

# FACULTAD DE ESTUDIOS ESTADÍSTICOS

## MÁSTER EN CIENCIA DE DATOS E INTELIGENCIA DE NEGOCIOS

Curso 2024/2025

---

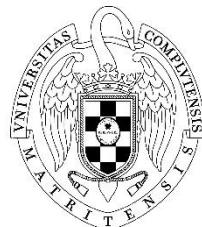
### Trabajo de Fin de Máster

***Titulo: Predicción de la inactividad de locales comerciales en Madrid mediante técnicas de Machine Learning***

**Alumno: Alejandro Bazán Guardia**

**Tutor: Belén Rodríguez-Cánovas**

Septiembre de 2025



UNIVERSIDAD COMPLUTENSE  
MADRID



*Quiero agradecer especialmente a mi madre, Judith, por su amor incondicional y apoyo constante; a mis tíos, Jenny y Jorge, por su cariño y aliento; a mis compañeros por su colaboración; y a la Facultad de Estadística de la Universidad Complutense por brindarme los conocimientos y recursos necesarios para completar esta tesis.*

## Resumen

Este Trabajo de Fin de Máster aborda la predicción del cierre de locales en la ciudad de Madrid, combinando datos administrativos del Ayuntamiento (2020–2024) enriquecidos con renta per cápita y población de cada área urbana. Se construyó una variable objetivo binaria (activo/inactivo) y se integraron más de ocho millones de registros, aplicando procesos de limpieza, estandarización y selección de variables (Boruta) antes del modelado. El conjunto de algoritmos evaluados incluye regresión logística, árboles de decisión, Random Forest, SVM, MLP, XGBoost, KNN y ensambles (Voting/Stacking), validados mediante validación cruzada estratificada y métricas. Los mejores resultados se obtuvieron con SVM, regresión logística, MLP y Voting. Las variables más determinantes fueron la renta media del entorno, el tipo de acceso (puerta de calle) y la pertenencia a determinadas secciones. Además, se identificaron patrones que permiten caracterizar los tipos de negocios más vulnerables y las zonas con mayor potencial de inversión en Madrid. El trabajo demuestra que modelos relativamente parsimoniosos, como el MLP compacto, ofrecen resultados robustos y estables, al tiempo que subraya la necesidad de mejorar la calidad de los datos. Como transferencia práctica, se propone desplegar una API interna y acompañar las predicciones con una explicación de estas con el método SHAP, lo que contribuirá tanto al Ayuntamiento de Madrid como a la ciudadanía, apoyando la toma de decisiones en política pública y la gestión comercial.

**Palabras clave:** locales , aprendizaje automático, datos administrativos, Madrid.

## Abstract

This Master's thesis addresses the prediction of business closures in the city of Madrid, combining administrative records from the City Council (2020–2024) enriched with per capita income and population data for each urban area. A binary target variable (active/inactive) was constructed and more than eight million records were integrated, followed by data cleaning, standardization, and feature selection (Boruta) prior to modeling. The set of evaluated algorithms includes logistic regression, decision trees, Random Forest, SVM, MLP, XGBoost, KNN, and ensemble methods (Voting/Stacking), validated through stratified cross-validation and standard metrics. The best results were achieved with SVM, logistic regression, MLP, and Voting. The most influential variables were neighborhood income, street-level access, and membership in specific territorial sections. In addition, patterns were identified that make it possible to characterize the types of businesses most vulnerable to closure and to highlight areas with greater investment potential across Madrid. The findings show that relatively parsimonious models, such as a compact MLP, can deliver robust and stable performance while also emphasizing the importance of improving data quality. As a practical contribution, the thesis proposes deploying an internal API and complementing predictions with explainability techniques (SHAP), thereby supporting decision-making for public policy and commercial management. Ultimately, the research provides value both to the Madrid City Council and to its citizens by guiding investment strategies and fostering sustainable urban development.

**Keywords:** business premises, machine learning, administrative data, Madrid

# Índice

Capítulo 1. Introducción .....	1
1.1 Contextualización del problema .....	1
1.2 Planteamiento del problema.....	3
1.3 Estado del arte .....	5
1.3.1 Enfoques fundacionales .....	5
1.3.2 Consolidación del machine learning.....	5
1.3.3 Auge de las redes neuronales para clasificación binaria.....	7
1.3.4 Dominio de los ensamblados avanzados .....	10
1.3.5 Integración de fuentes y modelos híbridos .....	11
1.4 Objetivos .....	12
Capítulo 2. Marco Teórico y Conceptual .....	13
Capítulo 3. Metodología .....	17
Capítulo 4. Datos .....	18
4.1 Fuente .....	18
4.2 Descripción de Variables.....	19
4.3 Construcción de la Variable Objetivo ("actividad") .....	20
4.4 Limitaciones del Dataset .....	21
Capítulo 5. Preparación de los Datos.....	21
5.1 Primer análisis exploratorio .....	21
5.2 Segundo análisis exploratorio después de la limpieza .....	24
5.3 Limpieza de Datos .....	31
5.3.1 Imputación de Datos Faltantes.....	31
5.3.2 Tratamiento de Coordenadas Geográficas.....	32
5.3.3 Recodificación de Variables .....	34
5.3.4 Revisión y Normalización del Texto .....	34
5.4 Preprocesamiento de los datos.....	35
5.4.1. Identificación y selección de variables categóricas .....	35
5.4.2 Transformación de variables categóricas: codificación one-hot.....	36
5.4.3 Observacion de variables no relevantes.....	36
5.4.4 Incorporación de la variable temporal para la segmentación.....	37
5.4.5 División del conjunto de datos en entrenamiento y prueba .....	37
5.4.6 Estandarización de variables numéricas .....	37
5.5 Selección de variables .....	38
5.5.1 Boruta .....	38
5.5.2 RFECV .....	39
5.5.3 Stepwise Logistic Regression .....	39
5.5.4 SBF (Sequential Backward Feature Selection).....	40
5.5.5 SHAP y XGBoost.....	41
5.5.6 Comparación de métodos de selección de variables.....	42
Capítulo 6. Modelización .....	43
6.1 Modelos entrenados: .....	43

6.1.1 Regresión logística .....	43
6.1.2 Arbol de decisión.....	45
6.1.3 XGBoost.....	47
6.1.4 KNN .....	49
6.1.5 Random Forest.....	50
6.1.6 SVM .....	52
6.1.7 Redes Neuronales .....	54
6.2 Ensamblado de modelos .....	55
6.2.1 VotingClassifier.....	55
6.2.2 StackingClassifier.....	57
Capítulo 7. Resultados.....	58
7.1 Comparación de modelos .....	58
7.2 Prueba de cambio de semilla .....	61
7.3 Análisis del modelo ganador .....	61
Capítulo 8. Conclusiones, recomendaciones y líneas futuras de investigación .....	66
8.1 Conclusiones principales .....	66
8.2 Recomendaciones .....	67
Bibliografía: .....	68

## Índice de tablas

<b>Tabla 1:</b> Lista y descripción de variables.....	19
<b>Tabla 2:</b> Renta media promedio per capita por distrito y año. Top 30 .....	27
<b>Tabla 3:</b> Renta Media por Barrio y Año (Ordenada Descendentemente). Top 20 .....	28
<b>Tabla 4:</b> Distritos con mayor % de locales inactivos (años 2023 y 2024) .....	29
<b>Tabla 5:</b> Barrios con mayor % de locales inactivos (años 2023 y 2024). Top 20 .....	30
<b>Tabla 6:</b> Comparación de Métodos de Selección de Variables: Precisión (Accuracy), AUC y Nº de Variables.....	43
<b>Tabla 7:</b> Resultados de validación cruzada .....	43
<b>Tabla 8:</b> Reporte de clasificación (test) .....	44
<b>Tabla 9:</b> Matriz de confusión .....	45
<b>Tabla 10:</b> Métricas de Evaluación por Clase y Globales — Árbol de decisión.....	45
<b>Tabla 11:</b> Resultados de validación cruzada .....	47
<b>Tabla 12:</b> Reporte de clasificación XGBoost .....	48
<b>Tabla 13:</b> Métricas de Clasificación del Modelo KNN en el Conjunto de Prueba .....	49
<b>Tabla 14:</b> Reporte de Clasificación – Test con Random Forest .....	51
<b>Tabla 15:</b> Reporte de Clasificación – Test con Random Forest .....	51
<b>Tabla 16:</b> Métricas de Desempeño – Modelo SVM en Test .....	53
<b>Tabla 17:</b> Matriz de confusión .....	53
<b>Tabla 18:</b> Matriz de confusión – MLP Red Neuronal.....	54
<b>Tabla 19:</b> Métricas Globales – MLP Red Neuronal.....	54
<b>Tabla 20:</b> Matriz de confusión – VotingClassifier .....	56
<b>Tabla 21:</b> Métricas– VotingClassifier.....	56
<b>Tabla 22:</b> Matriz de confusión StackingClassifier .....	57
<b>Tabla 23:</b> Reporte de Clasificación Completo StackingClassifier.....	58
<b>Tabla 24:</b> Comparación de métrica de los modelos .....	58
<b>Tabla 25:</b> Verdaderos negativos y especificidad de todos los modelos .....	59
<b>Tabla 26:</b> Top barrios con más actividad en comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas en Chamartín.....	63
<b>Tabla 27:</b> Epígrafe más representado de la sección comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas en el barrio de prosperidad .....	64
<b>Tabla 28:</b> Locales sección manufacturera interior y agrupado .....	65

# Índice de Ilustraciones

<b>Ilustración 1:</b> La distribución comercial en España: Número de ocupados en comercio al por menor .....	1
<b>Ilustración 2:</b> Tipología establecimientos comerciales .....	2
<b>Ilustración 3:</b> Crecimiento de sectores económicos en Madrid.....	3
<b>Ilustración 4:</b> Número de malas clasificaciones usando criterios basados en el puntaje z.....	5
<b>Ilustración 5:</b> Arquitectura de red neuronal MLP para la predicción de quiebra empresarial .....	6
<b>Ilustración 6:</b> Predicción vs. abandono real de clientes (Random Survival Forest) .....	7
<b>Ilustración 7:</b> Estadísticas de datos en los experimentos de Dianping y Yelp.....	8
<b>Ilustración 8:</b> Ejemplo de arquitectura del perceptrón multicapa. ....	9
<b>Ilustración 9:</b> Precisión del modelo de aprendizaje profundo en la muestra de comprobación. ....	10
<b>Ilustración 10:</b> Precisiones de clase y modelo.....	11
<b>Ilustración 11:</b> Informe nube de palabras y ranking de locales .....	12
<b>Ilustración 12:</b> Predicción de quiebras a nivel de empresa en la economía española .....	14
<b>Ilustración 13:</b> Tipos de métodos de aprendizaje automático para la previsión empresarial .....	15
<b>Ilustración 14:</b> Dos enfoques del aprendizaje automático para la toma de decisiones .....	16
<b>Ilustración 15:</b> Dos enfoques del aprendizaje automático para la toma de decisiones .....	24
<b>Ilustración 16:</b> Total de registros en la base de datos .....	24
<b>Ilustración 17:</b> Valores de población y de renta media per cápita anual (promedio, mínimo y máximo) por barrio en Madrid (2020-2024).....	25
<b>Ilustración 18:</b> Numero de registros del dataset por año y mes .....	25
<b>Ilustración 19:</b> Listado de variables del dataset .....	25
<b>Ilustración 20:</b> Distribución de datos por año del dataset .....	26
<b>Ilustración 21:</b> Matriz de correlación de variables numéricas.....	27
<b>Ilustración 22:</b> Evolución de la Renta Media por Distrito y Año .....	27
<b>Ilustración 23:</b> Distribución de la variable objetivo .....	28
<b>Ilustración 24:</b> Distribución de la variable objetivo de los años 2023 y 2024 .....	29
<b>Ilustración 25:</b> Análisis de frecuencia de palabras: Secciones Comerciales .....	31
<b>Ilustración 26:</b> Mapa de Madrid verificando correcta imputación de latitud y longitud.....	33
<b>Ilustración 27:</b> Mapa de Madrid verificando correcta imputación de latitud y longitud año 2023.....	33
<b>Ilustración 28:</b> Mapa de Madrid verificando correcta imputación de latitud y longitud año 2024.....	33
<b>Ilustración 29:</b> Variables train .....	38
<b>Ilustración 30:</b> Sharp Summary Plot.....	42
<b>Ilustración 31:</b> Sharp Bar Plot .....	42
<b>Ilustración 32:</b> Matriz de confusión .....	44
<b>Ilustración 33:</b> Curva ROC.....	44
<b>Ilustración 34:</b> Árbol de decisión.....	46
<b>Ilustración 35:</b> Curva de ROC - Árbol de decisión .....	46
<b>Ilustración 36:</b> Matriz de confusión - XGBoost .....	48
<b>Ilustración 37:</b> Curva de ROC - XGBoost.....	48
<b>Ilustración 38:</b> Matriz de confusión - KNN .....	50
<b>Ilustración 39:</b> Curva de ROC - KNN .....	50
<b>Ilustración 40:</b> Matriz de confusión Random Forest .....	52
<b>Ilustración 41:</b> Curva ROC - Random Forest.....	52
<b>Ilustración 42:</b> Curva ROC SVM .....	53
<b>Ilustración 43:</b> Curva ROC – Red neuronal MLP .....	55
<b>Ilustración 44:</b> Curva ROC – VotingClassifier.....	56
<b>Ilustración 45:</b> Grafico de la composición de StackingClassifier .....	57
<b>Ilustración 46:</b> Curva ROC StackingClassifier .....	58
<b>Ilustración 47:</b> Curva ROC – Todos los modelos .....	59
<b>Ilustración 48:</b> Métricas – Todos los modelos .....	59
<b>Ilustración 49:</b> Matriz de confusión – Todos los modelos .....	60
<b>Ilustración 50:</b> Sharp Summary Plot.....	62
<b>Ilustración 51:</b> Mapa 1 de locales de comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas la sección en Chamartín .....	63
<b>Ilustración 52:</b> Mapa 2 de locales de comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas la sección en el barrio de Prosperidad, Chamartín .....	63
<b>Ilustración 53:</b> Epígrafe comercio al por menor de prendas de vestir en establecimientos especializados en el barrio de Prosperidad, Chamartín.....	64
<b>Ilustración 54:</b> Mapa de locales inactivos de la sección .....	65

# Capítulo 1. Introducción

## 1.1 Contextualización del problema

Madrid, como capital del país y epicentro de su actividad económica, ha desarrollado a lo largo de las últimas décadas un tejido comercial sólido y diverso. El comercio minorista y los servicios de proximidad se han consolidado como pilares fundamentales en la estructura urbana, no solo por su capacidad para dinamizar el consumo, sino también por su contribución directa al empleo y al producto interior bruto (PIB) regional. En términos de superficie comercial, la ciudad encabeza el ranking nacional con más de 1,5 millones de metros cuadrados, triplicando el crecimiento medio del resto de comunidades autónomas en los últimos diez años. Esta concentración se traduce en alrededor de 60.000 empleos directos en el sector, lo que equivale al 26 % del total nacional, según el informe de la Asociación Nacional de Grandes Empresas de Distribución (2024).

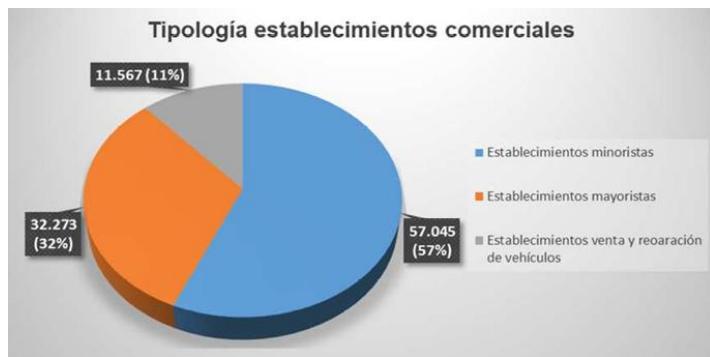
**Ilustración 1:** La distribución comercial en España: Número de ocupados en comercio al por menor (*miles de personas, tercer trimestre de 2024*).

	Total	Establecimientos no especializados	Alimentos, bebidas y tabaco en establecimientos especializados	Alimentos, bebidas y tabaco en establecimientos especializados	Resto
<b>Total</b>	<b>2.032</b>	<b>608</b>	<b>343</b>	<b>593</b>	<b>488</b>
Andalucía	356	103	61	107	86
Aragón	52	17	13	16	7
Asturias	49	22	7	12	9
Baleares	56	16	7	23	10
Canarias	142	56	14	44	29
Cantabria	24	10	4	6	4
Castilla y León	91	21	19	26	25
Castilla La Mancha	75	19	12	19	25
Cataluña	366	108	67	97	94
Valencia	218	69	24	57	69
Extremadura	39	12	7	10	11
Galicia	100	34	14	28	25
Madrid	300	74	68	100	59
Murcia	51	15	9	14	13
Navarra	25	10	4	6	4
País Vasco	73	22	11	26	15
La Rioja	10	2	3	2	3
Ceuta	7	1	1	3	2

**Fuente:** Informe económico del Comercio de la Asociación Nacional de Grandes Empresas de Distribución, 2024.

Según el informe El comercio de Madrid en cifras, elaborado por la Comunidad de Madrid (2025), el comercio representa el 12,2 % del PIB regional, de los cuales el comercio minorista aporta un 6,1 %. A comienzos de 2024, se contabilizaban 57.045 establecimientos minoristas activos, además de 32.273 comercios mayoristas y más de 11.500 puntos de venta y reparación de vehículos. El empleo ligado al sector comercial ascendía a casi 305.000 personas, lo que equivale al 8,9 % del total de ocupados en la región. Además, Madrid dispone de una red de 110 centros y parques comerciales que suman cerca de 3,2 millones de metros cuadrados, configurando un ecosistema de consumo robusto y variado. En paralelo, el comercio electrónico no ha dejado de crecer: solo en el tercer trimestre de 2024, la facturación del canal online en España superó los 24.000 millones de euros, y el 62,8 % de los madrileños afirmaron haber realizado compras por esta vía en los tres meses previos.

**Ilustración 2: Tipología establecimientos comerciales**



**Fuente:** Informe *El Comercio de Madrid en cifras* de la Comunidad de Madrid, 2024

No obstante, esta fortaleza se ha visto puesta a prueba desde 2020, cuando la irrupción de la pandemia por COVID-19 trastocó por completo los patrones de consumo y la operativa de los comercios. El cierre temporal de locales, la migración hacia entornos digitales y el encarecimiento de los costes operativos marcaron un punto de inflexión. Aunque en 2023 la facturación del sector creció un 5,2 % y las ventas online se multiplicaron por 2,6 respecto a 2019, según datos del informe de la Asociación Nacional de Grandes Empresas de Distribución (2024), la recuperación ha sido desigual. Algunas tipologías comerciales lograron adaptarse a la nueva realidad, mientras que otras quedaron rezagadas o incluso desaparecieron. El volumen de negocio en la Comunidad de Madrid ya supera en más de un 15 % los niveles prepandemia (Comunidad de Madrid, 2025), pero estas cifras agregadas ocultan diferencias marcadas entre barrios, distritos y zonas según su carácter turístico o residencial. Incluso los centros comerciales, tradicionalmente motores del consumo, han tenido que reinventarse incorporando actividades de ocio y experiencias para no perder competitividad frente al e-commerce (Escudero-Gómez, 2024).

En este escenario, el conflicto entre Rusia y Ucrania ha añadido un nuevo factor de inestabilidad. De acuerdo con un informe publicado por el Ministerio de Agricultura, Pesca y Alimentación (2024), entre marzo de 2022 y febrero de 2024 se produjo un notable incremento en el valor de las importaciones españolas de productos agroalimentarios, especialmente cereales y aceites procedentes de Ucrania. Este aumento se debió tanto a la subida de precios en los mercados internacionales como a la habilitación de corredores humanitarios que facilitaron las exportaciones ucranianas. Como consecuencia, sectores como la hostelería y la restauración que son fuertemente dependientes de estos insumos vieron incrementados sus costes operativos. A ello se sumaron las restricciones comerciales a productos rusos y la necesidad de diversificar socios comerciales, lo cual llevó a España a importar desde mercados más lejanos como Brasil, Estados Unidos o Canadá, impactando también en los precios al consumidor final (Ministerio de Agricultura, Pesca y Alimentación, 2024).

En este contexto de transformación, la economía madrileña ha demostrado signos de recuperación. Según un informe del Ayuntamiento de Madrid (2025), durante el año 2024 todos los sectores económicos crecieron: la industria lo hizo en un 1,3 %, la construcción en un 2,5 % y los servicios, principal motor económico, en un 3,7 %. Dentro del sector servicios, destacaron las actividades vinculadas al apoyo empresarial, con crecimientos del 4,4 % en el área de información y comunicaciones y del 3,9 % en las actividades

profesionales, científicas y técnicas. Estas cifras reflejan el dinamismo de una ciudad en constante evolución, que enfrenta retos globales con una estructura económica resiliente y en transformación.

**Ilustración 3: Crecimiento de sectores económicos en Madrid**

Ciudad de Madrid	2024
Industria	1,3
Construcción	2,5
Servicios	3,7
Comercio, transporte, almacenamiento y hostelería	3,4
Información y comunicaciones	4,4
Actividades financieras y de seguros	1,6
Actividades inmobiliarias	3,8
Actividades profesionales, científicas y técnicas; actividades administrativas y servicios auxiliares	3,9
Administración pública, sanidad, educación y servicios sociales	4,4
PIB	3,2

Fuente: SG Estadística Ayuntamiento de Madrid

Tras analizar el contexto, se optó por trabajar con una base de datos proporcionada por el Ayuntamiento de Madrid, la cual recopila información sobre todos los locales con actividad económica. Para esta investigación, se delimitó el periodo de estudio entre enero de 2020 y diciembre de 2024, incorporando además indicadores económicos y demográficos con el fin de construir una base de datos más robusta. Esta incluye variables como la geolocalización, la situación jurídica, los epígrafes de actividad económica, el tipo de acceso y el estado operativo de los locales. Al estar segmentada por distritos, barrios y secciones censales, representa una oportunidad valiosa para llevar a cabo un análisis integral de la evolución del comercio local, así como para desarrollar herramientas predictivas que resulten útiles tanto en la planificación pública como en la toma de decisiones del sector empresarial.

## 1.2 Planteamiento del problema

En un momento histórico marcado por transformaciones profundas en las ciudades, sorprende que, a pesar de la creciente disponibilidad de datos administrativos y del peso estratégico del comercio en la estructura urbana de Madrid, aún escaseen los estudios que analicen los cierres de locales desde una perspectiva territorial detallada. La mayoría de las investigaciones disponibles se enfocan en escalas agregadas como los centros comerciales o las zonas metropolitanas, lo que impide captar con precisión las dinámicas que se producen a nivel de barrio, calle o incluso manzana (Escudero-Gómez, 2024). Esta ausencia de enfoque microterritorial representa un obstáculo para diseñar políticas efectivas de conservación y reactivación del comercio local, especialmente en un contexto donde la pandemia, la aceleración digital y los cambios en los hábitos de consumo han dejado huellas desiguales.

En paralelo, estudios recientes han señalado que ciertos atributos del comercio, como su ubicación, el tipo de producto ofrecido, o incluso el ambiente percibido por el consumidor, pueden influir directamente en la probabilidad de cierre. No obstante, estos factores no siempre coinciden con las características tradicionalmente valoradas por la demanda (Kupfer et al., 2024). Asimismo, la incorporación de herramientas avanzadas, como los modelos de aprendizaje automático, ha abierto nuevas posibilidades para analizar estos fenómenos. Por ejemplo, en el ámbito sanitario, estas técnicas han permitido detectar

patrones de cierre ligados a variables estructurales, demográficas y territoriales (Park et al., 2024), aportando una visión más rica y contextualizada sobre la supervivencia de pequeñas entidades.

Aplicado al comercio minorista, se ha observado que aquellos negocios orientados hacia prácticas sostenibles o modelos de economía circular presentan una mayor capacidad de adaptación ante crisis prolongadas. Así lo evidencian modelos predictivos basados en redes neuronales profundas, los cuales han demostrado que estos enfoques no solo responden a una demanda emergente, sino que también aportan resiliencia frente a la incertidumbre (Uribe-Toril et al., 2022).

Desde esta perspectiva, el presente trabajo se plantea una serie de preguntas que guían la investigación y están directamente vinculadas con la situación de los locales madrileños tras la pandemia. ¿Qué factores explican que un local permanezca activo o pase a estar inactivo en la ciudad de Madrid? ¿Existen patrones diferenciados de cierre o supervivencia según el sector económico al que pertenece el establecimiento (hostelería, comercio, servicios personales, etc.)? ¿Hasta qué punto el barrio o distrito en el que se ubica el local influye en su continuidad operativa? ¿Cómo se relacionan las características socioeconómicas del entorno, como la renta media per cápita o la densidad poblacional por distrito, con el riesgo de cierre?

Para responder a estas cuestiones, se plantea el desarrollo de un modelo de predicción del cierre de locales en Madrid a partir de la aplicación de técnicas de aprendizaje automático supervisado. La investigación se apoyará en una base de datos administrativa que recoge información detallada sobre cada establecimiento, incluyendo coordenadas geográficas, tipo de acceso, clasificación de actividad económica, rótulo comercial y situación administrativa. A partir de la variable actividad creada, se construirá la codificación binaria que permitirá distinguir entre locales activos e inactivos, considerando cierres definitivos, bajas administrativas, reconversiones y situaciones especiales como obras.

El objetivo es entrenar y evaluar distintos modelos de clasificación que permitan identificar los factores asociados al riesgo de cierre. Para ello, se utilizará un enfoque comparativo basado en la implementación de diversos algoritmos de machine learning, como la regresión logística, los árboles aleatorios (Random Forest), el método de los vecinos más cercanos (K-Nearest Neighbors), máquinas de vectores soporte (Support Vector Machines), redes neuronales y modelos de gradiente como XGBoost y redes neuronales. Asimismo, se explorarán técnicas de ensamblado, incluyendo métodos de votación (Voting Classifier), con el fin de combinar las fortalezas de distintos algoritmos y mejorar el rendimiento predictivo.

Esta estrategia no solo permitirá evaluar la capacidad de predicción de cada técnica, sino también interpretar la relevancia de las variables explicativas en la probabilidad de cierre. El objetivo final es construir una herramienta robusta que contribuya a una planificación urbana más informada y a una toma de decisiones estratégicas por parte de actores públicos y privados en el ámbito comercial.

En definitiva, este trabajo busca contribuir a una comprensión más precisa y anticipatoria del comportamiento del comercio urbano en Madrid, en un contexto de acelerada transformación social, económica y tecnológica.

### 1.3 Estado del arte

La predicción de la supervivencia y el cierre de negocios ha evolucionado desde métodos estadísticos tradicionales hacia complejas arquitecturas de machine learning, consolidándose como un campo interdisciplinar que incorpora análisis econométrico, espacial, lingüístico y ensamblado de modelos.

#### 1.3.1 Enfoques fundacionales

Hasta finales del siglo XX, la predicción empresarial se apoyaba en métodos estadísticos clásicos. Cox (1972) desarrolló el modelo de riesgos proporcionales que facilita estimar la probabilidad de eventos como el cierre empresarial, incorporando variables macroeconómicas y sectoriales. Este modelo permitió cuantificar, por ejemplo, el impacto del desempleo o el crecimiento económico en la supervivencia de empresas, marcando un avance significativo en la comprensión cuantitativa del problema.

Por su parte, Altman (1968) introdujo el Z-score, una herramienta basada en ratios financieros que permite identificar empresas con alto riesgo de quiebra. Su simplicidad y efectividad facilitaron su adopción en la industria financiera, sirviendo como base para análisis binarios de éxito o fracaso empresarial. El modelo calcula un valor Z a partir de datos contables, y ese valor se interpreta de forma directa: si  $Z > 2.99$ , la empresa es solvente; si  $Z < 1.81$ , está en riesgo de quiebra; y si Z se encuentra entre 1.81 y 2.99, entra en una “zona gris” donde la clasificación es incierta. Para facilitar su uso sin apoyo computacional, Altman analizó los errores de clasificación en distintos rangos de Z y determinó que el punto óptimo de corte se encuentra en  $Z = 2.675$ , valor que minimiza los errores. Así, el Z-score no solo se consolida como un modelo predictivo robusto, sino también como una herramienta práctica y accesible para analistas financieros y gestores, incluso en contextos con recursos limitados.

**Ilustración 4:** Número de malas clasificaciones usando criterios basados en el puntaje z

TABLE 7  
NUMBER OF MISCLASSIFICATIONS USING VARIOUS Z SCORE CRITERIONS

Range of Z	Number Misclassified	Firms
1.81-1.98	5	2019, 1026, 1014, 1017, 1025
1.98-2.10	4	2019, 1014, 1017, 1025
2.10-2.67	3	2019, 1017, 1025
2.67-2.68	2	2019, 1025
2.68-2.78	3	2019, 2033, 1025
2.78-2.99	4	2019, 2033, 2032, 1025

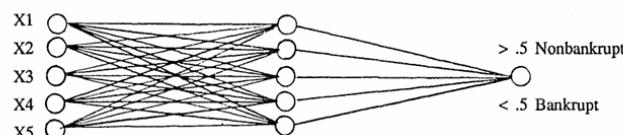
Fuente: Altman (1968)

#### 1.3.2 Consolidación del machine learning

La primera década del siglo XXI evidenció un crecimiento significativo en la aplicación de algoritmos de machine learning en problemas empresariales. Breiman (2001) diseñó Random Forest, un algoritmo de ensamblado de árboles que mejoró la generalización en problemas de clasificación, ofreciendo, además, indicadores interpretables sobre la relevancia de las variables. Simultáneamente, Cortes y Vapnik (1995) propusieron la Support Vector Machine (SVM), técnica robusta frente a alta dimensionalidad y ruido, que se convirtió rápidamente en una referencia en la predicción de riesgo empresarial.

En la misma línea, Odom y Sharda (1990) se enfocaron en analizar la capacidad de las redes neuronales Multilayer Perceptron (MLP) para captar relaciones no lineales entre indicadores financieros y el resultado de quiebra. Su investigación utilizó una arquitectura de tres capas ocultas con funciones sigmoidales, experimentando mejoras significativas en el área bajo la curva (AUC > 0,85) respecto a métodos estadísticos convencionales. Destacaron que las MLP, aplicadas con técnicas de regularización adecuadas, ofrecen potencia predictiva superior manteniendo control del sobreajuste. Este estudio abrió camino para posteriores investigaciones que fortalecieron la posición de las redes neuronales en este campo. Un ejemplo representativo de esta línea de trabajo lo encontramos en el estudio de los autores, donde entrenaron una red neuronal compuesta por una capa de entrada con cinco nodos (correspondientes a diferentes razones financieras), una capa oculta de cinco neuronas y una capa de salida con un solo nodo, encargado de clasificar a las empresas como quebradas o no. La red fue entrenada mediante el algoritmo de retropropagación del error, ajustando progresivamente los parámetros de aprendizaje y momento para mejorar su rendimiento. A pesar del elevado número de iteraciones necesarias para alcanzar la convergencia (191.400), los resultados fueron notables: la red clasificó correctamente la totalidad de las empresas en el conjunto de entrenamiento, superando claramente a métodos estadísticos como el análisis discriminante. Este trabajo no solo evidenció el potencial de las MLP en contextos financieros, sino que también puso de relieve la importancia de una configuración cuidadosa y un entrenamiento adecuado del modelo para obtener resultados fiables.

**Ilustración 5:** Arquitectura de red neuronal MLP para la predicción de quiebra empresarial



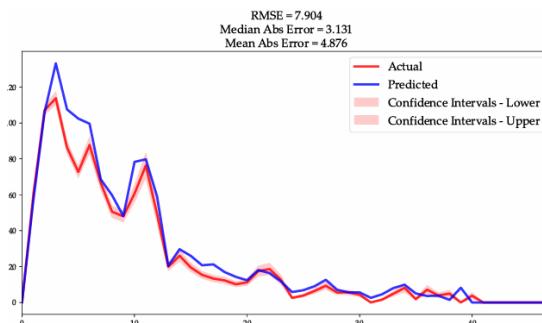
**Fuente 5:** Odom y Sharda (1990)

Con el avance del hardware y la acumulación de grandes volúmenes de datos en la década de 2010, Mogensen et al. (2012) evaluaron de forma sistemática la eficacia de los algoritmos Random Forest en el contexto del análisis de supervivencia, empleando métricas como la Brier score y curvas de error de predicción para comparar su rendimiento frente a modelos clásicos como la regresión de Cox. Su estudio, aplicado a datos clínicos con censura (pacientes con ictus), mostró que Random Forest — especialmente en sus versiones adaptadas al análisis de supervivencia como randomSurvivalForest y cforest — proporciona predicciones más robustas y menos

sensibles a los extremos que las técnicas paramétricas tradicionales. Esta capacidad de adaptación a datos de alta dimensión, sin requerir supuestos sobre la distribución subyacente, consolida a Random Forest como una herramienta versátil en contextos donde la variable objetivo es binaria y el desenlace está influido por factores complejos o parcialmente observables. En este sentido, su aplicación resulta altamente pertinente en escenarios como la predicción de la inactividad futura de locales comerciales, donde el modelo debe capturar interacciones no lineales y tratar con incertidumbre en los datos.

Sobreiro et al. (2022) extendieron esta línea de investigación enfocándose en la predicción del valor de vida y abandono de clientes en el sector retail, enfrentando la heterogeneidad de los datos transaccionales y sociodemográficos mediante modelos híbridos basados en Random Forest y Random Survival Forests. Su enfoque integró múltiples fuentes de información para capturar tanto el comportamiento de compra como la dinámica temporal del cliente. La evaluación del modelo de supervivencia mostró un alto rendimiento predictivo, con un Integrated Brier Score (IBS) de 0.08, un RMSE de 7.9 y un Error Absoluto Medio (MAE) de 4.88 clientes, evidenciando la precisión del modelo al predecir la deserción a lo largo de 40 meses. Además, se identificaron como variables más influyentes el monto total facturado, la frecuencia de uso y el número de accesos. Estos resultados confirman que la combinación de técnicas de ensamblado y análisis temporal proporciona mejoras sustanciales en la capacidad explicativa y predictiva de los modelos de retención de clientes. La figura del estudio ilustra visualmente este desempeño, mostrando el estrecho ajuste entre las predicciones y los datos reales de abandono, junto con los intervalos de confianza que respaldan la robustez del modelo.

**Ilustración 6:** Predicción vs. abandono real de clientes (Random Survival Forest)



Fuente: Sobreiro et al. (2022)

### 1.3.3 Auge de las redes neuronales para clasificación binaria

Tras el avance que supuso el uso de modelos de ensamblado y técnicas no paramétricas, la siguiente etapa en el desarrollo metodológico en predicción empresarial ha sido protagonizada por las redes neuronales, especialmente ante la llegada de grandes volúmenes de datos heterogéneos y no estructurados.

En este contexto, un ejemplo es la investigación de Li et al. (2022), el cual presentaron un modelo avanzado llamado RSPE (Restaurant Survival Prediction and Explanation) orientado a predecir de forma binaria si un local comercial permanecerá activo (1) o

cerrará (0) en un horizonte temporal futuro, combinando datos estructurados tradicionales con información no estructurada proveniente de reseñas de clientes en plataformas digitales. Este modelo integra un mecanismo de co-atención que selecciona los fragmentos textuales más relevantes para la predicción, junto con una red neuronal gráfica que captura las complejas interacciones entre usuarios y establecimientos, permitiendo detectar patrones difíciles de percibir con enfoques convencionales. Los experimentos realizados sobre conjuntos de datos reales de distintas regiones mostraron que RSPE mejora la precisión predictiva hasta en un 6,8% frente a métodos que solo usan variables numéricas, además de proporcionar explicaciones claras basadas en los textos de las reseñas para facilitar la interpretación y la toma de decisiones por parte de gestores y autoridades. La tabla de conjuntos de datos muestra la variabilidad de tasas de cierre en diferentes ciudades, subrayando la necesidad de modelos robustos que integren diversos tipos de datos. Asimismo, las nubes de palabras y las estadísticas de tasa de fallo evidencian que términos relacionados con la calidad del servicio y satisfacción del cliente correlacionan con la supervivencia de los locales, lo que RSPE aprovecha para generar predicciones más precisas y explicativas

*Ilustración 7: Estadísticas de datos en los experimentos de Dianping y Yelp*

dataset	#res- taurant	#closure restaurant	#closure ratio
Dian Ping	SH	10251	3312
	BJ	5067	1308
	GZ	1932	509
Yelp	NV	4764	223
	AZ	6623	258
	ON	5688	209

**Fuente:** Li et al. (2022)

En paralelo, Lahmí y Bekiros (2024) examinan la capacidad de diversos modelos de machine learning, incluyendo redes neuronales, para anticipar quiebras corporativas usando datos financieros cualitativos multivariados. Su enfoque destaca el auge de las redes neuronales artificiales como herramientas altamente eficaces para problemas de clasificación binaria, especialmente en contextos donde se busca determinar si una empresa está en riesgo de quiebra (clase positiva) o no (clase negativa).

A través de un diseño experimental riguroso, compararon el rendimiento de cuatro tipos distintos de redes neuronales: redes de retropropagación (BPNN), redes neuronales probabilísticas (PNN), redes de funciones de base radial (RBFNN) y redes neuronales de regresión generalizada (GRNN). Sus resultados revelaron que las redes GRNN y RBFNN no solo superaron significativamente a los modelos estadísticos tradicionales, sino también a otras arquitecturas de redes neuronales, alcanzando precisiones altas. El estudio subraya especialmente el valor de las redes neuronales en la minimización de errores tipo I (falsos negativos), que en este contexto implican clasificar erróneamente a una empresa en riesgo como solvente. La GRNN, por ejemplo, alcanzó una precisión del 99.96%, sensibilidad del 99.91% y especificidad del 100%, mientras que la RBFNN logró un 100% de sensibilidad, lo que la convierte en una arquitectura clave para detectar con anticipación señales de riesgo empresarial. Este auge de las redes neuronales para tareas de clasificación binaria se debe a su capacidad para modelar relaciones no lineales complejas, procesar datos cualitativos

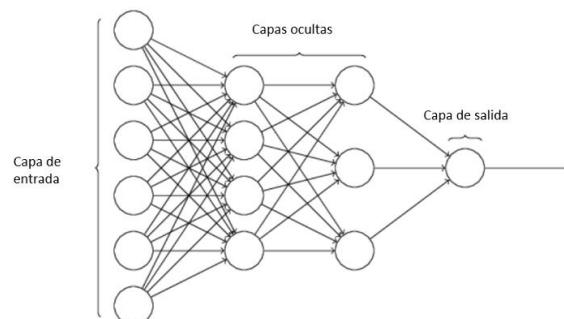
sin necesidad de supuestos estadísticos restrictivos, y adaptarse rápidamente incluso con muestras pequeñas. Además, su desempeño mejora la toma de decisiones en tiempo real al reducir significativamente el riesgo de pasar por alto entidades potencialmente insolventes. Así, Lahmiri y Bekiros no solo validan la eficacia de las redes neuronales en el pronóstico de quiebras, sino que también colocan a la GRNN y RBFNN en el centro de la como las mejores prácticas actuales en clasificación binaria dentro del ámbito financiero.

En los últimos años, las redes neuronales han cobrado un protagonismo creciente en el campo de la clasificación binaria, especialmente en el ámbito de la predicción empresarial. Su capacidad para procesar grandes volúmenes de datos, identificar patrones no lineales y adaptarse a distintos contextos las ha convertido en una herramienta de referencia frente a los modelos estadísticos tradicionales.

Una muestra destacada de esta tendencia es el estudio de Romero Martínez et al. (2021), que aplica redes neuronales profundas de tipo feedforward para predecir el fracaso empresarial en una muestra amplia de más de 61.000 compañías europeas. El modelo desarrollado logró una precisión del 94% al clasificar correctamente empresas activas frente a aquellas en situación concursal, utilizando información contable del año anterior. Este resultado no solo confirma la eficacia del enfoque, sino también su solidez, ya que fue validado sobre una muestra independiente distinta a la de entrenamiento.

Los autores subrayan que uno de los factores determinantes en el rendimiento del modelo fue la correcta configuración de hiperparámetros como la función de activación, el número de capas ocultas, el dropout y la regularización. Estos elementos permiten que la red aprenda de forma generalizable, evitando sobreajustes que comprometan su aplicabilidad en datos nuevos. Además, el estudio identificó variables clave en la predicción binaria, como el tamaño del activo, los ingresos de explotación y la solvencia financiera. La importancia de estas variables sugiere que el modelo es capaz de captar características estructurales que inciden directamente en el riesgo de inactividad o cierre de una unidad económica. Esta capacidad resulta especialmente útil en contextos donde se desea anticipar si un local comercial seguirá activo o cesará operaciones en el corto o mediano plazo. Por último, el uso de métricas como el AUC, la sensibilidad y la especificidad permite evaluar con mayor precisión el desempeño del modelo en términos de falsos positivos y negativos, lo cual es crucial en problemas donde una clasificación errónea puede tener consecuencias significativas, como ignorar un posible cierre inminente.

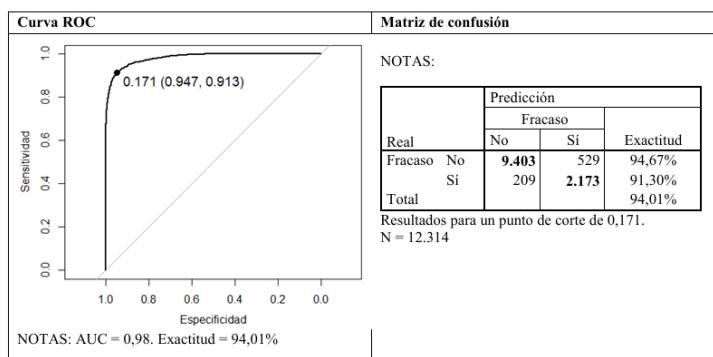
**Ilustración 8:** Ejemplo de arquitectura del perceptrón multicapa.



**Fuente:** Romero Martínez et al. (2021),

El estudio muestra que el modelo de aprendizaje profundo, al ser evaluado con una muestra de comprobación independiente no utilizada en el entrenamiento, alcanzó un alto nivel de precisión, con un AUC de 0,9816 y una exactitud global del 94,01%. Además, logró clasificar correctamente el 94,67% de las empresas activas (especificidad) y el 91,30% de las empresas fracasadas (sensibilidad), lo que demuestra su solidez para predecir de forma fiable el estado futuro de una empresa, evitando problemas de sobreajuste y confirmando su capacidad de generalización.

**Ilustración 9:** Precisión del modelo de aprendizaje profundo en la muestra de comprobación.



**Fuente:** Romero Martínez et al. (2021),

#### 1.3.4 Dominio de los ensamblados avanzados

En los años recientes, en la literatura sobre predicción de quiebras, los modelos de ensamblado avanzados, en particular Extreme Gradient Boosting (XGBoost), han demostrado un rendimiento notable frente a métodos tradicionales. El estudio de Shetty, Musa y Brédart (2022) compara XGBoost con redes neuronales profundas y máquinas de vectores soporte (SVM) utilizando datos financieros básicos de más de 3.700 PYMEs belgas. A pesar de trabajar con un conjunto reducido de variables, rentabilidad sobre activos, ratio de liquidez y solvencia, XGBoost alcanzó una precisión global del 83%, superando ligeramente a los otros modelos y mostrando una mayor estabilidad en la clasificación de empresas solventes. Aunque los autores reconocen una limitación inherente debido a la superposición de clases en los datos, los resultados confirman que XGBoost, como técnica de ensamblado, ofrece un equilibrio sólido entre simplicidad y potencia predictiva, especialmente en contextos con datos heterogéneos o desequilibrados como el de las pequeñas y medianas empresas.

**Ilustración 10: Precisiones de clase y modelo**

Method	Class/Total	Precision	Recall	f1-Score
Neural Net	0	85	79	82
	1	79	82	82
	Total	82	81	82
SVM	0	85	81	83
	1	80	84	82
	Total	83	83	83
XGBoost	0	84	81	83
	1	81	82	83
	Total	83	82	83

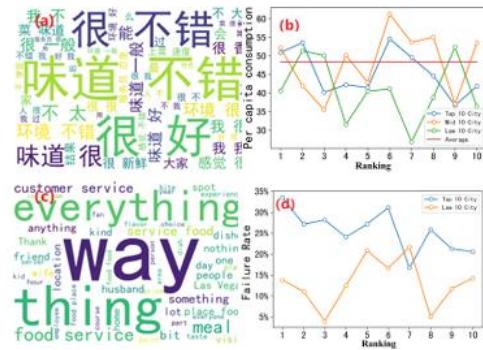
**Fuente:** Shetty, Musa y Brédart (2022)

En el ámbito de los modelos de ensamblado aplicados a problemas de predicción binaria, el trabajo de Sobreiro et al. (2022) constituye una referencia relevante. Su investigación se orienta a estimar la deserción de clientes en un entorno real, definiendo como variable objetivo un estado binario (1 = abandono, 0 = permanencia activa) sobre el cual construyen un modelo híbrido que combina Random Survival Forests con técnicas de clustering. Este enfoque integra la capacidad de los bosques aleatorios para modelar relaciones no lineales con la segmentación adaptativa de los datos, generando submodelos especializados para diferentes grupos de comportamiento. Aunque su aplicación original está dirigida a la predicción de churn, la metodología resulta análoga para escenarios donde se busca anticipar cambios de estado en entidades, como la estimación de si locales permanecerán activos o pasarán a inactivos. Los resultados empíricos muestran que el ensamblado híbrido reduce significativamente métricas de error como el Integrated Brier Score y el Mean Absolute Error frente a modelos individuales, además de aportar mayor estabilidad frente a la variabilidad de los datos. Las conclusiones destacan que la combinación de técnicas de aprendizaje supervisado con métodos de segmentación no solo incrementa la precisión, sino que también mejora la resiliencia de las predicciones en contextos dinámicos, posicionando este tipo de ensamblados avanzados como herramientas de alto valor para la toma de decisiones estratégicas en entornos de negocio caracterizados por la incertidumbre y la necesidad de anticipar estados futuros.

### 1.3.5 Integración de fuentes y modelos híbridos

Con la creciente disponibilidad de datos geoespaciales y textuales desde 2020, han surgido modelos que exploran estas nuevas fuentes para enriquecer la predicción empresarial. Li et al. (2022) propusieron un modelo que combina variables espaciales (localización geográfica, movilidad urbana) con análisis de contenido textual proveniente de reseñas de clientes, con el objetivo de anticipar cierres en restaurantes. Su enfoque buscó aprovechar la complementariedad entre información estructurada y no estructurada para mejorar la interpretabilidad y la precisión. Los resultados mostraron aumentos significativos en ambos aspectos, resaltando que los factores cualitativos y contextuales son cruciales para entender el fenómeno de la supervivencia empresarial.

**Ilustración 11: Informe nube de palabras y ranking de locales**



Fuente: Li et al. (2022)

Finalmente estudio de Stempień y Ślepaczuk (2024) consistió en probar diferentes modelos híbridos para predecir la evolución de dos series financieras muy distintas: el índice S&P 500 y Bitcoin, usando más de veinte años de datos históricos. Para ello combinaron modelos estadísticos tradicionales (ARIMA y ARFIMA), que detectan patrones lineales, con técnicas más avanzadas como máquinas de soporte vectorial (SVM), XGBoost y redes neuronales LSTM, capaces de encontrar relaciones no lineales y más complejas. Aplicaron dos formas de hibridación: una en la que los modelos trabajan por separado y luego se suman sus resultados, y otra donde la predicción de un modelo alimenta directamente al siguiente. Además de medir la precisión de las predicciones (RMSE, MAE), evaluaron si estos modelos podían traducirse en estrategias prácticas y resistentes frente a cambios bruscos en el mercado. Los resultados mostraron que las combinaciones ARIMA-SVM y ARIMA-LSTM lograron predicciones más estables que los modelos individuales, sobre todo en escenarios de alta incertidumbre.

#### 1.4 Objetivos

- General: El propósito principal de este trabajo es diseñar un modelo predictivo capaz de estimar la probabilidad de cierre futuro de locales comerciales en la ciudad de Madrid. Dado que el problema plantea una tarea de clasificación binaria si en un futuro estaría local activo o no activo, se han empleado diversas técnicas de machine learning que permiten modelar relaciones complejas entre variables geográficas, económicas, demográficas y comerciales. Se han probado modelos tanto lineales como no lineales, incluyendo regresión logística, árboles de decisión, bosques aleatorios, máquinas de vectores de soporte (SVM), redes neuronales (MLP), modelos de boosting (XGBoost) y ensambles como StackingClassifier. Esta variedad de algoritmos permite comparar enfoques con diferentes capacidades de modelado y niveles de interpretabilidad, con el objetivo de seleccionar el modelo que ofrezca el mejor equilibrio entre rendimiento predictivo y comprensión del fenómeno estudiado. La selección de variables se realizó con Boruta para reducir dimensionalidad y centrar el análisis en las variables más relevantes.

Específicos:

- Organizar y preparar el conjunto de datos disponibles, asegurando la calidad, limpieza y transformación adecuada de las variables relacionadas con la ubicación del local, el tipo de actividad, la situación censal, los ingresos del área y otros factores contextuales.
- Comparar diferentes algoritmos de predicción como árboles de decisión, modelos de ensamble o redes neuronales, con el fin de seleccionar aquel que ofrezca mejores resultados en cuanto a precisión y capacidad de generalización en la clasificación de locales abiertos o cerrados.
- Entrenar y validar los modelos construidos, utilizando criterios objetivos de evaluación como son los siguientes: exactitud, matriz de confusión, área bajo la curva ROC y métodos de validación cruzada para reducir el riesgo de sobreajuste.
- Analizar la relevancia de las variables utilizadas, con el objetivo de identificar los factores que tienen mayor peso en la supervivencia o el cierre de un local comercial en la ciudad.
- Clasificar los locales según su nivel de riesgo de cierre, permitiendo generar alertas tempranas o mapas de calor que visualicen zonas críticas y orienten futuras acciones por parte de las administraciones gubernamentales, empresas interesadas o la misma ciudadanía.

## Capítulo 2. Marco Teórico y Conceptual

La historia de los modelos de aprendizaje automático aplicados a la toma de decisiones comienza a principios de siglo. Según el artículo de Breiman (2001), Random Forests se define como un conjunto de clasificadores de tipo árbol, donde cada uno se construye a partir de un vector aleatorio independiente, pero con la misma distribución, y cuya predicción final se obtiene mediante votación mayoritaria de los árboles. Los Decision Trees son la base estructural de estos clasificadores, construidos sin poda y mediante divisiones sucesivas de nodos con base en características seleccionadas aleatoriamente o por combinaciones lineales. Bagging, también propuesto previamente por el autor, consiste en generar múltiples conjuntos de entrenamiento mediante remuestreo con reemplazo, y entrenar clasificadores sobre estos para luego agregarlos. El concepto de VotingClassifier se encuentra implícito en la lógica del modelo: cada árbol emite un voto, y la clase más votada es la predicción del modelo completo.

Años más tarde, la atención comenzó a dirigirse hacia problemas empresariales específicos. En su estudio, Smith y Álvarez (2022) analizan la predicción de quiebras empresariales utilizando el algoritmo XGBoost, el cual se basa en árboles de decisión y optimiza una función de pérdida que combina errores de predicción con un componente de penalización por complejidad, incorporando técnicas de regularización para prevenir el sobreajuste. Este modelo asigna puntuaciones a las empresas según sus características, las cuales se suman y transforman en probabilidades mediante una función logística. Los autores también discuten el uso de la regresión logística como modelo tradicional, destacando su simplicidad y ausencia de supuestos sobre la distribución de variables, aunque limitada frente a relaciones no lineales. El problema se aborda como una clasificación binaria (empresa en quiebra o no), evaluado mediante métricas como la exactitud, sensibilidad, especificidad, precisión y F1-score. Además, se analizan las curvas ROC y su área bajo la curva (AUC) como indicadores de desempeño general del modelo, aunque para datos desbalanceados se considera más adecuada la curva Precision-Recall y su respectiva métrica (AUPRC). Para optimizar el modelo, se aplican técnicas como la

validación cruzada de k pliegues y la búsqueda en rejilla, con el objetivo de ajustar los hiperparámetros y evitar el sobreajuste durante el entrenamiento.

En esta misma línea temporal, según Smith y Álvarez (2022), evaluar el desempeño de un modelo requiere métricas que aborden diferentes perspectivas. La exactitud (accuracy) mide la proporción de aciertos totales, aunque en clases desbalanceadas puede ser engañosa. Para estos casos, recomiendan métricas como la sensibilidad (recall), que mide la capacidad de detectar correctamente casos positivos, y la precisión (precision), que calcula cuántos de los casos predichos como positivos lo son realmente. Los autores destacan la importancia del  $F_1$  score, que combina precisión y sensibilidad de forma armónica, resultando útil cuando es necesario equilibrar falsos positivos y falsos negativos. También señalan que la métrica AUC (Área Bajo la Curva ROC) evalúa la capacidad del modelo para distinguir entre clases, mientras que en problemas con clases poco frecuentes es preferible la métrica AUPRC (Área Bajo la Curva Precisión–Recall) por ofrecer una representación más fiel de la detección de casos positivos.

**Ilustración 12:** Predicción de quiebras a nivel de empresa en la economía española

**Table 4** Comparison to other machine learning methods for 1 year prior predictions

Metric	XGBoost	Logistic	Light GBM	Shallow NN	Deep NN	R. Forest	SVM (R)	SVM (L)
Accuracy	0.87	0.91	0.89	0.91	0.91	0.92	0.90	0.91
Sensitivity	0.73	0.25	0.61	0.32	0.30	0.37	0.16	0.18
Specificity	0.88	0.99	0.92	0.98	0.98	0.99	0.99	0.99
Precision	0.42	0.66	0.47	0.60	0.62	0.79	0.66	0.71
$F_1$	0.54	0.36	0.53	0.42	0.41	0.51	0.26	0.29
MCC	0.49	0.37	0.47	0.39	0.39	0.51	0.30	0.33
AUC	0.81	0.81	0.76	0.65	0.64	0.68	0.58	0.59

NN neural network, SVM support vector machine with (R) radial, (L) linear kernel

**Fuente:** Smith y Alvarez (2022)

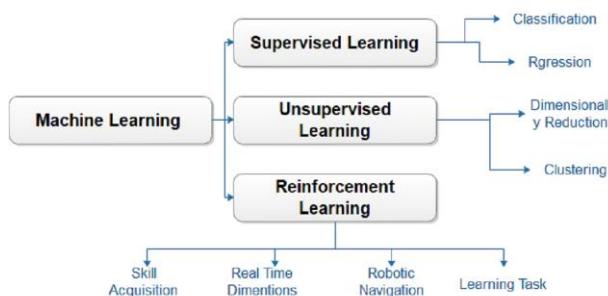
Un paso importante se dio también en 2022 con el trabajo de Uribe-Toril et al., cuyo modelo aplicado en su investigación es una Artificial Neural Network (ANN), concretamente un Multilayer Perceptron (MLP), el cual se compone de múltiples capas ocultas de neuronas conectadas entre sí. En su caso, el MLP se diseñó con tres capas ocultas de 100, 150 y 200 neuronas, respectivamente, utilizando la función de activación ReLU para las capas ocultas y sigmoide para la capa de salida. Para evaluar la capacidad predictiva del modelo, se utilizó la métrica de accuracy, que representa el porcentaje de aciertos obtenidos sobre el total de predicciones, logrando una precisión del 98.6%. A fin de evitar problemas de overfitting, el conjunto de datos se dividió en 80% para entrenamiento y 20% para prueba, una técnica conocida como train/test split, además de aplicar procedimientos de validación cruzada (cross-validation) para asegurar la generalización del modelo. Durante la etapa de preprocesamiento, las variables categóricas, como los nombres de las empresas, fueron transformadas mediante one-hot encoding, permitiendo así su uso en la red neuronal al convertir las categorías en vectores binarios compatibles con el modelo de aprendizaje automático.

Con la consolidación de estas bases, nuevos enfoques aparecieron. En el estudio de Vallapuram et al. (2022), el desarrollo de enfoques supervisados y la clasificación binaria se centran en predecir la supervivencia de negocios. Para ello, los autores formulan una

tarea de clasificación supervisada binaria, donde la etiqueta a predecir indica si un negocio permanece abierto o cierra en un periodo determinado. Este enfoque requiere contar con datos históricos etiquetados, es decir, información de negocios que efectivamente sobrevivieron o no, lo cual permite entrenar modelos predictivos. Entre los modelos utilizados se encuentran árboles de decisión potenciados (GBDT), máquinas de vectores de soporte (SVM), regresión logística (LR) y perceptrones multicapa (MLP), evaluados principalmente mediante el área bajo la curva ROC (AUC), dada la naturaleza desequilibrada del conjunto de datos. Además, para mejorar la interpretación de los modelos se emplea la técnica LIME, que permite comprender las decisiones del modelo al resaltar qué características influyen en la predicción de cierre o supervivencia. Este enfoque no solo busca precisión en la clasificación, sino también ofrecer explicaciones comprensibles y útiles para los dueños de negocios, lo que distingue este trabajo de enfoques previos puramente cuantitativos o difíciles de interpretar.

Con el paso del tiempo, los modelos comenzaron a diversificarse en aplicaciones más amplias. Según Ahamed et al. (2023), el aprendizaje automático se divide en tres enfoques principales: aprendizaje supervisado, no supervisado y por refuerzo, cada uno con aplicaciones específicas en la predicción y toma de decisiones empresariales. El aprendizaje supervisado se utiliza comúnmente para tareas de clasificación y regresión, donde los modelos aprenden a partir de datos etiquetados para predecir comportamientos futuros, mientras que el aprendizaje no supervisado identifica patrones y estructuras ocultas mediante técnicas como el clustering o la reducción de dimensionalidad, sin necesidad de datos etiquetados. Por su parte, el aprendizaje por refuerzo se aplica en contextos dinámicos donde un agente optimiza sus decisiones a través de la retroalimentación del entorno.

**Ilustración 13:** Tipos de métodos de aprendizaje automático para la previsión empresarial

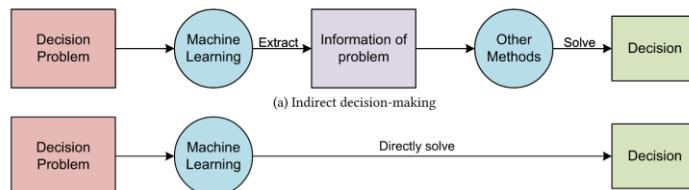


**Fuente:** Ahamed et al. (2023)

La evolución reciente muestra un salto hacia aplicaciones en ciudades inteligentes y modelos complejos. Según Zheng et al. (2024), más de la mitad de la población mundial vive actualmente en ciudades, las cuales se han convertido en centros clave de actividad económica. En este contexto, los autores señalan que el uso de tecnologías avanzadas, como el aprendizaje automático, es esencial para construir ciudades inteligentes que impulsen el desarrollo sostenible y mejoren la calidad de vida. Destacan que la creación de modelos inteligentes que apoyen la toma de decisiones urbanas constituye un elemento central para alcanzar estos objetivos. Los autores también destacan que el aprendizaje automático ya se aplica en planificación urbana, transporte y sanidad. Esto evidencia su

utilidad frente a retos como la distribución del uso de suelo, la ubicación de infraestructuras públicas o la optimización del tráfico.

**Ilustración 14:** Dos enfoques del aprendizaje automático para la toma de decisiones



Fuente: Zheng et al. (2024)

Los autores distinguen tres enfoques principales: supervisado, no supervisado y por refuerzo. Mientras los dos primeros se orientan al análisis de datos históricos, el aprendizaje por refuerzo introduce un enfoque adaptativo, capaz de tomar decisiones en tiempo real para maximizar recompensas específicas. Resaltan que los modelos de refuerzo profundo pueden reducir de forma drástica los tiempos de decisión urbana, acelerando la asignación de funciones en ciudades inteligentes. El autor hace entender que el machine learning aplicado a la toma de decisiones puede abordarse de dos maneras principales. En un enfoque indirecto, los modelos se entrena para extraer información clave del problema, como patrones o predicciones, que luego sirven de entrada a otros métodos que generan la decisión final. En contraste, el enfoque directo permite que el modelo aprenda de manera end-to-end a transformar los datos de entrada en decisiones, ajustando sus parámetros mediante funciones de pérdida como el error cuadrático medio o la entropía cruzada.

En el ámbito de la modelización de supervivencia, Langbein et al. (2024) sostienen que técnicas de machine learning como los bosques aleatorios de supervivencia, los árboles potenciados y las redes neuronales adaptadas a este tipo de análisis se han consolidado como herramientas eficaces. Estas metodologías han ganado protagonismo debido a su capacidad para modelar grandes volúmenes de datos y captar relaciones complejas que los enfoques tradicionales, como el modelo de riesgos proporcionales de Cox, no siempre logran representar adecuadamente. En particular, los bosques aleatorios de supervivencia, al tratarse de métodos de ensamble basados en árboles de decisión, ofrecen una alternativa robusta y no paramétrica para predecir tiempos hasta eventos, siendo especialmente útiles en contextos con alta dimensionalidad y presencia de censura. Por otro lado, las redes neuronales orientadas al análisis de supervivencia permiten capturar patrones no lineales y efectos que varían en el tiempo, adaptándose a estructuras de datos complejas mediante funciones de riesgo aprendidas automáticamente. Este tipo de modelos resulta especialmente adecuado cuando las suposiciones de proporcionalidad del riesgo no se cumplen, lo que las convierte en herramientas de alto potencial para investigaciones biomédicas y clínicas donde la dinámica temporal es clave.

Por su parte, en su estudio, Lee et al. (2024) describen el aprendizaje supervisado como un enfoque donde modelos predictivos se entrena con datos históricos que contienen variables de entrada y salidas conocidas, como el número de tiendas, ventas promedio,

población residente y población trabajadora, lo que permite al modelo aprender patrones para hacer predicciones precisas. Utilizan distintos modelos supervisados, incluyendo regresión lineal y redes neuronales profundas, que se ajustan a los datos mediante entrenamiento supervisado. Aunque el enfoque principal es la predicción de variables continuas, los autores indican que esta metodología también puede adaptarse a clasificación binaria, permitiendo etiquetar unidades comerciales como “exitosas” o “en riesgo” en función de características relevantes. Además, subrayan la importancia de la selección de características, utilizando análisis de correlación para identificar las variables más influyentes y optimizar el rendimiento del modelo. Finalmente, la validación del modelo se realiza mediante métricas como el error cuadrático medio (RMSE), que permite medir la precisión de las predicciones frente a los datos reales, asegurando la confiabilidad del sistema propuesto.

Finalmente, según el autor Lu et al. (2024), entre los modelos de aprendizaje automático más relevantes para tareas de predicción se encuentran Random Forest (RF), Redes Neuronales Artificiales (Neural Networks, NN) y Gradient Boosted Decision Trees (GBDT). El modelo Random Forest se basa en un conjunto de árboles de decisión entrenados sobre distintos subconjuntos del conjunto de datos mediante muestreo aleatorio, y combina sus predicciones mediante promedios, lo cual mejora la precisión y reduce el riesgo de sobreajuste. Por su parte, las Redes Neuronales Artificiales imitan la estructura del cerebro humano a través de capas interconectadas de neuronas, que transforman las entradas mediante funciones de activación y se entrena mediante retropropagación para minimizar el error. En el caso del modelo GBDT, se construyen árboles de forma secuencial, donde cada árbol nuevo aprende a corregir los errores (residuos) del anterior, optimizando progresivamente el rendimiento del modelo. Para evaluar la precisión de estas técnicas, el autor emplea la métrica RMSE (Root Mean Square Error), la cual calcula la raíz cuadrada del promedio de los errores al cuadrado, penalizando con mayor intensidad aquellos errores de predicción más grandes y proporcionando una medida sólida de la exactitud del modelo en contextos de regresión.

## Capítulo 3. Metodología

Para explicar la metodología SEMMA se toma como referencia principal el trabajo de Firas (2023), quien la define de la siguiente manera: “The SEMMA process model consists of 5 main stages namely, sampling, exploring, modifying, modeling, and assessing.”

A partir de esta definición, se detallan a continuación las cinco fases que conforman el flujo SEMMA:

### 1. Sample (Muestreo).

En esta primera etapa se extrae un subconjunto representativo del conjunto de datos original, con el fin de acelerar las iteraciones posteriores sin perder fidelidad estadística. El muestreo puede realizarse de forma aleatoria o estratificada, de modo que se conserven las proporciones de la variable objetivo y se eviten sesgos en el entrenamiento de los modelos (Firas, 2023)

## **2. Explore (Exploración).**

Sobre la muestra seleccionada, se llevan a cabo análisis descriptivos y gráficos como histogramas, diagramas de dispersión, estadísticas de resumen además para identificar tendencias, valores atípicos y relaciones entre variables. Esta fase orienta la generación de hipótesis y la planificación de transformaciones en las etapas posteriores (Firas, 2023).

## **3. Modify (Modificación).**

Consiste en la limpieza de datos como imputación de valores faltantes, corrección de inconsistencias y la transformación de variables , además de creación de dummies, normalización, ingeniería de atributos para preparar el conjunto de datos de manera óptima de cara al modelado (Firas, 2023)

## **4. Model (Modelado).**

Con los datos ya preparados, se entrena n distintos algoritmos de minería de datos o aprendizaje automático como los árboles de decisión, redes neuronales, técnicas de boosting, etc.), aprovechando la capacidad de SAS Enterprise Miner para configurar, calibrar y comparar modelos en paralelo de forma ágil (Firas, 2023)

## **5. Assess (Evaluación).**

Finalmente, se evalúa el desempeño de cada modelo mediante métricas como AUC-ROC, precisión y recall, y se validan los resultados con técnicas de hold-out y validación cruzada. Esta etapa asegura que el modelo seleccionado cumpla con los objetivos de negocio antes de su despliegue (Firas, 2023)

# **Capítulo 4. Datos**

## **4.1 Fuente**

La base de datos utilizada en este trabajo de fin de máster fue elaborada a partir de la integración de diferentes fuentes oficiales. En una primera fase, se recopilaron y unificaron los archivos CSV correspondientes a Locales y Actividades publicados mensualmente por el Ayuntamiento de Madrid en su portal web, abarcando el periodo comprendido entre enero de 2020 y diciembre de 2024.

Posteriormente, esta base inicial se enriqueció mediante la incorporación de información procedente del Banco de Datos del propio Ayuntamiento de Madrid, el cual proporciona datos desagregados por local y año sobre la renta per cápita neta anual y la población. La combinación de ambas fuentes permitió construir una base de datos consolidada y adaptada a los objetivos de este estudio, sirviendo como soporte principal para el análisis desarrollado en la investigación.

## 4.2 Descripción de Variables

El fichero contiene variables organizadas en diferentes grupos:

- Variables geográficas y administrativas: incluyen identificadores y nombres de distrito, barrio, y sección censal donde se ubica cada local, permitiendo la localización espacial precisa dentro de la ciudad de Madrid.
- Coordenadas geográficas: se registran tanto en sistema UTM (X, Y) como en latitud y longitud (WGS84), para facilitar georreferenciación y análisis espacial.
- Tipo de acceso al local: variables que describen el tipo y características del acceso al local, como código y descripción del tipo de acceso (por ejemplo, puerta a la calle).
- Situación del local: estado actual del local, indicando si está abierto, cerrado, dado de baja, etc.
- Dirección del edificio y acceso: información detallada sobre la vía y número del edificio y del acceso específico al local.
- Datos comerciales: nombre comercial o rótulo del local.
- Clasificación económica (CNAE): códigos y descripciones que categorizan la actividad económica del local según la Clasificación Nacional de Actividades Económicas.
- Fecha de extracción del registro: año y mes de actualización del fichero.
- Variables derivadas para modelado: en particular, la variable objetivo-binaria actividad que indica si el local está activo (1) o no (0).
- Variables sociodemográficas: población residente en el barrio y renta neta media anual por persona en el distrito, que aportan contexto socioeconómico.

**Tabla 1: Lista y descripción de variables**

Categoría	Variable	Descripción
1. Identificadores Geográficos y Administrativos	id_local	Identificador único del local
	id_distrito_local	Código del distrito municipal
	desc_distrito_local	Nombre del distrito
	id_barrio_local	Código del barrio dentro del distrito
	desc_barrio_local	Nombre del barrio
	cod_barrio_local	Código oficial del barrio

<b>2. Coordenadas Geográficas</b>	latitud_local	Latitud geográfica del local (WGS84)
	longitud_local	Longitud geográfica del local (WGS84)
<b>3. Tipo de Acceso al Local</b>	id_tipo_acceso_local	Código del tipo de acceso (valor nulo permitido)
	desc_tipo_acceso_local	Descripción del tipo de acceso (e.g., "Puerta de Calle")
<b>4. Situación del Local</b>	desc_situacion_local	Estado del local (Abierto, Cerrado, Baja, etc.)
<b>5. Dirección del Edificio y Acceso</b>	clase_vial_acceso	Tipo de vía del acceso al local
	desc_vial_acceso	Nombre de la vía de acceso
	nom_acceso	Denominación del acceso
	num_acceso	Número del acceso
	cal_acceso	Calificador del número de acceso
<b>6. Datos Comerciales</b>	rotulo	Nombre comercial del local
<b>7. Clasificación Económica (CNAE)</b>	id_seccion	Código de sección CNAE
	desc_seccion	Descripción de la sección CNAE
	id_division	Código de división CNAE
	desc_division	Descripción de la división CNAE
	id_epigrafe	Código del epígrafe CNAE
	desc_epigrafe	Descripción del epígrafe CNAE
<b>8. Fecha del reporte</b>	Fecha_Reporte	Año y mes del informe de extracción del fichero
	Mes	Mes del reporte
	Año	Año del reporte
<b>9. Variable objetivo</b>	actividad	Variable binaria para modelado predictivo (1=Abierto, 0=No)
	desc_situacion_local	Descripción del estado del local
<b>10. Variables Sociodemográficas del Entorno</b>	Total_Poblacion	Población residente del barrio del local
	Renta_Media	Renta neta media anual por persona en el distrito

**Fuente:** Elaboración propia

#### 4.3 Construcción de la Variable Objetivo ("actividad")

La variable objetivo actividad se construye a partir de la variable desc\_situacion\_local que describe el estado del local. Para simplificar el análisis y enfocarlo en la actividad económica, se codifica como variable binaria:

- 1 (Activo): si el local está en estado "Abierto".
- 0 (Inactivo): si el local presenta cualquier otra situación (Cerrado, Baja, etc.).

Esta transformación permite desarrollar modelos predictivos binarios que identifiquen la probabilidad de que un local esté en funcionamiento activo.

#### 4.4 Limitaciones del Dataset

- Problemas de calidad en variables textuales: Algunas variables de tipo texto, como los nombres de barrios, distritos y rótulos comerciales, presentan inconsistencias y errores de escritura (faltas ortográficas, variantes en mayúsculas/minúsculas, espacios adicionales, etc.), lo que dificulta su unificación y análisis preciso.
- Inconsistencias en variables numéricas también se detectaron, donde el uso de separadores decimales y miles no era homogéneo (puntos y comas usados de manera incorrecta o mezclada), complicando la correcta interpretación y tratamiento de estos datos.
- Ausencia de variable fecha en tablas mensuales originales: Los archivos descargados para cada mes no contenían explícitamente la variable de fecha de reporte, por lo que fue necesario crearla e integrarla manualmente para poder identificar y organizar temporalmente los registros desde 2020 hasta 2024.
- Integración de múltiples tablas: La consolidación de los datos mensuales en una base única requirió un proceso de depuración, normalización y análisis, para asegurar la coherencia y homogeneidad, dada la variabilidad y errores presentes en los archivos originales.

### Capítulo 5. Preparación de los Datos

#### 5.1 Primer análisis exploratorio

El análisis exploratorio comenzó con la carga del archivo "Actividades Económicas de Madrid.csv". Este dataset se presentó información establecimientos económicos registrados en la ciudad de Madrid, incluyendo datos de identificación, ubicación geográfica, información estructural de los locales y edificios, tipos de acceso, estado operativo y clasificación de actividades económicas según la normativa vigente.

Uno de los aspectos más relevantes de la exploración inicial fue el análisis detallado de valores faltantes. Se evidenció una distribución desigual de los mismos, afectando en algunos casos a más del 90% de los registros en ciertas columnas. Por ejemplo, la variable "id\_local?" presentaba un nivel crítico de incompletitud, con cerca del 90% de los valores ausentes, mientras que id\_local, en cambio, aparecía más completa. Esta duplicidad apuntó a posibles errores en el proceso de consolidación de datos, lo cual se convirtió en un hallazgo clave de cara a las decisiones sobre depuración y consolidación de variables, por lo que se decidió fusionar las dos variables

En cuanto a la información geográfica, las coordenadas UTM (coordenada\_x\_local y coordenada\_y\_local) no contenían valores faltantes en términos técnicos, es decir, no presentaban Na's, pero se identificaron numerosos registros con coordenadas (0,0), que en la práctica representaban ubicaciones inválidas. Estas observaciones ponían de manifiesto la necesidad de realizar un análisis geoespacial adicional para validar y transformar estos datos a un formato geográfico estándar como latitud y longitud.

También se detectaron problemas recurrentes en la calidad de los datos categóricos. Varias variables presentaban codificación inconsistente, como el caso de los nombres de barrios, donde el mismo código podía estar vinculado a distintas descripciones ("Palos de Frontera" vs. "Palos de la Frontera"), o viceversa. Este tipo de errores terminológicos podía introducir sesgos importantes en los análisis posteriores si no se corregían mediante procesos de estandarización adecuados o el aumento de valores únicos.

En el ámbito de la clasificación económica, las variables id\_sección, id\_división e id\_epígrafe mostraban tanto valores nulos como registros con códigos atípicos (por ejemplo, "-1" o "PT"), que no formaban parte del sistema CNAE. Asimismo, se observaron discordancias entre los códigos y sus respectivas descripciones, lo cual evidenció deficiencias en la calidad del mapeo entre identificadores y etiquetas descriptivas.

Una parte significativa del análisis se centró en las variables relacionadas con edificios y accesos. Mientras las variables del edificio principal como son por ejemplo la clase\_vial\_edificio y desc\_vial\_edificio, nom\_edificio, presentaban un patrón uniforme de 39 valores faltantes, las variables de acceso mostraban niveles mucho más altos de faltantes. La variable id\_vial\_acceso, por ejemplo, contaba con más de cinco millones de registros vacíos, lo que planteaba interrogantes sobre su utilidad real para los análisis posteriores. A pesar de estas carencias, algunas variables descriptivas de acceso no presentaban valores nulos, lo que sugiere la existencia de problemas de integración entre distintas bases de datos administrativas.

El análisis de la variable temporal Fecha\_Reporte, representada en formato YYYYMM, permitió observar que no existían valores faltantes en esa columna. Sin embargo, otras variables temporales como fx\_carga, fx\_datos\_ini y fx\_datos\_fin presentaban niveles muy altos de ausencia de datos. Este patrón indicaba posibles deficiencias en la captura de metadatos asociados al proceso de carga de registros o la falta de actualización de ciertas variables.

En relación con los estados operacionales de los establecimientos, se identificaron múltiples categorías como "abierto", "cerrado", "baja", "baja reunificación", "en obras" o "uso vivienda". Esta diversidad ofrecía información del estado de los locales a nivel específico, aunque también requería de un tratamiento posterior de recodificación para facilitar su análisis desde una perspectiva más predictiva.

Durante esta fase inicial se detectaron múltiples anomalías vinculadas a la codificación de caracteres. Entre los errores más comunes se encontraron sustituciones incorrectas como "0" por "o", problemas con la representación de la letra "ñ", fallos en el uso de tildes, así como la presencia de espacios en blanco irregulares y un uso poco sistemático de mayúsculas y minúsculas. Estos problemas afectaban particularmente a las variables

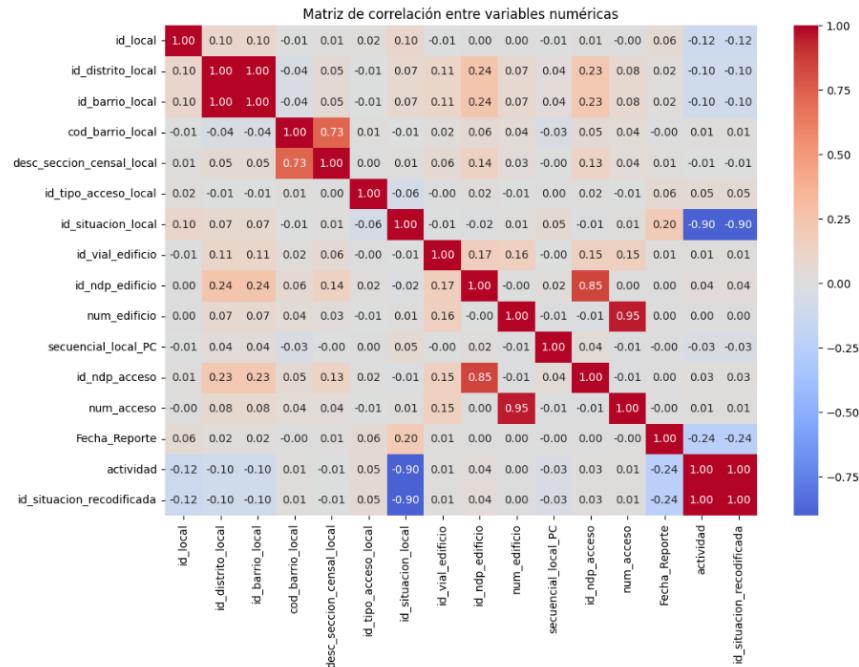
tipo categórico y condicionaban su agrupación y análisis correcto si no se aplicaba un proceso de normalización adecuado.

Finalmente, se observaron redundancias informativas entre variables conceptualmente relacionadas, como las de edificios y accesos, y entre distintas formas de representación de la misma información. Esta redundancia no solo añadía complejidad al análisis, sino que también obligaba a definir criterios claros sobre qué variables conservar, cuáles consolidar y cuáles eliminar en la fase de depuración posterior. Así, el análisis exploratorio permitió establecer un diagnóstico completo sobre el estado del dataset original, sentando las bases para las decisiones de limpieza, transformación y preparación necesarias para abordar análisis más avanzados de carácter descriptivo y predictivo.

Todo este proceso generó múltiples valores distintos en el análisis, destacando la presencia de valores únicos causados por errores de escritura en diversas variables de tipo texto. Además, se identificaron variables con un alto porcentaje de valores nulos. Este análisis permitió evaluar cuáles de estas variables podían conservarse para conformar un dataset adecuado y sin complicaciones. Todo el análisis fue realizado con Python, lo cual facilitó la ejecución de acciones necesarias para la preparación de los datos previos a la aplicación de técnicas de machine learning.

Por último, generó una matriz de correlación entre variables numéricas para identificar relaciones lineales que pudieran influir en la calidad del dataset. En ella, se observaron fuertes correlaciones positivas entre variables como id\_ndp\_edificio, num\_edificio, secuencial\_local\_PC y num\_acceso, lo que sugiere redundancia de información y la posibilidad de reducir la dimensionalidad eliminando algunas de estas variables. Además, se detectaron correlaciones perfectas entre id\_local, id\_distrito\_local e id\_barrio\_local, lo cual indica que estas variables probablemente están jerárquicamente relacionadas. Por otro lado, la variable actividad mostró una correlación negativa muy fuerte (-0.90) con id\_situacion\_local y su versión recodificada, lo que evidencia una relación inversa significativa que podría ser útil para modelos de clasificación o predicción. Este análisis permitió identificar variables clave, detectar colinealidad y orientar decisiones para la limpieza y selección de variables, asegurando así un dataset más limpio, eficiente y adecuado para aplicar técnicas de machine learning.

**Ilustración 15:** Dos enfoques del aprendizaje automático para la toma de decisiones



Fuente: Elaboración propia

## 5.2 Segundo análisis exploratorio después de la limpieza

En el presente análisis exploratorio después de la limpieza se empleó el software SAS, aplicando los códigos aprendidos a lo largo del máster. La base de datos original, en formato CSV, fue transformada exitosamente en un conjunto de datos SAS, lo que permitió trabajar de manera eficiente con las variables disponibles.

A partir de este mismo programa se generaron automáticamente todas las tablas y gráficos incluidos en el análisis, utilizando los procedimientos y rutinas implementadas en los códigos SAS. Entre las tareas realizadas destacan el cálculo de rentas medias por año y distrito, la ordenación de dichas rentas, la elaboración de gráficos de evolución temporal, la distribución de la variable de actividad, el análisis de frecuencias de palabras y la creación de un gráfico que permite identificar los términos o conceptos más repetidos en la variable sección de local, la cual hace referencia al rubro al que pertenece cada establecimiento.

Se ve en la siguiente figura el numero de registros que tiene nuestro dataset que son de 8318562 observaciones

**Ilustración 16:** Total de registros en la base de datos



Fuente: Elaboración propia

La tabla presenta un resumen estadístico de la población total y la renta media per cápita anual en los barrios de Madrid entre 2020 y 2024. La población promedio por barrio es de 33.019 habitantes, con valores que oscilan entre 1.444 y 75.829, mientras que la renta media anual se sitúa en 19.082 euros, con un rango de 10.239 a 30.506 euro

**Ilustración 17:** Valores de población y de renta media per cápita anual (promedio, mínimo y máximo) por barrio en Madrid (2020-2024)

The SAS System						
The MEANS Procedure						
Variable	N	N Miss	Minimum	Maximum	Mean	Std Dev
Total_Poblacion	8318562	0	1444.00	75829.00	33019.36	14767.19
Renta_Media	8318562	0	10239.00	30506.00	19082.64	5728.48

Fuente: Elaboración propia

La Figura muestra la distribución de registros por año y mes entre 2020 y 2024, con un total de 8.318.562 datos. Se aprecia un aumento progresivo desde 1.444.693 registros en 2020 hasta 2.007.958 en 2024, con valores mensuales estables salvo la ausencia de datos en abril de 2022, el cual no se pudo obtener dichos datos del ayuntamiento

**Ilustración 18:** Número de registros del dataset por año y mes

The SAS System													
The FREQ Procedure													
Frequency	Table of Anio by Mes												
	Anio	1	2	3	4	5	6	7	8	9	10	11	12
2020	119744	119834	119910	119975	120136	120343	120470	120530	120674	120878	121084	121115	1444693
2021	121280	121419	121619	121778	121911	122036	122213	122294	122405	123395	125737	125881	1471963
2022	126005	126107	126815	0	127103	127262	127334	127445	0	166382	166439	166587	1387479
2023	166651	166722	166752	166923	167046	167202	167333	167439	167544	167618	167625	167664	2006519
2024	167741	167751	167759	167831	167833	168017	168723	168830	168936	164574	164923	164990	2007908
Total	701421	701833	702855	576507	704029	704854	706073	706538	579559	742847	745808	746238	8318562

Fuente: Elaboración propia

La Figura 18 presenta las 29 variables del dataset, que incluyen identificadores, descripciones textuales, datos geográficos (como distrito, barrio, latitud y longitud), información socioeconómica (Total\_Poblacion, Renta\_Media) y variables temporales (Fecha\_Reporte, Mes, Anio). Esta estructura combina datos espaciales, económicos y temporales, permitiendo un análisis completo del conjunto.

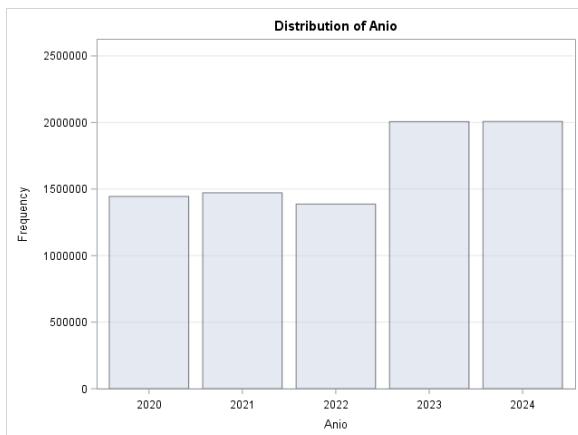
**Ilustración 19:** Listado de variables del dataset

Variables in Creation Order					
#	Variable	Type	Len	Format	Format
1	id_local	Num	8	BEST12	BEST32
2	id_distrito_local	Num	8	BEST12	BEST32
3	des_c_distrito_local	Char	19	\$19.	\$19.
4	id_barrio_local	Num	8	BEST12	BEST32
5	des_c_barrio_local	Char	20	\$20.	\$20.
6	cod_barrio_local	Num	8	BEST12	BEST32
7	id_tipo_acceso_local	Num	8	BEST12	BEST32
8	des_c_tipo_acceso_local	Char	12	\$12.	\$12
9	des_c_situacion_local	Char	8	\$8.	\$8.
10	clase_vial_acceso	Char	10	\$10.	\$10.
11	des_c_vial_acceso	Char	40	\$40.	\$40.
12	nom_acceso	Char	3	\$3.	\$3.
13	num_acceso	Num	8	BEST12	BEST32
14	cal_acceso	Char	2	\$2.	\$2.
15	rotulo	Char	72	\$72.	\$72.
16	id_seccion	Char	1	\$1.	\$1.
17	des_c_seccion	Char	88	\$88.	\$88.
18	id_division	Char	2	\$2.	\$2.
19	des_c_division	Char	123	\$123.	\$123.
20	id_epigrafe	Char	8	\$8.	\$8.
21	des_c_epigrafe	Char	155	\$155.	\$155.
22	Fecha_Reporte	Num	8	BEST12	BEST32
23	actividad	Num	8	BEST12	BEST32
24	latitud_local	Num	8	BEST12	BEST32
25	longitud_local	Num	8	BEST12	BEST32
26	Total_Poblacion	Num	8	BEST12	BEST32
27	Renta_Media	Num	8	BEST12	BEST32
28	Mes	Num	8	BEST12	BEST32
29	Anio	Num	8	BEST12	BEST32

**Fuente:** Elaboración propia

En la siguiente figura se muestra un gráfico de barras que presenta la cantidad de observaciones por año del conjunto de datos, donde se aprecia que en 2023 y 2024 se registran más observaciones.

**Ilustración 20:** Distribución de datos por año del dataset



**Fuente:** Elaboración propia

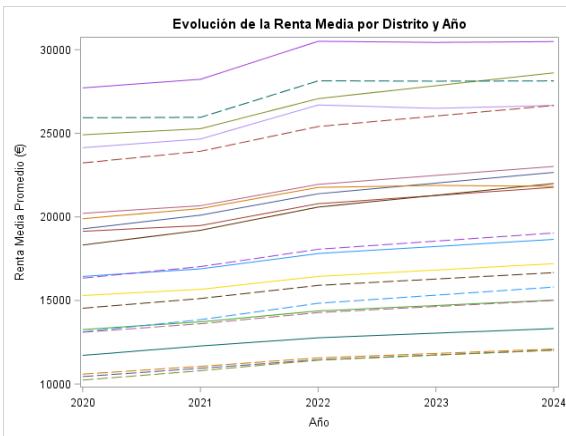
**Ilustración 21: Matriz de correlación de variables numéricas**

Pearson Correlation Coefficients, N = 8318562 Prob >  r  under H0: Rho=0														
	Fecha_Reporte	Mes	Renta_Media	Total_Poblacion	Anio	actividad	cod_barrio_local	id_barrio_local	id_distrito_local	id_local	id_tipo_acceso_local	latitud_local	longitud_local	num_acceso
Fecha_Reporte	1.00000	0.01810	0.13085	0.07981	0.99970	-0.23678	-0.00125	0.02429	0.02430	0.05672	0.06346	-0.00584	0.00493	-0.00091
		<.0001	<.0001	<.0001	<.0001	<.0001	0.0003	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	0.0087
Mes	0.01810	1.00000	0.00971	-0.00879	-0.00636	-0.00497	0.00113	-0.00763	-0.00764	0.06877	0.03156	0.00634	0.00067	-0.00038
			<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	0.2553
Renta_Media	0.13085	0.00971	1.00000	-0.30983	0.13063	0.05689	-0.01944	-0.56414	-0.56401	-0.09390	0.03860	0.60521	0.01852	-0.01064
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
Total_Poblacion	0.07891	-0.00879	-0.30983	1.00000	-0.10404	-0.05817	0.22941	0.22961	0.06407	-0.03323	-0.12603	-0.08792	0.03852	
		<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
Anio	0.99970	-0.00636	0.13063	0.07713	1.00000	-0.23667	-0.00127	0.02448	0.02449	0.05656	0.06270	-0.00597	0.00492	-0.00090
			<.0001	<.0001		<.0001	<.0001	0.0002	<.0001	<.0001	<.0001	<.0001	<.0001	0.0094
actividad	-0.23678	-0.00497	0.05689	-0.10404	-0.23667	1.00000	0.01197	-0.09741	-0.09743	-0.12467	0.05319	0.04032	0.00094	0.00897
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	0.0070
cod_barrio_local	-0.00125	0.00113	-0.01944	-0.06817	-0.00127	0.01197	1.00000	-0.03703	-0.04030	-0.00982	0.00889	0.20642	-0.11680	0.04178
			0.0003	0.0011		<.0001	0.0002	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
id_barrio_local	0.02429	-0.00763	-0.56414	0.22941	-0.05741	-0.03703	1.00000	0.99999	0.09758	-0.00526	-0.17956	0.49574	0.07616	
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
id_distrito_local	0.02430	-0.00764	-0.56401	0.22981	0.02449	-0.05743	-0.04030	0.99999	1.00000	0.09780	-0.00528	-0.18024	0.49808	0.07802
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
id_local	0.05672	0.00677	-0.09090	0.06407	0.05589	-0.12467	-0.00982	0.09788	0.09780	1.00000	0.02441	-0.01531	0.02577	-0.00010
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	0.7785
id_tipo_acceso_local	0.06348	0.03156	0.05880	-0.03323	0.06270	0.05319	0.06865	-0.00268	-0.00528	0.2441	1.00000	0.02885	0.00941	-0.00819
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
latitud_local	-0.00584	0.00634	0.06521	-0.12603	-0.00597	0.04032	0.20642	-0.17958	-0.18024	-0.01531	0.02585	1.00000	0.18190	0.06621
			<.0001	<.0001		<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
longitud_local	0.00493	0.00067	0.01852	-0.08792	0.02492	0.00094	-0.11680	0.49574	0.49606	0.02577	0.00941	0.18190	1.00000	0.04345
			<.0001	0.0521		<.0001	<.0001	0.0070	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001
num_acceso	-0.00091	-0.00039	-0.01064	0.03852	-0.00090	0.00897	0.04178	0.07816	0.07802	-0.00010	-0.00819	0.06521	0.04346	1.00000
			0.0087	0.2553		<.0001	0.0094	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001	<.0001

Fuente: Elaboración propia

En el siguiente gráfico se observa cómo la renta media per cápita anual por distrito ha ido en aumento entre 2020 y 2024

**Ilustración 22: Evolución de la Renta Media por Distrito y Año**



Fuente: Elaboración propia

En la siguiente tabla se presenta la renta media anual promedio, dividida por distrito y año. Se observa que los residentes de Chamberí han registrado la mayor renta media durante los últimos años, alcanzando su nivel más alto en 2024. Por otro lado, en 2020, los distritos de Usera y Vallecas mostraron las rentas medias más bajas dentro del mismo período.

**Tabla 2: Renta media promedio per capita por distrito y año. Top 30**

Año	Distrito	Renta Media Promedio (€)
2022	CHAMARTIN	30506
2024	CHAMARTIN	30489
2023	CHAMARTIN	30437

2024	CHAMBERI	28621
2021	CHAMARTIN	28233
2022	SALAMANCA	28140
2024	SALAMANCA	28135
2023	SALAMANCA	28119
2023	CHAMBERI	27848
2020	CHAMARTIN	27719
2022	CHAMBERI	27076
2022	MONCLOA-ARAVACA	26694
2024	MONCLOA-ARAVACA	26669
2024	RETIRO	26665
2023	MONCLOA-ARAVACA	26497
2023	RETIRO	26036
2021	SALAMANCA	25956
2020	SALAMANCA	25932
2022	RETIRO	25407
2021	CHAMBERI	25275
2020	CHAMBERI	24913
2021	MONCLOA-ARAVACA	24659
2020	MONCLOA-ARAVACA	24138
2021	RETIRO	23925
2020	RETIRO	23227
2024	FUENCARRAL-EL PARDO	23017
2024	ARGANZUELA	22658
2023	FUENCARRAL-EL PARDO	22482
2023	ARGANZUELA	22021
2024	CENTRO	21997

**Fuente:** Elaboración propia

En cuanto a los barrios, Castilla (ubicado en el distrito de Chamartín), Ciudad Jardín (en Hortaleza) y El Viso (en el distrito de Chamartín) registraron la mayor renta media promedio en 2022. Por otro lado, Zofío y San Fermín, ambos situados en el distrito de Usera, fueron los que obtuvieron las rentas medias más bajas en 2020.

**Tabla 3: Renta Media por Barrio y Año (Ordenada Descendentemente). Top 20**

Año	Barrio	Renta Media Promedio (€)
2022	castilla	30506
2022	ciudad jardin	30506
2022	el viso	30506
2022	hispanoamerica	30506
2022	nueva espana	30506
2022	prosperidad	30506
2024	castilla	30489
2024	ciudad jardin	30489
2024	el viso	30489
2024	hispanoamerica	30489
2024	nueva espana	30489
2024	prosperidad	30489
2023	castilla	30437
2023	ciudad jardin	30437
2023	el viso	30437
2023	hispanoamerica	30437
2023	nueva espana	30437
2023	prosperidad	30437
2024	almagro	28621
2024	arapiles	28621

**Fuente:** Elaboración propia

Se realizan análisis sobre la variable objetivo binaria “actividad”, donde el valor 1 indica que el local está activo en ese momento, y el valor 0 que está inactivo. En primer lugar, se genera un análisis para todo el conjunto de datos, y luego se realizan análisis específicos para los años 2023 y 2024.

**Ilustración 23: Distribución de la variable objetivo**

**Distribución general de la variable binaria Actividad**

The FREQ Procedure

actividad	Frequency	Percent
0	1605040	19.29
1	6713522	80.71

**Fuente:** Elaboración propia

**Ilustración 24:** Distribución de la variable objetivo de los años 2023 y 2024

**Distribución de la variable binaria Actividad por año (con porcentaje)**

The FREQ Procedure

Anio=2023

actividad	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	627023	31.25	627023	31.25
1	1379496	68.75	2006619	100.00

**Distribución de la variable binaria Actividad por año (con porcentaje)**

The FREQ Procedure

Anio=2024

actividad	Frequency	Percent	Cumulative Frequency	Cumulative Percent
0	533977	26.59	533977	26.59
1	1473931	73.41	2007908	100.00

**Fuente:** Elaboración propia

En la siguiente tabla se observa que, en 2023, Puente Vallecas fue el distrito con menor número de locales activos, seguido por Latina en ese mismo año. Por otro lado, en 2024, Chamberí y Salamanca registraron la mayor cantidad de locales activos.

**Tabla 4:** Distritos con mayor % de locales inactivos (años 2023 y 2024)

Anio	Distrito	Locales Inactivos	Total, de Locales	% Inactivos
2023	PUENTE DE VALLECAS	65420	140211	46.66
2023	LATINA	53974	128674	41.95
2023	CARABANCHEL	68161	166655	40.90
2023	CIUDAD LINEAL	58591	147742	39.66
2023	HORTALEZA	34208	88024	38.86
2023	VILLAVERDE	36313	95943	37.85
2023	USERA	28766	76815	37.45
2023	TETUAN	47819	128857	37.11
2023	VILLA DE VALLECAS	18868	57276	32.94
2023	FUENCARRAL-EL PARDO	33944	106356	31.92
2023	VICALVARO	9242	29419	31.42
2023	SAN BLAS-CANILLEJAS	23419	79504	29.46
2023	MORATALAZ	9730	35154	27.68
2023	BARAJAS	7631	28161	27.10
2023	MONCLOA-ARAVACA	14908	65895	22.62
2023	CHAMARTIN	20452	94784	21.58
2023	ARGANZUELA	17174	82149	20.91
2023	CENTRO	36313	176439	20.58
2023	RETIRO	9293	56707	16.39
2023	SALAMANCA	17535	118364	14.81
2023	CHAMBERI	15262	103390	14.76
2024	PUENTE DE VALLECAS	53275	128703	41.39
2024	LATINA	44374	120296	36.89
2024	CARABANCHEL	55844	157005	35.57
2024	CIUDAD LINEAL	48540	141049	34.41
2024	HORTALEZA	29248	85881	34.06
2024	VILLAVERDE	30078	90031	33.41
2024	USERA	23880	72997	32.71
2024	TETUAN	41031	130015	31.56
2024	VILLA DE VALLECAS	15862	54880	28.90

2024	FUENCARRAL-EL PARDO	29408	104306	28.19
2024	VICALVARO	7900	28535	27.69
2024	SAN BLAS-CANILLEJAS	19868	79309	25.05
2024	MORATALAZ	8526	34232	24.91
2024	BARAJAS	6863	29072	23.61
2024	MONCLOA-ARAVACA	12734	67342	18.91
2024	ARGANZUELA	15564	83758	18.58
2024	CHAMARTIN	18233	101373	17.99
2024	CENTRO	32422	193255	16.78
2024	RETIRO	8736	60462	14.45
2024	SALAMANCA	17586	133820	13.14
2024	CHAMBERI	14005	111587	12.55

**Fuente:** Elaboración propia

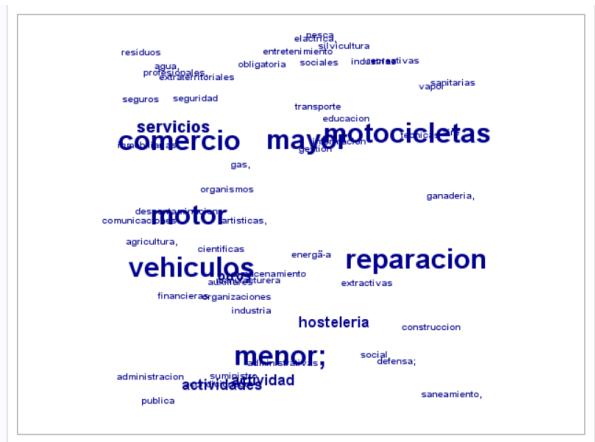
**Tabla 5: Barrios con mayor % de locales inactivos (años 2023 y 2024). Top 20**

Posición	Año	Barrio	Locales Inactivos	Total, Locales	% Inactivos
1	2023	Los Carmenes	5,399	9,127	59.15%
2	2023	Entrevías	10,019	18,117	55.30%
3	2023	Valdeacederas	12,317	22,928	53.72%
4	2024	Los Carmenes	4,344	8,142	53.35%
5	2023	Amposta	2,813	5,462	51.50%
6	2023	San Isidro	13,043	25,423	51.30%
7	2023	Puerta del Ángel	18,418	36,474	50.50%
8	2024	Entrevías	8,156	16,108	50.63%
9	2023	Zofío	4,295	8,638	49.72%
10	2023	Pinar del Rey	17,039	34,578	49.28%
11	2023	Campamento	4,894	9,974	49.07%
12	2024	Valdeacederas	10,080	20,877	48.28%
13	2023	Palomeras Sureste	9,909	20,674	47.93%
14	2023	San Diego	17,401	36,613	47.53%
15	2023	Almenara	8,482	18,009	47.10%
16	2023	Ventas	16,349	35,137	46.53%
17	2024	Amposta	2,349	5,060	46.42%
18	2023	Orcasur	2,307	5,099	45.24%
19	2024	Zofío	3,544	7,847	45.16%
20	2024	Puerta del Ángel	15,020	33,377	45.00%
21	2024	San Isidro	10,581	23,524	44.98%
22	2024	Pinar del Rey	14,127	31,775	44.46%
23	2023	Comillas	7,569	17,046	44.40%
24	2024	Campamento	4,065	9,189	44.24%
25	2023	Lucero	7,871	17,879	44.02%
26	2023	Numancia	12,959	29,791	43.50%
27	2023	Palomeras Bajas	7,271	16,768	43.36%
28	2023	Puerta Bonita	8,390	19,362	43.33%
29	2023	Berruguete	8,277	19,155	43.21%
30	2023	Portazgo	7,861	18,248	43.08%

**Fuente:** Elaboración propia

Las palabras que se observan con más claridad son "comercio", "vehículos", "reparación", "motor" y "motocicletas" indican la sección de los locales comerciales más comunes. El tamaño de cada palabra refleja su frecuencia relativa en el conjunto de datos. Esto sugiere que sectores relacionados con la venta y reparación de vehículos, así como el comercio al por menor, son predominantes en la muestra. Otras palabras como "hostelería" y "servicios" también emergen, lo que indica la presencia de actividades complementarias en el ámbito de la restauración y servicios.

**Ilustración 25: Análisis de frecuencia de palabras: Secciones Comerciales**



**Fuente:** Elaboración propia

## 5.3 Limpieza de Datos

El proceso de limpieza del dataset original sobre actividades económicas en la ciudad de Madrid supuso un desafío considerable debido a la complejidad estructural de los datos. Esta sección detalla las metodologías aplicadas para gestionar valores faltantes, recodificar variables, depurar contenido textual y garantizar la coherencia e integridad del conjunto de datos final.

Durante el análisis exploratorio inicial, se identificó una cantidad significativa de valores nulos. Se procedió a evaluar su distribución y relevancia para el estudio, descartando aquellas variables con más del 65% de valores ausentes y sin utilidad analítica directa. Asimismo, se eliminaron registros que presentaban carencias generalizadas en múltiples columnas, imposibilitando su imputación, especialmente en variables críticas como las de geolocalización.

Además, como parte del proceso de integración de fuentes, se detectaron variables redundantes con nombres inconsistentes, producto de errores en la unión de distintos datasets. Por ejemplo, las variables id\_local y id\_local? correspondían al mismo campo, por lo que se procedió a su fusión de estos.

### **5.3.1 Imputación de Datos Faltantes**

Una parte sustancial de la limpieza consistió en la imputación de valores faltantes en columnas clave. Para ello, se aplicaron estrategias basadas en duplicados de alta confianza y lógica relacional entre variables, implementadas con herramientas de procesamiento en Python.

En particular, se abordó la imputación de campos relacionados con la identificación territorial como son las variables `desc_distrito_local`, `id_distrito_local`, `desc_barrio_local` y `id_barrio_local`. Se utilizó la variable `id_local` como clave principal, buscando registros duplicados con esta misma identificación que tuvieran información completa, y relacionando estos a los valores incompletos a medida que se inputaba

cada observación. Se ejecuta códigos para hallar coincidencias entre la información encontrada en el dataset.

De igual forma, se imputaron campos relacionados con las características estructurales del local, como la calle de acceso, utilizando filtros condicionales sobre el id\_local y técnicas de agrupamiento. Además, se validaron y completaron las relaciones entre identificadores y descripciones categóricas, ya que, tras imputar los valores de barrios y distritos de ciertos locales, fue posible determinar la calle de acceso correspondiente. Se emplearon las bibliotecas pandas y NumPy para el manejo eficiente de valores nulos e infinitos, facilitando la preparación del dataset antes de su uso en tareas de modelado y análisis predictivo.

### *5.3.2 Tratamiento de Coordenadas Geográficas*

El conjunto de datos original incluía las variables coordenadas\_x\_local y coordenada\_y\_local, correspondientes a coordenadas en el sistema UTM, específicamente que abarca la ciudad de Madrid. Este sistema es comúnmente utilizado en cartografía técnica por su precisión métrica, ya que permite trabajar con distancias lineales en metros. Sin embargo, para determinadas aplicaciones analíticas, visualizaciones geográficas o integraciones con otras fuentes de datos espaciales, resulta más conveniente emplear coordenadas geográficas en forma de latitud y longitud (sistema WGS84), las cuales son ampliamente reconocidas en herramientas como Google Maps, por ejemplo.

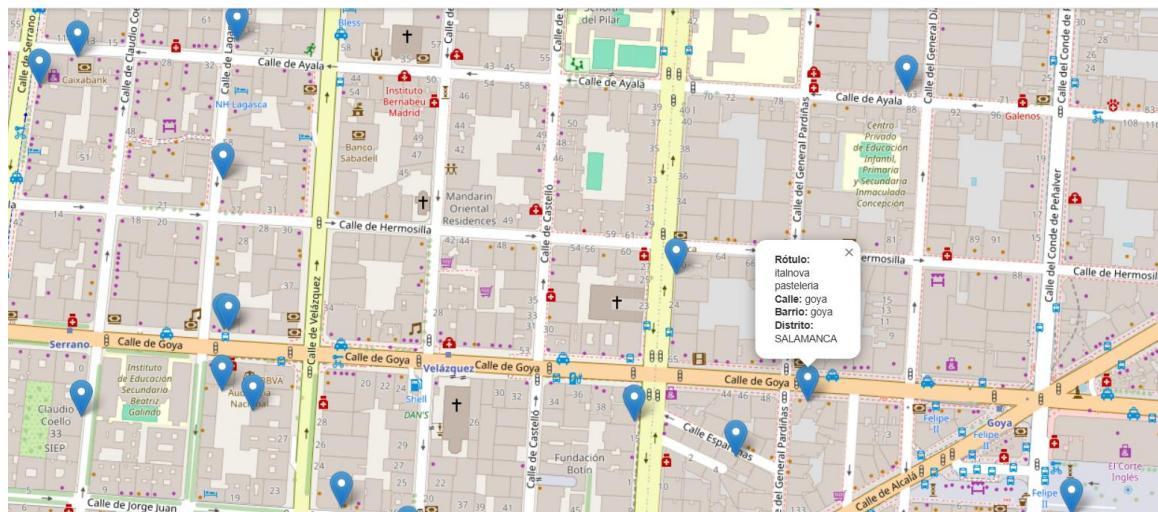
Por esta razón, se procedió a la conversión de las coordenadas UTM a latitud (latitud\_local) y longitud (longitud\_local). Este proceso se realizó utilizando la librería pyproj, que permite transformar coordenadas entre distintos sistemas de referencia geodésicos. En concreto, se definió la proyección UTM con los parámetros apropiados y se empleó un objeto Transformer para realizar la transformación fila a fila, verificando previamente que los valores no fueran nulos ni igual a cero.

Adicionalmente, se detectaron múltiples registros con valores faltantes o erróneos en las variables de coordenadas, lo cual dificultaba su uso posterior en análisis espaciales o representaciones cartográficas. Para solucionar esto, se implementaron estrategias de imputación progresiva. En primer lugar, se construyó una clave compuesta a partir de atributos del local (como el identificador, calle, número o barrio) y se utilizaron registros con coordenadas válidas como referencia para imputar aquellos incompletos. Posteriormente, se flexibilizó el análisis reduciendo el número de variables clave cuando no existían coincidencias exactas. Este enfoque permitió hallar las ubicaciones exactas de los locales, mejorando así la calidad y cobertura de los datos geográficos.

La transformación a coordenadas de latitud y longitud no solo facilitó la imputación de datos mediante la comparación entre registros, sino que también fue fundamental para la representación visual en mapas interactivos, utilizando herramientas como Folium. Esta visualización permitió validar la imputación, al comprobar en un mapa de Madrid la correcta ubicación de los locales. Para ello, se integraron en un script de Python los puntos correspondientes a cada local, incluyendo leyendas con el barrio, distrito, rótulo

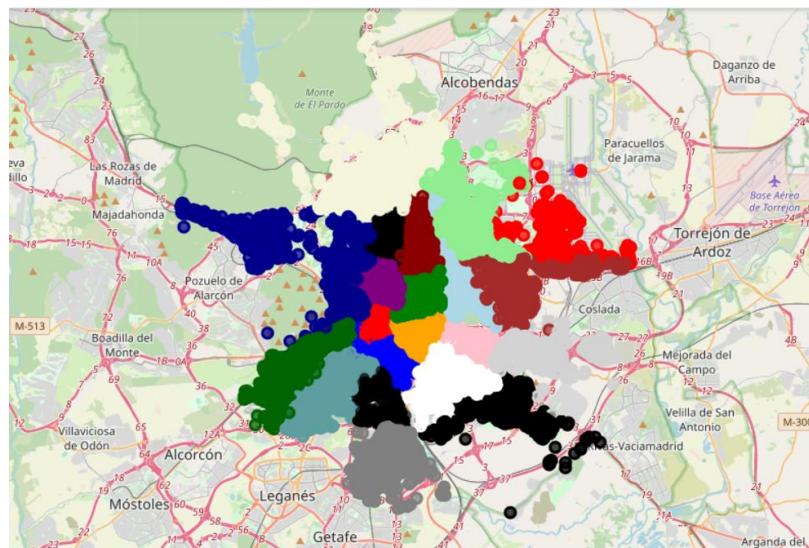
y calle asociada, lo que permitió contrastar visualmente los datos con el mapa oficial y verificar su precisión, como se muestra en la siguiente figura.

**Ilustración 26:** Mapa de Madrid verificando correcta imputación de latitud y longitud



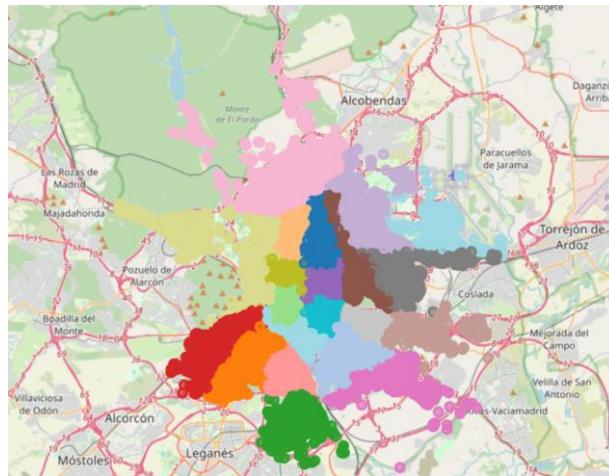
Fuente: Elaboración propia

**Ilustración 27:** Mapa de Madrid verificando correcta imputación de latitud y longitud año 2023.



Fuente: Elaboración propia

**Ilustración 28:** Mapa de Madrid verificando correcta imputación de latitud y longitud año 2024.



Fuente: Elaboración propia

### **5.3.3 Recodificación de Variables**

La recodificación de variables categóricas fue una tarea esencial para consolidar criterios homogéneos en el dataset y preparar las variables para análisis posteriores. Un ejemplo representativo fue la transformación de la variable `desc_situacion_local`, que presentaba hasta siete denominaciones diferentes para describir el estado de un local: "abierto", "cerrado", "baja", "en obras", "uso vivienda" y "baja R"

Dado que el objetivo era modelar la apertura o cierre de un local como variable binaria, se unificaron todas las categorías no activas bajo una única denominación que sería "Inactivo", manteniendo solo "Abierto" como valor positivo. De esta transformación se derivó una nueva variable llamada actividad, codificada como 1 si el local estaba abierto y 0 en cualquier otro caso.

#### *5.3.4 Revisión y Normalización del Texto*

Se realizó un proceso de revisión, limpieza y normalización de todas las variables categóricas presentes en el conjunto de datos, con especial atención a aquellas columnas con contenido textual como por ejemplo desc\_barrio\_local, desc\_distrito\_local, rotulo, desc\_epigrafe, desc\_division. La finalidad de este proceso fue garantizar la coherencia interna del dataset, evitar duplicidades que pudieran inducir a errores de interpretación, y asegurar la correcta integración de fuentes externas como los datos de renta y población por barrio y distrito.

Uno de los principales desafíos detectados fue la existencia de registros repetidos o mal interpretados debido a variaciones en el uso de mayúsculas, tildes, caracteres especiales y espacios en blanco al inicio o final de las cadenas. Para resolver estos problemas se aplicaron técnicas de limpieza sistemática que incluyeron la conversión de todas las cadenas a minúsculas, la eliminación de espacios innecesarios y la normalización de caracteres mediante la librería unidecode. Este tratamiento permitió, por ejemplo, unificar registros como “Chamberí”, “CHAMBERI”, “Chamberi (sin tilde)” o “chámbperi” bajo una única representación coherente.

Además de la normalización básica, se implementó un control más riguroso sobre las descripciones utilizando expresiones regulares y reglas personalizadas de reemplazo. Se identificaron y corrigieron errores frecuentes de escritura en diversas variables, como por ejemplo “vehaaculos” en lugar de “vehículos”, “hosteleraa” por “hostelería” o “educacion” en vez de “educación”. Estos errores surgían tanto de fallos en la digitalización como de problemas de codificación al momento de importar los archivos. Para corregirlos, se construyó un diccionario de reemplazo con las versiones correctas de los términos, que se aplicó a cada variable relevante mediante funciones de transformación en pandas.

Con el objetivo de detectar de manera más automática posibles errores ortográficos, se incorporó también la herramienta `pyspellchecker`, una biblioteca que permite identificar palabras inexistentes en el idioma español. Esta técnica fue especialmente útil para columnas como `rotulo`, `desc_epigrafe`, `desc_division` y `desc_seccion`, donde la cantidad de valores únicos era muy elevada y dificultaba una revisión manual completa. Una vez detectadas las palabras sospechosas, se realizaron revisiones caso por caso para determinar si debían corregirse, eliminarse o recodificarse bajo una categoría común como “sin actividad”.

Otro aspecto crítico fue el aseguramiento de la correspondencia unívoca entre variables categóricas y sus identificadores numéricos asociados. Por ejemplo, se comprobó que cada valor de `desc_barrio_local` correspondiera a un único código en `cod_barrio_local`, y viceversa. Ante la detección de casos en los que un mismo código aparecía vinculado a distintas descripciones, se optó por conservar aquella descripción que era más frecuente o la que coincidía con fuentes externas oficiales. En paralelo, se consolidaron todos los registros que representaban la misma entidad bajo una forma común, eliminando así las ambigüedades internas del dataset.

En lo que respecta a la integración con otras fuentes, como los datos socioeconómicos por barrio, se aplicó una estrategia avanzada de normalización que permitiera comparar cadenas de texto de diferentes archivos sin errores. Para ello, se desarrollaron funciones que, además de eliminar tildes y artículos definidos (“el”, “la”, “los”, “las”), reestructuraban las cadenas en un formato estandarizado, permitiendo así emparejar con precisión los barrios entre distintos datasets, incluso cuando existían pequeñas diferencias en la escritura o presentación.

Este proceso de limpieza textual no solo favoreció la calidad del análisis estadístico y la visualización posterior, sino que también fue clave para evitar que las inconsistencias en los datos afectaran la precisión de modelos predictivos o tareas de imputación posteriores. Las técnicas aplicadas se implementaron principalmente con las bibliotecas pandas, unidecode, unicodedata, re y `pyspellchecker`, seleccionadas por su eficacia y versatilidad en la manipulación de texto dentro del entorno de análisis de datos en Python.

## 5.4 Preprocesamiento de los datos

### 5.4.1. Identificación y selección de variables categóricas

En una primera etapa del preprocesamiento, se definió un subconjunto de variables categóricas consideradas de interés para el análisis, entre las que se incluían desc\_distrito\_local, desc\_barrio\_local, desc\_tipo\_acceso\_local y desc\_sección. Estas variables fueron seleccionadas en base a criterios de relevancia contextual y dan mejor descripción al local. También se aplicó un filtro adicional sobre este subconjunto categórico con el objetivo de identificar variables con baja cardinalidad, entendida como aquellas que presentan un número reducido de categorías. Esta decisión metodológica responde a una práctica común en el tratamiento de datos categóricos que es limitar la transformación a variables que, al ser codificadas mediante técnicas como one hot encoding, no generen muchas columnas por ejemplo la variable referente a epígrafe que tiene muchos valores.

#### *5.4.2 Transformación de variables categóricas: codificación one-hot*

Una vez identificadas las variables categóricas con cardinalidad controlada, se procedió a su transformación mediante la técnica de codificación one-hot. Esta técnica convierte cada categoría en una variable binaria independiente, también conocida como variable dummy. Con el fin de evitar redundancias y posibles problemas de colinealidad derivados de la creación de múltiples variables dummies altamente correlacionadas, se optó por eliminar una de las categorías posibles en cada variable transformada. Esto se logró mediante el parámetro drop\_first=True, el cual excluye automáticamente la primera categoría como referencia, manteniendo la información completa del conjunto sin introducir dependencia lineal entre las nuevas variables generadas.

#### *5.4.3 Observación de variables no relevantes*

Completada la transformación de variables categóricas, se llevó a cabo una fase de depuración orientada a no contar aquellas columnas consideradas no relevantes para el análisis o potencialmente perjudiciales para la calidad del modelo. Dentro de esta categoría se incluyeron identificadores únicos como id\_local, coordenadas geográficas (latitud\_local, longitud\_local), y otras variables que, o bien aportaban escasa información explicativa, o bien podían introducir sesgos no deseados, como descripciones textuales (desc\_epigrafe, desc\_division) o derivadas temporales (Mes, Año).

Asimismo, se aplicó un criterio de exclusión adicional para remover dummies que podrían haber sido generadas a partir de variables eliminadas previamente. Esto se realizó mediante la identificación de prefijos comunes en los nombres de las columnas, como id\_, desc\_epigrafe\_ o desc\_division\_, entre otros, garantizando así la limpieza completa de variables potencialmente redundantes o irrelevantes.

Finalmente, con el objetivo de asegurar que el conjunto de datos resultante estuviera compuesto exclusivamente por variables numéricas, se identificaron y eliminaron aquellas columnas que aún conservaban tipos de datos no numéricos, como object o string. Este control final de calidad permitió consolidar un conjunto de datos homogéneo y listo para la siguiente fase del preprocesamiento.

#### *5.4.4 Incorporación de la variable temporal para la segmentación*

Una vez finalizado el proceso de limpieza, se reincorporó al conjunto de datos la variable Fecha\_Reporte, previamente excluida para evitar su eliminación accidental durante los pasos anteriores. Esta columna contiene información temporal sobre cada observación y resulta de particular importancia para realizar una segmentación cronológica del conjunto de datos, que es especialmente relevante cuando se pretende evaluar la capacidad predictiva de los modelos en un contexto prospectivo. La incorporación de esta variable temporal permite simular un entorno de predicción realista, respetando la secuencia cronológica natural de los datos.

#### *5.4.5 División del conjunto de datos en entrenamiento y prueba*

Con el propósito de evaluar de manera objetiva el rendimiento del modelo, se dividió el conjunto de datos en dos subconjuntos: entrenamiento (train) y prueba (test), utilizando como criterio la variable Fecha\_Reporte. Las observaciones anteriores a enero de 2024 fueron asignadas al conjunto de entrenamiento, mientras que aquellas correspondientes a enero de 2024 en adelante formaron parte del conjunto de prueba. Este tipo de partición basada en el tiempo es especialmente adecuado para tareas de predicción en series temporales o escenarios en los que la información evoluciona con el tiempo.

#### *5.4.6 Estandarización de variables numéricas*

Finalmente, se aplicó un proceso de estandarización sobre todas las variables numéricas incluidas en los conjuntos de entrenamiento y prueba. La estandarización se llevó a cabo utilizando la técnica de normalización tipo Z-score, implementada a través de la clase StandardScaler de la biblioteca scikit-learn. Esta técnica transforma cada variable para que tenga media cero y desviación estándar uno, lo que garantiza una contribución equitativa de todas las variables al proceso de aprendizaje del modelo. Es importante destacar que el escalador fue ajustado únicamente sobre el conjunto de entrenamiento, y posteriormente aplicado tanto al conjunto de entrenamiento como al de prueba. Esta práctica se usa para evitar cualquier tipo de fuga de información del conjunto de prueba al modelo, preservando así la validez del proceso de evaluación.

Estas serían las variables por las cuales se hará la selección de variables:

#### **Ilustración 29: Variables train**

```
Variables finales en X_train_scaled (25 variables):
1: Total_Poblacion
2: Renta_Media
3: desc_tipo_acceso_local_interior
4: desc_tipo_acceso_local_pc asociado
5: desc_tipo_acceso_local_puerta calle
6: desc_seccion_actividades artisticas, recreativas y de entretenimiento
7: desc_seccion_actividades de organizaciones y organismos extraterritoriales
8: desc_seccion_actividades financieras y de seguros
9: desc_seccion_actividades inmobiliarias
10: desc_seccion_actividades profesionales, cientificas y tecnicas
11: desc_seccion_actividades sanitarias y de servicios sociales
12: desc_seccion_administracion publica y defensa; seguridad social obligatoria
13: desc_seccion_agricultura, ganaderia, silvicultura y pesca
14: desc_seccion_comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas
15: desc_seccion_construccion
16: desc_seccion_educacion
17: desc_seccion_hosteleria
18: desc_seccion_industria manufacturera
19: desc_seccion_industrias extractivas
20: desc_seccion_informacion y comunicaciones
21: desc_seccion_otros servicios
22: desc_seccion_sin actividad
23: desc_seccion_suministro de agua, actividades de saneamiento, gestion de residuos y descontaminacion
24: desc_seccion_suministro de energia electrica, gas, vapor y aire acondicionado
25: desc_seccion_transporte y almacenamiento
```

Fuente: Elaboración propia

## 5.5 Selección de variables

En esta sección se presentan los distintos métodos empleados para la selección de variables, con el objetivo de identificar aquellas características con mayor capacidad explicativa y reducir la complejidad del modelo. Para ello, se combinaron enfoques estadísticos y de aprendizaje automático que permiten evaluar la relevancia de las variables desde distintas perspectivas como son la importancia en modelos de árboles, significancia estadística en regresión logística, rendimiento en validación cruzada o influencia marginal mediante valores SHAP. Tal como señala Xie et al. (2023), la integración de múltiples mecanismos de selección y evaluación de variables, incluyendo métodos estadísticos y algoritmos basados en árboles, junto con validación cruzada y explicaciones SHAP, garantiza que sólo los factores más relevantes y representativos se utilicen en la predicción, optimizando la capacidad explicativa y el rendimiento del modelo. Esta estrategia garantiza una selección adecuada y representativa de los factores más determinantes en la predicción de la actividad o inactividad de los locales en el entorno urbano analizado. Esto mejora el análisis de los modelos que aplicaremos de machine learning.

### 5.5.1 Boruta

Con el objetivo de reducir la dimensión del conjunto de datos y conservar únicamente las variables con mayor capacidad explicativa, se aplicó el algoritmo Boruta sobre una muestra aleatoria de 100.000 registros del conjunto de entrenamiento. Este algoritmo se basa en un modelo de Random Forest, configurado con una profundidad máxima de 5 y ponderación de clases para manejar posibles desequilibrios en la variable objetivo.

Boruta compara la importancia de cada variable real con la de variables aleatorias generadas a partir de permutaciones. A través de un proceso iterativo, determina si

cada variable original aporta más información que el azar. En este caso, el algoritmo completó 100 iteraciones, al cabo de las cuales confirmó un total de 13 variables como relevantes y descartó 10

Entre las variables seleccionadas destacan indicadores como Total\_Poblacion y Renta\_Media. Estas variables serán empleadas en las fases posteriores de modelado, al representar los factores con mayor influencia identificados de forma robusta por el algoritmo.

Variables: Variables seleccionadas por Boruta: ['Total\_Poblacion', 'Renta\_Media', 'desc\_tipo\_acceso\_local\_pc asociado', 'desc\_tipo\_acceso\_local\_puerta calle', 'desc\_seccion\_actividades artisticas, recreativas y de entretenimiento', 'desc\_seccion\_actividades financieras y de seguros', 'desc\_seccion\_actividades profesionales, cientificas y tecnicas', 'desc\_seccion\_actividades sanitarias y de servicios sociales', 'desc\_seccion\_comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas', 'desc\_seccion\_educacion', 'desc\_seccion\_hosteleria', 'desc\_seccion\_otros servicios', 'desc\_seccion\_sin actividad']

### 5.5.2 RFECV

Como parte del proceso de reducción de la dimensionalidad y selección de características relevantes, se utilizó la técnica Recursive Feature Elimination with Cross-Validation (RFECV). Este método permite identificar de forma automática el subconjunto óptimo de variables explicativas, eliminando de manera recursiva aquellas que aportan menos al rendimiento del modelo, basándose en validación cruzada.

Para ello, se trabajó con una muestra aleatoria de 50.000 registros del conjunto de entrenamiento, con el fin de reducir la carga computacional del proceso. Se empleó como estimador un modelo de Random Forest con 100 árboles y parámetros por defecto, en combinación con una validación cruzada estratificada de 5 particiones. El criterio de evaluación utilizado fue la el accuracy).

El algoritmo evaluó distintas combinaciones de variables, eliminando en cada iteración la menos significativa, hasta encontrar el conjunto que maximiza el rendimiento predictivo promedio a través de las particiones. El resultado final fue la selección de dos variables: Renta\_Media y desc\_seccion\_sin actividad

Este tipo de análisis es especialmente útil cuando se busca simplificar el modelo manteniendo una alta capacidad predictiva, y ofrece un enfoque complementario al método Boruta, permitiendo contrastar resultados y evaluar la estabilidad de las variables seleccionadas.

### 5.5.3 Stepwise Logistic Regression

Como complemento a los métodos anteriores de selección de variables, se aplicó un enfoque basado en regresión logística con selección hacia adelante y hacia atrás, conocido como stepwise. Esta técnica consiste en iterar sucesivamente sobre los

posibles predictores, añadiendo o eliminando variables en función de su significancia estadística hasta alcanzar un conjunto óptimo de características explicativas para el modelo.

Previamente a la ejecución del stepwise, se realizó una serie de preprocesamientos sobre una muestra estratificada de 500.000 observaciones del conjunto de entrenamiento. En primer lugar, se aplicó un umbral de varianza mínima para eliminar aquellas variables que presentaban escasa variabilidad, lo cual suele indicar escasa capacidad predictiva. Posteriormente, se eliminaron las variables altamente correlacionadas entre sí con el fin de reducir la multicolinealidad, que podría distorsionar la estimación de los coeficientes del modelo logístico.

El resultado final de este procedimiento fue la identificación de un conjunto amplio de 17 variables seleccionadas, que incluyen tanto variables numéricas como Renta\_Media, como una diversidad de variables categóricas transformadas a dummies, tales como la variable desc\_seccion y desc\_tipo\_acceso\_local . Esto indica que el modelo logró captar patrones relevantes desde distintas dimensiones socioeconómicas.

La selección stepwise proporciona, por tanto, una perspectiva basada en la significancia estadística individual y conjunta de las variables. Este enfoque es particularmente útil cuando se desea interpretar el impacto de cada variable en términos probabilísticos.

Variables: ['desc\_seccion\_hosteleria', 'desc\_seccion\_educacion', 'Total\_Poblacion', 'desc\_seccion\_actividades sanitarias y de servicios sociales', 'desc\_tipo\_acceso\_local\_puerta calle', 'desc\_seccion\_actividades financieras y de seguros', 'desc\_seccion\_informacion y comunicaciones', 'desc\_seccion\_industria manufacturera', 'desc\_seccion\_actividades inmobiliarias', 'desc\_seccion\_sin actividad', 'desc\_tipo\_acceso\_local\_pc asociado', 'desc\_seccion\_construccion', 'desc\_seccion\_actividades de organizaciones y organismos extraterritoriales', 'Renta\_Media', 'desc\_seccion\_transporte y almacenamiento', 'desc\_seccion\_otros servicios', 'desc\_seccion\_administracion publica y defensa; seguridad social obligatoria']

#### 5.5.4 SBF (*Sequential Backward Feature Selection*)

Esta vez se implementó una estrategia de eliminación secuencial hacia atrás, llamado SBF, utilizando un modelo base de regresión logística. Este método consiste en comenzar con el conjunto completo de variables explicativas e ir eliminando, de manera iterativa, aquellas que menos contribuyen al desempeño predictivo del modelo, medido en este caso a través del área bajo la curva ROC (AUC).

Dado el coste computacional elevado de este tipo de procedimientos, se utilizó una muestra estratificada de 50.000 observaciones del conjunto de entrenamiento, tamaño suficiente para preservar la representatividad de la distribución original y asegurar la viabilidad del proceso. La validación cruzada se implementó con tres particiones para reducir el tiempo de procesamiento sin comprometer excesivamente la robustez de la

evaluación. La configuración del selector se estableció para que determinara automáticamente el número óptimo de variables a conservar, en función del rendimiento del modelo.

Variables: Total\_Poblacion, desc\_tipo\_acceso\_local\_pc asociado, desc\_tipo\_acceso\_local\_puerta calle, desc\_seccion\_actividades artisticas, recreativas y de entretenimiento, desc\_seccion\_actividades inmobiliarias, desc\_seccion\_actividades sanitarias y de servicios sociales, desc\_seccion\_administracion publica y defensa; seguridad social obligatoria, desc\_seccion\_comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas, desc\_seccion\_construccion, desc\_seccion\_educacion, desc\_seccion\_hosteleria, desc\_seccion\_industria manufacturera, desc\_seccion\_sin actividad

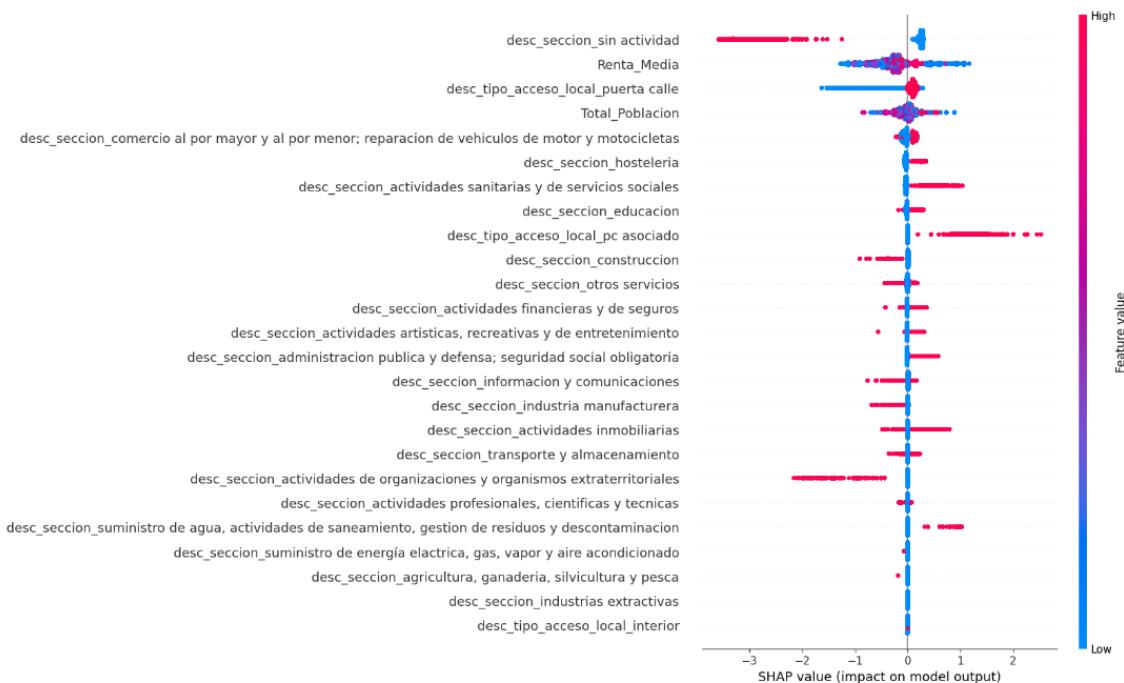
### 5.5.5 SHAP y XGBoost

Para complementar los métodos de selección supervisada previamente aplicados, se recurrió a un enfoque basado en modelos mediante SHAP (SHapley Additive exPlanations), técnica que permite interpretar la contribución de cada variable a las predicciones individuales realizadas por un modelo complejo como XGBoost. SHAP no se limita a determinar qué variables son importantes en promedio, sino que cuantifica con precisión la influencia marginal de cada característica en cada instancia del conjunto de datos.

El modelo de referencia fue un clasificador XGBoost entrenado sobre la totalidad del conjunto de entrenamiento escalado, con una configuración estándar optimizada para obtener buen rendimiento y evitar sobreajuste. Una vez ajustado el modelo, se construyó un objeto explainer sobre el conjunto de entrenamiento, que permitió estimar los valores SHAP sobre el conjunto de prueba.

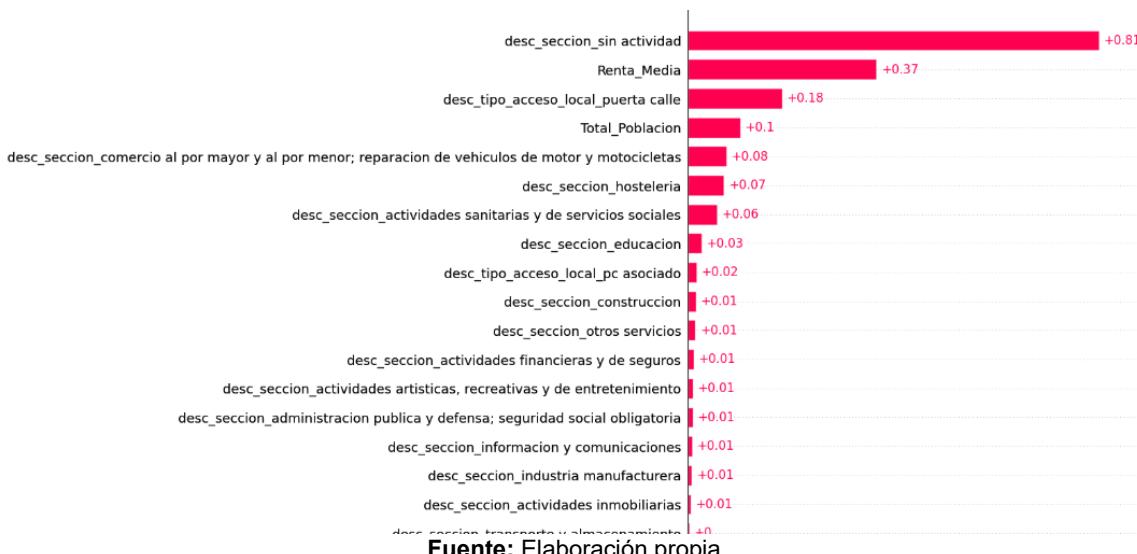
Variables: ['desc\_seccion\_sin actividad', 'Renta\_Media', 'desc\_tipo\_acceso\_local\_puerta calle', 'Total\_Poblacion', 'desc\_seccion\_comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas', 'desc\_seccion\_hosteleria', 'desc\_seccion\_actividades sanitarias y de servicios sociales', 'desc\_seccion\_educacion', 'desc\_tipo\_acceso\_local\_pc asociado', 'desc\_seccion\_construccion', 'desc\_seccion\_otros servicios', 'desc\_seccion\_actividades financieras y de seguros', 'desc\_seccion\_actividades artisticas, recreativas y de entretenimiento', 'desc\_seccion\_administracion publica y defensa; seguridad social obligatoria', 'desc\_seccion\_informacion y comunicaciones', 'desc\_seccion\_industria manufacturera', 'desc\_seccion\_actividades inmobiliarias', 'desc\_seccion\_transporte y almacenamiento', 'desc\_seccion\_actividades de organizaciones y organismos extraterritoriales', 'desc\_seccion\_actividades profesionales, cientificas y tecnicas', 'desc\_seccion\_suministro de agua, actividades de saneamiento, gestion de residuos y descontaminacion', 'desc\_seccion\_suministro de energia electrica, gas, vapor y aire acondicionado', 'desc\_seccion\_agricultura, ganaderia, silvicultura y pesca', 'desc\_seccion\_industrias extractivas', 'desc\_tipo\_acceso\_local\_interior']

**Ilustración 30: Sharp Summary Plot**



Fuente: Elaboración propia

**Ilustración 31: Sharp Bar Plot**



Fuente: Elaboración propia

### 5.5.6 Comparación de métodos de selección de variables

El método boruta destaca como la opción más equilibrada para la selección de variables, al combinar un rendimiento competitivo con una complejidad moderada. Con un AUC de 0.8495, muy cercano al valor más alto obtenido entre los métodos evaluados, demuestra una sólida capacidad para discriminar entre clases. Su accuracy de 0.8674 confirma un desempeño estable y comparable con alternativas más complejas. Un aspecto clave es su capacidad de reducir el conjunto de variables a 17.

Esta combinación de precisión, discriminación y simplicidad convierte a BORUTA en la elección más adecuada para los modelos de machine learning que se implementarán en esta investigación.

**Tabla 6:** Comparación de Métodos de Selección de Variables: Precisión (Accuracy), AUC y Nº de Variables

Método	Accuracy	AUC	Nº Variables
RFECV	0.8848	0.8220	2
BORUTA	0.8674	0.8495	17
STEPWISE	0.8674	0.8495	17
SBF	0.8775	0.8407	13
SHAP	0.8674	0.8502	25

Fuente: Elaboración propia

## Capítulo 6. Modelización

### 6.1 Modelos entrenados:

#### 6.1.1 Regresión logística

En esta sección se llevó a cabo la aplicación del modelo de regresión logística, con el objetivo de predecir la clase de la variable objetivo en función de un conjunto de variables previamente seleccionadas mediante el método Boruta. Para garantizar un enfoque riguroso y reproducible, se estableció una semilla aleatoria 12345 y se aumentó el número máximo de iteraciones permitidas en el optimizador, asegurando así la convergencia del modelo.

Se procedió a una búsqueda eficiente de hiperparámetros utilizando la técnica RandomizedSearchCV, que permite explorar aleatoriamente combinaciones dentro de un espacio definido. En este caso, se limitaron los valores a una regularización de tipo 'l2', con dos solvers compatibles ('liblinear' y 'saga'), y un rango logarítmico para el parámetro C, entre 0.001 y 100. Se utilizó una validación cruzada de tres particiones para cada combinación, evaluando el rendimiento mediante la métrica F1-score, con el fin de equilibrar precisión y exhaustividad.

Como resultado de esta búsqueda aleatoria, se seleccionó como mejor configuración la combinación de  $C = 2.918$ ,  $\text{penalty} = \text{'l2'}$  y  $\text{solver} = \text{saga}$ . Esta configuración proporciona un equilibrio entre la capacidad de generalización del modelo y su complejidad, manteniendo una penalización suave para evitar el sobreajuste. Una vez identificados los hiperparámetros óptimos, se procedió a realizar una validación cruzada adicional para evaluar el rendimiento general del modelo. Se calcularon métricas clave como precisión, recall, F1-score, accuracy y área bajo la curva ROC,

**Tabla 7:** Resultados de validación cruzada

Métrica	Media	Desviación estándar
Accuracy	0.9102	$\pm 0.0097$
Precision	0.9147	$\pm 0.0122$
Recall	0.9838	$\pm 0.0031$
F1	0.9479	$\pm 0.0052$
ROC AUC	0.8237	$\pm 0.0247$

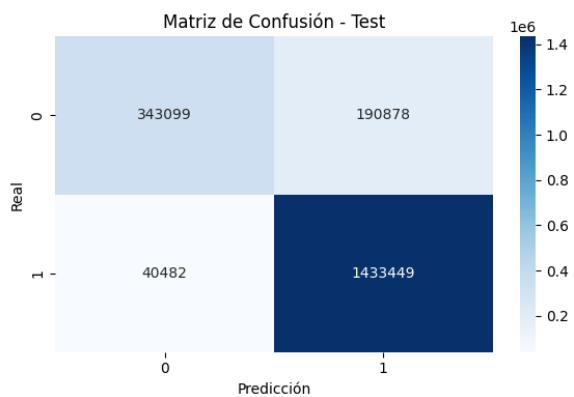
**Fuente:** Elaboración propia

**Tabla 8: Reporte de clasificación (test)**

Clase	Precision	Recall	F1-score	Support
0	0.89	0.64	0.75	533,977
1	0.88	0.97	0.93	1,473,931
<b>Accuracy</b>			0.88	2,007,908
<b>Macro avg</b>	0.89	0.81	0.84	2,007,908
<b>Weighted avg</b>	0.89	0.88	0.88	2,007,908

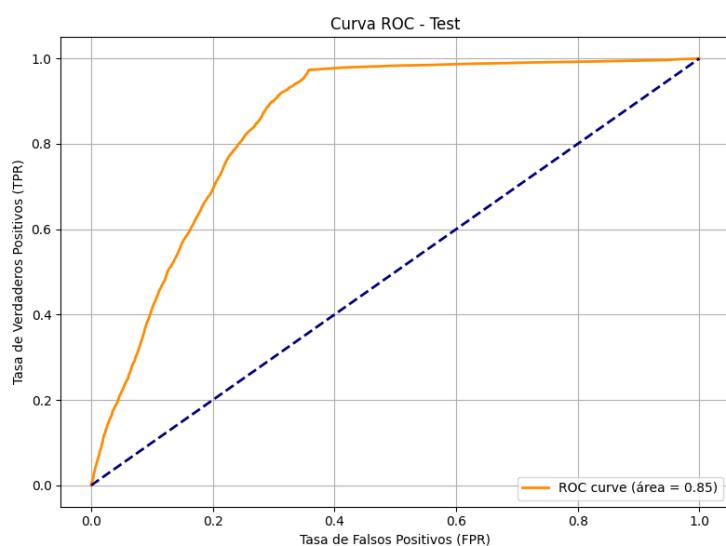
**Fuente:** Elaboración propia

**Ilustración 32: Matriz de confusión**



**Fuente:** Elaboración propia

**Ilustración 33: Curva ROC**



**Fuente:** Elaboración propia

El modelo de Regresión Logística presentó, en la validación cruzada, una exactitud promedio del 91,02%, con una precisión del 91,47% y un recall del 98,38%, lo que refleja una gran capacidad para identificar correctamente los casos positivos. El F1-

Score fue de 94,79% y el AUC de 0.8237, indicando una buena capacidad de diferenciación entre clases. En el conjunto de prueba, alcanzó una exactitud global del 88%. La clase 1 presentó un recall de 0,97, lo que confirma que el modelo detecta casi todos los casos positivos, aunque con una precisión de 0,88 debido a un número moderado de falsos positivos. Para la clase 0, la precisión fue de 0,89 y el recall de 0,64, evidenciando mayores dificultades en la detección de negativos. La matriz de confusión mostró un elevado número de aciertos en positivos y un rendimiento menor en negativos. La curva ROC arrojó un AUC de 0.85, reafirmando una buena capacidad discriminativa.

### *6.1.2 Árbol de decisión*

En esta sección se llevó a cabo la aplicación del modelo de árbol de decisión. Para garantizar la reproducibilidad del experimento, se fijó una semilla aleatoria con valor 12345, lo que asegura consistencia en los resultados durante cada ejecución.

A continuación, se procedió a la optimización de hiperparámetros utilizando la técnica GridSearchCV, que permite realizar una búsqueda exhaustiva sobre un conjunto predefinido de combinaciones de parámetros. En particular, se exploraron los valores de profundidad máxima del árbol (`max_depth`), el número mínimo de muestras requeridas para dividir un nodo interno (`min_samples_split`) y el número mínimo de muestras que debe tener una hoja (`min_samples_leaf`). Para cada combinación, se aplicó validación cruzada con cinco particiones (5-fold), evaluando el desempeño mediante la métrica F1-score, con el objetivo de capturar un buen balance entre precisión y recall.

Como resultado de esta búsqueda sistemática, se seleccionó como mejor configuración aquella que presentaba un equilibrio entre la complejidad del árbol y su capacidad de generalización. Los hiperparámetros óptimos obtenidos fueron: `max_depth = 5`, `min_samples_split = 2` y `min_samples_leaf = 4`. Esta configuración permitió limitar la profundidad del árbol para reducir el riesgo de sobreajuste, al tiempo que se mantuvieron niveles adecuados de particionamiento y tamaño de las hojas para capturar patrones relevantes en los datos.

Una vez identificado el mejor conjunto de hiperparámetros, se utilizó el modelo resultante para generar predicciones sobre el conjunto de prueba.

**Tabla 9:** Matriz de confusión

	<b>Predicción: Positivo</b>	<b>Predicción: Negativo</b>	<b>Total, Real</b>
Real: Positivo	198,796	335,181	533,977
Real: Negativo	46,822	1,427,109	1,473,931
Total, Predicción	245,618	1,762,290	2,007,908

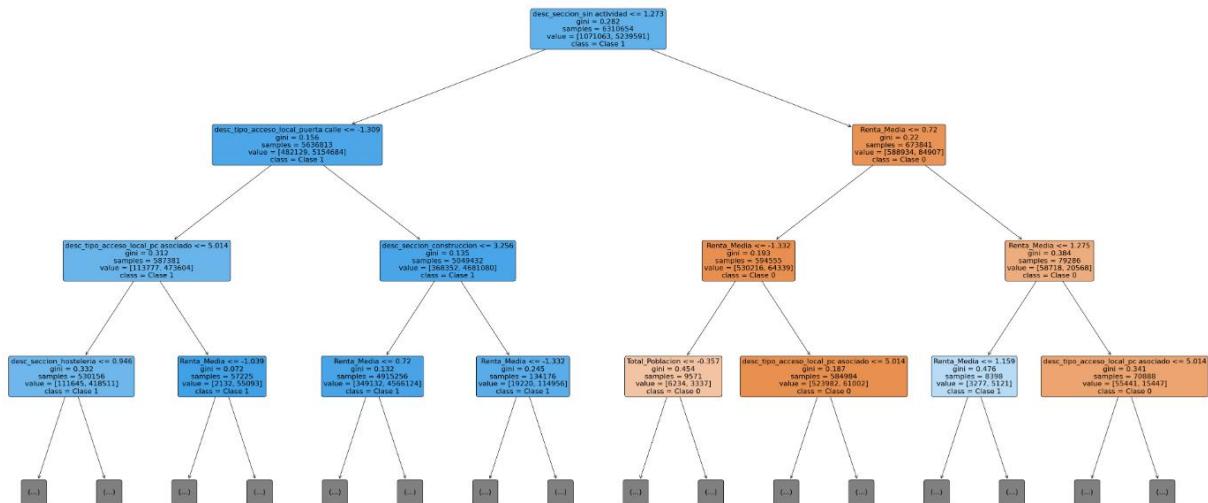
**Fuente:** Elaboración propia

**Tabla 10:** Métricas de Evaluación por Clase y Globales — Árbol de decisión

Clase	Precisión	Recall	F1-score	Sopporte
0	0.81	0.37	0.51	533,977
1	0.81	0.97	0.88	1,473,931
<b>Accuracy</b>			0.81	2,007,908
<b>Macro promedio</b>	0.81	0.67	0.70	2,007,908
<b>Promedio ponderado</b>	0.81	0.81	0.78	2,007,908

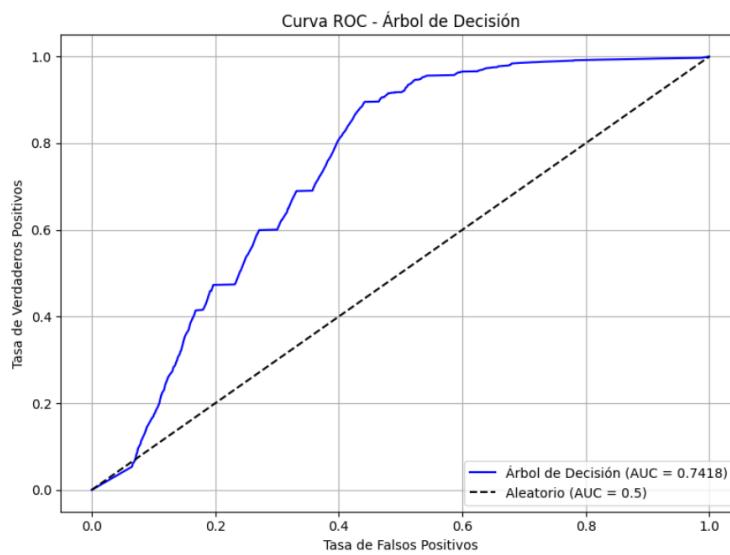
**Fuente:** Elaboración propia

**Ilustración 34: Árbol de decisión**



**Fuente:** Elaboración propia

**Figura 35:**  
**Ilustración 35: Curva de ROC - Árbol de decisión**



**Fuente:** Elaboración propia

El modelo de Árbol de Decisión, con una profundidad máxima de 4 niveles, alcanzó una exactitud global del 0,81 en el conjunto de prueba, mostrando un buen desempeño especialmente en la identificación de la clase 1. Esta clase obtuvo un recall de 0,97 y una precisión de 0,81, lo que indica que la gran mayoría de los

casos negativos reales fueron correctamente detectados, aunque con cierta presencia de falsos positivos. En contraste, la clase 0 presentó un desempeño más limitado, con un recall de 0,37 y la misma precisión de 0,81, lo que señala dificultades para reconocer correctamente los casos positivos. El puntaje F1 fue de 0,88 para la clase 1 y de 0,51 para la clase 0, evidenciando un mejor equilibrio entre precisión y exhaustividad en la clase negativa.

El análisis del árbol revela que la variable más influyente para la clasificación fue desc\_sección\_sin\_actividad, que segmenta el conjunto en dos grandes grupos, seguidos por otras variables como desc\_tipo\_acceso\_local\_puertacalle, y Renta\_Media, que refinan la. La matriz de confusión evidencia un sesgo hacia la predicción de la clase positiva, con un número considerable de falsos positivos, y la curva ROC, con un AUC de 0.7418, confirma una capacidad baja en comparación a los otros modelos.

### 6.1.3 XGBoost

El objetivo fue predecir la clase de la variable objetivo a partir de un conjunto de variables previamente seleccionadas mediante el método Boruta. Para asegurar la reproducibilidad de los resultados, se fijó una semilla aleatoria con valor 12345 en el modelo base.

Para optimizar el rendimiento del clasificador, se llevó a cabo una búsqueda exhaustiva de hiperparámetros mediante la técnica GridSearchCV. En esta búsqueda se evaluaron diferentes combinaciones de los siguientes parámetros: profundidad máxima del árbol (max\_depth), tasa de aprendizaje (learning\_rate), número de árboles (n\_estimators) y proporción de muestras empleadas en cada iteración (subsample). Se fijó la métrica de evaluación en AUC-ROC, dado que se trata de un indicador robusto y adecuado para problemas con posibles desbalances en las clases.

Como resultado de esta exploración sistemática, se seleccionó como mejor configuración aquella que combinaba un max\_depth = 5, learning\_rate = 0.1, n\_estimators = 200 y subsample = 0.8. Esta combinación permite construir un modelo no excesivamente profundo, con un ritmo de aprendizaje moderado y una muestra parcial de observaciones en cada iteración, lo cual contribuye a reducir el sobreajuste y mejorar la capacidad de generalización.

Una vez identificados los hiperparámetros óptimos, se realizó una validación cruzada con el modelo ajustado, aplicando diversas métricas de evaluación como accuracy, precision, recall, F1-score y AUC. Esta evaluación permitió obtener una visión integral del rendimiento del modelo sobre diferentes subconjuntos del conjunto de entrenamiento, incluyendo tanto medidas de clasificación directa como métricas basadas en probabilidades.

*Tabla 11: Resultados de validación cruzada*

Métrica	Media	Desviación estándar
Accuracy	0.9121	± 0.0002
Precision	0.9143	± 0.0001
Recall	0.9867	± 0.0001
F1-Score	0.9491	± 0.0001

ROC AUC	0.8635	$\pm 0.0005$
---------	--------	--------------

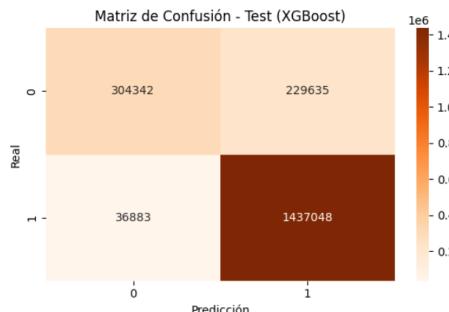
**Fuente:** Elaboración propia

**Tabla 12:** Reporte de clasificación XGBoost

Clase	Precision	Recall	F1-score	Sopporte
0	0.89	0.57	0.70	533,977
1	0.86	0.97	0.92	1,473,931
<b>Accuracy global</b>			0.87	2,007,908
<b>Macro avg</b>	0.88	0.77	0.81	2,007,908
<b>Weighted avg</b>	0.87	0.87	0.86	2,007,908

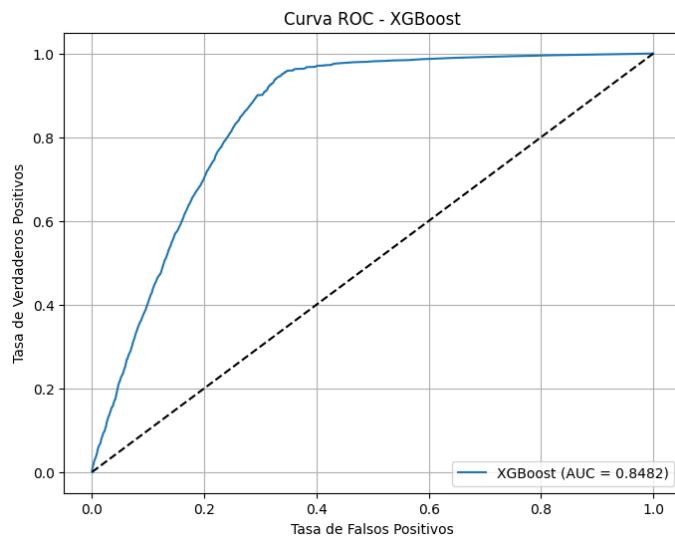
**Fuente:** Elaboración propia

**Ilustración 36:** Matriz de confusión - XGBoost



**Fuente:** Elaboración propia

**Ilustración 37:** Curva de ROC - XGBoost



**Fuente:** Elaboración propia

El modelo XGBoost presenta un desempeño estable destacando por su alta capacidad de detección de la clase positiva con un recall de 0.9867 y F1-score de 0.9491 en test, lo que indica que identifica correctamente la gran mayoría de casos relevantes, aunque a con un menor recall en la clase 0 con valor de 0.57, lo que genera un número considerable de falsos positivos. Con un AUC promedio de

0.8635 en validación cruzada y 0.8491 en test, el modelo muestra una buena capacidad discriminativa y resultados consistentes, lo que lo hace especialmente adecuado para escenarios donde es prioritario minimizar falsos negativos

#### 6.1.4 KNN

En esta sección se aplicó el modelo K-Nearest Neighbors (KNN) y dado que es un algoritmo que puede ser computacionalmente costoso en conjuntos de datos grandes, se optó por realizar un muestreo estratificado sobre el conjunto de entrenamiento, seleccionando 100.000 observaciones de forma aleatoria pero representativa. Esta estrategia permitió reducir el tiempo de entrenamiento sin comprometer significativamente la representatividad de los datos.

Para facilitar el preprocesamiento y la evaluación del modelo, se implementó un pipeline que incluía dos pasos: la estandarización de las variables mediante StandardScaler y el clasificador KNeighborsClassifier. Esta estandarización es fundamental en KNN, dado que las distancias euclidianas utilizadas por el modelo son sensibles a la escala de las variables.

Con el objetivo de encontrar la mejor configuración posible del modelo, se aplicó una búsqueda exhaustiva de hiperparámetros mediante GridSearchCV. La rejilla de búsqueda incluyó combinaciones de los siguientes parámetros: número de vecinos (n\_neighbors), esquema de ponderación (weights), y métrica de distancia (metric). Para esta prueba inicial, se redujo el espacio de búsqueda a combinaciones relativamente pequeñas: n\_neighbors = [3, 5], weights = 'distance' y metric = 'euclidean', priorizando la eficiencia del ajuste.

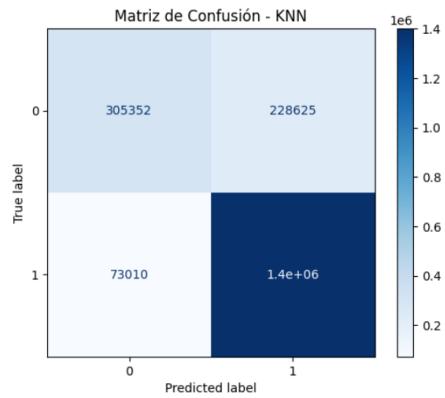
Tras la búsqueda, se identificó como mejor configuración aquella compuesta por n\_neighbors = 5, weights = 'distance' y metric = 'euclidean'. Esta configuración permitió al modelo ponderar más las observaciones cercanas y utilizar una métrica estándar de distancia que se ajusta bien a datos previamente estandarizados.

**Tabla 13:** Métricas de Clasificación del Modelo KNN en el Conjunto de Prueba

Clase	Precisión	Recall	F1-Score	Soporte
0	0.81	0.57	0.67	533,977
1	0.86	0.95	0.90	1,473,931
<b>Accuracy</b>			0.85	2,007,908
<b>Macro Avg</b>	0.83	0.76	0.79	2,007,908
<b>Weighted Avg</b>	0.85	0.85	0.84	2,007,908

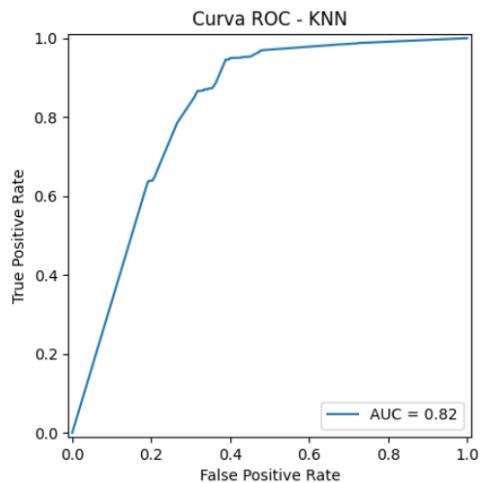
**Fuente:** Elaboración propia

**Ilustración 38: Matriz de confusión - KNN**



Fuente: Elaboración propia

**Ilustración 39: Curva de ROC - KNN**



Fuente: Elaboración propia

El modelo KNN obtiene un rendimiento adecuado, con un accuracy del 86% y un AUC de 0.8189, lo que refleja una buena capacidad discriminativa entre clases, aunque algo menor que la observada en modelos más complejos como XGBoost. Presenta un mejor balance entre precisión y recall que en la clase negativa con valores de 0.81 y 0.57 respectivamente en comparación con XGBoost, reduciendo parcialmente los falsos positivos, pero aumenta los falsos negativos. En la clase 1, logra un recall de 0.86 y un F1-score de 0.95, lo que indica una correcta detección de la mayoría de los casos relevantes.

#### 6.1.5 Random Forest

En esta sección se aplicó el modelo Random Forest y dado el tamaño del conjunto de entrenamiento y con el fin de evitar problemas de memoria durante el ajuste de hiperparámetros, se realizó un submuestreo aleatorio estratificado de 100.000

observaciones. Esta medida permitió mantener un equilibrio entre rendimiento computacional y representatividad de los datos. A continuación, se definió una rejilla reducida de hiperparámetros para aplicar una búsqueda exhaustiva mediante GridSearchCV. La malla contempló combinaciones de los siguientes parámetros: número de árboles (n\_estimators), profundidad máxima del árbol (max\_depth), número mínimo de muestras para dividir un nodo interno (min\_samples\_split) y número mínimo de muestras por hoja (min\_samples\_leaf). La validación se llevó a cabo utilizando una validación cruzada estratificada con tres particiones y se empleó como métrica de evaluación principal el F1-score, dada su utilidad en contextos con clases potencialmente desbalanceadas.

Como resultado del proceso de ajuste, se obtuvo como configuración óptima un modelo con n\_estimators = 150, max\_depth = 10, min\_samples\_split = 2 y min\_samples\_leaf = 1. Esta combinación proporcionó un equilibrio adecuado entre complejidad y generalización, permitiendo capturar relaciones no lineales sin incurrir en sobreajuste.

Con los hiperparámetros óptimos definidos, se llevó a cabo una nueva validación cruzada sobre el conjunto de entrenamiento submuestreado, calculando métricas clave como precisión, recall, F1-score, accuracy y área bajo la curva ROC. Esta evaluación permitió cuantificar el rendimiento del modelo y su estabilidad en distintos subconjuntos del entrenamiento.

Finalmente, se entrenó el modelo final con los datos submuestreados y se evaluó su desempeño sobre el conjunto de prueba completo. Se generaron las predicciones de clase y probabilidades, calculando la matriz de confusión y un reporte detallado de clasificación.

**Tabla 14:** Reporte de Clasificación – Test con Random Forest

Métrica	Media	Desviación Estándar
Accuracy	0.9108	± 0.0009
Precisión	0.9140	± 0.0005
Recall	0.9854	± 0.0012
F1-Score	0.9483	± 0.0006
ROC AUC	0.8464	± 0.0019

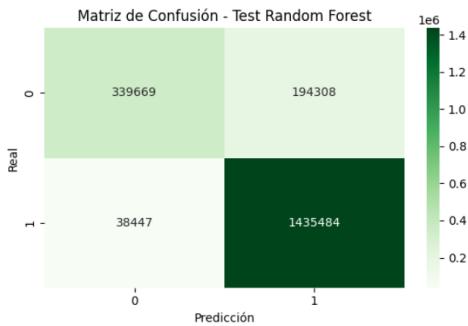
Fuente: Elaboración propia

**Tabla 15:** Reporte de Clasificación – Test con Random Forest

Clase	Precisión	Recall	F1-Score	Soporte
0	0.90	0.64	0.74	533,977
1	0.88	0.97	0.93	1,473,931
<b>Accuracy</b>			0.88	2,007,908
<b>Macro Avg</b>	0.89	0.81	0.83	2,007,908
<b>Weighted Avg</b>	0.89	0.88	0.88	2,007,908

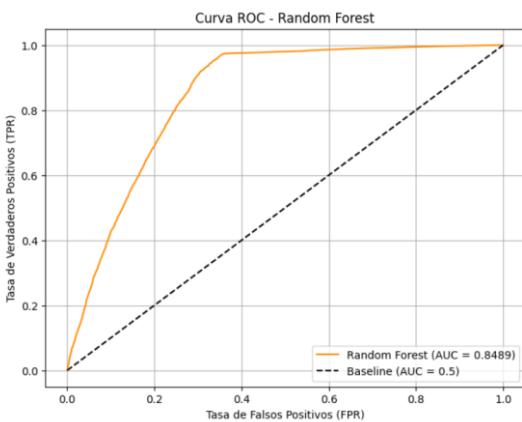
Fuente: Elaboración propia

**Ilustración 40: Matriz de confusión Random Forest**



Fuente: Elaboración propia

**Ilustración 41: Curva ROC - Random Forest**



Fuente: Elaboración propia

El modelo Random Forest evaluado muestra un desempeño sobresaliente, destacando por su alta capacidad de detección de la clase 1 con un recall de 0.9854 y un equilibrio adecuado entre precisión y sensibilidad con el valor de F1-Score de 0.9483. La matriz de confusión revela un número reducido de falsos negativos y una ligera tendencia a clasificar en favor de la clase 1, lo que incrementa los falsos positivos. El valor de AUC de 0.8489 confirma una buena capacidad discriminativa, mientras que las métricas por clase evidencian un rendimiento superior en la detección de la clase 1 respecto a la clase 0. El modelo presenta un buen rendimiento.

### 6.1.6 SVM

Para complementar el análisis, se implementó un modelo de Support Vector Machine (SVM), un algoritmo ampliamente reconocido por su capacidad para encontrar fronteras de decisión óptimas en espacios de alta dimensionalidad. Debido a su alto coste computacional, especialmente cuando se aplica sobre grandes volúmenes de datos, fue necesario aplicar un submuestreo del conjunto de entrenamiento. En este caso, se seleccionaron 100.000 observaciones representativas para acelerar tanto el ajuste del modelo como la búsqueda de

hiperparámetros, sin comprometer significativamente la distribución de clases ni la capacidad del modelo para generalizar.

El modelo base definido fue una instancia de SVC de scikit-learn, configurado inicialmente sin estimación de probabilidades, lo cual permitió optimizar el tiempo de entrenamiento. A continuación, se construyó una rejilla de búsqueda limitada, centrada en dos valores del hiperparámetro de regularización C (0.1 y 1), manteniendo el kernel de tipo lineal (kernel='linear'), dada la alta dimensionalidad del conjunto de datos y con el objetivo de reducir la complejidad computacional.

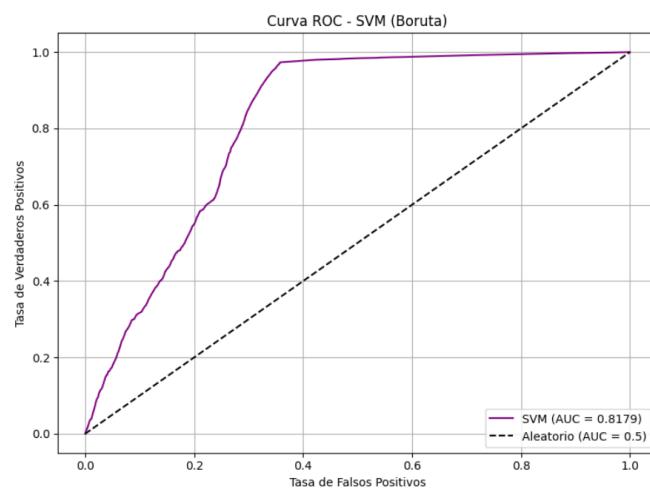
Para el proceso de ajuste de hiperparámetros se empleó GridSearchCV con validación cruzada priorizando la métrica F1-score como criterio de optimización. Esta decisión se tomó para mitigar posibles efectos del desbalance de clases, y también para acelerar el proceso dada la alta carga computacional de este tipo de modelos. Como resultado, el modelo óptimo fue un clasificador SVM con kernel lineal y C = 1. Una vez identificado el mejor estimador, se evaluó su rendimiento sobre el conjunto de prueba completo y se calcularon las métricas fundamentales.

**Tabla 16:** Métricas de Desempeño – Modelo SVM en Test

Métrica	Valor
Accuracy	0.8848
Precisión	0.8825
Recall	0.9727
F1-Score	0.9254

**Fuente:** Elaboración propia

**Ilustración 42:** Curva ROC SVM



**Fuente:** Elaboración propia

**Tabla 17:** Matriz de confusión

	Predicho: 0	Predicho: 1
Real: 0	343,039	190,938
Real: 1	40,309	1,433,622

**Fuente:** Elaboración propia

El modelo SVM evaluado obtiene un buen desempeño en el conjunto de prueba, con un accuracy de 0.8848 y un F1-Score de 0.9254, reflejando un buen balance entre precisión con un valor de 0.8825 y recall con 0.9727. La matriz de confusión muestra un elevado número de verdaderos positivos y verdaderos negativos, lo que indica una alta capacidad para identificar la clase 1, aunque con cierta tendencia a clasificar instancias como positivas. El valor de AUC-ROC de 0.8179 confirma una capacidad discriminativa adecuada, aunque inferior a la observada entre los modelos anteriores.

#### 6.1.7 Redes Neuronales

Para evaluar el uso de redes neuronales en la tarea de clasificación, se empleó un modelo Multi-Layer Perceptron (MLP) implementado con el clasificador MLPClassifier de la librería scikit-learn. Este tipo de red neuronal, también llamada perceptrón multicapa, está formada por varias capas de neuronas conectadas entre sí, que aplican funciones de activación no lineales. Esto permite al modelo aprender patrones complejos en los datos que no siguen relaciones directas o proporcionales (es decir, relaciones no lineales), algo que los modelos lineales tradicionales no pueden capturar, mejorando así su capacidad de clasificación.

Dada la naturaleza computacionalmente intensiva de este tipo de modelos se aplicaron diversas estrategias para optimizar el proceso. El espacio de búsqueda de hiperparámetros fue definido de forma acotada pero representativa, incluyendo configuraciones con una o dos capas ocultas de 50 y 100 neuronas respectivamente, la función de activación ReLU, diferentes valores de regularización que es el valor alpha, y estrategias de aprendizaje constantes o adaptativas. Se empleó early stopping para evitar el sobreajuste, y se limitó a 200 iteraciones por configuración. La selección de hiperparámetros se realizó mediante GridSearchCV con validación cruzada priorizando la métrica F1-score para balancear la precisión y el recall en escenarios con clases desbalanceadas.

El modelo óptimo encontrado fue una red neuronal con una capa oculta de 100 neuronas, función de activación ReLU, regularización alpha = 0.0001. Este modelo fue reentrenado con el conjunto completo de entrenamiento utilizando las variables seleccionadas por Boruta.

**Tabla 18:** Matriz de confusión – MLP Red Neuronal

	Predictión Negativa	Predictión Positiva
Clase Negativa	343,294	190,683
Clase Positiva	40,696	1,433,235

**Fuente:** Elaboración propia

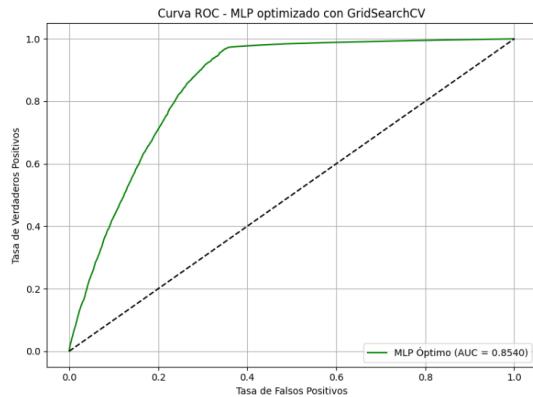
**Tabla 19:** Métricas Globales – MLP Red Neuronal

Métrica	Valor
Accuracy	0.8848
Precisión	0.8826
Recall	0.9724

F1-Score	0.9253
----------	--------

Fuente: Elaboración propia

Ilustración 43: Curva ROC – Red neuronal MLP



Fuente: Elaboración propia

El modelo de red neuronal MLP presenta un desempeño sólido y bueno en la tarea de clasificación. La curva ROC muestra un área bajo la curva, el valor AUC, de 0.8540, lo que indica una alta capacidad discriminativa para diferenciar entre clases positiva y negativa. La curva se aproxima al vértice superior izquierdo del gráfico, evidenciando una buena relación entre la tasa de verdaderos positivos y la tasa de falsos positivos. En la matriz de confusión, se observa que el modelo clasifica correctamente un alto número de instancias de ambas clases, con 1,433,235 verdaderos positivos y 343,294 verdaderos negativos. Sin embargo, persiste un volumen considerable de falsos positivos lo que podría implicar costos dependiendo del contexto de aplicación, y un menor número de falsos negativos lo que refleja una elevada sensibilidad. El modelo alcanza una exactitud Accuracy de 88.48%, con una precisión de 88.26%, lo que significa que la gran mayoría de las predicciones positivas corresponden efectivamente a la clase positiva. El Recall de valor 97.24% es particularmente alto, reflejando que el modelo detecta casi la totalidad de los casos positivos reales. El F1-Score de 92.53% confirma un balance adecuado entre precisión y sensibilidad.

## 6.2 Ensamblado de modelos

### 6.2.1 VotingClassifier

Con el objetivo de mejorar el rendimiento del sistema de clasificación, se construyó un modelo de ensamblado utilizando la técnica de soft voting. Para ello, se seleccionaron previamente las variables más relevantes a través del algoritmo Boruta, y se escalaron los conjuntos de entrenamiento y prueba para asegurar la correcta convergencia de los modelos involucrados. El ensamblado se construyó con tres clasificadores base que serían el Random Forest, el modelo XGBoost y la red neuronal tipo ML, estos tres modelos también tuneados y buscando los mejores parámetros.

La combinación de modelos se implementó mediante VotingClassifier de scikit-learn combinando las probabilidades de predicción de cada estimador base. Se asignaron pesos diferenciados a cada modelo con el objetivo de reflejar la confianza relativa en el rendimiento individual de cada uno. Tras entrenar el ensamblador sobre los datos seleccionados, se evaluó su desempeño sobre el conjunto de prueba.

**Tabla 20:** Matriz de confusión – VotingClassifier

Clase Real \ Predicha	Predicha 0	Predicha 1
Real 0	337,584	196,393
Real 1	38,984	1,434,947

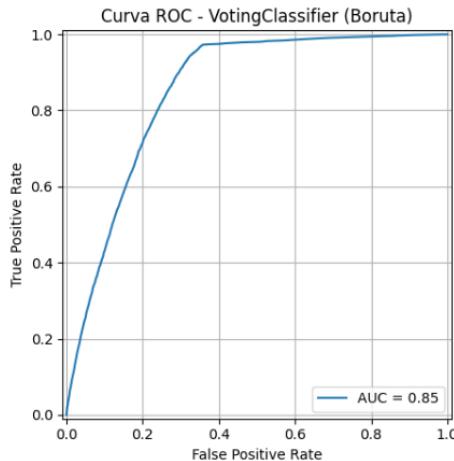
Fuente: Elaboración propia

**Tabla 21:** Métricas– VotingClassifier

Clase / Métrica	Precisión	Recall	F1-Score	Soporte
<b>Clase 0</b>	0.90	0.63	0.74	533,977
<b>Clase 1</b>	0.88	0.97	0.92	1,473,931
<b>Accuracy</b>	—	—	0.88	2,007,908
<b>Macro promedio</b>	0.89	0.80	0.83	2,007,908
<b>Weighted promedio</b>	0.88	0.88	0.88	2,007,908

Fuente: Elaboración propia

**Ilustración 44:** Curva ROC – VotingClassifier



Fuente: Elaboración propia

El modelo VotingClassifier con selección de características mediante Boruta presenta un rendimiento global competitivo, con un AUC de 0.8541, lo que indica una elevada capacidad de discriminación entre clases. La curva ROC evidencia que, en la mayoría de los umbrales de decisión, el modelo logra una alta tasa de verdaderos positivos con una tasa moderada de falsos positivos, acercándose al rendimiento ideal. En la matriz de confusión, se registran un valor alto de falsos positivos a comparación de los falsos negativos. Este comportamiento sugiere una ligera inclinación del modelo hacia la clasificación 1, lo que eleva la sensibilidad, pero puede impactar en la precisión de la clase negativa.

### 6.2.2 StackingClassifier

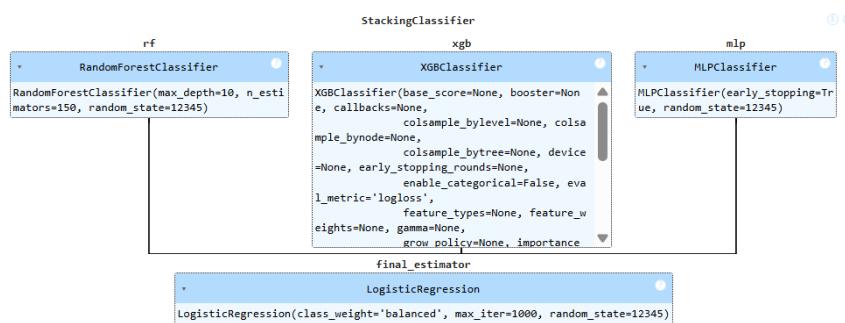
A diferencia del ensamblado por votación, el apilamiento (stacking) permite combinar múltiples clasificadores base mediante un modelo final que aprende a partir de las predicciones generadas por los modelos individuales. Este enfoque suele aportar una mejora en el rendimiento, al aprovechar patrones complementarios entre modelos heterogéneos. En este caso, el StackingClassifier se construyó utilizando tres modelos base previamente optimizados que serían Random Forest, XGBoost y MLPClassifier, esto con los mejores hiperparámetros.

Como modelo se utilizó una regresión logística con ajuste de pesos de clase, pensada para compensar posibles desbalanceos en los datos y mejorar la sensibilidad frente a la clase minoritaria. El modelo fue entrenado utilizando validación cruzada interna sobre las predicciones de los clasificadores base.

Al igual que en modelos anteriores, se utilizó el subconjunto de variables seleccionadas mediante el método boruta, asegurando que solo se incluyeran características estadísticamente relevantes para la tarea de clasificación. Los datos se transformaron a arrays NumPy para mejorar el rendimiento computacional durante el entrenamiento.

Tras el entrenamiento del modelo, se evaluó su rendimiento en el conjunto de prueba utilizando métricas como la matriz de confusión, el reporte de clasificación y el AUC, complementadas con visualizaciones de la curva ROC y la propia matriz. Este enfoque permitió analizar tanto el rendimiento global como el comportamiento por clase.

**Ilustración 45:** Gráfico de la composición de StackingClassifier



Fuente: Elaboración propia

**Tabla 22:** Matriz de confusión StackingClassifier

Clase Real \ Predicha	Predicha 0	Predicha 1
Real 0	347,595	186,382
Real 1	132,987	1,340,944

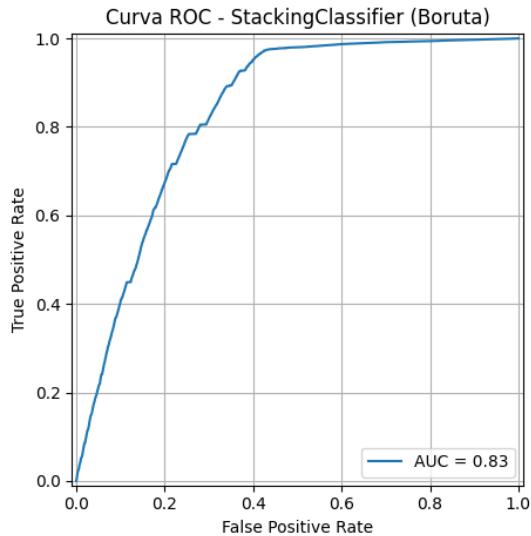
Fuente: Elaboración propia

**Tabla 23:** Reporte de Clasificación Completo StackingClassifier

Clase / Métrica	Precisión	Recall	F1-Score	Soporte
Clase 0	0.72	0.65	0.69	533,977
Clase 1	0.88	0.91	0.89	1,473,931
Accuracy	—	—	0.84	2,007,908
Macro promedio	0.80	0.78	0.79	2,007,908
Weighted promedio	0.84	0.84	0.84	2,007,908

**Fuente:** Elaboración propia

**Ilustración 46:** Curva ROC StackingClassifier



**Fuente:** Elaboración propia

El StackingClassifier obtiene un AUC de 0.83, lo que indica una capacidad de discriminación aceptable, aunque inferior a la observada en otros modelos evaluados. La curva ROC refleja que el modelo mantiene una buena proporción de verdaderos positivos frente a falsos positivos, pero con un rendimiento balanceado comparación con alternativas como el VotingClassifier. En la matriz de confusión, el modelo clasifica correctamente 1,340,944 instancias de la clase positiva y 347,595 de la clase negativa. Sin embargo, presenta un número elevado de falsos negativos con 132,987, lo que implica que una fracción importante de casos positivos no es detectada. Asimismo, se registran 186,382 falsos positivos, evidenciando cierta dificultad para discriminar con precisión la clase negativa. El reporte de clasificación muestra una precisión para la clase positiva de 0.88 y un recall de 0.91, lo que indica que el modelo es competente identificando la mayoría de los casos positivos reales, aunque no alcanza los valores de recall de modelos previamente analizados.

## Capítulo 7. Resultados

### 7.1 Comparación de modelos

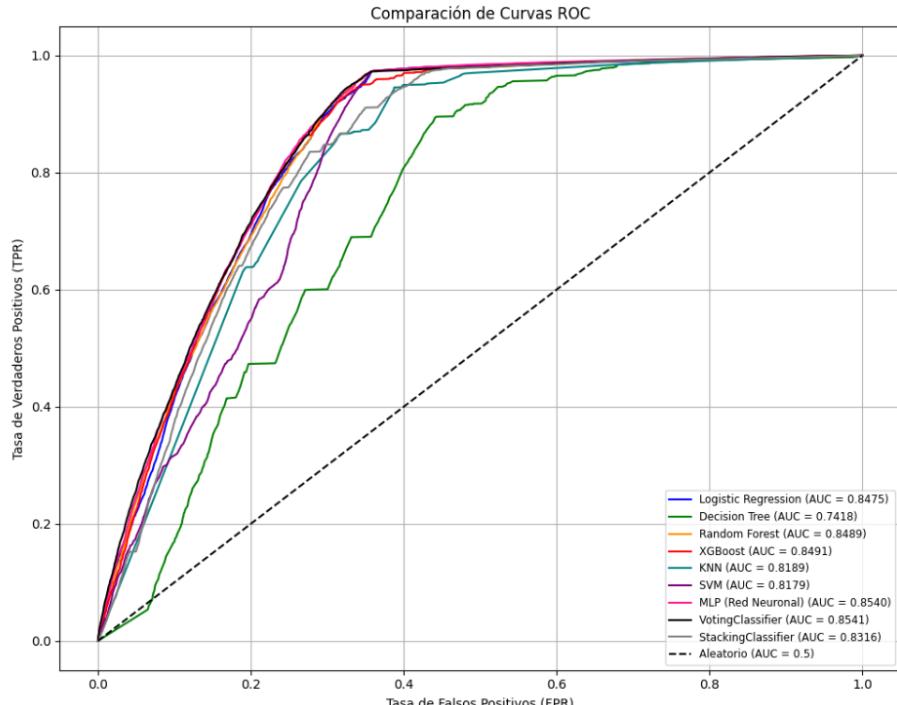
**Tabla 24:** Comparación de métrica de los modelos

Modelo	Accuracy	Precision	Recall	F1 Score	AUC
StackingClassifier	0.88	0.84	0.91	0.89	0.83

VotingClassifier	0.8828	0.8796	0.9736	0.9242	0.8541
MLP (Red Neuronal)	0.8848	0.8826	0.9724	0.9253	0.8540
XGBoost	0.8673	0.8622	0.9750	0.9151	0.8491
Random Forest	0.8841	0.8808	0.9739	0.9250	0.8489
Logistic Regression	0.8848	0.8825	0.9725	0.9253	0.8475
StackingClassifier	0.8409	0.8780	0.9098	0.8936	0.8316
KNN	0.8498	0.8597	0.9505	0.9028	0.8189
SVM	0.8848	0.8825	0.9727	0.9254	0.8179
Decision Tree	0.8098	0.8098	0.9682	0.8820	0.7418

Fuente: Elaboración propia

Ilustración 47: Curva ROC – Todos los modelos



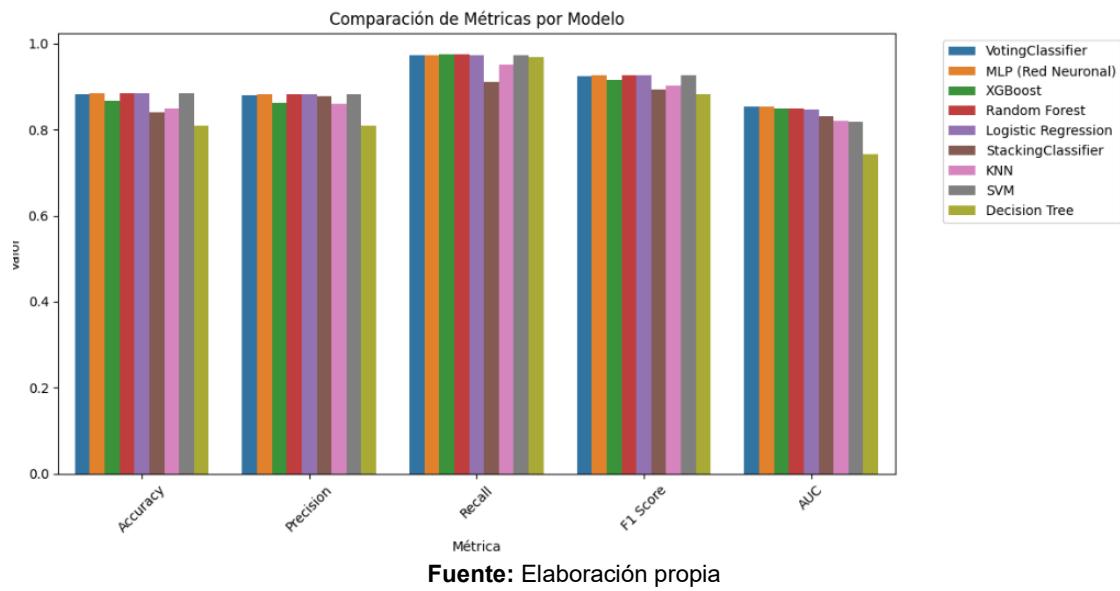
Fuente: Elaboración propia

Tabla 25: Verdaderos negativos y especificidad de todos los modelos

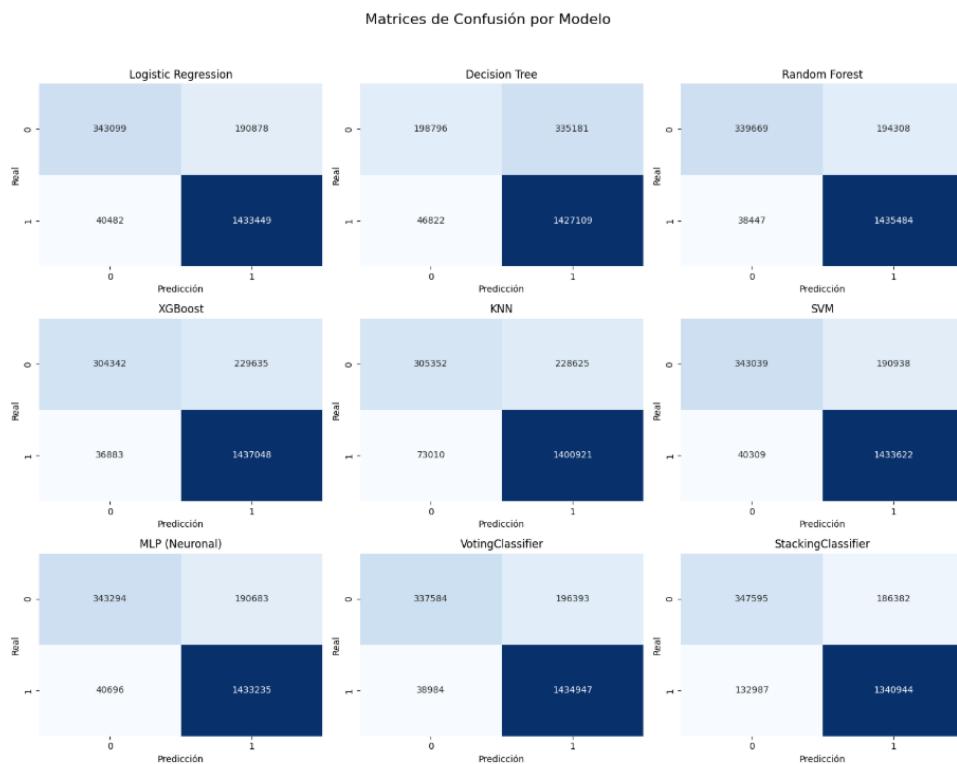
Modelo	Verdaderos Negativos (TN)	Especificidad (TNR)
StackingClassifier	347 595	0.6510
MLP	343 294	0.6429
Logistic Regression	343 099	0.6425
SVM	343 039	0.6424
Random Forest	339 669	0.6361
VotingClassifier	337 584	0.6322
KNN	305 352	0.5718
XGBoost	304 342	0.5700
Decision Tree	198 796	0.3723

Fuente: Elaboración propia

Ilustración 48: Métricas – Todos los modelos



**Ilustración 49:** Matriz de confusión – Todos los modelos



**Fuente:** Elaboración propia

El modelo MLP - Red Neuronal se selecciona como la opción óptima debido a que ofrece el mejor equilibrio entre sensibilidad y especificidad, manteniendo, al mismo tiempo, métricas globales sobresalientes como accuracy, precision, F1-score y AUC. Esta elección no se fundamenta en un único indicador, sino en un análisis integral que combina curvas ROC, matrices de confusión y tablas comparativas, evidenciando cómo el MLP logra capturar de forma equilibrada tanto positivos como negativos en un escenario donde la clase positiva es mayoritaria. En cuanto a su capacidad de discriminación, la curva ROC y el AUC son métricas determinantes.

El MLP alcanza un AUC de 0.8540, superior al obtenido por Decision Tree con valor de 0.7418 y SVM con 0.8179, y muy cercano al VotingClassifier con un valor de 0.8541. Más allá del valor absoluto, la forma de la curva refleja un mejor desempeño en la zona media de la tasa de falsos positivos, lo que implica que, incluso con niveles moderados de falsos positivos, mantiene una alta tasa de verdaderos positivos.

En términos de métricas globales, el MLP presenta resultados consistentes y de alto rendimiento: accuracy de 0.8848, precision de 0.8841, recall de 0.9724 y F1-score de 0.9253. Destaca especialmente su combinación de un recall muy elevado, que asegura la detección de la mayoría de los casos positivos, junto con una precision competitiva. Esto demuestra un control eficaz de los falsos positivos, superando a modelos como XGBoost, que obtiene una precision de 0.8622 y que, pese a un recall similar, genera más errores de este tipo.

Además, la elección del MLP se alinea con el principio de parsimonia o Navaja de Occam, el cual establece que, ante modelos con un rendimiento comparable, debe preferirse aquel que presente menor complejidad, ya que es menos probable que se ajuste al ruido de los datos y más probable que generalice adecuadamente a nuevos escenarios (Domingos, 1999). En este estudio, aunque el VotingClassifier presenta un AUC ligeramente superior, el MLP iguala o supera su desempeño en métricas clave como la especificidad y lo hace con una estructura más simple, reduciendo riesgos de sobreajuste.

## 7.2 Prueba de cambio de semilla

Con el objetivo de evaluar la estabilidad del modelo seleccionado frente a variaciones en la inicialización aleatoria, se comparó el rendimiento del estimador óptimo obtenido en el GridSearch con semilla 12345 con el de dos modelos entrenados utilizando dos semillas diferentes que son 33215 y 54321. Para cada configuración se calcularon las métricas F1-score y AUC sobre el conjunto de prueba. Los resultados evidencian que el modelo base alcanzó un F1-score de 0.9253 y un AUC de 0.8540, mientras que las variantes con semillas alternativas obtuvieron valores muy similares:

La semilla 33215: F1 = 0.9255, AUC = 0.8536 y la semilla 54321: F1 = 0.9249, AUC = 0.8543.

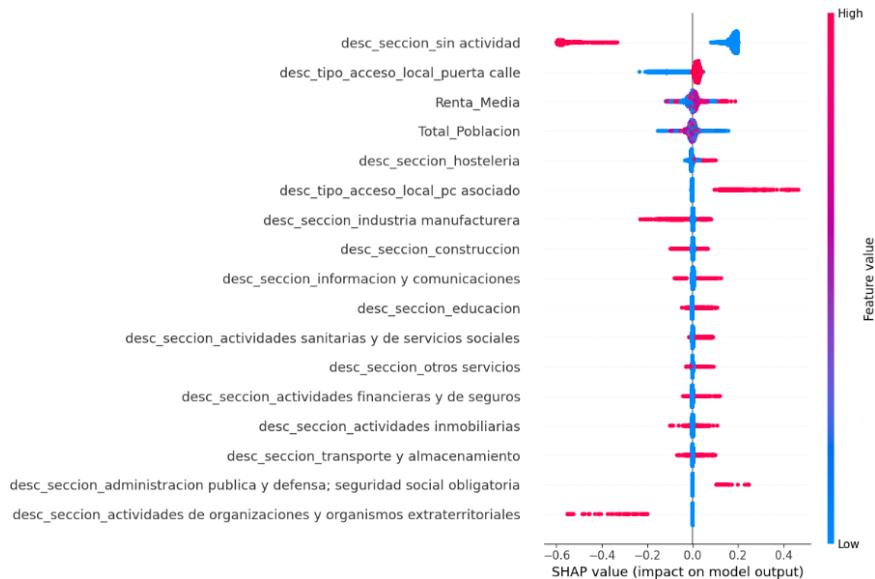
Esta consistencia en las métricas confirma que el rendimiento del MLP es el adecuado con la semilla base y no hay mucha variación en las métricas.

## 7.3 Análisis del modelo ganador

Se realizó un análisis del modelo ganador, una red neuronal MLP Classifier, con el objetivo de interpretar sus variables mediante los métodos SHAP y LIME. A través de estas técnicas, se identificaron las variables que influyen en el modelo, determinando cuáles son más importantes y cuáles tienden a clasificar hacia la clase 0 o la clase 1. Este análisis

resulta fundamental para explicar, en un contexto real, las razones y factores que afectan las predicciones del modelo. El gráfico de SHAP ofrece una visualización detallada del impacto relativo de cada variable predictora en la clasificación, permitiendo determinar si un local comercial está activo o inactivo. Esta metodología brinda una interpretación precisa de cómo cada característica contribuye a las predicciones, facilitando la identificación no solo de la importancia relativa de las variables, sino también de la dirección y magnitud de su influencia.

**Ilustración 50:** Sharp Summary Plot



**Fuente:** Elaboración propia

En el estudio sobre la actividad económica de la ciudad de Madrid, la variable que ejerce mayor influencia es `desc_seccion_sin_actividad`. Como es lógico, esta variable indica que, si un local no tiene actividad registrada, se predice que no está activo. La segunda variable más relevante es `desc_tipo_acceso_local_puerta calle`. Los locales con acceso directo desde la calle impulsan las predicciones hacia el lado positivo, lo que sugiere que la visibilidad y accesibilidad son factores clave para el dinamismo económico en Madrid. En cambio, cuando no existe este tipo de acceso el impacto suele ser negativo. En tercer lugar, `Renta_Media` muestra un patrón claro: las zonas con rentas medias más elevadas contribuyen de forma positiva al resultado, mientras que rentas bajas tienen un efecto generalmente negativo. Le sigue `Total_Poblacion`, donde las áreas más pobladas tienden a tener un impacto positivo, aunque de forma algo más moderada que la renta. Luego, `desc_seccion_hosteleria` presenta una relación positiva: la presencia significativa de locales de hostelería se asocia con un mayor impulso en la probabilidad que esos locales estén activos.

En el grupo intermedio de relevancia aparecen variables como `desc_seccion_industria manufacturera`, `desc_seccion_construcion`, `desc_seccion_informacion y comunicaciones` y `desc_seccion_educacion`, cuyos valores altos tienen impactos variados: en algunos casos favorecen el resultado, mientras que en otros lo reducen, lo que refleja una relación más dependiente de otros factores.

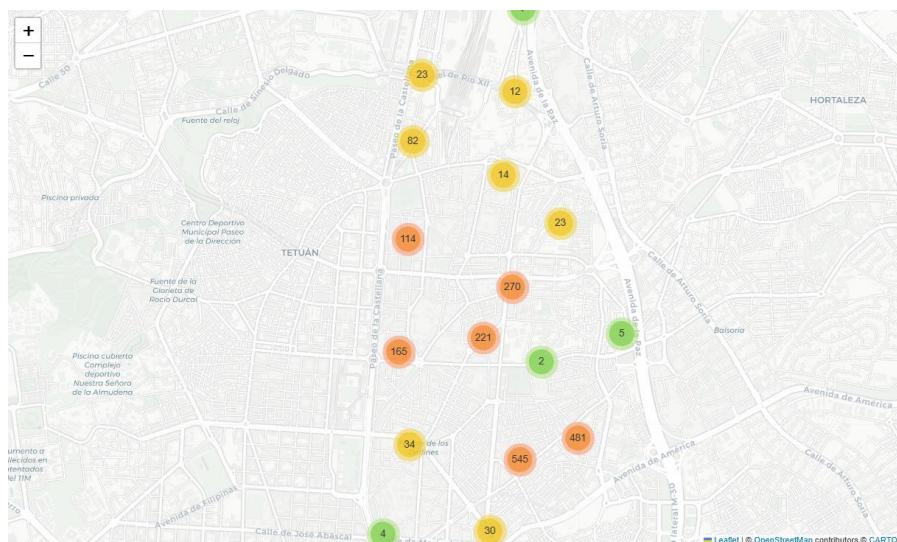
Finalmente, variables como desc\_seccion\_actividades sanitarias y de servicios sociales, desc\_seccion\_otros servicios, desc\_seccion\_actividades financieras y de seguros, desc\_seccion\_actividades inmobiliarias, desc\_seccion\_transporte y almacenamiento, desc\_seccion\_administracion publica y defensa; seguridad social obligatoria y desc\_seccion\_actividades de organizaciones y organismos extraterritoriales presentan impactos reducidos en el modelo.

**Tabla 26:** Top barrios con más actividad en comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas en Chamartín

desc_distrito_local	desc_barrio_local	cantidad
CHAMARTIN	prosperidad	609
CHAMARTIN	hispanoamerica	482
CHAMARTIN	ciudad jardin	344
CHAMARTIN	el viso	251
CHAMARTIN	nueva españa	219
CHAMARTIN	castilla	127

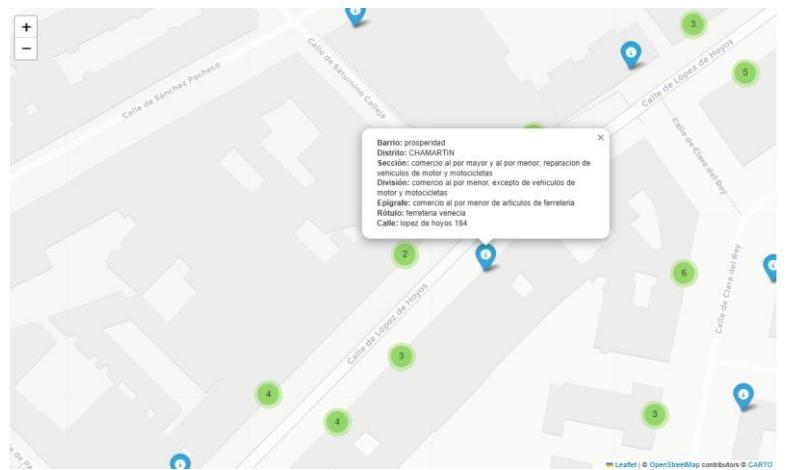
**Fuente:** Elaboración propia

**Ilustración 51:** Mapa 1 de locales de comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas la sección en Chamartín



**Fuente:** Elaboración propia

**Ilustración 52:** Mapa 2 de locales de comercio al por mayor y al por menor; reparacion de vehiculos de motor y motocicletas la sección en el barrio de Prosperidad, Chamartín



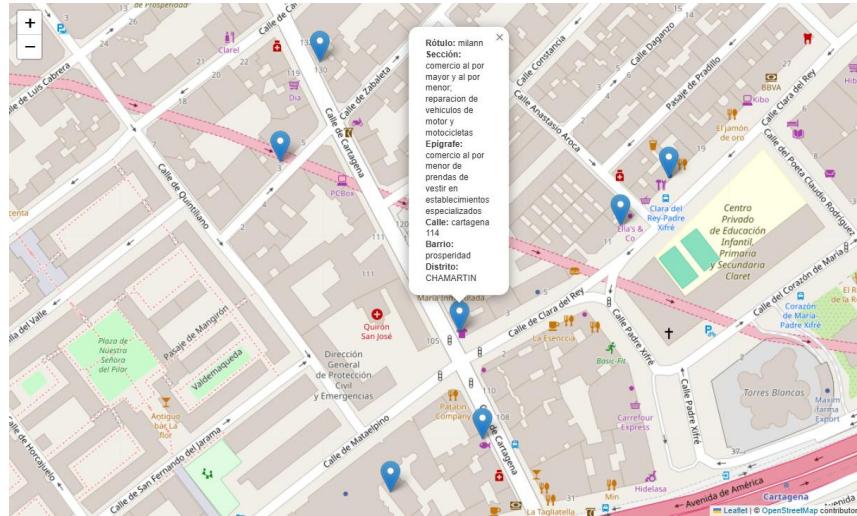
**Fuente:** Elaboración propia

**Tabla 27:** Epígrafe más representado de la sección comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas en el barrio de prosperidad

Distrito	Barrio	Sección	Epígrafe	Conteo
CHAMARTIN	Prosperidad	Comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas	Comercio al por menor de prendas de vestir en establecimientos especializados	48
CHAMARTIN	Prosperidad	Comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas	Comercio al por menor en establecimientos no especializados, con predominio en productos alimenticios, bebidas y tabaco (autoservicio)	36
CHAMARTIN	Prosperidad	Comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas	Comercio al por menor de frutas y hortalizas sin obrador	30
CHAMARTIN	Prosperidad	Comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas	Taller de reparación de automóviles especializado en mecánica y electricidad	25
CHAMARTIN	Prosperidad	Comercio al por mayor y al por menor; reparación de vehículos de motor y motocicletas	Farmacia	21

**Fuente:** Elaboración propia

**Ilustración 53:** Epígrafe comercio al por menor de prendas de vestir en establecimientos especializados en el barrio de Prosperidad, Chamartín



**Fuente:** Elaboración propia

**Tabla 28:** Locales sección manufacturera interior y agrupado

<u>desc_distrito_local</u>	<u>desc_barrio_local</u>	<u>conteo</u>
SALAMANCA	recoletos	36
RETIRO	los jeronimos	8
BARAJAS	aeropuerto	6
MONCLOA-ARAVACA	el plantio	6
CHAMARTIN	castilla	5
CHAMARTIN	ciudad jardin	4
CHAMARTIN	el viso	2
ARGANZUELA	atocha	1

**Fuente:** Elaboración propia

**Ilustración 54:** Mapa de locales inactivos de la sección manufactura industrial locales interior y agrupado



**Fuente:** Elaboración propia

## Capítulo 8. Conclusiones, recomendaciones y líneas futuras de investigación

### 8.1 Conclusiones principales

- El análisis de más de 8 millones de registros entre 2020 y 2024 ha demostrado que los modelos de Machine Learning pueden anticipar con notable precisión los patrones de actividad e inactividad comercial en la ciudad de Madrid. Asimismo, el muestreo de estos datos resulta una estrategia eficaz para optimizar la aplicación de estas técnicas. Con métricas de rendimiento cercanas al 90 % de exactitud y superiores al 92 % en F1-score, se confirma que estos algoritmos no solo son aplicables en entornos empresariales, sino que también aportan un valor significativo a la administración pública y a la ciudadanía, al transformar información administrativa, como es un censo, en conocimiento estratégico para la planificación urbana y económica.
- La renta media de los barrios, el tipo de acceso al local y la pertenencia a secciones clasificadas como el tipo de sección, renta, población, se consolidan como factores claves para explicar la permanencia o cierre de los negocios. Esta conclusión es especialmente relevante para la política pública, ya que demuestra que la supervivencia comercial no depende solo de factores individuales del negocio, sino del ecosistema urbano y económico en el que se ubica.
- Este trabajo ha demostrado que la combinación de una adecuada preparación de los datos, la selección precisa de variables y la aplicación de algoritmos de Machine Learning permite no solo anticipar con un alto grado de certeza qué locales tienden a cerrar, sino también comprender mejor las dinámicas urbanas que condicionan esa decisión. La comparación entre diferentes modelos puso en evidencia que alternativas relativamente simples, como el MLP Classifier con un número reducido de nodos y variables de entrada, pueden ofrecer resultados más sólidos y estables que configuraciones excesivamente complejas, como ensamblados avanzados, cuya mejora en error de validación no justifica el incremento en complejidad. Este hallazgo subraya la importancia de priorizar la simplicidad y la interpretabilidad en el diseño de modelos predictivos, ya que no solo reduce el riesgo de sobreajuste, sino que también favorece la robustez y la replicabilidad de los resultados. Finalmente, la búsqueda de configuraciones de hiperparámetros no debe centrarse en alcanzar un valor óptimo puntual, sino en identificar entornos estables que garanticen consistencia y generalización del modelo en distintos escenarios.
- Durante el desarrollo de este trabajo se ha constatado la presencia de diversos problemas de calidad en los datos, como la ausencia de valores, diferencias en la escritura y el orden de las variables a lo largo de los meses, así como registros inconsistentes. Entre estos últimos destacan, por ejemplo, nombres de barrios o distritos escritos de forma distinta, errores en el uso de tildes, o epígrafes y rótulos mal redactados. Estas inconsistencias hacen imprescindible realizar una limpieza y una exploración exhaustiva del conjunto de datos antes de seleccionar variables o aplicar métodos de machine learning, ya que, si no se corrigen, pueden obtenerse

resultados erróneos o sesgados. Por ello, se recomienda estandarizar nombres y formatos, imputar o eliminar valores faltantes siguiendo criterios justificados, y documentar cada paso del preprocesado para garantizar la reproducibilidad y la fiabilidad de los modelos.

## 8.2 Recomendaciones

- La mejor forma de aprovechar el valor de los modelos predictivos desarrollados es desplegarlos mediante una API accesible para las diferentes áreas municipales. De este modo, cualquier departamento de la administración pública como es la de urbanismo, licencias, comercio e inspección, podrá consultar de manera inmediata el riesgo de inactividad de un local, integrando la predicción en sus flujos de trabajo cotidianos. A su vez, la publicación de parte de estos datos en un portal de Open Data fomentaría la innovación abierta, facilitando la colaboración con universidades, startups y ciudadanos interesados en contribuir al desarrollo urbano.
- No solo basta con predecir o aplicar un modelo de Machine Learning sino también es imprescindible que las personas que analicen los datos sean en entes gubernamentales o un ciudadano comprendan qué variables han influido en cada predicción. La integración de herramientas de explicabilidad como SHAP o LIME permitiría ofrecer, junto al resultado del modelo, un desglose de los factores más determinantes. Esto aumentaría la confianza en el sistema, facilitaría la trazabilidad de las decisiones y reforzaría la transparencia hacia la ciudadanía.
- Las predicciones no deben limitarse a un uso técnico, sino convertirse en insumos para el diseño de políticas. Las áreas con mayor probabilidad de cierres deberían priorizarse en programas de dinamización comercial, ayudas económicas o incentivos fiscales, mientras que aquellas con menor riesgo pueden recibir recursos destinados a consolidación y diversificación. Así, el modelo se transforma en una herramienta de planificación territorial basada en evidencia.
- Se recomienda institucionalizar un marco de gobernanza que asegure la estandarización de diccionarios de variables, el registro obligatorio de metadatos como la fecha de reporte, la validación de datos y la normalización de textos. Estas medidas no solo mejorarán el desempeño de los modelos actuales, sino que sentarán las bases para futuros proyectos de Ciencia de Datos en el Ayuntamiento y la Comunidad de Madrid, así como integrar a la base de datos de los diferentes entes gubernamentales para realizar análisis cruzado

## Bibliografia:

- Ahamed, S. F., Vijayasankar, A., Thenmozhi, M., Rajendar, S., Bindu, P., & Subha Mastan Rao, T. (2023). Machine learning models for forecasting and estimation of business operations. *The Journal of High Technology Management Research*, 34(1), 100455. <https://doi.org/10.1016/j.hitech.2023.100455>
- Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23(4), 589–609. <https://doi.org/10.2307/2978933>
- Asociación Nacional de Grandes Empresas de Distribución (ANGED). (2024). *Informe económico 2024*. <https://anged.es/el-sector/informe-anual-anged/>
- Ayuntamiento de Madrid. (2025). *Informe mensual de la coyuntura económica: Mayo 2025* [Informe]. Dirección General de Economía. [https://www.madrid.es/UnidadesDescentralizadas/UDCObservEconomico/002%20Indicadores%20Economicos/Ficheros/INFORME\\_SITUACION\\_COYUNTURA\\_ECONOMIA\\_MADRID\\_CIUDAD\\_2025\\_05\\_09.pdf](https://www.madrid.es/UnidadesDescentralizadas/UDCObservEconomico/002%20Indicadores%20Economicos/Ficheros/INFORME_SITUACION_COYUNTURA_ECONOMIA_MADRID_CIUDAD_2025_05_09.pdf)
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Comunidad de Madrid. (2025, abril 5). *El comercio de Madrid en cifras*. Dirección General de Comercio, Consumo y Servicios. [https://www.comunidad.madrid/sites/default/files/el\\_comercio\\_de\\_madrid\\_en\\_cifras\\_5\\_abril\\_2025.pdf](https://www.comunidad.madrid/sites/default/files/el_comercio_de_madrid_en_cifras_5_abril_2025.pdf)
- Cortes, C., & Vapnik, V. (1995). Support vector networks. *Machine Learning*, 20(3), 273–297. <https://doi.org/10.1007/BF00994018>
- Cox, D. R. (1972). Regression models and life tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2), 187–220. <https://www.jstor.org/stable/2985181>
- Domingos, P. (1999). The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery*, 3(4), 409–425. <https://doi.org/10.1023/A:1009868929893>
- Escudero-Gómez, L. A. (2024). Shopping centers challenging decline: Competitive strategies in three case studies from Madrid's urban area. *Journal of Retailing and Consumer Services*, 79, 103826. <https://doi.org/10.1016/j.jretconser.2024.103826>
- Firas, O. (2023). A combination of SEMMA & CRISP-DM models for effectively handling big data using formal concept analysis based knowledge discovery: A data mining approach. *World Journal of Advanced Engineering Technology and Sciences*, 8(1), 009–014. <https://doi.org/10.30574/wjaets.2023.8.1.0147>
- Kupfer, A.-K., Marchand, A., & Hennig-Thurau, T. (2024). Explaining physical retail store closures in digital times. *Journal of Retailing*, 100(4), 512–531. <https://doi.org/10.1016/j.jretai.2024.07.001>
- Lahmiri, S., & Bekiros, S. (2024). Can machine learning approaches predict corporate bankruptcy? Evidence from a qualitative experimental design. *ESCA School of*

*Management.*

[https://cadmus.eui.eu/bitstream/handle/1814/66068/Machine\\_learning.pdf?sequence=1&isAllowed=y](https://cadmus.eui.eu/bitstream/handle/1814/66068/Machine_learning.pdf?sequence=1&isAllowed=y)

Park, Y., Kim, D., Jeon, J., & Kim, K. (2024). Predictors of medical and dental clinic closure by machine learning methods: Cross-sectional study using empirical data. *Journal of Medical Internet Research*, 26, e46608. <https://doi.org/10.2196/46608>

Langbein, S. H., Krzyziński, M., Spytek, M., Baniecki, H., Biecek, P., & Wright, M. N. (2024). Interpretable machine learning for survival analysis (No. arXiv:2403.10250). *arXiv*. <https://doi.org/10.48550/arXiv.2403.10250>

Lee, S., Ko, S., Roudsari, A. H., & Lee, W. (2024). A deep learning model for predicting the number of stores and average sales in commercial district. *Data & Knowledge Engineering*, 150, 102277. <https://doi.org/10.1016/j.datak.2024.102277>

Li, X., Li, Z., Sun, Y., & Wang, F. (2022). A joint learning framework for restaurant survival prediction and explanation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)* (pp. 3456–3466). ACM. <https://doi.org/10.1145/3534678.3539307>

Lu, J., Zheng, X., Nervino, E., Li, Y., Xu, Z., & Xu, Y. (2024). Retail store location screening: A machine learning-based approach. *Journal of Retailing and Consumer Services*, 77, 103620. <https://doi.org/10.1016/j.jretconser.2023.103620>

Ministerio de Agricultura, Pesca y Alimentación. (2024). *Efecto de la guerra de Ucrania en el comercio exterior agroalimentario y pesquero español (marzo 2022-febrero 2024)*. Subdirección General de Análisis, Coordinación y Estadística. <https://www.mapa.gob.es/es/>

Mogensen, U. B., Ishwaran, H., & Gerds, T. A. (2012). Evaluating random forests for survival analysis using prediction error curves. *Journal of Statistical Software*, 50(11), 1–23. <https://doi.org/10.18637/jss.v050.i11>

Odom, M., & Sharda, R. (1990). A neural network model for bankruptcy prediction. In *Proceedings of the IEEE International Joint Conference on Neural Networks* (Vol. 2, pp. 163–168). IEEE. <https://doi.org/10.1109/IJCNN.1990.137710>

Romero Martínez, J. M., García García, J. A., & Cárdenas Montoya, A. (2021). La utilidad del deep learning en la predicción del fracaso empresarial. *Revista de Métodos Cuantitativos para la Economía y la Empresa*, 32, 250–270. <https://doi.org/10.46661/revmetodoscuanteconempresa.6274>

Shetty, S., Musa, M., & Brédart, X. (2022). Bankruptcy prediction using machine learning techniques. *Journal of Risk and Financial Management*, 15(1), 35. <https://doi.org/10.3390/jrfm15010035>

Smith, M., & Alvarez, F. (2022). Predicting firm-level bankruptcy in the Spanish economy using extreme gradient boosting. *Computational Economics*, 59(1), 263–295. <https://doi.org/10.1007/s10614-020-10078-2>

Sobreiro, V. A., Kimura, H., Edina, S., Mello, J. C. C. B. S., Oliveira, B. C., & Mariani, V. C. (2022). Hybrid random forest survival model to predict customer lifetime value in the retail sector. *Electronics*, 11(20), 3328. <https://doi.org/10.3390/electronics11203328>

Stempień, D., & Ślepaczuk, R. (2024). Hybrid models for financial forecasting: Combining econometric, machine learning, and deep learning models. *arXiv preprint arXiv:2505.19617*. <https://arxiv.org/abs/2505.19617>

Uribe-Toril, J., Ruiz-Real, J. L., Galindo Durán, A. C., Torres Arriaza, J. A., & de Pablo Valenciano, J. (2022). The circular economy and retail: Using deep learning to predict business survival. *Environmental Sciences Europe*, 34(1), 2. <https://doi.org/10.1186/s12302-021-00582-z>

Vallapuram, A. K., Nanda, N., Kwon, Y. D., & Hui, P. (2022). Interpretable business survival prediction. In *Proceedings of the 2021 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 99–106). IEEE. <https://doi.org/10.1145/3487351.3488353>

Xie, Z., Hao, J., Zhang, Y., Liu, W., Chen, L., & Wang, M. (2023). The prediction of in-hospital mortality in chronic kidney disease patients with coronary artery disease using machine learning models. *Frontiers in Public Health*, 11, 1118438. <https://doi.org/10.3389/fpubh.2023.1118438>

Zheng, Y., Hao, Q., Wang, J., Gao, C., Chen, J., Jin, D., & Li, Y. (2024). A survey of machine learning for urban decision making: Applications in planning, transportation, and healthcare. *ACM Computing Surveys*, 57(4), 99:1–99:41. <https://doi.org/10.1145/3695986>