*Master in Management + Analytics*

# Desarrollo de modelos para la predicción de los precios del ganado vacuno argentino

**Autoría**: Arnaudo, Damián Ezequiel
**Año**: 2025

UNIVERSIDAD TORCUATO DI TELLA

MASTER IN MANAGEMENT + ANALYTICS

# DESARROLLO DE MODELOS PARA LA PREDICCIÓN DE LOS PRECIOS DEL GANADO VACUNO ARGENTINO

**TESIS**

Damián Ezequiel Arnaudo

Diciembre 2024

Tutor: Ignacio Pace

**Resumen**

La producción de carne vacuna no solo es una parte importante de la economía argentina, también es una parte fundamental de la dieta de los argentinos y un símbolo ante el mundo. Pero a pesar de ser un producto tan importante muchas veces se desconoce el esfuerzo que requiere producirlo y la volatilidad que puede tener su precio. Cuando un productor decide producir ganado vacuno se está involucrando en un proceso prolongado en el tiempo; este ciclo productivo puede durar al menos dos años y comprende distintas etapas desde que el animal nace hasta que está listo para su comercialización. Una vez transcurrido este ciclo, el productor tiene la difícil tarea de buscar precio que represente el esfuerzo realizado durante todo el ciclo.

El Mercado Agroganadero (MAG), antes conocido como Mercado de Liniers, es un mercado concentrador de hacienda, formador y orientador de los precios ganaderos en todo el país. El MAG funciona similar al de la bolsa de comercio; donde los únicos autorizados a operar se denominan consignatarios; estos funcionan de intermediarios entre el productor y el mercado. Pero a diferencia de la bolsa de comercio, donde cada producto operable ya se encuentra disponible, los consignatarios deben conseguir productores dispuestos a vender su hacienda en las diferentes categorías que se comercializan en el MAG.

El valor de la hacienda depende de diferentes variables, algunas en las que el productor puede tener injerencia como es la categoría del ganado (macho entero joven, novillito, novillo, vaca, vaquillona, toro), la estacionalidad o el peso. Pero también existen variables referidas al escenario económico y climático del país que afectan al precio y tanto el productor como el consignatario no pueden predecir.

Este trabajo busca desarrollar una herramienta que pueda ayudar a los productores a comprender mejor las variables que influyen en el precio del mercado ganadero y optimizar los beneficios en cada venta. Y para esto se recopilo información desde 2019 a 2022 del MAG y de las diferentes variables que consideramos necesarias para explicar el comportamiento de los precios del ganado vacuno. Y se pusieron a prueba un modelo XGBoost y un modelo de Red Neuronal obteniendo no solo resultados con un nivel admisible de predicción, además, se logró analizar la eficiencia de un modelo frente al otro.

**Abstract**

Beef production is not only an important part of the Argentine economy, but also a fundamental part of the Argentine diet and a symbol before the world. But despite being such an important product, the effort required to produce it and the volatility of its price are often unknown. When a producer decides to produce cattle, he is getting involved in a long process; this productive cycle can last at least two years and includes different stages from the time the animal is born until it is ready to be marketed. Once this cycle has elapsed, the producer has the difficult task of seeking a price that represents the effort made during the entire cycle.

The Mercado Agroganadero (MAG), formerly known as Mercado de Liniers, is a cattle market that concentrates livestock, and is responsible for the formation and orientation of cattle prices throughout the country. The MAG functions similarly to the stock exchange, where the only ones authorized to operate are called consignees, who act as intermediaries between the producer and the market. But unlike the stock exchange, where each tradable product is already available, the consignees must find producers willing to sell their cattle in the different categories that are traded in the MAG.

The value of the cattle depends on different variables, some of which the producer can influence, such as the category of the cattle (young entire male, young steer, steer, calf, cow, heifer, bull), seasonality or weight. But there are also variables related to the economic and climatic scenario of the country that affect the price and that both the producer and the consignee cannot predict.

This work seeks to develop a tool that can help producers to better understand the variables that influence the price of the cattle market and optimize the benefits in each sale. And for this we collected information from 2019 to 2022 from MAG and the different variables that we consider necessary to explain the behavior of cattle prices. An XGBoost model and a Neural Network model were tested, obtaining not only results with an admissible level of prediction, but also analyzing the efficiency of one model compared to the other.

# Índice

Índice de Ecuaciones ..... 5

Índice de Figuras ..... 5

Índice de Tablas ..... 6

1. Introducción ..... 7

    1.1. Contexto ..... 7

    1.2. Problema ..... 9

    1.3. Objetivo ..... 10

2. Datos ..... 11

3. Metodología ..... 18

    3.1. XGBoost ..... 18

    3.2. Redes Neuronales ..... 25

4. Resultados ..... 30

    4.1. XGBRegressor ..... 30

    4.2. Redes Neuronales ..... 36

    4.3. XGBRegressor vs Redes Neuronales ..... 41

5. Conclusiones ..... 42

6. Uso Práctico ..... 43

7. Referencias ..... 45

Apéndice A. Compilado y Normalizado ..... 45

Apéndice B. Operaciones Totales x Consignatario ..... 46

Apéndice C. Operaciones Totales x Departamento (75%) ..... 47

Apéndice D. Grid Search ..... 48

# Índice de Ecuaciones

**Ecuación 1.** Proceso de boosting ..... 18
**Ecuación 2.** Boosting, modelo Inicial ..... 19
**Ecuación 3.** Boosting, cálculo de residuos ..... 19
**Ecuación 4.** Boosting, modelo final ..... 19
**Ecuación 5.** Boosting, funcion de perdida ..... 19
**Ecuación 6.** Boosting, actualización de los pesos ..... 19
**Ecuación 7.** Red Neuronal ..... 25
**Ecuación 8.** Red Neuronal, minimización de la función de perdida MSE ..... 26
**Ecuación 9.** Red Neoronal, descenso del gradiente ..... 26
**Ecuación 10.** Red Neuronal, mitigación de sobreajuste ..... 26
**Ecuación 11.** Red Neuronal, representación ciclica (seno y cosceno) del tiempo (nro de semana y año) ..... 28

# Índice de Figuras

**Figura 1**. Variación del Precio en el Tiempo x Categoría ..... 13
**Figura 2.** Variación del Precio x Categoría ..... 14
**Figura 3.** Precio vs Cotizaciones (Enero 2019 - Diciembre 2022) ..... 15
**Figura 4.** Precio vs Precipitaciones (Promedio por Semana) ..... 16
**Figura 5.** Precio vs Temperaturas (Promedio por Semana) ..... 16
**Figura 6.** Operaciones Totales x Semana ..... 17
**Figura 7.** Variables Categóricas ..... 20
**Figura 8.** Dataset tras One-Hot Encoding ..... 21
**Figura 9.** Dataset repartido en train y test ..... 22
**Figura 10.** Rendimiento del modelo XGBRegressor ..... 23
**Figura 11.** Performance Primer XGBRegressor ..... 30
**Figura 12.** Importancia de Características en el modelo XGBRegressor ..... 32
**Figura 13.** Precios Reales VS Predicciones del modelo XGBRegressor ..... 33
**Figura 14.** Histograma de Errores XGBRegressor ..... 34
**Figura 15.** Performance XGBRegressor Final ..... 35
**Figura 16.** Performance Primer Red Neuronal ..... 37
**Figura 17.** Permutation Importance Redes Neuronales ..... 38
**Figura 18.** Precios Reales VS Predicciones del modelo Redes Neuronales ..... 39

**Figura 19.** Histograma de Errores Redes Neuronales ..... 40

# Índice de Tablas

**Tabla 1.** Head & Tail del Dataset Compilado y Normalizado ..... 12
**Tabla 2.** Resultados Ambos Modelos (Train) ..... 41
**Tabla 3.** Resultado Ambos Modelos (Test) ..... 42

# 1. Introducción

## 1.1. Contexto

La aptitud de los suelos, la variedad de climas y la abundancia de recursos naturales a lo largo de extensas regiones confieren al país importantes ventajas comparativas para la producción de ganados y carnes bovinos. Esto ha hecho que la actividad haya ido evolucionando desde sus inicios, contribuyendo al crecimiento y desarrollo del país, sustentando las economías regionales, generando empleo; hoy la ganadería argentina es responsable de cerca de 5 de cada 100 empleos argentinos y 1 de cada 4 empleos en la agroindustria.

Y no solo es una parte importante de la economía argentina, también es una parte fundamental de la dieta de los argentinos y un símbolo ante el mundo. Abasteciendo al mercado interno e insertándose en el contexto internacional como proveedor de alimentos se ha convertido en una de las actividades más relevantes del Sistema Agroalimentario Argentino. En 2023 la producción de carne vacuna fue de 3.286.135 toneladas, de la cuales se exportaron 960.000 toneladas, representando el 7,5% de las exportaciones nacionales.

A pesar de ser un producto tan importante muchas veces se desconoce el esfuerzo que requiere producirlo y la volatilidad que puede tener su precio. El ciclo productivo de la carne en Argentina es un proceso complejo que incluye varias y diferentes etapas que, a su vez, se complementan. Esto va desde la cría y el cuidado del ganado en una etapa inicial. Continua en una segunda etapa en la que los productores seleccionan el ganado para su posterior engorde y ceba, en esta etapa se alimentan y cuidan hasta que alcancen el peso adecuado para la faena en busca de maximizar el rendimiento de carne por animal. Y finalizando el ciclo con la distribución y comercialización de la carne. Una característica muy interesante de este ciclo es su duración, la cual viene determinada por la lentitud del proceso productivo, debido a las restricciones del ciclo biológico del animal.

La cadena de comercialización de ganados y carne vacuna tiene la característica de desarrollar diferentes actividades en su interior desde la etapa de cría hasta el consumidor final por medio de diversos canales. Los destinos posibles de la hacienda vendida, cuando se trata de crías, pueden incluir el engorde, la reposición de vientres o, eventualmente, la cría de toros. Una vez que el ganado alcanza el peso mínimo adecuado, su destino será la faena. El engorde se trata de terneros destetados que son vendidos para su terminación, mientras que la faena comercializa la hacienda terminada. A su vez, las vías de comercialización pueden ser directas o indirectas. La vía directa es cuando la operación se realiza entre productores o entre el productor y el establecimiento de faena. Y la vía indirecta es cuando la misma operación la realiza un consignatario que hará de intermediario entre ambas partes por medio de un remate de feria o mercado concentrador.

Existen tres mercados concentradores en el país, El Mercado Agroganadero (MAG), antes conocido como Mercado de Liniers, el Mercado de Córdoba y el Mercado de Rosario (ROSGAN). Se ubican en torno a los centros urbanos más poblados del país donde la demanda es más concentrada. El MAG es el más importante en la comercialización con destino a faena, opera a diario de lunes a viernes, y los bovinos vendidos en el mercado son enviados inmediatamente a

faena. Por medio del sitio web del mercado, todo remitente de hacienda puede seguir su venta en forma inmediata en cuanto a kilaje, precio, comprador, etc.

El Mercado Agroganadero (MAG), además de ser un mercado concentrador de hacienda, es formador y orientador de los precios ganaderos en todo el país. El MAG funciona similar a la bolsa de comercio; donde los únicos autorizados a operar son los consignatarios, funcionando como intermediarios entre el productor y el mercado. Pero a diferencia de la bolsa de comercio, donde cada producto operable ya se encuentra disponible, los consignatarios deben conseguir productores dispuestos a vender su hacienda en las diferentes categorías que se comercializan en el MAG.

El valor de la hacienda depende de diferentes variables, algunas en las que el productor puede tener injerencia como es la categoría del ganado (macho entero joven, novillito, novillo, vaca, vaquillona, toro), la estacionalidad o el peso. Pero también existen variables referidas al escenario económico y climático del país que afectan al precio y tanto el productor como el consignatario no pueden predecir. Por lo que muchas veces un productor entrega su hacienda a un consignatario sin estar seguro de a qué precio se venderá, si es el mejor momento para hacerlo, o si el peso actual de su hacienda es el más rentable.

A pesar de la importancia que la ganadería vacuna tiene para Argentina no se hallaron trabajos similares al que se va a proponer más adelante. Lo más similar es un artículo titulado "Impacts of changes in market fundamentals and price momentum on hedging live cattle", realizado por Coffey, Tonsor y Schroeder (2018) en Colorado Iowa, Estados Unidos, que busca predecir el valor de contrato de futuros sobre ganado vivo negociado en la Bolsa Mercantil de Chicago que se utiliza ampliamente para cubrir el riesgo del precio al contado del ganado vivo. Para lograr esto el estudio cuantifica los efectos de los cambios en los factores económicos fundamentales y las tendencias de los precios en la predictibilidad de la base del ganado vivo.

Al ser tan diferente la forma de comercializar el ganado vivo en Argentina respecto a Estados Unidos, al no usar contrato de futuros en nuestro país, se analizaron otros productos que son igual de representativos para un país como es la carne para el nuestro y se halló un trabajo titulado "Predictores del precio de maíz blanco en Jalisco y Michoacán", López-García, Martínez-Damián y Arana-Coronado (2022) donde analizan los factores que influyen en la determinación del precio del maíz blanco en los estados de Jalisco y Michoacán, dos de las principales regiones productoras de este cereal en México. A pesar de existir un indicador del precio proporcionado por las bolsas internacionales, en México no existe una bolsa que proporcione una señal adecuada sobre el comportamiento futuro de los precios del maíz blanco en este país. Es por esto que mediante el uso de modelos autorregresivos integrados de media móvil (ARIMA) los autores se enfocan en identificar y modelar los predictores claves que afectan la variabilidad de los precios del maíz blanco en estos mercados regionales.

Otra alternativa fue tomar como referencia dos análisis hechos sobre ciertos activos financieros como son el caso de la plata, el oro y el cobre. El primero de este análisis surge en Perú y es titulado *"Desarrollo de un modelo para la predicción del precio del cobre empleando herramientas de Machine Learning"*, Fosca Gamarra *(2020)* elige el cobre como eje del análisis ya que Perú, país de donde surge el artículo, es el segundo mayor productor del mundo. En este trabajo estructuran dos modelos sobre un conjunto de datos históricos para poder predecir el precio del *commodity*: uno de regresión lineal y otro SVR, para luego centrarse en optimizar el modelo SVR a través de la implementación de algoritmos de selección de hiperparámetros.

Mientras que el segundo estudio titulado *"Predictor alternativo del precio de los activos financieros. El caso de la plata y el oro"*, Martínez, Améstica-Rivas, Parisi y Gurrola (2021) abordan el análisis de factores alternativos que pueden predecir el comportamiento de los precios de los activos financieros, específicamente en los mercados de la plata y el oro. Utilizando información de los precios de cierre semanales del oro y de la plata durante el período 2015 - 2018, busca comprobar la eficacia del modelo ARIMA optimizado con fuerza bruta operacional para pronosticar en el mercado financiero el precio de estos metales preciosos, que tradicionalmente son considerados refugios de valor en tiempos de incertidumbre económica.

Finalmente, se revisaron estudios sobre mercados estacionarios, como el inmobiliario, encontrándose un trabajo titulado *"Extracción de datos y modelo predictor para el precio de alquiler de viviendas de Barcelona"*, realizado por Óscar Fernández Batalla (2021), cuyo objetivo es predecir el precio de alquiler de viviendas en la ciudad de Barcelona. Un aspecto particularmente interesante de este estudio, que no se había observado en investigaciones previas, es el uso de un programa basado en *web scraping* para la obtención de datos. Esta herramienta permitió navegar por portales inmobiliarios y extraer los atributos presentes en los distintos anuncios publicados, lo que facilitó la recopilación de información relevante para el análisis predictivo.

## 1.2. Problema

Cuando un productor decide producir ganado vacuno se está involucrando en un proceso prolongado en el tiempo; este ciclo productivo puede durar al menos dos años y comprende distintas etapas desde que el animal nace hasta que está listo para su comercialización.

Hasta que el ganado alcanza el peso adecuado para la faena el productor debe afrontar diversos costos como pueden ser el alimento y las vacunas que requieren los animales, y el transporte necesario para llevarlo hasta los mercados concentradores entre otros. Sumado a esto el productor puede correr diversos riesgos, que van desde la falta de agua para su ganado hasta la pérdida de un animal.

A pesar de tener certeza de los costos que va a afrontar y de conocer el mercado como al cliente a quien ira dirigido su ganado una vez transcurrido este ciclo, el productor no cuenta con la certeza de cuan beneficioso puede ser todo su esfuerzo, en especial en escenarios de tanta volatilidad como el actual donde vender una semana antes o una semana después puede significar un cambio sustancial en el precio de la hacienda.

El Mercado Agroganadero (MAG) desempeña un papel clave como referente de precios para los productores al momento de comercializar su hacienda. En este contexto, el MAG elabora dos índices de gran relevancia: el Índice Novillo Mercado Agroganadero (INMG) y el Índice General Mercado Agroganadero (IGMAG). El INMG se calcula en función de todos los novillos en pie vendidos en un día determinado, específicamente aquellos lotes que cumplen con criterios como, por ejemplo, "novillos mestizos con peso superior a 430 kg". En contraste, el IGMAG se calcula como el promedio de todas las operaciones en pie realizadas en un día específico. De este modo, mientras que el primero es altamente específico y representa únicamente a una categoría particular de ganado vacuno, el segundo tiene un enfoque general y abarca todas las categorías. Sin embargo, ambos índices presentan limitaciones significativas, ya que cada categoría de ganado vacuno posee ciclos y precios particulares que no son adecuadamente

reflejados por estos índices, lo que reduce su utilidad en la toma de decisiones para los productores.

En la actualidad, los productores enfrentan una gran incertidumbre a la hora de determinar el precio de su ganado, debido a la falta de índices que reflejen adecuadamente las particularidades de cada categoría de ganado vacuno. Los índices actuales, al ser demasiado generales o demasiado específicos, no logran captar las variaciones propias de los ciclos y características de cada tipo de ganado, lo que dificulta la toma de decisiones informadas. Esta problemática genera un escenario en el que los productores no pueden contar con herramientas precisas que les permitan predecir de manera confiable los precios de su hacienda, lo que incrementa los riesgos asociados a sus transacciones comerciales.

## 1.3. Objetivo

El objetivo principal de este trabajo es desarrollar una herramienta analítica y predictiva que permita a los productores ganaderos comprender de manera más clara y profunda las variables que inciden en la determinación del precio de venta de sus animales en el mercado ganadero. Para ello, se integran tanto las variables intrínsecas del ganado, como la categoría (por ejemplo, terneros, vacas de cría, etc.) y el peso de los animales, como factores externos que impactan directamente en el contexto económico y productivo, tales como las condiciones meteorológicas (temperatura, lluvias), los costos asociados a la alimentación animal y la cotización del dólar, entre otros elementos. Este enfoque tiene como objetivo brindar una visión integral de los factores que afectan la fluctuación de los precios, permitiendo a los productores tomar decisiones más informadas y estratégicas al momento de comercializar su ganado.

Para alcanzar este objetivo, se realizó una recopilación de datos históricos correspondientes a las operaciones comerciales diarias que se efectuaron en el Mercado Agropecuario (MAG) durante un período de cuatro años (2019-2022). La base de datos obtenida no solo incluye información sobre las características del ganado, como su categoría y peso, sino también datos relativos a las condiciones meteorológicas (como la temperatura promedio y la cantidad de lluvia en el día de la operación), así como el valor del alimento (que incide en los costos de producción) y la cotización del dólar en cada jornada de venta. Esta amplia recopilación de información permite tener una visión más precisa y contextualizada de los factores que influyen en la determinación de los precios de venta del ganado vacuno.

Una vez que se integraron todas estas variables, se implementaron dos modelos predictivos avanzados: un modelo basado en el algoritmo XGBoost y otro utilizando Redes Neuronales Artificiales. Ambos modelos fueron entrenados y evaluados con el fin de predecir el precio de venta de los lotes de ganado con la mayor precisión posible. La comparación entre estos dos enfoques permitirá identificar cuál de ellos tiene un mejor desempeño predictivo en función de las características particulares de los datos y el comportamiento del mercado ganadero. Además, no solo se buscará la predicción precisa, sino también el análisis comparativo de la eficiencia de cada modelo, considerando variables como el tiempo de procesamiento, la capacidad de generalización y la robustez frente a fluctuaciones imprevistas en los datos de entrada.

El propósito final de este estudio es diseñar una herramienta práctica que pueda ser utilizada directamente por los productores ganaderos para optimizar su toma de decisiones. Esta herramienta proporcionará predicciones de precios basadas en las condiciones actuales del

mercado, facilitando así una mejor planificación de las ventas, maximización de beneficios y una mayor competitividad en un mercado cada vez más dinámico y globalizado. Además, se ofrecerán recomendaciones específicas sobre cuándo y cómo vender el ganado, basadas en la comprensión profunda de las interrelaciones entre las variables externas e internas, y su influencia en los precios de venta.

# 2. Datos

El *dataset* tiene origen en diferentes fuentes con el objetivo de conformar un escenario que contemple diferentes contextos que participan en la determinación del valor del ganado vacuno a la hora de su comercialización.

La columna vertebral del *dataset* proviene del Mercado Agroganadero (MAG), de donde se obtuvieron más de 500.000 operaciones realizadas desde enero 2019 hasta diciembre 2022. Cada operación representa un lote de animales que un productor envió al mercado a través de un consignatario para su comercialización. Las variables que trae son la fecha del día de operación, la localidad y provincia de donde proviene el ganado, la categoría del ganado, el número de cabezas que conforman el lote, el peso total del lote y el precio en pesos ($) por kilo al que se vendió (esta variable es la que se intentara poder predecir). Las categorías que comercializa el MAG para la Faena son Novillito, Novillo, Vaquillona, Vaca, Toro y Macho Entero Joven (MEJ), y dentro de cada una existe una subdivisión por rangos de kilos que puede pesar el animal. Del MAG también se obtuvo los rangos de kilos de cada categoría que determinan cada subcategoría, y con las variables ya descritas se determinó a cuál pertenece cada lote.

En el MAG es imprescindible que al final de cada día el número de animales operados en los distintos lotes sea el mismo que los animales ingresados. Pero a veces sucede que hay animales que sufren algún daño en el viaje hacia el mercado impidiendo que puedan ser comercializados dentro de un lote, debiendo ser vendidos de manera simbólica a precios ínfimos. Al ser los valores de estas operaciones muy diferentes a otros de las mismas características fue necesario depurarlos para evitar que pueden llegar a corromper la capacidad de previsión de los modelos.

Dado que el Mercado Agropecuario (MAG) informa el origen de los animales en cada lote, detallando localidad y provincia como variables categóricas nominales, se optó por cruzar estos datos con el Servicio de Normalización de Direcciones y Unidades Territoriales de Argentina, obtenido de datos.gob.ar, con el fin de transformarlas en variables numéricas. Esta modificación facilitó el manejo del origen de cada lote; sin embargo, al realizar el análisis por localidad, se observó un agrupamiento excesivamente amplio, y al hacerlo por provincia, se evidenció una categorización demasiado genérica.

Al segmentar los datos por localidad, se observó que no existían cantidades significativas de lotes en ninguna localidad específica. Sin embargo, en localidades relativamente cercanas geográficamente, los lotes presentaban características y precios similares. Por otro lado, al realizar la agrupación por provincia, se observó una tendencia opuesta: el número de lotes crecía significativamente, pero las características y los precios diferían considerablemente entre ciudades de una misma provincia ubicadas a gran distancia. En este sentido, no es lo mismo un

lote proveniente del norte de Buenos Aires que uno originado en el sur de la misma provincia. Por lo tanto, además de transformar los campos de localidad y provincia, se decidió incorporar el campo "departamento". De este modo, cada provincia fue subdividida, lo que permitió que las localidades cercanas con condiciones geográficas similares se agruparan en un mismo departamento, mejorando la precisión en el análisis territorial.

El segundo *dataset* busca representar el contexto económico al tomar la cotización diaria histórica del dólar para la compra y para la venta. De la página del Banco de la Nación Argentina se obtuvo la cotización diaria histórica del dólar oficial; mientras que de la página del diario Ámbito Financiero se obtuvo la cotización diaria histórica del dólar paralelo para la compra y para la venta. Y para manejar un único tipo de cambio oficial y un único tipo de cambio paralelo se calculó el promedio diario entre el valor de compra y el de venta.

Del Ministerio de Agricultura, Ganadería y Pesca (MAGyP) se adquirió la cotización mensual histórico del maíz por tonelada en el Mercado de Rosario que permitió sumar un enfoque del valor del alimento necesario para la hacienda. Al igual que sucede con los mercados de ganado vacuno existen los mercados que operan granos y que cumplen la función de concentradores y referentes de precios; de todos los comercios que hoy existe se tomó el de Rosario por tener una mayor relevancia al resto.

El último aspecto considerado para completar el *dataset* fue el contexto climatológico. Gracias al proyecto IRI creado por la *Columbia Climate School* se logró obtener el histórico diario de la temperatura mínima y la temperatura máxima y los milímetros de precipitación de cada departamento. Y como complemento se generaron tres nuevas variables, los milímetros precipitados al cuadrado, los milímetros precipitados acumulados y el número de días continuos sin lluvia.

Tras normalizar y transformar los distintos conjuntos de datos de manera individual se unificaron en un único *dataset* respetando la frecuencia diaria. Y como el periodo de tiempo con el que este trabajo buscará predecir el precio del ganado vacuno será semanal, se transformó la variable fecha en la variable *week* que contiene el número de la semana y el año. Finalmente, se creó la variable *week_num* con el número de la semana con la que se busca encontrar la estacionalidad de los precios a lo largo año. De esta manera, el *dataset* quedo conformado de la siguiente manera:

***Tabla 1.*** *Head & Tail del Dataset Compilado y Normalizado*

| | week | consignatario | depto | categoria | subcategoria | cabezas | kilos_prom | precio | Dólar | dolar_blue | maiz | lluvia | lluvia2 | lluvia_acumulada | dias_sin_lluvia | temp_max | temp_min |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2019-00 | 445 | 6084 | 3 | 30 | 8 | 391,25 | 50 | 37,7 | 39,5 | 5,54591 | 1,801312 | 3,244725 | 12,002372 | 0 | 26,57852 | 19,28792 |
| **1** | 2019-00 | 445 | 6084 | 3 | 30 | 7 | 370 | 52,8 | 37,7 | 39,5 | 5,54591 | 1,801312 | 3,244725 | 12,002372 | 0 | 26,57852 | 19,28792 |
| **2** | 2019-00 | 32 | 6448 | 10 | 44 | 8 | 380 | 47 | 37,7 | 39,5 | 5,54591 | 0,580093 | 0,336508 | 12,6637133 | 0 | 26,85949 | 18,52129 |
| **3** | 2019-00 | 32 | 6448 | 10 | 44 | 1 | 370 | 42 | 37,7 | 39,5 | 5,54591 | 0,580093 | 0,336508 | 12,6637133 | 0 | 26,85949 | 18,52129 |
| **4** | 2019-00 | 32 | 6448 | 9 | 43 | 18 | 435 | 34 | 37,7 | 39,5 | 5,54591 | 0,580093 | 0,336508 | 12,6637133 | 0 | 26,85949 | 18,52129 |
| **…** | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| **535235** | 2022-52 | 538 | 6119 | 9 | 42 | 13 | 400 | 220 | 179,25 | 344 | 42,6271 | 0,001753 | 3,07E-06 | 10,63561295 | 0 | 36,19468 | 19,08709 |
| **535236** | 2022-52 | 538 | 6119 | 9 | 42 | 3 | 290 | 200 | 179,25 | 344 | 42,6271 | 0,001753 | 3,07E-06 | 10,63561295 | 0 | 36,19468 | 19,08709 |
| **535237** | 2022-52 | 538 | 6119 | 9 | 42 | 8 | 336,25 | 230 | 179,25 | 344 | 42,6271 | 0,001753 | 3,07E-06 | 10,63561295 | 0 | 36,19468 | 19,08709 |
| **535238** | 2022-52 | 538 | 6672 | 4 | 32 | 16 | 303,13 | 345 | 179,25 | 344 | 42,6271 | 2,58E-05 | 6,68E-10 | 16,89954503 | 0 | 36,47673 | 15,56308 |
| **535239** | 2022-52 | 538 | 6672 | 4 | 32 | 9 | 360 | 325 | 179,25 | 344 | 42,6271 | 2,58E-05 | 6,68E-10 | 16,89954503 | 0 | 36,47673 | 15,56308 |

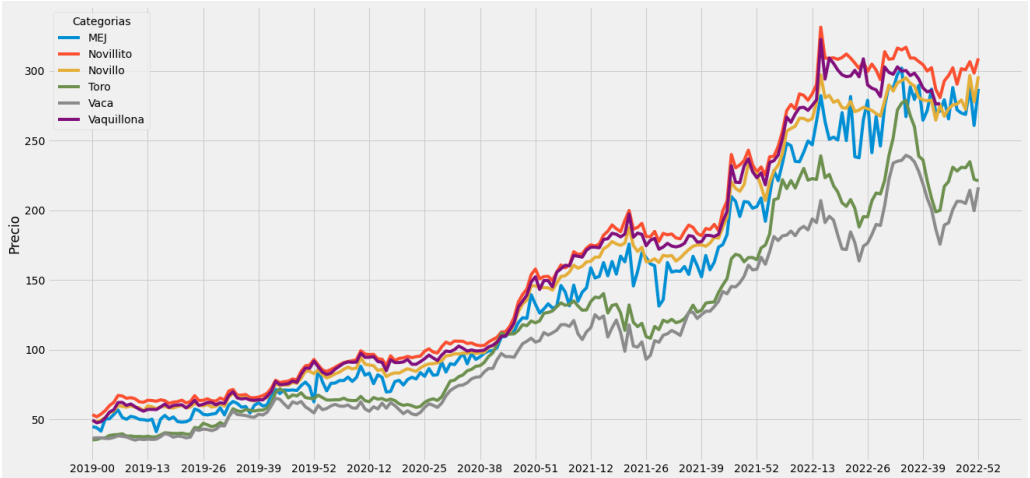
Con el *dataset* ya normalizado se generaron diferentes gráficos para poder comprender mejor la relación entre las variables, especialmente la relación con el precio que es la variable por predecir.

El gráfico presente en la Figura 1 fue diseñado para mostrar la evolución temporal de los precios de cada categoría de ganado comercializada en el MAG, con un enfoque en cómo estos precios cambian a lo largo del periodo de 2019-2022.

Dado el contexto económico de Argentina durante este período, caracterizado por altos niveles de inflación, fluctuaciones en la cotización del dólar y cambios en los costos de producción (como el precio de los alimentos y las condiciones climáticas), no es sorprendente que los precios del ganado muestren una tendencia ascendente a lo largo del tiempo. Este comportamiento es claramente visible en el gráfico de líneas (Figura 1), que refleja un aumento sostenido de los precios conforme se avanza en los años del *dataset*. A medida que se aprecia esta progresión, se hace evidente cómo el mercado ganadero fue afectado por estos factores externos, lo que refuerza la importancia de entender cómo los diferentes elementos (clima, economía, tipo de cambio) influyen en el precio final de la venta.

Este gráfico permite observar visualmente la dinámica del mercado en función del tiempo, ayudando a contextualizar los incrementos o descensos en los precios y proporcionando una base para explorar otras relaciones más profundas entre las variables internas del ganado y factores externos que afectan el mercado.

***Figura 1**. Variación del Precio en el Tiempo x Categoría*

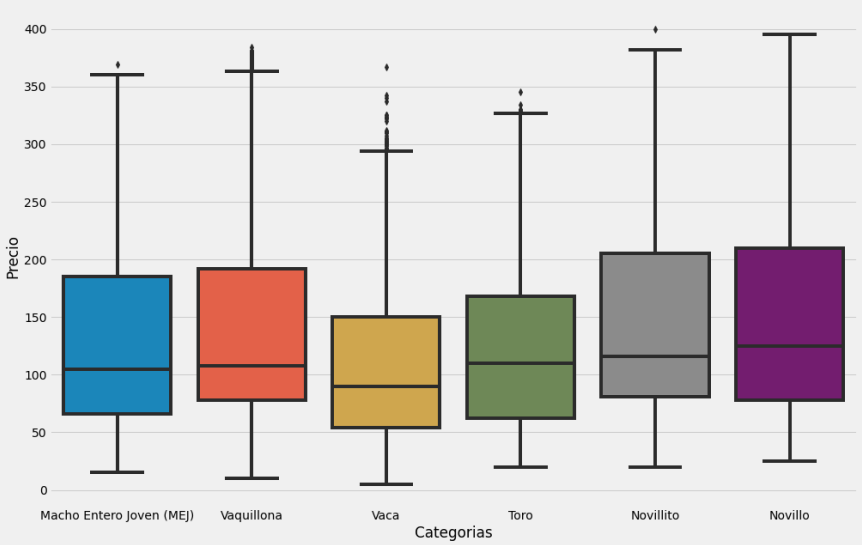



*Fuente: elaboración propia en base a datos del MAG.*

Para profundizar la comprensión del comportamiento de los precios en relación con las categorías de ganado se armó el *boxplot* de la Figura 2. Este gráfico permite observar que la mayor parte de los precios en cada categoría se concentra en un rango relativamente estrecho, que va desde los $50 hasta los $200, lo que indica una baja variabilidad en los precios dentro de cada categoría. Este comportamiento sugiere que, a pesar de las fluctuaciones generales en el mercado ganadero, los precios para una misma categoría de ganado tienden a mantenerse en

un rango predecible, con una variación moderada. En la mayoría de las categorías, la mediana de los precios se encuentra más cerca del cuartil inferior, en torno a los $100.
Es relevante señalar que, en la mayoría de las categorías de ganado, el *boxplot* no muestra valores atípicos (*outliers*), o bien muestra solo algunos pocos casos aislados. En ciertos casos, no hay ningún valor atípico visible. Este comportamiento sugiere que las transacciones en el mercado son bastante consistentes, con escasos casos extremos donde los precios se desvíen significativamente de la norma.

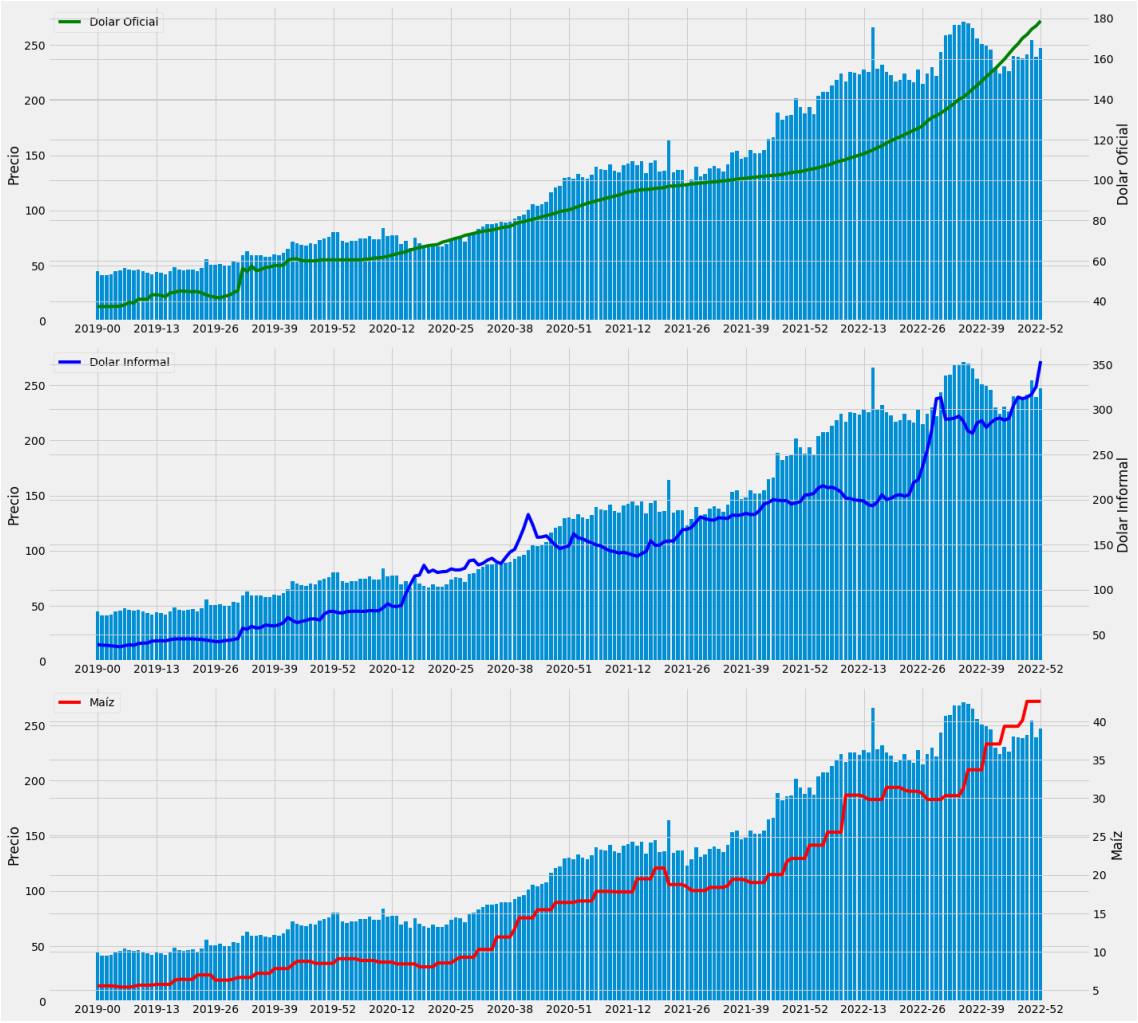
***Figura 2.*** *Variación del Precio x Categoría*




*Fuente: elaboración propia en base a datos del MAG.*

En el siguiente análisis (Figura 3) se compara el comportamiento de los precios con tres variables externas al MAG, el dólar oficial, el dólar informar, y el valor del kilo de maíz en el mercado local en transcurso de las semanas entre enero 2019 y diciembre 2022. En los tres casos se ve un crecimiento similar de los precios operados en el MAG con las variables económicas, pero a pesar de esto no podemos asegurar una relación directa, sino que es posible que la similitud se deba a que todas las variables fueron atravesadas por el mismo contexto económico que vivió el país.
Es importante destacar que el valor del kilo de maíz no solo creció de manera constante igual que los precios del MAG, sino que los movimientos de ambas variables son prácticamente idénticos, lo cual pude tomarse como un indicio de que existe una fuerte relación entre ambas variables. Dado que el maíz es un insumo fundamental para la alimentación del ganado vacuno, es probable que la relación sea tan estrecha como lo demuestra el gráfico.
Existe un periodo muy corto, correspondiente a las últimas semanas de 2022, donde el movimiento de ambas variables se desincroniza. Este hecho aislado coincide con la aparición del denominado “dólar soja” que representa la intervención del gobierno en la cotización del tipo de cambio para los productores que liquidaban granos. Dicho programa se dividió en cuatro etapas, comenzando en septiembre de 2022 y extendiéndose hasta mediados de 2023.

***Figura 3.*** *Precio vs Cotizaciones (Enero 2019 - Diciembre 2022)*



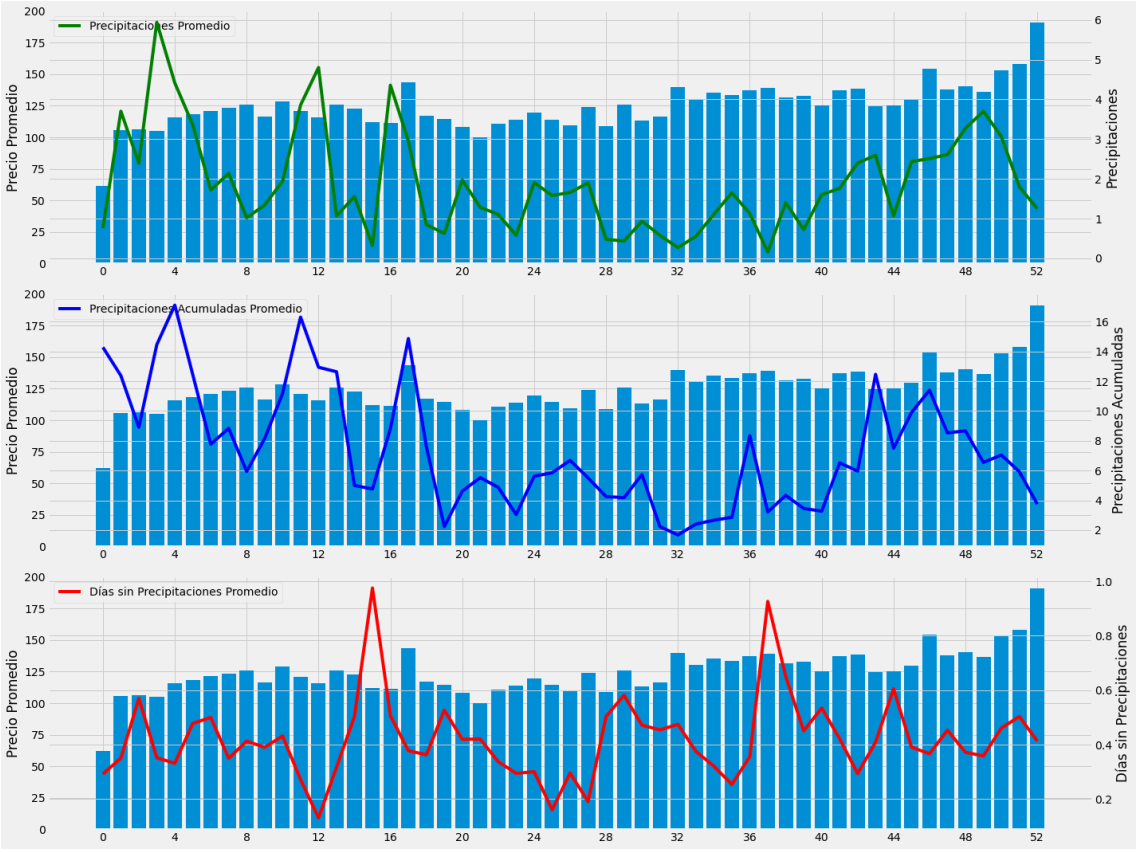

*Fuente: elaboración propia en base a datos del MAG, Banco de la Nación Argentina, diario Ámbito Financiero y MAGyP*

En los gráficos de las Figuras 4 y 5 volvemos a comparar los precios operados en el MAG con variables externas, primero lo hacemos con las precipitaciones y luego con las temperaturas. Estas variables tienen la característica de ser estacionales, es decir, que las estaciones del año traen cambios en el clima que son predecibles, como las bajas temperaturas en invierno y las bajas temperaturas en verano. Con las precipitaciones sucede lo mismo, hay estaciones donde el nivel de precipitaciones es mayor, mientras que en otras las precipitaciones escasean.

Ante esta característica de las variables se optó por tomar como medida de tiempo solo el numero de la semana en el año y calcular el promedio del precio como de las precipitaciones y las temperaturas máximas y mínimas en cada semana.

Para la construcción del gráfico comparativo entre precios y precipitaciones (Figura 4), además de la variable "milímetros precipitados", se consideraron también los milímetros precipitados acumulados y el número de días consecutivos sin lluvia. Aunque no se pudo identificar una relación significativa entre estas variables, resulta evidente la estacionalidad de las lluvias a lo largo del año.

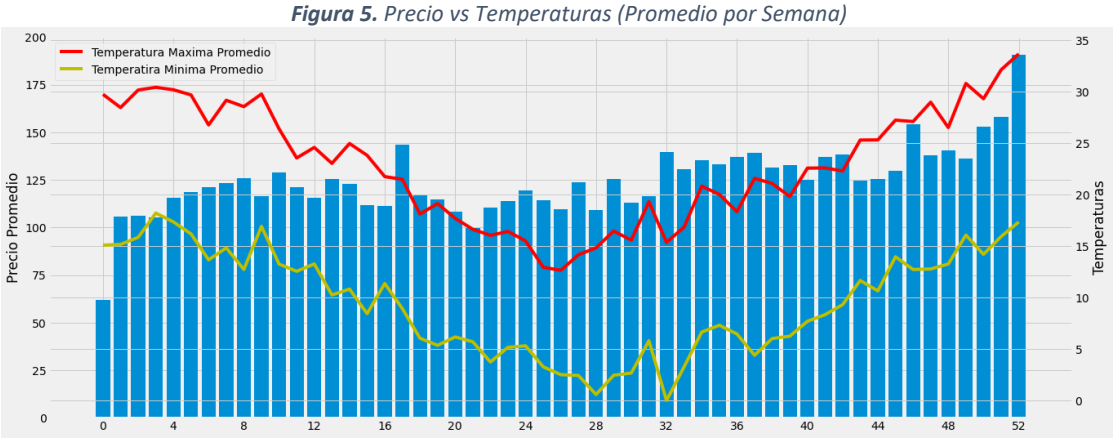
***Figura 4.*** *Precio vs Precipitaciones (Promedio por Semana)*




*Fuente: elaboración propia en base a datos del MAG y Columbia Climate School.*

Con respecto a las temperaturas, se observa un patrón estacional aún más marcado a lo largo de las 52 semanas del año. Sin embargo, resulta complejo establecer una relación directa entre las temperaturas y los precios. A pesar de que se aprecia una similitud entre el incremento de los precios y el aumento de la temperatura en la segunda mitad del año, dicho aumento podría estar más relacionado con las festividades que se celebran en los últimos meses del año y el inicio de las vacaciones, en lugar de una influencia directa del clima.

***Figura 5.*** *Precio vs Temperaturas (Promedio por Semana)*




*Fuente: elaboración propia en base a datos del MAG y Columbia Climate School.*

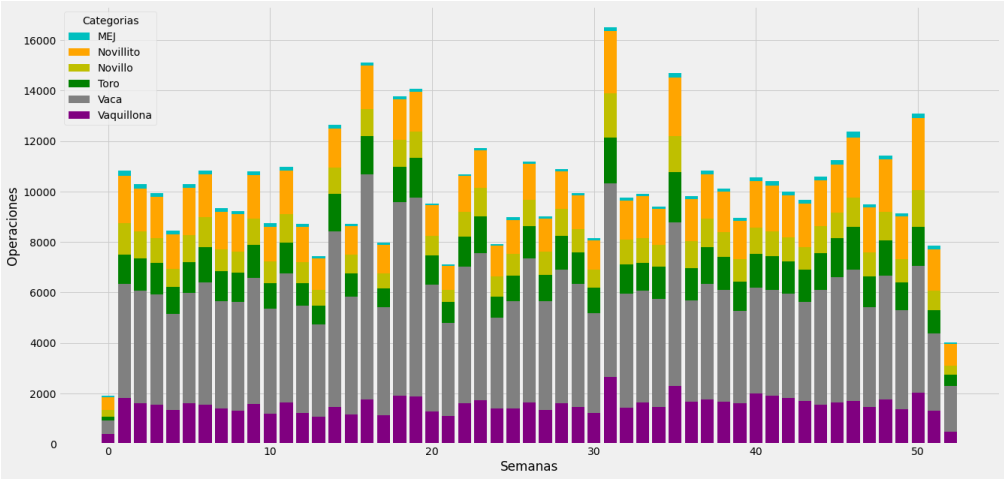
En el análisis de la participación de los consignatarios dentro del mercado, se elaboró un gráfico (Apéndice B) que permite visualizar la distribución del volumen de operaciones realizadas por cada uno de los 47 consignatarios registrados en el MAG entre enero de 2019 y diciembre de 2022. Este gráfico es crucial para entender cómo la concentración de operaciones de un pequeño grupo de consignatarios puede influir en el precio del ganado y la dinámica del mercado en general.

El análisis de los datos revela un patrón de alta concentración de operaciones, ya que el 50% del total de las transacciones están concentradas en apenas 10 consignatarios, lo que representa aproximadamente el 21% de todos los consignatarios del mercado. Este hallazgo indica que una porción significativa de las operaciones es dominada por un grupo reducido de actores.

En el Apéndice C, se continúa con el análisis del peso relativo de algunos valores dentro del conjunto de posibles que puede asumir una variable específica, y cómo estos factores inciden en la determinación del precio del ganado. En esta ocasión, se examina el origen geográfico del ganado, es decir, el departamento de procedencia de todos los lotes operados entre enero de 2019 y diciembre de 2022. Los resultados obtenidos indican que las operaciones de ganado se concentran principalmente en departamentos ubicados en la región centro del país, especialmente en las provincias de Buenos Aires, Córdoba, Entre Ríos y Santa Fe, destacándose particularmente el norte de la provincia de Buenos Aires.

Finalmente, mediante el gráfico presentado en la Figura 6, se puede visualizar la distribución semanal de las operaciones realizadas en el Mercado Agroganadero (MAG) entre enero de 2019 y diciembre de 2022, segmentadas por categoría. Este análisis permite no solo identificar la estacionalidad del mercado de ganado en general, sino también evaluar la variación de cada categoría a lo largo del año. Un hallazgo relevante de este estudio es la preeminencia de las operaciones de vaca en el total de las transacciones, así como la notable diferencia en el volumen de operaciones entre el primer y el segundo semestre del año, con un incremento significativo en la actividad durante la segunda mitad, seguido por una caída abrupta en las tres semanas comprendidas entre la Navidad y el Año Nuevo.

***Figura 6.*** *Operaciones Totales x Semana*




*Fuente: elaboración propia en base a datos del MAG.*

# 3. Metodología

Se optó por evaluar dos metodologías distintas con el objetivo de no solo obtener predicciones precisas, sino también para realizar una comparación entre ambas. Este enfoque permitió analizar tanto las fortalezas como las limitaciones de cada metodología, proporcionando una perspectiva más completa sobre su desempeño relativo. La implementación de estos modelos se basó en los principios establecidos por Friedman, Hastie y Tibshirani (2001), quienes detallan técnicas fundamentales en el diseño de modelos estadísticos en su obra *The Elements of Statistical Learning*.

## 3.1. XGBoost

Por su rendimiento superior en problemas de regresión y clasificación la primera metodología seleccionada fue XGBoost. Esta metodología tiene un enfoque de ensamblaje que mejora la precisión de modelos de aprendizaje supervisado. A diferencia de métodos como el *bagging*, que entrena modelos independientes, el *boosting* construye modelos de manera secuencial, donde cada nuevo modelo intenta corregir los errores del anterior. Su base se encuentra en el principio de utilizar modelos simples o "débiles", generalmente árboles de decisión poco profundos, para concluir en un modelo fuerte.

El proceso de *boosting* se basa en la minimización de una función de pérdida, que mide la discrepancia entre las predicciones del modelo y los valores reales. Esto se logra aplicando un algoritmo de optimización que ajusta de manera secuencial un nuevo modelo a los errores de las predicciones previas. Este enfoque permite reducir el sesgo y, en muchos casos, también la varianza.

El proceso de *boosting* puede ser formalizado de la siguiente manera. Dado un conjunto de datos $(x_i, y_i)$ donde i=1, 2,…,n , el objetivo es construir un modelo de predicción $\hat{y}$ que minimice una función de pérdida, como el error cuadrático medio (ECM) para problemas de regresión. La idea básica detrás del *boosting* es que, en cada iteración m, se entrena un modelo débil $h_m(x)$ que minimiza la pérdida ponderada:

***Ecuación 1.*** *Proceso de boosting*

$$\sum_{i=1}^{n} w_i \, \mathcal{L}(Y_i, h_m(X_i))$$

donde $w_i$ son pesos que reflejan la importancia de cada observación en la iteración actual. Los pesos se actualizan después de cada iteración, incrementándose para las observaciones mal predichas y decreciendo para las correctamente predichas.

En el contexto de la regresión, el algoritmo de *Boosting* se puede describir de la siguiente manera:

Se comenzó con un modelo inicial, típicamente la media de las observaciones Y:

***Ecuación 2.*** *Boosting, modelo Inicial*

$$F_0(x) = \arg min_c \sum_{i=1}^{n} \mathcal{L}(Y_i, c)$$

Para cada iteración m, se calcularon los residuos:

***Ecuación 3.*** *Boosting, cálculo de residuos*

$$r_{im} = Y_i - F_{m-1}(x_i)$$

Finalmente, se actualizó el modelo global combinando el modelo anterior con el nuevo modelo débil:

***Ecuación 4.*** *Boosting, modelo final*

$$F_m(x) = F_{m-1}(x) + \nu h_m(x)$$

donde ν es una tasa de aprendizaje que controla la contribución de cada modelo débil al modelo final. Este paso es crucial, ya que una tasa de aprendizaje demasiado alta puede llevar a sobreajuste, mientras que una tasa demasiado baja puede hacer que el modelo converja lentamente.

La elección de la función de pérdida $\mathcal{L}$ es fundamental en el boosting, ya que determina la forma en que se ajustan los modelos. En el caso de la regresión, la función de pérdida más comúnmente utilizada es el error cuadrático medio:

***Ecuación 5.*** *Boosting, funcion de perdida*

$$\mathcal{L}(y, \hat{y}) = (y - \hat{y})^2$$

Sin embargo, el *boosting* también puede aplicarse con otras funciones de pérdida, como la pérdida absoluta o pérdidas robustas que son menos sensibles a valores atípicos.

La actualización de los pesos en cada iteración se puede expresar en términos de la derivada de la función de pérdida con respecto a las predicciones del modelo:

***Ecuación 6.*** *Boosting, actualización de los pesos*

$$w_i^{(m)} = -\frac{\partial \mathcal{L}(Y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}$$

Este enfoque permite que el modelo se enfoque en las observaciones que son más difíciles de predecir, mejorando así la precisión general.

Para implementar el modelo de predicción, fue necesario realizar previamente un exhaustivo preprocesamiento de los datos, una fase crítica en cualquier proyecto de *machine learning*. Esta etapa es esencial porque tiene un impacto directo en la calidad, rendimiento y efectividad del modelo final. En términos generales, el preprocesamiento busca limpiar, transformar y preparar los datos de manera que el modelo pueda aprender de ellos de forma eficiente. Un conjunto de datos mal preparado puede llevar a modelos poco precisos o incluso a errores durante el entrenamiento y la predicción. Por lo tanto, una correcta preparación de los datos es crucial para el éxito del proyecto.

En este caso, el preprocesamiento comenzó con la limpieza de los datos. Durante esta fase, se identificaron y resolvieron varios problemas comunes en los *datasets* crudos, como la presencia de valores nulos en algunas filas y columnas. Estos valores nulos son problemáticos porque la mayoría de los modelos de machine *learning* no pueden manejarlos adecuadamente sin un tratamiento previo. Además, uno de los problemas más frecuentes que se abordó fue el escalado de las variables, ya que el modelo no puede procesar variables con rangos numéricos muy dispares. En este caso, por ejemplo, el precio del maíz estaba en toneladas, mientras que el precio del ganado estaba en kilos, lo que generaba grandes diferencias en la escala de los valores. Este tipo de disparidad puede afectar negativamente el rendimiento del modelo, ya que algunas variables pueden dominar la predicción solo por su escala. Por lo tanto, se realizó un escalado adecuado de las variables para garantizar que todas tuvieran un impacto equilibrado en el modelo.

Como los modelos generalmente no pueden manejar directamente las variables categóricas, como son el caso de las variables *consignatario, depto, categoría, subcategoria, week* fue necesario convertirlas en una representación numérica. Para abordar este problema, se utilizó una de las técnicas más comunes y efectivas en este tipo de situaciones: el *One-Hot Encoding*. Esta técnica convierte cada categoría en una nueva columna binaria, en la que cada columna representa una categoría única de la variable original. Así, en lugar de tener una sola columna con categorías de texto, el *dataset* se transforma en una serie de columnas binarias (0s y 1s) donde cada fila representa la pertenencia a una categoría específica. Por ejemplo, la variable *categoria* tiene seis categorías posibles, después de aplicar *One-Hot Encoding*, esta variable será representada por seis columnas, y cada fila tendrá un 1 en la columna correspondiente a su departamento y 0 en las otras. De esta forma, se puede representar la información categórica en una forma numérica que el modelo pueda entender.

***Figura 7.*** *Variables Categóricas*

| | consignatario | depto | categoria | subcategoria | week |
|---|---|---|---|---|---|
| count | 535240 | 535240 | 535240 | 535240 | 535240 |
| unique | 49 | 241 | 6 | 12 | 209 |
| top | 465 | 6826 | 9 | 43 | 2020-31 |
| freq | 54617 | 17726 | 247835 | 149771 | 7346 |

Al aplicar la técnica de *One-Hot Encoding* a las variables categóricas mediante una función en Python, se observó un incremento significativo en el número de columnas del conjunto de datos. De las 17 columnas originales, se pasó a un total de 529 columnas, excluyendo la variable objetivo, que en este caso corresponde al precio. Este aumento en la dimensionalidad del dataset es consecuencia de la transformación de la variable *consignatario*, que dio lugar a 49 nuevas columnas; la columna *depto*, que generó 241 columnas; *categoria*, que resultó en 6 columnas adicionales; *subcategoria*, que derivó en 12 columnas; y finalmente, *week*, que se expandió en 209 columnas. Este fenómeno resalta el impacto que puede tener la transformación de variables categóricas en la estructura de un conjunto de datos, dado que cada categoría produce nuevas columnas, lo cual incrementa considerablemente la dimensionalidad. Tal expansión debe ser cuidadosamente gestionada en la fase de entrenamiento de modelos, ya que puede inducir a problemas de sobreajuste o a un uso ineficiente de los recursos computacionales disponibles.

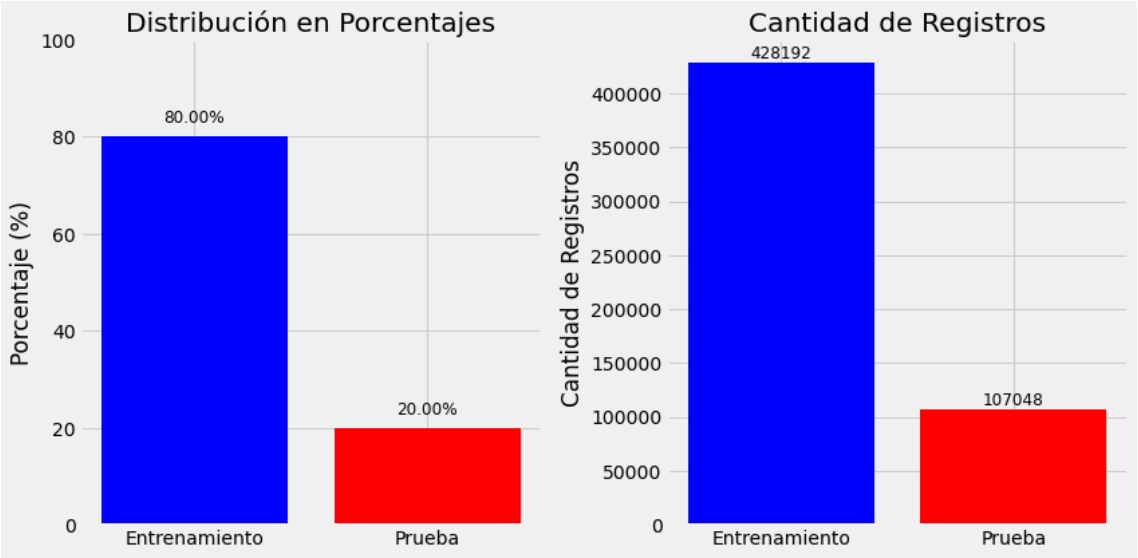
***Figura 8.*** *Dataset tras One-Hot Encoding*

```
RangeIndex: 535240 entries, 0 to 535239
Columns: 529 entries, cabezas to week_2022-52
dtypes: Sparse[float64, 0](529)
memory usage: 104.1 MB
```

Una vez completado el preprocesamiento, el *dataset* estuvo listo para ser utilizado en el modelo de regresión XGBRegressor. En este punto, se procedió a dividir el conjunto de datos en dos subconjuntos: uno para el entrenamiento del modelo y otro para prueba. Para la división, se optó por una estrategia comúnmente utilizada, que asigna el 80% de los datos al conjunto de entrenamiento y el 20% restante al conjunto de prueba. Este tipo de división permite que el modelo sea entrenado con la mayor cantidad posible de datos, a la vez que se reserva una porción de los datos para evaluar su rendimiento en datos no vistos, lo que ayuda a evitar el sobreajuste y asegura que el modelo generalice bien.

El conjunto de entrenamiento resultante constó de 428.192 registros, mientras que el conjunto de prueba quedó con 107.048 registros. Esta distribución es común en la mayoría de los proyectos de machine *learning*, ya que proporciona suficientes datos tanto para entrenar el modelo de forma efectiva como para evaluar su capacidad de generalización en un conjunto de datos independiente. Con esta división, el modelo podría ser entrenado de manera robusta y, posteriormente, evaluado utilizando el conjunto de prueba, lo que ofrece una buena estimación de su desempeño en situaciones del mundo real.

*Figura 9. Dataset repartido en train y test*

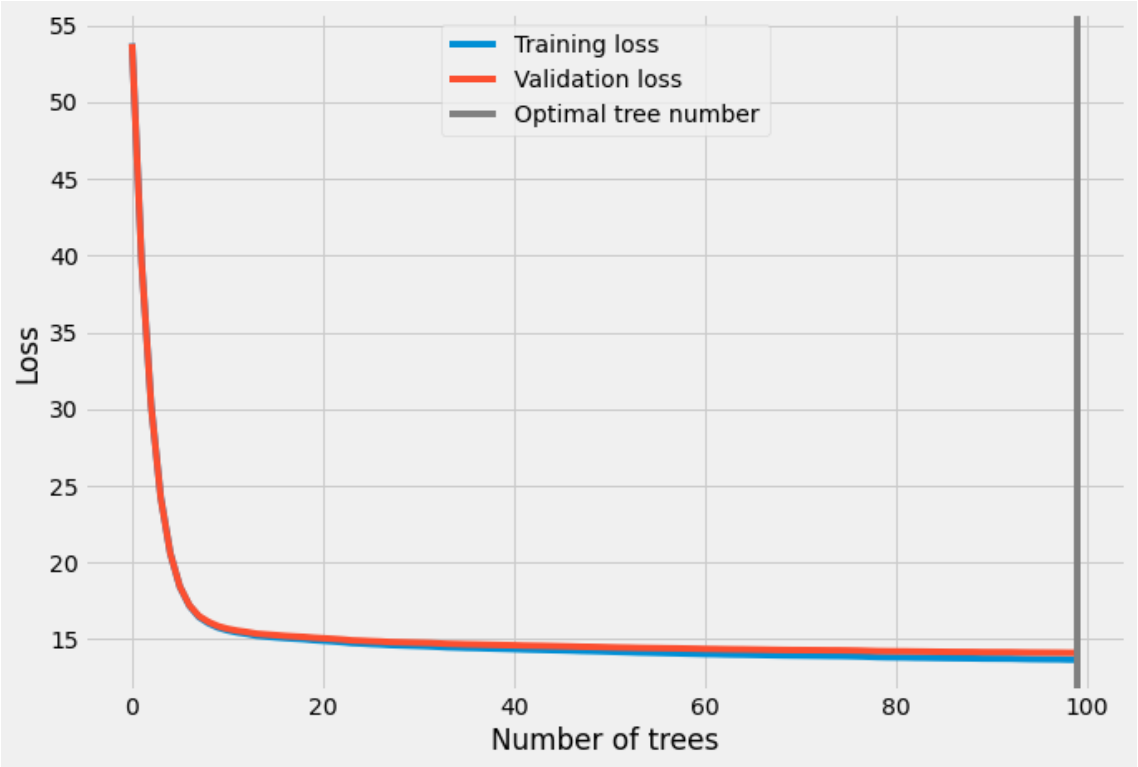



Para lograr el mejor resultado posible del modelo fue necesario hacer más de un procesamiento. Inicialmente, se puso a prueba el modelo utilizando los hiperparámetros con los valores predeterminados, y posteriormente se trabajó en afinar los hiperparámetros con el fin de mejorar la eficiencia de la predicción.

Aunque los resultados fueron bastante buenos, se buscó identificar si existía algún margen de mejora ajustando los parámetros del modelo. Para lograr esto, se empleó la técnica de *Grid Search*, en la cual se probaron todas las combinaciones posibles de los parámetros dentro de un rango previamente definido.

El modelo XGBRegressor tiene una propiedad que almacena el número de la mejor iteración (o el número de árboles) en que el modelo alcanzó su mejor desempeño durante el entrenamiento, gracias a esto es posible definir el rango del hiperparámetro de numero de árboles. En la Figura 10 se presenta un gráfico que visualiza el rendimiento del modelo XGBRegressor a medida que se añaden árboles al modelo durante el proceso de entrenamiento. Específicamente, el gráfico muestra la evolución del RMSE (*Root Mean Squared Error*, o Raíz del Error Cuadrático Medio) tanto en el conjunto de entrenamiento como en el conjunto de validación, a medida que se entrenan más árboles (o iteraciones). Finalmente, es interesante ver como a partir de los 50 árboles la mejora del modelo se suaviza.

***Figura 10.*** *Rendimiento del modelo XGBRegressor*




Aquí es donde se puso realmente a prueba la capacidad de procesamiento con la que contamos, ya que la búsqueda de los hiperparámetros que optimizan la eficiencia del modelo es un proceso extremadamente demandante para cualquier equipo de cómputo. Este proceso es crucial, ya que implica la determinación de los valores más adecuados para una serie de parámetros, de modo que el modelo pueda generalizar mejor y obtener los mejores resultados posibles. La tarea en cuestión consistió en explorar de manera sistemática todas las combinaciones posibles de los valores dentro de los rangos predefinidos para cada uno de los hiperparámetros seleccionados.

En otras palabras, la búsqueda se realizó mediante un producto cartesiano entre los valores de cada parámetro dentro de su respectivo rango, lo que implicó evaluar todas las combinaciones posibles entre los n parámetros seleccionados. Este enfoque provocó que la cantidad de cálculos necesarios se multiplicara exponencialmente, alcanzando más de 500 cálculos para llegar al objetivo. Como consecuencia, el proceso no solo se volvió computacionalmente intensivo, sino también progresivamente más complejo a medida que aumentaba el número de hiperparámetros a ajustar.

Este tipo de búsqueda, conocido como *Grid Search*, demanda una cantidad considerable de recursos de procesamiento, lo que puede generar un estrés significativo en la capacidad de cómputo, especialmente si se emplean modelos complejos o grandes volúmenes de datos. En el Apéndice D, se puede observar de manera ilustrativa un fragmento del trabajo que este proceso implicó para el computador, visualizando cómo se generaron y evaluaron distintas combinaciones posibles de los hiperparámetros seleccionados. Esto proporciona una idea más clara de la magnitud del esfuerzo computacional que implicó llevar a cabo la tarea de optimización.

Los parámetros ajustados en el proceso de optimización incluyen los siguientes:

- Tasa de aprendizaje (*learning_rate*): Este hiperparámetro controla el tamaño de los pasos dados por el modelo durante el proceso de aprendizaje. Un valor bajo favorece un aprendizaje más gradual, lo que puede mejorar la precisión del modelo al reducir el riesgo de sobreajuste. El valor por defecto de este parámetro es 0.3, y en este caso, el rango de búsqueda considerado fue [0.01, 0.1].
- Profundidad máxima del árbol (*max_depth*): Este parámetro establece la profundidad máxima de los árboles de decisión en el modelo. Un valor más alto puede generar árboles más complejos, lo que incrementa el riesgo de sobreajuste, mientras que valores más bajos podrían llevar a un modelo insuficientemente complejo. El valor por defecto es 6, y el rango de búsqueda evaluado fue [3, 5, 7, 10].
- peso mínimo de los hijos (*min_child_weight*): Este parámetro determina el peso mínimo que deben tener las muestras en los nodos hijos para que un nodo se divida. Un valor elevado puede hacer que el modelo sea más conservador, evitando divisiones en nodos con pocos datos. El valor por defecto de este hiperparámetro es 1, y el rango de búsqueda abarcó los valores [1, 3, 5].
- Fracción de muestras utilizadas en cada árbol (*subsample*): Controla la fracción de las muestras que se utilizan para entrenar cada árbol. Un valor más bajo puede reducir el sobreajuste, ya que introduce aleatoriedad en el entrenamiento de los árboles. El valor por defecto es 1.0, y el rango de búsqueda fue [0.5, 0.7].
- Fracción de características utilizadas (*colsample_bytree*): Este parámetro define la fracción de características que se usarán para construir cada árbol de decisión. Un valor menor favorece la regularización del modelo al evitar que dependa demasiado de un conjunto limitado de características. El valor por defecto es 1.0, y en la búsqueda se evaluaron los valores [0.5, 0.7].
- Número de estimadores (*n_estimators*): Este hiperparámetro especifica el número de árboles que se entrenarán en el modelo. Un mayor número de estimadores puede mejorar el rendimiento del modelo, pero también aumenta la posibilidad de sobreajuste. El valor por defecto es 100, y en este caso, el rango de búsqueda comprendió entre 50, el valor de la mejor iteración obtenida previamente en el primer procesamiento (Figura 10), y ese valor incrementado en 50 unidades.

Una vez realizada la búsqueda de hiperparámetros mediante GridSearchCV, los valores finales seleccionados para los parámetros fueron los siguientes:

- colsample_bytree = 0.7
- learning_rate = 0.1
- max_depth = 10
- min_child_weight = 1
- n_estimators = 149
- subsample = 0.7

Estos valores fueron los utilizados para entrenar por segunda vez el modelo XGBoost, con el objetivo de obtener el mejor rendimiento posible.

## 3.2. Redes Neuronales

La segunda metodología seleccionada fueron las redes neuronales, una herramienta poderosa para modelar dependencias entre las variables de entrada y las variables de salida de una manera flexible y no lineal. A diferencia de los modelos estadísticos tradicionales, como la regresión lineal, que suponen relaciones lineales entre las variables, las redes neuronales pueden capturar interacciones complejas y no lineales sin necesidad de especificar explícitamente una forma funcional para la relación entre las variables. En el contexto de la regresión, las redes neuronales tienen la capacidad de modelar relaciones no lineales entre las variables predictoras y la variable objetivo.

Las redes neuronales son un conjunto de modelos computacionales inspirados en la estructura y el funcionamiento del cerebro humano. Su principal fortaleza radica en su capacidad para modelar relaciones no lineales complejas entre las variables de entrada y salida. Estas redes consisten en unidades llamadas "neuronas", organizadas en capas de manera jerárquica. Cada capa recibe señales de la capa anterior y pasa su salida a la siguiente capa. La red más básica tiene tres tipos de capas: la capa de entrada, las capas ocultas, y la capa de salida. En el caso de un problema de regresión, la capa de salida usualmente tendrá una sola neurona, cuya salida será el valor continuo que la red intenta predecir.

Las redes neuronales se entrenan ajustando los pesos de las conexiones entre las neuronas para minimizar el error en las predicciones. Este proceso de ajuste de pesos se realiza mediante el algoritmo de retropropagación, que usa el gradiente del error con respecto a los pesos para actualizar los mismos de manera iterativa. Específicamente, para cada ejemplo de entrenamiento, se calcula el error en la salida de la red, y luego se propaga este error hacia atrás a través de la red para ajustar los pesos en cada capa.

Supongamos que tenemos un conjunto de datos de entrenamiento $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, donde $x_i$ es un vector de características y $y_i$ es la variable objetivo que queremos predecir. La salida de una red neuronal para un conjunto de entradas $x$ es:

***Ecuación 7.*** *Red Neuronal*

$$\hat{y} = f_o(x) = f(W_2\, h(W_2\, x + b_1) + b_2)$$

donde:

- $x$ es el vector de entrada,
- $W_1$ es la matriz de pesos de la capa de entrada a la capa oculta,
- $b_1$ es el sesgo de la capa oculta,
- $h(\cdot)$ es una función de activación (comúnmente una función no lineal como la sigmoide o ReLU),
- $W_{12}$ es la matriz de pesos de la capa oculta a la capa de salida,
- $b_2$ es el sesgo de la capa de salida.

En este contexto, la red neuronal para un problema de regresión está tratando de aprender los parámetros $\theta = \{W_1, b_1, W_2, b_2\}$ de tal forma que la predicción $\hat{y}$ sea lo más cercana posible al valor real $y$. En general, la función de activación en la capa de salida es lineal, es decir, $f(x) = x$, ya que se busca una salida continua para el problema de regresión.

El entrenamiento de una red neuronal se basa en la minimización de una función de pérdida. En el caso de la regresión, una de las funciones de pérdida más comunes es el error cuadrático medio (MSE), que se define como:

***Ecuación 8.*** *Red Neuronal, minimización de la función de perdida MSE*

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f_\theta(x_i)\right)^2$$

El objetivo del entrenamiento es encontrar los parámetros $\theta$ que minimicen esta función de pérdida. Para ello, se utiliza un algoritmo de optimización, siendo uno de los más populares el descenso del gradiente. Este algoritmo ajusta los parámetros de la red en la dirección opuesta al gradiente de la función de pérdida con respecto a los parámetros:

***Ecuación 9.*** *Red Neoronal, descenso del gradiente*

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta L(\theta_t)$$

donde $\eta$ es la tasa de aprendizaje, un hiperparámetro que controla el tamaño del paso en cada iteración, y $\nabla_\theta L(\theta_t)$ es el gradiente de la función de pérdida con respecto a los parámetros en el tiempo $t$.

A medida que el algoritmo de descenso de gradiente avanza, los pesos de la red se ajustan para mejorar las predicciones, reduciendo el error cuadrático medio sobre el conjunto de entrenamiento.

La fase de propagación hacia adelante es el proceso en el que las entradas $x_i$ se pasan a través de la red, capa por capa, para generar la predicción $\hat{y}_i$. Este proceso se describe mediante la fórmula que mencionamos anteriormente. En esta fase, la red realiza una estimación de la variable dependiente $y_i$ para cada ejemplo de entrenamiento.

Por otro lado, la retropropagación es el algoritmo mediante el cual los errores de predicción son "propagados hacia atrás" desde la capa de salida hasta las capas anteriores, de modo que se pueda ajustar el gradiente de la función de pérdida con respecto a los pesos y sesgos de la red. En este proceso, se aplican reglas de la cadena para calcular cómo cada parámetro contribuye al error total y, con base en esto, actualizar los parámetros utilizando el algoritmo de descenso de gradiente.

Uno de los desafíos al entrenar redes neuronales es el sobreajuste, donde el modelo se ajusta demasiado a los datos de entrenamiento y pierde capacidad de generalización. Para mitigar este problema, se pueden aplicar técnicas de regularización. Una de las más comunes es la regularización L2 (también conocida como "decadencia de peso"), que agrega un término de penalización en la función de pérdida:

***Ecuación 10.*** *Red Neuronal, mitigación de sobreajuste*

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f_\theta(x_i)\right)^2 + \lambda \sum_j \theta_j^2$$

donde $\lambda$ es un hiperparámetro que controla la fuerza de la regularización, y $\theta_j$ representa los pesos de la red. Este término penaliza los valores grandes de los parámetros, evitando que el modelo se sobreajuste.

Otra técnica común de regularización es la regularización L1, que penaliza la suma de los valores absolutos de los pesos. Además, técnicas como *dropout* y *early stopping* también son útiles para prevenir el sobreajuste.

Las redes neuronales ofrecen varias ventajas en problemas de regresión que las convierte en herramientas poderosas cuando se trata de datos con interacciones complicadas o patrones que no pueden ser capturados por modelos lineales o más simples.

Sin embargo, las redes neuronales también presentan ciertos desafíos. Uno de los más relevantes es la necesidad de grandes cantidades de datos para entrenar modelos efectivos, especialmente en redes profundas. Además, las redes pueden ser sensibles a la elección de los hiperparámetros, como la arquitectura de la red, la tasa de aprendizaje, y el número de neuronas en las capas ocultas. También es importante tener en cuenta la complejidad computacional, ya que el entrenamiento de redes neuronales, especialmente en redes profundas, puede ser intensivo en términos de tiempo y recursos.

Para implementar el modelo de predicción basado en Redes Neuronales, fue necesario realizar un preprocesamiento adecuado de los datos. Como se explicó en el modelo anterior, gran parte del proceso de preprocesamiento se llevó a cabo de manera progresiva durante la construcción del conjunto de datos, lo que permitió optimizar el tiempo y facilitar la adaptación de los datos a las especificidades de cada modelo.

Al igual que en el modelo de XGBoost, las Redes Neuronales no pueden manejar directamente variables categóricas. Por ello, se recurrió nuevamente a la técnica de *One-Hot Encoding*, que convierte cada categoría en una nueva columna binaria. En este enfoque, cada columna representa una categoría única de la variable original, transformando las variables categóricas en un conjunto de columnas binarias (0s y 1s), donde cada fila indica la pertenencia de una observación a una categoría específica. Esta técnica se implementó en las variables *consignatario, depto, categoría* y *subcategoría*.

Sin embargo, en el caso de la variable *week*, se optó por una transformación diferente, con el fin de evitar un aumento excesivo en el número de columnas del conjunto de datos. A medida que crece el número de columnas, aumenta de manera significativa el poder de procesamiento necesario para cualquier modelo, y en particular para Redes Neuronales, que son especialmente sensibles a la dimensionalidad de los datos.

La variable *week* posee una naturaleza categórica, pero además tiene un componente temporal que puede ser útil para la predicción. El valor de *week* es una combinación de dos factores: el número de la semana y el año. Aunque esta variable no es numérica en el sentido tradicional (como una característica continua), sí presenta una relación temporal significativa. Por ello, se decidió tratarla de manera distinta. En lugar de convertirla directamente en una serie de columnas binarias, se separó en dos variables: una para el número de la semana (*week_num*) y otra para el año (*year*), con cuatro posibles valores (2019, 2020, 2021 y 2022).

Dado que las semanas siguen un ciclo anual (después de la semana 52, vuelve la semana 1), se decidió transformar las variables *week_num* y *year* en características cíclicas. Esta transformación permite incorporar la estructura cíclica de las semanas y los años en el modelo, lo que mejora la capacidad predictiva al reflejar de manera más precisa la relación temporal entre los valores.

Para lograrlo, se aplicaron funciones trigonométricas, específicamente seno y coseno, para representar las semanas y los años como características continuas. Estas funciones permiten modelar la circularidad del tiempo de manera eficaz, ya que los valores cercanos en el ciclo estarán representados por números próximos, sin importar su distancia en la escala numérica directa.

Las fórmulas para calcular las representaciones cíclicas de las variables *week_num* y *year* son las siguientes:

***Ecuación 11.*** *Red Neuronal, representación ciclica (seno y cosceno) del tiempo (nro de semana y año)*

$$week\_sin = \sin\left(\frac{2\pi \times week\_num}{N}\right) \qquad year\_sin = \sin\left(\frac{2\pi \times year}{N}\right)$$

$$\text{week\_cos} = \cos\left(\frac{2\pi \times week\_num}{N}\right) \qquad \text{year\_cos} = \cos\left(\frac{2\pi \times year}{N}\right)$$

Donde:

- N es el número total de periodos (52 para semanas, 2022 para años).
- 2π es el ciclo completo de un círculo en radianes (ya que las funciones seno y coseno son periódicas cada 2π radianes).

Este enfoque generó cuatro nuevas columnas (*week_sin, week_cos, year_sin* y *year_cos*), que representan las semanas y los años de manera cíclica. De este modo, el modelo es capaz de entender que las semanas y los años no son variables lineales, sino que siguen un ciclo anual, mejorando la captura de patrones temporales en los datos.

De manera similar al enfoque utilizado en el modelo anterior, se procedió a dividir el conjunto de datos en dos subconjuntos principales: uno destinado al entrenamiento del modelo y otro para la evaluación de su rendimiento en datos no vistos. En esta ocasión, se optó nuevamente por una división estándar, asignando el 80% de los datos al conjunto de entrenamiento y reservando el 20% restante para el conjunto de prueba. Esta división es comúnmente utilizada ya que permite entrenar el modelo con una proporción significativa de los datos disponibles, a la vez que se mantiene una muestra representativa para probar su capacidad de generalización. Este enfoque ayuda a prevenir el sobreajuste o *overfitting*, una condición en la que el modelo se ajusta demasiado a los datos de entrenamiento, perdiendo capacidad para generalizar a nuevos datos.

Al igual que en el modelo de XGBoost, el proceso de implementación del modelo de Redes Neuronales también requirió una serie de pasos adicionales de preprocesamiento y ajuste fino. Inicialmente, se evaluó el rendimiento del modelo utilizando los valores predeterminados de los

hiperparámetros que se ofrecen en la librería *sklearn*. Este primer procesamiento del modelo de redes neuronales mostró resultados prometedores.

A pesar de los resultados altamente satisfactorios obtenidos en el modelo inicial, se intentó mejorar el rendimiento del modelo mediante la optimización de hiperparámetros utilizando *GridSearchCV*. Este proceso tiene como objetivo encontrar los mejores valores para parámetros clave del modelo, como el número de capas ocultas, el número de neuronas por capa, la tasa de aprendizaje, y otros parámetros relevantes, que pueden influir directamente en la capacidad predictiva del modelo.

En el proceso de optimización, se ajustaron varios parámetros clave del modelo, que influyen directamente en su capacidad predictiva. A continuación, se describen los parámetros ajustados:

Tamaño de las capas ocultas (*hidden_layer_sizes*): Este parámetro determina la arquitectura de la red neuronal, especificando el número de capas ocultas y el número de neuronas en cada capa. Un número mayor de capas y neuronas puede aumentar la capacidad del modelo para aprender patrones complejos, pero también puede incrementar el riesgo de sobreajuste. El valor por defecto es (100), que representa una sola capa oculta con 100 neuronas, y en la búsqueda se evaluaron las configuraciones [(100,), (100, 50), (200, 100), (50, 50, 50)].

Número máximo de iteraciones (*max_iter*): Este parámetro define el número máximo de iteraciones que el optimizador realizará durante el entrenamiento del modelo. Un valor mayor permite que el modelo se ajuste de manera más exhaustiva, pero también puede incrementar el tiempo de cómputo. El valor por defecto es 200, y los valores evaluados en este caso fueron [500, 1000, 1500].

Tasa de regularización (*alpha*): Este parámetro controla la penalización sobre los pesos del modelo para evitar el sobreajuste. Un valor más alto de alpha aumenta la regularización, restringiendo la magnitud de los pesos. El valor por defecto es 0.0001, y se probaron los valores [0.0001, 0.001, 0.01].

Tasa de aprendizaje (*learning_rate*): Este parámetro regula cómo se ajusta la tasa de aprendizaje durante el entrenamiento. Existen diferentes estrategias de ajuste: *constant* mantiene la tasa fija, *invscaling* la reduce conforme aumenta el número de iteraciones, y adaptive ajusta la tasa dependiendo del rendimiento. El valor por defecto es *constant*, y se probaron las opciones [*constant*, *invscaling*, *adaptive*].

Algoritmo de optimización (*solver*): Este parámetro determina el algoritmo utilizado para minimizar la función de error. El valor por defecto es *adam*, un optimizador eficiente que combina técnicas de gradiente estocástico y momentos adaptativos, y se evaluaron las opciones [*adam*', *sgd*].

Función de activación (*activation*): Este parámetro define la función de activación utilizada en las neuronas de la red neuronal. La función de activación introduce no linealidad, permitiendo que el modelo capture patrones complejos. El valor por defecto es *relu*, y se probaron las funciones [*relu*, *tanh*, *logistic*].

Sin embargo, se presentaron desafíos significativos en términos de tiempo de cómputo. A medida que la búsqueda de hiperparámetros se extendía, el proceso se volvió excesivamente

largo, alcanzando más de 72 horas sin llegar a una solución óptima. Este problema es común cuando se realiza una búsqueda *GridSearchCV*, especialmente cuando se exploran un número elevado de combinaciones de parámetros o cuando el modelo en sí tiene una alta complejidad computacional, como es el caso de las Redes Neuronales.

En consecuencia, debido al alto costo computacional y el tiempo de espera, se tomó la decisión de detener el proceso de optimización de hiperparámetros y trabajar con el modelo previamente entrenado.

# 4. Resultados

Tras todo el esfuerzo puesto en preparar los datos para que puedan ser procesados por los modelos XGBoost y Redes Neuronales amerita hacer un análisis profundo a los resultados obtenidos por ambos modelos. En ambos casos, el primer objetivo buscado y que puede determinar el éxito o fracaso de todo el trabajo realizado es la predicción del precio del ganado vacuno. Sin embargo, esto no significa que sea lo único que se pueda obtener como resultado de un modelo. A continuación, se presentarán diferentes métricas que se pudieron obtener a partir de los resultados tanto del modelo XGBoost como del modelo de Redes Neuronales.

## 4.1. XGBRegressor

Inicialmente, se puso a prueba el modelo utilizando los hiperparametros con los valores predeterminados. Este primer procesamiento del modelo mostró resultados satisfactorios, como se puede observar Figura 11, donde los valores de $R^2$ son bastante altos (cerca de 0.97 en ambos conjuntos), lo que indica que el modelo tiene un buen poder predictivo y es capaz de explicar casi el 97% de la varianza en los datos. Además, la diferencia entre las métricas de rendimiento de entrenamiento y prueba es pequeña. El hecho de que el modelo tenga un $R^2$ muy similar en ambos conjuntos y las diferencias en MSE y RMSE sean mínimas sugiere que no hay un problema serio de *overfitting*.

***Figura 11.*** *Performance Primer XGBRegressor*

```
### Train performance ###
MAX ERROR train  203.73194885253906
MSE train  186.9982137299523
RMSE train  13.674729018519976
R^2 train:  0.9665423687267372

### Test performance ###
MAX ERROR test  171.78369140625
MSE test  199.29580054502642
RMSE test  14.117216458814621
R^2 test:  0.9643533286045768
```

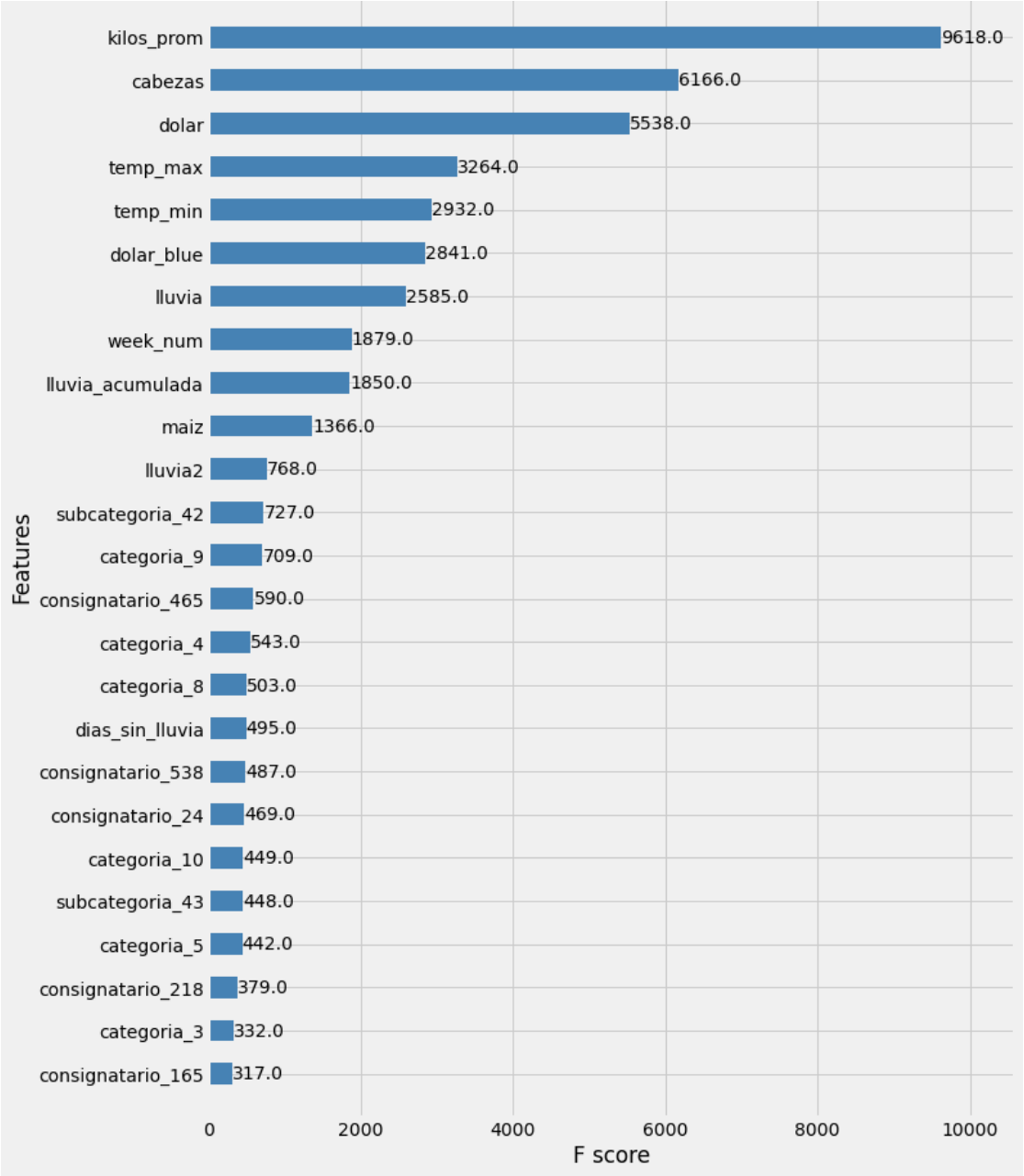
No obstante, se debe tener en cuenta el Error Máximo (203.73 en entrenamiento y 171.78 en prueba) el cual es considerablemente alto, aunque este no refleja el rendimiento general, sino el peor caso.

Los resultados obtenidos al ejecutar nuevamente el modelo XGBRegressor con los mejores hiperparámetros seleccionados a través de un GridSearchCV muestran una mejora significativa tanto en los datos de entrenamiento como en los de prueba, en comparación con el primer intento. Al optimizar los hiperparámetros, se logró no solo reducir el Error Máximo, sino también mejorar ligeramente la capacidad de generalización del modelo. Esto resalta la importancia de realizar una búsqueda cuidadosa de hiperparámetros y la efectividad del proceso de GridSearchCV para encontrar configuraciones que optimicen el rendimiento del modelo.

A través del gráfico de Importancia de Características en el modelo XGBRegressor fue posible interpretar y comprender qué características tuvieron mayor influencia en las predicciones del modelo. Como se puede observar en la Figura 12 para este modelo la variable que más influyo a la hora de predecir el precio del ganado vacuno fue *kilom_prom*. Esta variable se puede colocar a la cabeza de un conjunto de diez características que, sin lugar a duda, son clave en las predicciones del modelo. Esto sugiere que el modelo está basando una gran parte de sus predicciones en estas variables específicas.

Otra interpretación que permite determinar este gráfico es cuáles son aquellas características que tienen baja importancia y que probablemente no aportan mucho al modelo. Esto es especialmente útil para reducir la dimensionalidad y mejorar la eficiencia del modelo, ya que menos características pueden llevar a tiempos de entrenamiento más rápidos y una menor complejidad computacional. En este caso, se observa que entre las 25 características más importante no aparece ningún departamento, lo cual nos da la pauta que podríamos eliminarla para futuros entrenamientos.

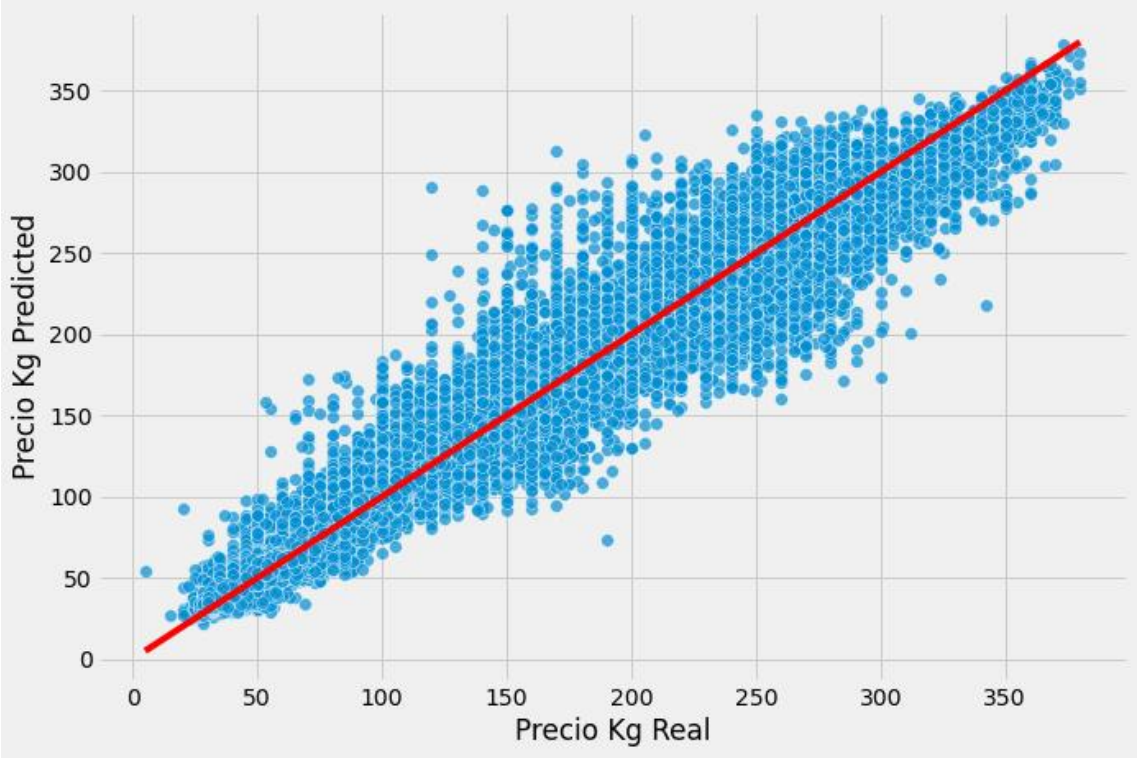
***Figura 12.*** *Importancia de Características en el modelo XGBRegressor*




Otro de los resultados importantes que obtuvimos al ejecutar el modelo fue el gráfico de comparación entre los precios reales y las predicciones generadas por el modelo. Este gráfico resulta ser una herramienta muy útil para evaluar visualmente cómo de bien está funcionando el modelo de regresión, en este caso, el modelo XGBRegressor. El gráfico utilizado es un gráfico de dispersión, también conocido como *scatter plot*, que aparece en la figura 13, y permite observar de manera clara cómo se ajustan las predicciones del modelo a los valores reales de los datos. Al analizar este gráfico, se puede observar que la dispersión de los puntos alrededor de la línea roja, que representa la relación perfecta entre los valores predichos y los valores reales, nos da una indicación clara de que el modelo está haciendo un trabajo bastante bueno en la predicción para la mayoría de las instancias. Esto es, la mayoría de los puntos caen

relativamente cerca de esta línea, lo que significa que el modelo ha acertado con las predicciones de forma bastante precisa.

***Figura 13.*** *Precios Reales VS Predicciones del modelo XGBRegressor*

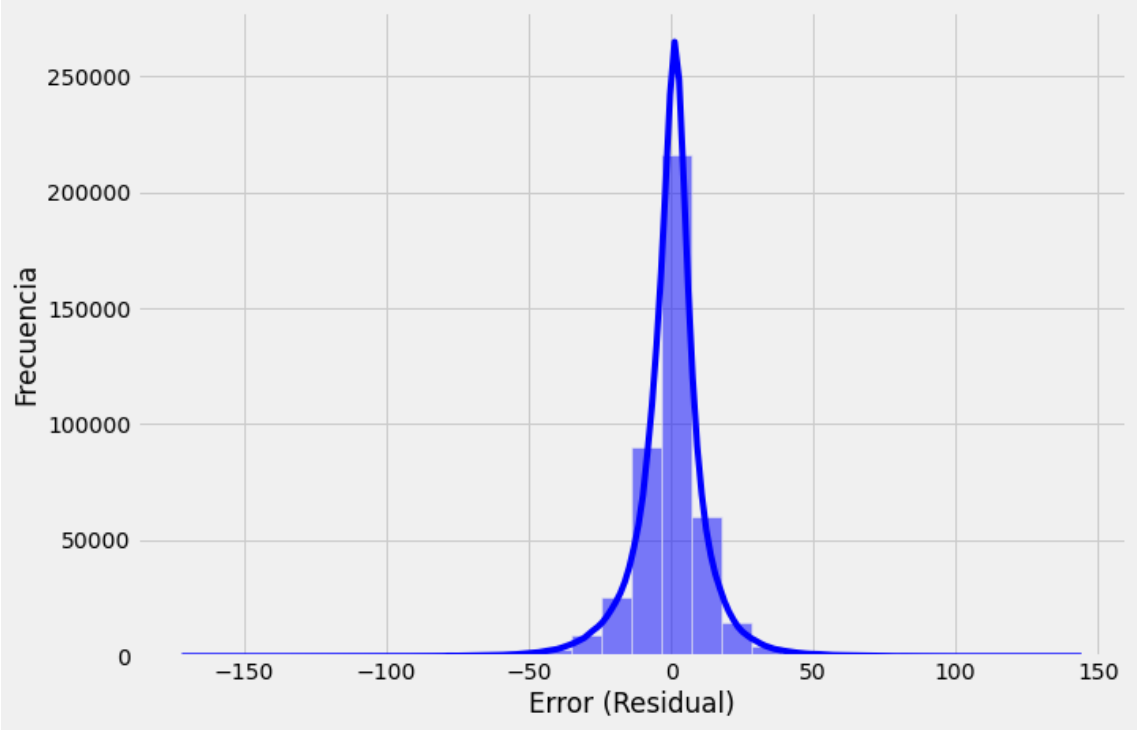



Sin embargo, no todos los puntos siguen este patrón. Algunos casos aislados se alejan significativamente de la línea roja, lo que sugiere que para esos casos el modelo no fue tan preciso. Estos puntos que se desvían en gran medida de la línea de referencia podrían ser valores atípicos, también conocidos como *outliers*. Los *outliers* son observaciones que se desvían de manera considerable de la tendencia general de los datos y suelen representar casos excepcionales o inusuales. Estos valores atípicos son complicados para los modelos de *machine learning*, ya que suelen generar grandes errores de predicción. Esto se debe a que el modelo intenta ajustar sus predicciones para todo el conjunto de datos, pero no siempre puede predecir correctamente para esos casos inusuales que no siguen el patrón general. Por lo tanto, es posible que algunos de estos puntos aislados sean difícilmente predecibles para el modelo, dado su comportamiento atípico.

Continuando con las diferencias que existen entre el precio real de la carne vacuna y el precio obtenido en la predicción por el modelo se generó el Histograma de Errores, el cual es clave para analizar la calidad de las predicciones del modelo. El Histograma de Errores que se muestra en la Figura 14 representa la distribución de las diferencias entre las predicciones del modelo y los valores reales (observados). Este gráfico, en su simplicidad, proporciona mucha información respecto del modelo XGBRegressor. En primer lugar, el dibujo de curva en forma de campana es debido a que los errores siguen una distribución normal y esto nos da el primer indicio que el modelo está haciendo un buen trabajo.

Otro punto a favor del histograma es que los valores están bastante centrados en torno a cero, sugiriendo que el modelo, en promedio, no tiene sesgo significativo. Además, al tener la campana muy estrecha sugiere que el modelo tiene una baja varianza, y esto puede ser una señal positiva de que las predicciones son más estables.

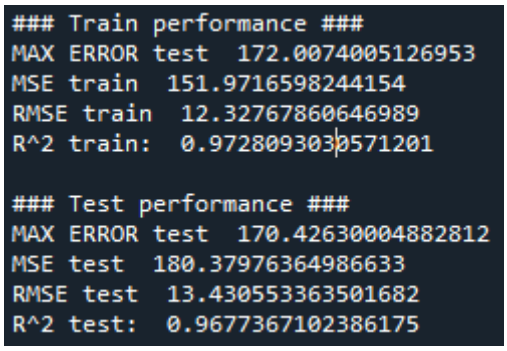
***Figura 14.*** *Histograma de Errores XGBRegressor*




Solo hay una observación negativa del histograma que merece ser destacada: la existencia de más errores positivos que negativos. Aunque la diferencia no es grande, esto podría ser un indicio de que hay alguna característica del modelo o de los datos que está provocando un pequeño sesgo en las predicciones.

Pero como se planteó en el inicio de este trabajo, el objetivo es buscar predecir el precio del valor de la carne vacuna y ningún sentido tendrían estos gráficos analizados si no se lograra un resultado aceptable. Y esto se logró tras un arduo trabajo de preprocesamiento y de un extenso tiempo de cómputo en el entrenamiento del modelo. Tras ajustar los hiperparámetros y reprocesar el modelo las métricas obtenidas, las cuales se pueden ver tanto para entrenamiento como para prueba en la figura 15, y que iremos desglosando e interpretando son Error Máximo, Error Cuadrático Medio (MSE), Raíz del Error Cuadrático Medio (RMSE), y Coeficiente de Determinación ($R^2$).

***Figura 15.*** *Performance XGBRegressor Final*




Además de la mejora en el Error Máximo, se observó una ligera pero positiva mejora en el $R^2$ respecto a la ejecución inicial. En el segundo intento, el $R^2$ se mantuvo en torno al 97%, lo que implica que el modelo ahora es capaz de explicar un 97% de la variabilidad en los datos, tanto en el conjunto de entrenamiento como en el de prueba. Este ligero incremento en el $R^2$ es un buen indicativo de que el modelo ha logrado generalizar aún mejor, capturando relaciones más sutiles entre las características y las predicciones, sin sobreajustarse a los datos de entrenamiento.

El error Máximo, es relativamente alto (170.426 en el conjunto de prueba) puede ser preocupante, ya que podría significar que el modelo está cometiendo errores significativos en algunas instancias, posiblemente debido a *outliers* o características no representadas adecuadamente en el modelo. Esto concuerda con el gráfico de comparación entre los precios reales y las predicciones generadas por el modelo, donde ya se observaban algunos valores atípicos.

El MSE en el conjunto de prueba es 180.379, lo que es más alto que el MSE en el conjunto de entrenamiento (151.971). Esto indica que el modelo está cometiendo más errores en el conjunto de prueba que en el de entrenamiento, lo cual es común cuando el modelo generaliza menos bien a datos no vistos o cuando el modelo está ligeramente sobreajustado (*overfitted*). Sin embargo, el aumento en el MSE no es drástico, lo que sugiere que el modelo todavía está funcionando bien, aunque podría necesitar una pequeña mejora.

El RMSE de 13.430 es ligeramente mayor en el conjunto de prueba que en el conjunto de entrenamiento (12.327). Este aumento en el RMSE es esperado, ya que generalmente se observa que los modelos tienen un rendimiento ligeramente peor en el conjunto de prueba, especialmente si hay alguna forma de sobreajuste. A pesar de este aumento, el RMSE sigue siendo relativamente bajo, lo que sugiere que el modelo, en general, está haciendo predicciones razonablemente precisas.

El $R^2$ en el conjunto de prueba es 0.9677, lo que sigue siendo un valor muy alto y sugiere que el modelo aún explica aproximadamente el 96.77% de la variabilidad en los datos de prueba. Aunque este valor es ligeramente inferior al $R^2$ en el conjunto de entrenamiento (0.97280), sigue siendo un excelente resultado, lo que indica que el modelo tiene una muy buena capacidad de generalización. Un $R^2$ cercano a 1 en el conjunto de prueba sugiere que el modelo es capaz de hacer predicciones bastante precisas en datos no vistos.

La ligera diferencia en el rendimiento entre el conjunto de entrenamiento y el de prueba (específicamente en el MSE y RMSE) puede ser un indicio de que el modelo ha sobreajustado a los datos de entrenamiento, ya que este sobreajuste puede generar que el modelo pierde capacidad para generalizar a nuevos datos o datos atípicos (*outliers*).

A pesar de estas pequeñas diferencias entre el conjunto de entrenamiento y prueba, con valores de $R^2$ superiores al 96% en ambos casos y errores relativamente bajos en términos de RMSE, nos indica que el modelo tiene un rendimiento general excelente.

## 4.2. Redes Neuronales

Al igual que en el modelo de XGBoost, el proceso de implementación del modelo de Redes Neuronales también requirió una serie de pasos adicionales de preprocesamiento y ajuste fino. Inicialmente, se evaluó el rendimiento del modelo utilizando los valores predeterminados de los hiperparámetros que se ofrecen en la librería. Este primer procesamiento del modelo de redes neuronales mostró resultados prometedores, con un excelente desempeño tanto en el conjunto de entrenamiento como en el conjunto de prueba. Como se muestra en la Figura 16, los valores de $R^2$ para ambos conjuntos superan el 96%, lo que indica que el modelo es capaz de explicar más del 96% de la variabilidad en la variable dependiente. Este alto valor de $R^2$ sugiere que el modelo tiene un buen poder predictivo y es capaz de capturar adecuadamente las relaciones subyacentes entre las variables independientes y la variable objetivo. Este resultado es indicativo de un modelo bien ajustado, con una capacidad de predicción robusta.

Asimismo, las métricas de RMSE (Raíz del Error Cuadrático Medio) y MSE (Error Cuadrático Medio) fueron relativamente bajas, lo que significa que las predicciones del modelo están bastante cerca de los valores reales. El RMSE, al expresarse en las mismas unidades que la variable objetivo, proporciona una medida directa de la magnitud del error, mientras que el MSE evalúa el promedio de los errores al cuadrar las diferencias entre las predicciones y los valores reales. Ambos valores son indicadores positivos del rendimiento del modelo.

No obstante, a pesar de estos resultados positivos, es importante resaltar que el modelo de redes neuronales presenta un pequeño sobreajuste. Esto se observa en el leve aumento de los errores en el conjunto de prueba en comparación con el conjunto de entrenamiento, lo que sugiere que, aunque el modelo se ajustó bien a los datos de entrenamiento, su capacidad para generalizar a datos nuevos podría ser ligeramente mejorada. Este tipo de comportamiento es común cuando un modelo de aprendizaje automático tiene un ajuste demasiado preciso a las muestras de entrenamiento y, por lo tanto, no es tan efectivo en situaciones con nuevos datos.

A pesar de este pequeño sobreajuste, los resultados generales son altamente satisfactorios. El modelo de redes neuronales ha mostrado ser robusto y prometedor, con una gran capacidad para predecir los valores de la variable objetivo en datos no vistos.

*Figura 16. Performance Primer Red Neuronal*

```
### Train performance ###
MAX ERROR train:  191.1174873279999
MSE train:  130.1943840732808
RMSE train:  11.410275372368574
R^2 train:  0.9767056828550033

### Test performance ###
MAX ERROR test:  181.77734904517825
MSE test:  182.59316692013047
RMSE test:  13.512703908549557
R^2 test:  0.9673408139938151
```

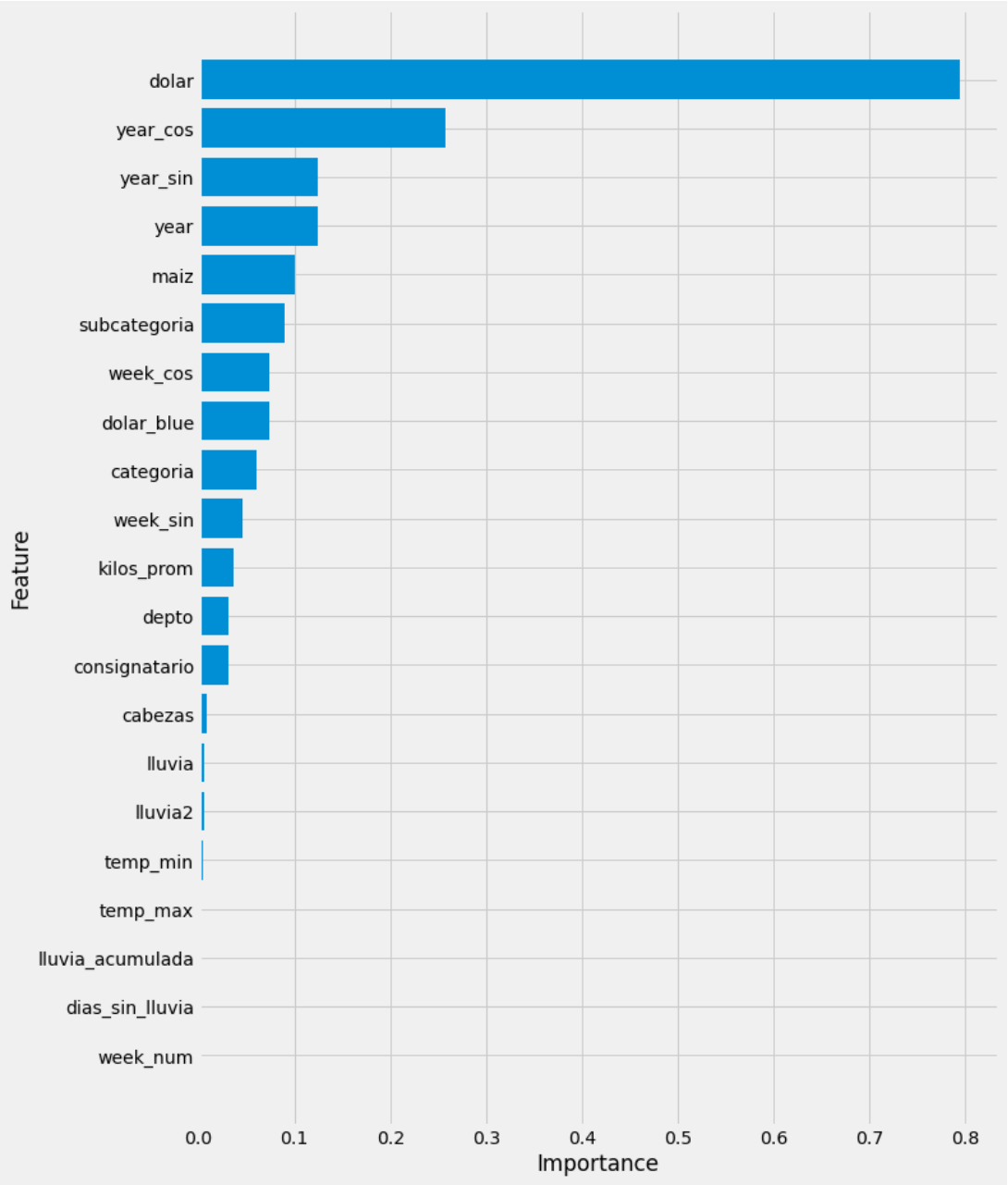
Y al igual que en el modelo XGBoost, se intentó mejorar el rendimiento del modelo de Redes Neuronales mediante la optimización de hiperparámetros utilizando *GridSearchCV*. Sin embargo, se presentaron desafíos significativos en términos de tiempo de cómputo. En consecuencia, se tomó la decisión de detener el proceso de optimización de hiperparámetros y trabajar con el modelo previamente entrenado, donde los resultados obtenidos hasta ese momento con el modelo original eran prometedores, con un $R^2$ superior al 96% en ambos conjuntos de datos, demostrando un buen poder predictivo y un rendimiento robusto.

Con el modelo de Redes Neuronales, también se tiene la posibilidad de observar diversos resultados que nos indican la performance del modelo y las características del *dataset*. Pero, las redes neuronales tienen un comportamiento complejo debido a la no linealidad y la interdependencia entre las características, lo que hace que la interpretación de las características sea más difícil que en modelos de árboles de decisión, lo cual hace imposible obtener una medida global de la importancia de las características tal cual el modelo anterior.

Sin embargo, existen maneras de obtener una visualización similar para redes neuronales utilizando otras técnicas, como *Permutation Importance*. El método de *Permutation Importance* permite ver cómo cambia el rendimiento del modelo cuando se permutan aleatoriamente las características. Cuanto más caiga el rendimiento, más importante es esa característica. En la Figura 17, podemos ver plasmada esta técnica, la cual arroja que la característica que más influye en la predicción del precio de la carne vacuna es la cotización del dólar. Esta característica está a la cabeza de un conjunto de diez características que sin lugar a duda el modelo está basando muchas de sus predicciones en ellas.

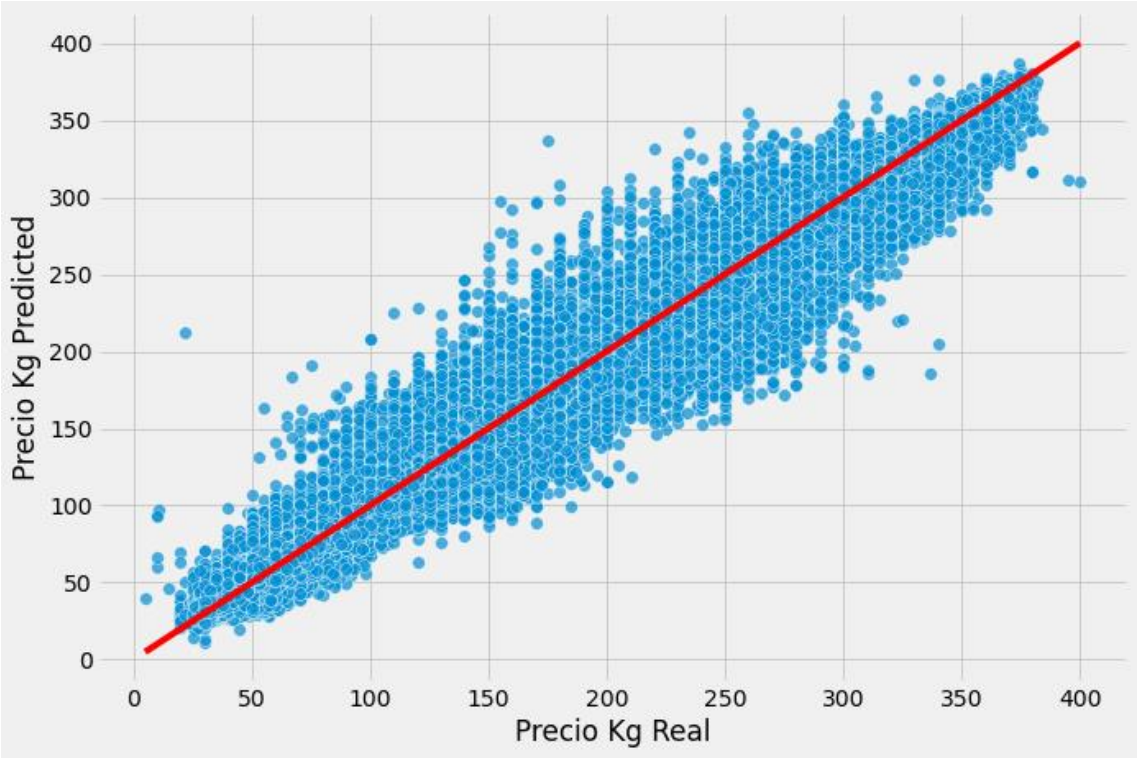
Otra interpretación que se puede hacer de este gráfico es cuáles son aquellas características que tienen baja importancia y que probablemente no aportan mucho al modelo. Observamos que temp_max, lluvia_acumulada, días_sin_lluvia, y week_num no afecta a la caída del rendimiento. Es muy interesante ver que week_num no tenga gran impacto en la predicción, pero si lo tienen las variables week_cos y week_sin que son producto de week_num. Esto confirma la eficacia usar funciones trigonométricas para modelar la circularidad del tiempo.

*Figura 17. Permutation Importance Redes Neuronales*




Al igual que en el modelo anterior, el resultado importante que si se logró obtener al ejecutar el modelo de redes neuronales fue el gráfico de comparación entre los precios reales y las predicciones generadas por el modelo. El gráfico de dispersión que aparece en la figura 18, permite observar de manera clara cómo se ajustan las predicciones del modelo a los valores reales de los datos. Y al analizar este gráfico, podemos ver que la dispersión de los puntos alrededor de la línea roja, que representa la relación perfecta entre los valores predichos y los valores reales, nos da una indicación clara de que el modelo de redes neuronales también está haciendo un trabajo bastante bueno en la predicción para la mayoría de las instancias. Esto es, la mayoría de los puntos caen relativamente cerca de esta línea, lo que significa que el modelo también ha acertado con las predicciones de forma bastante precisa.

***Figura 18.*** *Precios Reales VS Predicciones del modelo Redes Neuronales*



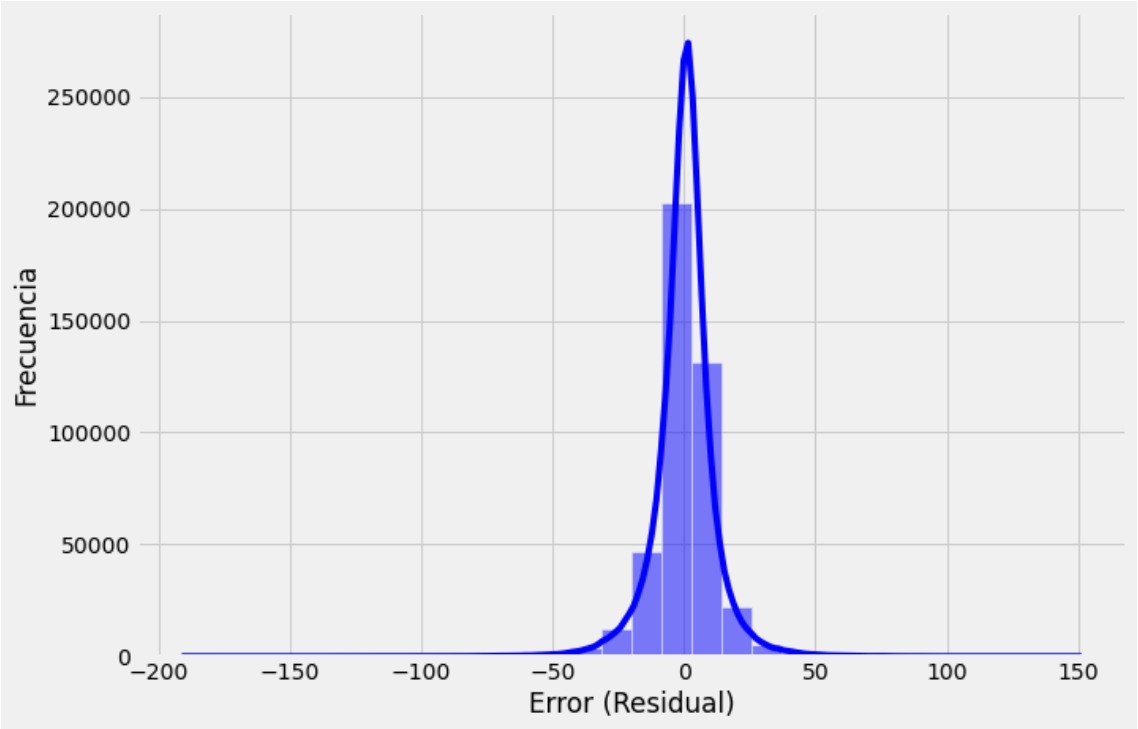

Y, de la misma forma que en el modelo de XGBoost, no todos los puntos siguen este patrón. Algunos casos aislados se alejan significativamente de la línea roja, lo que sugiere que para esos casos el modelo no fue tan preciso. De la misma forma que los observamos en el modelo de XGBoost, estos puntos que se desvían en gran medida de la línea de referencia son observaciones que se desvían de manera considerable de la tendencia general de los datos y suelen representar casos excepcionales o inusuales (*outliers*).

También se obtuvo el histograma de errores, el cual es clave para analizar la calidad de las predicciones del modelo. Lo que se observa en la Figura 19 es la distribución de las diferencias entre las predicciones del modelo y los valores reales (observados). Este gráfico que parece tan simple, está diciendo mucho respecto del modelo. En primer lugar, el dibujo de curva en forma de campana es debido a que los errores siguen una distribución normal y esto indica que el modelo está haciendo un buen trabajo.

Otro punto a favor del histograma es que los valores están bastante centrados en torno a cero, sugiriendo que el modelo de Redes Neuronales, en promedio, no tiene sesgo significativo. Además, al tener la campana muy estrecha sugiere que el modelo tiene una baja varianza, y esto puede ser una señal positiva de que las predicciones son más estables.

La única observación negativa del histograma es la existencia de más errores negativos que positivos, aunque no es grande la diferencia, esto puede ser un indicio de que hay alguna característica del modelo o de los datos que está provocando un pequeño sesgo en las predicciones.

*Figura 19. Histograma de Errores Redes Neuronales*




Como se mencionó previamente en este trabajo, uno de los objetivos fundamentales es predecir el valor del precio de la carne vacuna. Debido a limitaciones de recursos de cómputo, no fue posible completar la búsqueda de hiperparámetros de manera exhaustiva. En consecuencia, se implementó un modelo de redes neuronales utilizando los hiperparámetros con los valores por defecto. A pesar de esta restricción, como se verá a continuación, el modelo ha mostrado resultados bastante satisfactorios en términos de precisión y capacidad de generalización.

El rendimiento del modelo en el conjunto de entrenamiento presenta un Error Máximo de 191.12, lo que es relativamente alto, pero no alarmante. Este valor podría sugerir que en algunas instancias específicas el modelo comete errores significativos, lo que puede ser indicativo de la presencia de datos atípicos o de una complejidad del modelo que aún no se ha ajustado completamente. Sin embargo, no es un valor excesivamente alto, y considerando que los modelos de redes neuronales pueden ser sensibles a las variaciones en los datos, esto no necesariamente indica un problema mayor.

El MSE en el conjunto de entrenamiento es de 130.19, lo que es razonablemente bajo, y se traduce en un RMSE de 11.41. El RMSE, al igual que el MSE, nos ayuda a entender la magnitud de los errores, y en este caso, refleja que el modelo está haciendo predicciones relativamente precisas en el conjunto de entrenamiento, con un error promedio de alrededor de 11.41, lo cual es aceptable dadas las características del problema. El valor de $R^2$ en este conjunto es de 0.9767, lo que sugiere que el modelo explica aproximadamente el 97.67% de la variabilidad de los precios de la carne en los datos de entrenamiento, lo cual es una excelente indicación de que el modelo está capturando bien la relación subyacente entre las variables.

Cuando se observa el rendimiento del modelo en el conjunto de prueba, el Error Máximo disminuye ligeramente a 181.78, lo cual es una mejora con respecto al conjunto de entrenamiento, aunque sigue siendo un valor relativamente alto. El MSE en el conjunto de prueba es de 182.59, lo que es más alto que el MSE en el conjunto de entrenamiento (130.19), indicando que el modelo comete errores ligeramente mayores en el conjunto de prueba. Esto puede ser indicativo de un ligero sobreajuste (*overfitting*), ya que el modelo ha sido entrenado para adaptarse bien a los datos de entrenamiento, pero presenta una leve pérdida de precisión en datos no vistos. No obstante, este aumento no es excesivo, lo que sugiere que el modelo sigue manteniendo un buen rendimiento en general.

El RMSE en el conjunto de prueba es de 13.51, lo que es un poco mayor que el valor observado en el conjunto de entrenamiento (11.41), como es de esperar en la mayoría de los modelos, donde los datos de prueba tienden a mostrar una ligera disminución en el rendimiento en comparación con los datos de entrenamiento. Sin embargo, el error sigue siendo bajo, lo que indica que el modelo continúa haciendo predicciones bastante ajustadas.

Finalmente, el valor de $R^2$ en el conjunto de prueba es de 0.9673, lo que, aunque es ligeramente inferior al valor de entrenamiento (0.9767), sigue siendo un resultado muy favorable. Esto sugiere que el modelo es capaz de generalizar bien a nuevos datos, explicando aproximadamente el 96.7% de la variabilidad en los precios de la carne vacuna en el conjunto de prueba.

En resumen, los resultados del segundo modelo propuesto muestran un rendimiento excelente, con valores de $R^2$ superiores al 96% tanto en el conjunto de entrenamiento como en el de prueba, lo que indica que el modelo es capaz de capturar de manera efectiva la relación entre las variables. Aunque hay ligeras diferencias entre los errores en ambos conjuntos, los resultados siguen dentro de un rango aceptable, sugiriendo que el modelo tiene una capacidad notable para predecir los precios de la carne vacuna con alta precisión.

## 4.3. XGBRegressor vs Redes Neuronales

A continuación, una comparativa exhaustiva entre los resultados obtenidos por el modelo XGBRegressor y el modelo de Redes Neuronales, considerando diversas métricas de rendimiento clave. Ambas metodologías tienen características y ventajas que podrían influir en la decisión de cuál utilizar en función de los requisitos del problema y las limitaciones del entorno. A continuación, se comparan ambos modelos en base a las métricas de Error Máximo**,** MSE**,** RMSE**,** $R^2$ y Generalización, con un análisis sobre sus fortalezas y debilidades.

***Tabla 2.*** *Resultados Ambos Modelos (Train)*

| | XGBoost | Red Neuronal |
|---|---|---|
| **Max Error** | 172,0074 | 191,1174 |
| **MSE** | 151,9716 | 130,1943 |
| **RMSE** | 12,3276 | 11,4102 |
| **$R^2$** | 0,9728 | 0,9767 |

*Tabla 3. Resultado Ambos Modelos (Test)*

| | XGBoost | Red Neuronal |
|---|---|---|
| Max Error | 170,4263 | 181,7773 |
| MSE | 180,3797 | 182,5931 |
| RMSE | 13,4305 | 13,5127 |
| $R^2$ | 0,9677 | 0,9673 |

En términos de MSE y RMSE, el modelo de Redes Neuronales muestra un rendimiento ligeramente superior en el conjunto de entrenamiento, con un MSE de 130.19 frente a 151.971 de XGBRegressor, y un RMSE de 11.41 frente a 12.327. Sin embargo, en el conjunto de prueba, XGBRegressor mantiene un rendimiento más consistente en comparación con Redes Neuronales, donde el MSE y RMSE son ligeramente más altos. Ambos modelos muestran una capacidad de generalización sólida, con pequeñas diferencias en los valores de $R^2$ entre entrenamiento y prueba, lo que sugiere un comportamiento robusto y bien ajustado de ambos modelos.

En resumen, los dos modelos presentan un rendimiento similar en términos de capacidad de generalización y precisión, con pequeñas variaciones entre ellos en las métricas de error y ajuste. Ambos muestran un excelente desempeño, con algunas diferencias que podrían influir dependiendo de los objetivos específicos del análisis.

# 5. Conclusiones

Este trabajo nació con un objetivo claro, desarrollar una herramienta analítica y predictiva que permita a los productores ganaderos optimizar su toma de decisiones, facilitando la planificación de sus ventas, maximización de beneficios y una mayor competitividad. Y para lograrlo se buscó comprender de manera más clara y profunda las variables que inciden en la determinación del precio de venta de sus animales en el mercado ganadero.

Al abordar la investigación fue una gran sorpresa descubrir que, a pesar de ser la carne vacuna un producto tan importante y representativo para nuestro país y nuestra idiosincrasia como argentinos, hasta la fecha no existía ningún análisis similar al que se logró en este trabajo. Y no solo que no existía análisis alguno de como él precio del ganado en pie puede variar en diferentes escenarios. Si no que tampoco existen herramientas que faciliten al productor la posibilidad de identificar en un solo lugar todas las variables que forman los diferentes escenarios que influye en el precio que va a tener su ganado cuando intente venderlo, y mucho menos que esta misma herramienta le dé una predicción del precio al que puede llegar su ganado.

La creación del *dataset* para este trabajo fue un proceso complejo y exigente que requirió un esfuerzo considerable para asegurar que estuviera en condiciones óptimas para entrenar y evaluar tanto un modelo XGBoost como un modelo de Redes Neuronales. El *dataset* proviene de múltiples fuentes heterogéneas, lo que implicó una ardua tarea de integración y limpieza de datos. A lo largo del proceso, muchas de las variables iniciales debieron ser replanteadas o modificadas para garantizar su consistencia y relevancia en el análisis. Este trabajo de

refinamiento no solo incluyó la resolución de problemas de formato y de calidad, sino también una constante iteración para ajustar las características del *dataset* a las necesidades del modelo, lo que resultó en varias idas y vueltas antes de obtener una versión final lista para la experimentación.

Los resultados obtenidos con los modelos XGBoost y Redes Neuronales fueron muy satisfactorios, especialmente al considerar que se trabajó inicialmente con configuraciones de hiperparámetros no ajustadas. Ambos modelos demostraron un alto poder predictivo desde las primeras etapas de procesamiento, superando las expectativas iniciales. Sin embargo, fue el modelo de Redes Neuronales el que sobresalió, demostrando una precisión superior en comparación con el modelo XGBoost. Esta diferencia en el desempeño puede atribuirse a la capacidad inherente de las redes neuronales para modelar relaciones no lineales complejas, lo que les otorga una ventaja en la predicción de precios en mercados tan dinámicos y multifactoriales como el ganadero.

A pesar de que el modelo XGBoost no alcanzó la misma precisión que las Redes Neuronales, sus resultados no fueron en absoluto despreciables. De hecho, el desempeño de XGBoost fue más que satisfactorio para el propósito del análisis y, además, permitió realizar una búsqueda más exhaustiva de los hiperparámetros, lo que resultó en una mejora significativa en su capacidad predictiva. Esta mejora permitió que el modelo XGBoost igualara, en términos de precisión, los resultados obtenidos por las Redes Neuronales, a pesar de las limitaciones de poder computacional disponibles. Es importante resaltar que, debido a la exigencia de recursos computacionales, no fue posible llevar a cabo una búsqueda más profunda de hiperparámetros para el modelo de Redes Neuronales, lo que limita su optimización completa dentro del alcance del presente trabajo.

En conclusión, la elección entre XGBoost y Redes Neuronales debe basarse principalmente en la disponibilidad de poder computacional, ya que ambos modelos pueden ofrecer una performance excelente en la tarea de predicción de precios del ganado. Si bien las Redes Neuronales presentaron un mejor desempeño en términos de precisión, el modelo XGBoost resultó ser una alternativa muy viable y eficaz, especialmente cuando los recursos computacionales son limitados. Este trabajo ha puesto de manifiesto la importancia de contar con herramientas predictivas robustas que permitan a los productores ganaderos anticiparse a las fluctuaciones del mercado, optimizando así sus decisiones y mejorando su posicionamiento en un entorno económico cada vez más competitivo. Por último, se destaca la necesidad de continuar desarrollando y perfeccionando este tipo de herramientas, incorporando nuevas fuentes de datos y avanzando en técnicas de modelado, para proporcionar a los productores ganaderos una herramienta aún más precisa y accesible que les permita enfrentarse a los desafíos del mercado con mayor seguridad y eficiencia.

# 6. Uso Práctico

Ante la incertidumbre del precio que el productor podría obtener por su ganado en el Mercado Agroganadero, sumado a los costos de transporte y los riesgos asociados a la pérdida de peso del ganado durante el traslado (desbaste) o posibles lesiones y muertes antes de la venta, el productor podría optar por vender al frigorífico local, a pesar de que el precio que este ofrezca sea inferior. La seguridad proporcionada por el frigorífico en términos de la transacción puede

ser un factor determinante, ya que la oferta en el Mercado Agroganadero no siempre proporciona una referencia clara del valor real del ganado. Así, para que el productor elija el Mercado Agroganadero, debe haber una diferencia significativa entre el precio ofertado por este y el del frigorífico local, una diferencia que contemple los costos adicionales mencionados y que permita al productor obtener una ganancia superior.

En este contexto, la predicción de precios de ganado desarrollada en el presente trabajo tiene el potencial de ayudar al productor a maximizar sus ganancias y mitigar los riesgos asociados con la incertidumbre en el mercado. Supongamos que el productor necesita vender una parte de su hacienda compuesta por Novillitos, los cuales, por su peso, se encuentran en la subcategoría de menos de 390 kg. Este tipo de ganado es muy específico, de alta calidad y altamente demandado en el mercado.

Para simplificar la simulación, supongamos que el clima para la semana siguiente se pronostica cálido, sin lluvias, lo que es un factor relevante dado que la falta de precipitaciones garantiza la accesibilidad del camión para el retiro del ganado del campo, según los informes del Servicio Meteorológico Nacional. Además, se considera el consignatario con el cual el productor llevará a cabo la operación en el Mercado Agroganadero, presumiendo que trabaja habitualmente con dos consignatarios: Martin G. Lalor S.A. y Saenz Valiente, Bullrich y Cía.

Las variables conocidas por el productor incluyen la categoría, subcategoría, número de cabezas, kilos promedio del ganado a vender, el departamento en el que se encuentra la hacienda, la semana en la que se desea realizar la operación, las condiciones climáticas (temperatura y precipitaciones) en dicha semana y los consignatarios con los que podría operar. Sin embargo, existen tres variables adicionales que aún no se conocen: la cotización del dólar oficial, la cotización del dólar paralelo y el precio del kilo de maíz. Estas variables se pueden suponer para crear diferentes escenarios y evaluar la combinación de condiciones que permita al productor decidir si le resulta más rentable vender su ganado en el Mercado Agroganadero o en el frigorífico local.

En cuanto a la cotización del dólar oficial y paralelo, se pueden proyectar dos escenarios: uno en el que los valores se mantienen estables y otro en el que se experimenta una suba significativa. Respecto al precio del kilo de maíz, se pueden establecer tres posibles escenarios: uno en el que el precio baja debido a una oferta abundante y baja demanda, otro en el que el precio se mantiene estable, y finalmente, uno en el que el precio aumenta debido a una alta demanda frente a una oferta limitada. De esta manera, se pueden generar 24 escenarios posibles que serán evaluados por el modelo para predecir el precio en cada uno de ellos.

Con los resultados de estas predicciones, el productor podrá tomar decisiones informadas sobre la opción más rentable para la venta de su ganado, en función de los posibles escenarios que se presenten. Esto le permitirá maximizar su rentabilidad económica en cada operación, minimizando los riesgos asociados y optimizando su estrategia de comercialización.

# 7. Referencias


Coffey, B. K., Tonsor, G. T., & Schroeder, T. C. (2018). Impacts of changes in market fundamentals and price momentum on hedging live cattle. Journal of Agricultural and Resource Economics, 18-33.

Fernández Batalla, O. (2021). Extracción de datos y modelo predictor para el precio de alquiler de viviendas de Barcelona.

Fosca Gamarra, A. (2020) Desarrollo de un modelo para la predicción del precio del cobre empleando herramientas de Machine Learning.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. Springer Series in Statistics. Springer.

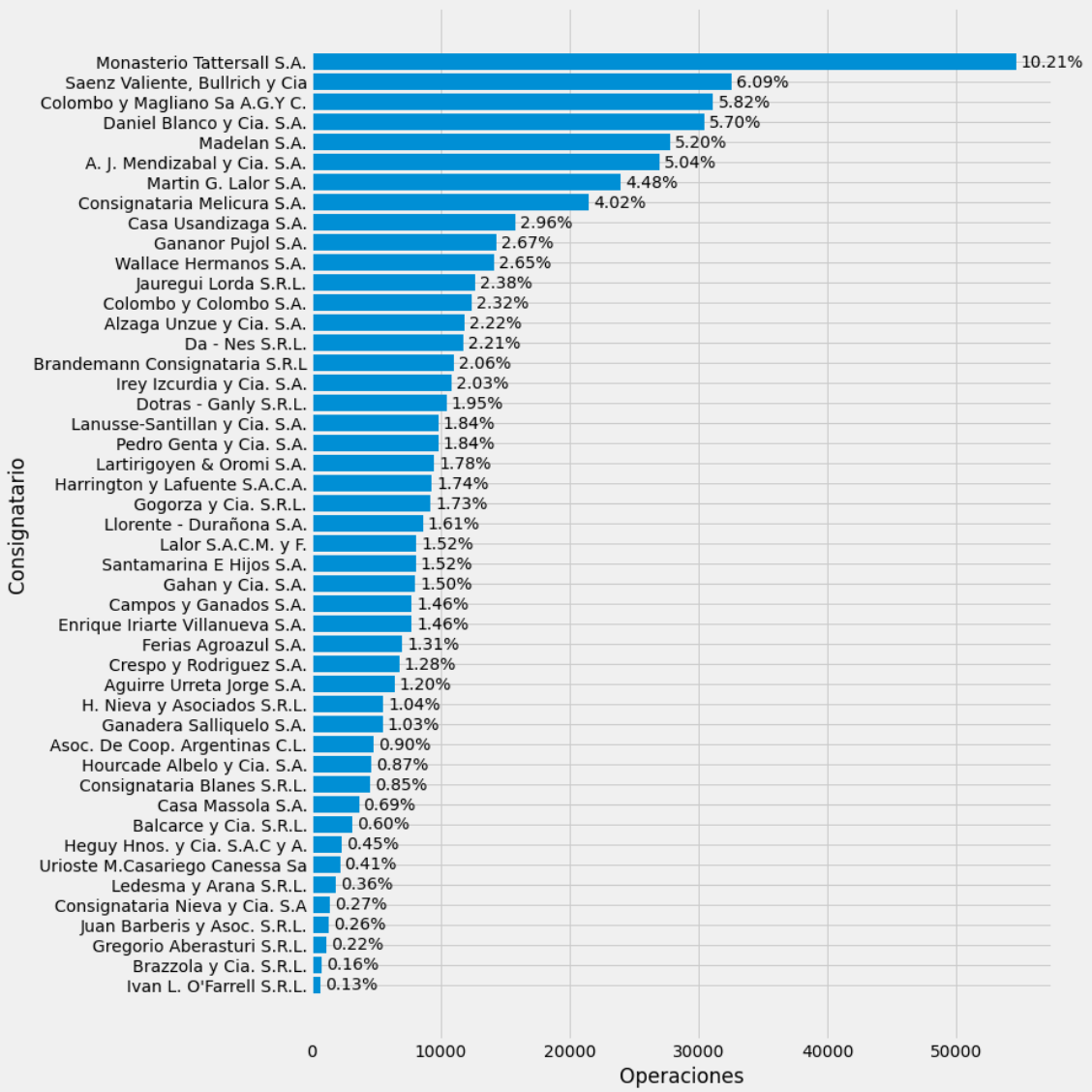
López-García, M. D. R., Martínez-Damián, M. Á., & Arana-Coronado, J. J. (2022). Predictores del precio de maíz blanco en Jalisco y Michoacán. Revista mexicana de ciencias agrícolas, 13(2), 261-272.

Martínez, J., Améstica-Rivas, L., Parisi, A., & Gurrola, C. (2021). Predictor alternativo del precio de los activos financieros. El caso de la plata y el oro. Oikos Polis, 6(2), 30-54.


# Apéndice A. Compilado y Normalizado

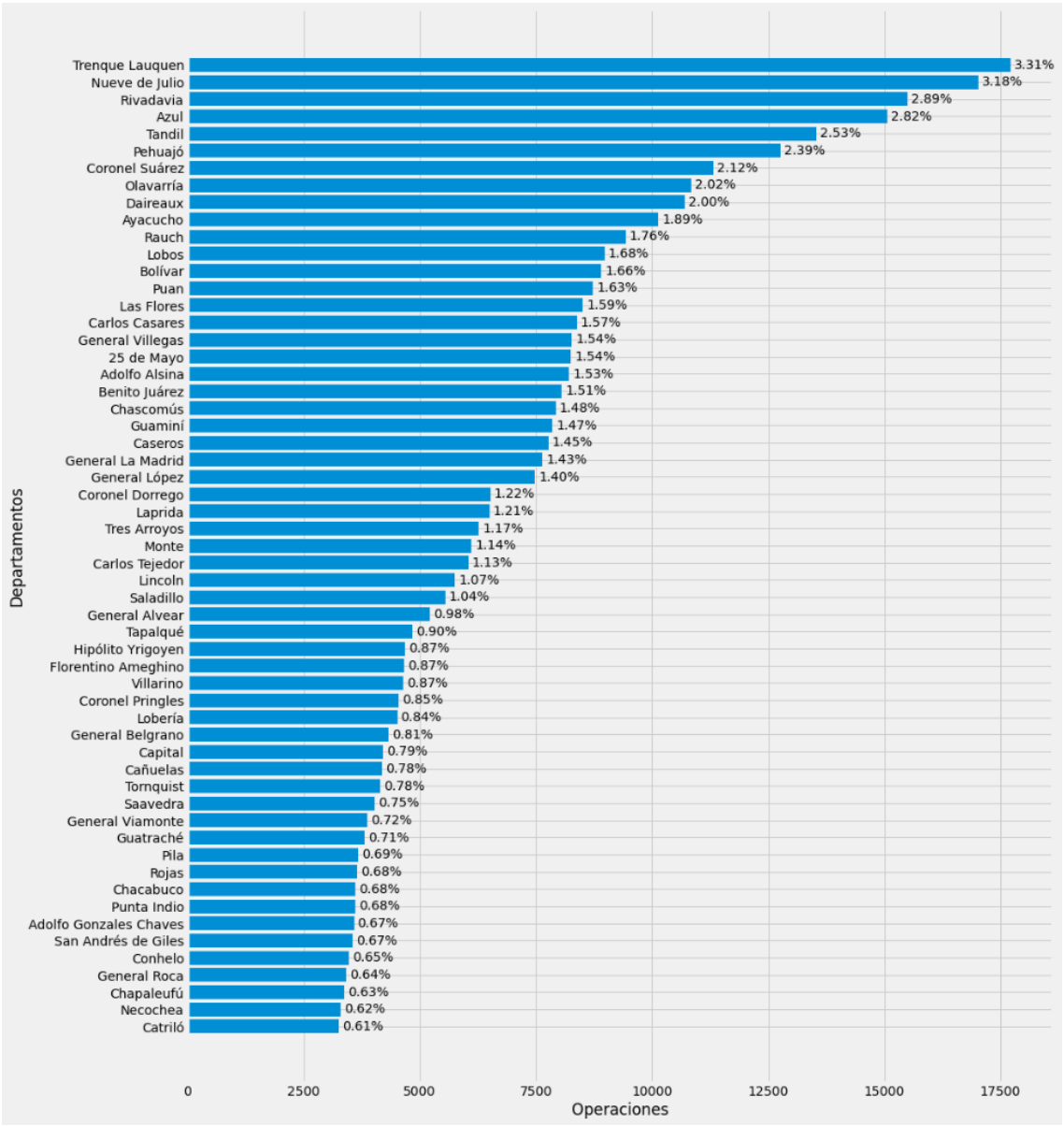
```
RangeIndex: 535240 entries, 0 to 535239
Data columns (total 17 columns):
 #   Column            Non-Null Count   Dtype
---  ------            --------------   -----
 0   week              535240 non-null  object
 1   consignatario     535240 non-null  int64
 2   depto             535240 non-null  int64
 3   categoria         535240 non-null  int64
 4   subcategoria      535240 non-null  int64
 5   cabezas           535240 non-null  int64
 6   kilos_prom        535240 non-null  float64
 7   precio            535240 non-null  float64
 8   dolar             535240 non-null  float64
 9   dolar_blue        535240 non-null  float64
 10  maiz              535240 non-null  float64
 11  lluvia            535240 non-null  float64
 12  lluvia2           535240 non-null  float64
 13  lluvia_acumulada  535240 non-null  float64
 14  dias_sin_lluvia   535240 non-null  int64
 15  temp_max          535240 non-null  float64
 16  temp_min          535240 non-null  float64
dtypes: float64(10), int64(6), object(1)
memory usage: 69.4+ MB
```

# Apéndice B. Operaciones Totales x Consignatario




Fuente: elaboración propia en base a datos del MAG

# Apéndice C. Operaciones Totales x Departamento (75%)




*Fuente: elaboración propia en base a datos del MAG.*

# Apéndice D. Grid Search

```
[CV 4/5; 237/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=99,
objective=reg:squarederror, subsample=0.5
[CV 4/5; 237/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=99,
objective=reg:squarederror, subsample=0.5;, score=0.959 total time= 3.1min
[CV 2/5; 239/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=149,
objective=reg:squarederror, subsample=0.5
[CV 2/5; 239/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=149,
objective=reg:squarederror, subsample=0.5;, score=0.960 total time= 2.9min
[CV 5/5; 240/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=149,
objective=reg:squarederror, subsample=0.7
[CV 5/5; 240/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=1, n_estimators=149,
objective=reg:squarederror, subsample=0.7;, score=0.961 total time= 2.9min
[CV 4/5; 242/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=50,
objective=reg:squarederror, subsample=0.7
[CV 4/5; 242/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=50,
objective=reg:squarederror, subsample=0.7;, score=0.955 total time= 2.2min
[CV 5/5; 243/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=99,
objective=reg:squarederror, subsample=0.5
[CV 5/5; 243/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=99,
objective=reg:squarederror, subsample=0.5;, score=0.959 total time= 2.7min
[CV 3/5; 245/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=149,
objective=reg:squarederror, subsample=0.5
[CV 3/5; 245/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=3, n_estimators=149,
objective=reg:squarederror, subsample=0.5;, score=0.960 total time= 3.3min
[CV 1/5; 247/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=50,
objective=reg:squarederror, subsample=0.5
[CV 1/5; 247/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=50,
objective=reg:squarederror, subsample=0.5;, score=0.957 total time= 2.2min
[CV 2/5; 248/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=50,
objective=reg:squarederror, subsample=0.7
[CV 2/5; 248/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=50,
objective=reg:squarederror, subsample=0.7;, score=0.956 total time= 2.3min
[CV 5/5; 249/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=99,
objective=reg:squarederror, subsample=0.5
[CV 5/5; 249/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=99,
objective=reg:squarederror, subsample=0.5;, score=0.959 total time= 3.4min
[CV 2/5; 251/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=149,
objective=reg:squarederror, subsample=0.5
[CV 2/5; 251/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=5, min_child_weight=5, n_estimators=149,
objective=reg:squarederror, subsample=0.5;, score=0.960 total time= 4.2min
[CV 1/5; 253/288] START colsample_bytree=0.7, learning_rate=0.1, max_depth=7, min_child_weight=1, n_estimators=50,
objective=reg:squarederror, subsample=0.5
[CV 1/5; 253/288] END colsample_bytree=0.7, learning_rate=0.1, max_depth=7, min_child_weight=1, n_estimators=50,
objective=reg:squarederror, subsample=0.5;, score=0.961 total time= 2.5min
```