uc3m | Universidad Carlos III de Madrid

Grado en Estadística y Empresa
2023-2024

Trabajo Fin de Grado

"Análisis y predicción de los precios de las viviendas en la ciudad de Madrid"

Andrés Rubio Lafuente

Tutor

Daniel Ruiz Nodar

Getafe, Julio 2024

RESUMEN

El presente Trabajo de Fin de Grado es una investigación que tiene por objetivo analizar y predecir con la mayor exactitud posible el precio de las viviendas en la ciudad de Madrid. El punto de partida del trabajo comienza con la extracción de los datos, que se realiza mediante una tarea de "web scraping" sobre un portal de viviendas online. En esta tarea de extracción, se han obtenido las características de las viviendas de todos los distritos de la ciudad de Madrid.

Para alcanzar dicho objetivo, lo primero es realizar una tarea de extracción, transformación y carga de los datos o ETL. Se realiza una transformación de las características creando variables nuevas y se eliminan aquellas que no proporcionan información útil para la construcción de los modelos. A continuación, se realiza un análisis exploratorio de los datos o EDA, para conocer en mayor profundidad las variables con las que se va a trabajar. Una vez acabado el análisis exploratorio, se realiza un preprocesamiento de los datos para poder aplicar las técnicas de aprendizaje automático, tanto de aprendizaje supervisado como no supervisado.

Cuando se da por terminado el preproceso, se procede a realizar un análisis exhaustivo del conjunto de datos mediante el ajuste de diversos modelos de aprendizaje automático. En nuestro caso, tenemos que abordar un problema de regresión pues la variable respuesta es el precio de la vivienda, que es una variable numérica continua, y la métrica de evaluación que se utiliza es el R^2. Entre las técnicas de aprendizaje supervisado que se utilizan para resolver este problema de regresión, podemos destacar la calidad de los resultados del Random Forest. También se aplican técnicas de aprendizaje no supervisado con el objetivo de buscar una nueva agrupación de las viviendas mediante los algoritmos de clustering jerárquico y no jerárquico.

Finalmente, se exponen los modelos que logran obtener las mejores métricas de evaluación, se extraen los conocimientos más relevantes de los resultados y se elaboran las conclusiones.

Palabras Clave

Aprendizaje Automático, Aprendizaje Supervisado, Aprendizaje No Supervisado, Predicciones, ETL, EDA, Preproceso, Random Forest, Clustering.

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

1. INTRODUCCIÓN, DATOS Y OBJETIVOS

1.1. Introducción

Antes de comenzar con el trabajo, se quiere situar al lector en el contexto adecuado para que resulte más sencillo entender el interés que hay en la realización de este estudio. En los últimos años ha habido 2 catástrofes que han provocado que muchos países, entre ellos España, entren en una etapa de recesión. Las 2 catástrofes mencionadas son la pandemia provocada por el coronavirus SARS-CoV-2 o COVID-19, que irrumpió a finales de 2019, y la guerra entre Rusia y Ucrania, que estalló a principios de 2022.

En España, la pandemia del COVID-19 paralizó todas las operaciones que se estaban realizando hasta ese momento y obligó a establecer un periodo de cuarentena. En este periodo de cuarentena, muchos ciudadanos españoles no pudieron continuar con sus actividades cotidianas como ir a la oficina o a la escuela. La inactividad afectó a numerosos sectores, entre ellos la construcción y la hostelería y restauración, donde muchos negocios se vieron obligados a cerrar. Por otro lado, la guerra entre Rusia y Ucrania ha afectado de manera indirecta a muchos países que no participaban en la guerra, pero que han sufrido las consecuencias de la contienda.

Debido a la importancia de Rusia y Ucrania como exportadores de productos alimenticios, entre ellos el trigo, la guerra contribuyó a que el precio de los granos alcanzara niveles máximos históricos. La guerra desencadenó la peor crisis energética global que ha habido desde la década de 1970. Los precios de los energéticos aumentaron en muchas partes del mundo al tiempo que los países disminuían o suspendían la compra de combustibles fósiles procedentes de Rusia (Mpoke Bigg, 2023).

En este periodo de crisis se han visto consecuencias económicas muy negativas en todos los sectores. De entre todos los sectores de la economía, este trabajo se centra en el sector inmobiliario, en especial, en las viviendas de la capital del territorio español. A forma de mostrar las consecuencias de estos eventos en el sector inmobiliario, se ha obtenido información del Instituto Nacional de Estadística o INE.

Como se observa en la figura 1, el Índice de Precios de Consumo (IPC) ha experimentado un aumento notable a nivel nacional. La figura en cuestión muestra el IPC en relación con la vivienda y otros factores asociados, como la electricidad, el agua, el gas y otros combustibles. Es evidente que el incremento del IPC se intensificó durante las fases iniciales de la pandemia del COVID-19 y de la guerra entre Rusia y Ucrania. Sin embargo, desde finales de 2022 se aprecia una disminución considerable en el IPC, el cual se ha mantenido relativamente estable desde comienzos de 2023.

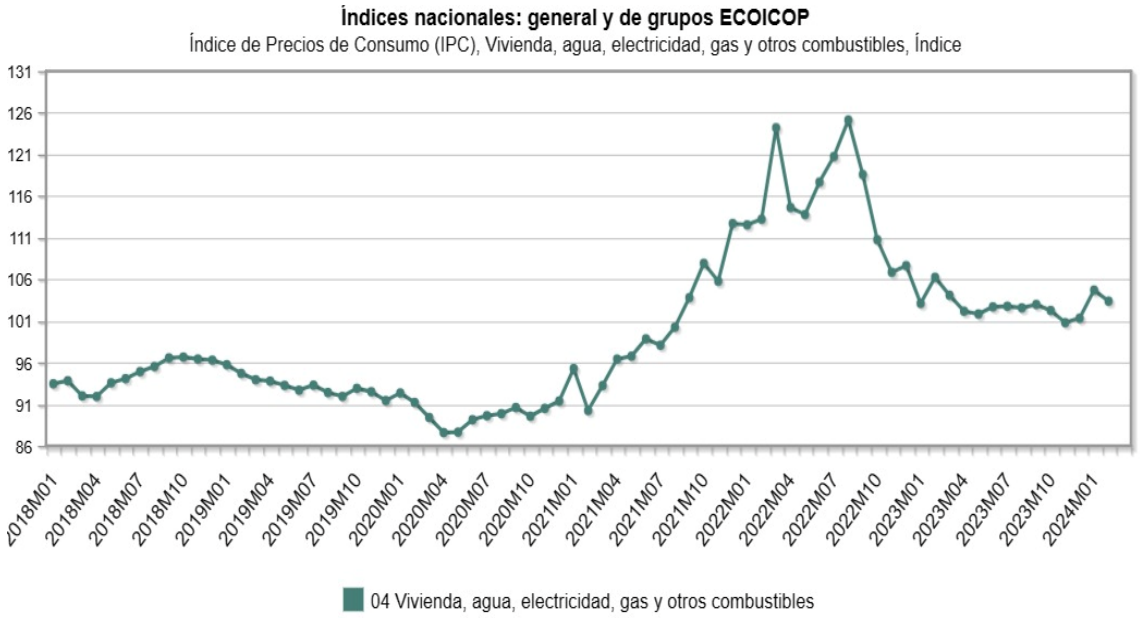

Figura 1 Evolución del IPC a nivel nacional: vivienda, agua, electricidad, gas y otros combustibles

Fuente: INE (2024)

En la figura 2 se puede observar la evolución del Índice de Precios de la Vivienda (IPV) en la comunidad de Madrid. Desde 2018, el IPV ha mostrado una tendencia al alza constante, sufriendo muy pocas disminuciones a lo largo de su trayectoria hasta la actualidad. Aunque es cierto que desde comienzos de 2019 hasta finales de 2020 el crecimiento del IPV ha sido más moderado, en los últimos años la pendiente ha aumentado a un ritmo mayor.

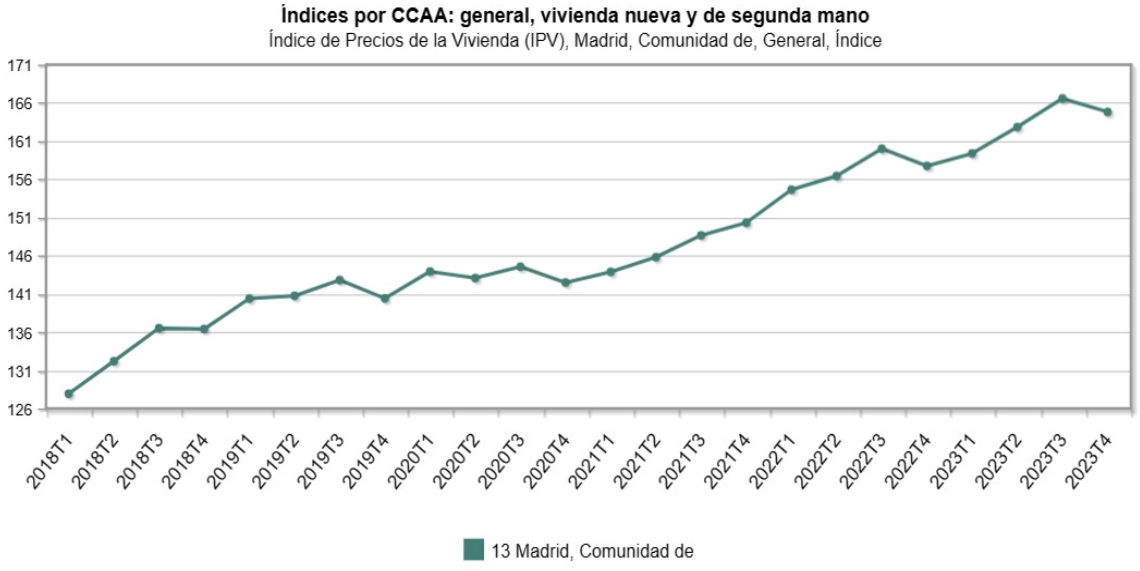

Figura 2 Evolución del IPV en la Comunidad de Madrid

Fuente: INE (2024)

En la figura 3 vemos el crecimiento del PIB del sector inmobiliario a nivel nacional. Aunque, con la pandemia del COVID-19 el PIB disminuyó significativamente, en la actualidad ha logrado recuperarse y se mantiene por encima del nivel previo a la pandemia.

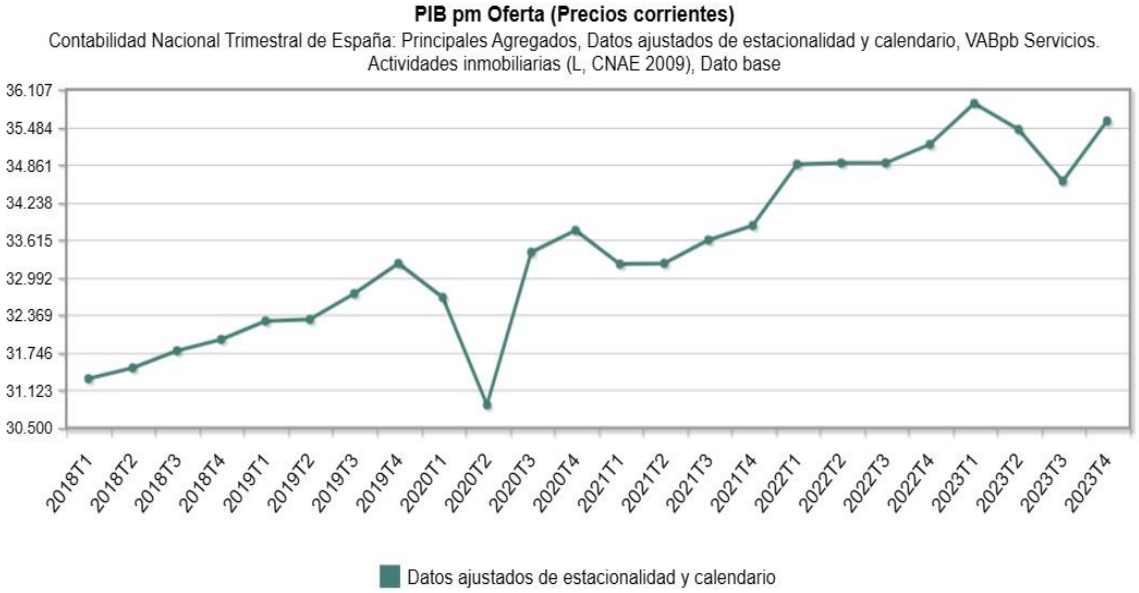

Figura 3 Evolución del PIB de actividades inmobiliarias a nivel nacional

Fuente: INE (2024)

En la figura 4 vemos la trayectoria de los tipos de interés de las viviendas a nivel nacional. A pesar de que los tipos de interés, tanto fijo como variable, se habían mantenido y en algunos casos disminuido hasta finales de 2019, estos están sufriendo un crecimiento importante en los 2 últimos años. Respecto al nivel previo a la pandemia, ha habido un incremento notable de los tipos de interés. En particular, el aumento ha sido de 0,6 y 0,8 puntos porcentuales para el interés fijo y variable, respectivamente.

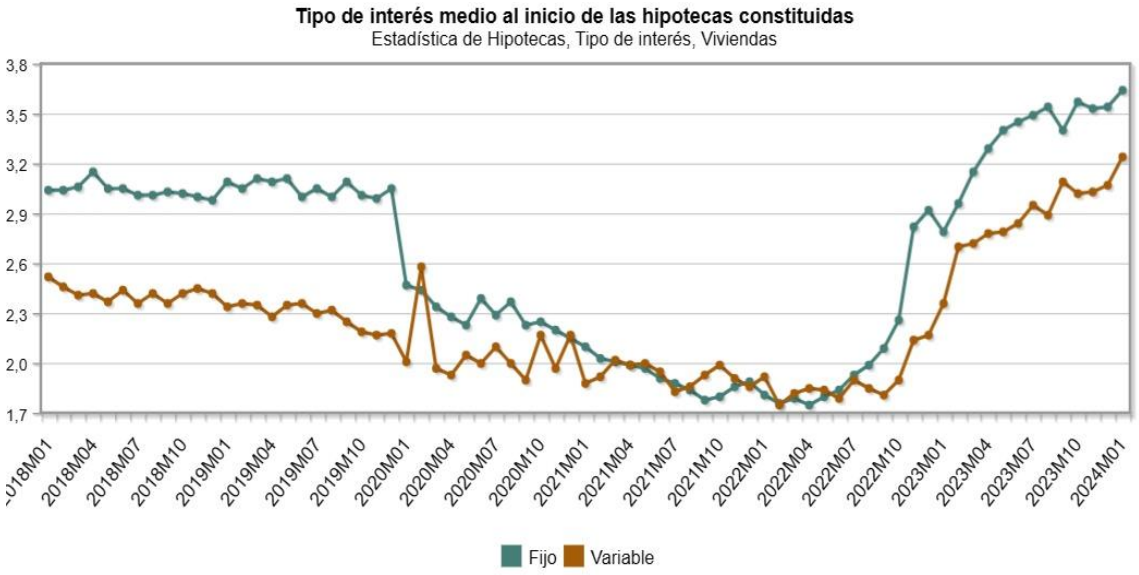

Figura 4 Evolución de los tipos de interés en las hipotecas de las viviendas a nivel nacional

Fuente: INE (2024)

Todos estos gráficos sugieren que estamos viviendo una etapa de crisis en la que los precios de las viviendas han aumentado considerablemente. Y no solo eso, el aumento de los tipos de interés dificulta aún más la situación del sector inmobiliario, pues ahora resulta mucho más complicado que el ciudadano promedio consiga pagar su hipoteca al banco.

Teniendo en cuenta todos estos factores, con la inflación de los precios de las viviendas uno podría estar interesado en entender cuáles son los factores más importantes que intervienen a la hora de determinar el precio de una vivienda. Y teniendo los datos necesarios, podríamos estar interesados en desarrollar un modelo que nos permita predecir el precio de una vivienda aleatoria en base a una serie de reglas. Pues bien, eso es precisamente lo que se pretende lograr con este trabajo de fin de grado. Un trabajo en el que se realiza un estudio que, cual camaleón, se podría adaptar a otras características y a otras zonas geográficas, y con el cual se podría obtener unos resultados igualmente competentes.

Es un trabajo que despierta el interés del autor pues busca encontrar una manera de alcanzar su objetivo de forma que se pueda aplicar a un caso real. Es por eso que se decide realizar una tarea de "web scraping" para extraer información sobre las viviendas de un área real, como es la ciudad de Madrid. Este trabajo resulta interesante pues tiene una aplicación real y lo podría realizar una empresa inmobiliaria que entre nueva en el mercado y quiera desarrollar un modelo de predicción de los precios de las viviendas. Le podría interesar las reglas de decisión que utilizan los competidores para determinar el precio de una vivienda y cuáles son las características que consideran más importantes. Por lo cual, no sólo es un trabajo académico, sino que se podría implementar en el mundo laboral si así se desease. Además, este trabajo académico parece bastante popular en la comunidad científica, pues encontramos numerosos estudios que abordan el mismo tema, como se cita a continuación.

El aprendizaje automático está creciendo más rápidamente en esta década. Muchas aplicaciones y algoritmos evolucionan en el aprendizaje automático día a día. Una de esas aplicaciones que se encuentra en las revistas es la predicción del precio de las casas. Los precios de la vivienda están aumentando cada año, lo que ha necesitado el modelado de la predicción del precio de las casas. Estos modelos construidos, ayudan a los clientes a comprar una casa adecuada para sus necesidades (Thamarai & Malarvizhi, 2020).

¿Cómo utilizar algoritmos de aprendizaje automático para predecir el precio de las casas? Es un desafío para obtener el resultado más cercano posible basado en el modelo construido. Para una casa específica, el precio se determina por la ubicación, el tamaño, el tipo de casa, la ciudad, el país, las reglas de impuestos, el ciclo económico, el movimiento de la población, la tasa de interés y muchos otros factores que podrían afectar la demanda y la oferta (Lu et al., 2017).

1.2. Datos y Objetivos

El objetivo principal de este trabajo es lograr desarrollar un modelo de machine learning que, en base a una serie de características, logre predecir el precio de una vivienda determinada en la ciudad de Madrid. Para alcanzar dicho objetivo, se utilizan una serie de herramientas y técnicas estadísticas centradas en el aprendizaje supervisado y no supervisado, en las cuales profundizaremos más adelante. La realización del trabajo se lleva a cabo en Google Collaborate utilizando el lenguaje de programación Python y se puede ver con mayor detalle en el anexo B.

La base de datos que se utiliza procede de una tarea de "web scraping" realizada sobre la página web de pisos.com. En la tarea de extracción de los datos, se ha realizado un análisis exhaustivo de cómo está estructurado el contenido html en la página web y de cómo acceder a los elementos relevantes. Véase el Anexo A si se quiere obtener más información sobre cómo se ha realizado la extracción y creación de la base de datos.

Sobre la obtención de los datos es importante resaltar que la tarea de web scraping puede resultar ilegal si se utiliza para fines ilícitos o malintencionados. En este caso, el objetivo de la extracción de los datos es tener un conjunto de datos actual y sobre una región real que se pueda utilizar para el estudio en cuestión. En ningún caso se utilizan estos datos para otro fin que no sea este trabajo académico. Aunque hay información personal disponible en la página web, como el teléfono móvil o el correo para contactar con la persona o entidad que oferta la vivienda, en ningún caso se utiliza esta información por respeto de la privacidad. En concreto, la información que ha sido extraída no tiene carácter personal y ha sido puesta en Internet a disposición de todos los usuarios, luego cualquiera puede acceder a ella. En principio se podría utilizar esta información sin ningún problema, teniendo en cuenta lo mencionado anteriormente.

Una vez explicado esto, podemos pasar a describir el conjunto de datos que se utiliza para el estudio. En la tabla 1 podemos ver la información de todas las características que se utilizan para el trabajo. A continuación, se verá la sección 2, donde se presenta la metodología empleada para el trabajo, la sección 3, donde se analizan los resultados obtenidos, y la sección 4, donde se exponen las principales conclusiones del trabajo.

Variable	**Tipo de Variable**	**Información sobre la Variable**
Precio	Numérica Continua	Contiene el precio de la vivienda y será nuestra variable respuesta
Superficie construida	Numérica Continua	Contiene el área o la superficie construida de la vivienda
Habitaciones	Numérica Discreta	Contiene el número de habitaciones que hay en la vivienda
Baños	Numérica Discreta	Contiene el número de baños que hay en la vivienda
Distrito	Categórica Multiclase	Contiene el distrito en el que se encuentra la vivienda

Tipo de vivienda	Categórica Multiclase	Contiene el tipo de vivienda que puede ser un piso, chalet, dúplex …
Piscina	Categórica Binaria	Toma valor 1 si la vivienda tiene piscina y 0 en caso contrario
Terraza	Categórica Binaria	Toma valor 1 si la vivienda tiene terraza y 0 en caso contrario
Jardín	Categórica Binaria	Toma valor 1 si la vivienda tiene jardín y 0 en caso contrario
Garaje	Categórica Binaria	Toma valor 1 si la vivienda tiene garaje y 0 en caso contrario
Trastero	Categórica Binaria	Toma valor 1 si la vivienda tiene trastero y 0 en caso contrario
Calefacción	Categórica Binaria	Toma valor 1 si la vivienda tiene calefacción y 0 en caso contrario
Aire acondicionado	Categórica Binaria	Toma valor 1 si la vivienda tiene aire acondicionado y 0 en caso contrario
Ascensor	Categórica Binaria	Toma valor 1 si la vivienda tiene ascensor y 0 en caso contrario

Tabla 1 Descripción del conjunto de variables

2. DESCRIPCIÓN DE LA METODOLOGÍA EMPLEADA

En esta sección del trabajo, se presenta la metodología de las técnicas de aprendizaje automático que se han empleado. El aprendizaje automático o "machine learning" es un campo de la inteligencia artificial que ha ganado mucha popularidad en los últimos años. Es por ello que este trabajo se centra en este área, en especial en 2 de sus tareas: aprendizaje supervisado y no supervisado. Para la descripción de las técnicas de aprendizaje automático, se ha utilizado los apuntes de 3 asignaturas del grado de Estadística y Empresa: análisis multivariante (Grané Chávez, notas de clase, 2023), aprendizaje automático para el análisis de datos (García Diéguez, notas de clase, 2023) y métodos estadísticos en minería de datos (Gonzalo de Alba, notas de clase, 2023). A continuación, se explican las técnicas de aprendizaje supervisado y no supervisado que se han empleado en el trabajo.

2.1. Aprendizaje Supervisado

En el aprendizaje supervisado nuestro objetivo es entrenar un modelo que, en base a una serie de predictores, nos ayude a entender el comportamiento de los datos para predecir la variable respuesta. En función de la variable respuesta, podemos dividir los problemas en 2 tipos: problemas de clasificación, en los cuales la variable respuesta es una variable categórica, y problemas de regresión, donde es una variable numérica continua. En nuestro caso, la variable respuesta o "target" es el precio de la vivienda, el cual toma valores numéricos en un espacio continuo, luego tenemos un problema de regresión.

Existen multitud de métodos que se pueden aplicar para resolver este problema, desde regresión lineal simple hasta regresión de Ridge o mejor selección de subconjuntos. Sin embargo, en este trabajo se van a probar otros métodos de aprendizaje automático en los que se profundizará más adelante. Estos son: el modelo de los k vecinos más cercanos (KNN), el árbol de decisión, las máquinas de vectores de soporte, el Gradient Boosting, el Random Forest y las redes neuronales.

2.1.1. KNN

El KNN o k vecinos más cercanos es un algoritmo de aprendizaje automático vago, pues realmente no se construye ningún modelo, sino que se clasifican las instancias en función del número de vecinos que tenga más cerca de un grupo o de otro, utilizando distancias. Típicamente se utiliza la distancia euclídea, cuya fórmula se rige por la expresión (2.1).

$$d(x_i, x_j)^2 = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + d(x_{ik} - x_{jk})^2 \qquad (2.1)$$

Donde:

$$\mathrm{i, j} = \text{Observaciones i} - \text{ésima, j} - \text{ésima del conjunto de datos}$$

$$\{1, 2, \ldots \mathrm{k}\} = \text{Número de variables o características}$$

Con el KNN, si por ejemplo definimos un valor para el número de vecinos k = 5, entonces el algoritmo busca los 5 vecinos más cercanos a la instancia a predecir. Como tenemos

un problema de regresión, para cada característica se toma el valor promedio ponderado de los 5 vecinos más cercanos a la instancia a predecir.

A pesar de que el KNN es un algoritmo muy sencillo y fácil de implementar, tiene algunas limitaciones. Es un método muy sensible a los datos con ruido y a las características irrelevantes, que pueden afectar de forma negativa al rendimiento del modelo. Para aplicar este algoritmo de aprendizaje, es necesario realizar un tratamiento previo de los datos. Se precisa normalizar las características numéricas para que las variables con un rango más amplio no tengan mayor influencia que las demás. Para la normalización de las variables utilizamos la estandarización, cuya fórmula viene definida en (2.2).

$$x'_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j} \qquad (2.2)$$

Donde:

$$x_{ij} = Valor\ de\ la\ característica\ j\ para\ la\ instancia\ i$$

$$\mu_j = Valor\ medio\ de\ la\ característica\ j$$

$$\sigma_j = Desviación\ típica\ de\ la\ característica\ j$$

Debemos tener en cuenta estos aspectos antes de aplicar el modelo de los k vecinos más cercanos, pues de otra forma se obtendrán resultados erróneos. Además, para mejorar el rendimiento del modelo se pueden ajustar una serie de hiperparámetros entre los que destacamos el número de vecinos y la medida de distancia utilizada.

2.1.2. Árbol de Decisión

Los modelos de árbol de decisión crean un árbol compuesto por nodos, de forma que en cada nodo se toma una decisión y se subdivide en los nodos siguientes. La decisión que tomamos en cada nodo se realiza sobre la variable más relevante y para determinarla podemos utilizar varias medidas. Las medidas típicas en problemas de clasificación son la entropía y el índice de Gini, pero en problemas de regresión buscamos la reducción de la varianza en vez de la reducción de la impureza. Para logar esto, utilizamos el MSE o "Mean Squared Error", en español "Error Cuadrático Medio", cuya fórmula se puede ver en la expresión (2.3).

$$MSE = \frac{1}{n}\sum_{I}(y_i - \hat{y}_i)^2 \qquad (2.3)$$

Donde:

$$y_i = \text{Valor real de la variable respuesta para la observación i} - \text{ésima}$$

$$\hat{y}_i = \text{Valor predicho de la variable respuesta para la observación i} - \text{ésima}$$

$$n = \text{Número total de observaciones}$$

Así pues, la variable del conjunto de datos que minimice el error cuadrático medio será la que se introduzca en el nodo del árbol y sobre la cual se tomará la decisión. Como muchos otros algoritmos de aprendizaje automático, el árbol de decisión se ve

influenciado por la presencia de datos atípicos o "outliers", con lo cual habrá que tratar con ellos antes de poder aplicar el modelo. Una de las ventajas del árbol de decisión es que es muy sencillo de interpretar y visualizar, pues podemos ver la importancia de las variables y las reglas de decisión que se toman en cada nodo. Con los árboles de decisión ajustamos los siguientes hiperparámetros: la profundidad máxima del árbol y el número mínimo de instancias que debe haber en cada nodo para subdividir el árbol.

2.1.3. Máquinas de Vectores de Soporte

Con las máquinas de vectores de soporte o SVM's, en los problemas de regresión, a diferencia de los problemas de clasificación, el objetivo es que las instancias queden dentro del margen del hiperplano. Por tanto, se trata de encontrar el hiperplano que mejor se ajuste a los datos de entrenamiento y que minimice el error en la predicción; es decir, que deje el menor número de instancias fuera del hiperplano.

Para los problemas no lineales, podemos utilizar las funciones kernel, que trazan un hiperplano no lineal. En concreto, utilizamos el kernel lineal (2.4), que es equivalente a hacer una regresión lineal, y el kernel radial (2.5), que se puede ver como una suma ponderada de similaridades entre los 2 vectores soporte. Los hiperparámetros que se ajustan son el parámetro de regularización C, que controla el equilibrio entre maximizar el margen y minimizar la pérdida, y gamma, que controla la influencia de una observación en el entrenamiento.

$$k(x_i, x_j) = x_i^T \cdot x_j \qquad (2.4)$$

$$k(x_i, x_j) = exp(\frac{-\|x_i - x_j\|^2}{2\sigma^2}) \qquad (2.5)$$

2.1.4. Gradient Boosting

El Gradient Boosting es una de las técnicas de aprendizaje supervisado que forman parte de los modelos de conjunto o "ensembles". En los modelos de conjunto se construyen múltiples modelos base y se agregan para crear un modelo final. Estos modelos base son dependientes y homogéneos, y se denominan "weak learners", pues son modelos que obtienen un error similar al azar. En Gradient Boosting los modelos base suelen ser modelos de árbol de decisión con poca profundidad, también llamados tocones. Estos modelos base se construyen de manera secuencial de forma que cada modelo se centra en corregir los errores del modelo anterior. Este proceso se repite para todos los modelos base y en cada iteración se busca reducir el error residual del modelo anterior. Para reducir este error, se utiliza el descenso del gradiente que nos indica la dirección que debemos tomar para ajustar el modelo de forma correcta.

Como hiperparámetros se ajustan el número de modelos base que se crean y la velocidad de aprendizaje del algoritmo. Una velocidad de aprendizaje muy baja y un número de modelos base muy grande provocará que haya sobreajuste o sobreaprendizaje de los datos de entrenamiento y el modelo no generalizará bien con los datos de validación. Por otro lado, con el Gradient Boosting podemos conocer la importancia de las variables del modelo, y realizar selección de características si fuera preciso. En el caso de los

problemas de regresión, las variables más importantes son aquellas que más ayudan a reducir la varianza del modelo entrenado.

En este trabajo se implementa el Extreme Gradient Boosting o XGBoost, que es uno de los algoritmos más populares en los últimos años. El XGBoost es un algoritmo paralelizable que nos permite trabajar con grandes cantidades de datos y cuyo tiempo de entrenamiento es muy corto. Además, nos permite controlar el sobreajuste del modelo mediante términos de regularización y podemos definir nuestras propias funciones de pérdida.

2.1.5. Random Forest

El Random Forest, al igual que el Gradient Boosting, es un modelo de conjuntos. El Random Forest es una técnica de Bagging o "Boostrap Aggregation", que consiste en la construcción independiente de los modelos base utilizando muestreo aleatorio con reemplazamiento. El Random Forest extiende estos modelos y, además de utilizar el muestreo aleatorio con reemplazamiento del Bagging, le añade la randomización de variables en los modelos base, que suelen ser árboles de decisión.

En los árboles de decisión del Random Forest se realiza la randomización de variables, donde se selecciona un número reducido de ellas y, para los problemas de regresión, se suele utilizar un tercio del total de características. Este modelo de aprendizaje utiliza la estimación "Out of Bag", que consiste en que el error para cada observación se obtiene usando solamente los árboles en los que no se utilizó dicho dato para construirlo. Al igual que el Gradient Boosting, con este método podemos ver la importancia de las variables. Los hiperparámetros que se ajustan son: el número de árboles o modelos base que se crean y el tamaño del subconjunto de variables seleccionado para la construcción de cada modelo base.

2.1.6. Redes Neuronales

Las redes neuronales están compuestas por capas de entrada, capas ocultas y capas de salida. La capa de entrada tiene tantas neuronas como variables hay en el conjunto de datos. Las entradas para cada neurona se multiplican por unos pesos, se suman y se transforman utilizando la función de activación. La capa de salida puede tener tantas neuronas como clases haya para problemas de clasificación y una única neurona para problemas de regresión.

El número de capas ocultas se determina en función de la complejidad del problema, cuántas más capas ocultas haya, mejor podremos captar las relaciones complejas de los datos. En cuanto a la función de activación, en las capas ocultas se utiliza la función relu y, dado que tenemos un problema de regresión, en la capa de salida no utilizamos ninguna función. La función de coste que se utiliza para problemas de regresión es el error cuadrático medio o MSE, que ya se ha definido en la expresión (2.3).

Los hiperparámetros que se ajustan son: el número máximo de iteraciones que utiliza la red neuronal para encontrar los pesos óptimos y el tamaño de las capas ocultas. También podríamos haber ajustado hiperparámetros de regularización como alpha o podríamos

haber ajustado la tasa de aprendizaje, pero entonces el tiempo de entrenamiento del modelo se incrementaría exponencialmente. Si se tuviese un ordenador tan potente como para realizar estos ajustes en un periodo de tiempo relativamente corto, entonces sí se haría, pero la realidad es que contamos con recursos limitados.

2.1.7. Ajuste de Hiperparámetros

El ajuste de hiperparámetros es un paso clave que nos acerca al modelo que mejor se ajusta a los datos. En cada modelo de aprendizaje supervisado podemos establecer una serie de valores para cada hiperparámetro y dependiendo de sus valores, el ajuste del modelo será mejor o peor. Debemos tener especial cuidado en esta parte pues no queremos obtener un modelo que se sobreajuste a los datos de entrenamiento y generalice erróneamente para los datos de validación. Es preciso realizar una pausa y comentar sobre la validación cruzada y las validaciones interna y externa.

La validación cruzada hace referencia a la división del conjunto de datos en un conjunto de entrenamiento y en otro de validación. En concreto, la validación cruzada divide los datos en k "folds" o partes iguales, se entrena el modelo con k – 1 partes y se valida con la parte restante, repitiéndose este proceso hasta que cada parte haya actuado como conjunto de validación. Si tenemos un conjunto de datos con 3000 filas y se utiliza validación cruzada con k = 3 folds, entonces se van a crear 3 partes de 1000 filas cada una, se entrenará el modelo con 2 partes y se validará con la parte restante, repitiéndose el proceso 3 veces. Para obtener la métrica de evaluación global del modelo, la calculamos como una media de los resultados de las métricas obtenidas en cada iteración. Para este ejemplo, suponiendo que tenemos un problema de regresión y que $R^2 = \{0.55, 0.65, 0.75\}$, obtenemos que el R^2 global del modelo será 0.65.

En el proceso de entrenamiento del modelo, podemos distinguir 2 etapas: una en la que se realiza el ajuste de hiperparámetros y otra en la que se entrena el modelo tras haber hallado los mejores valores de los hiperparámetros. A la etapa de ajuste de hiperparámetros se le denomina validación interna y la etapa de entrenamiento del modelo se le llama validación externa. En este trabajo, para la validación interna se ha utilizado validación cruzada con k = 10 folds, y para la validación externa se ha utilizado el método "holdout" = 3/4, que realiza una partición aleatoria sin reemplazamiento de los datos, de los cuales 3/4 se dedican al entrenamiento del modelo y 1/4 a validación.

Para ajustar hiperparámetros se pueden utilizar diversos métodos, pero en este trabajo se van a ver solamente 2 de ellos, estos son el "Grid Search" y el "Random Search".

El "Grid Search" o la búsqueda en rejilla es un método de ajuste de hiperparámetros en el que se prueban todas y cada una de las posibles combinaciones de hiperparámetros que se hayan definido. Mientras que el "Random Search" o la búsqueda aleatoria es un método en el que se prueban de forma aleatoria algunas de las posibles combinaciones de hiperparámetros. En las tablas 2 y 3 podemos ver la diferencia entre ambos métodos, en los cuales se han considerado los hiperparámetros del árbol de decisión. El ejemplo se ha realizado ajustando el número mínimo de instancias que debe haber en cada nodo para ser subdividido (min_samples_split) y la profundidad máxima del árbol (max_depth). En

la tabla 2 vemos que se han comprobado todas las 16 posibles combinaciones de los hiperparámetros, mientras que en la tabla 3 se han comprobado 8 de ellas de forma aleatoria.

En cuanto a la interpretación de los resultados y a la elección de un enfoque para ajustar hiperparámetros, podemos decir que con el "Grid Search" estamos seguros de que vamos a encontrar la mejor combinación de hiperparámetros y con ello el mejor modelo. Pero, si queremos ajustar muchas combinaciones de hiperparámetros, debemos tener en cuenta que el tiempo de ejecución se va a elevar desmesuradamente. Dado que en muchas ocasiones no tenemos los recursos suficientes para ello, nos decantamos por el "Random Search", con el cual no estamos 100% seguros de que vayamos a obtener la mejor combinación de hiperparámetros, pero obtendremos un ajuste bastante adecuado. Por lo tanto, en situaciones reales con recursos limitados, utilizamos el enfoque que ofrece la búsqueda aleatoria de hiperparámetros.

		min_samples_split			
		5	6	7	8
max_depth	2	(2, 5)	(2, 6)	(2, 7)	(2, 8)
	3	(3, 5)	(3, 6)	(3, 7)	(3, 8)
	4	(4, 5)	(4, 6)	(4, 7)	(4, 8)
	5	(5, 5)	(5, 6)	(5, 7)	(5, 8)

Tabla 2 Búsqueda en rejilla de hiperparámetros

		min_samples_split			
		5	6	7	8
max_depth	2		(2, 6)		(2, 8)
	3	(3, 5)		(3, 7)	
	4		(4, 6)	(4, 7)	
	5	(5, 5)		(5, 7)	

Tabla 3 Búsqueda aleatoria de hiperparámetros

2.1.8. Métricas de Evaluación del Modelo

Para aplicar las métricas de evaluación del modelo, debemos distinguir entre los problemas de clasificación y los de regresión, pues utilizan métricas diferentes. Como en nuestro caso, tenemos un problema de regresión, no podemos utilizar métricas como la accuracy o la precisión, sino que recurrimos al error cuadrático medio (MSE) y al R^2. Entre estas 2 métricas, nos decantamos por el R^2 pues con el MSE los datos que están muy mal predichos tienen mucho peso en la media. Por tanto, la métrica de evaluación de

los modelos que se ha utilizado es R^2, que nos indica el porcentaje de variabilidad de la variable respuesta explicada por nuestro modelo. Podemos obtener el valor de R^2 de acuerdo con la expresión (2.6).

$$R^2 = 1 - \frac{RSS}{TSS} \qquad (2.6)$$

A su vez, obtenemos la suma residual de cuadrados (RSS) y la suma total de cuadrados (TSS) conforme a las expresiones (2.7) y (2.8) respectivamente.

$$RSS = \sum_i (y_i - \widehat{y_i})^2 \qquad (2.7)$$

$$TSS = \sum_i (y_i - \bar{y})^2 \qquad (2.8)$$

Donde:

$$\mathrm{y_i} = \text{ Valor real de la variable respuesta para la observación i} - \text{ésima}$$

$$\widehat{y_i} = \text{Valor predicho de la variable respuesta para la observación i} - \text{ésima}$$

$$\bar{\mathrm{y}} = \text{ Valor medio de los valores reales de la variable respuesta}$$

Completamos la definición de esta métrica de evaluación estableciendo los criterios para determinar si un modelo es adecuado o no. La métrica R^2 está definida en el intervalo [0,1], donde un valor cercano a 0 representa un modelo pésimo y un valor cercano a 1 representa un modelo satisfactorio. En nuestro caso, nos va a interesar aquel modelo que logre maximizar el valor de esta métrica.

2.2. Aprendizaje No Supervisado

A diferencia del aprendizaje supervisado, el aprendizaje no supervisado tiene por objetivo encontrar patrones en el conjunto de datos, en el que no existe una variable respuesta. Dentro del aprendizaje no supervisado podemos distinguir multitud de técnicas, pero en este trabajo nos centramos en el agrupamiento de instancias con el algoritmo de clustering k-medias y el clustering jerárquico. A continuación, procedemos a profundizar más sobre estas técnicas.

2.2.1. Clustering Jerárquico

El clustering jerárquico se encuentra dentro del análisis de conglomerados y consiste en que, dado un conjunto de datos sobre los que se ha calculado una medida de distancia, se agrupan los datos más cercanos en clústeres o grupos. Esta medida de distancia puede ser la distancia euclídea, la de Manhattan, la de Mahalanobis o cualquier otra distancia, que dependerá del conjunto de datos que se tenga y del problema a resolver.

En el clustering jerárquico no conocemos el número de clústeres o grupos que se deben formar, sino que el objetivo de esta técnica es determinar el número de clústeres óptimo que mejor representa al conjunto de datos. Para determinar el número de clústeres más adecuado, podemos tomar 2 enfoques distintos: el algoritmo jerárquico divisivo, que parte de un único conglomerado y va subdividiendo en clústeres cada vez más pequeños, y el

algoritmo jerárquico aglomerativo, que parte de n conglomerados y va agrupando en clústeres cada vez más grandes.

En este caso, utilizamos el algoritmo jerárquico aglomerativo y agrupamos los clústeres utilizando el método de Ward, que agrupa los clústeres de forma que se minimice la varianza dentro del nuevo clúster. Para ver de forma gráfica el resultado del clustering jerárquico se crea un dendrograma, que muestra la estructura jerárquica de los conglomerados.

2.2.2. Algoritmo de Clustering K-Medias

El algoritmo de clustering no jerárquico k-medias agrupa el conjunto de datos en un número predefinido de clústeres, que son internamente homogéneos y externamente heterogéneos. El algoritmo de las k-medias comienza seleccionando aleatoriamente k puntos del conjunto de datos que actúan como centroides, se calcula la distancia de cada punto a los k centroides y se asigna cada punto al centroide que esté más próximo. Una vez que se han asignado todos los puntos, se calculan los nuevos centroides para los k clústeres y se repite el proceso sucesivamente hasta que el criterio ya no mejore.

Este algoritmo es bastante potente y sencillo de utilizar, pero presenta una desventaja pues debemos determinar el número de conglomerados que se van a formar antes de aplicar el algoritmo. Sin embargo, este inconveniente se puede cubrir fácilmente mediante el clustering no jerárquico o utilizando el método de la silueta media. El método de la silueta media evalúa la calidad de la clasificación realizada para cada punto del conjunto de datos, de acuerdo con la expresión en (2.9). El número k de conglomerados que maximice la silueta media será el número óptimo de clústeres que se incluirán en el algoritmo k-medias.

$$s(i) = \frac{b(i) - a(i)}{máx\{a(i), b(i)\}} \qquad (2.9)$$

Donde:

$$\text{a(i)} = \text{Distancia media entre i y todos los puntos de su propio clúster}$$

$$\text{b(i)} = \text{Distancia media entre i y todos los puntos del clúster más cercano distinto al suyo}$$

Una vez explicada la metodología que se ha empleado para realizar el trabajo, tanto las técnicas de aprendizaje supervisado como no supervisado, se procede a exponer y analizar los resultados obtenidos.

3. ANÁLISIS DE RESULTADOS

3.1. ETL: Extracción, Transformación y Carga

Los datos han sido extraídos mediante una tarea de "web scraping" sobre la página web de pisos.com. Se ha escogido esta página web porque la extracción de la información era sencilla y porque el contenido html estaba mejor estructurado que otras páginas web. Es importante mencionar que se intentó realizar la misma actividad con otras páginas web como la de Idealista, pero tenía más restricciones. La página web de Idealista bloqueaba el acceso a los robots tras un par de intentos de acceso a la página. También había otras restricciones como CAPTCHA, que el robot no podía resolver y había que resolverlo de forma manual. Aunque en un principio se diseñó el "web scraping" para realizarlo sobre la página web de Idealista, se vio que había demasiados problemas y dificultades. Por tanto, se realizó sobre la página web de pisos.com, que ofrecía prácticamente la misma información, era más sencillo de manejar y no presentaba tantas limitaciones.

Es de importancia resaltar que la extracción de los datos se quería realizar en principio sobre todas las viviendas disponibles en la página web de pisos.com, en su momento alrededor de 6500 viviendas. Estas 6500 viviendas estaban distribuidas en páginas, cada una conteniendo 30 viviendas, y para acceder a las 30 siguientes había que pasar de página. Por tanto, se diseñó un proceso para ir accediendo a todas las viviendas de la página, a pasar de página y repetir este proceso hasta que no hubiera más páginas. Pero, cuando llegaba a la vivienda número 3000 en la página 100, el robot al pasar de página no saltaba a la siguiente y accedía a las viviendas 3001 – 3030, sino que volvía a la primera página. Es decir, que la página web estaba diseñada únicamente para mostrar 3000 viviendas o 100 páginas. A pesar de estas limitaciones, las aproximadamente 3000 instancias son suficientes para realizar el estudio y no se precisa recoger más información. Teniendo estas limitaciones en cuenta, podemos pasar a explicar las actividades de la ETL.

Los datos en su momento fueron extraídos de forma trivial y, en algunos casos no aportaban información útil, por lo que era preciso transformar las características. Las transformaciones que se han realizado comienzan por cambiar la variable *direccion,* que contenía la dirección de la vivienda. La variable *direccion* realmente se podría haber utilizado con otro fin, pues podríamos haber obtenido el nombre de la calle y agrupar las viviendas por calles. Pero, parece demasiado minucioso llegar a tal nivel de detalle pues no tenemos suficientes viviendas como para agruparlas por calles. También habría que tener en cuenta que hay calles muy pequeñas que igual contendrían 2 viviendas y otras como la calle Alcalá que podría contener cientas.

La agrupación de viviendas por calles era una idea compleja y se vio superada por la agrupación de viviendas por distritos, una agrupación de mayor utilidad. Es por ello que se decide transformar la variable *direccion* extrayendo la información importante de esta variable, que sigue la siguiente estructura: tipo de vivienda + texto. En este caso, nos interesa extraer el tipo de vivienda que nos puede servir de gran utilidad para agrupar las

viviendas. Podemos ver la transformación completa de la dirección de la vivienda en la tabla 4.

Piso en venta en Calle de Sondica	**→**	**Piso**
Atico en venta en Calle del Duque de Rivas	**→**	**Atico**
Chalet en venta en Hortaleza	**→**	**Chalet**

Tabla 4 Transformación de la variable dirección de la vivienda

Tras transformar la variable *direccion*, se decide que se debe transformar el formato de las variables que en principio serán numéricas: *precio, superficie construida, habitaciones* y *baños.* Para las variables *superficie construida*, *habitaciones* y *baños* se transforman los valores que no se pueden cambiar a numéricos, como las cadenas de texto, en valores faltantes. Para la variable *precio,* eliminamos los valores faltantes y los valores de tipo texto que no se pueden cambiar a numéricos, valores del tipo: {A consultar, nan}. Aunque es cierto que estas viviendas se podrían haber utilizado como datos de validación de los modelos de aprendizaje supervisado, la realidad es que se elimina una cantidad minúscula de instancias, alrededor de 15 viviendas. Podemos ver la transformación que se realiza para el precio de la vivienda en la tabla 5 y para la superficie construida en la tabla 6.

230.000 euros	**→**	**230.000**
1.370.000 euros	**→**	**1.370.000**
1.750.000 euros	**→**	**1.750.000**

Tabla 5 Transformación de la variable precio de la vivienda

91 metros cuadrados	**→**	**91**
78 metros cuadrados	**→**	**78**
39 metros cuadrados	**→**	**39**

Tabla 6 Transformación de la variable superficie construida de la vivienda

Una vez realizadas estas transformaciones, se decide analizar las variables *planta* e *inmobiliaria.* Al calcular el número de valores faltantes que había en la variable *planta*, nos damos cuenta de que son más de 1/3 de los datos, por lo que se decide eliminar esta variable del conjunto de datos. En cuanto a la variable *inmobiliaria,* no podemos agrupar las viviendas por inmobiliarias pues hay demasiados valores distintos. Se intentó agrupar las viviendas por las inmobiliarias más famosas y crear una categoría llamada "resto" que agrupara a las inmobiliarias más pequeñas, pero no fue posible. Esta categoría "resto" contenía la mayoría de las viviendas y no se podía realizar una agrupación adecuada de

las viviendas por inmobiliaria, por tanto se eliminó esta característica del conjunto de datos.

La siguiente variable que se transforma es el *distrito,* que es una variable de tipo texto con la siguiente estructura: texto + Distrito + texto; de la que debemos extraer la información del distrito. Podemos ver la transformación completa del distrito de la vivienda en la tabla 7.

Las Aguilas (Distrito Latina. Madrid Capital)	**→**	**Latina**
Buenavista (Distrito Carabanchel. Madrid Capital)	**→**	**Carabanchel**
Vallehermoso (Distrito Chamberi. Madrid Capital)	**→**	**Chamberi**

Tabla 7 Transformación de la variable distrito de la vivienda

La última transformación que se debe realizar afecta al resto de variables que no se han mencionada hasta ahora, que son las variables binarias. Estas variables toman valor "no" o "si" dependiendo de si la vivienda posee o no un atributo determinado. Debemos transformar estas variables y codificarlas como 0 y 1 en vez de "no" y "si".

Con estas transformaciones de las variables, ya tenemos el conjunto de datos final que se utilizará para aplicar los métodos de aprendizaje automático, el siguiente paso es guardar y cargar los datos. Una vez que realizamos esto, podemos dar por terminada la etapa de la ETL y pasamos a realizar el análisis exploratorio de los datos o EDA.

3.2. EDA: Análisis Exploratorio de los Datos

Tras finalizar la fase de extracción, tratamiento y carga de los datos o ETL, a continuación es de vital importancia llevar a cabo un análisis exploratorio de los datos o EDA. El objetivo principal de este análisis es comprender la naturaleza de los datos con los que estamos trabajando y determinar qué tareas será necesario realizar durante la fase del preproceso.

3.2.1. Exploración y Tratamiento de Valores Atípicos

Antes de tomar cualquier paso para explorar los datos, debemos examinar las variables en busca de valores atípicos o "outliers". Debemos identificar dichos valores atípicos y manejarlos de forma que no afecten al rendimiento de los modelos de aprendizaje supervisado. La decisión que se toma es que se conservan las observaciones débilmente atípicas, aquellas que están cerca de los límites definidos por los diagramas de caja, y se eliminan las observaciones extremadamente atípicas, aquellas que están muy lejos de los límites definidos. Para determinar qué observaciones son débilmente atípicas y cuáles son extremadamente atípicas, se establece la siguiente regla de decisión: todos aquellos valores que queden fuera del intervalo de decisión se considerarán extremadamente atípicos y se eliminarán del conjunto de datos. Podemos obtener la fórmula que se ha utilizado para crear el intervalo de decisión en la expresión (3.1).

$$Intervalo\ de\ decisión = [Q_1 - 3 \cdot IQR\ ;\ Q_3 + 3 \cdot IQR] \qquad (3.1)$$

Donde:

$$IQR = Q_3 - Q_1 \qquad (3.2)$$

El IQR es el rango intercuartílico, definido en (3.2), y se decide que se eliminan aquellos valores que sean menores que el primer cuartil menos 3 veces el rango intercuartílico, y aquellos valores que sean mayores que el tercer cuartil más 3 veces el rango intercuartílico. Este es el intervalo de decisión que se ha definido, y los valores que queden fuera de él se consideran valores extremos que se deben eliminar, pues no son una representación fiel de los datos. Como ya se ha mencionado, esta eliminación de los valores atípicos nos resultará de gran ayuda cuando apliquemos las técnicas de aprendizaje supervisado, ya que hay modelos que se ven afectados por la presencia de "outliers".

Ahora bien, para determinar los valores atípicos se decide realizar diagramas de caja o "boxplots" de las variables numéricas agrupadas por distritos. Además, en los boxplots se ordenan los distritos de mayor a menor precio medio de las viviendas, de forma que a la izquierda están los distritos con las viviendas más lujosas y a la derecha los distritos con las viviendas más asequibles. En las figuras 5 a 8 podemos ver los boxplots de los datos brutos; es decir, incluyendo los valores atípicos demasiado extremos, mientras que en las figuras 9 a 12 vemos los boxplots una vez se han eliminado estos valores extremadamente atípicos.

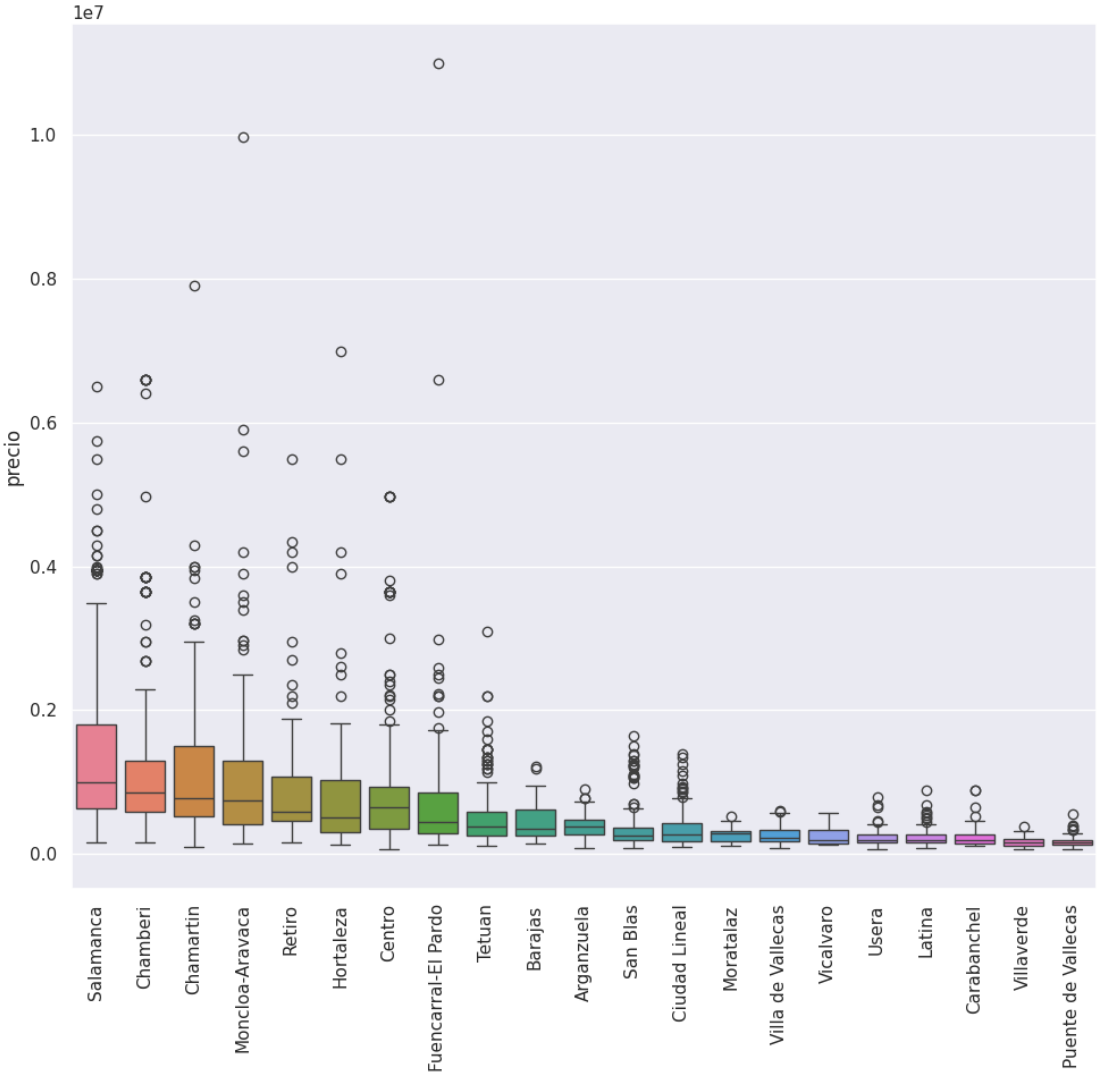

Figura 5 Diagrama de caja del precio de las viviendas por distrito

En la figura 5 observamos los boxplots del precio de las viviendas, nuestra variable respuesta, agrupadas por distritos. A simple vista podemos ver que hay algunos valores extremadamente atípicos como pueden ser viviendas de 11 millones de euros en Fuencarral-El Pardo o de 10 millones de euros en Moncloa-Aravaca. Estos valores atípicos los debemos eliminar del conjunto de datos, pero seguro que encontramos muchos más si aplicamos la regla de decisión que se ha definido en (3.1). Como hay valores que son demasiado atípicos, estos dificultan mucho la labor de explorar los outliers para los distritos más económicos como Vicálvaro o Villaverde.

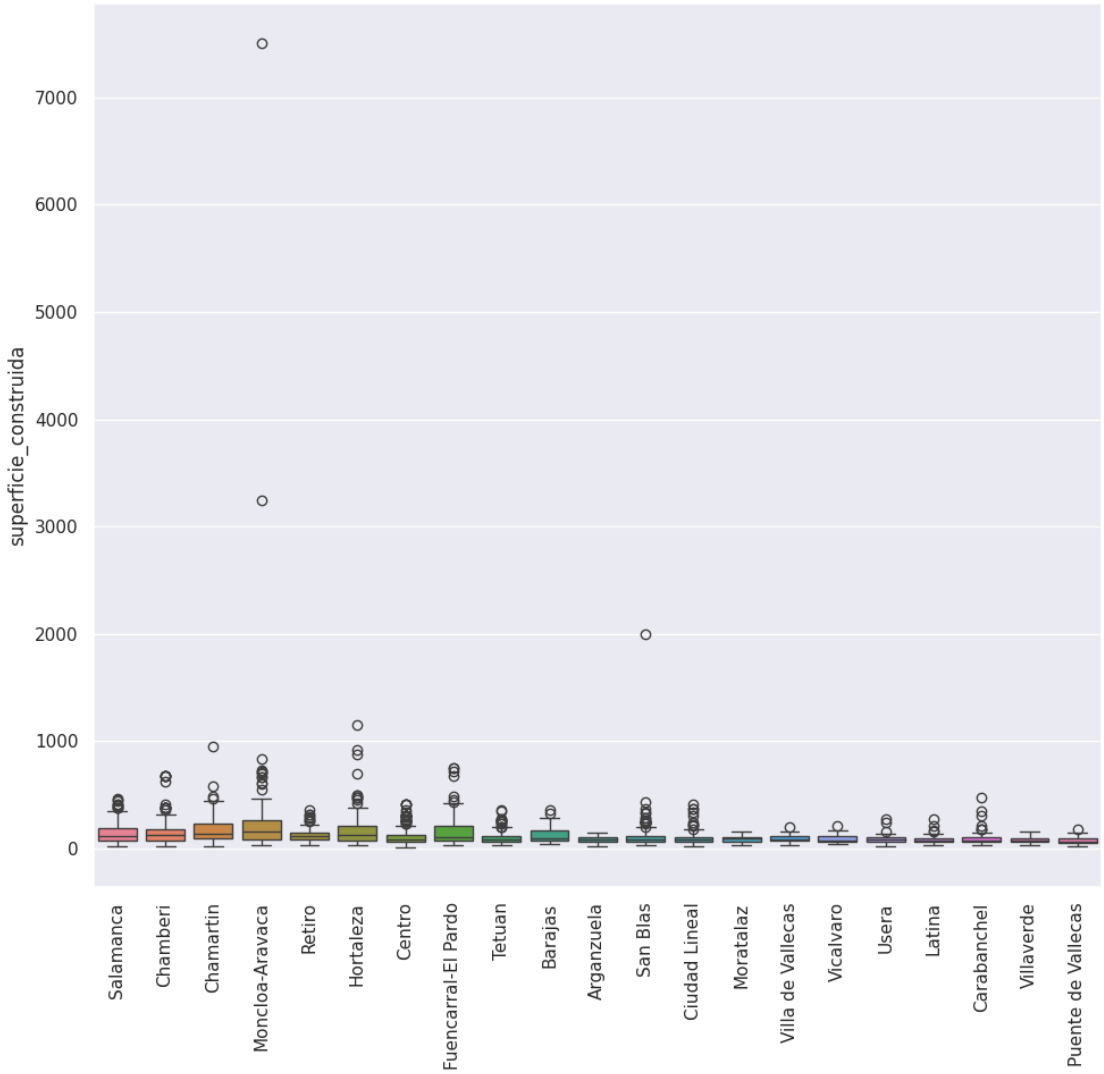

Figura 6 Diagrama de caja de la superficie construida de las viviendas por distrito

En la figura 6 observamos los boxplots de la superficie construida de las viviendas agrupadas por distritos. A simple vista podemos ver que hay algunos valores extremadamente atípicos como pueden ser viviendas de más de 7000 metros cuadrados construidos en Moncloa-Aravaca o de 2000 metros cuadrados en San Blas. Como hay valores que son demasiado atípicos, estos dificultan mucho la labor de explorar los outliers para todos los distritos, en esta figura no podemos observar aspectos demasiado significativos del conjunto de datos.

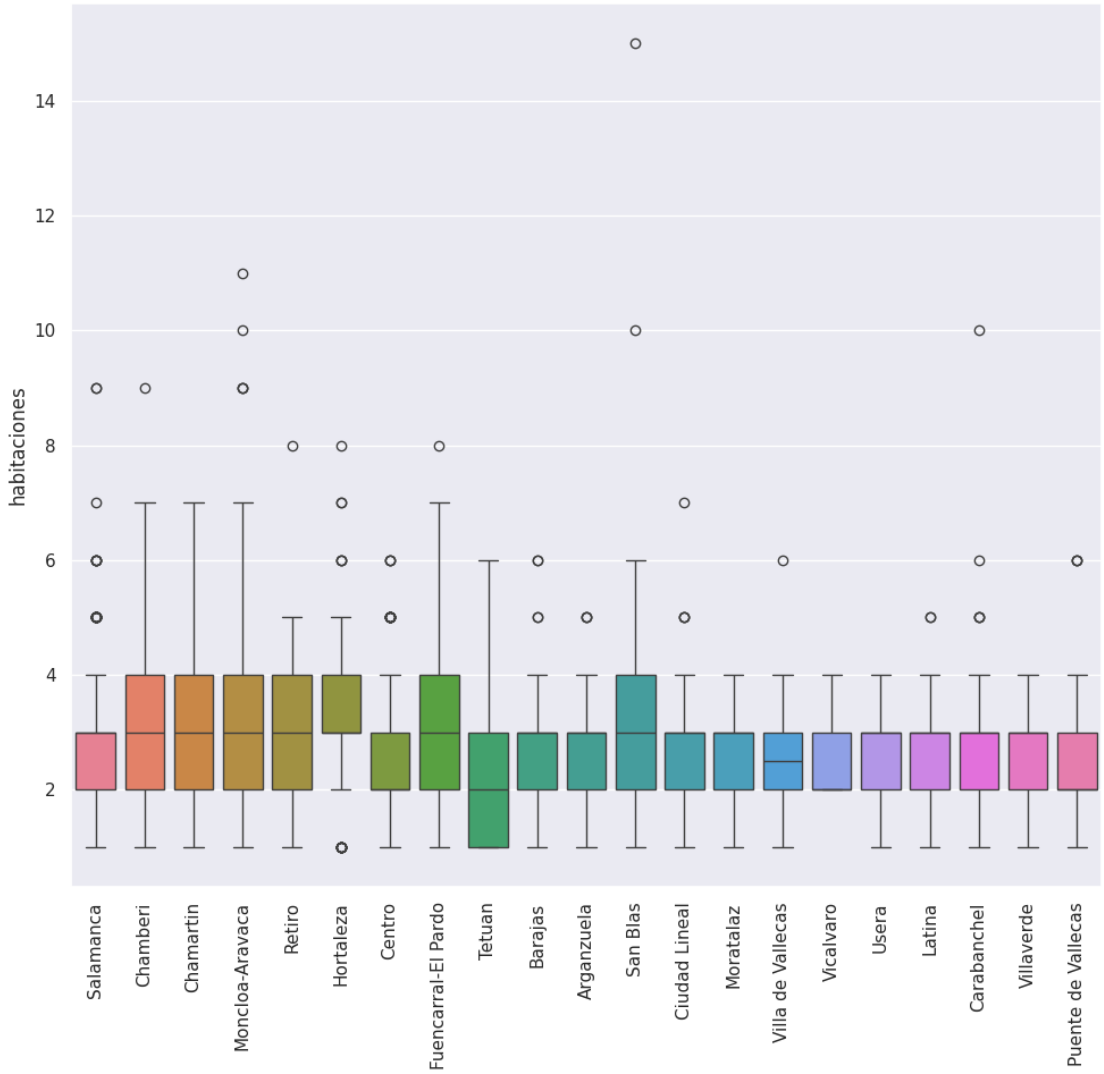

Figura 7 Diagrama de caja del número de habitaciones por distrito

En la figura 7 observamos los boxplots del número de habitaciones de las viviendas agrupadas por distritos. A simple vista podemos ver que hay algunos valores extremadamente atípicos como pueden ser viviendas de 9 habitaciones en Salamanca o de 15 habitaciones en San Blas. Aunque es cierto que hay valores demasiado atípicos, el número de habitaciones está definido en un rango de valores reducido, de 1 a 15, y podemos ver claramente los outliers para cada distrito.

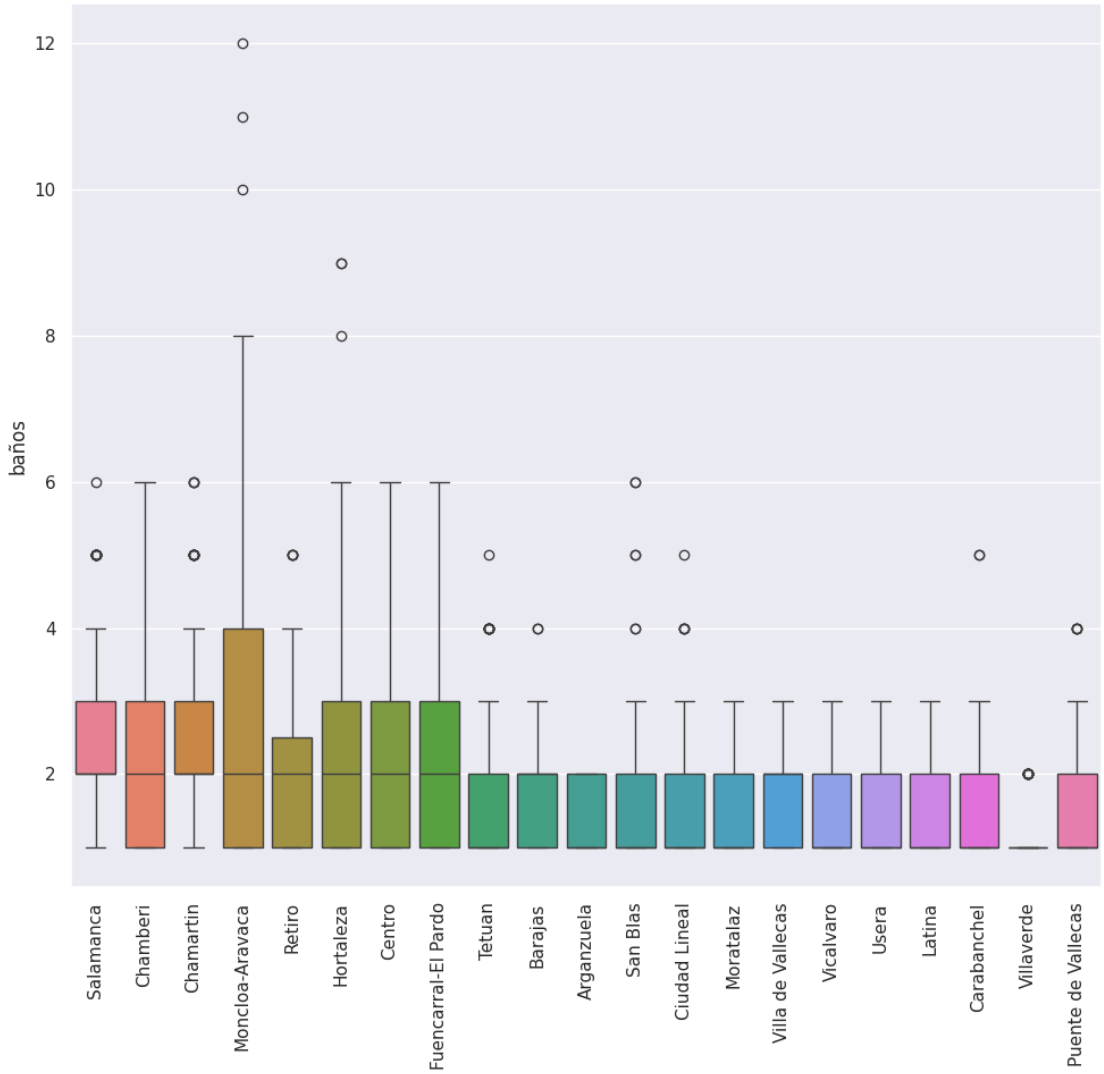

Figura 8 Diagrama de caja del número de baños por distrito

En la figura 8 observamos los boxplots del número de baños de las viviendas agrupadas por distritos. A simple vista no podemos ver que haya muchos valores extremadamente atípicos, y de hecho se observan pocos valores atípicos. Sin embargo, resulta llamativo que en el distrito de Moncloa-Aravaca una vivienda con 8 baños no se considera algo atípico y que el 25% de las viviendas en este distrito tienen 4 baños o más. En este caso, no podemos aplicar nuestra regla de decisión, pues en el distrito de Villaverde tenemos que Q1 = Q3, y por tanto, IQR = 0. Si aplicásemos nuestra regla de decisión, veríamos que el intervalo de decisión resultante nos indicaría que nos quedásemos con todas las viviendas que están entre el primer y el tercer cuartil y entonces, estaríamos eliminando el 50% de las viviendas de este distrito. Por lo que, para evitar eliminar instancias del conjunto de datos de forma innecesaria, no se aplica la regla de decisión para el número de baños de la vivienda.

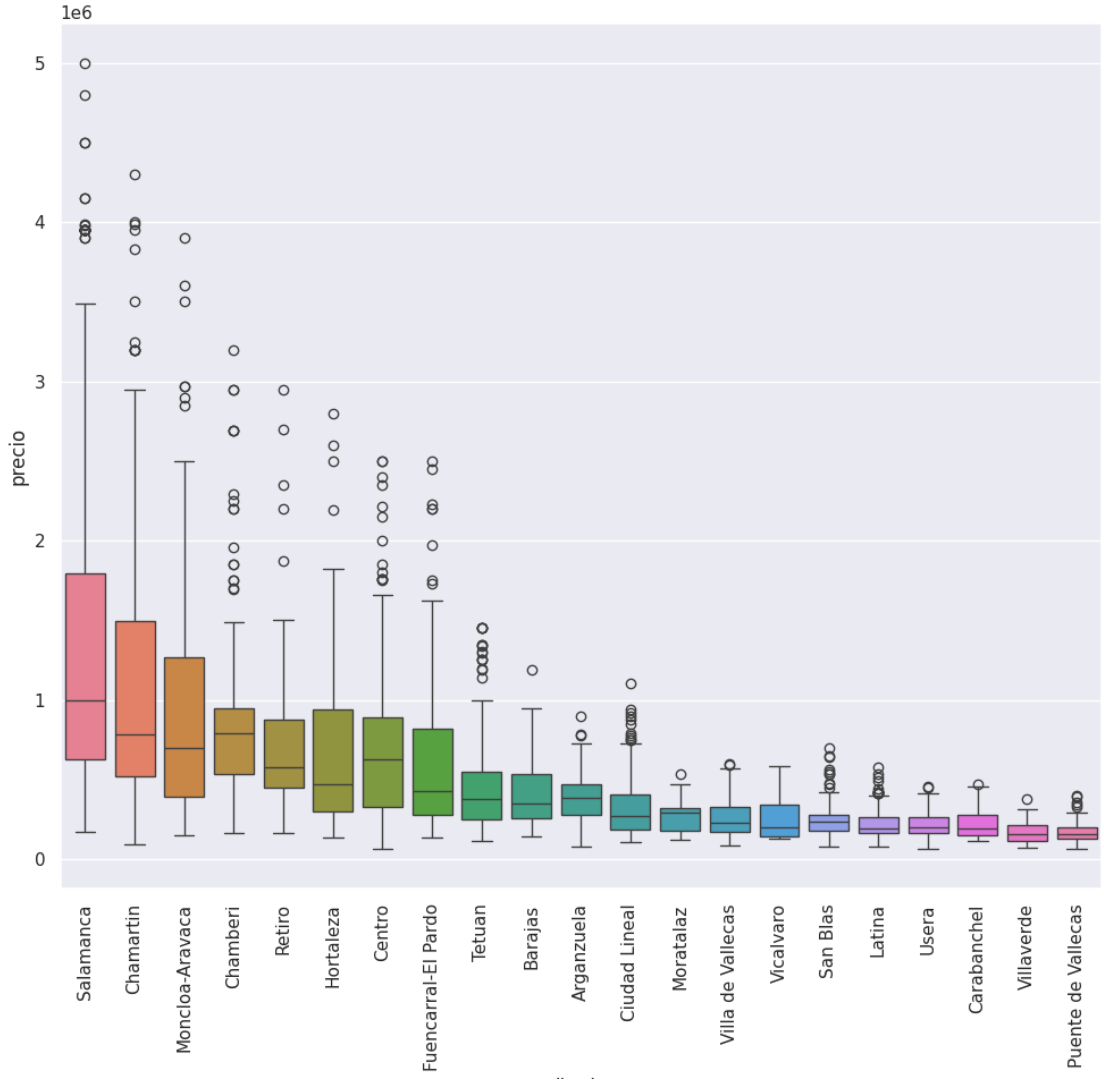

Figura 9 Diagrama de caja del precio de las viviendas por distrito sin valores atípicos extremos

En la figura 9 observamos boxplots del precio de las viviendas, nuestra variable respuesta, agrupadas por distritos, y habiéndose eliminado los valores extremadamente atípicos. Como podemos ver, el precio de las viviendas en la figura 5 antes estaba comprendido en un rango de 0 a 11 millones de euros y ahora en la figura 9 está comprendido en un rango de 0 a 5 millones de euros. Es cierto que sigue habiendo bastantes valores atípicos, pero estos son los valores que hemos denominado como “débilmente atípicos”, y se espera que estas observaciones no afecten al rendimiento de los modelos de aprendizaje supervisado.

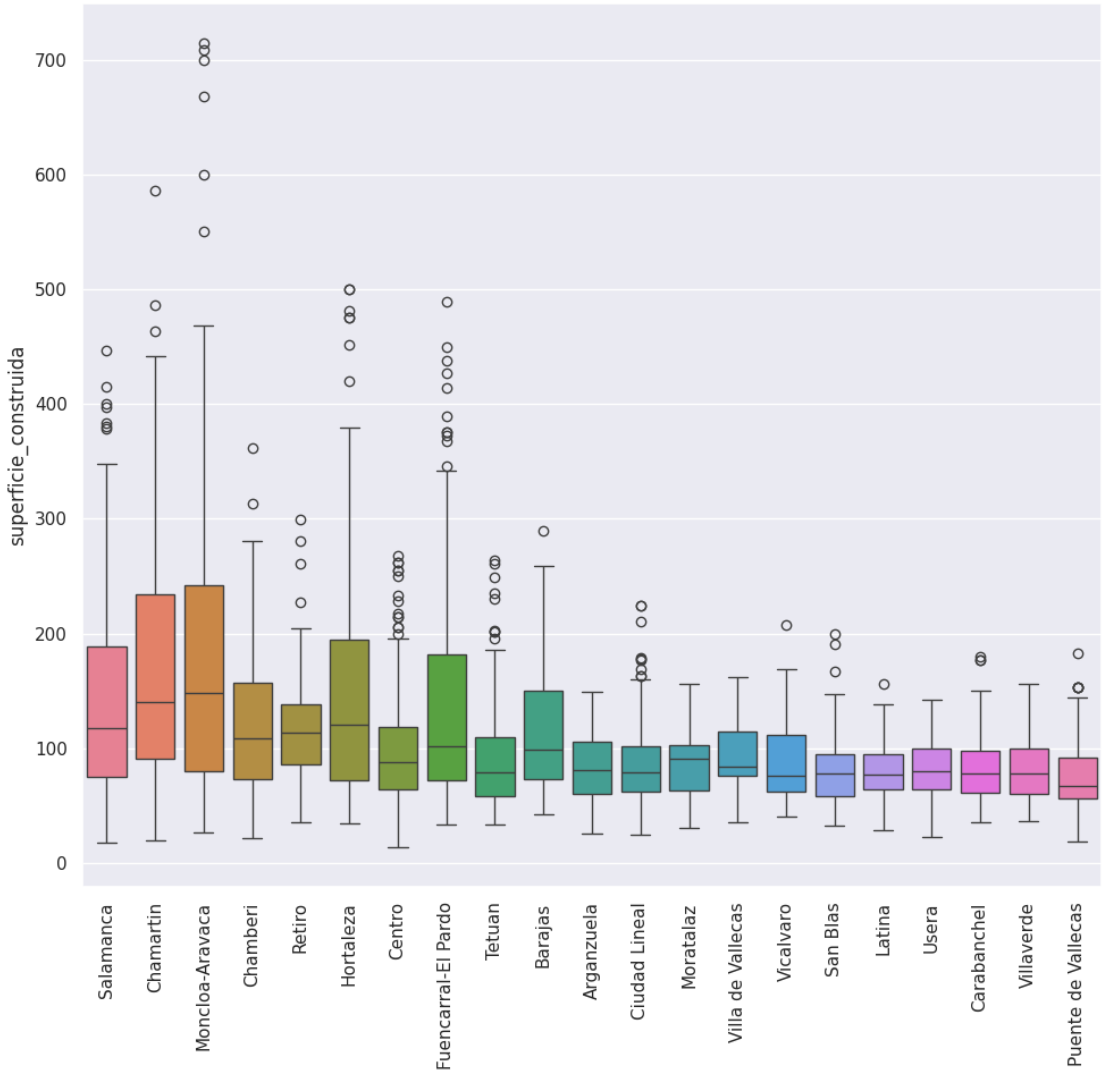

Figura 10 Diagrama de caja de la superficie construida de las viviendas sin valores atípicos extremos

En la figura 10 observamos boxplots de la superficie construida de las viviendas agrupadas por distritos, y habiéndose eliminado los valores extremadamente atípicos. Como podemos ver, la superficie construida de las viviendas en la figura 6 antes estaba comprendido en un rango de 0 a 7500 metros cuadrados y ahora en la figura 9 está comprendido en un rango de 0 a 750 metros cuadrados. Es cierto que sigue habiendo muchos valores atípicos, pero estos son valores "débilmente atípicos", y se espera que estas observaciones no afecten al rendimiento de los modelos de aprendizaje supervisado.

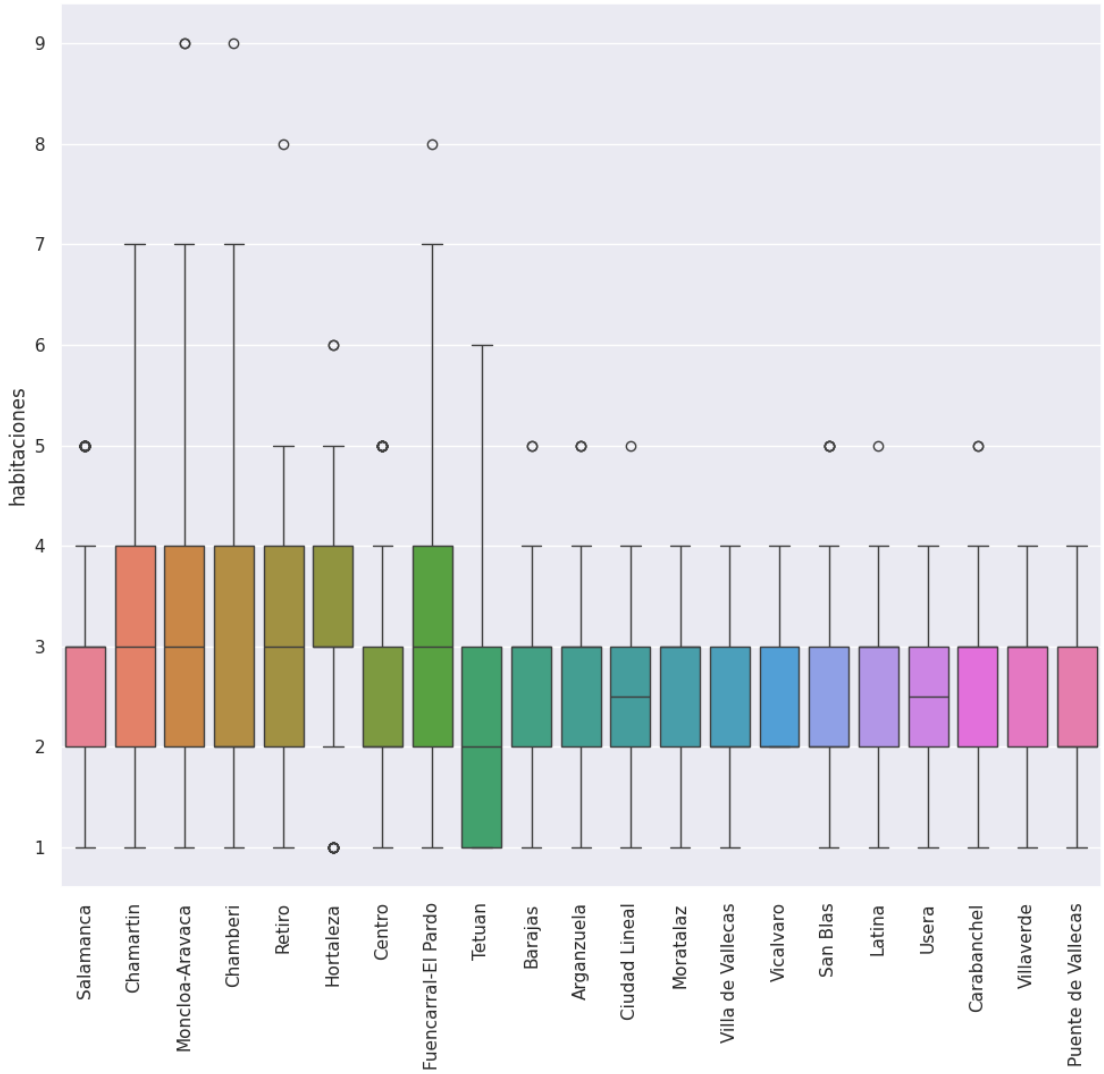

Figura 11 Diagrama de caja del número de habitaciones de las viviendas sin valores atípicos extremos

En la figura 11 observamos boxplots del número de habitaciones de las viviendas agrupadas por distritos, y habiéndose eliminado los valores extremadamente atípicos. Como podemos ver, el número de habitaciones de las viviendas en la figura 7 antes estaba comprendido en un rango de 1 a 15 habitaciones y ahora en la figura 11 está comprendido en un rango de 1 a 9 habitaciones. No observamos prácticamente ningún valor atípico y los que se observan son valores “débilmente atípicos”, y se espera que estas observaciones no afecten al rendimiento de los modelos de aprendizaje supervisado.

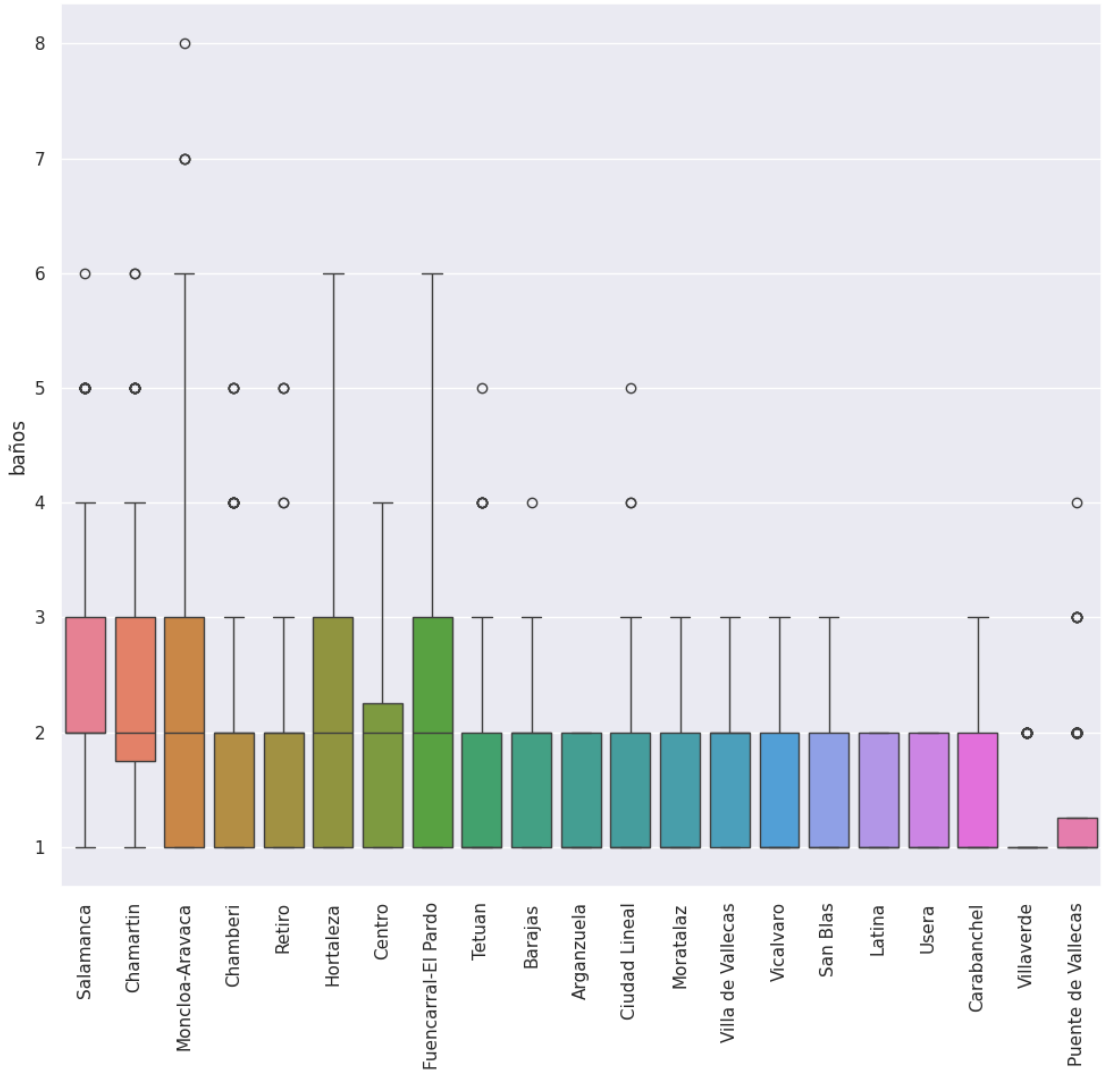

Figura 12 Diagrama de caja del número de baños de las viviendas sin valores atípicos extremos

En la figura 12 observamos boxplots del número de baños de las viviendas agrupadas por distritos, y habiéndose eliminado los valores extremadamente atípicos. Como podemos ver, el número de baños de las viviendas en la figura 8 antes estaba comprendido en un rango de 1 a 12 baños y ahora en la figura 12 está comprendido en un rango de 1 a 8 baños. No observamos prácticamente ningún valor atípico y los que se observan son valores "débilmente atípicos", y se espera que estas observaciones no afecten al rendimiento de los modelos de aprendizaje supervisado.

En la tabla 8, podemos ver cómo cambia el número de instancias a medida que vamos eliminando valores atípicos extremos. En esta tabla aparece el número de instancias que tenemos tras eliminar los valores atípicos extremos de la variable numérica considerada. Por ejemplo, tras eliminar los valores extremadamente atípicos de la variable *precio,* tenemos 2794 instancias; es decir, se han eliminado 124 instancias. Y repetimos este proceso para las variables *superficie construida y habitaciones,* donde se eliminan 15 y 18 instancias, respectivamente.

	Filas	Diferencia
Original	2918	-
Precio	2794	124
Superficie Construida	2779	15
Habitaciones	2761	18

Tabla 8 Valores atípicos extremos eliminados y nuevas dimensiones de los datos

Una vez que se han manejado los valores atípicos del conjunto de datos, se procede a realizar un análisis exploratorio básico de los datos.

3.2.2. Análisis Exploratorio Básico de los Datos

Comenzamos el análisis exploratorio básico de los datos obteniendo las dimensiones del conjunto de datos; es decir, el número de instancias (viviendas) y el número de columnas (variables, atributos o características). En concreto, estamos trabajando con 2761 viviendas sobre las que se han medido 14 características, como podemos ver en los resultados de la tabla 9.

	Filas	Columnas
Dimensiones	2761	14

Tabla 9 Dimensiones del conjunto de datos

El siguiente paso es realizar un análisis de las variables examinando si contienen valores faltantes o no, cuyos resultados podemos ver en la tabla 10. Obtenemos que hay algunas viviendas en las que no aparece toda la información necesaria y se obtienen valores faltantes. Esto es algo totalmente normal porque debemos recordar que el conjunto de datos se ha extraído de una página web; es decir, son datos reales. Por lo tanto, es muy probable que a los anunciantes de las viviendas, ya sean los propietarios o las inmobiliarias, se les haya olvidado poner algunas características de los inmuebles en el anuncio de la vivienda. Como vemos, obtenemos valores faltantes para las variables *baños, habitaciones y superficie_construida.* Ya hemos identificado las variables que poseen valores faltantes, pero de momento no hay que tratar con ellos, pues esta es una tarea que deberemos realizar durante la fase de preproceso.

Características	Valores Faltantes
tipo_vivienda	0
distrito	0
precio	0
piscina	0
terraza	0

jardin	0
garaje	0
trastero	0
calefaccion	0
aire_acondicionado	0
ascensor	0
superficie_construida	3
habitaciones	98
baños	103

Tabla 10 Valores faltantes

La siguiente acción a realizar es obtener el tipo de variables con las que estamos trabajando, lo cual podemos ver en la tabla 11. A priori, tenemos 2 variables categóricas definidas como “object” y 12 variables numéricas, definidas como “int64” y “float64”. Sin embargo, sabemos que el número de variables categóricas y numericas es diferente, pues en la ETL hemos cambiado las variables binarias, que tenían clases “no” y “si” a 0 y 1, luego aparecen como variables numéricas (int64) en la tabla 11. Es decir, realmente tenemos 10 variables categóricas y 4 variables numéricas, pero nos falta ver si son variables categóricas multiclase o binarias, y numéricas discretas o continuas.

Característica	**Tipo de Variable**
tipo_vivienda	object
distrito	object
precio	int64
piscina	int64
terraza	int64
jardin	int64
garaje	int64
trastero	int64
calefaccion	int64
aire_acondicionado	int64
ascensor	int64
superficie_construida	float64
habitaciones	float64
baños	float64

Tabla 11 Tipo de variables

Hemos identificado que tenemos 4 variables numéricas que son el precio, la superficie construida, el número de habitaciones y el número de baños de la vivienda. Y también hemos identificado que tenemos 10 variables categóricas, de las cuales 2 son variables categóricas multiclase, el distrito y el tipo de vivienda. Las restantes 8 variables categóricas son binarias y verifican si la vivienda posee o no una característica determinada y estas son el ascensor, el trastero, el garaje, el aire acondicionado, la calefacción, la terraza, el jardín y la piscina.

Para ver si las variables numéricas son discretas o continuas, realizamos un resumen básico de las características, que podemos ver en la tabla 12. En esta tabla vienen resumidas algunas características importantes de las variables como el número de valores no faltantes, la media, la desviación típica, el mínimo, el máximo y los percentiles. Si analizamos bien esta tabla, podemos ver que las variables *habitaciones* y *baños* son variables numéricas discretas pues están definidas en un intervalo reducido de valores enteros y positivos. En particular, el número de habitaciones está limitado entre 1 y 9, y el número de baños entre 1 y 8. Por otro lado, las variables *precio* y *superficie_construida* son variables numéricas continuas, pues toman valores enteros y positivos, pero en un rango mucho más amplio. En particular, el *precio* está definido entre 60.353 y 5.000.000 de euros, y la *superficie_construida* entre 14 y 714 metros cuadrados. Por tanto, tras la descripción básica de las variables numéricas, podemos concluir que hay 2 variables numéricas continuas y 2 variables numéricas discretas.

Operación	**precio**	**superficie**	**habitaciones**	**baños**
count	2.761	2.758	2.663	2.658
mean	610.504	110,22	2,67	1,82
std	667.625	74,42	1,09	1,02
min	60.353	14	1	1
max	5.000.000	714	9	8
25%	210.000	65	2	1
50%	369.000	89	3	2
75%	742.350	128	3	2

Tabla 12 Descripción básica de las variables numéricas

Como podemos ver en la tabla 12, las variables numéricas varían bastante en rango lo cual puede afectar a ciertos modelos de aprendizaje automático. Esta será una de las tareas o problemas con los que deberemos lidiar en la fase del preprocesamiento de los datos.

Una vez realizada la descripción básica de las variables, podemos pasar a visualizar los datos mediante técnicas de análisis univariante y multivariante.

3.2.3. Visualización de los Datos

Una vez que ya sabemos con qué tipo de variables estamos tratando, realizamos un histograma de las variables numéricas para ver su distribución, el cual podemos ver en la figura 13. En general, todas las variables numéricas tienen una distribución asimétrica positiva que se podría asemejar a una distribución Gamma, pero para determinar esto con certeza habría que profundizar realizando pruebas estadísticas y contrastes de hipótesis y ese no es el propósito de este trabajo. Lo que sí podemos extraer de este gráfico es que podemos agrupar alrededor del 90% de las viviendas en unos intervalos de características bastante reducidos. Podemos decir con casi total seguridad que la mayoría de las viviendas tiene entre 1 y 4 habitaciones, entre 1 y 3 baños, un precio menor que 1 millón de euros, y una superficie construida menor que 200 metros cuadrados.

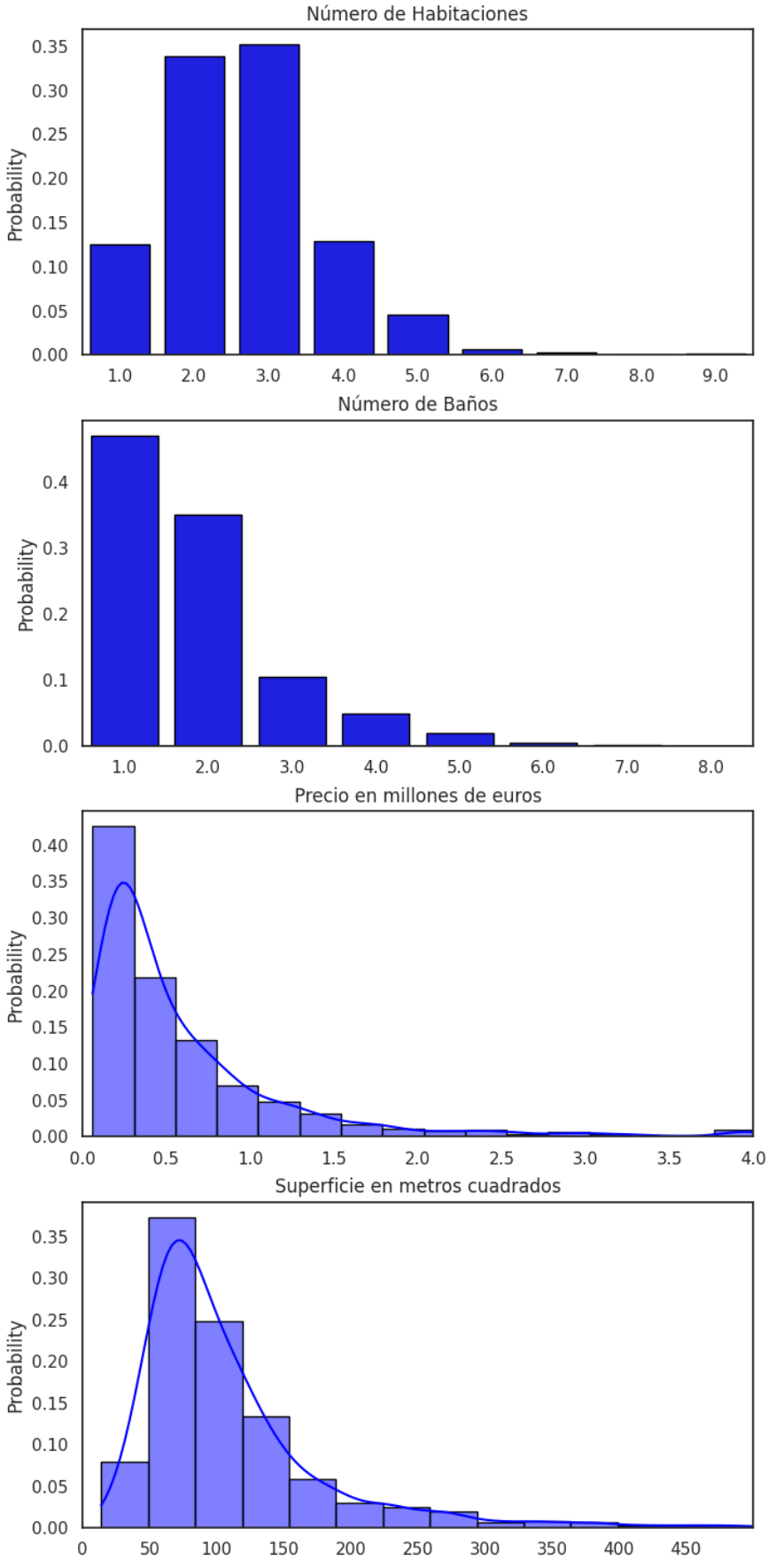

Figura 13 Histograma de las variables numéricas

En la figura 14, podemos ver los histogramas que se han realizado para las variables categóricas multiclase. En particular, para la distribución del *tipo de vivienda*, podemos extraer que aproximadamente el 90% de las viviendas son pisos, mientras que la proporción de dúplex, casas, chalets, áticos y apartamentos es del 10% restante. También podemos ver que la distribución de la variable *distrito* es variada, aunque es cierto que hay distritos como Salamanca y Centro, que tienen una mayor proporción de viviendas ofertadas, alrededor del 11%, y otros como Barajas y Moratalaz, cuya proporción no llega al 2%.

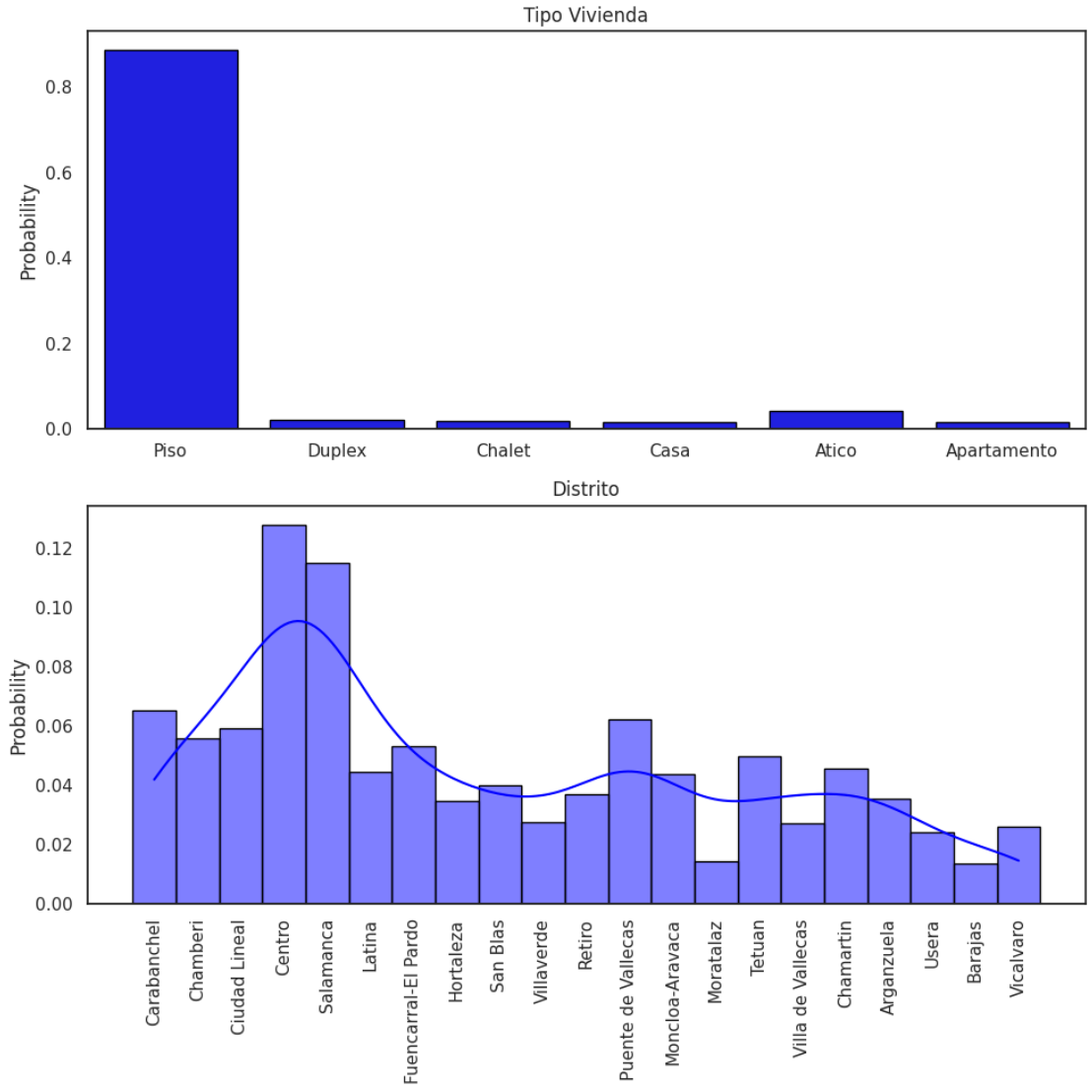

Figura 14 Histograma de las variables categóricas multiclase

Para obtener una idea de la cantidad de viviendas que se ofrecen en cada distrito, podemos ver una visualización en la figura 15. En esta figura podemos ver el tamaño de cada distrito así como su posición geográfica respecto al centro de la ciudad y el número de viviendas ofertadas. Como se puede observar, la mayoría de las viviendas ofertadas se concentran en los distritos más céntricos, mientras que los distritos que se encuentran en la periferia tienen una menor oferta de viviendas, pese a su mayor tamaño. Por ejemplo,

si comparamos el distrito de Villa de Vallecas y el de Salamanca, vemos que el distrito de Villa de Vallecas es mucho más extenso en cuanto a superficie, pero la oferta de viviendas es mucho menor. Esto puede deberse a que el distrito de Villa de Vallecas tiene un menor número de viviendas construido, pues es un distrito bastante nuevo. También se observa otra diferencia bastante notable en la posición geográfica, pues tenemos una mayor frecuencia de viviendas ofertadas al norte y al oeste de la ciudad que al sur y al este. Para ver esta diferencia, contémplense por ejemplo los distritos de Fuencarral-El Pardo y Villa de Vallecas o Vicálvaro.

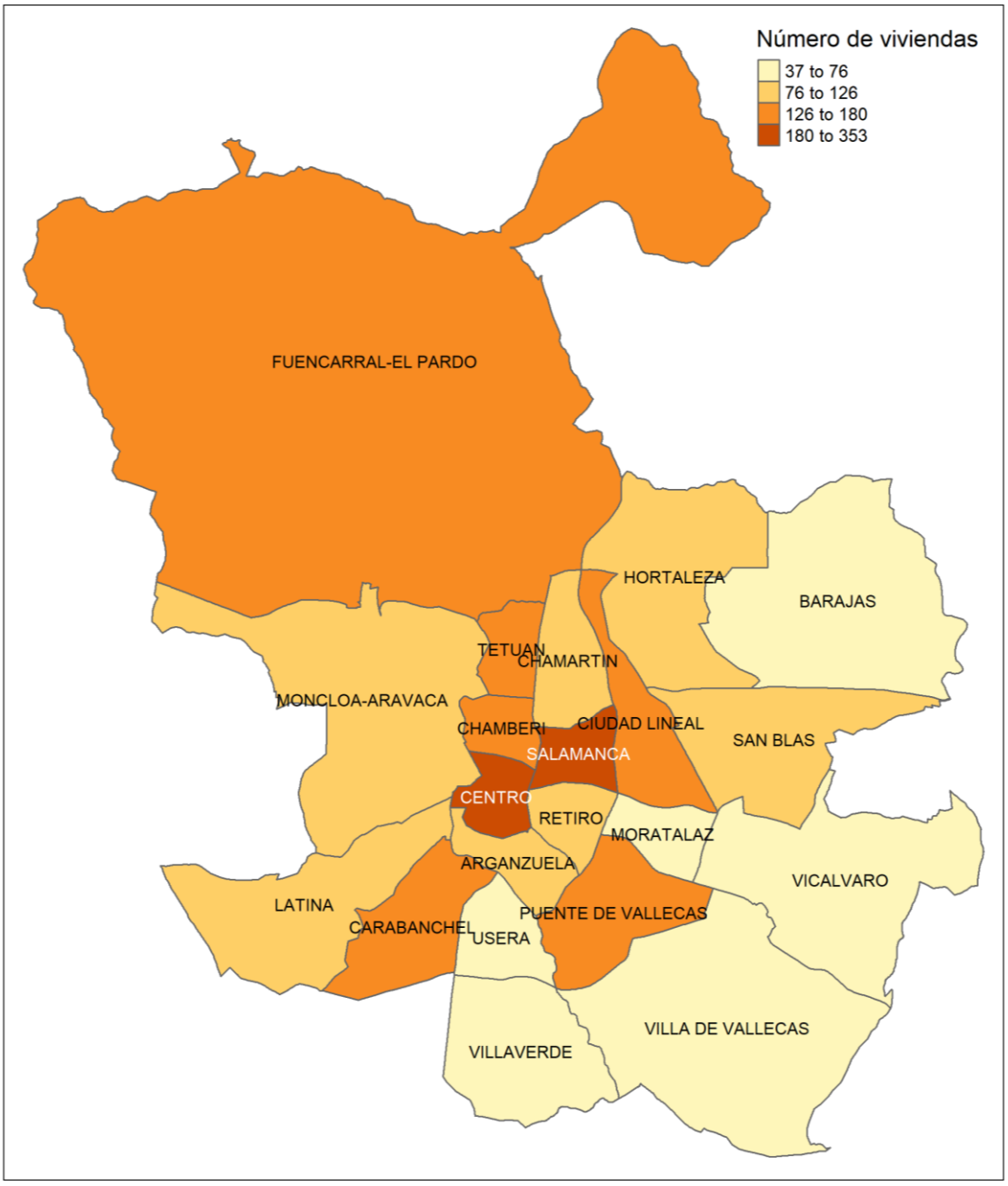

Figura 15 Mapa de calor de las viviendas por distrito

Después de analizar los histogramas para las variables numéricas y categóricas multiclase, ahora observamos los gráficos de barras para las variables categóricas binarias en la figura 16. Como se puede observar en los gráficos de barras, las 8 variables categóricas del conjunto de datos son variables binarias que toman valor 0 o 1 dependiendo de si la vivienda posee una característica o no. De estos gráficos, podemos extraer aproximadamente que un 45% de las viviendas poseen aire acondicionado, un 60% calefacción y ascensor, un 20% garaje y trastero, un 45% terraza y un 10% jardín y piscina.

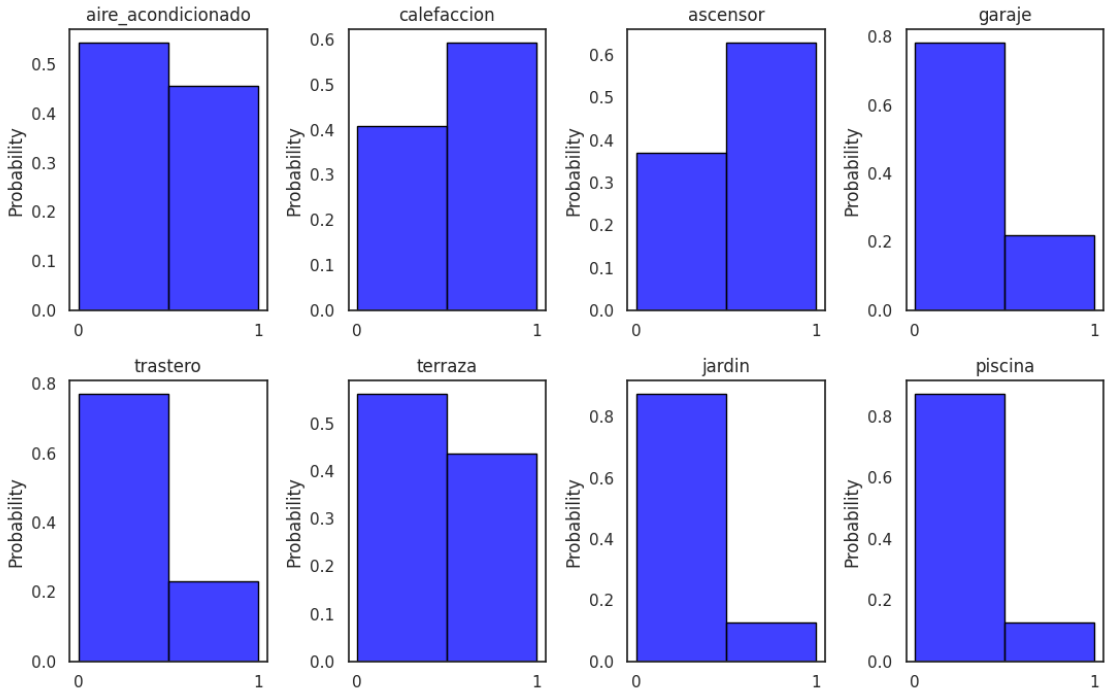

Figura 16 Histograma de las variables categóricas

Tras haber analizado los histogramas de todas las variables, pasamos a ver las posibles relaciones que hay entre ellas. Podemos alcanzar dicho propósito calculando la matriz de correlaciones de Pearson, que podemos ver en la figura 17. Nos interesan principalmente las relaciones entre nuestra variable respuesta *precio* y las variables predictoras. Todas las variables binarias tienen un coeficiente de correlación con la variable respuesta positivo pero que en ningún caso llega a 0.3, por lo cual las variables están muy poco correladas. Las variables *superficie construida* y *baños* tienen un coeficiente de correlación con la variable respuesta alrededor de 0.8, con lo cual las variables están altamente correladas. Por su parte, la variable *habitaciones* tiene un coeficiente alrededor del 0.5, lo cual nos indica una correlación moderada.

Por otro lado, nos podrían interesar las relaciones entre las características *baños, habitaciones* y *superficie construida,* que son variables dependientes. Es decir, si se aumenta la superficie construida, el comportamiento típico es que aumente el número de habitaciones y baños, pues hay un mayor espacio para la construcción. Observamos que

tienen unos coeficientes de correlación positivos bastante moderados e incluso altamente correlados, en el rango de 0.6 a 0.8. Esto simplemente refuerza la idea de que existe una cierta dependencia entre estas características. Sin embargo, la relación no es tan fuerte como para decir que alguna de estas variables no sea significativa por ser una combinación lineal de las otras, con lo cual se mantienen todas las características.

Por último, analizamos las relaciones entre las variables *piscina, jardín, garaje* y *trastero* y obtenemos unos coeficientes de correlación moderados entre 0.3 y 0.6. Parece lógico que si una vivienda posee piscina, entonces también puede poseer jardín o que si una vivienda tiene trastero, entonces puede poseer garaje. Estas características como piscina o trastero son en este sentido secundarias y los atributos como jardín o garaje se consideran principales. Es decir, una vivienda puede poseer jardín y no tiene por qué poseer piscina, pero si consideramos la situación inversa, el comportamiento normal es que si una vivienda tiene piscina, entonces también tendrá jardín.

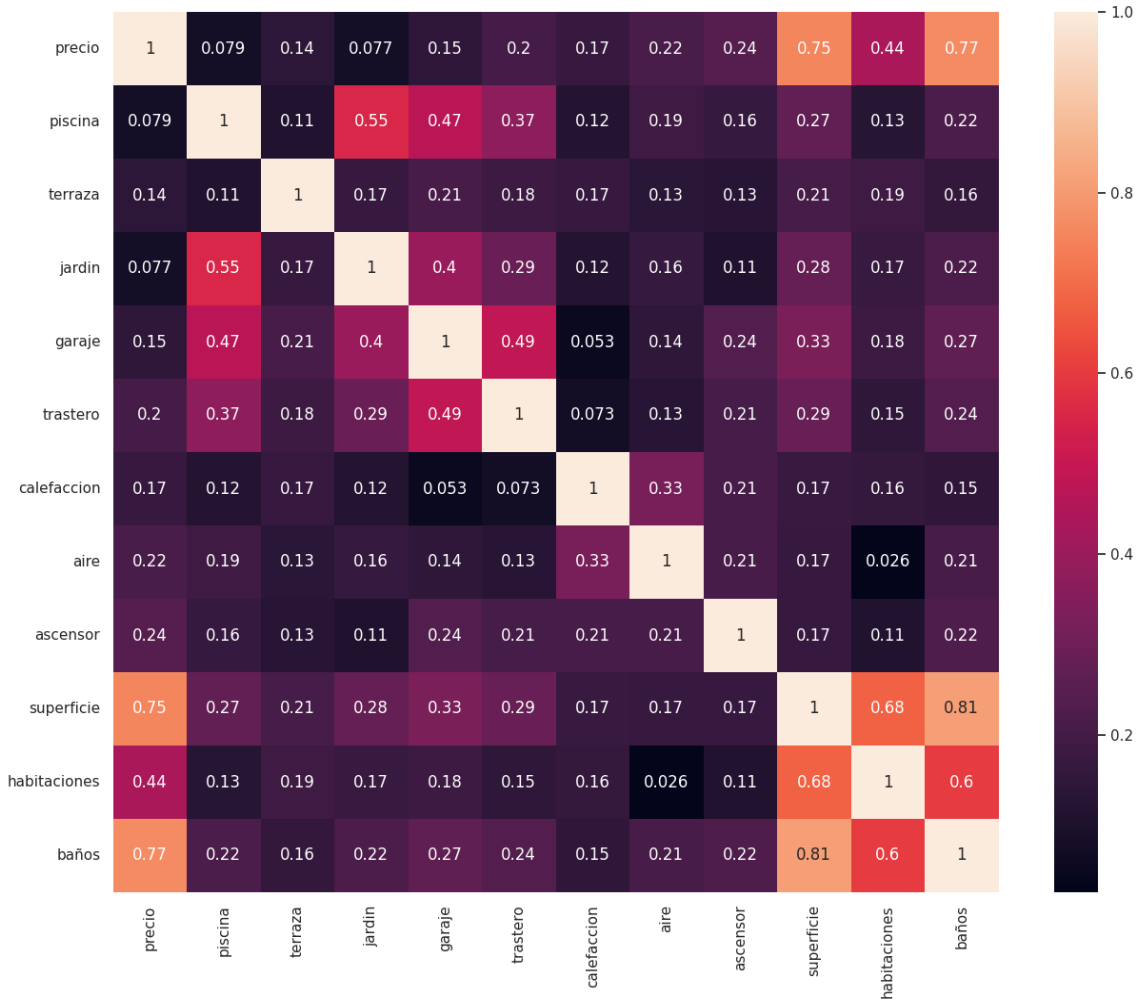

Figura 17 Matriz de correlaciones de Pearson

Ahora vamos a visualizar gráficamente el precio medio y mediano de las viviendas según el distrito que consideremos, que podemos ver en la figura 18. Como vemos hay bastantes diferencias en los precios de las viviendas entre los distritos, dándose la máxima

diferencia entre los distritos de Salamanca y Puente de Vallecas. El precio medio de una vivienda en Salamanca es aproximadamente 7 veces más caro que el precio medio de una vivienda en Puente de Vallecas. Sin embargo, esta información no es del todo relevante porque las características de las viviendas de Salamanca probablemente son muy diferentes a las de Puente de Vallecas.

La información más relevante que podemos extraer de este gráfico es la diferencia que hay entre el precio medio y el precio mediano de las viviendas por distrito. En el distrito Salamanca, vemos que el precio mediano de una vivienda es de 1 millón de euros, mientras que el precio medio es de 1.4 millones de euros. Por otro lado, en el distrito Puente de Vallecas, el precio mediano de una vivienda prácticamente se solapa con el precio medio, alrededor de 200.000 euros. La realidad del distrito Salamanca es que el precio de las viviendas se encuentra alrededor de 1 millón de euros; es decir, utilizamos la mediana como medida de centralidad. Utilizar la media en este caso resultaría menos correcto pues este no sería el precio de una vivienda seleccionada de forma aleatoria en este distrito, ya que la media se ve muy afectada por los valores atípicos, a pesar de que se han eliminado los valores extremadamente atípicos.

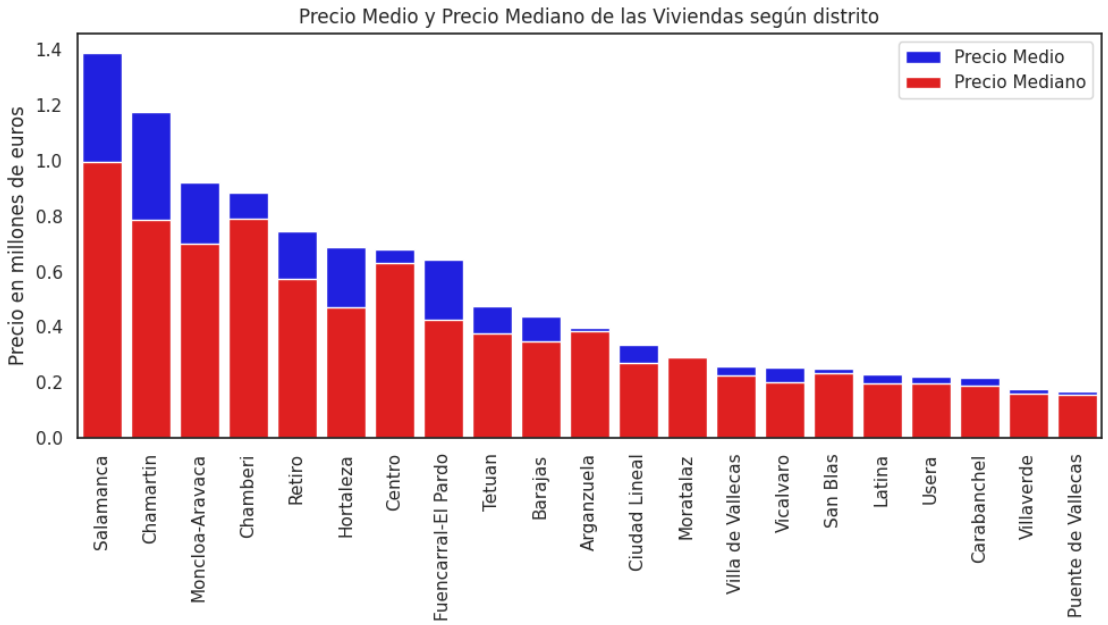

Figura 18 Precio medio y mediano de las viviendas según el distrito

El siguiente paso es la realización de un gráfico de dispersión entre las variables numéricas del conjunto de datos, que podemos ver en la figura 19. En este gráfico, podemos ver la dispersión de las variables así como un histograma de las mismas, el cual hemos analizado previamente. En muchos gráficos observamos bastantes puntos que se pueden trazar en la misma línea, porque debemos recordar que las variables *baños* y *habitaciones* son numéricas discretas. Entonces, para un mismo número de baños o habitaciones, hay valores que varían mucho como el precio o la superficie construida, que recordemos son variables continuas. De estos gráficos podemos extraer mucha información sobre las variables y su comportamiento. En este caso, ya conocemos

bastante bien cómo se comportan las variables, pero en otro caso con variables peor definidas o con datos más complejos, nos resultaría de mucha utilidad.

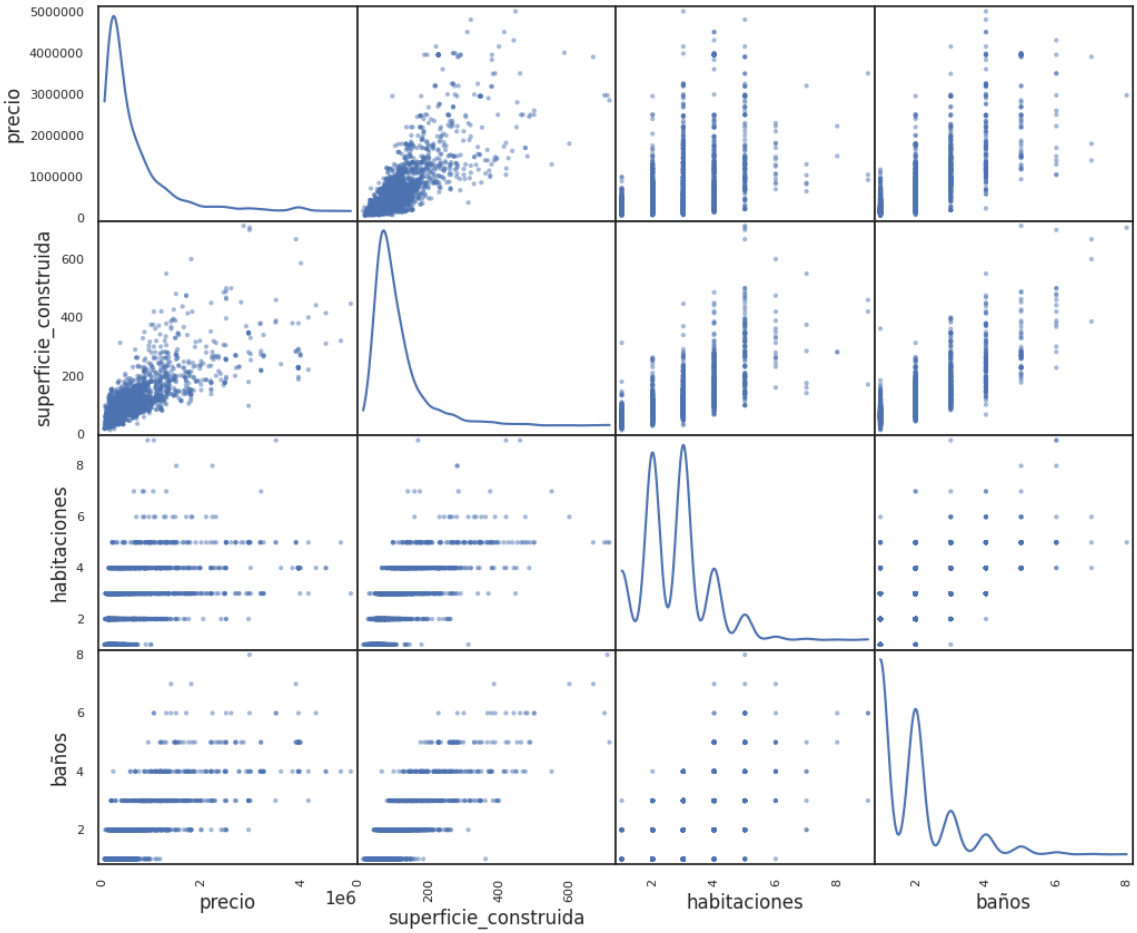

Figura 19 Diagrama de dispersión de las variables numéricas

La última parte del EDA es la realización de los gráficos qq-plot que podemos ver en la figura 20. Estos gráficos nos indican si las variables numéricas se ajustan a una distribución normal. Aunque es cierto que se deberían realizar las respectivas pruebas y contrastes de hipótesis, podemos establecer casi con total certeza que las variables no se aproximan a una distribución normal. Esto viene respaldado por los histogramas que hemos observado en la figura 13, en los que veíamos que las variables se ajustaban más a una distribución gamma. Aunque, de nuevo habría que realizar los contrastes de hipótesis correspondientes.

Una vez finalizada la etapa del EDA, podemos pasar a realizar el preprocesamiento de los datos para poder aplicar las técnicas de aprendizaje automático.

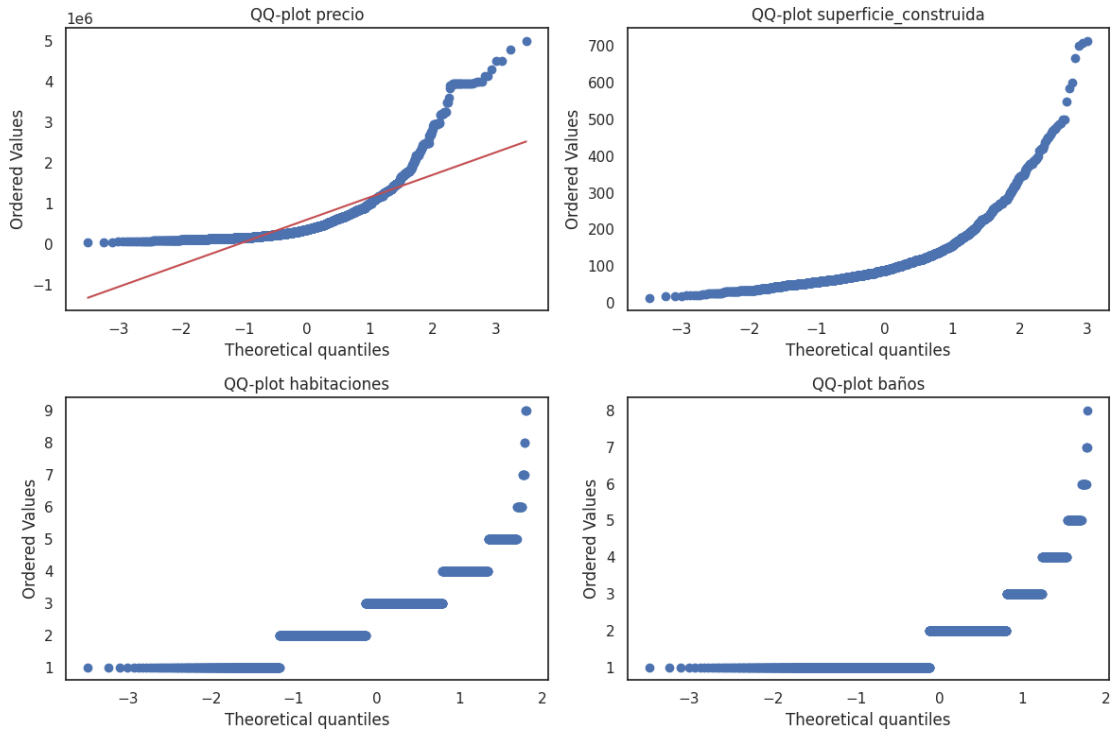

Figura 20 QQ-Plot de las variables numéricas

3.3. Preprocesamiento de los datos

El preprocesamiento de los datos es la siguiente tarea que tenemos que realizar después de haber completado el EDA. En este caso, en el EDA hemos identificado que tenemos algunas variables que presentan valores faltantes, que las variables numéricas tienen escalas de medición diferentes y que hay algunas variables categóricas multiclase.

La primera parte del preproceso es el tratamiento de los valores faltantes y por lo general hay 2 estrategias: la eliminación de valores faltantes o la imputación de valores faltantes. La eliminación de valores faltantes comprende 2 casos: podemos eliminar las columnas con un porcentaje de valores faltantes superior a un umbral, por ejemplo del 60%, o podemos eliminar las filas que tengan valores faltantes. La imputación por su parte puede ser de 2 tipos: imputación univariante e imputación multivariante.

La imputación univariante generalmente sustituye los valores faltantes por la media o mediana del atributo si la variable es numérica y por la moda si la variable es categórica. Aunque, también se puede sustituir por un valor aleatorio del conjunto de datos que siga una distribución normal con media y desviación típica calculada a partir del atributo. Por su parte, la imputación multivariante sustituye los valores faltantes por un modelo construido a partir del resto de características. La imputación multivariante suele ser la mejor solución para el tratamiento de los valores faltantes porque considera el resto de características del conjunto de datos. Los métodos de imputación univariante no suelen ser adecuados porque sustituyen el valor faltante por un valor fijo o aleatorio sin tener en cuenta el resto de características.

Vamos a ver un ejemplo que nos ayude a ver por qué un método es mejor que el otro. Supongamos un caso en el que conocemos que el número medio de habitaciones de las viviendas es 2 y que tenemos 2 viviendas con valores faltantes para este atributo. Si utilizamos imputación univariante, entonces 2 viviendas sin información sobre el número de habitaciones, una con 700 metros cuadrados de superficie construida y con 6 baños, y otra con 22 metros cuadrados y 1 baño, tendrán el mismo número de habitaciones, lo cual no parece tener demasiado sentido. En cambio, si utilizamos imputación multivariante con un modelo KNN, podremos obtener que la primera vivienda está muy cerca de viviendas que tienen 5 habitaciones y que la segunda vivienda es cercana a viviendas con 1 única habitación.

Como métodos para tratar con valores faltantes hemos justificado que la imputación multivariante por lo general es mejor que la imputación univariante. Ahora la pregunta a contestar sería: ¿qué método es mejor: eliminar valores faltantes o imputar valores faltantes? Si se tienen pocos valores faltantes y la eliminación de estos no supone consecuencias negativas graves en el modelo, entonces se pueden eliminar los valores faltantes. Pero, en la mayoría de casos lo mejor es mantener las instancias con valores faltantes pues con la imputación multivariante, se pueden sustituir estos valores y se pueden lograr resultados muy satisfactorios. En este caso, se ha decidido aplicar imputación multivariante con el modelo de los k vecinos más cercanos (KNN) pues se obtienen resultados de gran calidad.

Después de haber llegado a un método para manejar los valores faltantes, pasamos a la próxima tarea del preproceso. La siguiente tarea consiste en tratar con las variables numéricas, pues tienen escalas de medición diferentes. Si no gestionamos estas variables, podríamos introducir sesgos en nuestros modelos predictivos, lo que afectaría a la interpretación de los resultados y la precisión de las predicciones. Podemos resolver este problema aplicando la técnica de estandarización de los datos, cuya expresión viene definida en (2.2). Con esta transformación de las variables numéricas, los datos siguen una distribución con media 0 y desviación típica 1. En principio, ya habríamos acabado con el preproceso de las variables numéricas y podríamos pasar al preproceso de las variables categóricas.

En las variables categóricas normalmente tenemos que realizar tareas como el tratamiento de valores faltantes, pero en este caso no tenemos valores faltantes. La única acción que debemos realizar es la codificación binaria, pues tenemos 2 variables categóricas multiclase. La variable *distrito* tiene 21 categorías, luego se crearán 21 variables binarias de acuerdo con la expresión definida en (3.3). Para la variable *tipo_vivienda* se procede de forma similar, y dado que tiene 6 categorías, entonces se crearán 6 variables binarias.

$$I_{ij} = \begin{matrix} 1 & si\ \ la\ vivienda\ i\ pertenece\ al\ distrito\ j \\ 0 & en\ caso\ contrario \end{matrix} \qquad (3.3)$$

También se podrían haber creado n – 1 variables binarias para las n categorías de las variables multiclase. Es decir, para la variable distrito que tiene 21 clases, se crearían 20 variables binarias y la variable que no se incluye se manifestaría cuando las 20 variables binarias tomasen valor 0. Pero si utilizásemos este método, entonces no podríamos

conocer la importancia de la categoría excluida para los modelos de aprendizaje supervisado. Es por ello, que se decide crear 1 variable dummy para cada clase o categoría de la variable categórica multiclase.

Todos estos pasos y tareas de preprocesamiento de los datos los decidimos realizar en una tubería o pipeline. Este pipeline contiene las actividades del preproceso y es el que debemos usar para aplicar las técnicas de aprendizaje automático. Además, este pipeline nos va a permitir ajustar los hiperparámetros de los modelos de aprendizaje supervisado, sin tener que aplicar el preproceso directamente a los datos. Una vez definido el preproceso, podemos pasar a aplicar las técnicas y modelos de aprendizaje supervisado.

3.4. Aprendizaje Supervisado

La siguiente fase del trabajo es el entrenamiento y validación de los modelos de aprendizaje supervisado. Para realizar esto, primero debemos dividir los datos en 2 conjuntos, uno que se va a utilizar para el entrenamiento del modelo y otro para la validación. Para todos los modelos de aprendizaje supervisado se va a utilizar tanto validación interna para el ajuste de hiperparámetros como validación externa para el entrenamiento del modelo. Para la validación interna utilizamos la validación cruzada con k = 10 folds y para la validación externa usamos el método holdout = 3/4, en el cual 3/4 de los datos se dedican al entrenamiento del modelo y 1/4 a la validación. Para el ajuste de hiperparámetros se considera el “Grid Search” o la búsqueda en rejilla y el “Random Search” o la búsqueda aleatoria. Si se necesitase más información sobre el ajuste de hiperparámetros, se puede encontrar explicado más detalladamente en la sección 2.1.7.

Una vez que se ha definido la validación interna y externa, debemos establecer para los modelos que se van a entrenar, los hiperparámetros que se van a ajustar. Recordemos que los modelos que se consideran son el modelo de los k vecinos más cercanos, el árbol de decisión, las máquinas de vectores de soporte con el kernel lineal y radial, los métodos de conjunto con el Random Forest y el Gradient Boosting, y la red neuronal de perceptrón multicapa. En las tablas 13 a 19 podemos ver los resultados de cada modelo de aprendizaje supervisado que se ha entrenado utilizando para el ajuste de hiperparámetros la búsqueda en rejilla y la búsqueda aleatoria, respectivamente. Estas tablas contienen información sobre las métricas de evaluación del modelo para los datos de entrenamiento y evaluación, sobre los mejores valores de los hiperparámetros y sobre el tiempo de ejecución de cada modelo.

Para el modelo de los k vecinos más cercanos, decidimos ajustar entre 2 y 10 vecinos y consideramos la distancia euclídea, la de Manhattan y la de Minkowski. Podemos visualizar los resultados del ajuste de hiperparámetros para el KNN en la tabla 13. Tanto con la búsqueda en rejilla como con la búsqueda aleatoria de hiperparámetros, obtenemos el mismo resultado; obtenemos que la mejor combinación de hiperparámetros es utilizar la distancia de Manhattan y considerar los 4 vecinos más cercanos a la instancia del conjunto de validación. En cuanto al tiempo de ejecución, éste varía entre 11 y 29 segundos, dependiendo de si utilizamos la búsqueda aleatoria o en rejilla. Aunque el

tiempo de ejecución es bastante corto, nos decantamos por el "Random Search" pues encuentra la mejor combinación de hiperparámetros prácticamente 3 veces más rápido.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	metric: manhattan n_neighbors: 4	0,8091	0,8156	29,06
Random Search	metric: manhattan n_neighbors: 4	0,8091	0,8156	10,84

Tabla 13 Resultados del modelo de KNN

Para el árbol de decisión, ajustamos valores entre 2 y 19 para la profundidad máxima del árbol, y entre 10 y 29 para el número mínimo de instancias que debe haber en cada nodo para ser subdividido. Podemos visualizar los resultados del ajuste de hiperparámetros para el árbol de decisión en la tabla 14. Con la búsqueda en rejilla obtenemos que los mejores valores de los hiperparámetros son establecer una profundidad máxima de 19 y considerar mínimo 27 instancias para subdividir el árbol. Mientras que con la búsqueda aleatoria de hiperparámetros, obtenemos 15 de profundidad máxima y 21 instancias para subdividir el árbol. En cuanto al tiempo de ejecución, éste varía entre 10 y 330 segundos, dependiendo de si utilizamos la búsqueda aleatoria o en rejilla. Los resultados de las métricas de evaluación del modelo no cambian significativamente de un método a otro, el R^2 se encuentra alrededor de 0.80. En este caso, claramente nos decantamos por el "Random Search" pues encuentra una buena combinación de hiperparámetros 33 veces más rápido y además, generaliza mejor para los datos de validación.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	max_depth: 19 min_samples_split: 27	0,7904	0,8064	329,13
Random Search	max_depth: 15 min_samples_split: 21	0,7875	0,8178	9,26

Tabla 14 Resultados del modelo de árbol de decisión

Para las máquinas de vectores de soporte, ajustamos valores entre 100.000 y 1.000.000 para el coeficiente de penalización y entre 0.01 y 1 para el hiperparámetro gamma. Podemos visualizar los resultados del ajuste de hiperparámetros para el kernel lineal en la tabla 15. Obtenemos el mismo resultado con la búsqueda en rejilla y la búsqueda aleatoria de hiperparámetros; obtenemos que el mejor valor del coeficiente de penalización es 1.000.000. En cuanto al tiempo de ejecución, éste no varía prácticamente

entre los 2 métodos situándose en 64 segundos, por lo que a priori no importaría demasiado qué enfoque utilizar.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	C: 1000000	0,7169	0,7463	63,79
Random Search	C: 1000000	0,7169	0,7463	64,18

Tabla 15 Resultados del modelo de máquinas de vectores de soporte con kernel lineal

Por su parte, podemos visualizar los resultados del ajuste de hiperparámetros para el kernel radial en la tabla 16. Para el kernel radial obtenemos que la mejor combinación de hiperparámetros con la búsqueda en rejilla es un coeficiente de penalización de 1.000.000 y un gamma de 0.1, mientras que con la búsqueda aleatoria obtenemos un coeficiente de 500.000 y un gamma de 0.032. En cuanto al tiempo de ejecución, éste varía entre 53 y 174 segundos, dependiendo de si utilizamos la búsqueda aleatoria o en rejilla. En este caso, habría un "tradeoff" o intercambio entre qué se prioriza más: obtener una mejor métrica de evaluación del modelo u obtener un tiempo de ejecución más corto. En este caso, nos interesa más obtener una mejor métrica de evaluación del modelo pues el tiempo de ejecución no es demasiado elevado, no supera los 3 minutos. Por lo que, optaríamos por utilizar la búsqueda en rejilla que nos permite obtener la mejor combinación de los hiperparámetros.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	C: 1000000 gamma: 0.1	0,8416	0,8666	173,17
Random Search	C: 500000 gamma: 0.032	0,8077	0,84	52,75

Tabla 16 Resultados del modelo de máquinas de vectores de soporte con kernel radial

Para el Random Forest, ajustamos valores entre 10 y 200 modelos base, y entre el 10% y el 75% de características utilizadas para crear cada modelo base. Podemos visualizar los resultados del ajuste de hiperparámetros para el Random Forest en la tabla 17. Obtenemos que la mejor combinación de hiperparámetros para la búsqueda en rejilla es 200 modelos base y un 25% de características para crear cada modelo base, mientras que con la búsqueda aleatoria obtenemos 50 modelos base y 50% de características. En cuanto al tiempo de ejecución, éste varía entre 44 y 93 segundos, dependiendo de si utilizamos la

búsqueda aleatoria o en rejilla. En este caso, dado que se obtienen métricas de evaluación de entrenamiento y de test muy similares con los 2 enfoques, nos inclinaríamos por la búsqueda aleatoria, pues es más rápido.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	max_features: 0.25 n_estimators: 200	0,8369	0,8734	92,21
Random Search	max_features: 0.50 n_estimators: 50	0,836	0,8686	43,63

Tabla 17 Resultados del modelo de Random Forest

Para el Gradient Boosting, ajustamos valores entre 10 y 200 modelos base, y entre 0.01 y 0.50 para la velocidad de aprendizaje del modelo. Podemos visualizar los resultados del ajuste de hiperparámetros para el Random Forest en la tabla 18. Obtenemos que la mejor combinación de hiperparámetros para la búsqueda en rejilla es 50 modelos base y una velocidad de aprendizaje de 0.10, mientras que con la búsqueda aleatoria obtenemos 10 modelos base y una velocidad de aprendizaje de 0.30. En cuanto al tiempo de ejecución, éste varía entre 26 y 41 segundos, dependiendo de si utilizamos la búsqueda aleatoria o en rejilla. En este caso, dado que se obtienen métricas de evaluación de entrenamiento y de test muy similares con los 2 enfoques, nos inclinaríamos por la búsqueda aleatoria, pues es más rápido.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Grid Search	learning_rate: 0.1 n_estimators: 50	0,8249	0,8585	40,02
Random Search	learning_rate: 0.3 n_estimators: 10	0,8155	0,8468	25,13

Tabla 18 Resultados del modelo de Gradient Boosting

Para la red neuronal perceptrón multicapa, ajustamos valores entre (64,32) y (256,128) para el tamaño de las capas ocultas y entre 1.000 y 2.000 para el número máximo de iteraciones. Podemos visualizar los resultados del ajuste de hiperparámetros para el Random Forest en la tabla 19. En este caso, se realiza únicamente “Random Search” porque el tiempo de ejecución se elevaría enormemente si utilizásemos el “Grid Search”. Con la búsqueda aleatoria obtenemos que los mejores valores de los hiperparámetros son 2 capas ocultas con 256 y 128 neuronas, y con hasta 2.000 iteraciones. El tiempo de

ejecución de este modelo es exageradamente elevado en comparación con los otros modelos, pues casi llega a los 1.500 segundos; es decir, más de 24 minutos.

Ajuste	Hiperparámetros	R2 Train	R2 Test	Tiempo
Random Search	max_iter: 2000 hidden_layer_sizes: (256, 128)	0,8221	0,8484	1.487,52

Tabla 19 Resultados del modelo de red neuronal perceptrón multicapa

Las tablas 20 y 21 son muy interesantes pues contienen la información necesaria para decidir qué modelo de aprendizaje supervisado es mejor. Para tomar la decisión, debemos fijarnos en las métricas de evaluación para los datos de entrenamiento y de test y en el tiempo de ejecución del modelo. Por lo general, el enfoque que ofrece la búsqueda en rejilla es poco realista y, aunque siempre obtenemos la mejor combinación de hiperparámetros, el tiempo de ejecución es bastante más elevado en comparación con la búsqueda aleatoria. Lamentablemente, en el mundo real no tenemos tiempo infinito para probar todas las posibles combinaciones de hiperparámetros, sino que tenemos recursos limitados. Por lo que debemos probar algunas combinaciones de hiperparámetros y decidir cuál es la mejor; es decir, con qué modelo se obtienen mejores resultados. Este enfoque es precisamente el de la búsqueda de hiperparámetros de forma aleatoria o "Random Search".

Si nos guiamos por los resultados del "Grid Search", obtenemos que el SVM con kernel radial es el modelo con mejor rendimiento, mientras que con el enfoque del "Random Search" obtenemos que éste es el cuarto mejor modelo. En cambio, para el modelo del Random Forest obtenemos que es el segundo mejor modelo si usamos "Grid Search" y el mejor si usamos "Random Search". De estos resultados podemos extraer que el Random Forest, a diferencia del SVM con kernel radial, proporciona unos resultados más consistentes y emplea menor tiempo para el entrenamiento del modelo. El único modelo que se podría comparar con el Random Forest en términos de eficiencia sería el Gradient Boosting, que tiene un tiempo de ejecución menor pero ofrece peores resultados en cuanto al R^2. Como vemos, los modelos de conjuntos son los que tienen un mejor rendimiento, pues ofrecen métricas de evaluación bastante decentes y emplean un periodo de tiempo bastante reducido para el entrenamiento del modelo.

Finalmente, podemos concluir que el mejor modelo es el Random Forest, que obtiene una métrica de evaluación R^2 de aproximadamente 0.84 para los datos de entrenamiento y 0.87 para los datos de validación. Es decir, con nuestro modelo somos capaces de explicar un 84% y un 87% de la variabilidad de la variable respuesta para los datos de entrenamiento y validación, respectivamente. Esto es, con el modelo del Random Forest no solo obtenemos buenos resultados en la etapa de entrenamiento, sino que evitamos el sobreajuste y conseguimos que el modelo generalice muy bien para los datos de validación. Además, el tiempo de ejecución del modelo es relativamente breve, pues no

excede de 1 minuto. Por lo que, para la elección del modelo final nos decantamos por el Random Forest con 50 modelos base y con un 50% de características seleccionadas de forma aleatoria, que se usan para entrenar cada modelo base.

Modelo	**Hiperparámetros**	**R2 Train**	**R2 Test**	**Tiempo**
SVM Kernel Radial	C: 1000000 gamma: 0.1	0,8416	0,8666	173,17
Random Forest	max_features: 0.25 n_estimators: 200	0,8369	0,8734	92,21
Gradient Boosting	learning_rate: 0.1 n_estimators: 50	0,8249	0,8585	40,02
Red Neuronal	max_iter: 2000 hidden_layer_sizes: (256, 128)	0,8221	0,8484	1.487,52
KNN	metric: manhattan n_neighbors: 4	0,8091	0,8156	29,06
Árbol de Decisión	max_depth: 19 min_samples_split: 27	0,7904	0,8064	329,13
SVM Kernel Lineal	C: 1000000	0,7169	0,7463	63,79

Tabla 20 Resultados del ajuste de hiperparámetros con el Grid Search

Modelo	**Hiperparámetros**	**R2 Train**	**R2 Test**	**Tiempo**
Random Forest	max_features: 0.50 n_estimators: 50	0,836	0,8686	43,63
Red Neuronal	max_iter: 2000 hidden_layer_sizes: (256, 128)	0,8221	0,8484	1.487,52
Gradient Boosting	learning_rate: 0.3 n_estimators: 10	0,8155	0,8468	25,13
KNN	metric: manhattan n_neighbors: 4	0,8091	0,8156	10,84

SVM Kernel Radial	C: 500000 gamma: 0.032	0,8077	0,84	52,75
Árbol de Decisión	max_depth: 15 min_samples_split: 21	0,7875	0,8178	9,26
SVM Kernel Lineal	C: 1000000	0,7169	0,7463	64,18

Tabla 21 Resultados del ajuste de hiperparámetros con el Random Search

En la figura 21 podemos observar gráficamente los resultados del entrenamiento y evaluación de los modelos para los métodos de la búsqueda en rejilla y la búsqueda aleatoria de hiperparámetros.

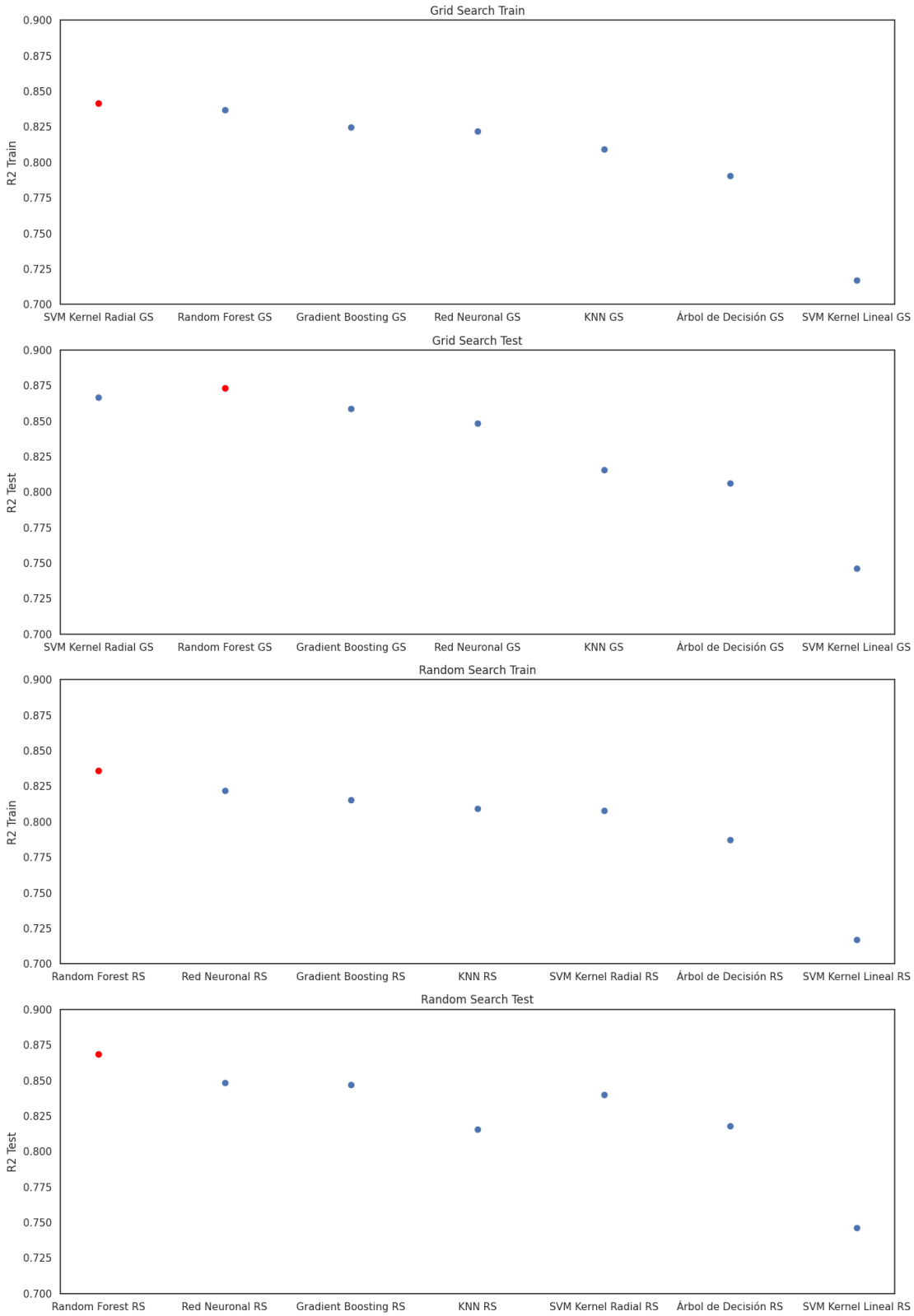

Figura 21 Gráfico comparativo de las métricas de evaluación de entrenamiento y de test y de los métodos de búsqueda en rejilla y búsqueda aleatoria

Aparte del modelo del Random Forest, se entrena el modelo del árbol de decisión con una profundidad máxima de 15 y con 21 instancias en cada nodo para seguir subdividiendo. En el Anexo C podemos encontrar un enlace que contiene un documento en formato pdf

en el que podemos visualizar el árbol de decisión que se crea. Esta visualización nos ayuda a comprender cómo se toman las decisiones en el modelo y qué variables son más importantes para hacer buenas predicciones. Las variables que aparecen en los nodos del árbol son aquellas que el modelo considera más importantes para reducir la varianza y realizar predicciones precisas.

En la tabla 22, podemos ver la importancia de las variables del conjunto de datos que ha sido determinada por el modelo del Random Forest. La importancia de las variables se refiere a cuánto cambia la variabilidad de la variable respuesta dependiendo de si se incluye o no la variable predictora. En nuestro caso, las variables de mayor relevancia de nuestro conjunto de datos son la superficie construida y el número de baños de la vivienda. En la tabla 22 sólo se muestra la importancia de las 10 variables más relevantes, pero en el anexo C se puede ver la importancia de todas y cada una de las variables del conjunto de datos.

Característica	**Importancia**
superficie_construida	0,4783
baños	0,2446
Ciudad Lineal	0,0912
habitaciones	0,0416
Piso	0,0138
Atico	0,0105
trastero	0,0104
calefaccion	0,0104
garaje	0,0101
Duplex	0,0097

Tabla 22 Importancia de las variables predictoras

En la tabla 23 podemos observar algunas de las predicciones del precio de las viviendas que se obtienen con el modelo de Random Forest. Esta tabla también incluye el valor real del precio de las viviendas y las características de la vivienda. En el Anexo C podemos ver todas las predicciones que se han realizado para los datos de validación y su comparación con el valor real de la variable respuesta.

Característica	**Valor**		
# instancia	1.462	600	1.813
precio predicho	437.153	1.208.449	488.980
precio real	469.900	1.275.000	548.000
tipo de vivienda	Piso	Piso	Piso

distrito	Fuencarral-El Pardo	Hortaleza	Centro
piscina	0	1	0
terraza	0	1	0
jardin	0	1	0
garaje	0	1	0
trastero	0	1	0
calefaccion	1	1	1
aire acondicionado	1	1	1
ascensor	1	1	0
superficie construida	98	208	75
habitaciones	3	4	2
baños	2	3	1

Tabla 23 Predicciones realizadas por el modelo de Random Forest

Una vez encontrado el mejor modelo de aprendizaje supervisado, pasamos a aplicar las técnicas de aprendizaje no supervisado.

3.5. Aprendizaje No Supervisado

En cuanto a las técnicas de aprendizaje no supervisado, nos interesa obtener una nueva agrupación de las viviendas mediante técnicas de clustering jerárquico y no jerárquico. Es conocido que las técnicas de aprendizaje no supervisado se caracterizan porque los datos no tienen etiqueta; es decir, no hay variable respuesta. Por lo que, para aplicar este tipo de herramientas debemos desechar nuestro "target", que contiene el precio de las viviendas del conjunto de datos. Se desecha la variable respuesta, pues nuestra intención es lograr obtener una nueva agrupación de las viviendas sin tener en cuenta el precio de las mismas, que podría condicionar sustancialmente la formación de los clústeres.

Con las técnicas de clustering, nuestro objetivo es agrupar las viviendas de nuestro conjunto de datos en función de sus características. Es cierto que ya poseemos una variable categórica multiclase que agrupa las viviendas por distritos. Sin embargo, aunque esta agrupación geográfica es totalmente correcta, con esta clasificación estamos considerando 21 distritos, lo cual parece un número bastante elevado para las menos de 3000 instancias del conjunto de datos. Por lo que, con el fin de encontrar una manera más resumida de agrupar las viviendas, decidimos aplicar técnicas de clustering jerárquico y no jerárquico.

Como ya se ha mencionado anteriormente, para obtener una nueva agrupación de las viviendas, debemos encontrar el número de clústeres óptimo que se formarán. Para ello, aplicamos el algoritmo de clustering jerárquico aglomerativo y el método de la silueta media. Primero aplicamos el algoritmo de clustering jerárquico aglomerativo, en el que partimos de n conglomerados y se van uniendo hasta que solo queda 1. Obtenemos la

visualización de la estructura jerárquica de los conglomerados y cómo se van agrupando con el dendrograma de la figura 22. En este gráfico, podemos ver claramente que se forman 2 grupos, el primero representado por el color naranja y el segundo, mucho más numeroso, por el color verde.

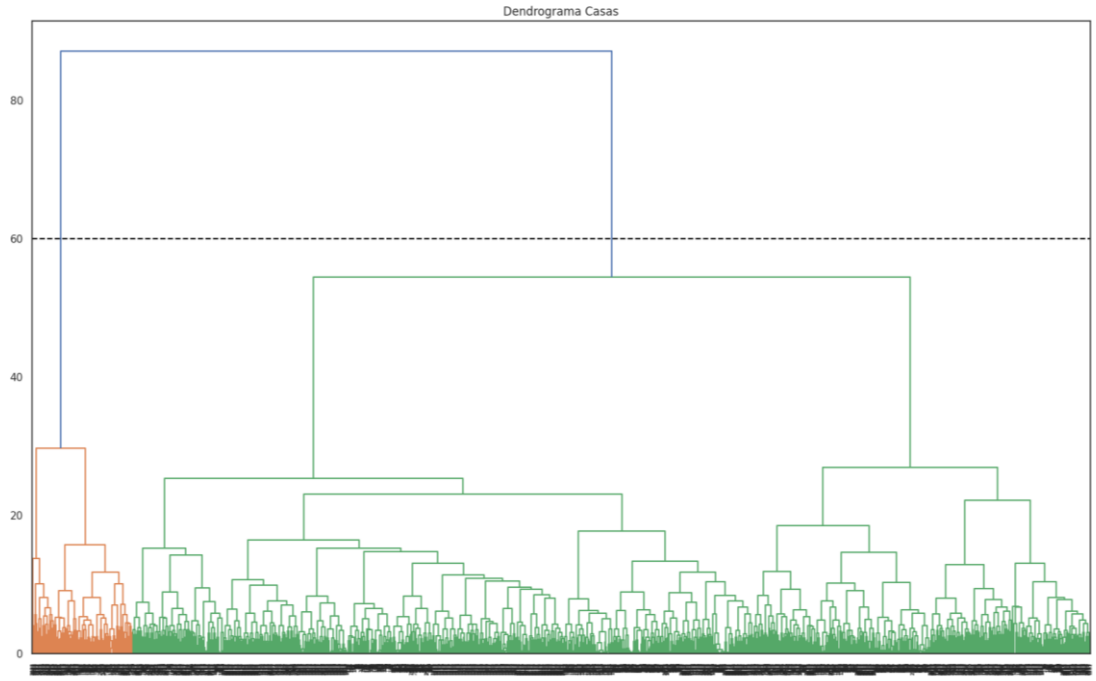

Figura 22 Dendrograma de las viviendas

A continuación, empleamos la técnica de la silueta media, que nos da una medida de la calidad de la agrupación para un número k de clústeres. En este caso, se ha decidido utilizar un número reducido para el número de clústeres k, entre 2 y 10. Podemos examinar los resultados que se obtienen con esta técnica en la figura 23, y obtenemos que el número de clústeres óptimo es 2. Por tanto, tanto el algoritmo de clustering jerárquico aglomerativo como la técnica de la silueta media, proporcionan el mismo resultado.

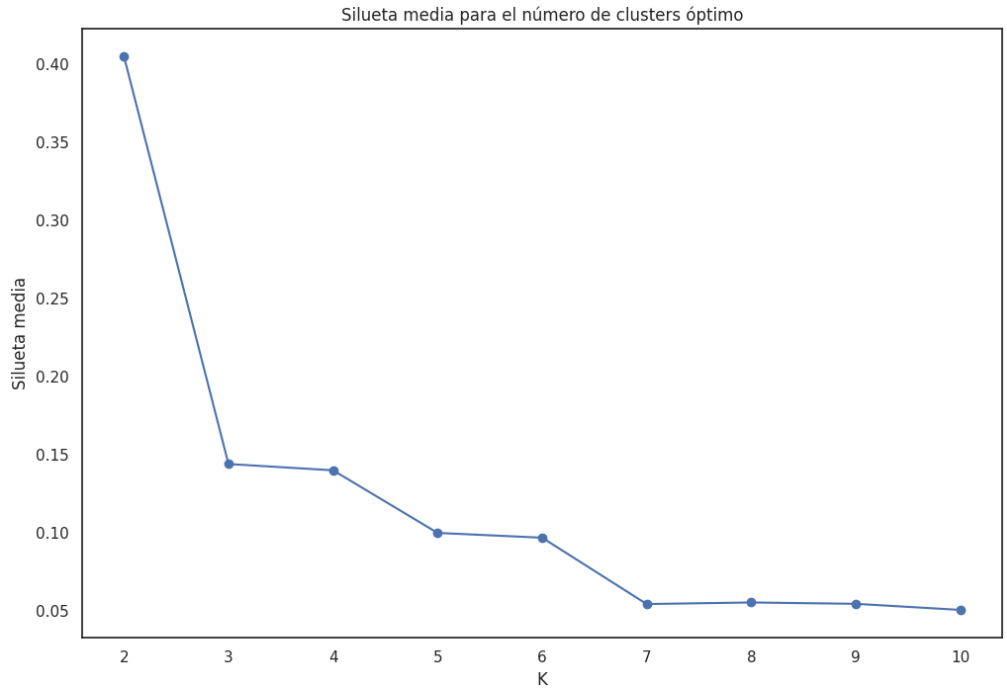

Figura 23 Método de la silueta media para determinar el número de clústeres óptimo

Una vez determinado el número de clústeres óptimo, aplicamos el algoritmo de las k-medias y obtenemos las características medias para cada clúster. Como podemos ver en la figura 24, el clúster 0 podría decirse que agrupa a las viviendas lujosas y el clúster 1 agrupa a las viviendas más asequibles. En las figuras 24 a 29 podemos ver las características medias de cada clúster formado y las diferencias entre ambos clústeres.

Comenzando con las variables numéricas, en la figura 24 percibimos que el precio medio de una vivienda en el clúster 0 es de 1.2 millones de euros mientras que en el clúster 1 no llega a 400.000 euros. También observamos diferencias en la superficie construida que es prácticamente de 200 metros cuadrados para el grupo 0 y de 75 para el grupo 1. En cuanto al número de habitaciones y baños, si redondeamos al número entero más cercano, en media obtenemos que el grupo 0 tiene 4 habitaciones y 3 baños, mientras que el grupo 1 tiene 2 habitaciones y 1 baño.

Continuando con las variables categóricas multiclase, en la figura 25 extraemos que los distritos más comunes para el grupo 0 son Salamanca, Centro, Chamartín, Moncloa-Aravaca, Chamberí y Fuencarral-El Pardo, mientras que para el grupo 1 son Centro, Salamanca, Carabanchel, Puente de Vallecas, Ciudad Lineal y Latina. En las figuras 26 y 27, observamos la posición geográfica de los distritos con mayor número de viviendas ofertadas para cada clúster y percibimos que hay una separación geográfica importante. En la figura 28, vemos un gráfico de barras del tipo de viviendas ofertadas en cada clúster y observamos que, aunque en los 2 clústeres predomina la oferta de pisos, en el clúster 0 hay una mayor proporción de áticos, chalets, casas y dúplex respecto al clúster 1.

Finalizando con las variables categóricas binarias en la figura 29, podemos ver aproximadamente que para el clúster 0 el 60% de las viviendas poseen aire acondicionado y terraza, el 70% calefacción, el 80% ascensor, el 40% garaje y trastero, el 25% jardín y

piscina. En cambio, para el clúster 1 el 40% de las viviendas poseen aire acondicionado, el 50% calefacción, el 55% ascensor, el 15% garaje y trastero, el 35% terraza, el 8% jardín y piscina.

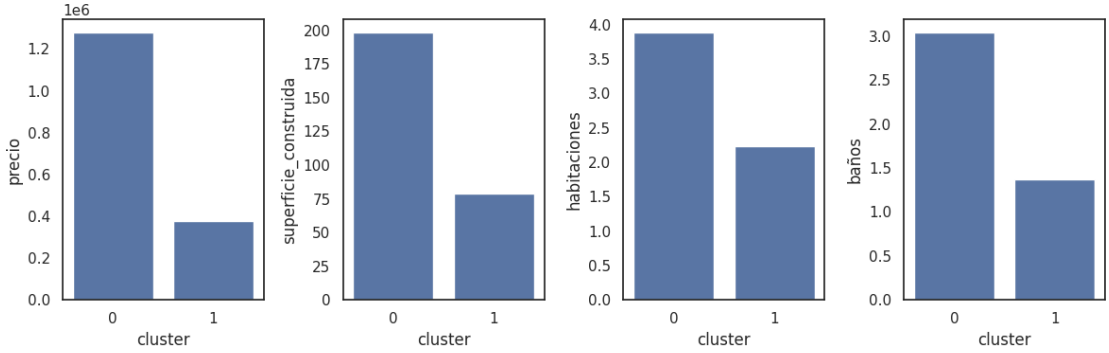

Figura 24 Características numéricas de cada clúster

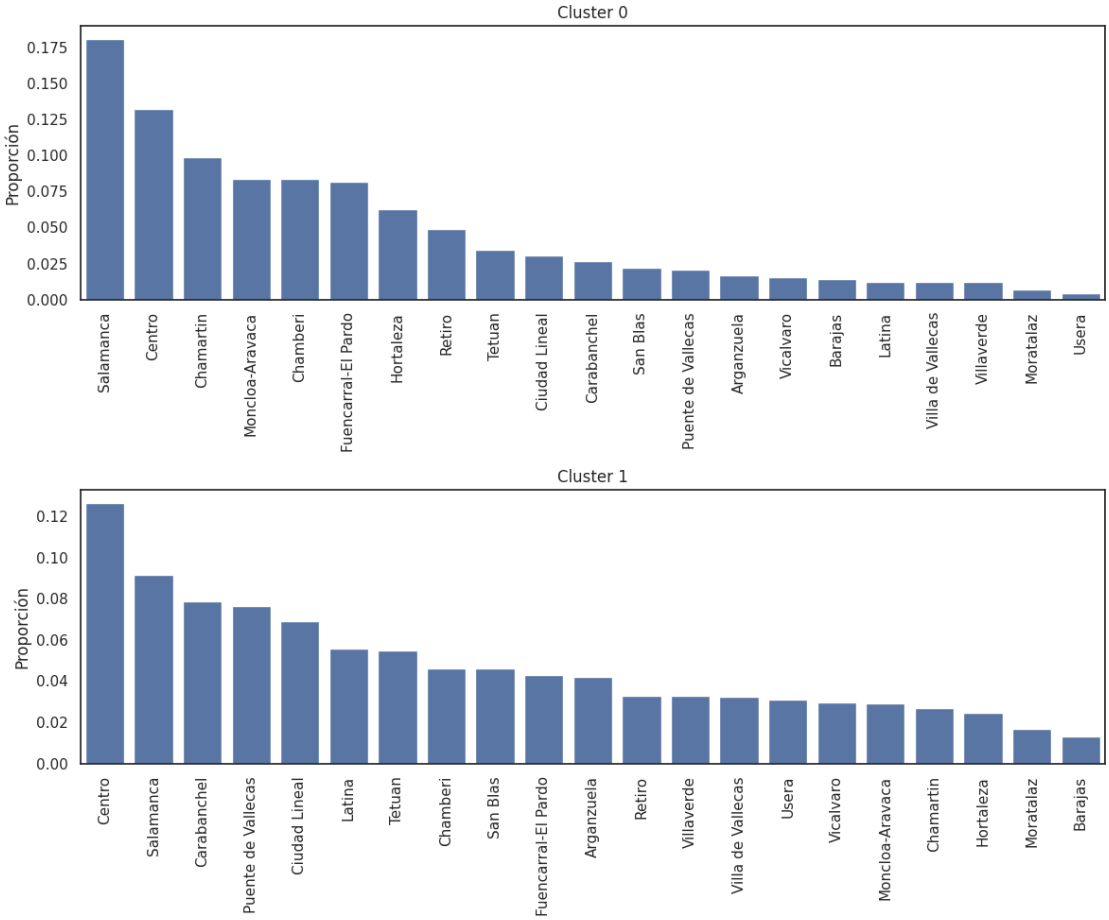

Figura 25 Características distrito de cada clúster

Figura 26 Mapa de los distritos de Madrid con mayor oferta de viviendas para el clúster 0

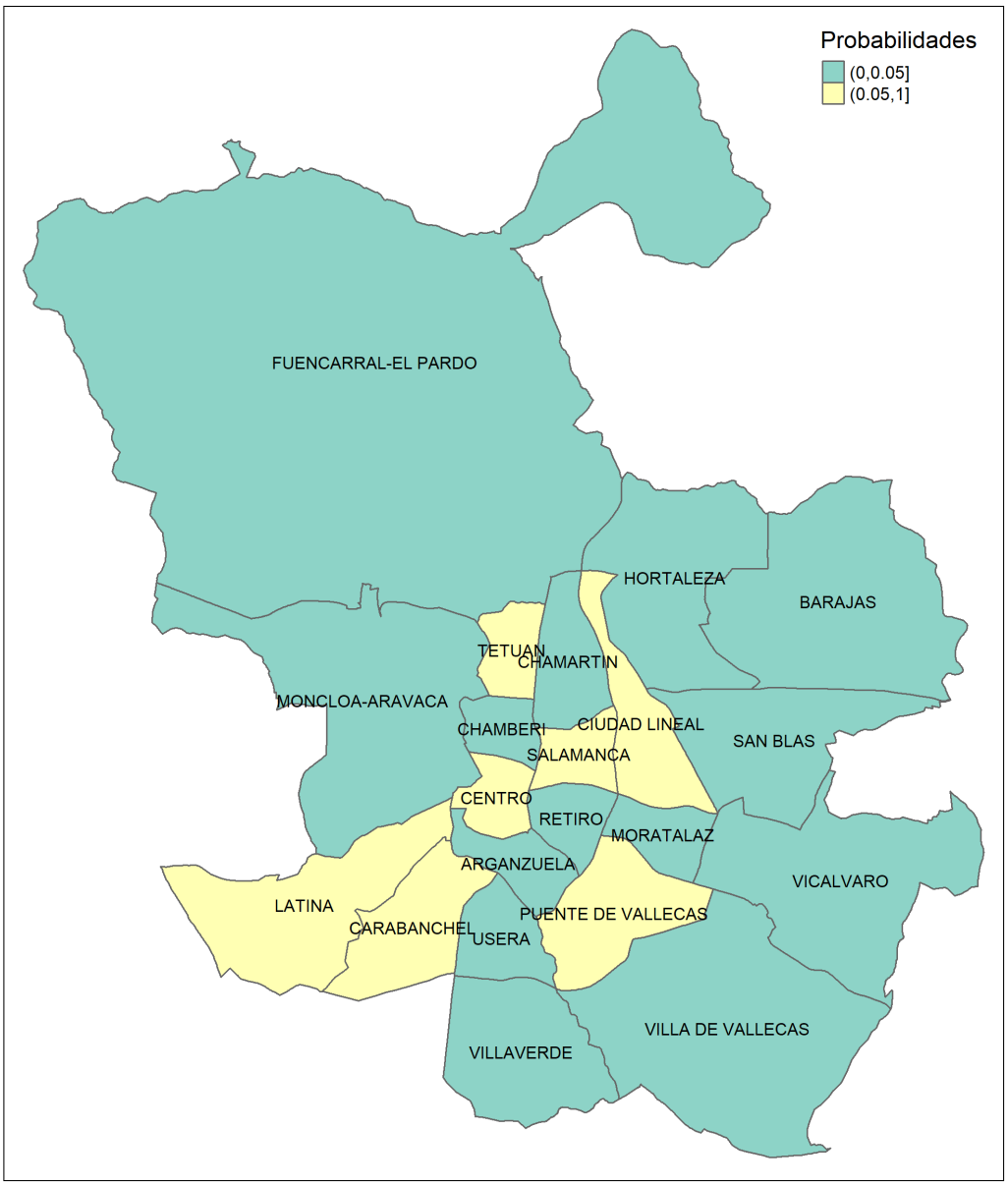

Figura 27 Mapa de los distritos de Madrid con mayor oferta de viviendas para el clúster 1

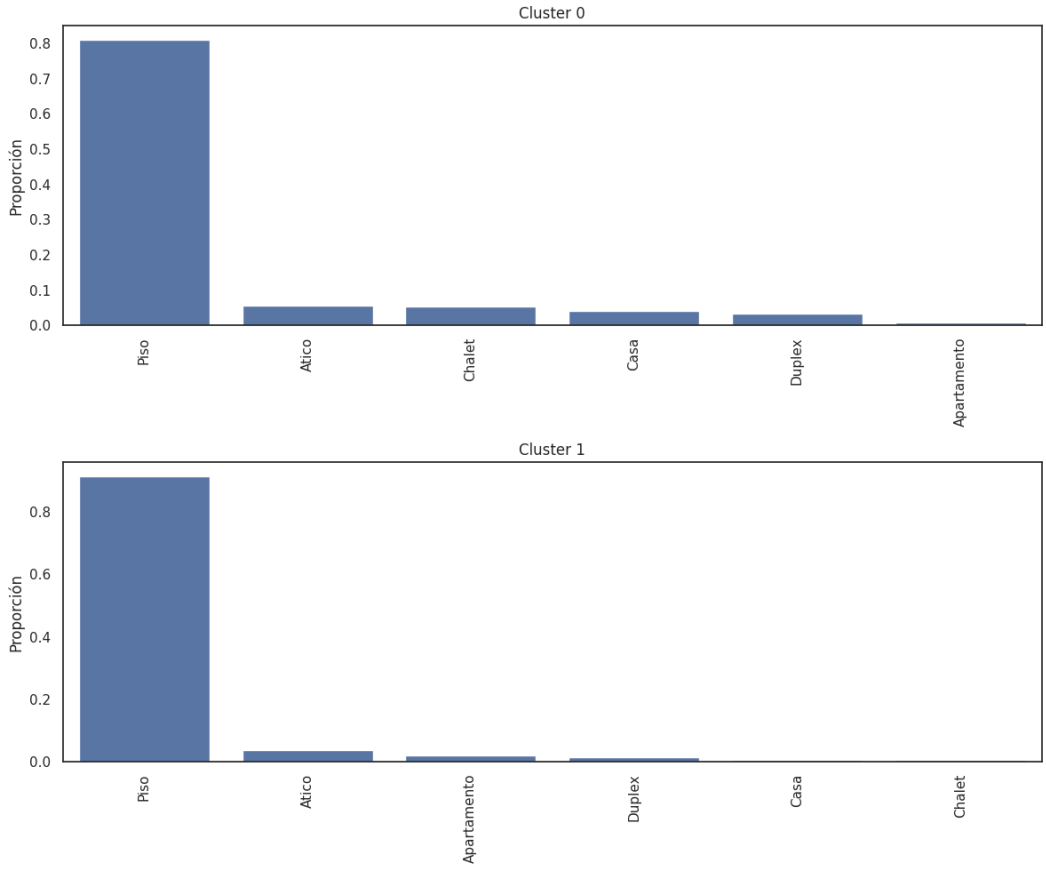

Figura 28 Características tipo de vivienda de cada clúster

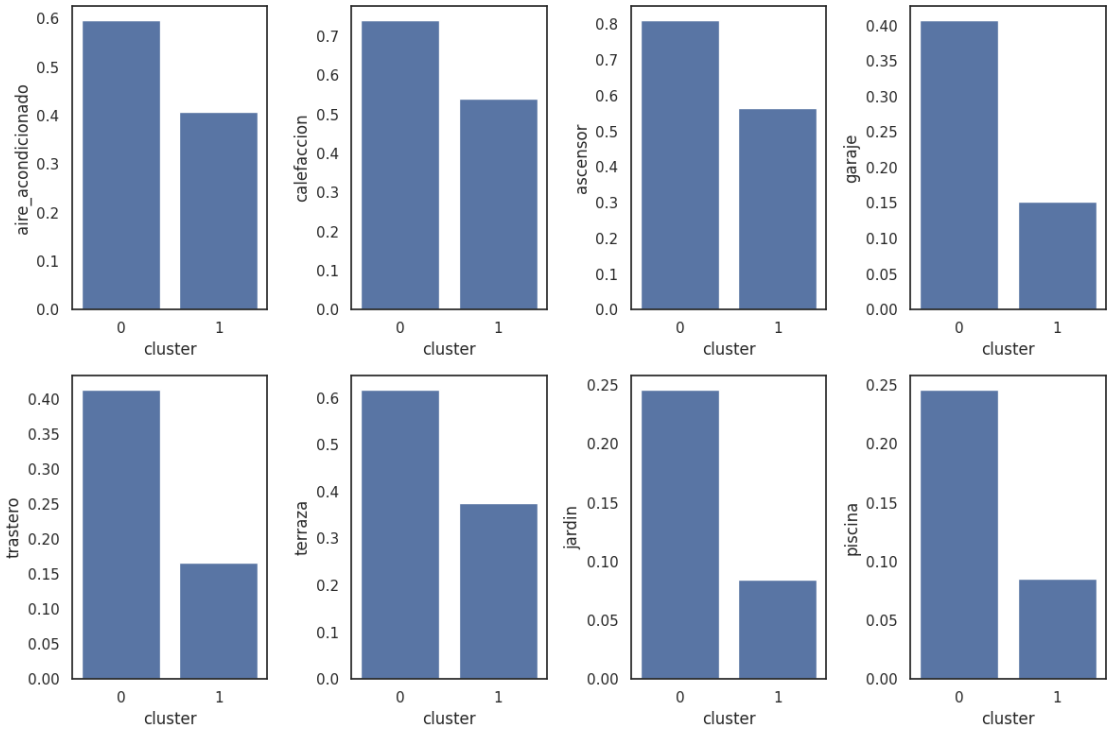

Figura 29 Características binarias de cada clúster

Para ver de forma más detallada los resultados de las características medias para cada grupo, calculamos una serie de intervalos de confianza. Podemos ver resumidas las características medias de las viviendas de cada clúster así como los intervalos de confianza para la media al 95% en las tablas 24 y 25. Los intervalos de confianza para la media se han creado basándose en la distribución t-student, y el intervalo toma la forma de la expresión (3.4). Podemos ver la expresión del estadístico que se utiliza para crear el intervalo de confianza en la fórmula (3.5).

$$IC(\bar{x}) = [\bar{x} \pm t_{est} \cdot \frac{\sigma}{\sqrt{n}}] \qquad (3.4)$$

Donde:

$$t_{est} \sim t((1 + (1 - \alpha))/2\,, n - 1) \qquad (3.5)$$

Como podemos ver en las tablas 24 y 25, los intervalos de confianza que se crean para la media son muy estrechos, debido principalmente a que el tamaño del conjunto de datos es grande, luego la desviación típica de la media es pequeña. Si analizamos los resultados de las tablas podemos ver que, por ejemplo para el precio de la vivienda, el rango del clúster 0 es de 135.000 euros mientras que para el clúster 1 es de 25.000 euros. Es decir, que la desviación típica de la media para el clúster 0 es mayor que para el clúster 1, lo que conlleva unos intervalos de confianza más anchos para el clúster 0. Este comportamiento se repite para otras características como puede ser la superficie construida de la vivienda, con un rango para el grupo 0 de 14 metros cuadrados y para el grupo 1 de 2 metros cuadrados. Podemos concluir que los intervalos de confianza para el grupo 1 son más estrechos y por tanto más precisos que los del grupo 0, debido a que el tamaño muestral del grupo 1 es prácticamente 3 veces mayor al del grupo 0. La interpretación de los intervalos de confianza es la misma para todas las características y clústeres que se consideren. Si tuviésemos 100 muestras de datos, entonces podríamos afirmar que, en el 95% de los casos estos intervalos contendrían el verdadero valor de la media.

Característica	**Media**	**IC Inferior**	**IC Superior**
precio	1.277.620	1.209.220	1.346.010
superficie_construida	198.2	191.41	204.99
habitaciones	3.88	3.81	3.95
baños	3.04	2.96	3.12
piscina	0.25	0.21	0.28
terraza	0.62	0.58	0.65
jardin	0.25	0.21	0.28
garaje	0.41	0.37	0.44
trastero	0.41	0.38	0.45
calefaccion	0.74	0.71	0.77

aire_acondicionado	0.6	0.56	0.63
ascensor	0.81	0.78	0.84

Tabla 24 Intervalo de confianza al 95% para la media de las características principales del clúster 0

Característica	**Media**	**IC Inferior**	**IC Superior**
precio	375.168	362.693	387.644
superficie_construida	79.13	77.94	80.33
habitaciones	2.23	2.2	2.26
baños	1.37	1.35	1.4
piscina	0.09	0.07	0.1
terraza	0.37	0.35	0.4
jardin	0.08	0.07	0.1
garaje	0.15	0.14	0.17
trastero	0.17	0.15	0.18
calefaccion	0.54	0.52	0.56
aire_acondicionado	0.41	0.39	0.43
ascensor	0.56	0.54	0.59

Tabla 25 Intervalo de confianza para la media al 95% de las características principales del clúster 1

Una vez finalizado el análisis de resultados de las técnicas de aprendizaje automático, pasamos a la elaboración de conclusiones.

4. CONCLUSIONES

Tras haber analizado los resultados y haber estudiado el comportamiento del conjunto de datos, pasamos al último paso de la investigación: la elaboración de conclusiones.

Se ha realizado una tarea de extracción de datos en la página web del portal de viviendas de pisos.com. Partiendo del conjunto de datos original, se han realizado algunas transformaciones de las variables para lograr que fuesen más significativas. También, se ha realizado un análisis de los valores atípicos y se han eliminado aquellos valores demasiado extremos, pues no parecían ser una muestra representativa de la población. Tras ello, se ha realizado un análisis exploratorio de los datos y se ha descubierto que el conjunto de datos está compuesto por cerca de 3000 instancias y 14 características, de las cuales 2 son numéricas continuas, 2 numéricas discretas, 2 categóricas multiclase y 8 categóricas binarias. Se han aplicado los ajustes necesarios en la fase de preproceso, como son la codificación binaria para las variables categóricas multiclase, la estandarización de las variables numéricas y el tratamiento de los valores faltantes mediante la imputación multivariante con el modelo del KNN.

Se ha alcanzado un modelo de aprendizaje supervisado que es capaz de predecir el comportamiento de la variable respuesta de forma muy satisfactoria. Este modelo es el Random Forest con 50 modelos base y con un 50% de características utilizadas de forma aleatoria para construir cada modelo base. Con este modelo se obtiene una métrica de evaluación R^2 de aproximadamente 0.84 y 0.87 para los datos de entrenamiento y de validación, respectivamente. Es decir, que hemos logrado obtener un modelo que no solo se ajusta bien a los datos de entrenamiento, sino que además generaliza muy bien para los datos de validación. Además, hemos obtenido que las variables más importantes para determinar el precio de una vivienda son la superficie construida y el número de baños.

Este modelo de Random Forest se podría utilizar para futuros estudios, siempre y cuando no cambien demasiado la tendencia de los precios de las viviendas ni las reglas de decisión. También, se podría ajustar el modelo para realizar el mismo estudio sobre las viviendas de otras ciudades grandes como Barcelona, pues aunque las reglas de decisión seguramente no cambien mucho, los precios de las viviendas por su parte variarán considerablemente. Seguramente el precio por metro cuadrado será muy diferente y se considerarán otras características más importantes. En adición, este estudio lo podría realizar por ejemplo una inmobiliaria que operara en Castilla y León, y que quisiese adaptarse al mercado de las viviendas en Madrid.

Para próximos estudios de investigación, queda pendiente por obtener un conjunto de datos que tenga en cuenta la posición de la vivienda en el mapa de la ciudad de Madrid. Si se tuviesen en cuenta las coordenadas de la vivienda, latitud y longitud, se podría obtener un análisis con un mayor nivel de detalle y con unos resultados aún mejores. Con estas coordenadas, por ejemplo, podríamos calcular la distancia de la vivienda a un centro educativo o el número de centros educativos en un radio de 2 kilómetros. Otras variables

que se podrían considerar serían cercanía a parques, acceso al transporte público, servicios de salud, tiendas y comercios…

En cuanto a las técnicas de clustering jerárquico y no jerárquico, hemos podido ver que las viviendas se pueden agrupar en 2 clústeres. El primer clúster se podría decir que representa las viviendas lujosas de nuestro conjunto de datos, mientras que el segundo, mucho más numeroso, representa las viviendas más asequibles. Si redondeamos los resultados de las características más importantes de cada clúster, podemos extraer que el primer grupo de viviendas tiene un precio medio de 1.270.000 euros, una superficie construida de 198 metros cuadrados, 4 habitaciones y 3 baños. Mientras que el segundo grupo de viviendas tiene un precio medio de 375.000 euros, una superficie construida de 80 metros cuadrados, 2 habitaciones y 1 baño.

Finalmente, podemos concluir que se han realizado con éxito todas las tareas de aprendizaje automático consideradas y que se ha cumplido correctamente con todos los objetivos planteados. Hemos logrado entrenar un modelo que predice de manera bastante precisa el precio de las viviendas en la ciudad de Madrid y hemos logrado obtener una nueva agrupación de las viviendas en 2 grupos claramente distintos.

5. REFERENCIAS

1. Mpoke Bigg, M. (28 de febrero, 2023). Guerra en Ucrania: 6 consecuencias que ha tenido en el mundo. *The New York Times*. Guerra en Ucrania: 6 consecuencias que ha tenido en el mundo - The New York Times (nytimes.com)

2. Instituto Nacional de Estadística (2024). Índices Nacionales: general y de grupos ECOICOP [Datos]. Índices nacionales: general y de grupos ECOICOP(50902) (ine.es).

3. Instituto Nacional de Estadística (2024). Índices por CCAA: general, vivienda nueva y de segunda mano [Datos]. Índices por CCAA: general, vivienda nueva y de segunda mano (25171) (ine.es)

4. Instituto Nacional de Estadística (2024). PIB pm Oferta (Precios corrientes) [Datos]. PIB pm Oferta (Precios corrientes)(30678) (ine.es)

5. Instituto Nacional de Estadística (2024). Tipo de interés medio al inicio de las hipotecas constituidas [Datos]. Tipo de interés medio al inicio de las hipotecas constituidas(24457) (ine.es)

6. Thamarai, M., & Malarvizhi, S. P. (2020). House Price Prediction Modeling Using Machine Learning. *International Journal of Information Engineering & Electronic Business*, *12*(2). template.doc (mecs-press.org)

7. Lu, S., Li, Z., Qin, Z., Yang, X. & Siow Mong Goh, R. (2017). A hybrid regression technique for house prices prediction. *IEEE International Conference on Industrial Engineering and Engineering Management (*pp. 319–323). A hybrid regression technique for house prices prediction | IEEE Conference Publication | IEEE Xplore

ANEXO A DATOS WEB SCRAPING

En los siguientes enlaces viene información sobre cómo se ha realizado la tarea de web scraping y cómo se ha obtenido la base de datos que se ha utilizado en este presente trabajo.

1. Enlace a los datos utilizados para el trabajo:

Trabajo-Fin-de-Grado/Datos.csv at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

2. Enlace al código utilizado para el web scraping:

Trabajo-Fin-de-Grado/Final.py at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

ANEXO B DOCUMENTOS PYTHON

En los siguiente enlaces se puede obtener información sobre cómo se ha realizado el trabajo en Python y los resultados que se han obtenido.

3. Enlace al código utilizado para la realización del trabajo:

Trabajo-Fin-de-Grado/Trabajo_Fin_Grado_LU.html at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

4. Enlace a las funciones utilizadas para la realización del trabajo:

Trabajo-Fin-de-Grado/Paquete_Funciones_TFG_LU.ipynb at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

ANEXO C MODELOS FINALES

En los siguientes enlaces se puede ver la representación gráfica del modelo de árbol de decisión, la importancia de las variables y las predicciones determinadas por el modelo de Random Forest.

5. Enlace al documento de visualización del árbol de decisión:

Trabajo-Fin-de-Grado/arbol_decision.pdf at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

6. Enlace al documento que contiene la importancia de las variables:

Trabajo-Fin-de-Grado/importancia.xlsx at main · rbooantt/Trabajo-Fin-de-Grado (github.com)

7. Enlace al documento que contiene las predicciones y los valores reales de la variable respuesta precio:

Trabajo-Fin-de-Grado/Predicciones.xlsx at main · rbooantt/Trabajo-Fin-de-Grado (github.com)