Universidad Internacional de La Rioja

Escuela Superior de Ingeniería y Tecnología

Máster Universitario en Análisis y Visualización de Datos Masivos

Modelo predictivo para valoración financiera de activos inmobiliarios

Trabajo Fin de Máster

Presentado por: Paredero Gil, Daniel

Directora: García Martínez, Yamila

Madrid
07/02/2020

Resumen

Nota: El objetivo de este trabajo es desarrollar un modelo para la predicción de precios de referencia de activos inmobiliarios a través de variables informacionales de dichos activos tales como la localización geográfica o la superficie del activo.

Para el desarrollo de este modelo se usarán datos procedentes de transacciones de compraventa, pertenecientes a un portal web inmobiliario. Dichos datos serán procesados y estudiados con el fin de aplicar técnicas de inteligencia artificial, centradas en modelos de aprendizaje supervisado. Estos modelos de aprendizaje supervisado tendrán como input las variables elegidas y obtendrán una estimación del precio del activo.

Tras obtener los resultados se evaluarán las predicciones y se comprobará que las tasas de error estén por debajo de un umbral establecido.

Finalmente, las conclusiones obtenidas servirán para establecer vías de trabajo futuro, basadas en incluir un mayor número de variables, por un lado, referentes a los activos inmobiliarios, pero también relativas a los factores económicos de la región geográfica.

Palabras Clave: sector inmobiliario, modelo predictivo, precios, valoración inmobiliaria.

Abstract

Note: The objective of this work is to develop a model for prediction of reference prices for real estate assets through informational variables such as geographical location or surface of the asset.
For the development of this model, data from sales transactions, belonging to a real estate web portal, will be used. This dataset will be processed and studied in order to apply artificial intelligence techniques, focused on supervised learning models. These supervised learning models will have those variables as input and will obtain an estimation of the price of the asset.
After obtaining the results, the predictions will be evaluated, and it will be verified that the error rates are below a set threshold.
Finally, the conclusions obtained will serve to establish future work paths, based on including a greater number of variables, on the one hand, referring to real estate assets, but also related to the economic factors of the geographical region.

Keywords: real estate, predictive model, prices, real estate assessment.

Contenido

Índice de tablas

Índice de figuras

1. Introducción

La valoración efectiva de activos inmobiliarios supone en la actualidad uno de los retos a los que se somete la economía mundial, estando muy presentes las consecuencias de la crisis económica surgida en Estados Unidos por las hipotecas *subprime* y que se extendió a todo el mundo (Tong, H. & Wei, S., 2008).

Figura 1. Evolución del porcentaje de hipotecas subprime sobre el total la década anterior a la crisis de 2008. (Pozzi, S, 2017)

El aumento de las hipotecas de riesgo o *subprime* estuvo asociado con una burbuja en el precio de los activos inmobiliarios dado que la facilidad de crédito a mayor riesgo propició un aumento en los precios sin que se viese afectado el consumo (Sanders, A, 2008). La consecuencia fue la crisis y el desplome del precio de la vivienda.
Este trabajo pretende construir un modelo de predicción de precios de referencia para activos inmobiliarios con el objetivo de tener una herramienta de consulta que permita estudiar los precios de los activos y poder evaluar su evolución. De esta forma, se reduce el riesgo de sufrir una nueva burbuja en el precio de la vivienda.

1.1 Justificación

El problema tratado se centra en los métodos de valoración del precio de activos inmobiliarios. La forma habitual de realizar una valoración de los activos es a través de la tasación, sin embargo, este procedimiento cuenta con puntos negativos.

Uno de estos puntos es el alto coste que puede alcanzar realizar una tasación. Otro punto es el hecho de que la naturaleza de una tasación está intrínsecamente ligada al momento en el que se realizada, debido a que tiene está influenciada directamente por las variables macroeconómicas temporales. Este es el motivo por el que las tasaciones pierden validez al cabo de un tiempo, debiéndose volver a realizar (Vedo Núnez, M.,

2016). Debido al coste de las tasaciones, esta valoración se suele realizar cuando la transacción de compraventa está en un punto avanzado del proceso. Esto muestra que existe un periodo del proceso, que va desde la publicación en el mercado del activo hasta la tasación en el que existe una valoración financiera diferente sobre el activo inmobiliario. Esta valoración previa a la tasación es el precio de referencia del activo, y la forma de establecerse es a día de hoy, principalmente heurística (Obaid, M., 2008. Eficiencia en tasaciones dentro del mercado inmobiliario) Por otro lado, la componente temporal de las tasaciones hace que el sector inmobiliario mayorista tenga que realizar tasaciones periódicas sobre sus activos para evaluar la evolución del mercado y poder analizar sus carteras, generando un gasto adicional.

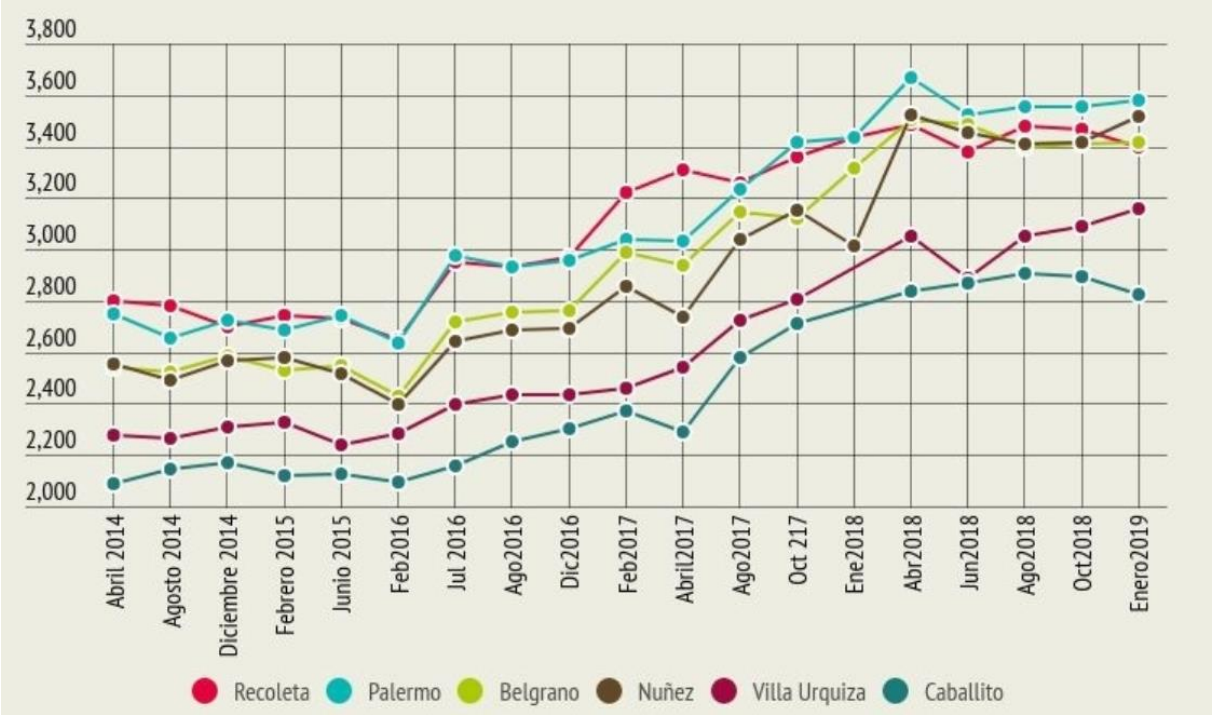

Figura 2. Evolución temporal del precio en dólares de vivienda usada de 2 habitaciones en 6 barrios de Buenos Aires desde 2014 a 2019. (Maure, 2019)

Es por esto por lo que se plantea en este trabajo el desarrollo de un sistema de valoración financiera basado en modelos predictivos.

1.2 Planteamiento del trabajo

El modelo que se plantea se basa en algoritmos de Inteligencia Artificial para establecer predicciones sobre los precios (UNIR, 2019. Técnicas de Inteligencia Artificial). Concretamente se enfoca como un problema de aprendizaje supervisado, en el que el algoritmo dispone en la fase de entrenamiento de los datos, i.e. las características de los activos inmobiliarios, y las etiquetas, i.e. los precios. A través del entrenamiento el

modelo asocia precios con características del inmueble y es capaz de dar predicciones ante nuevos datos que nunca han sido tratados por el modelo.

1.3 Estructura del trabajo

En el capítulo 1 se muestra una introducción al tema tratado y la motivación que existe para abordar dicho problema.

En el capítulo 2 se presenta la situación actual del sector inmobiliario a nivel global y estudios relacionados con la predicción de valoraciones financieras.

El capítulo 3 contiene los objetivos del trabajo, mostrando el objetivo principal y desglosándolo en los objetivos específicos. Se establece también la metodología seguida en el proyecto.

En el capítulo 4 se hace el desarrollo específico de la contribución del trabajo. Se explican las tecnologías usadas para el trabajo y se muestra el tratamiento de los datos para ajustarlos al modelo. Además, se realiza un análisis previo de las características del conjunto de datos y adecuación para ser el input del modelo. Por último, tiene lugar el entrenamiento de un modelo predictivo basado en Random Forest, explicando en qué consiste el algoritmo y cuáles son los resultados, realizando para ello una comparación de distintas configuraciones paramétricas.

En el capítulo 5 se desarrollan las conclusiones a los resultados mostrados previamente y se plantean las líneas de trabajo futuro.

2. Contexto y estado del arte

El sector inmobiliario y el establecimiento de un sistema de predicción de valoraciones inmobiliarias es un tema de capital importancia en la economía y cuenta con numerosos estudios al respecto que lo avalan.

2.1 Contexto

El sector de inmobiliario ha sufrido una gran transformación a nivel mundial en las dos últimas décadas. En primer lugar, la Gran Recesión de 2008 tuvo su origen en el modelo de hipotecas *subprime* de Estados Unidos y expandió por todo el mundo, incluido Argentina, país en el que se centrará el proyecto (Pérez, P. E & Féliz, M., 2019). Esto creó una recesión económica en la que el mayor indicador fue el aumento del impago de hipotecas. Este aumento de los niveles de morosidad fue causado por la creación de burbujas inmobiliarias con subidas de precios en el sector inmobiliario muy superiores a la tendencia general de la economía en la mayoría de los países afectados. Las consecuencias más inmediatas fueron las restricciones de crédito por parte de las entidades financieras, especialmente a particulares, y la consecuente introducción de grandes fondos de inversión a oportunidades de negocio generadas por esta situación (Tong, H. & Wei, S., 2008).

Figura 3. Evolución histórica del euríbor (Finect, 2020)

En la Figura 3 se puede ver la bajada histórica del euríbor como un indicador de la crisis financiera e inmobiliaria experimentada en Europa (Hakala J.O., 2016). Este es el índice

de referencia europeo de los tipos de interés interbancario y ligado está ligado a las hipotecas variables. El resto de los índices interbancarios mundiales sufrieron la misma tendencia de subida antes de la crisis, lo cual aceleró el pico de morosidad de los años 2007 y 2008 al aumentar los tipos de interés (Borrallo Egea, F.A. & Hierro Recio, L.A., 2015).

De esta forma, muchos países, especialmente iberoamericanos, experimentaron la entrada de nuevos *players* en sus sectores inmobiliarios, en forma de fondos de capitales invirtiendo de forma masiva, actuando de inyecciones de capital en el sistema y cambiando el panorama inmobiliario (Ocampo, A., 2009). Sin embargo, este sector ha sido testigo también de la confluencia de dos tendencias contrapuestas que son las que finalmente han moldeado la situación actual. Por un lado, estos *players* han creado un escenario en el que existe una mayor capacidad por su parte de controlar el precio de los activos inmobiliarios en el mercado, gracias al control de la información del que disponen en contraposición con el tradicional sector minorista. Por otro lado, la expansión de internet ha tenido como consecuencia la aparición de portales de compraventa alternativos, que ofrecen mayores posibilidades a los usuarios y una mayor accesibilidad en comparación con las tradicionales inmobiliarias (Gruber, 2016).

2.2 Estado del arte

El escenario descrito anteriormente ha derivado en un contexto de creciente interés en el tema de estudio del proyecto. Los avances en la materia de estudio se pueden dividir en 2 categorías principales.

La primera categoría son los trabajos centrados en la elaboración de índices de referencia como indicadores del precio de los activos inmobiliarios. En esta línea se tienen índices como el desarrollado por el portal web inmobiliario Fotocasa (Fotocasa, 2020), el cual es un índice de referencia para el FMI y se desarrolla a partir de los datos procedentes del portal web a partir de 2005. En este sentido, los estudios acerca del sector latinoamericano son más escasos, teniendo por ejemplo el artículo de Domínguez Prost, R.F., en el que hace énfasis en la escasez de estudios en Argentina. En este artículo, se construye un índice a través de la realización de una regresión hedónica centrada en la situación geográfica de los activos inmobiliarios y su distancia a puntos de referencia.

La segunda categoría de trabajos centrados en la materia del proyecto son los estudios que abordan modelos de valoraciones financieras o precios de referencia para activos inmobiliarios, bien a través de la propia predicción del valor, bien a través de evaluar el

impacto que tienen variables relativas a los activos en el valor. En la línea de la predicción de la valoración se tiene estudios como el de Vedo Núñez, M. el cual realiza una descripción de la situación del sector inmobiliario en el mundo y en España en particular y aborda modelos estadísticos para predecir los precios. En la línea de la evaluación del peso que tienen variables de los activos están estudios como el de Hlaczikm, M.M, en el que se ve cómo varía la estrategia de venta según las características de las propiedades en el partido de La Plata. También está el estudio de Montero, J.M. & Fernández-Avilés, G. en el que se evalúa el peso de los efectos espaciales en el precio de la vivienda en España.

Siguiendo la misma línea se encuentra el estudio de Liu, J.G., Zhang, X.L. & Wu, W.P. (2006), en el cual se realiza un modelo de predicción de los precios de activos inmobiliarios, usando para ello algoritmos de redes neuronales. Este estudio, al igual que el desarrollado en el este trabajo, se basa en un método hedónico para estimar los precios de los activos.

A pesar de haber numerosos estudios sobre la materia, el denominador común es la menor cantidad de trabajos publicados centrados en los mercados latinoamericanos y la abundancia de estudios centrados en aplicar técnicas estadísticas tradicionales sin plantear la perspectiva del Big Data.

3. Objetivos concretos y metodología de trabajo

3.1. Objetivo general

Se plantea el desarrollo de un sistema de valoración que constituya una alternativa a los métodos heurísticos del establecimiento de precios de referencia, y fundamente en la evolución de los datos de las propias ventas, la evolución de los precios, de forma que facilite el proceso de toma de decisiones en las transacciones de compraventa de inmuebles, a la par que abarate el coste de realizar tasaciones periódicas.

3.2. Objetivos específicos

El objetivo principal del trabajo se divide en varios hitos específicos necesarios para su consecución:

- Búsqueda y captura de datos pertenecientes a transacciones inmobiliarias. Dado que el sector inmobiliario es un sector con una menor implantación de las nuevas tecnologías, este punto afronta cierta complejidad. Sin embargo, gracias a los nuevos portales web, tales como el portal argentino Properati, se puede acceder a fuentes fiables de datos. Estas empresas son más recientes que las inmobiliarias tradicionales, pero tienen una implantación total de las nuevas tecnologías, lo cual permite establecer mayor confianza en la calidad del dato.
- Tratamiento de los datos para obtener un conjunto con una calidad del dato aceptable. Aunque los nuevos portales web inmobiliarios disponen de datos más confiables y disponibles, en cualquier trabajo de estas características es necesario tratar los datos y establecer filtros de calidad.
- Análisis de los datos para conocer las características de distribución. Es necesario establecer hipótesis sobre el conjunto de datos y comprobar si estas hipótesis se aceptan a través del estudio de las características de las variables. De esta forma, se puede establecer los niveles de correlación que existen entre las variables
- Aplicación del algoritmo Random Forest sobre el conjunto de datos con el objetivo de entrenar un modelo y evaluar sobre un subconjunto de datos de test el error de las predicciones sobre los datos reales.
- Evaluar los resultados del proyecto y establecer líneas de continuación para mejorar y extender los resultados.

3.3. Metodología del trabajo

El trabajo comienza con la captura de datos de la aplicación de Google BigQuery proporcionado por la empresa Properati SA. En dicha aplicación, la empresa publica de forma periódica sus datos de transacciones de todas las categorías y pasan a estar disponibles en forma de tablas que se pueden consultar con SQL. Tras estudiar las variables que aporta la empresa, es necesario delimitar el marco del proyecto de forma que se procede a establecer unos filtros sobre los datos y a extraer únicamente las variables que se consideran de especial relevancia. El poder acceder al entorno Google BigQuery permite una consulta más ágil sobre el conjunto de datos, y facilitar el proceso de decisión de filtros y campos, además de asegurar una mayor calidad en el dato.

Tras el establecimiento de la consulta final de extracción de los datos iniciales, se procede a cargar dichos datos en base de datos, entorno en el que se realizarán las transformaciones de los datos. Se crea una estructura de 2 tablas, la primera en la que se cargan los datos en bruto y la segunda en la que se cargan los datos tras pasar los filtros de calidad que se establecen. Junto con Tableau y SQL se realiza un análisis previo de los datos para conocer la distribución y correlación entre las variables y de esta forma evaluar si las variables seleccionadas y los filtros establecidos son adecuados al problema que se quiere tratar. Por último, se delimita el conjunto de datos a un subconjunto de tal forma que sea representativo del total de datos, con el objetivo de seleccionar un subconjunto computacionalmente adecuado a las limitaciones de hardware del proyecto.

Con los datos ya preparados, se procede a realizar la carga a Jupyter, entorno de Python en el que se desarrolla el entrenamiento del modelo predictivo. Previo a la aplicación del modelo, es necesario adaptar las variables al formato de codificación que exige el algoritmo. Una vez realizado este paso, se divide el conjunto de datos en un dataset de entrenamiento y otro de validación y entrena el modelo sobre el primero.

Para estudiar el comportamiento del algoritmo y su adecuación sobre el conjunto de datos, se realizan varias parametrizaciones, comprobando los resultados de rendimiento de los modelos entrenados con diferentes configuraciones de hiperparámetros.

Por último, se evalúan los resultados de rendimiento del modelo y se fija una línea de trabajo futuro en base a las conclusiones de este trabajo.

4. Desarrollo específico de la contribución

El primer punto a tratar es conseguir los datos necesarios para el entrenamiento del modelo y la obtención de los objetivos fijados anteriormente. En primer lugar, se describirán las tecnologías usadas en la extracción, transformación y tratamiento de los datos en el desarrollo del modelo. Se procederá a continuación a explicar dónde y cómo se han obtenido los datos. Después, se llevará a cabo la fase de estudio y transformación de los datos, fase crucial para en último lugar entrenar el modelo predictivo y evaluar su rendimiento.

4.1 Marco tecnológico

En el desarrollo del trabajo se han usado varias tecnologías de distintas áreas en el ámbito del tratamiento de datos.

En cuanto a lenguajes de programación, se han usado tanto el lenguaje de consultas SQL como Python. Por un lado, el trabajo con SQL se ha llevado a cabo con Google BigQuery y Microsoft SQL Server y, por otro lado, el trabajo con Python se ha desarrollado en el intérprete Jupyter.

Además, para la ayuda en la comprensión previa de las características de los datos se ha usado Tableau.

A continuación, se profundiza en estas tecnologías.

4.1.1 Structured Query Language (SQL)

El Lenguaje Estructurado de Consulta o SQL por sus siglas en inglés es, según IBM, un lenguaje estandarizado para definir y manipular datos en una base de datos relacional, cuyas declaraciones son ejecutadas por un gesto de bases de datos (IBM, 2020). Fue desarrollado por IBM en 1974 y es desde la década de 1980 estándar ANSI e ISO. Es un lenguaje sencillo y de dominio específico en el tratamiento de datos.

En el desarrollo del trabajo se ha usado SQL en dos puntos: en primer lugar, en la captura de datos a través de la plataforma Google BigQuery, y en segundo lugar en el tratamiento previo de los datos en Microsoft SQL Server.

Google BigQuery es un almacén de datos para empresas que abstrae del mantenimiento y desarrollo de una infraestructura y hardware de almacenamiento de grandes conjuntos de datos (Google Cloud, 2020). Se puede acceder a través de Cloud

Console, interactuando a través de SQL con la consola, y con diversas API's y aplicaciones de terceros. Además, es una herramienta que usan las empresas para poner a disposición del público sus "open datasets" de forma más estructurada que en ficheros planos y con mayor capacidad de almacenamiento.

Microsoft SQL Server es el gestor de bases de datos perteneciente a Microsoft, a través del cual se pueden almacenar datos estructurados en una base de datos relacional y consultarnos mediante el lenguaje SQL.

4.1.2 Lenguaje Python

Python es un lenguaje de programación de alto nivel y con alta potencia de rendimiento. Se trata de un lenguaje orientado a objetos, pero con una estructura de alto nivel que se complementa con un código sencillo y multitud de bibliotecas orientadas a problemas de un amplio espectro (Python, 2020). Debido a estos puntos, Python es uno de los lenguajes de programación más extendidos en el análisis de datos (van Rossum, 2009).

El desarrollo en Python del proyecto ha tenido lugar en el entorno Jupyter Notebook y se ha centrado en el entrenamiento del modelo predictivo de Random Forest a través de la biblioteca Scikit-learn.

Jupyter Notebook es un entorno de desarrollo para Python orientado al análisis de datos, que permite desarrollar cuadernos de trabajo con el código y los resultados de las ejecuciones-

Scikit-learn es una biblioteca de Python centrada en algoritmos de *machine learning* y funciones para evaluar el rendimiento de estos algoritmos. El trabajo se ha centrado en el módulo del algoritmo Random Forest.

4.2 Origen de datos

El sector inmobiliario es, por lo explicado anteriormente, un sector económico importante y en el que los datos son especialmente relevantes a la hora del desarrollo del negocio. Es por esto por lo que son pocas las empresas dedicadas al sector que proveen de sus propios datos como "open data". Properati SA es un portal web inmobiliario argentino, dedicado a la publicación de anuncios de activos para transacciones de compra venta y alquiler y que opera en varios países de Latinoamérica, tales como Argentina, Perú o Colombia. Desde su apertura en 2012 se ha posicionado como un portal líder en el

sector latinoamericano y referente gracias a la publicación periódica que realiza de sus propios datos para el uso libre. (Properati SA, 2019)

El primer punto a tratar a la hora de recoger los datos es el de los campos disponibles y si se trata de campos de relevancia para el objetivo del trabajo. En el apartado “Data” de la web de Properati se puede ver un listado de los campos que están disponibles, encontrándose lo siguiente:

Campo	***Descripción***
type	Tipo de aviso (Propiedad, Desarrollo/Proyecto).
country	País en el que está publicado el aviso (Argentina, Uruguay, Colombia, Ecuador, Perú)
id	Identificador del aviso. No es único: si el aviso es actualizado por la inmobiliaria (nueva versión del aviso) se crea un nuevo registro con la misma id pero distintas fechas: de alta y de baja.
start_date	Fecha de alta del aviso.
end_date	Fecha de baja del aviso.
created_on	Fecha de alta de la primera versión del aviso.
place	Campos referidos a la ubicación de la propiedad o del desarrollo.
lat	Latitud.
lon	Longitud.
l1	Nivel administrativo 1: país.
l2	Nivel administrativo 2: usualmente provincia.
l3	Nivel administrativo 3: usualmente ciudad.
l4	Nivel administrativo 4: usualmente barrio.
property	Campos relativos a la propiedad (vacío si el aviso es de un desarrollo/proyecto).
operation	Tipo de operación (Venta, Alquiler).
type	Tipo de propiedad (Casa, Departamento, PH…).
rooms	Cantidad de ambientes (útil en Argentina).
bedrooms	Cantidad de dormitorios (útil en el resto de los países).
bathrooms	Cantidad de baños.
surface_total	Superficie total en m².
surface_covered	Superficie cubierta en m².
price	Precio publicado en el anuncio.

currency	Moneda del precio publicado.
price_period	Periodo del precio (Diario, Semanal, Mensual)
title	Título del anuncio.
description	Descripción del anuncio.
development	Campos relativos al desarrollo inmobiliario (vacío si el aviso es de una propiedad).
status	Estado del desarrollo (Terminado, En construcción, ...)
name	Nombre del desarrollo.
short_description	Descripción corta del anuncio.
description	Descripción del anuncio.

Tabla 1. Esquema de avisos (Properati SA, 2019)

Este esquema de datos está disponible para su descarga tanto en CSV como en Google BigQuery. Dado que según nos indica la empresa, los datos disponibles en CSV cuentan con limitaciones de volumetría y tipología, se decide realizar la descarga a través de Google BigQuery.

Se accede a Google BigQuery a través de Google Console, usando para ello un correo electrónico de Gmail. En la plataforma, Properati SA dispone de una tabla que contiene todos los avisos inmobiliarios publicados en su portal, con los campos indicados en la Tabla 2.

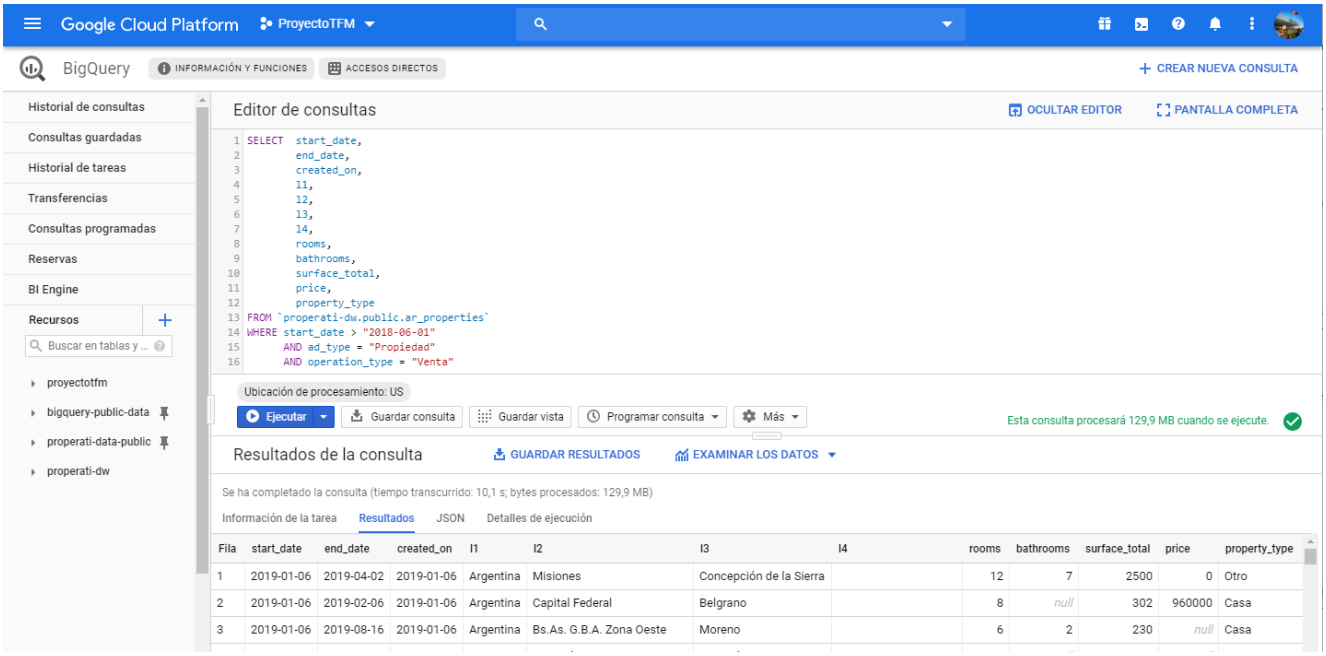

Figura 4. Ejemplo de consulta BigQuery a Properati SA

Para acotar el alcance, se decide filtrar la consulta a la tabla, de manera que el estudio se centre en un conjunto de datos más delimitado. Los filtros aplicados en la captura de datos son los siguientes:

- type: Propiedad
- country: Argentina.
- created_on: mayor de 01/06/2018
- Operación: Venta
- Moneda: USD

Además de aplicar los filtros descritos, tras analizar la descripción y posible utilidad en el estudio de los campos disponibles se decide realizar la extracción sobre los siguientes campos: start_date, end_date , created_on, l1, l2, l3, l4, type, rooms, bathrooms, surface_total, price. De estos campos, el campo price será la variable a predecir en el modelo y el resto serán las variables predictoras.

Una vez se ha generado la consulta, se selecciona la opción de Guardar resultados que realiza una exportación de datos a un fichero csv con el formato de la query realizada. A partir de esto el siguiente paso será cargar dicho fichero csv en la base de datos.

4.3 Procesado de los datos

La interfaz de Google Cloud Platform, a través de la cual se han obtenido los datos, permite realizar un estudio de los resultados de las consultas realizadas a través de su aplicación Data Studio. En este contexto, esta aplicación resulta de gran utilidad para realizar visualizaciones ad hoc de los resultados obtenidos en la consulta de manera rápida. Sin embargo, se ha optado por no usar dicha aplicación debido a que los datos necesitan un procesado previo para establecer mayor calidad en el dato.

Todo el procesado será realizado en base de datos mediante SQL y una vez alcanzada la calidad del dato deseada, se procederá a usar Tableau para explorar los datos y comprender sus características mediante visualizaciones, con el objetivo de extraer el conjunto de datos de entrenamiento de los algoritmos.

4.3.1 Carga a base de datos

A pesar de disponer de los datos en un fichero de texto plano csv, se decide cargar la información en base de datos. Python permite el tratamiento datos en csv gracias a

distintas librerías y es una opción que se plantea a la hora de abordar la fase del procesado de los datos, sin embargo, se decide cargar la información en base de datos por diversos motivos:

- Facilitar el acceso a los datos a través del gestor de base de datos: el lenguaje SQL permite acceder rápidamente a los datos necesarios, además de modificarlos y actualizarlos.

- Centralizar los datos usados en el desarrollo del proyecto en tablas en la misma base de datos: si se modifican los datos con distintos propósitos se pueden guardar en diferentes tablas y disponer de todas ellas en un repositorio accesible y unificado.

- Ayudar a la calidad del dato: dado que la extracción realizada es un conjunto de datos estructurado, las restricciones de calidad del dato que aporta una base de datos, tales como los tipos de dato o el evitar duplicados, son de utilidad para el procesado de la información.

La carga inicial se realiza en una tabla llamada dbo.TBL_STAGE, la cual se crea en la base de datos con los siguientes campos:

Campo	***Formato***
start_date	date, null
end_date	date, null
created_on	date, null
l1	nvarchar(max), null
l2	nvarchar(max), null
l3	nvarchar(max), null
l4	nvarchar(max), null
rooms	int, null
bathrooms	int, null
surface_total	int, null
price	float, null
property_type	nvarchar(max), null

Tabla 2. Campos y formato de tabla stage

Debido a que la extracción se realizó desde BigQuery, se puede realizar la carga inicial en una tabla con los formatos de campo ya establecidos, en lugar de cargar previamente a una tabla con todos los campos en formato texto. Esto es debido a que BigQuery contiene los datos en tablas con esquemas definidos, por lo que se replica el esquema de origen en la tabla de carga.

4.3.2 Calidad del dato

Para continuar, se debe establecer un marco de calidad del dato consistente en una serie de validaciones sobre la integridad de los datos. Se aplican las siguientes validaciones a los campos:

Campo	***Validaciones***
l1	• Longitud menor a 50 caracteres. • No nulo. • Distinto de vacío.
l2	• Longitud menor a 50 caracteres. • No nulo. • Distinto de vacío.
l3	• Longitud menor a 250 caracteres. • No nulo. • Distinto de vacío.
l4	• Longitud menor a 250 caracteres. • No nulo. • Distinto de vacío.
rooms	• Mayor que 0. • No nulo.
bathrooms	• Mayor que 0. • No nulo.
surface_total	• Mayor que 10. • No nulo.
price	• Mayor que 5000. • No nulo.
property_type	• Longitud menor a 50 caracteres. • No nulo. • Distinto de vacío.

Tabla 3. Marco de calidad del dato.

Dichas validaciones se aplican en forma de filtros, de forma que aquellos registros que no cumplan con todas las validaciones son excluidos del conjunto de datos.

Por último, se realiza la normalización de los campos de texto de la tabla. Debido a que los datos proceden de la web de Properati SA, los problemas en la normalización de los campos de texto son improbables, sin embargo, es necesario realizar la comprobación previa de que estos campos están correctamente normalizados.

Comprobar la normalización es una tarea complicada y en este caso se ha decidido aplicar algoritmos de clustering sobre cada campo de texto de la tabla dbo.TBL_STAGE de tal manera que se realicen clústeres agrupando por similitud del

texto en base a diferentes algoritmos. Para aplicar estas técnicas se ha usado la aplicación open source OpenRefine.

Tras aplicar diferentes algoritmos sobre los campos l1, l2, l3, l4 y property_type se detectan únicamente problemas de normalización en 2 categorías de los campos l3 y l4, referentes a las tildes ortográficas.

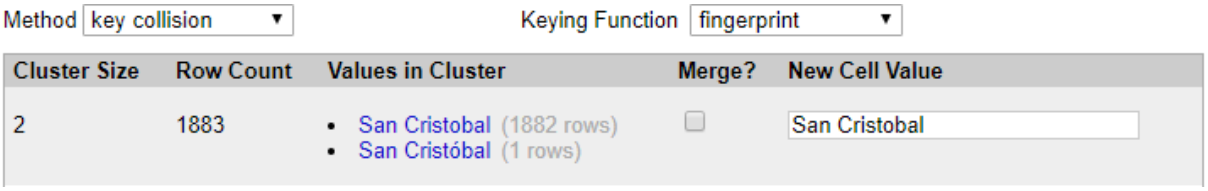

Figura 5. Algoritmo de clustering sobre campo l3

Una vez aplicada la normalización de los campos y el marco de calidad del dato, se carga el perímetro de registros resultante a una tabla llamada: dbo.PERIMETRO_INMUEBLES, renombrando los campos:

Campo origen	***Campo destino***	***Formato destino***
created_on	Fecha	date, not null
price	Precio	float, not null
rooms	Habitaciones	int, not null
bathrooms	Baños	int, not null
surface_total	Superficie	int, not null
l3	Ciudad	nvarchar(250), not null
l4	Barrio	nvarchar(250), not null
property_type	Tipo	nvarchar(50), not null

Tabla 4. Campos tabla final

El código de las transformaciones SQL realizadas está disponible en el Anexo 7.1 Calidad del dato.

4.3.2 Análisis de los datos

En el siguiente punto se desarrolla el análisis de los datos de los que se dispone una vez se ha alcanzado una calidad en el dato aceptable. Se utiliza SQL y Tableau en el cálculo de medidas y visualizaciones de los datos, para conocer la distribución y características de cada uno de los campos que se van a utilizar. Conocer los datos nos permitirá además acotar el dataset final para tener una volumetría que se ajuste a las capacidades de hardware disponibles en el proyecto.

En primer lugar, es necesario ver la distribución de los datos por cada uno de los campos para conocer sus características principales. Una vez hecho esto, se tiene que estudiar la posible correlación de los datos, para finalmente poder preparar un conjunto de datos de entrenamiento que sea representativo y suficientemente grande.

Dentro del conjunto de datos se dispone de 8 campos, de los cuales, 1 es una variable de fecha, 3 son variables categóricas y 4 son variables numéricas.

La **variable de fecha** se distribuye de forma homogénea desde julio de 2018 hasta diciembre de 2019, el rango de fechas seleccionado.

Las **variables categóricas** son Ciudad, Barrio y Tipo:

- Hay un total de 650 ciudades en el conjunto de datos, distribuidas por toda Argentina, habiendo una media de 25 ciudades por provincia, siendo Córdoba la provincia con más ciudades, 121 en total y La Rioja la provincia con menos ciudades, 1 en total. Se observa por lo tanto una desviación estándar muy alta, con un valor de 29.4.

- Hay un total de 720 divisiones municipales o barrios, habiendo una media de 2 barrios por ciudad, siendo Pilar la ciudad con más barrios, 113 en total y habiendo 598 ciudades con registros en un único barrio. Se observa por lo tanto una desviación estándar con un valor de 6.5.

- Hay un total de 10 tipos de activos inmobiliarios. Departamento y Casa son las tipologías más comunes, habiendo 228,6 mil registros de tipo Departamento y 72,6 mil registros de tipo Casa. Las tipologías menos comunes son Depósito y Cochera con 69 y 34 registros respectivamente.

Las **variables numéricas** son Habitaciones, Baños, Superficie y la variable objetivo, Precio. La siguiente tabla recoge las principales medidas de la distribución estadística de dichas variables:

Campo	Media	Desviación Típica	Mínimo	Percentil 25	Percentil 50	Percentil 75	Máximo
Baños	1.68	0.97	1	1	1	2	20
Habitaciones	3.2	1.59	1	2	3	4	35
Superficie	338.18	3165.312	10	57	95	207	193549
Precio	308680	1196788.48	5000	99000	157000	265000	38444052

Tabla 5. Medidas de variables numéricas

La tabla muestra como claramente existen valores atípicos en la parte final de la distribución, indicando que la distribución de la cantidad de activos por cada uno de estos campos es asimétrica. Será necesario quitar estos valores atípicos del dataset de entrenamiento para evitar que se produzcan dispersiones en los algoritmos.

Para continuar con el análisis, se van a establecer unas consideraciones iniciales con relación a la correlación existente entre los datos y se procederá a comprobar que dichas consideraciones se cumplen:

- El número de habitaciones tiene una proporcionalidad directa con el precio del activo.
- El número de habitaciones tiene una proporcionalidad directa con el precio del activo.
- Las habitaciones y los baños están relacionado con la superficie de forma directamente proporcional.
- El precio sufre una tendencia alcista a lo largo del tiempo debido a la acción de la alta inflación en Argentina, lo cual puede provocar distorsiones en periodos de tiempo grandes.

Para comprobar estos puntos, se conecta la base de datos a Tableau para visualizar la distribución de los datos pivotando los campos según las relaciones que se necesitan ver.

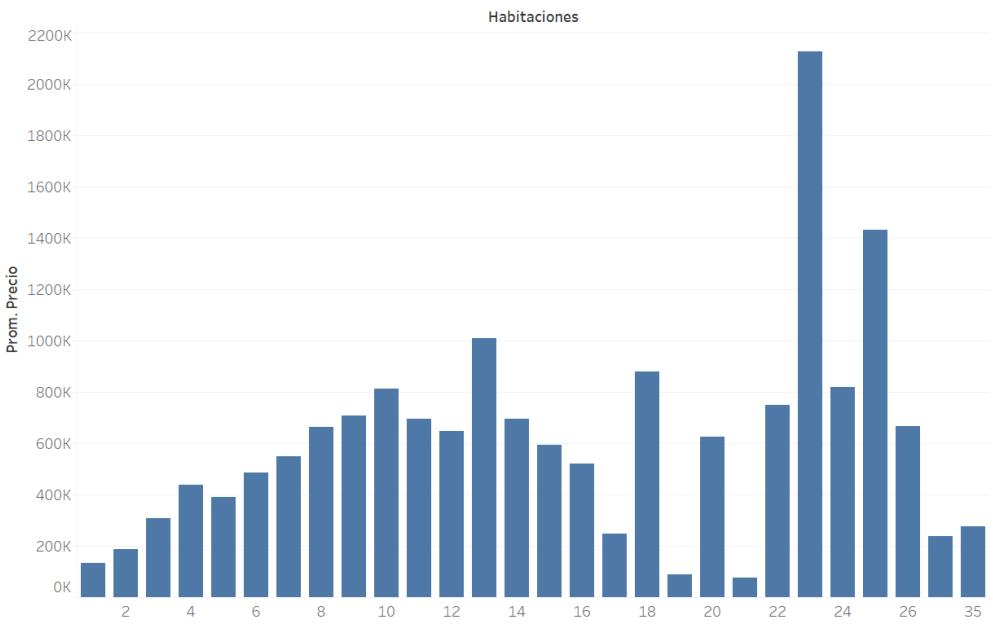

Figura 6. Precio promedio por número de habitaciones

Se observa en la Figura 6 una tendencia ascendente acorde a lo que se había asumido en las consideraciones iniciales hasta las 10 habitaciones. A partir de las 10 habitaciones el precio no sigue ninguna distribución reconocible. Esto puede deberse a

que la cantidad de activos con un número de habitaciones superior a 10 no es representativo y suponen valores típicos.

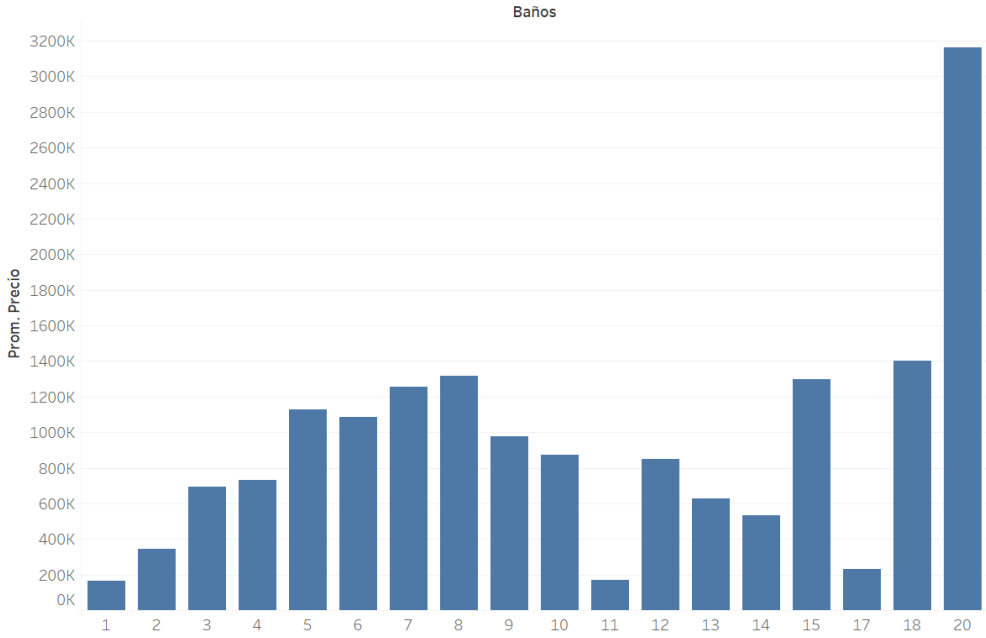

Figura 7. Precio promedio por número de baños

De forma análoga a las habitaciones, el precio medio que se ve en la Figura 7 es ascendente de forma directamente proporcional a la cantidad de baño hasta 8. A partir de 8 baños el precio no sigue ninguna distribución reconocible.

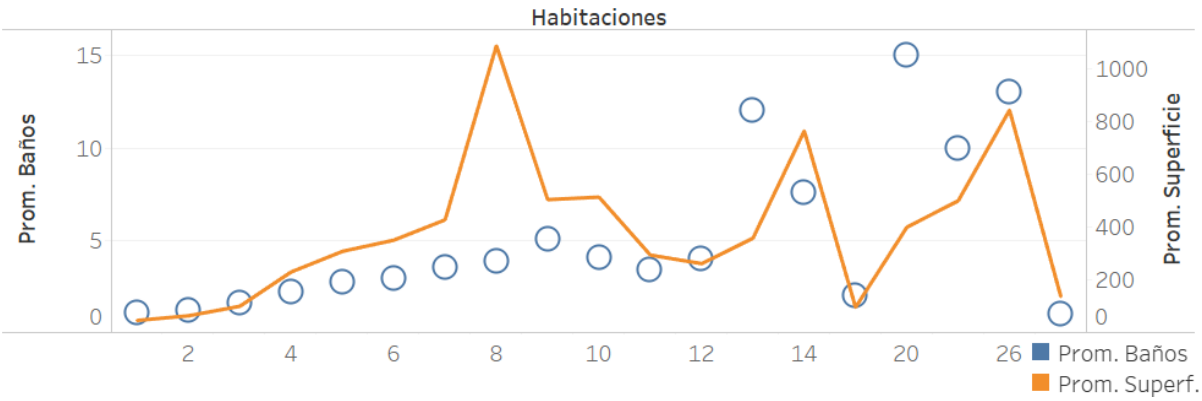

Figura 8. Promedio de baños y superficie por habitaciones

En la Figura 8 se muestra la relación entre el número de habitaciones, el número de baños y la superficie promedio del conjunto de datos. Nuevamente se observa que para valores menores a 10 habitaciones la correlación es alta, como se asumía en las

consideraciones iniciales.

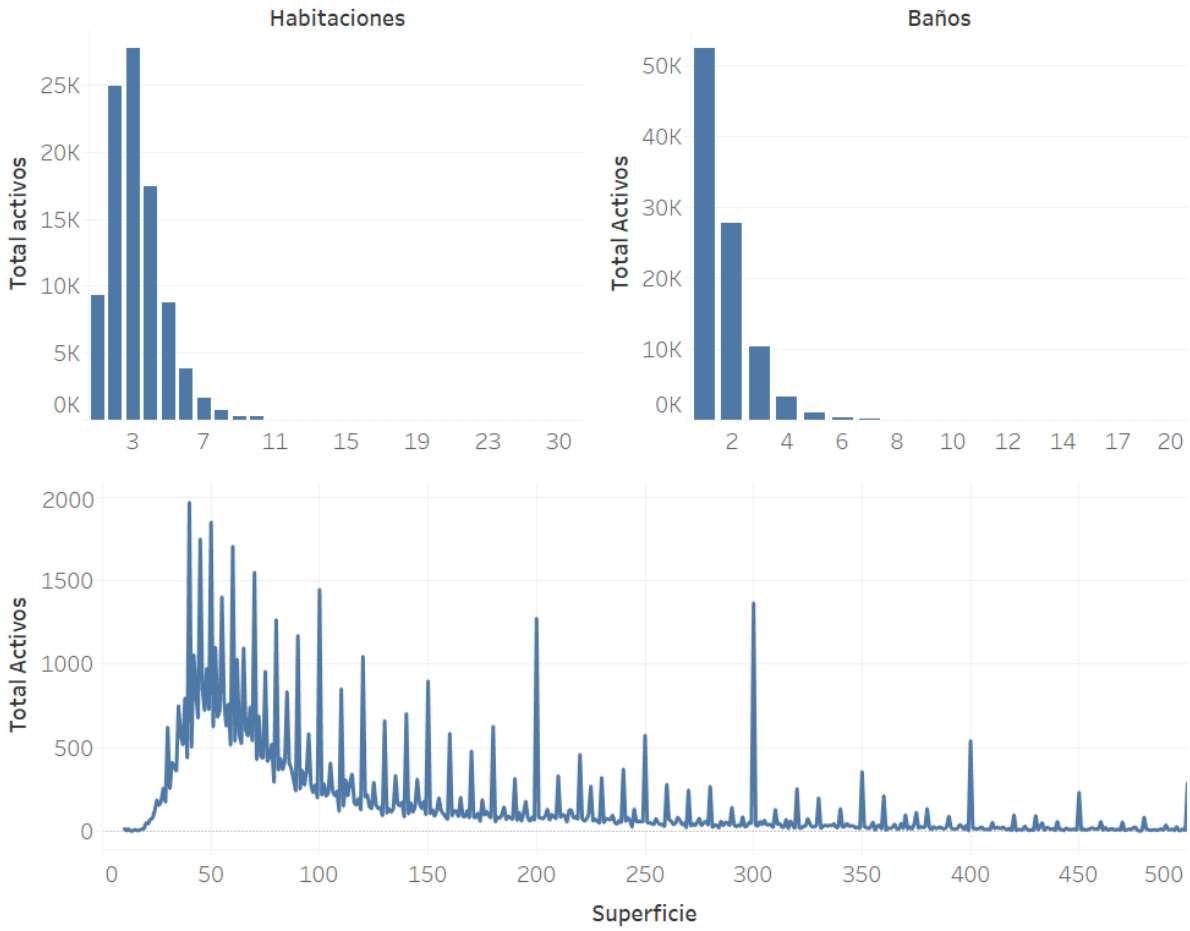

Figura 9. Distribuciones por total de activos

Como se podía ver en la Tabla 5, la distribución de la cantidad de activos para Habitaciones, Baños y Superficie de la Figura 9 es asimétrica positiva, con una cola hacia la derecha muy alargada.

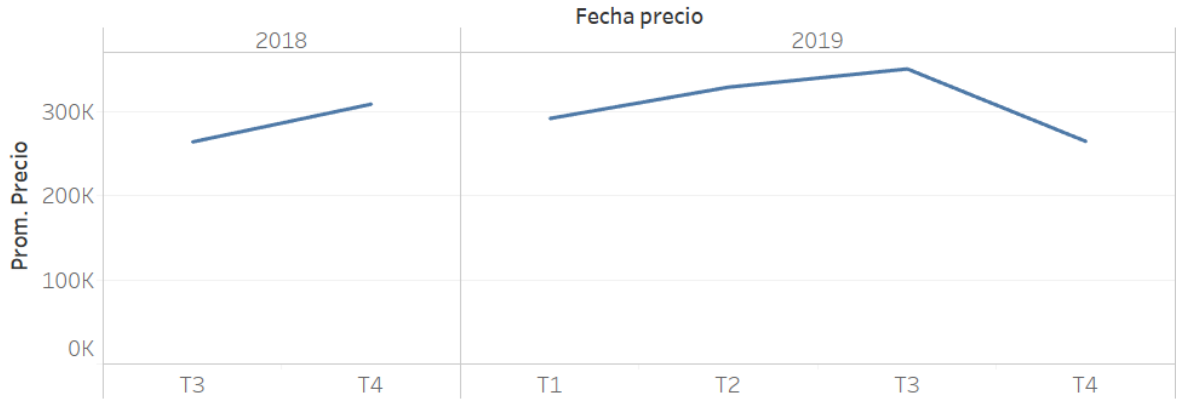

Figura 10. Precio promedio por trimestre

Por otro lado, el precio medio por trimestre muestra una tendencia ligeramente ascendente, como se puede ver en la Figura 10, sin embargo, no se muestra

suficientemente significativa como para que la inflación pueda influir de forma negativa en la fase de aplicación de los algoritmos.

Para obtener un dataset más representativo y evitar que los valores atípicos distorsionen el resultado de los algoritmos aplicados, se procederá a excluir los activos inmobiliarios con más de 10 habitaciones o más de 9 baños y los activos inmobiliarios con una superficie mayor a 1000 metros cuadrados.

Por último, como se indicó al inicio de la sección, la aplicación de los algoritmos se realizará sobre un subconjunto que sea representativo y con un volumen de datos menor, de forma que sea más abordable con respecto a las limitaciones computacionales del proyecto.

Para obtener el dataset final, junto con los filtros para evitar los valores atípicos, se aplica un filtro adicional sobre las ciudades en las que se encuentran los activos inmobiliarios, teniendo en total los siguientes filtros sobre el conjunto de datos:

- 2 ciudades: Vicente López y Palermo, entre las cuales existen registros pertenecientes a 13 barrios distintos. Dichas ciudades tienen un número suficientemente alto de activos inmobiliarios y la distribución del total de activos en cada ciudad es suficientemente homogénea y representativa entre cada uno de sus barrios, tal y como se puede ver en la Figura 10.
- Tipología del activo distinta a *Lote* y *Otro.* Son 2 tipologías con pocos activos inmobiliarios y una dispersión muy alta debido a que son categorías que recogen diferentes tipologías.
- Superficie del activo inmobiliario menor a 1000 metros cuadrados.
- Activos con menos de 10 habitaciones.
- Activos con menos de 9 baños.

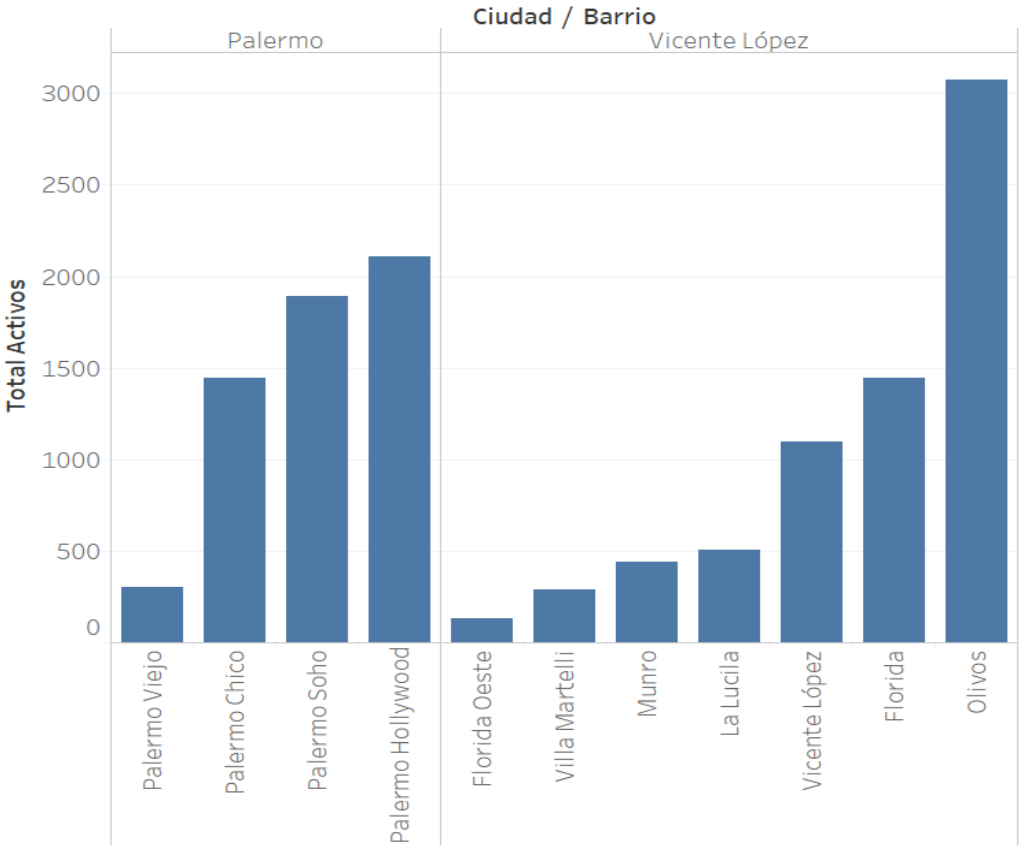

Figura 11. Distribución del total de activos por barrio

Las medidas de distribución estadística del dataset resultante son las siguientes:

Campo	Media	Desviación Típica	Mínimo	Percentil 25	Percentil 50	Percentil 75	Máximo
Baños	1.65	0.91	1	1	1	2	7
Habitaciones	2.93	1.4	1	2	3	4	10
Superficie	110.53	94.06	12	52	77	135	980
Precio	353844.24	493587	6000	151179	230000	359007	13800600

Tabla 6. Medidas de variables numéricas del dataset final

Se puede observar en la Tabla 6 como estas medidas tienen una desviación típica menor con respecto a las calculadas en el conjunto de datos original en la Tabla 5. Se conservan similares las medias y los valores máximos están acotados por los filtros que se han puesto. Esto indica que la dispersión en este nuevo conjunto de datos es menor y el resultado de los algoritmos será más preciso.

Con estos criterios se obtiene un conjunto de datos final con 12,9 mil registros, a partir del cual se crearán los datasets de entrenamiento y de test con los que se trabajará en el siguiente punto. Las variables predictoras usadas en el algoritmo serán Ciudad, Barrio, Tipología, Habitaciones, Baños y Superficie y la variable respuesta será el precio del activo inmobiliario.

4.4 Modelo Predictivo

El trabajo realizado hasta ahora ha consistido en capturar los datos, establecer unas restricciones y transformaciones de calidad del dato y por último el análisis de los datos disponibles con el objetivo de obtener un dataset óptimo para la tarea que se va a realizar a continuación.

Para la ejecución de esta parte del trabajo se usará, tal y como se indicó en el punto 4.1, un entorno de trabajo formado por cuadernos de Jupyter como IDE para el desarrollo del código en Python, y las bibliotecas Pandas, para manipulación de datos y Scikit-learn de machine learning, la cual dispone de los algoritmos y distintas funciones orientadas a evaluar el rendimiento de los modelos.

El dataset de activos inmobiliarios se cargará en Jupyter. A partir de este, se generarán 2 datasets para entrenamiento y validación y posteriormente se entrenará un modelo a través del algoritmo Random Forest. Por último, se probarán varias parametrizaciones diferentes para evaluar cual es mejor.

El código a través del cual se desarrollan los modelos está disponible en el Anexo.

4.4.1 Codificación One-hot

En primer lugar, se guarda el dataset generado en el punto 4.3 en formato csv. Tal como se ha visto, el dataset cuenta con 3 variables categóricas (Ciudad, Barrio, Tipología), 3 variables numéricas (Habitaciones, Baños y Superficie) y la variable respuesta, de formato numérico.

Para la aplicación de los algoritmos disponibles en la librería Scikit-learn será necesario codificar las variables categóricas a vectores *dummy* mediante una codificación One-hot, en la que cada variable categórica se transforma en tantas variables como valores tenga y las únicas combinaciones posibles son aquellas en las que un único valor es igual a 1 y el resto es igual a 0.

Esta codificación es necesaria debido a que el algoritmo que se va a usar únicamente tiene como inputs variables numéricas. Una codificación numérica estándar en la que cada categoría es un número no es adecuada, debido a que el algoritmo realiza comparaciones entre los valores numéricos.

Vicente López	1	0
Palermo	0	1

Tabla 120. Ejemplo de codificación One-hot para variable Ciudad

Pandas dispone de una función para codificar de esta forma las variables categóricas. Para realizar la codificación se carga el fichero CSV con el dataset, mediante la función *read_csv*, generando de esta forma un *dataframe* de pandas con 8 campos. A continuación, se aplica la función *get_dummies*, que codifica las variables categóricas tal y como se ha explicado.

El resultado final es la transformación de las variables Ciudad, Barrio y Tipología en 3 vectores con las siguientes dimensiones:

- Ciudad: dimensión 2 (2 valores originales).
- Barrio: dimensión 13 (13 valores originales).
- Tipo: dimensión 5 (5 valores originales).

4.4.2 Datasets de entrenamiento y validación

Una vez se dispone de los datos codificados en un formato adecuado para el algoritmo que se va a usar, es necesario dividir el conjunto de datos en 2 subconjuntos, uno para entrenamiento y otro para validación.

Mediante la función presente en Scikit-learn *train_test_split,* perteneciente al módulo *model_selection*, se puede dividir un conjunto de datos. Para el modelo a desarrollar se genera el dataset de entrenamiento con un 80% de los datos originales y el dataset de validación con el 20% restante. El uso de esta función es indicado porque consigue que la generación de ambos datasets sea aleatoria y de esta forma se tiene un dataset de validación con una distribución similar, lo cual minimiza el error.

Si se desea repetir el entrenamiento con un nuevo conjunto de hiperparámetros, es necesario guardar los datasets de entrenamiento y validación en formato CSV y de esta forma disponer de los mismos datos para nuevas parametrizaciones del modelo.

4.4.3 Evaluación del rendimiento del modelo

A la hora de evaluar el rendimiento del modelo, es necesario establecer una serie de métricas que cuantifiquen cómo de bien se están ajustando las predicciones, junto con unas cotas sobre estos valores que indiquen que el modelo obtenido es suficientemente bueno.

La principal métrica que se va a evaluar es el error relativo medio, el cual se calcula tomando un estimador dado por la siguiente fórmula:

$$\frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i}$$

Donde y_i es el valor del precio de cada activo inmobiliario del conjunto de dados de validación, $\hat{y}_i$ es el valor del precio estimado para ese activo inmobiliario y N es el total de registros del conjunto de datos.

Este estimador mide la desviación en valor absoluto media de los precios estimados con respecto a los precios reales y es un indicador relativo del error en las predicciones en el modelo. El valor de la diferencia entre la predicción y el precio real está acotado entre 0 e infinito, por lo que el cociente con el precio real es un valor también acotado inferiormente por 0 y con un valor menor a 1 si la diferencia es menor al propio valor del activo inmobiliario y mayor a 1 si la diferencia es mayor que el precio real. El promedio de estos cocientes cuantifica por lo tanto la magnitud del promedio de los errores relativos.

Alternativamente se usará el estimador complementario:

$$1 - \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i}$$

Como un indicador de la precisión del modelo. Como se ha explicado, el valor del error está acotado inferiormente por 0, por lo que el valor de la precisión está acotado superiormente por 1, alcanzando este valor si todos los precios estimados son iguales al precio real. Si el error es mayor que 1, se tiene que en promedio las diferencias entre valores son superiores al propio valor real, por lo que se tendría una precisión negativa.

Otro conjunto de indicadores de utilidad para evaluar la correlación entre precio estimado y precio real son conjuntamente la pendiente de la recta de regresión y el coeficiente de correlación de Pearson de un precio sobre otro. La pendiente de la recta de regresión del precio real sobre el estimado viene dada por la siguiente fórmula:

$$\frac{\sum_{i=1}^{N}(y_i - \bar{y})(\hat{y}_i - \hat{\bar{y}})}{\sum_{i=1}^{N}(\hat{y}_i - \hat{\bar{y}})^2}$$

Dado que este valor mide la pendiente que toma la recta de regresión del precio real sobre el estimado (García Pérez & Vélez Ibarrola, 1993), por definición de los resultados, un valor más cercano a 1 indica un mejor rendimiento del modelo, puesto

que se necesita un resultado dado por una recta de regresión identidad, donde valor real = valor estimado: $\hat{y}_i = y_i$.

El coeficiente de correlación de Pearson viene dado por la siguiente fórmula:

$$\frac{N\sum_{i=1}^{N} y_i\hat{y}_i - \sum_{i=1}^{N} y_i \sum_{i=1}^{N} \hat{y}_i}{\sqrt{N(\sum_{i=1}^{N} {y_i}^2 - \left(\sum_{i=1}^{N} y_i\right)^2}\sqrt{N(\sum_{i=1}^{N} {\hat{y}_i}^2 - \left(\sum_{i=1}^{N} \hat{y}_i\right)^2}}$$

Este coeficiente mide la correlación lineal entre dos variables (García Pérez & Vélez Ibarrola, 1993). A diferencia de la covarianza, este coeficiente está acotado entre -1, equivalente a correlación lineal inversa total, y 1, correlación lineal directa total. El valor 0 corresponde con total ausencia de correlación lineal. Debido a que se van a evaluar en este coeficiente 2 vectores de precios reales y predicciones, es de esperar que la correlación sea positiva. En este caso, como se ha explicado anteriormente, dado que el rendimiento perfecto del modelo es que $\hat{y}_i = y_i$ para cada registro, el resultado ideal es que la correlación entre el valor estimado y el valor real sea total y el valor del coeficiente alcance el valor 1.

De manera general, el coeficiente de correlación de Pearson y la pendiente de la recta de regresión miden en qué medida los resultados se ajustan a una recta y cuánto se acerca dicha recta a la recta identidad.

Dados los indicadores descritos anteriormente, se procede a realizar un cálculo del valor máximo de error aceptado. Este valor máximo será el indicador que determinará si el modelo entrenado es aceptado, en el caso de que el error del modelo sea menor, o no, en el caso contrario.

Para calcular el error máximo aceptado, se procede a realizar una estimación de los precios sencilla, basada en el precio medio agrupado por cada una de las variables predictoras: Ciudad, Barrio, Habitaciones, Baños y Superficie. De esta forma se tendrá un precio medio para cada una de las diferentes combinaciones de valores de las variables predictoras, el cual se procede a comparar con el precio real del activo inmobiliario con esas características.

Se calcula el precio medio sobre el conjunto de datos de entrenamiento y se cruza con el conjunto de datos de validación. Sobre este cruce, se calcula el error entre el precio real y el precio medio, dado por el indicador de error descrito. En el caso en el que un activo inmobiliario no tenga precio medio en la agrupación del dataset de

entrenamiento, se le da el valor por defecto de 0. Con estos cálculos se obtiene un valor de error de 0.359.

Por lo tanto, si la aplicación del algoritmo Random Forest al conjunto de datos genera un modelo con un error menor a 0.359, este modelo será aceptado, al mejorar la estimación sencilla que se ha realizado.

4.4.4 Random Forest

El algoritmo de Random Forest se trata actualmente de uno de los algoritmos de aprendizaje supervisado más usados. Fue presentado por primera vez en 2001 por el estadístico Leo Breiman.

Este algoritmo está basado en árboles de decisión, consistiendo en una combinación a través de la técnica de bagging, de dichos predictores, en el que cada árbol se entrena con un subconjunto aleatorio del dataset de entrenamiento, muestreado de forma independiente y con repetición (Breiman, 2001). Este algoritmo utiliza la técnica de bagging, consistente en la generación de distintas versiones de un predictor, las cuales se ensamblan para generar un algoritmo predictor agregado (Breiman, 1994).

Se puede aplicar la técnica de bagging a cualquier algoritmo de aprendizaje supervisado, aunque el algoritmo más frecuentemente usado es Random Forest debido a que la combinación de árboles de decisión mediante bagging genera unos resultados muy precisos con un gran rendimiento. Se puede demostrar que el error está acotado y es convergente cuando el número de árboles usados en el algoritmo crece (Breiman, 2001).

En la documentación de la biblioteca Scikit-learn (Scikit-learn, 2020) se muestran los algoritmos de Random Forest disponibles. La biblioteca cuenta con un clasificador, *RandomForestClassifier* para problemas de clasificación y un predictor, *RandomForestRegressor* para problemas de regresión. En el problema abordado por el proyecto se usará el algoritmo de regresión.

El siguiente paso es usar la función *RandomForestRegressor* para entrenar el modelo sobre el conjunto de datos de entrenamiento. Se dividen tanto el conjunto de entrenamiento como el de validación en 2 subconjuntos, el primero de los cuales contiene las variables input del modelo y el segundo la variable objetivo, el precio. El algoritmo recibe ambos subconjuntos durante el entrenamiento, al tratarse de aprendizaje supervisado, y en la fase de validación con el subconjunto de variables

predictoras se generar un vector de precios estimados por el modelo, el cual se compara con los precios reales para calcular el error y la precisión.

Entre los principales hiperparámetros que se pueden ajustar en el algoritmo, se encuentran los siguientes:

- Número de estimadores (por defecto = 10): número de árboles que se usarán en el modelo.
- Criterio (por defecto = 'mse'): función que mide la calidad de una división de un nodo. El valor por defecto es el error cuadrático medio (mse),
- Profundidad máxima (por defecto = sin profundidad máxima): profundidad máxima que se permite a cada árbol del algoritmo. Si no se fija un valor, los árboles crecerán hasta que todos sus nodos sean hojas, pudiendo darse un sobreajuste.
- Muestras mínimas para dividir el nodo (por defecto = 2).
- Muestras mínimas para ser nodo hoja (por defecto = 1): número mínimo de muestras requeridas para que el nodo sea una hoja.

Hay hiperparámetros que influyen directamente en la precisión del algoritmo, como el número de estimadores y otros que influyen en el sobreajuste del modelo, como la profundidad máxima o las muestras mínimas para ser nodo hoja. También existen hiperparámetros para modificar la configuración de los subconjuntos de datos para cada árbol de decisión. Para ver el resto de hiperparámetros del algoritmo, se puede consultar la documentación de la biblioteca (Scikit-learn, 2020): https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html#sklearn.ensemble.RandomForestRegressor

Uno de los puntos negativos del algoritmo Random Forest es que, a diferencia de otros algoritmos como las redes neuronales o los árboles de decisión, este no tiene una representación gráfica intuitiva. Está compuesto por cientos de árboles de decisión, los cuales pueden alcanzar cientos de nodos de profundidad. Scikit-learn ofrece funciones para exportar los árboles del modelo que se implementa a un formato texto que puede representarse gráficamente.

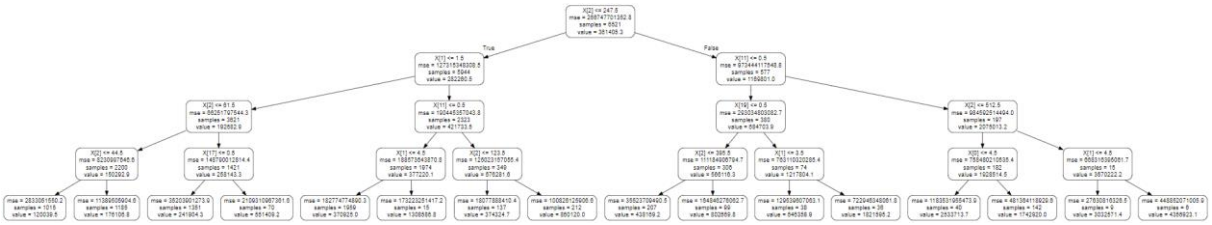

Figura 13. Árbol de decisión podado con profundidad 4

En la Figura 13 se muestra un árbol de decisión resultado de una implementación del algoritmo sobre los datos del proyecto. En esta implementación se estableció una limitación en la profundidad del árbol en 4 nodos máximo. Las implementaciones con profundidades sin acotar generan árboles de cientos de nodos de profundidad.

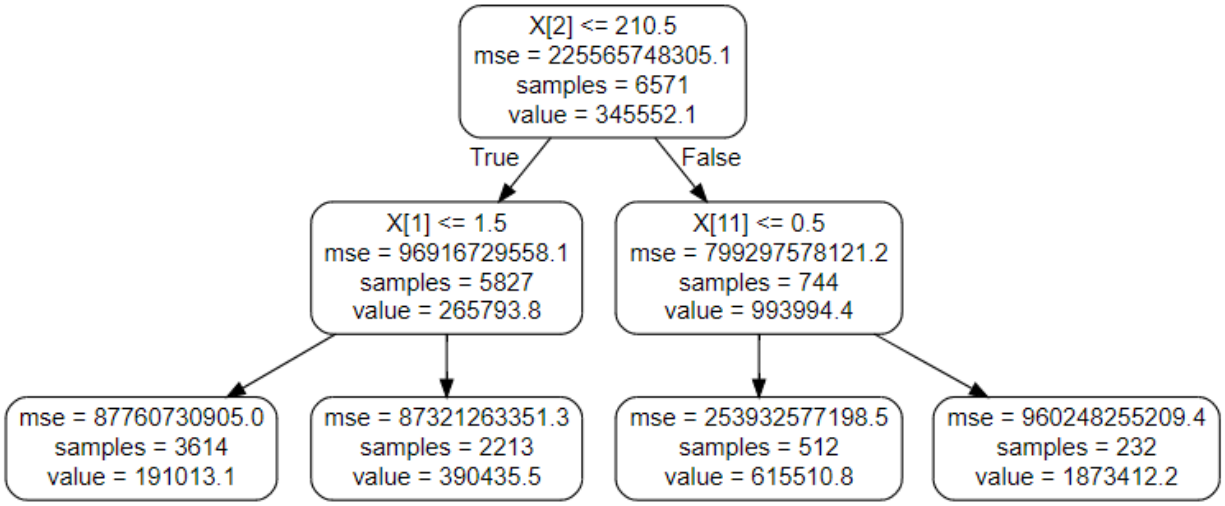

Figura 14. Árbol de decisión podado con profundidad 2.

En la Figura 14 se puede ver un árbol sobre el conjunto de datos, con profundidad máxima ajustada a 2 nodos. De esta forma, gracias a que el árbol es muy simple, se puede ver cómo el algoritmo genera árboles de decisión y cuáles son los criterios que va aplicando. En la Figura se muestra el nodo de arriba como el nodo raíz y los nodos de abajo como los nodos hoja. Adicionalmente, dentro de cada nodo se muestra, por orden de arriba abajo:

- Variable y condición sobre la variable a evaluar en la división del nodo. Las variables se muestran como un vector en el que X[1] es Habitaciones, X[2] es Baños y sucesivamente.
- Error cuadrático medio del nodo, la función de medida de la división de un nodo por defecto del algoritmo.
- Registros que contiene el nodo.

- Predicción del precio para cada uno de los activos que pertenecen al nodo.

Las divisiones de los nodos se realizan según desigualdades, por este motivo ha sido necesario realizar la codificación One-hot sobre las variables categóricas.

A continuación, se va a proceder a probar diferentes parametrizaciones sobre el conjunto de datos y evaluar el rendimiento y precisión de cada una de ellas en base a los indicadores de referencia. Adicionalmente, el algoritmo permite ver el peso de cada una de las variables de entrada en el modelo, por lo que se puede evaluar si algunas variables tienen un peso significativamente mayor a otras.

Las parametrizaciones se realizan pivotando sobre 3 hiperparámetros: número de estimadores, profundidad máxima de los árboles y muestras mínimas para ser nodo hoja. El resto de hiperparámetros quedan fijados con los valores por defecto de los hiperparámetros del algoritmo.

Parametrización 1

La primera parametrización del modelo se realizará con los valores por defecto indicados. Con esta parametrización el error esperado es alto, pues el número de árboles de decisión empleados es bajo:

- Número de estimadores: 10
- Profundidad máxima: sin acotar
- Muestras mínimas para ser nodo hoja: 1

El valor del error con esta parametrización es de 0,245, aproximadamente un 25% de error. Se tiene por lo tanto que la precisión del modelo es ligeramente superior al 75%. El valor del error es significativamente inferior al error máximo aceptado, por lo que se tiene que el algoritmo Random Forest genera un modelo mejor que las estimaciones basadas en la media.

Por otro lado, el coeficiente de Pearson es de 0,785, un valor de correlación directa bastante alto. Además, la pendiente de la recta de regresión es de 0,82, lo que indica que los precios predichos por el modelo son ligeramente inferiores a los precios reales, observando un sesgo en las predicciones.

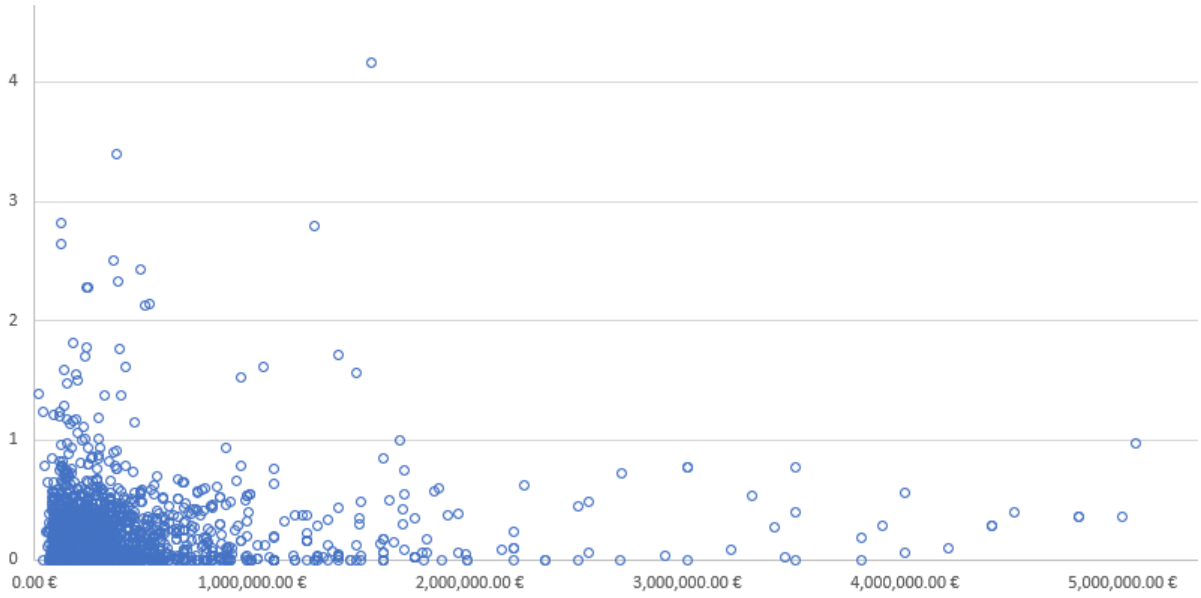

Figura 15. Distribución del error relativo por precio en parametrización 1.

En la Figura 15 se puede ver como el error relativo absoluto es mayor en precios más bajos. Esto posiblemente se debe a que los activos inmobiliarios con un precio mayor son menos frecuentes en el dataset y debido a sus características de precio tienen menos variedad en lo relativo a los campos del modelo tales como la situación geográfica, superficie o tipología.

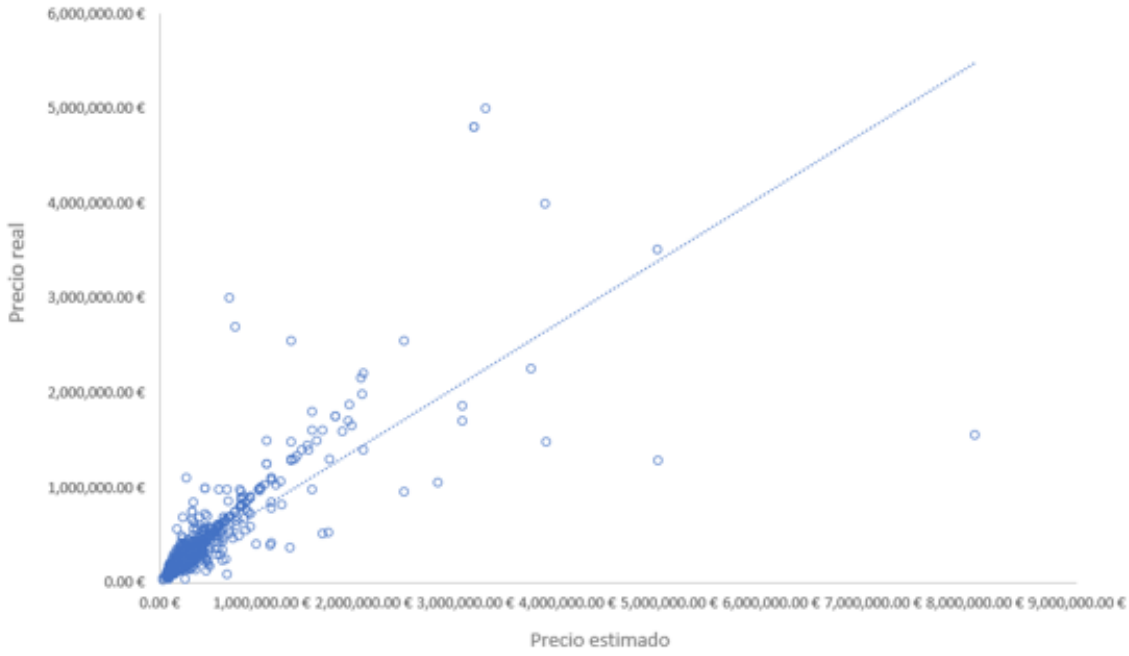

Figura 16. Correlación entre precio real y estimado en parametrización 1.

En la Figura 16 se muestra la distribución del precio real y el estimado. Como se ha calculado, el coeficiente de Pearson es alto y se puede ver como sigue una distribución lineal.

Parametrización 2

Se aumenta el número de estimadores para comprobar si de esta forma se reduce el error, manteniendo el resto de hiperparámetros fijos:

- Número de estimadores: 200
- Profundidad máxima: sin acotar
- Muestras mínimas para ser nodo hoja: 1

El valor del error con esta parametrización desciende a 0.199, un error de un 5% menos, reduciéndose a aproximadamente un 20% de error. El coeficiente de correlación de Pearson es de 0,923, muy superior al valor obtenido en la parametrización 1 y cercano a 1, mientras que la pendiente de la recta de regresión es 1 por lo que el ajuste lineal de los precios predichos y reales es casi exacto.

Estos valores indican que el modelo tiene un mejor rendimiento con esta parametrización y muestra una menor diferencia entre los precios estimados y los precios reales.

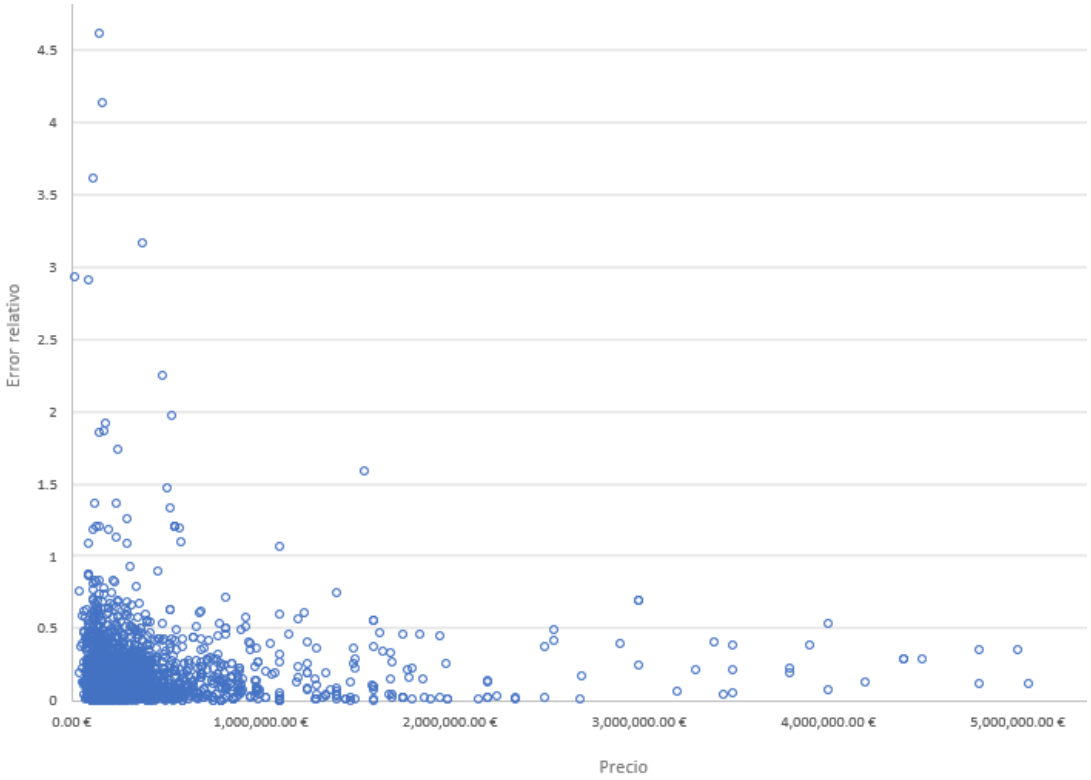

Figura 17. Distribución del error relativo por precio en parametrización 2.

En la Figura 17 se puede observar la misma tendencia que en la Figura 14 de la parametrización 1, con un mayor error en precios bajos. Además, se puede observar que el error relativo tiene mayor concentración en valores menores a 0.5, en comparación con la parametrización 1.

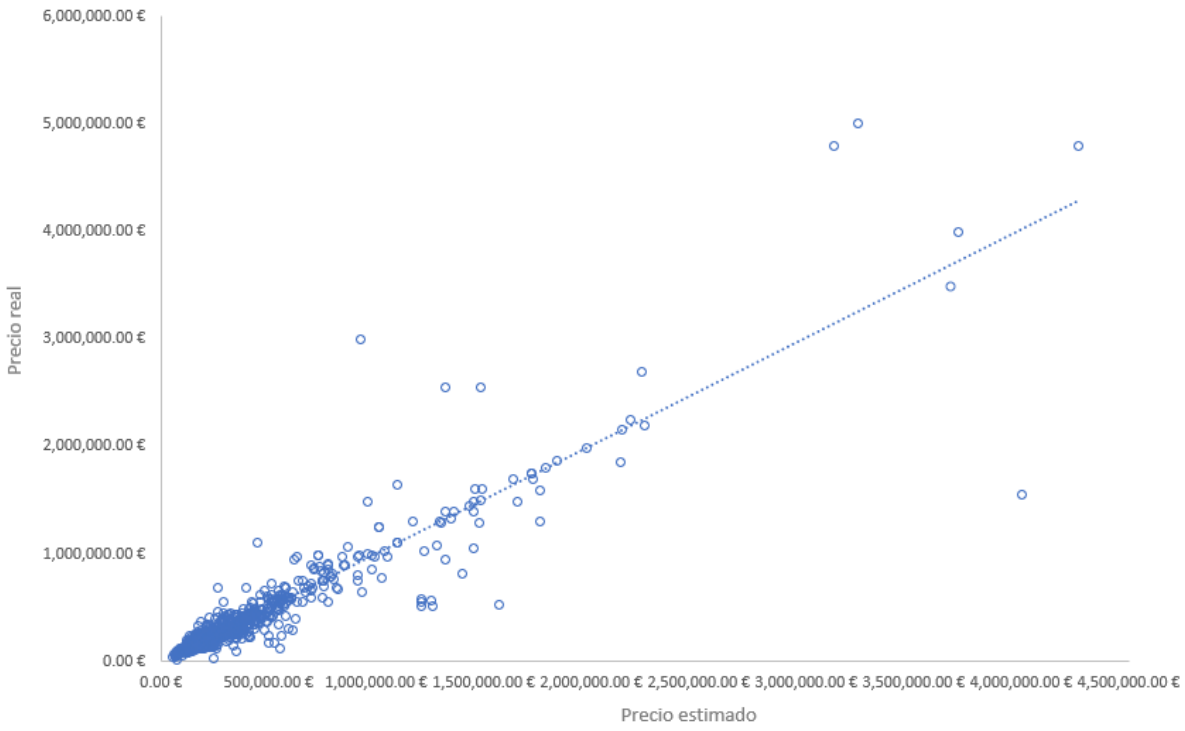

Figura 18.Correlación entre precio real y estimado en parametrización 2.

En la Figura 18 se muestra una mayor concentración de los precios en su línea de tendencia central, tal y como se espera por su valor del coeficiente de Pearson.

Parametrización 3

Dado que el error se ha reducido aumentando el número de estimadores, se realiza una segunda estimación con más estimadores para comprobar si el descenso del error continúa:

- Número de estimadores: 400
- Profundidad máxima: sin acotar
- Muestras mínimas para ser nodo hoja: 1

Con esta parametrización se alcanza un error de 0,178 y un coeficiente de Pearson de 0,89. Además la pendiente de la recta de correlación es de 1, por lo que se puede concluir que esta parametrización es la que menor error tiene y la que ajusta de forma más lineal los precios reales y predichos. Aunque se trata de la parametrización con el error más bajo de todas las realizadas, el descenso del error con respecto a la parametrización 2 es bajo en comparativa al aumento de estimadores que se ha realizado en el modelo.

En la siguiente Figura 19 se muestra la gráfica de distribución del error. Es significativa la mayor concentración de los errores relativos por debajo de la franja de 0,5 que se puede observar. Esta es una tendencia creciente según las parametrizaciones aumentan el número de estimadores.

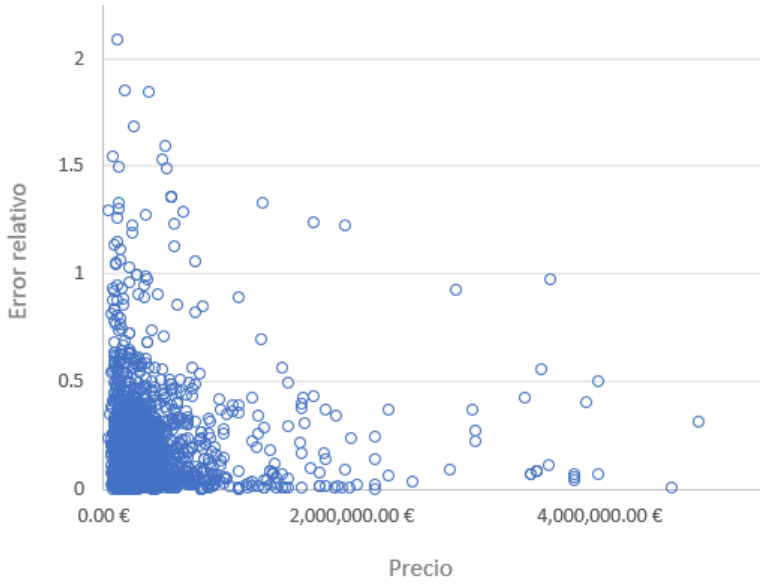

Figura 19. Distribución del error en la parametrización 3.

Parametrización 4

El aumento de estimadores no ha supuesto un descenso significativo del error en el modelo, por lo que el algoritmo puede estar sobreajustando el modelo. Para ello, se realiza una nueva parametrización con el mismo número de estimadores, pero limitando la profundidad máxima de los árboles a 10 y ajustando el tamaño del nodo para ser hoja a 5:

- Número de estimadores: 400
- Profundidad máxima: 10
- Muestras mínimas para ser nodo hoja: 5

Bajo esta parametrización el error asciende a 0,248, un error muy superior al que se estaba obteniendo en parametrizaciones anteriores. El cociente de correlación es de 0.89, un valor alto, pero también inferior al de parametrizaciones anteriores.

La hipótesis de que el modelo esté sobreajustando no parece acertada, dado que limitando la profundidad de los árboles el error aumenta. Se realizan unas pruebas adicionales con configuraciones paramétricas iguales, pero aumentando la

profundidad máxima de los árboles del algoritmo, de cara a confirmar el rechazo a la hipótesis. Ninguna de las pruebas realizadas supera el error de 0.199 obtenido en la parametrización 2, por lo que el algoritmo parece converger a un valor de error de aproximadamente 0,19 para el conjunto de datos y las variables del modelo.

Parametrización 5

Tomando la parametrización 3, la cual ha obtenido los mejores resultados de rendimiento, se va a analizar el peso que tiene cada una de las variables en el modelo. La biblioteca Sckit-learn tiene implementada una función que devuelve un vector con la importancia de cada variable en el algoritmo. Esta importancia mide cuánto mejora la predicción el añadir la variable al modelo (Scikit-learn, 2020).

La función usada es *feature_importances_* y debido a que las variables de entrada están codificadas One-hot, es necesario agrupar el vector resultante, sumando la importancia que cada variable dummy tiene sobre su variable categórica. El resultado es el siguiente:

- Habitaciones: 0.103126392
- Baños: 0.092266968
- Superficie: 0.54058988
- Ciudad: 0.003775523
- Barrio: 0.215752167
- Tipología: 0.04448907

Se puede apreciar que la superficie es la variable más determinante, con gran diferencia con respecto a la segunda variable en importancia, el barrio. Por el otro lado, la ciudad tiene una importancia significativamente menor al resto. Este punto es intuitivo al tener en cuenta que la superficie del activo está relacionada con las habitaciones y baños que este puede tener, reduciendo importancia a estas otras 2 variables. En la Figura 20 se puede ver gráficamente la distribución de la importancia de cada variable.

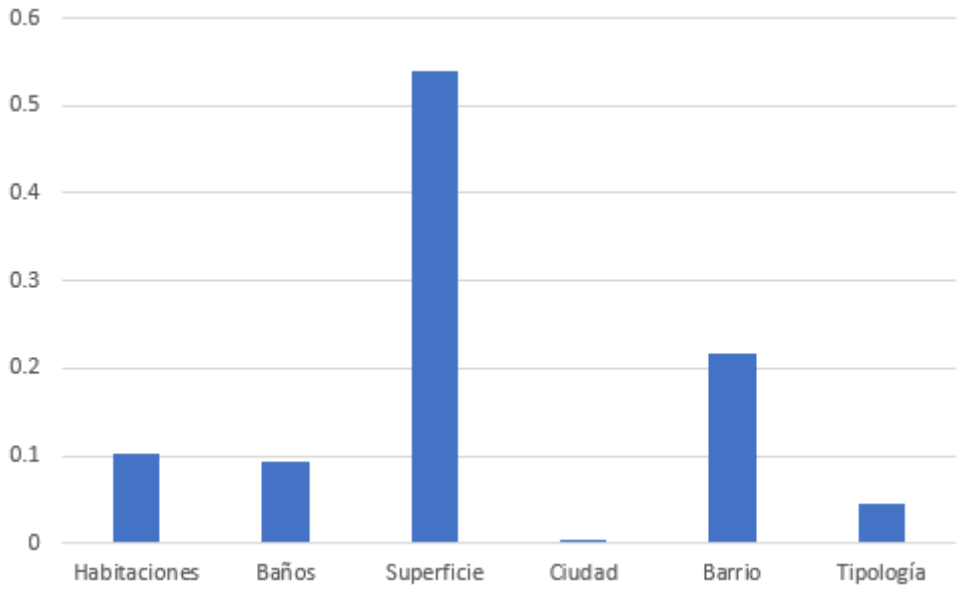

Figura 20. Importancia de cada variable en el modelo.

Dados estos resultados, se va a comprobar qué impacto tiene sobre el rendimiento del modelo el excluir la variable ciudad. El impacto esperado es reducido, debido a la importancia obtenida, por lo que, si el modelo alcanza un rendimiento equiparable con una variable menos, será mejor al requerir menos carga computacional y menos trabajo de recogida de datos.

Para ello, en primer lugar, es necesario excluir las 2 variables dummy en las que está codificada la variable Ciudad de los datasets de validación y entrenamiento. Una vez eliminadas estas variables, se ajusta el código para que el algoritmo entrene el modelo sobre los nuevos datos, realizándose la siguiente parametrización:

- Número de estimadores: 400
- Profundidad máxima: sin acotar
- Muestras mínimas para ser nodo hoja: 1
- Variables de entrada: Habitaciones, Baños, Superficie, Barrio y Tipología.

Entrenando el modelo con esta configuración sobre las variables indicadas, se obtiene un valor de error de 0.21, un valor inferior al de las parametrizaciones 1 y 4. Los valores del ajuste lineal de los precios son de 0,87 el coeficiente de correlación y 1,1 la pendiente de la recta de regresión, por lo que la correlación está en valores similares a las parametrizaciones anteriores, aunque la pendiente indica que el modelo estima los precios ligeramente por encima de los precios reales. Esto demuestra lo que indicaba el vector de importancias del algoritmo, la variable Ciudad no aporta una cantidad de información suficientemente relevante, siendo prácticamente indiferente en el modelo.

La variable Barrio aporta suficiente información geográfica. Este resultado es interesante debido a que se pueden presentar variantes del modelo que no informen de la ciudad y variantes que sí la recojan, en función del coste o la accesibilidad que suponga recoger la información de ciudad.

En la Figura 21 se muestra la gráfica de distribución de los errores, mostrando una distribución similar a la de la parametrización 1.

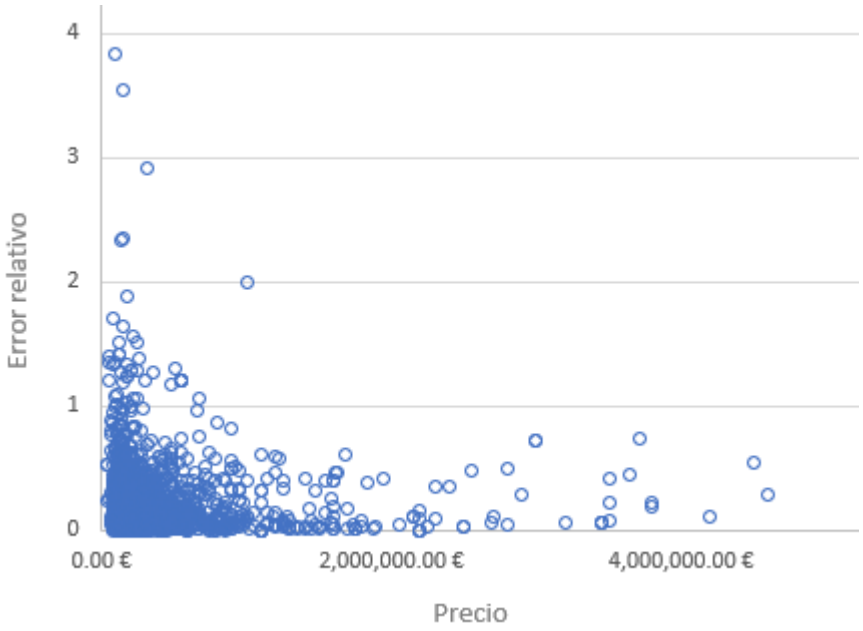

Figura 21. Distribución del error relativo por precio en parametrización 5.

A continuación, se muestra el vector de importancias de las variables usadas en el nuevo modelo y se puede ver en la en la Figura 22 la distribución de la importancia. La distribución no varía y es prácticamente igual a la mostrada en el modelo de 6 variables, debido a que la variable Ciudad tenía una importancia cercana a 0.

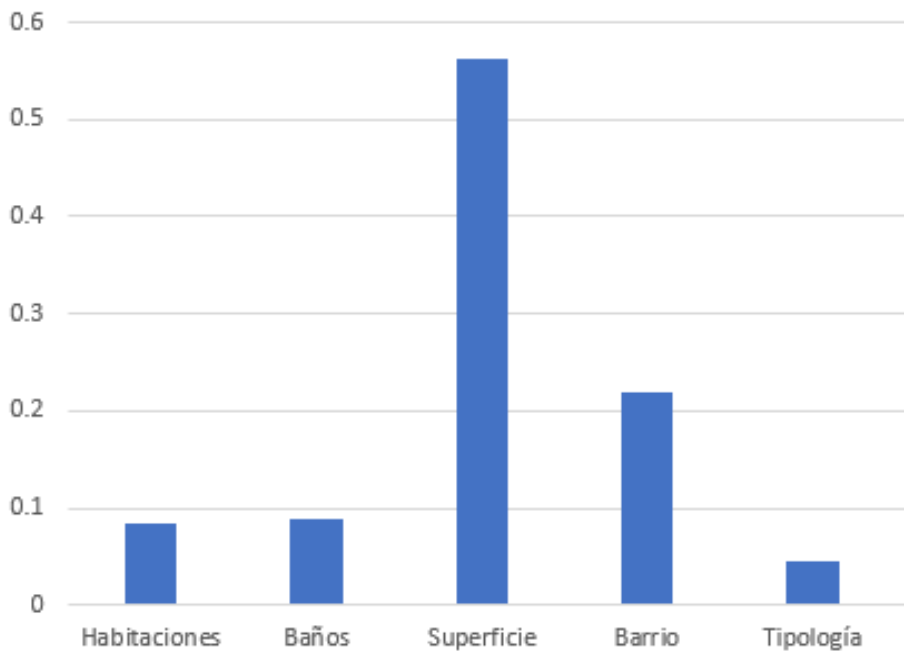

Figura 22. Importancia de las variables en el modelo excluyendo la ciudad.

- Habitaciones: 0.08437104
- Baños: 0.08829979
- Superficie: 0.5629956
- Barrio: 0.21884774
- Tipología: 0.04548583

5. Conclusiones y trabajo futuro

5.1. Conclusiones

La valoración de activos inmobiliarios supone en la actualidad uno de los retos de la economía, debido a la imperfección de los métodos de valoración actuales tales como las tasaciones. Existen muchas metodologías que cambian en función del objetivo de esta, lo cual afecta a la propia valoración, tanto en el periodo de validez como en el objetivo (Vedo Núnez, M., 2016).

Este trabajo se ha centrado en la construcción de un modelo de predicción de precios de referencia para activos inmobiliarios a través de variables relativas a dichos activos. Se ha identificado la necesidad de tener un método de valoración alternativo a los existentes, que constituyera una herramienta a través de la cual el usuario pueda saber de una forma eficiente y precisa cuál es el precio de su activo inmobiliario. Para ello se ha recopilado información de transacciones de compraventa (distribuida por Properati SA con licencia CC BY 3.0). A partir de esto se ha procedido a realizar una transformación y análisis de los datos para prepararlos como input del modelo. Por último, se ha usado el algoritmo de Random Forest para, a través del aprendizaje supervisado, obtener las predicciones.

De esta forma, se han planteado una serie de objetivos centrados en la obtención del modelo predictivo. El primer objetivo ha consistido en obtener una fuente de datos fiable y con información relativa a las transacciones. A continuación, el siguiente objetivo ha sido asegurar la calidad de los datos de forma que no existiesen valores atípicos que pudiesen alterar las predicciones, y que todos los datos estuviesen en el formato adecuado para el modelo propuesto. Por último, el objetivo principal ha sido aplicar los datos preparados en el modelo predictivo a través de Random Forest y evaluar los resultados.

Para el primer objetivo, se ha procedido a extraer los datos de la empresa Properati SA, la cual publica sus precios a través de Google BigQuery. Esta interfaz permite acceder a los datos de forma instantánea y realizar consultas SQL sobre la información publicada. Adicionalmente, la empresa tiene publicado un listado de los campos disponibles y su descripción, lo cual permite decidir qué campos son adecuados a los objetivos del trabajo. Tras analizar estos campos, se han incluido en el conjunto de datos del trabajo los siguientes: habitaciones, baños, superficie, ciudad, barrio y precio. Los 5

primeros campos se han incluido con el objetivo de ser las variables predictoras y el último con el objetivo de ser la variable predicha.

Tras descargar los datos, para el siguiente objetivo de evaluar la calidad del dato se ha desarrollado un marco de calidad. Este marco de calidad contiene unas validaciones sobre los tipos de dato y filtros para eliminar los valores atípicos del conjunto de datos del trabajo. Una vez se han aplicado estas validaciones, el siguiente paso ha sido realizar un análisis más exhaustivo de los datos, lo cual ha permitido detectar más valores atípicos. De esta forma, se ha obtenido un conjunto de datos final en el que la distribución de los datos se ajustase a unas hipótesis previas tales como que los activos con mayor superficie tuviesen mayor precio o que la superficie de los activos tuviese correlación directa con el número de habitaciones y el número de baños.

Por último, para el objetivo final se ha dividido el conjunto de datos en dos subconjuntos, uno de entrenamiento, sobre el cual se aplica el modelo y otro de test, sobre el cual se analizan los resultados del modelo. A continuación, se ha desarrollado un análisis del algoritmo que se ha utilizado. Este estudio del algoritmo ha permitido conocer los hiperparámetros y cuál es su efecto sobre el rendimiento del modelo y sobre los resultados en función del conjunto de datos. Finalmente, para la evaluación de los resultados del modelo se han establecido los indicadores de error y precisión, junto con el indicador de correlación de predicciones y junto con estos indicadores se ha procedido a establecer una cota máxima de error aceptable a través de aplicar dichos indicadores a un modelo de predicción básico basado en las medias. Estos indicadores se han aplicado en la ejecución de distintas configuraciones paramétricas del algoritmo, llevándose a cabo 5 configuraciones diferentes, en las cuales se han modificado los hiperparámetros de Random Forest y aplicado los indicadores descritos como medidas del rendimiento del modelo y de su aceptabilidad. El error más bajo obtenido ha sido de un 17,8%, lo cual ha superado ampliamente la cota superior de error aceptable del 35,9%, por lo tanto, los resultados han sido satisfactorios, aunque con un amplio margen de mejora.

Los resultados en las distintas parametrizaciones que se han realizado han mostrado en líneas generales que, para el dataset de estudio, el aumento de estimadores en el algoritmo, i.e., árboles de decisión, se traduce en un aumento de la precisión del modelo hasta un determinado número de estimadores, a partir de 400, en el que se ha detectado una ralentización en el descenso del error del modelo. Para descartar que este descenso en la reducción del error se debiese a un sobreajuste, se ha comprobado que el modelo no estuviese provocando sobreajuste al aumentar el número de estimadores. Esto se

ha comprobado a través de realizar una poda a los árboles de decisión del algoritmo, limitando la profundidad de estos. El resultado ha sido negativo, dado que, con el mismo número de árboles de decisión, limitando la profundidad de estos no se han mejorado los resultados, pudiéndose descartar la hipótesis del sobreajuste. Esto concuerda con los estudios de Leo Breiman, creador del algoritmo Random Forest, en los que demuestra que el aumento de estimadores no produce sobreajuste, pero el error converge a un valor mayor que 0 (Breiman, L. 2001).

Adicionalmente, el análisis del peso de las variables escogidas en el modelo ha mostrado unos resultados muy desiguales para cada una de las variables: mientras que la variable superficie obtiene un valor de 54%, el resto de variables obtienen valores muy inferiores, siendo la variable barrio la segunda en importancia con un 21%. La variable ciudad obtiene una importancia de un 0,37%, revelándose insignificante en el modelo. Estos resultados indican que la elección de las variables se puede optimizar, de manera que se incluyan variables que igualen los resultados de importancia de todas las variables y den como resultado un modelo más robusto.

De manera general, se puede concluir que la aplicación del algoritmo Random Forest al problema de la valoración de activos inmobiliarios ha sido exitosa, obteniéndose unos resultados alentadores de cara a ampliar el alcance de este trabajo en futuros estudios.

5.2. Líneas de trabajo futuro

Las líneas de trabajo que se abren tras analizar los resultados obtenidos se dividen principalmente en 2 caminos. Por un lado, está la posibilidad de realizar un estudio comparativo entre distintos modelos basados en algoritmos alternativos y evaluar el rendimiento de otros algoritmos. Por otro lado, se plantea la línea de expandir el modelo basado en Random Forest a través de ampliar las variables de entrada y el alcance del conjunto de datos de entrenamiento.

Los resultados del error obtenidos, junto con las propiedades del algoritmo Random Forest, llevan a plantear la continuación del estudio proponiendo el desarrollo de modelos alternativos basados en otros algoritmos. El objetivo es estudiar el error de otros modelos bajo las mismas condiciones aplicadas en este trabajo, para obtener resultados de error y evaluar si estos son más precisos. En esta línea se han desarrollado trabajos tales como el de Jian-Guo Liu, Xiao-Li Zhang y Wei-Ping Wu (Liu, J.G., Zhang, X.L. & Wu, W.P. (2006) : Application of Fuzzy Neural Network for Real

Estate Prediction) en el cual se propone un modelo basado en redes neuronales con unos resultados altamente favorables.

Por último, la importancia irregular que se ha obtenido de las variables en el modelo lleva a una segunda línea de trabajo futuro, consistente en el desarrollo de un nuevo modelo a través del ajuste de las variables de entrada. Adicionalmente, es necesario expandir el alcance del conjunto de datos para poder generalizar los resultados del modelo a otras áreas geográficas y aumentar el rango temporal con el objetivo de obtener más datos para el modelo. En el modelo actual se han obviado los datos macroeconómicos dentro de las variables de input, lo cual puede provocar desajustes en los resultados, especialmente si se amplía el rango temporal del conjunto de datos. Es por esto por lo que una de las líneas de continuación propuestas es el estudio de la importancia de la información macroeconómica a través del análisis de que cuáles son las variables que mejor codifican esta información, junto con la expansión geográfica de los datos de entrada.

6. Bibliografía

Breiman, L. (1994). Bagging Predictors

Breiman, L. (2001). Random Forests

Borrallo Egea, F.A. & Hierro Recio, L.A. (2015) Revista de Economía Mundial, núm. 41: La eficiencia de la política monetaria durante la crisis económica mundial.

Calvo, G.A., Leiderman, L. & Reinhart, C.M. (1993) Capital inflows and real exchange rate appreciation in Latin America: the role of external factors.

Chazallet, S. (2016). Python 3: los fundamentos del lenguaje. Barcelona, España: Ediciones ENI.

Domínguez Prost, E.R. (2019) Construcción de un índice de precios inmobiliario para la Ciudad Autónoma de Buenos Aires a partir de datos espaciales.

Finect (2020) Evolución histórica del euríbor. Recuperado el 10 de 02 de 2020 de https://www.finect.com/blogs/vivienda_e_inmobiliario/articulos/evolucion-historica-futura-euribor

Fotocasa. (2020). Índice inmobiliario Fotocasa. Recuperado el 10 de 02 de 2020 de https://www.fotocasa.es/indice/#/filter/eyJ0cmFuc2FjdGlvbiI6ImJ1eSJ9

García, J. (2008). De la quimera inmobiliaria al colapso financiero. Crónica de un desenlace anunciado.

García Pérez, A. & Vélez Ibarrola, R. (1993). Principios de inferencia estadística.

Google Cloud. (2020) Introducción a BigQuery. Recuperado el 10 de 01 de 2020 de https://cloud.google.com/bigquery/what-is-bigquery?hl=es-419

Gruber, G. (2016) La contribución de Properati al mercado inmobiliario argentino. https://medium.com/@gabrielgruber/la-contribuci%C3%B3n-de-properati-al-mercado inmobiliario-argentino-bdcd2043f21f

Hakala, J.O. (2016) Estudio sobre el mercado interbancario a un año en la Unión Europea

Hlaczikm, M.M. (2009) Tiempo en el mercado y precio de venta: un análisis para el mercado inmobiliario del partido de La Plata

IBM – Structured Query Language (SQL). (2020) Recuperado el 10 de 01 de 2020 de https://www.ibm.com/support/knowledgecenter/SSCJDQ/com.ibm.swg.im.dashdb.sql.ref.doc/doc/c0004100.html

Liu, J.G., Zhang, X.L. & Wu, W.P. (2006). Application of Fuzzy Neural Network for Real Estate Prediction.

Maure Inmobiliaria (2019). Evolución del valor del m2 según la UADE. Recuperado de www.maureinmobiliaria.com/evolucion-del-valor-del-m2-uade/

Montero, J.M. & Fernández-Avilés, G. (2017) La importancia de los efectos espaciales en la predicción del precio de la vivienda. Una aplicación geoestadística en España.

Obaid, M. (2008). Eficiencia en tasaciones dentro del mercado inmobiliario.

Ocampo, A. (2009) Latin America and the global financial crisis

Pérez, P. E, & Féliz, M. (2019) La crisis económica y sus impactos sobre la política de empleo e ingresos en Argentina

Pozzi, S. (2017). El País: Hipotecas subprime: La crisis con la que empezó todo.

Properati SA. (2019). Recuperado el 09 de 01 de 2019 de https://www.properati.com.ar/data/

Python (2020) applications for Python. Recuperado el 17 de 01 de 2020 de https://www.python.org/about/apps/

Roig Hernando, J., Gras Alomà, R., & Soriano Llobera, J. M. (2015). Análisis y pronóstico del precio de la vivienda en España: Modelo econométrico desde una perspectiva conductual.

van Rossum, G. (2009). Tutorial de Python.

Sanders, A, (2008). Journal of Housing Economics: The subprime crisis and its role in the financial crisis.

Scikit-learn. (2020). Ensemble methods. Recuperado el 15 de 01 de 2020 de https://scikit-learn.org/stable/modules/ensemble.html

Sehgal, M. (2017). Data Science Solutions: Laptop Startup to Cloud Scale Data Science Workflow.

Tong, H. & Wei, S. (2008). Real effects of the subprime mortgage crisis: is it a demand or a finance shock?

Vedo Núnez, M. (2016). Valoración contable de los inmovilizados inmobiliarios a través de modelos predictivos. Un análisis para el sector bancario.

Universidad Internacional de La Rioja. (2019). Análisis e interpretación de datos - Tema 3. Relación entre variables. La Rioja, España.

Universidad Internacional de La Rioja. (2019). Métodos de almacenamiento de información. La Rioja, España.

Universidad Internacional de La Rioja. (2019). Técnicas de Inteligencia Artificial – Tema 2. Árboles de decisión. La Rioja, España.

Anexos

Todo el código implementado en el desarrollo del proyecto se encuentra disponible en el siguiente enlace de GitHub:

https://github.com/DParedero/Masters_Dissertation

Anexo I. Código SQL

Análisis hipótesis previas

```
--TOTAL DE CIUDADES
-----------------------------------------------------------------
SELECT COUNT(DISTINCT Ciudad) TOT_CIUDADES
FROM dbo.PERIMETRO_INMUEBLES


--MEDIA Y DESVIACION TIPICA DE CIUDADES POR PROVINCIA
-----------------------------------------------------------------
SELECT  AVG(TOT_CIUDADES)
              ,STDEVP(TOT_CIUDADES)
FROM (
              SELECT  Provincia
                            ,COUNT(DISTINCT Ciudad) TOT_CIUDADES
              FROM dbo.PERIMETRO_INMUEBLES
              GROUP BY Provincia
       ) TBL


--PROVINCIA CON MÁS CIUDADES
-----------------------------------------------------------------
SELECT TOP 1 Provincia
                     ,COUNT(DISTINCT Ciudad) TOT_CIUDADES
FROM dbo.PERIMETRO_INMUEBLES
GROUP BY Provincia
ORDER BY 2 DESC



--TOTAL DE BARRIOS
-----------------------------------------------------------------
SELECT COUNT(DISTINCT Barrio) TOT_BARRIOS
FROM dbo.PERIMETRO_INMUEBLES



--MEDIA Y DESVIACION TIPICA DE BARRIOS POR CIUDAD
-----------------------------------------------------------------
SELECT  AVG(TOT_BARRIOS)
              ,STDEVP(TOT_BARRIOS)
FROM (
              SELECT  Barrio
                            ,COUNT(DISTINCT Barrio) TOT_BARRIOS
              FROM dbo.PERIMETRO_INMUEBLES
              GROUP BY Ciudad
       ) TBL
```

```
--CIUDAD CON MÁS BARRIOS
-----------------------------------------------------------------
SELECT TOP 1 Ciudad
                    ,COUNT(DISTINCT Barrio) TOT_BARRIOS
FROM dbo.PERIMETRO_INMUEBLES
GROUP BY Ciudad
having COUNT(DISTINCT Barrio) = 1
ORDER BY 1 DESC


--TOTAL DE TIPOLOGÍAS
-----------------------------------------------------------------
SELECT COUNT(DISTINCT Tipo) TOT_TIPO
FROM dbo.PERIMETRO_INMUEBLES


--TOTAL ACTIVOS POR TIPOLOGIA
-----------------------------------------------------------------
SELECT  Tipo
            ,COUNT(1) TOT_ACTIVOS
FROM dbo.PERIMETRO_INMUEBLES
GROUP BY Tipo
ORDER BY 2 DESC
```

Tabla de medidas de variables numéricas

```
--TABLA CAMPOS NUMERICOS
-----------------------------------------------------------------
SELECT 'Habitaciones' AS Campo
            ,Media
            ,[Desviación Típica]
            ,[Mínimo]
            ,[Percentil 25]
            ,[Percentil 50]
            ,[Percentil 75]
            ,[Máximo]
FROM
            (SELECT  Agrupador
                        ,AVG(Habitaciones*1.0)            AS Media
                        ,STDEVP(Habitaciones*1.0)  AS [Desviación Típica]
                        ,MIN(Habitaciones)                AS [Mínimo]
                        ,MAX(Habitaciones)                AS [Máximo]
             FROM dbo.PERIMETRO_INMUEBLES
             GROUP BY Agrupador
            ) Medidas

            LEFT JOIN
            (SELECT DISTINCT Agrupador
                                    ,PERCENTILE_DISC(0.25) WITHIN GROUP
(ORDER BY Habitaciones ASC) OVER (PARTITION BY Agrupador) as [Percentil 25]
                                    ,PERCENTILE_DISC(0.5) WITHIN GROUP
(ORDER BY Habitaciones ASC) OVER (PARTITION BY Agrupador) as [Percentil 50]
                                    ,PERCENTILE_DISC(0.75) WITHIN GROUP
(ORDER BY Habitaciones ASC) OVER (PARTITION BY Agrupador) as [Percentil 75]

             FROM dbo.PERIMETRO_INMUEBLES
            ) Percentiles
            ON Medidas.Agrupador = Percentiles.Agrupador
```

```
UNION

SELECT 'Baños' AS Campo
              ,Media
              ,[Desviación Típica]
              ,[Mínimo]
              ,[Percentil 25]
              ,[Percentil 50]
              ,[Percentil 75]
              ,[Máximo]
FROM
              (SELECT  Agrupador
                           ,AVG(Baños*1.0)            AS Media
                           ,STDEVP(Baños*1.0)  AS [Desviación Típica]
                           ,MIN(Baños)                AS [Mínimo]
                           ,MAX(Baños)                AS [Máximo]
               FROM dbo.PERIMETRO_INMUEBLES
               GROUP BY Agrupador
              ) Medidas

              LEFT JOIN
              (SELECT DISTINCT Agrupador
                                          ,PERCENTILE_DISC(0.25) WITHIN GROUP
(ORDER BY Baños ASC) OVER (PARTITION BY Agrupador) as [Percentil 25]
                                          ,PERCENTILE_DISC(0.5) WITHIN GROUP
(ORDER BY Baños ASC) OVER (PARTITION BY Agrupador) as [Percentil 50]
                                          ,PERCENTILE_DISC(0.75) WITHIN GROUP
(ORDER BY Baños ASC) OVER (PARTITION BY Agrupador) as [Percentil 75]

               FROM dbo.PERIMETRO_INMUEBLES
              ) Percentiles
              ON Medidas.Agrupador = Percentiles.Agrupador

UNION

SELECT 'Superficie' AS Campo
              ,Media
              ,[Desviación Típica]
              ,[Mínimo]
              ,[Percentil 25]
              ,[Percentil 50]
              ,[Percentil 75]
              ,[Máximo]
FROM
              (SELECT  Agrupador
                           ,AVG(Superficie*1.0)       AS Media
                           ,STDEVP(Superficie*1.0)    AS [Desviación Típica]
                           ,MIN(Superficie)                 AS [Mínimo]
                           ,MAX(Superficie)                 AS [Máximo]
               FROM dbo.PERIMETRO_INMUEBLES
               GROUP BY Agrupador
              ) Medidas

              LEFT JOIN
              (SELECT DISTINCT Agrupador
                                          ,PERCENTILE_DISC(0.25) WITHIN GROUP
(ORDER BY Superficie ASC) OVER (PARTITION BY Agrupador) as [Percentil 25]
                                          ,PERCENTILE_DISC(0.5) WITHIN GROUP
(ORDER BY Superficie ASC) OVER (PARTITION BY Agrupador) as [Percentil 50]
                                          ,PERCENTILE_DISC(0.75) WITHIN GROUP
(ORDER BY Superficie ASC) OVER (PARTITION BY Agrupador) as [Percentil 75]
```

```
             FROM dbo.PERIMETRO_INMUEBLES
            ) Percentiles
            ON Medidas.Agrupador = Percentiles.Agrupador

UNION

SELECT 'Precio' AS Campo
            ,Media
            ,[Desviación Típica]
            ,[Mínimo]
            ,[Percentil 25]
            ,[Percentil 50]
            ,[Percentil 75]
            ,[Máximo]
FROM
            (SELECT  Agrupador
                        ,AVG(Precio*1.0)           AS Media
                        ,STDEVP(Precio*1.0) AS [Desviación Típica]
                        ,MIN(Precio)               AS [Mínimo]
                        ,MAX(Precio)               AS [Máximo]
             FROM dbo.PERIMETRO_INMUEBLES
             GROUP BY Agrupador
            ) Medidas

            LEFT JOIN
            (SELECT DISTINCT Agrupador
                                    ,PERCENTILE_DISC(0.25) WITHIN GROUP
(ORDER BY Precio ASC) OVER (PARTITION BY Agrupador) as [Percentil 25]
                                    ,PERCENTILE_DISC(0.5) WITHIN GROUP
(ORDER BY Precio ASC) OVER (PARTITION BY Agrupador) as [Percentil 50]
                                    ,PERCENTILE_DISC(0.75) WITHIN GROUP
(ORDER BY Precio ASC) OVER (PARTITION BY Agrupador) as [Percentil 75]

             FROM dbo.PERIMETRO_INMUEBLES
            ) Percentiles
            ON Medidas.Agrupador = Percentiles.Agrupador
ORDER BY 1
```

Error predicción basada en la media

```
SELECT SUM(ABS(PREC.Precio -
COALESCE(ESTIM.Precio_MEDIO,0))*1.0/PREC.Precio)/COUNT(*)
FROM          dbo.TBL_FINAL PREC

            LEFT JOIN
            (SELECT        AVG(PRECIO) AS Precio_MEDIO
                        ,Habitaciones
                        ,Baños
                        ,Superficie
                        ,Barrio
                        ,CIUDAD
             FROM dbo.TBL_FINAL
             GROUP BY  Habitaciones
                           ,Baños
                           ,Superficie
                           ,Barrio
                           ,CIUDAD
            ) ESTIM
            ON            PREC.Baños = ESTIM.Baños
                  AND PREC.Ciudad = ESTIM.Ciudad
```

```
                    AND PREC.Barrio = ESTIM.Barrio
                    AND PREC.Habitaciones = ESTIM.Habitaciones
                    AND PREC.Superficie = ESTIM.Superficie
```

Anexo II. Código Python

Codificación One-hot

```
import pandas as pd

df = pd.read_csv('./Data/Dataset.csv')
df = pd.get_dummies(df)

df_results.to_csv('./Data/Data_Onehot.csv', sep = "|")
```

Dataset de entrenamiento y validación

```
from sklearn.model_selection import train_test_split
import pandas as pd


df = pd.read_csv('./Data/Data_Onehot.csv')


#Se divide una partición de 80% dataset de entrenamiento y 20% dataset
de validación

df_train, df_test = train_test_split(df, test_size=0.2)


#Se exporta a csv para disponer de la partición en diferentes
implementaciones

df_train.to_csv('./Data/Train.csv', sep = "|")
df_test.to_csv('./Data/Test.csv', sep = "|")
```

Modelo Random Forest

```
import sklearn as sl
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import pandas as pd


df_train.read_csv('./Data/Train.csv')
df_test.read_csv('./Data/Test.csv')


#Variables predictoras y la variable objetivo (precio)

Predictor_train = df_train.drop('PRECIO', axis = 1)
Predictor_test = df_test.drop('PRECIO', axis = 1)

Target_train = np.array(df_train['PRECIO'])
Target_test = np.array(df_test['PRECIO'])
```

```
#Aplicación del modelo de Random Forest y ajuste a los datos de
entrenamiento

predictor = RandomForestRegressor(n_estimators = 10)
predictor = predictor.fit(Predictor_train, Target_train)


#Creación del vector de prediccions a través del predictor ajustado
prediction = predictor.predict(Predictor_test)


#Exportación de los resultados a csv
RESULTS_COLUMNS = ['PRECIO PREDICCION', 'PRECIO REAL']

i=0
l = []
while i < len(prediction):
    l.append([prediction[i], Target_test[i]])
    i = i + 1


df_results = pd.DataFrame(np.array(l).reshape(len(l),len(l[1])),
columns = RESULTS_COLUMNS)

df_results.to_csv('./Data/result.csv', sep = "|")
```