Modelo de Predicción del Precio de la Vivienda en el Valle de San Nicolás

Robin Andrés Soto Hincapié

Estefanía David Rodríguez

Monografía presentada para optar al título de Especialista en Analítica y Ciencia de Datos

Tutor
Julián David Arias Londoño, Doctor (PhD)

Universidad de Antioquia
Facultad de Ingeniería

Especialización en Analítica y Ciencia de Datos

Medellín, Antioquia, Colombia

2021

Cita	(Soto Hincapié & David Rodríguez, 2021)
Referencia **Estilo APA 7 (2020)**	Soto Hincapié R. A. & David Rodríguez E. (2021). Modelo de Predicción del Precio de Vivienda en el Valle de San Nicolás [Trabajo de grado especialización]. Universidad de Antioquia, Medellín, Colombia.

Especialización en Analítica y Ciencia de Datos, Cohorte II.

Universidad de Antioquia - www.udea.edu.co

Rector: John Jairo Arboleda Céspedes

Decano/Director: Jesús Francisco Vargas Bonilla

Jefe departamento: Diego Jose Luis Botia Valderrama

Tabla de contenido

Lista de tablas

Lista de figuras

Siglas, acrónimos y abreviaturas

Esp.	Especialista
UdeA	Universidad de Antioquia
Itr	Iteración
MAPE	Median Absolute Percentage Error
$\boldsymbol{R^2}$	Coeficiente de determinación
RMSE	Root mean squared error
MedAE	Median Absolute Error
ML	Machine Learning
GBT	Gradient Boosting Tree
RNN	Red Neuronal Recurrente
ANN	Red Neuronal Artificial
XGBoost	Extreme Gradient Boosting
Stopwords	Palabras vacías

Resumen

A través del presente documento se busca explorar las mejores técnicas de aprendizaje supervisado para predicción de precios de vivienda para un conjunto de datos de 2.481 registros extraídos del portal fincaraiz.com. Inicialmente se tratarán algunos aspectos como el planteamiento del problema y los objetivos, luego se hará un recorrido a través de las distintas técnicas de aprendizaje supervisado usadas en la actualidad, haciendo énfasis en aspectos clave como la selección de hiperparámetros o la idoneidad de los modelos a la hora de abordar el problema de predicción de precios. Posterior a esto se realizará una revisión del conjunto de datos con el cual se pretende realizar el proceso de modelado, partiendo de las características básicas del dataset y luego ahondando en algunos hallazgos logrados a partir del análisis descriptivo. Una vez se tenga conocimientos sobre el dataset, se procederá a explicar el diseño de las pruebas y las distintas iteraciones realizadas, con el objetivo de que los lectores puedan replicar el proceso experimental [1]. Finalmente se hará una explicación de los resultados obtenidos y las implicaciones del uso de modelos basados en técnicas de ensamble y redes neuronales, en contraste con otras técnicas estadísticas como la Regresión Lineal Múltiple y la Regresión Ridge.

Palabras clave — Machine Learning, Redes Neuronales, Regresión, Aprendizaje Supervisado, Árboles de decisión, Gradiente Descendente, Random Forest.

[1] El siguiente repositorio de git contiene los notebooks usados en el proceso: https://github.com/andres-soto-h/monografia-udea-eacd

Abstract

This document seeks to explore the best supervised learning techniques for house price prediction for a dataset of 2,481 records extracted from the fincaraiz.com portal. Initially, some aspects such as the statement of the problem and the objectives will be treated, then it will be explained the different supervised learning techniques used today, emphasizing key aspects such as the selection of hyperparameters or models suitability when it comes to address the problem of price prediction. After this, a review of the data used for the modeling process will be carried out, starting from the basic characteristics of the dataset and then delving into some findings obtained from the descriptive analysis. Once the dataset is known, we will proceed to explain the design of the tests and the different iterations carried out, with the aim that the readers can replicate the experimental process [2]. Finally, an explanation will be made of the results obtained and the implications of the use of models based on assembly techniques and neural networks, in contrast to other statistical techniques such as Multiple Linear Regression and Ridge Regression.

Keywords — Machine Learning, Neural Networks, Regression, Supervised Learning, Decision Trees, Gradient Descent, Random Forest.

[2] The following git repository contains the notebooks used in the process: https://github.com/andres-soto-h/monografia-udea-eacd

Introducción

Mediante el presente trabajo se pretende abordar el uso de modelos de aprendizaje supervisado, con distintos modelos de machine learning basados en regresión, esto con el objetivo de predecir los precios de la vivienda para los municipios ubicados en el valle de San Nicolás, comprendido por Rionegro, Guarne, El Carmen de Viboral, El Retiro, El Santuario, Marinilla, La Ceja, La Unión, y San Vicente ubicados en el departamento de Antioquia. Para esto se utilizó como insumo, la información obtenida a partir del portal fincaraiz.com a través de la técnica de web scraping, dónde se obtuvo un dataset inicial que posteriormente fue procesado y transformado para ajustarse a los distintos modelos de machine learning analizados a lo largo del documento.

La problemática de predecir los precios de vivienda a partir de una serie de características, es ampliamente conocida en la literatura (Park & Bae, 2015; Phan, 2018; Varma, Sarma, Doshi, & Nair, 2018), para el caso de Colombia se tienen múltiples análisis dónde se trata de hacer la estimación de precios en grandes ciudades como Bogotá o Medellín (Téllez, 2021; Tissnesh, 2014), así mismo se tiene un referente de un modelo de precios para el municipio de Rionegro (Grajales, 2019); lo anterior, sumado a la creciente demanda de unidades de vivienda en el Valle de San Nicolás, hace relevante conocer el estado del arte de las técnicas de machine learning para predicción de precios, su capacidad para modelar datasets compuestos por información de distintos municipios y cómo es posible estructurar y desplegar estos modelos usando distintas librerías con el lenguaje de programación python.

1. Planteamiento del problema

Según datos de Cornare, el Valle de San Nicolás ha presentado un crecimiento del 6,68% en su población entre los años 2015 y 2020, y se proyecta una tasa de crecimiento de 6,32% para el año 2025 esperando llegar a cerca de 459 mil habitantes (Cornare, 2015), así mismo, los datos macroeconómicos relacionados con la producción y el empleo sitúan a esta zona como el segundo centro económico más importante del departamento (Isaza, 2019). El crecimiento sostenido de la población observado en las últimas décadas, está directamente relacionado con la demanda de nuevas unidades de vivienda, que según análisis de la Cámara de Comercio del Oriente Antioqueño, se ha dado principalmente en los municipios de Rionegro y La Ceja, por el aumento en la solicitud de licencias de construcción (1583 para el año 2019) (Isaza, 2019).

Dicho lo anterior, predecir de manera acertada los precios de la vivienda en función al tipo de inmueble, área construida, ubicación, entre otras características relevantes de cada propiedad es importante, porque a partir de datos históricos permite conocer el precio para un determinado inmueble en función de sus características, así mismo, hacer un análisis de la dinámica de los precios, permite establecer cuando un proyecto de vivienda presenta un precio razonable en función a las características del mismo, o cuando se puede presentar un sobre precio atribuible a la especulación o una alta demanda en ciertos sectores del Valle de San Nicolás. Para abordar este reto se utilizarán diferentes técnicas de aprendizaje supervisado, con el fin de tener los resultados más acertados posibles y ofrecer a los usuarios finales de la solución, una estimación confiable.

1.1 Antecedentes

Uno de los referentes que se tiene para realizar el análisis de los precios de vivienda, es el modelo diseñado en el trabajo de grado de (Grajales, 2019) en éste, la autora plantea el uso de 5 técnicas de machine learning con el objetivo de abordar el problema de modelamiento de los precios de vivienda para el municipio de Rionegro, los resultados obtenidos se muestran a continuación:

Figura 1.

Resultado de modelamiento precios Rionegro (Grajales, 2019)

	R2	RMSE	MAE	MAPE	SMAPE
Regresión Lineal	0.530519	9.658456e+07	7.552920e+07	0.249191	0.109879
Decision Tree	0.551575	9.439382e+07	6.075658e+07	0.176933	0.088676
Random Forest	0.702875	7.683654e+07	5.148748e+07	0.152312	0.074897
Gradient Bosting Machine	0.710440	7.585210e+07	5.510707e+07	0.167122	0.080208
SVM	0.034413	1.385142e+08	1.064484e+08	0.339682	0.154847

De lo anterior se tiene que los mejores resultados obtenidos en este trabajo, corresponden al modelo de Random Forest y Gradient Boosting, con un MAPE del 16% en promedio y un R^2 del 70% aproximadamente.

Otro referente sobre el cual revisar resultados, es el trabajo realizado en 2021 por (Téllez, 2021) en él también se aborda la problemática de predicción de precios de vivienda, pero en este caso para la ciudad de Bogotá. Al igual que en el caso anterior, se evalúa el resultado a través de 5 modelos y 3 métricas de desempeño (Evaluación del error), un factor interesante a considerar es que este análisis se realiza particionando los modelos a partir del estrato socioeconómico y obteniendo el desempeño final a través del promedio de los submodelos. Para este caso se obtuvieron los siguientes resultados:

Figura 2.

Resultado promedio para el modelo de precios de vivienda en Bogotá (Téllez, 2021)

	Regresion Lineal	Arboles de Decision	Random Forest	Red Neuronal
Promedio R^2	0,77	0,82	0,93	0,43
Promedio RMSE	376142332	213913139	125768700	594806075
Promedio CV	0,33	0,24	0,17	0,66

De la **Figura 2** es posible concluir que el modelo con mejor desempeño es el Random Forest, así mismo el pobre desempeño en la red Neuronal, puede atribuirse a una arquitectura que no se ajusta al problema o alguna falencia en el preprocesamiento de los datos o entrenamiento del modelo

2. Justificación

El problema de predicción de precios de vivienda en el Valle de San Nicolás, es relevante porque busca dar solución a una necesidad latente en el mercado inmobiliario, dada la creciente demanda en unidades de vivienda. Como se ha mencionado anteriormente, el valle de San Nicolás, ubicado en la subregión del departamento de Antioquia, presenta un gran potencial de crecimiento y expansión, y se ha consolidado en las últimas décadas como un foco de crecimiento económico y demográfico en el departamento. Conocer de manera acertada, las variables más representativas que contribuyen a predecir el precio para un inmueble, ayuda a que las partes involucradas en el proceso de compraventa, puedan entender la dinámica del mercado y tomar decisiones informadas al momento de buscar u ofrecer una propiedad.

3. Objetivos

3.1 Objetivo general

Implementar modelos de machine learning en un conjunto de datos del sector inmobiliario para conocer la capacidad de predicción de las técnicas de aprendizaje supervisado más utilizadas.

3.2 Objetivos específicos

- Implementar distintos modelos de machine learning acordes a la problemática de predicción de precios en el sector inmobiliario.
- Explorar el estado del arte en técnicas de aprendizaje supervisado y otros trabajos similares que den una guía sobre cómo abordar el problema de predicción de precios de vivienda.
- Comprobar la capacidad predictiva de los modelos usando distintas técnicas de validación cruzada y estratificación del dataset.
- Implementar una estructura multimodal de redes neuronales para un problema de predicción de precios.
- Desplegar los mejores modelos encontrados en una plataforma de nube que permita conocer su comportamiento en datos nuevos.

4. Problema de investigación

A través del presente trabajo se busca establecer el mejor o los mejores modelos de aprendizaje supervisado para la predicción de precios de vivienda en el Valle de San Nicolás que permitan establecer un precio de compra y venta confiable para un inmueble, dependiendo de sus características y ubicación geográfica.

5. Marco teórico

En esta sección del documento se ilustra al lector sobre algunos conceptos de gran importancia al momento de abordar el proceso de modelado a partir de técnicas de machine learning.

5.1 Python [3][4]

Es un lenguaje de programación de código abierto que tiene como características principales ser un lenguaje interpretado, es decir, su código es sencillo y legible para personas con poca experiencia programando y no requiere de un compilador para ser ejecutado por un sistema predeterminado; es multiplataforma, por lo tanto, las líneas de código se pueden ejecutar en cualquier sistema operativo (windows, linux, mac, Android, entre otros); es de tipado dinámico, es decir, permite que un objeto pueda tomar diferentes valores de distintos tipos en diferentes momentos. En python no hay que declarar con anterioridad cuál es el tipo de dato que se va a almacenar en el objeto, sino que su declaración se hace de manera automática según el contenido de esta (a menos que se especifique) y es multiparadigma, lo que implica que acepta diferentes modelos de desarrollo (orientado a objetos, funcional o imperativo).

Imperativo: la programación imperativa implica especificar paso a paso las instrucciones que va a ejecutar el programa creado.

Funcional: basado en el uso de funciones matemáticas, permite operar con datos de entrada y salida, el uso de estas funciones permite ahorrar muchas líneas de código.
Dentro de las ventajas de uso de python está el uso de librerías que permite no solo ahorrar líneas de código sino reutilizar funciones o programas que ya otros han creado ahorrándote tiempo de ejecución e implementación.

[3] ¿Qué es python? (Alvarez 2003)
[4] Python (Python.org 2021)

Orientado a Objetos: la programación orientada a objetos ofrece la particularidad en la forma de obtener los resultados. Los objetos manipulan los objetos de entrada para la obtención de resultados específicos (salidas) donde cada objeto nos ofrece una función específica y también nos permite la agrupación de bibliotecas o librerías (pythones.net, 2019).

5.2 Web Scraping [5] [6]

Es una técnica que sirve para extraer información de páginas web de forma "automatizada". Es importante aclarar que la palabra automatizada se usa con cautela, ya que esto es posible siempre y cuando la estructura de la página web se mantenga constante o igual, en la realidad esto no siempre sucede, por lo tanto, si cambia la estructura de la página web, el código utilizado para realizar el web scraping de la página de interés puede quedar obsoleto o desactualizado.

El uso del web scraping tiene diferentes finalidades, la de nuestro proyecto tenía como finalidad obtener información o datos del precio de las viviendas que están en venta en el valle de San Nicolás y sus características para construir un conjunto de datos que nos sirva posteriormente para hacer uso de metodologías de modelamiento y resolver nuestro problema. Pero su uso también puede utilizarse para agregar contenido, optimizar precios, mejorar reputación online entre otras finalidades.

Las librerías que se utilizaron en el proceso de web scrapping se presentan a continuación:

selenium: permite probar y registrar las interacciones con una aplicación web y luego repetir estas interacciones las veces que se deseen. Esta librería está basada exclusivamente en HTML y JavaScript (IONOS España, 2020; Muthukadan, 2018).

beautifulsoup4: esta librería permite extraer información de contenido en formato HTML o XML. Para usarla se requiere de un parser que transforma un documento HTML en un árbol de objetos python para hacer más sencilla la extracción de la información (Python Software Foundation, 2021).

[5] ¿Qué es el web scraping? (IONOS España 2020)
[6] Web scraping (Lafuente 2019)

lxml: es una biblioteca de python para un fácil manejo de archivos HTML y XML. Permite analizar documentos grandes de forma más rápida y tiene una fácil conversión de los datos a tipo de datos de python para hacer más sencilla su manipulación (Richter, 2021; de la Vega, 2021).

5.3 Machine Learning [7]

El aprendizaje automático es la ciencia de hacer que las computadoras actúen sin estar programadas explícitamente (Andrew Ng).

Se considera una disciplina del campo de la inteligencia artificial, la cual dota a los ordenadores de la capacidad de identificar patrones de los datos y posteriormente realizar predicciones. Esta capacidad de aprendizaje por parte de los ordenadores se da a través de algoritmos o modelos tales como regresiones, modelos de ensamble, kernels, entre otros. Adicionalmente, el uso de estos algoritmos se puede guiar desde una definición de la necesidad en un contexto supervisado, no supervisado y por refuerzo.

- **Análisis Supervisado:** se entiende por análisis supervisado cuando se cuenta con un conjunto de datos que tiene información de la variable a modelar y la tarea a realizar se resuelve a través de una predicción o clasificación, es decir, el proceso de aprendizaje por parte del ordenador se da a través de datos etiquetados, es decir, se conoce previamente el valor de la variable respuesta, de manera que su aprendizaje se mejore a través de comparar el resultado obtenido con el real.

- **Análisis No Supervisado:** se entiende por análisis no supervisado cuando la necesidad es encontrar patrones o agrupaciones buscando similitudes, en este contexto, el conjunto de insumos no cuenta con una variable target o con información previa de una variable respuesta de interés, ya que no necesariamente la necesidad es modelar el comportamiento

[7] Machine learning - definiciones (Iberdrola SA 2021)

de la variable respecto a otras sino identificar agrupaciones o realizar reducción de dimensionalidad con un conjunto de variables del conjunto de datos.

A continuación se presentan los algoritmos usados en al menos uno de los experimentos con la finalidad de predecir el precio de la vivienda a través de las características principales de la vivienda y sus alrededores.

5.4 Regresión Ridge [8]

Se entiende por regresión cuando realizamos una predicción numérica o cuantitativa a partir de unas entradas.

El algoritmo más conocido en estos casos de uso es la regresión lineal, donde se asume una relación lineal entre la variable respuesta y las regresoras y tiene como objetivo predecir el valor de la variable respuesta en función de las variables regresoras. El objetivo es estimar los coeficientes que acompañan a las variables regresoras que menor error produzca.

En ese orden de idea, la regresión ridge es una regresión lineal regularizada, lo que involucra un término de penalización en la función de costo, en este caso, el término adicional en la función de costo es la suma de los pesos al cuadrado. Su efecto produce modelos más simples que tienden a generalizar mejor. Su uso es útil si se sospecha de correlación entre las variables de entrada ya que estima coeficientes más pequeños para estas variables.

5.5 Modelo de Random Forest [9]

Un bosque aleatorio es una combinación de árboles como predictores [Breiman 2001], el objetivo del uso de los bosques aleatorios es mitigar una de las desventajas del uso de los árboles de decisión que es el problema de generalización o sobreajuste de los árboles, es decir,

[8] Regularización Lasso L1, Ridge L2 y ElasticNet (Martinez 2020)

[9] Random Forest (Martinez 2020)

la varianza. Al ser una combinación de árboles, la mitigación de que el modelo se aprenda los datos se da a través de un proceso denominado Bagging o Bootstrap [Breiman 1996], donde su objetivo es construir cada uno de los árboles usando una submuestra de los datos (es decir, no se toma todos los datos para entrenar los árboles) permitiendo en cada construcción del árbol tener un resultado individual, finalmente, el resultado del bosque aleatorio es el promedio (en el caso de regresión) o la tasa de la clase que más se repite (en el caso de la clasificación).

5.6 Modelo de Gradient Boosting [10]

El modelo de Gradiant Boosting (GBT), en español, algoritmo de aumento del gradiente, también hace parte de la familia de modelos de ensamble, parte de la idea inicial del Random Forest que es un conjunto de árboles de decisión, donde su entrenamiento se hace de forma secuencial, la finalidad es que el árbol tenga información del error del árbol anterior para mejorarlo (de ahí el nombre de aumento del gradiente). El valor final que arroja el modelo es el promedio de todas las predicciones o la clase de mayor clasificación.

5.7 Modelo XGBoost [11]

El XGBoost es una versión mejorada del GBT, se conoce como el gradiente de refuerzo regularizado. La gran utilidad de este algoritmo viene de la palabra regularización, una desventaja del GBT parte del proceso de ramificación ya que este proceso se hace a través del algoritmo de recursive binary splitting, este, evalúa la división de cada predictor con el objetivo de optimizar una medida: entropía, Gini, RSS, entre otras. En cada división intenta seleccionar la mejor predictora y el mejor umbral que divide los datos de tal forma que optimice la métrica elegida, por lo que suele favorecer las variables con más niveles o categorías o la de mayor frecuencia.

Una ventaja del XGBoost, es que usa un proceso de regularización en la selección de las predictoras, permitiendo minimizar el sobreajuste en la construcción del modelo; adicionalmente

[10] Gradient Boosting con Python (Amat 2020)
[11] Complete Guide to Parameter Tuning in XGBoost with codes in Python (Aarshay 2021)

la implementación del paquete de XGBoost presenta algunas mejoras computacionales frente a la implementación del GBT en python.

5.8 Modelos de Redes Neuronales [12] [13] [14]

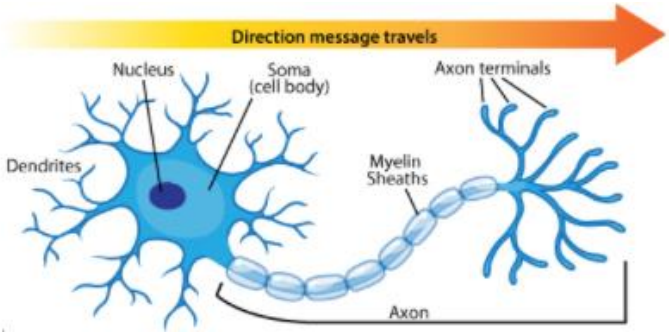
Son un tipo de algoritmos de machine learning bioinspirado, es decir, tratan de simular un comportamiento biológico, en este caso, el de las redes biológicas. Aunque en la práctica no siguen un proceso igual al biológico si coinciden en tres principios básicos:

- Se basa en unidades básicas sencillas que son capaz de realizar tareas simples
- Utiliza múltiples unidades básicas en paralelo para resolver problemas más complejos
- Involucra elementos no lineales lo que le da mayor capacidad de modelamiento al sistema.

Para entender el funcionamiento básico de una red neuronal artificial, se resalta el proceso natural de una neurona biológica donde las partes principales de conexiones que conforman una neurona biológica son: las dendritas que son las encargadas de recibir los estímulos de otras neuronas o de información externa y el axón que es el encargado de entregar esta información a otras neuronas. El tipo de respuesta que tiene una neurona es de activarse frente al estímulo o de no activarse.

Figura 3.

Diagrama de una neurona biológica.



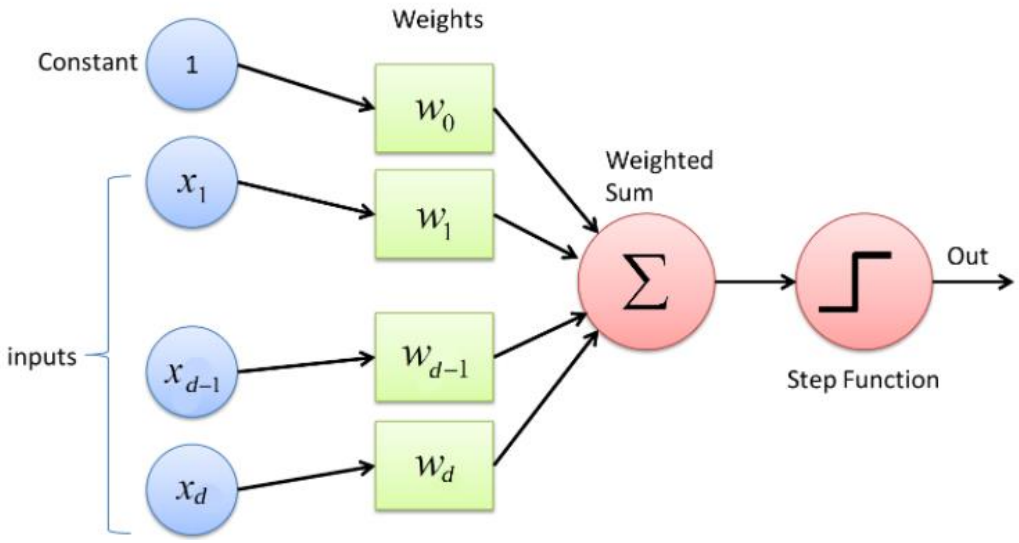
***Fuente**: Notas de clase fundamentos de deep learning (Ramos & Arias, 2020)*

[12] Tema 8. Redes Neuronales (Moujahid 2008)
[13] Deep Learning - introducción practica (Torres 2018)
[14] Introducción al deep Learning - Notas de clase (Ramos and Arias 2020)

Simulando este comportamiento en las redes neuronales artificiales, este proceso se hace a través del algoritmo del perceptrón. Donde las conexiones del exterior están dados por las variables (inputs) y la conexión del axón es la conexión final del perceptrón, en el siguiente gráfico hace referencia al Out.

Figura 4.

Diagrama de un perceptrón - Redes Neuronales.



Fuente: Notas de clase fundamentos de deep learning (Ramos & Arias, 2020)

Finalmente, el algoritmo del perceptrón tiene la finalidad de darle pesos a las variables de entrada y la operación interna sería una suma sopesada de la combinación entre las entradas y los pesos asignados a cada una de las entradas.

Dentro de las arquitecturas de las redes neuronales se resaltan las siguientes:

Redes Neuronales Multicapas

Una de las limitaciones del perceptrón simple es que la separación entre patrones se da a través de un hiperplano, una de las formas de solventar esta limitación es la inclusión de capas ocultas en la arquitectura de la red que permita tener regiones de decisión de formas convexas. También se conoce con el nombre de red de retropropagación ya que se suele entrenar con un algoritmo de retropropagación de errores, es decir, los errores se propagan en sentido inverso a las entradas para actualizar los pesos de la red.

Redes Neuronales Recurrentes

Son un tipo de arquitectura de red neuronal utilizada para analizar datos de series temporales o secuenciales. Se caracteriza por recibir información de entrada sino que contempla la salida de la capa anterior para generar su propia salida, es decir, la red neuronal recurrente tiene tanto información del presente como del pasado reciente.

5.9 Búsqueda de Hiperparámetros [15]

La finalidad del uso de un modelo es predecir la variable respuesta, en nuestro caso, el precio de la vivienda de futuras observaciones, es decir, de casos que el modelo actualmente no ha observado o "visto" con anterioridad. Si sólo tomamos como referencia la métrica del conjunto de entrenamiento estaríamos haciendo una evaluación errónea de los resultados, ya que la métrica de entrenamiento suele ser más optimista. Uno de los pasos a realizar en la construcción del modelo es la búsqueda de hiperparámetros, siguiendo la idea inicial, se tiene como insumos diferentes estrategias de validación cruzada que permita ir evaluando el modelo con diferentes valores en sus hiperparámetros en un conjuntos de datos que no ha sido para entrenar.
Dentro de las estrategias de validación cruzada se puede encontrar las siguientes opciones:

[15] Validación de modelos predictivos (Amat 2020)

- *Validación Simple:* que implica dividir el conjunto en entrenamiento y prueba de forma aleatoria
- *Validación cruzada Leave One Out:* es un método iterativo que emplea como conjunto de entrenamiento todas las observaciones disponibles excepto una, que se excluye para emplearla como validación.
- *K - Fold:* consiste en dividir los datos de forma aleatoria en *k* grupos de aproximadamente el mismo tamaño, *k-1* grupos se emplean para entrenar el modelo y uno de los grupos se emplea como validación. Este proceso se repite *k* veces utilizando un grupo distinto como validación en cada iteración. El proceso genera *k* estimaciones del error cuyo promedio se emplea como estimación final.

6. Metodología

6.1 Descripción de los datos

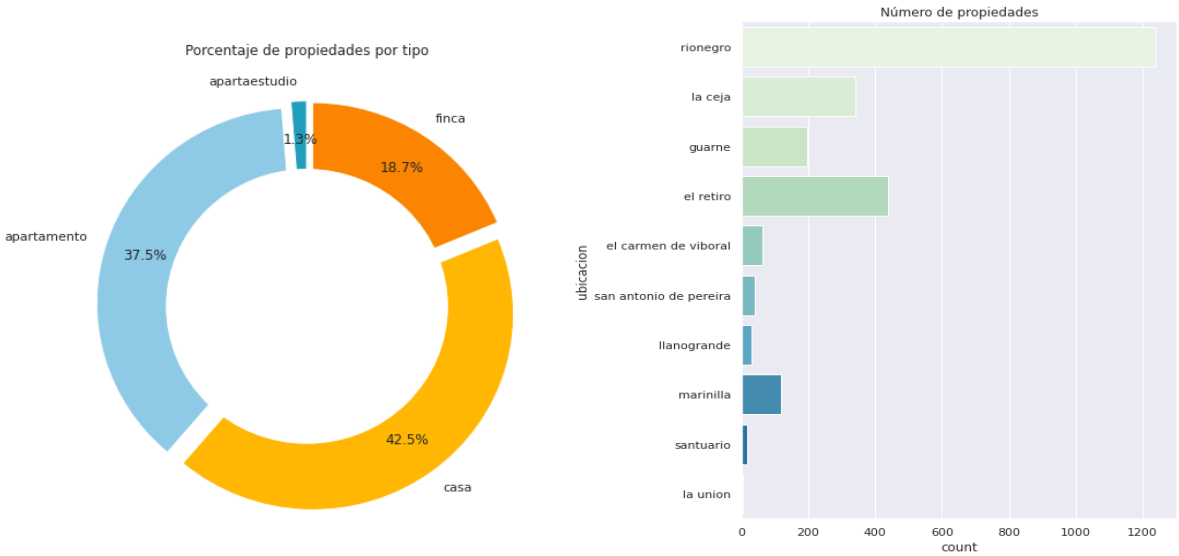
Para la obtención de la información se realizó un Web Scrapping al portal web de Finca Raiz (fincaraiz.com.co), un portal dedicado al mercado de bienes raíces para la venta y arriendo en todo el territorio de Colombia, donde se encuentran anuncios clasificados de constructoras, inmobiliarias o particulares con una gran acogida.

A partir del web scrapping se consolidó una base con información de 2.485 viviendas nuevas o usadas, entre apartamentos, casas y casas fincas. La mayoría son viviendas usadas y una gran proporción está ubicada en Rionegro. Es de anotar que del municipio de San Vicente de Ferrer no se obtuvo ningún registro. Debido a que el portal puede añadir o remover anuncios con cierta frecuencia, se optó por descargar los datos en un archivo en formato de texto plano (csv) con un tamaño aproximado de 3 MB, el cual posteriormente fue subido a un repositorio de github público, desde el cual se realiza su descarga para ser procesado en los notebooks.

En el primer gráfico de la **Figura 5** se puede observar la distribución de inmuebles por cada tipo de propiedad y cantidad de inmuebles en cada municipio.

Figura 5.

Distribución de los datos por tipo de propiedad y municipio.



De la información que se recolectó de cada propiedad se resaltan las siguientes variables:

Características principales:

- Tipo de vivienda: usada o nueva
- Precio en pesos colombianos (Variable independiente)
- Área construida en metros cuadrados
- Número de habitaciones
- Número de baños
- Número de parqueaderos
- Antigüedad del inmueble
- Municipio
- Tipo de propiedad : casa, apartamento o casa finca
- Descripción del inmueble en venta
- Estrato socioeconómico del sector

Características propias de la vivienda como:

- Si tiene balcón, chimenea, cuarto de servicio, cocina dotada, jardín, parqueadero de visitantes, despensa, calentador, si pertenece al área urbana o al área rural. Es de anotar que no todas las propiedades especifican todas las características, por lo tanto, solo se marcó como 1 las que si las mencionan.

Características complementarias como:

- Si la propiedad está ubicada cerca de universidades o colegios, salón, piscina, canchas de fútbol, cuarto de servicio, zona bbq, acceso pavimentado, tanques de agua, vivienda bifamiliar, finca ganadera, puerta eléctrica, entre otras. Es de anotar que no todas las propiedades especifican todas las características, por lo tanto, solo se marcó como 1 las que si las mencionan.

Información de identificación:

- url
- título

Permitiendo consolidar una base final con 184 columnas. Donde 3 de ellas contienen información general de las características internas y externas de las propiedades, que posteriormente fueron codificadas usando la estratégia one-hot-encoding. Las variables url, título y descripción fueron removidas al hacer el ajuste de los modelos. Así mismo, en una de las iteraciones se implementó el modelo back of words con el objetivo de extraer información adicional a partir de la descripción de la propiedad.

El proceso de extracción de la información puede encontrarse en el Notebook: 01_02_WEB_SCRAPING_UPT_12082021, y el detalle de las variables usadas y su descripción se encuentra en la **Tabla 10** del anexo del documento**.**

Figura 6.

Descarga y lectura de datos usando pandas en Google Colaboratory

```
#Descargar datasets desde github
!git clone https://github.com/andres-soto-h/monografia-udea-eacd.git
```

```
#Lectura del dataset transformado
df_propiedades=pd.read_csv('/content/monografia-udea-eacd/df_prop_clean_12082021.csv', delimiter=';', encoding='latin1')
```

Una vez se realizó la obtención de los datos a partir de web scraping, se realizaron algunos ajustes para estandarizar el nombre de las variables y limpiar caracteres especiales. A partir de este archivo se realiza una lectura usando la librería pandas de python, para transformar el archivo en un dataframe, y poder realizar los respectivos análisis descriptivos y modelamiento.

Otro aspecto importante a mencionar, es que se realizó la partición de los datos de manera estratificada, tratando de tener la mejor distribución entre la información de entrenamiento y prueba, para esto se realizaron varias iteraciones, realizando estratificación por municipio, por estrato socioeconómico, ajustando modelos sin estratificación e incluso realizando un modelo para cada uno de los municipios por separado, en el apartado 5.3 del documento se tratará más al respecto de este tema.

6.2 Análisis descriptivo de los datos

Como se mencionó inicialmente, el dataset de precios de casas está compuesto por 2485 observaciones las cuales se encuentran distribuidas en distintos municipios y tipos de propiedades. Al analizar la variable de precio podemos observar que el precio promedio de las propiedades extraídas de finca raíz, corresponde a 720,7 millones de pesos, y el 75% de las viviendas / apartamentos cuestan menos de 930 millones de pesos. También es de destacar que la propiedad más barata tiene un valor de 95 millones de pesos y corresponde a un aparta estudio ubicado en el municipio de Guarne. Así mismo existen 20 propiedades en el dataset con un precio que ronda los 2500 millones de pesos, ubicadas principalmente en Rionegro y Llanogrande, de éstas 11 son fincas y 9 son casas con áreas entre 180 y 16.800 metros cuadrados.

Tabla 1.

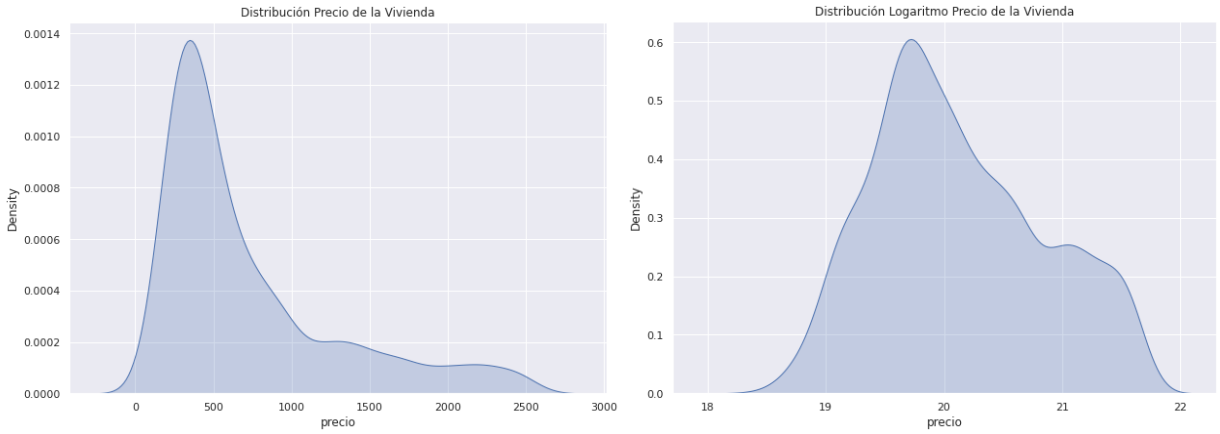
Datos estadísticos para algunas variables cuantitativas del dataset

		Precio	Area en m2	# habitaciones	# baños
Count	$	2.485	2.485	2.485	2.485
Mean	$	720.797.675	7.162	3	3
Std	$	573.546.484	212.576	1	1
min	$	95.000.000	3	1	1
25%	$	320.000.000	75	3	2
50%	$	490.000.000	146	3	3
75%	$	930.000.000	349	4	4
max	$	2.500.000.000	10.000.000	13	12

Respecto al área de las propiedades, en promedio se tiene un valor de 7162 metros cuadrados, lo cual se explica principalmente por las fincas y propiedades con grandes extensiones, aún así el 75% de los datos cuenta con menos de 350 metros cuadrados[16], entre los datos destaca una propiedad ubicada en Llanogrande con un área de 1000 hectáreas y un precio de 1700 millones.

Figura 7.

A la izquierda distribución de la variable precio. A la derecha log(precio)

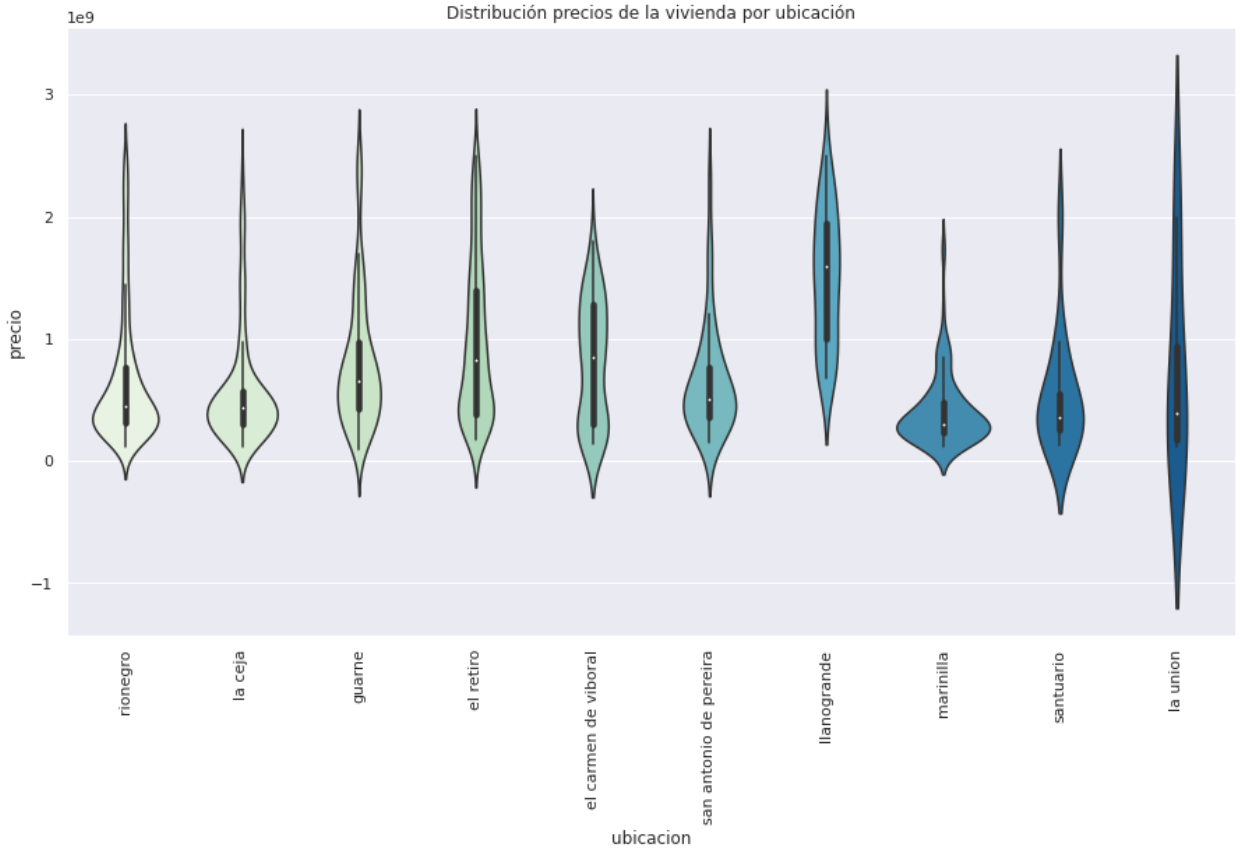


Como se aprecia en la **Figura 7**, el precio de las propiedades tiene una distribución sesgada a la derecha, debido a que la mayoría de los anuncios del portal finca raíz para estas ubicaciones se encuentran en un promedio de 720,8 millones, pero también se tienen otras propiedades como fincas, con características diferenciales, o ubicadas en zonas exclusivas dónde el precio por metro cuadrado es mucho más elevado. Debido a que se observa una distribución sesgada en la variable de precio, se toma la decisión de realizar una transformación sobre ésta usando el logaritmo, con esta puede verse un cambio en la distribución de los datos, donde se aprecia que quedan más centrados respecto a la media.

[16] Para una distribución no simétrica, la media no siempre es el mejor indicador para evaluar el comportamiento.

Figura 8.

Gráfico de violín para la variable precio a nivel de municipio



En la **Figura 8** se aprecia la distribución del precio para los distintos municipios analizados, de esta podemos observar como la distribución de los precios entre Rionegro, La ceja y San Antonio de Pereira se comportan más o menos de manera similar. En el caso del municipio de El Retiro se observa que su distribución está sesgada a la derecha, aunque también hay un número importante de observaciones con precios superiores a la media. Finalmente, para el municipio de La Unión dónde solo se cuenta con 4 observaciones, puede verse una distribución mucho más aplanada, pero un número de observaciones tan bajo, no resulta significativo para comprender la dinámica de precios de esta ubicación.

En la siguiente tabla (**Tabla 2**) podemos ver un comparativo de los precios por ubicación y tipo de propiedad, del cual se tiene que la propiedad con el precio promedio más bajo es el apartaestudio nombrado previamente; así mismo las fincas son las propiedades que presentan el precio promedio más alto. Comparadas casas y apartamentos, las casas también presentan un

precio promedio superior, aunque es de destacar que en El Retiro, el precio promedio de las casas es de 1.309 Millones, que junto con Llanogrande, tiene uno de los precios más altos para este tipo de propiedad. En tanto a los apartamentos, las propiedades con precio promedio más bajo se encuentran ubicadas en los municipios de Santuario, Carmén de Viboral y La Ceja.

Tabla 2.

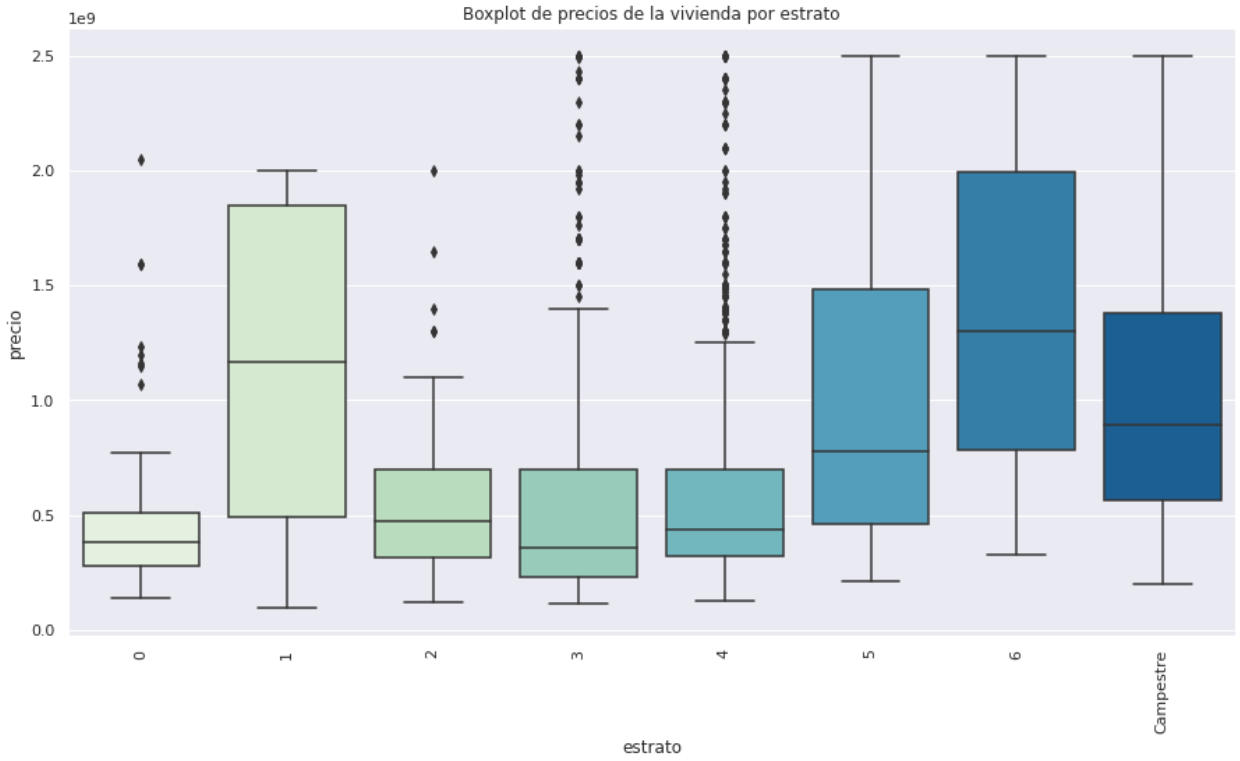
Precio promedio según el tipo de propiedad y ubicación – cifras en millones de pesos

Tipo_propiedad	Ubicacion	Precio promedio
apartaestudio	el retiro	$ 325
	guarne	$ 132
	la ceja	$ 205
	marinilla	$ 192
	rionegro	$ 320
apartamento	el carmen de viboral	$ 212
	el retiro	$ 437
	guarne	$ 250
	la ceja	$ 244
	la union	$ 120
	llanogrande	$ 1.007
	marinilla	$ 253
	rionegro	$ 342
	san antonio de pereira	$ 339
	santuario	$ 130
Casa	el carmen de viboral	$ 772
	el retiro	$ 1.309
	guarne	$ 835
	la ceja	$ 601
	llanogrande	$ 1.723
	marinilla	$ 568
	rionegro	$ 880
	san antonio de pereira	$ 779
finca	el carmen de viboral	$ 1.035
	el retiro	$ 1.402
	guarne	$ 878
	la ceja	$ 1.388
	la union	$ 928
	llanogrande	$ 1.708
	marinilla	$ 599
	rionegro	$ 1.245
	san antonio de pereira	$ 1.340
	santuario	$ 557

En la **Figura 9** podemos observar el comportamiento de la variable precio de acuerdo al estrato de la propiedad. Según esta información, es claro que a mayor estrato socioeconómico, mayor es el precio de la vivienda. En total se tienen 6 propiedades en estrato 1 de las cuales 5 corresponden a fincas con extensiones de más de 1500 metros cuadrados, para las demás propiedades su valor se encuentra por encima de los 300 millones.

Figura 9.

Boxplot de los precios de vivienda según el estrato socioeconómico



De la gráfica anterior también es importante resaltar que para los estratos 3 y 4 existe una cantidad significativa de valores que podrían considerarse atípicos, para el caso del estrato 3, el precio promedio se ubica en 546 millones, sin embargo de las 584 propiedades ubicadas en este estrato, hay 139 que tienen un precio superior a 700 millones hasta un máximo de 2500, que es el mayor valor registrado en el dataset. Para el caso del estrato 4 ocurre algo similar, el precio promedio de las viviendas se encuentra en torno a los 600 millones de pesos, pero de las 947 propiedades ubicadas en el estrato, hay 232 con un valor superior a los 700 millones de pesos y con un precio promedio de 1310 millones.

Dicho esto, es importante mencionar que, dentro de las iteraciones realizadas en la búsqueda del mejor modelo para el pronóstico de precios, se contempló la posibilidad de usar la variable estrato, para estratificar los datos y tener una mejor distribución entre los conjuntos de train y test.

6.3 Diseño

6.3.1 Modelos

Tomando como referencia trabajos similares que se han realizado en la predicción del precio de una vivienda y de los algoritmos estudiados en la especialización, se contemplan los siguientes modelos para predecir el precio de una vivienda en el Valle de San Nicolás con los datos obtenidos a partir proceso de web scraping.

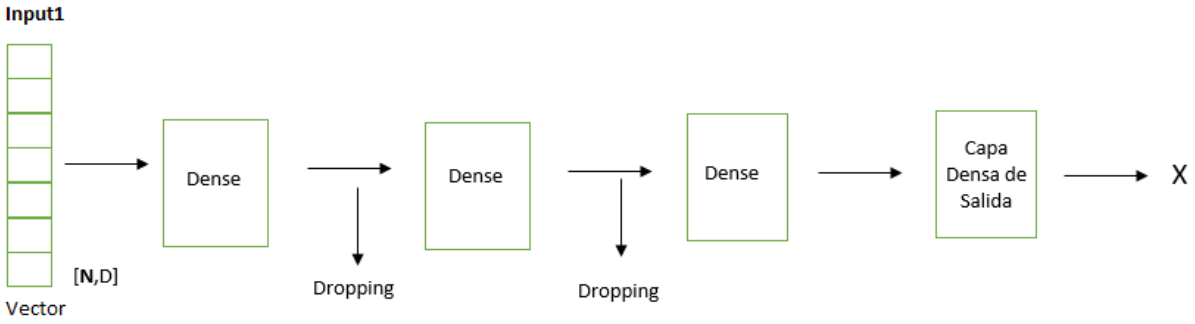
Los modelos son:

1. Random Forest
2. Gradient Boosting
3. Extrem Gradient Boosting
4. Neuronal Network

Para el caso de la red neuronal, se contemplaron dos experimentos con el conjunto de datos. La primera arquitectura de red toma información de las características del inmueble tanto internas como externas, para este caso, se consideró una arquitectura multicapa con 3 capas densas (sin contar la capa de salida) y donde la cantidad de neuronas se determina en el proceso de búsqueda de hiperparámetros. En la **Figura 10** se encuentra el esquema de la primera arquitectura contemplada.

Figura 10.

Esquema de arquitectura multicapa de la red neuronal contemplada en el primer experimento.

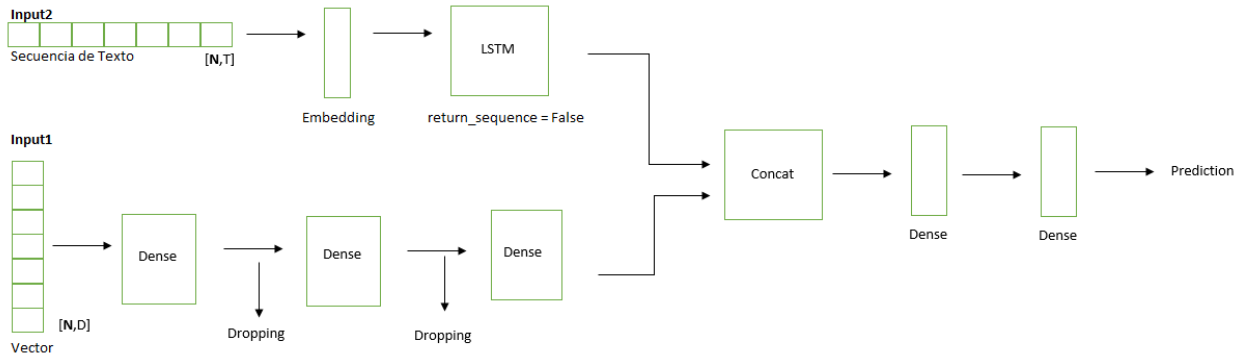


La segunda arquitectura de red contemplada implica la inclusión de dos entradas diferentes. La primera entrada contempla como información las variables que describen las características de la vivienda, la segunda entrada contempla una secuencia de palabras donde se selecciona las 500 palabras de mayor frecuencia extraídas del análisis de texto realizado a la columna *Descripcion* que contiene información que proporciona el vendedor para la venta del inmueble.

La **Figura 11** describe la arquitectura contemplada en el proceso de modelación.

Figura 11.

Esquema de arquitectura multimodal para una arquitectura de redes neuronales.



De la **Figura 11**, se resalta que la arquitectura de la red neuronal contempla dos entradas, como primera entrada se tiene un vector con las características más representativas del inmueble en el que se contempla una red secuencial con varias capas densas ocultas; la segunda entrada es una secuencia de texto (procesada a través de la técnica de word embedding) y contempla una

estructura de red recurrente (LSTM: Long Short Term Memory), las salidas de ambas estructuras se unen con una capa de concatenación, donde posteriormente se aplican dos capas densas adicionales hasta obtener la predicción final.

6.3.2 Métricas [17] [18] [19] [20]

Tomando como base la referenciación de trabajos similares que se han realizado en la predicción del precio de una vivienda, se consideran 4 métricas para la selección y evaluación de los modelos:

Tabla 3.

Métricas usadas para la evaluación de modelos.

Métrica	Fórmula	Descripción
Mediana del Error Absoluto (MedAE)	$median\left(\sum\lvert y_i - \hat{y}_i\rvert\right)$	La pérdida se calcula tomando la mediana de todas las diferencias absolutas entre el objetivo y la predicción. El error absoluto mediano es particularmente interesante porque es robusto a valores atípicos.
Error Porcentual Absoluto Medio (MAPE)	$\frac{1}{N}\sum_{i=1}^{N}\left\lvert\frac{y_i - \hat{y}_i}{y_i}\right\rvert$	Es un indicador del desempeño del Pronóstico de Demanda que mide el tamaño del error (absoluto) en términos porcentuales.
Coeficiente de Determinación R2	$R^2 = \frac{\sum(\hat{Y}_i - \bar{Y})^2}{\sum(Y_i - \bar{Y})^2}$	El coeficiente de determinación es la proporción de la varianza total de la variable explicada por la regresión. El coeficiente de determinación, también llamado R cuadrado, refleja la bondad del ajuste de un modelo a la variable que pretender explicar.
RMSE	$\sqrt{\frac{\sum(\hat{Y}_i - Y_i)^2}{N}}$	El error cuadrático medio (RMSE) mide la cantidad de error que hay entre dos conjuntos de datos. En otras palabras, compara un valor predicho y un valor observado o conocido

[17] Median absolute error (Bonnin 2017)

[18] Error Porcentual Absoluto Medio (MAPE) en un Pronóstico de Demanda (Gestiondeoperaciones.net 2015)

[19] ¿Qué es el error cuadrático medio RMSE? (Gabri 2018)

[20] Coeficiente de determinación (López 2017)

6.4 Etapa de Modelación

Para iniciar con el proceso de modelación del precio de la vivienda se definen unas pautas o procedimientos para transformar y limpiar la información que se utilizará dentro del esquema de modelación. En donde se resaltan las siguientes decisiones:

a. Se elimina información de las viviendas de La Unión ya que este municipio sólo cuenta con 4 datos, por lo tanto, no hay información suficiente para generalizar los resultados a toda la ubicación.
b. Se establece el esquema de validación del modelo, en donde se define como métrica principal (elección de los hiperparámetros y mejor modelo) el Median Absolute Error (MedAE). La intención de usar esta métrica en la elección del modelo es para compensar el efecto del sesgo a la derecha y cola pesada que se observa en la distribución del precio de las viviendas.
c. El esquema de validación utilizado para todo el proceso de selección de la metodología de modelación, búsqueda de hiperparámetros y evaluación de las métricas se compone de una división del conjunto de datos en 3 secciones (entrenamiento, validación y prueba).

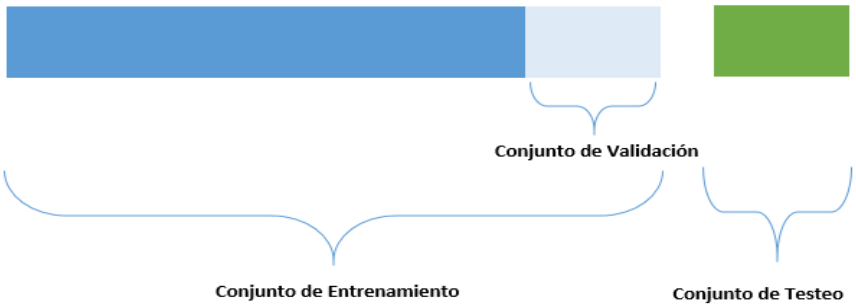
- **Entrenamiento:** representa el 90% del total de los datos y se utiliza para realizar el ajuste del modelo.

- **Validación:** representa el 10% del total de los datos de entrenamiento y se utiliza para realizar la validación cruzada para la selección de los hiperparámetros para cada uno de los modelos.

- **Prueba:** representa el 10% del total de los datos y se utiliza para validar el ajuste de los modelos.

Figura 12.

Partición de datos para la estrategia de validación cruzada.



d. Se calculan adicionalmente, las métricas del MAPE, R2, RMSE. El MAPE y R2 permiten comparar los resultados del ajuste con los resultados de la referenciación en trabajos similares donde el R2 es de alrededor del 70%, adicionalmente, son métricas de más fácil interpretación del error.
e. Se realizan iteraciones en el ajuste y búsqueda de hiperparámetros variando la división del conjunto de datos estratificando la división por alguna variable de interés.
f. Para cada iteración de modelación se realiza el ajuste de los siguientes modelos: random forest, gradient boosting y xgboost.
g. Se realiza el proceso de búsqueda de hiperparámetros. Para la búsqueda, se utilizó una combinación entre la estrategia de K- Folds, con K = 5 y boostraping, es decir, se selecciona diferentes conjuntos de datos conservando una proporción de 90-10 y se realiza la división en 5 grupos, contemplando como mallas de hiperparámetros los siguientes valores:

Random Forest y Gradiant Boosting	**XGBoost**
n_estimator: [100, 120, 150]	**max_depth**: [6, 7, 8, 9]
max_depth: [6, 7, 8, 9]	**learning_rate:** [0.05, 0.1]
min_sample_split: [4, 5, 6]	**reg_alpha:** [0.01, 0.1, 0.5]
min_sample_leaf: [2, 3, 4]	

A continuación, se resaltan los diferentes experimentos realizados en el proceso de ajuste de los modelos, los objetivos que se quieren alcanzar son:

- Considerar metodologías de modelación más robustas
- Realizar ingeniería de variables
- Mejorar el desempeño del modelo
- Referenciación de otros trabajos similares

Tabla 4.

Descripción de las iteraciones realizadas.

Iteración	Descripción
1	Se realiza la división del conjunto de datos en entrenamiento y prueba, estratificado por la variable ubicación. En esta iteración se utiliza la métrica del MAPE para seleccionar los hiperparámetros de los diferentes algoritmos seleccionados y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
2	Se realiza la división del conjunto de datos en entrenamiento y prueba, estratificado por la variable ubicación. En esta iteración se utiliza la métrica del ***Median Absolute Error (MedAE)*** para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT, XGBoost. Se cambia la métrica MAPE, ya que la distribución del precio de las viviendas contempladas presenta sesgo a la derecha y cola pesada, por lo que se considera una métrica robusta a la presencia de datos atípicos. Adicionalmente, se calcularon las métricas de RMSE, MAPE, R2 para la validación final de su ajuste. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).

3	Se divide el conjunto de datos en estratificado por la variable estrato, esta idea sale de la revisión que se realizó del estado del arte de otros trabajos similares al que estamos realizando. En esta iteración se utiliza la métrica del Median Absolute Error para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT. Adicionalmente, se calculan las métricas de RMSE, MAPE, R2. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
4	En este experimento se realiza la construcción de un modelo individual para cada ubicación del valle de San Nicolás contemplada, por la cantidad de datos en cada lugar de interés, se unifican teniendo presente en la agrupación aquellas localidades que presentan un comportamiento similar en su distribución. Formándose 5 grupos así: Grupo 1: El Carmen de Viboral y Guarne Grupo 2: El Retiro y Llanogrande Grupo 3: La Ceja y San Antonio de Pereira Grupo 4: Marinilla y El Santuario Grupo 5: Rionegro La métrica utilizada es MedAE para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT, XG Boost. Adicionalmente, se calculan las métricas de RMSE, MAPE, R2. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).

5	En esta iteración se realiza la construcción de un modelo individual para cada estrato, por la cantidad de datos en cada una de las categorías de la variable de interés, se unifican algunos niveles. Formando 4 Grupo 1: Estrato 0, 1 y 2 Grupo 2: Estrato3 Grupo 3: Estrato4 Grupo 4: Estrato 5 y 6 En esta iteración se usa la métrica de MedAE para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT. Adicionalmente, se calculan las métricas de RMSE, MAPE, R2. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
6	Se divide el conjunto de datos estratificado por la variable ubicación. Se adiciona como variables explicativas, las 500 palabras más frecuentes del resultado del ejercicio de minería de texto realizado a la columna *descripción* del conjunto de datos. En este experimento se utiliza la métrica del MedAE para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. En esta iteración se contempló el algoritmo GBT y XGBoost. Adicionalmente, se calculan las métricas de RMSE, MAPE, R2. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
7	Se realiza el ajuste de la variable precio estratificado por ubicación. Cómo variables regresoras sólo se utilizan las 500 palabras más frecuentes del resultado del ejercicio de minería de texto usando unigramas. En esta iteración se utiliza la métrica del MedAE para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. En esta iteración se contempló el algoritmo GBT y XGBoost. Adicionalmente, se calculan las métricas de RMSE, MAPE, R2. En esta iteración se quería evaluar cuánto aportan las variables del texto en la explicación de la variable precio, sin contemplar las otras variables características del inmueble.

	Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
8	Se realiza la división del conjunto de datos en entrenamiento y prueba, estratificado por la variable ubicación. En este experimento se utiliza la métrica del MedAE para seleccionar los hiperparámetros de los diferentes algoritmos a ajustar y para validar el ajuste del modelo en un nuevo conjunto de datos. Los algoritmos contemplados son: Random Forest, GBT y XGBoost. En esta iteración sólo se considera como variables regresoras las básicas: tipo (usada o nueva), area (en mtrs2), baños, garajes, antigüedad vivienda, latitud, longitud, estrato, ubicación, tipo_propiedad (casa, apartamento, finca o apartaestudio), balcon (si o no), zonas verdes (si o no), conjunto cerrado (si o no), zona infantil (si o no), supermercados o comercios cerca (si o no), colegios o universidades cerca (si o no) y transporte público cerca (si o no). También pensando en la facilidad en el despliegue del modelo y de la recolección de información posteriormente. Las variables contempladas en este experimento presentan una importancia mayor a cero. Un segundo experimento se realizó añadiendo las variables de latitud y longitud al conjunto de datos para modelar. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).
9	Contemplando la misma métrica de ajuste y realizando una partición de los datos en entrenamiento y prueba estratificado por la variable ubicación, se realiza el ajuste de los modelos: Random Forest, GBT y XGBoost considerando las siguientes variables: Se considera como variables regresoras las básicas: tipo (usada o nueva), area (en mtrs2), baños, garajes, antigüedad vivienda, latitud, longitud, estrato, ubicación, tipo_propiedad (casa, apartamento, finca o apartaestudio), balcon (si o no), zonas verdes (si o no), conjunto cerrado (si o no), zona infantil (si o no), supermercados o comercios cerca (si o no), colegios o universidades cerca (si o no) y transporte público cerca (si o no) y las primeras 500 variables de texto más importante tomado del análisis de texto de la variable Descripción. Entre las iteraciones se realiza el ejercicio usando la variable precio sin ninguna transformación y otra iteración con la misma familia de algoritmo usando como transformación para y = log1p(precio).

10 En este último experimento, se contemplaron dos estructuras de modelos: una red multicapa, donde la entrada es un vector con las características internas y externas del inmueble (variables contempladas en el experimento 8 y 9) y una red neuronal con 2 entradas. En esta segunda arquitectura, la primera entrada es un vector con las características propias del inmueble (sólo se utiliza las variables del experimento 8 y 9) y la segunda entrada corresponde a las 500 palabras de mayor frecuencia de la columna *Descripción.* Las arquitecturas de las redes se explicaron en el literal C. Diseño.

En estas iteraciones, se tomó como medida para realizar el ajuste y búsqueda de hiperparámetros el MAPE.

En el proceso de búsqueda de hiperparámetros, el hiperparámetro iterado fue el número de neuronas en cada capa densa de la arquitectura. En la primera estructura (primera entrada) se tiene 3 capas densas y en la estructura de concatenación se tiene otra capa densa antes de la capa densa final.

Para el caso de la segunda entrada, en la etapa del Embedding se tiene que la longitud máxima de las frases es de 127 palabras, se toma la decisión de considerar una longitud máxima de 80 palabras.

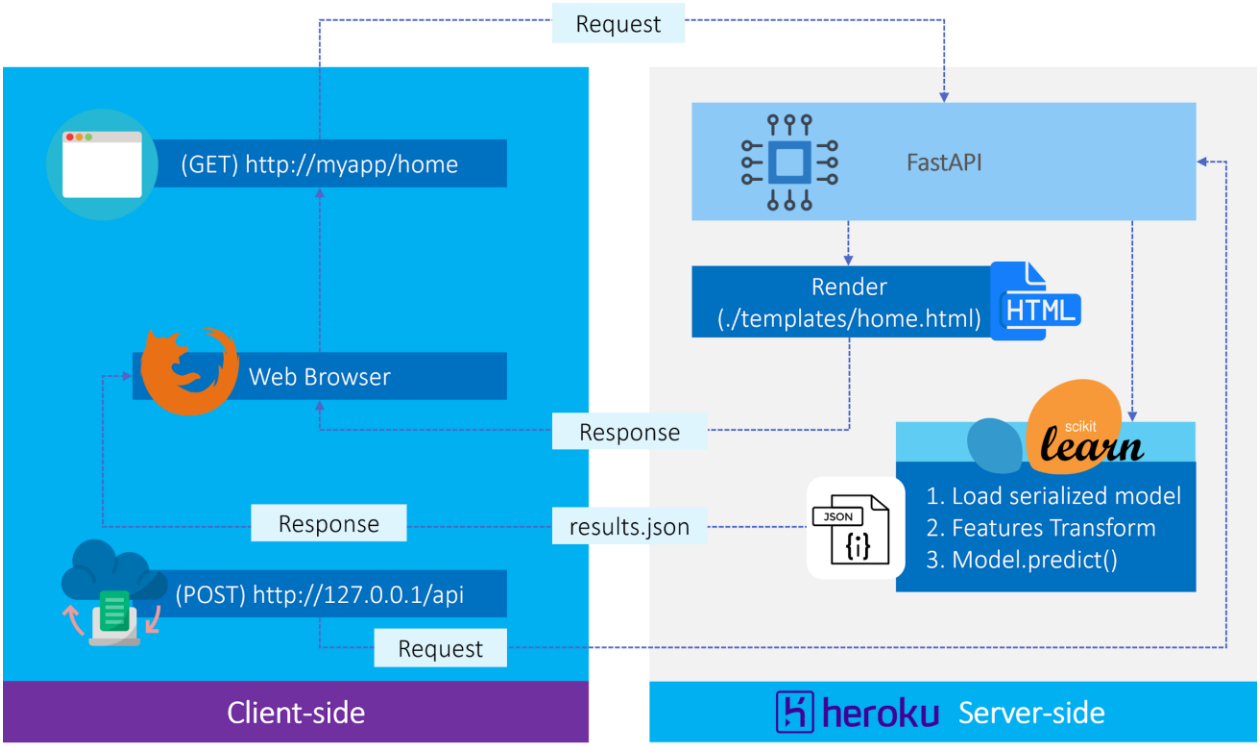
La variable del precio se utiliza en este experimento sin ninguna transformación.

6.5 Despliegue del modelo

Para realizar la implementación del modelo, se realizará la creación de una aplicación web, a través de esta será posible ingresar las principales características de la propiedad en la interfaz gráfica de usuario, posteriormente los datos serán enviados al servidor, dónde el mejor modelo seleccionado será cargado y recibirá los datos ingresados por el usuario. Finalmente se presentarán los resultados de la estimación en pantalla.

Figura 13.

Arquitectura conceptual de la solución.



A la derecha del esquema se presentan los elementos que intervienen en el lado del cliente, el usuario de la aplicación ingresa, llena los datos y hace el envío a través de una petición de tipo POST, estos datos llegan al servidor (lado derecho de la Figura) y son procesados por FastAPI, dónde se invoca al modelo pre entrenado para realizar la predicción.

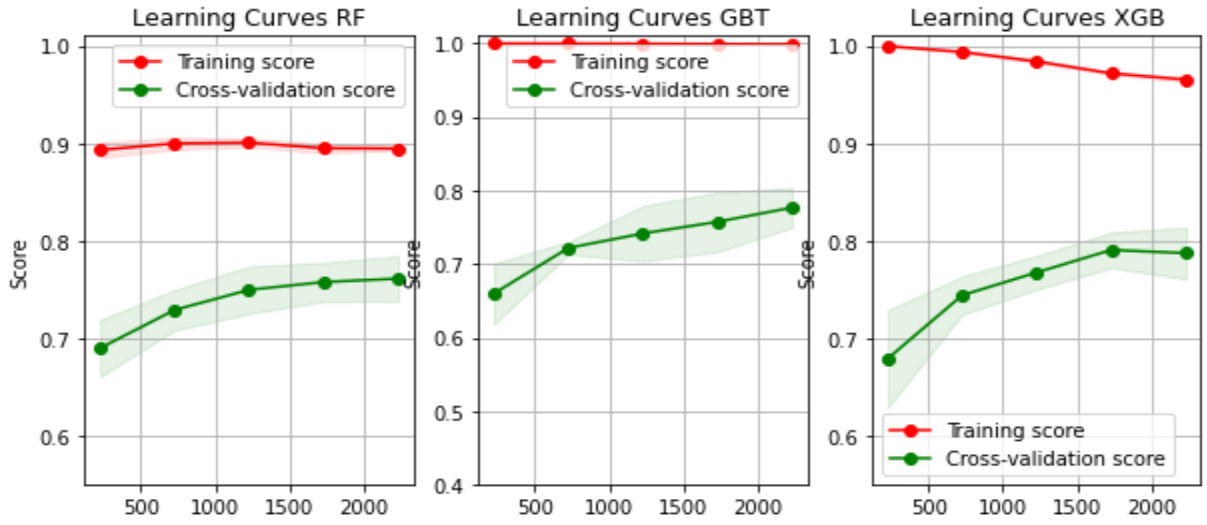
7. RESULTADOS

7.1 Curva de Aprendizaje

Dentro del diseño de la validación del modelo, se inició con una partición inicial de un 80% para entrenamiento y un 20% para test. Del ajuste realizado en la iteración 1 y 2, se realizó la curva de aprendizaje para cada una de las metodologías empleadas, usando como modelo aquellos con mejor parámetros. Los resultados obtenidos para la iteración de tomar datos estratificados por ubicación se presentan en la **Figura 14**.

Figura 14.

Curvas de aprendizaje para 3 de los Modelos.



En donde se observa que todavía hay posibilidad de mejorar las métricas de entrenamiento y test si me presenta más información, de manera que las métricas se acercaran más entre sí. Por el comportamiento que se evidencia en estas gráficas se toma la decisión de pasar el esquema de validación de una división 80 - 20 a 90 - 10, con el objetivo de tener más información en el entrenamiento, dado que la posibilidad de obtener mayor información a través de web scraping es limitada.

7.2 Procesamiento de Texto

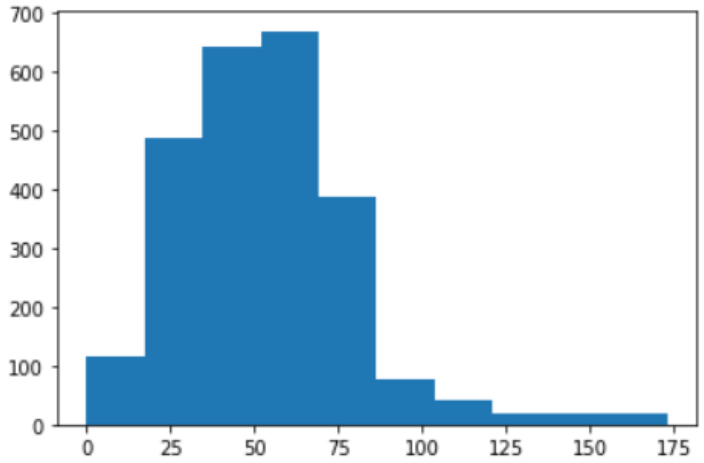
En el ejercicio de web scraping una de las columnas que se obtuvo fue la descripción que el vendedor realizaba para la venta del inmueble. Se tomó esta información y se realizó un procesamiento de texto para extraer las palabras de mayor frecuencia para usar esta información posteriormente en los ajustes de los modelos.

Dentro del tratamiento de las palabras se eliminaron números, signos de puntuación, tildes y se convirtieron todas las palabras a minúsculas. Adicionalmente se eliminaron las stopwords en español que se descargaron desde un repositorio en github[21]. También se realizo un proceso de stemming con la librería nltk y se ajustaron algunas palabras de forma manual.

Posteriormente, se realizó un proceso de tokenización de las palabras y se construyó un count vectorizer que tiene el conteo de las palabras y, este resultado se usa adicionalmente para evaluar la longitud del texto. El resultado del análisis de la distribución de la longitud del texto, se utiliza en la construcción de la arquitectura de la red neuronal recurrente. En la **Figura 15**. Se aprecia la distribución de la longitud del texto de la columna *descripción.*

Figura 15.

Distribución longitud de las frases de la columna descripción ya procesada.



[21] https://raw.githubusercontent.com/Alir3z4/stop-words/master/spanish.txt

Se puede resaltar, que la longitud máxima es de aproximadamente 175 palabras y que la mayor concentración se presenta en el intervalo de 30 a 60 palabras.

7.3 Ajustes del Modelo Random Forest, GBT y XG Boost

En cada experimento realizado se hallaron los mejores parámetros basados en la metodología de validación cruzada, realizando en cada iteración un proceso de bagging y tomando los datos estratificados por la variable ubicación, luego del resultado de la búsqueda de hiperparámetros se realiza el ajuste final con los parámetros hallados y se evalúa los resultados de cada modelo comparando cuál presenta el mejor resultado.

En la **Tabla 5** se visualiza el resultado de la búsqueda de hiperparámetros para las iteraciones más representativas dentro del trabajo; se especifican los parámetros finales del mejor modelo tanto para el Random Forest como para las iteraciones con el GBT considerando la variable respuesta sin ninguna transformación y transformada.

Tabla 5.

Resultados de búsqueda hiperparámetros en el Random Forest y Gradient Boosting para los experimentos más relevantes.

Itr	Escenario	Modelo	max depth	min samples leaf	min samples split	n estimators
2	Estratificación por Ubicación	rf	9	4	4	150
2	Estratificación por Ubicación	rf, y_transf	9	2	5	120
2	Estratificación por Ubicación	gbt	9	3	4	150
2	Estratificación por Ubicación	gbt, y_trans	9	4	4	120
3	Estratificación por Estrato	rf	9	3	4	100
3	Estratificación por Estrato	rf, y_transf	9	2	4	150

3	Estratificación por Estrato	gbt	9	2	4	150
3	Estratificación por Estrato	gbt, y_trans	8	2	5	150
6	Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	rf	9	2	6	100
6	Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	rf, y_transf	9	2	4	100
6	Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	gbt	8	2	6	100
6	Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	gbt, y_trans	7	2	6	100
8	Estratificación por Ubicación, considerando variables básicas (no latitud, no longitud)	rf	9	3	4	150
8	Estratificación por Ubicación, considerando variables básicas (no latitud, no longitud)	rf, y_transf	9	2	5	100
8	Estratificación por Ubicación, considerando variables básicas (no latitud, no longitud)	gbt	7	2	4	100
8	Estratificación por Ubicación, considerando variables básicas (no	gbt, y_trans	9	3	4	150

	latitud, no longitud)					
9	Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	rf	9	2	4	150
9	Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	rf, y_transf	9	2	6	150
9	Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	gbt	9	2	4	100
9	Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	gbt, y_trans	8	3	4	150

De manera general, se observa que en la mayoría de los casos, el valor para la profundidad de los árboles es de 9 y la cantidad mínima para las divisiones es de 4 muestras.

En la **Tabla 6** se presentan los resultados de la búsqueda de hiperparámetros para el modelo de XGBoost. En el caso de este algoritmo, sólo se realizaron iteraciones contemplando la variable respuesta sin transformar.

Tabla 6.

Resultados de búsqueda hiperparámetros en el XGBoost para los experimentos más relevantes.

Itr	Escenario	n estimators	max depth	reg alpha
2	Estratificación por Ubicación	100	6	0.01
3	Estratificación por Estrato	100	5	0.01
6	Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	100	8	0.01
8	Estratificación por Ubicación, considerando variables básicas (no latitud, no longitud)	100	5	0.01
9	Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	100	8	0.01

En los resultados de la búsqueda de hiperparámetros del XGBoost, el número de estimadores es el predeterminado en la función, es decir, el número de árboles contemplados dentro de los mejores modelos es de 100 árboles con una tasa de aprendizaje de 0.1. De los resultados obtenidos, se observa que el valor del parámetro de regularización es igual para todos los experimentos y es de 0.01. La profundidad de los árboles varía según el experimento.

Conocidos los parámetros de los diferentes modelos, se comparan las métricas obtenidas del ajuste del modelo.

En la **Tabla 7**, se visualizan los resultados de la métrica del MedAE en cada una de las iteraciones realizadas, usando como variable respuesta el precio de la vivienda sin ninguna transformación. De las iteraciones realizadas, se observa que el error promedio está en $81.000.000, el experimento 8 presenta el menor error, en este caso, se construyó un GBT usando los datos estratificados por la variable ubicación y tomando información de las 17 variables principales. El problema en esta iteración es que la métrica de entrenamiento y prueba se encuentran distantes entre sí, por lo que hay presencia de sobreajuste.

Tabla 7.

Resumen de métricas obtenidas en 3 de los principales modelos - MedAE[22].

Median Absolute Error	modelo_rf		modelo_gbt		modelo_xgb	
Iteraciones	Train	Test	Train	Test	Train	Test
Iteración2 - Modelo Estratificado por Ubicación	53.483.616	73.400.178	7.774.991	74.172.436	43.208.272	85.208.512
Iteración 3 - Modelo Estratificado por Estrato	56.672.364	74.602.796	4.558.753	70.208.590	55.503.136	77.231.296
Iteración 4- Modelos por Ubicación	49.991.239	78.901.368	4.566.614	82.882.543	23.596.125	77.401.634
Iteración 5 - Modelos por Estratificación	29.170.987	79.996.425	9.870.335	94.432.750	293.291	191.375.335
Iteración 6 - Estratificación por Ubicación Imp > 0	52.765.731	70.089.281	12.860.288	75.182.943	43.208.272	85.208.512
Iteración 7.1 - Estratificación por Ubicación Sólo Variables de Texto unigrama	135.435.883	143.418.159	45.052.016	146.570.517	68.071.680	145.012.864
Iteración 7.2 - Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	130.923.619	149.068.845	35.351.596	142.113.036	48.176.160	133.618.240
Iteración 8.1 - Estratificación por Ubicación, considerando variables básicas	53.370.768	70.321.431	15.182.611	67.389.854	45.697.168	72.452.272
Iteración 8.2 - Estratificación por Ubicación, considerando variables básicas 2	59.264.527	73.215.843	36.762.557	73.080.386	59.268.584	74.319.680
Iteración 9 - Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	53.988.177	76.014.114	14.602.577	69.267.769	17.400.506	77.883.712

Tomando como métrica el MAPE, la iteración con menor métrica en test se presenta en la iteración 3, donde se realiza una estratificación de los datos por la variable Estrato y un ajuste con el gbt. Igualmente, se resalta que aunque las mejores métricas se tienen con el gbt, hay presencia de sobreajuste en los modelos ajustados. Este problema se logra mejorar en la iteración 8.1 donde se realiza un ajuste con las 17 variables básicas (incluyendo latitud y longitud) con el Xgboost.

[22] El modelo del experimento 8.1 incluye las variables de latitud y longitud, mientras que en el modelo 8.2 se excluyeron, esto se da porque para el despliegue no se tendrá información de latitud y longitud del inmueble.

Tabla 8.

Resumen de métricas obtenidas en tres de los principales modelos - MAPE.

MAPE	modelo_rf		modelo_gbt		modelo_xgb	
Iteración	Train	Test	Train	Test	Train	Test
Iteración 2 - Modelo Estratificado por Ubicación	16,30%	22,07%	2,74%	20,62%	11,88%	21,39%
Iteración 3 - Modelo Estratificado por Estrato	16,99%	22,96%	1,79%	19,99%	15,37%	21,32%
Iteración 4- Modelos por Ubicación	25,21%	26,18%	22,36%	26,60%	6,87%	23,72%
Iteración 5 - Modelos por Estratificación	10,59%	26,51%	3,38%	31,77%	6,41%	33,75%
Iteración 6 - Estratificación por Ubicación Imp > 0	15,69%	21,89%	4,03%	20,60%	11,88%	21,39%
Iteración 7.1 - Estratificación por Ubicación Sólo Variables de Texto unigrama	37,65%	36,78%	14,79%	35,53%	20,88%	34,60%
Iteración 7.2 - Estratificación por Ubicación Sólo Variables de Texto uni y bigrama	37,57%	39,06%	12,22%	35,35%	16,34%	34,37%
Iteración 8.1 - Estratificación por Ubicación, considerando variables básicas	15,88%	21,17%	4,70%	20,87%	12,44%	20,51%
Iteración 8.2 - Estratificación por Ubicación, considerando variables básicas 2	17,26%	21,51%	10,05%	21,58%	16,22%	21,05%
Iteración 9 - Estratificación por Ubicación, considerando variables básicas y de texto (uni y bigramas)	16,24%	22,48%	4,31%	20,51%	5,18%	21,27%

De las iteraciones realizadas, se tiene como modelo candidato el XGBoost de la iteración 8.1, donde se logra tener un MAPE del 20% para test y del 12% para los datos de entrenamiento.

Tomando como referencia la métrica del median absolute error, el modelo candidato es el Random Forest o XGBoost de la iteración 8.1 (este por simplicidad en las variables) ya que el random forest de la iteración 6 muestra una leve mejora en las métricas del RF.

7.4 Ajuste y Búsqueda de Hiperparámetros de las Redes Neuronales

El último experimento realizado con el conjunto de datos es la implementación de dos estructuras de redes neuronales, se desea observar si con esta última iteración las métricas presentan alguna mejora.

Como se ha mencionado en secciones anteriores, en este experimento se consideraron dos estructuras de red. La primera estructura contempla una única entrada con la matriz de características de la vivienda como son: Área, Estrato, antigüedad de la vivienda, número de baños, número de habitaciones, número de garajes, tipo de vivienda, ubicación, si es nueva o no, si tiene balcón, zona verdes, si es un conjunto cerrado, si tiene zona infantil, comercios o colegios cercanos; la segunda arquitectura de red contempla dos entradas, la primera consiste de las características de la vivienda, la segunda consiste en un texto procesado a través de word embedding y una red recurrente (LSTM), texto que contiene la información de las palabras más importantes extraídas de la columna *Descripción* del conjunto de datos.

Para estas iteraciones sólo se contempló en la búsqueda de hiperparámetros el número de neuronas de la estructura multicapa.

En la **Figura 16** se visualiza la construcción de la primera estructura de red en python, haciendo uso para su construcción, búsqueda y ajuste la librería keras de tensorflow.

Figura 16.

Estructura arquitectura red multicapa con vector de características como entrada

```
def build_model(n_units=100 ,n_layers=3, drop_rate=0.1, input_shape = 32):

  model = keras.Sequential()
  model.add(Input(shape=input_shape))

  neuron_list=[]
  neuron_list.append(n_units)

  for n in range(1, n_layers):
    neuron_list.append(neuron_list[n-1]/2)

  for n in neuron_list:
    model.add(Dense(units=n, activation = 'relu'))
    model.add(Dropout(drop_rate))

  model.add(Dense(1))

  optimizer = tf.keras.optimizers.Adam(learning_rate=0.001,
      beta_1=0.9,beta_2=0.999, epsilon=1e-07,amsgrad=False,
      name="Adam")

  model.compile(loss='mean_absolute_percentage_error',
                optimizer=optimizer,
                metrics=['mean_absolute_percentage_error'])

  return model
```

En la gráfica anterior, también se observa que para la construcción de la primera arquitectura de red, la función recibe como parámetros:

n_units: el número de neuronas de la primera capa densa

n_layers: número de capas densas de la arquitectura (no incluye la capa densa final)

drop_rate: parámetro para la capa de dropout de la arquitectura

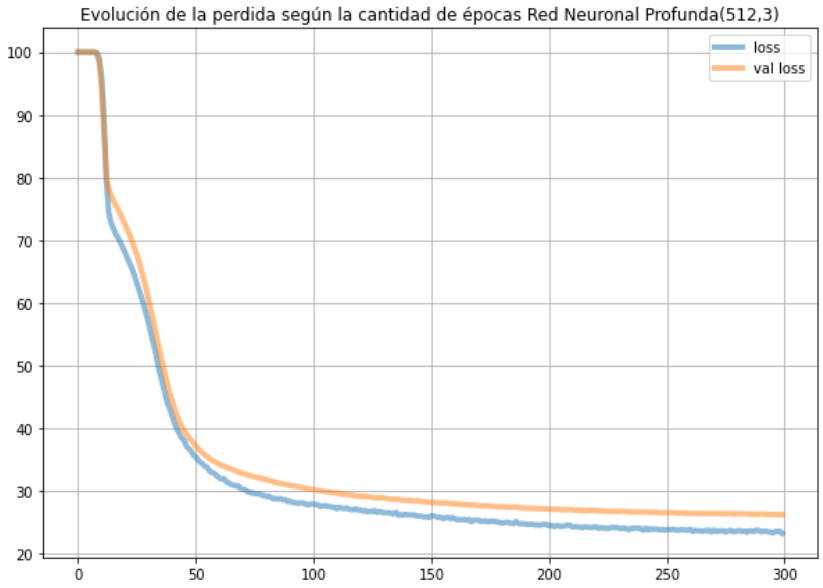
input_shape: especifica el tamaño de entrada de la red

La activación usada en cada una de las capas densas es una “relu”, el algoritmo de optimización utilizado es Adam y la métrica utilizada para la selección de hiperparámetros es el MAPE.

Se toma el resultado de la búsqueda de hiperparámetros, donde el número de neuronas elegidas son: (512,256,128) y se gráfica la evolución de la pérdida de entrenamiento y validación variando el número de épocas, su comportamiento se puede visualizar en la **Figura 17**.

Figura 17.

Evolución de la pérdida de la red neuronal multicapa con input las características de la vivienda.



De la **Figura 17** se puede observar que al incrementar el número de épocas la función de pérdida aún puede tomar menores valores sin afectar el resultado entre entrenamiento y prueba. Aunque ya se empieza a visualizar una estabilidad en la pérdida de ambos conjuntos de datos, todavía se observa la posibilidad de mejora.

El MAPE obtenido con 512 neuronas en la primera capa y 300 épocas para el entrenamiento del modelo es de 26.14% en el conjunto de validación y 26.16% en el conjunto de entrenamiento.

En la **Figura 18** se visualiza la construcción de la segunda arquitectura de red en python, haciendo uso para su construcción, búsqueda y ajuste la librería keras de Tensorflow.

Figura 18.

Arquitectura red multicapa y red recurrente.

```
def build_model2(param):

    inp1 = Input(shape=(X_train_scaled.shape[1],), name="Input_Feautures")

    neuron_list=[]
    neuron_list.append(param['unit_1'])

    for n in range(1, param['n_layers']):
      neuron_list.append(neuron_list[n-1]/2)

    x = Dense(param['unit_1'], activation="relu", name="dense1")(inp1)

    for idx, n in enumerate(neuron_list[1:]):

      x = Dense(units=n, activation = 'relu', name="Dense_"+str(idx))(x)
      x = Dropout(param['drop_rate'], name="Dropout"+str(idx))(x)

    inp2 = Input(shape=(NewX_train.shape[1],), name="Input_Text")
    cc1 = Embedding(input_dim=(tokenizer.num_words+1), output_dim=32, mask_zero=True)(inp2)
    cc2 = LSTM(param['cells_number'], return_sequences=False)(cc1)

    cc3 = tf.concat([x, cc2],axis=1)
    cc4 = Dense(10, activation='relu', name="Dense_concat")(cc3)
    cc5 = Dropout(param['drop_rate'], name="Dropout_concat")(cc4)
    output = Dense(1, name="Output")(cc5)

    model = Model(inputs=[inp1, inp2], outputs=output)

    optimizer = tf.keras.optimizers.Adam(learning_rate=param['lr'],beta_1=0.9,beta_2=0.999,epsilon=1e-07,amsgrad=False,name="Adam")

    model.compile(loss='mean_absolute_percentage_error',optimizer=optimizer, metrics=['mean_absolute_percentage_error'])
```

En comparación con la estructura anterior, se adiciona la estructura de la red recurrente y el proceso de concatenación de ambos resultados. La búsqueda de hiperparámetros sólo se realiza para la primera estructura, ya que el número de neuronas contempladas en la capa densa después del proceso de concatenación es de 10 neuronas.

El resultado de la cantidad de neuronas a considerar en la búsqueda de hiperparámetros es de 512 neuronas en la primera capa y por cada capa adicional se considera n/2 de neuronas, los parámetros del mejor modelo se presentan en la **Tabla 9**.

Tabla 9.

Parámetros del mejor modelo de la segunda arquitectura de la red neuronal

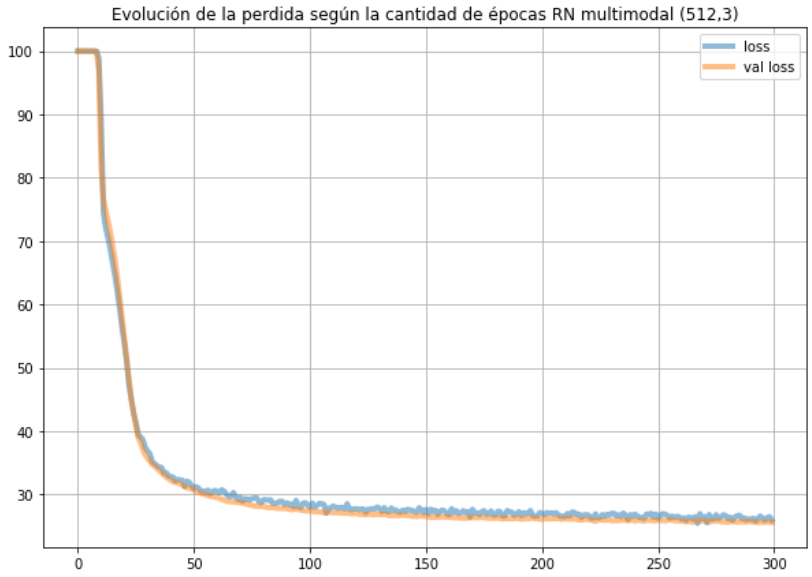
Parámetro	Valor	Parámetro	Valor
batch_size	32	lr	0.001
cells_number	10	n_layers	3
drop_rate	0.1	steps_per_epoch	70
epochs	100	unit_1	512

El MAPE obtenido con 512 neuronas en la primera capa y 100 épocas es de 27,25% .

Luego de seleccionar la cantidad de neuronas a través de la búsqueda de hiperparámetros, se ajusta el mejor modelo y se evalúa para el modelo seleccionado la evolución de la pérdida al variar el número de épocas, su comportamiento se puede visualizar en la **Figura 19**.

Figura 19.

Evolución de la pérdida arquitectura solo características.



En la **Figura 19** se aprecia la mejora en la función de pérdida a medida que el número de épocas incrementa. Un valor de épocas bueno puede ser de 200 o más épocas, debido a que el modelo todavía tiene alternativa de seguir aprendiendo.

7.5 Puesta en producción de modelos

Como se había mencionado en la sección de metodología, se realizó un desarrollo para desplegar el modelo en internet, y poder obtener resultados de precios a partir de una serie de atributos que deben ser ingresados a través de un formulario.

Para la publicación de la aplicación se utilizó el servicio de nube gratuito heroku.com, y a través del servidor web uvicorn y la librería FastApi, se hicieron disponibles 2 modelos para la

predicción de precios. En ambos casos el modelo usado es un XGBoost con una profundidad máxima de 5 y un alfa de 0.01, la diferencia radica en las variables usadas a la hora de entrenar y guardar los resultados.

Los parámetros finales de cada modelo se guardaron usando la librería joblib de sklearn y posteriormente se cargaron a la aplicación web[23] .

Lista de variables usadas en el modelo 1:

tipo, area_m2, banos,garajes, antiguedad, estrato, ubicacion, tipo_propiedad, balcon, zonas_verdes, en_conjunto_cerrado, zona_infantil, supermercados_ccomerciales, colegios_universidades, trans_publico_cercano, habitaciones

Lista de variables usadas en el modelo 2:

tipo, tipo_propiedad, area_m2, banos, estrato, habitaciones, ubicacion

Al ingresar a la aplicación se le pide al usuario ingresar una serie de atributos básicos relacionados con la propiedad, si el usuario marca la última casilla en SÍ**,** se habilitan nuevos campos para diligenciar. Luego de diligenciar los campos se da clic en el botón **Enviar datos**.

Figura 20.

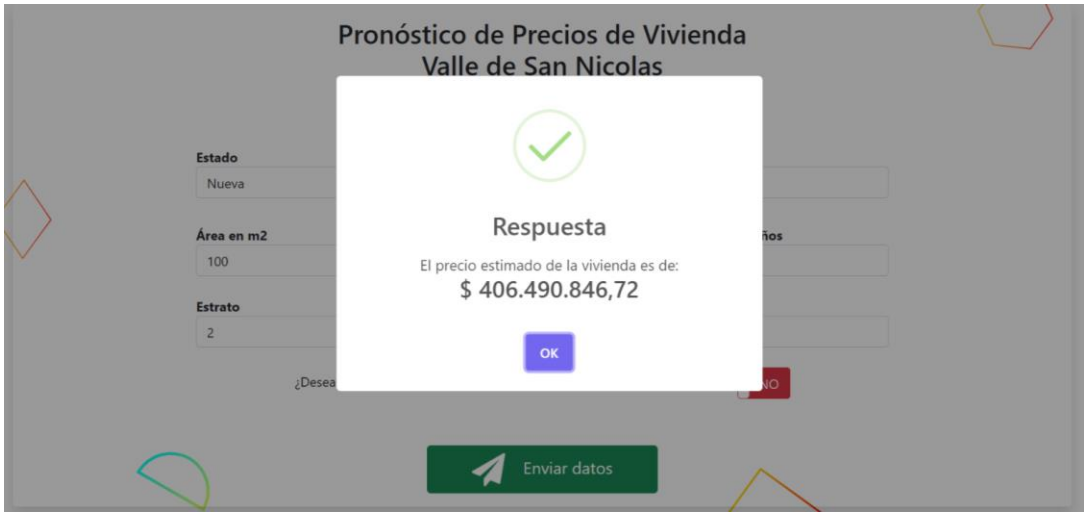
Capturas de la aplicación.

[23] Link de la aplicación: https://tranquil-thicket-90234.herokuapp.com/home

Los datos del formulario viajan hacia el servidor a través de una petición http (POST) y la aplicación los transforma para ingresarlos en el modelo pre entrenado y generar el pronóstico, una vez se complete el proceso, se envía una respuesta al cliente, mostrando en pantalla el resultado.

Figura 21.

Capturas de la aplicación.



Por último, el usuario podrá ver los resultados de los precios de acuerdo a las distintas características en una tabla al final de la página web, como se observa en la **Figura 22**.

Figura 22.

Capturas de la aplicación.

8. Discusión

Como objetivo inicial se pretende brindar una herramienta que permita predecir el precio de una vivienda en el Valle de San Nicolás contemplando como información de entrada las características propias de la vivienda. La motivación principal se da por el crecimiento que ha tenido el sector inmobiliario en los últimos años, que a su vez ha tenido efectos en la construcción de nuevas vías, construcción de espacios de recreación, aumento de industria y comercio y la cercanía a universidades, colegios y el aeropuerto.

Otra de las motivaciones, fue poner en práctica lo aprendido en la especialización; usando como solución analítica diferentes modelos que permitan describir la relación entre el precio de una vivienda en función de sus características principales. Aunque se probaron cuatro grupos de modelos (Random Forest, Gradient Boosting, XGBoost y Redes Neuronales), se puede evidenciar en los resultados del ajuste de los modelos valores muy similares en algunas de las métricas utilizadas.

Dentro de las iteraciones contempladas se resalta la importancia de estratificar la información no solo en la división del conjunto de datos en entrenamiento y prueba sino también contemplar la misma estratificación en el proceso de cross validation en la búsqueda de hiperparámetros. También se requería probar el efecto que tiene la información registrada por el vendedor a la hora de ofrecer su inmueble en la página de finca raíz en el ajuste de los modelos, logrando apreciar las palabras más importantes y evaluar su impacto en el ajuste. Aunque no proporcionó una mejora significativa se aprecia un pequeño aporte al ajuste de los modelos, experimentando diferentes procesamientos de la información para futuras iteraciones, ya que por sí solo, el campo de Descripción de las propiedades logra aportar un MAPE del 35% en el conjunto de prueba.

Enfocados en elegir un modelo final para desplegar, el modelo más simple logró obtener un buen ajuste dentro de las iteraciones contempladas, por lo que se elige un modelo con buen resultado en su ajuste y además parsimonioso. El modelo seleccionado para el despliegue final de la herramienta es el desarrollado en el experimento 8.2 donde se obtuvo un MAPE del 21% en la base de prueba y del 16% en la base de entrenamiento, utilizando como modelo el XGBoost. Con esta iteración se logra mejorar el sobreajuste del modelo que se observó con el GBT y tener buenas métricas en ambos conjuntos de datos (entrenamiento y prueba).

Finalmente, se quería anexar dentro de los experimentos realizados el último tema en modelación visto en la especialización que estaba enfocado en las redes neuronales. Dentro de esta iteración se generaron arquitecturas multicapa y multimodales, además de hacer uso de las redes recurrentes para el ajuste de datos no tabulados. Aunque su ajuste no mejoró significativamente la métrica del MAPE (se obtuvo un error del 24%) se resalta su utilidad en la explicación del precio de vivienda usando diferentes tipo de datos (tabular y texto) dentro del modelo.

9. Conclusiones

1. Los modelos estadísticos como la regresión lineal múltiple, la regresión por mínimos cuadrados ordinarios y la regresión Ridge, han sido de gran utilidad a la hora de predecir una variable de salida en función a una serie de variables de entrada, cuando se tiene claridad sobre una relación de causalidad y correlación entre estas. Las técnicas más avanzadas de machine learning, parten de las bases de la teoría estadística y las integran con los avances en computación permitiendo sortear algunas limitaciones de la teoría estadística clásica logrando incluso superar en algunos aspectos a los enfoques clásicos.

2. Las técnicas de ensamble que dan lugar a modelos como Random Forest, Gradient Boosting y XGBoost han representado un gran avance en los modelos de aprendizaje supervisado, a tal punto que XGBoost y Random Forest son unos de los primeros referente cuando se trata de modelar un problema de regresión, por la interpretabilidad de sus resultados y el buen desempeño que presentan.

3. Para el caso del dataset de viviendas en el Valle de San Nicolas, se puede concluir que los modelos con mejor resultado a la hora de predecir los precios tomando como referencia el MAPE fueron el Random Forest y el XGBoost, siendo destacable la posibilidad de implementar algunas medidas de regularización L1 y L2 en este último.

4. Dentro de las arquitecturas de redes neuronales propuestas se definió un modelo multimodal compuesto por una red neuronal recurrente para el análisis de texto (Descripción de los inmuebles), y una red neuronal multicapa para procesar el vector de características de los inmuebles, al usar esta arquitectura se tuvo un resultado que estuvo un poco por debajo del desempeño de los mejores modelos, pero que fue destacable dada la versatilidad del framework TensorFlow y Keras para implementar este tipo de arquitecturas.

10. Recomendaciones

- Es posible que el resultado obtenido en los modelos con redes neuronales, haya obedecido a un tamaño del dataset bastante limitado, por lo cual se sugiere reintentar la experimentación con un dataset mucho más robusto que permita explotar al máximo la capacidad de estos modelos, sin correr el riesgo de tener un sobreajuste por la cantidad de parámetros que llegan a tener las redes neuronales profundas.

- Al momento de realizar el análisis de precios de vivienda se tiene una cantidad muy elevada de variables a considerar, tanto de la propiedad como de su entorno, en ese punto es muy importante validar que las variables que se van a contemplar en el modelo si tengan una "alta" ocurrencia, de modo que se evite llenar el dataset con variables que poco aportan a la descripción de la variable de salida, pero que sí pueden representar un problema, al encontrarnos con una capacidad de cómputo limitada o sufrir problemas de convergencia en los algoritmos de optimización que soportan los modelos de ML.

11. Referencias

Cornare. (2015). *Análisis socioeconómico del Oriente Antioqueño*. Cornare. Retrieved October 31, 2021, from https://www.cornare.gov.co/Plan-crecimiento-verde/Anexo1.Analisis-Socioeconomico-Oriente-Antioqueno.pdf

Grajales, Y. (2019). *Modelo De Predicción De Precios De Viviendas En El Municipio De Rionegro Para Apoyar La Toma De Decisiones De Compra Y Venta De Propiedad Raíz* (Master thesis). UNIVERSIDAD PONTIFICIA BOLIVARIANA.

IONOS España. (2020). Tutorial para usar Selenium: conceptos básicos y primeros pasos - IONOS. Retrieved November 21, 2021, from https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/tutorial-de-selenium-webdriver/

Isaza, J. D. (2019). *Concepto Economico Oriente Antioqueño*. Cámara de comercio Oriente Antioqueño.

Muthukadan, B. (2018). 2. Getting Started — Selenium Python Bindings 2 documentation. Retrieved November 21, 2021, from https://selenium-python.readthedocs.io/getting-started.html

Park, B., & Bae, J. K. (2015). Using machine learning algorithms for housing price prediction: The case of Fairfax County, Virginia housing data. *Expert systems with applications*, *42*(6), 2928–2934.

Phan, T. D. (2018). Housing price prediction using machine learning algorithms: the case of melbourne city, australia. *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)* (pp. 35–42). Presented at the 2018 International Conference on Machine Learning and Data Engineering (iCMLDE), IEEE.

pythones.net. (2019). ¿Qué es Python? Definición, características y sus ventajas. Retrieved

November 7, 2021, from https://pythones.net/que-es-python-y-sus-caracteristicas/

Python Software Foundation. (2021). beautifulsoup4 · PyPI. Retrieved November 21, 2021, from https://pypi.org/project/beautifulsoup4/

Ramos, R., & Arias, J. (2020). 2.1 - The Perceptron — Fundamentos de Deep Learning. Retrieved November 14, 2021, from https://rramosp.github.io/2021.deeplearning/content/U2.01%20-%20The%20Perceptron.html

Richter, S. (2021). lxml - Processing XML and HTML with Python. Retrieved November 21, 2021, from https://lxml.de/

Téllez, V. (2021). *Método Automático Para LaPredicción Del Avalúo Comercial de Un Inmueble En La Ciudad De Bogotá* (Undergraduate thesis). UNIVERSIDAD CATÓLICA DE COLOMBIA.

Tissnesh, A. (2014). *Determinantes De La OfertaDe Vivienda Nueva En Medellín* (Undergraduate thesis). Universidad EAFIT.

Varma, A., Sarma, A., Doshi, S., & Nair, R. (2018). House price prediction using machine learning and neural networks. *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 1936–1939). Presented at the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), IEEE.

de la Vega, R. (2021). Introducción a la biblioteca Python lxml | Pharos. Retrieved November 21, 2021, from https://pharos.sh/introduccion-a-la-biblioteca-python-lxml/

12. Anexos

Tabla 10.
Lista de Variables del Dataset

Columna	#	Dtype	Descripción
tipo	2481	object	Indica el tipo de propiedad
url	2481	object	url del sitio web del anuncio
titulo	2481	object	Título del anuncio
precio	2481	float64	Precio de la propiedad
area_m2	2481	float64	Área en M2 de la propiedad
habitaciones	2481	float64	Nro. de Habitaciones de la propiedad
banos	2481	float64	Nro. de Baños de la propiedad
garajes	2481	float64	Nro. de Garajes de la propiedad
descripcion	2481	object	descripción del anuncio
antiguedad	2481	object	antigüedad del anuncio
balcon	2481	float64	Indica si la propiedad tiene balcón
bano_auxiliar	2481	float64	indica si la propiedad cuenta con baño auxiliar
comedor	2481	float64	Indica si cuenta con comedor
estudio	2481	float64	Indica si cuenta con estudio
servicios_publicos	2481	float64	Indica si la propiedad tiene servicios

ventilacion_natural	2481	float64	Indica si la propiedad tiene ventilación natural
zona_de_lavanderia	2481	float64	Indica si la propiedad tiene Zona_De_Lavanderia
en_conjunto_cerrado	2481	float64	Indica si la propiedad está en En_Conjunto_Cerrado
garaje_parqueadero(s)	2481	float64	Número de Parqueaderos de la propiedad
gimnasio	2481	float64	Indica si la propiedad tiene Gimnasio
piscina	2481	float64	Indica si tiene piscina
salon_comunal	2481	float64	Indica si tiene salón comunal
vivienda_multifamiliar	2481	float64	Indica si es multifamiliar
zona_infantil	2481	float64	Indica si cuenta con zonas infantiles
zonas_verdes	2481	float64	Indica si cuenta con zonas verdes
bombas_de_gasolina	2481	float64	Indica si queda cerca de bomba de gasolina
cerca_centro_comercial	2481	float64	Indica si queda cerca de centro comercial
comodas_vias_de_acceso	2481	float64	Indica si tiene vías de acceso
restaurantes	2481	float64	Indica si queda cerca de restaurantes
seguridad	2481	float64	Indica si está ubicado en zona segura
supermercados_ccomerciales	2481	float64	Indica si queda cerca de supermercados
trans_publico_cercano	2481	float64	Indica si queda cerca de transporte público
otros_datos	2481	object	otros datos del anuncio

latitud	2481	float64	Latitud de la propiedad
longitud	2481	float64	Longitud de la propiedad
bano_independiente	2481	float64	Indica si las habitaciones tienen baño independiente
closet	2481	float64	Las habitaciones cuentan con closet
cocina_integral	2481	float64	La propiedad cuenta con cocina integral
cocina_tipo_americano	2481	float64	La propiedad cuenta con cocina tipo americano
instalacion_de_gas	2481	float64	La propiedad cuenta con red de gas
acceso_pavimentado	2481	float64	La propiedad cuenta con acceso pavimentado
canchas_deportivas	2481	float64	Cuenta con canchas deportivas
garaje(s)	2481	float64	Número de garajes
cerca_de_zona_urbana	2481	float64	Indica si queda Cerca_De_Zona_Urbana
parques_cercanos	2481	float64	Indica si cuenta con Parques CERCANOS
zona_comercial	2481	float64	Indica si se encuentra en zona comercial
zona_residencial	2481	float64	Indica si se encuentra en zona residencial
estrato	2481	int64	Estrato de la propiedad
terraza	2481	float64	Terraza de la propiedad
patio	2481	float64	Patio de la propiedad
alcantarillado	2481	float64	Alcantarillado de la propiedad

arboles_frutales	2481	float64	Árboles en FRUTALES de la propiedad
bano_compartido	2481	float64	Baño en COMPARTIDO de la propiedad
en_casa	2481	float64	La propiedad tiene casa
establo	2481	float64	Establo de la propiedad
pozo_de_agua_natural	2481	float64	Pozo_De_Agua_Natural de la propiedad
area_rural	2481	float64	Área en RURAL de la propiedad
ascensor	2481	float64	Ascensor de la propiedad
parqueadero_visitantes	2481	float64	Parqueadero de VISITANTES de la propiedad
porteria_recepcion	2481	float64	Portería / RECEPCIÓN enla propiedad
vigilancia	2481	float64	Vigilancia de la propiedad
sobre_via_principal	2481	float64	Está ubicada Sobre_Via_Principal
cochera	2481	float64	Cuenta con cochera
calentador	2481	float64	Cuenta con calentador
chimenea	2481	float64	Cuenta con chimenea
comedor_auxiliar	2481	float64	Cuenta con comedor auxiliar
cuarto_de_servicio	2481	float64	Cuenta con cuarto de servicio
deposito_bodega	2481	float64	La propiedad tiene un depósito
hall_de_alcobas	2481	float64	La propiedad tiene hall de alcobas

parqueadero_interno	2481	float64	Hay un parqueadero interno
piso_de_alta_resistencia	2481	float64	Los pisos son de alta resistencia
piso_en_baldosa_marmol	2481	float64	Los pisos son de marmol
piso_en_cemento	2481	float64	Los pisos son de cemento
bahia_exterior_de_parqueo	2481	float64	Hay una bahía exterior de parqueo
cancha_de_squash	2481	float64	Cuenta con cancha de squash
en_condominio	2481	float64	Indica si la propiedad está en un condominio
garaje_cubierto	2481	float64	Indica si el garaje está cubierto
oficina_de_negocios	2481	float64	Indica si es un edificio de negocios
planta_electrica	2481	float64	Indica si cuenta con planta eléctrica
porteria_vigilancia	2481	float64	Indica si cuenta con portería
senderos_ecologicos	2481	float64	Indica si tiene senderos ecológicos
tanques_de_agua	2481	float64	La propiedad cuenta con tanques de agua
sobre_via_secundaria	2481	float64	La propiedad está ubicada en una vía secundaria
jardin	2481	float64	Indica si tiene Jardín
vista_panoramica	2481	float64	Indica si cuenta con vista panorámica
colegios_universidades	2481	float64	Está cerca de colegios y universidades
zona_campestre	2481	float64	La propiedad está en zona campestre

barra_estilo_americano	2481	float64	Tiene barra americana
citofono	2481	float64	Tiene citófono
circuito_cerrado_de_tv	2481	float64	Tiene Circuito cerrado de TV
salon_de_juegos	2481	float64	Cuenta con salón de juegos
despensa	2481	float64	Cuenta con despensa
cancha_de_futbol	2481	float64	Cuenta con cancha de futbol
lote_en_construccion	2481	float64	Es lote en construcción
cocina_equipada	2481	float64	La cocina se encuentra equipada
zona_industrial	2481	float64	Está en zona industrial
ascensores_comunales	2481	float64	Tiene ascensores comunales
con_administrador	2481	float64	Tiene administrador
sauna_turco_jacuzzi	2481	float64	Tiene turco / jacuzzi
shut_de_basura	2481	float64	Tiene shut de basuras
todos_los_servicios	2481	float64	Cuenta con todos los servicios
ubicada_en_edificio	2481	float64	Está ubicada en un edificio
zona_de_bbq	2481	float64	Cuenta con zona bbq
bosque_nativo	2481	float64	Propiedad con bosque nativo
en_edificio	2481	float64	La propiedad está ubicada en edificio

en_zona_residencial	2481	float64	Es una zona residencial
jardines_exteriores	2481	float64	Hay jardines exteriores
area_urbana	2481	float64	Está en el área urbana del municipio
asador	2481	float64	Tiene asador
vigilancia_privada_24*7	2481	float64	Vigilancia_Privada_24*7
zona_de_camping	2481	float64	Cuenta con zona de camping
vigilancia_24x7	2481	float64	Vigilancia en 24X7 de la propiedad
zona_de_hamacas	2481	float64	Cuenta con zona de hamacas
alarma_contra_incendio	2481	float64	Tiene alarma contra incendio
deteccion_de_humo	2481	float64	Cuenta con detectores de humo
lic_de_construccion	2481	float64	Cuenta con licencia de construcción
locales_comerciales	2481	float64	Hay locales comerciales cerca
piso_en_madera	2481	float64	Los pisos son en madera
disponibilidad_wifi	2481	float64	Existe red WiFi
en_zona_comercial	2481	float64	Es una zona comercial
cerca_a_sector_comercial	2481	float64	Hay zonas comerciales cerca de la propiedad
nacimientos_de_agua	2481	float64	La propiedad tiene nacimientos de agua
bano_de_servicio	2481	float64	Hay un baño de servicio

servicio_de_internet	2481	float64	Hay cobertura de internet en la zona
rio_quebrada_cercano(a)	2481	float64	Hay ríos o quebradas cercanos
galpon	2481	float64	Cuenta con un galpón
pesebrera	2481	float64	Cuenta con una pesebrera
cancha_de_baloncesto	2481	float64	Cuenta con cancha de baloncesto
invernadero	2481	float64	Cuenta con invernadero
kiosko	2481	float64	Cuenta con kiosko
sala_de_internet	2481	float64	Cuenta con sala de internet
cancha_de_tennis	2481	float64	Cuenta con cancha de Tennis
alarma	2481	float64	Cuenta con sistema de alarmas
sistema_de_riego	2481	float64	Cuenta con sistema de riegos
con_vivienda	2481	float64	Cuenta con una vivienda construida
casa_de_trabajadores	2481	float64	Cuenta con casa para empleados
bahias_de_parqueo	2481	float64	Cuenta con bahías de parqueo
aire_acondicionado	2481	float64	Cuenta con aire acondicionado
amoblado	2481	float64	La propiedad está amoblada
cocina_de_lena	2481	float64	Cuenta con cocina de leña
escalera_de_emergencia	2481	float64	Tiene escaleras de emergencia

gabinete_de_incendios	2481	float64	La propiedad tiene gabinete de incendios
ascensor(es)_inteligente(s)	2481	float64	Cuenta con ascensor inteligente
industrial	2481	float64	Se encuentra ubicada en zona industrial
banos_comunales	2481	float64	Cuenta con baños comunales
banos_mixtos	2481	float64	Cuenta con baños mixtos
control_de_acceso_digital	2481	float64	Cuenta con control de acceso digital
servicios_independientes	2481	float64	Los servicios son independientes
vivienda_bifamiliar	2481	float64	Vivienda es BIFAMILIAR
rociadores_de_agua	2481	float64	Tiene rociadores en el jardín
finca_ganadera	2481	float64	Es una finca ganadera
corrales	2481	float64	Tiene corrales
cuarto_de_conductores	2481	float64	Cuenta con cuarto de conductores
con_casa_club	2481	float64	Cuenta con casa club
finca_agricola	2481	float64	La propiedad es una finca agrícola
finca_agroganadera	2481	float64	La propiedad es una finca agroganadera
cableado_de_red	2481	float64	Cuenta con cableado de red
puerta_de_seguridad	2481	float64	Tiene puerta de seguridad
puerta_electrica	2481	float64	Puerta ELÉCTRICA en la propiedad

mezzanine	2481	float64	Tiene mezzanine
esquinero	2481	float64	Cuenta con esquineros
acceso_para_camiones	2481	float64	Tiene acceso para camiones
sensor_de_movimiento	2481	float64	Tiene sensores de movimiento
auditorio	2481	float64	Tiene un auditorio
patio_interno	2481	float64	Tiene un patio interno
cuarto_de_servicio.1	2481	float64	Cuenta con cuarto de servicio
cuarto_de_escoltas	2481	float64	Cuenta con cuarto de escoltas
panoramica_un_lado	2481	float64	Uno de los lados tiene vista panorámica
cocineta	2481	float64	La propiedad tiene cocineta
finca_avicola	2481	float64	Es una finca avícola
piso_en_alfombra	2481	float64	La propiedad tiene piso con alfombra
jaula_de_golf	2481	float64	Tiene jaula de golf
con_cerca_electrica	2481	float64	Cuenta con cerca eléctrica
oficinas_administrativas	2481	float64	Cuenta con oficinas administrativas
salon_de_videoconferencias	2481	float64	Cuenta con salón de video conferencias
pasaje_comercial	2481	float64	Se encuentra en pasaje comercial
en_club	2481	float64	Está cerca de un club

con_casa_prefabricada	2481	float64	La casa es prefabricada
finca_cafetera	2481	float64	Es finca cafetera
ubicacion	2481	object	ubicación del anuncio - municipio
3.5+ MB			

En la **Tabla 11.** Se relaciona los diferentes notebooks construidos para el desarrollo del web scraping, limpieza de los datos y todos los experimentos realizados en la fase de modelación y validación de los modelos.

Tabla 11.
Notebooks del desarrollo del trabajo.

Carpeta	Notebook	Descripción
Web Scraping y Limpieza	01_01_WEB_SCRAPING	contiene el código inicial del proceso de web scraping a la página de finca raíz.
Web Scraping y Limpieza	01_02_WEB_SCRAPING_UPT_12082021	se actualiza el cambio para extraer adicionalmente otros campos, se actualiza ya que la estructura de la página cambió un poco en la estructura de la información.
Web Scraping y Limpieza	02_01_Limpieza_Depuracion_Datos	realiza el proceso de limpieza y orden de la información extraída del web scraping
Web Scraping y Limpieza	02_02_Limpieza_Depuracion_Datos	realiza el proceso de limpieza y orden de la información extraída del web scraping del 12082021
Etapa de Modelamiento	01_01_Analisis_Descriptivo	contiene el análisis descriptivo realizado al conjunto de datos de interés
Etapa de Modelamiento	02_01_Primera_Iteracion	hace referencia al primer ajuste del modelo
Etapa de Modelamiento	03_01_Iteraciones_Modelado_MAPE	contiene el primer experimento de modelación
Etapa de Modelamiento	03_02_Iteraciones_Modelado_x_Ub	contiene el segundo experimento de

	icacion	modelación
Etapa de Modelamiento	03_03_Iteraciones_Modelado_x_Estrato	contiene el tercer experimento de modelación
Etapa de Modelamiento	03_04_Iteraciones_Modelado_modelos_x_ubicacion	contiene el cuarto experimento de modelación
Etapa de Modelamiento	03_05_Iteraciones_Modelado_modelos_x_estrato	contiene el quinto experimento de modelación
Etapa de Modelamiento	03_06_Iteraciones_Modelado_x_Texto_x_Ubicacion_v2	contiene el sexto y séptimo experimento de modelación. Para realizar el séptimo se realiza el ajuste del modelo sólo con las variables de texto.
Etapa de Modelamiento	03_08_Iteraciones_Modelado_x_Ubicacion_Variables_Basicas	contiene el octavo experimento de modelación
Etapa de Modelamiento	03_09_Iteraciones_Modelado_Var_Basicas_Texto_x_Ubicacion	contiene el noveno experimento de modelación
Etapa de Modelamiento	03_10_Iteraciones_Modelado_Var_Basicas_Texto_x_Ubicacion_RNN2	contiene el décimo experimento de modelación
Etapa de Modelamiento	03_11 Modelos Para Despliegue	contiene los modelos ajustados, guardados en .joblib y desplegados
Etapa de Modelamiento	04_01_CurvaAprendizaje	contiene el resultado de las primeras curvas de aprendizaje de la primera experimentación
Etapa de Modelamiento	04_02_CurvaAprendizaje	contiene el resultado de las curvas de aprendizaje del mejor modelo de la segunda experimentación de cada modelo
Etapa de Modelamiento	05_01_Importancia_Variables_x_Ubicacion	evaluar la importancia de variables del segundo experimento
Etapa de Modelamiento	05_02_Importancia_Variables_x_Ubicacion_variables_basicas	evaluar la importancia de variables del octavo experimento