Neiva, 23 de mayo del 2023

Señores

CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD SURCOLOMBIANA

Ciudad

El (Los) suscrito(s):

Francisco Javier Imbachi Rivas, con C.C. No. 1075319047,

Anderson Arley Ramírez Charry, con C.C. No. 1004035472,

Autor(es) de la tesis y/o trabajo de grado titulado **MODELO DE MACHINE LEARNING PARA LA ESTIMACION DEL VALOR COMERCIAL DE UN INMUEBLE EN LA CIUDAD DE NEIVA**
presentado y aprobado en el año **2023** como requisito para optar al título de **Matemático**;

Autorizo (amos) al CENTRO DE INFORMACIÓN Y DOCUMENTACIÓN de la Universidad Surcolombiana para que, con fines académicos, muestre al país y el exterior la producción intelectual de la Universidad Surcolombiana, a través de la visibilidad de su contenido de la siguiente manera:

- Los usuarios puedan consultar el contenido de este trabajo de grado en los sitios web que administra la Universidad, en bases de datos, repositorio digital, catálogos y en otros sitios web, redes y sistemas de información nacionales e internacionales “open access” y en las redes de información con las cuales tenga convenio la Institución.

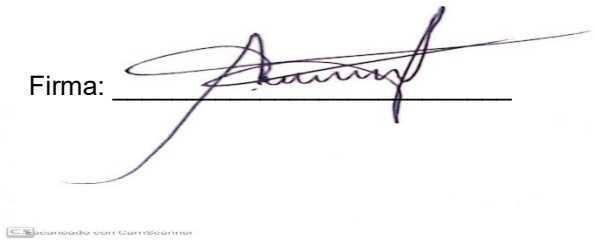
- Permita la consulta, la reproducción y préstamo a los usuarios interesados en el contenido de este trabajo, para todos los usos que tengan finalidad académica, ya sea en formato Cd-Rom o digital desde internet, intranet, etc., y en general para cualquier formato conocido o por conocer, dentro de los términos establecidos en la Ley 23 de 1982, Ley 44 de 1993, Decisión Andina 351 de 1993, Decreto 460 de 1995 y demás normas generales sobre la materia.

- Continúo conservando los correspondientes derechos sin modificación o restricción alguna; puesto que, de acuerdo con la legislación colombiana aplicable, el presente es un acuerdo jurídico que en ningún caso conlleva la enajenación del derecho de autor y sus conexos.

De conformidad con lo establecido en el artículo 30 de la Ley 23 de 1982 y el artículo 11 de la Decisión Andina 351 de 1993, “Los derechos morales sobre el trabajo son propiedad de los autores” , los cuales son irrenunciables, imprescriptibles, inembargables e inalienables.

EL AUTOR/ESTUDIANTE:

Anderson Arley Ramirez Charry

Firma: 

EL AUTOR/ESTUDIANTE:

Francisco Javier Imbachi RIvas

Firma: Francisco Javier Imbachi Rivas

TÍTULO COMPLETO DEL TRABAJO: MODELO DE MACHINE LEARNING PARA LA ESTIMACION DEL VALOR COMERCIAL DE UN INMUEBLE EN LA CIUDAD DE NEIVA

AUTOR O AUTORES:

Primero y Segundo Apellido	Primero y Segundo Nombre
Imbachi Rivas Ramírez Charry	Francisco Javier Anderson Arley

DIRECTOR Y CODIRECTOR TESIS:

Primero y Segundo Apellido	Primero y Segundo Nombre

ASESOR (ES):

Primero y Segundo Apellido	Primero y Segundo Nombre
Roldán Jiménez	Diego Gerardo

PARA OPTAR AL TÍTULO DE: Matemático

FACULTAD: Ciencias Exactas y Naturales

PROGRAMA O POSGRADO: Matemática aplicada

CIUDAD: Neiva **AÑO DE PRESENTACIÓN: 2023** **NÚMERO DE PÁGINAS: 70**

TIPO DE ILUSTRACIONES (Marcar con una **X**)**:**

Diagramas___Fotografías___Grabaciones en discos_**X**__ Ilustraciones en general__ Grabados____ Láminas___Litografías___Mapas___Música impresa___Planos___Retratos___ Sin ilustraciones___Tablas o Cuadros__

SOFTWARE requerido y/o especializado para la lectura del documento: PDF

MATERIAL ANEXO:

PREMIO O DISTINCIÓN *(En caso de ser LAUREADAS o Meritoria)*: Ninguno

PALABRAS CLAVES EN ESPAÑOL E INGLÉS:

Español	**Inglés**
1. Aprendizaje Automático	Machine Learning
2. Bosques Aleatorios	Random Forest
3. XGBoost	XGBoost
4. Redes Neuronales	Neural Networks

RESUMEN DEL CONTENIDO: (Máximo 250 palabras)

El objetivo principal de este trabajo de grado es ajustar un modelo de machine learning para así determinar el valor comercial de los inmuebles, a partir de su ubicación geográfica en Neiva y características principales tales como: tipo de inmueble, área, estrato, piso, habitaciones, parqueaderos, baños, antigüedad, etc. Dicha información es obtenida y analizada por medio de la técnica de Web Scraping, a partir de datos del mercado y ofertas de inmuebles similares, que se encuentran en las diferentes plataformas de ventas de inmuebles en Neiva. mediana y macroempresa se observaron pequeñas diferencias en el ajuste de ambos enfoques. Finalmente, se tiene que el enfoque bayesiano genera mejores resultados y además, se observó que los ingresos de las empresas en Colombia reportaron un decaimiento en el año 2020 frente al año 2019. Es decir, que la pandemia si afecto significativamente a las empresas colombiana y especialmente a las microempresas y pequeñas empresas.

Por tal motivo, en este proyecto se implementó las técnicas de Machine Learning (Random Forest, XGBoost y Redes Neuronales) con el cual se puede determinar el valor de los inmuebles en la ciudad de Neiva. De esta manera, se concluyó que el mejor modelo para la predicción del valor comercial de un inmueble en la ciudad de Neiva es el de random forest, al presentar valores óptimos en las métricas de desempeño.

ABSTRACT: (Máximo 250 palabras)

The main objective of this work is to adjust a machine learning model in order to determine the commercial value of the properties, based on their geographical location in Neiva and main characteristics such as: type of property, area, stratum, floor, rooms, parking lots, bathrooms, age, etc. Said information is obtained and analyzed through the Web Scraping technique, based on market data and offers of similar properties, which are found on the different real estate sales platforms in Neiva.

For this reason, in this project Machine Learning techniques (Random Forest, XGBoost and Neural Networks) were implemented, with which the value of real estate in the city of Neiva can be determined. In this way, it was concluded that the best model for predicting the commercial value of a property in the city of Neiva is the random forest model, as it presents optimal values in the performance metrics.

APROBACION DE LA TESIS

Nombre Asesor de Tesis: Diego Gerardo Roldán Jiménez

Firma: Diego Gerardo Roldan Jimenez

Nombre Jurado: Álvaro Javier Cangrejo

Firma:

Nombre Jurado: Oswaldo Delgado

Firma:

MODELO DE MACHINE LEARNING PARA LA ESTIMACIÓN DEL VALOR COMERCIAL DE UN INMUEBLE EN LA CIUDAD DE NEIVA

UNIVERSIDAD
SURCOLOMBIANA

FRANCISCO JAVIER IMBACHI RIVAS
ANDERSON ARLEY RAMIREZ CHARRY

Universidad Surcolombiana
Facultad de Ciencias Exactas y Naturales
Programa de Matemática Aplicada
Neiva (Huila), 2023

MODELO DE MACHINE LEARNING PARA LA ESTIMACIÓN DEL VALOR COMERCIAL DE UN INMUEBLE EN LA CIUDAD DE NEIVA

Trabajo de Grado para Optar por el título de Matemático

Dirigido por: Diego Gerardo Roldan Jimenez

Realizado por: FRANCISCO JAVIER IMBACHI RIVAS
ANDERSON ARLEY RAMIREZ CHARRY

Universidad Surcolombiana
Departamento de Ciencias Exactas y Naturales
Programa de Matemática Aplicada
Neiva (Huila), 2023

Agradecimientos

Queremos agradecerle al profesor Diego Roldan Jiménez, por su ilimitada colaboración, gracias por su conocimiento y paciencia para guiarnos a través de este trabajo. A mi papá Jhon Faiver Imbachi, el cual me hizo posible el estudio en esta universidad sustentándome económicamente mientras cumplía el sueño de ser un profesional, a mi mamá Silvia Helena Rivas Torres y mi hermana Paula Andrea Imbachi Rivas por su fe en Dios y apoyo todo este tiempo universitario y en la vida académica. Al matemático Juan Sebastián Suarez por sus aportes y conocimientos del web scraping que fueron valiosos para el desarrollo del trabajo.

De mi parte, agradecer infinitamente a mi querida madre Nury Charry, que ha sido el pilar para cumplir con ser matemático y me ha tenido paciencia en momentos difíciles, agradecer a mi hermano Jair Ramiréz Charry que también ha sido un apoyo importante en muchos aspectos y siempre me aconsejo para no desistir de este sueño, agradecer a los docentes que han sido parte del camino universitario por transmitirme los conocimientos necesarios para hoy poder estar aquí, agradecer a mis tías y primas que con su granito de arena me han apoyado de alguna manera para superar los obstáculos, y agradecer a todos mis amigos y amigas, que a pesar de no poder escribirlos uno a uno acá , ellos saben que han sido importantes para llegar al final de este primer paso en mi vida.

Dedicatoria

Dedicamos los frutos de este trabajo a toda nuestra familia. El mayor agradecimiento a nuestros padres que nos apoyaron y aguantaron los malos y no tan malos momentos. Gracias por enseñarnos a enfrentar la adversidad en lugar de perder la cabeza o morir en el intento.

También dedico este trabajo a mi esposa María Rojas. Por tu paciencia, por tu comprensión, por tu compromiso, por tu fuerza y por tu amor. Debo pedirte disculpas porque te has visto directamente afectado por las consecuencias del trabajo. De hecho, ella me ayudó a lograr el equilibrio y me permitió alcanzar mi máximo potencial. Nunca dejaré de estar agradecido por eso.

Este trabajo está dedicado a mi madre Nury Charry, quien ha puesto toda su voluntad y esfuerzo en mí, espero algún día poder enorgullecerla y demostrarle que todo ha valido la pena.

TABLA DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

Resumen

El objetivo principal de este trabajo de grado es ajustar un modelo de machine learning para así determinar el valor comercial de los inmuebles, a partir de su ubicación geográfica en Neiva y características principales tales como: tipo de inmueble, área, estrato, piso, habitaciones, parqueaderos, baños, antigüedad, etc. Dicha información es obtenida y analizada por medio de la técnica de Web Scraping, a partir de datos del mercado y ofertas de inmuebles similares, que se encuentran en las diferentes plataformas de ventas de inmuebles en Neiva.

Por tal motivo, en este proyecto se implementó las técnicas de Machine Learning (Random Forest, XGBoost y Redes Neuronales) con el cual se puede determinar el valor de los inmuebles en la ciudad de Neiva. De esta manera, se concluyó que el mejor modelo para la predicción del valor comercial de un inmueble en la ciudad de Neiva es el de random forest, al presentar valores óptimos en las métricas de desempeño.

Palabras Clave: Machine Learning, XGBoost,Random Forest,Redes Neuronales.

Abstract

The main objective of this work is to adjust a machine learning model in order to determine the commercial value of the properties, based on their geographical location in Neiva and main characteristics such as: type of property, area, stratum, floor, rooms, parking lots, bathrooms, age, etc. Said information is obtained and analyzed through the Web Scraping technique, based on market data and offers of similar properties, which are found on the different real estate sales platforms in Neiva.

For this reason, in this project Machine Learning techniques (Random Forest, XGBoost and Neural Networks) were implemented, with which the value of real estate in the city of Neiva can be determined. In this way, it was concluded that the best model for predicting the commercial value of a property in the city of Neiva is the random forest model, as it presents optimal values in the performance metrics.

Keywords: Machine Learning, XGBoost, Random Forest, Neural Networks.

Capítulo 1

1. Introducción

El mercado inmobiliario en Colombia es una de las grandes industrias del país, debido a que generan un valor agregado a su economía, ya que son los responsables de la compra y venta de inmuebles indispensables para la sociedad. En concordancia con lo anterior, si se desea aumentar la compra y venta de inmuebles, se debe incluir él avaluó comercial. Dado que el avalúo comercial, permite una estimación del valor de un inmueble al momento de ser puesto a la venta.

El avalúo comercial en Colombia, es un método realizado por un experto, llamado perito o avaluador, el cual esta certificado por la resolución IGAC 620 del 2008. Dicho método consiste en analizar el inmueble al que se le quiere determinar el valor comercial, basado en sus características tipo terreno y construcción, para luego ser comparado con inmuebles del mercado que tienen características similares a los del inmueble a evaluar.

Con base al aumento constante en la producción de datos, los profesionales se han visto en la tarea de utilizar múltiples técnicas que permiten acceder a algoritmos predictivos potentes. En los últimos años, el machine learning se ha convertido en una de estas técnicas la cual es interdisciplinaria, debido a que se puede aplicar prácticamente en todos los ámbitos de investigación académico e industrial.

El machine learning reúne algoritmos que permiten reconocer patrones presentes en los datos y ajustar con ellos modelos que los representan. Por esta razón, se han presentado aumentos en las investigaciones que buscan implementar métodos predictivos, obteniendo buenos resultados, desarrollando redes neuronales y técnicas de machine learning. Algunas de las investigaciones son (Rodrigo, 2020) “Machine learning con Python” en la ciudad de New York; (Martínez y Téllez, 2021) “Método automático para la predicción del avalúo comercial de un inmueble en la ciudad de Bogotá”, lo cual muestran la búsqueda de métodos predictivos que facilitan la predicción del valor de un avalúo comercial.

En Colombia, el avalúo comercial no suele ser muy solicitado por los propietarios de los inmuebles, debido a que presentan ciertas desventajas en algunos factores como: tiempos extensos de entrega una vez solicitado, altos costos en el pago del servicio y margen de error considerable en la estimación del valor comercial del inmueble.

Así pues, el presente trabajo propone el ajuste de modelos de machine learning y redes neuronales, para la predicción del valor comercial de los inmuebles en la ciudad de Neiva, dado que la ciudad es la capital de uno de los departamentos de Colombia, la cual viene en los últimos años ampliándose de forma territorial, causando un auge en términos de compra, venta y construcción de inmuebles. (Del Rio, 2021).

El presente trabajo de investigación se realizó durante el año 2022 y constó de diez capítulos. El primer capítulo se basó en la introducción, donde se detalla el contexto en el que se desarrolló y además se justificó la importancia de estudiar el tema. Se suele incluir una breve descripción de los objetivos del estudio y de la estructura que se seguirá en el trabajo. En el segundo capítulo, se explicó con detalle el problema de investigación que se abordará en el estudio. Se describen las razones por las cuales el problema es relevante y se presentan preguntas que ayuden a enfocar el estudio.

En el tercer capítulo se realizó una revisión de la literatura sobre el tema de investigación. Se recopilaron estudios previos relacionados con el problema de investigación y se presentaron las principales teorías y enfoques que existen en el tema. En el cuarto capítulo se estableció el marco teórico, donde se indicaron los conceptos relevantes y necesarios para la compresión del presente trabajo. Además, se incluyeron las matemáticas que se presentan en los modelos de machine learning y redes neuronales, para su construcción y ajuste en los datos.

En el capítulo quinto se describió el objetivo general y los objetivos específicos que se persiguen en el trabajo de investigación. En el sexto capítulo se detalló la metodología, la cual muestra el procedimiento para la realización del trabajo, paso a paso. Dicho capítulo abarcó las cinco actividades que se presentan en un proyecto de datos.

En el séptimo capítulo, se presentaron los resultados obtenidos de los procedimientos descritos en la metodología. Además, se proporcionaron comentarios pertinentes acerca de los resultados, para su evaluación e interpretación. En el octavo capítulo, se enunciaron las conclusiones que se obtienen tras discutir los resultados conseguidos, teniendo en cuenta el propósito del trabajo y las limitaciones existentes. Así mismo, se presentaron recomendaciones para mejoras adicionales.

En el noveno capítulo, se incluyeron las referencias bibliográficas que se han utilizado en el trabajo. Se presenta una lista ordenada alfabéticamente de los autores citados en el texto y se describen las fuentes consultadas. Por último. en el décimo capítulo se incluyeron los anexos que complementan la información presentada en el trabajo. En este caso son los scripts donde se encuentra el código en lenguaje de programación Python el cual se hizo posible la metodología.

Capítulo 2

2. Planteamiento del problema

En los últimos años, el mercado inmobiliario en la ciudad de Neiva se ha impulsado en términos de comercialización e iniciación de obras. (Del Rio, 2021), esto, se debe principalmente a estrategias adoptadas por el gobierno, a programas como 'Mi casa ya'. Con base a esto, determinar el valor de un inmueble llega ser un factor importante para la adquisición del mismo, dado que al determinar el verdadero valor comercial del inmueble garantiza que su transacción no se distorsione y, en última instancia, se convierta en un precio irrazonable para todas las partes involucradas en el negocio.

Actualmente en Colombia, los valores comerciales se vienen determinando por medio de un avalúo. La entidad o persona solicitante, podrá solicitar la elaboración de un avalúo a entidades privadas de propiedad raíz, con domicilio en el municipio donde se encuentren ubicados el o los inmuebles en objeto de avalúo.

La solicitud de realización de los avalúos de que trata el Decreto 1420 de 1998(pág. 3), deberá presentarse por la entidad interesada en forma escrita, firmada por el representante legal o su delegado legalmente autorizado, señalando el motivo del avalúo y entregando a la entidad encargada numerosos documentos legales referente al inmueble. Después de que el solicitante entregue los documentos legales, el plazo para la realización de los avalúos es de máximo 30 días hábiles. La cual designará para el efecto uno de los peritos privados o avaluadores que se encuentren registrados y autorizados por ella, cumpliendo la ley 1673 del 2003, por la cual se reglamenta la actividad del avaluador.

El perito o avaluador hace la visita al inmueble, analizando los factores bases los cuales son: terreno y construcción, que están impuestos en la resolución IGAC 620 de 2008. El método más utilizado para encontrar dichos factores es el método comparativo o de mercado, que consiste en analizar inmuebles que tengan características similares a las del objeto en valoración, para así poder determinar el valor comercial del inmueble.

Con lo anterior, se puede concluir que un avalúo suele ser una tarea tardía y llena de pasos formales, los cuales afectan el tiempo de los que quieren invertir en el mercado inmobiliario. Según (Borrero Ochoa, 2000), el método comparativo o de mercado, conduce a una mayor margen de error (cercano al 10 %). Dos avaluadores que utilicen este método podrían tener una diferencia entre sus avalúos de hasta el 20 % (margen de error de 10 % para cada uno). (pág. 51) Lo que genera varios riesgos y desventajas que pueden llegar a perjudicar el patrimonio o los ingresos del comprador o vendedor de inmuebles. Ya que se tomará mucho tiempo pactar un acuerdo entre la compra y venta del inmueble. Los honorarios de un perito tienen un alto costo, puesto que depende del valor comercial del inmueble, generalmente la tarifa mínima es de $300.000 por

inmueble cuyo valor comercial es de $100.000.000. Si el costo del inmueble es mayor, se le suma a los $300.000 el 1 por mil. Forzando al comprador y vendedor de inmuebles a no contratar con dicho servicio, lo que hará determinar el valor comercial del inmueble por su propia cuenta, basado en opiniones subjetivas o suposiciones sin las debidas comprobaciones del mercado.

En un mundo cada vez más digital, las empresas apuestan por plataformas que permitan a los clientes realizar él avaluó comercial de sus inmuebles en minutos, utilizando una variedad de datos analíticos.

No obstante, debemos de ser cuidadosos con la información, ya que muchas de estas plataformas inmobiliarias manejan datos desactualizados, haciendo que afecten en los sesgos a la hora de determinar los valores comerciales de los inmuebles. Para ello aplicaremos la transformación de datos que consiste en crear las bases de datos y tablas de información, llevando a cabo la limpieza de datos desactualizados, lo que soluciona el error de los sesgos a la hora de determinar los valores comerciales de los inmuebles en la ciudad de Neiva.

Finalmente, para problemas inmobiliarios, los inmuebles a ser considerados en el estudio serán únicamente departamentos y casas, ya que la predicción para otro tipo de inmuebles, como lotes, requiere tipos de técnicas distintas a las aplicadas en los inmuebles descritos anteriormente. Por eso es mejor trabajar con estas propiedades, además, son las más buscadas por la gente.

2.1 Pregunta de Investigación

¿Cómo determinar el valor comercial de los inmuebles en la ciudad de Neiva, ofreciendo mejores resultados de manera precisa y rápida sin la elaboración de un avalúo por parte de un perito?

Capítulo 3

3. Estado del arte y justificación

3.1 Estado del arte

El machine learning se ha convertido en una herramienta esencial en la predicción de precios de viviendas. A continuación, se presentan algunas de las investigaciones más relevantes en esta área, a nivel internacional y en el contexto colombiano:

Nivel Internacional:

- (Jha *et al.*, 2020), desarrolló un modelo de predicción de precios de inmuebles, utilizando datos reales de las propiedades del condado de Volusia de Florida de diez años. Con el objetivo de predecir el precio de vivienda de un inmueble. Para desarrollar dicho modelo utilizaron algoritmos de machine learning como: Regresión logística, Arboles de decisión, Voting classifer y XGBoost. De los cuales, el algoritmo XGBoost ofreció mejores resultados en comparación a los demás algoritmos empleados.
- (Antón, 2020), un modelo para la predicción de precios del mercado de un inmueble en la ciudad de Valencia empleando Redes Neuronales Profundas, y una herramienta como el web scraping de donde se obtuvo un valor de 15,66 % y 18,55 % sobre el precio, el cual se considera por el autor como satisfactorio y de cierta utilidad, pero esperando en un futuro un modelo con más características.
- (Ho *et al.*, 2021), sobre predicción de precios con algoritmos de machine learning, donde se aplica tres algoritmos de aprendizaje automático en Hong Kong que incluyen: máquina de vectores de soporte (SVM), bosque aleatorio (RF) y máquina de aumento gradiente (GBM) en la evaluación de los precios de las propiedades y luego compara los resultados de estos algoritmos. Concluyen que el aprendizaje automático ofrece una técnica alternativa prometedora en la valoración de la propiedad e investigación de la predicción de precio de las propiedades.
- (Huang, 2019), que busca predecir el valor de las viviendas a través de machine learning, utiliza los datos inmobiliarios de los tres condados en los Ángeles, California, Estados Unidos. Se demuestra que, a través de este experimento, la selección de características es un proceso muy importante, y que los métodos basados en árboles son muy útiles cuando se enfrenta a un modelo con una gran cantidad de características en la predicción del valor de la vivienda.

  Nivel Colombia:

- (Grajales *et al.*, 2019), describe en su estudio, la técnica cuya función es la extracción de datos en las diferentes páginas web de ventas de inmuebles, con el objetivo de predecir los precios de inmuebles en Rio Negro. Utilizando técnicas

de machine learning y Deep learning para desarrollar el modelo. Los resultados mostraron que los factores más importantes a la hora de determinar el precio de una propiedad son el área de la casa, tipo de vivienda y el estrato. Lo cual se logró evaluar un modelo de gradient boosting que pudiese predecir dichos resultados.

- (Martínez y Téllez, 2021) , con el objetivo de predecir los valores comerciales de los inmuebles en la ciudad de Bogotá utilizo: Random Forest, regresión lineal, Arboles de decisión y redes neuronales a partir de datos de páginas web. Los resultados que obtuvieron fue que el mejor modelo es Random Forest, dado que sus medidas de desempeño fueron superiores respecto a los demás modelos.

3.2 Justificación

La predicción de precios de vivienda en Colombia es un tema de gran importancia debido a la relevancia que tiene este sector en la economía del país. La toma de decisiones informadas y precisas en el mercado inmobiliario es crucial para los compradores, vendedores, inversores y tomadores de decisiones gubernamentales. Por esta razón, se ha visto un aumento en el uso de modelos de machine learning para predecir los precios de vivienda en Colombia.

En primer lugar, como describe (De La Hoz *et al.*, 2019),los modelos de machine learning son capaces de analizar grandes cantidades de datos y detectar patrones y tendencias que los humanos no pueden. Esto permite que los modelos de machine learning sean más precisos y confiables que los métodos tradicionales de predicción de precios de vivienda. Según un estudio de (Maisueche, 2019), "los modelos de aprendizaje automático se están convirtiendo rápidamente en herramientas valiosas para la predicción de precios de vivienda debido a su capacidad para manejar grandes cantidades de datos".

En segundo lugar, los modelos de machine learning son capaces de tener en cuenta una variedad de factores que influyen en el precio de la vivienda, como la ubicación, el tamaño, la edad y las características del vecindario. Esto significa que los modelos de machine learning pueden proporcionar predicciones más precisas y detalladas que los métodos tradicionales.Los modelos de aprendizaje automático pueden reconocer patrones en los datos que los expertos humanos pueden no haber considerado, y por lo tanto, facilitar predicciones más exactas y eficientes".

En tercer lugar, los modelos de machine learning pueden adaptarse y mejorar con el tiempo a medida que se recopila más información y se actualizan los datos. Esto significa que los modelos de machine learning pueden proporcionar predicciones más precisas y útiles a medida que se recopilan más datos sobre el mercado inmobiliario colombiano. Según un estudio de (Martín, 2021), "los modelos de aprendizaje automático son capaces de mejorar con el tiempo a medida que se actualiza la información y se ajusta el modelo, lo que los convierte en herramientas valiosas para la predicción de precios de vivienda a largo plazo".

Este modelo busca ayudar a que el valor comercial sea consistente de acuerdo con lo que se presenta en el mercado inmobiliario en la ciudad de Neiva. Siendo una herramienta para las personas que desean invertir en el mercado inmobiliario, ya sea en la compra o venta de inmuebles. Además, puede llegar a ser una herramienta no solo para los expertos en el mercado inmobiliario, sino también para los principiantes que se desean iniciar en este mercado, identificando las diferentes variables importantes que inciden a la hora de determinar el valor comercial de los inmuebles en Neiva, por medio de los patrones en la recolección de la información.

En conclusión, los modelos de machine learning son una herramienta valiosa para la predicción de precios de vivienda en Neiva, debido a su capacidad para manejar grandes cantidades de datos, considerar una variedad de factores, adaptarse y mejorar con el tiempo. Su uso puede ayudar a tomar decisiones más informadas y precisas en el mercado inmobiliario, lo que puede tener un impacto positivo en la economía del país.

Capítulo 4

4. Marco Teórico

Para el desarrollo del presente trabajo, se van a considerar los siguientes elementos fundamentales desde la teoría.

4.1 Inmueble como Bien Raíz

Según el autor Jorge Valencia Jaramillo en su obra "Derecho Civil. Parte General", el inmueble es "todo bien cuya característica es la estabilidad, la fijeza o la inmovilidad en relación con el suelo y su incorporación a él, de modo que no se puede trasladar sin que se altere su naturaleza o se deteriore"(Valencia, 2014)

Por otro lado, el autor David Martínez Carreño en su libro "Bienes y derechos reales", menciona que,"un inmueble es el bien que se encuentra ligado de manera indisoluble al suelo, por lo que no es susceptible de traslación de lugar, ni física ni jurídica, sino mediante un acto formal que implica una transferencia de dominio"(D. Martínez, 2012).

En resumen, el inmueble en Colombia se refiere a los bienes raíces que tienen una fijeza o inmovilidad en relación con el suelo y que, por tanto, no pueden ser trasladados sin alterar su naturaleza. Esta definición se encuentra establecida en el Código Civil y es complementada por otras normas y doctrinas jurídicas.

4.2 Avalúo como Herramienta Tradicional

El avalúo es un "proceso mediante el cual se determina el valor de un bien, ya sea inmueble, mobiliario, intangible u otro tipo de activo, a través de la aplicación de métodos y técnicas especializadas"(Kozak, 2015). Según (Pérez, 2019), el avalúo es una disciplina que se encarga de estimar el valor de los bienes, considerando diversos factores como la ubicación, las características físicas y funcionales, el estado de conservación, la oferta y la demanda, entre otros.

La importancia del avalúo radica en su utilidad para diversos fines, como la toma de decisiones en materia de inversión, la determinación de impuestos, la obtención de créditos y la liquidación de patrimonios (A. Torres, 2017). Según (Carreño, 2018), el avalúo es una herramienta clave en el ámbito de los negocios, ya que permite a los propietarios de bienes conocer el valor real de sus activos y tomar decisiones informadas acerca de su gestión.

En el marco teórico del avalúo, se consideran diversas metodologías y técnicas para la determinación del valor de los bienes, como el método comparativo de mercado, el método de costos y el método de ingresos (Kozak, 2015). Además, es importante destacar que el avalúo debe ser realizado por profesionales capacitados y objetivos, que garanticen la precisión y la

imparcialidad en la valoración de los bienes (Pérez, 2019).

En resumen, el avalúo es un proceso esencial en el ámbito financiero y de los negocios, que permite la valoración precisa de los bienes y la toma de decisiones informadas. Para llevar a cabo un avalúo adecuado, se deben considerar diversas metodologías y técnicas especializadas, así como contar con profesionales capacitados y objetivos que garanticen la imparcialidad en el proceso.

4.3 Avalúo comercial

Permite determinar el valor de la propiedad con el objetivo de que el valor sea de tipo "comercial" esto quiere decir, para la compra, venta, arrendamiento o cualquier tipo de transacción de carácter comercial.

4.4 Mercado inmobiliario en Colombia

Según (Geltner *et al.*, 2013), el mercado inmobiliario es un mercado local en el que los precios de los bienes raíces están influenciados por las características de la propiedad y su entorno, así como por las tendencias macroeconómicas y las políticas gubernamentales. Además, estos autores señalan que la oferta y la demanda son factores clave en la determinación del precio de las propiedades y que existen diferentes métodos de valuación que se utilizan para establecer el valor de mercado de los inmuebles.

Por su parte, (C. Torres y Mora, 2019) afirman que el mercado inmobiliario en Colombia ha experimentado un crecimiento sostenido en los últimos años, impulsado por el aumento en la demanda de vivienda y por la expansión de la economía. Según estos autores, los precios de los bienes raíces en Colombia están influenciados por factores como la ubicación, la oferta y la demanda, los costos de construcción y los aspectos legales y tributarios relacionados con la propiedad.

En resumen, el mercado inmobiliario en Colombia es un mercado local en el que la oferta y la demanda son factores clave en la determinación del precio de los bienes raíces, y en el que influyen diferentes factores como la ubicación, la calidad, la disponibilidad y las políticas gubernamentales relacionadas con la propiedad.

4.5 Web scraping

El web scraping es una técnica de extracción de datos que consiste en la recopilación automatizada de información de diferentes sitios web a través de la utilización de software especializado. Según (Kim *et al.*, 2020), el web scraping se ha vuelto una técnica muy popular en el campo del análisis de datos debido a su capacidad para recopilar grandes cantidades de información de manera rápida y eficiente. Además, esta técnica puede ser utilizada para diferentes propósitos, como la minería de datos, la investigación de mercado y la monitorización de precios, entre otros.

El web scraping se realiza mediante la utilización de bots o crawlers, que navegan por los sitios web, extrayendo la información relevante y almacenándola en una base de datos. Según (Carvalho *et al.*, 2020), el proceso de web scraping puede ser automatizado utilizando diferentes herramientas y lenguajes de programación, como Python, R y JavaScript, entre otros. Además, es importante destacar que el uso de web scraping puede tener implicaciones legales y éticas, por lo que se debe tener cuidado en su aplicación y asegurarse de cumplir con las normativas pertinentes.

En resumen, el web scraping es una técnica valiosa en el análisis de datos que permite la recopilación automatizada de información de sitios web. Esta técnica puede ser utilizada para diferentes propósitos y se puede automatizar utilizando diferentes herramientas y lenguajes de programación. Es importante tener en cuenta las implicaciones legales y éticas del uso de web scraping y asegurarse de cumplir con las normativas correspondientes.

4.6 Machine Learning ("Aprendizaje automático")

El aprendizaje automático o machine learning se define como una rama de la inteligencia artificial que permite a los sistemas informáticos mejorar su desempeño en una tarea T y una medida de rendimiento P, a través de la experiencia E, sin necesidad de programación explícita para cada tarea. Esta tecnología es esencial para predecir los precios de las propiedades ya que puede analizar grandes conjuntos de datos y detectar patrones complejos, lo que sería difícil para los seres humanos (Mitchell *et al.*, 2007).

Además, el machine learning permite construir modelos predictivos precisos y personalizados para cada caso, lo que proporciona una toma de decisiones más informada y precisa. Según autores como (Hastie *et al.*, 2009), el aprendizaje supervisado es especialmente útil para la predicción de precios de inmuebles, utilizando modelos de regresión para predecir los valores de precios en función de las características de las propiedades.

Por otra parte, el aprendizaje no supervisado, como destaca (Breiman, 2001), es importante para detectar patrones y anomalías en grandes conjuntos de datos, lo que puede ayudar en la identificación de áreas o zonas donde los precios de las propiedades podrían ser más bajos o más altos que el promedio. En resumen, el machine learning es una herramienta fundamental para la predicción de precios de propiedades debido a su capacidad para analizar grandes conjuntos de datos y detectar patrones complejos, lo que permite construir modelos predictivos precisos y personalizados para cada caso.

4.7 Aprendizaje supervisado

El aprendizaje supervisado es un tipo de algoritmo de aprendizaje automático que utiliza un conjunto de datos etiquetados para entrenar un modelo que pueda predecir etiquetas para datos no etiquetados. Según (Hastie *et al.*, 2009), el aprendizaje supervisado se refiere a la tarea de aprender una función que mapee una entrada a una salida deseada, a partir de ejemplos de entrenamiento que consisten en pares de entrada-salida.

En el contexto de la predicción de precios de inmuebles, el aprendizaje supervisado es importante porque permite entrenar un modelo que pueda predecir el precio de una propiedad en función de sus características. Para ello, es necesario contar con un conjunto de datos de entrenamiento etiquetado que contenga información sobre las características de las propiedades y sus precios de venta.

Según (Wu *et al.*, 2018) , el aprendizaje supervisado ha sido ampliamente utilizado en la predicción de precios de inmuebles, especialmente en la industria inmobiliaria y el sector financiero. En particular, los modelos de regresión son comúnmente utilizados en la predicción de precios de inmuebles, ya que permiten predecir un valor numérico continuo como el precio de venta.

En resumen, el aprendizaje supervisado es un enfoque importante para la predicción de precios de inmuebles, ya que permite entrenar un modelo para predecir el precio de una propiedad en función de sus características. Los modelos de regresión son comúnmente utilizados en esta tarea y el aprendizaje supervisado ha sido ampliamente utilizado en la industria inmobiliaria y el sector financiero para la predicción de precios de inmuebles.

4.8 Matemáticas del Machine Learning

En este capítulo se dan a conocer las matemáticas que aglutinan un proyecto de datos basado en técnicas de machine learning. Las cuales están presentes en el ajustes de modelos y elección del modelo. Donde se exponen los materiales y métodos que hacen posible el desarrollo de dichas etapas de forma teórica matemática.

4.8.1 Construcción de modelos de machine learning

4.8.2. Random Forest

Random Forest es un algoritmo basado en técnicas de aprendizaje automático, que tiene la necesidad de disponer un conjunto de datos que posee información de la variable respuesta (aprendizaje supervisado). Para ello se divide de forma aleatoria el conjunto de datos (data frame) en dos subconjuntos; uno de entrenamiento y otro de test, siendo mayor el subconjunto de entrenamiento.

En efecto ambos subconjuntos poseen variables (X, Y) donde:

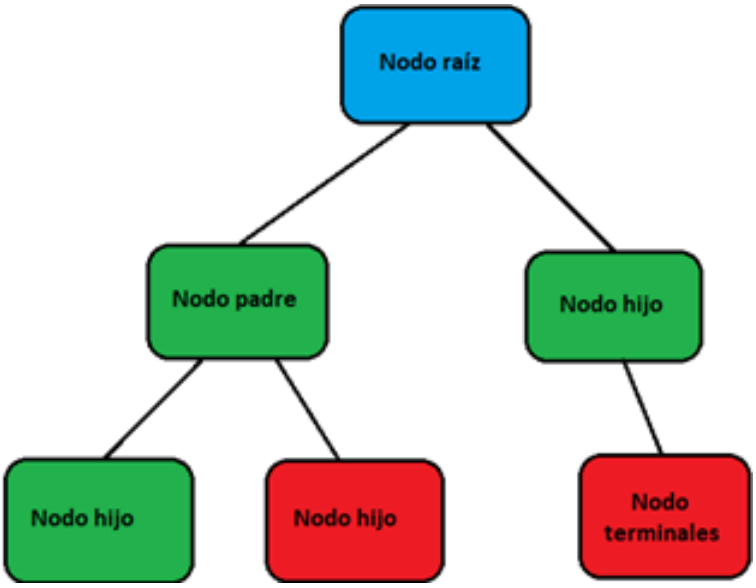
X = Variables predictoras (independientes) dado $X = x_1, x_2, x_3, \ldots, x_i \; i = 1, 2, 3 \ldots n.$

Y = Variable respuesta (dependientes) dado $Y = y$

El algoritmo está formado por un conjunto de árboles predictores individuales, los cuales son grafos conexos acíclicos dirigidos formados por un conjunto finito de nodos conectados. A continuación, (Ejea Carbonell y Alcalá Nalvaiz, 2017) define las principales características asociadas con su presentación gráfica de la Figura 1.

- Un nodo es la unidad sobre la que se construye un árbol.
- Un nodo puede tener varios nodos a los que apunte llamados nodos hijos.
- Se dice que un nodo es un nodo padre si posee al menos un nodo hijo.
- Al nodo sin antecesores se le llama nodo raíz y es único.
- Hermanos son aquellos nodos que comparten el mismo padre.
- Los nodos terminales u hojas son aquellos que no tiene ramificaciones o nodos hijos.
- Rama es el camino (unión de arcos simples) desde la raíz a un nodo hoja.
- El grado de un nodo es el número de descendientes directos que posee, y su nivel el número de arcos que han de recorrerse de la raíz, a la cual se le presupone nivel 1.
- El grado de un árbol es el máximo de los grados de sus nodos, y su altura el máximo de los niveles. (p.5)

Figura 1: Esquema de un árbol predictor.



Cada uno de los árboles están entrenados con una muestra aleatoria extraída del subconjunto de datos de entrenamiento con ayuda del procedimiento "Bootstrap". El cual define según (Ejea Carbonell y Alcalá Nalvaiz, 2017), como la creación de un número B de muestras con reemplazamiento del mismo tamaño que el conjunto inicial y ajusta un estadístico para cada una de ellas.

En cada iteración Bootstrap, se utiliza cada vez aproximadamente $\frac{2}{3}$ de los datos para ajustar el árbol correspondiente y el tercio restante se considera observaciones out of bag (OBB),

es decir, las muestras no seleccionadas para crear árboles. Este resultado viene de tomar el límite.

$$\lim_{n\to\infty}\left(1-\frac{1}{n}\right)^{n}=e^{-1}\approx 0,36 \tag{1}$$

Pues $\left(1-\frac{1}{n}\right)^{n}$ es la probabilidad de que un dato no se seleccione por una muestra Bootstrap de tamaño n cuando $n\to\infty$

Sabiendo las variables predictoras que tenemos, se realiza una selección de muestro aleatorio del número de variables predictoras a considerar en cada uno de los árboles. Este método se basa en la raíz cuadrada del total de variables predictores.

$$m=\sqrt{p} \tag{2}$$

Donde

m = Número de variables predictoras consideradas en cada división.

p = Total de variables predictoras.

Finalmente, las predicciones de los diferentes árboles se promedian, obteniendo como resultado una predicción general que sea más precisa que los árboles individuales, esto debido a la poca varianza que se consigue en el modelo, ya que Random Forest permite romper con la correlación entre los árboles generados en el proceso, haciendo una selección de m predictores antes de evaluar cada división. De modo que, un predictor influyente no será elegido en las divisiones de los nodos en un promedio de $\frac{p-m}{p}$, permitiendo que otros predictores puedan ser seleccionados.

A continuación, se demuestra la reducción de la varianza cuando existe un incremento en el número de árboles de decisión.

Supongamos que tenemos M árboles de decisión, cada uno representado por un modelo $M=\{M_1,M_2,M_3,\cdots,M_i\}$, donde cada uno de ellos va a tener asociada una salida $Y=\{y_1,y_2,y_3,\cdots,y_i\}$, y esa salida va a tener asociada a su vez una media (μ) y una varianza (σ) suponiendo que será la misma para todos los árboles de decisión.

Ahora vamos a ver qué sucede cuando tomamos el valor promedio o el valor esperado de todo el conjunto de árboles de decisión pensado como un ensamble.

Entonces si calculamos el valor esperado como;

$$E\left[\frac{1}{M}\sum_{i=1}^{M}y_i\right]=\frac{1}{M}\sum_{i=1}^{M}E\left[y_i\right]=\frac{1}{M}\cdot M\cdot\mu=\mu \tag{3}$$

Resultado evidente, el cual tenemos M modelos cada uno con su promedio y tomamos el valor esperado de todos ellos obteniendo μ.

Ahora veamos qué pasa con la varianza;

$$Var\left[\frac{1}{M}\sum_{i=1}^{M} y_i\right] = \frac{1}{M^2}\sum_{i=1}^{M} Var\left(y_i\right) = \frac{1}{M^2}\cdot M\cdot\sigma^2 = \frac{\sigma^2}{M} \tag{4}$$

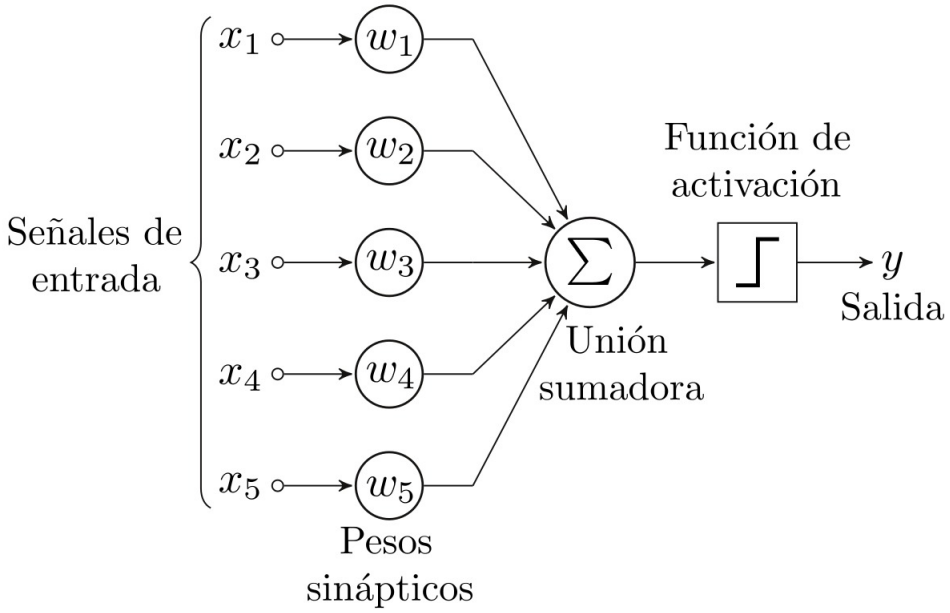
De esta manera, considerando muchos árboles que tienen las mismas características y distribuciones podemos concluir, que la varianza total del conjunto o ensamble es inferior a la varianza de cada uno de los árboles de decisión pensados por separados.

4.8.3 Red Neuronal profunda

Una red neuronal profunda al igual que random forest, es un algoritmo basado en técnicas de aprendizaje automático, del tipo "aprendizaje automático supervisado". El algoritmo consigue modelar neuronas biológicas del cerebro, de esas con masa redonda, un núcleo y unas ramificaciones. Dichas neuronas artificiales se les conoce como perceptrón, y, por tanto, una unidad neuronal. El perceptrón efectúa cálculos para detectar características o tendencias en los datos de entrada.

En un perceptrón podemos distinguir una entrada de señales, un nodo y una salida, tal como lo muestra la Figura 2. A continuación, veremos cómo funciona un perceptrón del punto de vista teórico matemático.

Figura 2: Perceptrón o neurona artificial.



Nota: Adaptado de Rodrigo, J.(2021). Redes Neuronales con Python.[Figura].(https://www.cienciadedatos.net/documentos/py35-redes-neuronales-python.html)

Las señales de entrada, es decir, la información que recibe nuestro perceptrón, son variables predictoras (independientes). Los n-valores de entrada son multiplicados por sus respectivos pesos, dicho de otra manera, el vector entrada es multiplicado por el vector peso, teniendo como resultado una combinación lineal de las n-valores de entrada y los pesos; algo que denominamos función de ponderación.

$$X * W^t = (x_1, x_2, x_3, \ldots, x_n) * \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{pmatrix} = \sum_{i=1}^{n} x_i * w_i \tag{5}$$

Después de obtener nuestra función de ponderación, generamos información la cual será transmitida a las conexión de salida por medio de una función de activación. La función de activación según (Rodrigo, 2021) modifica el valor resultado o impone un límite que se debe sobrepasar para ser propagado a la salida. Las funciones de activación más conocidas o más usadas son:

- **Función Escalón, (similar a la función binaria)**.

  Es una función discontinua cuyo valor es 0 para cualquier argumento negativo, y 1 para cualquier argumento positivo.

$$\Phi(x) = \begin{cases} 0 \; si \; x < 0 \\ 1 \; si \; x \; \geq 0 \end{cases} \tag{6}$$

- **Función Sigmoidal**.

  Función en forma de "S" que transforma los valores introducidos a una escala (0,1), donde los valores altos tienen manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0 (Rodrigo, 2021).

$$\Phi(x) = \frac{1}{1 + e^{-x}} \tag{7}$$

- **Función Rectificadora (Relu)**.

  Función que permite el paso de todos los valores positivos sin cambiarlos, pero asigna todos los valores negativos a cero (Rodrigo, 2021).

$$\Phi(x) = max(0, x), \; siendo \; x \geq 0 \tag{8}$$

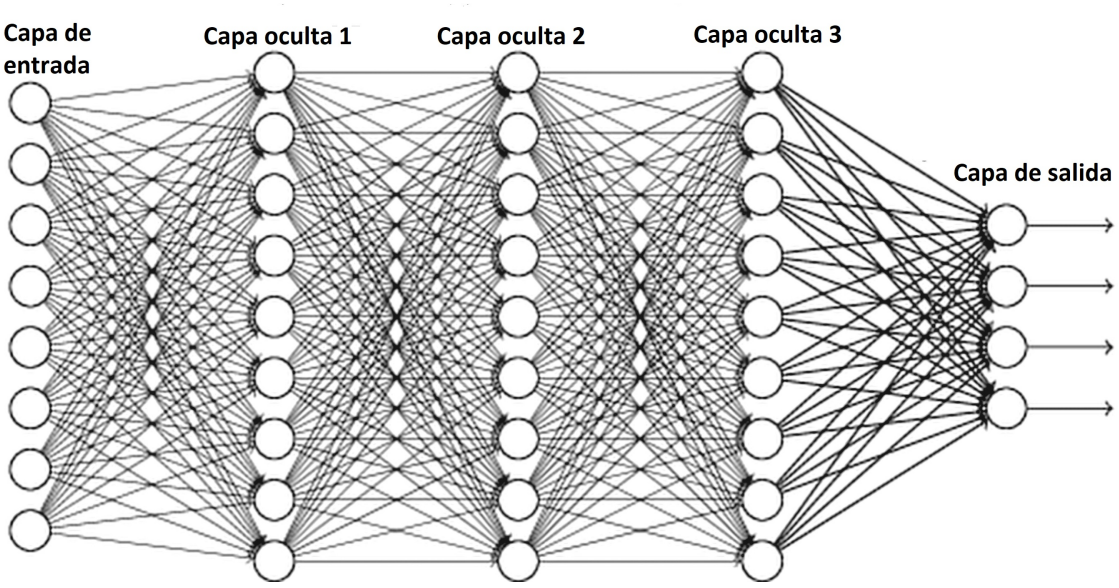
- **Función Tangente Hiperbólica**.

  La función tangente hiperbólica transforma los valores introducidos a una escala (-1,1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a -1 (Rodrigo, 2021).

$$\Phi(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{9}$$

Ahora que se ha terminado de describir una neurona o perceptrón. Podemos definir, una red neuronal como una serie de capas de neuronas. Específicamente, todas las neuronas de una capa se conectan a las neuronas de la siguiente capa tal como lo muestra la Figura 3.

Figura 3: Esquema de redes neuronales.



Nota: Adaptado de Rodrigo, J.(2021). Redes Neuronales con Python.[Figura].(https://www.cienciadedatos.net/documentos/py35-redes-neuronales-python.html)

Una red neuronal aprende relaciones mucho más complejas entre los predictores y la variable respuesta, superando así, las limitaciones que presentan trabajar con una sola neurona o perceptrón. Ya que un perceptrón, solo sirve para clasificar problemas linealmente separables, cosa que ya se podía hacer mediante métodos estadísticos y de una forma mucho más eficiente.

Como se ha mencionado anteriormente, el proceso de aprendizaje, es el verdadero potencial de este algoritmo. Lo cual consiste en ajustar el valor de los pesos y bías de tal forma que, las

predicciones que se generen, tengan el menor error posible. Gracias a esto, el modelo tiene la posibilidad de darle importancia relativamente distinta a cada entrada, afectando pues a su propia salida y, en definitiva, a la salida global de la red neuronal dada una serie de entradas.

La idea del proceso de aprendizaje parece ser sencilla, para ello se requiere la combinación de métodos matemáticos, conocidos como, el algoritmo de retropropagación (backpropagation) y la optimización por descenso de gradiente (gradient descent).

El método Backpropagation permiten cuantificar la influencia que tiene cada peso y bías de la red en sus predicciones, haciendo uso de la regla de la cadena, el cual, de manera resumida, es la derivada parcial del error respecto a un parámetro (peso o bías), donde se mide cuanta responsabilidad ha tenido ese parámetro en el error cometido. Una vez calculado el error respecto a un parámetro (peso o bías), el descenso de gradiente (gradient descent), es el algoritmo optimizador que permite determinar cuánto y cómo modificar los pesos y bías del modelo para reducir su error. Estos dos procesos se repiten hasta que la red sea suficientemente buena.

4.8.4. XGBoost

El algoritmo de XGBOOST, como nos explica (Espinosa, 2020) consiste en un ensamblado secuencial de árboles de decisión. Conocido como Classification and Regression Trees.

Los árboles funcionan de manera que cada árbol se agrega secuencialmente con el fin de aprender del resultado de los árboles previos y corregir el error producido por estos mismos, hasta llegar al resultado de que ya no se pueda corregir más este error. De este modo se aplica el "gradiente descendiente". A diferencia del aumento del gradiente, el XGBOOST utiliza una expansión de Taylor para aproximar la función de pérdida, y el modelo tiene un mejor sesgo y varianza de compensación, generalmente usando menos árboles de decisión con el fin de obtener una mayor precisión. Para encontrar más sobre las ecuaciones que rigen este proceso matemáticamente se puede encontrar en el artículo de (Qin *et al.*, 2021).

Luego, se tiene un funcionamiento del XGBOOST expuesto por (Espinosa, 2020) , como en los siguientes pasos:

1. Se obtiene un árbol inicial F_0 para predecir la variable objetivo "y", el resultado se asocia con un residual $(y - F_0)$

2. Se obtiene un nuevo árbol h_1 que ajusta el error del paso previo.

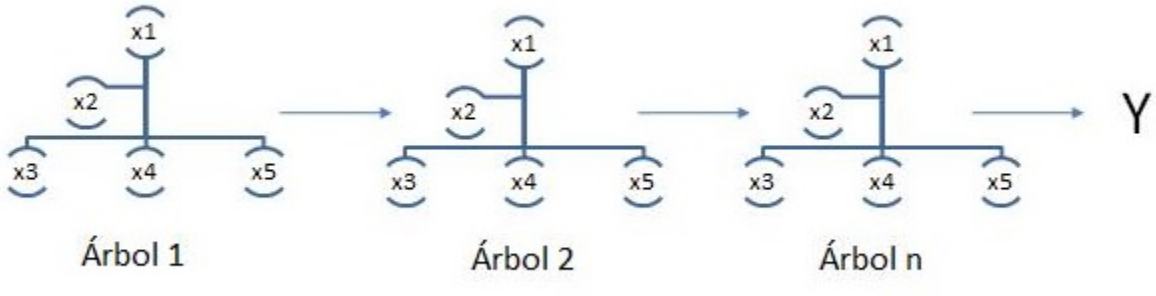
3. Los resultados de F_0 y h_1 se combinan para obtener el árbol F_1, donde el error cuadrático medio de F_1 será menor que el de F_0

$$F_1(x) < -F_0(x) + h_1(x) \tag{10}$$

4. Este proceso se sigue iterativamente hasta que el error es minimizado lo más posible de la siguiente forma:

$$F_m(x) < -F_{m-1}(x) + h_m(x) \tag{11}$$

Figura 4: Algoritmo XGBoost



Nota: Adaptado de Espinosa, J.(2020). Aplicación de Algoritmos Random forest y XGBoost en una base de solicitudes de tarjetas de crédito.[Figura].(https://www.revistaingenieria.unam.mx/numeros/v21n3-02.php)

De modo que, en este algoritmo los árboles de decisión se crean de forma secuencial como se puede observar en la Figura 4.Además, los pesos juegan un papel importante en XGBOOST. Se asignan pesos a todas las variables independientes que luego se introducen en el árbol de decisión que predice resultados. El peso de las variables predichas incorrectamente por el árbol se incrementa y estas variables luego se alimentan de un segundo árbol de decisión.

Estos clasificadores o predictores individuales luego se ensamblan para dar a un modelo sólido y más preciso. Lo mejor de XGBOOST es que aborda de forma inteligente estos dos problemas. Se debe tener en cuenta que se puede usar la función, base alumno, árbol indistintamente.

Por último, describiendo el análisis de (Espinosa, 2020), el XGBOOST utiliza una extensión de árboles definidos por el usuario y que se deben definir correctamente los parámetros para evitar un sobreajuste en el modelo.

4.9. Métricas de desempeño

4.9.1 Coeficiente de determinación

Según (González, 2018) el coeficiente de determinación (R^2) indica el ajuste o la bondad de un modelo y se utiliza a menudo con fines descriptivos, mostrando que la variable in-

dependiente elegida también explica la variabilidad de su variable dependiente R^2 se define como:

$$R^2 = 1 - \frac{\sum_{i=1}^{M} (real_i - estimado_i)^2}{\sum_{i=!}^{M} (real_i - estimado_i)^2} \tag{12}$$

El coeficiente de determinación tiene la útil propiedad de que su escala es intuitiva y va de 0 a 1, donde 0 indica que el modelo propuesto no mejora las predicciones del modelo promedio y 1 indica predicciones perfectas. Las mejoras en el modelo de regresión dan como resultado un aumento proporcional en R^2

4.9.2. Error cuadrático medio

Este criterio de evaluación, como lo define (J. Martínez, 2020), es usado regularmente para problemas de regresión. Sobre todo, en aprendizaje automático supervisado.

Este error, es básicamente la raíz cuadrada de la diferencia entre el valor real y el valor estimado, es decir la ecuación que la rige es:

$$\sqrt{error\ cuadratico} = \sqrt{(real\ -\ estimado)^2} \tag{13}$$

Siendo así, se calcula el error medio en cada punto. Luego se llama M al número total de puntos y nos queda la fórmula del Error Cuadrático (MSE) y su ecuación es

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (real_i - estimado_i)^2} \tag{14}$$

4.9.3. Validación cruzada

La validación cruzada es una técnica para evaluar modelos de machine learning. Técnica que ayuda a comparar y seleccionar el modelo de machine learning mediante la base de entrenamiento, para entrenar y evaluar al mismo tiempo(Rodrigo, 2020) .

Consiste en dividir la tabla de datos en k partes iguales. Donde se dejará uno de los pliegues para la validación y los pliegues restantes se dejarán para el entrenamiento del modelo. Este proceso es iterativo, hasta que todos los pliegues sean utilizados como validación.

En cada proceso iterativo, se calcula el desempeño del modelo. Para que por último promediemos los desempeños y así obtener el desempeño esperado del modelo.

4.9.4. Criterio AIC

El criterio AIC (Akaike Information Criterion) es una medida de la calidad del ajuste de un modelo estadístico. El AIC es una herramienta comúnmente utilizada en el campo del aprendizaje automático para seleccionar el mejor modelo entre un conjunto de modelos candidatos.

El AIC se calcula como,

$$AIC = -2log(L) + 2k \tag{15}$$

Donde L es la función de verosimilitud del modelo y k es el número de parámetros en el modelo. El modelo con el valor más bajo de AIC se considera el mejor modelo.

El uso del criterio AIC en modelos de aprendizaje automático se ha discutido en varias publicaciones científicas. Por ejemplo, en un artículo de 2004, (Konishi y Kitagawa, 2008) destacan la importancia del AIC en la selección del modelo y lo comparan con otros criterios de selección de modelos.

Además, en un artículo de 2004, (Anderson y Burnham, 2004) discuten la teoría detrás del criterio AIC y lo aplican a la selección del modelo en diferentes disciplinas científicas. También señalan que el criterio AIC es ampliamente utilizado y preferido por los investigadores debido a su capacidad para seleccionar modelos más simples con un buen ajuste.

En conclusión, el criterio AIC es una herramienta valiosa para seleccionar el mejor modelo en el aprendizaje automático y ha sido ampliamente discutido en la literatura científica.

4.9.5. Criterio BIC

El criterio BIC (Bayesian Information Criterion), también conocido como criterio de Schwartz, es una herramienta útil en la selección de modelos en el aprendizaje automático. Este criterio se utiliza para comparar diferentes modelos estadísticos y determinar cuál de ellos es el más probable, dadas las observaciones y los datos.

Según (Akaike, 1973), el criterio BIC es una medida de la complejidad del modelo y de su ajuste a los datos. El criterio BIC se calcula a partir del logaritmo de la función de verosimilitud, el número de parámetros del modelo y el tamaño de la muestra:

$$BIC = -2log(L) + klog(n) \tag{16}$$

Donde L es la función de verosimilitud, k es el número de parámetros y n es el tamaño de la muestra. El modelo con el valor más bajo de BIC se considera el modelo más probable.
El criterio BIC tiene una interpretación intuitiva: penaliza los modelos con muchos parámetros y recompensa los modelos que se ajustan bien a los datos con pocos parámetros. De esta manera, evita el sobreajuste y selecciona el modelo que mejor representa la complejidad subyacente en los datos.

El criterio BIC es ampliamente utilizado en la selección de modelos de aprendizaje automático. Por ejemplo, en la selección de modelos de regresión lineal, modelos de mezcla de Gaussianas y modelos de redes neuronales (Ripley, 2007).

En conclusión, el criterio BIC es un enfoque estadístico eficaz para seleccionar modelos de aprendizaje automático. El uso de esta herramienta ayuda a encontrar el modelo más probable y evita el sobreajuste en los datos.

4.9.6. Criterio DIC

El criterio de información bayesiano deviance (DIC, por sus siglas en inglés) es una medida utilizada para comparar modelos de machine learning. Este criterio se basa en la idea de que un modelo mejor es aquel que logra ajustarse adecuadamente a los datos observados, pero que a su vez no es demasiado complejo.

La fórmula del DIC se puede expresar como:

$$DIC = -2 * log(p(D|M)) + 2pD \tag{17}$$

Donde $p(D|M)$ es la probabilidad posterior predictiva de los datos bajo el modelo M y pD es el número efectivo de parámetros del modelo, que se puede calcular como la diferencia entre el valor medio de la deviance y la deviance del modelo ajustado.

El DIC es una medida útil para la selección de modelos en diversos campos, como la biología, la epidemiología, la ingeniería y las ciencias sociales. Sin embargo, es importante tener en cuenta que el DIC no es perfecto y puede presentar limitaciones en ciertas situaciones.

Según (Spiegelhalter *et al.*, 2002), el DIC es una medida que combina tanto la capacidad predictiva del modelo como su complejidad, lo que lo hace útil para la comparación de modelos en términos de capacidad predictiva. Por otro lado,(Lee y Wagenmakers, 2014) argumentan que el DIC puede ser demasiado sensible a la elección de la distribución previa, lo que puede afectar su capacidad para comparar modelos.

En cualquier caso, el DIC es una herramienta útil para la selección de modelos en el ámbito del machine learning y ha sido ampliamente utilizado en la literatura científica. Además, existen diversas variantes del DIC, como el DIC bayesiano y el DIC centrado, que pueden adaptarse a diferentes situaciones y necesidades (Celeux *et al.*, 2006) (Gelman *et al.*, 2014) .

En resumen, el criterio DIC es una medida útil para la comparación de modelos de machine learning en términos de capacidad predictiva y complejidad. Aunque presenta ciertas limitaciones, su uso está ampliamente extendido en la literatura científica y es una herramienta valiosa para la selección de modelos en diversos campos.

4.10. Marco legal

4.10.1 Decreto 1420 del 24 de julio de 1998

Por el cual se reglamentan el artículo 37 de la Ley 9 de 1989, el artículo 27 del Decreto-ley 2150 de 1995, los artículos 56, 61, 62, 67, 75, 76, 77, 80, 82, 84 y 87 de la Ley 388 de 1997 y el artículo 11 del Decreto-Ley 151 de 1998, que hacen referencia al tema de avalúos.

4.10.2 Resolución IGAC 620 del 23 de septiembre de 2008

Por la cual se establecen los procedimientos para los avalúos ordenados dentro del marco de la Ley 388 de 1997.

4.10.3 Ley 1673 del 19 de julio de 2013

Por la cual se reglamenta actividad del avaluador y se dictan otras disposiciones.

4.10.4 Decreto 556 del 14 de marzo de del 2014

Por el cual se reglamenta la Ley 1673 del 2013.

4.10.5 Ley 388 del 18 de julio de 1997

Por la cual se modifica la Ley 9 de 1989, y la Ley 3 de 1991 y se dictan otras disposiciones.

4.10.6. Norma Técnica Sectorial Colombiana NTS I 01

Contenido de Informes de Valuación de Bienes Inmuebles Urbanos.

Capítulo 5

5. Objetivos

5.1 Objetivo General

- Ajustar un modelo para determinar los valores comerciales de los inmuebles en la ciudad de Neiva en el año 2022, a partir de técnicas de aprendizaje de máquinas y extracción de datos de páginas web inmobiliarias.

5.2 Objetivos Específicos

- Generar un conjunto de datos a partir de la recolección, análisis y transformación de la información encontrada en las diferentes páginas web del mercado inmobiliario.
- Proponer el ajuste de ciertos modelos (Random Forest, XGBoost y Redes Neuronales Profundas) de acuerdo al comportamiento natural de los datos.
- Evaluar la calidad del modelo en términos de métricas de error, para la determinación del valor comercial de los inmuebles en Neiva.

Capítulo 6

6. Metodología

Para la construcción del modelo de machine learning, primero se investigó el proceso que los peritos o avaluadores realizan para determinar el valor comercial de un inmueble, el cual es uno de los métodos más utilizados y que está registrado en la resolución IGAC 620 de 2008, llamado método comparativo o de mercado que consiste en el estudio de las ofertas o transacciones recientes, de bienes semejantes y comparables al inmueble que se le quiere determinar el valor comercial.

Basado en lo anterior, se realizó una metodología a partir del flujo de actividades que hay en un proyecto de datos, siendo un total de 5 actividades como se muestran a continuación:

6.1 Recolección y extracción de los datos

Debido a que la ciudad de Neiva no cuenta con bases de datos al público sobre ofertas o transacciones recientes de inmuebles, se decide utilizar las páginas web inmobiliarias (Buriticá, Félix Trujillo y Finca raíz) para la recolección de los datos.

Dicha tarea se decidió ejecutar de forma automatizada ya que la cantidad de inmuebles que se presentaron en la recolección, sumándole sus características, hizo que dicha tarea al ejecutarla de forma manual conllevara mucho tiempo, además de ser tediosa e inoficiosa. Por lo cual nos apoyamos en el método automatizado de recolección web, llamado "web scraping". Lo cual, antes de aplicarlo, se debe tener en cuenta aspectos cómo: accesibilidad de los datos de origen y análisis de patrones de los datos.

Lo que en efecto surgen pasos para la recolección, donde a continuación se exponen:

a) Se indagó las tres páginas inmobiliarias con el objetivo de ubicar los datos que queríamos recolectar. Donde para ello, se realizó una indagación de forma paralela con la interfaz que ofrece las páginas al público; y la interfaz programada en niveles múltiples de etiquetas HTML. Con fin de encontrar las rutas de código que contienen los datos necesarios para la recolección.

Durante la indagación pudimos visualizar que entre las páginas inmobiliarias elegidas no coinciden todos los datos que se puede presentar en un inmueble, como también errores de edición y datos nulos. Donde más adelante tendríamos que decidir en descartar las páginas web que no nos brindan los suficiente datos para la recolección.

b) Como segundo paso para llevar a cabo el método de recolección web scraping en las tres páginas inmobiliarias, se utilizó el lenguaje de programación Python con ayuda de las librerías webdriver, selenium y time las cuales permite realizar tareas automatizadas de páginas web desarrolladas en lenguajes HTML, además se utilizó la librería pandas con el

objetivo de la recopilación de la información en archivo XLSX.

c) Por consiguiente, se desarrolló algoritmos los cuales obtienen las múltiples direcciones web de los inmuebles (casa y apartamento) en venta que aparecen en las tres páginas inmobiliarias por separado, ya que cada página web inmobiliaria viene estructurada de manera diferente. De los cuales se obtuvo en las inmobiliarias Buriticá, Félix Trujillo y Finca raíz: 80, 194 y 383 respectivamente, guardándose por archivos separados las direcciones web que le pertenecen a cada inmobiliaria.

d) Por último, se desarrollan nuevos algoritmos con la tarea de ingresar a cada dirección web obtenida anteriormente e ir recolectando los datos publicados de los inmuebles, los cuales se presentaron datos de características tipo terreno y localización que son: dirección, barrio, estrato y precio por metro cuadrado. Como también datos de características tipo construcción del inmueble que son: número de habitaciones, número de baños, número de parqueaderos, área construida, área privada, antigüedad, administración, piso, tipo de inmueble, estado del inmueble y por último valor del inmueble.

Durante el proceso, se presentan algunas complicaciones. Las cuales mencionaremos a continuación, tomando decisiones al respecto:

- Las direcciones web pertenecientes a la página inmobiliaria Buriticá mostraron errores de digitación en los datos tipo terreno, localización y construcción mencionados anteriormente. Los cuales afectan en los patrones de búsqueda del web scraping, como también datos nulos que son importante y no reemplazables, por ejemplo: dirección, estrato, número de habitaciones y valor del inmueble. Por lo que se decide no seguir trabajando con esta página web la recolección y tampoco la extracción para la construcción de la base de datos.

- Pudimos evidenciar que Finca raíz es un sitio web inmobiliario que ofrece una recopilación de todos los inmuebles publicados por las tantas páginas inmobiliarias de la ciudad de Neiva, la cual entraría Félix Trujillo que es una página local y entre otras más. Por lo que solo se decidió trabajar con Finca raíz y no seguir con Félix Trujillo y otras páginas que ofrecen inmuebles en Neiva. Evitando que se presentaran datos repetidos.

- En el sitio web Finca raíz, las características (número de parqueaderos, administración, antigüedad, tipo de inmueble, piso y estado del inmueble) presentaron irregularidades. Dado que, si el inmueble no contaba con parqueadero o no pagaba administración y si en algunos casos el vendedor no especificaba la antigüedad, piso o el estado del inmueble, se evidenciaba en la página web la ausencia de estas características. Lo que en consecuencia afectaba el algoritmo de recolección y también de ser imposible recopilar la información en una tabla. Por lo que la solución en ese momento fue guardar todas las características en una columna.

  Damos por terminado el proceso de recolección de datos a los 383 inmuebles del sitio web finca raíz. Por lo que se obtiene un archivo xlsx con las siguientes columnas:

número de inmueble, link, características, precio, dirección y barrio que se muestra en la Figura 5.

La cual se le aplicara la otra parte de esta actividad llamada extracción de datos.

	link	caracteristicas	precio	direccion	barrio
0	https://www.fir	[['Habitaciones', '3'], ['Baños', '5'], ['I	585.000.000	CONDOMINIO RESERVA D	CONJUNTO RESERV/
1	https://www.fir	[['Habitaciones', '3'], ['Baños', '2'], ['I	120.000.000	CALLE 22 1A 07	CONJUNTO CERRAD
2	https://www.fir	[['Habitaciones', '3'], ['Baños', '3'], ['I	650.000.000	CALLE 8 No. 52-149 COND	ipanema - Neiva
3	https://www.fir	[['Habitaciones', '5'], ['Baños', '3'], ['I	205.000.000	calle 22 #43 @58	Los Colores - Neiva
4	https://www.fir	[['Habitaciones', '3'], ['Baños', '3'], ['I	40.500.000	calle 46 Sur # 33-178	CIUDADELA MESOP(
5	https://www.fir	[['Habitaciones', '2'], ['Baños', '2'], ['I	160.000.000	VILLA MAGDALENA CALLE	VILLA MAGDALENA
6	https://www.fir	[['Habitaciones', '3'], ['Baños', '3'], ['I	270.000.000	Calle 6C #24A-74	Barrio la Gaitana - N
7	https://www.fir	[['Habitaciones', '5'], ['Baños', '3'], ['I	140.000.000	BARRIO LIMONAR CARRE	limonar - Neiva
8	https://www.fir	[['Habitaciones', '3'], ['Baños', '2'], ['I	320.000.000	EDIFICIO TORRE 8-34 APT	ipanema - Neiva
9	https://www.fir	[['Habitaciones', '3'], ['Baños', '2'], ['I	300.000.000	CONDOMINIO AMARANT	AMARANTO CLUB H
10	https://www.fir	[['Habitaciones', '3'], ['Baños', '4'], ['I	620.000.000	CONDOMINIO RESERVA D	ipanema - Neiva
11	https://www.fir	[['Habitaciones', '3'], ['Baños', '3'], ['I	220.000.000	Calle 8b #38-50	Barrio Ipanema Nei
12	https://www.fir	[['Habitaciones', '3'], ['Baños', '1'], ['I	85.000.000	Cra 36#30-36sur	IV CENTENARIO LAD
13	https://www.fir	[['Habitaciones', '3'], ['Baños', '1'], ['I	85.000.000	Cra 36#30-36 SUR	IV CENTENARIO LAD
14	https://www.fir	[['Habitaciones', '3'], ['Baños', '1'], ['I	104.000.000	calle 25 A No. 36-68	Parque residencial (
15	https://www.fir	[['Habitaciones', '3'], ['Baños', '3'], ['I	520.000.000	CALLE 8 No. 52-45 CONJU	CAMINOS DE ORIEN
16	https://www.fir	[['Habitaciones', '5'], ['Baños', '5'], ['I	620.000.000	Codigo: 5204525. Código	SEVILLA - Neiva
17	https://www.fir	[['Habitaciones', '3'], ['Baños', '2'], ['I	280.000.000	Calle 56 No. 17 71	CONDOMINIO AMA
18	https://www.fir	[['Habitaciones', '3'], ['Baños', '2'], ['/	130.000.000	Calle 76 c#2w	CALAMARI - Neiva
19	https://www.fir	[['Habitaciones', '5'], ['Baños', '4'], ['I	226.500.000	CONJUNTO RESIDENCIAL	bosques de Tamarir

Figura 5: Archivo xlsx de los 383 inmuebles recolectados.

Teniendo en cuenta el archivo final que se obtuvo mediante la recolección de datos en los inmuebles del sitio web Finca raíz, se procedió a la extracción de datos necesarios para la construcción del data frame.

Para emplear dicha actividad se utilizó la minería de datos que consiste en encontrar patrones en grandes volúmenes de conjuntos de datos. Lo cual es necesario para solucionar la complicación que manteníamos hasta el momento con los datos recolectados.

El lenguaje de programación Python con las librerías pandas y replace fueron las herramientas que se utilizaron para completar con éxito dicha actividad. Donde a continuación, se explica el proceso:

Dado que el archivo de recolección, algunas características de los inmuebles estaban guardadas en una sola columna, se procedió a separarlas en columnas diferentes. Para ello se sube el archivo de recolección en Python convirtiendo las columnas existentes en listas y guardándolas por el nombre establecido anteriormente, excepto la columna características la cual se llamó propiedades.

Según lo realizado, pudimos notar que en la lista llamada propiedades, cada elemento de dicha lista contenía todas las características de cada inmueble. Donde esas características

Python las guardo como cadenas de caracteres que representan texto, llamado formato tipo string; tal como lo muestra la Figura 6. Además, cada elemento en formato tipo string estaban acompañados por caracteres especiales como comillas simples y corchetes, donde esos caracteres especiales debían ser eliminados; ya que lo que interesaba era lo que estaba por dentro de ellos que son la recolección de las características de los inmuebles.

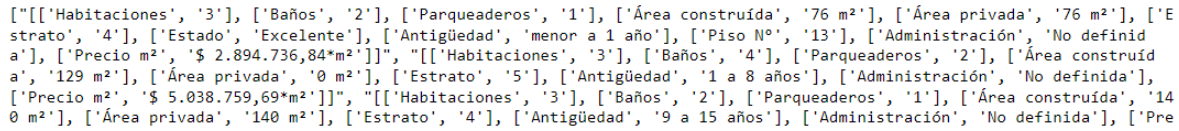


Figura 6: Extracción de la información datos tipo string.

Por lo que en Python se estableció dos patrones de selección: el primer patrón corresponde a la selección de los corchetes y el segundo patrón a la selección de las comillas simples. Por consiguiente, se construyó un algoritmo que desempeña 3 ejecuciones:

- La primera ejecución consistió en recorrer cada elemento de la lista propiedades aplicando el primer patrón establecido, lo cual se selecciona los corchetes y se van eliminando.
- La segunda ejecución consistió en volver a recorrer cada elemento de la lista propiedades aplicando el segundo patrón, lo cual seleccionó las comillas simples que se presentaron y las fue eliminando. Además, se guardó en listas por inmueble todas las características que se iban extrayendo en otra lista creada llamada características.
- La tercera ejecución se creó otra lista llamada objetos, donde a medida que terminaba la ejecución dos, las características guardadas en listas por inmueble con su nombre y valor; ya sea cualitativo o cuantitativo, se fueran guardando en sub listas por separado dentro de la lista objetos, tal como lo muestra la Figura 7.

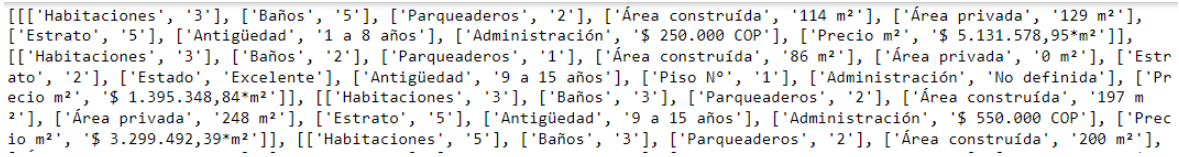


Figura 7: Extracción de la información datos tipo lista.

Cabe aclarar que el resultado del algoritmo con las tres ejecuciones, presentaron corchetes y comillas simples, pero ya no esta vez como caracteres en formato tipo string, si no de la forma donde los corchetes representan listas y las comillas representan que el dato es texto, ya que Python los presenta de esa forma.

Dado que todavía no habíamos logrado separar las características que se presentan en columnas, con el fin de crear un data frame piloto. Se sigue trabajando en Python de forma cómoda, ya que los elementos de la lista propiedades son más fáciles de manipular por estar guardados en sub listas. A continuación, se sigue detallando este proceso:

Teniendo en cuenta que las características de los inmuebles quedaron guardadas sin caracteres especiales y como sub listas, donde la primera posición de cada una de las sub listas corresponde al nombre de la característica del inmueble y la segunda posición corresponde al valor de ella, se consigue averiguar por medio de un algoritmo; ¿Cuántas? y ¿cuáles? son las características guardadas en la lista llamada objetos.

El algoritmo consistió en pedirle a Python que recorriera la primera posición de cada una las sub listas pertenecientes a la lista objetos, ya que en esa posición se presentaban los nombres de las características, además se le pidió que esos nombres los fuera guardando en una lista llamada variables, lo cual nos arrojó 12 nombres de características las cuales son:

- Habitaciones
- Baños
- Parqueaderos
- Área construida
- Área privada
- Estrato
- Estado
- Antigüedad
- Piso N°
- Administración
- Precio m^2
- Tipo de apartamento

Teniendo ya claro las características que se presentan en la lista objetos, se procede a extraer en forma de dupla las características del inmueble con el nombre y su valor.

Lo cual se desarrolló un algoritmo que va seleccionando una por una las características de búsqueda guardadas en la lista variables, donde además va recorriendo las sub listas buscando las características que le pertenece a cada inmueble y guardando cada una de ellas con su valor en una sola lista llamada estructura.

Llegado el caso que la característica no apareciera en las sub listas de cada inmueble se agregaba el nombre de la característica acompañada de un "na"; lo que significa que no la tiene.

Ya guardadas todas las características de los inmuebles en orden en una sola lista llamada estructura, se averigua el rango de elementos donde cambia de característica dicha lista.

Lo cual se desarrolla un nuevo algoritmo que recorre la lista estructura desde la posición inicial hasta la posición final; con el fin de que me arroje la posición de la lista cuando empiezan a cambiar de característica.

En función de lo anterior procedimos a particionar las características, dándole como tarea a Python la toma de los rangos adquiridos anteriormente; donde aparece cada una de las características de todos los inmuebles, lo cual va recorriendo la lista estructura para ir seleccionando y guardando en listas separadas las 12 características mencionadas.

Por último, se emplea un algoritmo que une las listas que teníamos antes y despues del proceso de minería de datos por columnas separadas; donde en total salieron 16. Lo cual se convierte en un data frame guardado en un archivo xlsx.

Damos por terminado la actividad de recolección y extracción de los datos, donde en el Cuadro 1 se presentan los datos que pertenecen al data frame con su variable, descripción y etiqueta.

Cuadro 1: Datos que pertenecen al data frame.

Variable	Descripción	Etiqueta
Link	Contiene las direcciones web específicas de los inmuebles de la página inmobiliaria Finca raíz.	Links
Dirección	Contiene la posición geográfica del inmueble en la ciudad de Neiva, escrita en formato calle o carrera.	Dirección
Barrio	Contiene los nombres de las extensiones de tierra en Neiva, limitadaspor líneas imaginarias establecidas.	Barrio
Habitaciones	Contiene el número de habitaciones que posee el inmueble.	Habitaciones
Baños	Contiene el número de baños que posee el inmueble.	Baños
Parqueaderos	Contiene el número de parqueaderos que posee el inmueble.	Parqueaderos
Área construida	Expresa el valor en la unidad metro cuadrado, de la superfie del terreno en donde se ha construido el inmueble.	Área construida
Área privada	Expresa el valor en la unidad metro cuadrado, de la superficie al interior del inmueble.	Área privada
Estrato	Expresa la estratificación socieconómica de los inmuebles que deben recibir servicios públicos en 2, 3, 4, 5 y campestre.	Estrato
Antigüedad	Indica el número de años trancurridos desde la construcción del inmueble hasta su última rehabilitación, por medio de intervalos los cuales son: menor a 1 año, 1 a 8 años, 9 a 15 años, 16 a 30 años y más de 30 años.	Antigüedad
Administración	Expresa el valor monetario a pagar de algunos inmuebles y si no es de ser así se presenta como no definida.	Administración
Precio por metro cuadrado	Indica el valor monetario a pagar de algunos inmuebles y si no es de ser así se presenta como no definida.	Precio m^2
Estado	Expresa el valor monetario por metro cuadrado del inmueble.	Estado
Piso	Indica el número de pisos, si el inmueble es una casa o en que piso se encuentra si el inmueble es un aparatamento.	Piso
Tipo	Indica si el inmueble es casa o apartamento.	Tipo
Precio	Presenta el valor comercial del inmueble.	Precio

6.2. Transformación y Limpieza de los datos

Despues de la recolección y extracción de los datos, se detectaron errores en el data frame; los cuales hacían imposible el análisis de los datos y la construcción del modelo. A continuación, se presentan por separado los errores con su debida solución.

6.2.1 Eliminación de datos repetidos

Teniendo en cuenta el data frame donde contiene 383 inmuebles, se procede a la detección de datos repetidos en Excel. Para ello se utilizó la función quitar duplicados en los inmuebles que contenían exactamente los mismos valores en las columnas. Los cuales dio un total de 4 inmuebles repetidos eliminados, dejando 379 inmuebles en total.

6.2.2 Limpieza de caracteres especiales

Debido a que los caracteres especiales que referencian unidades de medidas, como por ejemplo metro cuadrado (m^2) y precio ($), resultan inconvenientes para el análisis y construcción del modelo. Por lo que se empleó la función buscar y reemplazar de Excel, donde se busca los caracteres especiales "(m^2)","($)" y se reemplazan por vacío.

Luego procedimos a eliminar las comas (,) que referenciaban unidades de mil y también los puntos (.) como separadores de unidades decimales. Por último, se convierte a tipo numérico las categorías de las columnas que contienen solo números.

6.2.3 Conversión de las direcciones a latitud y longitud

Dado que las direcciones de los inmuebles pertenecientes a la columna dirección se encuentran en formato tipo texto, se procedió a convertirlas a valor numérico obteniéndolas en coordenadas latitud y longitud.

Para ello se sube la columna "dirección" a una hoja de cálculo de Google donde se desarrolla un algoritmo de geo codificación, utilizando una extensión de la hoja de cálculo llamada Apps Script, donde se maneja java script como lenguaje de programación.

Luego de tener las direcciones en coordenadas latitud y longitud, con la herramienta Google My Maps se creó un mapa personalizado de la ciudad de Neiva añadiendo la ubicación de los 379 inmuebles; tal como lo muestra la Figura 8.

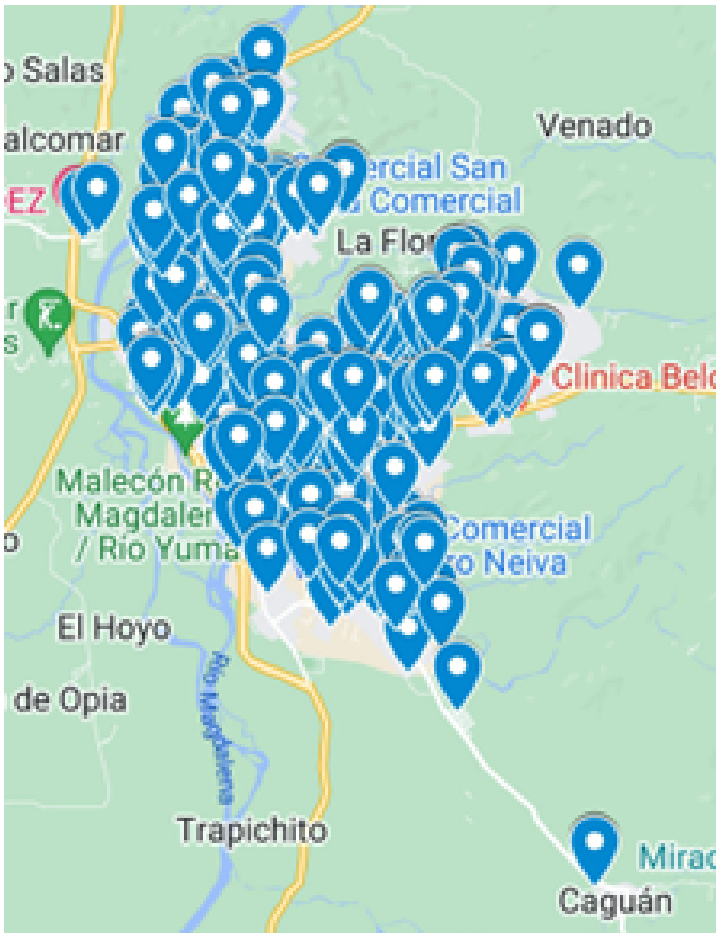


Figura 8: Mapa personalizado de la ciudad de Neiva con la ubicación de los 379 inmuebles.

6.2.4 Imputación de datos vacíos en las columnas del data frame

Se cargó el data frame a Python con la ayuda de la librería "pandas", donde se visualizó con la función ".head( )" si los datos estaban estructurados para la lectura con Python, tal como lo muestra la Figura 9.

	Unnamed: 0	Links	Dirección	Latitud	Longitud	Barrio	Habitaciones	Baños	Parqueaderos	Area construida	Area privada	Estrato
0	0	https://www.fincaraiz.com.co/inmueble/apartame...	Cra. 52 #11-60, Neiva, Huila, Colombia	2.937702	-75.245074	CONJUNTO RESERVA DE LA SIERRA - Neiva	3	5	2	114.0	129.0	5
1	1	https://www.fincaraiz.com.co/inmueble/casa-en-...	Cl. 22 #1A-07, Neiva, Huila, Colombia	2.933942	-75.300573	CONJUNTO CERRADO BRISAS DEL MAGDALE - Neiva	3	2	1	86.0	0.0	2
2	2	https://www.fincaraiz.com.co/inmueble/casa-en-...	Cl. 8 #52-149, Neiva, Huila, Colombia	2.931560	-75.271617	ipanema - Neiva	3	3	2	197.0	248.0	5

Figura 9: Encabezado del data frame.

Posteriormente utilizamos la función ".info( )" la cual nos presentó un resumen conciso del marco de datos, presentado en la Figura 10. Donde pudimos evidenciar: las columnas del data frame, el número de datos no vacíos y los tipos de datos al que pertenece cada columna.

```
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       379 non-null    int64
 1   Links            379 non-null    object
 2   Dirección        379 non-null    object
 3   Latitud          379 non-null    float64
 4   Longitud         379 non-null    float64
 5   Barrio           379 non-null    object
 6   Habitaciones     379 non-null    int64
 7   Baños            379 non-null    int64
 8   Parqueaderos     379 non-null    int64
 9   Area construida  379 non-null    float64
 10  Area privada     379 non-null    float64
 11  Estrato          379 non-null    object
 12  Antiguedad       341 non-null    object
 13  Administracion   379 non-null    object
 14  Precio_m2        379 non-null    float64
 15  Estado           117 non-null    object
 16  Piso             118 non-null    float64
 17  Tipo             379 non-null    object
 18  Precio           379 non-null    int64
```

Figura 10: Información del data frame.

Dicho resumen nos permitió identificar que columnas como: antigüedad, estado y piso presentaron un número de datos vacíos.

Además, los datos de las columnas estrato y administración no estaban siendo leídos correctamente; ya que Python los leyó como tipo texto (object) y no como tipo entero (int64), esto se debió a que algunos inmuebles en la columna estrato presentaban estrato tipo "campestre" y en la columna administración los datos vacíos aparecían como "no definida".

Según lo anterior se procedió a resolver la complicación de los datos vacíos en las columnas mencionadas, donde se planteó una alternativa para mitigar los datos faltantes llamada imputación de datos.

La cual estima los datos ausentes en base a los valores válidos en el data frame, empleando las matemáticas que van detrás de los números pseudoaleatorios.

Para ello analizamos la variable antigüedad que está definida como variable tipo texto (object), y está clasificada en: 1 a 8 años, 9 a 15 años, 16 a 30 años, más de 30 años y menor a 1 año. Donde se presentaron 341 datos no vacíos de un total de 379. Tal como lo muestra la Figura 11.

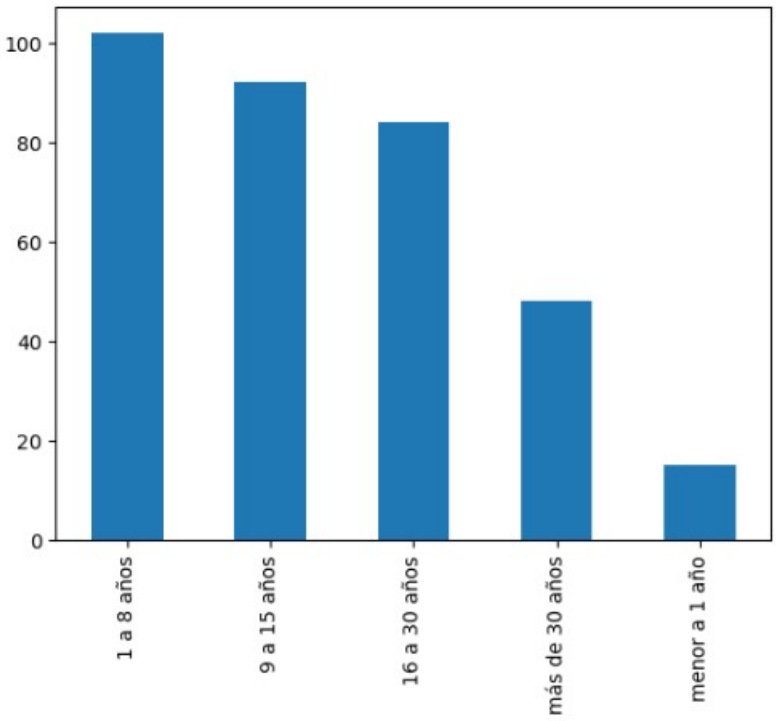


Figura 11: Diagrama de las antigüedades de los inmuebles.

Además, se comprobó que ningún inmueble presentara dos clasificaciones al mismo tiempo, evidenciado en la Figura 12. Con dicha información se calculó los porcentajes que pertenecen a cada clasificación con respecto al total de los datos.

Antiguedad	1 a 8 años	16 a 30 años	9 a 15 años	menor a 1 año	más de 30 años	All
Antiguedad						
1 a 8 años	102	0	0	0	0	102
16 a 30 años	0	84	0	0	0	84
9 a 15 años	0	0	92	0	0	92
menor a 1 año	0	0	0	15	0	15
más de 30 años	0	0	0	0	48	48
All	102	84	92	15	48	341

Figura 12: Clasificaciones de la antigüedad de los inmuebles.

- 1 a 8 años 26.912929 %
- 9 a 15 años 24.274406 %
- 16 a 30 años 22.163588 %

- Más de 30 años 12.664908 %
- Menor a 1 año 3.957784 %

Despues de haber realizado el análisis a la variable antigüedad, con base a las 5 clasificaciones se realizó la imputación de datos. Proceso que se llevó a cabo mediante el desarrollo de un algoritmo que identificó los valores vacíos, los cuales fueron rellenados por un valor igual a 1 por medio de la función ".fillna( )"; luego se aplicó una función ".choice.random( )" la cual hizo que los valores iguales a 1 fueran reemplazados por las clasificaciones de manera aleatoria y uniforme.

Dichos resultados se muestran en la Figura 13, donde podemos evidenciar la imputación de datos manteniendo la uniformidad de las clasificaciones, es decir, antes de la imputación de datos los cuales fueron mostrados en la Figura 7.

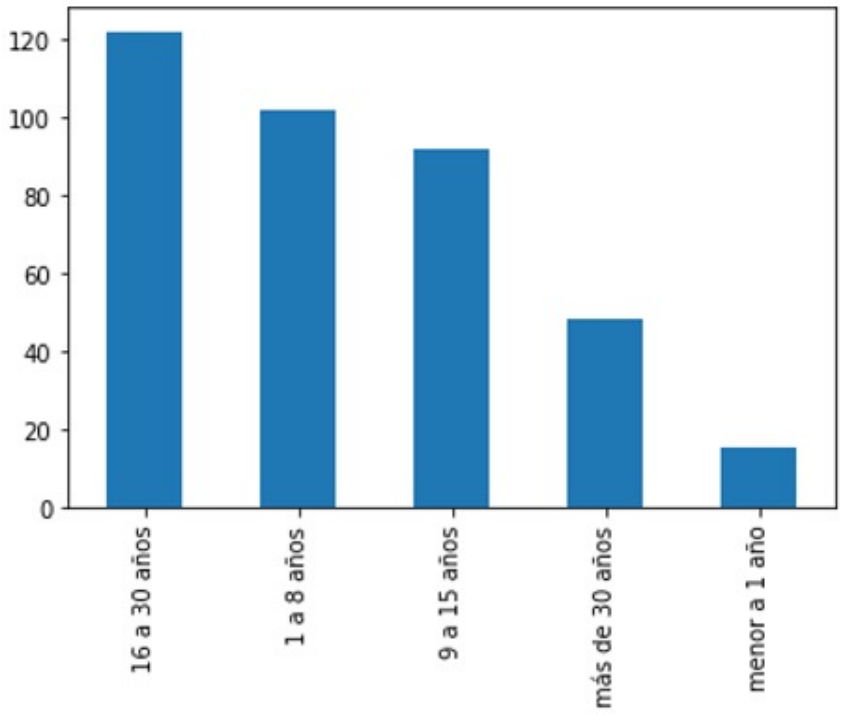


Figura 13: Imputación de datos de la antigüedad.

Despues se revisó la variable "piso" la cual presentó 118 datos no vacíos de un total 379, donde se realizó una imputación a los datos vacíos desarrollando un algoritmo que utiliza la función ".fillna( )"; la cual filtra los datos vacíos y le agregó un valor igual a 1, estableciendo que el inmueble es de un piso si es casa o de ser apartamento se encuentra en el primer piso.

Por otra parte, también se evidencio que la variable "estado" presentó un número de 117 datos no vacíos. Por lo que se hizo un análisis a la variable, con el objetivo de dar una alternativa para la mitigación de datos vacíos. Dicho análisis nos permitió ver que la variable es tipo texto y tomaba dos valores que son: "Bueno" y "excelente". Debido a esta información

se decidió omitir la variable "estado" ya que los valores que toma no son relevantes influyendo una diferencia en el valor comercial.

Por último, se tomó la variable "administración" para transformar los valores "no definido" los cuales representan valores vacíos, ya que los valores que puede tomar la variable son precios o precios nulos en caso de no pagar. De esta forma, se aplicó un filtro en la columna "administración" donde se utilizó una función ".replace( )" para reemplazar los valores "no definido" por un valor numérico igual a 1. Dado que lo siguiente fue multiplicar por un valor adecuado (1000 COP) a los precios de administración menores a este mismo. Obteniendo como resultado que los inmuebles que no facturan administración no influyeran en el valor comercial.

6.2.5 Categorización de datos en la variable estrato y antigüedad

La variable estrato tenía definidos algunos valores como "campestre", lo cual no estipula un valor numérico asociado a la estratificación socioeconómica de un inmueble. Por consecuencia, se planteó relacionar el valor "campestre" a un valor numérico. De esta manera, se localizaron inmuebles ubicados en la misma zona y presentaban un estrato igual a 4; lo que implica una relación con el valor "campestre". De este modo, se empleó un algoritmo basado en la función ".loc[ ]" que seleccionó estos valores y los igualó a 4.

En cuanto a la variable antigüedad se realizó una conversión categórica a numérica, con el objetivo de preparar la variable a posteriores análisis y por supuesto al ajuste del modelo. La conversión se logró bajo un algoritmo, el cual estableció un rango de las clasificaciones que toma dicha variable utilizando la función "range", que resultó en un total de 5. Por lo tanto, los valores numéricos asociados a cada una de las clasificaciones son (0,1,2,3 y 4), por último, se reemplaza los valores categóricos por los numéricos empleando una función ".repleace( )".

6.2.6 Norma vectorial en las coordenadas latitud y longitud

Teniendo en cuenta que las coordenadas latitud y longitud definen la posición de un punto sobre el esferoide (tierra), se formaron vectores con cada una de las coordenadas de los 379 inmuebles, de lo cual se calcularon las magnitudes. Dicho proceso se realizó, subiendo el data frame a Python donde se desarrolló el algoritmo capaz de calcular las magnitudes a través de la norma vectorial.

$$\|v\| = \sqrt{(latitud)^2 + (longitud)^2} \tag{18}$$

Los valores obtenidos fueron guardados en la columna con etiqueta "Posición".

6.2.7 Similitud de coseno entre las columnas dirección y barrio

Dado que las columnas dirección y barrio están en formato texto, se planteó una conversión a formato numérico aplicando la similitud de coseno entre las dos columnas. Este proceso se desarrolló en Python, donde se empleó la función "text_vector". Lo cual recorrió

las columnas dirección y barrio de los inmuebles, tomo sus palabras y las codificó en vectores.

Luego de tener vectorizadas las direcciones y el barrio de los inmuebles, a cada palabra se le asignó teóricamente una dimensión diferente, donde el valor en cada dimensión correspondió al número de veces que la letra aparece en los vectores. En consecuencia, se construyó un algoritmo utilizando la fórmula de similitud de coseno, calculando los valores de similaridad entre los vectores en término de las palabras que se contienen. Por último, los valores de similaridad que están claramente delimitados en [0,1], se guardaron en una columna con etiqueta "Direccion_barrio".

$$cos\theta = \frac{A \cdot B}{|A||B|} \tag{19}$$

Donde, A y B son vectores atributos.

$A \cdot B$ producto escalar.

$\|A\|$ magnitud del vector atributo A.

$\|B\|$ magnitud del vector atributo B.

Fórmula de similitud de coseno.

El código utilizado en la transformación de los datos en Python se presenta a detalle en el Anexo, como también el data frame generado.

6.3. Análisis Exploratorio de datos

En esta parte, es importante tener en cuenta que antes de iniciar con un entrenamiento del modelo predictivo, se debe realizar una exploración de los datos para entender mejor la información que se tiene en cada variable y también detectar algunos errores comunes. Un ejemplo de esto:

- Una variable numérica que no está siendo reconocida como texto, que tenga espacios vacíos, o sea reconocida como tipo texto.
- La variable de tipo numérico se haya introducido una palabra en lugar de un número.
- Una variable que contenga valores incoherentes, es decir, que tenga texto o un "0" en el precio del inmueble.
- No conocer el número de observaciones disponibles, si están incompletas, valores ausentes que son importantes para crear los modelos, ya que no aceptan observaciones incompletas.

Lo primero que se hizo fue revisar la totalidad de los datos que se tenían, debido a que Random forest tiene valores por defecto para cada uno de los hiperparámetro y dado esto

no podemos saber de antemano si estos son los óptimos para ajustar el modelo, para ello se identifica los valores óptimos de los hiperparámetro con las estrategias de validación, como la validación cruzada. En la siguiente Figura 14 se evidencia la totalidad de los datos en cada una de las columnas.

```
datos_finca.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 379 entries, 0 to 378
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Unnamed: 0       379 non-null    int64
 1   Links            379 non-null    object
 2   Dirección        379 non-null    object
 3   Latitud          379 non-null    float64
 4   Longitud         379 non-null    float64
 5   Barrio           379 non-null    object
 6   Habitaciones     379 non-null    int64
 7   Baños            379 non-null    int64
 8   Parqueaderos     379 non-null    int64
 9   Area construida  379 non-null    float64
 10  Area privada     379 non-null    float64
 11  Estrato          379 non-null    object
 12  Antiguedad       379 non-null    int64
 13  Administracion   379 non-null    int64
 14  Precio_m2        379 non-null    float64
 15  Estado           379 non-null    object
 16  Piso             379 non-null    float64
 17  Tipo             379 non-null    object
 18  Precio           379 non-null    int64
dtypes: float64(6), int64(7), object(6)
memory usage: 56.4+ KB
```

Figura 14: Información del data frame.

Siendo así, se puede validar que tenemos en cada una de las características un total de 379 datos correspondientes en cada inmueble. Además, este análisis exploratorio de datos nos puede dar pistas sobre qué variables son adecuadas como predictores para los modelos.

Luego, que se verificó que están los datos, se procedió a ver el mapa de calor de la matriz de correlación (Figura 18), se analizó las variables predictoras con mayor grado de correlación frente a la variable respuesta(precio). Posteriormente, se observó las medidas estadísticas como son: media, desviación estándar, valor máximo y cuartiles; quedando todo establecido en el apartado 7.2 de análisis de resultados.

Al construir un modelo, es importante estudiar la distribución de la variable de respuesta porque, en última instancia, eso es lo que le interesa predecir. En este caso es la variable precio; la gráfica de su distribución (Figura 20) se puede ver en el apartado 7.2 de análisis de resultados.

6.4 Desarrollo de modelos preliminares

En esta actividad, se ajustaron modelos de machine learning, para responder a los objetivos propuestos. Los modelos se ajustaron a través de técnicas de diseño de algoritmos de "aprendizaje automático". A causa de que existen un gran número de estas técnicas basadas en la predicción de datos, se realizó un estado del arte de los diversos casos de estudio similares al proyecto. De las cuales se escogieron tres técnicas más frecuentes en estos casos:

Random Forest, XGBoost y Redes neuronales profundas.

Estos modelos se ajustaron en el entorno de Google Colab, por su interfaz, lenguaje de programación Python y visualización de las gráficas y variables. Estando en Colab se procedió a buscar las librerías en Python que nos proporcionó la base para el ajuste de dichos modelos. A continuación, se pueden evidenciar las librerías en el Cuadro 2.

Cuadro 2: Librerías a utilizar en el ajuste de los modelos.

Librería	**Utilidad**
matplotlib.pyplot	Permite crear y personalizar gráficos a partir de listas y arrays.
Numpy	Permite crear vectores y matrices.
Pandas	Manipulación y análisis de datos.
Scikit-learn (sklearn)	Herramienta básica para programar y estructurar los sistemas de análisis de datos y modelo estadísticos.
Math	Acceso a las funciones matemáticas.
Statistics	Acceso a las funciones estadísticas de datos numéricos

Teniendo en cuenta el data frame conseguido en la etapa de procesamiento de datos, se eliminaron las columnas: Links, dirección y barrio . Puesto que son columnas que no aportan información relevante al desarrollo de los modelos, como es el caso de la columna Links.

En el caso de las columnas dirección y barrio por ser columnas tipo texto no es posible identificar una correlación, por lo cual se crearon nuevas columnas de tipo numérico como: latitud, longitud, posición y direccion_barrio derivadas de estas dos columnas. Debido a esto, el data frame utilizado en el ajuste de modelos de machine learnning, contiene las siguientes columnas que se exponen en la Figura 15.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 379 entries, 0 to 378
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Latitud           379 non-null    float64
 1   Longitud          379 non-null    float64
 2   Posicion          379 non-null    float64
 3   Habitaciones      379 non-null    int64
 4   Baños             379 non-null    int64
 5   Parqueaderos      379 non-null    int64
 6   Area construida   379 non-null    float64
 7   Area privada      379 non-null    float64
 8   Antiguedad        379 non-null    int64
 9   Administracion    379 non-null    int64
 10  Precio_m2         379 non-null    float64
 11  Piso              379 non-null    float64
 12  Estrato           379 non-null    int64
 13  Direccion_barrio  379 non-null    float64
 14  Tipo              379 non-null    int64
 15  Estado            379 non-null    int64
 16  Precio            379 non-null    int64
dtypes: float64(8), int64(9)
memory usage: 50.5 KB
```

Figura 15: Información del data frame.

Antes de proceder al diseño de los algoritmos de aprendizaje automático, se dividió el data frame en el 75 % de los datos para entrenamiento y el 25 % de los datos para prueba de forma aleatoria. Empleando como variable respuesta "Precio" y como predictoras todas las otras variables disponibles.

6.4.1. Random Forest

El primer modelo se ajustó utilizando el algoritmo Random Forest. Donde se cargaron los datos de entrenamiento y de test con sus variables predictoras y variable respuesta. Además, se hizo uso de hiperparámetros mencionados en el capítulo de las matemáticas, los cuales se exponen con sus respectivos valores y descripciones:

- n_estimators = Número de árboles incluidos en el modelo.
- criterion = Ganancia de información, error cuadrático.
- max_depth = Profundidad máxima que puede alcanzar los árboles. En este caso 2 es el número mínimo observaciones para que el nodo pueda dividirse.
- max_features = Número de predictores considerados en cada división. En este caso utiliza todos los predictores.
- Obb_score = Sirve como estimación del error del test. Si se calcula o no el OOB, por defecto es false ya que aumenta el tiempo de entrenamiento.
- N_Jobs = Número de cores empleados para el entrenamiento. En este casos utiliza todos los cores disponibles.
- random_state = Semilla para que los resultados sean reproducibles.

Como en todo estudio predictivo, no solo es importante ajustar el modelo, sino también cuantificar su capacidad para predecir nuevas observaciones. Dicho esto, una vez entrenado el modelo se evaluó la capacidad predictiva empleando el conjunto de test y el error cuadrático medio (RMSE), lo cual nos arrojó un error que se evidencia en el capítulo 7.

El modelo preliminar se entrenó utilizando 10 árboles (n_estimators=10) y manteniendo el resto de hiperparámetros con su valor por defecto. Al ser hiperparámetros, no se puede saber cuáles son sus valores adecuados con anterioridad; ya que, dependiendo del comportamiento de los datos, los hiperparámetros varían. Para ello se empleó estrategias de validación: out-of-bag y validación cruzada.

En la implementación de "RandomForestRegressor", la métrica de validación devuelta como Obb_score (out-of-bag) es el $\boldsymbol{R^2}$ y en cv-error (validación cruzada) es el $\boldsymbol{RMSE}$. En conclusión, cuando se busca el valor óptimo de un hiperparámetro con dos métricas distintas ($\boldsymbol{R^2}$,$\boldsymbol{RMSE}$), el resultado obtenido no suele ser el mismo. Lo fundamental es que las dos métricas establezcan las mismas regiones de interés.

Con base en lo anterior, se procedió a calcular el número adecuado de árboles (n_estimators), con el fin de ahorrar recursos computacionales. Dado que en Random Forest, el añadir árboles mejora los resultados del modelo y además, no produce sobre ajuste por exceso de árboles. Sin embargo, al no ser un hiperparámetro crítico, el añadir más árboles una vez el resultado se estabiliza; es una pérdida de recursos computacionales. Lo cual se empleó las dos métricas de validación mencionadas, donde se obtuvo una región de interés presentada en el Capítulo 7, para así poder estimar el número de árboles adecuados.

Después, se buscó el valor óptimo de uno de los hiperparámetros más importante del Random Forest el cual es "máx_features", ya que reduce la correlación que hay entre los árboles. Dado que indica el número de predictores considerados en cada división, haciendo que un predictor influyente no sea elegido para todos los árboles. Es decir, otros predictores pueden ser seleccionados en cada división sin tener en cuenta su valor influyente. Dicho esto, se empleó las dos métricas de validación mencionadas, donde se obtuvo una región de interés presentada en el Capítulo 7, para así poder estimar el número óptimo de predictores considerados en cada división.

Ya que hasta el momento, el análisis de los hiperparámetros se había realizado de manera individual, entendiendo su impacto en el modelo y además identificando regiones de interés. Se debió tener en cuenta que cada hiperparámetro interactúa con los demás. Lo cual, se recurrió al método grid search, o también conocido como random search; para el análisis de varias combinaciones de hiperparámetros.

Este método hizo una búsqueda completa de todas las combinaciones de hiperparámetros, teniendo en cuenta las regiones de interés obtenidas en el Obb_score (out-of-bag) y en cv-error (validación cruzada). Luego, se evaluó todas las combinaciones posibles de hiperparámetros, empleando la métrica de validación R^2 y $RMSE$. Debido a esto, se escogió la combinación de hiperparámetros con mejores resultados en las métricas de validación, la cual se expone en el Capítulo 7.

Por último, como resultado de optimizar los hiperparámetros, se consiguió reducir el error $RMSE$ del modelo empleando el conjunto de test. En el Capítulo 7 se evidencia la reducción de dicho error.

6.4.2. Redes Neuronales Profundas.

Las Redes Neuronales Profundas fue el segundo algoritmo que se empleó para ajustar un nuevo modelo a los datos. Para ello se volvió a utilizar él data frame inicial empleado en el Random Forest, de lo cual se realizó dos tipos de preprocesamiento: binarización (One hot ecoding) de las variables tipo texto (object) y estandarización de las variables tipo numérico (int y float). En base a esto se obtuvo un nuevo data frame apropiado, el cual se usó en el algoritmo de Redes Neuronales Profundas. En el capítulo 7 se expone el data frame obtenido en el preprocesamiento.

Luego de obtener él data frame preprocesado, se definió la arquitectura de la Red Neu-

ronal Profunda, es decir, el número de capas ocultas y el número de neuronas de cada una. Debido a esto, se basó en el trabajo de (Antón, 2020), el cual sugiere un número de capas ocultas igual o mayor a 2; ya que es capaz de aprender relaciones complejas entre variables. Además, se buscó identificar el número de épocas por las que se debe entrenar el modelo, para así encontrar el punto donde el modelo alcanza un régimen estacionario (no cambian esencialmente los errores ni de entrenamiento ni de test).

Aun así, se probaron varios números de capas ocultas, neuronas por capas y épocas, obteniendo como resultado los valores correspondientes para cada uno de ellos. En el Capítulo 7 en la sección 3.2 se presentan los resultados de dichos valores.

En cuanto al entrenamiento de la red neuronal, se implementó hiperparámetros que modifican el entrenamiento y su desempeño. A continuación, se muestran los más influyentes:

- **activation**: función de activación de las capas ocultas.
- **solver**: algoritmo de optimización utilizado para aprender los pesos y bias de la red.
- **alpha**: método de regularización que se utiliza para reducir el sobreajuste de la red neuronal.
- **learning_rate_init**: es el valor inicial de la tasa de aprendizaje de la red neuronal.

Con base en lo anterior, se tuvo en cuenta las regiones de interés de los hiperparámetros empleados en el artículo de (Rodrigo, 2020); y se realizó una búsqueda de todas las combinaciones de hiperparámetros utilizando el método random search por validación cruzada (RandomizedSearchCV). De acuerdo a la métrica de validación cruzada, la combinación de hiperparámetros con mejores resultados se presenta en el Capítulo 7.

Finalmente, se evaluó la capacidad predictiva de la red neuronal profunda haciendo uso del conjunto de test. Por lo cual se obtuvo un error RMSE que se evidencia en el Capítulo 7.

6.4.3. XGboost

El tercero y último algoritmo para el ajuste de un nuevo modelo fue el XGBoost, en el cual se escogió los mismos datos de entrenamiento y test que se usó en el algoritmo Random Forest con su preprocesamiento; ya que el elemento fundamental de los dos algoritmos son los árboles de decisiones.

El algoritmo XGBoost se construyó teniendo en cuenta ciertos hiperparámetros que influyen en el entrenamiento y su desempeño, los cuales son un total de 9 hiperparámetros:

- nthread: número de hilos computacionales que serán usados.
- booster: tipo de modelo de regresión usado por defecto.
- objective: tipo de tarea de regresión que realizara.

- learning_rate: reducción de tamaño de paso utilizado para evitar el sobreajuste.
- max_depth: determina la profundidad con la que se permite que crezca cada árbol durante cualquier ronda de impulso.
- min_child_weight: número mínimo de hojas, ayuda a reducir la complejidad.
- subsample: porcentaje de muestras utilizados por árbol.
- colsample_bytree: Porcentaje de características utilizadas por un árbol.
- n_estimators: número de árboles que se desean construir.

Para el entrenamiento y desempeño de un modelo más robusto se utilizó el método random search por validación cruzada (RandomizedSearchCV), lo cual realizó el mismo procedimiento que en los algoritmos anteriores, es decir, hizo una búsqueda de las posibles combinaciones de los hiperparámetros dada una región específica, eligiendo la combinación de hiperparámetros de mejor desempeño. Dichas regiones de búsqueda de cada uno de los hiperparámetros junto a la combinación con mejor desempeño, se encuentran en el Capítulo 7.

Por último, se empleó la métrica (RMSE) para evaluar la capacidad predictiva del modelo dado el conjunto de test. Lo cual nos dio error que se encuentra en el Capítulo 7.

Para una mayor profundización acerca del desarrollo de los modelos preliminares, consultar en el anexo.

6.5. Elegir el modelo

En esta actividad, se analizó los resultados de las medidas de desempeño generadas en el ajuste de los 3 modelos. Cabe recordar que el entrenamiento para los 3 modelos fue el mismo, es decir, se empleó el mismo data frame y el mismo porcentaje de datos para entrenamiento y test. A continuación, se presentan las medidas de desempeño utilizadas para evaluar los modelos ajustados en la actividad anterior.

- **metrics.root_mean_squared_error**: la raíz del error cuadrático medio (RMSE).
- **metric_r2_score**: coeficiente de determinación.

Como se observa en el capítulo 7, los modelos que presentan un mejor coeficiente de determinación es el random forest y seguido está el XGBoost, es por esto que dichos modelos presentan un mejor ajuste a los precios de vivienda en el conjunto de datos. Por otro lado, los resultados obtenidos en el RMSE de cada modelo, se pudo observar que los modelos que presentan un menor error son random forest y seguido está el XGBoost , esto da a entender que el valor de la predicción del precio de vivienda, es más cercano respecto al valor real y conocido de la vivienda.

Por otro lado, también se analizó los criterios de calidad de modelos estadístico los cuales son:

- **AIC: Criterio de información de Akaike**.
- **BIC: Criterio de información bayesiano**
- **DIC: Criterio de información de desviación**.

Estas medidas tienen en cuenta la bondad de ajuste como la complejidad del modelo, cuanto menor sea el valor de estos criterios mejor será el modelo.

Se calculó el valor de los tres criterios para cada modelo empleando las formulas mencionadas en el marco teórico, teniendo en cuenta el número de parámetros utilizados en cada uno de los modelos, al igual que el mismo número de variables predictoras y porcentaje del data frame para entrenamiento y test.

Luego de observar los resultados, se puede concluir que el modelo que presenta el mejor desempeño fue el Random Forest, por encima del XGBoost y redes neuronales profundas. Lo cual es el modelo de regresión óptimo para predecir los valores comerciales de los inmuebles en la ciudad de Neiva.

Capítulo 7

7. Análisis e interpretación de Resultados

7.1 Data Frame

Como se había mencionado en el capítulo de Metodología, el data frame se construyó a partir de los datos, debidamente transformados; los cuales fueron obtenidos y extraídos de la página web finca raíz. El resultado es un data frame en Python, la cual puede almacenarse en archivo xlsx mediante la instrucción " name_data_frame.xlsx".

Tras la obtención y extracción de los datos de la página web finca raíz, se obtuvo un data frame con 383 inmuebles, con 16 columnas distintas, correspondientes a las distintas características de las que se extrajo información, como se puede ver en la Figura 16.

	Links	Direccion	Barrio	obitacione	Baños	Parqueadero	rea construid	rea privad	Estrato	ntigueda	dministrac	Precio_m2	Estado	Piso	Tipo	Precio
0	https://www	CONJUNTO	CONJUNT	3	2	1	76 m²	76 m²	4	Excelente	menor a 1	13	No definic	$ 2.894.73	na	220.000.000
1	https://www	Cra 46 # 8-3	Ipanema -	3	4	2	129 m²	0 m²	5	na	1 a 8 años	na	No definic	$ 5.038.75	na	650.000.000
2	https://www	carrera 25 2	Neiva la n	3	2	1	140 m²	140 m²	4	na	9 a 15 año	na	No definic	$ 2.071.42	na	468.000.000
3	https://www	Carrera 36 B	La Floresta	3	2	na	198 m²	190 m²	4	Excelente	1 a 8 años	2	No definic	$ 2.363.63	na	470.000.000
4	https://www	CALLE 13 No	Los Martir	7	4	1	280 m²	0 m²	3	Excelente	9 a 15 año	2	No definic	$ 1.678.57	na	183.357.000
5	https://www	MISAEL PAS	MISAEL PA	2	1	1	43 m²	43 m²	4	na	menor a 1	na	No definic	$ 4.264.11	na	155.000.000
6	https://www	Cra 7p 36-55	PALERMO	3	2	na	91 m²	91 m²	2	na	9 a 15 año	na	No definic	$ 1.703.29	na	430.000.000
7	https://www	calle 28a # 3	El tesoro -	3	3	na	160 m²	0 m²	0	na	na	na	No definic	$ 2.687.50	na	320.000.000
8	https://www	CALLE 21A N	SEVILLA -	6	5	na	340 m²	340 m²	3	na	1 a 8 años	na	No definic	$ 941.176,	na	260.000.000
9	https://www	Calle 22 6-61	Sevilla - N	2	1	na	98 m²	0 m²	2	na	16 a 30 añ	na	No definic	$ 1.234.69	na	646.000.000
10	https://www	casa muy bi	Neiva la n	6	2	1	91 m²	91 m²	2	na	1 a 8 años	na	No definic	$ 2.857.14	na	140.000.000

Figura 16: Data Frame sin modificar.

Luego, se realizó la detección de inmuebles repetidos en el data frame, por lo que se procedió a eliminarlos, tras lo cual quedaron 379 inmuebles. Sin embargo, no todos los inmuebles cuentan con todos las columnas de características completas, por lo que se procedió a imputar los datos faltantes en las columnas de características, para poder aprovechar el potencial de la información no faltante.

Por consiguiente, se eliminó las columnas del data frame que no aportan información relevante para el análisis y ajuste de modelos. Además, se transformaron columnas de características de tipo cualitativo (object) a tipo cuantitativo (int o float), anteriormente explicado en el capítulo de metodología en la sección de transformación de datos y desarrollo de modelos.

Con base a lo anterior, se obtuvo un data frame de 379 inmuebles, con todas las columnas de características con valores definidos, como se puede observar en la Figura 17.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 379 entries, 0 to 378
Data columns (total 17 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Latitud           379 non-null    float64
 1   Longitud          379 non-null    float64
 2   Posicion          379 non-null    float64
 3   Habitaciones      379 non-null    int64
 4   Baños             379 non-null    int64
 5   Parqueaderos      379 non-null    int64
 6   Area construida   379 non-null    float64
 7   Area privada      379 non-null    float64
 8   Antiguedad        379 non-null    int64
 9   Administracion    379 non-null    int64
 10  Precio_m2         379 non-null    float64
 11  Piso              379 non-null    float64
 12  Estrato           379 non-null    int64
 13  Direccion_barrio  379 non-null    float64
 14  Tipo              379 non-null    int64
 15  Estado            379 non-null    int64
 16  Precio            379 non-null    int64
dtypes: float64(8), int64(9)
memory usage: 50.5 KB
```

Figura 17: Información del Data Frame final.

7.2 Análisis de datos

En la Figura 18, se puede observar el grado de correlación de las variables predictoras frente a la variable respuesta (Precio).

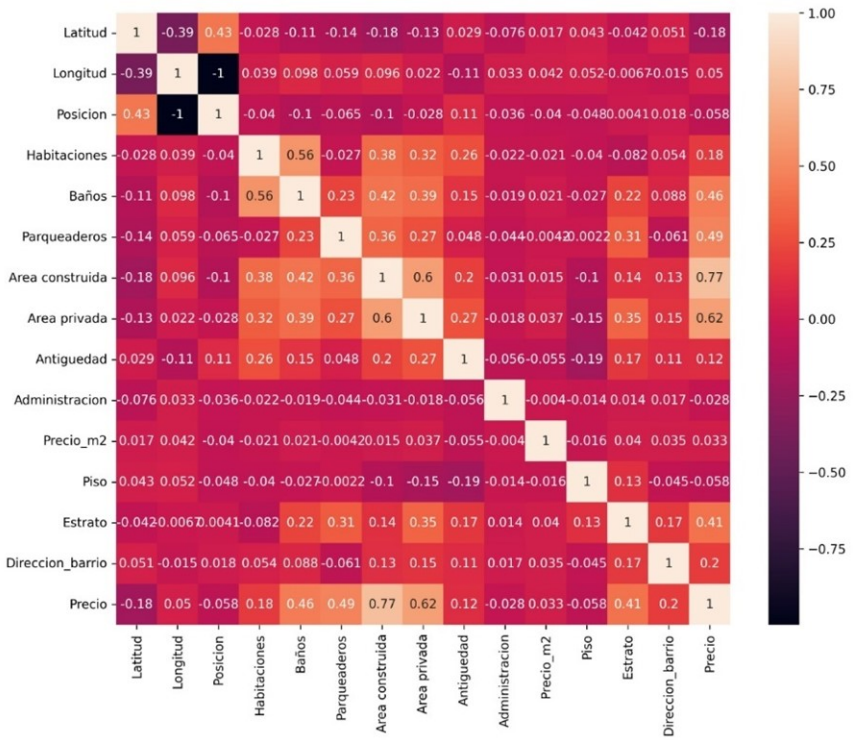


Figura 18: Mapa de calor de la matriz de correlación.

Como se puede evidenciar en el mapa de calor de la matriz de correlación (Figura 18), las variables predictoras con mayor grado de correlación son:

- Baños.
- Parqueaderos.
- Área construida.
- Área privada.
- Estrato.
- Dirección_barrio

En la siguiente Figura 19, se pueden observar los parámetros o medidas estadísticas como son: media, desviación estándar, valor máximo y cuartiles.

Que hacen parte de la estadística unidimensional; las cuales fueron calculadas a las variables predictoras y variable respuesta.

```
datos_finca.describe()
```

	Unnamed: 0	Latitud	Longitud	Habitaciones	Baños	Parqueaderos	Area construida	Area privada	Precio_m2	Piso	Precio
count	379.000000	379.000000	379.000000	379.000000	379.000000	379.000000	379.000000	379.000000	3.790000e+02	379.000000	3.790000e+02
mean	189.000000	2.933503	-75.277423	3.540897	2.559367	1.108179	149.818005	114.704892	9.448266e+06	1.598945	3.269939e+08
std	109.552118	0.021184	0.014650	1.592074	1.207795	1.064786	121.986012	106.146090	1.287613e+08	1.757102	2.813870e+08
min	0.000000	2.814687	-75.336438	0.000000	0.000000	0.000000	1.000000	0.000000	1.358168e+04	1.000000	1.450000e+07
25%	94.500000	2.925765	-75.288371	3.000000	2.000000	1.000000	86.000000	62.000000	1.567958e+06	1.000000	1.600000e+08
50%	189.000000	2.933942	-75.279344	3.000000	2.000000	1.000000	113.000000	98.000000	2.056738e+06	1.000000	2.400000e+08
75%	283.500000	2.946043	-75.268225	4.000000	3.000000	1.000000	180.000000	139.000000	2.793568e+06	1.000000	3.850000e+08
max	378.000000	2.978188	-75.227635	15.000000	10.000000	10.000000	1400.000000	900.000000	2.500000e+09	15.000000	2.500000e+09

Figura 19: Medidas estadísticas.

A continuación, se presenta en la Figura 20, la distribución de la variable respuesta (Precio). La cual tiene una distribución asimétrica con una cola positiva debido a que, unos pocos inmuebles, tiene un precio muy superior a la media.

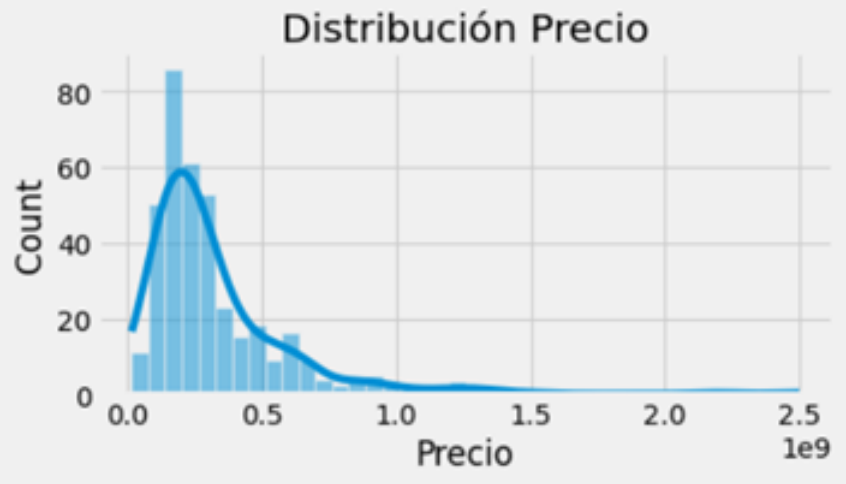


Figura 20: Distribución de la variable respuesta.

7.3 Desarrollo de modelos preliminares

Una vez obtenido, tratado y analizado el data frame, se procedió al ajuste y optimización de los modelos predictivos basados en los algoritmos de Random Forest, Redes Neuronales Profundas y XGBoost. Para ello, como se ha indicado en el capítulo de Metodología, se empleó la librería sklearn para la construcción de la arquitectura de cada uno de los algoritmos, teniendo en cuenta los hiperparámetros que influyen en el entrenamiento y resultado de los modelos.

7.3.1 Random Forest

El primer algoritmo que se empleo fue el Random Forest el cual se define su arquitectura, basado en los hiperparámetros influyentes para el entrenamiento del modelo:

- n_estimators = 10
- criterion = 'Squared_error'
- max_depth = 'None'
- max_features = 'auto'
- Obb_score = false
- N_Jobs = -1

- random_state = 123

Cabe destacar, que el modelo fue entrenado utilizando 10 árboles y los demás hiperparámetros con su valor por defecto. Luego, se evaluó la capacidad predictora del modelo empleando el error RMSE y el conjunto de test, lo cual nos arrojó un error del 7.21 %.

Optimización de hiperparámetros del Random Forest

Cuando se busca optimizar ciertos hiperparámetros, se tiene como objetivo mejorar el error RMSE obtenido respecto de los datos de test, es decir, al tratar de predecir con el modelo resultante los datos de test, el error sea menor.

Aclarado esto, los siguientes hiperparámetros que se trataron de optimizar fueron el n_estimators (número de árboles), max_features (número de predictores) y max_depth (número de hojas).

Para ello, los hiperparámetros n_estimators y max_features se les realizó una búsqueda individual empleando las métricas de validación Obb_score y cv-error, de las cuales se obtuvo una región de interés donde se evidencia la estabilidad del error R^2 y RMSE, para cada una de las métricas de validación respectivamente.
A continuación, se presenta los resultados que se obtuvieron por medio de las Figura 21.

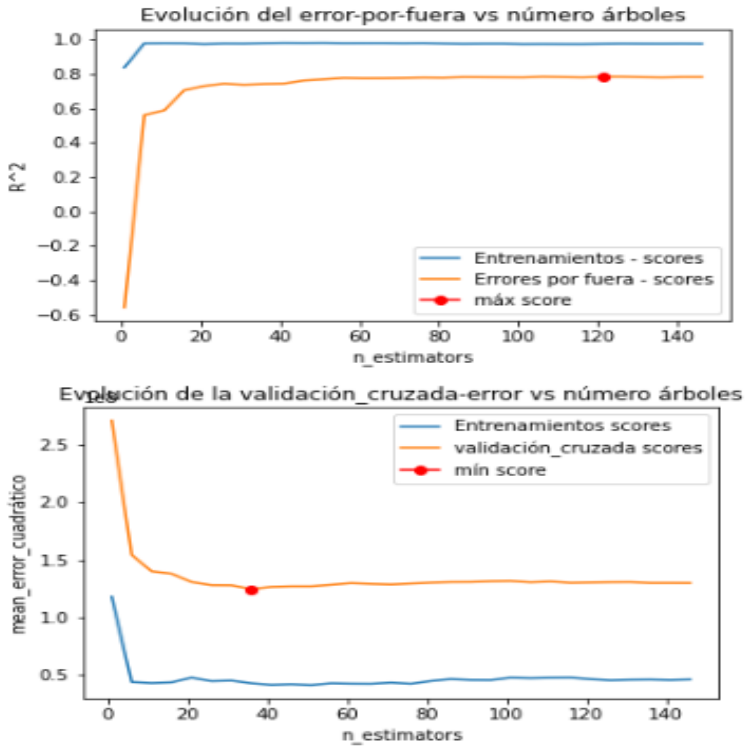


Figura 21: Resultados Obb_score y cv-error para n_estimators.

Según la Figura 21 las métricas indicaron que, entre 36 y 121 árboles, el error del modelo

se estabiliza.

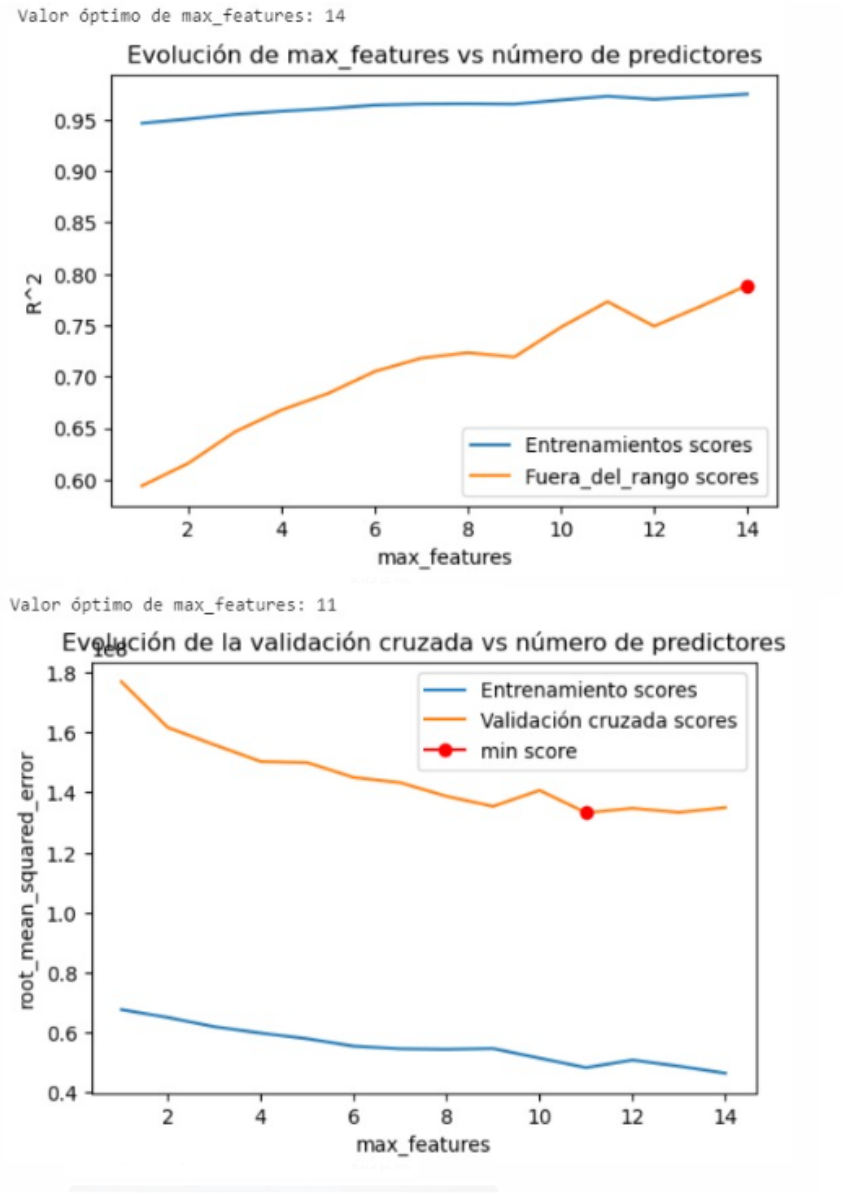


Figura 22: Resultados Obb_score y cv-error para max_features.

Según la Figura 22 las dos métricas determinaron, que la región optima del max_features se encuentra entre 11 y 14.

Por último, se recurrió al análisis de varias combinaciones de hiperparámetros empleando el método grid search, ya que cada hiperparámetro interacciona con los demás. Dicho esto, se establecieron regiones de interés para los hiperparámetros n_estimators, max_features y max_depth, con el objetivo de realizar múltiples combinaciones entre los valores de cada hiperparámetros y así identificar la mejor combinación.

Cabe resaltar que el método grid search también se basa en las dos métricas de validación Obb_score y cv-error, las cuales presentaron los siguiente resultados.

	oob_r2	max_depth	max_features	n_estimators
0	0.75301	NaN	12.0	150.0
1	0.75301	NaN	13.0	150.0
16	0.75301	22.0	13.0	150.0
15	0.75301	22.0	12.0	150.0

Figura 23: Resultados del método grid search basado en Obb_score.

En la Figura 23, se obtuvo diferentes combinaciones de valores para los tres hiperparámetros, utilizando el método grid search basado en Obb_score, de las cuales se identificó la mejor que es: max_depth = 22.0, max_features = 12.0 y n_estimators = 150.0 con una proporción de confiabilidad en el R^2 de 0.748837.

	param_max_depth	param_max_features	param_n_estimators	mean_test_score	std_test_score	mean_train_score
11	8	13	150	-1.282807e+08	7.997700e+07	-5.511842e+07
17	22	13	150	-1.291942e+08	8.032202e+07	-5.481804e+07
2	None	13	150	-1.291942e+08	8.032202e+07	-5.481804e+07
14	20	13	150	-1.291948e+08	8.032142e+07	-5.482051e+07

Figura 24: Resultados del método grid search basado en cv-error.

Dada la Figura 24, se obtuvo diferentes combinaciones de valores para los tres hiperparámetros, utilizando el método grid search basado en cv-error, de las cuales se identificó la mejor que es: max_depth = 8, max_features = 13 y n_estimators = 150.0 con un error negativo del RMSE del -124688003.98615666.

Una vez identificados los mejores hiperparámetros los cuales son max_depth = 8, max_features = 13 y n_estimators = 150, se reentreno el modelo indicando los valores óptimos en sus argumentos. Tras la optimización de los hiperparámetros, se consiguió reducir el error RMSE del modelo de 7.21 % a 5.32339 % respecto al conjunto de test. Dicho esto, las predicciones del modelo final se alejan en promedio 53,223,393 COP del valor real.

7.3.2 Redes Neuronales Profundas

El siguiente algoritmo que se usó, fue las Redes Neuronales Profundas. Para ello, primero se realizó dos tipos de preprocesamiento en el data frame conocidos como: binarización (One

hot ecoding) de las variables tipo texto (object) y estandarización de las variables tipo numérico (int y float). A continuación, se presenta en la Figura 25, la información del data frame despues haberle aplicados los dos tipos de preprocesamiento.

```
Data columns (total 18 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Latitud                    284 non-null    float64
 1   Longitud                   284 non-null    float64
 2   Posicion                   284 non-null    float64
 3   Habitaciones               284 non-null    float64
 4   Baños                      284 non-null    float64
 5   Parqueaderos               284 non-null    float64
 6   Area construida            284 non-null    float64
 7   Area privada               284 non-null    float64
 8   Administracion             284 non-null    float64
 9   Precio_m2                  284 non-null    float64
 10  Piso                       284 non-null    float64
 11  Estrato                    284 non-null    float64
 12  Direccion_barrio           284 non-null    float64
 13  Antiguedad_1 a 8 años      284 non-null    float64
 14  Antiguedad_16 a 30 años    284 non-null    float64
 15  Antiguedad_9 a 15 años     284 non-null    float64
 16  Antiguedad_menor a 1 año   284 non-null    float64
 17  Antiguedad_más de 30 años  284 non-null    float64
dtypes: float64(18)
memory usage: 40.1 KB
```

Figura 25: Información del data frame luego del preprocesamiento.

Luego, se diseñó la arquitectura de la Red Neuronal Profunda que está compuesta por 2 capas ocultas de 100 neuronas cada una, definiendo un entrenamiento del modelo igual a 200 épocas.

Optimización de hiperparámetros de las Redes Neuronales Profundas

En este punto, se procedió a optimizar los hiperparámetros mencionados en la metodología, los cuales son: activation, solver, alpha y learning_rate_init. Por lo cual, se utilizó el método random search por validación cruzada para así poder encontrar la mejor combinación de hiperparámetros, con el objetivo de que el modelo obtenga un buen desempeño.

Con base a lo anterior, el método random search se basó en las regiones de interés del articulo (Rodrigo, 2020), donde se tuvo como resultado la combinación de hiperparámetros con mejor desempeño acorde a las métricas de validación cruzada, la cual se presenta a continuación:

- activation = 'tanh'
- solver = 'lbfgs'
- alpha = 0.001
- learning_rate_init = 0.01

Por último, se evaluó el modelo final prediciendo el conjunto de test, obteniendo como resultado un error RMSE de 122,135,650 COP. Esto significa que las predicciones del modelo final se alejan en promedio a dicho valor, con respecto al valor real.

7.3.3 XGBoost

El último algoritmo que se empleó fue el XGBoost, utilizando el mismo data frame del modelo Random Forest (Figura 10), donde al igual que los algoritmos anteriores, se construyó una arquitectura basada en hiperparámetros influyentes para el entrenamiento del modelo, los cuales se mencionan con una breve descripción en el capítulo de Metodología.

Dicho esto, el algoritmo cuenta con 9 hiperparámetros que se debieron de optimizar para obtener mejores predicciones, ya que si se toman los valores de los hiperparámetros por defecto, el modelo tiende a tener un margen de error alto en las predicciones.

Optimización de hiperparámetros del XGBoost

Dada la elevada cantidad de hiperparámetros que tiene el modelo XGBoost, se empleó una estrategia de grid search para identificar la mejor combinación de hiperparámetros con la que se tienen mejores predicciones.

Para ello, se estipulo ciertas regiones de búsqueda, para cada hiperparámetro con base al artículo de (Rodrigo y Ortiz, 2022). Por lo cual, el método grid search realizo las múltiples combinaciones de hiperparámetros, escogiendo la combinación de mejores resultados. A continuación, se presenta dicha combinación de hiperparámetros.

- Booster = 'gbtree'
- Nthread = 4
- Objective = 'reg:squarederror'
- learning_rate = 0.01
- max_depth = 4
- min_child_weight = 4
- subsample = 0.5
- colsample_bytree = 1
- _estimators = 500

Una vez identificado los mejores hiperparámetros se entrena el modelo indicando los valores óptimos en sus combinaciones. Por último, se utilizó el conjunto de test para evaluar la capacidad predictiva del modelo, lo cual nos arrojó un RMSE de 52,228,275 COP. Esto significa que las predicciones del modelo final se alejan en promedio a dicho valor, con respecto al valor real.

7.4 Elección del modelo

Con base a las métricas de desempeño (RMSE y R^2) y los criterios de evaluación de modelos estadísticos (AIC, BIC y DIC) empleadas en los 3 modelos, se obtuvo los siguientes resultados, los cuales tiene como objetivo elegir el modelo con mejor desempeño. A continuación, se presenta en el Cuadro 3 el resultado de las métricas de desempeño generadas por cada uno de los modelos

Cuadro 3: Resultados obtenidos según las métricas de desempeño.

	Modelos		
Metricas de desempeño	**Random Forest**	**Redes Neuronales Profundas**	**XGBoost**
RMSE	53,223,393	122,135650	52,228,275
R^2	0.93	0.53	0.9
AIC	3409.143	3432.9759	3897.8516
BIC	3986.205	4001.92	4223.1296
DIC	16.0906	16.1285	44.78

De acuerdo a los resultados presentados en la Cuadro 3, podemos elegir el modelo con mejor desempeño, el cual es el Random Forest, seguido el XGBoost y por último el de Redes Neuronales Profundas. Debido a que en el Random Forest su coeficiente de determinación es mejor a los otros dos modelos, lo que significa un mejor ajuste a los precios de los inmuebles en el data frame.

Según el RMSE, el modelo Random Forest vuelve a obtener un desempeño superior a los otros dos modelos, esto da a entender que el valor de la predicción de precio del inmueble, es más cercano respecto al valor real y conocido del inmueble.

Finalmente, los valores (AIC, BIC y DIC) más bajos fueron obtenidos por el modelo Random Forest, lo que indica que es el mejor modelo para predecir el precio de los inmuebles. El modelo de Random Forest incluye un bosque de arboles que puede modelar relaciones complejas entre las variables predictoras y la variable respuesta, lo que resulta en un mejor ajuste que los otros dos modelos.

Capítulo 8

8. Conclusiones y Recomendaciones

En este trabajo se logró el planteamiento, construcción e implementación de métodos automáticos, a partir de la utilización de una técnica de Web Scraping, técnica ejecutada por la herramienta selenium para la recolección de la información, la cual fue de gran utilidad para el entrenamiento, prueba y validación de los modelos ajustados por 3 algoritmos de Machine Learning (Random Forest, XGBoost y Redes neuronales profundas). Donde se eligió el mejor modelo ajustado a los datos teniendo en cuenta las métricas de desempeño R^2, RMSE y CV.

De esta manera, se concluyó que el mejor modelo para la predicción del valor comercial de un inmueble en la ciudad de Neiva es el de random forest, al presentar valores óptimos en las métricas de desempeño. Esto demuestra que el modelo de random forest presenta un mayor ajuste a los valores de los inmuebles del conjunto de datos, una mayor proximidad entre los datos pronosticados con los observados y finalmente una menor dispersión.

A partir de la recolección de la información con web scraping y de una transformación de ella misma utilizando Python, se elaboró un data frame con un total de 379 inmuebles y 12 columnas. Sin embargo, se reconoce la limitación en la ejecución del web scraping, dado que solo se logró la recolección para una página web inmobiliaria; por lo cual no se pudo comparar la calidad de la información obtenida en distintas páginas web y evaluar si se tienen similitudes en la información que presenta. No obstante, el número de inmuebles y columnas fue óptimo, debido a que supero el mínimo de los datos esperados (248 inmuebles) para el análisis de la información y ajuste de modelos que se tiene en cuenta en una investigación (Zhao *et al.*, 2019).

Según los resultados presentados, mediante la implementación de un modelo se consiguió determinar el valor comercial de los inmuebles en la ciudad de Neiva , ofreciendo mejores predicciones de manera precisa y rápida, sin la ayuda de un perito o avaluador. De esta forma se puede dar una herramienta a las personas que quieran invertir o comprar una vivienda en este municipio.

Por otro lado, quedan recomendaciones futuras de trabajo, el uso de librerías más eficientes en el proceso del web scraping como por ejemplo “Scrapy”, con el objetivo de no ser bloqueados por diferentes páginas web y así poder aumentar el número de inmuebles al data frame.

Por último, queda pendiente la implementación de algoritmos de machine learning de aprendizaje profundo, con la asignación de puntajes estéticos a partir de imágenes y combinado con Random Forest, para mostrar una mejora del rendimiento en la predicción del valor comercial del inmueble.

Capítulo 9

Referencias

Akaike, H. (1973). Information theory and an extension of the maxmum likelihood principale. En *2ˆ¡nd¿international symposium of information theory* (pp. 267–281).

Anderson, D., y Burnham, K. (2004). Model selection and multi-model inference. *Second. NY: Springer-Verlag*, *63*(2020), 10.

Antón, A. (2020). *Predicción del precio en el mercado de viviendas en la ciudad de valencia mediante redes neuronales en el año 2020* (Tesis Doctoral no publicada). Universitat Politècnica de València.

Borrero Ochoa, Ó. (2000). Avalúos de inmuebles y garantías. *Bogotá: Bhandar Editores Ltda*.

Breiman, L. (2001). Statistical modeling: The two cultures (with comments and a rejoinder by the author). *Statistical science*, *16*(3), 199–231.

Carreño, J. (2018). El avalúo y su importancia en el ámbito empresarial. *Científica Multidisciplinaria*, *6(1)*, 95–104.

Carvalho, F. P., Martins, P. R., y Andrade, R. (2020). Web scraping as a tool for data analysis: A systematic review of literature. *Journal of Information Systems Engineering and Management*, *5(1)*, 1–13.

Celeux, G., Forbes, F., Robert, C. P., y Titterington, D. M. (2006). Deviance information criteria for missing data models.

De La Hoz, E. J., De La Hoz, E. J., y Fontalvo, T. J. (2019). Metodología de aprendizaje automático para la clasificación y predicción de usuarios en ambientes virtuales de educación. *Información tecnológica*, *30*(1), 247–254.

Del Rio, J. (2021). *Bucaramanga, entre las ciudades con más viviendas vendidas este año.* Descargado de https://www.vanguardia.com/economia/local/bucaramanga-entre-las-ciudades-con-mas-viviendas-vendidas-este-ano-JD4498902

Ejea Carbonell, D. G., y Alcalá Nalvaiz, J. T. (2017). Árboles de regresión. algunos algoritmos y extensiones a métodos de consenso.

Espinosa, J. J. (2020). Application of random forest and xgboost algorithms based on a credit card applications database. *Ingeniería, investigación y tecnología*, *21*(3).

Gelman, A., Carlin, J. B., Stern, H. S., y Rubin, D. B. (2014). *Bayesian data analysis (vol. 2).* Taylor & Francis Boca Raton.

Geltner, D., Miller, N., Clayton, J., y Eichholtz, P. (2013). *Commercial real estate analysis and investments. cengage learning.* Inc.

González, L. (2018). *Evaluando el error en los modelos de regresión. aprendeia.* Descargado de https://aprendeia.com/evaluando-el-error-en-los-modelos-de-regresion/

Grajales, Y. V., y cols. (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* (Tesis de Master no publicada). Escuela de Ingenierías.

Hastie, T., Tibshirani, R., y Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2). Springer.

Ho, W. K., Tang, B.-S., y Wong, S. W. (2021). Predicting property prices with machine learning algorithms. *Journal of Property Research*, *38*(1), 48–70.

Huang, Y. (2019). Predicting home value in california, united states via machine learning modeling. *Statistics, optimization & information computing*, *7*(1), 66–74.

Jha, S. B., Pandey, V., Jha, R. K., y Babiceanu, R. F. (2020). Machine learning approaches to real estate market prediction problem: a case study. *arXiv preprint arXiv:2008.09922*.

Kim, Y., Kim, J., y Kim, J. (2020). Data-driven competitive intelligence for e-commerce: A web scraping approach. *A web scraping approach. Journal of Business Research*, 238–246.

Konishi, S., y Kitagawa, G. (2008). Information criteria and statistical modeling.

Kozak, M. (2015). Métodos y técnicas de avalúo de bienes inmuebles. *del Instituto de Investigación de la Facultad de Ingeniería Geológica, Minera, Metalúrgica y Geográfica*, *18(35)*, 37–49.

Lee, M. D., y Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge university press.

Maisueche, A. (2019). Utilización del machine learning en la industria 4.0.

Martín, D. (2021). *Aplicación de modelos de aprendizaje automático en microcontroladores* (B.S. thesis). Universitat Politècnica de Catalunya.

Martínez, D. N., y Téllez, V. J. (2021). Método automático para la predicción del avalúo comercial de un inmueble en la ciudad de bogotá.

Martínez, D. (2012). Bienes y derechos reales. *Universidad Externado de Colombia.*, 107.

Martínez, J. (2020). *Error cuadrático medio para regresión. iartificial.* Descargado de https://www.iartificial.net/error-cuadratico-medio-para-regresion/

Mitchell, T. M., y cols. (2007). *Machine learning* (Vol. 1). McGraw-hill New York.

Pérez, C. (2019). El avalúo como disciplina y su relevancia en la economía. *Economía Administración*, *4(7)*, 62–73.

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., y Liu, P. (2021). Xgboost optimized by adaptive particle swarm optimization for credit scoring. *Mathematical Problems in Engineering*, 1–18.

Ripley, B. D. (2007). *Pattern recognition and neural networks*. Cambridge university press.

Rodrigo, J. A. (2020). Random forest con python. *Ciencia de Datos*.

Rodrigo, J. A. (2021). Redes neuronales con python. *Ciencia de Datos*.

Rodrigo, J. A., y Ortiz, J. (2022). *Skforecast: time series forecasting with python and scikit-learn.*

Spiegelhalter, D., Nicola, G., Carlin, B. P., y van der Linde, A. (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *64*(4), 583–639.

Torres, A. (2017). La importancia del avalúo en la determinación del valor de los bienes. *de Administración y Finanzas*, *6(1)*, 41–52.

Torres, C., y Mora, M. (2019). Evaluación de los determinantes del precio de los inmuebles en colombia. *Cuadernos de Administración*, *35(64)*, 131–146.

Valencia, J. (2014). Derecho civil. parte general. *Temis*.

Wu, Y., Li, H., y Duan, Y. (2018). A review of housing price prediction models. , *19*, 226–233.

Zhao, Y., Chetty, G., y Tran, D. (2019). Deep learning with xgboost for real estate appraisal. En *2019 ieee symposium series on computational intelligence (ssci)* (pp. 1396–1401).

Capítulo 10

10. Anexo

En el siguiente enlace se encuentra un Drive donde se puede evidenciar el data frame y el código utilizado para la ejecución de los modelos en el entorno Google Colaboraty:

https://drive.google.com/drive/folders/1yFP-0SYgn_RTu3Vmc_iTeoxMRasnVsk0?usp=share_link