COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

PREDICCIÓN DE TARGETS DE M&A MEDIANTE MACHINE LEARNING: DESARROLLO DE UNA HERRAMIENTA DE SCREENING BASADA EN DATOS FINANCIEROS PÚBLICOS

Autor: Adriana Zugasti Álvarez
Director: David Martín-Corral

Madrid
Junio 2026

Declaración de originalidad

Declaro bajo mi responsabilidad que el Proyecto presentado con el título **Predicción de Targets de M&A mediante Machine Learning: Desarrollo de una Herramienta de Screening basada en Datos Financieros Públicos** e la ETS de Ingeniería – ICAI de la Universidad Pontificia Comillas en el curso académico 2025-2026 es de mi autoría y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Uso de Inteligencia Artificial[1]

Declaro bajo mi responsabilidad que (indicar la opción correcta):

- ☐ No he utilizado Inteligencia Artificial en la elaboración del presente documento.
- ☒ He utilizado Inteligencia Artificial en la elaboración del presente documento y/o del Anexo B siempre en las condiciones permitidas por la Universidad Pontificia Comillas, es decir, aplicando el Nivel 2 de la Escala de Evaluación de Perkins et al. (2024): "*La IA puede utilizarse para actividades previas a la tarea, como la lluvia de ideas, la descripción y la investigación inicial. Este nivel se centra en el uso de la IA para la planificación, las síntesis y la generación de ideas, pero las evaluaciones deben hacer hincapié en la capacidad de desarrollar y refinar estas ideas de forma independiente*". En concreto, las Inteligencia Artificial ha sido empleada para:

La Inteligencia Artificial ha sido empleada como apoyo en la organización y estructuración del Trabajo Fin de Grado, facilitando la ordenación de ideas, la planificación de apartados y la revisión de la redacción, sin alterar el contenido técnico, los resultados ni las conclusiones desarrolladas por la autora.

También se ha utilizado para la traducción y adaptación de apartados al inglés, así como como apoyo auxiliar en la generación y depuración de código para tareas concretas de integración de datos, especialmente en el proceso de fuzzy matching entre PitchBook y Compustat. La validación del dataset, el análisis de resultados y las decisiones metodológicas fueron realizadas por la autora.

Firmado (alumno): Adriana Zugasti Álvarez

Fecha: 01/07/2026

[1] Esta declaración se refiere al uso de la Inteligencia Artificial generativa para realizar los documentos del Proyecto (Anexo B y Memoria). No aplica a Proyectos donde, por su naturaleza, deban emplear inteligencia artificial como parte de los mismos (aplicación de técnicas de aprendizaje automático, redes neuronales, análisis de datos…)

Autorización para la entrega del Proyecto

El Director del Proyecto	El co-Director del Proyecto (si aplica)
MARTIN CORRAL CALVO DAVID - 03909768K Digitally signed by MARTIN CORRAL CALVO DAVID - 03909768K Date: 2026.07.01 14:45:01 +02'00'	
Fdo:	Fdo:
Fecha: 01/07/2026	Fecha:

COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

GRADO EN INGENIERÍA EN TECNOLOGÍAS DE TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

PREDICCIÓN DE TARGETS DE M&A MEDIANTE MACHINE LEARNING: DESARROLLO DE UNA HERRAMIENTA DE SCREENING BASADA EN DATOS FINANCIEROS PÚBLICOS

Autor: Adriana Zugasti Álvarez
Director: David Martín-Corral

Madrid
Junio 2026

PREDICCIÓN DE TARGETS DE M&A MEDIANTE MACHINE LEARNING

Autor: Zugasti Álvarez, Adriana
Director: Martín-Corral, David
Entidad Colaboradora: ICAI - Universidad Pontificia Comillas

RESUMEN DEL PROYECTO

La identificación de targets es una de las fases más intensivas del proceso de M&A. Antes de analizar una operación en detalle, los equipos revisan un universo muy amplio de compañías, filtran sectores, estudian ratios financieros y comparan posibles oportunidades. Este trabajo parte de esa necesidad: convertir una parte del screening inicial en un proceso más sistemático, reproducible y escalable.

El foco se sitúa en empresas cotizadas de Estados Unidos porque existe una mayor disponibilidad de información financiera estandarizada y una base histórica amplia de operaciones. La base de datos combina operaciones completadas de PitchBook con estados financieros de Compustat. Con esa información se construye una etiqueta binaria: target cuando una empresa fue adquirida y no target cuando pertenece al grupo de control. A partir de esto, se construye un pipeline completo de machine learning que combina la ingeniería de features, tratamiento del desbalanceo de clases y comparativa de doce modelos de machine learning, culminando en un dashboard web interactivo para la exploración de resultados con informes automatizados.

Palabras clave: M&A, machine learning, screening, Random Forest, SMOTE, XGBoost, Pitchbook, Compustat, clasificación binaria

1. Introducción

El M&A constituye una vía clave de crecimiento corporativo, ya que permite a las empresas entrar en nuevos mercados, acceder a capacidades estratégicas, ganar escala o reforzar su posición competitiva. Sin embargo, la fase inicial de identificación de targets suele seguir siendo intensiva en trabajo manual. Los analistas revisan bases de datos, ratios financieros, comparables, sectores y noticias para construir una lista preliminar de posibles candidatos. Este proceso aporta criterio cualitativo, pero también introduce subjetividad y limita la escala del análisis cuando el universo inicial incluye miles de compañías.

En este contexto, el Machine Learning puede aportar valor como primera capa de priorización. Las empresas adquiridas no comparten necesariamente una única característica financiera, sino combinaciones de tamaño, liquidez, rentabilidad, deuda, valoración, crecimiento o sector. Por ello, el problema se formula como una clasificación binaria: a partir de información financiera previa al evento, el modelo estima la probabilidad de que una empresa pertenezca a la clase target. Esta probabilidad se utiliza como un score de adquisición, útil para generar rankings y orientar el análisis posterior.

2. Definición del proyecto

El objetivo principal del TFG es desarrollar una herramienta end-to-end de screening de targets de M&A. Para ello, el trabajo cubre cuatro etapas: construcción del dataset, limpieza y transformación de datos, entrenamiento y comparación de modelos, y desarrollo de un dashboard web que permita explotar los resultados. La muestra parte de operaciones completadas de PitchBook entre 2005 y 2020, con targets estadounidenses cotizados y tamaño mínimo de operación de 100 millones de dólares. Posteriormente, estas compañías se cruzan con datos financieros de Compustat mediante un proceso de integración y fuzzy matching.

Tras la integración inicial, el dataset contiene 5.071 observaciones y 51 columnas, incluyendo 1.589 empresas target. Durante la fase de limpieza se eliminan columnas que podrían generar data leakage, se tratan valores nulos, se revisan outliers financieros y se eliminan variables redundantes por correlación o multicolinealidad. El dataset final queda formado por 4.940 observaciones, de las cuales 1.552 son targets y 3.388 no targets, y 26 variables predictoras. La partición utilizada es 70/30, con 3.458 observaciones para entrenamiento y 1.482 para test.

3. Descripción del modelo/ sistema/ herramienta

La arquitectura del trabajo se plantea como un pipeline completo de datos, modelo y visualización. Todos los archivos que componen el pipeline de la Figura I se pueden encontrar en el repositorio de github.[2]

Primero se extraen y combinan las fuentes de información. Después se calculan ratios financieros como deuda total, margen EBITDA, ROA, leverage, capex intensity, market capitalization, EV/EBITDA, asset turnover o cash ratio. A continuación se entrena una serie de modelos supervisados y, finalmente, el modelo seleccionado se integra en un dashboard web desarrollado con Flask, HTML y JavaScript.

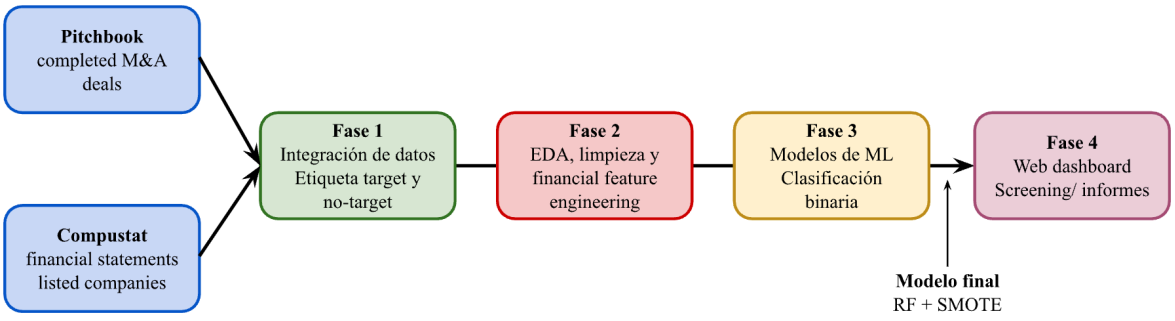


Figura I. Arquitectura general del pipeline end-to-end desarrollado en el TFG.
Fuente: elaboración propia
**** El modelo no sustituye el análisis del analista: prioriza compañías para reducir el universo inicial de screening.***

Se entrenan distintas familias de modelos: regresión logística como benchmark lineal, Random Forest, Gradient Boosting, XGBoost y LightGBM como modelos no lineales basados en árboles. Además, se

[2] Los csv originales descargados de Pitchbook y Compustat no han sido subidos por razones legales, pero el dataset final compuesto para el modelaje sí.

prueban variantes con GridSearchCV, class_weight balanced, regularización y SMOTE. Esta última técnica se aplica únicamente sobre el conjunto de entrenamiento para evitar data leakage. La evaluación se realiza sobre el test original, prestando especial atención al AUC-ROC, a los targets identificados, a la accuracy y al nivel de overfitting.

El dashboard transforma el modelo en una herramienta práctica. Incluye una página de Executive Summary, un M&A Screener para filtrar y ordenar empresas, una sección de Company Deep Dive para analizar una compañía concreta, y un simulador que permite introducir datos financieros de una empresa hipotética y obtener su score estimado. De esta forma, el resultado no queda limitado a un modelo entrenado en Python, sino que se convierte en una aplicación utilizable para apoyar la toma de decisiones inicial.

4. Resultados

Los resultados muestran que los modelos lineales no capturan suficiente señal para este problema. La regresión logística obtiene un AUC-ROC de 0,5602 y solo identifica 11 targets en el conjunto de test, por lo que no resulta adecuada para screening. En cambio, los modelos basados en árboles mejoran claramente la capacidad de separación entre targets y no targets, con AUC superiores a 0,70.

El mejor equilibrio global lo ofrece Random Forest con SMOTE. Este modelo alcanza un AUC-ROC de 0,7260, una accuracy en test del 71,39 % e identifica 130 targets, manteniendo un overfitting moderado. Gradient Boosting con SMOTE queda muy cerca, con AUC-ROC de 0,7253 y 142 targets identificados. XGBoost y LightGBM detectan más targets en algunas variantes, pero presentan menor AUC o mayor número de falsos positivos. En particular, LightGBM balanced identifica 297 targets, pero genera 324 falsos positivos, lo que reduciría la utilidad práctica del ranking al introducir demasiado ruido para el analista.

Modelo	Accuracy train	Accuracy test	AUC-ROC	Targets identificados	Overfitting
Regresión logística	68,42 %	68,29 %	0,5602	11	No
Random Forest SMOTE	86,73 %	71,39 %	0,7260	130	Moderado
Gradient Boosting SMOTE	89,54 %	71,79 %	0,7253	142	Moderado
XGBoost SMOTE	98,12 %	70,24 %	0,7166	167	Alto
LightGBM SMOTE	87,55%	70,72%	0,7198	166	Moderado

Tabla I. Comparación simplificada de resultados de los principales modelos en test.

Por tanto, los resultados deben interpretarse desde la lógica de una herramienta de screening. Un mayor número de targets detectados no implica necesariamente un mejor modelo si viene acompañado de demasiados falsos positivos. El modelo seleccionado adopta un enfoque más conservador: prioriza empresas con una señal financiera más clara, aunque no detecte todos los targets reales. En el dashboard, el score generado debe entenderse como una medida relativa de similitud con targets históricos, no como

una probabilidad exacta de adquisición. Su utilidad está en ordenar el universo inicial y ayudar al analista a decidir qué compañías revisar primero, complementando después el resultado con análisis financiero, estratégico y cualitativo.

5. Conclusiones

La principal conclusión del proyecto es que los datos financieros públicos sí contienen señal útil para apoyar la fase inicial de target screening, aunque no permiten predecir una adquisición con certeza. El modelo final no debe interpretarse como una decisión automática de inversión, sino como una herramienta de priorización. Su aportación está en reducir el universo inicial, hacer el proceso más trazable y ayudar al analista a empezar desde una lista ordenada por probabilidad estimada de adquisición.

Los objetivos planteados se han cubierto: se ha construido un dataset combinando PitchBook y Compustat, se ha aplicado un proceso completo de limpieza y transformación, se han entrenado y comparado múltiples modelos de clasificación, se ha seleccionado Random Forest con SMOTE como mejor alternativa y se ha desarrollado un dashboard interactivo para explotar los resultados. Las aportaciones principales son la creación de un pipeline reproducible de datos financieros y M&A, la comparación práctica de técnicas de modelado sobre un problema desbalanceado, y la integración del modelo en una herramienta visual orientada al uso real en screening.

El proyecto también presenta limitaciones. La herramienta utiliza únicamente empresas cotizadas estadounidenses y variables financieras públicas, por lo que deja fuera factores privados y estratégicos muy relevantes en una adquisición, como conversaciones entre comprador y vendedor, sinergias, intención del equipo directivo, presión de accionistas o contexto competitivo. Esto podría ser algo interesante a mirar como trabajo futuro.

código github demo dashboard

6. Referecias

Bain & Company. (2024). Global M&A deal value on track to reach $3.5 trillion in 2024.

Beckenstrater, G. (2024). Predicting mergers and acquisitions using machine learning. University of Cape Town.

Breiman, L. (2001). Random Forests. Machine Learning, 45, 5-32.

Campbell, J. L., Elfrink, E., Irons, C. y Moon, J. (2024). What is the Deal?: Predicting M&A Outcomes with Machine Learning.

Chawla, N. V., Bowyer, K. W., Hall, L. O. y Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

James, G., Witten, D., Hastie, T. y Tibshirani, R. (2021). An Introduction to Statistical Learning. Springer.

PitchBook. Mergers & Acquisitions Data; WRDS. Compustat North America - Fundamentals Annual.

PREDICTING M&A TARGETS THROUGH MACHINE LEARNING

Autor: Zugasti Álvarez, Adriana
Supervisor: Martín-Corral, David
Collaborating Entity: ICAI - Universidad Pontificia Comillas

PROJECT SUMMARY

Target identification is one of the most relevant and least scalable stages of the mergers and acquisitions process. Before valuing a company, negotiating a transaction or analysing synergies, M&A teams must review a broad universe of companies and decide which candidates deserve deeper analysis. This Final Degree Project addresses that practical need by developing a Machine Learning-based screening tool using public financial data. The objective is not to replace the analyst or to predict acquisitions with certainty, but to make the first screening stage more systematic, reproducible and efficient.

The project focuses on listed companies in the United States because this market offers broad access to standardised financial information and a large historical base of M&A transactions. The dataset combines completed transactions from PitchBook with annual financial statements from Compustat, accessed through WRDS. From this integration, a binary target variable is built: companies acquired in the selected period are labelled as targets, while companies not acquired form the non-target control group.

Keywords: M&A, target screening, Machine Learning, binary classification, Random Forest, SMOTE, PitchBook, Compustat, dashboard.

1. Introduction

M&A is a key corporate growth mechanism, allowing firms to enter new markets, acquire strategic capabilities, gain scale or strengthen their competitive position. However, the initial identification of acquisition targets is still highly manual. Analysts review databases, financial ratios, comparable transactions, sectors and market information to create a preliminary list of companies. This process benefits from professional judgement, but it is time-consuming, subjective and difficult to scale when the initial universe includes thousands of firms.

Machine Learning can add value as a first prioritisation layer. Acquired companies are unlikely to share a single financial characteristic; instead, acquisition likelihood depends on combinations of size, liquidity, profitability, leverage, valuation, growth and sector information. For this reason, the project is formulated as a binary classification problem. Using financial information available before the event, the model estimates the probability that each company belongs to the target class. This probability becomes an acquisition likelihood score that can be used to rank firms for further review.

2. Project definition

The main objective of the project is to develop an end-to-end M&A target screening tool. The work is structured into four stages: dataset construction, data cleaning and transformation, model training and comparison, and the development of a web dashboard to use the results. The initial M&A sample is extracted from PitchBook and includes completed transactions between 2005 and 2020, US-listed targets and a minimum deal size of 100 million dollars. These companies are then matched with Compustat financial data through an integration process that includes fuzzy matching.

After the first integration, the dataset contains 5,071 observations and 51 columns, including 1,589 target companies. During the cleaning stage, columns that could cause data leakage are removed, missing values are treated, financial outliers are reviewed and redundant variables are eliminated based on correlation and multicollinearity. The final modelling dataset contains 4,940 observations, 1,552 targets, 3,388 non-targets and 26 predictive variables. A 70/30 train-test split is used, with 3,458 observations for training and 1,482 for final testing.

3. Model & tool description

The project architecture follows a full data-model-visualisation pipeline. First, the two main sources are extracted and combined. Then, financial features are calculated, including total debt, EBITDA margin, ROA, leverage, capex intensity, market capitalisation, EV/EBITDA, asset turnover and cash ratio. Next, several supervised models are trained and compared. Finally, the selected model is integrated into a web dashboard developed with Flask, HTML and JavaScript. All the files that compose the pipeline described in Figure II can be found in github.[3]

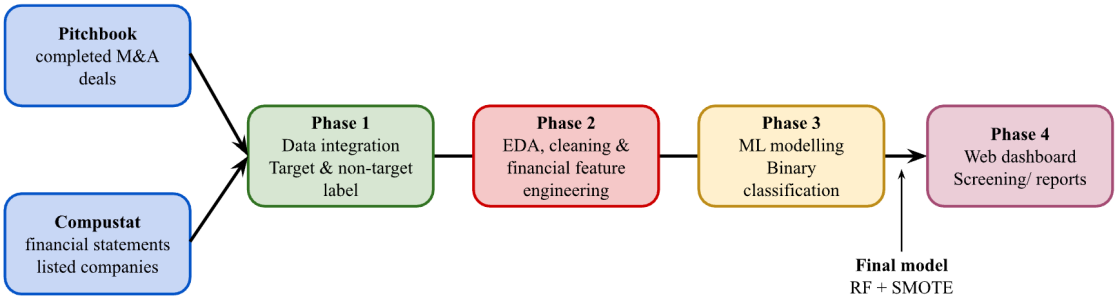


Figure II. General architecture of the pipeline developed in the project.
Source: self-made

**** The model supports the analyst by ranking companies; it does not replace strategic judgement or due diligence.***

The modelling stage compares logistic regression as a linear benchmark with non-linear tree-based models: Random Forest, Gradient Boosting, XGBoost and LightGBM. Several variations are tested, including GridSearchCV hyperparameter tuning, class_weight balanced, manual regularisation and

[3] The original cvs files downloaded from Pitchbook and Compustat are not uploaded for legal reasons, but the built final dataset with both cross-matched is.

SMOTE. SMOTE is applied only to the training set in order to avoid data leakage, while the final evaluation is always performed on the original test set. The comparison considers AUC-ROC, accuracy, identified targets and overfitting, because accuracy alone is not sufficient in an imbalanced screening problem.

The dashboard turns the model into a practical tool. It includes an Executive Summary, an M&A Screener to filter and rank companies, a Company Deep Dive page to analyse one firm in detail, and a simulator where the user can enter the financial data of a hypothetical company and obtain its estimated acquisition score. Therefore, the project does not end with a Python model, but translates the output into an interface that can support an analyst during the initial screening process.

4. Results

The results show that the linear model is not sufficient for this problem. Logistic regression obtains an AUC-ROC of 0.5602 and identifies only 11 targets in the test set, so it is not suitable as a screening tool. In contrast, tree-based models improve the separation between targets and non-targets, with AUC values above 0.70.

The best overall balance is obtained by Random Forest with SMOTE. This model reaches an AUC-ROC of 0.7260, a test accuracy of 71.39% and correctly identifies 130 targets, while keeping overfitting at a moderate level. Gradient Boosting with SMOTE is very close, with an AUC-ROC of 0.7253 and 142 identified targets. XGBoost and LightGBM identify more targets in some versions, but they either produce lower AUC values or introduce more false positives. In particular, LightGBM balanced identifies 297 targets, but also generates 324 false positives, which would reduce the practical usefulness of the ranking by adding too much noise for the analyst.

Model	Accuracy train	Accuracy test	AUC-ROC	Targets identified	Overfitting
Regresión logística	68,42 %	68,29 %	0,5602	11	No
Random Forest SMOTE	86,73 %	71,39 %	0,7260	130	Moderate
Gradient Boosting SMOTE	89,54 %	71,79 %	0,7253	142	Moderate
XGBoost SMOTE	98,12 %	70,24 %	0,7166	167	High
LightGBM SMOTE	87,55%	70,72%	0,7198	166	Moderate

Table II. Simplified comparison of the main models on the test set.

Therefore, the results should be interpreted from the perspective of a screening tool. A higher number of identified targets does not necessarily imply a better model if it also leads to too many false positives. The selected model follows a more conservative approach: it prioritises companies with a clearer financial signal, even if it does not capture all real targets. In the dashboard, the generated score should be understood as a relative measure of similarity to historical targets, rather than an exact probability of

acquisition. Its value lies in ranking the initial universe of companies and helping the analyst decide which firms to review first, before complementing the output with financial, strategic and qualitative analysis.

5. Conclusion

The main conclusion is that public financial data contain useful signal for supporting the first stage of target screening, although they are not enough to predict acquisitions with certainty. The final model should be interpreted as a prioritisation tool rather than an automatic investment decision. Its contribution lies in reducing the initial universe, improving traceability and helping the analyst start from a ranked list of companies based on estimated acquisition likelihood.

The project objectives have been met. A dataset combining PitchBook and Compustat has been built, a complete cleaning and transformation process has been applied, multiple classification models have been trained and compared, Random Forest with SMOTE has been selected as the final model, and an interactive dashboard has been developed to use the results. The main contributions are the creation of a reproducible M&A and financial data pipeline, the practical comparison of modelling techniques in an imbalanced classification problem, and the integration of the model into a visual tool designed for real screening use.

The main limitations come from the scope of the data. The tool uses only US-listed companies and public financial variables, so it does not capture private or strategic factors that are central to an acquisition, such as buyer interest, synergies, management intentions, shareholder pressure or negotiations. This a future project could look into.

código github demo dashboard

6. References

Bain & Company. (2024). Global M&A deal value on track to reach $3.5 trillion in 2024.

Beckenstrater, G. (2024). Predicting mergers and acquisitions using machine learning. University of Cape Town.

Breiman, L. (2001). Random Forests. Machine Learning, 45, 5-32.

Campbell, J. L., Elfrink, E., Irons, C. y Moon, J. (2024). What is the Deal?: Predicting M&A Outcomes with Machine Learning.

Chawla, N. V., Bowyer, K. W., Hall, L. O. y Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861-874.

James, G., Witten, D., Hastie, T. y Tibshirani, R. (2021). An Introduction to Statistical Learning. Springer.

PitchBook. Mergers & Acquisitions Data; WRDS. Compustat North America - Fundamentals Annual.

Índice de la memoria

Índice de figuras

Índice de tablas

Capítulo 1. INTRODUCCIÓN

1.1 CONTEXTO

Las fusiones y adquisiciones, conocidas como M&A (Mergers & Acquisitions) por sus siglas en inglés, son una de las formas más rápidas que tienen las empresas para crecer. Una compañía puede comprar otra para entrar en un nuevo mercado, acceder a tecnología, ganar cuota, reforzar su posición frente a competidores o incorporar capacidades que tardaría años en desarrollar internamente. Por eso, el M&A no se puede entender solo como una operación financiera, sino también como una decisión estratégica.

El peso económico de esta actividad es muy alto. Bain & Company estimó que el valor global de las operaciones de M&A alcanzaría aproximadamente 3,5 billones de dólares en 2024, un 15 % más que el año anterior, y que en los años siguientes seguiría subiendo, situando el volumen global en torno a los 4 billones de dólares en el entorno de 2026 (Bain & Company, 2024). Este dato muestra que, aunque el mercado se había ralentizado en los años anteriores por la subida de tipos de interés, el encarecimiento de la financiación y la incertidumbre macroeconómica, las operaciones corporativas siguen siendo una herramienta fundamental para las empresas.

Además, el M&A forma parte de una industria mucho más amplia: la banca de inversión. The Business Research Company (2026) define la banca de inversión como el conjunto de servicios financieros que incluye captación de capital, underwriting y asesoramiento en fusiones y adquisiciones. Dentro de este mercado, el segmento de asesoramiento en M&A fue el mayor en 2025, con un valor aproximado de 49.570 millones de dólares y una cuota del 33,35 % del total del mercado global de banca de inversión (The Business Research Company, 2026). Por lo tanto, la identificación, análisis y ejecución de operaciones de adquisición no es una actividad secundaria. Es una de las áreas principales de los servicios financieros.

En este contexto, uno de los pasos más importantes es identificar qué empresas pueden ser adquiridas. Antes de valorar una compañía, negociar un precio o estudiar sinergias, hay que decidir qué compañías merece la pena analizar. Esta primera fase se conoce como screening de targets. Aunque pueda parecer una fase preliminar, condiciona todo el proceso posterior. Si el equipo parte de una lista

mal construida, puede dedicar tiempo a empresas que no tienen sentido estratégico o, peor aún, dejar fuera oportunidades relevantes.

El problema es que este screening suele ser muy manual. Los equipos de M&A, corporate development, private equity o banca de inversión revisan bases de datos, informes sectoriales, estados financieros, múltiplos, noticias y comparables. Después, con esa información, preparan una lista de posibles candidatos. Este proceso requiere criterio financiero y conocimiento del sector, pero también consume mucho tiempo. En un mercado grande, un analista puede revisar decenas de empresas con detalle, pero no miles de forma eficiente. En este sentido, la literatura sobre M&A destaca que la fase de identificación de targets es una de las más intensivas en recursos dentro del proceso, ya que implica filtrar un universo amplio de compañías hasta llegar a un conjunto reducido que encaje estratégicamente con el comprador (DePamphilis, 2019).

A esto se suma que el análisis inicial puede ser subjetivo. Dos analistas pueden ver los mismos datos y priorizar empresas distintas. Esto no significa que el criterio humano sea un problema. Al contrario, en M&A sigue siendo imprescindible. Sin embargo, sí muestra que puede haber valor en contar con una herramienta que ordene el universo inicial de empresas de una forma más sistemática. Tal y como señalan Rosenbaum y Pearl (2020), la selección de comparables y targets potenciales depende en gran medida del juicio del analista, lo que introduce variabilidad en los resultados. Un modelo no sustituye la decisión final, pero puede servir como primer filtro.

Este proyecto parte de esa idea. El objetivo es desarrollar una herramienta de screening de targets de M&A utilizando Machine Learning y datos financieros públicos. La herramienta busca estimar la probabilidad de que una empresa cotizada estadounidense sea adquirida, a partir de variables procedentes de sus estados financieros. Después, esa probabilidad se utiliza para generar un ranking de compañías y facilitar el análisis inicial.

La propuesta tiene sentido porque los estados financieros de una empresa contienen información relevante sobre su situación. Variables como tamaño, rentabilidad, liquidez, endeudamiento, crecimiento, valoración o eficiencia operativa pueden reflejar características que hacen que una empresa sea más o menos atractiva como target. Una empresa adquirida no tiene por qué cumplir una única condición. Lo más probable es que el patrón esté en la combinación de varias variables. Ahí es donde el Machine Learning puede aportar valor.

Este trabajo no pretende predecir de forma perfecta qué empresas van a ser adquiridas. Eso no sería realista. Una operación de M&A depende de muchos factores que no aparecen en los estados financieros: conversaciones privadas, estrategia del comprador, intención del equipo directivo, presión de accionistas, regulación, timing de mercado o situación competitiva. Sin embargo, que no se pueda observar todo no significa que no se pueda extraer señal de los datos disponibles. La utilidad del modelo está en reducir el universo inicial y ayudar al analista a empezar desde una base mejor ordenada.

Por lo tanto, el proyecto se plantea como un pipeline completo: datos, modelo y dashboard. Primero se construye un dataset combinando operaciones de M&A con información financiera pública. Después se entrenan distintos modelos de clasificación. Por último, se diseña una aplicación que permite utilizar el modelo como herramienta de screening. Esta última parte es importante porque un modelo aislado tiene valor limitado si sus resultados no se pueden consultar de forma clara.

1.2 EVOLUCIÓN DEL MERCADO DE M&A

La actividad de M&A cambia mucho según el ciclo económico. Cuando hay liquidez, intereses bajos y confianza en el crecimiento, las empresas tienen más incentivos para comprar. Cuando suben los intereses, se encarece la deuda o aumenta la incertidumbre, muchas operaciones se retrasan o se cancelan. Aun así, las adquisiciones siguen siendo una vía muy utilizada para crecer, especialmente cuando el crecimiento orgánico es lento o insuficiente.

En los últimos años, el mercado ha estado marcado por un entorno más complejo. Después de la pandemia, la inflación, los tipos de interés altos y la incertidumbre geopolítica dificultaron muchas operaciones. Según el análisis de SMU (2023), el periodo post pandemia obligó a las empresas a replantear sus estrategias de M&A, ya que el encarecimiento de la financiación y la volatilidad de los mercados habían reducido el número de operaciones y aumentado la cautela de los inversores. Además, factores como las disrupciones en las cadenas de suministro, los cambios en los hábitos de consumo y una mayor presión regulatoria han añadido nuevas capas de complejidad al proceso de adquisición. La recuperación progresiva del mercado ha impulsado operaciones estratégicas, especialmente en sectores como tecnología, salud y energía, donde la pandemia aceleró tendencias estructurales como la digitalización o la transición energética (Bain & Company, 2024).

En paralelo, el uso de tecnología ha ganado peso dentro del proceso de M&A. La inteligencia artificial y las herramientas analíticas están permitiendo identificar oportunidades de adquisición de forma más eficiente y agilizar tareas como la due diligence, la valoración o la preparación de comités de inversión, aunque el juicio humano sigue siendo clave (Levy, 2026). Esta tendencia se enmarca en una evolución más amplia del sector financiero, donde la adopción de analítica avanzada está creciendo de forma sostenida. De hecho, se espera que el mercado global de banca de inversión aumente significativamente en los próximos años, impulsado en parte por el uso de estas tecnologías (The Business Research Company, 2026).

En este contexto, Estados Unidos tiene un papel especialmente relevante. Es uno de los principales centros financieros del mundo, cuenta con mercados de capitales muy desarrollados y concentra un gran número de empresas cotizadas. Además, ofrece una ventaja clave para este proyecto: la disponibilidad de información financiera pública, estandarizada y accesible. Esto permite comparar compañías de forma homogénea y construir un dataset consistente.

Por este motivo, el proyecto se centra en empresas cotizadas estadounidenses. No se trata de que sea el único mercado relevante, sino de que ofrece mejores condiciones para entrenar un modelo de este tipo: mayor disponibilidad de datos, más operaciones históricas y mejor cobertura en bases de datos financieras.

A pesar de estos avances, el M&A sigue siendo una actividad compleja. Una adquisición implica valorar activos, negociar con accionistas, analizar riesgos, integrar equipos y justificar el precio pagado. Además, muchas operaciones fallan porque las sinergias esperadas no se materializan o porque el comprador paga demasiado. Las primas de adquisición reflejan estas expectativas de valor, pero también aumentan el riesgo si no se cumplen (Brixius et al., 2025).

Por ello, la fase de selección inicial es especialmente importante. Elegir mejor los targets no garantiza el éxito de una operación, pero sí mejora el punto de partida. Un proceso de screening más estructurado permite priorizar mejor las empresas y dedicar más tiempo al análisis estratégico, reduciendo ineficiencias desde las primeras etapas.

1.3 MOTIVACIÓN DEL PROYECTO

La motivación principal del proyecto surge de una pregunta práctica: si existen tantos datos financieros disponibles, ¿por qué no utilizarlos para hacer el screening inicial de targets de forma más eficiente?

En M&A, el proceso inicial de revisión de empresas es necesario, pero también muy manual y difícil de escalar cuando el universo es amplio. Aquí es donde un modelo puede aportar valor. A partir de datos financieros históricos, un clasificador puede identificar patrones asociados a empresas que han sido adquiridas y utilizarlos para estimar la probabilidad de adquisición de nuevas compañías.

El resultado no debe interpretarse como una predicción exacta, sino como una herramienta de apoyo que permite ordenar empresas y priorizar el análisis. De esta forma, el modelo ayuda a reducir el universo inicial, hacer el proceso más sistemático y liberar tiempo del analista para tareas que requieren juicio humano.

Por tanto, el objetivo es comprobar si los datos financieros públicos contienen suficiente señal como para mejorar el screening inicial. Aunque el modelo no sea perfecto, si consigue concentrar una parte relevante de los targets en un subconjunto más manejable, ya aporta valor práctico en el proceso de M&A.

1.4 OBJETIVOS DEL PROYECTO

El objetivo general de este Trabajo Fin de Grado es desarrollar una herramienta de screening de targets de fusiones y adquisiciones mediante Machine Learning, utilizando datos financieros públicos de empresas cotizadas estadounidenses.

Para alcanzar este objetivo general, se plantean los siguientes objetivos específicos:

- Construir un dataset de empresas estadounidenses cotizadas combinando información de operaciones de M&A con datos financieros públicos. Para ello se utilizan operaciones completadas procedentes de PitchBook y variables financieras extraídas de Compustat.
- Aplicar un proceso completo de limpieza y preparación de datos. Esto incluye revisar valores nulos, eliminar duplicados, tratar outliers, analizar correlaciones, transformar variables y preparar el dataset para el entrenamiento.

- Entrenar y comparar distintos modelos de clasificación supervisada. En concreto, se prueban modelos como regresión logística, Random Forest, Gradient Boosting y XGBoost, junto con técnicas de ajuste de hiperparámetros.
- Evaluar los modelos con métricas adecuadas al problema y compararlos más allá de los números. Esto permite entender mejor si el modelo es útil como herramienta de screening dentro del contexto de M&A.
- Seleccionar el modelo con mejor equilibrio entre rendimiento e interpretación. El objetivo no es elegir el algoritmo más complejo, sino el que mejor se adapte al problema y a su uso práctico.
- Diseñar una herramienta visual tipo dashboard. Esta aplicación debe permitir consultar empresas, generar un ranking de probabilidad de adquisición y analizar casos concretos de forma sencilla.

El resultado final es un pipeline end-to-end: datos, modelo y visualización. La aportación principal está en conectar técnicas de Machine Learning con una necesidad real del proceso de M&A. De esta forma, el proyecto ofrece una base práctica para hacer más eficiente la fase inicial de identificación de targets.

1.5 METODOLOGÍA Y PLANIFICACIÓN

La metodología del proyecto sigue un enfoque secuencial, desde la construcción del dataset hasta la visualización final de los resultados. Primero se define el alcance del análisis, centrado en empresas cotizadas estadounidenses y operaciones de M&A completadas dentro del periodo seleccionado. Después se combinan dos fuentes principales: PitchBook, utilizada para identificar las operaciones y construir la variable objetivo, y Compustat, utilizada para obtener las variables financieras de las empresas.

Una vez construido el dataset, se realiza una fase de análisis exploratorio, limpieza y transformación de datos. En esta etapa se revisan valores nulos, duplicados, outliers, distribución de variables, correlaciones y posibles problemas de calidad. También se preparan las variables para el entrenamiento de los modelos y se separa el dataset en conjuntos de entrenamiento y test.

La siguiente fase consiste en entrenar y comparar distintos modelos de clasificación supervisada. Se prueban modelos con distintos niveles de complejidad, desde regresión logística hasta modelos

basados en árboles como Random Forest, Gradient Boosting y XGBoost. También se tiene en cuenta el desbalanceo de clases, ya que las empresas adquiridas representan una proporción menor frente al total de compañías. La evaluación se realiza mediante métricas como AUC-ROC, recall, precisión y matriz de confusión.

Por último, el modelo seleccionado se integra en una herramienta visual tipo dashboard. El objetivo de esta aplicación es transformar el resultado del modelo en una herramienta práctica de screening, permitiendo consultar empresas, generar rankings de probabilidad de adquisición y facilitar el análisis inicial de posibles targets.

La planificación del trabajo se divide en seis fases principales mostradas en la Figura 1: definición del proyecto, captura e integración de datos, análisis exploratorio y limpieza, modelado, desarrollo del dashboard y cierre de la memoria. Esta planificación permitió organizar el proyecto de forma progresiva, dejando primero cerrada la parte de datos y modelos antes de pasar a la visualización final.

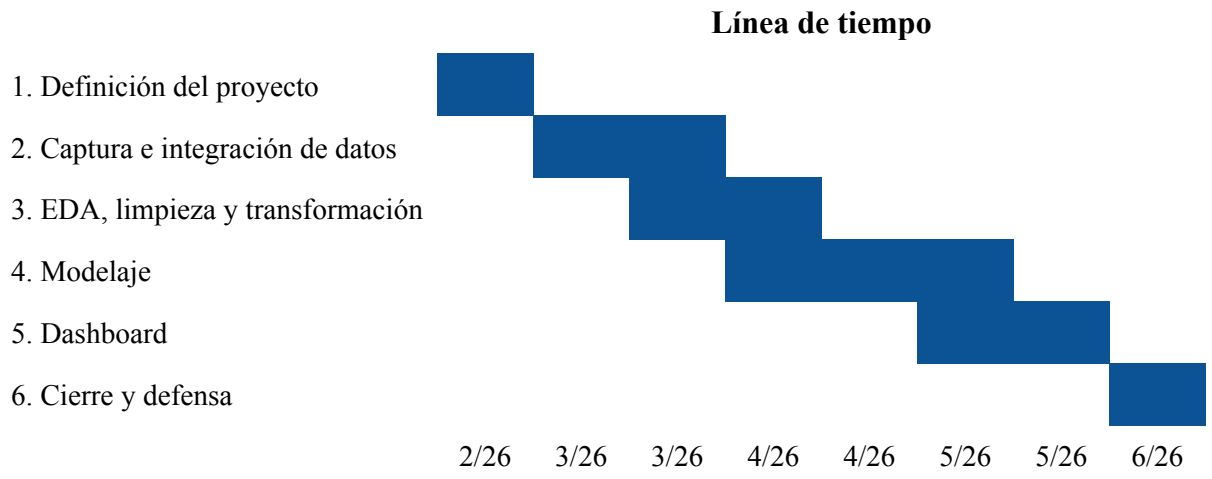


Figura 1. Cronología simplificada del proyecto

A partir del Capítulo 3 empieza la definición del proyecto.

Capítulo 2. ESTADO DE LA TÉCNICA

2.1 PROCESO DE M&A Y EVOLUCIÓN DE TARGET SCREENING

2.1.1 Concepto general de M&A

Las fusiones y adquisiciones, o M&A, agrupan operaciones en las que una empresa se combina con otra o adquiere el control total o parcial de ella. En una adquisición, la empresa compradora se denomina "acquirer" y la empresa adquirida se denomina "target". Aunque en la práctica los términos fusión y adquisición suelen aparecer juntos, no son exactamente lo mismo. En una fusión, dos empresas se integran para formar una entidad conjunta. En una adquisición, una empresa compra otra y pasa a controlar sus activos, operaciones o participación accionarial.

Las razones que llevan a una empresa a adquirir otra pueden ser muy distintas. Una adquisición puede buscar crecimiento, entrada en nuevos mercados, acceso a tecnología, reducción de costes, diversificación, aumento de cuota de mercado o creación de sinergias (Lee, 2005; Routhu et al., 2023). Por tanto, el M&A no se limita a una decisión financiera. También es una decisión estratégica que afecta a la posición futura de la empresa.

Además, las operaciones de M&A suelen tener un impacto relevante sobre el mercado. Los anuncios de adquisición pueden modificar la valoración de las empresas implicadas y transmitir información sobre expectativas de crecimiento, estrategia y creación de valor (Andrade et al., 2001; Campbell et al., 2025). Esto explica por qué el estudio de estas operaciones ha sido un tema recurrente en finanzas corporativas.

2.1.2 Fases principales de una adquisición

El proceso completo de una adquisición suele dividirse en varias fases. Primero, el comprador define su estrategia de crecimiento. Después, identifica empresas que podrían encajar con esa estrategia. Esta fase se conoce como target selection o target screening. Más adelante se realiza la due diligence, donde se analiza con mayor detalle la situación financiera, legal, operativa y estratégica del target. Si la operación avanza, se pasa a la valoración, negociación y cierre. Finalmente, una vez completada la

adquisición, llega la integración post-adquisición, donde se combinan equipos, procesos, sistemas y culturas empresariales (Jemison & Sitkin, 1986; Routhu et al., 2023).

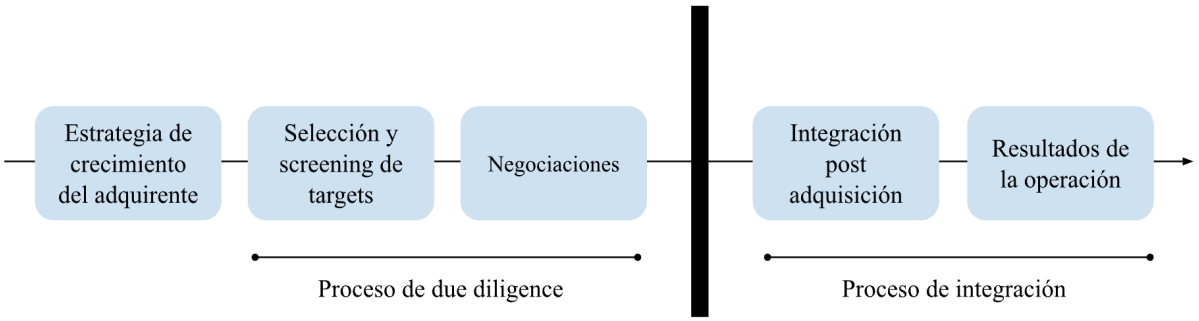


Figura 2. Proceso simplificado de una adquisición.

Fuente: elaboración propia, adaptado de Jemison y Stikin (1986) y Routhu et al. (2023).

Este TFG no estudia todo el proceso de M&A descrito en la Figura 2. El foco está en la segunda fase: la identificación de targets. Esta fase es importante porque condiciona todo lo que ocurre después. Si el screening inicial es poco preciso, el equipo puede perder tiempo analizando empresas que no encajan o puede dejar fuera oportunidades relevantes. Por eso, aunque sea una fase previa, tiene impacto directo en la eficiencia del proceso.

2.1.3 Target screening tradicional

Tradicionalmente, la identificación de targets se ha basado en una combinación de análisis financiero, conocimiento sectorial y criterio profesional. Los analistas revisan bases de datos, informes de mercado, estados financieros, múltiplos de valoración, operaciones comparables y noticias. Con esa información construyen una lista de empresas que podrían ser atractivas para un comprador. Después, esa lista se filtra hasta llegar a un conjunto reducido de compañías que merecen un análisis más profundo.

Este enfoque tiene ventajas. Permite incorporar conocimiento cualitativo, entender el contexto estratégico de la operación y valorar factores que no aparecen directamente en una base de datos. Por ejemplo, un analista puede detectar si una empresa encaja con una estrategia de expansión geográfica, si tiene una marca difícil de replicar o si existe una lógica industrial clara detrás de una posible adquisición. Estos elementos siguen siendo necesarios y no se pueden sustituir completamente con un modelo.

Sin embargo, el proceso tradicional también tiene limitaciones. Las más relevantes son la subjetividad, el alto consumo de tiempo y el uso limitado de datos (Lukander, 2025). La subjetividad aparece porque muchas decisiones dependen del juicio de expertos. Esto no es negativo en sí mismo, pero puede generar diferencias entre analistas. Dos personas pueden revisar el mismo universo de empresas y llegar a conclusiones distintas.

El consumo de tiempo también es un problema. Analizar targets con métodos tradicionales puede requerir muchas horas, especialmente si se quieren calcular ratios, comparar métricas, revisar información histórica y estudiar varias empresas al mismo tiempo (Lukander, 2025). Por último, el uso limitado de datos implica que muchas veces el análisis se centra en un número reducido de variables o en criterios cualitativos difíciles de sistematizar.

Esta limitación se vuelve más clara cuando el universo inicial es muy amplio. Un analista puede revisar con detalle diez, veinte o cincuenta empresas. Pero si el universo inicial tiene miles de compañías, el proceso manual deja de ser escalable. En ese contexto, una herramienta que ayude a ordenar empresas de forma automática puede aportar valor. No para decidir qué empresa comprar, sino para priorizar qué empresas analizar primero.

2.1.4 Primeros trabajos sobre predicción de targets

La idea de predecir qué empresas pueden convertirse en targets no es nueva. Uno de los trabajos clásicos en este campo es el de Palepu (1986), que plantea la predicción de targets como un problema empírico basado en características observables de las empresas. Este enfoque ya partía de una idea parecida a la de este proyecto: ciertas variables financieras pueden estar relacionadas con la probabilidad de que una empresa sea adquirida.

Los primeros modelos utilizaban técnicas estadísticas tradicionales, como regresión logística o análisis discriminante. Estos modelos eran útiles porque permitían trabajar con variables financieras e interpretar el signo y peso de cada factor. Sin embargo, tenían una limitación importante: normalmente asumían relaciones lineales entre variables y resultado. En un problema como M&A, esto puede ser demasiado simple. La probabilidad de adquisición no depende de una sola métrica financiera, sino de la combinación de tamaño, rentabilidad, liquidez, endeudamiento, crecimiento, sector y contexto de mercado.

Con el tiempo, han aparecido enfoques más avanzados basados en Machine Learning. Estos modelos permiten capturar relaciones no lineales y combinaciones de variables que no son tan fáciles de detectar con técnicas tradicionales. En la predicción de M&A, varios trabajos recientes han probado modelos como Random Forest, Gradient Boosting, LightGBM, redes neuronales o modelos ensemble (Beckenstrater, 2024; Campbell et al., 2025; Lukander, 2025).

2.1.5 Machine Learning aplicado al screening de M&A

El Machine Learning encaja bien en este problema porque la identificación de targets no depende de una única variable. Una empresa no se convierte en target solo por tener baja deuda, alta rentabilidad o mucho crecimiento. Lo más probable es que el patrón esté en la combinación de varias características. Algunas son financieras, como tamaño, liquidez, rentabilidad, endeudamiento o crecimiento. Otras pueden estar relacionadas con el sector, el contexto macroeconómico o la propia dinámica del mercado de M&A.

Los modelos supervisados permiten aprender a partir de ejemplos históricos. En este caso, el modelo observa empresas que fueron adquiridas y empresas que no lo fueron. A partir de sus variables financieras, intenta identificar patrones asociados a la clase target. Después, aplica esos patrones a nuevas empresas para estimar su probabilidad de adquisición.

Este enfoque no significa que el modelo sepa por qué una empresa va a ser comprada. Tampoco implica que pueda observar todos los factores que influyen en una operación. Muchas variables importantes son privadas: conversaciones entre comprador y vendedor, intenciones estratégicas, presión de accionistas, negociaciones, sinergias internas o decisiones del equipo directivo. Por eso, el objetivo no es predecir una adquisición con certeza. El objetivo es generar un ranking útil para el screening inicial.

Varios trabajos recientes muestran que los modelos no lineales pueden aportar valor en este contexto. En un estudio sobre resultados de operaciones de M&A, los modelos de Random Forest y Gradient Boosting ofrecieron mejores resultados que enfoques lineales para predecir retornos posteriores a la adquisición (Campbell et al., 2025). En otra investigación centrada en predecir M&A, los modelos avanzados superaron a la regresión logística, y LightGBM y los modelos ensemble obtuvieron el mejor rendimiento (Beckenstrater, 2024). Estos resultados apoyan la idea de que los patrones de M&A pueden no ser completamente lineales.

2.1.6 Automatización y uso de IA en deal sourcing

La aplicación de IA al M&A no se limita a trabajos académicos. En la práctica, el deal sourcing también está cambiando. Cada vez más equipos utilizan herramientas que combinan bases de datos, señales de mercado, modelos predictivos y dashboards para identificar oportunidades. El objetivo es ampliar la parte superior del funnel y detectar empresas que podrían pasar desapercibidas en un proceso manual.

Algunas plataformas analizan información pública y privada, actividad digital, crecimiento de empleados, financiación, tráfico web, noticias, sentimiento de mercado o señales de contratación. Con estos datos generan rankings o alertas de empresas que podrían tener interés estratégico. Este tipo de herramientas no sustituyen el criterio del analista, pero sí permiten revisar más compañías y reducir tareas repetitivas (Balan et al., 2025; Deloitte, 2025; McKinsey & Company, 2026).

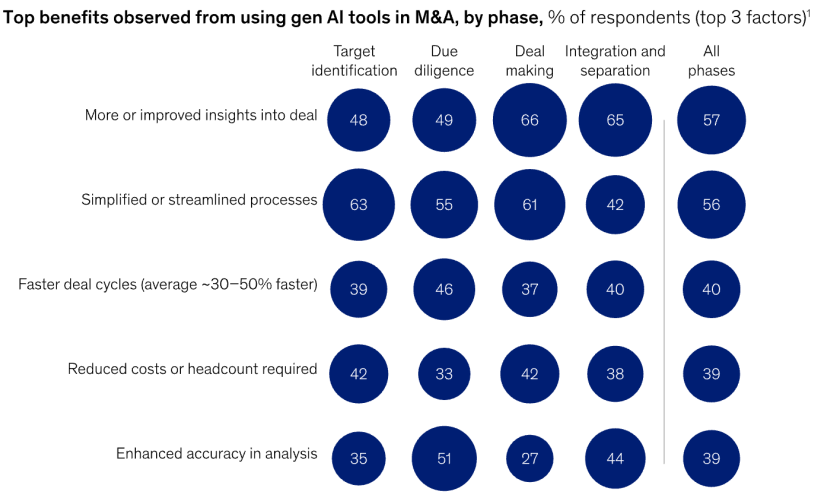


Figura 3. Principales beneficios observados al utilizar herramientas de IA generativa en M&A.

Fuente: (McKinsey & Company, 2026)

La Figura 3 muestra los principales beneficios observados por profesionales de M&A que ya utilizan herramientas de IA generativa. En la fase de identificación de targets, el beneficio más mencionado es la simplificación de procesos, con un 63 % de respuestas. También destacan la mejora en los insights sobre la operación, con un 48 %, la reducción de costes o necesidad de personal, con un 42 %, y la aceleración de los ciclos de deal, con un 39 % (McKinsey & Company, 2026).

Estos datos son relevantes para este TFG porque refuerzan la idea de que la identificación de targets es una fase donde la IA puede aportar valor de forma directa. En un proceso tradicional, el screening depende mucho de revisión manual, experiencia previa y criterios difíciles de sistematizar. Con herramientas basadas en datos, el analista puede partir de una lista más ordenada y dedicar más tiempo a interpretar los resultados, estudiar sinergias y valorar el encaje estratégico.

Este proyecto sigue esa misma lógica, pero con un alcance más concreto. No se busca automatizar una operación de M&A completa, sino construir una herramienta de screening basada en datos financieros públicos. El modelo estima una probabilidad de adquisición y genera un ranking de empresas, que sirve como primer filtro para apoyar el análisis inicial.

2.1.7 Limitaciones del enfoque basado en datos

El uso de Machine Learning en M&A también tiene limitaciones. La primera es la calidad y disponibilidad de los datos. Una gran parte de las operaciones de M&A involucra empresas privadas, que no están obligadas a publicar información financiera detallada. Esto implica que variables clave como ingresos, márgenes, deuda o crecimiento no siempre están disponibles o son comparables. Además, incluso en bases de datos profesionales, la cobertura de empresas privadas es incompleta y heterogénea. Por ejemplo, estudios recientes señalan que más del 70 % de las empresas a nivel global son privadas, pero solo una fracción de ellas dispone de datos financieros estructurados accesibles para análisis cuantitativo (Lukander, 2025). Esto limita el alcance de cualquier modelo basado en datos públicos y explica por qué muchos trabajos, incluido este TFG, se centran en empresas cotizadas, donde la información es más consistente.

La segunda limitación es la interpretabilidad. Algunos modelos avanzados, como Random Forest o Gradient Boosting, pueden ofrecer mejores resultados predictivos que modelos lineales, pero son más difíciles de explicar. En un contexto como M&A, donde las decisiones implican grandes volúmenes de capital, no basta con obtener una probabilidad de adquisición. Es necesario entender qué variables están influyendo en el resultado y si la predicción tiene sentido desde un punto de vista financiero y estratégico. Por ejemplo, un modelo puede asignar alta probabilidad a una empresa por una combinación de ratios financieros, pero sin una explicación clara, el analista puede no confiar en el resultado o no saber cómo utilizarlo en la práctica (Balan et al., 2025; Lukander, 2025).

La tercera limitación es el riesgo de sesgo. Los modelos se entrenan con datos históricos, por lo que tienden a reproducir patrones del pasado. Esto puede generar varios problemas. Por un lado, pueden favorecer empresas que se parecen a targets tradicionales, como compañías de cierto tamaño, sector o perfil financiero. Por otro lado, pueden penalizar empresas innovadoras o en fases tempranas, cuyos indicadores financieros todavía no reflejan su valor estratégico. Además, el propio fenómeno de M&A es relativamente raro en comparación con el número total de empresas, lo que genera un problema de desbalanceo en los datos: hay muchas más empresas no adquiridas que adquiridas. Esto puede hacer que el modelo aprenda a predecir mayoritariamente la clase negativa si no se aplican técnicas específicas para corregir este desequilibrio (Beckenstrater, 2024).

Por último, también existe el riesgo de sesgos derivados de la selección de variables. Si el modelo solo utiliza datos financieros, deja fuera factores clave como la estrategia del comprador, sinergias potenciales, cultura empresarial o negociaciones internas, que son determinantes en una operación real. Por eso, este tipo de herramienta debe entenderse como un apoyo al analista. Su valor está en ordenar información y priorizar empresas, pero no en sustituir el análisis completo ni la toma de decisiones final.

2.1.8 Enfoque adoptado en este TFG

Este proyecto adopta un enfoque concreto y limitado. No intenta predecir todo el resultado de una operación ni sustituir el proceso completo de M&A. Se centra en la fase inicial de screening y utiliza únicamente datos financieros públicos de empresas cotizadas estadounidenses. Esta decisión reduce el alcance, pero hace que el análisis sea más reproducible.

La aportación principal está en conectar tres elementos: datos financieros públicos, modelos de clasificación supervisada y una herramienta visual de screening. A diferencia de una base de datos tradicional, el objetivo no es solo permitir búsquedas manuales. El objetivo es generar una probabilidad estimada de adquisición y ordenar empresas según esa probabilidad. A diferencia de un estudio puramente académico, el proyecto también busca traducir el modelo a una herramienta práctica mediante un dashboard.

Este enfoque no elimina la incertidumbre del M&A. Una adquisición siempre dependerá de factores internos, privados y estratégicos que no se pueden observar completamente desde fuera. Sin embargo, sí permite trabajar mejor con la información disponible. Si el modelo consigue identificar patrones en

empresas que históricamente han sido adquiridas, puede servir como primer filtro para guiar el análisis. En ese sentido, el valor del proyecto no está en predecir el futuro con certeza, sino en hacer el proceso de screening más sistemático, rápido y defendible.

2.2 DATOS FINANCIEROS PÚBLICOS Y BASES DE DATOS UTILIZADAS

El proyecto utiliza datos financieros públicos porque permiten trabajar con información estructurada, comparable y reproducible. Los estados financieros recogen información sobre tamaño, liquidez, endeudamiento, rentabilidad, crecimiento, valoración y eficiencia operativa. Estas dimensiones no explican por sí solas una adquisición, pero sí pueden reflejar características que hacen que una empresa sea más o menos atractiva como posible target.

La fuente utilizada para identificar operaciones de M&A es PitchBook. Su función dentro del proyecto es proporcionar la información necesaria para construir la variable objetivo: las empresas que aparecen como targets en operaciones completadas se etiquetan como ma_target = 1. El análisis se limita a operaciones completadas entre 2005 y 2020, con targets estadounidenses cotizados y tamaño mínimo de 100 millones de dólares, con el fin de trabajar con una muestra más consistente y con mejor cobertura de datos.

La fuente financiera principal es Compustat, accesible a través de WRDS. Compustat aporta información anual de empresas cotizadas, incluyendo variables de balance, cuenta de resultados, flujos de caja y datos de mercado. A partir de estas variables se calculan ratios como total_debt, ebitda_margin, leverage, capex_intensity, roa, current_ratio, market_cap, ev_ebitda, asset_turnover y cash_ratio. La construcción concreta del dataset, los filtros aplicados y el proceso de cruce entre PitchBook y Compustat se detallan en el Capítulo 3.

La principal ventaja de este enfoque es que permite construir un pipeline reproducible y aplicar el modelo a un universo amplio de empresas cotizadas. Su limitación principal es que deja fuera información privada y cualitativa que suele ser clave en M&A, como conversaciones entre comprador y vendedor, sinergias, presión de accionistas, calidad del equipo directivo o intención estratégica del comprador. Por tanto, los resultados deben interpretarse como una señal financiera de apoyo, no como una explicación completa del proceso de adquisición.

2.3 MACHINE LEARNING SUPERVISADO Y CLASIFICACIÓN BINARIA

2.3.1 Concepto general de Machine Learning

El Machine Learning es una rama de la inteligencia artificial que permite construir modelos capaces de aprender patrones a partir de datos. En lugar de programar una regla fija para cada situación, el modelo ajusta sus parámetros utilizando ejemplos históricos y después aplica lo aprendido a nuevos casos. Esta idea es especialmente útil cuando el problema depende de muchas variables y no existe una regla sencilla que explique el resultado (Mitchell, 1997; James et al., 2021).

En este proyecto, el Machine Learning se utiliza para estimar la probabilidad de que una empresa sea adquirida. El modelo no recibe instrucciones directas del tipo "si una empresa tiene este nivel de deuda, entonces será target". Lo que hace es analizar empresas que fueron adquiridas y empresas que no lo fueron, buscando patrones en sus variables financieras. Después, utiliza esos patrones para asignar una probabilidad de adquisición a otras compañías.

Los modelos de Machine Learning suelen clasificarse en aprendizaje supervisado, no supervisado, semi-supervisado y por refuerzo (Talaei Khoei & Kaabouch, 2023). En este TFG se utiliza aprendizaje supervisado, porque el dataset contiene una variable objetivo conocida: si la empresa fue adquirida o no.

2.3.2 Tipos de aprendizaje

Dentro de Machine Learning existen varios enfoques. Los más habituales son aprendizaje supervisado, aprendizaje no supervisado y aprendizaje por refuerzo. La diferencia principal está en el tipo de información disponible durante el entrenamiento.

En el aprendizaje supervisado, cada observación del dataset tiene una salida conocida. El modelo aprende a partir de pares entrada-salida. Es decir, recibe variables explicativas y una etiqueta asociada. En este TFG, las variables explicativas son los datos financieros de las empresas y la etiqueta indica si la empresa fue adquirida o no. Por tanto, el proyecto se plantea como un problema de aprendizaje

supervisado. La Figura 4 muestra un esquema visual simplificado del procedimiento de aprendizaje supervisado.

Dentro del aprendizaje supervisado existen dos tipos principales de problemas: regresión y clasificación. La regresión se utiliza cuando la variable que se quiere predecir es continua, mientras que la clasificación se utiliza cuando la variable objetivo pertenece a un conjunto limitado de clases (Sanz Bobi, 2024). En este proyecto, la etiqueta tiene dos valores posibles: target y no target. Por eso, se trata de un problema de clasificación binaria.

En el contexto de este proyecto, una observación tendría esta forma:

$$Empresa\ A \rightarrow variables\ financieras \rightarrow target = 1$$
$$Empresa\ B \rightarrow variables\ financieras \rightarrow target = 0$$

A partir de muchos ejemplos de este tipo, el modelo intenta aprender qué combinaciones de variables aparecen con mayor frecuencia en empresas que acaban siendo adquiridas. Después, cuando recibe una nueva empresa, estima su probabilidad de pertenecer a la clase target.

Esta probabilidad es más útil que una simple etiqueta. En un problema de screening, no interesa únicamente saber si el modelo clasifica una empresa como target o no target. Lo más útil es ordenar empresas de mayor a menor probabilidad estimada. Así, el analista puede empezar revisando aquellas que el modelo considera más parecidas a targets históricos.

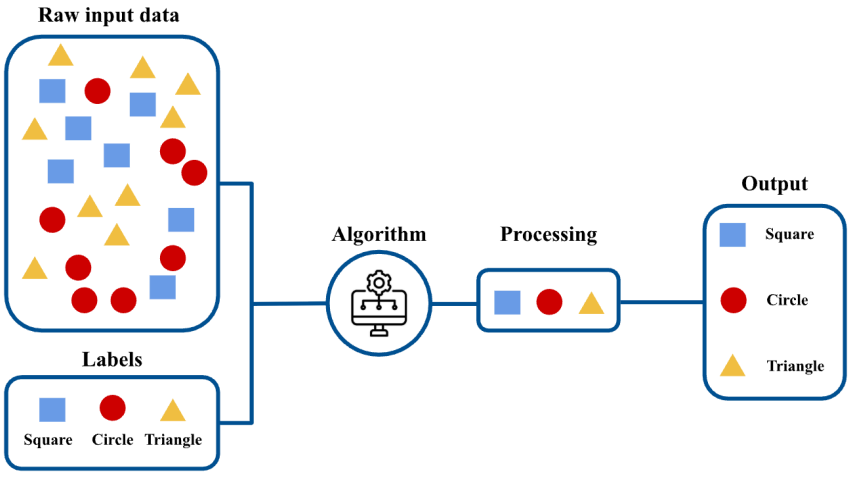


Figura 4. Esquema de aprendizaje supervisado.

Fuente: elaboración propia a partir de materiales de James et al., 2021.

2.3.3 Variables explicativas y variable objetivo

En un modelo supervisado, las variables explicativas son las características que el modelo utiliza para aprender. En este proyecto, esas variables proceden de los estados financieros de las empresas. Se incluyen indicadores de tamaño, liquidez, endeudamiento, rentabilidad, crecimiento y valoración. Estas variables actúan como señales que pueden estar relacionadas con la probabilidad de adquisición.

La variable objetivo es la etiqueta que el modelo intenta predecir. En este caso, la variable objetivo toma dos valores: 1 si la empresa fue adquirida y 0 si no fue adquirida. Esta etiqueta se construye a partir de la información de operaciones de M&A obtenida de PitchBook y se combina con las variables financieras procedentes de Compustat. Hay que tener cuidado porque existe riesgo de data leakage. Esto ocurre cuando el modelo utiliza información que no habría estado disponible en el momento real de la predicción. En este proyecto, para evitar ese problema, las variables financieras se toman antes de la adquisición. Así, el modelo aprende con información que podría haber sido utilizada en un proceso real de screening.

2.3.4 Sesgo, varianza, underfitting y overfitting

Al entrenar modelos predictivos aparece siempre un equilibrio entre sesgo y varianza. El sesgo aumenta cuando el modelo es demasiado simple y no consigue representar bien la estructura del problema, mientras que la varianza aumenta cuando el modelo se adapta demasiado a los datos concretos de entrenamiento y pierde capacidad de generalización sobre datos nuevos (Sanz Bobi, 2024). En el primer caso aparece underfitting, con errores altos tanto en entrenamiento como en test. En el segundo aparece overfitting, donde el modelo puede obtener muy buenos resultados en entrenamiento, pero empeorar claramente al aplicarse sobre el conjunto de test.

La Figura 5, a continuación, muestra la relación entre la complejidad del modelo, el sesgo y la varianza. A medida que aumenta la complejidad, el modelo tiene más capacidad para capturar patrones, pero también aumenta el riesgo de ajustarse al ruido del conjunto de entrenamiento. Por ello, el punto más relevante es la zona intermedia, donde se alcanza un equilibrio entre ambos efectos. En este proyecto, este equilibrio es especialmente importante porque el objetivo no es memorizar empresas históricas adquiridas, sino identificar patrones que puedan aplicarse a empresas no vistas.

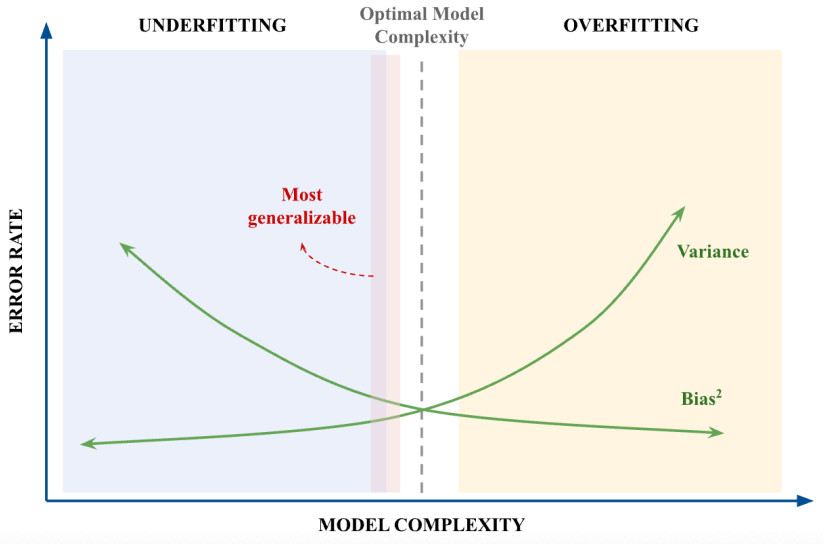


Figura 5. Esquema relacionando la varianza y el sesgo con la complejidad del modelo.

Fuente: elaboración propia a partir del material de la asignatura (Sanz Bobi, 2024)

Para reducir estos riesgos se utilizan varias estrategias: separar correctamente los conjuntos de entrenamiento y test, aplicar validación cruzada durante el ajuste de hiperparámetros, controlar la complejidad de los modelos y evaluar los resultados con métricas adecuadas al problema. En este caso, debido al desbalanceo de clases, no basta con analizar la *accuracy*; también es necesario revisar métricas como AUC-ROC, *recall*, precisión y matriz de confusión, que permiten entender mejor la capacidad real del modelo para identificar targets (Beckenstrater, 2024; Sanz Bobi, 2024).

2.3.5 Validación cruzada

La validación cruzada permite estimar de forma más robusta el rendimiento de un modelo. En lugar de depender de una única partición entre entrenamiento y validación, el dataset se divide en varios bloques o folds. El modelo se entrena varias veces, dejando cada vez un fold distinto para validación. Al final, se calcula una media de los resultados obtenidos.

Este procedimiento ayuda a comprobar si el rendimiento del modelo depende demasiado de una partición concreta de los datos. También es útil para comparar modelos y ajustar hiperparámetros. Según los apuntes de la asignatura, la validación cruzada se utiliza para estimar la capacidad predictiva sobre casos no vistos y para proteger frente al overfitting, especialmente cuando el número de observaciones es limitado (Sanz Bobi, 2024).

En este proyecto, la validación cruzada se utiliza durante la fase de entrenamiento y ajuste de modelos. Sin embargo, el conjunto de test final se mantiene separado para evaluar el modelo seleccionado. Esto evita que la evaluación final esté contaminada por decisiones tomadas durante el entrenamiento.

2.3.6 Aplicación al screening de targets

La aplicación de aprendizaje supervisado al screening de targets permite convertir un problema de análisis financiero en un problema de clasificación. El modelo aprende a partir de empresas históricas y estima una probabilidad de adquisición para nuevas compañías. Esa probabilidad se utiliza después para generar un ranking.

Esta forma de interpretar el modelo es importante. El objetivo no es que el algoritmo tome una decisión de adquisición. Tampoco se espera que prediga todos los targets reales. El objetivo es reducir el universo inicial y ayudar al analista a priorizar empresas. Una empresa con probabilidad estimada alta no se interpreta como una adquisición segura, sino como una compañía que merece una revisión más profunda.

Por tanto, el Machine Learning se utiliza como una herramienta de apoyo. El modelo aporta una primera capa de análisis sistemático y reproducible. Después, el analista debe complementar el resultado con información sectorial, estrategia del comprador, análisis de sinergias, valoración y criterio profesional.

En este sentido, la clasificación supervisada encaja bien con el propósito del TFG. Permite aprender de operaciones pasadas, aplicar ese aprendizaje a empresas no observadas y construir una herramienta práctica para mejorar la eficiencia del screening inicial de M&A.

2.4 MODELOS DE MACHINE LEARNING PARA CLASIFICACIÓN

2.4.1 Criterios para elegir un modelo

Una vez definido el problema como una clasificación binaria, el siguiente paso es elegir qué modelos tiene sentido probar. No existe un único algoritmo que sea siempre mejor. La elección depende del tipo de datos, del tamaño de la muestra, del número de variables, del desbalanceo de clases, de la interpretabilidad necesaria y del objetivo final del proyecto.

En este caso, el objetivo es estimar la probabilidad de que una empresa cotizada estadounidense sea target de M&A. Por tanto, el modelo debe cumplir varias condiciones. Primero, debe poder trabajar con variables financieras numéricas. Segundo, debe ofrecer una salida interpretable como probabilidad o score de adquisición. Tercero, debe ser capaz de capturar relaciones no lineales, porque la probabilidad de adquisición no depende de una sola variable financiera. Cuarto, debe generalizar bien, ya que no interesa memorizar operaciones históricas, sino detectar patrones que puedan aplicarse a empresas no vistas. Por último, debe tener un nivel razonable de interpretabilidad, ya que el resultado se va a utilizar como herramienta de apoyo para analistas.

La selección de modelos también debe equilibrar simplicidad y capacidad predictiva como mencionado anteriormente. Queremos un buen trade-off entre sesgo y varianza para tener una complejidad óptima para el modelo que se ajuste bien a los datos. Un modelo demasiado simple puede no capturar patrones relevantes. Un modelo demasiado complejo puede ajustarse al ruido del dataset y perder capacidad de generalización. Por eso, en un proyecto aplicado no tiene sentido probar solo el modelo más complejo. Es mejor comparar varias familias de modelos y analizar qué aporta cada una.

En problemas de M&A, varios trabajos recientes han seguido esta misma lógica. En lugar de utilizar un solo algoritmo, suelen comparar modelos lineales, modelos basados en árboles, métodos ensemble y, en algunos casos, redes neuronales o modelos híbridos. La razón es que el fenómeno que se quiere estudiar es complejo y no se conoce de antemano qué estructura de modelo va a capturar mejor la señal existente en los datos (Beckenstrater, 2024; Campbell et al., 2025; Lukander, 2025).

2.4.2 Modelos lineales

Los modelos lineales suelen utilizarse como primera referencia en problemas de clasificación porque son simples, rápidos e interpretables. En este TFG se utiliza la regresión logística como modelo base para comprobar si las variables financieras contienen una señal lineal suficiente para distinguir entre empresas target y no target.

Aunque se denomina regresión, la regresión logística se aplica a problemas de clasificación binaria. En este caso, estima la probabilidad de que una empresa sea target a partir de sus variables financieras, transformando el resultado mediante una función logística para obtener valores entre 0 y 1 (Sanz Bobi, 2024; James et al., 2021). Su principal ventaja es que permite interpretar de forma relativamente sencilla qué variables aumentan o reducen la probabilidad estimada de adquisición.

Sin embargo, este modelo también tiene una limitación importante: asume relaciones lineales entre las variables explicativas y la probabilidad de pertenecer a la clase positiva. En el contexto de M&A, esta hipótesis puede ser demasiado restrictiva, ya que una empresa no se convierte en target por una única variable aislada, sino por la combinación de factores como tamaño, rentabilidad, deuda, liquidez, valoración o sector. Por ello, la regresión logística se utiliza en este proyecto como benchmark, no como modelo final esperado. Su función es servir como referencia sencilla e interpretable frente a modelos más flexibles capaces de capturar relaciones no lineales.

2.4.3 Árboles de decisión

Los árboles de decisión son modelos no lineales que dividen el espacio de variables mediante reglas sucesivas. En cada nodo, el modelo selecciona una variable y un umbral que separan mejor las observaciones según la clase objetivo. El resultado final es una estructura tipo árbol, donde cada camino desde la raíz hasta una hoja representa una secuencia de decisiones.

Su ventaja principal es que son fáciles de interpretar. Un árbol puede leerse como un conjunto de reglas. Por ejemplo, podría separar empresas según tamaño, después según rentabilidad y después según endeudamiento. Esta lógica se parece bastante al razonamiento humano, donde se van aplicando filtros para decidir si una empresa merece un análisis más profundo (Sanz Bobi, 2024).

Los árboles también capturan relaciones no lineales e interacciones entre variables sin necesidad de especificarlas manualmente. Esto los hace atractivos para problemas financieros, donde muchas relaciones no son estrictamente lineales.

El problema es que un único árbol suele ser inestable. Pequeños cambios en los datos de entrenamiento pueden generar árboles distintos. Además, si el árbol crece demasiado, puede ajustarse en exceso al conjunto de entrenamiento. Esto produce overfitting. Por tanto, aunque los árboles son útiles para entender la lógica de partición, normalmente no se utilizan solos cuando se busca un modelo más robusto.

Esta limitación dio lugar al uso de métodos ensemble basados en árboles, como Random Forest y Gradient Boosting. Estos modelos combinan muchos árboles para mejorar la capacidad predictiva y reducir la inestabilidad de un único árbol.

2.4.4 Métodos ensemble

Los métodos ensemble combinan varios modelos para obtener una predicción final más robusta. La idea es que un conjunto de modelos puede funcionar mejor que un único modelo, siempre que los modelos individuales aporten cierta diversidad. En clasificación, las predicciones pueden combinarse mediante voto mayoritario, media de probabilidades o técnicas más avanzadas como stacking (Sanz Bobi, 2024; Hastie et al., 2009).

Esta familia de modelos es especialmente relevante en este proyecto. Los datos financieros pueden contener ruido, valores extremos, relaciones no lineales y combinaciones complejas de variables. Un único modelo puede ser demasiado rígido o demasiado sensible. En cambio, un ensemble puede reducir parte de esa inestabilidad.

Existen tres enfoques principales dentro de los métodos ensemble: bagging, boosting y stacking. Este último no se utilizará y se omite en esta sección.

Dentro de los *ensemble* destacan dos enfoques principales. El primero es bagging, que entrena modelos en subconjuntos aleatorios del dataset y combina sus resultados para reducir la varianza; Random Forest pertenece a esta familia. El segundo es boosting, que construye modelos de forma secuencial, haciendo que cada nuevo árbol corrija errores de los anteriores y reduciendo el sesgo; Gradient Boosting, XGBoost y LightGBM pertenecen a este grupo.

En estudios recientes sobre M&A, los modelos ensemble aparecen con frecuencia. Beckenstrater (2024) compara regresión logística con Random Forest, LightGBM, LSTM, TabTransformer y un ensemble. Campbell et al. (2025) también prueban distintos modelos y encuentran que los modelos no lineales basados en árboles superan a modelos lineales en la predicción de resultados post-adquisición. Esto refuerza la idea de que, para problemas de M&A, los modelos capaces de capturar no linealidades suelen ser más adecuados que los enfoques puramente lineales.

2.4.5 Random Forest

Random Forest es un método ensemble basado en bagging. Entrena muchos árboles de decisión sobre distintas muestras del dataset y, además, introduce aleatoriedad en la selección de variables utilizadas en cada división del árbol. Después combina las predicciones de todos los árboles para obtener una decisión final o una probabilidad estimada (Breiman, 2001).

La ventaja principal de Random Forest es que reduce la varianza de los árboles individuales. Un árbol aislado puede sobreajustarse fácilmente. En cambio, al combinar muchos árboles entrenados con distintas muestras y distintas variables, el modelo se vuelve más estable. Esto lo hace útil en datasets con ruido o con relaciones no lineales.

Random Forest también tiene otra ventaja importante para este proyecto: permite estimar la importancia de las variables. Esto es relevante porque el objetivo no es solo obtener una probabilidad, sino también entender qué características financieras parecen estar asociadas a una mayor probabilidad de adquisición.

En el contexto de M&A, Random Forest ha sido utilizado en varios trabajos previos. Algunos estudios encuentran que este modelo funciona bien en predicción de takeovers o resultados de operaciones, especialmente cuando existen interacciones entre variables financieras, de mercado y del deal (Beckenstrater, 2024; Campbell et al., 2025). También es un modelo atractivo porque ofrece un buen equilibrio entre rendimiento, robustez e interpretación.

Aun así, Random Forest no está libre de limitaciones. Puede perder interpretabilidad frente a modelos lineales, puede generar modelos pesados si se utilizan muchos árboles y sus probabilidades pueden necesitar calibración si se quieren interpretar de forma estricta.

2.4.6 Gradient Boosting

Gradient Boosting es otro método ensemble basado en árboles, pero funciona de forma secuencial. En lugar de entrenar muchos árboles independientes, construye árboles uno detrás de otro. Cada nuevo árbol intenta corregir los errores del conjunto de árboles anterior. El modelo final es la suma de muchos árboles débiles, ajustados de forma gradual para mejorar la predicción (Friedman, 2001).

La principal ventaja de Gradient Boosting es su capacidad para capturar relaciones complejas. Al ir corrigiendo errores de forma iterativa, puede ajustar patrones no lineales con mucha precisión. En este TFG, Gradient Boosting tiene sentido porque el problema puede contener patrones sutiles. Una empresa puede parecer target no por una sola métrica, sino por una combinación de tamaño, rentabilidad, liquidez, crecimiento y endeudamiento. Un modelo secuencial puede capturar este tipo de interacciones mejor que un modelo lineal.

La limitación principal es que Gradient Boosting puede ser sensible a los hiperparámetros. Parámetros como la tasa de aprendizaje, el número de árboles y la profundidad de cada árbol influyen mucho en el rendimiento.

2.4.7 XGBoost y otros modelos de boosting avanzado

XGBoost, o Extreme Gradient Boosting, es una implementación optimizada de gradient boosting. Fue diseñado para mejorar eficiencia, escalabilidad y control del overfitting. Incluye regularización, manejo eficiente de datos dispersos, paralelización y distintas opciones de ajuste que lo han convertido en uno de los modelos más utilizados en problemas de datos tabulares (Chen & Guestrin, 2016).

La ventaja de XGBoost frente a Gradient Boosting tradicional es que suele ofrecer mayor rendimiento y más control sobre la complejidad del modelo. En problemas financieros, XGBoost puede ser útil porque maneja relaciones no lineales y variables con distinta importancia. También puede capturar interacciones sin tener que definirlas manualmente. Sin embargo, esa potencia tiene un coste. Si no se controla bien, puede sobreajustarse, especialmente cuando el dataset no es muy grande o cuando la clase positiva es minoritaria.

Existen otros modelos de boosting avanzado, como LightGBM y CatBoost. LightGBM está diseñado para trabajar de forma eficiente con datasets grandes y suele funcionar muy bien en problemas tabulares.

2.4.8 Otros modelos considerados

Además de los modelos finalmente entrenados, existen otras familias de algoritmos que podrían aplicarse a problemas de clasificación, como k-Nearest Neighbors, Support Vector Machines, redes neuronales o modelos híbridos. kNN clasifica una observación según sus vecinos más cercanos, mientras que SVM busca una frontera que separe las clases maximizando el margen entre ellas. Ambos modelos pueden ser útiles en ciertos contextos, pero presentan limitaciones para este proyecto: son más sensibles a la escala de las variables, pueden tener mayor coste computacional y su interpretación resulta menos directa cuando se trabaja con un universo amplio de empresas financieras (Sanz Bobi, 2024).

También se han utilizado redes neuronales y arquitecturas más avanzadas en trabajos recientes de M&A, especialmente cuando se dispone de grandes volúmenes de datos, series temporales o información no estructurada, como noticias, informes anuales o transcripciones de resultados (Beckenstrater, 2024; Hastie et al., 2009; Talaei Khoei & Kaabouch, 2023). Sin embargo, estos modelos suelen requerir más datos, más ajuste y ofrecen menor interpretabilidad, lo que no encaja del todo con el objetivo de este TFG: construir una herramienta de screening reproducible y comprensible para el analista.

Por último, algunos trabajos combinan varios algoritmos mediante modelos híbridos o ensembles complejos. Aunque este enfoque puede mejorar el rendimiento cuando los modelos capturan patrones distintos, también aumenta la complejidad y el riesgo de overfitting si no se valida correctamente (Lukander, 2025). Por este motivo, en este trabajo se priorizan modelos individuales más estándar y adecuados para datos tabulares financieros: regresión logística, Random Forest, Gradient Boosting, XGBoost y LightGBM. Esta selección permite comparar distintos niveles de complejidad sin perder interpretabilidad ni claridad metodológica.

2.4.9 Modelo extra de la literatura académica - LightGBM

Como prueba adicional, se decidió entrenar también un modelo LightGBM. Esta decisión no formaba parte del planteamiento inicial del modelaje, pero se incorporó al observar que algunos trabajos

recientes sobre predicción de operaciones de M&A utilizan este algoritmo y obtienen buenos resultados. En concreto, Beckenstrater (2024) identifica LightGBM como uno de los modelos con mejor rendimiento dentro de un enfoque de Machine Learning aplicado a la predicción de M&A. Por tanto, se incluyó como extensión del análisis para comprobar si podía mejorar los resultados obtenidos con Random Forest, Gradient Boosting y XGBoost.

LightGBM, propuesto por Ke et al. (2017), es una implementación eficiente de Gradient Boosting basada en árboles de decisión. Igual que otros modelos de boosting, construye árboles de forma secuencial, de manera que cada nuevo árbol intenta corregir los errores cometidos por los anteriores. Su principal ventaja es que está diseñado para entrenar de forma más rápida y eficiente, especialmente en datasets grandes o con muchas variables. A diferencia de otros algoritmos que hacen crecer los árboles por niveles, LightGBM utiliza una estrategia leaf-wise, es decir, prioriza la división de las hojas que generan mayor ganancia. Esta característica puede mejorar el rendimiento, aunque también puede aumentar el riesgo de overfitting si no se controla correctamente la complejidad del modelo.

Aunque LightGBM no se había trabajado como modelo principal en la asignatura, sí está directamente relacionado con los contenidos vistos sobre Gradient Boosting y modelos ensemble. Por este motivo, su inclusión resulta coherente como prueba adicional, pero no sustituye a los modelos centrales del proyecto.

2.4.10 Modelos seleccionados para este TFG

A partir de esta revisión, se seleccionan cuatro modelos principales para el desarrollo del proyecto: regresión logística, Random Forest, Gradient Boosting y XGBoost. La selección busca cubrir distintos niveles de complejidad y distintas formas de capturar patrones.

La regresión logística se utiliza como modelo base. Es sencilla, interpretable y adecuada para clasificación binaria. Permite comprobar si existe una señal lineal básica en las variables financieras. También sirve como referencia para evaluar si los modelos más complejos aportan una mejora real.

Random Forest se incluye porque es un modelo robusto, no lineal y adecuado para datos tabulares. Puede capturar interacciones entre variables y reduce la inestabilidad de los árboles individuales mediante bagging. Además, permite analizar importancia de variables, lo que ayuda a interpretar los resultados desde una perspectiva financiera.

Gradient Boosting se incorpora porque es un modelo secuencial capaz de mejorar progresivamente corrigiendo errores anteriores. Puede detectar patrones más complejos que una regresión logística y suele funcionar bien en problemas estructurados.

XGBoost se incluye como versión más avanzada y regularizada del boosting. Su uso permite comprobar si una implementación optimizada mejora el rendimiento frente a Gradient Boosting tradicional. También permite controlar mejor la complejidad mediante hiperparámetros.

Cabe destacar que dentro de esta selección se harán múltiples versiones de cada modelo modificando parámetros, aplicando validación cruzada, regularización y balance de clases para obtener el mejor modelo posible para el trabajo. Asimismo, esta selección permite comparar modelos lineales y no lineales, modelos simples y complejos, y modelos con distintos equilibrios entre sesgo, varianza e interpretabilidad. No se pretende demostrar que un algoritmo sea universalmente superior. El objetivo es identificar qué enfoque funciona mejor para el problema concreto de screening de targets con datos financieros públicos.

A continuación, la Tabla 2 muestra un resumen de los modelos argumentados en el apartado 2.4.

Familia de modelo	Ejemplos	Ventajas principales	Limitaciones principales	Papel en este TFG
Modelos lineales	Regresión logística, LASSO, Ridge	Simples, rápidos e interpretables	Capturan peor relaciones no lineales	Se usa regresión logística como benchmark
Árboles de decisión	CART	Interpretables y no lineales	Inestables y propensos a overfitting	Base conceptual de modelos ensemble
Modelos basados en distancia o margen	kNN, SVM	Útiles en ciertos problemas de clasificación	Sensibles a escala, dimensionalidad e hiperparámetros	Se revisan, pero no se priorizan
Bagging	Random Forest	Reduce varianza, robusto, útil con datos tabulares	Menos interpretable que modelos lineales	Modelo principal comparado
Boosting	Gradient Boosting, XGBoost	Alto rendimiento y captura de interacciones complejas	Mayor riesgo de overfitting si no se ajusta bien	Se prueban ambos
Redes neuronales	MLP, LSTM, TabTransformer	Muy flexibles y capaces de modelar relaciones complejas	Requieren más datos y son menos interpretables	Se consideran como alternativa futura
Modelos híbridos	Stacking, ensembles mixtos	Pueden mejorar rendimiento combinando modelos	Más complejos y menos transparentes	Posible línea futura

Tabla 1. Resumen de los modelos revisados para el trabajo.

Fuente: elaboración propia a partir de Sanz Bobi (2024), Beckenstrater (2024), Campbell et al. (2025), Lukander (2025) y Talaei Khoei y Kaabouch (2023).

2.5 DESBALANCEO DE CLASES Y TÉCNICAS DE TRATAMIENTO

2.5.1 Problema del desbalanceo de clases

Un dataset está desbalanceado cuando una de las clases aparece con mucha más frecuencia que la otra. En este proyecto, la clase positiva son las empresas adquiridas, es decir, los *targets*, mientras que la clase negativa son las empresas que no fueron adquiridas durante el periodo analizado. Este problema es habitual en predicción financiera, ya que muchos eventos relevantes, como impagos, fraudes, quiebras o adquisiciones, ocurren con menor frecuencia que los casos negativos.

El desbalanceo afecta directamente al entrenamiento del modelo. Muchos algoritmos buscan reducir el error global, por lo que pueden aprender a favorecer la clase mayoritaria. En este caso, un modelo podría clasificar correctamente muchas empresas no target, pero fallar precisamente en las empresas que más interesan: los targets. Esto sería especialmente problemático en una herramienta de *screening*, donde el objetivo no es solo acertar la clase mayoritaria, sino identificar empresas potencialmente adquiribles (He & Garcia, 2009; Sanz Bobi, 2024). En el contexto de M&A, esta dificultad es esperable, ya que las adquisiciones son eventos relativamente poco frecuentes y el desbalanceo es uno de los principales retos en la predicción de targets (Beckenstrater, 2024).

2.5.2 Limitaciones de la accuracy

Por esta razón, la *accuracy* no puede ser la única métrica de evaluación. En datasets desbalanceados, un modelo puede obtener una accuracy aparentemente alta simplemente clasificando casi todas las empresas como no target. Desde el punto de vista del proyecto, ese modelo tendría poco valor, porque no detectaría oportunidades de adquisición. Por ello, es necesario analizar también métricas como *recall*, precisión, F1-score, matriz de confusión y AUC-ROC. Estas métricas permiten entender cuántos targets reales se identifican, cuántos falsos positivos se generan y qué capacidad tiene el modelo para separar ambas clases a distintos umbrales de decisión (Fawcett, 2006; He & Garcia, 2009).

Esta idea es importante para interpretar los resultados del proyecto. En M&A screening, un falso positivo significa que el modelo marca una empresa como posible target aunque no haya sido adquirida. Esto puede implicar revisar una empresa que finalmente no era una oportunidad real. Un

falso negativo, en cambio, significa que el modelo no detecta una empresa que sí fue adquirida. Desde una perspectiva de screening, este segundo error puede ser más relevante, porque implica perder una posible oportunidad.

2.5.3 Enfoques para tratar el desbalanceo

Existen varias formas de tratar el desbalanceo. La primera consiste en modificar la distribución de clases en el conjunto de entrenamiento, mediante técnicas como undersampling, oversampling o SMOTE. El undersampling reduce ejemplos de la clase mayoritaria, aunque puede eliminar información útil. El oversampling aumenta la presencia de la clase minoritaria duplicando observaciones, pero puede aumentar el riesgo de overfitting. SMOTE, en cambio, genera observaciones sintéticas de la clase minoritaria a partir de ejemplos cercanos, evitando duplicar directamente observaciones existentes (Chawla et al., 2002).

Enfoque	Técnica	Idea principal	Ventaja	Limitación
Modificación de datos	Undersampling	Reducir ejemplos de la clase mayoritaria	Reduce el desequilibrio y el coste computacional	Puede eliminar información útil
Modificación de datos	Oversampling	Duplicar ejemplos de la clase minoritaria	Aumenta la presencia de la clase positiva	Puede aumentar el riesgo de overfitting
Modificación de datos	SMOTE	Crear ejemplos sintéticos de la clase minoritaria	Genera nuevos puntos en lugar de duplicar observaciones	Puede crear ejemplos poco realistas si se aplica mal
Modificación del algoritmo	Class weighting	Penalizar más los errores sobre la clase minoritaria	No altera el dataset original	Depende de la sensibilidad del modelo a los pesos
Ajuste de decisión	Cambio de threshold	Modificar el umbral de clasificación	Permite ajustar recall y precisión	Requiere definir qué error es más costoso

Tabla 2. Técnicas habituales para tratar datasets desbalanceados

Fuente: elaboración propia a partir de Chawla et al. (2002), He y Garcia (2009), Sanz Bobi (2024) y Dube y Verster (2023).

La segunda estrategia consiste en modificar el algoritmo mediante técnicas como *class weighting*. En este caso, el modelo penaliza más los errores cometidos sobre la clase minoritaria. Esto resulta útil cuando fallar un target es más costoso que fallar un no target. La tercera posibilidad es ajustar el umbral de clasificación, ya que muchos modelos devuelven una probabilidad estimada y no solo una etiqueta. Modificar ese umbral permite cambiar el equilibrio entre recall y precisión.

2.5.4 Oversampling, undersampling y SMOTE

El undersampling consiste en reducir el número de observaciones de la clase mayoritaria. Mientras que el overfitting aumenta la presencia de la clase minoritaria. En este TFG se presta atención a SMOTE, o Synthetic Minority Oversampling Technique, intenta resolver parte de ese problema. En lugar de duplicar directamente ejemplos de la clase minoritaria, genera observaciones sintéticas. Para ello, selecciona una observación minoritaria, busca vecinos cercanos de la misma clase y crea un nuevo punto en el espacio de variables entre ambos. De esta forma, el modelo recibe más ejemplos de la clase minoritaria, pero no copias exactas de observaciones existentes (Chawla et al., 2002).

La principal ventaja de SMOTE es que permite ampliar la clase minoritaria de forma más informativa que el oversampling aleatorio. Esto puede mejorar la capacidad del modelo para detectar positivos. En problemas financieros desbalanceados, las técnicas de balanceo pueden mejorar el rendimiento de los clasificadores, aunque su efecto depende del algoritmo y de las características del dataset (Dube & Verster, 2023; Noh, 2023).

Sin embargo, SMOTE también tiene riesgos. Al generar puntos sintéticos, puede crear observaciones que no representen empresas reales. Esto es más probable cuando la clase minoritaria es muy dispersa o cuando los targets tienen perfiles financieros muy distintos entre sí. También puede generar puntos cerca de la frontera entre clases y aumentar la confusión del modelo. Por eso, SMOTE debe usarse con cuidado y siempre evaluando su impacto sobre el conjunto de test para evitar riesgo de data leakage que puede hacer que el rendimiento del modelo parezca mejor de lo que realmente es.

2.5.5 Relación con el screening de M&A

El desbalanceo de clases tiene una relación directa con el objetivo de este TFG. En un proceso de screening de M&A, no todas las empresas tienen la misma importancia para el modelo. La clase negativa es mayoritaria, pero la clase positiva es la que más interesa detectar. Por eso, el modelo debe evaluarse pensando en la utilidad del ranking, no solo en el porcentaje total de aciertos.

Esto no significa que haya que maximizar el recall a cualquier coste. Si el modelo marca demasiadas empresas como targets, el ranking pierde utilidad porque obliga al analista a revisar demasiados falsos positivos. El objetivo es encontrar un equilibrio. Interesa identificar una parte relevante de los targets reales, pero manteniendo un conjunto de candidatos suficientemente manejable.

Por eso, el tratamiento del desbalanceo no es un detalle técnico secundario. Afecta a la forma en la que el modelo aprende, a las métricas que deben analizarse y a la utilidad práctica de la herramienta. En este proyecto, las técnicas de balanceo se entienden como una forma de mejorar la detección de targets sin perder de vista la capacidad de generalización del modelo.

La aplicación concreta de estas técnicas al dataset del proyecto se describe más adelante, en el capítulo de análisis y comparación de modelos. En esta sección solo se presenta la base teórica necesaria para entender por qué se prueban estrategias como class weighting y SMOTE.

2.6 MÉTRICAS DE EVALUACIÓN

Una vez entrenados los modelos, es necesario evaluar si realmente funcionan para el objetivo del proyecto. En clasificación binaria, no basta con saber cuántas predicciones son correctas en total. También hay que entender qué tipo de errores comete el modelo. Esto es especialmente importante en datasets desbalanceados, como ocurre en este TFG, donde la clase target representa una proporción menor que la clase no target.

2.6.1 Matriz de confusión

La matriz de confusión resume las predicciones de un modelo comparando la clase real con la clase estimada. En este proyecto, la clase positiva es target y la clase negativa es no target.

	Predicción: target	**Predicción: no target**
Real: target	True Positive (TP)	False Negative (FN)
Real: no target	False Positive (FP)	True Negative (TN)

Tabla 3. Matriz de confusión para clasificación binaria.

Fuente: elaboración propia a partir de Sanz Bobi (2024) y Fawcett (2006).

En la Tabla 4, cada celda tiene una interpretación concreta. Un true positive es una empresa adquirida que el modelo identifica correctamente como target. Un true negative es una empresa no adquirida que el modelo clasifica correctamente como no target. Un false positive es una empresa que el modelo marca como target, aunque realmente no fue adquirida. Un false negative es una empresa adquirida que el modelo no detecta.

En el contexto del screening de M&A, los falsos negativos son especialmente relevantes porque representan targets reales que el modelo deja fuera. Los falsos positivos también importan, porque generan trabajo adicional para el analista. Sin embargo, revisar algunos falsos positivos puede ser aceptable si el modelo ayuda a identificar más targets reales dentro de un universo amplio.

2.6.2 Accuracy

La accuracy mide el porcentaje total de predicciones correctas:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

Es una métrica sencilla y fácil de interpretar. Sin embargo, en problemas desbalanceados puede ser engañosa. Si la mayoría de empresas son no targets, un modelo puede obtener una accuracy alta clasificando casi todo como no target. En ese caso, el modelo parecería bueno, pero no sería útil para detectar oportunidades de adquisición.

Por este motivo, en este proyecto la accuracy se analiza, pero no se utiliza como única métrica de evaluación. Es necesario complementarla con métricas centradas en la clase positiva.

2.6.3 Recall o sensibilidad

El recall mide qué proporción de targets reales identifica correctamente el modelo:

$$Recall = \frac{TP}{TP+FN}$$

En este TFG, el recall responde a una pregunta directa: de todas las empresas que realmente fueron adquiridas, ¿cuántas detecta el modelo? Por eso es una métrica importante para el screening. Un recall bajo significa que el modelo deja fuera muchos targets reales. Un recall alto indica que el modelo recupera una mayor parte de las oportunidades históricas.

Aun así, maximizar el recall sin controlar otras métricas puede ser problemático. Un modelo podría marcar demasiadas empresas como target y así aumentar el recall, pero también generaría muchos falsos positivos. Por eso, el recall debe interpretarse junto con la precisión.

2.6.4 Precisión

La precisión mide qué proporción de las empresas clasificadas como target realmente pertenecen a la clase target:

$$Precisión = \frac{TP}{TP+FP}$$

En este proyecto, la precisión indica la calidad de la lista de candidatos generada por el modelo. Si la precisión es baja, significa que muchas empresas marcadas como posibles targets no fueron adquiridas. Esto reduce la utilidad práctica del ranking, porque obliga al analista a revisar demasiadas empresas poco relevantes.

La precisión y el recall suelen estar relacionados. Si se baja el umbral de clasificación, el modelo tiende a identificar más targets, pero también puede generar más falsos positivos. Si se sube el umbral, el modelo será más exigente, pero puede dejar fuera más targets reales. Por eso, el objetivo no es maximizar una métrica de forma aislada, sino encontrar un equilibrio razonable para el problema.

2.6.5 F1-score

El F1-score combina precisión y recall en una sola métrica. Se calcula de la siguiente forma:

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$$

Esta métrica es útil cuando se quiere evaluar el equilibrio entre detectar targets y evitar demasiados falsos positivos. En un dataset desbalanceado, el F1-score puede aportar más información que la accuracy, porque da peso a los errores cometidos sobre la clase positiva.

Aun así, el F1-score también tiene limitaciones. Resume dos métricas en un único valor, pero no muestra directamente el número de falsos positivos y falsos negativos. Por eso, en este proyecto se interpreta junto con la matriz de confusión.

2.6.6 Curva ROC y AUC-ROC

Muchos modelos de clasificación no generan solo una etiqueta final, sino una probabilidad estimada. Para convertir esa probabilidad en una clase, se define un umbral. Por ejemplo, si el umbral es 0,5, una empresa con probabilidad superior al 50 % se clasifica como target.

La curva ROC permite analizar el rendimiento del modelo para distintos umbrales. Representa la tasa de verdaderos positivos frente a la tasa de falsos positivos:

$$TPR = \frac{TP}{TP+FN} \qquad FPR = \frac{FP}{FP+TN}$$

El área bajo la curva ROC, conocida como AUC-ROC, mide la capacidad general del modelo para separar targets y no targets. Un AUC cercano a 0,5 indica que el modelo apenas distingue entre clases. Un AUC más alto indica mejor capacidad de separación. Esta métrica es útil porque no depende de un único umbral de decisión. La Figura 6, muestra un rango de ROCs con sus respectivas AUCs como referencia.

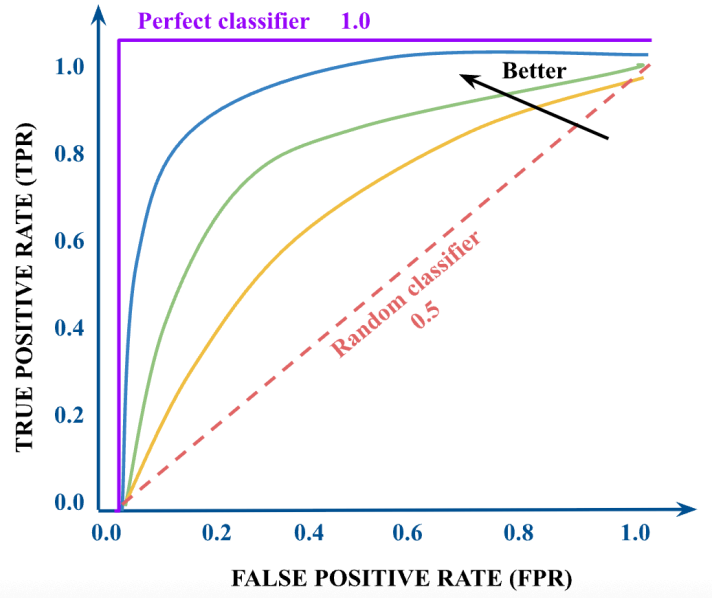


Figura 6. Visualización de distintas curvas ROC con su AUC.

Fuente: elaboración propia a partir de Fawcett (2006) y Sanz Bobi (2024).

En este proyecto, el AUC-ROC se utiliza como una de las métricas principales para comparar modelos. Permite evaluar si el modelo asigna probabilidades más altas a empresas que realmente

fueron adquiridas que a empresas no adquiridas. Esto encaja bien con el objetivo del TFG, que no es solo clasificar empresas, sino construir un ranking de probabilidad de adquisición.

2.6.7 Interpretación de las métricas en el screening de M&A

La elección de métricas debe estar conectada con el uso real de la herramienta. En un modelo de screening, no interesa únicamente acertar muchas observaciones. Interesa identificar targets reales y ordenar empresas de forma útil para el analista.

Por eso, la matriz de confusión permite ver el tipo de errores cometidos. El recall indica cuántos targets reales se recuperan. La precisión mide la calidad de los candidatos marcados como target. El F1-score resume el equilibrio entre ambas. El AUC-ROC mide la capacidad general del modelo para separar clases a distintos umbrales.

En conjunto, estas métricas permiten evaluar el modelo desde una perspectiva técnica y práctica. Un modelo útil para este proyecto no tiene que predecir todas las adquisiciones. Debe ser capaz de concentrar una parte relevante de los targets reales en una lista priorizada de empresas, reduciendo el universo inicial y facilitando el análisis posterior.

2.7 ARQUITECTURA WEB: CLIENTE-SERVIDOR Y APLICACIÓN A HERRAMIENTAS DE SCREENING

La información de esta sección está sacada de los recursos de la asignatura PAT (Programación de Aplicaciones Telemáticas) de la Universidad Pontificia Comillas, ICAI.

2.7.1 El modelo cliente-servidor

La mayoría de las aplicaciones que se usan a través de un navegador siguen un mismo patrón de funcionamiento conocido como arquitectura cliente-servidor. El cliente es el programa que el usuario tiene delante, normalmente el navegador web, y el servidor es un programa que está siempre en ejecución esperando peticiones, normalmente alojado en otro ordenador o, durante el desarrollo, en la misma máquina del programador.

La comunicación entre ambos sigue siempre el mismo ciclo: el cliente envía una petición indicando qué quiere hacer, el servidor la procesa y devuelve una respuesta. Ninguno de los dos actúa por

iniciativa propia hacia el otro fuera de ese ciclo; es siempre el cliente quien abre la conversación. Este modelo es el que sustenta prácticamente toda la web actual, desde una página informativa sencilla hasta una aplicación compleja de análisis de datos.

2.7.2 El protocolo HTTP y sus métodos principales

La comunicación entre cliente y servidor se realiza mediante el protocolo HTTP (HyperText Transfer Protocol). Cada petición HTTP incluye un método que indica la acción que se quiere realizar sobre un recurso del servidor. En el desarrollo de aplicaciones web los dos métodos más utilizados son:

- **GET:** Se emplea para solicitar información sin modificar nada en el servidor.
- **POST:** Se emplea cuando el cliente necesita enviar información al servidor para que la procese, normalmente en forma de un paquete de datos (el "cuerpo" de la petición) que no resulta práctico incluir en la dirección web.

Cada respuesta del servidor incluye un código de estado que informa de cómo ha ido la petición.

- Los códigos que comienzan por 2 indican éxito (la respuesta más habitual es el código 200, "OK")
- Los que comienzan por 4 indican un error originado por la petición del cliente (por ejemplo, pedir un recurso que no existe)
- Los que comienzan por 5 indican un error en el propio servidor.

2.7.3 Separación entre frontend y backend

En el desarrollo de aplicaciones web es habitual distinguir dos capas con responsabilidades distintas.

El **frontend** es la parte que se ejecuta en el navegador del usuario. Está compuesto por:

- HTML, que define la estructura y el contenido de la página
- CSS, que define su apariencia visual (colores, tipografías, distribución de los elementos)
- JavaScript, que añade comportamiento dinámico, permitiendo que la página reaccione a las acciones del usuario sin necesidad de recargarse por completo.

El **backend** es la parte que se ejecuta en el servidor. Es responsable de la lógica de negocio: acceder a los datos, aplicar cálculos o modelos, y devolver los resultados al frontend en un formato que este pueda interpretar.

Esta separación permite que ambas partes evolucionen de forma independiente. El frontend no necesita saber cómo se han calculado los datos que muestra, y el backend no necesita saber cómo se van a presentar visualmente esos datos.

2.7.4 API REST y formato JSON

Cuando el frontend necesita datos del backend, lo habitual es que ambos se comuniquen a través de una API REST (Application Programming Interface, siguiendo el estilo arquitectónico REST). En la práctica, esto significa que el backend expone una serie de direcciones (rutas o "endpoints"), cada una asociada a una función concreta del servidor, a las que el frontend puede hacer peticiones HTTP. Por ejemplo, una ruta puede estar dedicada a devolver un listado de elementos, y otra distinta a devolver el detalle de un elemento concreto.

El formato habitual para intercambiar datos en estas peticiones es JSON (JavaScript Object Notation), un formato de texto estructurado en pares de clave y valor que resulta sencillo de generar desde el backend y de interpretar desde el frontend, independientemente del lenguaje de programación utilizado en cada lado.

Capítulo 3. CAPTURA E INTEGRACIÓN DE DATOS

Este capítulo describe el proceso de construcción del dataset utilizado en el proyecto. Para ello, se explican las fuentes de datos empleadas, PitchBook y Compustat, los filtros aplicados en cada una y el procedimiento seguido para integrar ambas bases y definir la variable objetivo del modelo.

3.1 FUENTES DE DATOS Y CONSTRUCCIÓN DEL DATASET

El primer paso del proyecto consiste en construir el dataset que se utilizará para entrenar los modelos de clasificación. Para ello se combinan dos tipos de información: operaciones históricas de M&A y datos financieros de empresas cotizadas. La lógica es sencilla: primero se identifican empresas que fueron adquiridas y después se buscan sus variables financieras antes de la operación. Con esa información se construye la clase positiva del modelo. Después se añade un grupo de empresas no adquiridas, que funciona como grupo de control.

El dataset se construye a partir de dos fuentes principales: PitchBook y Compustat. PitchBook se utiliza para obtener información sobre operaciones de M&A completadas. Compustat, accesible a través de WRDS, se utiliza para extraer información financiera anual de empresas cotizadas estadounidenses. Cabe mencionar que el acceso a ambas plataformas se pudo gracias a la Universidad de Illinois Urbana-Champaign, en la cual realicé mi intercambio. La combinación de ambas bases permite transformar un problema financiero en un problema de clasificación binaria: target frente a no target.

3.1.1 Extracción de operaciones desde PitchBook

PitchBook es una plataforma de datos financieros utilizada habitualmente en banca de inversión, private equity, consultoría y corporate finance. Proporciona información sobre compañías, inversores, operaciones, múltiplos, fechas, tamaños de transacción y participantes en deals. En este proyecto se utiliza como fuente para identificar qué empresas fueron adquiridas y, por tanto, para construir la etiqueta positiva del modelo.

La extracción de PitchBook se centra en operaciones de M&A completadas. Se aplican varios filtros para limitar el universo inicial y construir una muestra más consistente. En concreto, se seleccionan operaciones completadas entre 2005 y 2020, con targets estadounidenses cotizados y un tamaño mínimo de operación de 100 millones de dólares.

La restricción temporal permite trabajar con una ventana suficientemente amplia para cubrir distintos ciclos de mercado. El periodo incluye años previos a la crisis financiera, la crisis de 2008, la recuperación posterior y los años anteriores a la pandemia. Esto ayuda a que el modelo no dependa únicamente de un contexto económico concreto.

El filtro geográfico se justifica por la disponibilidad y calidad de la información financiera en Estados Unidos. Las empresas cotizadas estadounidenses publican información financiera de forma estandarizada, y Compustat ofrece buena cobertura para este mercado. Además, Estados Unidos concentra un volumen elevado de operaciones de M&A y se utiliza con frecuencia como referencia en estudios empíricos de finanzas corporativas.

El tamaño mínimo de 100 millones de dólares se utiliza para centrarse en operaciones con mayor relevancia económica y mejor cobertura en bases de datos. Las operaciones pequeñas suelen tener menos información disponible y pueden responder a dinámicas distintas. Esta decisión mejora la calidad del dataset, aunque también limita el alcance del modelo. Por tanto, los resultados no deben interpretarse como directamente aplicables a empresas privadas, targets pequeños o transacciones de menor tamaño.

Tras aplicar los filtros anteriores, se obtiene un universo inicial de 9.312 operaciones. Estas operaciones sirven como punto de partida para identificar las empresas adquiridas y construir la clase target del modelo. El Anexo 1 muestra una captura de la interfaz de Pitchbook como referencia una vez seleccionados los filtros.

3.1.2 Extracción de datos financieros desde Compustat

Compustat se utiliza como fuente principal de información financiera. Es una base de datos ampliamente utilizada en investigación financiera y contiene información contable y de mercado de empresas cotizadas, principalmente de Estados Unidos y Canadá. En este proyecto se accede a Compustat a través de WRDS, que permite descargar datos estructurados por empresa y año.

La información extraída de Compustat incluye variables procedentes del balance, la cuenta de resultados, el estado de flujos de caja y datos de mercado. Entre las variables brutas descargadas se incluyen activos totales, ventas, resultado neto, caja, deuda a largo plazo, deuda a corto plazo, cuentas a cobrar, cuentas a pagar, capital circulante, depreciación, EBITDA, gastos de ventas y administración, capex, acciones en circulación, precio de cierre, código SIC y mercado de cotización.

A partir de estas variables brutas se calculan ratios financieros derivados. Estos ratios permiten resumir la situación de la empresa desde distintas perspectivas. En concreto, se calculan indicadores de rentabilidad, endeudamiento, liquidez, tamaño, valoración y eficiencia operativa. Entre ellos se incluyen EBITDA margin, leverage, capex intensity, ROA, current ratio, market capitalization, EV/EBITDA, asset turnover, cash ratio y total debt.

El objetivo de esta extracción no es solo obtener datos financieros de las empresas adquiridas. También se obtiene información de empresas cotizadas que no fueron adquiridas durante el periodo analizado. Estas empresas forman la clase no target y permiten al modelo comparar patrones entre compañías adquiridas y no adquiridas.

La extracción inicial de Compustat incluye aproximadamente 200.000 registros anuales correspondientes a unas 23.000 empresas cotizadas estadounidenses. El Anexo 2 muestra una captura de la interfaz de Compustat a través de WRDS como referencia donde se muestran los pasos que había que seguir para obtener un .csv con los datos buscados. A partir de esta base se construye el universo financiero sobre el que se realiza el cruce con PitchBook.

3.1.3 Integración de PitchBook y Compustat

Una vez extraídas ambas fuentes, el siguiente paso es combinarlas. Esta integración es una de las partes más importantes del proyecto, porque de ella depende la correcta definición de la variable objetivo.

El principal reto es que PitchBook y Compustat no siempre utilizan los mismos identificadores de empresa. En algunos casos, el cruce puede hacerse mediante identificadores comunes. Sin embargo, en otros casos es necesario utilizar el nombre de la empresa como referencia principal. Para resolver este problema se utiliza un proceso de fuzzy matching.

El fuzzy matching permite comparar nombres de empresas aunque no coincidan exactamente. Esto es necesario porque una misma compañía puede aparecer escrita de formas distintas en cada base de datos. Por ejemplo, una fuente puede incluir sufijos como “Inc.”, “Corporation” o “Ltd.”, mientras que otra puede omitirlos. También puede haber diferencias por abreviaturas, signos de puntuación o cambios menores en el nombre.

En este proyecto, el script utilizado para realizar el fuzzy matching y combinar ambas bases fue creado con apoyo de inteligencia artificial, ya que la construcción de este tipo de algoritmo de emparejamiento quedaba fuera del foco principal del trabajo. Su uso se limita a una tarea auxiliar de integración de datos. El objetivo del TFG no es desarrollar un algoritmo de matching, sino construir y evaluar una herramienta de screening de targets mediante Machine Learning.

El proceso de matching se revisa con criterio conservador. Los casos dudosos se descartan para evitar introducir etiquetas incorrectas. Esta decisión reduce el número de targets disponibles, pero mejora la fiabilidad del dataset final. En un problema de clasificación supervisada, una etiqueta mal asignada puede afectar al aprendizaje del modelo, por lo que se prioriza la calidad del cruce frente a maximizar el número de observaciones.

Como resultado del cruce entre PitchBook y Compustat, se identifican 1.589 empresas target. Estas empresas reciben la etiqueta ma_target = 1. El resto del dataset se completa con empresas no adquiridas seleccionadas como grupo de control, que reciben la etiqueta ma_target = 0.

3.1.4 Construcción de la variable objetivo

La variable objetivo del modelo se denomina ma_target. Toma el valor 1 cuando la empresa fue adquirida en una operación de M&A incluida en PitchBook y correctamente emparejada con Compustat. Toma el valor 0 cuando la empresa pertenece al grupo de control y no aparece como adquirida en el periodo analizado.

Esta definición permite plantear el proyecto como un problema de clasificación binaria. Cada fila del dataset representa una empresa-año con sus variables financieras asociadas y una etiqueta que indica si la empresa fue target o no.

Es importante que las variables financieras utilizadas correspondan a información disponible antes de la operación. Si se utilizaran datos posteriores a la adquisición, el modelo estaría aprendiendo con

información que no habría estado disponible en un proceso real de screening. Por eso, la construcción del dataset busca respetar la lógica temporal del problema: primero existen los datos financieros, después ocurre o no ocurre la adquisición.

Además, algunas columnas procedentes de PitchBook se conservan como información contextual. Por ejemplo, pueden aparecer variables como fecha del deal, tamaño de la operación o nombre de la compañía en PitchBook. Sin embargo, estas variables solo existen para empresas target. Por tanto, no se utilizan como variables predictoras del modelo, ya que introducirán información directa sobre la clase objetivo. Sí pueden ser útiles más adelante en el dashboard, para filtrar o mostrar información adicional de operaciones históricas.

3.1.5 Dataset inicial resultante

Tras la integración de fuentes, el dataset inicial contiene 5.071 filas y 51 columnas. El Anexo 3 muestra una tabla de las variables con su significado antes de la fase de limpieza y transformación. De estas observaciones, 1.589 corresponden a empresas target y el resto a empresas no target. Esta estructura genera un dataset desbalanceado, ya que la clase positiva representa una proporción menor que la clase negativa.

Este desbalanceo es coherente con el problema real. En el mercado, la mayoría de empresas no son adquiridas en un periodo determinado. Por tanto, no se fuerza una muestra perfectamente equilibrada desde el inicio. El tratamiento del desbalanceo se aborda más adelante, durante la fase de modelado, mediante técnicas específicas como class weighting y SMOTE.

El dataset inicial no se utiliza directamente para entrenar los modelos. Antes debe pasar por una fase de análisis exploratorio, limpieza y transformación. En esa fase se revisan valores nulos, outliers, correlaciones, multicolinealidad y selección final de variables. El objetivo de este capítulo es describir cómo se construye la base raw del proyecto, que después se prepara para el entrenamiento.

En conjunto, esta fase permite pasar de dos fuentes independientes, PitchBook y Compustat, a un dataset único orientado a Machine Learning. Esta base constituye el punto de partida del pipeline desarrollado en el TFG: datos, modelo y dashboard.

Capítulo 4. EDA, LIMPIEZA Y TRANSFORMACIÓN

Este capítulo describe el proceso seguido para convertir el dataset inicial en una base preparada para el entrenamiento de los modelos. La fase incluye análisis exploratorio, tratamiento de valores nulos, revisión de outliers, análisis de correlaciones, control de multicolinealidad, selección final de variables y separación entre entrenamiento y test. Todos los archivos de código están en github.

4.1 OBJETIVO DEL ANÁLISIS EXPLORATORIO

El análisis exploratorio de datos, o EDA, permite entender la estructura del dataset antes de entrenar cualquier modelo. En esta fase no se busca todavía obtener resultados predictivos. El objetivo es detectar problemas de calidad, revisar la distribución de las variables y decidir qué transformaciones son necesarias para que el modelo pueda aprender de forma correcta.

Esta fase es especialmente importante en un proyecto basado en datos financieros. Las variables contables suelen tener distribuciones muy asimétricas, valores extremos y escalas muy distintas. Por ejemplo, una empresa pequeña puede tener activos de pocos millones, mientras que una compañía grande puede tener activos de cientos de miles de millones. Si estos aspectos no se revisan antes del modelado, pueden afectar al entrenamiento y a la interpretación de los resultados.

El dataset inicial utilizado en esta fase contiene 5.071 filas y 51 columnas. Incluye variables procedentes de Compustat, información contextual de PitchBook y la variable objetivo ma_target, que indica si una empresa fue adquirida o no.

4.2 TRATAMIENTO DE VALORES NULOS

El primer problema detectado fue la presencia de valores nulos. En total, el dataset inicial contenía 34.531 valores faltantes. Sin embargo, no todos los nulos tenían el mismo significado. Como se puede ver en el Figura 7, una parte importante procedía de columnas de PitchBook, como deal_date, deal_size, pb_company, pb_industry_sector o pb_investors. Estas columnas solo tienen información para empresas que fueron adquiridas. Por tanto, sus valores nulos en empresas no target no representan errores, sino ausencia natural de información de deal.

deal_type2	95.82
pb_industry_group	68.92
pb_industry_sector	68.92
pb_investors	68.88
deal_size	68.66
deal_date	68.66
pb_company	68.66
match_score	68.66
current_ratio	20.47
wcap	20.23
xsga	16.68
market_cap	9.68
prcc_f	9.47
dp	4.32
sich	3.45
ebitda	3.39
oibdp	3.39
capex_intensity	2.72
capx	2.72
csho	2.50
pb_industry_code	2.45
ap	0.81
rect	0.69
dltt	0.35
lt	0.20
dlc	0.18
che	0.02
cash_ratio	0.02

dtype: float64

Por esta razón, las columnas de contexto de PitchBook se separaron del dataset de modelado. Estas variables no se utilizan para entrenar el modelo porque solo existen cuando ya se sabe que hubo una operación. Incluirlas generaría data leakage, es decir, el modelo estaría utilizando información posterior o directamente asociada al evento que se quiere predecir. Aun así, se conservan en una copia separada del dataset porque pueden ser útiles más adelante para el dashboard.

Tras eliminar estas columnas del dataset de modelado, quedaron 42 columnas. Los valores nulos restantes se concentraban principalmente en variables financieras como current_ratio, wcap, xsga, market_cap y prcc_f. Eliminar directamente todas las filas con algún valor nulo habría reducido el dataset de 5.071 a 3.160 filas. Esto supondría perder 1.911 observaciones, una reducción demasiado alta para un problema donde la clase target ya es minoritaria.

Figura 7. % de nulls por variable

Por este motivo, se siguió una estrategia intermedia. Primero se eliminaron las filas con seis o más valores nulos, porque representaban observaciones con poca información útil. Después se imputaron los valores restantes usando la mediana de cada variable. Esta decisión está respaldada por la literatura en análisis de datos y machine learning, donde se recomienda el uso de la mediana frente a la media en presencia de distribuciones asimétricas y valores extremos (James et al., 2021; Han et al., 2012). En estos contextos, la media puede quedar muy distorsionada por empresas extremas, mientras que la mediana representa mejor el valor típico de la muestra.

Después de eliminar las filas con más nulos, el dataset quedó formado por 4.940 observaciones: 1.552 targets y 3.388 no targets. El ratio resultante sigue siendo desbalanceado, pero conserva una base suficiente para entrenar modelos y refleja mejor la naturaleza real del problema, donde las empresas adquiridas son minoría.

```
Filas actuales: 5071
Filas sin ningún nulo: 3160
Filas perdidas: 1911
```

	n_nulos
0	3160
1	395
2	675
3	439
4	115
5	156
6	103
7	16
8	9
9	1
10	1
14	1

```
Filas después: 4940
Targets conservados: 1552
No targets: 3388
```

Figura 8. Antes y después de eliminar los valores nulos graves e interpolar por la mediana.

4.3 ANÁLISIS DE OUTLIERS

Después de revisar los valores nulos, se analizaron los outliers. En datos financieros, la presencia de valores extremos es habitual. No siempre son errores. Muchas veces reflejan diferencias reales de tamaño, sector o modelo de negocio. Por eso, no se eliminaron automáticamente. Cada caso se interpretó teniendo en cuenta el significado financiero de la variable.

Primero se generaron boxplots de las variables numéricas. Como las variables estaban en escalas muy distintas, también se creó una versión normalizada entre 0 y 1 mediante MinMaxScaler para comparar mejor la distribución relativa de cada variable. Esta visualización mostró que las variables de tamaño absoluto, como activos totales, ventas, capitalización bursátil, pasivos o deuda total, tenían muchos outliers hacia la derecha. Esto era esperable, ya que en el mercado hay pocas empresas muy grandes y muchas empresas medianas o pequeñas.

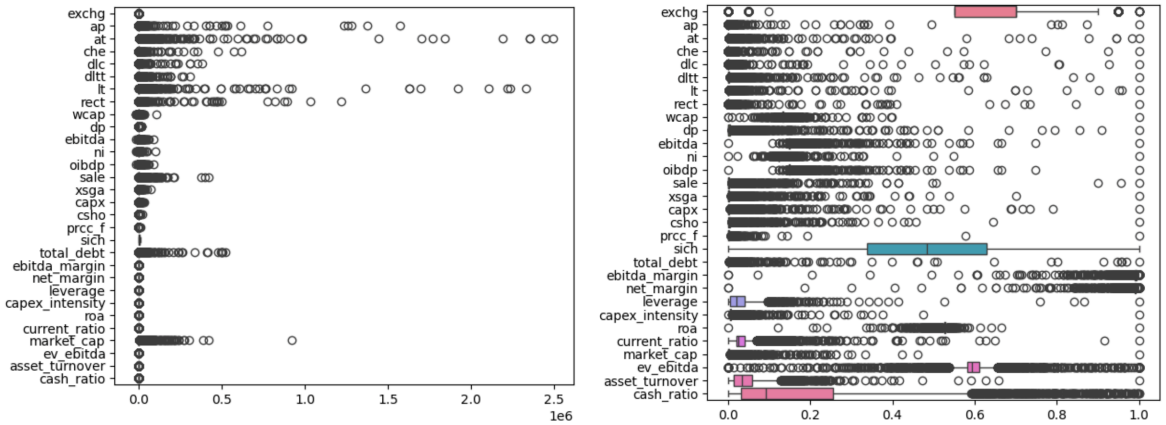


Figura 9. Antes y después de normalización de variables numéricas

La Tabla 5 muestra un resumen de los outliers revisados a continuación.

En las variables de tamaño absoluto, como at, sale, market_cap, lt y total_debt, se observó una concentración de valores cerca de cero y muchos outliers hacia la derecha. Este patrón es normal en datos financieros. El universo de empresas cotizadas incluye muchas compañías medianas o pequeñas y un número reducido de empresas muy grandes. Por tanto, estos outliers no se interpretaron como errores, sino como una consecuencia esperable de la distribución del tamaño empresarial. Aun así, este resultado confirmó la necesidad de escalar las variables antes de entrenar los modelos.

En las variables de rentabilidad, como ebitda_margin, net_margin y roa, aparecieron valores negativos extremos. Al revisar los casos concretos (Anexo 4 muestra un ejemplos de estos casos), se observó que varios correspondían a empresas biotecnológicas, farmacéuticas o compañías en etapas tempranas, como Northwest Biotherapeutics, Sarepta Therapeutics o Pharmasset. Este tipo de empresas puede presentar ventas reducidas o pérdidas operativas elevadas antes de generar ingresos estables, lo que distorsiona los márgenes. Por tanto, los valores negativos no se eliminaron de forma automática, ya que reflejan situaciones reales de negocio.

También se revisaron los valores positivos extremos. En ROA aparecieron empresas como Sabine Royalty Trust o Marine Petroleum Trust, con rentabilidades muy superiores a la mediana. Estos casos tampoco se consideraron errores. Responden a modelos de negocio específicos, con estructuras de activos distintas a las de una compañía industrial tradicional. Lo mismo ocurre con algunas empresas con márgenes altos, donde la estructura operativa puede generar ratios superiores a los de otros sectores.

La variable ev_ebitda fue una de las más ruidosas. Presentó valores negativos muy bajos y positivos muy altos. Esto ocurre porque el múltiplo EV/EBITDA es muy sensible cuando el EBITDA es bajo, negativo o cercano a cero. En el dataset aparecieron valores extremos en empresas como Tableau Software, Trulia o Eloqua en el lado negativo, y MedImmune, Concur Technologies o ExactTarget en el lado positivo. Aunque esta variable introduce ruido, se decidió mantenerla porque EV/EBITDA es una métrica muy utilizada en valoración y M&A.

Las variables leverage y capex_intensity también presentaron casos extremos. En leverage, algunos valores elevados correspondían a empresas con niveles de deuda muy altos en relación con sus activos. En capex_intensity, algunos valores se explicaban por empresas con ventas muy reducidas y capex relativamente alto. Estos casos pueden ser poco representativos, pero no necesariamente incorrectos. Por eso, se conservaron tras comprobar que no eran errores evidentes de carga.

Las variables sich y exchg se interpretaron de forma distinta. Aunque aparecen como numéricas, realmente son códigos. sich representa el sector SIC de la empresa y exchg identifica el mercado donde cotiza. Por tanto, sus boxplots no deben interpretarse como distribuciones continuas tradicionales. En fases posteriores se mantiene esta precaución al interpretar su importancia en los modelos.

En conjunto, la revisión mostró que los valores extremos estaban justificados por el sector, el tamaño o el modelo de negocio de las empresas. No se detectaron errores suficientes como para aplicar una eliminación masiva de observaciones. Además, eliminar todos los outliers habría podido borrar información relevante, especialmente en M&A, donde algunas empresas adquiridas tienen perfiles financieros atípicos.

Variable	Patrón observado	Interpretación
at, sale, market_cap, lt, total_debt	Valores muy altos hacia la derecha	Diferencias reales de tamaño entre empresas
ebitda_margin, net_margin, roa	Valores negativos extremos	Empresas en pérdidas o en fase pre-revenue
ev_ebitda	Rango muy amplio, con valores negativos y positivos altos	Métrica sensible a EBITDA bajo o negativo
leverage	Valores elevados en empresas con alta deuda relativa	Situaciones financieras reales, no errores automáticos
capex_intensity	Valores altos cuando las ventas son reducidas	Ratio sensible en empresas pequeñas o con baja facturación
sich, exchg	Distribución amplia de códigos	Variables codificadas, no continuas

Tabla 4. Resumen de los outliers revisados.

4.4 CORRELACIONES ENTRE VARIABLES

Una vez tratados los valores nulos y revisados los outliers, se analizó la relación entre variables mediante una matriz de correlaciones. Este paso permite detectar relaciones fuertes entre variables, posibles redundancias y señales iniciales respecto a la variable objetivo.

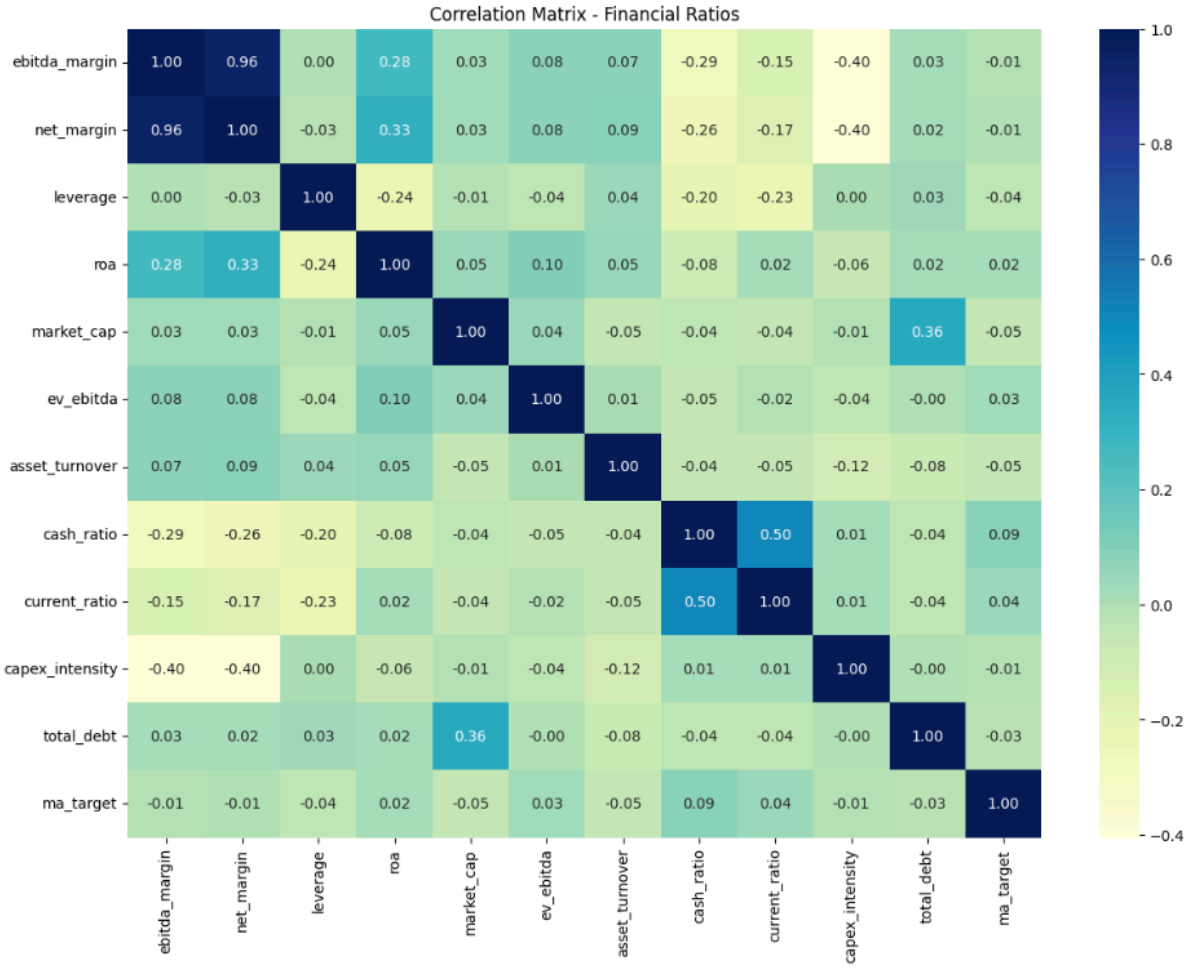


Figura 10. Matriz de correlaciones de los ratios financieros.

La matriz mostró que ninguna variable financiera tenía una correlación alta con ma_target. Esto es relevante porque indica que la condición de target no depende de una única variable de forma directa. La señal parece estar repartida entre varias características financieras. Esta conclusión encaja con la lógica del problema: una empresa no se convierte en target solo por tener mucha caja, poca deuda o una determinada rentabilidad. Lo más probable es que la probabilidad de adquisición dependa de combinaciones de variables.

Este resultado justifica el uso de modelos no lineales en fases posteriores. Modelos como Random Forest o Gradient Boosting pueden capturar interacciones entre variables que una regresión logística simple podría no detectar.

También se identificaron correlaciones esperables entre algunas variables. Por ejemplo, market_cap y total_debt presentaron una correlación positiva moderada, lo que tiene sentido porque las empresas más grandes suelen tener mayor deuda absoluta. La correlación más alta apareció entre ebitda_margin y net_margin, con un valor aproximado de 0,96. Ambas variables miden rentabilidad desde perspectivas muy cercanas. Para evitar redundancia, se eliminó net_margin y se mantuvo ebitda_margin, que resulta más relevante en un contexto de valoración y M&A.

4.5 MULTICOLINEALIDAD Y SELECCIÓN DE VARIABLES

Además de la matriz de correlaciones, se calculó el Variance Inflation Factor, o VIF, para detectar multicolinealidad. El VIF mide hasta qué punto una variable puede explicarse por el resto de variables. Valores muy altos indican que existe redundancia y que varias variables están aportando información casi idéntica (Sanz Bobi, 2024; O'Brien, 2007).

	feature	VIF
10	ebitda	inf
12	oibdp	inf
19	total_debt	6.572028e+07
5	dltt	2.011813e+07
4	dlc	1.654214e+07
2	at	4.645681e+02
6	lt	4.121579e+02
7	rect	2.298609e+01
1	ap	1.946828e+01
3	che	1.493838e+01
9	dp	9.632968e+00
15	capx	8.804863e+00
21	net_margin	7.941018e+00
20	ebitda_margin	7.762432e+00
26	market_cap	6.318855e+00
13	sale	5.235323e+00
14	xsga	3.282408e+00
16	csho	2.400045e+00
8	wcap	2.139725e+00
11	ni	1.668467e+00
29	cash_ratio	1.572201e+00
25	current_ratio	1.540687e+00
24	roa	1.266714e+00
22	leverage	1.233728e+00
23	capex_intensity	1.125692e+00
28	asset_turnover	1.088623e+00
0	exchg	1.086133e+00
17	prcc_f	1.056060e+00
18	sich	1.053111e+00
27	ev_ebitda	1.024834e+00

El análisis mostró VIF infinito para ebitda y oibdp. Esto se debe a que ambas variables eran perfectamente colineales en el dataset. Para evitar duplicidad, se eliminó oibdp y se mantuvo ebitda. También aparecieron VIFs muy elevados en total_debt, dltt y dlc. En este caso, total_debt se construye como suma de deuda a largo plazo y deuda a corto plazo. Por tanto, mantener las tres variables habría introducido información duplicada. Se decidió conservar total_debt y eliminar dltt y dlc.

Estas decisiones reducen la redundancia del dataset y simplifican el espacio de variables sin perder información financiera relevante. La selección final busca mantener variables interpretables y útiles para M&A, evitando que el modelo aprenda relaciones artificiales derivadas de duplicidades contables.

Figura 11. Tabla de los VIFs por variable.

Variable eliminada	Motivo
net_margin	Correlación muy alta con ebitda_margin
oibdp	Colinealidad perfecta con ebitda
dltt	Componente de total_debt
dlc	Componente de total_debt

Tabla 5. Variables eliminadas durante la selección final

4.6 TRANSFORMACIÓN FINAL DEL DATASET

Tras la limpieza y selección de variables, se preparó el dataset final para el modelado. Primero se eliminaron identificadores y columnas que no debían usarse como predictores, como gvkey, fyear, tic, datadate, conm y company_clean. Estas variables sirven para identificar empresas o fechas, pero no deben actuar como señales financieras dentro del modelo.

El dataset final quedó compuesto por 4.940 observaciones y 26 variables predictoras. Las variables finales incluyen indicadores de tamaño, liquidez, deuda, rentabilidad, valoración, eficiencia operativa, sector y mercado de cotización.

Las 26 variables finales son: *exchg, ap, at, che, lt, rect, wcap, dp, ebitda, ni, sale, xsga, capx, csho, prcc_f, sich, total_debt, ebitda_margin, leverage, capex_intensity, roa, current_ratio, market_cap, ev_ebitda, asset_turnover y cash_ratio*.

Después se dividió el dataset en entrenamiento y test, usando una partición 70/30. El conjunto de entrenamiento quedó formado por 3.458 observaciones y el conjunto de test por 1.482 observaciones. Esta separación permite entrenar los modelos con una parte de los datos y evaluar su rendimiento sobre empresas no utilizadas durante el aprendizaje.

Por último, se aplicó estandarización mediante StandardScaler. La estandarización transforma las variables para que tengan media 0 y desviación típica 1.

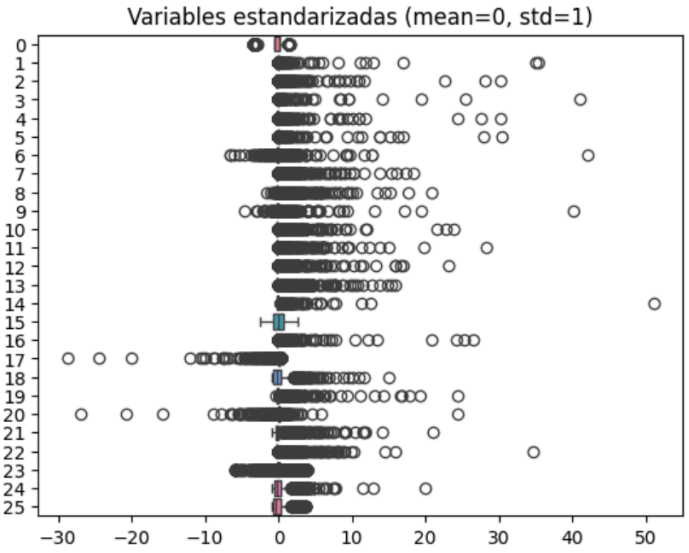


Figura 12. Boxplot de variables estandarizadas.

El resultado final de esta fase es un dataset limpio, imputado, sin duplicidades evidentes y preparado para entrenar modelos de clasificación. Este dataset se guarda junto con la partición train/test y los nombres de las variables, para asegurar que la fase de modelado sea reproducible.

Capítulo 5. ANÁLISIS Y MODELAJE

Este capítulo recoge la fase de entrenamiento, comparación y selección de modelos. A partir del dataset limpio generado en el capítulo anterior, se entrenan distintos modelos de clasificación supervisada con el objetivo de estimar la probabilidad de que una empresa sea target de M&A. La comparación se realiza con métricas de rendimiento en test, prestando especial atención al AUC-ROC, al número de targets identificados y al nivel de overfitting.

5.1 PLANTEAMIENTO DEL MODELAJE

El problema se formula como una clasificación binaria. La variable objetivo es ma_target, donde el valor 1 identifica empresas adquiridas y el valor 0 identifica empresas no adquiridas. El dataset final contiene 4.940 observaciones y 26 variables predictoras. La partición utilizada es 70 % para entrenamiento y 30 % para test (Sanz Bobi, 2024). De esta forma, el modelo aprende con 3.458 observaciones y se evalúa sobre 1.482 observaciones no utilizadas durante el entrenamiento.

Se prueban cuatro familias principales de modelos. Primero se utiliza regresión logística como modelo base. Después se entrenan modelos basados en árboles: Random Forest, Gradient Boosting y XGBoost. Esta selección permite comparar un modelo lineal sencillo con modelos no lineales capaces de capturar interacciones entre variables. Este enfoque es coherente con trabajos previos de Machine Learning aplicado a M&A, donde los modelos basados en árboles suelen ofrecer mejores resultados que los modelos lineales cuando existen relaciones no lineales entre variables financieras (Campbell et al., 2025; Beckenstrater, 2024).

5.2 REGRESIÓN LOGÍSTICA

La regresión logística se utiliza como primer modelo porque es sencilla, interpretable y adecuada para clasificación binaria. Su función dentro del proyecto es servir como benchmark. Es decir, permite comprobar si existe una señal lineal básica en las variables financieras antes de pasar a modelos más complejos.

Logit Regression Results

Dep. Variable:	y	No. Observations:	3458
Model:	Logit	Df Residuals:	3431
Method:	MLE	Df Model:	26
Date:	Tue, 09 Jun 2026	Pseudo R-squ.:	0.02719
Time:	16:54:06	Log-Likelihood:	-2093.4
converged:	True	LL-Null:	-2151.9
Covariance Type:	nonrobust	LLR p-value:	1.642e-13

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.8294	0.039	-21.307	0.000	-0.906	-0.753
exchg	-0.1609	0.039	-4.138	0.000	-0.237	-0.085
ap	0.9357	0.447	2.095	0.036	0.060	1.811
at	-1.6697	1.266	-1.319	0.187	-4.151	0.812
che	-0.2632	0.246	-1.072	0.284	-0.744	0.218
lt	1.0138	1.310	0.774	0.439	-1.554	3.582
rect	-0.4158	0.356	-1.167	0.243	-1.114	0.283
wcap	-0.0365	0.088	-0.416	0.677	-0.208	0.135
dp	0.1379	0.142	0.970	0.332	-0.141	0.416
ebitda	0.1407	0.176	0.802	0.423	-0.203	0.485
ni	0.0751	0.059	1.268	0.205	-0.041	0.191
sale	0.1933	0.116	1.670	0.095	-0.034	0.420
xsga	-0.1333	0.099	-1.352	0.177	-0.327	0.060
capx	-0.2684	0.159	-1.687	0.092	-0.580	0.043
csho	-0.0536	0.082	-0.656	0.512	-0.214	0.107
prcc_f	-0.0162	0.048	-0.335	0.737	-0.111	0.078
sich	-0.1220	0.039	-3.135	0.002	-0.198	-0.046
total_debt	0.2479	0.208	1.191	0.234	-0.160	0.656
ebitda_margin	0.0321	0.049	0.651	0.515	-0.064	0.129
leverage	-0.1227	0.047	-2.594	0.009	-0.215	-0.030
capex_intensity	-0.1366	0.064	-2.140	0.032	-0.262	-0.012
roa	0.0644	0.051	1.257	0.209	-0.036	0.165
current_ratio	-0.0697	0.045	-1.554	0.120	-0.158	0.018
market_cap	-0.1164	0.141	-0.825	0.410	-0.393	0.160
ev_ebitda	0.0863	0.038	2.283	0.022	0.012	0.160
asset_turnover	-0.1074	0.045	-2.412	0.016	-0.195	-0.020
cash_ratio	0.2326	0.046	5.103	0.000	0.143	0.322

Figura 13. Output del modelo de Regresión Logística

El modelo se entrena usando las 26 variables finales. La regresión logística obtiene un Pseudo R^2 de 0,027, lo que indica que la capacidad explicativa lineal es baja. Algunas variables aparecen como estadísticamente significativas ($p < 0,05$):

- cash_ratio (0.233, p=0.000) → mayor liquidez = más probable ser target, tiene sentido financiero
- exchg (-0.161, p=0.000) → la bolsa donde cotiza influye

- sich (-0.122, p=0.002) → el sector industrial influye
- ev_ebitda (0.086, p=0.022) → mayor valoración = más probable ser target
- leverage (-0.123, p=0.009) → menos deuda = más probable ser target
- capex_intensity (-0.137, p=0.032) → menos inversión en capex = más probable
- asset_turnover (-0.107, p=0.016) → interesante, menos eficiencia operativa = más probable
- ap (0.936, p=0.036) → más cuentas a pagar = más probable

Desde un punto de vista financiero, varias de estas relaciones tienen sentido. Por ejemplo, una mayor liquidez puede hacer que una empresa sea más atractiva como target, mientras que un mayor apalancamiento puede reducir su atractivo.

Sin embargo, el rendimiento predictivo del modelo es limitado. En entrenamiento obtiene una accuracy del 68,42 %. En test obtiene una accuracy del 68,29 % y un AUC-ROC de 0,5602. Aunque la accuracy parece aceptable a primera vista, en realidad es poco informativa. Debido al desbalanceo de clases, un modelo que clasificara casi todo como no target podría alcanzar una accuracy parecida.

La matriz de confusión confirma este problema. De los 466 targets reales del conjunto de test, la regresión logística solo identifica correctamente 11. El resto se clasifican como no targets. Por tanto, el recall de targets es muy bajo y el modelo no sirve como herramienta de screening.

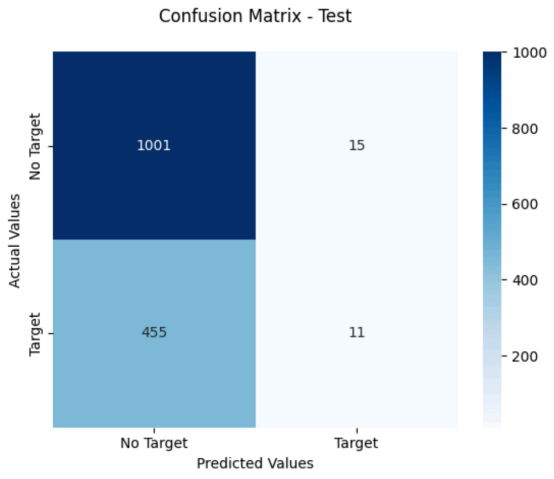


Figura 14. Matriz de confusión de la regresión logística en test.

La conclusión de este primer modelo es clara. La regresión logística apenas mejora a un clasificador aleatorio en términos de AUC-ROC y no detecta prácticamente ningún target. Esto encaja con lo observado en el EDA: ninguna variable individual presentaba una correlación fuerte con ma_target.

La señal parece depender de combinaciones de variables, por lo que tiene sentido pasar a modelos no lineales.

5.3 RANDOM FOREST

5.3.1 Modelo base

El siguiente modelo entrenado es Random Forest. Este algoritmo combina muchos árboles de decisión entrenados sobre muestras aleatorias del dataset y promedia sus predicciones. Su ventaja principal es que puede capturar relaciones no lineales e interacciones entre variables, reduciendo la varianza respecto a un único árbol de decisión (Breiman, 2001).

El primer Random Forest se entrena con 100 árboles, max_depth sin límite y max_features igual a la raíz cuadrada del número de variables. Como el dataset tiene 26 features, se utilizan aproximadamente 5 variables por árbol, criterio habitual en clasificación con Random Forest.

El modelo base mejora claramente a la regresión logística. Como se puede ver en la Figura 15, en test obtiene una accuracy del 71,12 % y un AUC-ROC de 0,7136. Además, identifica correctamente 134 targets, frente a los 11 identificados por la regresión logística. Esto muestra que el modelo ya está capturando una señal no lineal relevante.

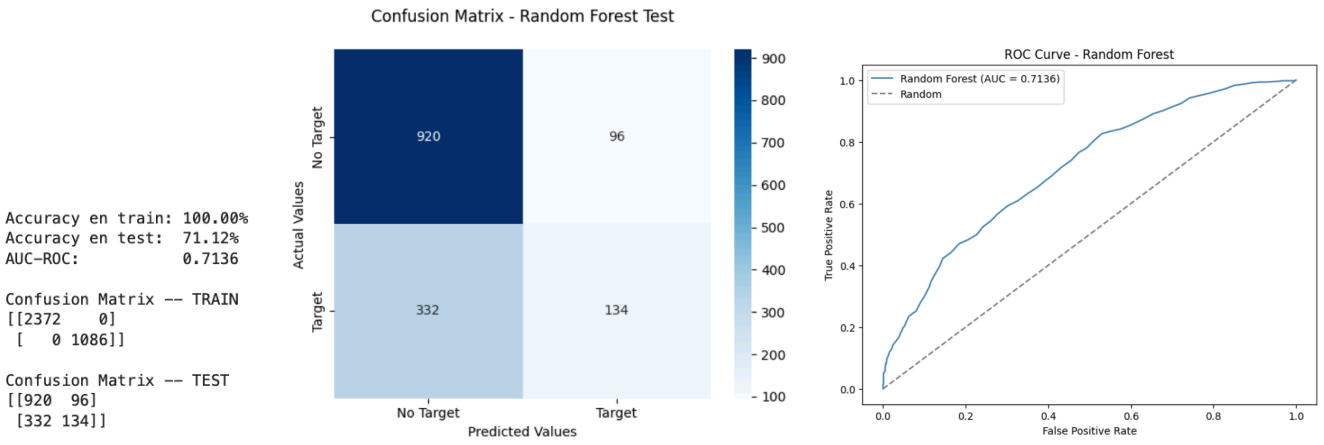


Figura 15. Matriz de confusión y ROC del Random Forest base.

Sin embargo, también aparece un problema claro de overfitting. En entrenamiento alcanza una accuracy del 100 %, mientras que en test se queda en torno al 71 %. El modelo memoriza perfectamente el conjunto de entrenamiento, pero no generaliza igual de bien a empresas nuevas.

La importancia de variables del modelo base muestra que las variables más relevantes son market_cap, ev_ebitda, che, cash_ratio y csho. Todas ellas tienen sentido financiero. Están relacionadas con tamaño, valoración y liquidez, factores que pueden influir en el atractivo de una empresa como target. Además, la importancia está bastante repartida entre variables, lo que sugiere que el modelo no depende de una única métrica.

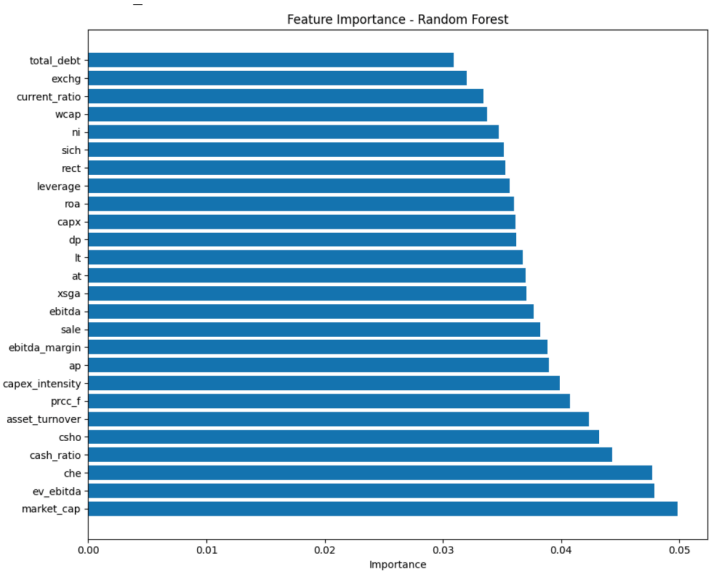


Figura 16. Random Forest baseline feature importance.

5.3.2 Ajuste de hiperparámetros con GridSearchCV

Para intentar reducir el overfitting, se aplica GridSearchCV con validación cruzada estratificada de 5 folds. Se prueban distintas combinaciones de n_estimators, max_features, max_depth, min_samples_leaf y criterion. La métrica de optimización es AUC-ROC, ya que el dataset está desbalanceado y el objetivo es evaluar la capacidad de separación entre targets y no targets.

El mejor modelo encontrado utiliza criterion = entropy, max_depth = None, max_features = 7, min_samples_leaf = 1 y n_estimators = 150. En validación cruzada alcanza un AUC-ROC de 0,7251. Al evaluarlo sobre el test, obtiene una accuracy del 70,45 % y un AUC-ROC de 0,7224. Identifica 128 targets.

Aunque el AUC mejora respecto al Random Forest base, el overfitting no desaparece. La accuracy en entrenamiento sigue siendo del 100 %. Esto indica que la búsqueda de hiperparámetros no soluciona

por sí sola el problema. El modelo sigue siendo muy flexible y aprende demasiado bien el conjunto de entrenamiento.

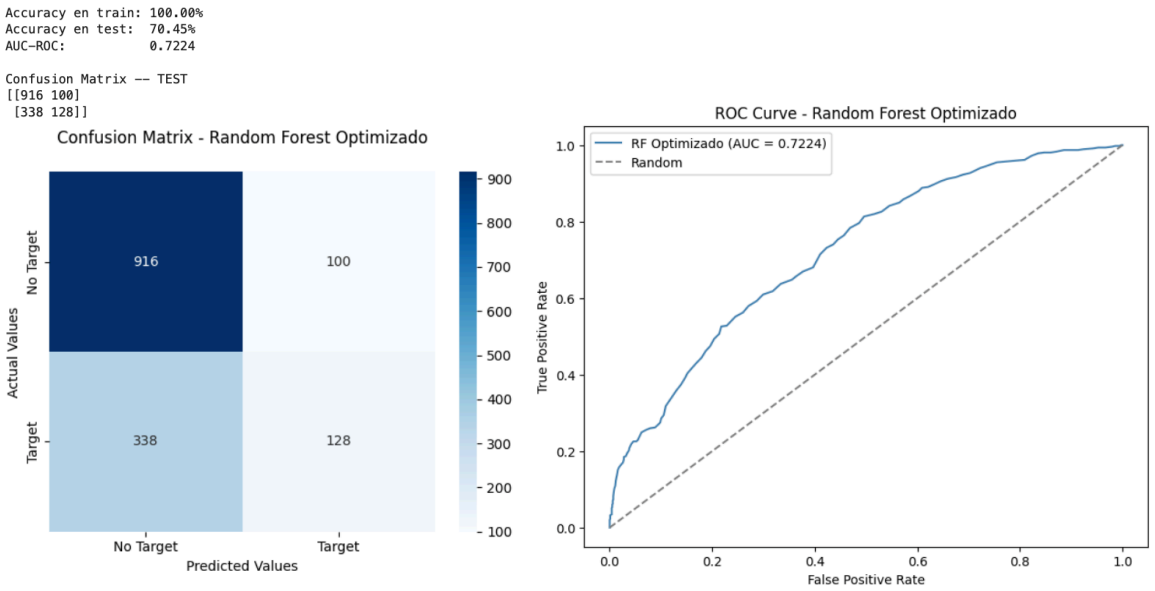


Figura 17. Resultados modelo RF optimizado.

5.3.3 Random Forest con class_weight balanced

El siguiente paso consiste en introducir class_weight = balanced. Esta opción penaliza más los errores cometidos sobre la clase minoritaria. En este caso, los targets reciben un peso mayor que los no targets, porque aparecen con menor frecuencia en el dataset. Aplicando la fórmula:

$$peso = \frac{total\ filas}{nº\ clases \cdot filas\ en\ esa\ clase},$$

se obtiene aproximadamente un peso de 0,73 para los no targets y de 1,59 para los targets. Esto implica que un error al clasificar un target como no target tiene más del doble de impacto que el error contrario. Esta estrategia se utiliza habitualmente en problemas desbalanceados para que el modelo preste más atención a la clase positiva (He & Garcia, 2009).

El Random Forest balanceado obtiene una accuracy del 71,46 % en test y un AUC-ROC de 0,7204. Identifica correctamente 144 targets, más que el modelo optimizado con GridSearch. Sin embargo, también mantiene una accuracy del 100 % en entrenamiento. Por tanto, class_weight mejora ligeramente la detección de targets, pero no controla el overfitting.

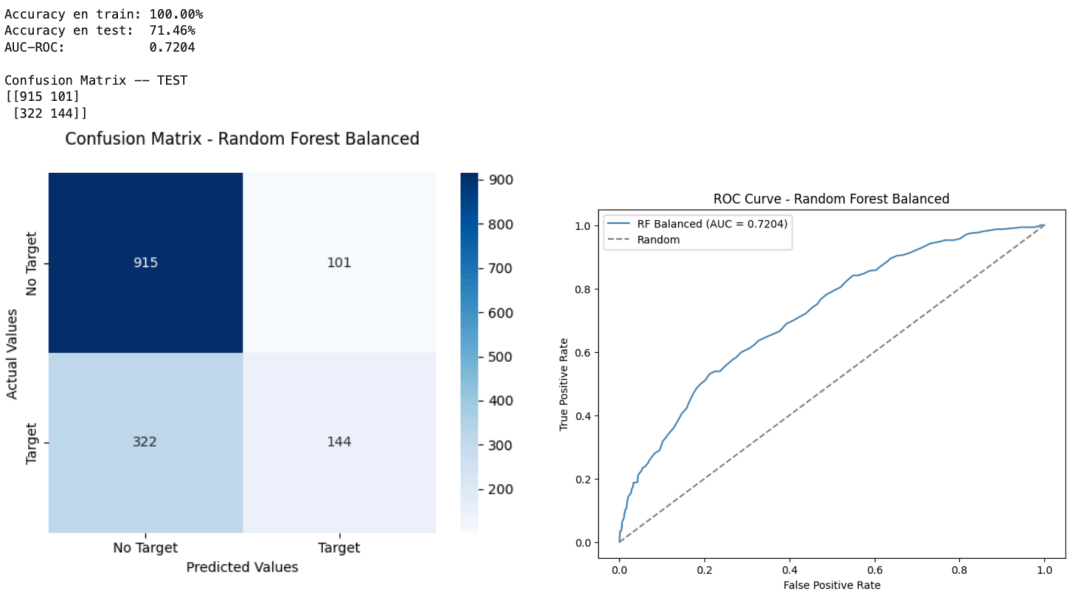


Figura 18. Resultados RF balanceado.

5.3.4 Regularización forzada

Como el GridSearch no limitó la profundidad de los árboles, se prueba una regularización manual. El objetivo es controlar la complejidad del modelo y reducir la diferencia entre train y test. Para ello se exploran tres alternativas: limitar max_depth, aumentar min_samples_leaf y combinar ambas restricciones.

- Limitar max_depth:

```
max_depth=3 | Train: 68.6% | Test: 68.6% | AUC: 0.6906
max_depth=5 | Train: 71.1% | Test: 69.6% | AUC: 0.7057
max_depth=10 | Train: 85.3% | Test: 71.1% | AUC: 0.7232
max_depth=15 | Train: 99.4% | Test: 71.3% | AUC: 0.7246
max_depth=20 | Train: 100.0% | Test: 71.2% | AUC: 0.7271
```

Figura 19. Prueba y error limitando max_depth.

Cuando simplificas el modelo (menos profundidad, más muestras en hoja), reduces overfitting pero también baja el AUC. En cambio cuando lo dejas complejo, mejora el AUC pero el accuracy en train es de 100%, lo cual indica overfitting. Por lo tanto, el punto de equilibrio está en max_depth=10. Probamos ajustar el modelo:

```
Accuracy en train: 85.25%
Accuracy en test:  71.05%
AUC-ROC:           0.7232

Confusion Matrix -- TEST
[[947  69]
 [360 106]]
```

Figura 20. Resultados regularización por max_depth.

- Aumentar min_samples_leaf:

```
min_samples_leaf=1 | Train: 100.0% | Test: 70.4% | AUC: 0.7224
min_samples_leaf=5 | Train: 95.3% | Test: 70.7% | AUC: 0.7242
min_samples_leaf=10 | Train: 87.7% | Test: 71.1% | AUC: 0.7203
min_samples_leaf=20 | Train: 80.5% | Test: 70.0% | AUC: 0.7120
min_samples_leaf=50 | Train: 74.7% | Test: 70.4% | AUC: 0.7059
```

Figura 21. Prueba y error para aumentar min_samples_leaf.

Se exige un mínimo de 5 observaciones en cada nodo hoja para reducir la complejidad del modelo. Esto evita que los árboles creen ramas muy específicas basadas en pocos ejemplos, que es una de las causas principales del overfitting en Random Forest. Al aumentar min_samples_leaf a 5, el modelo obtiene una accuracy del 95,32 % en entrenamiento, 70,72 % en test y AUC-ROC de 0,7242. Identifica 122 targets, pero el overfitting sigue siendo alto.

```
Accuracy en train: 95.32%
Accuracy en test:  70.72%
AUC-ROC:           0.7242

Confusion Matrix -- TEST
[[926  90]
 [344 122]]
```

Figura 22. Resultados regulariación por min_samples_leaf.

- Combinación:

  Por último, la combinación de max_depth = 10 y min_samples_leaf = 5 reduce más el overfitting, con 80,51 % en train y 70,85 % en test, pero baja el número de targets identificados a 101. Se combinan ambas restricciones simultáneamente para aplicar una regularización más agresiva. La combinación actúa desde dos ángulos, limitando tanto la profundidad global del árbol como la especificidad de sus nodos finales, buscando el mejor equilibrio posible entre bias y variance.

```
Accuracy en train: 80.51%
Accuracy en test:  70.85%
AUC-ROC:           0.7204

Confusion Matrix -- TEST
[[949  67]
 [365 101]]
```

Figura 23. Resultados regularización combinada.

Ninguna opción es claramente superior en AUC. La combinación reduce más el overfitting, pero pierde capacidad de detección de targets. El modelo con min_samples_leaf = 5 identifica más targets, pero sigue sobreajustando bastante. Por equilibrio entre rendimiento y generalización, se toma max_depth = 10 como versión regularizada de referencia para Random Forest.

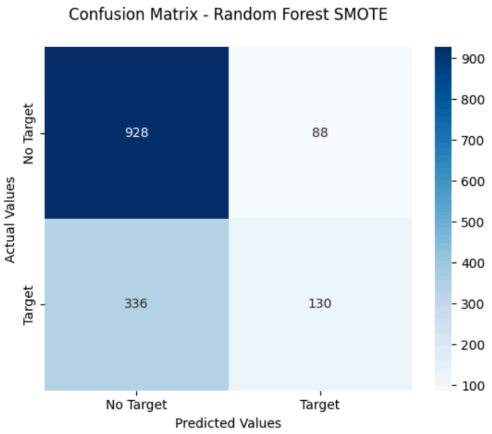
5.3.5 Random Forest con SMOTE

Después se prueba SMOTE, una técnica de oversampling que genera ejemplos sintéticos de la clase minoritaria en lugar de duplicar observaciones existentes (Chawla et al., 2002). En este proyecto se aplica únicamente sobre el conjunto de entrenamiento. El test no se modifica, porque debe representar el problema real y permitir una evaluación limpia.

Se utiliza sampling_strategy = 0,5. Esto significa que, después de aplicar SMOTE, la clase target queda con la mitad de observaciones que la clase no target en el conjunto de entrenamiento. El número de targets en train pasa de 1.086 a 1.186, mientras que los no targets se mantienen en 2.372. Es una aplicación conservadora de SMOTE, ya que solo genera 100 observaciones sintéticas.

El Random Forest con SMOTE mantiene max_depth = 10 para controlar el overfitting. El resultado es una accuracy del 86,73 % en entrenamiento, 71,39 % en test y AUC-ROC de 0,7260. Identifica correctamente 130 targets y genera 88 falsos positivos.

```
Before SMOTE, count label '1': 1086
Before SMOTE, count label '0': 2372

After SMOTE, shape X_train: (3558, 26)
After SMOTE, shape y_train: (3558,)
After SMOTE, count label '1': 1186
After SMOTE, count label '0': 2372

Accuracy en train: 86.73%
Accuracy en test:  71.39%
AUC-ROC:           0.7260

Confusion Matrix -- TEST
[[928  88]
 [336 130]]
```



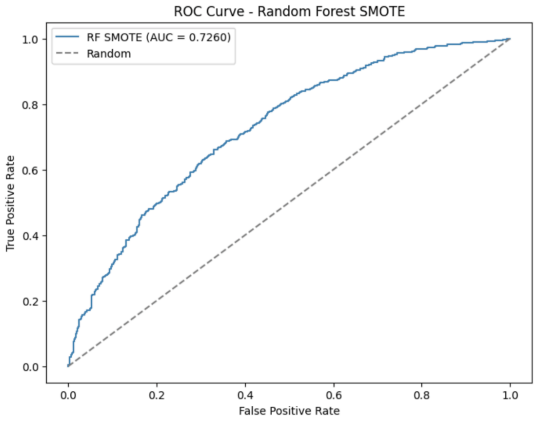


Figura 24. Resultados RF con SMOTE.

Esta variante obtiene el mejor AUC dentro de los modelos Random Forest y mantiene un overfitting moderado. Aunque identifica menos targets que la versión balanced, generaliza mejor y presenta una matriz de confusión más equilibrada. Por eso, se considera la mejor variante de Random Forest.

5.4 GRADIENT BOOSTING

5.4.1 Modelo base

Una vez explorado Random Forest, se prueba Gradient Boosting. La diferencia principal es que Random Forest entrena árboles independientes en paralelo, mientras que Gradient Boosting construye árboles de forma secuencial. Cada nuevo árbol intenta corregir los errores cometidos por los anteriores (Friedman, 2001). Esta lógica puede ser útil en este proyecto porque los targets son precisamente los casos difíciles que los modelos anteriores no siempre detectan.

El Gradient Boosting base se entrena con 100 estimadores y max_features = sqrt. En test obtiene una accuracy del 70,78 % y un AUC-ROC de 0,7083. Identifica 106 targets. Su rendimiento es inferior al Random Forest con SMOTE. A cambio, presenta menos overfitting, con una accuracy de entrenamiento del 77,10 % frente al 70,78 % en test.

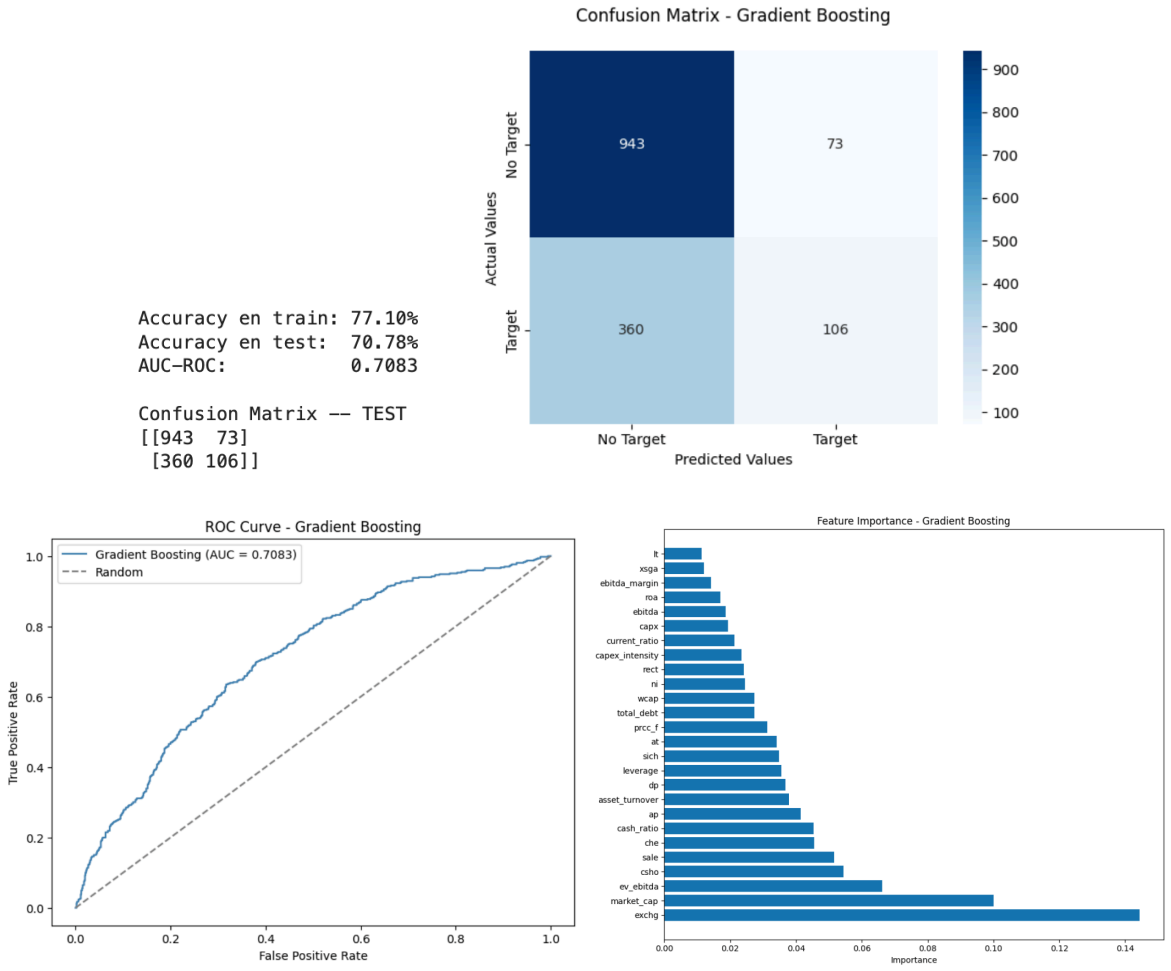


Figura 25. Resultados Gradient Boosting base

La importancia de variables muestra un comportamiento llamativo: exchg aparece como la variable más importante, con un peso mucho mayor que el resto. Esta variable representa el mercado donde cotiza la empresa. Aunque no debe interpretarse como una causa directa de adquisición, puede estar capturando información real sobre el tipo de compañía. Por ejemplo, las empresas cotizadas en NASDAQ o NYSE suelen tener mayor liquidez, mayor visibilidad y más seguimiento por parte de analistas que empresas en mercados OTC.

Para comprobar si exchg estaba distorsionando el modelo, se entrenó una versión eliminando esta variable. El resultado empeoró: el AUC-ROC bajó a 0,7007 y los targets identificados pasaron de 106 a 98. Por tanto, se decide mantener exchg, pero interpretarlo con cautela como una variable de contexto de mercado, no como una variable causal.

5.4.2 Gradient Boosting optimizado

Como segunda opción, se aplica GridSearchCV con la variable exchg, para ajustar los hiperparámetros del Gradient Boosting. Se prueban combinaciones de n_estimators, learning_rate y max_depth, manteniendo max_features = sqrt. El mejor modelo utiliza learning_rate = 0,01, max_depth = 7 y n_estimators = 300.

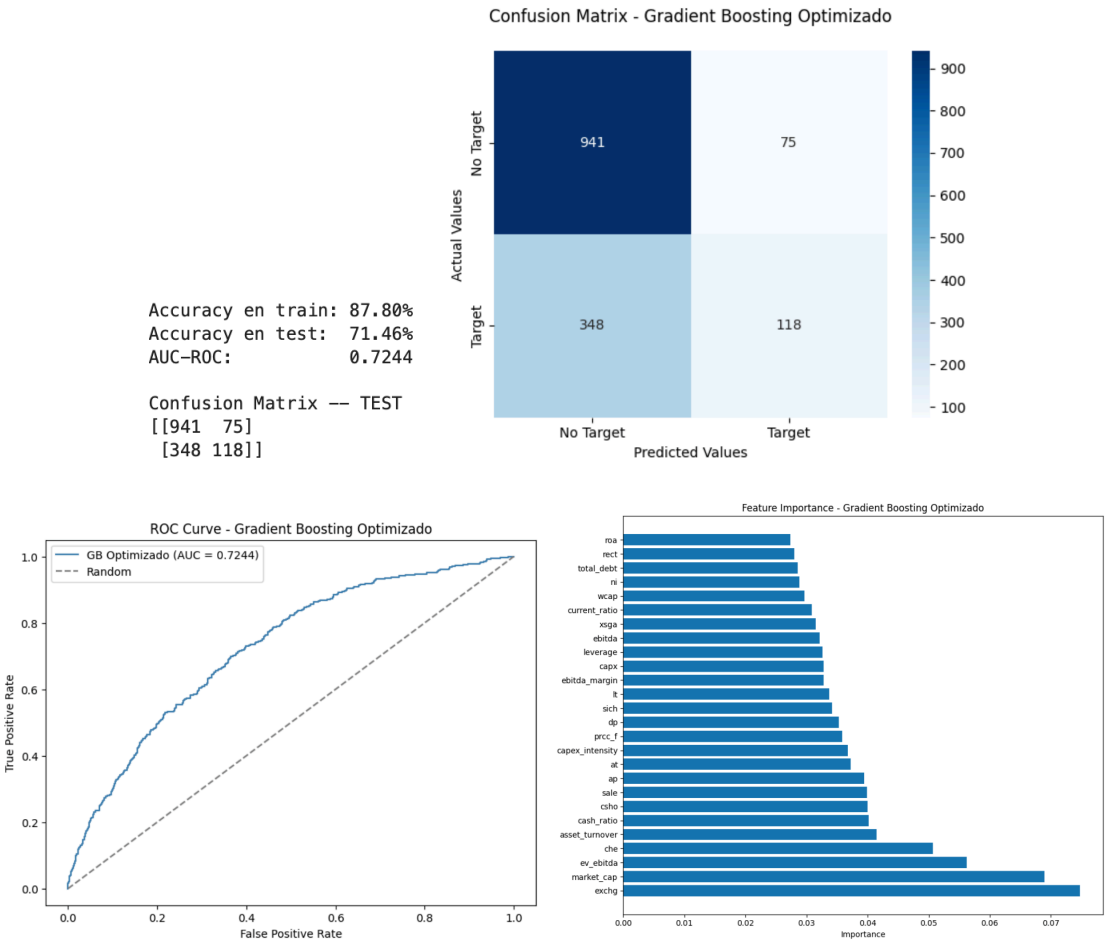


Figura 26. Resultados Gradient Boosting optimizado.

Con estos parámetros, el modelo obtiene una accuracy del 87,80 % en entrenamiento, 71,46 % en test y AUC-ROC de 0,7244. Identifica 118 targets. La mejora respecto al modelo base es clara. Además, la importancia de variables queda más repartida. exchg sigue apareciendo como la variable más relevante, pero su peso baja respecto al modelo base. Le siguen market_cap y ev_ebitda.

El modelo optimizado queda muy cerca del Random Forest con SMOTE en AUC, aunque identifica menos targets. La diferencia entre ambos modelos es pequeña, pero Random Forest con SMOTE sigue siendo ligeramente superior en capacidad discriminativa.

5.4.3 Gradient Boosting con SMOTE

También se prueba Gradient Boosting con SMOTE, usando los mejores hiperparámetros del modelo anterior. Igual que en Random Forest, SMOTE se aplica solo sobre el conjunto de entrenamiento.

El modelo obtiene una accuracy del 89,54 % en entrenamiento, 71,79 % en test y AUC-ROC de 0,7253. Identifica 142 targets. Este es el mejor modelo hasta este punto en número de targets detectados, y también el que obtiene mayor accuracy en test. Sin embargo, el AUC-ROC sigue siendo ligeramente inferior al de Random Forest con SMOTE.

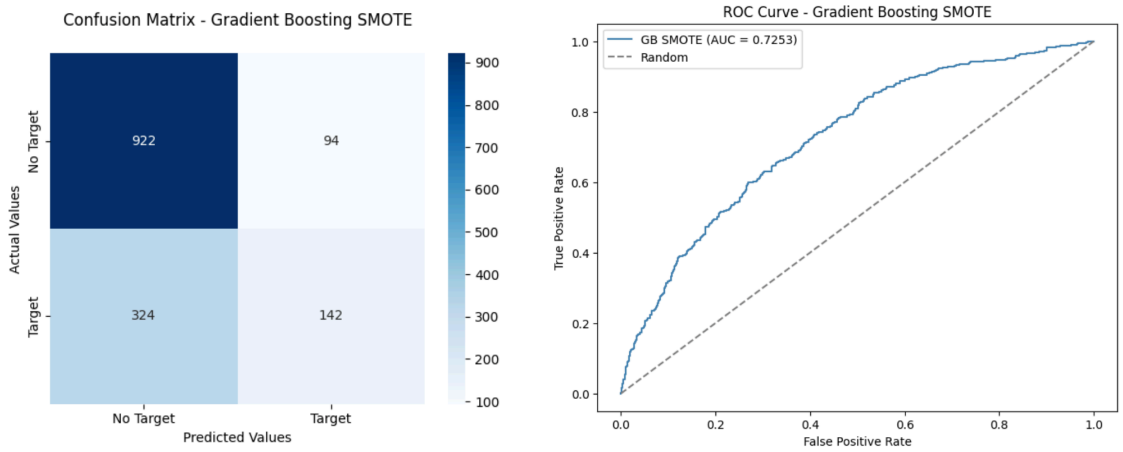


Figura 27. Resultados Gradient Boosting con SMOTE

La comparación muestra un trade-off. Gradient Boosting con SMOTE identifica más targets que Random Forest con SMOTE, pero no supera su AUC. Como el AUC evalúa la capacidad de separación del modelo a distintos umbrales, se mantiene Random Forest con SMOTE como candidato principal, aunque la diferencia entre ambos es muy reducida.

5.5 XGBOOST

5.5.1 Modelo base

El último algoritmo probado es XGBoost. Este modelo es una implementación optimizada de Gradient Boosting que incorpora regularización L1 y L2, paralelización y mayor control sobre la complejidad del modelo (Chen & Guestrin, 2016). En teoría, estas características podrían ayudar a controlar el overfitting observado en algunos modelos anteriores.

El XGBoost base obtiene una accuracy del 95,40 % en entrenamiento, 71,05 % en test y AUC-ROC de 0,7162. Identifica 159 targets, el mayor número hasta ese momento. Sin embargo, también muestra un overfitting alto, con una diferencia de más de 24 puntos entre train y test.

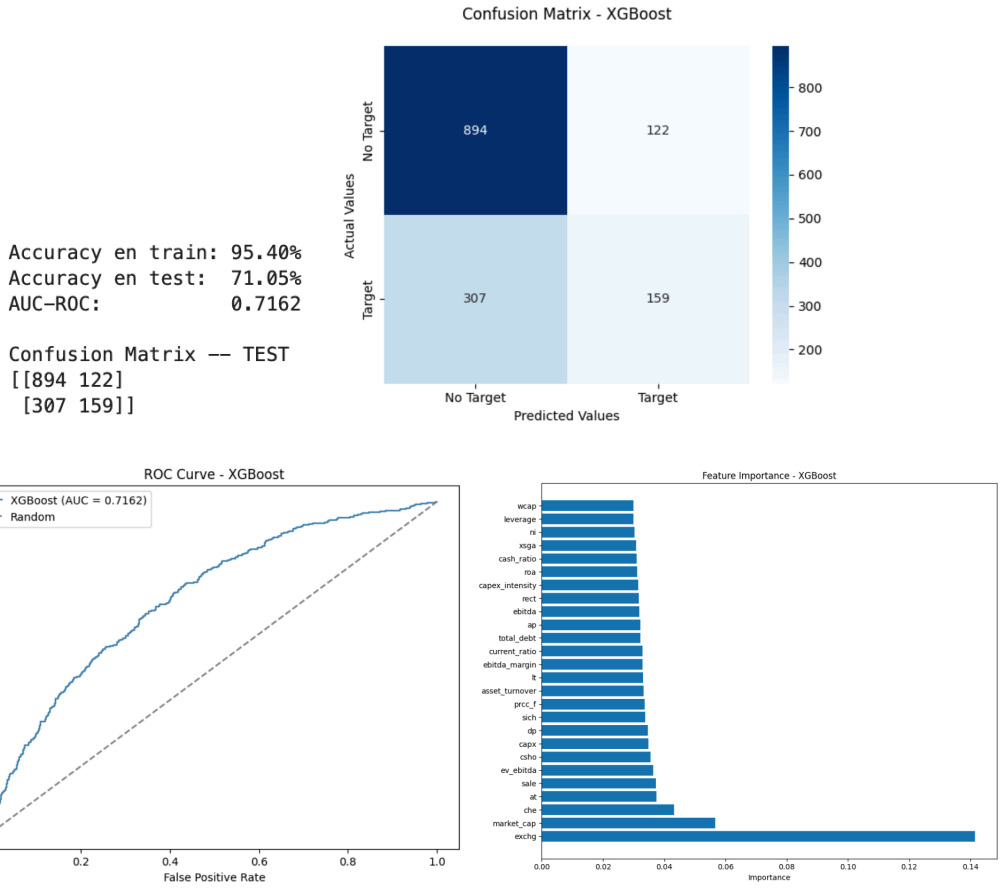


Figura 28. Resultados XGBoost base.

El comportamiento del modelo confirma que XGBoost es más agresivo identificando targets. Esto aumenta el recall, pero también aumenta los falsos positivos y reduce la estabilidad general del modelo. Además, el AUC queda por debajo del Random Forest con SMOTE y del Gradient Boosting con SMOTE.

También se revisó de nuevo la variable exchg, porque volvía a aparecer como muy importante. Al analizar la proporción de targets por mercado, se observó que las empresas en NASDAQ y NYSE tenían una proporción de targets claramente superior a las empresas OTC. En concreto, NASDAQ presentaba un 40,7 % de empresas adquiridas y NYSE un 34,9 %, frente al 13,6 % de OTC/Pink Sheets. Por tanto, exchg se mantiene en el modelo como variable informativa, aunque su interpretación se hace con cautela.

5.5.2 XGBoost optimizado

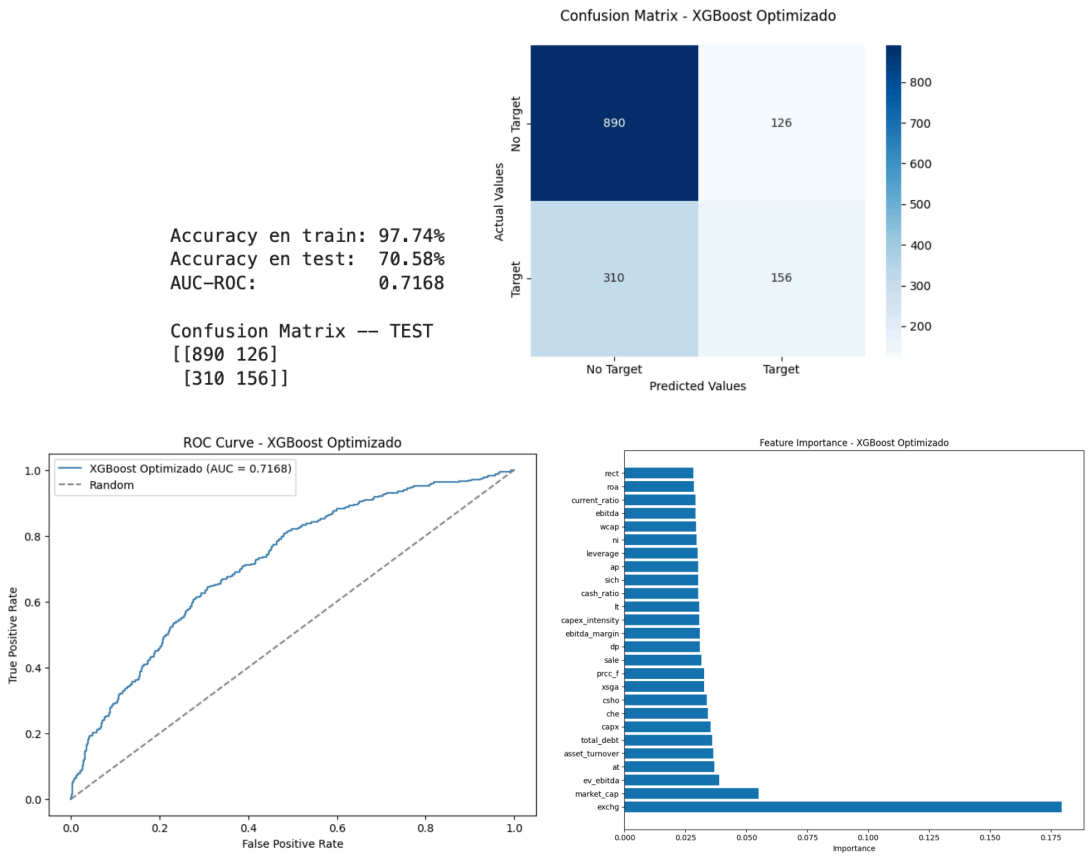


Figura 29. Resultados XGBoost optimizado.

Después se aplica GridSearchCV para ajustar n_estimators, max_depth, learning_rate, subsample y colsample_bytree. El mejor modelo utiliza colsample_bytree = 0,8, learning_rate = 0,05, max_depth = 7, n_estimators = 200 y subsample = 1,0.

El modelo optimizado obtiene una accuracy del 97,74 % en entrenamiento, 70,58 % en test y AUC-ROC de 0,7168. Identifica 156 targets. Aunque el AUC mejora ligeramente respecto al XGBoost base, el overfitting aumenta. Por tanto, el GridSearch no aporta una mejora suficiente.

5.5.3 XGBoost con SMOTE

Por último, se entrena XGBoost con SMOTE. Este modelo obtiene una accuracy del 98,12 % en entrenamiento, 70,24 % en test y AUC-ROC de 0,7166. Identifica 167 targets, el máximo de todos los modelos probados. Sin embargo, también presenta el mayor overfitting.

```
Accuracy en train: 98.12%
Accuracy en test:  70.24%
AUC-ROC:           0.7166

Confusion Matrix -- TEST
[[874 142]
 [299 167]]
```

Figura 30. Resultados XGBoost con SMOTE.

XGBoost identifica más targets que el resto de modelos, pero lo hace a costa de una pérdida de generalización. Para una herramienta de screening, identificar más targets es positivo, pero no si el modelo se ajusta demasiado al histórico y pierde fiabilidad sobre datos nuevos. Por este motivo, XGBoost no se selecciona como modelo final.

5.6 LIGHTGBM

5.6.1 Modelo base

El primer modelo LightGBM se entrenó con 100 estimadores, learning_rate = 0,1, num_leaves = 31, subsample = 0,8 y colsample_bytree = 0,8. El modelo base obtiene una accuracy del 97,48 % en entrenamiento y del 70,78 % en test, con un AUC-ROC de 0,7137.

La matriz de confusión muestra que el modelo identifica correctamente 165 de los 466 targets reales del conjunto de test. Esto supone un recall del 35,4 %, superior al del Random Forest con SMOTE.

Sin embargo, también genera 132 falsos positivos y presenta una diferencia muy elevada entre train y test. Por tanto, aunque detecta bastantes targets, el modelo base muestra un overfitting alto.

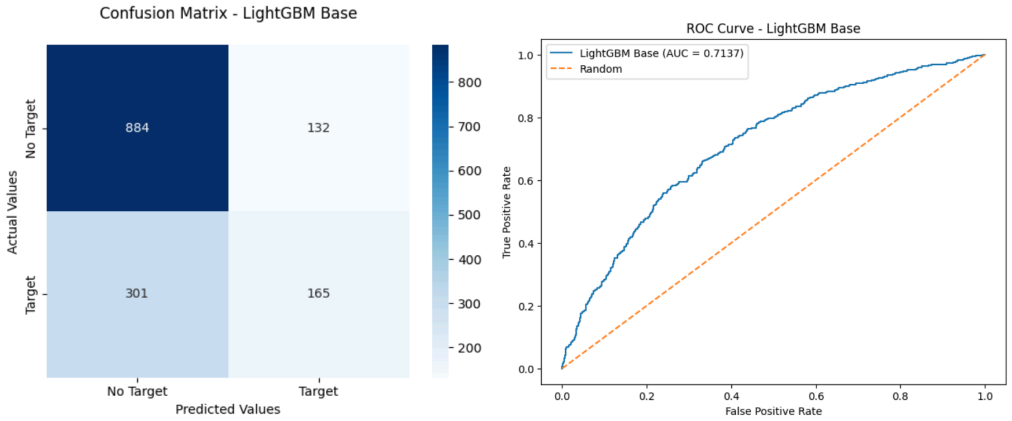


Figura 31. Resultados Modelo base LightGBM

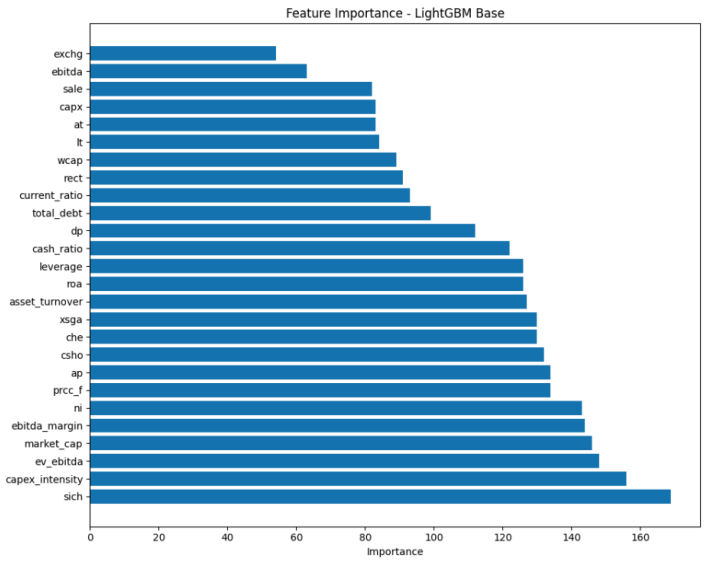


Figura 32. Variables de importancia del modelo LightGBM.

Asimismo, la Figura 31 muestra que las variables más relevantes son sich, capex_intensity, ev_ebitda, market_cap, ebitda_margin, ni, prcc_f y ap. Estas variables recogen información relacionada con sector, intensidad inversora, valoración, tamaño, rentabilidad y precio de mercado. La presencia de sich como variable más importante debe interpretarse con cautela, ya que no es una variable financiera continua, sino un código sectorial. Por tanto, indica que el sector puede estar aportando información relevante al modelo, pero no debe interpretarse como una relación causal directa.

Un aspecto interesante es que, a diferencia de lo observado en otros modelos del proyecto (Figura 26 y Figura 27), la variable exchg no aparece entre las más importantes. En modelos anteriores, exchg se situaba de forma recurrente entre las variables con mayor peso, junto con otras como market_cap, ev_ebitda, che, asset_turnover, cash_ratio, sale o ap. Estas variables reflejaban principalmente tamaño, valoración, liquidez y eficiencia operativa, y aparecían de forma bastante consistente en Random Forest, Gradient Boosting y XGBoost. Sin embargo, en este caso su importancia es mucho menor. Esto indica que LightGBM está capturando patrones distintos o redistribuyendo la relevancia entre otras variables, lo que refuerza la idea de que la interpretación de la importancia de variables depende del modelo utilizado y no es completamente estable entre algoritmos.

5.6.2 LightGBM optimizado

Después se aplica GridSearchCV para ajustar los hiperparámetros del modelo. La búsqueda selecciona colsample_bytree = 0,8, learning_rate = 0,05, max_depth = -1, n_estimators = 200, num_leaves = 15 y subsample = 0,8. El mejor AUC-ROC medio en validación cruzada es 0,7264.

Al evaluar el modelo optimizado sobre el conjunto de test, se obtiene una accuracy del 70,04 % y un AUC-ROC de 0,7194. La accuracy en entrenamiento baja a 87,59 %, por lo que el overfitting se reduce claramente respecto al modelo base. Sin embargo, esta regularización también reduce la capacidad de detección de targets: el modelo identifica 138 targets, frente a los 165 del LightGBM base.

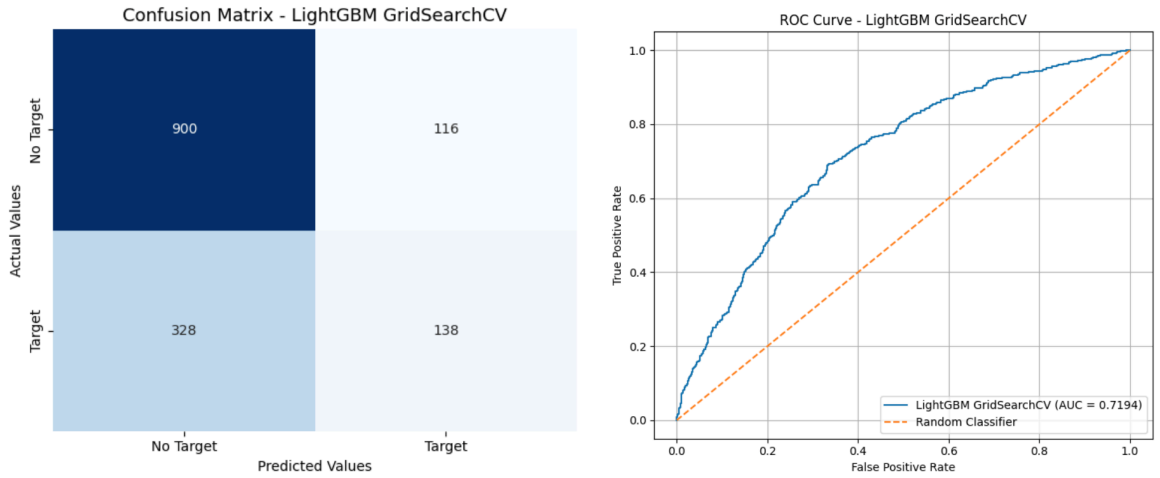


Figura 33. Resultados modelo LightGBM optimizado.

El modelo optimizado mejora ligeramente el AUC respecto al LightGBM base, pero no consigue superar a las mejores variantes de Random Forest y Gradient Boosting. Además, el número de targets identificados queda por debajo de otras alternativas.

5.6.3 LightGBM con class_weight balanced

También se prueba una versión con class_weight = balanced. Esta variante da más peso a la clase minoritaria, es decir, a las empresas target, con el objetivo de reducir los falsos negativos y aumentar el recall.

El resultado cambia de forma importante el comportamiento del modelo. La accuracy en entrenamiento es del 85,89 %, la accuracy en test baja al 66,73 % y el AUC-ROC queda en 0,7157. La matriz de confusión muestra que esta variante identifica 297 de los 466 targets reales, lo que equivale a un recall del 63,7 %. Es, por tanto, el modelo que más targets detecta entre todas las variantes de LightGBM.

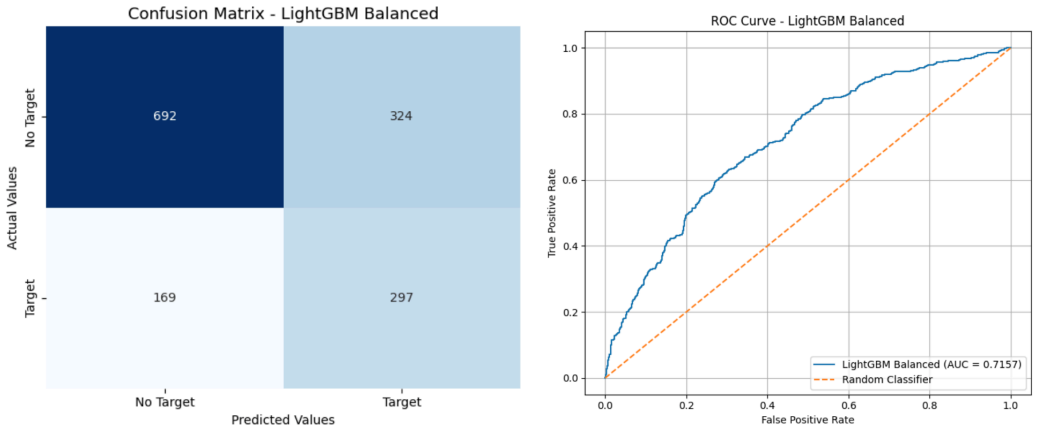


Figura 34. Resultados modelo LightGBM balanceado.

Sin embargo, esta mejora en recall se consigue a costa de aumentar mucho los falsos positivos. El modelo clasifica erróneamente como targets a 324 empresas que en realidad no fueron adquiridas. Para una herramienta de screening, detectar más targets puede ser positivo, pero si el número de falsos positivos es demasiado alto, el ranking pierde utilidad práctica porque genera demasiado ruido para el analista ya que acabaría analizando empresas que realmente no tenían perfil target. Por este motivo, aunque LightGBM balanced maximiza la detección de targets, no se considera una opción equilibrada para el modelo final.

5.6.4 LightGBM con SMOTE

Por último, se entrena LightGBM aplicando SMOTE sobre el conjunto de entrenamiento. Igual que en los modelos anteriores, el conjunto de test no se modifica para evitar data leakage.

Esta versión obtiene una accuracy del 87,55 % en entrenamiento, una accuracy del 70,72 % en test y un AUC-ROC de 0,7198. Identifica correctamente 166 targets y genera 134 falsos positivos. En comparación con el modelo base, mantiene prácticamente el mismo número de targets identificados, pero reduce el overfitting de forma clara. En comparación con el modelo optimizado, mejora tanto el AUC como el número de targets detectados.

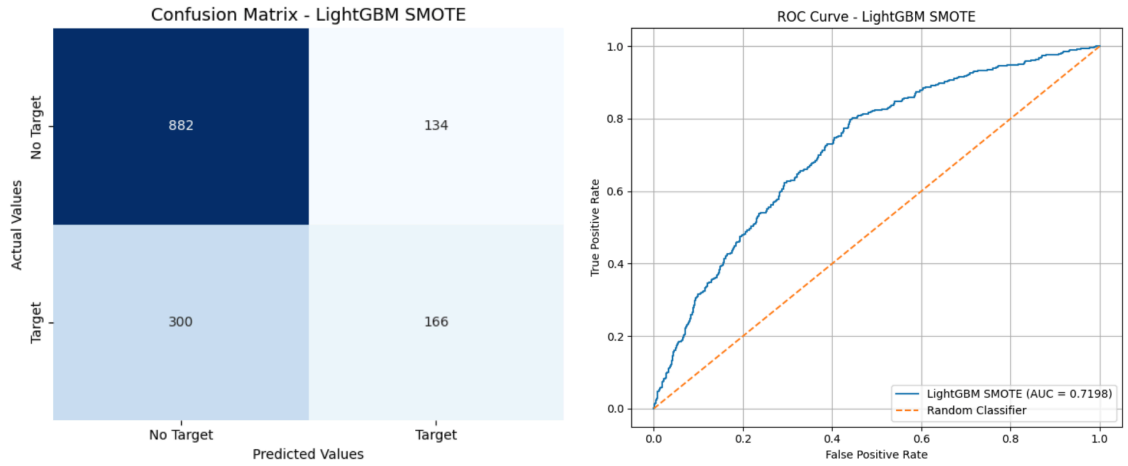


Figura 35. Resultados modelo LightGBM con SMOTE.

Capítulo 6. ANÁLISIS DE RESULTADOS

Este capítulo presenta los resultados obtenidos tras el entrenamiento de los modelos. Primero se comparan los distintos modelos, después se justifica la elección del modelo final y se analiza por qué es el más adecuado para el objetivo del proyecto. Por último, se describe cómo el modelo seleccionado se integra en una herramienta visual tipo dashboard para facilitar el screening de empresas y la interpretación de los resultados.

6.1 COMPARACIÓN FINAL DE MODELOS

Modelo	Accuracy train	Accuracy test	AUC-ROC	Targets identificados	Overfitting
Regresión logística	68,42 %	68,29 %	0,5602	11	No
Random Forest baseline	100,00 %	71,12 %	0,7136	134	Severo
Random Forest GridSearchCV	100,00 %	70,45 %	0,7224	128	Severo
Random Forest regularizado	85,25 %	71,05 %	0,7232	106	Moderado
Random Forest balanced	100,00 %	71,46 %	0,7204	144	Severo
Random Forest SMOTE	86,73 %	71,39 %	0,7260	130	Moderado
Gradient Boosting baseline	77,10 %	70,78 %	0,7083	106	Bajo
Gradient Boosting GridSearchCV	87,80 %	71,46 %	0,7244	118	Moderado
Gradient Boosting SMOTE	89,54 %	71,79 %	0,7253	142	Moderado
XGBoost baseline	95,40 %	71,05 %	0,7162	159	Alto
XGBoost GridSearchCV	97,74 %	70,58 %	0,7168	156	Alto
XGBoost SMOTE	98,12 %	70,24 %	0,7166	167	Alto
LightGBM base	97,48%	70,78%	0,7137	165	Alto
LightGBM GridSearchCV	87,59%	70,04%	0,7194	138	Moderado
LightGBM balanced	85,89%	66,73%	0,7157	297	Bajo
LightGBM SMOTE	87,55%	70,72%	0,7198	166	Moderado

Tabla 6. Comparativa final de modelos.

La Tabla 7 resume los principales modelos entrenados. Se comparan accuracy en train, accuracy en test, AUC-ROC, número de targets identificados y nivel de overfitting.

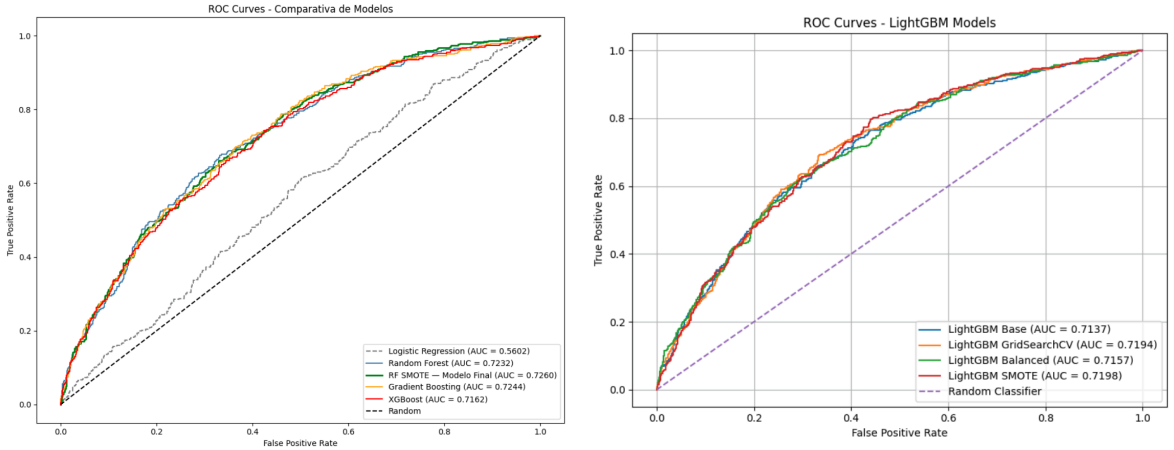


Figura 36. Comparativa de curvas ROC de los principales modelos.

La comparación muestra varias conclusiones. En primer lugar, la regresión logística queda claramente por debajo del resto de modelos. Su AUC-ROC es 0,5602 y solo identifica 11 targets, por lo que no resulta adecuada para una herramienta de screening. Esto confirma que la señal del dataset no está en relaciones lineales simples, sino en combinaciones más complejas de variables financieras.

En segundo lugar, los modelos basados en árboles mejoran de forma clara el rendimiento. Random Forest, Gradient Boosting, XGBoost y LightGBM obtienen AUC superiores a 0,70, lo que indica una mayor capacidad para separar targets y no targets. Sin embargo, no todos presentan el mismo equilibrio entre discriminación, recall y generalización.

XGBoost y LightGBM identifican más targets que Random Forest, especialmente en sus versiones base o con SMOTE. Sin embargo, estas mejoras en detección no siempre vienen acompañadas de una mejor capacidad discriminativa. XGBoost SMOTE identifica 167 targets, pero su AUC-ROC se queda en 0,7166 y presenta un overfitting alto. LightGBM SMOTE identifica 166 targets, pero su AUC-ROC es 0,7198, también inferior al de Random Forest con SMOTE.

El caso de LightGBM balanced es especialmente relevante. Este modelo identifica 297 targets, el mayor número de todos los modelos probados, y alcanza un recall del 63,7 %. Sin embargo, también genera 324 falsos positivos y reduce la accuracy en test hasta el 66,73 %. Para una herramienta de screening inicial, un recall alto puede ser atractivo, pero un número tan elevado de falsos positivos

reduce la utilidad práctica del ranking, ya que obligaría al analista a revisar demasiadas empresas señaladas erróneamente como posibles targets.

En tercer lugar, Gradient Boosting con SMOTE queda muy cerca del mejor resultado. Obtiene una accuracy en test del 71,79 %, identifica 142 targets y alcanza un AUC-ROC de 0,7253. Es una alternativa competitiva, pero sigue ligeramente por debajo del Random Forest con SMOTE en AUC.

En conjunto, el mejor equilibrio lo ofrece Random Forest con SMOTE. Este modelo obtiene el AUC-ROC más alto, 0,7260, mantiene una accuracy en test del 71,39 %, identifica 130 targets y presenta un overfitting moderado. Aunque no es el modelo que detecta más targets, sí es el que combina mejor capacidad discriminativa, estabilidad e interpretación.

6.2 SELECCIÓN DEL MODELO FINAL

Tras comparar todos los modelos, se selecciona Random Forest con SMOTE como modelo final del proyecto.

6.2.1 Argumentación

Tras comparar todos los modelos, se selecciona Random Forest con SMOTE como modelo final del proyecto. Esta decisión se basa en que ofrece el mejor equilibrio entre rendimiento, generalización e interpretación.

El modelo obtiene el mayor AUC-ROC, con 0,7260, lo que indica la mejor capacidad para separar empresas target y no target. Además, mantiene un *overfitting* moderado, con una *accuracy* del 86,73 % en entrenamiento y del 71,39 % en test. Aunque otros modelos identifican más targets, como XGBoost SMOTE o LightGBM balanced, lo hacen con menor AUC, mayor *overfitting* o demasiados falsos positivos. En concreto, LightGBM balanced identifica 297 targets, pero genera 324 falsos positivos, lo que reduciría la utilidad práctica del ranking.

Por tanto, Random Forest con SMOTE no es el modelo que más targets detecta, pero sí el que ofrece un ranking más equilibrado para una herramienta de *screening*. Además, permite analizar la importancia de variables de forma relativamente clara (Breiman, 2001), y SMOTE es una técnica adecuada para tratar datasets desbalanceados siempre que se aplique solo sobre el conjunto de entrenamiento para evitar *data leakage* (Chawla et al., 2002; He & Garcia, 2009).

6.2.2. Interpretación práctica del modelo

El score generado por el dashboard debe interpretarse como una señal relativa de atractivo financiero, no como una probabilidad exacta de adquisición. Por ejemplo, un score del 35 % no significa que la empresa tenga exactamente un 35 % de probabilidad real de ser adquirida, sino que su perfil financiero se parece más al de targets históricos que el de otras empresas con menor puntuación.

También es importante interpretar el recall dentro del uso práctico de la herramienta. Que el modelo identifique en torno al 30 % de los targets reales significa que consigue recuperar una parte relevante de las empresas adquiridas, pero no todas. Esto es coherente con un enfoque conservador: el objetivo no es marcar demasiadas compañías como targets, sino priorizar aquellas con una señal más clara y reducir el riesgo de generar una lista excesiva de falsos positivos.

Por tanto, el dashboard debe utilizarse como una primera capa de análisis. Una empresa con score alto merece una revisión más detallada, pero una empresa con score bajo no queda automáticamente descartada, ya que pueden existir factores estratégicos, cualitativos o privados que el modelo no observa. La herramienta ayuda a ordenar el universo inicial, pero la decisión final debe depender del criterio financiero y estratégico del analista.

6.3 DESARROLLO DEL DASHBOARD

Una vez seleccionado el modelo final, el siguiente paso del proyecto consiste en construir una herramienta que permita explorar sus resultados de una forma práctica, similar a como lo haría un analista de M&A, corporate development o private equity en su trabajo diario. Para ello se ha desarrollado un dashboard web interactivo siguiendo la arquitectura cliente-servidor descrita en el apartado 2.7, aplicando el modelo Random Forest con SMOTE sobre el universo completo de empresas estadounidenses disponibles en Compustat. Al final del capítulo se encuentran los links al código y demo.

6.3.1 Arquitectura general del dashboard

El dashboard está formado por tres archivos con responsabilidades diferenciadas, que se ejecutan sobre un servidor local mediante el framework Flask:

app.py	dashboard.html	dashboard.js
Backend	**Frontend**	**Lógica frontend**
Responsable de cargar el universo de empresas, calcular las variables financieras necesarias, aplicar el modelo entrenado y exponer los resultados a través de una serie de rutas	Define la estructura visual del dashboard: el encabezado, el menú de navegación lateral y las cuatro secciones principales (Executive Summary, M&A Screener, Company Deep Dive y Simulador), además del estilo visual de toda la interfaz.	Es responsable de comunicarse con el backend mediante peticiones HTTP, de procesar las respuestas recibidas y de actualizar dinámicamente el contenido de la página sin necesidad de recargarla.

Tabla 7. Descripción archivos responsables del funcionamiento del dashboard.

El dashboard se compone de tres archivos principales: app.py, dashboard.html y dashboard.js. El archivo app.py funciona como backend y se encarga de cargar los datos, calcular las variables financieras, aplicar el modelo entrenado y exponer los resultados mediante rutas. El archivo dashboard.html define la estructura visual de la aplicación, mientras que dashboard.js gestiona la interacción del usuario con la página y las peticiones al backend.

Al ejecutar la aplicación, Flask inicia un servidor local. Durante el arranque, el backend carga el universo de empresas, recalcula las variables derivadas necesarias y obtiene el M&A Likelihood Score para cada compañía. Después, cuando el usuario interactúa con el dashboard, el frontend envía peticiones HTTP al backend y recibe las respuestas en formato JSON, actualizando la página sin necesidad de recargarla completamente. El detalle técnico del flujo de funcionamiento, las rutas implementadas y los ejemplos de respuestas del servidor se recoge en el Anexo 5.1.

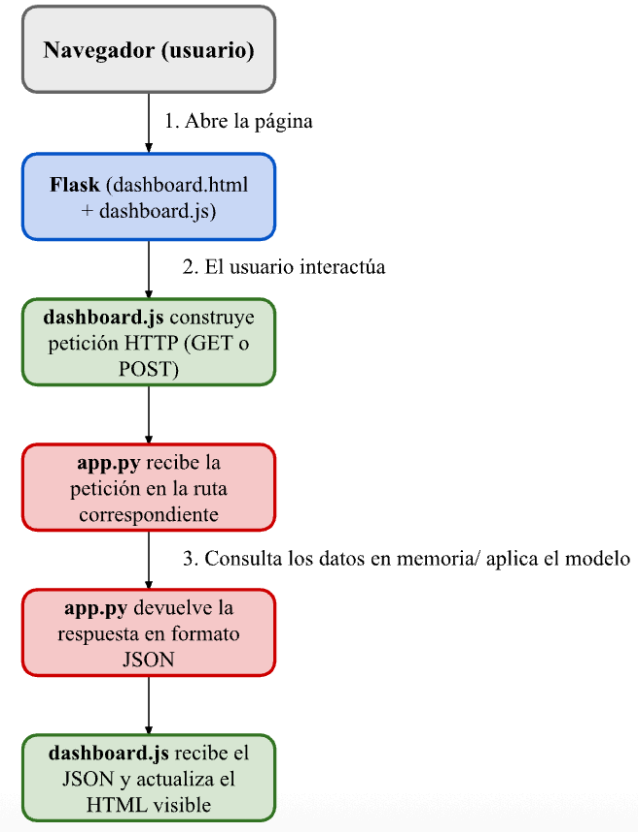


Figura 37. Esquema del flujo de funcionamiento del dashboard.

Fuente: elaboración propia.

Las rutas principales definidas en el backend, junto con la sección del dashboard a la que dan servicio, son las siguientes:

Ruta	Método	Sección del dashboard
/	GET	Sirve la página principal
/api/summary	GET	Executive Summary
/api/screener	GET	M&A Screener
/api/empresa/<ticker>	GET	Company Deep Dive
/api/simular	POST	Simulador
/api/tickers	GET	Selector de empresas (Deep Dive)

Tabla 8. Rutas principales del backend.

6.3.2 Carga de universo de empresas y cálculo del score

Antes de que el dashboard pueda mostrar ningún resultado, app.py necesita preparar los datos. USCompanies.csv contiene las variables financieras "en bruto" procedentes de Compustat (activos, ventas, EBITDA, deuda a corto y largo plazo, etc.) pero no las variables derivadas que utiliza el modelo, como el EV/EBITDA o el ratio de apalancamiento. La función **calcular_features()** del backend recalcula estas variables siguiendo exactamente la misma lógica empleada durante el entrenamiento del modelo, para garantizar que las predicciones sobre el universo completo sean coherentes con las obtenidas en la fase de modelado. Este proceso se detalla en el Anexo 5.2.

A continuación, la función **construir_tabla_empresas()** resuelve un problema propio de los datos de panel de Compustat: cada empresa aparece una vez por cada año fiscal del que existe información, de modo que una compañía con quince años de histórico genera quince filas distintas. Para que el dashboard muestre una única fila por empresa, esta función calcula dos puntuaciones por compañía:

1. Score reciente: el score correspondiente a su año fiscal más reciente (el que se utiliza como referencia principal en el resto del dashboard) y
2. Score atractivo: el score más alto de toda su historia, junto con el año en que se produjo. Esta segunda puntuación permite identificar en qué momento de su trayectoria financiera una empresa presentó el perfil más compatible con el de un target de adquisición.

El Anexo 5.1 muestra los outputs en la terminal del momento de arranque: la carga de datos y las respuestas del servidor con su código dependiendo de la petición GET.

6.3.3 Executive Summary

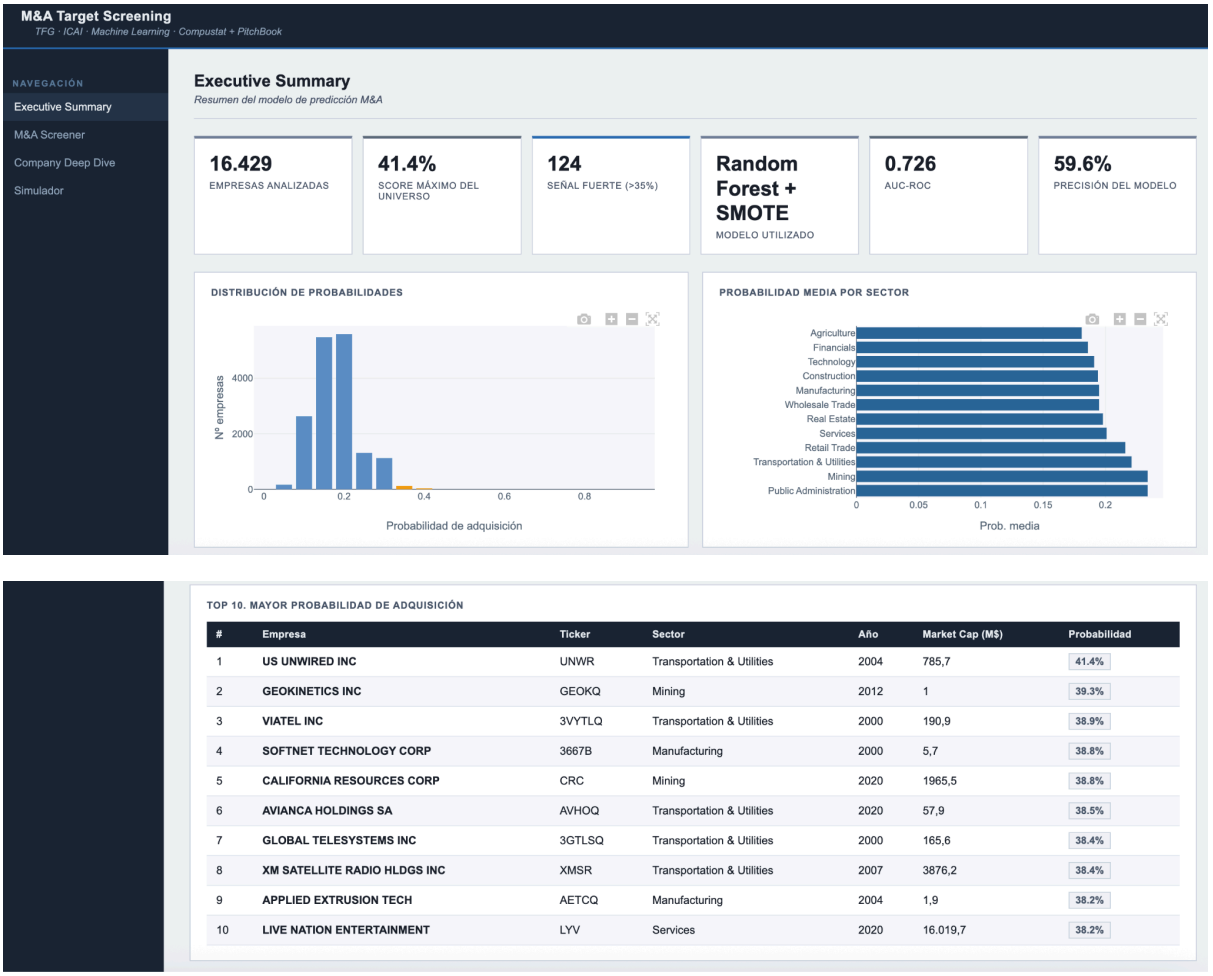


Figura 38. Captura de pantalla de la sección 'Executive Summary'.

Esta es la primera pantalla que ve el usuario al abrir el dashboard. Muestra un resumen general del universo analizado: el número de empresas, el score máximo alcanzado, el número de empresas con una señal de adquisición fuerte, el modelo utilizado y sus métricas principales (AUC-ROC y precisión), además de la distribución de scores en todo el universo, el score medio por sector y un listado con las diez empresas con mayor puntuación.

En el momento en que el navegador carga la página, la función **cargarSummary()** de dashboard.js se ejecuta automáticamente y realiza una petición GET a la ruta /api/summary. En el backend, la función **get_summary()** de app.py recibe esa petición, calcula los indicadores a partir de la tabla de empresas ya cargada en memoria (sin necesidad de volver a leer ningún archivo) y devuelve un objeto JSON

con los indicadores, la distribución de scores, el score medio por sector y el listado de las diez empresas principales. El frontend recibe esa respuesta y reparte sus distintos campos entre los elementos visuales correspondientes. El funcionamiento interno de está sección se explica en más detalle en el Anexo 5.5.1.

6.3.4 M&A Screener

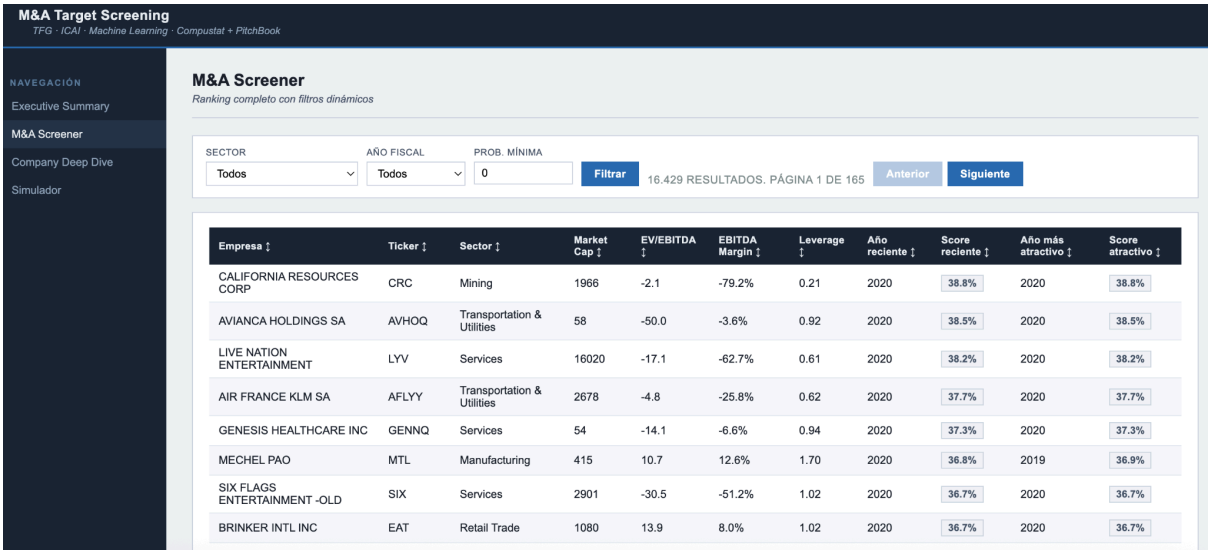


Figura 39. Captura de pantalla de la sección 'M&A Screener".

La sección M&A Screener permite explorar el universo completo de empresas mediante un ranking ordenable y filtrable. El usuario puede filtrar por sector, año fiscal y score mínimo, y ordenar los resultados según distintas variables financieras. Como el universo de observaciones es amplio, los resultados se muestran de forma paginada, devolviendo cien filas por petición para mantener la herramienta manejable.

Esta sección representa el uso principal del dashboard como herramienta de *screening*. Permite pasar de un universo amplio de compañías a una lista priorizada de posibles targets, reduciendo el trabajo manual inicial. Además, cada fila incluye tanto el score reciente como el score máximo histórico, lo que permite distinguir entre el atractivo actual de una empresa y el momento en el que tuvo un perfil financiero más parecido al de targets históricos.

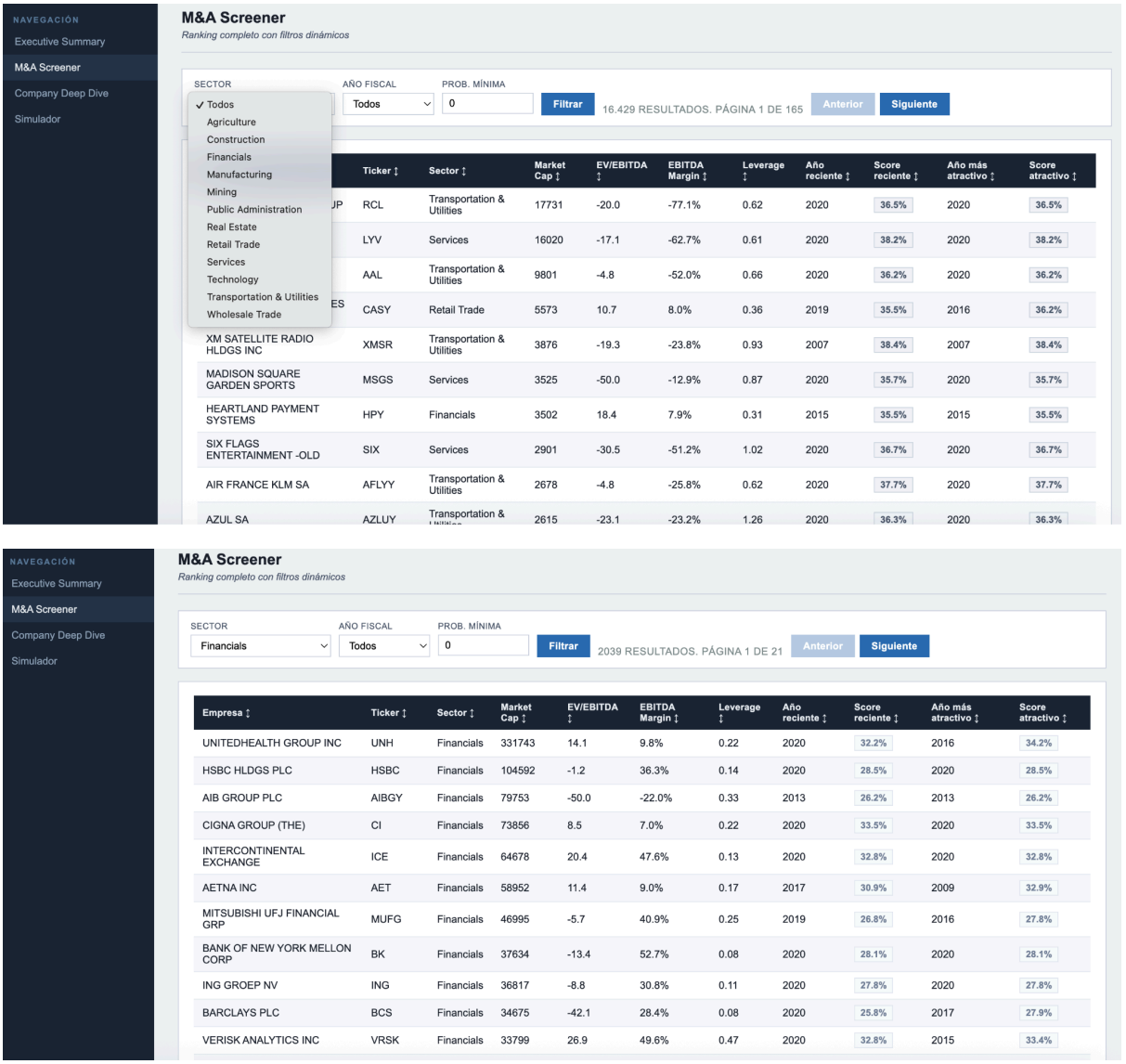


Figura 40. Ejemplo resultado de filtrado de datos por 'financials' y 'market cap'.

Cuando el usuario aplica un filtro, la función **aplicarFiltros()** de dashboard.js lee los valores seleccionados en los desplegables y el campo de score mínimo, construye una petición GET a /api/screener añadiendo esos valores como parámetros de la dirección y la envía al servidor. En el backend, la función **get_screener()** de app.py recoge esos parámetros, filtra la tabla de empresas en memoria, la ordena por score y devuelve únicamente la página solicitada junto con el número total de resultados y de páginas disponibles. El frontend utiliza esa información para construir la tabla, activar o desactivar los botones de "Anterior" y "Siguiente" según corresponda, y mostrar el contador de resultados.

La función **ordenarTabla()** permite además reordenar los resultados visibles haciendo clic sobre cualquier cabecera de columna, alternando entre orden ascendente y descendente sin necesidad de volver a consultar al servidor, ya que opera directamente sobre los datos de la página ya cargada en el navegador. Por ejemplo, la Figura 39 muestra como se ha filtrado por 'financials' (/api/screener?sector=Financials) y luego filtrado descendentemente por 'market_cap'. Los detalles de esta backend de esta sección se explican en el Anexo 5.5.2, y la lógica del frontend en Anexo 5.6.

6.3.5 Company Deep Dive

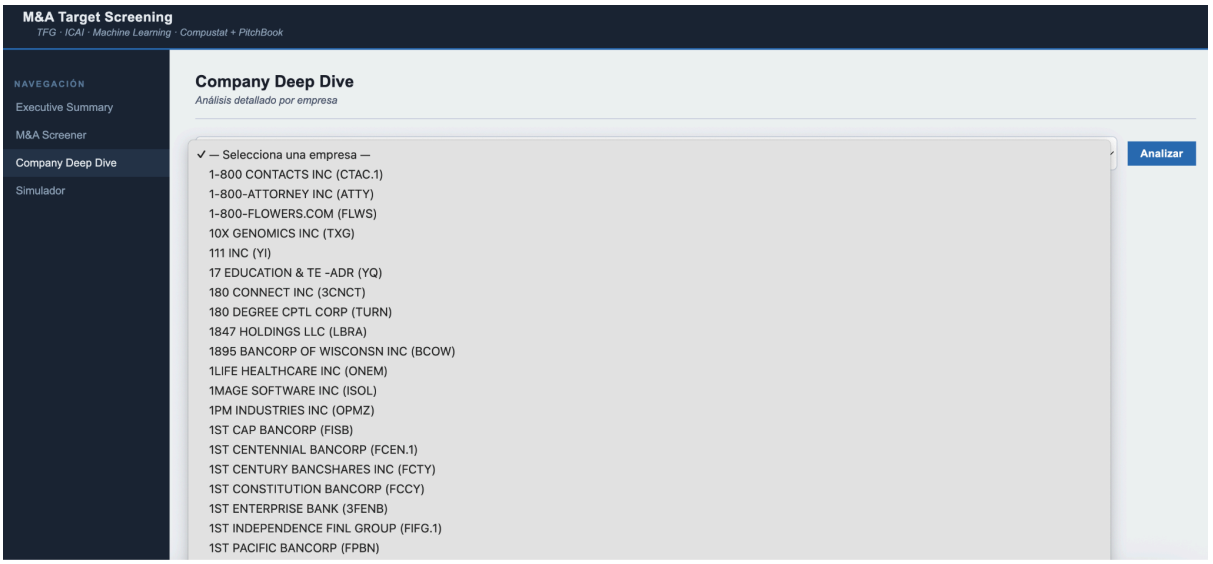


Figura 41. Captura de pantalla de la sección 'Company Deep Dive'.

Esta sección permite seleccionar una empresa concreta del universo y consultar su perfil financiero completo, su score de adquisición, su posición respecto a la mediana de su sector y la evolución de su score a lo largo de los años fiscales disponibles.

Desde el punto de vista técnico, la sección utiliza la ruta /api/empresa/<ticker>, que localiza la empresa seleccionada, calcula su señal, obtiene las medianas sectoriales y recupera el histórico de scores a partir de la base completa. El funcionamiento de esta ruta se explica en el Anexo 5.5.3. Por su parte, la representación visual mediante tarjetas, indicador tipo velocímetro, gráfico radar y línea temporal se recoge en el Anexo 5.7.

Además, el dashboard permite generar un informe HTML automático con el resumen de la empresa analizada. Este informe se descarga directamente desde el navegador y permite conservar una primera

ficha de análisis sin necesidad de almacenar información adicional en el servidor. La lógica de generación y descarga de informes se explica también en el Anexo 5.7.

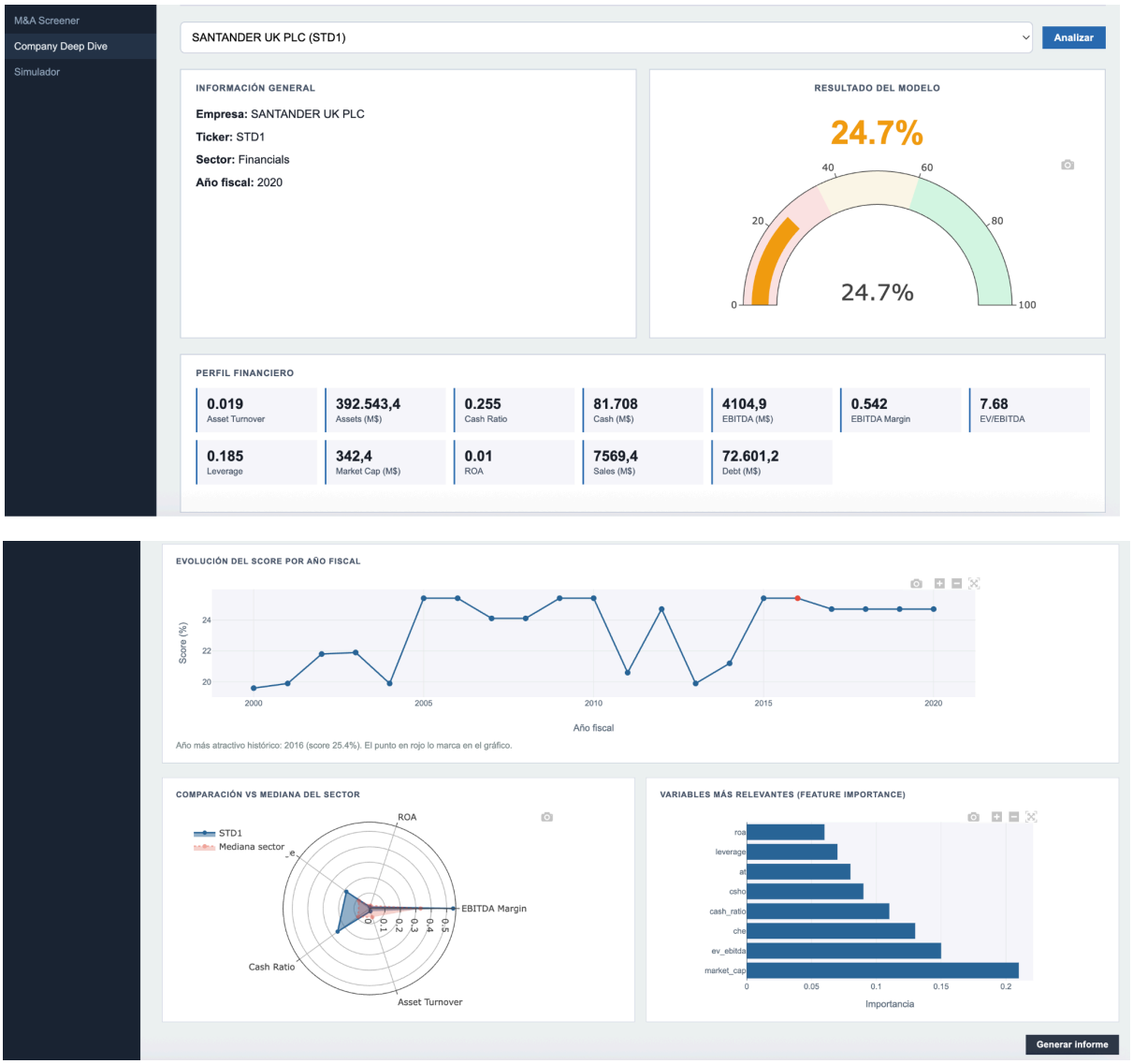


Figura 42. Ejemplo del perfil financiero de Santander UK con su informe HTML.

Informe M&A. SANTANDER UK PLC (STD1)

Generado el 30/6/2026 — TFG · M&A Target Screening · ICAI

SANTANDER UK PLC (STD1)

Sector: Financials | **Año:** 2020

Resultado del modelo

Probabilidad: **24.7%** . undefined

Perfil financiero

- **asset_turnover:** 0.019
- **at:** 392543.4
- **cash_ratio:** 0.255
- **che:** 81708
- **ebitda:** 4104.9
- **ebitda_margin:** 0.542
- **ev_ebitda:** 7.68
- **leverage:** 0.185
- **market_cap:** 342.4
- **roa:** 0.01
- **sale:** 7569.4
- **total_debt:** 72601.2

Variables más relevantes

1. market_cap (0.21)
2. ev_ebitda (0.15)
3. che (0.13)
4. cash_ratio (0.11)
5. csho (0.09)

Conclusión automática

STD1 no presenta el perfil típico de objetivo de adquisición según el modelo (24.7%). Su estructura financiera o valoración no es consistente con el universo de targets históricos.

Figura 42. Ejemplo del perfil financiero de Santander UK con su informe HTML.

Desde el punto de vista de un analista, la sección Company Deep Dive permite pasar del ranking general a una revisión más concreta de cada empresa. El score no debe interpretarse como una probabilidad exacta de adquisición, sino como una señal relativa de similitud con targets históricos. El indicador circular permite ver rápidamente la intensidad de esa señal; el perfil financiero muestra las variables principales que explican el caso; el gráfico de evolución permite comprobar si el score es puntual o si se mantiene en el tiempo; y la comparación con la mediana sectorial ayuda a interpretar si la empresa destaca frente a compañías similares.

Por ejemplo, en la Figura 41, Santander UK PLC obtiene un score de 24,7 %, lo que indica una señal moderada-baja. Aunque presenta métricas financieras relevantes, el indicador circular y el informe muestran que no se sitúa entre las compañías con perfil más claro de target. Además, la evolución histórica del score se mantiene en niveles relativamente estables, sin un salto claro que justifique priorizarla en una primera lista corta, salvo que existan factores estratégicos externos al modelo.

En cambio, la Figura 42 muestra US Unwired Inc., con un score de 41,4 %. El indicador circular refleja una señal más fuerte y la evolución temporal muestra un aumento progresivo hasta alcanzar su

máximo en el último año disponible. Esto sugiere que su perfil financiero se acerca más al de targets históricos y que el caso merecería una revisión más detallada. Aun así, el modelo mantiene un enfoque conservador: identifica una parte relevante de los targets reales, pero no todos, y busca priorizar empresas con una señal más clara sin generar una lista excesiva de falsos positivos. Por tanto, el dashboard ayuda a ordenar el universo inicial, pero la decisión final debe completarse con análisis financiero, estratégico y cualitativo.

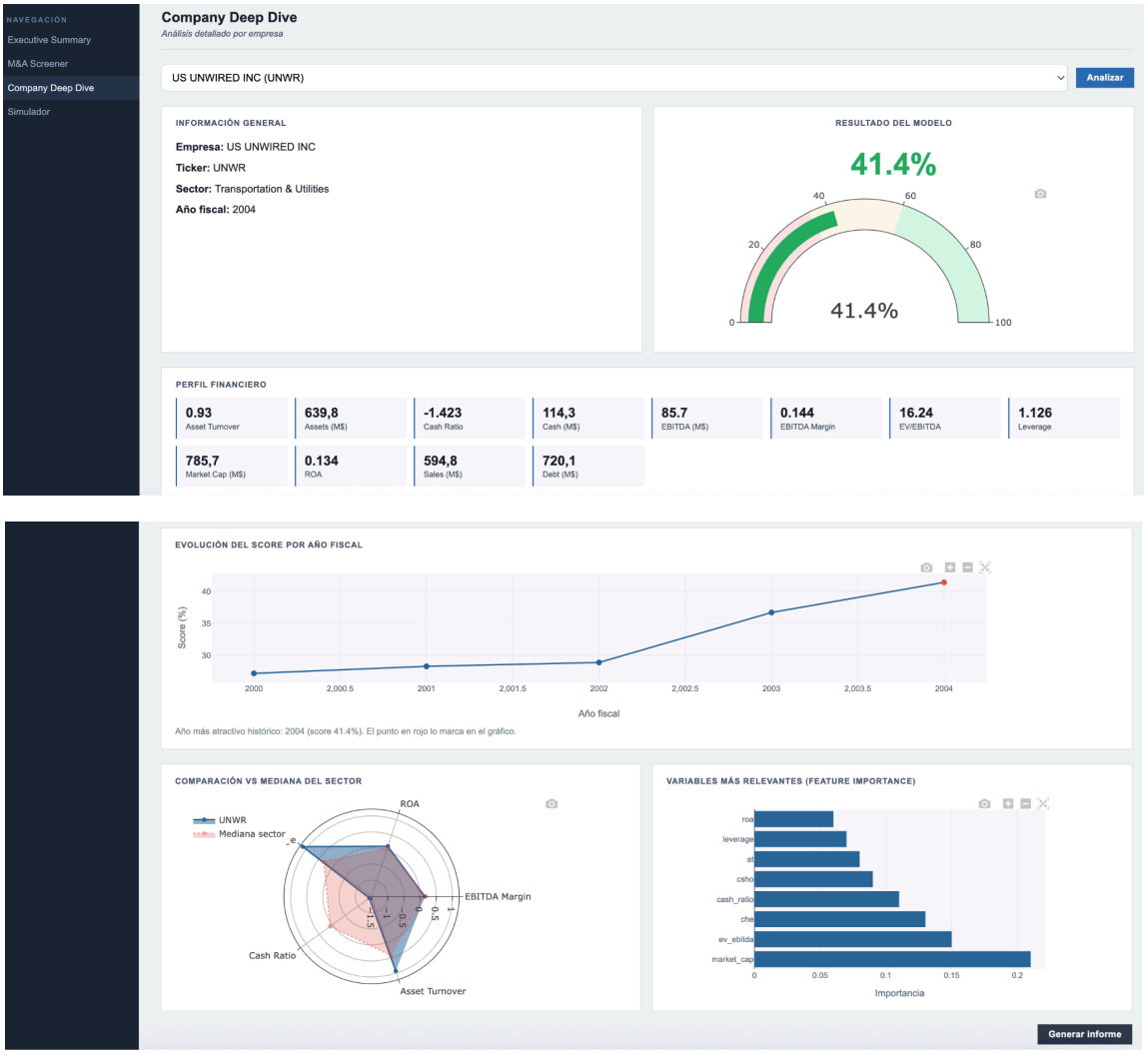
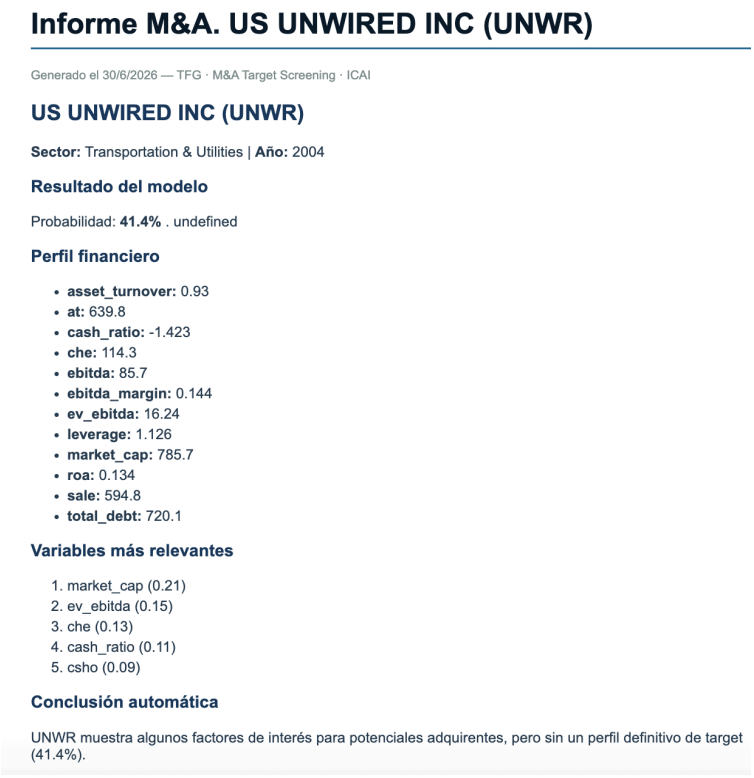


Figura 43. Ejemplo del perfil financiero de US Unwired Inc. Mejor valorizada.



Informe M&A. US UNWIRED INC (UNWR)

Generado el 30/6/2026 — TFG · M&A Target Screening · ICAI

US UNWIRED INC (UNWR)

Sector: Transportation & Utilities | **Año:** 2004

Resultado del modelo

Probabilidad: **41.4%** . undefined

Perfil financiero

- **asset_turnover:** 0.93
- **at:** 639.8
- **cash_ratio:** -1.423
- **che:** 114.3
- **ebitda:** 85.7
- **ebitda_margin:** 0.144
- **ev_ebitda:** 16.24
- **leverage:** 1.126
- **market_cap:** 785.7
- **roa:** 0.134
- **sale:** 594.8
- **total_debt:** 720.1

Variables más relevantes

1. market_cap (0.21)
2. ev_ebitda (0.15)
3. che (0.13)
4. cash_ratio (0.11)
5. csho (0.09)

Conclusión automática

UNWR muestra algunos factores de interés para potenciales adquirentes, pero sin un perfil definitivo de target (41.4%).

Figura 43. Ejemplo del perfil financiero de US Unwired Inc. Mejor valorizada.

6.3.6 Simulador

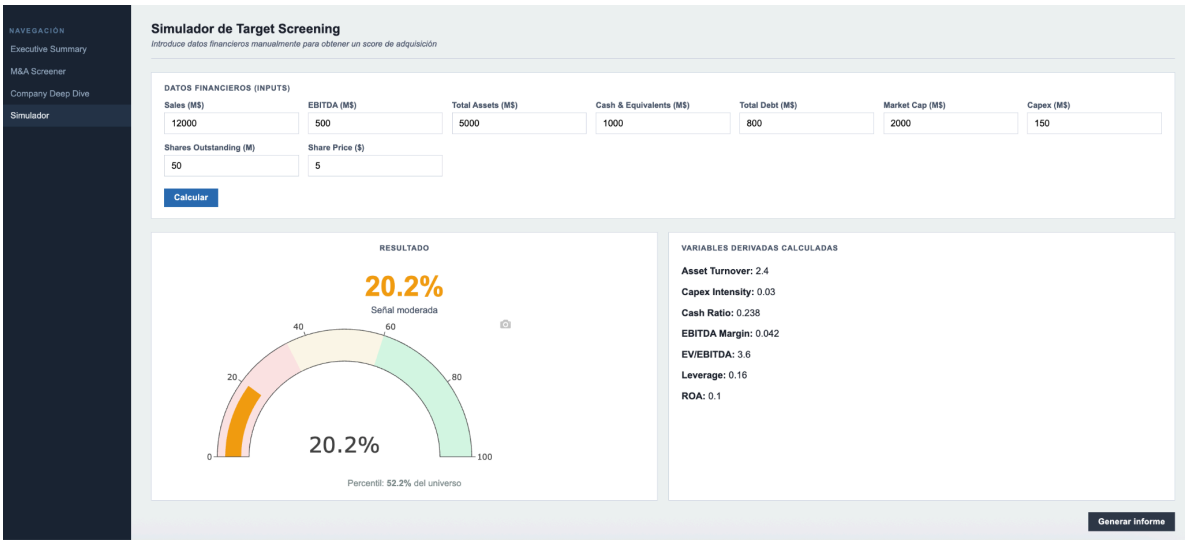


Figura 44. Captura de pantalla de la sección 'Simulator'.

Esta última sección permite introducir manualmente los datos financieros de una empresa hipotética y obtener su puntuación de adquisición estimada, sin necesidad de que dicha empresa exista en el universo de Compustat. Resulta útil, por ejemplo, para evaluar de forma rápida una compañía privada o para realizar análisis de sensibilidad sobre cómo cambiaría la puntuación de una empresa si variasen ciertos indicadores financieros.

A diferencia del resto de secciones, que únicamente consultan información ya calculada, esta sección envía datos al servidor para que sean procesados. Cuando el usuario rellena el formulario y pulsa "Calcular", la función **calcularSimulacion()** de dashboard.js recoge los valores introducidos en cada campo, los agrupa en un objeto y realiza una petición POST a la ruta /api/simular, incluyendo ese objeto como cuerpo de la petición en formato JSON. En el backend, la función simular() de app.py recibe esos datos, calcula las variables financieras derivadas necesarias siguiendo la misma lógica empleada en el resto del proyecto, construye con ellas el mismo vector de variables que utiliza el modelo entrenado y obtiene la probabilidad de adquisición correspondiente. El resultado se devuelve al frontend junto con el percentil que ocuparía esa empresa hipotética dentro del universo real, lo que permite al usuario contextualizar el resultado obtenido.

El funcionamiento técnico de la ruta /api/simular se explica en el Anexo 5.5.4. La lógica del frontend para enviar los datos mediante POST se detalla en el Anexo 5.6, y la representación visual del resultado junto con la generación del informe HTML se recoge en el Anexo 5.7.

6.3.7 Disponibilidad de la herramienta

El código completo del dashboard, junto con el resto del proyecto, está disponible públicamente en el siguiente repositorio:

https://github.com/azugasti18/TFG_GITT_AdrianaZugasti

Link para la demo del dashboard:

Demo dashboard TFG AdrianaZugasti.mov

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

7.1 CONCLUSIONES

El objetivo principal de este Trabajo Fin de Grado era desarrollar una herramienta de *screening* de targets de M&A mediante técnicas de *Machine Learning*, utilizando datos financieros públicos de empresas cotizadas estadounidenses. Este objetivo se ha cumplido mediante la construcción de un pipeline completo que integra tres fases principales: creación del dataset, entrenamiento y comparación de modelos predictivos, y desarrollo de un dashboard web para visualizar y explotar los resultados.

La primera conclusión del trabajo es que los datos financieros públicos sí contienen cierta señal útil para priorizar empresas dentro de un proceso de identificación de targets. Sin embargo, esta señal no permite predecir adquisiciones con certeza, algo coherente con la naturaleza del M&A. Una operación de adquisición no depende únicamente de ratios financieros, sino también de factores estratégicos, conversaciones privadas, sinergias potenciales, regulación, contexto de mercado y decisiones internas de compradores y vendedores. Por tanto, el modelo debe interpretarse como una herramienta de apoyo al análisis, no como un sustituto del criterio profesional.

La construcción del dataset ha sido una parte esencial del proyecto. Se combinaron operaciones completadas de M&A procedentes de PitchBook con información financiera de Compustat, creando una variable objetivo binaria que distingue entre empresas adquiridas y no adquiridas. Tras la integración de ambas fuentes, el dataset fue sometido a un proceso de limpieza, imputación de valores nulos, análisis de outliers, revisión de correlaciones, control de multicolinealidad y selección final de variables. Este proceso permitió transformar datos financieros brutos en una base preparada para entrenar modelos de clasificación supervisada.

Desde el punto de vista del modelaje, los resultados muestran que la regresión logística no es suficiente para este problema. Su AUC-ROC fue bajo y apenas consiguió identificar targets reales, lo que confirma que la probabilidad de adquisición no parece depender de una relación lineal simple entre variables financieras aisladas. En cambio, los modelos basados en árboles obtuvieron resultados claramente superiores, lo que indica que la señal se encuentra en combinaciones más complejas de

variables como tamaño, liquidez, rentabilidad, endeudamiento, valoración, sector y mercado de cotización.

Tras comparar todas las alternativas, se seleccionó Random Forest con SMOTE como modelo final. Este modelo obtuvo el mejor equilibrio entre rendimiento, estabilidad e interpretación, con un AUC-ROC de 0,7260, una accuracy en test del 71,39 % y 130 targets identificados en el conjunto de test. Aunque otros modelos como XGBoost o LightGBM detectaron un mayor número de targets, también presentaron mayor overfitting o un número más elevado de falsos positivos. En una herramienta de *screening*, el objetivo no es simplemente marcar el mayor número posible de empresas como targets, sino generar un ranking útil y razonable para priorizar el análisis.

La última aportación del trabajo ha sido el desarrollo de un dashboard web interactivo que integra el modelo seleccionado. La herramienta permite consultar un resumen ejecutivo del universo analizado, filtrar empresas mediante un *M&A Screener*, analizar compañías concretas en la sección *Company Deep Dive* y simular la probabilidad de adquisición de una empresa introducida manualmente. De esta forma, el proyecto no se queda en un modelo teórico, sino que se convierte en una herramienta funcional orientada a un uso práctico dentro de la fase inicial de análisis de M&A.

En conjunto, el trabajo demuestra que el *Machine Learning* puede aportar valor al proceso de *target screening*, especialmente cuando el universo inicial de empresas es amplio. Su utilidad principal está en ordenar compañías, detectar perfiles financieros similares a targets históricos y reducir el tiempo necesario para llegar a una lista inicial de candidatos. Aun así, la decisión final de analizar o adquirir una empresa debe seguir dependiendo del juicio financiero, estratégico y sectorial del analista.

7.2 LIMITACIONES DEL TRABAJO

La principal limitación del proyecto está relacionada con el alcance de los datos utilizados. El modelo se ha entrenado únicamente con empresas cotizadas estadounidenses, ya que estas ofrecen información financiera pública, homogénea y estructurada. Esta decisión mejora la calidad del análisis, pero deja fuera una parte muy importante del mercado de M&A: las empresas privadas. Muchas operaciones, especialmente en *middle market* y private equity, se realizan sobre compañías no cotizadas, por lo que los resultados no pueden extrapolarse directamente a todo el mercado.

Otra limitación relevante es la construcción de la variable objetivo. Una empresa etiquetada como no target no significa necesariamente que nunca haya sido considerada en una operación, sino

simplemente que no aparece como adquirida en PitchBook durante el periodo y bajo los filtros definidos. Además, el cruce entre PitchBook y Compustat requiere un proceso de emparejamiento entre compañías que, aunque se haya realizado de forma conservadora, puede introducir cierto margen de error.

También debe tenerse en cuenta que el modelo aprende de patrones históricos. El periodo analizado recoge distintos contextos económicos, pero el mercado de M&A cambia con los tipos de interés, la financiación disponible, la regulación, las valoraciones de mercado y las prioridades estratégicas de los compradores. Por tanto, si la herramienta se quisiera utilizar en un entorno real, sería necesario actualizar periódicamente los datos y reentrenar el modelo.

Por último, el modelo utiliza principalmente variables financieras estructuradas. Esto permite que el proceso sea reproducible, pero excluye factores cualitativos muy relevantes en M&A, como la calidad del equipo directivo, la posición competitiva, la existencia de sinergias, el interés de compradores concretos o la situación accionarial de la compañía. Por ello, el score generado debe interpretarse como una señal financiera relativa, no como una probabilidad definitiva de adquisición.

7.3 TRABAJOS FUTUROS

A partir de estas limitaciones, existen varias líneas de mejora. La primera sería ampliar las fuentes de datos. Además de la información financiera de Compustat, podrían incorporarse variables de mercado, evolución bursátil, múltiplos históricos, propiedad institucional, cobertura de analistas, noticias, información sectorial y datos macroeconómicos. Esto permitiría enriquecer el modelo con señales más cercanas a las que se utilizan en un proceso real de análisis de targets.

Una segunda mejora sería reformular el problema desde una perspectiva temporal. En este trabajo se plantea una clasificación binaria entre empresas adquiridas y no adquiridas. En futuras versiones, podría analizarse no solo si una empresa acaba siendo adquirida, sino cuándo ocurre la adquisición. Para ello podrían utilizarse ventanas temporales móviles, modelos de supervivencia o enfoques basados en datos panel, capturando mejor la evolución financiera de cada compañía antes de convertirse en target.

También sería interesante profundizar en la interpretabilidad del modelo. Aunque Random Forest permite analizar la importancia global de las variables, técnicas como SHAP permitirían explicar cada predicción individual. Esto sería especialmente útil en el dashboard, ya que el usuario no solo vería el

score de adquisición de una empresa, sino también qué factores concretos están impulsando ese resultado.

Otra línea de mejora sería evaluar el modelo con métricas más alineadas con el uso real del *screening*. Además del AUC-ROC o la accuracy, podrían utilizarse métricas como *precision at k*, *recall at k* o *lift*, que miden cuántos targets reales aparecen entre las primeras posiciones del ranking. Esto encajaría mejor con el trabajo de un analista, que normalmente no revisa todo el universo de empresas, sino una lista priorizada.

Finalmente, el dashboard podría evolucionar hacia una herramienta más completa mediante despliegue en la nube, actualización automática de datos, generación de informes en PDF, comparación entre empresas, filtros sectoriales avanzados y alertas cuando una compañía alcance un determinado score. Estas mejoras permitirían transformar el prototipo desarrollado en este TFG en una herramienta más cercana a una solución profesional de apoyo al *deal sourcing*.

En conclusión, este trabajo establece una base sólida para aplicar técnicas de *Machine Learning* al *screening* de targets de M&A. Aunque el modelo no sustituye el análisis financiero y estratégico tradicional, sí demuestra que es posible utilizar datos públicos para ordenar empresas de forma más sistemática, reproducible y eficiente, ayudando al analista a centrar su atención en las compañías con mayor similitud financiera a targets históricos.

Capítulo 8. BIBLIOGRAFÍA

Abhishek, A. (2024, Julio). Mergers and Acquisitions: Identifying Targets. LinkedIn. https://www.linkedin.com/pulse/mergers-acquisitions-identifying-targets-abhishek-ankur-ntxsc/

Arnold, C., Biedebach, L., Küpfer, A., & Neunhoeffer, M. (2024). The role of hyperparameters in machine learning models and how to tune them. Political Science Research and Methods, 12(4), 841–848. doi:10.1017/psrm.2023.61

Bain & Company. (2024, Diciembre). Global M&A deal value on track to reach $3.5 trillion in 2024—Bain & Company. Bain & Company. https://www.bain.com/about/media-center/press-releases/2024/global-ma-deal-value-on-track-to-reach-%243.5-trillion-in-2024bain--company?utm_source=chatgpt.com

Balan, Binesh and W, Ivy and Ni, Nikyi and Ma, Yiran and Ren, Maximilian and Huang, Junwen and Zhu, Sophia and Balan, Bijila, AI-Driven M&A - How Algorithms are Influencing Target Identification and Valuation (September 15, 2025). Available at SSRN: https://ssrn.com/abstract=5583110 or http://dx.doi.org/10.2139/ssrn.5583110

Beckenstrater, G. (2024). Predicting mergers and acquisitions using machine learning. (). University of Cape Town ,Faculty of Science ,Department of Computer Science. Retrieved from http://hdl.handle.net/11427/40791

Breiman, L., "Random Forests", Machine Learning, vol. 45, pp. 5–32, 2001.

Brixius, M., Perret, J. K., Schröder Jörg, & Kamilė, T. (2025). Determinants of M&A Acquisition Premiums on the European Market in the Period of 2009 to 2022. International Journal of Financial Studies, 13(4), 204. https://doi.org/10.3390/ijfs13040204

Campbell, John L. and Elfrink, Erik and Irons, Charles and Moon, James, What is the Deal?: Predicting M&A Outcomes with Machine Learning (October 01, 2024). Available at SSRN: https://ssrn.com/abstract=4987268 or http://dx.doi.org/10.2139/ssrn.4987268

Charilaou, P., & Battat, R. (2022). Machine learning models and over-fitting considerations. World journal of gastroenterology, 28(5), 605–607. https://doi.org/10.3748/wjg.v28.i5.605

Chawla, N. V., Bowyer, K. W., Hall, L. O. y Kegelmeyer, W. P., "SMOTE: Synthetic Minority Over-sampling Technique", Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785–794. https://doi.org/10.1145/2939672.2939785

Dube, L., & Verster, T. (2023). Enhancing classification performance in imbalanced datasets: A comparative analysis of machine learning models. AIMS. https://www.researchgate.net/profile/Lindani-Dube/publication/374734735_Enhancing_classification_performance_in_imbalanced_datasets_A_comparative_analysis_of_machine_learning_models/links/652d4e7f6725c324010cc0f1/Enhancing-classification-performance-in-imb

Fawcett, T., "An introduction to ROC analysis", Pattern Recognition Letters, vol. 27, no. 8, pp. 861–874, 2006.

Flannery, M. J., Hanousek, J., Shamshur, A., & Tresl, J. (2023). M&A Activity and the Capital Structure of Target Firms. Journal of Financial and Quantitative Analysis, 58(5), 2064–2095. doi:10.1017/S0022109022000436. https://www.cambridge.org/core/journals/journal-of-financial-and-quantitative-analysis/article/ma-activity-and-the-capital-structure-of-target-firms/CF35CADD69711E98AFABDF736BE198AB

Friedman, J. H., "Greedy Function Approximation: A Gradient Boosting Machine", The Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

He, H. y Garcia, E. A., "Learning from Imbalanced Data", IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263–1284, 2009.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). An introduction to statistical learning: With applications in R (2nd ed.). Springer. https://doi.org/10.1007/978-1-0716-1418-1

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30.

Kordestani, K., & Silva, R. (2026, January 14). Gen AI in M&A: From theory to practice to high performance. McKinsey. Retrieved June 28, 2026, from

https://www.mckinsey.com/capabilities/m-and-a/our-insights/gen-ai-in-m-and-a-from-theory-to-practice-to-high-performance

Lee, S. E. (2005). Policy framework in telecommunications mergers and acquisitions: A comparative analysis of merger review of the FCC and the DOJ/FTC (Order No. 3192422). Available from ProQuest One Business. (304996447). https://www.proquest.com/dissertations-theses/policy-framework-telecommunications-mergers/docview/304996447/se-2

Levy, B. (2026, June 23). Global M&A industry trends: 2026 mid-year outlook. PwC. Retrieved June 26, 2026, from https://www.pwc.com/gx/en/services/deals/trends.html

Lobo, B., Vieira, E., & Oliveira, J. (2026). Capital Market Reaction to Mergers and Acquisitions Announcements. Brazilian Business Review, 23, 1-22. https://doi.org/10.15728/bbr.2024.2201.en

Lukander, O. (2025). Predicting Merger and Acquisition Outcomes: A Machine Learning Approach. https://aaltodoc.aalto.fi/server/api/core/bitstreams/07272e9c-f49d-4907-b866-9cc7f76d5a12/content

Narteh-Kofi, E., Sampson2, E., Hattoh, E., & CFA. (2025, August). Optimizing Target Identification in the U.S. Capital Market Mergers and Acquisitions Through Artificial Intelligence: Implications for Financial Efficiency, Compliance, and National Economic Competitiveness. International Journal for Multidisciplinary Research. https://www.researchgate.net/profile/Ephraim-Narteh-Kofi/publication/394116691_Optimizing_Target_Identification_in_the_US_Capital_Market_Mergers_and_Acquisitions_through_Artificial_Intelligence_Implications_for_Financial_Efficiency_Compliance_and_National_

Noh, S.-H. (2023). Comparing the Performance of Corporate Bankruptcy Prediction Models Based on Imbalanced Financial Data. Sustainability, 15(6), 4794. https://doi.org/10.3390/su15064794

Palepu, K. G., “Predicting Takeover Targets: A Methodological and Empirical Analysis”, Journal of Accounting and Economics, vol. 8, no. 1, pp. 3–35, 1986.

Past, present, and future of sustainable finance: insights from big data analytics through machine learning of scholarly research. (2025). Annals of Operations Research, 345(2), 1061-1104. https://doi.org/10.1007/s10479-021-04410-8

PitchBook, Mergers & Acquisitions Data.
Usada para justificar PitchBook como fuente de datos de operaciones de M&A: tamaño, fecha, estructura, participantes, múltiplos, comparables y datos de compañías adquiridas.

Potynska Y. Current Trends in Processes of Mergers and Acquisitions in the Global Business Environment : the paper for the first bachelor's degree in higher education : Speciality 292 International Economic Relations / Scientific supervisor I. Kytsyuk ; Lesya Ukrainka Volyn National University. Lutsk, 2024. 54 p. https://evnuir.vnu.edu.ua/items/444ba79c-db7c-49fe-a53e-df5ee3b9fedf

R, Kishan and Velaga, Vasu and Chinta, Purna Chandra Rao and Jha, Krishna Madhav, Leveraging Machine Learning Techniques for Predictive Analysis in Merger and Acquisition (M&A) (July 08, 2023). Available at SSRN: https://ssrn.com/abstract=5102672 or http://dx.doi.org/10.2139/ssrn.5102672

Radoniqi, F. (2012). Essays on the impact of mergers on rivals (Order No. 3504447). Available from ProQuest One Business. (1011480496). https://www.proquest.com/dissertations-theses/essays-on-impact-mergers-rivals/docview/1011480496/se-2

Sanz Bobi, M. A. (2024). Material docente de la asignatura de Machine Learning I [Diapositivas de clase]. Universidad Pontificia Comillas, ICAI.

SMU. (2023). Post-pandemic M&A: Challenges and opportunities in the next normal. SMU. https://masters.smu.edu.sg/programme/msc_in_applied_finance/community-stories/postpandemic_ma_challenges_and_opportunities_in

Talaei Khoei, T., & Kaabouch, N. (2023). Machine Learning: Models, Challenges, and Research Directions. Future Internet, 15(10), 332. https://doi.org/10.3390/fi15100332

The Business Research Company. (2026, April). Global Investment Banking Market Briefing 2026. EMIS. https://www-emis-com.proxy2.library.illinois.edu/v2/documents/report/947258033

Universidad Pontificia Comillas, ICAI (2025). Material de la Asignatura - Programación de Aplicaciones Telemáticas. https://apicai.github.io/web-ejercicios-pat/index.html

WRDS, Compustat North America - Fundamentals Annual.
Usada para justificar Compustat como fuente de datos financieros anuales y trimestrales de empresas cotizadas de EE. UU. y Canadá, incluyendo balance, cuenta de resultados y cash flow.

ANEXO I: ALINEACIÓN DEL PROYECTO CON LOS ODS

Este proyecto contribuye principalmente a dos Objetivos de Desarrollo Sostenible de la Agenda 2030 de Naciones Unidas:

ODS 8 – Trabajo decente y crecimiento económico.

El desarrollo de herramientas que apoyen decisiones de inversión más fundamentadas contribuye a la eficiencia de los mercados financieros y, por extensión, a la asignación eficiente de capital en la economía. Una herramienta de M&A screening accesible y reproducible reduce las barreras de entrada para actores con menos recursos, democratizando el acceso a análisis cuantitativo que hoy está reservado a grandes instituciones financieras.

ODS 9 – Industria, innovación e infraestructura.

La aplicación de técnicas de machine learning y análisis de datos al ámbito financiero representa una forma concreta de innovación tecnológica en un sector de alto impacto económico. El proyecto combina ingeniería de datos, modelado estadístico y desarrollo de software en un pipeline reproducible que puede servir de base para trabajos futuros en la intersección de la tecnología y las finanzas.

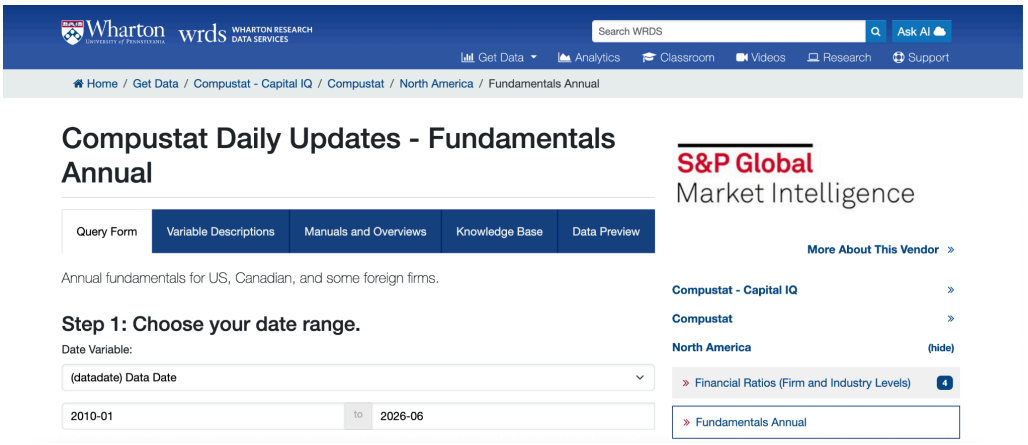
ANEXO II

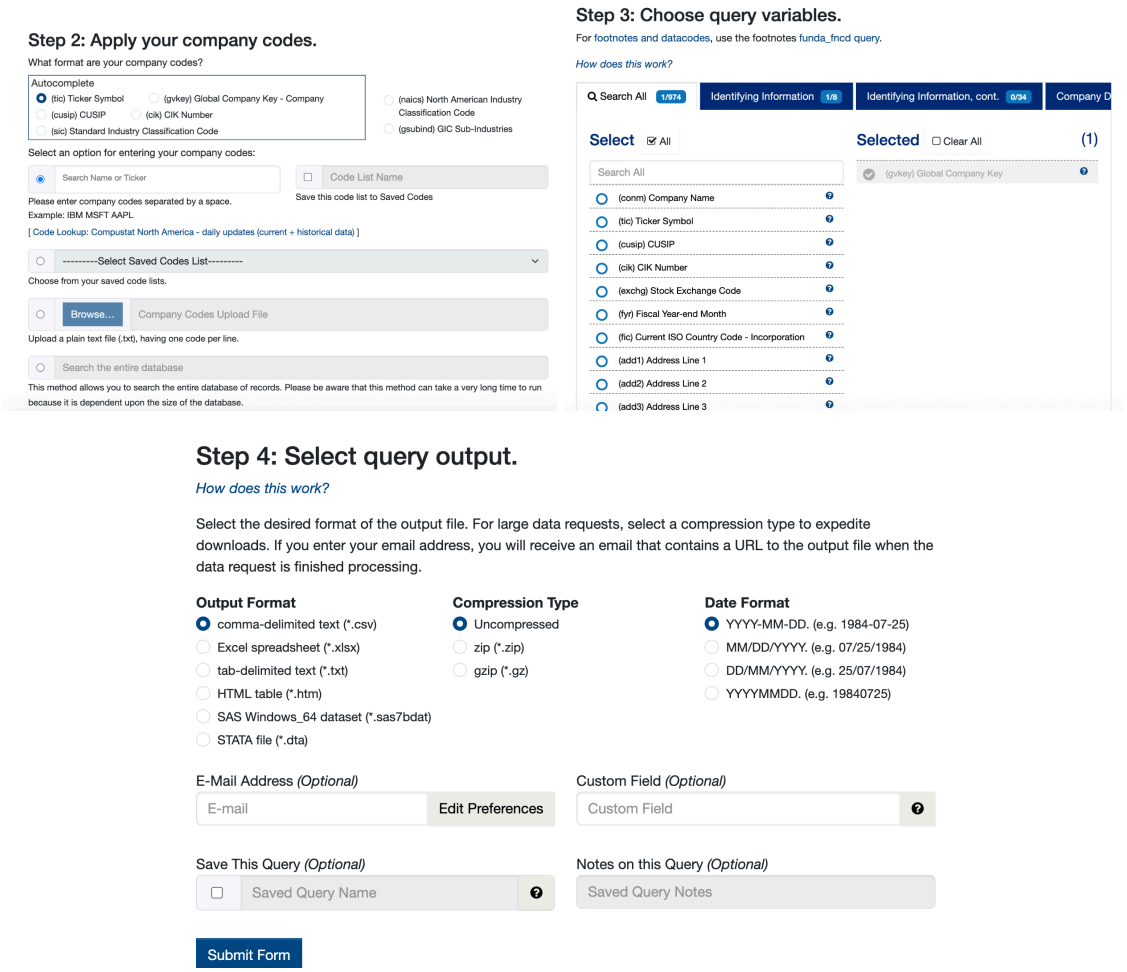
Anexo 1. Interfaz de PitchBook con filtros aplicados a operaciones de M&A

#	Companies (3,236)	Deal Date	Deal Type	Deal Type 2	Deal Size	Business Status	Investors	Primary Industry Code
2,025	Brand Connections	30-Jun-2008	Buyout/LBO	Corporate Dive...	150.00	Generating Revenue	VSS Capital Par...	Media and Inform...
2,026	Captain D's	27-Dec-2004	Buyout/LBO	Corporate Dive...	150.00	Generating Revenue	Charlesbank C...	Restaurants and B...
2,027	Catapult Learning	17-Mar-2008	Buyout/LBO		150.00	Generating Revenue	Chicago Growt...	Educational and T...
2,028	Columbian Chemic...	10-Nov-2009	Buyout/LBO	Secondary Buy...	150.00	Generating Revenue	One Equity Par...	Specialty Chemicals
2,029	CSG Systems Intern...	15-Mar-2005	Buyout/LBO		150.00	Profitable	Mercury Partn...	Systems and Infor...
2,030	CustomerContactC...	29-Dec-2016	Buyout/LBO	Secondary Buy...	150.00	Profitable	Everise (Sudhir ...	BPO/Outsource S...
2,031	Daily Racing Form	19-Oct-2007	Buyout/LBO	Secondary Buy...	150.00	Generating Revenue	Arlington Capit...	Publishing
2,032	DuBois Chemicals	30-Sep-2008	Buyout/LBO	Corporate Dive...	150.00	Profitable	New Canaan F...	Multi-line Chemic...
2,033	DURA Automotive S...	05-Jun-2020	Buyout/LBO		150.00	Generating Revenue	Bardin Hill Inve...	Road
2,034	eInstruction	04-Jul-2007	Buyout/LBO	Secondary Buy...	150.00	Profitable	HarbourVest P...	Educational Softw...
2,035	GTA TeleGuam	31-Dec-2004	Buyout/LBO		150.00	Generating Revenue	Shamrock Capi...	Telecommunicati...
2,036	HRI Properties	23-Oct-2014	Buyout/LBO		150.00	Generating Revenue	Almanac Realty...	Construction and ...
2,037	Invacare Supply Gr...	18-Jan-2013	Buyout/LBO	Add-on	150.00	Generating Revenue	AssuraMed (Mi...	Distributors (Heal...
2,038	Kasten (Business/Pr...	05-Oct-2020	Buyout/LBO	Add-on	150.00	Generating Revenue	Insight Partner...	Business/Producti...
2,039	Madison Logic	30-Nov-2016	Buyout/LBO		150.00	Profitable	Clarion Capital ...	Business/Producti...
2,040	Marietta Hospitality	17-Nov-2004	Buyout/LBO		150.00	Profitable	Ares Managem...	Personal Products
2,041	MBS Media Campus	01-Jun-2006	Buyout/LBO	Secondary Buy...	150.00	Generating Revenue	The Carlyle Gro...	Other Commercia...
2,042	MD Helicopters	01-Jul-2005	Buyout/LBO	Corporate Dive...	150.00	Generating Revenue	Patriarch Partn...	Aerospace and De...
2,043	Memjet	01-Jan-2012	Buyout/LBO	Secondary Buy...	150.00	Generating Revenue	George Kaiser ...	Industrial Supplie...
2,044	Neighborly	31-Jan-2011	Buyout/LBO	Secondary Buy...	150.00	Profitable	Ares Managem...	Other Commercia...
2,045	Nektar Therapeutic...	30-Dec-2020	Buyout/LBO	Asset Acquisiti...	150.00		HCRx	Buildings and Pro...
2,046	Nth Degree	26-Apr-2001	Buyout/LBO		150.00	Generating Revenue	Banc One Vent...	Media and Inform...
2,047	Omnium Worldwide	04-May-2007	Buyout/LBO	Add-on	150.00	Profitable	Quadrangle Gr...	Accounting, Audit ...
2,048	Optiv Security	22-Apr-2014	Buyout/LBO	Management ...	150.00	Profitable	Blackstone (NY...	IT Consulting and ...

Fuente: Pitchbook

Anexo 2. Interfaz de WRDS/Compustat con la selección de variables financieras.





Anexo 3. Tabla descriptiva de las variables del dataset final.

De izquierda a derecha: nombre de la variable en el modelo, nombre completo real, filas con datos completos, tipo de variable y definición.

Variable	Nombre completo	Non-null	Dtype	Definición
costat	Company Status	5071	str	Estado de la empresa en Compustat (activa/inactiva).
curcd	Currency Code	5071	str	Divisa en la que se reportan los datos (USD).
datafmt	Data Format	5071	str	Formato del dato (STD = estándar).
indfmt	Industry Format	5071	str	Formato de industria aplicado (INDL = industrial).
consol	Consolidation	5071	str	Tipo de consolidación contable (C = consolidado).
tic	Ticker Symbol	5071	str	Símbolo bursátil de la empresa.

datadate	Data Date	5071	str	Fecha de cierre del ejercicio fiscal.
gvkey	Global Company Key	5071	int64	Identificador único de empresa en Compustat.
conm	Company Name	5071	str	Nombre legal de la empresa.
exchg	Stock Exchange	5071	float64	Código de la bolsa donde cotiza la empresa (11=NYSE, 14=NASDAQ, 19=OTC).
fyear	Fiscal Year	5071	int64	Año fiscal del registro.
ap	Accounts Payable	5030	float64	Cuentas a pagar (pasivo corriente con proveedores), en millones USD.
at	Total Assets	5071	float64	Activos totales de la empresa, en millones USD.
che	Cash & Equivalents	5070	float64	Caja y activos líquidos equivalentes, en millones USD.
dlc	Debt in Current Liabilities	5062	float64	Deuda a corto plazo (vencimiento < 1 año), en millones USD. Eliminada por multicolinealidad con total_debt.
dltt	Long-Term Debt	5053	float64	Deuda a largo plazo (vencimiento > 1 año), en millones USD. Eliminada por multicolinealidad con total_debt.
lt	Total Liabilities	5061	float64	Pasivos totales de la empresa, en millones USD.
rect	Receivables	5036	float64	Cuentas a cobrar (deudores comerciales), en millones USD.
wcap	Working Capital	4045	float64	Capital circulante (activo corriente − pasivo corriente), en millones USD.
dp	Depreciation & Amortization	4852	float64	Dotación anual de amortización y depreciación, en millones USD.
ebitda	EBITDA	4899	float64	Resultado antes de intereses, impuestos, depreciación y amortización, en millones USD.
ni	Net Income	5071	float64	Beneficio neto del ejercicio, en millones USD.
oibdp	Operating Income Before D&A	4899	float64	Resultado operativo antes de D&A. Eliminada por colinealidad perfecta con ebitda (VIF = ∞).
sale	Net Sales	5071	float64	Ingresos netos por ventas, en millones USD.
xsga	SG&A Expenses	4225	float64	Gastos generales, de venta y administración, en millones USD.
capx	Capital Expenditures	4933	float64	Inversión en activos fijos (capex), en millones USD.
csho	Shares Outstanding	4944	float64	Número de acciones ordinarias en circulación, en millones.
prcc_f	Price Close (Fiscal)	4591	float64	Precio de cierre de la acción al final del ejercicio fiscal, en USD.
sich	SIC Code	4896	float64	Código SIC del sector industrial de la empresa.

company_clean	Nombre limpio	5071	str	Nombre de la empresa normalizado para el proceso de fuzzy matching con PitchBook.
ma_target	M&A target	5071	int64	Variable objetivo binaria: 1 si la empresa fue adquirida en el período 2005–2020 según PitchBook, 0 en caso contrario.
pb_company	PitchBook Company Name	1589	str	Nombre de la empresa en PitchBook. Solo existe para ma_target=1. Excluida del modelo (data leakage).
deal_date	Deal Close Date	1589	str	Fecha de cierre del deal M&A. Solo existe para ma_target=1. Excluida del modelo.
deal_size	Deal Size	1589	str	Tamaño del deal en USD. Solo existe para ma_target=1. Excluida del modelo.
deal_type2	Deal Type	212	str	Tipo específico de operación M&A. Solo existe para ma_target=1. Excluida del modelo.
match_score	Fuzzy Match Score	1589	float64	Puntuación de similitud del fuzzy matching (0–100). Umbral aplicado: 82%.
pb_industry_code	PB Industry Code	4947	str	Código de industria según clasificación PitchBook. Excluida del modelo.
pb_investors	PB Investors	1578	str	Inversores registrados en PitchBook. Solo existe para ma_target=1. Excluida del modelo.
pb_industry_sector	PB Industry Sector	1576	str	Sector industrial según PitchBook. Excluida del modelo.
pb_industry_group	PB Industry Group	1576	str	Grupo industrial según PitchBook. Excluida del modelo.
total_debt	Total Debt	5071	float64	Deuda total = dlc + dltt, en millones USD.
ebitda_margin	EBITDA Margin	5071	float64	Margen EBITDA = ebitda / sale. Mide la rentabilidad operativa.
net_margin	Net Margin	5071	float64	Margen neto = ni / sale. Eliminada por correlación 0,96 con ebitda_margin.
leverage	Leverage	5071	float64	Apalancamiento = total_debt / at. Proporción de activos financiados con deuda.
capex_intensity	Capex Intensity	4933	float64	Intensidad de capex = capx / sale. Proporción de ingresos destinada a inversión en capital.
roa	Return on Assets	5071	float64	Rentabilidad sobre activos = ni / at.
current_ratio	Current Ratio	4033	float64	Ratio corriente = wcap / lt. Mide la liquidez a corto plazo.
market_cap	Market Capitalization	4580	float64	Capitalización bursátil = csho × prcc_f, en millones USD.
ev_ebitda	EV/EBITDA	5071	float64	Múltiplo de valoración Enterprise Value / EBITDA. Estándar en valoración M&A.

asset_turnover	Asset Turnover	5071	float64	Rotación de activos = sale / at. Mide la eficiencia operativa.
cash_ratio	Cash Ratio	5070	float64	Ratio de caja = che / lt. Proporción de pasivos totales cubiertos con caja.

Anexo 4. Resultados de empresas con valores más extremos por variable.

```
--- ebitda_margin - Top 5 valores más extremos ---
                          conm  fyear  ma_target  ebitda_margin
1075  NORTHWEST BIOTHERAPEUTICS   2014          1     -66.738739
1695   SAREPTA THERAPEUTICS INC   2015          0     -66.738739
75                PHARMASSET INC   2010          1     -61.928431
1780            FUNCTION(X) INC   2012          0     -52.926225
1769              REGENETP INC   2018          0     -43.429303
--- Top 5 valores más altos ---
                          conm  fyear  ma_target  ebitda_margin
13              XTO ENERGY INC   2009          1       0.723258
126              AIRCASTLE LTD   2019          1       0.723258
127                      PIXAR   2005          1       0.723258
172          ATHLON ENERGY INC   2013          1       0.723258
355  KKR FINANCIAL HOLDINGS LLC   2013          1       0.723258

--- net_margin - Top 5 valores más extremos ---
                            conm  fyear  ma_target  net_margin
1075     NORTHWEST BIOTHERAPEUTICS   2014          1  -79.694444
1695      SAREPTA THERAPEUTICS INC   2015          0  -79.694444
75                  PHARMASSET INC   2010          1  -64.786275
1780              FUNCTION(X) INC   2012          0  -55.625937
4025  BELLICUM PHARMACEUTICALS INC   2014          0  -47.144862
```

Anexo 5. Implementación técnica del dashboard.

Este anexo resume la implementación técnica del dashboard desarrollado para visualizar y explotar los resultados del modelo de M&A screening. El objetivo no es documentar el código línea por línea, sino explicar los elementos principales que conectan el modelo entrenado con la herramienta visual: la preparación de datos, la aplicación del modelo, las rutas del backend y la comunicación con el frontend.

Anexo 5.1 Carga inicial de datos y modelo

Al ejecutar app.py, el servidor carga el universo de empresas y el modelo entrenado antes de que el usuario interactúe con el dashboard. Esto evita recalcular las predicciones en cada petición y mejora el rendimiento de la herramienta.

El proceso se inicia con la siguiente línea:

```
df_full, df_empresas, modelo_global, ES_DEMO = cargar_todo()
```

La función **cargar_todo()** centraliza la preparación inicial de la aplicación. Primero carga el universo de empresas, después importa el modelo guardado en formato pickle, aplica el modelo al universo completo y, finalmente, construye una tabla resumida con una fila por empresa.

De forma simplificada, el flujo es:

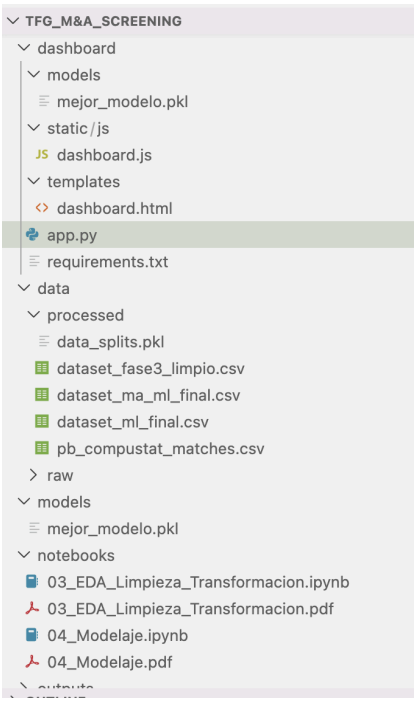
```
df = cargar_universo()

with open(MODEL_PATH, "rb") as f:
    modelo = pickle.load(f)

df = aplicar_modelo(df, modelo)
df_empresas = construir_tabla_empresas(df)
```

De esta forma, cuando el usuario abre el dashboard, los datos principales ya están cargados en memoria y las secciones pueden responder de forma más rápida.

Anexo 5.1.1. Consola al arrancar app.py con las respuestas del servidor.

```
(venv) (base) adrianazugasti@MacBook-Air-de-zugui dashboard % python app.py
⏳ Cargando universo de empresas...
   200,058 filas cargadas
   141,908 filas tras limpieza
Modelo aplicado al universo completo
   141,908 filas empresa-año -> 16,429 empresas únicas
Dashboard listo. 16,429 empresas unicas, 141,908 filas empresa-año en memoria
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
⏳ Cargando universo de empresas...
   200,058 filas cargadas
   141,908 filas tras limpieza
Modelo aplicado al universo completo
   141,908 filas empresa-año -> 16,429 empresas únicas
Dashboard listo. 16,429 empresas unicas, 141,908 filas empresa-año en memoria
 * Debugger is active!
 * Debugger PIN: 954-786-237
127.0.0.1 - - [30/Jun/2026 10:53:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:05] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:05] "GET /static/js/dashboard.js HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:06] "GET /api/summary HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:13] "GET /api/tickers HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:53:15] "GET /api/screener?pagina=1 HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:57:43] "GET /api/screener?pagina=1&sector=Financials&prob_min=0 HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 10:59:47] "GET /api/empresa/STD1 HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 11:01:38] "GET /api/empresa/UNWR HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 11:03:54] "POST /api/simular HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 11:03:58] "POST /api/simular HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 11:04:06] "POST /api/simular HTTP/1.1" 200 -
127.0.0.1 - - [30/Jun/2026 11:04:14] "POST /api/simular HTTP/1.1" 200 -
x^C
```

Anexo 5.1.2. Captura de la estructura de archivos del proyecto.



Anexo 5.2 Cálculo de variables financieras

El archivo USCompanies.csv contiene variables financieras brutas procedentes de Compustat. Sin embargo, el modelo final fue entrenado con un conjunto de variables que incluye ratios financieros derivados. Por ello, antes de aplicar el modelo, el backend recalcula esas variables siguiendo la misma lógica usada durante la creación del dataset de entrenamiento.

La función principal es **calcular_features()**:

```
def calcular_features(df):
    df = df.copy()

    df["total_debt"] = df["dltt"].fillna(0) + df["dlc"].fillna(0)
    df["market_cap"] = df["csho"] * df["prcc_f"]

    df["ebitda_margin"] = df["ebitda"] / df["sale"]
    df["roa"] = df["ebitda"] / df["at"]
    df["leverage"] = df["total_debt"] / df["at"]
    df["capex_intensity"] = df["capx"].fillna(0) / df["at"]
    df["asset_turnover"] = df["sale"] / df["at"]
```

```
        df["ev_ebitda"] = (df["market_cap"] + df["total_debt"] - df["che"]) /
df["ebitda"]

    denominador = (df["at"] - df["total_debt"]).replace(0, np.nan)
    df["cash_ratio"] = df["che"] / denominador

    return df
```

Esta función permite que el dashboard trabaje con las mismas variables utilizadas en la fase de modelaje. Por ejemplo, calcula la deuda total como la suma de deuda a largo y corto plazo, 'market cap' como acciones en circulación por precio de mercado, y ratios como leverage, ebitda_margin, asset_turnover o ev_ebitda.

Este paso es clave porque el modelo sólo puede generar predicciones coherentes si recibe las variables calculadas de la misma forma que durante el entrenamiento.

Anexo 5.3 Aplicación del modelo entrenado

Una vez calculadas las variables, el backend aplica el modelo Random Forest con SMOTE seleccionado como modelo final. Para ello se define una lista FEATURES, que contiene las 26 variables usadas por el modelo y mantiene exactamente el mismo orden que en el notebook de entrenamiento.

```
FEATURES = [
    "exchg", "ap", "at", "che", "lt", "rect", "wcap", "dp",
    "ebitda", "ni", "sale", "xsga", "capx", "csho", "prcc_f", "sich",
    "total_debt", "ebitda_margin", "leverage", "capex_intensity",
    "roa", "current_ratio", "market_cap", "ev_ebitda",
    "asset_turnover", "cash_ratio"
]
```

Mantener este orden es importante porque, al aplicar el modelo, las variables se introducen como un vector numérico. Si el orden cambiase, el modelo podría interpretar una variable como si fuera otra.
La predicción se realiza mediante la función **aplicar_modelo()**:

```
def aplicar_modelo(df, modelo):
    X = df[FEATURES].values
```

```
    df["prob_target"] = modelo.predict_proba(X)[:, 1]
    return df
```

El método **predict_proba()** devuelve la probabilidad de pertenecer a cada clase. En este caso se utiliza la probabilidad de la clase positiva, es decir, la probabilidad estimada de que la empresa sea target de M&A. Esta probabilidad se guarda en la columna prob_target, que después se utiliza para construir rankings, gráficos y señales de adquisición.

Anexo 5.4 Adaptación de datos panel a una tabla de empresas

Compustat tiene estructura de panel, por lo que una misma empresa puede aparecer en varios años fiscales. Para el dashboard, sin embargo, resulta más útil trabajar con una única fila por empresa en el ranking principal.

La función **construir_tabla_empresas()** resuelve este problema generando dos puntuaciones:

- score_reciente: score del año fiscal más reciente disponible.
- score_max: score máximo histórico de la empresa.
- año_max_score: año en el que se alcanza ese score máximo.

Fragmento simplificado:

```
def construir_tabla_empresas(df):
    df = df.sort_values(["tic", "fyear"])

    idx_reciente = df.groupby("tic")["fyear"].idxmax()
    base = df.loc[idx_reciente].copy()
    base = base.rename(columns={"prob_target": "score_reciente"})

    df_ordenado = df.sort_values(["tic", "prob_target", "fyear"])
    idx_max = df_ordenado.groupby("tic").tail(1).index

    maximos = df.loc[idx_max, ["tic", "fyear", "prob_target"]].rename(
        columns={"fyear": "año_max_score", "prob_target": "score_max"}
    )

    tabla = base.merge(maximos, on="tic", how="left")
    return tabla
```

Esta decisión permite que el dashboard distinga entre el atractivo actual de una empresa y el momento histórico en el que presentó un perfil financiero más parecido al de un target. En un proceso de screening, esta información es útil porque ayuda a contextualizar si la señal del modelo es reciente o responde a un perfil pasado.

Anexo 5.5 Explicación del backend por sección

5.5.1 Executive Summary

La ruta /api/summary calcula los indicadores principales del dashboard:

```
@app.route("/api/summary", methods=["GET"])
def get_summary():
    n_empresas = len(df_empresas)
    probs = df_empresas["score_reciente"].dropna()

    score_max = round(float(probs.max()) * 100, 1)
    señal_fuerte = int((probs > 0.35).sum())
```

Además del número de empresas, el score máximo y las empresas con señal fuerte, esta ruta devuelve el Top 10 de empresas, la distribución de probabilidades y el score medio por sector. Esta información alimenta la primera pantalla del dashboard.

5.5.2 M&A Screener

La ruta /api/screener permite filtrar el universo de empresas por sector, año fiscal y probabilidad mínima:

```
@app.route("/api/screener", methods=["GET"])
def get_screener():
    sector = request.args.get("sector", "")
    año = request.args.get("año", "")
    prob_min = float(request.args.get("prob_min", 0.0))
    pagina = int(request.args.get("pagina", 1))
    por_pag = 100
```

Después de aplicar los filtros, los resultados se ordenan por score_reciente y se devuelve solo la página solicitada:

```
df = df.sort_values("score_reciente", ascending=False)

inicio = (pagina - 1) * por_pag
df_pag = df.iloc[inicio: inicio + por_pag]
```

La paginación evita enviar demasiadas filas al navegador y hace que la herramienta sea más manejable para el usuario.

5.5.3 Company Deep Dive

La ruta /api/empresa/<ticker> permite analizar una empresa concreta. El backend localiza el ticker seleccionado, obtiene su score reciente, clasifica la señal como fuerte, moderada o débil y prepara su perfil financiero.

```
fila = df_empresas[df_empresas["tic"] == ticker]
row = fila.iloc[0]
prob = float(row.get("score_reciente", 0))

if prob >= 0.30:
    señal, color_cls = "Señal fuerte", "fuerte"
elif prob >= 0.18:
    señal, color_cls = "Señal moderada", "moderada"
else:
    señal, color_cls = "Señal débil", "debil"
```

Esta ruta también calcula las medianas del sector para comparar la empresa seleccionada con compañías similares:

```
df_sec = df_empresas[df_empresas["sector"] == sector]
medianas_sector = {
    k: round(float(df_sec[k].median()), 3)
    for k in financiero if k in df_sec.columns
}
```

Además, recupera el histórico de scores de la compañía a partir de df_full, lo que permite representar la evolución de la puntuación a lo largo de los años fiscales disponibles.

5.5.4 Simulador

La ruta /api/simular funciona mediante una petición POST, ya que recibe datos introducidos manualmente por el usuario. Primero extrae los valores financieros enviados desde el formulario:

```
datos = request.get_json()

sale = float(datos.get("sale", 0))
ebitda = float(datos.get("ebitda", 0))
at = float(datos.get("at", 0))
che = float(datos.get("che", 0))
total_debt = float(datos.get("total_debt", 0))
market_cap = float(datos.get("market_cap", 0))
```

Después calcula las variables derivadas necesarias:

```
ev_ebitda = (market_cap + total_debt - che) / ebitda if ebitda != 0 else 0
ebitda_margin = ebitda / sale if sale != 0 else 0
roa = ebitda / at if at != 0 else 0
leverage = total_debt / at if at != 0 else 0
```

Finalmente, construye el vector de entrada en el mismo orden que FEATURES y obtiene la probabilidad estimada:

```
X = np.array([[
    exchg, ap, at, che, lt, rect, wcap, dp,
    ebitda, ni, sale, xsga, capex, csho, prcc_f, sich,
    total_debt, ebitda_margin, leverage, capex_intensity,
    roa, current_ratio, market_cap, ev_ebitda,
    asset_turnover, cash_ratio
]])

prob = float(modelo_global.predict_proba(X)[0, 1])
```

El backend devuelve al frontend la probabilidad, la clasificación cualitativa, el percentil respecto al universo real y las variables derivadas calculadas. Esto permite utilizar el modelo con empresas que no estén incluidas en Compustat o con escenarios hipotéticos.

Anexo 5.6 Lógica del frontend

El archivo dashboard.js se encarga de conectar las acciones del usuario con las rutas del backend. Para ello utiliza **fetch()**, que permite realizar peticiones HTTP desde el navegador. Por ejemplo, cuando el usuario aplica filtros en el M&A Screener, el frontend construye una URL con parámetros:

```
const params = new URLSearchParams({ pagina: paginaActual });

if (sector) params.append("sector", sector);
if (año) params.append("año", año);
if (probMin) params.append("prob_min", probMin);

const res = await fetch(`/api/screener?${params}`);
const data = await res.json();
```

Después, con la respuesta recibida, la función **renderizarScreener()** actualiza la tabla HTML sin recargar la página completa.

En el simulador, el frontend envía los datos mediante una petición POST:

```
const res = await fetch("/api/simular", {
    method: "POST",
    headers: { "Content-Type": "application/json" },
    body: JSON.stringify(payload)
});
```

De esta manera, dashboard.js actúa como intermediario entre la interfaz y el backend: recoge inputs, envía peticiones, recibe respuestas JSON y actualiza los elementos visuales correspondientes.

Anexo 5.7 Renderizado visual e informes

El archivo dashboard.html define los elementos visuales que luego son rellenados dinámicamente por JavaScript. Por ejemplo, las tarjetas del Executive Summary tienen identificadores específicos:

```
<div class="kpi-valor" id="kpi-empresas">—</div>
<div class="kpi-valor" id="kpi-score-max">—</div>
<div class="kpi-valor" id="kpi-auc">—</div>
```

Cuando se carga la sección, dashboard.js recibe los datos de /api/summary y escribe los valores correspondientes en estos elementos.

El dashboard también utiliza Plotly para construir gráficos interactivos, como histogramas, gráficos por sector, indicadores tipo velocímetro, radar charts y líneas temporales. Además, se incluye una función para generar informes HTML descargables desde Company Deep Dive y desde el simulador. Esta descarga se realiza directamente desde el navegador:

```
function descargarHTML(htmlStr, nombre) {
    const blob = new Blob([htmlStr], { type: "text/html" });
    const a = document.createElement("a");
    a.href = URL.createObjectURL(blob);
    a.download = nombre;
    a.click();
}
```

Esto permite conservar un resumen de la empresa analizada o de la simulación realizada sin necesidad de almacenar información adicional en el servidor.