UNIVERSIDAD
DE ANTIOQUIA

Predicción del precio de vivienda en Antioquia

Camilo Gutiérrez Ramírez

Daniel Parra Holguín

Monografía presentada para optar al título de

Especialista en Analítica y Ciencia de Datos

Asesor

Sebastián Rodríguez Colina, MSc

Universidad de Antioquia

Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Colombia

2022

Cita	Gutiérrez Ramírez y Parra Holguin [1]
Referencia Estilo IEEE (2020)	[1] C. Gutiérrez Ramírez y D. Parra Holguín, “Predicción del Precio de Vivienda en Antioquia”, Trabajo de grado especialización, Especialización en Analítica y Ciencia de Datos, Universidad de Antioquia, Medellín, Antioquia, Colombia, 2022.

Especialización en Analítica y Ciencia de Datos, Cohorte III.

TABLA DE CONTENIDOS

Índice de figuras

Índice de tablas

1. RESUMEN EJECUTIVO

En el contexto del mercado inmobiliario el precio de venta tiende a variar significativamente en función de factores como el año de construcción, el estado en que se encuentra, la ubicación y el valor del metro cuadrado, es por esto que existe una necesidad de las personas que deseen comprar o vender una vivienda en estimar el valor del inmueble en cuestión.

El proyecto se enfocó en dar una estimación del precio de una vivienda usada en el departamento de Antioquia a partir de variables como área construida, número de habitaciones y baños, localización, entre otros. Se presentan diferentes algoritmos de inteligencia artificial para predecir los precios de la vivienda con una buena precisión que permitirán a distintos actores tener una aproximación para sus intereses económicos que se dan en el mercado de finca raíz. Una vez se tenga el modelo se disponibiliza un microservicio mediante un API.

El modelo se actualiza a través de procesos orquestados para obtener datos desde distintas fuentes, esto con el fin de poder entrenar continuamente el modelo y darle la cualidad al desarrollo de adaptarse a nuevos cambios en el mercado, derivados de las dinámicas sociales que impactan el precio de la vivienda, por ejemplo, nuevos focos de desarrollo urbanos, potenciales para inversión.

Palabras clave: Propiedad raíz, Web Scraping, Machine learning, MLOps,

El código fuente usado en este proyecto se puede encontrar en el siguiente enlace:

Repositorio: https://github.com/dparraho/monografia

2. DESCRIPCIÓN DEL PROBLEMA

El mercado inmobiliario tiene una participación importante en la economía del país, las cifras del 2021 del sector lo demuestran: 128.200 operaciones de financiación fueron desembolsadas, lo que se traduce en $15,8 billones de pesos. Durante este año el crecimiento fue superior al 80% respecto al año anterior. En cuanto a financiación el crecimiento en valor real fue del 116,7%. Esta tendencia positiva puede explicarse por los bajos intereses para préstamos hipotecarios impulsados por el Gobierno durante la emergencia sanitaria.

Al momento de realizar una transacción comercial de bienes raíces como adquisición o venta, conocer el valor del inmueble representa un desafío al depender de múltiples variables cuantitativas y cualitativas. El valor de una vivienda no depende solo de sus condiciones físicas como área construida y número de habitaciones, sino también del concepto personal del individuo que suele estar relacionado con la percepción de seguridad, paisajismo y ubicación, por lo que estimar el precio requiere de un algoritmo o modelo predictivo complejo. Dicha complejidad aumenta aún más si, además, los precios de las viviendas varían drásticamente con el tiempo; se debe considerar precios comerciales actualizados, comparar con los alrededores y capturar la variación de los precios en el tiempo. Sin embargo, la solución a estos problemas expandirá el alcance a entidades como bancos, grupos de inversión y personas naturales que realizan operaciones financieras, e inclusive puede impactar a desarrolladores inmobiliarios que operan con valores estimados y rentabilidades futuras.

Actualmente la estimación de precios se realiza manualmente con peritaje de expertos que miden características internas y externas de una casa, apartamento, finca, etc., y es un método legal para estimar el avalúo comercial de un inmueble. Sin embargo, esta técnica es costosa y puede fallar ya que el peritaje requiere la asistencia y medición manual del experto en todos los inmuebles a ser avaluados, asimismo, el uso que se da de esta estimación es sólo para inmuebles existentes y falla en incluir factores en el cálculo como cercanía a parques, centros comerciales, clínicas, centros educativos, dinámicas comerciales, entre otros.

Existen plataformas actualmente como Finca Raíz y Metro Cuadrado que facilitan la comunicación entre compradores y vendedores al disponer públicamente los inmuebles actuales a la venta, su precio y características internas típicas que un comprador consideraría en una búsqueda, sin embargo, estas páginas no permiten estimar el precio al que debe ser ofrecido la vivienda.

2.2 APROXIMACIÓN DESDE LA ANALÍTICA DE DATOS

Por la naturaleza del problema, se abarca con un modelo de regresión desde el paradigma de aprendizaje supervisado pues se tiene una variable a predecir continua. El aprendizaje supervisado busca encontrar una función que, dado unos datos de entrada, estime un valor cercano al output esperado en la data inicial. Este modelo de regresión podrá servir como respaldo técnico al momento de evaluar opciones de compra de vivienda; por ejemplo, la persona puede estimar el costo de cambiar de sector de la ciudad, de agregar una alcoba con un baño, etc. Las personas que deseen adquirir o vender una vivienda podrán realizar el avalúo del inmueble con nuestro modelo de aprendizaje de máquina. En el contexto del mercado

inmobiliario el precio de venta tiende a variar significativamente en función de factores como el año de construcción, el estado en que se encuentra, la ubicación y el valor del metro cuadrado, es por esto que se debe dar información precisa.

2.3 ORIGEN DE LOS DATOS

Se recopilaron datos de los portales MetroCuadrado[1], Finca Raíz[2] y Properati [1], este último cuenta con un dataset público y gratuito de inmuebles que se han publicado a la venta o en arriendo en toda Latinoamérica. Para el caso de Colombia cuenta con alrededor de un millón de propiedades, y además, Properati pone a disposición los últimos 15 años de precios de vivienda a través del servicio de almacenamiento BigQuery de Google. Es importante resaltar que será complementado regularmente con nuevas publicaciones de ventas, obtenidas a partir de extracción con técnicas de Web Scraping. Datos adicionales se obtienen de otras fuentes como OpenstreetMap para sitios de interés y accesibilidad vial, Google Earth Engine para índices de vegetación del terreno, Earth Data de la Nasa para pendientes del terreno y Datos Abiertos Colombia para divisiones geopolíticas de barrios y comunas.

2.4 MÉTRICAS DE DESEMPEÑO

El valor del precio de un inmueble es una medida cuantitativa continua. Sin embargo, es importante tener en cuenta que el precio tiene un rango de variación alto y no sigue una distribución normal, sino que por el contrario tiene una distribución con cola pesada a la derecha, es decir, en el conjunto de datos hay precios únicos muy grandes, aspecto que debe considerarse al escoger la métrica. Con todo lo anterior, se evaluará el desempeño del modelo con la Mediana del Error Porcentual Absoluto (MAPE), ya que es de fácil interpretación para el negocio, al mostrar unidades porcentuales del error que se comete en la estimación del precio de la mayoría de los inmuebles.

El éxito del producto depende de que los usuarios vean valor al uso de las consultas para sus transacciones de bienes raíces, por esta razón, un modelo que porcentualmente esté en un rango cercano al valor real es el objetivo del producto. Para determinar esos valores mínimos buscados del MAPE, se debe tener en cuenta la incertidumbre superior que significa estimar el precio de un inmueble rural en comparación con un inmueble urbano. Típicamente un inmueble rural se compone de grandes extensiones de tierra, y tiene pocos vecinos con los que se pueda realizar una comparación de precios comerciales, lo cual si es posible con inmuebles urbanos. Consecuentemente, el negocio decide que el desempeño del producto se divide para ambas zonas con valores mínimos de 9% para urbano y 15% para rural. En ventas típicas de Antioquia, una vivienda con precio de 400 millones COP tendría un error de 36 millones COP si se ubica en zona urbana y 60 millones COP si está en zona rural.

[1] https://www.metrocuadrado.com/

[2] https://www.fincaraiz.com.co/

3. DATOS

3.1 DATOS ORIGINALES

El dataset de Properati cuenta con alrededor de un millón de propiedades con información relevante como: tipo de la propiedad, latitud, longitud, precio original del aviso, área en m², precio por m², número de pisos, dormitorios, baños, descripción, barrio, fecha de publicación, entre otros. Se eliminaron las filas que tuvieran registros faltantes en la latitud y longitud ya que son factores que no se puede imputar acertadamente. Se seleccionaron únicamente los datos de viviendas en Antioquia, que corresponden al alcance de nuestro problema de negocio, como se describió en la sección 2. Se escogió esta área para tener en cuenta el precio en zonas rurales y urbanas, sin embargo, es importante aclarar que un porcentaje considerable de la población antioqueña se concentra en el Valle de Aburrá y el oriente de este. En la siguiente tabla se describen los campos disponibles y su respectiva descripción:

Tabla 1. Descripción de los campos disponibles.

Columna	Descripción
precio	El precio de la vivienda
lon	Longitud (Localización)
lat	Latitud (Localización)
bedrooms	Número de habitaciones
bathrooms	Número de baños
surface_total	Área en metros cuadrados de la propiedad
property_type	Tipo de propiedad: Apartamento o casa

3.2 DESCRIPTIVA

El dataset disponible de Properati pertenece a los años 2020 y 2021. Se tienen 3205 registros disponibles en Antioquia, cuyo rango de tiempo coincide el 60 % corresponden a datos del 2020 y el restante al 2021. Lo anterior debe ser tenido en cuenta ya que los modelos son optimizados basado en los datos utilizados durante la fase de entrenamiento y estos tienen este posible sesgo.

El precio de vivienda oscila entre 25 millones de pesos y 45 mil millones de pesos. El promedio son 820 millones de pesos y la mediana 420 millones. Como la media es mayor que la mediana se dice que la distribución de precios está sesgada a la derecha.

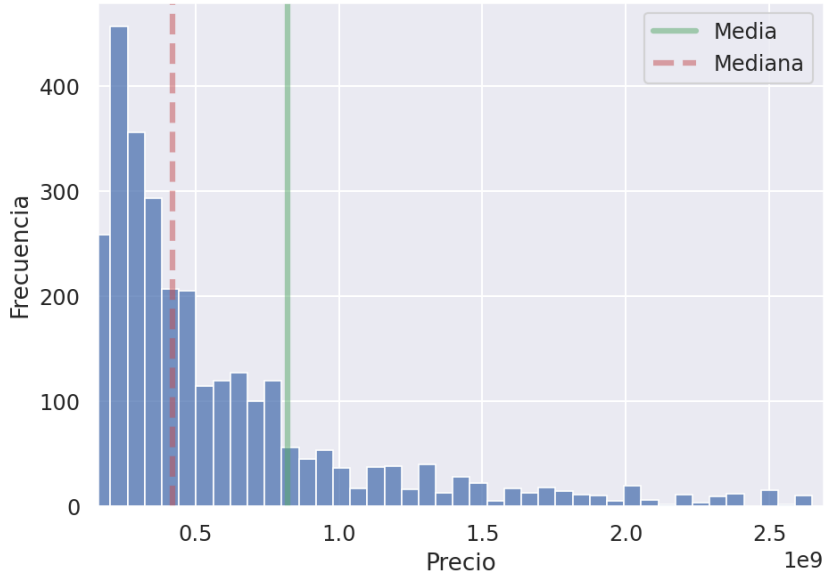

Figura 1. Distribución de precios en el dataset.

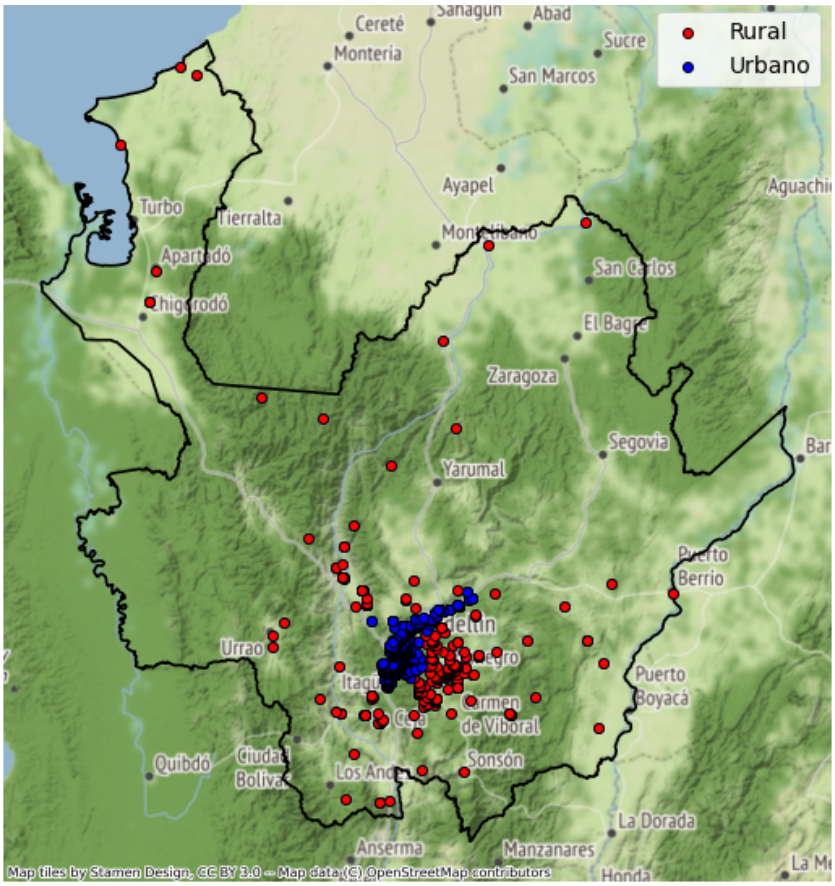

Figura 2. Localización de los datos en Antioquia

En la Figura *3*, se analizan los datos dentro del Área Metropolitana del Valle de Aburrá (AMVA), compuesta por los siguientes municipios: Medellín como ciudad núcleo, Barbosa, Girardota, Copacabana, Bello, Itagüí, Sabaneta, Envigado, La Estrella y Caldas. 2748 registros se encuentran dentro de esta división de los cuales el municipio con mayor cantidad de registros es Medellín con un (46.5%), le sigue Envigado (21.7%) y Bello con (12.9%) el resto de los municipios no supera el 20%. Comparando los precios de la ladera occidental de Medellín con los de la ladera oriental.

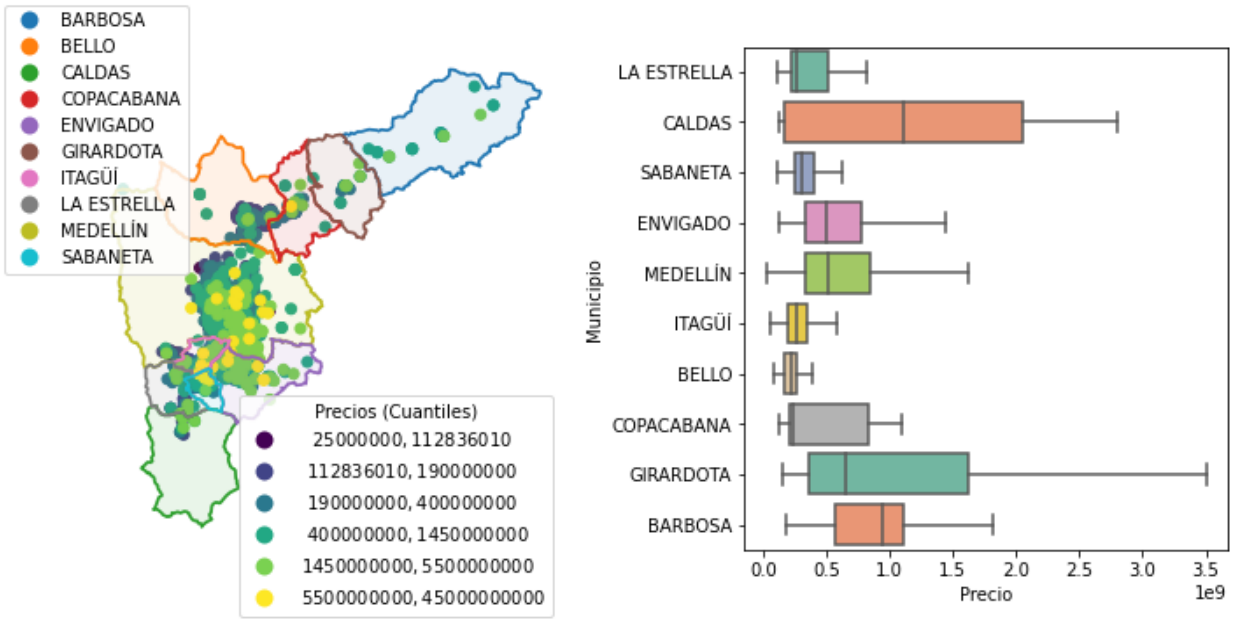

Figura 3. Análisis de datos en el AMVA

Luego se analiza la columna categórica Property Type. En la Figura *4*, la gráfica derecha muestra el recuento del número de muestras de cada tipo; el mayor número de registros corresponde a Apartamentos que es lógico teniendo en cuenta la alta densidad de éstos en el área metropolitana. Así mismo, en el gráfico de la izquierda se muestra la distribución de los precios según el tipo de propiedad. Se observa que los apartamentos tienden a ser menos costosos que las casas y que el mayor número de datos atípicos suceden principalmente en los lotes.

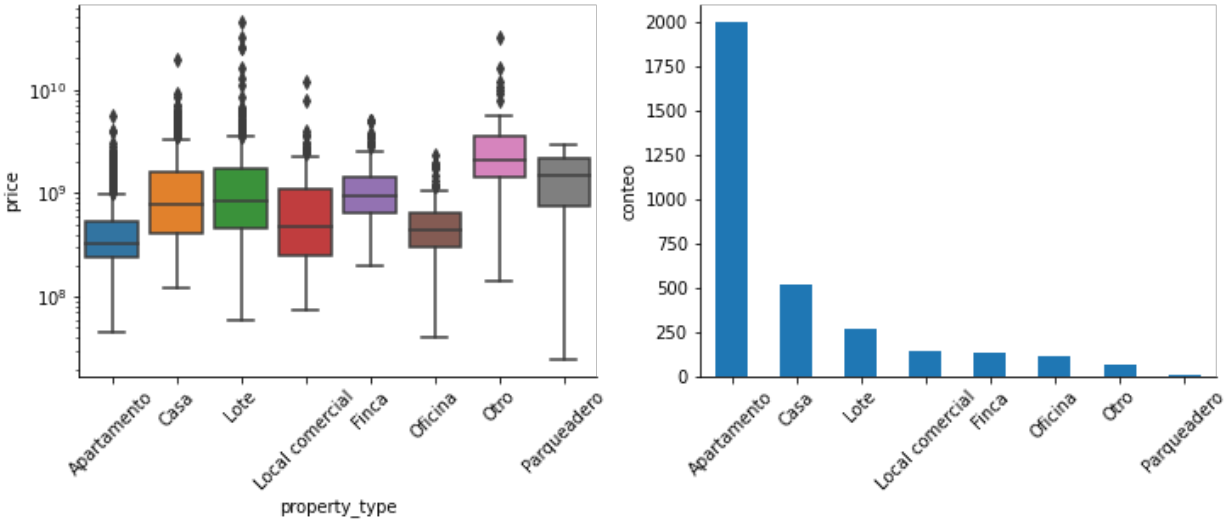

Figura 4. Diagrama de bigotes (Izquierda) y número de registros por tipo de propiedad.

Finalmente se analizan las variables predictoras y su relación con la variable objetivo, esto con el fin de descartar variables. Para ello, se realiza un análisis visual con gráficos de dispersión con el respectivo coeficiente de correlación. No se observa para ningún par de variables una dependencia lineal significativa, es decir una correlación superior a 0.7; las más altas son entre el número de habitaciones y de baño como se observa en la Figura *5*.

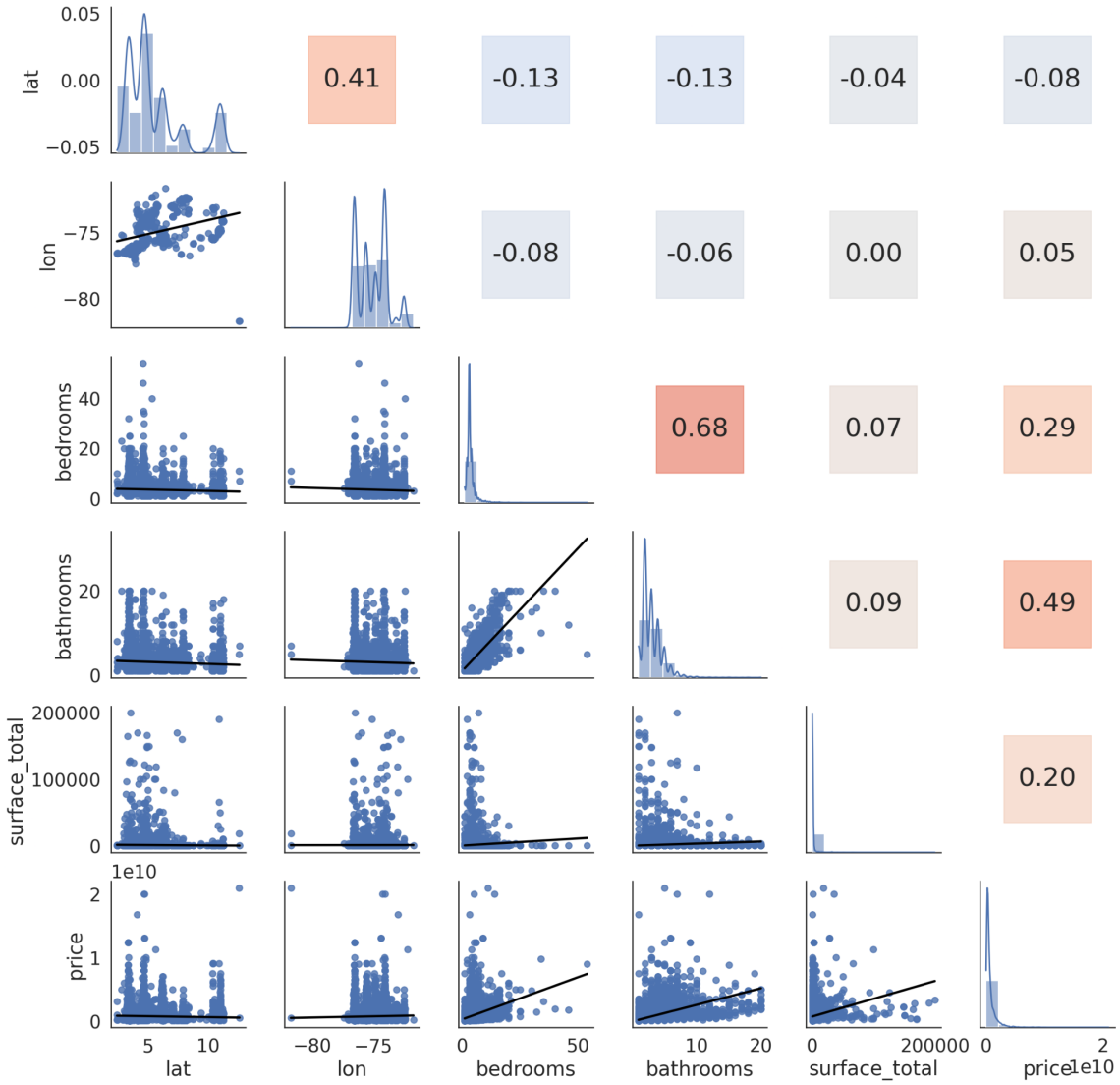

Figura 5. Análisis de las variables predictoras.

4. PROCESO DE ANALÍTICA

4.1 PIPELINE PRINCIPAL

4.1.1 PROCESO DE ELT

Para el proceso de Extracción, Cargue y Transformación de los datos se diseñó de la siguiente manera:

1. El flujo de los datos inicia con la recolección en páginas web mediante web scraping. El desarrollo del scrapper está diseñado para capturar la máxima cantidad de información en la vista de un inmueble en venta (ver Figura *6*), por lo que los datos en este paso están crudos, sin una estructura tabular fuerte definida. Los datos capturados se almacenan de manera persistente en archivos de texto plano.

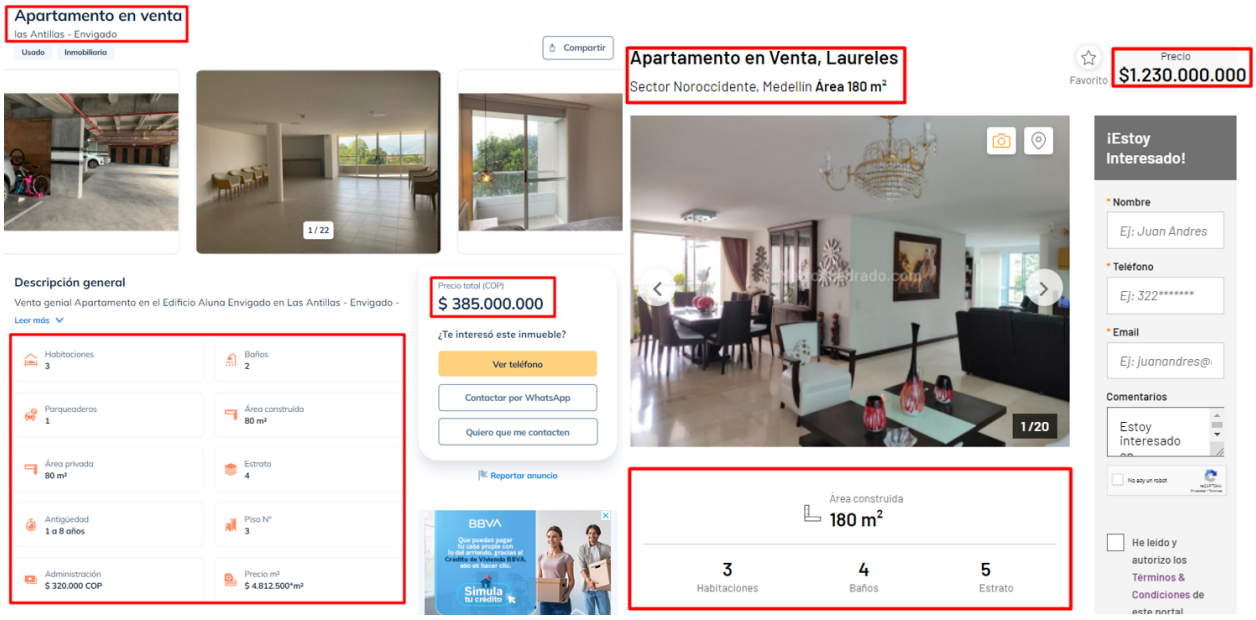

Figura 6. Portal Web de la plataforma Finca Raíz (izquierda) y Metro Cuadrado (derecha).

2. El siguiente paso en el pipeline interpreta los datos del archivo de texto plano y le da estructura para insertarlos en una base de datos relacional sin tipado de datos (String). Los filtros no son exigentes en esta etapa y los datos se insertan en un esquema de datos de crudos
3. Las páginas pueden variar en el total de datos proporcionados para un inmueble, por lo que en esta etapa se determina cuáles variables se utilizarán en el modelo, y de acuerdo a los rangos posibles de variación se realiza un tipo de los datos. Estos registros filtrados y procesados se almacenan en un esquema llamado “procesados”.

4. Como los vendedores pueden publicar en las páginas un mismo inmueble al mismo tiempo, el siguiente filtro que se considera en el flujo es de datos duplicados. Esto se logra a partir de la creación de una base de datos con el índice de los inmuebles únicos de acuerdo con la ubicación.
5. El feature engineering se realizará para los registros anteriores, y contará con datos tabulares resultado de consultas a una API que realiza operaciones geográficas sobre la ubicación (lat, lon) del inmueble, tales como vegetación, densidad vial, pendiente del terreno, entre otros. Estos datos se almacenarán en un esquema llamado “feature engineering o fe''. Los datos categóricos fueron transformados con el algoritmo One Hot Encoding.
6. El esquema Máster contará con los datos de los esquemas “procesados” y “feature engineering” cruzados con la base de inmuebles únicos. Este contendrá las tablas finales de entrenamiento y testing. En esta etapa, antes de consultar los datos primero se consultará a una tabla distribuciones, alimentada manualmente por el negocio. Esto último con el propósito de controlar el Data Drift, que será explicado posteriormente en la sección de “Consideraciones para el despliegue”.
7. El pipeline de extracción de datos finaliza en la anterior etapa, sin embargo, para efectos de control de distribuciones consumidas en la aplicación, los datos de sesión del usuario, inputs y outputs se almacenan, para su posterior análisis como estimación de la distribución de los datos que se ingresan al modelo en producción.

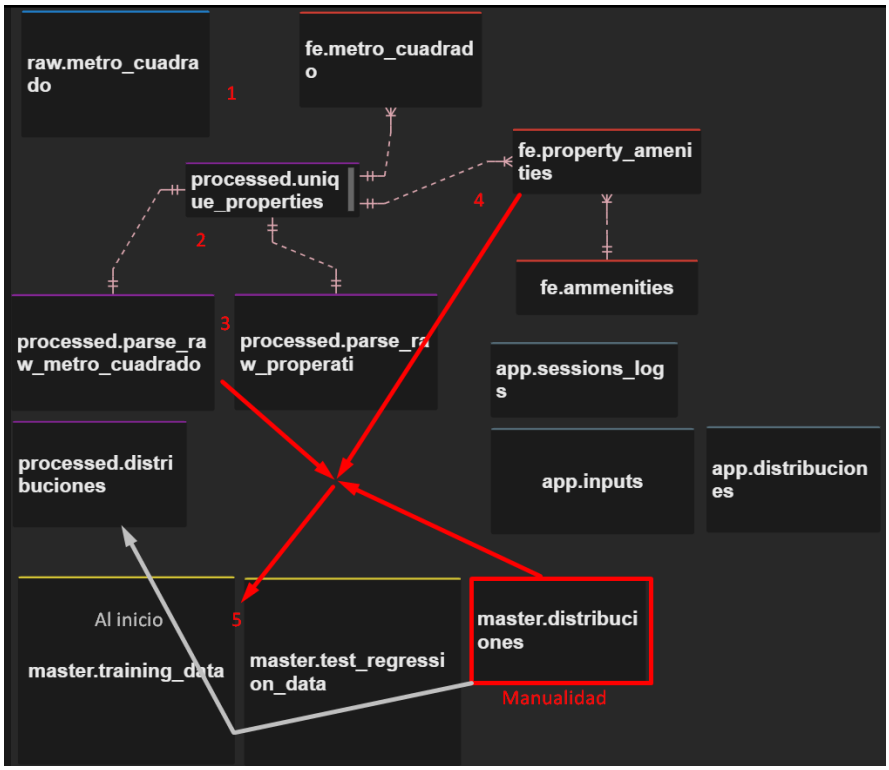

Figura 7. Modelo relacional de la base de datos.

4.3 MODELOS

Los modelos evaluados en el proyecto son LightGBM [2] y Random Forest Regressor[3], el primero es un algoritmo de refuerzo de gradientes (gradient boosting) basado en modelos de árboles de decisión y diseñado para ser distribuido y eficiente, es desarrollado por Microsoft, el segundo modelo es un ensamble de árboles de decisión.

Las configuraciones evaluadas en la grilla para la búsqueda de los hiperparámetros se muestran a continuación. El hiper parámetro seleccionado se señala en color rojo.

1. LightGBM:
   - ➔ metric: ["**l2**", "l1"]
   - ➔ num_leaves: [20, 30, 40, **80**, 100]
   - ➔ learning_rate: [**0.1**, 0.03, 0.003]

2. Random Forest:
   - ➔ bootstrap: [True, **False**]

[3] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html

- ➔ n_estimators: [**50**, 100, 200]
- ➔ max_depth: [null, 10, **20**, 30, 60]
- ➔ max_features: ['auto', **'sqrt'**]
- ➔ min_samples_leaf: [**1**, 2, 4]
- ➔ min_samples_split: [**2**, 5, 10]

4.4 MÉTRICAS

La métrica de desempeño escogida es MAPE, se calcula con la siguiente ecuación:

$$MAPE = median\left(\left|\frac{y-\hat{y}}{y}\right|\right)\%$$

Donde:

- y es el valor observado.
- $\hat{y}$ es el valor estimado por el modelo.

5. METODOLOGÍA

5.1 BASELINE Y EVOLUCIÓN

En la primera iteración se realizó una regresión lineal múltiple para interpretación de las relaciones entre las variables regresoras con la variable objetivo. Los resultados arrojaron que para la estimación del precio total, tener tipos de inmuebles como apartamento, oficina y casa, indican una relación inversamente proporcional al precio, algo que sin duda sería opuesto si se estimaran los precios por metro cuadrado. Los inmuebles de tipo lote o local comercial mostraron una relación positiva con el precio, asimismo, en términos de características internas, el número de baños mostró en este análisis una relación fuerte positiva con el precio del inmueble. La precisión medida con el MAPE para la regresión lineal correspondió con aproximadamente el 30% para el área urbana y 50% para el área rural.

La regresión lineal parte de la asunción de que las variables predictoras tienen una relación lineal con la variable objetivo, por lo que los coeficientes resultantes son constantes y tienen que ver con la pendiente y proporcionalidad de la relación. Sin embargo, el problema es complejo y no lineal, ya que para ciertas variables la relación no se puede establecer con un coeficiente constante, por ejemplo, en el caso de las coordenadas, una misma latitud puede variar significativamente su relación con el precio como función de las características internas del inmueble e incluso la longitud (e.g. acercarse o alejarse al Valle de Aburrá). También es evidencia de la no linealidad el desempeño de la regresión que estuvo alrededor de 30% para zonas urbanas y 50% para zonas rurales, muestra de una alta varianza en la predicción, además,

los porcentajes son muy altos si se comparan con un peritaje manual. Por esta razón en el segundo paso de la evolución del Baseline, se consideran otros modelos como LightGBM y RandomForest.

Luego de tener definido los modelos, el paso siguiente consistió en buscar variables relacionadas a partir de la localización de cada vivienda. Se realizó Feature Engineering de dos características que tradicionalmente han tenido relación con el precio de los inmuebles; la primera es el índice de vegetación normalizada (NDVI por sus siglas en inglés), el cual es un proxy del estado de la vegetación del suelo a partir de imágenes satelitales. El dato fue tomado de la API de Google Earth Engine [3] y se tomó el promedio de los 500 metros a la redonda de los inmuebles.

La segunda característica consistió en cuantificar las atracciones y lugares de interés como centros comerciales, restaurantes y parques dentro de un área de 500 metros de la vivienda (mínima distancia que puede llegarse a pie en diez minutos); este trabajo fue llevado a cabo con la ayuda de Open Street Map y la librería para Python OSMnx [4] . Adicionalmente, se realizarán diferentes iteraciones para validar que los nuevos features realmente tienen un impacto positivo en el desempeño del modelo.

En la siguiente tabla se resume el Feature Engineering:

Tabla 2. Campos feature engineering.

Campo	Descripción
NDVI	Índice de vegetación normalizada (NDVI por sus siglas en inglés) en un área de 500m
Food	Restaurantes en un área de 500m
Education	Universidades y colegios en un área de 500m
Noise Place	Bares y clubs nocturnos en un área de 500m
Entertainment	Parques, centros comerciales, estadios, gimnasios, piscinas en un área de 500m

5.2 VALIDACIÓN

Esta etapa es de suma importancia ya que se evalúa si el modelo está sobreentrenado, o tiene sesgos. Los modelos antes de ser útiles para la solución del problema deben de tener definido cuales son los hiperparametros con los que será entrenado; los hiperparámetros son una configuración externa al modelo que controla factores como el proceso de aprendizaje del modelo y determina los valores de los parámetros que el modelo tendrá una vez esté entrenado.

Los hiperparámetros fueron estimados mediante una búsqueda exhaustiva (Grid Search CV), en la que existe un paso de evaluación con validación cruzada: se entrenan varios modelos de ML en subconjuntos de los datos de entrada disponibles y se evalúa con el subconjunto complementario de los datos. Se evalúan diferentes métricas que permitan conocer la calidad del modelo. Se compara esta en los datos de entrenamiento y en los datos de validación. Si el error es alto puede ser debido a que se puede mejorar los hiperparámetros del modelo o realizar feature engineering.

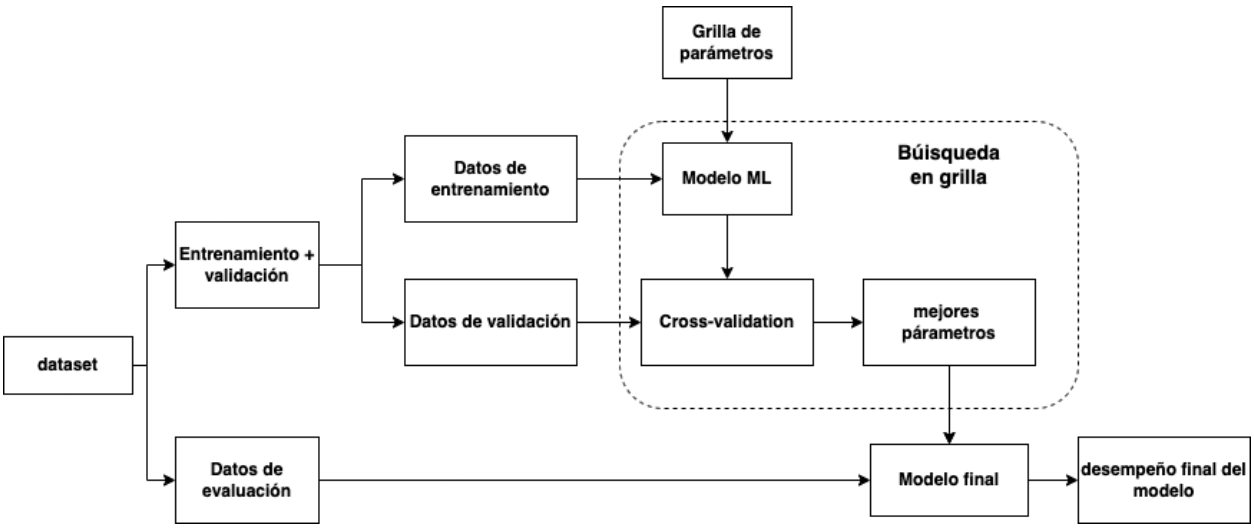

Figura 8. Flujo del dato para la validación.

Los datos se dividen en un dataset de entrenamiento (80%) y el 20% restante se utiliza para el dataset de prueba + validación. Los porcentajes se escogen según la cantidad de información disponible. Estos conjuntos cumplen las siguientes condiciones:

- Entrenamiento: Estos son los datos con los que se construye el modelo y se estiman los parámetros.
- Validación: Ayuda a determinar la configuración de los hiperparámetros y a seleccionar el mejor modelo.

- Prueba: Es el último subset del dataset que se mantiene aparte y sobre la cual se determinan las métricas que permite obtener la evaluación final del modelo.

La idea detrás de esto es conocer si el modelo ha aprendido propiedades y patrones de la población (generalización) y no simplemente ha memorizado las propiedades de los datos con los que fue entrenado. Al dividir los datos disponibles en tres conjuntos, se reduce drásticamente el número de muestras que pueden utilizarse para el aprendizaje del modelo, y los resultados pueden verse determinados por una elección aleatoria del par de conjuntos (de entrenamiento, de validación).

Una solución a este problema es un procedimiento llamado validación cruzada (CV para abreviar). Para la evaluación final debe seguir existiendo un conjunto de pruebas, pero el conjunto de validación ya no es necesario cuando se realiza la CV. En el enfoque utilizado en el proyecto el CV k-fold, el conjunto de entrenamiento se divide en k conjuntos más pequeños. El procedimiento para cada uno de los k "pliegues" consiste en entrenar un modelo utilizando k-1 de los pliegues como datos de entrenamiento; el modelo resultante se valida en la parte restante de los datos. La medida de rendimiento obtenida mediante la validación cruzada de k-fold es entonces la media de cada uno de los valores calculados.

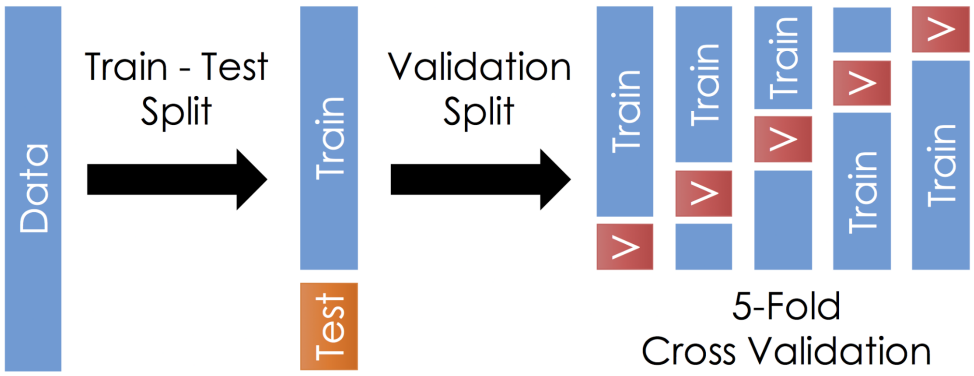

Figura 9. Metodología usada para dividir el dataset *[5]*

5.3 HERRAMIENTAS

Las herramientas usadas para el proyecto se relacionan a continuación:

Ambiente de desarrollo:

- Docker (con docker-compose)
- Debian - Ubuntu

Base de datos:

- PostgreSQL 14.2

Frameworks

- Apache Airflow

Librerías de Python:

- LightGBM
- Scikit-learn
- FastApi
- Pandas
- GeoPandas
- SqlAlchemy
- DBT
- Scrapy

Stack Tecnológico en Google Cloud

- ➔ Bigquery
- ➔ Firestore
- ➔ Google Cloud Storage
- ➔ Cloud Run
- ➔ Cloud Functions
- ➔ Compute Engine
- ➔ Google Kubernetes Engine
- ➔ Cloud Monitoring
- ➔ Container Registry

6. RESULTADOS

6.1 MÉTRICAS

En la Tabla *3* se presentan los resultados con la métrica para los diferentes modelos:

Tabla 3. MAPE de los diferentes modelos en el conjunto de prueba.

Modelo	Urbano	Rural
LightGBM	12.7	17.5
Random forest regression	11.7	11.1
Regresión lineal	29.6	51.3

El modelo final utilizado en el proyecto es el Random Forest. Esto debido a que tiene el error más bajo en áreas urbanas y rurales.

6.2 EVALUACIÓN CUALITATIVA

La curva de aprendizaje permite determinar el error en la predicción de un modelo de Machine Learning a medida que aumenta el tamaño del conjunto de entrenamiento. La curva de aprendizaje obtenida en el proyecto mostró que una adición de 500 inmuebles representa un aumento importante para el negocio en el desempeño del modelo.

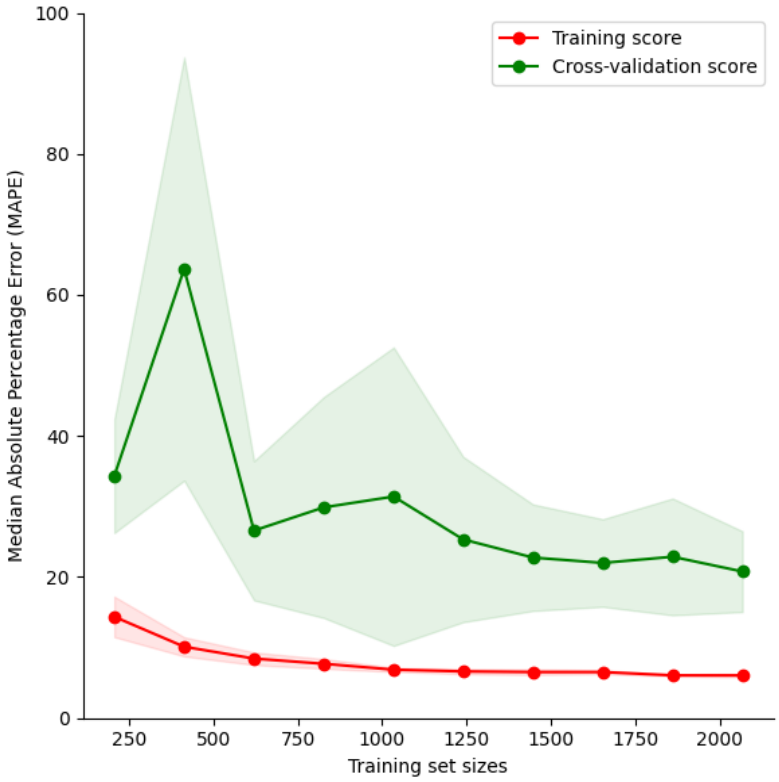

Figura 10. Curva de aprendizaje con el modelo LightGBM. Metrica Urbano y Rural combinada.

En la Figura *11* se presenta un diagrama de dispersión entre los valores reales y los valores predichos en espacio logarítmico. La mayoría de los valores oscilan en un ángulo de 45° representando una relación lineal, con algunos errores para los valores más altos y para los más bajos, no obstante, no se observa ningún patrón que sugiera un mayor enfoque para buscas alternativas de reentrenamiento.

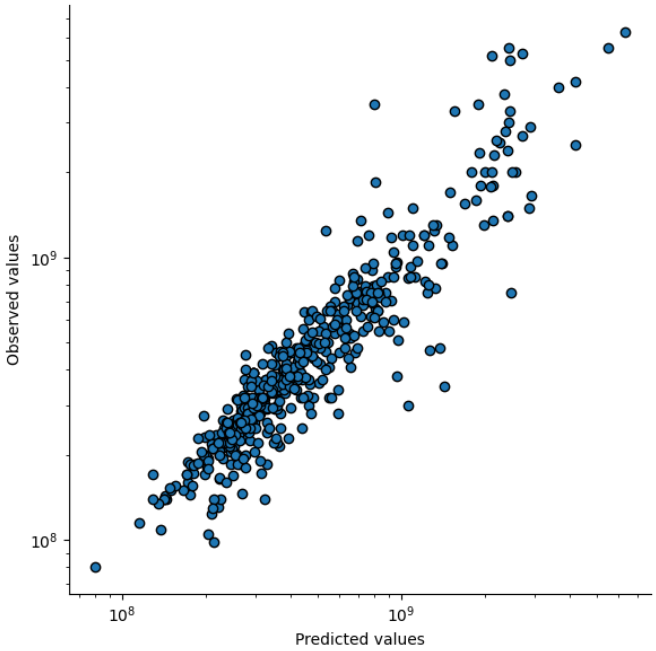

Figura 11. Gráfico de dispersión.

Aunque la interpretación de los coeficientes del modelo no es directa como con la regresión lineal, es posible obtener la importancia relativa de las características dentro del modelo. La Figura *12* muestra las características con su relevancia en el modelo LightGBM, se puede observar como el área del inmueble es la característica más importante, seguida de la ubicación, luego siguen los atractores y, finalmente, el número de baños y habitaciones. Los factores menos importantes para el modelo son el tipo de propiedad.

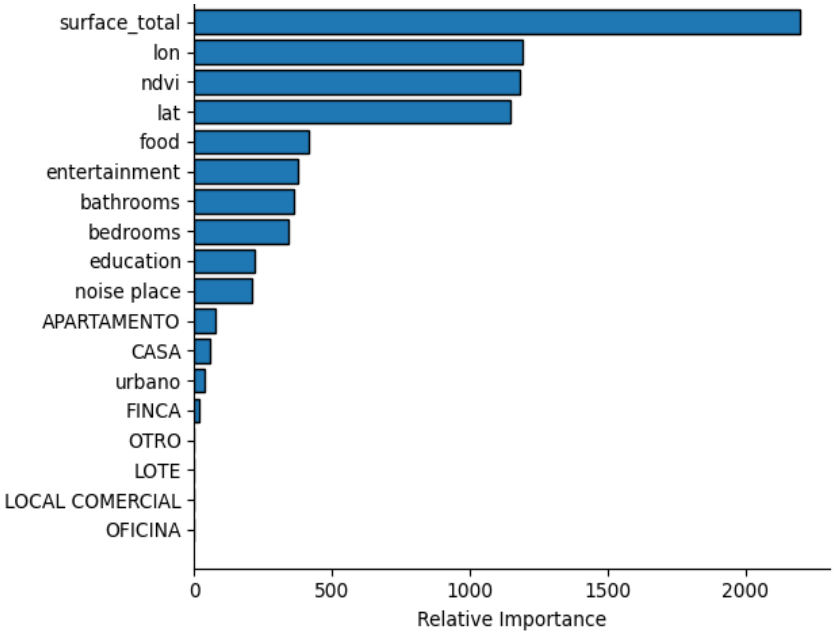

Figura 12. Importancia de los features según el modelo elegido.

6.3 CONSIDERACIONES DE PRODUCCIÓN

Arquitectura Producto Alto Nivel

La puesta en producción con una vista de alto nivel cuenta con 7 distintas capas que se aprecian en la Figura *13*. Las capas se despliegan en servicios de la nube que se describirán más adelante. El pipeline que orquesta la extracción de datos hasta el despliegue del modelo en la capa del servicio es con la herramienta Apache Airflow y se usan frameworks para transformación de datos como DBT.

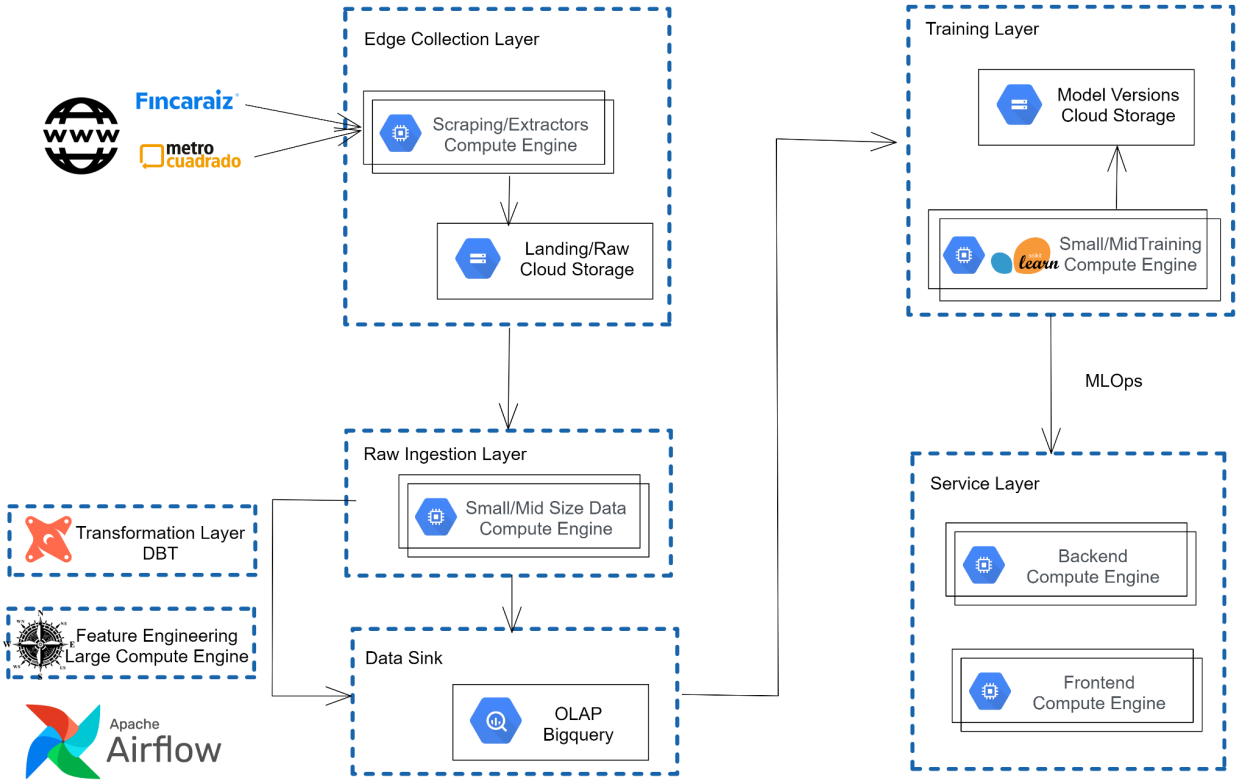

Figura 13. Arquitectura de alto nivel.

La primera capa de recolección contiene los desarrollos que obtienen datos de las distintas fuentes. Consiste en la ejecución de los scrappers en máquinas virtuales, para extraer datos de los portales Finca Raíz y Metro Cuadrado. El desarrollo se hace sobre Python con frameworks como Selenium, Beautiful Soup y Scrapy.

La segunda capa de ingestión de datos crudos obtiene el archivo en texto plano de la anterior etapa, almacenado en un volumen de datos persistente como Google Cloud Storage, y lo inserta en la base de datos. En un modelo de largo plazo y a gran escala, esta base de datos es llamada

el Datawarehouse. Se utiliza arquitectura de puertos y conectores para el desarrollo local con PostgreSQL por su fácil migración a BigQuery.

La capa de transformación se considera suficiente con las funciones y sintaxis SQL del DataWarehouse, por lo que el framework DBT se utiliza para la creación de tablas transformadas. Este framework facilita la productivización de transformaciones sobre los datos crudos, al permitir que con un archivo .sql ubicado en una ruta específica se cree una tabla procesada en el DataWarehouse.

Como se tiene pensada esta extracción, requiere un nivel alto de cómputo al tratarse de operaciones numéricas con datos geográficos, su inclusión añade costos considerables en servicios del proveedor de nube, pero mejora las métricas hasta los hitos establecidos.

La capa de entrenamiento requiere tener la capacidad de memoria y CPU para entrenar el modelo de acuerdo al volumen de datos estimado, este crece en el tiempo por lo que debería ser escalable. Para el desarrollo se utilizó Python con los framework Scikit-learn y TensorFlow. Dentro de esta capa residen los pipelines de despliegue de Machine Learning que se comentarán más adelante.

La capa del servicio tiene un backend y frontend. El primero tiene las siguientes tareas:

1. Predice con el modelo los inputs de los consumidores y devuelve una respuesta
2. Realiza la persistencia de los inputs y outputs de la sección usados para Data Drift. Esto es: antes de desplegar el modelo debe de hacerse una prueba de regresión que permitirá determinar si el último entrenamiento del modelo cumple con los requerimientos del negocio, es decir si mejora el desempeño del modelo. Para esto último juegan un papel importante dos conceptos el *Data drift* y el *Concept drift.*

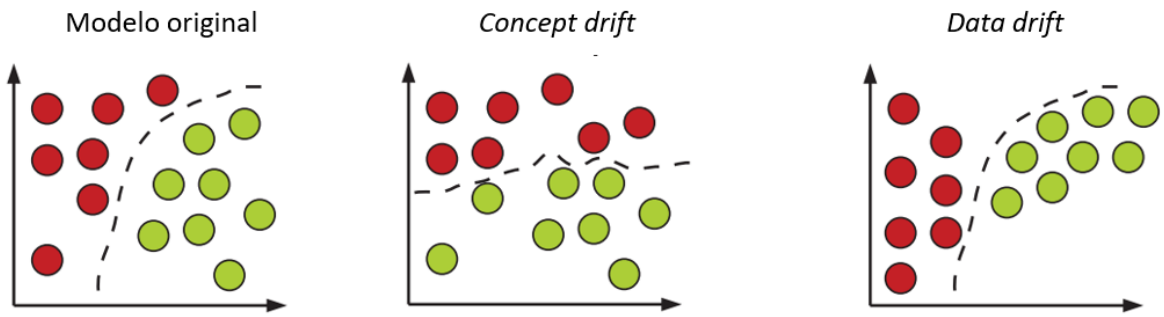

Figura 14. Tipos de drift en un modelo de Machine Learning. (Adaptación propia *[6]*).

3. Contiene el servicio de alarmas en caso de que los inputs del frontend estén por fuera de los rangos de entrenamiento de las variables, y no devuelva una predicción al frontend.

Arquitectura Producto Medio Nivel

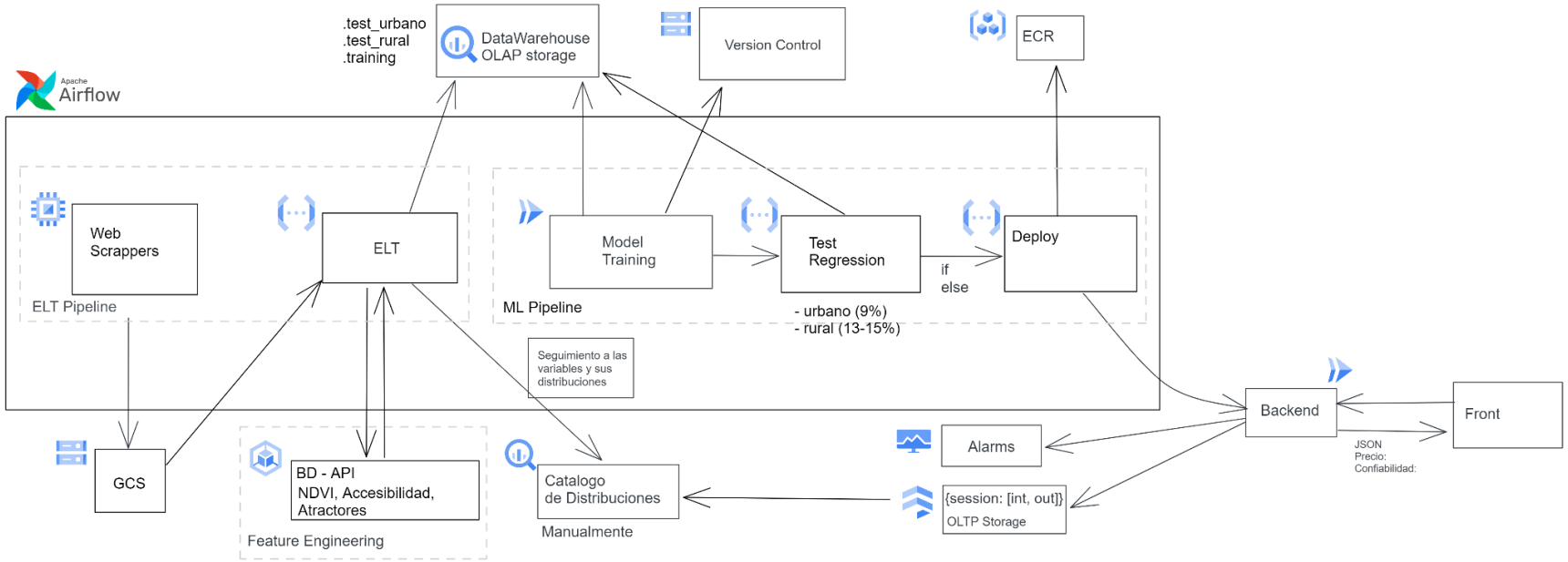

Figura 15. Arquitectura del pipeline principal a medio nivel

Consideraciones Productivas para Despliegue del Modelo

Los pasos explicados anteriormente corresponden a la ejecución completa del pipeline, desde la recolección hasta el despliegue. Sin embargo, el modelo tiene consideraciones adicionales de despliegue que no necesariamente están ligadas a la extracción de los datos en la fuente y que se explican a continuación:

1. El primer despliegue considerado (ver Figura *16* línea naranja) es el que se da a través del reentrenamiento de los modelos con la llegada de nuevos datos con los Web Scrappers (pipeline completo). La frecuencia de este pipeline se obtiene bajo dos criterios: en primer lugar, como el *concept drift* es intrínseco al problema por fenómenos macroeconómicos los precios de las viviendas siempre aumentan, y en Colombia esta frecuencia se puede considerar semestral. En segundo lugar, la curva de aprendizaje determina las ganancias en la métrica de desempeño con el aumento en los registros de entrenamiento adicionales. La curva de aprendizaje obtenida en el proyecto mostró que una adición de 300 inmuebles representa un aumento considerable en el desempeño del modelo. No se calcula cada vez que la operación comercial en las páginas web fuente tengan un incremento de este orden, sin embargo, se estima que sea por lo menos de un mes.
2. La segunda consideración de despliegue (ver Figura *16* línea azul) es producto de la analítica realizada por los científicos de datos durante todo el ciclo de vida del aplicativo. Con una nueva versión del código fuente que estos produzcan, el modelo se reentrena. En otras palabras, esta nueva ruta del pipeline iniciaría con la finalización del pipeline de DevOps que actualiza el código fuente del modelo en producción.
3. El tercer despliegue (ver Figura *16* línea verde) se da como amortiguación al Data Drift que se presente durante todo el ciclo de vida del aplicativo. Esto se logra a través del cruce entre las distribuciones estadísticas de las variables predictoras extraídas por los Scrapers, y las distribuciones consumidas en el aplicativo. Este análisis se realiza de manera manual en compañía del negocio, y se almacena en una base de datos de distribuciones (ver sección ELT). Los rangos que se determinen serán la capacidad del producto de responder a las consultas de los consumidores. Por ser esta la distribución real consumida en producción y el enfoque del negocio, se despliega el modelo para las mejores métricas en esta versión de los datos.

Pipelines de despliegue del Producto Medio Nivel

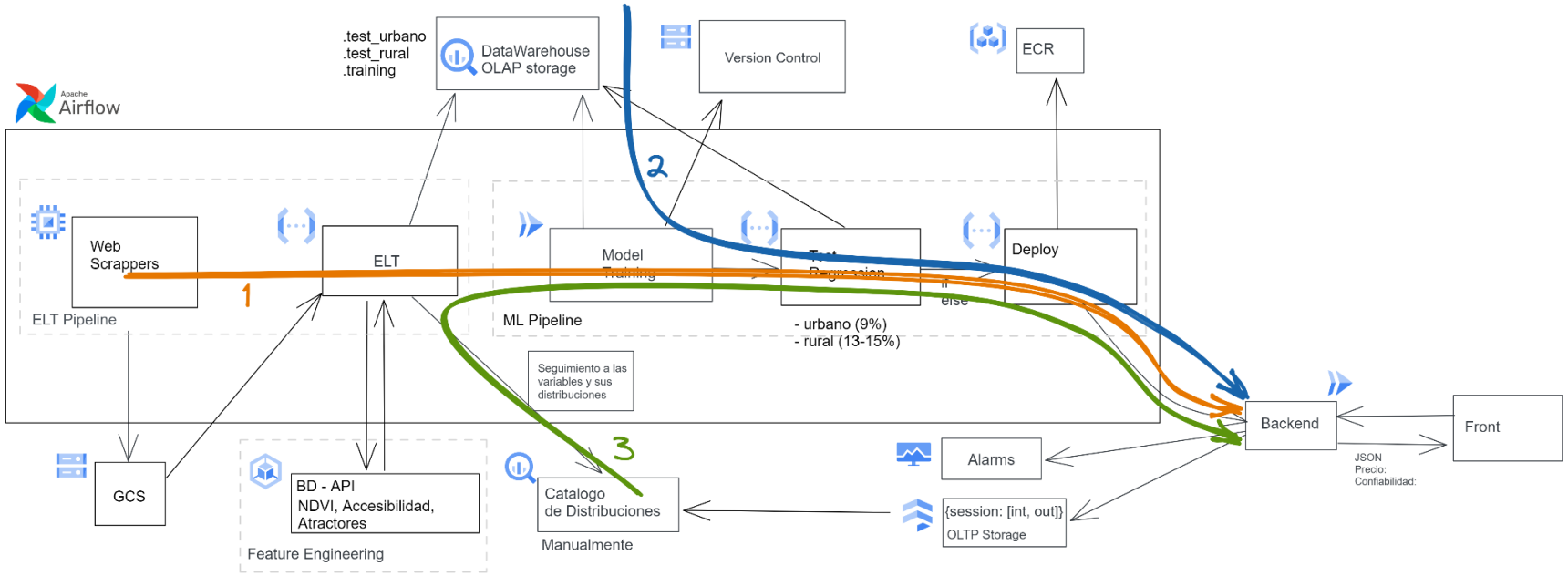

Figura 16. Pipelines para despliegue del modelo

7. CONCLUSIONES

Inicialmente calculamos una regresión lineal como modelo de referencia para entender los factores que determinan el valor comercial. Sin embargo, este modelo no captó el efecto dinámico y complejo que rige el negocio de la propiedad raíz. Dadas estas limitaciones implementamos modelos de Machine Learning (no lineales) como LightGBM y Random Forest que permiten tener mejores predicciones de los valores comerciales, pero con cierta pérdida de interpretabilidad de las características para finalmente tener un modelo con mejor desempeño en el MAPE con un error cercano al 12%.

La metodología propuesta se diferencia de otras aplicaciones porque permite una actualización continua de la información con páginas webs de propiedad raíz. El proyecto incluye nuevas características para estimar el valor comercial de las propiedades como cercanía a espacios públicos e índices de vegetación.

Para pasos siguientes, valdría la pena acotar el modelo a solo áreas urbanas o rurales según las necesidades del negocio, este nuevo enfoque puede ampliarse para incluir más variables en diferentes lugares. Una posible ampliación sería utilizar nuevos features como densidad vial, pendiente, elevación del terreno, entre otros, con el fin de complementar las características de las propiedades. En el futuro esperamos aumentar la información de las fuentes y las estrategias de recopilación, y desplegar el proyecto en la nube.

En este trabajo se discutieron los roles de los pipelines de ingeniería de datos y Machine learning, además de tres distintas metodologías para despliegue de los modelos y consideraciones en producción para un modelo alimentado con datos tabulares. Las metodologías tratan de mitigar el Concept Drift y Data Drift para un problema que por muchas evidencias varía con el tiempo, además, tiene en cuenta el ciclo de vida de despliegue de la analítica en desarrollo que pasa a producción. Una discusión adicional se podría dar para datos no estructurados, ya que pipeline de ingeniería de datos podría no ser el apropiado para el manejo de transformaciones, y por lo tanto el pipeline de Machine Learning se extendería en un paso adicional.

La propuesta se piensa evolucionar hasta presentar el ensamblaje de múltiples modelos para servir en el aplicativo de los usuarios finales. Esto con la finalidad de capturar las distintas hipótesis y performance que tienen los modelos, algunos podrían ser mejores en precios bajos y otros en los altos, por lo que le daría robustez al producto.

BIBLIOGRAFIA

[1] Properati, «Properati Data,» [En línea]. Available: https://www.properati.com.co/data/.

[2] «Repositorio LightGBM,» [En línea]. Available: https://github.com/microsoft/LightGBM .

[3] N. L. P. D. A. A. C. (. DAAC). [En línea]. Available: https://developers.google.com/earth-engine/datasets/catalog/MODIS_061_MOD13Q1 .

[4] G. Boeing, «OSMnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks,» *Computers, Environment and Urban Systems,* vol. 65, pp. 126-139, 2017.

[5] [En línea]. Available: https://qph.fs.quoracdn.net/main-qimg-724adf1ec221dbc16bcf5e91646d641b.

[6] [En línea]. Available: https://dl.acm.org/doi/10.1145/2523813.