

Proyecto Final

Desarrollo y Despliegue de un Sistema de Aprendizaje Automático Escalable con AWS SageMaker

Aplicación de técnicas de Data Science y Machine Learning para la para la gestión del inventario de una agencia inmobiliaria mediante scoring de inmuebles en un entorno Cloud escalable

Ivonis Florindo López – Programa Profesional de Data Science e Inteligencia Artificial, UNIR



1. Resumen

Este proyecto tiene como objetivo la implementación de un flujo de trabajo completo de ciencia de datos, desde la comprensión del negocio y análisis exploratorio de datos (EDA), hasta la ingeniería de características, entrenamiento, evaluación, despliegue y monitorización de un modelo de aprendizaje automático, utilizando las capacidades de Amazon SageMaker y otros servicios *Cloud* de AWS. Se aplican técnicas de visualización, detección de sesgos y monitorización para garantizar un modelo robusto y reproducible.

Se define la variable objetivo, inicialmente como una clasificación multiclase que indica el tiempo estimado de venta.

2. Entendimiento del negocio

Objetivo de negocio

Una inmobiliaria quiere anticipar la velocidad de venta de sus propiedades para priorizar campañas, ajustar precios y facilitar el asesoramiento de sus clientes.

Una predicción fiable del rango de 'Time To Sell' impacta en la rotación de inventario, en la comisión variable de los agentes y en la mejora en la toma de decisiones de las campañas de marketing.

Este modelo será complementario a un futuro *lead scoring*.

KPI principal

- F1-Macro $\geq 0,70$ en el modelo de clasificación multiclase
- 48 h de refresco de inferencias para mantener recomendaciones de *pricing* al día.

Criterio de éxito

Si el modelo supera el 70 % de F1-Macro y se integra en el *dashboard* de *pricing*, se espera reducir un 15 % el stock > 30 días en los próximos 6 meses.

3. Arquitectura *Cloud* y preparación del entorno

Arquitectura *Cloud*

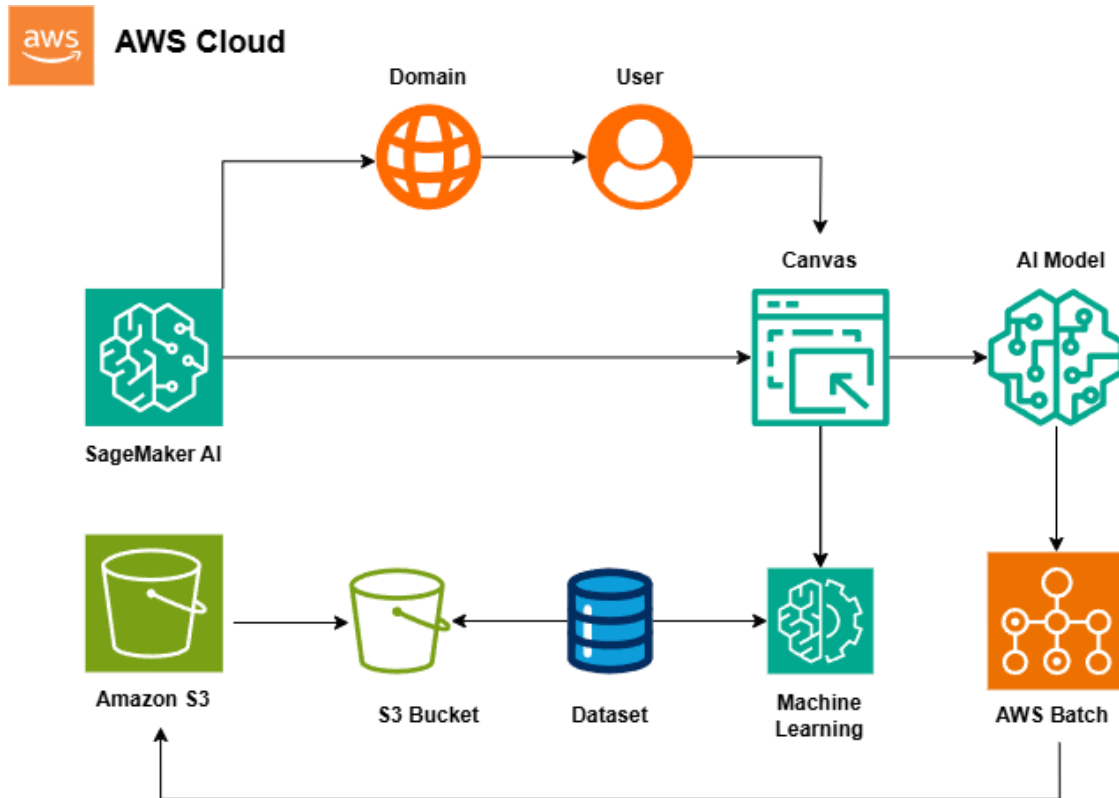


Figura 1. Arquitectura de AWS Cloud utilizada en el presente proyecto.

En la figura 1 se muestra la arquitectura *Cloud* que propuesta para cubrir el ciclo de vida del modelo. Se utilizarían servicios gestionados de AWS como SageMaker Canvas, Amazon S3 y procesamiento batch para generar predicciones a escala y monitorizarlas con Clarify.

Preparación del entorno

Amazon S3

En primer lugar se crea un *bucket* para el proyecto y se le nombra “ivo-unir”. Tal como se muestra en la figura 3. En él se almacenarán todos los archivos del flujo de trabajo como los datasets crudos y preprocesados, modelos y registros. Así se mantiene la gestión centralizada.

Crear bucket [Información](#)
Los buckets son contenedores de datos almacenados en S3.

Configuración general

Región de AWS
EE.UU. Este (Norte de Virginia) us-east-1

Tipo de bucket [Información](#)

☒ **Uso general**
Recomendado para la mayoría de los casos de uso y patrones de acceso. Los buckets de uso general son del tipo de bucket de S3 original. Permiten una combinación de clases de almacenamiento que almacenan objetos de forma redundante en múltiples zonas de disponibilidad.

☐ **Directorio**
Recomendado para casos de uso de baja latencia. Estos buckets utilizan únicamente la clase de almacenamiento S3 Express One Zone, que proporciona un procesamiento más rápido de los datos dentro de una única zona de disponibilidad.

Nombre del bucket [Información](#)
ivo-unir

Los nombres de los buckets deben tener entre 3 y 63 caracteres y ser únicos dentro del espacio de nombres global. Los nombres de los buckets también deben empezar y terminar con una letra o un número. Los caracteres válidos son a-z, 0-9, puntos (.) y guiones (-). [Más información](#)

Copiar la configuración del bucket existente: opcional
Solo se copia la configuración del bucket en los siguientes ajustes.

[Elegir el bucket](#)

Formato: s3://bucket/prefijo

Propiedad de objetos [Información](#)
Controle la propiedad de los objetos escritos en este bucket desde otras cuentas de AWS y el uso de listas de control de acceso (ACL). La propiedad de los objetos determina quién puede especificar el acceso a los objetos.

☒ **ACL deshabilitadas (recomendado)**
Todos los objetos de este bucket son propiedad de esta cuenta. El acceso a este bucket y sus objetos se especifica solo mediante políticas.


☐ **ACL habilitadas**
Los objetos de este bucket pueden ser propiedad de otras cuentas de AWS. El acceso a este bucket y sus objetos se puede especificar mediante ACL.

Propiedad del objeto
Aplicada al propietario del bucket

Figura 2. Creación de bucket de almacenamiento (Amazon S3)

SageMaker Canvas

Durante la creación del dominio de Canvas se seleccionó almacenamiento S3 personalizado usando el bucket s3://ivo-unir (Figura 3), y se configuran los roles de IAM y se crea un nuevo perfil de usuario con el que se abrirá Canvas

 **Canvas** [Más información sobre Canvas](#)

Genere predicciones precisas de machine learning, sin necesidad de código.

▼ **Configurar Canvas**

Configuración del almacenamiento de Canvas

Especificar la ubicación de los artefactos de Amazon S3
Especifique la ubicación de S3 en la que desea guardar los artefactos de Canvas para conjuntos de datos, modelos y predicciones. [Obtenga más información](#)

☐ **Sistema administrado**
Todos los artefactos generados se almacenarán en s3://sagemaker-us-east-1-316999731155

☒ **S3 personalizado**
Explorar Amazon S3 para seleccionar un bucket o ingresar un URI de S3

Q s3://ivo-unir X [Examinar S3](#)

Si desea garantizar los permisos necesarios para acceder a una ubicación de S3 personalizada, actualice los permisos del rol de ejecución predeterminado de SageMaker especificado en la configuración general. Esto incluye conceder acceso al bucket de S3 y, en el caso de los buckets de S3 de otra cuenta de AWS, garantizar el acceso tanto desde su cuenta como desde la otra cuenta. El bucket de S3 también debe tener la misma región que el dominio y los perfiles de usuario. Las políticas de IAM y los permisos de bucket incorrectos pueden provocar que los usuarios de la aplicación Canvas carezcan de los permisos necesarios para la creación de modelos. Para obtener información detallada, consulte la [documentación de Amazon S3](#)

Clave de cifrado - opcional
Cifre los datos. Elija una clave de KMS existente o introduzca el ARN de una clave.

Sin cifrado personalizado ▼

Figura 3. Desplegable de Canvas durante la configuración del dominio

Los archivos CSV se cargan en S3 (Figura 4) y se importan en Data Wrangler para la preparación, limpieza, transformación y exploración de los datos antes de entrenar el modelo.

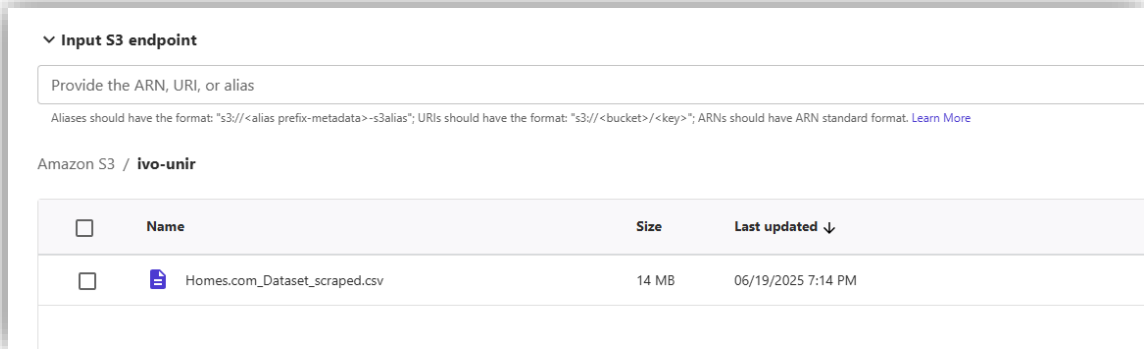


Figura 4. Vista de la ventana de importación de dataset desde Amazon S3

4. Preprocesamiento

Eliminación de variables irrelevantes

Se procede a eliminar las siguientes características:

- 'Street Info', 'Agency Name' y 'Agent Name' porque no aportan nada al problema de clasificación que afrontamos.
- 'Zip code' contiene 10.260 valores únicos, por lo que generará ruido.
- 'City' Contiene más de 30.000 valores únicos donde "Washington" supone el 2% del *dataset* y es el que más valores tiene.
- 'State' contiene un valor único.
- 'Status' contiene dos características juntas que no son necesarias: Fecha de Venta y Estado (*sold*) y, dado que ya existe "Days On Market", no se requiere la fecha de venta.

Normalización

Para garantizar compatibilidad SQL se normalizan los nombres de columnas eliminando espacios. De esta forma, las fórmulas utilizadas para la creación de nuevas características no generarán errores.

Revisión de tipos de datos

Se detectan varias variables en numéricas tipadas como *string*, por lo que se procede a cambiarlas a *float* o *long* (Figura 5).

Conversión de 'Sq Ft', 'Days on Market' (De ahora en adelante 'DOM') y 'Price' de *string* a *float/long*.

En el caso de 'Price' previamente se quita la coma usando 'Find and replace substring'. De esta forma el precio será 155000 en lugar de 155,000.

Se realiza el mismo procedimiento con "Sq Ft".

En el caso de 'DOM' cada celda presenta un patrón fijo: [Días]+[espacio]+[Days On Market] o [1]+[espacio]+[Day On Market]. Se aplica una doble transformación con "Find and Replace" reemplazando "Day/Days On Market/s". Seguidamente se pasa a tipo "long".

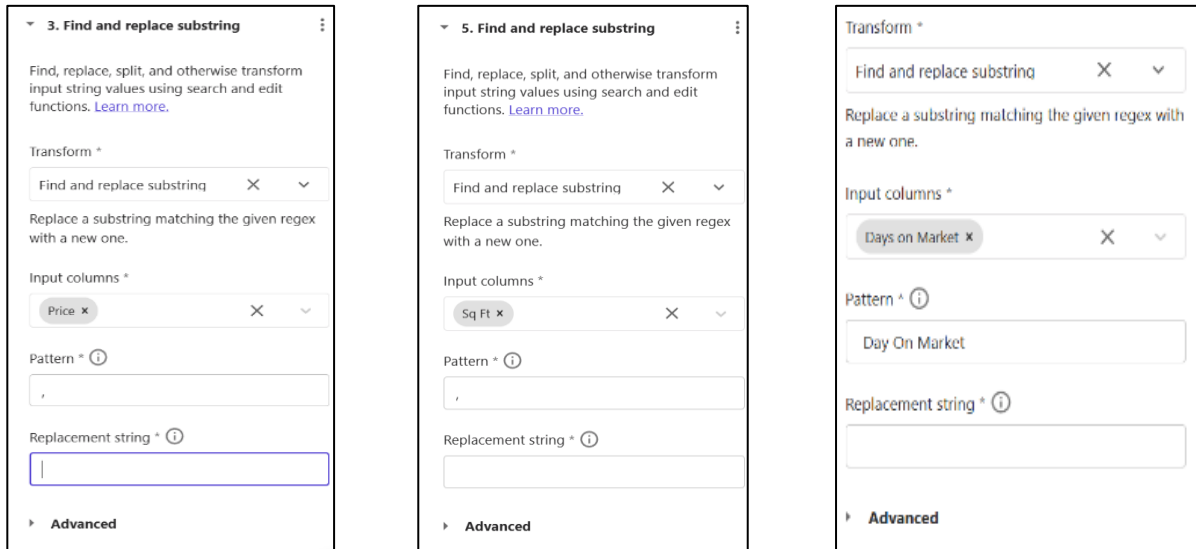


Figura 5 Figura 5. Vista de la Función “Find and replace substring” en Canvas

Revisión de datos faltantes

Se eliminan filas con valores faltantes en Price, Sq Ft y Price per Sq Ft para garantizar consistencia. Eran filas con nulos y celdas vacías en varias columnas, por lo que se decidió no imputar.

Se imputan Baths, Beds, Days on Market y Year Built con mediana aproximada para no perder registros que contienen información clave como el precio de venta.

Tras la limpieza y la imputación, se dispone de un dataset consistente y sin valores faltantes. Este consta de **24.093 registros** con variables numéricas y categóricas listas para su análisis y modelado.:

- ID: long.
- Price: float.
- Sq Ft: float.
- Price per Sq Ft: long.
- Days on Market: long.
- Beds: long.
- Baths: float.
- Year Built: long.
- Description: string (11 categorías).

Target leakage

La característica 'days_on_market' se elimina antes de entrenar porque es la propia variable que origina la etiqueta; mantenerla provocaría *target leakage* al estar disponible sólo tras la venta (Figura 6).

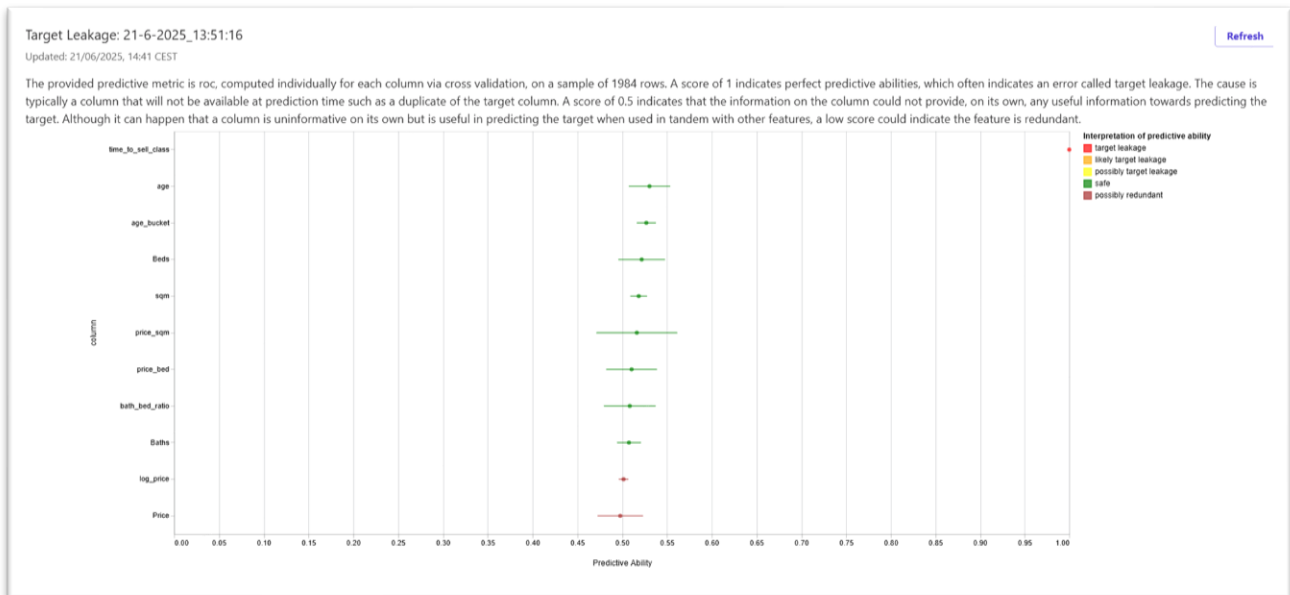


Figura 6. Gráfico ROC por variable para detección de leakage.

La figura 7 muestra el flujo de Data Wrangler de esta fase desde la revisión de tipos hasta que se exporta el dataset como usa_real_state_clean.csv

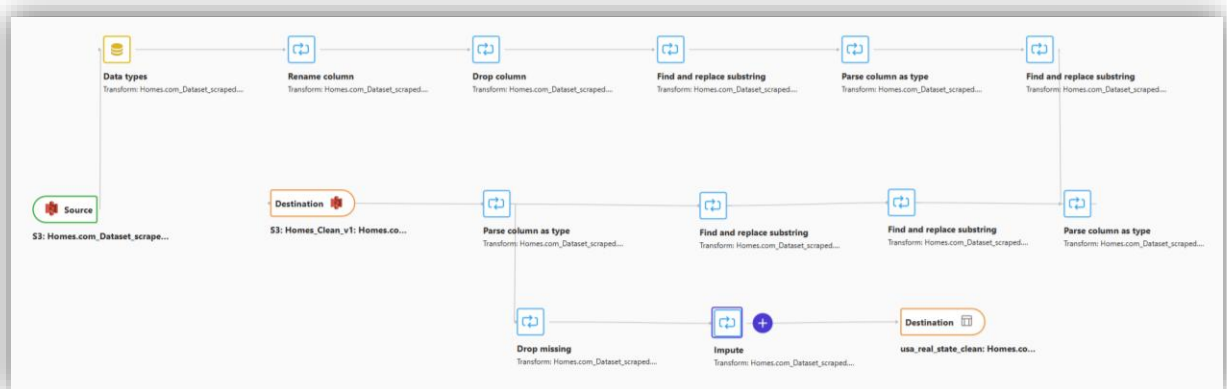


Figura 7. Flujo de transformaciones en Data Wrangler.

5. Feature Engineering

Age of property

Se transforma 'Year Built' en una nueva característica 'Age' que refleja directamente la antigüedad de la propiedad, una variable más informativa para el modelo de predicción.

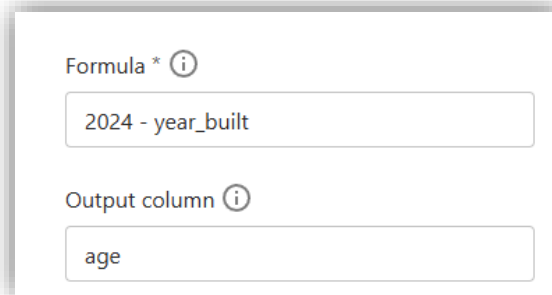


Figura 8. Cálculo de la columna age en Data Wrangler.

Medidas Internacionales

Se convierte la superficie de pies cuadrados (sqft) a metros cuadrados (sqm) y el precio unitario a precio por metro cuadrado (price_sqm) para facilitar la interpretación y coherencia con estándares internacionales. Posteriormente se eliminan las columnas originales para evitar redundancias. Se usaron las siguientes fórmulas:

$\text{sqm} = \text{Sqft} * 0,092903$

$\text{price_sqm} = \text{Price} / \text{sqm}$

Nueva columna 'time_to_sell_class'

Se genera la variable objetivo 'time_to_sell_class' mediante la categorización de 'days_on_market' en cinco rangos, permitiendo abordar el problema de predicción como clasificación multiclase.

Para valorar la selección de clases se observa la distribución de filas por tramos mediante la función filter rows.

- 115 filas se han vendido en 365 días o más.
- 23.978 propiedades, el 99.5%, se han vendido en menos de un año.
- 23.803 en menos de 9 meses.
- 23.200 en menos de 6 meses.
- 20.116 en menos de 3 meses.

- 17.806 en menos de 2 meses.
- 14.322 en menos de 1 mes.
- 11.190 en menos de 3 semanas.
- 9.457 en menos de 2 semanas.
- 6.642 en menos de una semana.

Se redefine la variable objetivo ‘time_to_sell_class’ en cinco categorías basadas en la distribución real de ‘days_on_market’, ajustando los rangos a la dinámica de ventas rápidas observada:

- Clase 0: ≤ 7 días (venta ultra-rápida).
- Clase 1: 8–14 días (venta rápida).
- Clase 2: 15–30 días (venta normal rápida).
- Clase 3: 31–90 días (venta moderada).
- Clase 4: > 90 días (venta lenta).

Se hace drop a las 115 filas que han tardado más de 365 días en venderse.

Se utiliza *Custom Formula* y se construye la variable time_to_sell_class mediante CASE WHEN que categoriza ‘days_on_market’ en las cinco clases anteriores, alineadas con la distribución real del conjunto de datos:

“CASE

WHEN days_on_market ≤ 7 THEN 0

WHEN days_on_market ≤ 14 THEN 1

WHEN days_on_market ≤ 30 THEN 2

WHEN days_on_market ≤ 90 THEN 3

ELSE 4

END”

Verificaciones finales

Mediante filtros en filas, se verifica que no haya valores negativos en las nuevas columnas tras las operaciones que se han realizado (Figura 9).

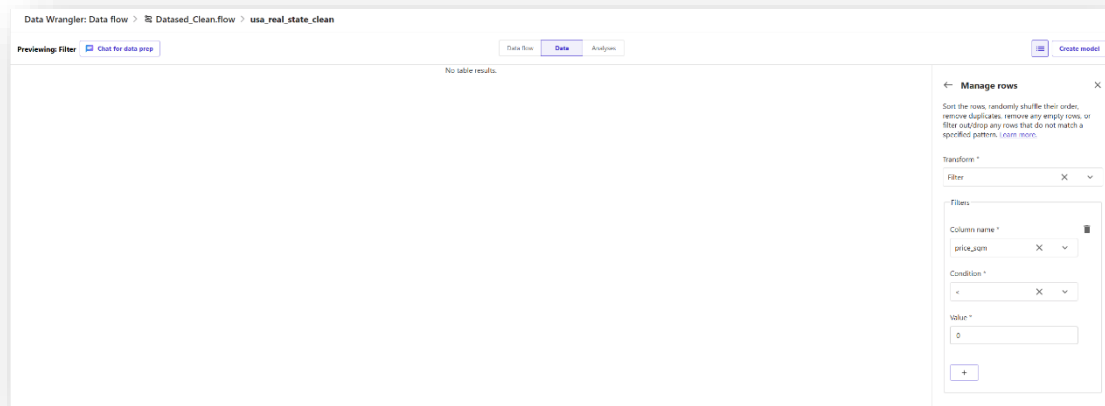


Figura 9. Filtro de valores cero en price_sqm.

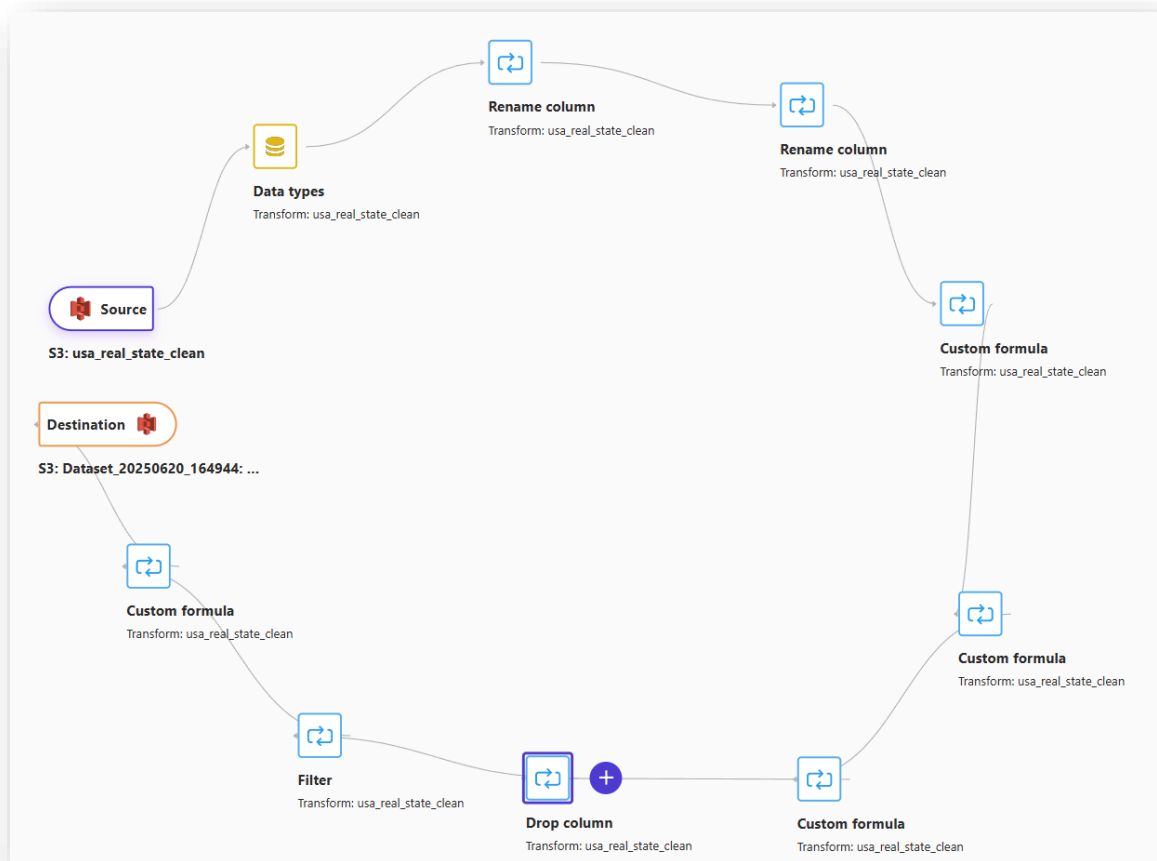


Figura 10. Flujo de pasos de limpieza en Data Wrangler.

6. Análisis exploratorio de los datos

Data quality and insights report v1

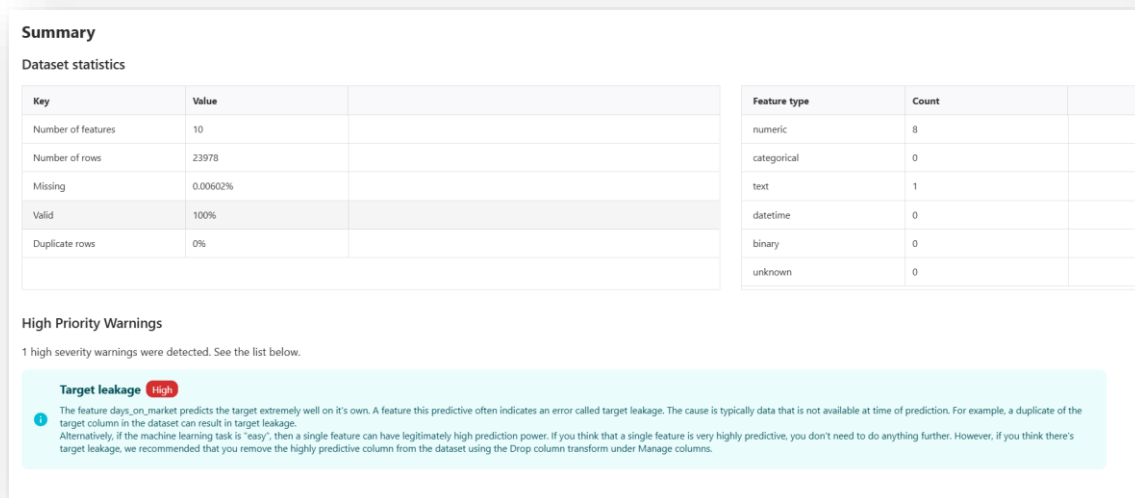


Figura 11. Resumen de estadísticas y alerta de fuga en Data Wrangler.

Se elimina 'days_on_market' para evitar target leakage, ya que su valor no está disponible en tiempo de predicción. Se hace lo mismo con 'Description', que podrá usarse para PNL. También se elimina 'id', que no aporta nada. A continuación, se obtiene un nuevo reporte.

Data quality and insights report v2

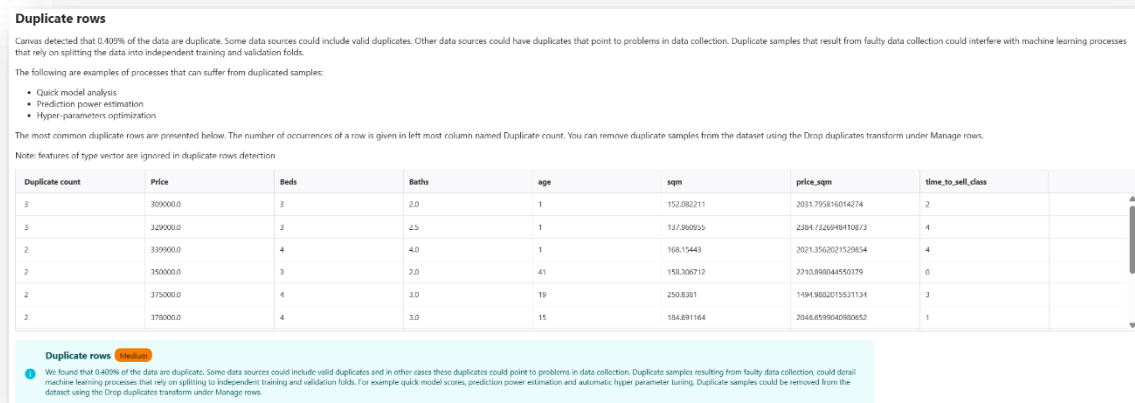


Figura 12. Detección de filas duplicadas en el dataset.

Según el reporte, se detectan un 0,409% de registros duplicados en el dataset. Estos duplicados podrían afectar la validación cruzada, el cálculo de métricas y el ajuste de hiperparámetros. Por ello, se eliminan usando la transformación *Drop Duplicates* para garantizar la independencia de los conjuntos de entrenamiento y validación.

Histograma del precio

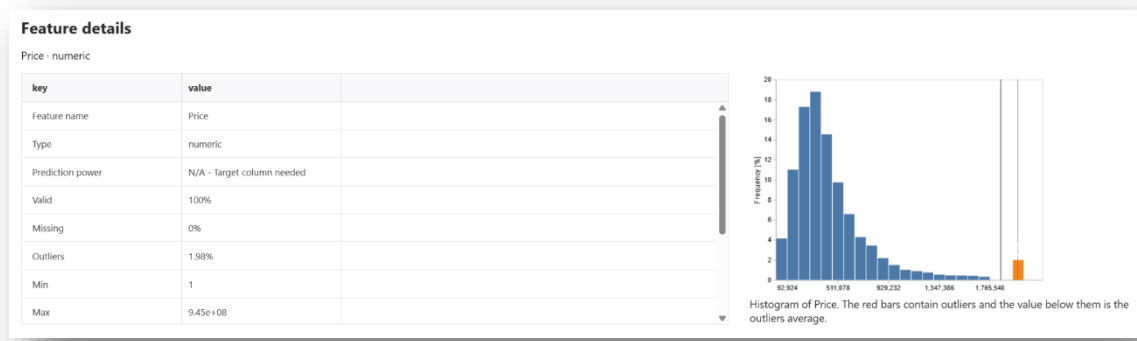


Figura 13. Histograma de precios y resumen de la característica.

Se detectan outliers en 'Price' (1.98% de los registros) con valores extremadamente altos. Se revisan y se decide eliminarlos para evitar distorsiones en el modelo. El corte se hace en USD 2.500.000, que es el 1% del dataset.

Histograma de 'beds'

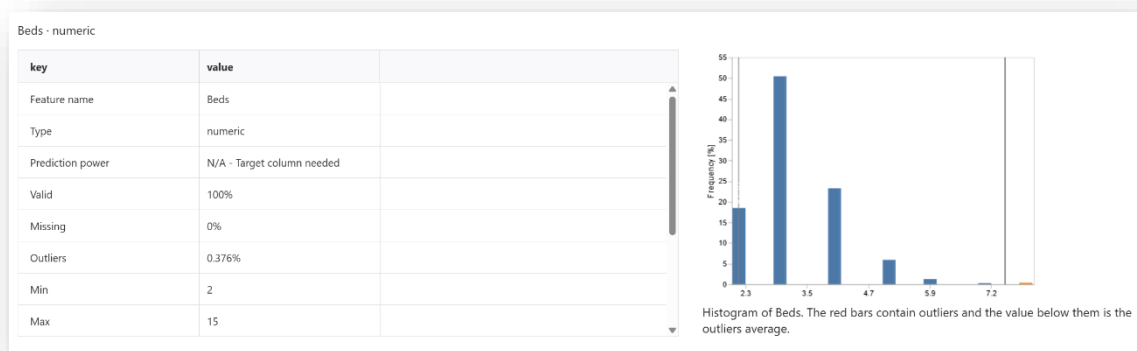


Figura 14. Histograma de habitaciones y resumen de la característica.

Se eliminan registros con más de 8 habitaciones para eliminar valores extremos poco representativos de la mayoría de propiedades, un total de 48 filas.

Histograma del precio por metro cuadrado

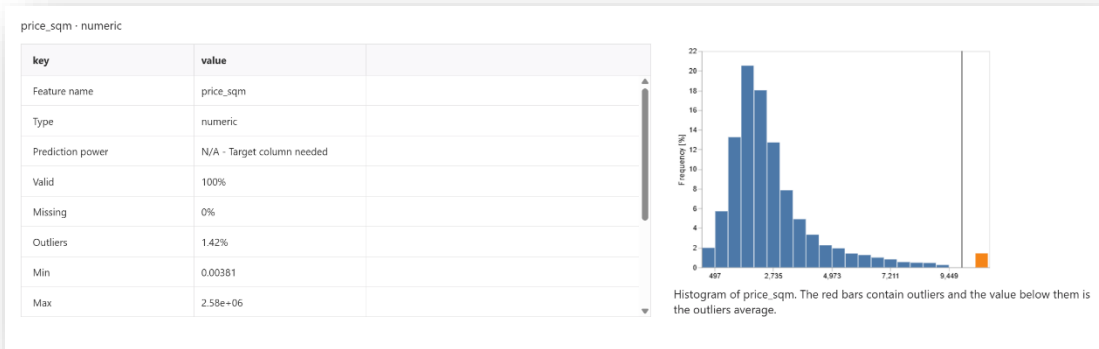


Figura 15. Histograma de 'price_sqm'

Se eliminan filas con valores de price_sqm superiores a 10.000 €/m², que representan aproximadamente el 1% de los registros, para evitar distorsiones en el entrenamiento del modelo. Llegados a este punto el total de registros que contiene el dataset es de 23.314.

Histograma del precio coloreado con beds

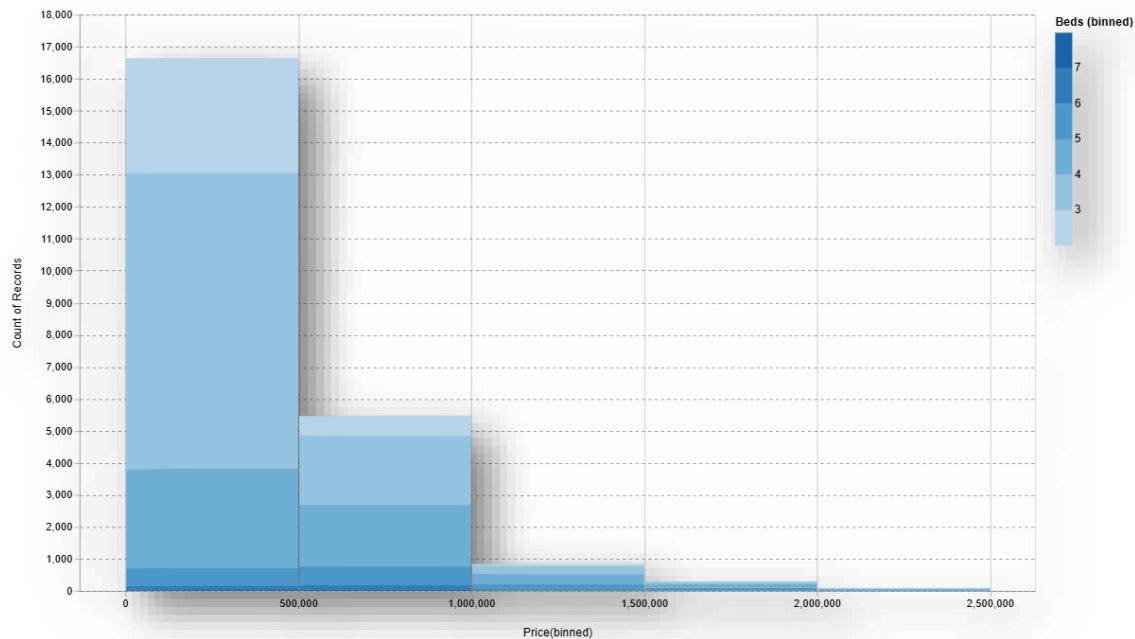


Figura 16. Histograma de precios coloreado por número de habitaciones.

Distribución fuertemente sesgada a la izquierda, concentrando la mayoría de los valores en rangos bajos de precio por metro cuadrado. Se observa una larga cola hacia la derecha que indica la presencia de inmuebles con precios significativamente altos, considerados como posibles valores atípicos.

Tabla resumen de características del dataset en su versión limpia

Table Summary: Untitled
Updated: 20/06/2025, 20:30 CEST

summary (string)	count (string)	mean (string)	stddev (string)	min (string)	max (string)
Price	23314	433919.9652140345	309817.28339912824	1.0	2495000.0
Beds	23314	3.226301792914129	0.8948978121191703	2	8
Baths	23314	2.3719867890537873	0.6901515213783875	0.5	6.0
age	23314	46.799004889765804	36.24579732051067	0	528
sqm	23314	190.98290949365193	904.7155116211873	17.837376	85267.209527
price_sqm	23314	2582.490736516847	1569.991916487431	0.003812934839242797	9987.79924153023
time_to_sell_class	23314	1.8412112893540362	1.4700744433559205	0	4

Figura 17. Tabla resumen de estadísticas descriptivas del dataset.

Heatmap de correlación entre variables

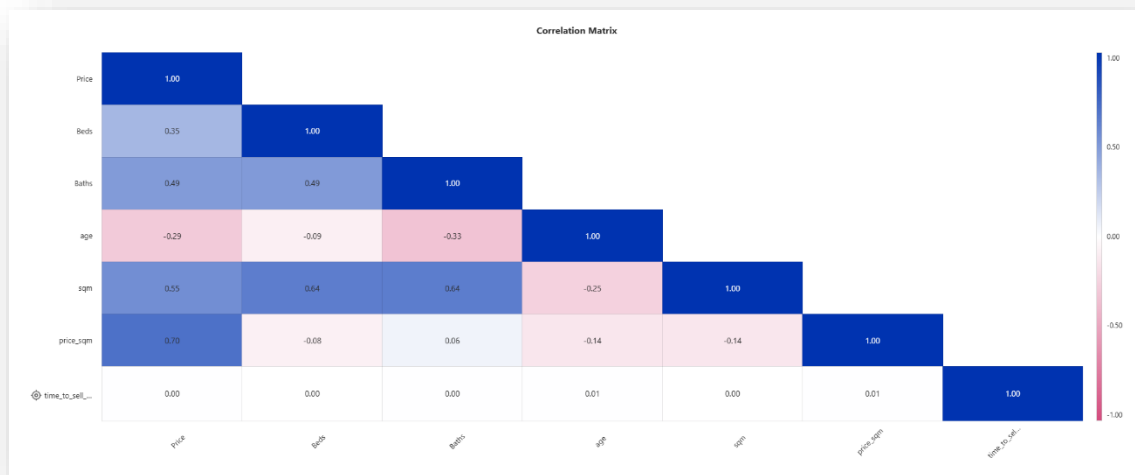


Figura 18. Heatmap de correlación entre variables.

La figura 18 muestra una alta correlación entre 'price' y 'price_sqm', coherente con la fórmula que vincula ambas variables. Se observa una correlación moderada entre 'sqm' y 'price', mientras que las demás variables presentan correlaciones bajas o nulas, sin indicar multicolinealidad crítica que afecte el rendimiento del modelo.

Scatterplot del precio vs time to sell coloreado por 'beds'

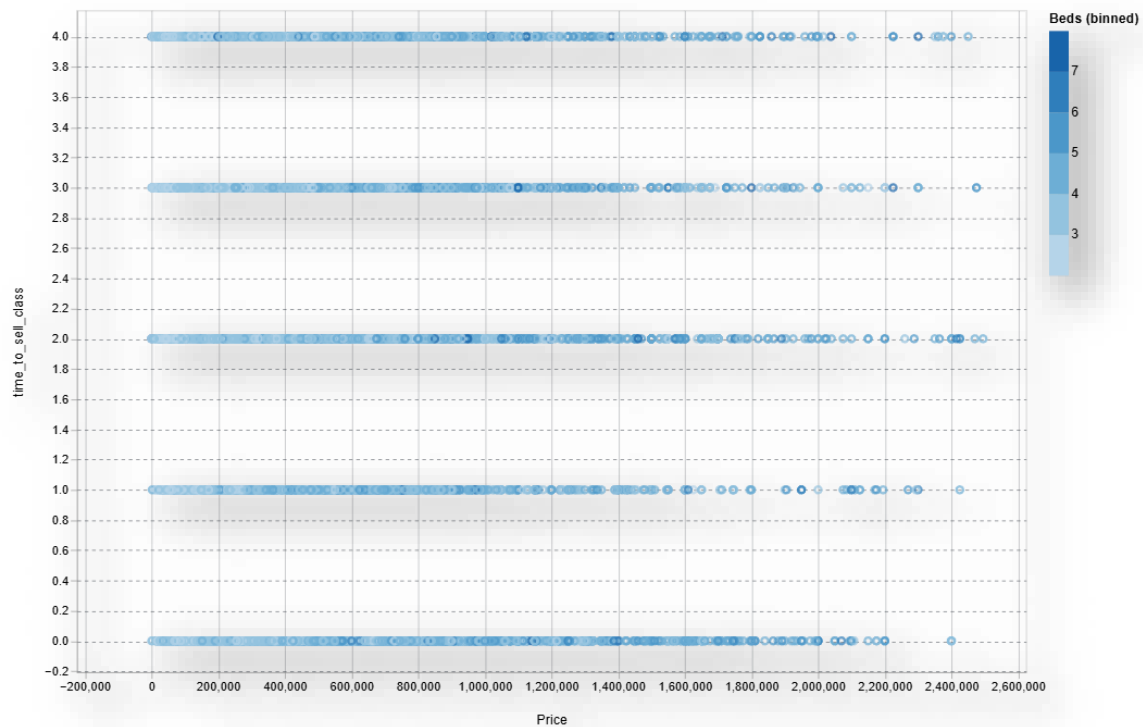


Figura 19. Scatterplot de precio vs tiempo de venta coloreado por habitaciones.

En los datos observados en la figura 19 se aprecia como el número de habitaciones influye en el precio pero no en el tiempo que se tarda en vender la propiedad. También se detecta que la variable 'time_to_sell_class' está contemplada como numérica, por lo que se cambia el tipo a *string*

Histograma de time_to_sell_class

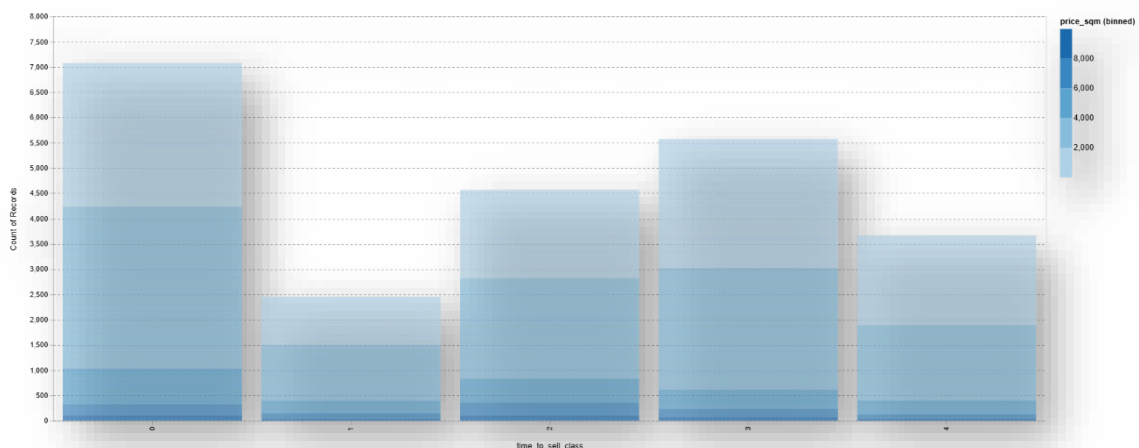


Figura 20. Histograma de tiempo de venta por clase coloreado por precio por metro cuadrado.

Tal como se observa en la figura 20, dado que la clase de propiedades que tardan más en venderse está poco representada, se considera necesario aplicar técnicas de balanceo para evitar que el modelo la pase por alto y mejore la precisión de la predicción para todos los casos.

- La mayoría de las propiedades se venden en menos de un mes, lo que confirma la alta rotación del mercado.
- Se observa leve relación entre superficie (sqm) y mayor tiempo en mercado: propiedades más grandes tienden a tardar más en venderse.
- No se detectan correlaciones excesivas entre variables numéricas, asegurando independencia de predictores claves.
- Se eliminaron registros con más de 6 baños para eliminar valores extremos poco representativos de la mayoría de propiedades, un total de 36 filas

Análisis de sesgo

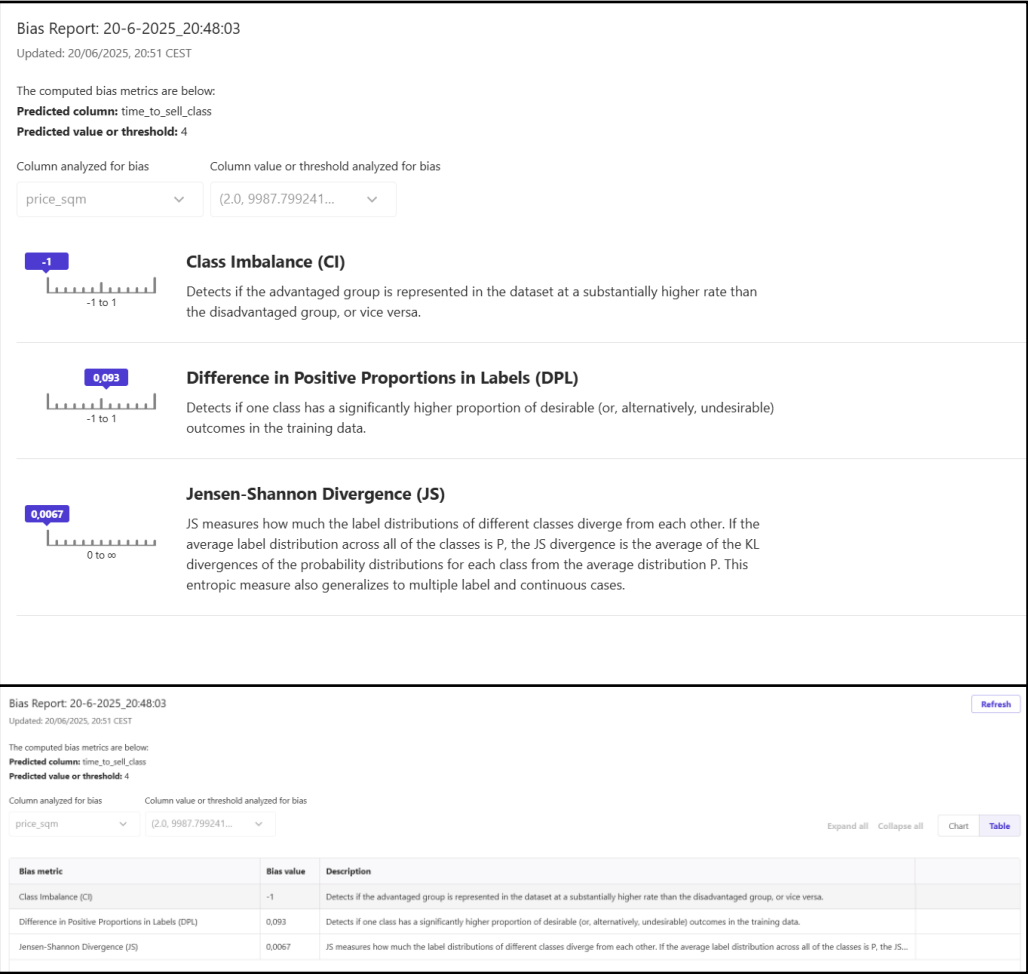


Figura 21. Tabla detallada de métricas de sesgo con valores y descripciones.

Se analiza si existe sesgo entre el precio por metro cuadrado y las propiedades que tardan más en venderse (clase 4). El informe (Figura 21) muestra un claro desequilibrio en la representación de este grupo, aunque la diferencia real en proporciones y la variación entre clases son bajas.

Resumen

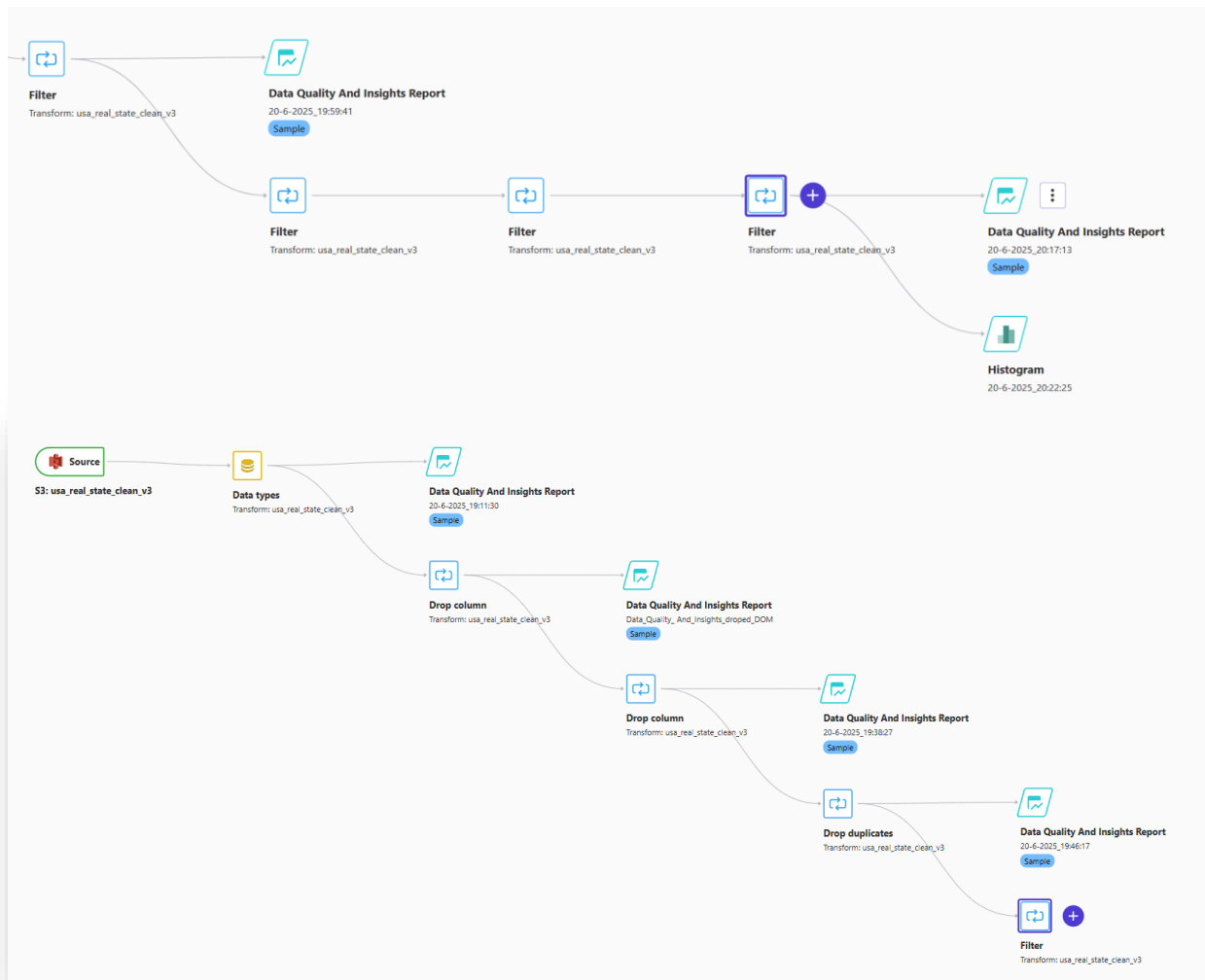


Figura 22. Flujo de Data Wrangler mostrando transformaciones y reportes de calidad.

Puede verse el flujo seguido en la etapa del Análisis Exploratorio del dataset en la Figura 22.

7. Entrenamiento y evaluación

Time-to-sell-5-class-predictor (3+ category model type)

V1 - Quick Build

Se inicia el entrenamiento con un Quick Build con máximo de 15 modelos y variable objetivo: 'time_to_sell_class' en un problema de clasificación multiclase. El tiempo máximo se ajusta a 30' (Figura 23).

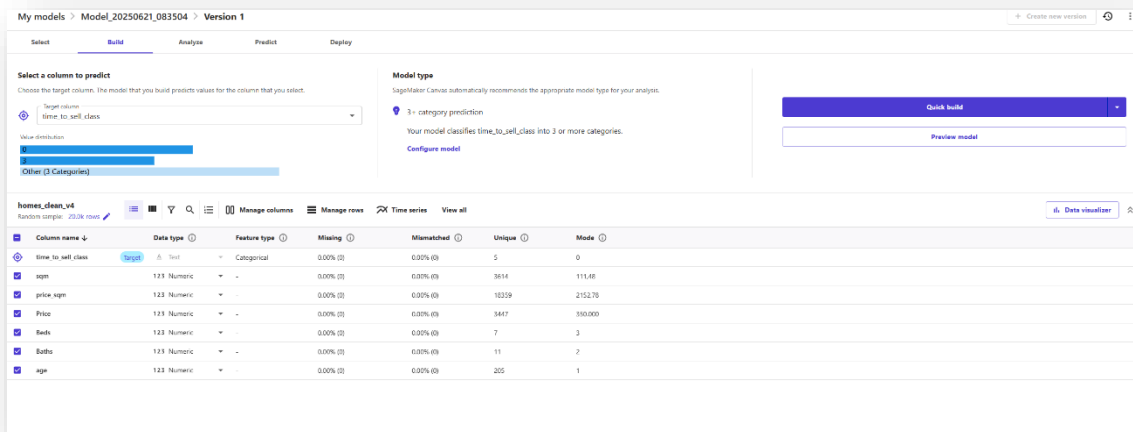


Figura 23. Configuración de Canvas para modelo de 3+ categorías con Quick Build.

Resultados del primer Quick Build: Balanced accuracy 0,24 establece la línea base frente al objetivo 0,70.

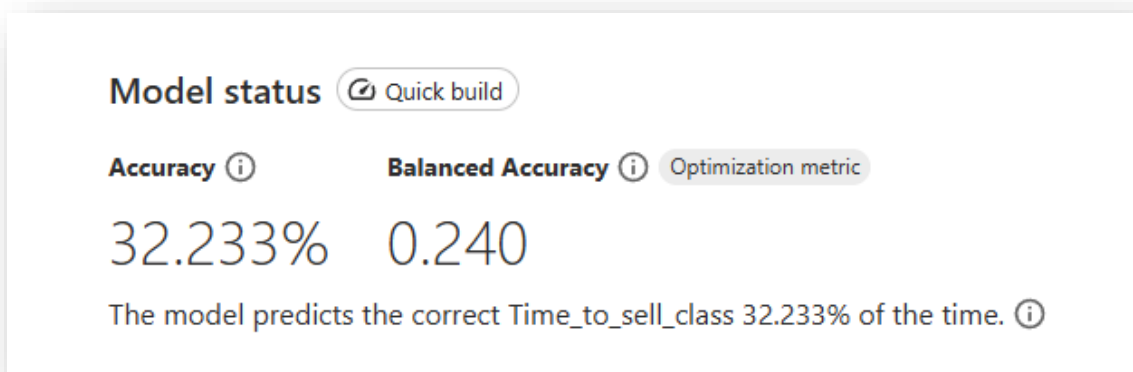


Figura 24 . Resultados del Quick Build con 32.233% de accuracy y 0.240 de balanced accuracy.

V2 - Standard Build

Se prueba el *Standard build* con la siguiente configuración de hiperparámetros (Figura 25). Se ajusta el máximo de candidatos a 15 y y 30' de tiempo de trabajo (Figura 26). Con esto se busca ver rápido si la optimización tiene efectos positivos rápidamente.

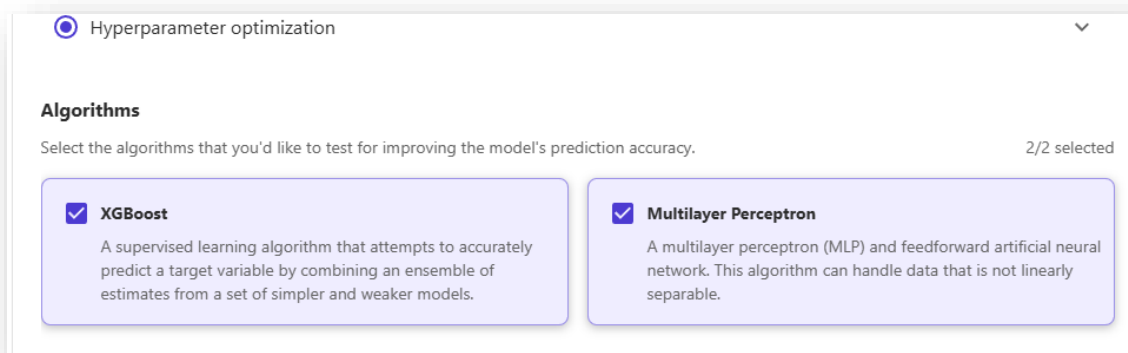


Figura 25. Selección de algoritmos para optimización de hiperparámetros: XGBoost y Multilayer Perceptron.

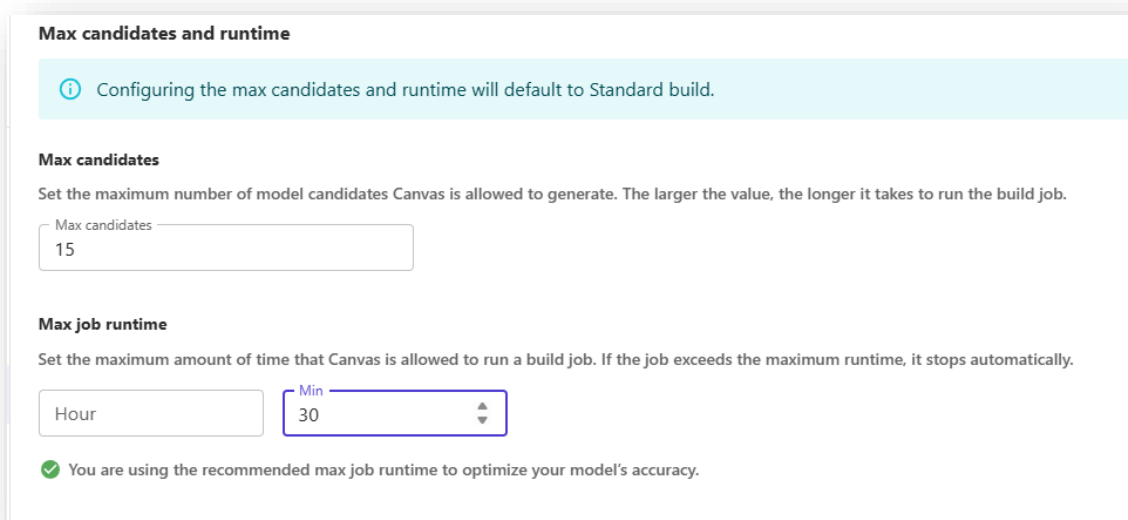


Figura 26. Max candidates and runtime de la V2 del modelo.

Resultados

Con este entrenamiento obtenemos 29.961% de accuracy y 0.212 F1 Macro (Figura 27), por lo que se confirma que sin nuevas variables y balanceo no hay mejora posible, se inicia reajuste en las características del dataset.

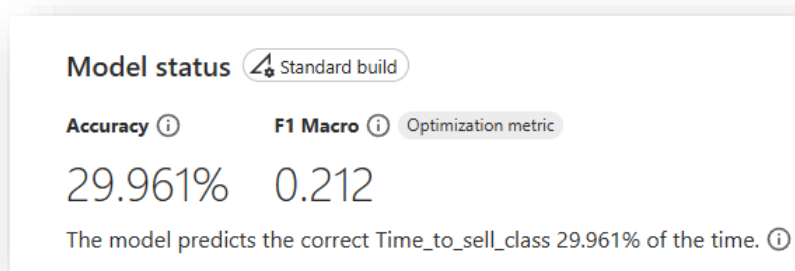


Figura 27. Resultados del Standard Build.

Time-to-sell-3-class-predictor (3+ category model type)

Nuevo enfoque de la variable objetivo

Tal como se observa en la Figura 28 se confirma que el desbalance es severo y que la falta de variables de localización impide al modelo hacer predicciones certeras.

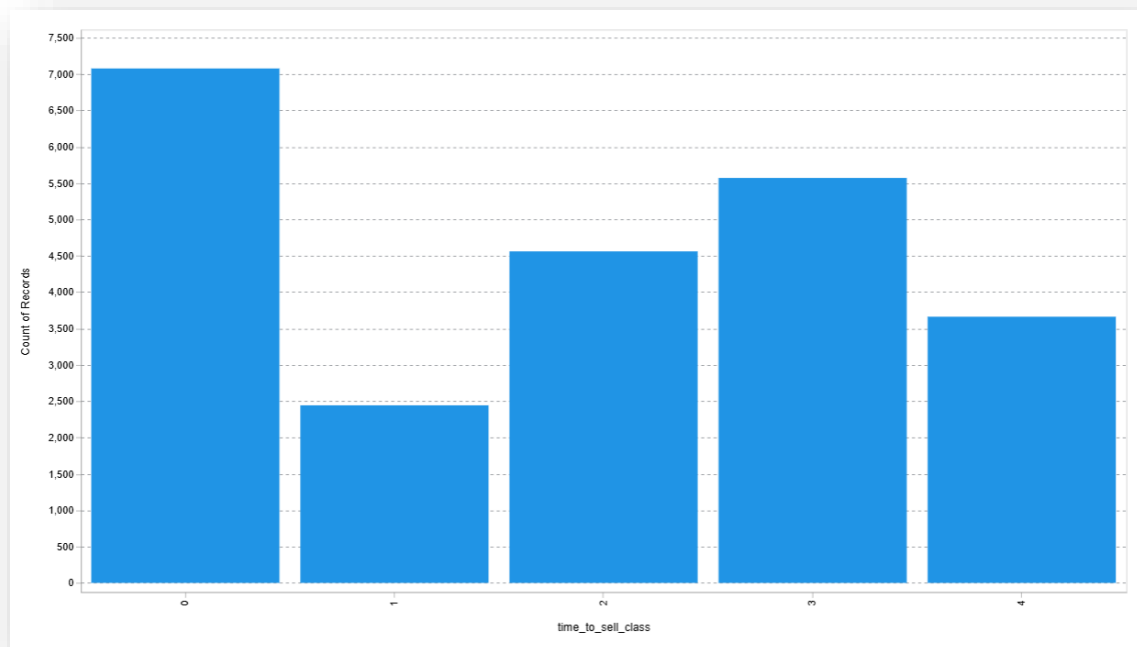


Figura 28. Distribución de clases en el dataset mostrando desbalance significativo.

- Clase **0** \approx 7 k registros
- Clase **1** \approx 2,4 k
- Clase **2** \approx 4,5 k
- Clase **3** \approx 5,6 k
- Clase **4** \approx 3,6 k

El 33 % del dataset está en la clase 0 y la clase 1 está muy poco representada.

Balanceo mediante 3 clases

- Se crea una nueva columna 'time_to_sell_3_class' que agrupa las clases 1-2 y 3-4, generando así tres clases: 0=quick, 1=moderate y 2=slow. (Figura 29)
- Será esta la nueva variable objetivo.

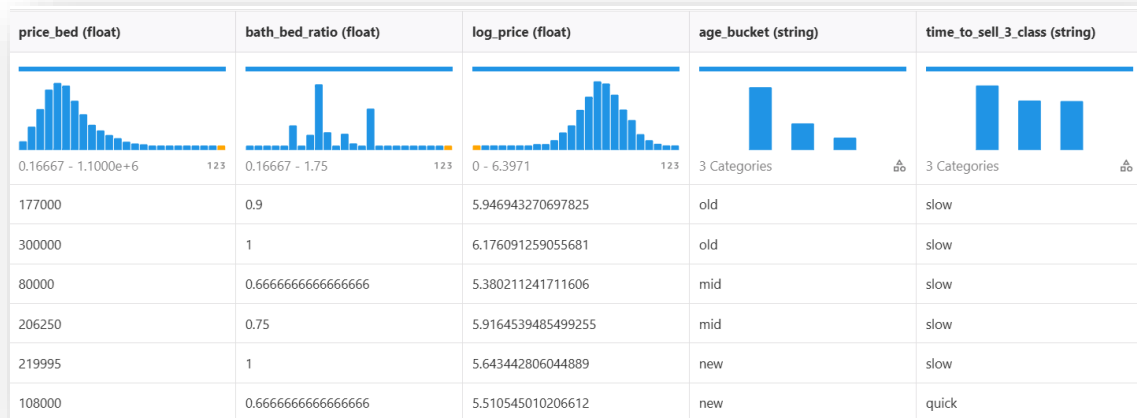


Figura 29. Vista previa del dataset procesado con variables transformadas y categorizadas.

Nuevas características

Tal como se observa en la Figura 29 se ha procedido a crear algunas nuevas características para mejorar los resultados:

- 'price_bed': precio por habitación.
- 'bath_per_bed_ratio': la ratio de baños por habitación.
- 'log_price': Precio transformado logarítmicamente.
- 'age_bucket': Categorización de la antigüedad de la propiedad en tres grupos: 'new' (0-5 años), 'mid' (6-30 años), y 'old' (más de 30 años).

Revisión de outliers

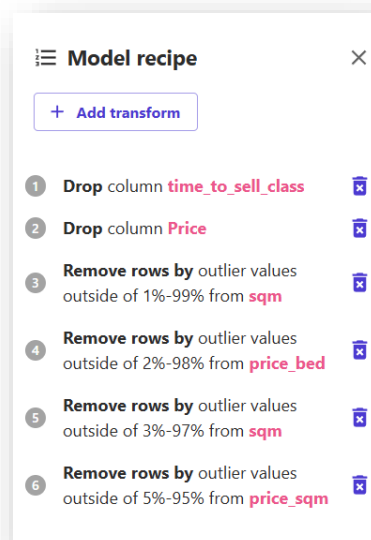


Figura 30. Model recipe.

Se realiza un nuevo preprocesado para tratar outliers usando un percentil de entre 1 y 5 según la característica.

V3 Standard Build, modelo multiclase

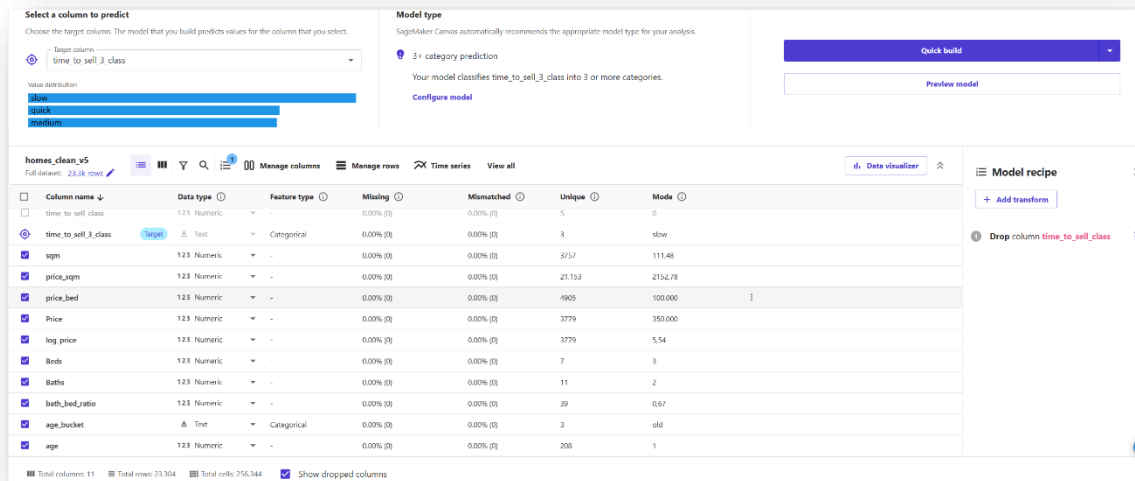


Figura 31. Configuración final del modelo con variable objetivo `time_to_sell_3_class` seleccionada.

Se deja fuera 'time_to_sell_class' ya que es la columna usada para generar la nueva variable objetivo. Se observa que 'Price' tiene relativamente pocos valores únicos en comparación con el total de registros, lo que sugiere redondeos o precios repetidos. Esto puede limitar la capacidad del modelo para aprender patrones más finos. Con el fin de aportar más detalle a la estimación se documenta esta limitación y se compensa usando variables derivadas como 'price_sqm' y 'log_price' (Figura 31).

Evaluación

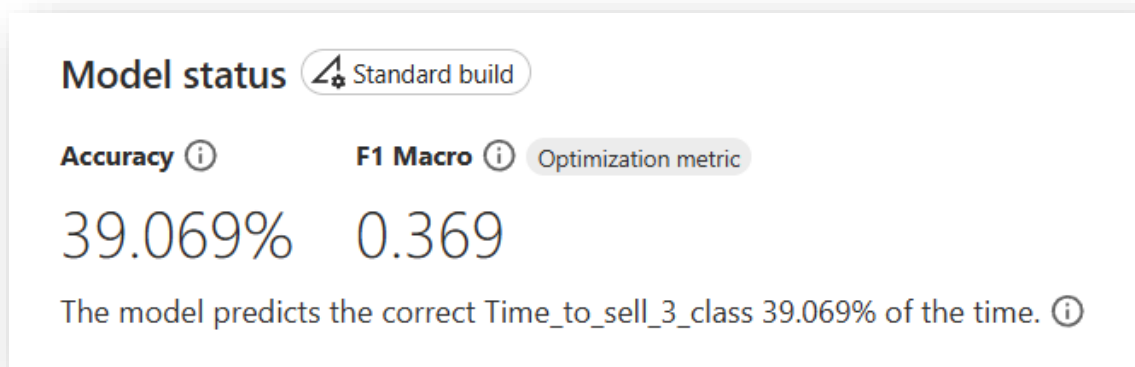


Figura 32. Modelo optimizado alcanzó 39.069% accuracy y 0.369 F1 Macro tras Standard Build.

Pese a la notable mejora del modelo, sigue sin acercarse al 0.7 objetivo (Figura 32). Dado este resultado la posibilidad de llevarlo a producción resulta poco adecuada. Se deberá realizar un nuevo estudio del Negocio así como así como una nueva analítica de los datos.

8. Registro y Despliegue

Registro del modelo

El modelo final se registra en el SageMaker Model Registry para gestionar versiones y facilitar su reutilización (Figura 33).

Add to Model Registry

Add model to the Model Registry so Studio users can catalogue, review and deploy the model in SageMaker Studio.



Selected version

VNaN  Ready Created Jun 21, 2025 1:44 PM

SageMaker Studio model group

Model group name

Use only letters, numbers, and dashes, up to 32 characters.

A model group helps you organize and track multiple versions of the same model. You can only specify one model group per model version.
 After you register a version from this model, all subsequent versions are registered to the same model group. 
[Learn more](#)

Cancel

Add

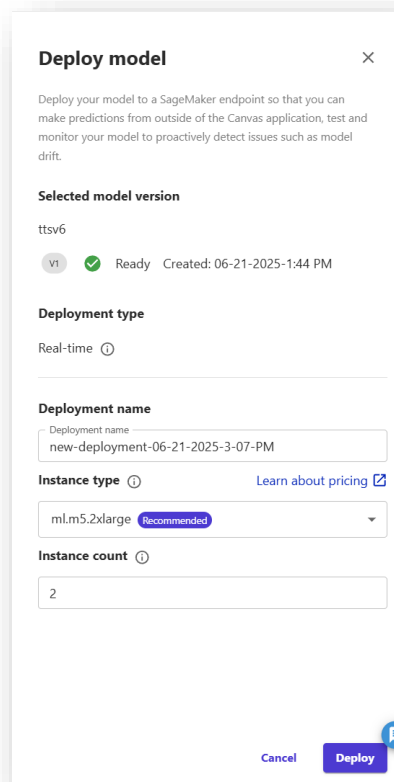
Figura 33. Registro del modelo

Despliegue

Se plantea desplegar el modelo mediante un endpoint en SageMaker para realizar inferencias en tiempo real o en modo batch. Por ejemplo, se podría automatizar la consulta de nuevos anuncios usando la API de Idealista y aplicar el modelo para asignar un scoring de venta y tiempo estimado de comercialización, integrándolo en un flujo de gestión de leads.

Para asegurar la calidad y estabilidad del modelo en producción, se utilizarán logs y métricas del servicio AWS CloudWatch y el panel de monitorización de SageMaker Studio. Se revisará el consumo de recursos, tiempos de respuesta y la precisión del modelo de forma periódica. Además, se guardarán métricas de inferencias en Amazon S3 para análisis históricos y posibles reentrenamientos.

No obstante en la Figura 34 se muestra un despliegue de ejemplo.



Deploy model ✕

Deploy your model to a SageMaker endpoint so that you can make predictions from outside of the Canvas application, test and monitor your model to proactively detect issues such as model drift.

Selected model version

ttsv6

v1 ✓ Ready Created: 06-21-2025-1:44 PM

Deployment type

Real-time ⓘ

Deployment name

Deployment name
new-deployment-06-21-2025-3-07-PM

Instance type ⓘ [Learn about pricing](#) ↗

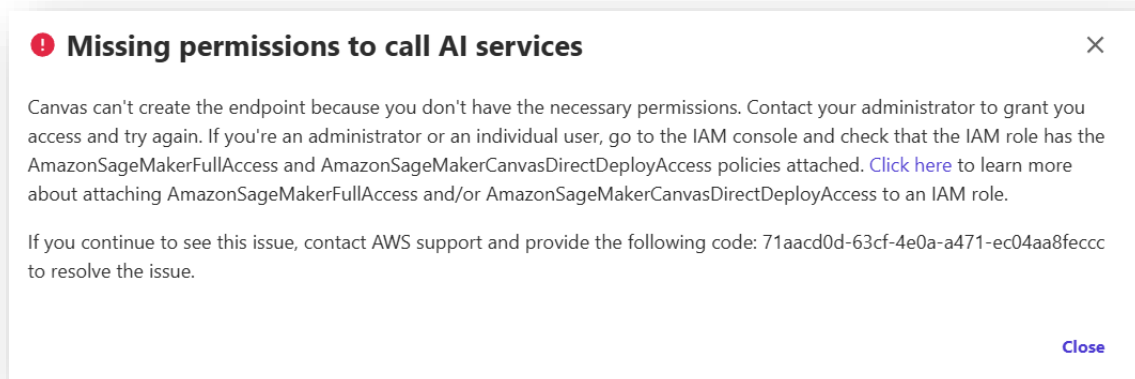
ml.m5.2xlarge Recommended

Instance count ⓘ

2

Cancel Deploy

Figura 34. Configuración de despliegue .



! Missing permissions to call AI services ✕

Canvas can't create the endpoint because you don't have the necessary permissions. Contact your administrator to grant you access and try again. If you're an administrator or an individual user, go to the IAM console and check that the IAM role has the AmazonSageMakerFullAccess and AmazonSageMakerCanvasDirectDeployAccess policies attached. [Click here](#) to learn more about attaching AmazonSageMakerFullAccess and/or AmazonSageMakerCanvasDirectDeployAccess to an IAM role.

If you continue to see this issue, contact AWS support and provide the following code: 71aacd0d-63cf-4e0a-a471-ec04aa8feccc to resolve the issue.

Close

Figura 35. Error de permisos.

El despliegue del modelo falló debido a permisos insuficientes en el entorno de laboratorio de AWS. Los labs educativos tienen políticas de IAM restrictivas que limitan el acceso a servicios como SageMaker endpoints para controlar costos y prevenir uso no autorizado.

9. Conclusiones, trabajo futuro y anexos.

El modelo desarrollado muestra resultados interesantes para estimar el tiempo de venta de propiedades usando variables clave como el precio normalizado, la superficie o la antigüedad. No obstante, la ausencia de variables de localización limita su precisión, ya que este tipo de mercado está muy influido por la cotización de la zona, una cuestión difícil de definir.

Un mismo inmueble con exactamente las mismas características, puede tener precios radicalmente distintos en dos ubicaciones diferentes, y en el dataset que se ha usado para realizar el entrenamiento contenía más de 30.000 valores diferentes en la columna 'city'. Con tan alta cardinalidad en los datos el modelo generaría ruido, por ello se decidió eliminarla.

Teniendo en cuenta los hallazgos obtenidos y en perspectiva de futuro, se plantea la posibilidad de incorporar más información sobre la cotización según la ubicación. Por ejemplo, se podría entrenar un modelo auxiliar que calcule la cotización promedio por ciudad y clasifique cada ubicación en rangos de precio (barata, media, cara, lujo). De esta forma se vería reflejada la influencia geográfica en el precio sin añadir ruido ni alta cardinalidad, mejorando así la precisión del modelo principal.

Con el propósito de controlar los gastos en futuros desarrollos, se ve conveniente realizar un mayor trabajo de conocimiento del negocio en el problema que se vaya a tratar. La problemática surgida con la mencionada columna 'city' y la posterior decisión de seguir adelante eliminándola ha generado un gasto evitable desde las primeras fases del proyecto.

En caso del desbalance de las clases objetivo, visto en el EDA, daba igualmente motivos suficientes como para no generar gastos en el entrenamiento del modelo sin haber creado nuevas clases más representativas.

Otra línea de ataque para los desbalances habría sido usar SMOTE o desponderación, la cual se considera también una línea de trabajo futuro junto con utilizar modelos de NLP que sean entrenados con las descripciones.