PREDICCIÓN DE ERRORES DEL MODELO BLACK-SCHOLES MEDIANTE MODELOS CLÁSICOS DE CIENCIA DE DATOS

Julián Rojas Ramírez
Código 9027554

Natalia María Tangarife Acevedo
Código 9027315

Proyecto Aplicado para optar al título de
Magíster en Ciencia de Datos

Director
Cristian Alejandro Torres Valencia

FACULTAD DE INGENIERÍA Y CIENCIAS
MAESTRÍA EN CIENCIA DE DATOS
SANTIAGO DE CALI, OCTUBRE 2025

Dedicatoria

Le dedicamos el resultado de este trabajo a nuestras familias.

Agradecimientos

Agradecemos a nuestro director de trabajo de grado, Cristian Alejandro Torres Valencia, quien desde su área de conocimiento aportó ideas innovadoras de alto nivel y dirigió este proyecto de investigación con las mejores intenciones, logrando un documento con rigurosidad científica y técnica.

Al programa de Maestría en Ciencia de Datos de la Pontificia Universidad Javeriana Cali, por brindarnos las herramientas metodológicas y técnicas necesarias para desarrollar este proyecto. Cada una de las materias cursadas constituyó un pilar fundamental para abordar las diferentes dimensiones de esta investigación: aprendizaje automático, análisis estadístico, procesamiento de datos, visualización de información y modelamiento predictivo, entre otras.

Gracias a los profesores de la maestría en Ciencia de Datos por fomentar en sus clases el pensamiento crítico, el rigor metodológico y el uso ético de los datos, incentivándonos constantemente hacia la investigación científica aplicada y la innovación tecnológica.

A la Pontificia Universidad Javeriana Cali, por su compromiso con la formación de profesionales íntegros y su apoyo institucional para el desarrollo de proyectos de investigación que contribuyen al avance del conocimiento científico en Colombia.

Finalmente, agradecemos a nuestras familias y amigos por su apoyo incondicional durante este proceso académico, su paciencia y motivación fueron fundamentales para alcanzar esta meta.

FICHA RESUMEN

TÍTULO DEL PROYECTO: Predicción de errores del modelo Black-Scholes mediante modelos clásicos de ciencia de datos

1. **ÁREA DE TRABAJO:** Ciencia de Datos Aplicada al Sector Financiero
2. **TIPO DE PROYECTO** (Aplicado, Innovación, Investigación): Aplicado
3. **ESTUDIANTE(S):** Julián Rojas Ramírez y Natalia María Tangarife Acevedo
4. **CORREO ELECTRÓNICO:**
   a) Julián Rojas hulian@javerianacali.edu.co
   b) Natalia Tangarife ntangari@javerianacali.edu.co
5. **DIRECCIÓN Y TELÉFONO:**
   a) Julián Rojas Carrera 105 A #70-03 (Bogotá)
   b) Natalia Tangarife Calle 77 Sur #35B71 Apto 2004. (Sabaneta - Antioquia)
6. **DIRECTOR:** Cristian Alejandro Torres Valencia
7. **VINCULACIÓN DEL DIRECTOR:** Universidad Javeriana Cali - Departamento Electrónica
8. **CORREO ELECTRÓNICO DEL DIRECTOR:** cristian.torres@javerianacali.edu.co
9. **PALABRAS CLAVE:** Opciones Europeas, Modelo Black-Scholes, Valoración Financiera, Modelos predictivos, Predicción de errores
10. **FECHA DE INICIO:** mayo de 2025
11. **DURACIÓN ESTIMADA (En meses):** 10 meses
12. **RESUMEN:** *La valoración precisa de opciones financieras es crucial para la toma de decisiones de inversión informada. El modelo Black-Scholes, aunque fundamental para la valoración teórica de opciones europeas, presenta limitaciones inherentes a sus supuestos que generan discrepancias con los precios reales del mercado. Este proyecto desarrolló un sistema de predicción basado en técnicas de ciencia de datos para predecir y corregir los errores sistemáticos del modelo Black-Scholes, utilizando un dataset de 40,337 contratos de opciones sobre 10 activos representativos del mercado estadounidense. Siguiendo la metodología CRISP-DM, se implementaron y compararon progresivamente seis modelos de aprendizaje automático de complejidad creciente, evaluando su capacidad para ajustar los precios teóricos. Los resultados demuestran que los modelos de ensamble logran las mayores reducciones del* RMSE*, con Random Forest como mejor modelo para opciones call (82,20 %) y XGBoost para opciones put (89,54 %) respecto al baseline, evidenciando que el aprendizaje automático puede complementar eficazmente los modelos teóricos de valoración financiera sin reemplazar su estructura fundamental.*

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Las opciones europeas son instrumentos financieros derivados cuya valoración se ha basado tradicionalmente en el modelo de Black-Scholes. Este modelo, aunque ampliamente utilizado en la industria financiera, se fundamenta en supuestos como la volatilidad constante y la eficiencia perfecta del mercado, condiciones que raramente se cumplen en escenarios reales. Como consecuencia, se generan diferencias sistemáticas entre el valor teórico estimado por el modelo y el precio real observado en el mercado, lo que impacta negativamente en la precisión de las decisiones de inversión, la cobertura de riesgos y la identificación de oportunidades de arbitraje.

Con el objetivo de reducir estas discrepancias, este trabajo integró técnicas de ciencia de datos para predecir y corregir los errores sistemáticos del modelo Black-Scholes en opciones europeas *call* y *put* sobre diez activos representativos del mercado estadounidense. El enfoque propuesto no reemplaza el modelo clásico sino que lo complementa: a partir del precio teórico calculado por Black-Scholes, un modelo de aprendizaje automático estima el error esperado y lo utiliza para ajustar dicho precio, acercándolo al valor real de mercado. El proyecto incluyó la construcción de un dataset de 40,337 contratos de opciones, el diseño e implementación de seis modelos predictivos bajo la metodología CRISP-DM, y la validación de su desempeño mediante métricas estadísticas estándar.

La pregunta central que orientó el trabajo fue: ¿cómo predecir la diferencia entre el valor teórico calculado por Black-Scholes y el valor real de mercado de opciones europeas *call* y *put* utilizando técnicas de aprendizaje automático? Para responderla, se estructuró una base de datos con variables financieras relevantes descargadas de Yahoo Finance, se entrenaron modelos predictivos de complejidad creciente sobre contratos de opciones reales, y se evaluó el impacto del ajuste propuesto en la precisión de la valoración. Los resultados evidenciaron mejoras superiores al 82 % en el RMSE para opciones call y al 89 % para opciones put respecto al modelo sin corrección, confirmando que el aprendizaje automático puede complementar eficazmente los modelos teóricos de valoración financiera.

1.1. Contextualizacion del proyecto

1.1.1. Definición del problema

Las opciones europeas son instrumentos financieros derivados cuyo precio es una variable fundamental en la toma de decisiones de inversión y gestión de riesgos. Para su valoración, el modelo de Black-Scholes (BS) ha sido el pilar teórico, proporcionando un valor estimado a partir de supuestos como la volatilidad constante y la eficiencia del mercado. Sin embargo, estas condiciones ideales no se cumplen en la realidad, lo que históricamente ha generado discrepancias significativas entre el precio teórico calculado por el modelo BS y el precio real observado en el mercado. Esta falta de precisión en la valoración ha representado un problema crítico en el mercado de opciones, ya que impacta negativamente en la efectividad de la gestión de portafolios, la cobertura de riesgos y la identificación de oportunidades de arbitraje.

Ante esta problemática, la evolución de la ciencia de datos y la disponibilidad de grandes volúmenes de información histórica abren la posibilidad de mejorar las estimaciones de precio. Por esta razón, este trabajo integró métodos de aprendizaje automático para abordar las limitaciones del modelo clásico. El estudio se centró en construir modelos predictivos que, en lugar de reemplazar el modelo Black-Scholes, estimaran sus errores para ajustar el valor teórico. Este método permitió mejorar la precisión en la valoración de opciones europeas para una toma de decisiones más informada, estableciendo una comparación objetiva entre el enfoque clásico y las estimaciones mejoradas con técnicas de aprendizaje automático.

1.1.2. Planteamiento del problema

Las opciones europeas son instrumentos financieros derivados que otorgan a su titular el derecho, pero no la obligación, de comprar (call) o vender (put) un activo subyacente a un precio predeterminado en una fecha futura específica. A diferencia de las opciones americanas, las europeas solo pueden ejercerse en su fecha de vencimiento, lo que introduce particularidades en su valoración y estrategias de inversión [3].

El precio de las opciones es una variable fundamental en los mercados financieros, ya que influye en la toma de decisiones de inversionistas, administradores de riesgo y entidades financieras. Para estimar su valor, se han desarrollado diversos modelos matemáticos, entre ellos el modelo de Black-Scholes (BS), que proporciona un valor teórico basado en supuestos como volatilidad constante, eficiencia de mercado y ausencia de costos de transacción. No obstante, estas hipótesis no siempre se cumplen en la realidad, lo que genera diferencias entre el precio teórico calculado por Black-Scholes y el precio de mercado de la opción [4].

Esta diferencia entre los valores teóricos y observados tiene implicaciones significativas en la gestión de portafolios, la cobertura de riesgos y la detección de oportunidades de arbitraje; dado que la falta de precisión en la valoración puede impactar negativamente la toma de decisiones financieras. Con la evolución de la ciencia de datos y la mayor disponibilidad de datos históricos del mercado, surge la posibilidad de mejorar las estimaciones de precio mediante modelos predictivos basados en técnicas de aprendizaje automático y análisis estadístico [5].

La mejora en la precisión de la valoración de opciones europeas no solo beneficia a inversionistas y entidades financieras en la toma de decisiones más informadas, sino que también contribuye al desarrollo de estrategias más robustas de gestión de riesgo y optimización de portafolios. Este estudio aporto un enfoque innovador al integrar métodos de ciencia de datos, lo que permitió una comparación objetiva entre las estimaciones derivadas de técnicas de aprendizaje automático y el enfoque clásico de Black-Scholes [6].

1.1.3. Formulación del problema - Pregunta principal:

- ¿Cómo predecir la diferencia entre el valor teórico calculado por el modelo de Black-Scholes y el valor real de mercado de las opciones tipo call y put utilizando técnicas de aprendizaje automático?

1.1.4. Preguntas de Sistematización

- ¿Qué variables explicativas del mercado financiero influyen significativamente en las diferencias entre el valor real y el teórico de una opción?
- ¿Qué técnicas de aprendizaje automático pueden mejorar la predicción de la diferencia entre el valor teórico y el valor de mercado de las opciones europeas call y put?
- ¿En qué medida el ajuste del valor teórico con el error predicho mejora la aproximación al precio real de mercado?
- ¿Qué nivel de error se mantiene luego del ajuste, y cómo se compara con el modelo tradicional de Black-Scholes sin corrección?
- ¿Cómo se comparan los modelos tradicionales de valoración como Black-Scholes (BS), con enfoques basados en aprendizaje automático en términos de precisión y aplicabilidad en el mercado financiero?

1.2. Objetivos.

1.2.1. Objetivo General

Desarrollar un sistema de predicción que estime las diferencias entre los valores reales y los precios teóricos de opciones europeas (call y put) calculados mediante el modelo Black-Scholes, utilizando técnicas de ciencia de datos para ajustar dichos valores y mejorar la precisión en su valoración, facilitando así una toma de decisiones de inversión más informada.

1.2.2. Objetivos Específicos

En el Cuadro 1.1 se relacionan los objetivos específicos con los resultados esperados en la sección correspondiente.

1. Diseñar y estructurar una base de datos con las variables del mercado financiero que impactan significativamente la discrepancia entre los valores reales y los precios teóricos de opciones europeas call y put según el modelo Black-Scholes, incorporando en su construcción, un procedimiento de preprocesamiento de las variables, así como técnicas de normalización para optimizar el desempeño de los modelos de predicción de errores en la valoración financiera.

2. Construir e implementar distintos modelos de aprendizaje automático para predecir con precisión los errores entre los valores reales y teóricos de las opciones europeas call y put, y utilizar dichos errores para ajustar el valor teórico calculado mediante el modelo de Black-Scholes.

3. Evaluar el rendimiento de los modelos de aprendizaje automático a partir de métricas de desempeño.

4. Medir el nivel de mejora en la estimación de precios luego de realizar el ajuste con el uso de modelos de aprendizaje automático, y compararlo con el valor teórico generado por el modelo Black-Scholes, para determinar así el ajuste que mejor se aproxima al valor real de mercado.

Capítulo 2

Marco de referencia

2.1. Marco Teórico

A continuación, se detalla un glosario de conceptos importantes requeridos para entender el contexto dentro del que se desarrolla este proyecto de tesis de maestría.

- **Definición de Opción Financiera:** Una opción es un contrato que otorga a su tenedor el derecho, pero no la obligación, de comprar o vender un activo subyacente a un precio determinado (precio de ejercicio o strike price) en o antes de una fecha determinada (fecha de vencimiento). Por este derecho, el comprador de la opción paga una prima al vendedor (o emisor) de la opción [2].
- **Activo Subyacente (S_0):** El activo subyacente en el marco de opciones europeas es el activo financiero sobre el cual se basa el contrato de opción. Puede ser una acción, un índice bursátil, una divisa o cualquier otro activo negociable [2].
- **Definición de Opción Europea:** Una Opción Europea es un contrato de opción que solo pueden ser ejercido por el tenedor en la fecha de vencimiento del contrato. No pueden ejercerse antes de esta fecha [2].
- **Definición del Modelo Black-Scholes (-Merton):** El modelo Black-Scholes-Merton es un modelo matemático que proporciona una fórmula teórica para determinar el precio de opciones de estilo europeo. El modelo calcula el precio de una opción de compra o venta basándose en el precio actual del activo subyacente, el precio de ejercicio, el tiempo hasta el vencimiento, la tasa de interés libre de riesgo y la volatilidad del activo [3] - [4] .
- **Definición de Opción de compra - Call:** Una opción de compra (call) otorga al tenedor el derecho a comprar el activo subyacente en una fecha determinada a un precio determinado [2].
- **Definición de Opción de venta - Put:** Una opción de venta (put) otorga al tenedor el derecho a vender el activo subyacente en una fecha determinada a un precio determinado [2].

- **Oportunidad de Arbitraje:** Una oportunidad de arbitraje en opciones se refiere a la posibilidad de obtener una ganancia libre de riesgo al aprovechar discrepancias entre el precio teórico de una opción (por ejemplo, calculado con el modelo Black-Scholes) y su precio real de mercado. Estas oportunidades surgen cuando los supuestos del modelo no se cumplen completamente, generando ineficiencias que pueden ser explotadas por operadores sofisticados [7].

- **Precio strike (K):** es el Precio del Ejercicio, que corresponde al precio fijo pactado en el contrato al que el comprador de la opción tiene el derecho más no la obligación de comprar (en una Call) o vender (en una Put) el activo en la fecha de vencimiento [2].

- **Moneyness:** es una medida que indica el grado de utilidad o beneficio potencial que tiene una opción financiera en función de su precio de ejercicio (K) comparado con el precio actual del activo subyacente (S0). Refleja qué tan favorable es una opción para ser ejercida en un momento dado. En otras palabras, representa la posición relativa de una opción respecto al mercado, y permite saber si ejercerla generaría ganancia (*in the money*), equilibrio (*at the money*) o pérdida (*out of the money*). Esta característica es clave para entender el valor intrínseco de la opción y su probabilidad de ejercicio [2].

- **Valor Intrínseco de una opción:** representa el beneficio inmediato que obtendría el titular de una opción si decidiera ejercerla en este preciso momento. Es decir, es el valor económico real que tiene la opción sin considerar el tiempo restante ni la volatilidad futura. Para una opción de compra *call* , el valor intrínseco es por tanto $max(S-K,0)$. Para una opción de venta *put*, es $max(K-S,0)$. Esto significa que si el activo vale más que el precio de ejercicio en una *call*, hay ganancia potencial, y si el activo vale menos que el precio de ejercicio en una *put*, también hay ganancia [2].

- **Terminología de Posición (Moneyness):** Para una adecuada interpretación de la valoración y el riesgo de una opción, es necesario establecer formalmente su posición relativa en el mercado o Moneyness. Esta clasificación es fundamental, ya que determina el valor intrínseco de la opción y su probabilidad de ser ejercida. En la Tabla 2.1: Clasificación de Opciones según Moneyness, se establecen las tres categorías principales *ITM, ATM, OTM* basándose en la condición de su valor intrínseco. Y posteriormente, en la Tabla 2.1: Clasificación Específica por Tipo de Opción , se detalla la condición matemática que debe cumplirse para cada tipo de opción *Call* y *Put* en relación con el precio del activo subyacente S_0 y el precio de ejercicio K [2].

Tabla 2.1: Clasificación de Opciones según Moneyness. *Fuente:* Adaptada de [2]

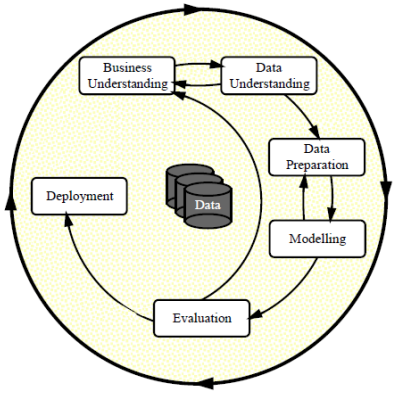
Término	Abreviatura	Definición	Valor Intrínseco (VI)
In The Money	ITM	La opción tiene un **valor intrínseco positivo**; si fuera ejercida inmediatamente, resultaría en una ganancia para el titular.	$VI > 0\$$
At The Money	ATM	El precio del subyacente es aproximadamente **igual** al precio de ejercicio. El valor intrínseco es cero o casi cero.	$VI \approx 0\$$
Out Of The Money	OTM	La opción **no tiene valor intrínseco**; si fuera ejercida inmediatamente, resultaría en una pérdida o costo innecesario.	$VI = 0\$$

Tabla 2.2: Clasificación Específica por Tipo de Opción. *Fuente:* Adaptada de [2]

Opción	ITM (In The Money)	ATM (At The Money)	OTM (Out Of The Money)
Call (Opción de Compra)	$S_0 > K\$$	$S_0 \approx K\$$	$S_0 < K\$$
Put (Opción de Venta)	$S_0 < K\$$	$S_0 \approx K\$$	$S_0 > K\$$

2.1.1. CRISP-DM Cross-Industry Standard Process for Data Mining como marco de trabajo

Como se describe en la figura 2.1, el CRISP-DM es un modelo de proceso integral propuesto para



latex

Figura 2.1: Metodología CRISP-DM para minería de datos. *Fuente:* Tomada del articulo [1].

estructurar y llevar a cabo proyectos de minería de datos que fue diseñado para ser independiente tanto del sector industrial como de la tecnología utilizada y que los proyectos fueran escalables al buscar una práctica estándar. Está compuesto por seis fases que se describen a continuación.

Inicia con la fase **Business Understanding (Comprensión del Negocio)**, esta fase se centra en definir el problema y establecer objetivos claros desde la perspectiva tanto de la investigación como del negocio; comprende principalmente las actividades: determinar objetivos del negocio, evaluar situación, determinar objetivos de minería de datos, producir plan del proyecto. Luego la metodología continua con la fase **Data Understanding (Comprensión de los Datos)**, que implica explorar la calidad y relevancia de los datos disponibles mediante análisis exploratorios, estadísticos y visualizaciones. Este análisis inicial es crucial para identificar patrones, valores atípicos y datos faltantes que puedan afectar el modelado en cualquier disciplina; esta fase comprende principalmente: Recolectar Datos Iniciales, Describir Datos, Explorar Datos, Verificar Calidad de los Datos. Como tercera fase se tiene **Data Preparation (Preparación de los Datos)**, que consiste en limpiar, transformar y combinar los datos para hacerlos aptos para el modelado. La calidad de los resultados finales depende en gran medida de esta etapa, la cual debe adaptarse a las particularidades de cada disciplina; comprende principalmente: Descripción del conjunto de Datos, Seleccionar Datos, Limpiar Datos, Construir Datos, Integrar Datos, Formatear Datos. La cuarta fase es el **Modeling (Modelado)**, que se centra en seleccionar y entrenar algoritmos de machine learning o aplicar métodos estadísticos para resolver el problema planteado. La elección del modelo adecuado depende de la complejidad del problema y de la calidad de los datos; aquí se Selecciona la Técnica de Modelado, se Genera un Diseño de Prueba, se Construye el Modelo y se Evalúa el Modelo. Se sigue con la fase **Evaluation (Evaluación)** donde se revisan tanto las métricas como el impacto en el contexto de aplicación. Aquí en esta fase es fundamental verificar que el modelo cumpla con los objetivos planteados, y para ello se utilizan métricas cuantitativas y cualitativas. Comprende principalmente los siguientes ítems, Evaluar resultados de los modelos con respecto a los criterios de éxito del negocio, Definir los Modelos Aprobados, Revisión del Proceso y Decisión sobre modelo más apropiado para el caso de negocio. Por último, se tiene la fase **Deployment (Despliegue)** que es esencial para trasladar los modelos desarrollados a entornos productivos, transformando los resultados en soluciones prácticas y operativas. Esta etapa abarca la integración inicial, la puesta en marcha y el mantenimiento continuo del modelo. Es fundamental considerar aspectos clave como la automatización, la infraestructura tecnológica, la escalabilidad, la seguridad y la monitorización, para garantizar que el sistema opere de manera óptima en entornos dinámicos [1].

2.1.2. Modelo Black-Scholes

El **modelo Black-Scholes** considera que el precio del activo subyacente se distribuye según una normal logarítmica para la que su varianza es proporcional al tiempo. Los supuestos de los que parte este modelo son los siguientes [3]:

1. El precio del activo sigue una distribución normal logarítmica, por lo que los rendimientos se

distribuyen normalmente. Esta distribución log-normal implica que los precios del activo nunca pueden ser negativos, lo cual es una característica fundamental de los mercados financieros.

2. El valor de los rendimientos es conocido y es directamente proporcional al paso del tiempo.
3. No hay costos de transacción, así que se puede establecer una cobertura sin riesgos entre el activo y la opción sin ningún costo.
4. Los tipos de interés son conocidos y constantes.
5. Durante el período de ejercicio, la acción subyacente no pagará dividendos.
6. Las opciones son de tipo europeo.

Para calcular el precio teórico de una opción europea mediante el modelo Black-Scholes, que será la base para calcular la variable objetivo de la modelación (Error = Precio_Real - Precio_BS), las fórmulas matemáticas propuestas por Black y Scholes [**?**] son:

Opción europea tipo *Call*:

$$c = S_0 N(d_1) - K e^{-rT} N(d_2) \tag{2.1}$$

Opción europea tipo *Put*:

$$p = K e^{-rT} N(-d_2) - S_0 N(-d_1) \tag{2.2}$$

donde los parámetros d_1 y d_2 se calculan como:

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}} \tag{2.3}$$

$$d_2 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - \frac{\sigma^2}{2}\right) T}{\sigma\sqrt{T}} = d_1 - \sigma\sqrt{T} \tag{2.4}$$

Los parámetros del modelo se definen como:

c	Precio de la opción de compra europea
p	Precio de la opción de venta europea
S_0	Precio spot del activo subyacente (precio actual en el mercado)
K	Precio de ejercicio (*strike*) de la opción
r	Tasa de interés libre de riesgo (anualizada)
T	Tiempo hasta el vencimiento de la opción (en años)
σ	Volatilidad del activo subyacente (desviación estándar de los rendimientos logarítmicos anualizados)

$N(d)$ Función de distribución acumulada de una distribución normal estándar

2.1.3. Modelos de Machine Learning

En el sector financiero, la predicción directa de precios de opciones a menudo combina modelos teóricos fundamentales como el **Black-Scholes**, con técnicas de aprendizaje automático, dado que los supuestos de los modelos teóricos no siempre se cumplen en la realidad generando diferencias entre los precios teóricos y los de mercado. Aquí es donde los modelos de aprendizaje automático (ML) aportan mucho valor, y son el objeto de este trabajo de grado, al aproximarse a predicciones de estas diferencias o "errores". Los modelos que se explorará son:

- **Regresión lineal.**

  La Regresión Lineal es uno de los modelos más sencillos y populares para predicción en problemas de regresión. Se basa en la suposición de que existe una relación lineal entre las variables independientes y la variable dependiente. El objetivo es encontrar la línea de mejor ajuste que minimice la suma de los errores cuadrados entre las predicciones y los valores reales.

  Esta técnica de modelado estadístico es ampliamente utilizada para establecer relaciones cuantitativas entre una variable dependiente y múltiples variables independientes. En el contexto de este estudio, se emplea para modelar el error de Black-Scholes (Diferencia) como función lineal de características observables de las opciones.

  El modelo de regresión lineal múltiple se expresa matemáticamente como:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip} + \varepsilon_i \tag{2.5}$$

  donde:

  y_i es el valor de la variable dependiente (Diferencia) para la observación i.
  x_{ij} es el valor de la j-ésima variable explicativa para la observación i.
  β_0 es el intercepto (constante del modelo).
  $\beta_1, \beta_2, \ldots, \beta_p$ son los coeficientes de regresión parciales.
  $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ es el término de error aleatorio.
  p es el número de variables explicativas.

  El método de **mínimos cuadrados ordinarios** OLS tiene como objetivo encontrar los valores de los coeficientes ($\hat{\beta}$) que minimizan la suma de los errores al cuadrado entre el error de Black-Scholes real (y_i) y el error que predice nuestro modelo ($\mathbf{x}_i'\beta$). Al elevar el error al cuadrado, el método se asegura de que los errores grandes sean penalizados fuertemente, forzando al modelo a buscar el mejor ajuste general a sus 30,365 observaciones de entrenamiento [8] [9].

- **Random Forest.**

  El Random Forest es una técnica que combina múltiples árboles de decisión para generar predicciones más robustas y precisas [10]. Este método aborda dos limitaciones fundamentales de los árboles de decisión simples: la alta varianza y la tendencia al sobreajuste.

  Random Forest se construye sobre la técnica de *bagging* (Bootstrap Aggregating), introducida por Breiman [11], que consiste en entrenar múltiples modelos sobre muestras aleatorias con reemplazo y promediar sus predicciones. Adicionalmente, Random Forest incorpora una selección aleatoria de variables predictoras en cada división del árbol. Para un conjunto de entrenamiento $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, el algoritmo construye B árboles de decisión independientes, cada uno entrenado sobre una muestra bootstrap $\mathscr{D}_b^*$ del conjunto original:

$$\hat{f}_{RF}(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^{B} T_b(\mathbf{x}) \tag{2.6}$$

  donde $T_b(\mathbf{x})$ representa la predicción del b-ésimo árbol para una observación con características $\mathbf{x}$, y B corresponde al número total de árboles en el bosque.

  La innovación clave de Random Forest respecto al bagging tradicional radica en la introducción de aleatoriedad adicional durante la construcción de cada árbol. En cada nodo de división, en lugar de evaluar todos los p predictores disponibles, el algoritmo selecciona aleatoriamente un subconjunto de m variables (donde típicamente $m \ll p$) y determina la mejor división únicamente entre estas variables seleccionadas. Esta restricción, conocida como *feature bagging* o *random subspace method*, reduce la correlación entre los árboles del bosque y mejora la capacidad de generalización del modelo [12].

  Para problemas de regresión, cada árbol individual T_b minimiza el error cuadrático medio en su muestra bootstrap, particionando recursivamente el espacio de predictores en regiones $R_1, R_2, \ldots, R_J$ y asignando a cada región el valor promedio de las observaciones contenidas:

$$T_b(\mathbf{x}) = \sum_{j=1}^{J} \bar{y}_{R_j} \cdot \mathbb{I}(\mathbf{x} \in R_j) \tag{2.7}$$

  donde $\bar{y}_{R_j}$ representa la media de la variable objetivo en la región R_j y $\mathbb{I}(\cdot)$ es la función indicadora.

  Una ventaja adicional de Random Forest es la disponibilidad del error *Out-of-Bag* (OOB), que proporciona una estimación insesgada del error de generalización sin necesidad de un conjunto de validación separado. Para cada árbol T_b, aproximadamente un tercio de las observaciones no son seleccionadas en la muestra bootstrap correspondiente; estas observaciones OOB se utilizan para evaluar el desempeño del árbol, y el promedio de estos errores constituye una estimación del error de prueba:

$$\text{MSE}_{OOB} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i^{OOB}\right)^2 \tag{2.8}$$

donde $\hat{y}_i^{OOB}$ representa la predicción promedio para la observación i utilizando únicamente los árboles para los cuales dicha observación fue out-of-bag.

- **XGBoost (eXtreme Gradient Boosting)**

  XGBoost representa una implementación optimizada del algoritmo de potenciación de gradiente (gradient boosting), desarrollada por Chen y Guestrin. A diferencia de Random Forest, que construye árboles de forma independiente y los promedia mediante agregación (bootstrap), XGBoost emplea un enfoque secuencial donde cada nuevo árbol se entrena para corregir los errores residuales del ensamble anterior, técnica conocida como potenciación (boosting).

  El modelo final es la suma de las predicciones de todos los árboles. Para un problema de regresión con n observaciones, esto se expresa como:

$$\hat{f}(x) = \sum_{k=1}^{K} f_k(x) \tag{2.9}$$

  donde f_k representa el k-ésimo árbol de decisión y K es el número total de árboles. En cada iteración t, XGBoost añade un nuevo árbol f_t que minimiza la función de pérdida $\mathscr{L}^{(t)}$, definida como:

$$\mathscr{L}^{(t)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) \tag{2.10}$$

  donde l es la función de pérdida (error cuadrático en este estudio), $\hat{y}_i^{(t-1)}$ es la predicción acumulada hasta la iteración anterior, y $\Omega(f_t)$ es un término de regularización que penaliza árboles demasiado complejos para evitar sobreajuste [13] [14].

- **K-Nearest Neighbors (KNN)**

  K-Nearest Neighbors (KNN) es un algoritmo de aprendizaje supervisado no paramétrico que realiza predicciones basándose en la similitud entre observaciones [15]. A diferencia de otros algoritmos que construyen modelos explícitos durante el entrenamiento, KNN es un método de *lazy learning* que almacena el conjunto de entrenamiento completo y realiza todo el cómputo en el momento de la predicción [16].

  En su formulación para regresión, dada una nueva observación x_0, el algoritmo identifica las k observaciones más cercanas en el conjunto de entrenamiento y calcula la predicción como el

promedio de sus valores de respuesta:

$$\hat{f}(x_0) = \frac{1}{k} \sum_{x_i \in N_k(x_0)} y_i \tag{2.11}$$

donde $N_k(x_0)$ representa el conjunto de los k vecinos más cercanos a x_0, y y_i es el valor de la variable respuesta de cada vecino. La cercanía entre observaciones se mide comúnmente mediante la distancia euclidiana:

$$d(x_i, x_j) = \sqrt{\sum_{m=1}^{p} (x_{im} - x_{jm})^2} \tag{2.12}$$

donde p es el número de variables predictoras.

El hiperparámetro principal del algoritmo es k, el número de vecinos considerados. Valores pequeños de k producen fronteras de decisión más flexibles pero con mayor varianza, mientras que valores grandes generan predicciones más suaves pero con mayor sesgo, lo que constituye un ejemplo clásico del compromiso sesgo-varianza [16].

Una consideración crítica en la implementación de KNN es la **normalización de variables**. Dado que el algoritmo se basa en distancias euclidianas, variables con escalas diferentes dominarían el cálculo de distancias. Para abordar esto, se aplica comúnmente la normalización Z-score:

$$z_{ij} = \frac{x_{ij} - \bar{x}_j}{s_j} \tag{2.13}$$

donde $\bar{x}_j$ y s_j son la media y desviación estándar de la variable j, calculadas exclusivamente sobre el conjunto de entrenamiento para evitar fuga de información (*data leakage*) [17].

- **Redes Neuronales Artificiales (ANNs).**

  Las Redes Neuronales Artificiales (ANNs) son sistemas de procesamiento inspirados en el cerebro humano, que utilizan capas de nodos para aprender representaciones no lineales de los datos [18].

  El Perceptrón Multicapa (MLP) es un tipo de red neuronal artificial compuesta por múltiples capas de neuronas interconectadas. A diferencia de los modelos lineales, las redes neuronales pueden aprender relaciones no lineales complejas entre las variables predictoras y la variable objetivo mediante la combinación de transformaciones lineales y funciones de activación no lineales como ReLU y tanh [19].

  La arquitectura básica de un MLP consta de tres tipos de capas: una capa de entrada que recibe las variables predictoras, una o más capas ocultas que procesan la información, y una capa de salida que genera la predicción. Cada neurona en una capa está conectada con todas las neuronas de la capa siguiente, lo que se conoce como arquitectura *feedforward* o de propagación

hacia adelante [20].

Para una neurona j en una capa oculta, la salida se calcula como:

$$a_j = f\left(\sum_{i=1}^{n} w_{ij} x_i + b_j\right) \tag{2.14}$$

donde x_i son las entradas, w_{ij} son los pesos de conexión, b_j es el sesgo (bias) y $f(\cdot)$ es la función de activación. Las funciones de activación más comunes son ReLU (*Rectified Linear Unit*), definida como $f(x) = \text{máx}(0, x)$ [21], y tanh, definida como $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$.

El entrenamiento de una red neuronal consiste en ajustar los pesos w_{ij} y sesgos b_j para minimizar una función de pérdida. Para problemas de regresión, se utiliza típicamente el error cuadrático medio (MSE). El proceso de ajuste se realiza mediante el algoritmo de retropropagación (*backpropagation*), que calcula los gradientes de la función de pérdida respecto a cada peso mediante la regla de la cadena [22]. El algoritmo de optimización Adam (*Adaptive Moment Estimation*) es ampliamente utilizado por su eficiencia y capacidad de adaptar la tasa de aprendizaje durante el entrenamiento [23].

Los principales hiperparámetros de una red MLP incluyen:

- **Arquitectura**: Número de capas ocultas y neuronas por capa.
- **Función de activación**: ReLU, tanh, sigmoid, entre otras.
- **Tasa de aprendizaje** (*learning rate*): Controla el tamaño de los ajustes en cada iteración.
- **Tamaño de lote** (*batch size*): Número de observaciones procesadas antes de actualizar los pesos.
- **Dropout**: Técnica de regularización que desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para prevenir el sobreajuste [24].
- **Epochs**: Número de veces que el modelo ve el conjunto de entrenamiento completo.

A diferencia de otros modelos como Random Forest o XGBoost, no existe una fórmula analítica para determinar los hiperparámetros óptimos de una red neuronal. Por esta razón, se recurre a técnicas de búsqueda como Grid Search (búsqueda exhaustiva) o Random Search (búsqueda aleatoria) para encontrar la mejor configuración [25].

Early Stopping. El *early stopping* es una técnica de regularización que detiene el entrenamiento de un modelo iterativo cuando el error evaluado sobre un conjunto de validación deja de mejorar durante un número predefinido de iteraciones consecutivas, denominado *paciencia* [19]. Su objetivo es prevenir el sobreajuste al identificar el punto en el que el modelo comienza a memorizar el ruido del conjunto de entrenamiento en lugar de capturar patrones generalizables. Esta técnica es aplicable a cualquier algoritmo de entrenamiento iterativo; en el presente estudio se emplea tanto en XGBoost, donde determina el número óptimo de rondas de boosting (`nrounds`), como en las redes neuronales MLP,

donde controla el número de épocas de entrenamiento.

Validación Cruzada (*k-fold Cross-Validation*). La validación cruzada es una técnica de evaluación que permite estimar el desempeño de un modelo sin necesidad de un conjunto de validación adicional. El procedimiento consiste en dividir el conjunto de entrenamiento en k subconjuntos de tamaño similar, denominados *pliegues* (*folds*). En cada iteración, uno de los pliegues se reserva como conjunto de validación y los $k-1$ restantes se utilizan para entrenar el modelo. Este proceso se repite k veces, de modo que cada pliegue actúa exactamente una vez como conjunto de validación. El desempeño final se calcula como el promedio de las k evaluaciones, proporcionando una estimación más robusta que una única partición [16]. En el presente estudio se utiliza validación cruzada con $k = 5$ pliegues para la optimización de hiperparámetros en los modelos XGBoost, KNN y MLP.

2.1.4. Importancia de Variables en Modelos de Ensemble

En el contexto de modelos de aprendizaje automático, la *importancia de variables* (feature importance) cuantifica la contribución relativa de cada variable predictora al desempeño del modelo. A diferencia de la regresión lineal, donde los coeficientes $\hat{\beta}$ proporcionan una medida directa del efecto marginal de cada variable, los modelos de ensemble como Random Forest y XGBoost son inherentemente no lineales e involucran interacciones complejas entre predictores, lo que impide obtener coeficientes interpretables de forma directa. Las métricas de importancia de variables permiten identificar qué predictores son más relevantes para la predicción, facilitando la interpretación del modelo y la comprensión de los factores que impulsan el fenómeno modelado [10].

Para Random Forest, la métrica más utilizada es la *importancia por permutación* (%IncMSE), propuesta por Breiman [10]. Esta métrica evalúa la importancia de una variable x_j midiendo cuánto se deteriora el error de predicción cuando los valores de dicha variable son permutados aleatoriamente en las observaciones out-of-bag (OOB). Formalmente, para cada árbol T_b del bosque, se calcula el error OOB antes y después de permutar x_j:

$$\text{Imp}(x_j) = \frac{1}{B}\sum_{b=1}^{B}\left(\text{MSE}_{OOB,b}^{\pi_j} - \text{MSE}_{OOB,b}\right) \tag{2.15}$$

donde $\text{MSE}_{OOB,b}$ es el error cuadrático medio de las observaciones out-of-bag para el árbol b, y $\text{MSE}_{OOB,b}^{\pi_j}$ es el error calculado tras permutar aleatoriamente los valores de x_j en dichas observaciones. Si una variable es relevante para la predicción, su permutación incrementará sustancialmente el error; si es irrelevante, el error permanecerá prácticamente inalterado. Los valores se reportan como incremento porcentual (%IncMSE) normalizado respecto a la variable de mayor importancia [16].

Para XGBoost, la métrica principal de importancia es el *Gain*, que mide la reducción acumulada en la función de pérdida atribuible a cada variable a lo largo de todos los árboles del modelo. Cuando una

variable x_j es seleccionada como criterio de división en un nodo, la ganancia en ese nodo cuantifica la mejora en la predicción resultante de dicha división. El Gain total de una variable se obtiene sumando estas contribuciones sobre todos los nodos de todos los árboles donde la variable fue utilizada [14]:

$$\text{Gain}(x_j) = \sum_{t=1}^{K} \sum_{s \in S_{j,t}} \Delta L_s \tag{2.16}$$

donde K es el número total de árboles, $S_{j,t}$ es el conjunto de nodos del árbol t donde x_j fue utilizada como variable de división, y ΔL_s es la reducción en la función de pérdida en el nodo s. Los valores se normalizan como porcentaje del Gain máximo para facilitar la comparación entre variables.

Ambas métricas cuantifican la relevancia predictiva de las variables, pero desde perspectivas complementarias. La importancia por permutación (%IncMSE) mide el impacto en el error al eliminar la información contenida en una variable, mientras que el Gain mide cuánto contribuye una variable a la reducción del error durante la construcción del modelo. En la práctica, ambas métricas tienden a identificar las mismas variables como las más importantes, aunque pueden diferir en el ordenamiento relativo debido a sus diferentes fundamentos metodológicos [16].

2.1.5. Métricas de Desempeño

Para determinar cuál modelo de aprendizaje automático predice con mayor precisión el error entre el valor teórico del modelo Black-Scholes y el precio real de mercado, se comparará su desempeño mediante métricas estadísticas estandarizadas. Esta evaluación permitirá identificar el enfoque más eficaz para capturar las discrepancias entre teoría y práctica en la valoración de opciones. Las métricas que se usarán son:

- **Error Cuadrático Medio (RMSE - Root Mean Squared Error):** se define como la raíz cuadrada de la media de los cuadrados de todos los errores en las predicciones numéricas y sirve como métrica de error de propósito general. [26]

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

  y_i = Es el valor real del error de la opción i, calculado como (Precio de Mercado - Precio BS).
  $\hat{y}_i$ = Es el valor del error predicho por el modelo para la misma opción i.
  N = número de observaciones

- **Error Absoluto Medio (MAE - Mean Absolute Error):** calcula el promedio de las diferencias absolutas entre los valores reales y los valores predichos por un modelo. Es una medida de la magnitud promedio de los errores sin considerar su dirección y es robusta a valores atípicos. [27]

$$MSE_{(x,y)} = \frac{1}{N}\sum_{i=1}^{N}(x_i - y_i)^2$$

x_i = valor del error que fue predicho por tu modelo
y_i = valor real del error (Valor real - Valor predicho BS)
N = número de observaciones

- **Coeficiente de Determinación (R^2):** indica la proporción de la varianza de la variable dependiente que es predecible a partir de las variables independientes en un modelo de regresión. Su valor, entre 0 y 1, mide qué tan bien un modelo se ajusta a los datos; un R^2 de 1 (100) significa que el modelo explica toda la variabilidad, mientras que un R^2 de 0 (0 %) indica que no explica ninguna variabilidad. Cuanto más cercano a 1 sea el valor, mayor será el ajuste y la capacidad predictiva del modelo. [28]

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(x_i - \bar{y}_i)^2}$$

y_i = Es el valor real del error de la opción i, calculado como (Precio de Mercado - Precio BS).
$\hat{y}_i$ = Es el valor del error predicho por el modelo para la misma opción i.
$\bar{y}$ = Es el promedio de todos los errores reales en tu conjunto de datos.
N = número de observaciones

- **Mejora porcentual:** La Mejora Porcentual mide cuánto mejor (o peor) es el rendimiento de un modelo en comparación con un punto de referencia o *baseline*.

$$Mejora\% = \left(1 - \frac{RSME_{modelo}}{RSME_{baseline}}\right) \times 100$$

$RSME_{modelo}$ = RMSE de cada modelo al predecir el error.
$RSME_{baseline}$ = RMSE del modelo de referencia Black - Scholes

2.2. Antecedentes

En los últimos años, se ha intensificado el interés por mejorar los métodos tradicionales de valoración de derivados financieros mediante la incorporación de técnicas de aprendizaje automático. Esta tendencia responde a las limitaciones inherentes a modelos clásicos como Black-Scholes, los cuales, a pesar de su valor histórico y su sólida base matemática, presentan supuestos restrictivos como la constancia de la volatilidad y la eficiencia del mercado, que no siempre se cumplen en la realidad.

En [29], se realiza un análisis comparativo entre el modelo de Black-Scholes y algoritmos de aprendizaje automático, especialmente redes neuronales artificiales (ANNs), aplicados a la predicción de

precios de opciones. El estudio destaca que la proliferación del *big data* y el avance en la capacidad de cómputo han permitido a los modelos de machine learning captar patrones no lineales complejos que el modelo clásico no considera. Los resultados obtenidos muestran que, en muchos casos, las técnicas de aprendizaje automático pueden superar la precisión del modelo de Black-Scholes, proporcionando no solo un reemplazo, sino como una alternativa complementaria. Este enfoque busca integrar métodos de ciencia de datos para mejorar la estimación de precios de opciones, sin embargo, a diferencia del estudio mencionado, este proyecto no pretende sustituir el modelo Black-Scholes, sino predecir los errores que este comete y ajustar sus salidas. El análisis que los autores presentan sobre la *"Predicción de precios de opciones"* [29] representa un enfoque complementario y original, ya que mantiene la estructura teórica del modelo clásico, pero la enriquece mediante técnicas modernas de predicción, con el objetivo de lograr una valoración más precisa y útil para la toma de decisiones de inversión.

Por otro lado, el estudio desarrollado en [30] analiza la predictibilidad de los precios de acciones en el mercado bursátil chino utilizando datos de alta frecuencia y métodos estadísticos como las cadenas de Markov y modelos de núcleos de difusión. Aunque se centra en acciones y no en opciones, este trabajo proporciona una base sólida sobre cómo ciertas características del mercado, como la volatilidad y el precio promedio, afectan la precisión de los modelos predictivos. Esto resulta relevante para el presente proyecto, ya que tales variables podrían ser consideradas como explicativas en la predicción de errores del modelo Black-Scholes.

la revisión presentada en [31] ofrece una visión amplia sobre el uso de inteligencia artificial en la inversión bursátil. Este estudio, que analiza más de 2300 publicaciones entre 1995 y 2019, clasifica las investigaciones en categorías como optimización de portafolios, predicción del mercado y análisis de sentimiento, evidenciando una evolución hacia modelos cada vez más especializados y robustos. Aunque no aborda directamente la valoración de opciones ni el modelo Black-Scholes, este trabajo respalda la pertinencia de aplicar enfoques de inteligencia artificial en contextos financieros complejos como el que se aborda en este proyecto.

En el estudio [32] se tenía como objetivo principal realizar un análisis comparativo a gran escala de diversos métodos de machine learning aplicados al problema de fijación de precios de activos en el mercado bursátil al medir las primas de riesgo de las acciones (es decir, la predicción de los rendimientos futuros de los activos), buscando identificar qué métodos (lineales o no lineales) eran superiores y qué señales predictivas eran las más dominantes.

Lo modelos utilizados en el estudio recorrieron desde los más simples hasta los más complejos, Modelos Lineales: Regresión por Mínimos Cuadrados Ordinarios (OLS) y Modelos Lineales Penalizados (como Elastic Net, Lasso y Ridge); Reducción de Dimensión: Regresión de Componentes Principales (PCR) y Mínimos Cuadrados Parciales (PLS); Extensiones No Lineales: Modelos Lineales Generalizados

(GLM) con expansiones de spline; Modelos Basados en Árboles: Random Forests (Bosques Aleatorios) y Gradient Boosted Regression Trees (GBRT); Redes Neuronales: Múltiples arquitecturas de redes neuronales feed-forward (NN), variando la profundidad desde una (NN1) hasta cinco (NN5) capas ocultas.

Las métricas de Evaluación sobre el rendimiento de los modelos incorporaron tanto métricas estadísticas como económicas: Métrica Estadística: La métrica principal fue el R^2 fuera de muestra (out-of-sample R^2), que mide qué tan bien un modelo predice los rendimientos en comparación con un pronóstico simple de cero y utilizaron una versión modificada de la prueba de Diebold-Mariano para determinar si las diferencias en la precisión predictiva entre los modelos eran estadísticamente significativas. Adicionalmente, incorporaron Métricas Económicas para medir el valor económico real de los pronósticos, construyendo carteras de inversión.

Adicionalmente, establecen el modelo de Regresión Lineal (RL) como el benchmark o línea base de rendimiento, validando el desempeño de los modelos de machine learning utilizados sistemáticamente contra los resultados arrojados por la Regresión Lineal. El objetivo de esta comparación es cuantificar el valor añadido de los modelos más complejos. De esta manera, se evalúa si la complejidad y la capacidad predictiva de cada modelo logra, como mínimo, superar el desempeño del modelo lineal más básico. A lo largo de su artículo, utilizan este benchmark lineal para demostrar que las ganancias de rendimiento de los modelos no lineales, como las redes neuronales y los árboles, son estadísticamente significativas y no superficiales. Adoptamos, por lo tanto, la misma lógica para validar los hallazgos de este trabajo.

Finalmente, con esta metodología de benchmarking sobre la regresión lineal, los autores Gu, Kelly y Xiu lograron concluir que los mejores modelos para predecir precios de activos son los modelos no lineales, y en particular las redes neuronales (específicamente NN3, con tres capas ocultas donde alcanzó su punto máximo y luego disminuyó) y los árboles (Random Forest y GBRT). Además, concluyeron que su ventaja predictiva se atribuye a su capacidad para capturar interacciones no lineales complejas entre los predictores, algo que los modelos lineales o semi-lineales no logran hacer. Adicionalmente encontraron que para el caso de finanzas debido a la menor cantidad de datos y la baja relación señal-ruido en comparación con campos como la visión por computadora, el aprendizaje "poco profundo"(shallow learning) supera al "profundo"(deep learning)

En conjunto, estos antecedentes demuestran que el uso de técnicas de ciencia de datos puede enriquecer significativamente los modelos tradicionales de valoración financiera. El enfoque del presente proyecto —predecir y ajustar los errores del modelo de Black-Scholes mediante modelos clásicos de aprendizaje automático— se distingue de los trabajos anteriores al centrarse en la mejora del modelo teórico sin desechar su estructura, proponiendo una solución intermedia entre la teoría financiera clásica y los enfoques basados en datos.

Capítulo 3

Preparación de los datos

En este capítulo se muestran los resultados de las actividades para dar cumplimiento al objetivo 1, el cual consistió en diseñar y estructurar una base de datos con las variables del mercado financiero que impactan significativamente la discrepancia entre los valores reales y los precios teóricos de opciones europeas call y put según el modelo Black-Scholes, incorporando en su construcción un procedimiento de preprocesamiento de las variables, así como técnicas de normalización para optimizar el desempeño de los modelos de predicción de errores en la valoración financiera.

El resultado obtenido fue un conjunto de datos robusto que captura las dinámicas del mercado de opciones durante el período de análisis, compuesto por 40,337 contratos de opciones distribuidos equilibradamente entre opciones call y put, abarcando 10 activos representativos del mercado estadounidense y 78 fechas de vencimiento diferentes. Este conjunto de datos constituye la base fundamental sobre la cual se desarrollarán los modelos predictivos en las siguientes etapas del proyecto.

3.1. Identificación de fuentes de datos y recolección para modelación

La construcción de la base de datos fue el primer paso fundamental de éste proyecto de Ciencia de Datos. Se utilizó Yahoo Finance como fuente de información. Esta plataforma se seleccionó como fuente única debido al acceso a datos históricos públicos, cadenas completas de opciones en tiempo real, y disponibilidad gratuita a través de la API de R mediante el paquete quantmod.

Se seleccionaron 10 activos subyacentes (tickers) basados en tres criterios metodológicos para garantizar la calidad y representatividad de la muestra. Criterio 1: Alta Liquidez, asegurando un volumen de negociación significativo para que los precios de mercado fueran representativos. Criterio 2: Diversidad Sectorial, incorporando empresas de tecnología, índices de mercado y activos con diferentes perfiles de volatilidad para capturar distintas dinámicas de mercado. Criterio 3: Disponibilidad de Opciones, seleccionando activos con múltiples precios de ejercicio (strikes) y vencimientos. Con base

en esto, se seleccionaron 10 tickers del mercado estadounidense: Apple (AAPL), Microsoft (MSFT), Alphabet (GOOGL), Meta (META), Nvidia (NVDA), AMD, Tesla (TSLA), Amazon (AMZN), ETF del S&P 500 (SPY) y ETF del Nasdaq-100 (QQQ). La recolección de datos se realizó el 6 de octubre de 2025 mediante un proceso automatizado en R como se muestra en la Figura 3.1. El script procesa cada ticker secuencialmente, descargando precios históricos del activo subyacente y todas las cadenas de opciones disponibles.

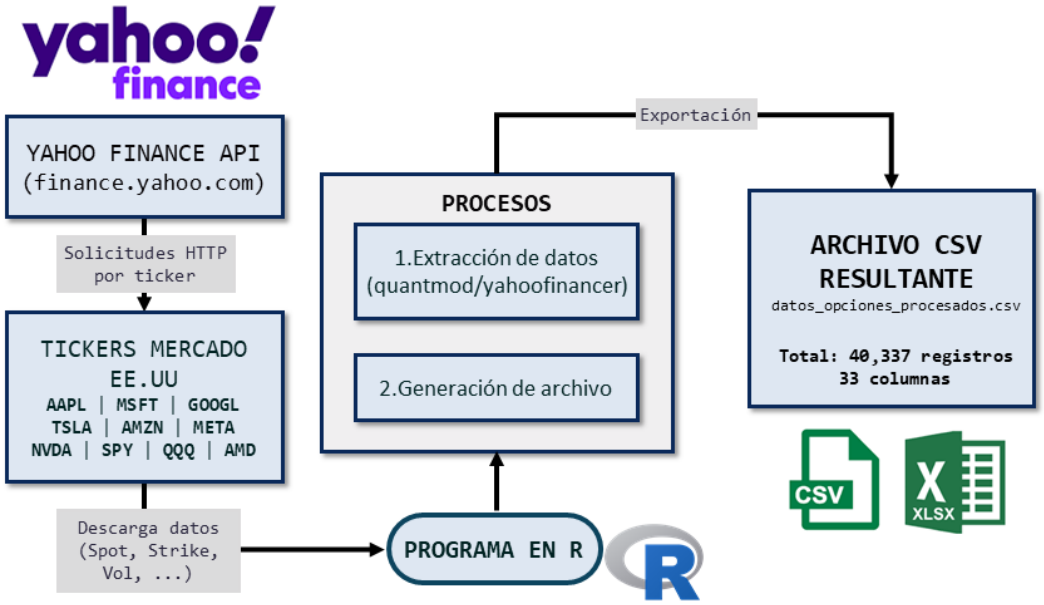


Figura 3.1: Proceso de extracción de datos financieros desde Yahoo Finance (R). *Fuente:* Elaboración propia.

El proceso resultó en 40,337 contratos de opciones distribuidos entre calls (20,756 contratos, 51.4 %) y puts (19,581 contratos, 48.6 %), abarcando 233 fechas de vencimiento desde octubre de 2025 hasta diciembre de 2026. La Tabla 3.1 presenta el resumen detallado por ticker, incluyendo volatilidad histórica calculada con rendimientos diarios de los últimos 252 días. Esta métrica revela diferencias notables: desde SPY con 19.45 % hasta TSLA con 70.98 %, permitiendo evaluar el comportamiento del modelo Black-Scholes bajo diferentes condiciones de mercado.

Tabla 3.1: Resumen de datos recolectados por *ticker*. *Fuente:* Elaboración propia.

Ticker	Precio Spot ($)	Volatilidad (%)	N° Calls	N° Puts	Vencimientos
AAPL	258.02	32.38	1,087	1,012	21
MSFT	517.35	24.64	1,471	1,365	21
GOOGL	245.35	32.13	1,068	1,003	21
TSLA	429.83	70.98	2,546	2,296	22
AMZN	219.51	33.53	994	937	21
META	710.56	36.56	2,454	2,209	21

Continúa en la siguiente página...

Tabla 3.1 – continuación

Ticker	Precio Spot ($)	Volatilidad (%)	N° Calls	N° Puts	Vencimientos
NVDA	187.62	49.49	2,625	2,412	21
SPY	669.21	19.45	3,919	3,991	32
QQQ	603.18	23.40	3,554	3,411	32
AMD	164.67	51.41	1,038	945	21
Total	–	–	20,756	19,581	233

3.2. Identificación de las variables relevantes para el análisis

Una vez establecida la fuente de datos y completada la recolección inicial, se procedió a identificar y estructurar las variables necesarias para abordar el problema de investigación. El proceso de identificación de variables busco organizar el conjunto de datos en cinco grupos: *las Variables Fundamentales del Modelo Black-Scholes, las Variables Proporcionadas por el Mercado de Opciones, el Diseño de Características Adicionales, las Variables Calculadas y Objetivo*, y *las Variables de Identificación*. Este enfoque de clasificación permitió asegurar que el dataset capturara tanto las condiciones teóricas como las variables suministradas por la API de yahoo finance. Estas variables fueron fundamentales para realizar nuestro proyecto de investigación.

3.2.1. Variables de Identificación

Este grupo se compone del **Ticker**, que es el identificador del activo subyacente y es crucial para segmentar el análisis por distintas dinámicas de mercado, y del **Tipo Opción** (*Call* o *Put*), el cual es indispensable para diferenciar y modelar los errores sistemáticos de valoración, dado que el modelo *Black-Scholes* presenta asimetrías de error notables entre ambos tipos de contratos.

3.2.2. Variables fundamentales del modelo Black-Scholes

El modelo Black-Scholes, en su formulación original, requiere exactamente cinco parámetros de entrada para calcular el precio teórico de una opción europea. Estas variables constituyen el núcleo fundamental del análisis y son indispensables para la implementación del modelo.

La primera variable es el **precio spot del activo subyacente** (S_0), obtenido directamente de Yahoo Finance como el precio de cierre ajustado en la fecha de análisis. Para Apple (AAPL), este valor fue de $258.02, mientras que para Tesla (TSLA) fue de $429.83.

La segunda variable fundamental es el **precio de ejercicio o strike** (K), extraído directamente de las cadenas de opciones disponibles. Para cada activo subyacente, se identificaron strikes que abarcaban un amplio espectro, en el caso de Apple, con un precio spot de $258.02, se encontraron opciones con

strikes que oscilaban desde $210 hasta valores superiores a $250.

El **tiempo hasta el vencimiento** (T) constituyó la tercera variable. Se calculó como la diferencia temporal entre la fecha de vencimiento de cada opción y la fecha de análisis (6 de octubre de 2025), expresada en años mediante la división por 365. En el dataset resultante, este parámetro presentó un rango desde 0 días ($T = 0$ años) para las opciones que vencían el mismo día de la extracción, hasta 837 días ($T = 2{,}29$ años) para las opciones de mayor plazo. Como se observa en la Figura 3.2, donde el eje horizontal representa los días hasta el vencimiento y el eje vertical muestra la frecuencia, las distribuciones para opciones *call* y *put* resultan prácticamente idénticas. Las opciones *call* presentan una media de 215 días (línea azul punteada, $SD = 233$ días), mientras que las put exhiben una media de 209 días (línea roja punteada, $SD = 230$ días). Esta similitud en las distribuciones de vencimientos—con una diferencia de apenas 6 días entre medias y medianas idénticas de 102 días—refleja que las cadenas de opciones ofrecen estructura temporal equivalente para ambos tipos de contratos. La distribución multimodal observada, con picos pronunciados en horizontes específicos, corresponde a los vencimientos estándar del mercado de opciones, concentrándose la mayor liquidez en horizontes de corto y mediano plazo (0-300 días).

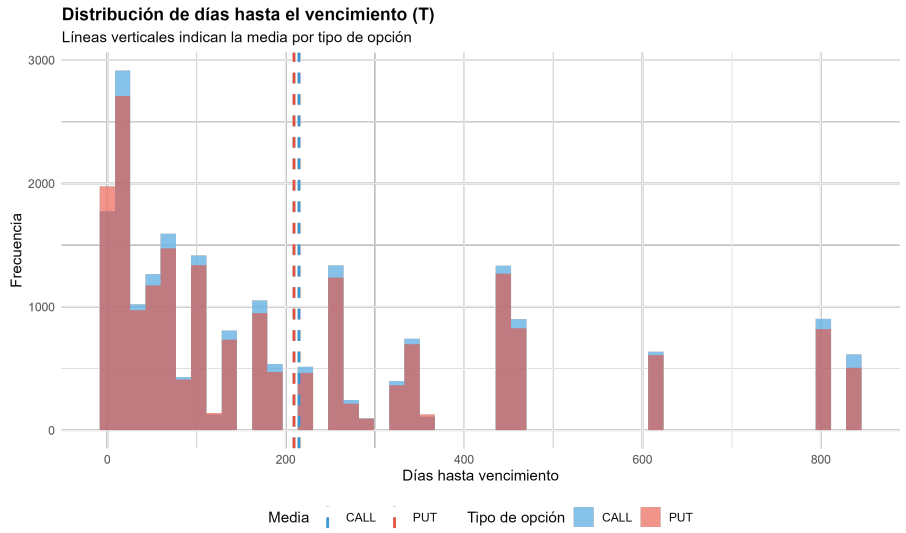


Figura 3.2: Distribución de días hasta el vencimiento (T). *Fuente:* Elaboración propia.

La **tasa libre de riesgo** (r) se obtuvo del índice `^IRX` de *Yahoo Finance*, que representa el rendimiento de los *Treasury Bills* (bonos del tesoro de EEUU) a 13 semanas. En la fecha de análisis, esta tasa se situó en 3.85 % anual. Se empleó una tasa única para todas las opciones, dado que las variaciones en la curva de rendimientos de corto plazo resultan insignificantes para el rango temporal analizado.

Finalmente, la **volatilidad histórica** (σ) se calculó mediante la desviación estándar de los rendi-

mientos logarítmicos diarios del último año de negociación (252 días hábiles), anualizada mediante multiplicación por $\sqrt{252}$. Como se observa en la Figura 3.3, donde el eje horizontal representa la volatilidad histórica anualizada en escala decimal (0.2 = 20 %) y el eje vertical muestra la frecuencia de contratos, la distribución de volatilidades presenta una estructura multimodal (múltiples picos) con una media de 36 % (línea roja punteada en 0.36) y desviación estándar de 16.5 %. Los picos observados corresponden a agrupaciones naturales de activos según su perfil de riesgo, extendiéndose desde instrumentos conservadores como SPY (19.45 %) hasta acciones de alta volatilidad como Tesla (70.98 %). Esta variabilidad resultó fundamental para evaluar el comportamiento del modelo *Black-Scholes* bajo diferentes condiciones de mercado. La volatilidad se explica por activo subyacente y no por tipo de opción.

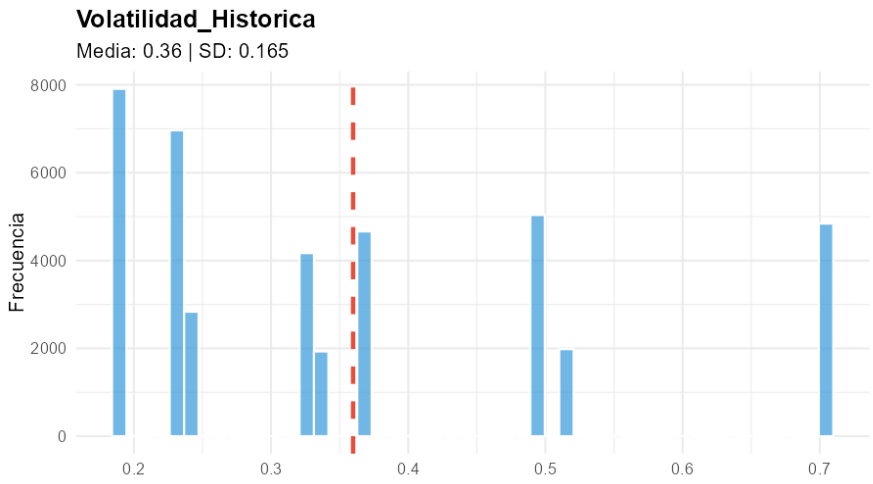


Figura 3.3: Distribución de volatilidad histórica (σ). *Fuente:* Elaboración propia.

3.2.3. Variables proporcionadas por el mercado de opciones

Más allá de los parámetros estrictamente necesarios para Black-Scholes, el mercado de opciones proporciona información adicional que se decidió incorporar al análisis.

La variable más importante en esta categoría es el **precio de mercado de la opción** (`Last`), que representa el último precio al cual se negoció el contrato. Esta variable constituye el *ground truth* del análisis, es decir, el valor real contra el cual se contrastan las predicciones teóricas. Sin este precio observado, no sería posible calcular el error del modelo.

La **volatilidad implícita** (IV) representa las expectativas del mercado sobre la volatilidad futura. *Yahoo Finance* la determina mediante inversión numérica de la fórmula de *Black-Scholes*: dado el precio de mercado observado, se calcula qué volatilidad habría que emplear para que el modelo reproduzca dicho precio. Como se observa en la Figura 3.4, donde el eje horizontal representa la volatilidad

implícita en escala decimal y el eje vertical muestra la frecuencia de contratos, la distribución presenta un marcado sesgo a la derecha con diferencias entre tipos de opciones. Las opciones *call* exhiben una media de 50.9 % (línea azul punteada en 0.509, $SD = 69{,}2\,\%$), mientras que las *put* muestran una media de 45.2 % (línea roja punteada en 0.452, $SD = 37{,}7\,\%$). Esta diferencia de 5.7 puntos porcentuales entre medias, junto con la sustancialmente mayor dispersión observada en las *call* (*SD* casi duplicada), refleja heterogeneidad en la composición del dataset y mayor presencia de valores extremos en opciones *call*. La IV promedio global de 48.1 % resultó significativamente superior a la volatilidad histórica promedio de 36 %, con una diferencia de 12.1 puntos porcentuales, indicando que el mercado anticipa mayor volatilidad que la reflejada en datos históricos.

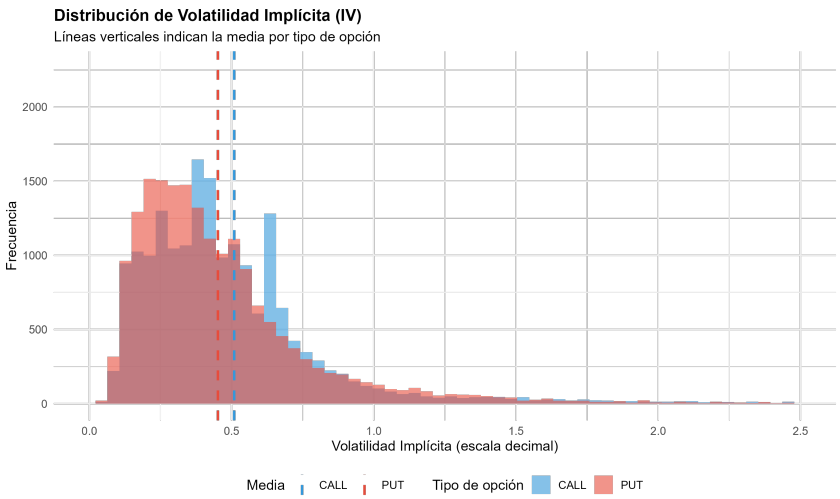


Figura 3.4: Distribución de volatilidad implícita (IV) por tipo de opción. *Fuente:* Elaboración propia.

Adicionalmente, se capturaron el **volumen de negociación** (`Vol`) y el **Open Interest** (OI). El volumen cuantifica el número de contratos negociados en la sesión, mientras que el *Open Interest* contabiliza el total de contratos abiertos sin liquidar. Estos indicadores de liquidez resultarían posteriormente útiles para identificar opciones con precios potencialmente poco confiables debido a baja actividad de negociación.

3.2.4. Variables calculadas y variable objetivo

Una vez recolectadas las variables fundamentales y de mercado, se procedió a calcular el **precio teórico de Black-Scholes** (`Precio_BS`) para cada opción utilizando la ecuacion estándar (2.1) y (2.2) para *call* y *put* respectivamente, y el calculo de la variable central de este analisis **Diferencia** , definida como el error del modelo Black-Scholes. Las cuales se desarrollaran a continuacion.

3.2.4.1. Cálculo del precio teórico: `Precio_BS`

La implementación del modelo *Black-Scholes* se realizó en el lenguaje R mediante una función que acepta los cinco parámetros requeridos y el tipo de opción (*call* o *put*) como argumentos. La función retorna el precio teórico calculado. La implementación utiliza la función `pnorm()` de R para evaluar la distribución normal acumulada, garantizando precisión numérica mediante algoritmos optimizados, La lógica general del procedimiento puede observarse en el diagrama de flujo mostrado en la Figura 3.5.

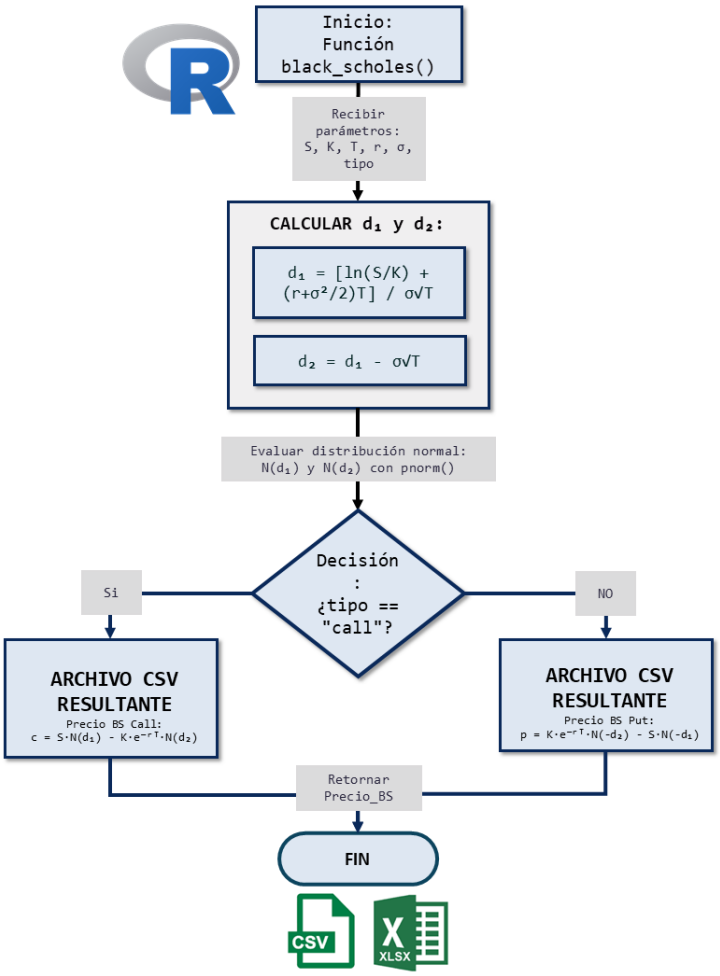


Figura 3.5: Cálculo del precio teórico Black–Scholes en R. *Fuente:* Elaboración propia.

Ejemplo ilustrativo de cálculo

Para ilustrar concretamente el proceso de cálculo, se presenta un ejemplo con una opción *call* sobre AAPL (Apple). La Tabla 3.2 detalla los parámetros de entrada y los resultados del cálculo paso a paso.

Tabla 3.2: Ejemplo de cálculo Black-Scholes: Opción CALL sobre AAPL. *Fuente:* Elaboración propia.

Parámetro	**Valor**
Parámetros de entrada:	
S (Precio spot)	\$258.02
K (Strike)	\$260.00
T (Tiempo)	32 días = 0.0877 años
r (Tasa libre riesgo)	3.858 %
σ (Volatilidad histórica)	32.379 %
Cálculos intermedios:	
d_1	0.0035
d_2	−0,0924
$N(d_1)$	0.5014
$N(d_2)$	0.4632
Resultados:	
Precio teórico BS	\$9.34
Precio de mercado observado	\$7.70
Diferencia	−\$1.64

Los valores intermedios ilustran la mecánica del modelo: d_1 ligeramente positivo (0.0035) indica que la probabilidad de ejercicio es apenas superior al 50 %, mientras que d_2 negativo (−0,0924) refleja el ajuste por volatilidad. Las probabilidades acumuladas $N(d_1) = 0{,}5014$ y $N(d_2) = 0{,}4632$ se combinan ponderadamente con los componentes spot y valor presente del *strike* para producir el precio teórico final. El procedimiento de cálculo descrito se aplicó sistemáticamente a cada una de las 40,337 opciones del dataset original. Para cada observación, se extrajeron los cinco parámetros requeridos (Strike, Precio_Actual, Dias_Vencimiento convertido a años, Volatilidad_Historica, Tasa_Libre_Riesgo) y se invocó la función Black-Scholes con el tipo de opción correspondiente. Los resultados se almacenaron en una nueva columna denominada `Precio_BS`. La Tabla 3.3 presenta estadísticas descriptivas de los precios teóricos calculados, segmentadas por tipo de opción.

Tabla 3.3: Estadísticas de precios teóricos Black-Scholes por tipo de opción. *Fuente:* Elaboración propia.

Tipo	**N**	**Media**	**Mediana**	**SD**	**Max**
CALL	20,756	\$93.85	\$51.97	\$113.17	\$705.97
PUT	19,581	\$60.14	\$5.28	\$174.38	\$2,186.64

3.2.4.2. Cálculo de la variable objetivo: `Diferencia`

Una vez calculados los precios teóricos Black-Scholes para cada opción del dataset, se procedió a definir y analizar formalmente la variable objetivo que guiará todo el análisis de machine learning subsecuente. Esta variable, denominada `Diferencia`, captura la discrepancia entre el precio observado en el mercado y el precio teórico calculado por el modelo. La variable objetivo se define matemáticamente como:

$$\text{Diferencia} = \text{Precio\_Real} - \text{Precio\_BS} = \text{Last} - \text{Precio\_BS} \tag{3.1}$$

Donde `Last` representa el último precio de negociación observado en el mercado, y `Precio_BS` corresponde al precio teórico calculado mediante la fórmula de Black-Scholes. Esta diferencia constituye el error de valoración del modelo, y su signo tiene interpretación económica directa: valores positivos indican que el mercado valora la opción por encima de la predicción teórica (subvaloración por parte del modelo), mientras que valores negativos señalan que el mercado valora la opción por debajo de la predicción (sobrevaloración por parte del modelo).

El objetivo fundamental de este estudio consiste en predecir esta variable `Diferencia` utilizando técnicas de machine learning, permitiendo así generar correcciones sistemáticas al modelo Black-Scholes que reduzcan el error de valoración. El análisis estadístico por tipo de opción revela patrones contrastantes documentados en la Tabla 3.4. Las opciones call presentan sesgo sistemático positivo con media de $10.65, indicando que Black-Scholes tiende a subvalorarlas (el mercado paga más de lo que predice el modelo). En contraste, las opciones put exhiben sesgo negativo con media de -$16.23, evidenciando que el modelo tiende a sobrevalorarlas (el mercado paga menos de lo previsto). Aunque las medianas de ambos tipos son cercanas a cero (-$0.39 para call, $0.13 para put), indicando que la mayoría de errores individuales son pequeños, las colas de las distribuciones difieren sustancialmente: las *call* presentan errores positivos extremos (máximo $935), mientras que las *put* muestran errores negativos extremos (mínimo -$987). Las opciones *put* exhiben además mayor dispersión absoluta (SD = $102 vs $89.0 para *call*) y menor concentración en el rango central (IQR = $3.91 vs $7.52). Los coeficientes de skewness con signos opuestos (6.01 para call, -6.19 para *put*) y valores de kurtosis excesivos (42.4 y 39.9 respectivamente) confirman distribuciones altamente no normales con colas pesadas en direcciones asimétricas. Esta divergencia fundamental entre tipos de opciones sugiere que los supuestos de *Black-Scholes* –particularmente la simetría implícita en la distribución log-normal de rendimientos– no se sostienen en la práctica del mercado real.

Tabla 3.4: Estadísticas descriptivas de la variable objetivo Diferencia por tipo de opción. *Fuente:* Elaboración propia.

Estadístico	CALL	PUT
N observaciones	20,756	19,581
Media	$10.65	−$16.23

Tabla 3.4 – continuación

Estadístico	CALL	PUT
Mediana	−$0.39	$0.13
Desviación estándar	$89.0	$102
RMSE	$89.6	$103
Mínimo	−$380	−$987
Cuartil 1 (Q1)	−$5.25	−$2.25
Cuartil 3 (Q3)	$2.27	$1.66
Máximo	$935	$243
Rango intercuartílico (IQR)	$7.52	$3.91
Skewness (asimetría)	6.01	−6.19
Kurtosis (exceso)	42.4	39.9

La Figura 3.6 visualiza estas diferencias mediante histogramas superpuestos, donde el eje horizontal representa el error en dólares y el eje vertical muestra la frecuencia de contratos. La línea negra punteada vertical marca el error cero (valoración perfecta), mientras que las líneas de color indican las medias por tipo: azul para *call* ($10.65, a la derecha de cero) y roja para *put* (-$16.23, a la izquierda de cero). Ambas distribuciones muestran alta concentración cerca del error cero, reflejando que la mayoría de opciones presentan errores de valoración pequeños. Sin embargo, las colas extendidas en direcciones opuestas evidencian los sesgos sistemáticos: la distribución azul (*call*) se extiende más hacia la derecha, capturando casos donde *Black-Scholes* subestima significativamente; la distribución roja (*put*) se extiende más hacia la izquierda, capturando casos de sobrevaloración extrema. Esta visualización confirma que el comportamiento del modelo Black-Scholes difiere fundamentalmente según el tipo de opción, justificando plenamente la necesidad de analizar *call* y *put* por separado en todas las etapas subsecuentes del estudio.

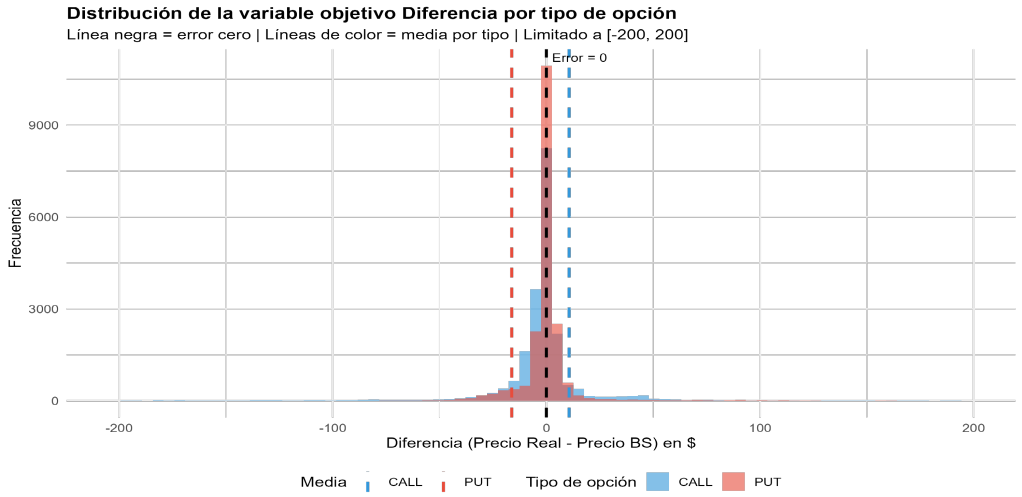


Figura 3.6: Distribución de la variable objetivo Diferencia por tipo de opción. *Fuente:* Elaboración propia.

3.2.5. Diseño de características adicionales

Reconociendo que las relaciones relevantes entre variables frecuentemente presentan naturaleza no lineal y compleja, se procedió a crear variables derivadas mediante técnicas de ingeniería de características (*feature engineering*) con el objetivo de capturar patrones más sutiles.

La primera variable derivada fue el **Moneyness**, calculado como el ratio S/K (precio spot dividido por strike). Un valor de *moneyness* igual a 1.0 indica que la opción se encuentra exactamente *at-the-money* (ATM); para opciones call, valores superiores a 1.0 señalan posiciones *in-the-money* (ITM) mientras que valores inferiores indican posiciones *out-of-the-money* (OTM); para opciones put, esta relación se invierte.

Como se observa en la Figura 3.7, donde el eje horizontal representa el ratio S/K y el eje vertical muestra la frecuencia de contratos, la distribución presenta diferencias notables entre tipos de opciones. Las opciones *call* exhiben una mediana de 1.05 (línea azul punteada), correspondiendo a posiciones ligeramente *in-the-money*, mientras que las put muestran una mediana de 1.12 (línea roja punteada), indicando posiciones predominantemente *out-of-the-money*. Al clasificar las opciones según umbrales estándar (OTM si $S/K < 0{,}95$, ATM si $0{,}95 \leq S/K \leq 1{,}05$, ITM si $S/K > 1{,}05$ para call; criterios invertidos para put), se observa que el 49.4 % de las call son ITM comparado con solo 24.0 % de las put, mientras que el 59.6 % de las put son OTM frente a 34.5 % de las call. Esta composición asimétrica del dataset refleja las condiciones de mercado prevalecientes durante el período de recolección.

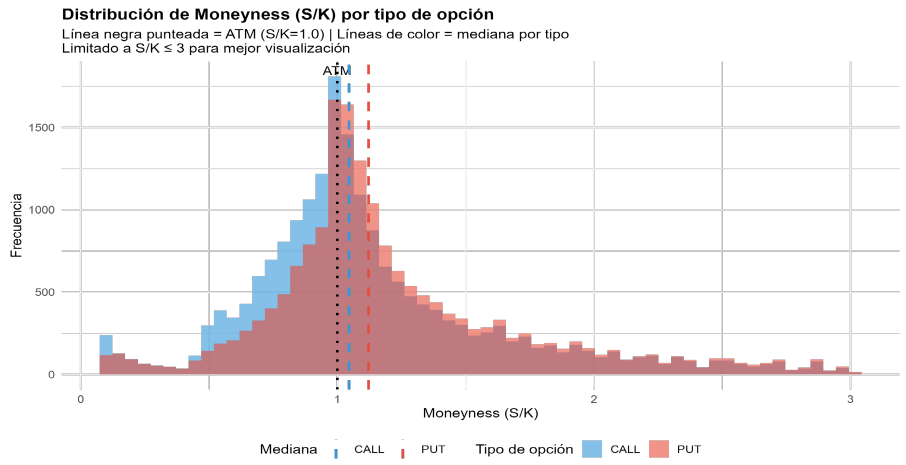


Figura 3.7: Distribución de Moneyness (S/K) por tipo de opción. *Fuente:* Elaboración propia.

En la Figura 3.8 se presenta el análisis de errores de Black-Scholes clasificando las opciones en tres grupos según moneyness (OTM con $S/K < 0{,}95$, ATM entre 0,95–1,05, e ITM con $S/K > 1{,}05$, aplicando criterios invertidos para opciones put). El eje vertical representa el error en dólares (Precio Real - Precio BS), mientras que el eje horizontal muestra los tres niveles de moneyness separados por tipo de opción. Los diamantes blancos indican la media de cada grupo, y la línea roja punteada horizontal marca el

error cero. El análisis cuantitativo revela patrones contrastantes entre tipos de opciones. Para opciones *call*, las ATM presentan el mejor desempeño (RMSE = $7.30), seguidas por ITM (RMSE = $33.2), mientras que las OTM exhiben el peor comportamiento (RMSE = $147) con error medio positivo de $38.5, indicando que *Black-Scholes* subestima sistemáticamente estas opciones. En contraste, para opciones *put*, las ATM también muestran el mejor desempeño (RMSE = $5.85), las OTM presentan excelente precisión (RMSE = $6.35, error medio de $1.06), pero las ITM exhiben errores extremos (RMSE = $210, error medio de -$68.7), evidenciando que *Black-Scholes* sobreestima dramáticamente estas posiciones.

Esta asimetría en los patrones de error entre *call* y *put* revela una limitación fundamental del modelo: la asunción de volatilidad constante a través de strikes y tipos de opciones resulta inadecuada para capturar la complejidad del mercado real.

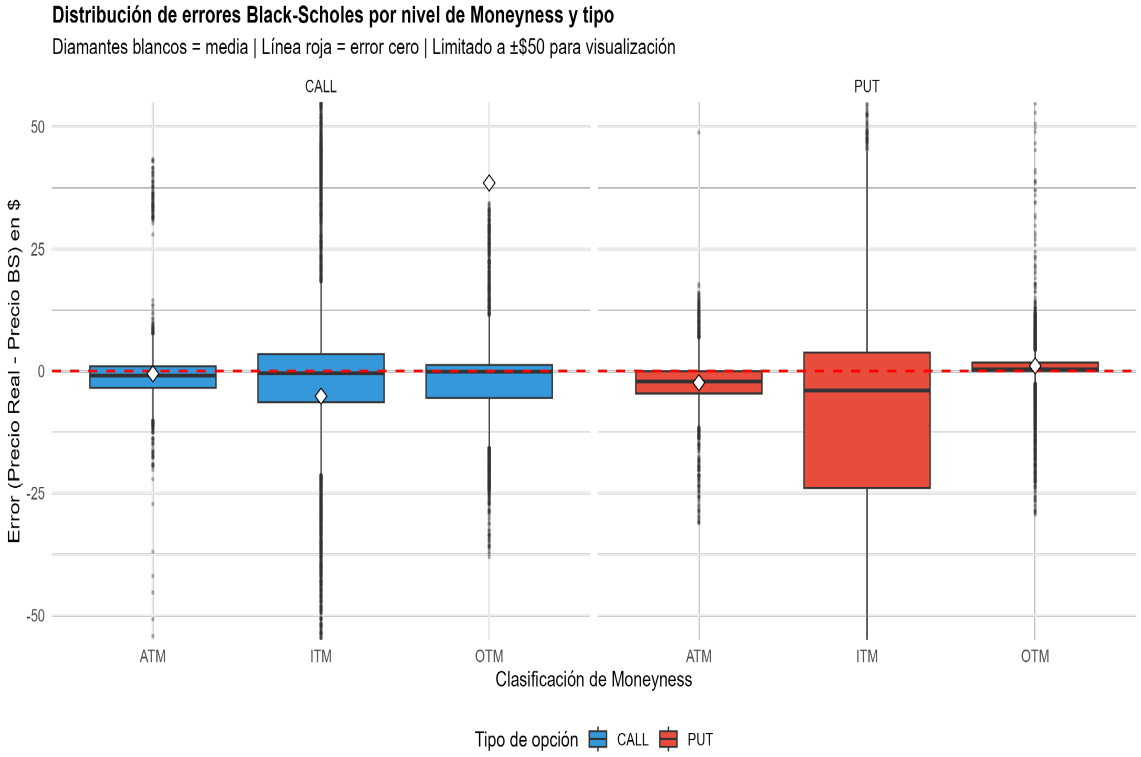


Figura 3.8: Distribución de errores Black-Scholes por nivel de Moneyness y tipo de opción. *Fuente:* Elaboración propia.

También se generó la variable **Log_Moneyness**, definida como $\ln(S/K)$. Esta transformación logarítmica se justifica teóricamente dado que *Black-Scholes* asume que los precios de los activos siguen una distribución log-normal en lugar de normal. Al emplear logaritmos, se trabaja en el espacio que el modelo considera matemáticamente relevante.

Como se observa en la Figura 3.9, las opciones *call* exhiben una mediana de 0.0446 (línea azul

punteada), mientras que las *put* muestran una mediana de 0.115 (línea roja punteada), reflejando nuevamente la mayor prevalencia de posiciones out-of-the-money en opciones put. Valores positivos de $\ln(S/K)$ indican que $S > K$, correspondiendo a *call in-the-money* o *put out-of-the-money*; valores negativos indican la situación inversa. Esta transformación captura de manera más simétrica las distancias relativas entre precio *Spot* y *Strike*.

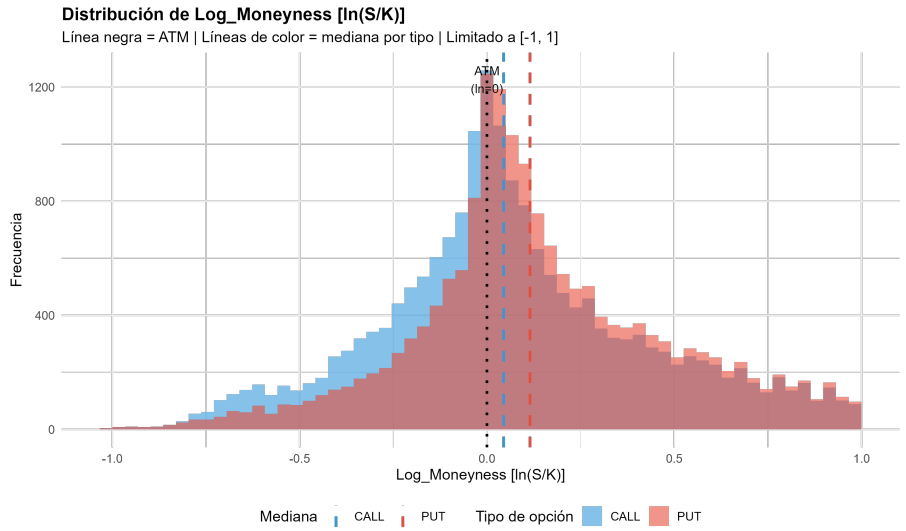


Figura 3.9: Distribución de Log_Moneyness $[\ln(S/K)]$ por tipo de opción. *Fuente:* Elaboración propia.

La variable **Vol_Diff** se definió como la diferencia entre la volatilidad implícita y la volatilidad histórica (IV $-\sigma$). Esta variable captura un aspecto fundamental: la discrepancia entre las expectativas del mercado sobre volatilidad futura (IV) y lo que sugieren los datos históricos recientes (σ). Como se observa en la Figura 3.10, donde el eje horizontal representa la diferencia de volatilidades en escala decimal y el eje vertical muestra la frecuencia de contratos, la distribución presenta un marcado sesgo a la derecha con diferencias notables entre tipos de opciones. Las opciones *call* exhiben una media de 0.147 (14.7 %, línea azul punteada) con desviación estándar de 0.665, mientras que las *put* muestran una media de 0.0949 (9.49 %, línea roja punteada) con desviación estándar de 0.341. Esta diferencia de 5.2 puntos porcentuales indica que las opciones call presentan mayor discrepancia entre volatilidad implícita e histórica, junto con sustancialmente mayor dispersión (*SD* casi duplicada). La concentración cerca de cero indica que para la mayoría de opciones, la volatilidad implícita y la histórica son relativamente similares. Sin embargo, las colas extensas hacia la derecha revelan casos donde el mercado anticipa volatilidad sustancialmente mayor que la observada históricamente, situación que resultará particularmente relevante para entender los errores sistemáticos del modelo *Black-Scholes* en el análisis posterior.

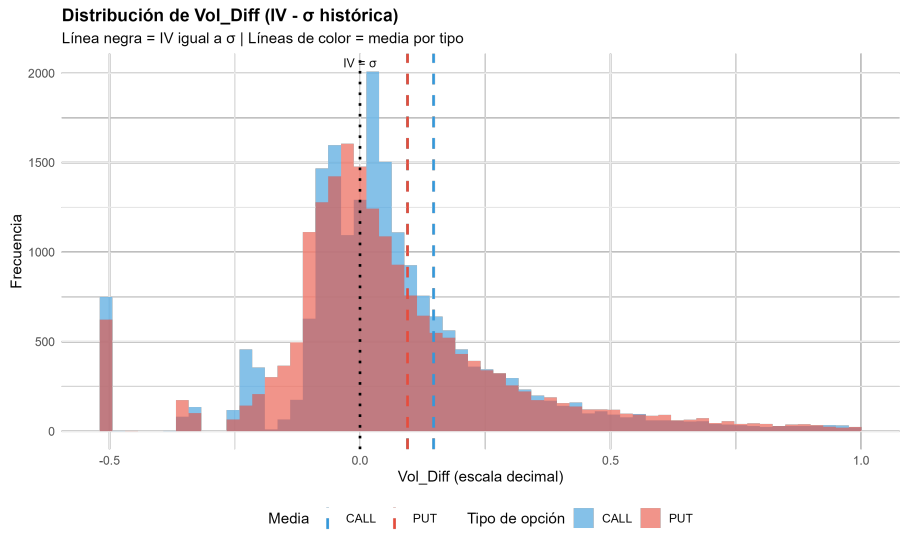


Figura 3.10: Distribución de Vol_Diff (IV - Volatilidad Histórica) por tipo de opción. *Fuente:* Elaboración propia.

Se creó también la variable **Strike_Normalizado** como $(K - S)/S$, que expresa la distancia del *Strike* respecto al precio *Spot* como porcentaje. Esta normalización permite comparaciones consistentes entre activos: un strike de \$230 para Apple (\$258.02) resulta comparable con un strike de \$250 para Tesla (\$429.83) en términos relativos. Como se observa en la Figura 3.11, las opciones *call* presentan una mediana de -0.0436 (línea azul punteada), indicando que el *Strike* típico se encuentra 4.36 % por debajo del precio *Spot*, mientras que las *put* muestran una mediana de -0.109 (línea roja punteada), con strikes 10.9 % por debajo del *spot*. Valores negativos indican *Strikes* inferiores al precio actual ($K < S$); valores positivos señalan *Strikes* superiores ($K > S$).

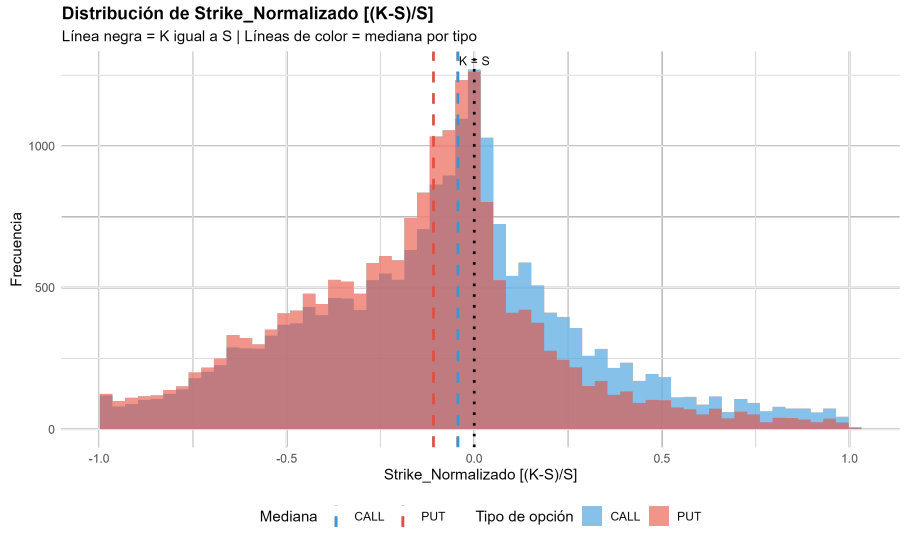


Figura 3.11: Distribución de Strike_Normalizado $[(K - S)/S]$ por tipo de opción. *Fuente:* Elaboración propia.

Finalmente, se incorporó la variable binaria **In_The_Money**: toma valor 1 si la opción está dentro del

dinero, 0 en caso contrario. Para opciones *call*, esto ocurre cuando $S > K$; para *put*, cuando $S < K$. El análisis del dataset reveló que el 57.4 % de las opciones *call* se encontraban in-the-money comparado con solo 31.3 % de las *put*, reflejando la composición asimétrica del mercado durante el período de recolección. Esta variable categórica permite a los modelos diferenciar entre opciones con valor intrínseco versus aquellas cuyo valor es exclusivamente temporal.

3.3. Análisis exploratorio de datos

El dataset original, producto de la recolección descrita en la sección anterior, contenía 40,337 observaciones correspondientes a contratos de opciones sobre 10 activos subyacentes. El primer paso consistió en realizar un diagnóstico exhaustivo de la calidad de estos datos para identificar potenciales problemas.

Identificación de valores faltantes

El análisis inicial reveló la presencia de valores faltantes en cinco variables del dataset. La Tabla 3.5 presenta el detalle de estas variables.

Tabla 3.5: Variables con valores faltantes en el dataset original. *Fuente:* Elaboración propia.

Variable	N Missing	N Valid	Total	% Missing
Vol (Volumen)	1,834	38,503	40,337	4.55 %
ChgPct (Cambio %)	32	40,305	40,337	0.08 %
OI (Open Interest)	14	40,323	40,337	0.03 %
Bid (Precio de compra)	1	40,336	40,337	0.00 %
Ask (Precio de venta)	1	40,336	40,337	0.00 %

Como se observa en la Tabla 3.5 y en la visualización de la Figura 3.12, la variable más afectada fue el volumen de negociación (Vol), con 4.55 % de valores faltantes. Este patrón resulta comprensible desde una perspectiva de microestructura de mercado: opciones con strikes muy alejados del precio spot o con vencimientos lejanos frecuentemente presentan baja o nula actividad de negociación en determinadas sesiones, resultando en valores faltantes para el volumen.

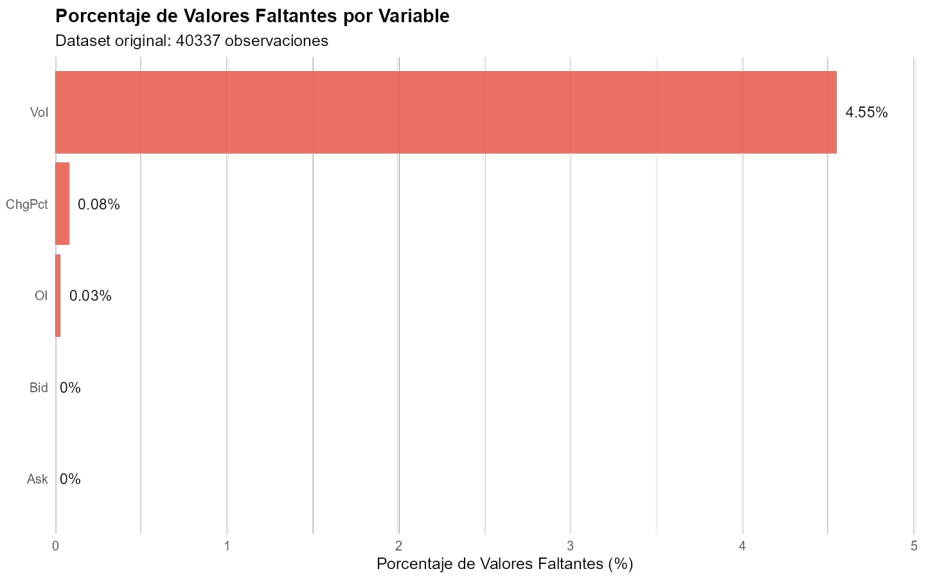


Figura 3.12: Distribución porcentual de valores faltantes por variable.

Detección de valores atípicos

Se procedió a identificar observaciones con valores atípicos (*outliers*) que pudieran distorsionar el análisis. Se analizaron sistemáticamente cinco variables clave desagregadas por tipo de opción: volatilidad implícita (IV), error de Black-Scholes (Diferencia), precio de mercado (Last), diferencia de volatilidades (Vol_Diff) y volatilidad histórica. La Tabla 3.6 presenta las estadísticas de outliers utilizando el criterio del rango intercuartílico (IQR) con factor 1.5.

Tabla 3.6: Estadísticas de outliers por variable y tipo de opción (criterio IQR). *Fuente:* Elaboración propia.

Variable	Tipo	N	Media	SD	% Outliers
IV	CALL	20,756	0.509	0.692	5.13 %
IV	PUT	19,581	0.452	0.377	5.82 %
Diferencia	CALL	20,756	$10.65	$89.0	19.0 %
Diferencia	PUT	19,581	−$16.23	$102	19.4 %
Last	CALL	20,756	$104.50	$129.33	5.86 %
Last	PUT	19,581	$43.91	$102.59	13.4 %
Vol_Diff	CALL	20,756	0.147	0.665	13.8 %
Vol_Diff	PUT	19,581	0.095	0.341	10.8 %
Volatilidad Histórica	CALL	20,756	0.362	0.166	0.00 %
Volatilidad Histórica	PUT	19,581	0.357	0.165	0.00 %

El análisis revela patrones diferenciados entre tipos de opciones. Para la volatilidad implícita, las opciones *put* presentan ligeramente mayor proporción de outliers (5.82 % vs 5.13 %), aunque las

medias son similares. Más notable resulta la diferencia en precios de mercado (Last), donde las *put* exhiben más del doble de proporción de outliers que las *call* (13.4 % vs 5.86 %), sugiriendo mayor heterogeneidad en la estructura de precios de opciones *put*. Para la variable objetivo (Diferencia), ambos tipos presentan proporciones similares de outliers (19.0 % y 19.4 %), reflejando que los errores extremos de *Black-Scholes* afectan ambos tipos de contratos de manera comparable. La volatilidad histórica no presenta outliers en ningún tipo, confirmando la estabilidad de esta métrica calculada.

Como se observa en la Figura 3.13, que presenta un análisis comparativo mediante z-scores normalizados, cada variable exhibe patrones distintos de valores atípicos. Sin embargo, se estableció un criterio selectivo de eliminación basado en la naturaleza de cada variable.

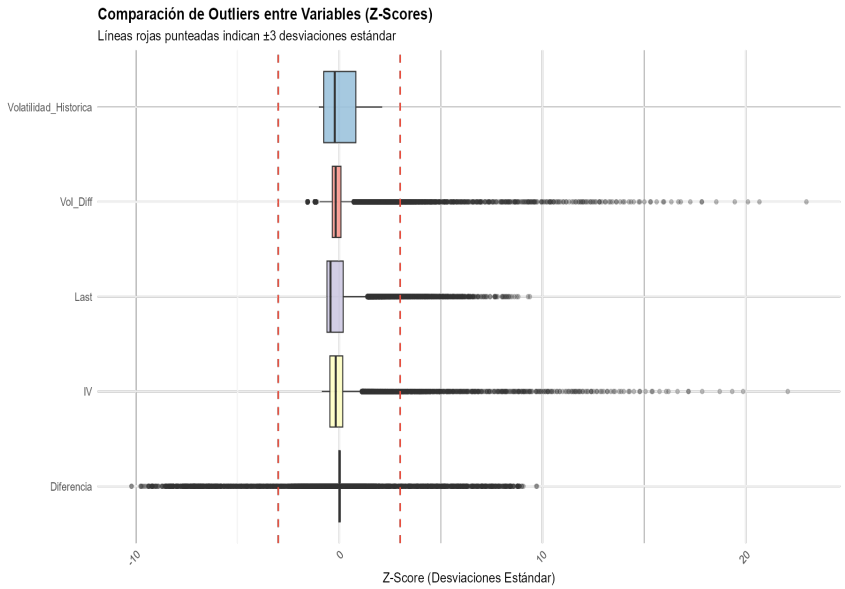


Figura 3.13: Panel comparativo de outliers entre variables mediante z-scores normalizados. Las líneas rojas punteadas indican umbrales de ±3 desviaciones estándar. *Fuente:* Elaboración propia.

Para la **volatilidad implícita (IV)**, como se observa en la Figura 3.14 que presenta la distribución por tipo de opción, se estableció un umbral de eliminación en IV > 2,15 (215 %), el cual filtra valores claramente imposibles incluso para los activos más volátiles del mercado en momentos de crisis. Este criterio identificó 432 observaciones en opciones call (2.08 % del total de call) y 155 en opciones put (0.79 % del total de put), evidenciando que las call presentan 2.6 veces más casos con volatilidades implícitas extremas que las put. Los paneles de la figura muestran claramente esta asimetría: la distribución de call exhibe una cola derecha más extendida hacia valores extremos, mientras que la distribución de put se concentra más fuertemente en el rango de volatilidades moderadas. Esta asimetría podría reflejar mayor incertidumbre o errores de cotización en opciones call con características particulares (típicamente OTM de corto plazo con baja liquidez).

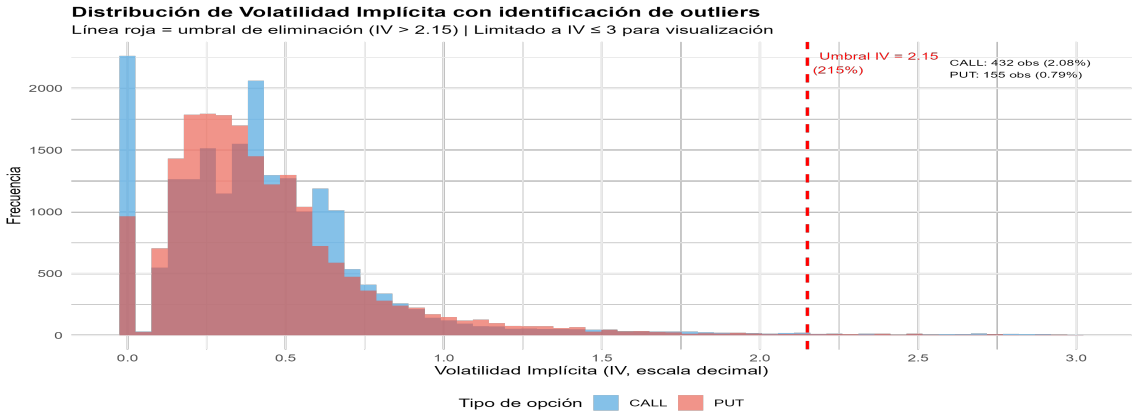


Figura 3.14: Distribución de volatilidad implícita por tipo de opción. La línea roja indica el umbral de eliminación (IV = 2.15). *Fuente:* Elaboración propia.

Para la **volatilidad histórica**, como se muestra en la Figura 3.15, la distribución resultó notablemente estable en ambos tipos de opciones, sin outliers detectados mediante el criterio IQR. No obstante, se aplicó un umbral preventivo en 1.5 (150 % anualizado) como medida conservadora contra posibles anomalías en el cálculo. Este umbral no eliminó ninguna observación en el dataset, confirmando que todos los activos analizados presentan volatilidades históricas dentro de rangos razonables.

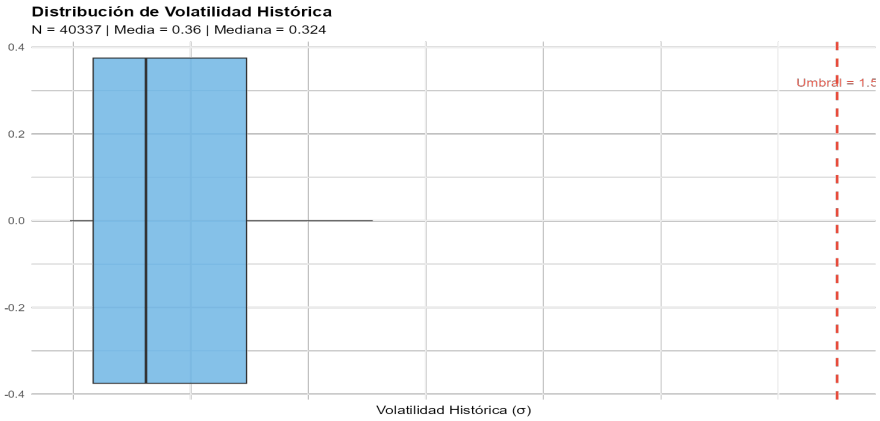


Figura 3.15: Distribución de volatilidad histórica. La línea roja punteada indica el umbral preventivo establecido en $\sigma = 1{,}5$. *Fuente:* Elaboración propia.

En contraste, los outliers detectados en **Diferencia**, **Last**, y **Vol_Diff** no fueron filtrados, independientemente del tipo de opción, ya que estos valores extremos representan fenómenos legítimos del mercado: errores genuinos del modelo Black-Scholes que constituyen precisamente el objeto de estudio; precios de mercado observados directamente cuya eliminación introduciría sesgos de selección; y divergencias reales entre volatilidades implícita e histórica que capturan expectativas del mercado. La Tabla 3.7 resume los criterios que se aplicaran en la siguiente sección.

Tabla 3.7: Criterios de tratamiento de outliers por variable. *Fuente:* Elaboración propia.

Variable	Criterio	Justificación	Acción
IV	IV > 2,15	Valores > 215 % son imposibles incluso en crisis (CALL: 432, PUT: 155 obs.)	Eliminar
Volatilidad Histórica	$\sigma > 1{,}5$	Volatilidades históricas > 150 % son extremadamente raras (0 obs. en ambos tipos)	Eliminar
Last	Ninguno	Precios de mercado observados - no filtrar	Retener
Diferencia	Ninguno	Errores extremos son parte del fenómeno a estudiar	Retener
Vol_Diff	Ninguno	Diferencias extremas de volatilidad son legítimas	Retener

Análisis de correlaciones entre variables

Una vez consolidadas todas las variables, se procedió a analizar sistemáticamente sus interrelaciones mediante matrices de correlación separadas por tipo de opción. Como se observa en las Figuras 6.6 y 6.7 (Anexo 6.3), donde los colores rojos indican correlaciones negativas, verdes representan correlaciones positivas, y la intensidad del color refleja la magnitud de la correlación, emergen patrones fundamentalmente distintos entre opciones *call* y *put*.

El análisis comparativo revela diferencias estructurales críticas. En primer lugar, la correlación entre precio de mercado (`Last`) y precio teórico (`Precio_BS`) difiere sustancialmente: 0.73 para opciones call versus 0.84 para opciones put, confirmando que Black-Scholes presenta mejor ajuste para puts. Esta correlación imperfecta en ambos casos representa precisamente el error sistemático que se busca modelar mediante técnicas de machine learning. Un hallazgo particularmente relevante es la presencia de correlaciones con signos opuestos entre call y put respecto a la variable objetivo (`Diferencia`). La Tabla 3.8 documenta las variables con mayor divergencia en sus patrones de correlación.

Tabla 3.8: Variables con patrones de correlación divergentes respecto a Diferencia. *Fuente:* Elaboración propia.

Variable	Cor. CALL	Cor. PUT	Diferencia Abs.
Strike_Normalizado	+0.50	−0.90	1.39
Last	+0.53	−0.45	0.98
Log_Moneyness	−0.39	+0.47	0.86
Precio_BS	−0.19	−0.86	0.67
Strike	+0.22	−0.44	0.65
Precio_Actual	−0.26	+0.25	0.50
Vol_Diff	−0.03	+0.31	0.34
Volatilidad_Histórica	+0.16	−0.16	0.32

El caso más notable corresponde a `Strike_Normalizado`, que presenta correlación fuertemente

positiva (+0.50) con el error en opciones call pero fuertemente negativa (−0.90) en opciones put, con una diferencia absoluta de 1.39. Esta divergencia dramática indica que la relación entre la distancia normalizada del strike y el error de Black-Scholes opera en direcciones completamente opuestas según el tipo de opción. Similarmente, `Log_Moneyness` exhibe signos opuestos (−0.39 para call, +0.47 para put), sugiriendo que la misma característica estructural de moneyness tiene efectos contrarios sobre el error del modelo dependiendo del tipo de derivado.

Estas divergencias fundamentales en los patrones de correlación confirman que las relaciones entre variables predictoras y el error de *Black-Scholes* no son uniformes, sino que dependen críticamente del tipo de opción. También se observó que `Moneyness` y `Log_Moneyness` exhiben correlación de 0.50 en ambos tipos, suficientemente baja para justificar la inclusión de ambas variables, ya que capturan aspectos complementarios de la relación strike-precio: `Moneyness` en espacio lineal y `Log_Moneyness` en el espacio logarítmico donde *Black-Scholes* opera matemáticamente.

Ninguna variable individual presenta una correlación dominante con la variable objetivo (`Diferencia`), pero los patrones difieren notablemente por tipo. Para opciones *call*, las correlaciones más fuertes corresponden a `Last` (+0.53) y `Strike_Normalizado` (+0.50). Para opciones *put*, destacan `Strike_Normalizado` (−0.90), `Precio_BS` (−0.86), y `Log_Moneyness` (+0.47). Este resultado sugiere que el error de Black-Scholes no puede predecirse mediante una única variable, sino que constituye un fenómeno multifactorial resultante de la interacción compleja entre múltiples factores, cuyas relaciones operan de manera fundamentalmente distinta según se trate de opciones *call* o *put*. Esta evidencia empírica justifica tanto la necesidad de modelos multivariados capaces de capturar interacciones no lineales, como la decisión metodológica de analizar ambos tipos de opciones por separado en todas las etapas del estudio.

Comparación entre precio teórico y precio de mercado

La Figura 3.16 presenta un diagrama de dispersión de esta relación para una muestra aleatoria de 5,000 opciones. La línea punteada negra representa la relación de igualdad perfecta (precio teórico = precio real). La dispersión significativa de puntos alrededor de esta línea evidencia las discrepancias sistemáticas que el modelo presenta. Se observan dos patrones claramente diferenciados:
(i) en opciones de bajo precio (< $20), *Black-Scholes* tiende a sobrestimar sistemáticamente los valores de mercado, generando una nube de puntos por encima de la línea de igualdad.

(ii) en opciones de mayor precio, la correlación es más fuerte –los puntos se acercan más a la línea diagonal– pero persiste dispersión considerable, indicando que aunque el modelo captura la tendencia general, los errores individuales pueden ser sustanciales.

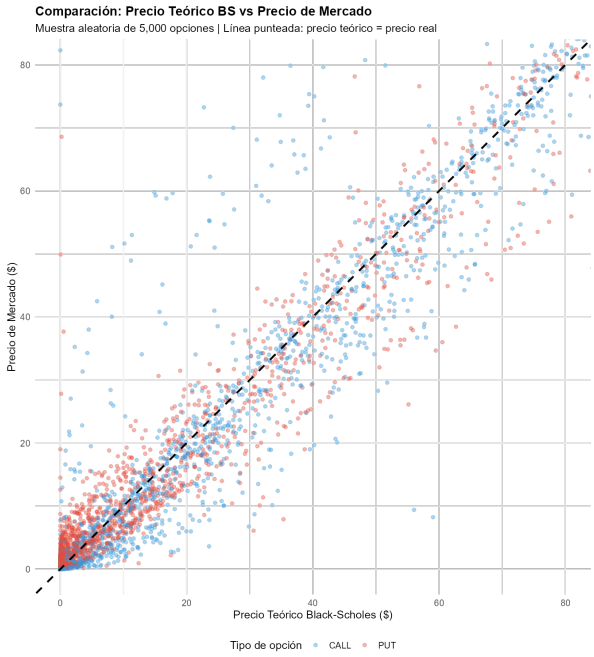


Figura 3.16: Comparación entre precios teóricos Black-Scholes y precios de mercado (muestra de 5,000 opciones). *Fuente:* Elaboración propia.

La correlación entre precio teórico y precio de mercado resultó de $r = 0{,}7587$ ($R^2 = 0{,}5756$), indicando que Black-Scholes explica aproximadamente el 58 % de la varianza en los precios de mercado. Desagregando por tipo de opción, se obtuvieron correlaciones de $r = 0{,}7386$ para calls y $r = 0{,}8552$ para puts, sugiriendo que el modelo presenta un mejor ajuste para opciones put que para opciones call en este conjunto de datos. El 42 % de varianza no explicada representa errores sistemáticos que varían según características estructurales de las opciones: el tipo de contrato (call vs put), el nivel de moneyness, el plazo hasta el vencimiento, y la discrepancia entre la volatilidad implícita del mercado y la volatilidad histórica empleada en el cálculo. La magnitud de estas discrepancias justifica plenamente el objetivo central de este estudio: desarrollar modelos de machine learning capaces de capturar estos patrones sistemáticos de error y generar correcciones más precisas que un simple ajuste por constante.

3.4. Limpieza y preparación de bases de datos

La calidad de los datos constituye un aspecto fundamental en cualquier análisis cuantitativo. Datos con valores faltantes, errores de medición, o valores atípicos extremos pueden comprometer seriamente la validez de los resultados y el desempeño de los modelos predictivos. Por esta razón, se implementó un proceso riguroso de limpieza y verificación de calidad de datos antes de proceder con el análisis y modelado. El proceso de limpieza se aplicó al conjunto completo de 40,337 observaciones utilizando los criterios y umbrales identificados durante el análisis exploratorio de la sección anterior. La limpieza se ejecutó en dos etapas secuenciales bien definidas: primero, la eliminación de

observaciones con valores faltantes en variables críticas para el modelado; segundo, la remoción de observaciones con volatilidades extremas que superan umbrales de plausibilidad económica.

Proceso de limpieza

Como se ilustra en la Figura 3.17, el proceso se ejecutó en dos etapas secuenciales. La primera etapa eliminó 1,860 observaciones (4.61 %) con valores faltantes en cualquiera de las variables críticas para el modelado (IV, Vol, OI), resultando en 38,477 observaciones. La segunda etapa removió 531 observaciones (1.32 % del total original) que excedían los umbrales establecidos para volatilidades extremas (IV > 2.15), produciendo un dataset limpio final de 37,947 observaciones con una tasa de retención del 94.07 %.

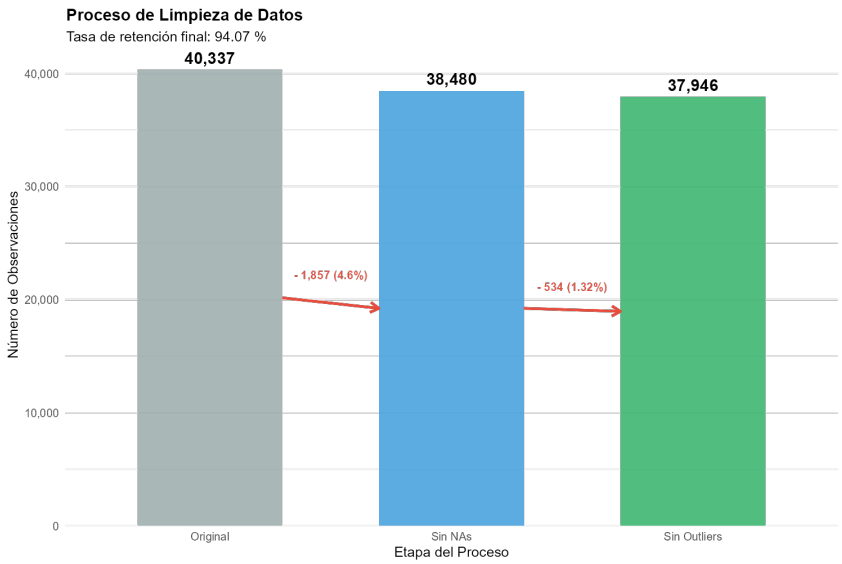


Figura 3.17: Proceso de limpieza del dataset. *Fuente:* Elaboración propia.

El dataset resultante de 37,947 observaciones cumple con todos los requisitos de calidad: ausencia total de valores faltantes, eliminación de volatilidades implausibles, y preservación de la variabilidad natural del mercado de opciones. La tasa de retención del 94.07 % indica que el proceso de limpieza fue selectivo pero no excesivamente restrictivo, manteniendo la representatividad del conjunto de datos. Este dataset limpio constituye la base confiable para la partición en conjuntos de entrenamiento y prueba, así como para el desarrollo y evaluación de los modelos predictivos en el capítulo siguiente.

Verificaciones de integridad

Finalmente, se ejecutaron verificaciones de integridad sobre el dataset limpio para confirmar que cumple con todas las restricciones lógicas y de dominio requeridas para el análisis subsecuente.

La Tabla 3.9 presenta los resultados de estas verificaciones, donde se puede comprobar que todas resultaron exitosas.

Tabla 3.9: Verificaciones de integridad del dataset limpio (N = 37,947). *Fuente:* Elaboración propia.

Verificación	Resultado	N° Obs.
Valores faltantes	Ninguno detectado	0
Registros duplicados	Ninguno detectado	0
Strike > 0	Cumple totalmente	37,947
Precio_Actual > 0	Cumple totalmente	37,947
Dias_Vencimiento >= 0	Cumple totalmente	37,947
Last > 0	Cumple totalmente	37,947
IV ≤ 2.15	Cumple totalmente	37,947
Volatilidad_Historica > 0	Cumple totalmente	37,947

3.5. Consolidación y estructura del dataset final

Al concluir el proceso de limpieza y creación de variables, se consolidó un dataset estructurado con 37,947 observaciones y 25 variables organizadas sistemáticamente en seis categorías funcionales, como se detalla en la Tabla 3.10. Esta estructura refleja la arquitectura lógica del análisis: desde las variables de identificación que permiten segmentar el análisis por activo y tipo de opción, pasando por los parámetros fundamentales requeridos por *Black-Scholes*, hasta las variables derivadas mediante ingeniería de características, los datos proporcionados por el mercado, los outputs del modelo teórico, y metadatos auxiliares para análisis complementarios.

Tabla 3.10: Estructura del dataset final consolidado (N = 37,947). *Fuente:* Elaboración propia.

Categoría	Variables	Tipo
Identificadores (2)	`Ticker`, `Tipo`	Categórica
Inputs BS (5)	`Precio_Actual` (S_0), `Strike` (K), `Dias_Vencimiento` (T), `Volatilidad_Historica` (σ), `Tasa_Libre_Riesgo` (r)	Numérica ($, días, %)
Variables de Mercado (4)	`Last`, `IV`, `Vol`, `OI`	Numérica ($, %, contratos)
Variables Derivadas (5)	`Moneyness` (S/K), `Log_Moneyness` [$\ln(S/K)$], `Vol_Diff` (IV$-\sigma$), `Strike_Normalizado` [$(K-S)/S$], `In_The_Money`	Numérica/Binaria (0/1)
Outputs (4)	`Precio_BS`, `Diferencia` (target), `Precio_Calibrado`, `Error_Calibrado`	Numérica ($)

Continúa en la siguiente página...

Tabla 3.10 – continuación

Categoría	Variables	Tipo
Metadata (5)	`Fecha_Descarga`, `Fecha_Vencimiento`, `Fecha_Vencimiento_Date`, `T_años`, `Moneyness_Grupo`	Fecha/Texto/Numérica/Categórica
Total: 25 variables		

Las 25 variables se distribuyen de la siguiente manera:

- **Identificadores (2 variables):** Permiten segmentar el análisis por activo subyacente (`Ticker`) y tipo de contrato (`Tipo`: CALL o PUT). Como se documentó en el análisis exploratorio, estas variables son fundamentales dado que el modelo Black-Scholes presenta patrones de error sistemáticamente diferentes entre opciones call y put.
- **Inputs de Black-Scholes (5 variables):** Los cinco parámetros estrictamente necesarios para calcular el precio teórico: precio spot del subyacente (`Precio_Actual`, S_0), precio de ejercicio (`Strike`, K), tiempo hasta vencimiento en días (`Dias_Vencimiento`, posteriormente convertido a años para el cálculo), volatilidad histórica anualizada del subyacente (`Volatilidad_Historica`, σ), y tasa de interés libre de riesgo anualizada (`Tasa_Libre_Riesgo`, r).
- **Variables de mercado (4 variables):** Información proporcionada directamente por el mercado de opciones que captura aspectos no contemplados en la formulación teórica de Black-Scholes: precio último negociado que constituye el *ground truth* del análisis (`Last`), volatilidad implícita que refleja las expectativas del mercado sobre volatilidad futura (`IV`), volumen de negociación diario (`Vol`), e interés abierto que contabiliza contratos sin liquidar (`OI`).
- **Variables derivadas (5 variables):** Características generadas mediante ingeniería de variables para capturar relaciones no lineales y patrones complejos: ratio precio-strike (`Moneyness`, S/K), su transformación logarítmica que opera en el espacio matemáticamente relevante del modelo (`Log_Moneyness`, $\ln(S/K)$), diferencia entre volatilidades que captura la discrepancia entre expectativas del mercado y datos históricos (`Vol_Diff`, IV $-\ \sigma$), strike normalizado que permite comparaciones consistentes entre activos (`Strike_Normalizado`, $(K-S)/S$), e indicador binario de posición moneyness (`In_The_Money`).
- **Outputs calculados (4 variables):** Resultados del cálculo de Black-Scholes y calibraciones subsecuentes: precio teórico calculado mediante las ecuaciones estándar (`Precio_BS`), error del modelo que constituye la variable objetivo del estudio (`Diferencia` = `Last` − `Precio_BS`), precio calibrado mediante ajuste con constante aditiva C (`Precio_Calibrado`), y error del modelo calibrado (`Error_Calibrado`).
- **Metadata auxiliar (5 variables):** Variables generadas para análisis complementarios, validaciones y trazabilidad: fecha de recolección de datos (`Fecha_Descarga`), fecha de vencimiento en formato texto original (`Fecha_Vencimiento`), fecha de vencimiento procesada en formato date

(`Fecha_Vencimiento_Date`), tiempo hasta vencimiento expresado en años para verificaciones (`T_años`), y clasificación categórica de moneyness en grupos OTM/ATM/ITM (`Moneyness_Grupo`).

Con 37,947 observaciones limpias y 25 variables sistemáticamente organizadas, se consolidó un dataset robusto y metodológicamente estructurado. Las 18 variables de las primeras cuatro categorías (identificadores, inputs BS, variables de mercado, y variables derivadas) constituyen el conjunto de predictores potenciales disponibles para los modelos de machine learning, mientras que `Diferencia` representa la variable objetivo cuya predicción precisa constituye el objetivo central de este estudio. Las 4 variables de outputs adicionales y las 5 de metadata proporcionan capacidades de validación cruzada, benchmarking contra modelos baseline, análisis segmentado por características estructurales de las opciones, y trazabilidad completa del proceso analítico.

3.6. Partición en conjuntos de entrenamiento y prueba

Una vez completado el proceso de limpieza y consolidación del dataset (37,947 observaciones con 25 variables), se procedió a realizar la partición en conjuntos de entrenamiento y prueba siguiendo las mejores prácticas metodológicas en machine learning [16, 17]. La decisión de ejecutar la limpieza previo a la partición se fundamenta en que las operaciones realizadas—eliminación de valores faltantes y de volatilidades atípicas—constituyen correcciones de calidad de datos que no implican aprendizaje de patrones ni generan riesgo de fuga de información (*data leakage*), a diferencia de transformaciones como normalización o imputación de valores que sí deben realizarse post-partición. Para la partición del dataset se consideraron dos variables categóricas fundamentales: (i) **Ticker (activo subyacente)**, dado que cada uno de los 10 activos presenta características distintivas en términos de volatilidad, liquidez y comportamiento de mercado; y (ii) **Tipo de opción (CALL/PUT)**, ya que el análisis exploratorio reveló patrones de comportamiento diferentes entre opciones call y put. La combinación de estas dos variables genera 20 agrupaciones (10 tickers × 2 tipos de opción), cada uno con suficientes observaciones para garantizar particiones robustas y representativas. Se utilizó la función `createDataPartition` del paquete `caret` en R, que implementa muestreo estratificado manteniendo proporciones aproximadas del 80 %-20 % en cada uno de los 20 estratos. La partición del dataset limpio de 37,947 observaciones asignó 30,365 observaciones (80 %) al conjunto de entrenamiento y 7,582 observaciones (20 %) al conjunto de prueba, como se detalla en la Tabla 3.11.

Tabla 3.11: Partición del dataset limpio en conjuntos de entrenamiento y prueba. *Fuente:* Elaboración propia.

Conjunto	Observaciones	Proporción	Uso
Entrenamiento	30,365	80.0 %	Desarrollo y ajuste de modelos
Prueba	7,582	20.0 %	Evaluación final de desempeño
Total	**37,947**	**100 %**	

La Figura 3.18 ilustra el proceso completo de partición, desde el dataset limpio hasta la generación de los archivos separados por tipo de opción.

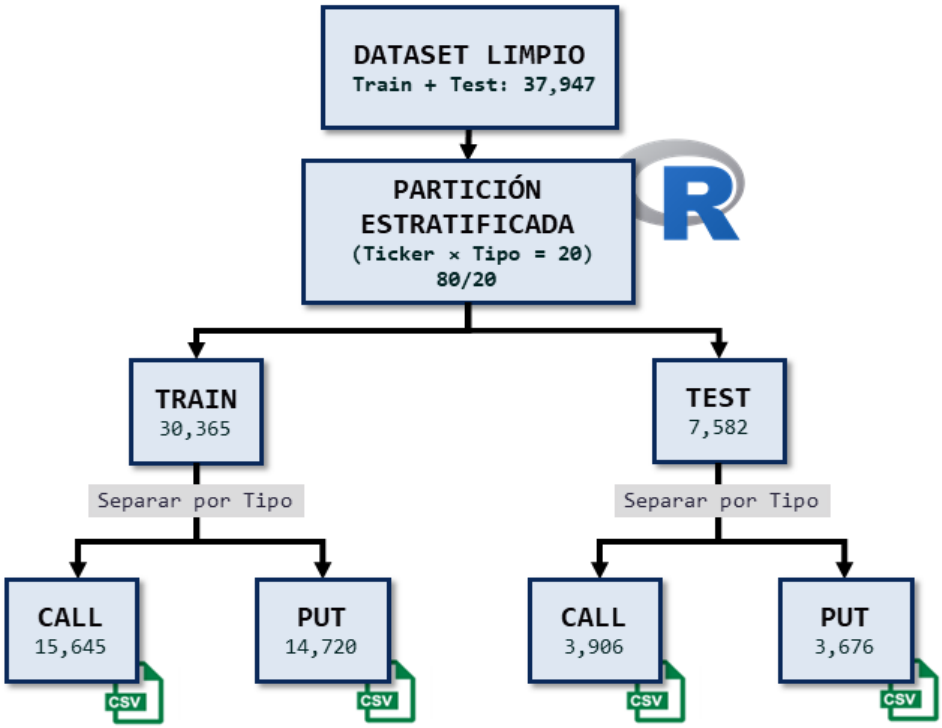


Figura 3.18: Diagrama del proceso de partición estratificada del dataset. *Fuente:* Elaboración propia.

Posterior a la partición, se verificó que las distribuciones de variables clave—moneyness (S/K), plazo al vencimiento (días), volatilidad implícita (IV), y precio spot—fueran comparables entre ambos conjuntos mediante pruebas de Kolmogorov-Smirnov. Los resultados presentados en la Tabla 3.12 confirman que no existen diferencias estadísticamente significativas entre los conjuntos de entrenamiento y prueba ($p > 0{,}05$ en todas las variables analizadas), validando que ambos conjuntos son representativos y que el proceso de partición fue exitoso.

Tabla 3.12: Pruebas de Kolmogorov-Smirnov: Comparación de distribuciones entre conjuntos. *Fuente:* Elaboración propia.

Variable	Estadístico KS	p-valor	Conclusión
Moneyness (S/K)	0.0116	0.384	No hay diferencia
Dias_Vencimiento	0.0109	0.467	No hay diferencia
Volatilidad_Implícita (IV)	0.0076	0.873	No hay diferencia
Precio_Actual (S)	0.0002	1.000	No hay diferencia

La Tabla 3.13 presenta la distribución detallada de observaciones por estrato en ambos conjuntos, confirmando que la proporción 80 %-20 % se mantiene consistentemente a través de todos los estratos

(tickers y tipos de opción).

Tabla 3.13: Distribución de observaciones por estrato (ticker y tipo de opción). *Fuente:* Elaboración propia.

Ticker	Tipo	Entrenamiento	Prueba	Total
AAPL	CALL	834	208	1,042
AAPL	PUT	777	194	971
AMD	CALL	818	204	1,022
AMD	PUT	728	181	909
AMZN	CALL	780	195	975
AMZN	PUT	721	180	901
GOOGL	CALL	823	205	1,028
GOOGL	PUT	756	189	945
META	CALL	1,813	453	2,266
META	PUT	1,588	397	1,985
MSFT	CALL	1,104	275	1,379
MSFT	PUT	1,015	253	1,268
NVDA	CALL	1,890	472	2,362
NVDA	PUT	1,832	458	2,290
QQQ	CALL	2,695	673	3,368
QQQ	PUT	2,591	647	3,238
SPY	CALL	2,951	737	3,688
SPY	PUT	3,001	750	3,751
TSLA	CALL	1,937	484	2,421
TSLA	PUT	1,711	427	2,138
Total (20 estratos)		**30,365**	**7,582**	**37,947**

A partir de este punto, el conjunto de prueba (7,582 observaciones) se mantuvo completamente aislado durante todo el desarrollo de modelos presentado en el Capítulo 4. Este conjunto no se utilizó en ninguna decisión de diseño, selección de hiperparámetros, o ajuste de modelos. Su único uso fue la evaluación final e imparcial del desempeño predictivo de los modelos desarrollados. Esta separación estricta constituye una práctica fundamental en ciencia de datos para evitar sobreajuste y obtener métricas de desempeño realistas que reflejen el comportamiento esperado del modelo. El conjunto de entrenamiento (30,365 observaciones), por su parte, se empleará en el Capítulo 4 para el desarrollo, entrenamiento y ajuste de todos los modelos predictivos, incluyendo el modelo baseline de corrección mediante constante aditiva, regresión lineal, Random Forest, XGBoost, KNN y Redes neuronales.

Capítulo 4

Modelación mediante Aprendizaje Automático

Este capítulo desarrolla los objetivos específicos 2,3 y 4 del proyecto, el cual consiste en construir e implementar distintos modelos de aprendizaje automático para predecir con precisión los errores entre los valores reales y teóricos de las opciones europeas call y put, y utilizar dichos errores para ajustar el valor teórico calculado mediante el modelo de Black-Scholes. El enfoque adoptado no busca reemplazar el modelo Black-Scholes, sino complementarlo mediante la predicción sistemática de sus errores, generando así precios ajustados más cercanos a los valores observados en el mercado.

Los resultados obtenidos en este capítulo demuestran que los modelos de machine learning pueden capturar patrones complejos en los errores de valoración que el modelo teórico no contempla, reduciendo significativamente la discrepancia entre precios teóricos y reales. Se implementaron y compararon tres técnicas de aprendizaje automático con diferentes niveles de complejidad: Regresión Lineal (modelo base), Random Forest y XGBoost, evaluando su capacidad para predecir la variable objetivo Diferencia definida en el capítulo anterior. Los scripts completos utilizados para el desarrollo de los modelos de este capítulo pueden consultarse en el repositorio público del proyecto: https://github.com/julian936/black-scholes-ml-correction.

4.1. Modelo Baseline: Predicción mediante Constante C

Antes de implementar modelos de aprendizaje automático de mayor complejidad, resulta fundamental establecer un punto de referencia o *baseline* que permita evaluar objetivamente las mejoras obtenidas. En esta sección se desarrolla el modelo más simple posible para la predicción del error de Black-Scholes: la estimación mediante una constante aditiva C. Este enfoque, aunque elemental, proporciona el umbral mínimo de desempeño que cualquier modelo posterior debe superar para

justificar su complejidad adicional. El modelo baseline parte de una observación empírica fundamental: si el modelo Black-Scholes presenta un sesgo sistemático en sus predicciones, entonces una corrección mediante una constante aditiva podría reducir el error promedio. Matemáticamente, el modelo ajustado se define como:

$$\text{Precio}_{\text{Ajustado}} = \text{Precio}_{\text{BS}} + C \tag{4.1}$$

donde C representa la constante de corrección que captura el sesgo promedio del modelo Black-Scholes. La variable objetivo del modelo es la **Diferencia** entre el precio de mercado y el precio teórico de Black-Scholes:

$$\text{Diferencia} = \text{Precio}_{\text{Real}} - \text{Precio}_{\text{BS}} \tag{4.2}$$

Al predecir esta diferencia con una constante C, el precio ajustado se aproxima al precio real de mercado. El error residual del modelo ajustado se expresa como:

$$\text{Error}_{\text{Residual}} = \text{Diferencia} - C = (\text{Precio}_{\text{Real}} - \text{Precio}_{\text{BS}}) - C \tag{4.3}$$

Este enfoque es conceptualmente equivalente a estimar una medida de tendencia central de la diferencia y utilizarla como predicción constante para todas las observaciones. Aunque ignora cualquier relación entre las características de las opciones y el error de valoración, establece el punto de partida contra el cual se medirán los modelos más sofisticados.

4.1.1. Proceso de modelado

La Figura 4.1 presenta el flujo metodológico del modelo baseline, desde la definición de variables hasta la evaluación del desempeño.

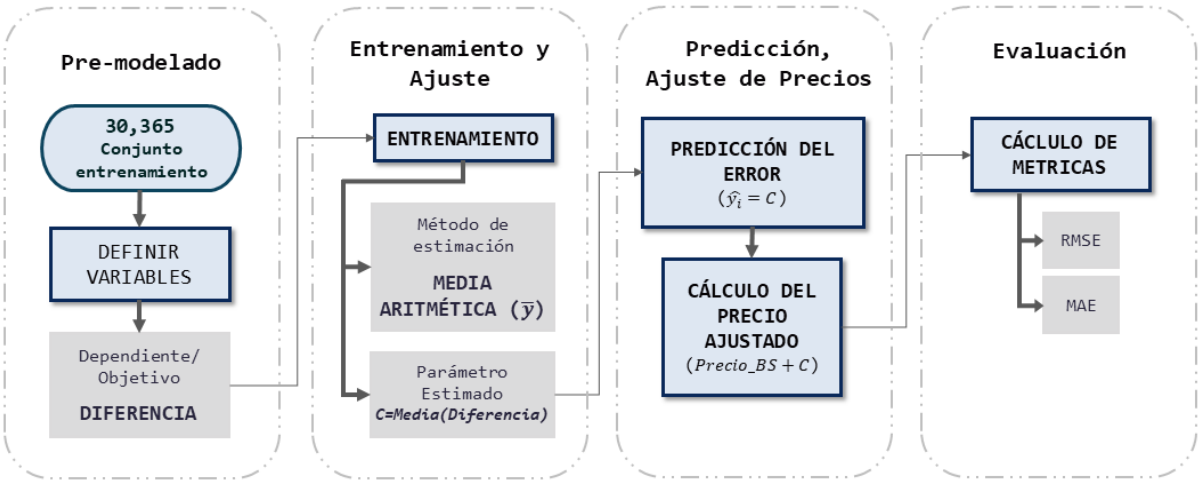


Figura 4.1: Diagrama metodológico del modelo baseline. *Fuente:* Elaboración propia.

El proceso se estructura en cuatro fases:

1. **Pre-modelado**: Definición de la variable objetivo (Diferencia) sobre el conjunto de entrenamiento (30,365 observaciones).
2. **Entrenamiento y Ajuste**: Estimación de la constante C mediante la media aritmética de la variable Diferencia.
3. **Predicción y Ajuste de Precios**: Aplicación del modelo para predecir el error ($\hat{y}_i = C$) y cálculo del precio ajustado como Precio_BS + C.
4. **Evaluación**: Cálculo de métricas de desempeño (RMSE, MAE) sobre el conjunto de prueba.

4.1.2. Análisis de la variable Diferencia y selección de C

La selección del valor óptimo de la constante C requiere un análisis de la distribución de la variable *Diferencia*. Este análisis se realizó sobre el conjunto de entrenamiento (N = 30,365 observaciones) para evitar fuga de información desde el conjunto de prueba.

4.1.2.1. Distribución de errores por tipo de opción

El análisis exploratorio del Capítulo 3 reveló patrones de error sistemáticamente diferentes entre opciones CALL y PUT. Las Figuras 4.2 y 4.3 presentan las distribuciones de la variable *Diferencia* para cada tipo de opción, junto con las tres medidas de tendencia central: media, mediana y moda.

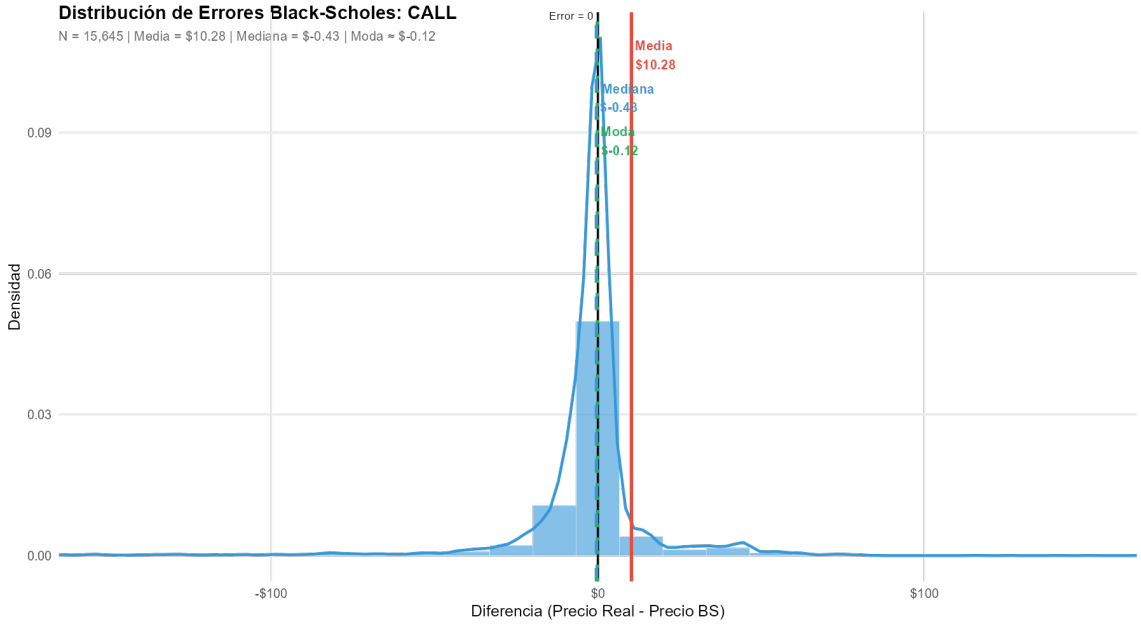


Figura 4.2: Distribución de la variable Diferencia para opciones CALL (N = 15,645). Las líneas verticales indican la media (roja, +$10,28), mediana (azul, −$0,43) y moda (verde, −$0,12). La línea negra representa el valor cero.

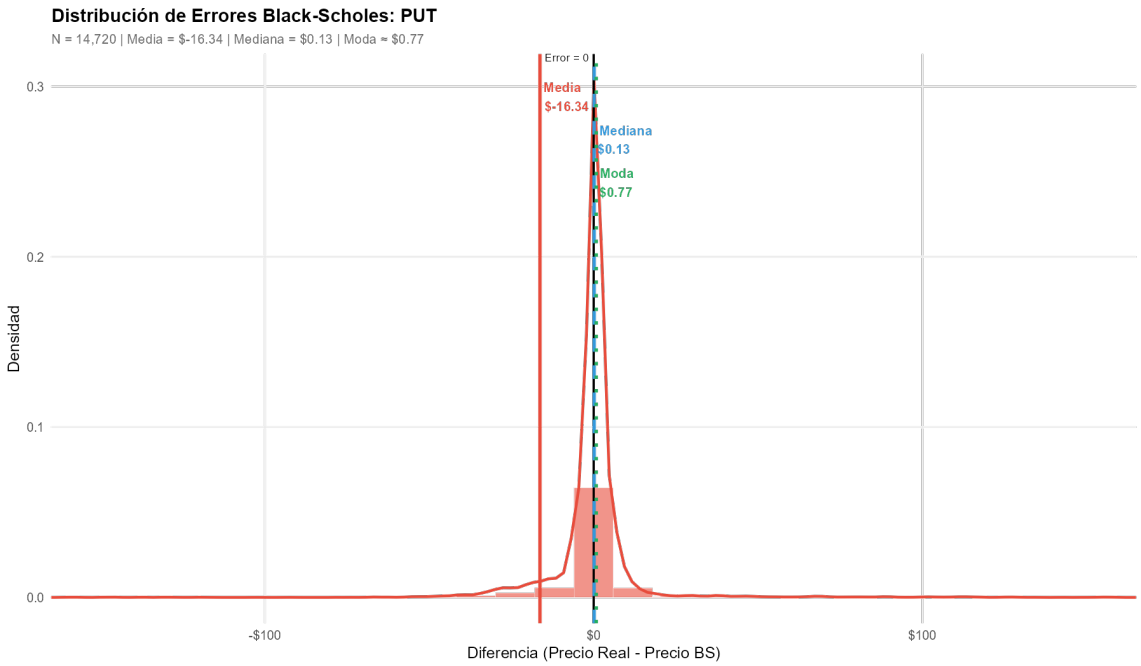


Figura 4.3: Distribución de la variable Diferencia para opciones PUT ($N = 14{,}720$). Las líneas verticales indican la media (roja, −$16,34), mediana (azul, +$0,13) y moda (verde, +$0,77). La línea negra representa el valor cero.

Las distribuciones revelan características importantes. En ambos casos, la mayoría de las opciones presentan errores pequeños cercanos a cero, como lo evidencian la mediana y la moda. Sin embargo, las colas de las distribuciones se extienden en direcciones opuestas: hacia valores positivos en opciones CALL y hacia valores negativos en opciones PUT.

4.1.2.2. Comparación de medidas de tendencia central

La Tabla 4.1 presenta la comparación de las tres medidas de tendencia central por tipo de opción.

Tabla 4.1: Comparación de medidas de tendencia central por tipo de opción.

Medida	**CALL** ($N = 15{,}645$)	**PUT** ($N = 14{,}720$)	**Diferencia absoluta**
Media	+$10,28	−$16,34	$26.62
Mediana	−$0,43	+$0,13	$0.56
Moda (kernel)	−$0,12	+$0,77	$0.89
Desviación Estándar	$88.69	$98.83	—

Los resultados revelan un patrón fundamental: mientras la mediana y la moda se encuentran cercanas a cero para ambos tipos de opciones, la media presenta valores sustancialmente diferentes de cero y

con signos opuestos. Esta diferencia tiene implicaciones directas para la selección de la constante C.

4.1.2.3. Selección de la media como estimador de C

La Figura 4.4 ilustra la comparación entre las tres medidas de tendencia central para cada tipo de opción.

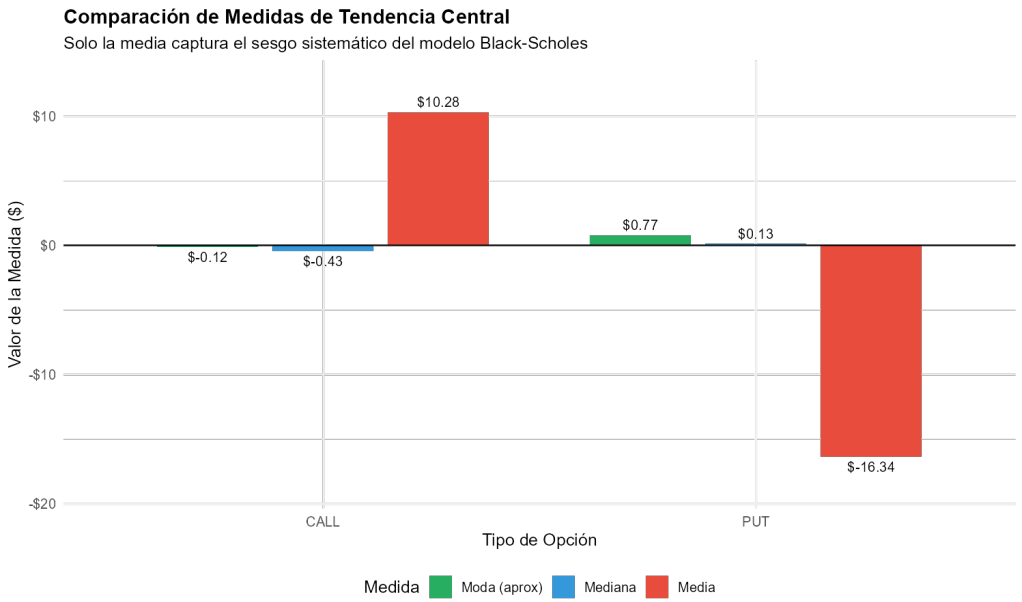


Figura 4.4: Comparación de medidas de tendencia central por tipo de opción. Solo la media captura el sesgo sistemático del modelo Black-Scholes; la mediana y la moda permanecen cercanas a cero.

Como se observa en la Figura 4.4, la mediana ($-\$0{,}43$ para CALL, $+\$0{,}13$ para PUT) y la moda ($-\$0{,}12$ para CALL, $+\$0{,}77$ para PUT) se encuentran cercanas a cero. Si se utilizara cualquiera de estas medidas como constante C, la corrección aplicada al modelo Black-Scholes sería prácticamente nula, ignorando el sesgo sistemático presente en la distribución.

La media aritmética, en contraste, captura este sesgo al ponderar proporcionalmente *todos* los valores de la distribución, incluyendo los errores de mayor magnitud. Aunque estos errores extremos son menos frecuentes, representan desviaciones económicamente significativas que el modelo Black-Scholes comete de manera sistemática.

Los valores de la media revelan el comportamiento diferenciado del modelo teórico:

- **Opciones CALL**: Media de $+\$10{,}28$, indicando que Black-Scholes **subvalora** sistemáticamente estas opciones (el mercado paga más de lo que predice el modelo).

- **Opciones PUT**: Media de −$16,34, evidenciando que Black-Scholes **sobrevalora** estas opciones (el mercado paga menos de lo predicho).

Por esta razón, se selecciona la **media** como estimador de la constante C, definiendo las siguientes correcciones:

$$C_{\text{CALL}} = +\$10{,}28 \quad \text{(corrige subvaloración)} \tag{4.4}$$

$$C_{\text{PUT}} = -\$16{,}34 \quad \text{(corrige sobrevaloración)} \tag{4.5}$$

4.1.3. Construcción y aplicación del modelo baseline

El proceso de construcción del modelo baseline sigue una secuencia metodológica que garantiza la validez de la evaluación:

1. **Cálculo de constantes**: Sobre el conjunto de entrenamiento, se calcula la media de la variable *Diferencia* separadamente para opciones CALL ($C_{\text{CALL}} = +\$10{,}28$) y PUT ($C_{\text{PUT}} = -\$16{,}34$).
2. **Aplicación del modelo**: Las constantes se aplican al conjunto de prueba según el tipo de cada opción, generando predicciones del error esperado.
3. **Evaluación**: Se calculan las métricas de desempeño comparando las predicciones (constante C) contra los valores reales de *Diferencia* en el conjunto de prueba.

El modelo resultante es:

$$\hat{\text{Diferencia}} = \begin{cases} +\$10{,}28 & \text{si la opción es CALL} \\ -\$16{,}34 & \text{si la opción es PUT} \end{cases} \tag{4.6}$$

4.1.4. Resultados del modelo baseline

La Tabla 4.2 presenta los resultados del modelo baseline evaluado sobre el conjunto de prueba, comparando el desempeño del modelo Black-Scholes original (equivalente a usar $C = 0$) contra el modelo corregido con la constante C calculada.

Tabla 4.2: Desempeño del modelo baseline sobre el conjunto de prueba ($N_{\text{test}} = 7{,}582$).

Tipo	Modelo	RMSE ($)	MAE ($)	Mejora RMSE
CALL ($N = 3{,}906$)	BS Original ($C = 0$)	98.42	25.96	—
	BS + C (Baseline)	**97.66**	32.33	+0,77 %
PUT ($N = 3{,}676$)	BS Original ($C = 0$)	105.02	22.78	—
	BS + C (Baseline)	**103.54**	34.06	+1,41 %

El modelo baseline logra mejoras modestas pero consistentes respecto al modelo Black-Scholes sin corrección:

- **Opciones CALL**: Reducción del RMSE de $98.42 a $97.66, representando una mejora del 0.77 %.
- **Opciones PUT**: Reducción del RMSE de $105.02 a $103.54, representando una mejora del 1.41 %.

4.1.5. Umbrales de referencia para modelos de Machine Learning

El modelo baseline establece los umbrales mínimos de desempeño que los modelos de aprendizaje automático desarrollados en las secciones siguientes deben superar para justificar su complejidad adicional:

Tabla 4.3: Umbrales baseline para evaluación de modelos de Machine Learning.

Tipo de Opción	RMSE Baseline ($)	Requisito para ML
CALL	97.66	RMSE < $97,66
PUT	103.54	RMSE < $103,54

Cualquier modelo que no logre superar estos umbrales no aporta valor predictivo adicional respecto a la simple corrección mediante constante, y por tanto no justificaría su implementación en un contexto práctico de valoración de opciones.

4.2. Regresión Lineal Múltiple

4.2.1. Proceso de modelado

La Figura 4.5 presenta el flujo metodológico completo del proceso de modelado mediante regresión lineal múltiple, desde la preparación de datos hasta la evaluación del desempeño predictivo.

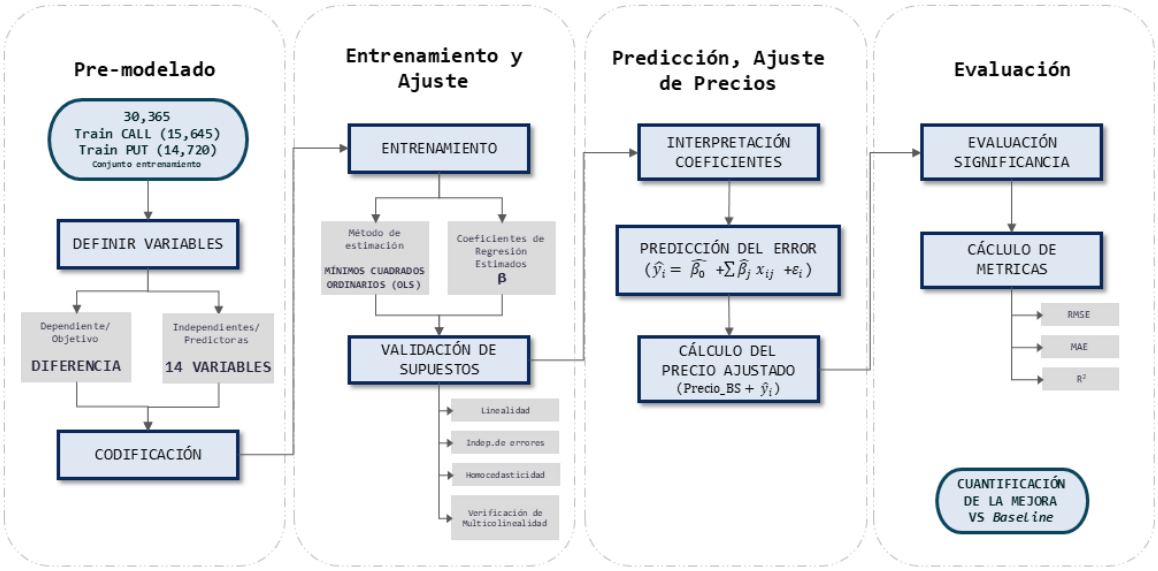


Figura 4.5: Diagrama del proceso de modelado mediante regresión lineal múltiple para la predicción del error de Black-Scholes.*Fuente:* Elaboración Propia.

El proceso se estructura en cuatro fases principales:

1. **Pre-modelado**: Preparación del conjunto de entrenamiento (30,365 observaciones), definición de la variable objetivo (Diferencia) y las 14 variables predictoras, junto con la codificación de variables categóricas (Ticker).
2. **Entrenamiento y Ajuste**: Estimación de los coeficientes $\hat{\boldsymbol{\beta}}$ mediante mínimos cuadrados ordinarios (OLS) y validación de los supuestos del modelo lineal.
3. **Predicción y Ajuste de Precios**: Aplicación del modelo entrenado para predecir el error $\hat{y}_i$ y cálculo del precio ajustado como Precio_BS + $\hat{y}_i$.
4. **Evaluación**: Cálculo de métricas de desempeño (RMSE, MAE, R^2) y cuantificación de la mejora respecto al baseline.

4.2.2. Construcción y entrenamiento del modelo

Del conjunto de variables disponibles en el dataset procesado, se seleccionaron 14 como variables explicativas potenciales, excluyendo las **variables de identificación** (ContractID, Fecha_Descarga, Fecha_Vencimiento), la **variable objetivo** (Diferencia) y sus derivadas (Error_Calibrado), el **precio de mercado** (Last) y el **precio teórico BS** (Precio_BS) ya que forman parte del cálculo de la variable Diferencia.

Dado que el análisis exploratorio reveló comportamientos diferenciados entre opciones CALL y PUT, se optó por entrenar **modelos separados** para cada tipo, permitiendo que los coeficientes capturen

las dinámicas específicas de cada instrumento. Las 14 variables explicativas empleadas se organizan en cuatro grupos:

1. **Inputs fundamentales de Black-Scholes:** Strike, Precio_Actual, Dias_Vencimiento, Volatilidad_Historica, Tasa_Libre_Riesgo
2. **Información del mercado de opciones:** Vol (volumen), OI (open interest), IV (volatilidad implícita)
3. **Variables derivadas (feature engineering):** Moneyness, Log_Moneyness, Vol_Diff, Strike_Normalizado, In_The_Money
4. **Variables categóricas:** Ticker (codificado mediante variables dummy, con AAPL como categoría de referencia)

El modelo de regresión lineal múltiple especificado para cada tipo de opción queda definido como:

$$\begin{aligned}\text{Diferencia}_i = {} & \beta_0 + \beta_1 \cdot \text{Strike}_i + \beta_2 \cdot S_i + \beta_3 \cdot T_i + \beta_4 \cdot \sigma_i + \beta_5 \cdot \text{Vol}_i + \beta_6 \cdot \text{OI}_i \\ & + \beta_7 \cdot \text{IV}_i + \beta_8 \cdot \left(\frac{S}{K}\right)_i + \beta_9 \cdot \ln\left(\frac{S}{K}\right)_i + \beta_{10} \cdot \left(\frac{K}{S}\right)_i \\ & + \beta_{11} \cdot \mathbf{1}_{\text{ITM},i} + \sum_{j=1}^{J} \gamma_j \cdot \mathbf{1}_{\text{Ticker}_j,i} + \varepsilon_i\end{aligned} \tag{4.7}$$

donde:

$S_i = \text{Precio\_Actual}_i$ (precio spot del activo subyacente)

$T_i = \text{Dias\_Vencimiento}_i$ (tiempo hasta expiración en días)

$\sigma_i = \text{Volatilidad\_Historica}_i$ (anualizada)

$\mathbf{1}_{\text{ITM},i}$, $\mathbf{1}_{\text{Ticker}_j,i}$ son funciones indicadoras binarias

γ_j son los coeficientes asociados a cada ticker (con AAPL como categoría de referencia)

$\varepsilon_i \sim N(0, \sigma^2)$ es el término de error aleatorio

La estimación de coeficientes $\hat{\boldsymbol{\beta}}$ se realizó mediante mínimos cuadrados ordinarios sobre 15,645 observaciones para opciones CALL y 14,720 para opciones PUT.

4.2.3. Validación de supuestos

Para validar el cumplimiento de los supuestos de la regresión lineal, se realizaron pruebas diagnósticas y análisis gráficos de residuos para ambos modelos. La Tabla 4.4 resume los resultados de las pruebas estadísticas.

Tabla 4.4: Resumen de pruebas diagnósticas para validación de supuestos.

Supuesto	**Test**	**CALL**		**PUT**	
		Estadístico	p-valor	Estadístico	p-valor
Linealidad	Visual	Ver Figura 6.1		Ver Figura 6.1	
Homocedasticidad	Breusch-Pagan	3,107.20	<0.001	7,598.56	<0.001
Normalidad	Anderson-Darling	591.51	<0.001	394.45	<0.001

Evaluación de Linealidad: Los gráficos de residuos versus valores ajustados (Figura 6.1, Anexo 6.1) muestran que ambos modelos exhiben violaciones severas del supuesto de linealidad:

- **CALL:** La curva LOESS revela un patrón no lineal pronunciado. Para valores ajustados cercanos a cero, los residuos se mantienen pequeños, pero para valores ajustados superiores a $200, los residuos descienden dramáticamente hasta −$500, formando una curvatura descendente. Además, se observan residuos positivos extremos (hasta +$800) dispersos para predicciones intermedias.
- **PUT:** Los valores ajustados son predominantemente negativos (rango de −$1,200 a $0). La curva LOESS muestra un patrón curvo que inicia con residuos altos (+$550) para valores muy negativos, desciende gradualmente, y cerca de cero presenta alta concentración de datos con residuos pequeños.

Implicación: Los patrones curvos indican que existen relaciones no lineales entre las variables explicativas y el error de Black-Scholes que el modelo lineal no puede capturar. Esto justifica la exploración de modelos no lineales como Random Forest y XGBoost.

Evaluación de Homocedasticidad: El test de Breusch-Pagan rechaza contundentemente la hipótesis nula de varianza constante para ambos modelos (CALL: $BP = 3{,}107{,}20$, $p < 0{,}001$; PUT: $BP = 7{,}598{,}56$, $p < 0{,}001$). Los gráficos Scale-Location (Figura 6.2, Anexo 6.1) ilustran gráficamente la heterocedasticidad:

- **CALL:** La curva LOESS forma un patrón en "U": la dispersión de residuos es mínima para valores ajustados cercanos a cero, pero aumenta considerablemente hacia ambos extremos, alcanzando valores de $\sqrt{|\text{residuos}|}$ superiores a 2 para predicciones mayores a $200.
- **PUT:** Se observa un patrón en forma de campana invertida. La varianza máxima ocurre para valores ajustados intermedios (alrededor de −$400 a −$600), mientras que la varianza es menor tanto para valores cercanos a cero como para valores muy negativos (−$1,200).

Implicación: El modelo comete errores de diferente magnitud dependiendo del nivel de predicción. Esto no afecta la validez de las predicciones ni los valores estimados de los coeficientes, pero sí reduce la confiabilidad de los p-valores para determinar qué variables son estadísticamente significativas.

Implicación: El modelo comete errores de diferente magnitud dependiendo del nivel de predicción. Esto no afecta la validez de las predicciones ni los valores estimados de los coeficientes, pero sí reduce la confiabilidad de los p-valores para determinar qué variables son estadísticamente significativas.

Evaluación de Normalidad de residuos: El test de Anderson-Darling rechaza la normalidad para ambos modelos ($p < 0{,}001$). Los gráficos Q-Q (Figura 6.3, Anexo 6.1) revelan desviaciones extremas:

- **CALL:** Presenta la violación más severa. Los cuantiles muestrales alcanzan valores de +12 en la cola superior (cuando lo esperado bajo normalidad sería aproximadamente +3), y hasta -6 en la cola inferior. Esto refleja una distribución con colas mucho más pesadas que la normal (curtosis = 42,29) y asimetría positiva pronunciada (skewness = 4,62), indicando mayor frecuencia de errores extremos positivos.
- **PUT:** También exhibe colas pesadas, con cuantiles muestrales que alcanzan +15 en el extremo superior y -6 en el inferior. Aunque la desviación es significativa, es menos extrema que en CALL (curtosis = 28,48, skewness = 1,03).

Implicación: Las colas pesadas indican la presencia de errores extremos (outliers) que el modelo lineal no puede ajustar adecuadamente. Estos corresponden a opciones donde el error de Black-Scholes es particularmente grande, típicamente opciones deep out-of-the-money o con características atípicas.

Evaluación de Multicolinealidad: Las matrices de correlación presentadas en el Capítulo 3 (Figuras 6.6 y 6.7) revelan correlaciones altas entre variables que capturan aspectos similares de la relación strike/precio:

- Moneyness (S/K)
- Log_Moneyness ($\ln(S/K)$)
- Strike_Normalizado ($(K-S)/S$)

Este comportamiento es esperado dado que las tres son transformaciones matemáticas de las mismas variables fundamentales: Strike (K) y Precio_Actual (S).
Implicación: La multicolinealidad no afecta la capacidad predictiva del modelo ni sesga las estimaciones de los coeficientes. Sin embargo, dificulta interpretar el efecto individual de cada variable, ya que cuando una cambia, las otras tienden a cambiar simultáneamente.

Conclusión sobre supuestos

Los tres supuestos evaluables (linealidad, homocedasticidad y normalidad) presentan violaciones severas. Este resultado es común en aplicaciones financieras donde las relaciones entre variables son inherentemente complejas y no lineales. Las violaciones observadas:

1. No invalidan las estimaciones puntuales de los coeficientes, que permanecen insesgadas.

2. Afectan la inferencia estadística (intervalos de confianza y p-valores), que debe interpretarse con cautela.
3. Justifican la exploración de modelos no lineales (Random Forest, XGBoost) que no requieren estos supuestos y pueden capturar las relaciones complejas identificadas en los residuos.

4.2.4. Resultados para opciones CALL

La Tabla 4.5 presenta los coeficientes estimados del modelo de regresión lineal para opciones CALL, entrenado sobre 15,645 observaciones. De las 14 variables especificadas en la fórmula del modelo, dos no fueron consideradas por presentar dependencia lineal perfecta. Concretamente, `Tasa_Libre_Riesgo` no fue incluida por ser constante en todas las observaciones (3.858 %), y `Vol_Diff` por constituir una combinación lineal exacta de `IV` y `Volatilidad_Historica`, ambas ya incluidas como predictores independientes. La variable categórica `Ticker`, con 10 niveles y `AAPL` como categoría de referencia, generó 9 variables *dummy*, de las cuales `TickerSPY` y `TickerTSLA` fueron igualmente descartadas por colinealidad. El modelo resultante estima 18 coeficientes efectivos más el intercepto (19 parámetros en total).

Tabla 4.5: Coeficientes estimados del modelo de regresión lineal para opciones CALL.

Variable	**Coef. ($\hat{\beta}$)**	**SE**	**t-valor**	**p-valor**	**Sig.**
(Intercepto)	27.5354	5.3531	5.14	<0.001	***
Strike	−0.1764	0.0065	−27.22	<0.001	***
Precio_Actual	0.1256	0.0093	13.55	<0.001	***
Dias_Vencimiento	−0.0139	0.0023	−6.01	<0.001	***
Volatilidad_Historica	6.1000	5.0506	1.21	0.2272	
Vol	−0.0000	0.0002	−0.07	0.9408	
OI	−0.0016	0.0001	−14.00	<0.001	***
IV	−11.8649	2.0652	−5.75	<0.001	***
Moneyness	1.2079	0.1087	11.11	<0.001	***
Log_Moneyness	−46.3126	2.0276	−22.84	<0.001	***
Strike_Normalizado	67.0763	1.4144	47.42	<0.001	***
In_The_Money	9.3143	1.5127	6.16	<0.001	***
Variables dummy de Ticker (referencia: AAPL)					
TickerAMD	9.7342	3.3932	2.87	0.0041	**
TickerAMZN	−6.6160	3.4522	−1.92	0.0553	.
TickerGOOGL	−0.6110	3.3104	−0.18	0.8536	
TickerMETA	21.1958	2.2253	9.52	<0.001	***
TickerMSFT	2.3919	2.2875	1.05	0.2957	
TickerNVDA	67.9408	2.8686	23.68	<0.001	***
TickerQQQ	−7.9410	1.6489	−4.82	<0.001	***

Códigos de significancia: *** p<0.001, ** p<0.01, * p<0.05, . p<0.1

R^2 = 0,4479, R^2 ajustado = 0,4473, Error estándar residual = $65,93

Los coeficientes revelan los principales factores del error de Black-Scholes en opciones CALL. **Strike_Normalizado** es la variable más significativa ($\hat{\beta} = 67{,}08,\ t = 47{,}42$): strikes más altos relativos al precio spot se asocian con mayores errores positivos de BS. **Log_Moneyness** presenta un efecto negativo pronunciado ($\hat{\beta} = -46{,}31,\ t = -22{,}84$), capturando la curvatura del *volatility smile* que el modelo BS no puede representar. **IV** muestra efecto negativo ($\hat{\beta} = -11{,}86,\ t = -5{,}75$): mayor volatilidad implícita se asocia con errores negativos (BS sobrestima). Entre los efectos por ticker, **NVDA** presenta el mayor efecto (+67,94, $t = 23{,}68$), reflejando la alta volatilidad característica de este activo. Volatilidad_Historica ($p = 0{,}227$) y Vol ($p = 0{,}941$) no son significativas. El modelo alcanza un $R^2 = 0{,}4479$, explicando el 44.79 % de la varianza del error con un error estándar residual de \$65.93. La Tabla 4.6 presenta las métricas de desempeño del modelo para opciones CALL.

Tabla 4.6: Desempeño del modelo de regresión lineal para opciones CALL.

Métrica	**Train**	**Test**	**Interpretación**
RMSE (\$)	65.89	73.15	Error cuadrático medio
MAE (\$)	28.47	30.71	Error absoluto medio
R^2	0.4479	0.4388	Varianza explicada (44 %)
R^2 ajustado	0.4473	—	Ajustado por # de variables
Comparación con Baseline (modelo constante):			
RMSE Baseline (\$)	—	97.66	
Mejora RMSE	—	25.10 %	(\$97.66 → \$73.15)
Mejora MAE	—	5.01 %	(\$32.33 → \$30.71)

El modelo alcanza un RMSE de \$73.15 en el conjunto de prueba, representando una mejora del 25.10 % respecto al baseline (\$97.66). La diferencia mínima entre métricas de entrenamiento y prueba (RMSE de \$65.89 vs \$73.15) indica ausencia de sobreajuste. Sin embargo, el $R^2 = 0{,}4388$ implica que el modelo deja el 56 % de la variabilidad sin explicar, sugiriendo relaciones no lineales e interacciones entre variables que la estructura lineal no captura. La aplicación práctica del modelo se evalúa mediante el precio ajustado $P_{LM} = P_{BS} + \hat{y}_i$. La Tabla 4.7 compara los errores antes y después de aplicar el ajuste.

Tabla 4.7: Comparación de errores en opciones CALL: BS original vs BS ajustado con regresión lineal.

Modelo	**RMSE (\$)**	**MAE (\$)**	**Mejora RMSE**
BS Original	98.42	25.96	—
BS + Regresión Lineal	73.15	30.71	25.68 %

Para ilustrar el funcionamiento del modelo, la Tabla 4.8 presenta casos representativos donde la

corrección redujo significativamente el error de valoración.

Tabla 4.8: Ejemplos de ajuste exitoso para opciones CALL

Ticker	Strike	S_0	Días	IV	Last	BS	$\hat{y}$	Ajustado	Error BS	Error Final
AMD	200	164.67	255	0.54	43.00	17.81	25.20	43.01	25.19	-0.01
SPY	850	669.21	802	0.16	16.06	38.77	-22.70	16.08	-22.71	-0.02
MSFT	460	517.35	255	0.33	87.91	83.67	4.25	87.92	4.24	-0.01
META	530	710.56	74	0.56	204.22	186.02	18.10	204.12	18.20	0.10
NVDA	1140	187.62	74	0.00	342.19	0.00	344.06	344.06	342.19	-1.87

Nota: S_0 = Precio actual del subyacente. Todos los valores monetarios en USD.

El caso de AMD ilustra una corrección casi perfecta: Black-Scholes subvalora la opción en $25.19, el modelo predice un ajuste de $25.20, y el error final es prácticamente cero (-$0.01). El ejemplo de NVDA es particularmente notable: se trata de una opción deep out-of-the-money donde Black-Scholes asigna precio cero, pero el mercado la cotiza en $342.19. El modelo captura este patrón y predice un ajuste de $344.06, reduciendo el error de $342.19 a solo -$1.87. Es importante notar que estas métricas agregadas no capturan completamente el comportamiento del modelo a nivel individual. El modelo mejora efectivamente la valoración en solo el 28.9 % de las opciones CALL, con casos donde la corrección puede empeorar el error original.

4.2.5. Resultados para opciones PUT

La Tabla 4.9 presenta los coeficientes estimados del modelo de regresión lineal para opciones PUT, entrenado sobre 14,720 observaciones. De la misma manera que en el modelo de regresión lineal para CALL, las variables `Tasa_Libre_Riesgo` y `Vol_Diff` no fueron consideradas por presentar dependencia lineal perfecta, así como las variables *dummy* `TickerSPY` y `TickerTSLA`, descartadas por colinealidad. El modelo resultante estima 18 coeficientes efectivos más el intercepto (19 parámetros en total).

Tabla 4.9: Coeficientes estimados del modelo de regresión lineal para opciones PUT.

Variable	**Coef. ($\hat{\beta}$)**	**SE**	**t-valor**	**p-valor**	**Sig.**
(Intercepto)	−31.9614	2.5595	−12.49	<0.001	***
Strike	0.1733	0.0036	48.45	<0.001	***
Precio_Actual	−0.1359	0.0046	−29.75	<0.001	***
Dias_Vencimiento	0.0260	0.0013	20.67	<0.001	***
Volatilidad_Historica	−36.8228	2.6211	−14.05	<0.001	***
Vol	0.0001	0.0001	1.00	0.3167	
OI	0.0001	0.0000	2.89	0.0038	**
IV	38.1323	1.3820	27.59	<0.001	***
Moneyness	0.5525	0.0514	10.74	<0.001	***

Continúa en la siguiente página

Tabla 4.9 – continuación

Variable	Coef. ($\hat{\beta}$)	SE	t-valor	p-valor	Sig.
Log_Moneyness	−37.1616	1.0185	−36.49	<0.001	***
Strike_Normalizado	−146.6886	0.6968	−210.51	<0.001	***
In_The_Money	12.2614	0.7654	16.02	<0.001	***
Variables dummy de Ticker (referencia: AAPL)					
TickerAMD	−2.4447	1.6792	−1.46	0.1455	
TickerAMZN	6.2599	1.6896	3.70	<0.001	***
TickerGOOGL	−0.9853	1.6202	−0.61	0.5431	
TickerMETA	−2.7089	1.0766	−2.52	0.0119	*
TickerMSFT	0.8757	1.1174	0.78	0.4332	
TickerNVDA	−25.1881	1.3840	−18.20	<0.001	***
TickerQQQ	2.1558	0.7835	2.75	0.0059	**

Códigos de significancia: *** p<0.001, ** p<0.01, * p<0.05, . p<0.1
R^2 = 0,9019, R^2 ajustado = 0,9018, Error estándar residual = $30,97

El modelo PUT presenta patrones notablemente diferentes al modelo CALL. **Strike_Normalizado** es nuevamente la variable más significativa, pero con efecto inverso y de mayor magnitud ($\hat{\beta} = -146{,}69$, $t = -210{,}51$), indicando que BS sobrestima sistemáticamente las opciones PUT con strikes elevados. **IV** muestra efecto opuesto al modelo CALL ($\hat{\beta} = 38{,}13$, $t = 27{,}59$): mayor volatilidad implícita implica errores positivos (BS subestima). Esta asimetría refleja el sesgo conocido del mercado de opciones (*put skew*). **Volatilidad_Historica**, no significativa para CALL, es aquí altamente significativa y negativa ($\hat{\beta} = -36{,}82$, $t = -14{,}05$), evidenciando poder predictivo diferenciado según el tipo de opción. Entre los tickers, **NVDA** presenta el efecto más extremo ($-25{,}19$, $t = -18{,}20$), indicando mayor sobrestimación de BS. Vol permanece no significativa ($p = 0{,}317$). La Tabla 4.10 presenta las métricas de desempeño del modelo para opciones PUT.

Tabla 4.10: Desempeño del modelo de regresión lineal para opciones PUT.

Métrica	Train	Test	Interpretación
RMSE ($)	30.95	31.43	Error cuadrático medio
MAE ($)	15.73	16.02	Error absoluto medio
R^2	0.9019	0.9078	Varianza explicada (90 %)
R^2 ajustado	0.9018	—	Ajustado por # de variables
Comparación con Baseline (modelo constante):			
RMSE Baseline ($)	—	103.54	
Mejora RMSE	—	69.65 %	($103.54 → $31.43)
Mejora MAE	—	52.97 %	($34.06 → $16.02)

El modelo PUT exhibe un desempeño sustancialmente superior al modelo CALL en todas las métricas.

El RMSE de $31.43 en prueba representa una mejora del 69.65 % respecto al baseline ($103.54), mientras que el R^2 = 0,9078 indica que el modelo explica más del 90 % de la variabilidad en los errores de Black-Scholes para opciones PUT. La consistencia entre entrenamiento y prueba (RMSE de $30.95 vs $31.43) confirma la ausencia de sobreajuste.

La Tabla 4.11 compara los errores antes y después de aplicar el ajuste para opciones PUT.

Tabla 4.11: Comparación de errores en opciones PUT: BS original vs BS ajustado con regresión lineal.

Modelo	**RMSE ($)**	**MAE ($)**	**Mejora RMSE**
BS Original	105.02	22.78	—
BS + Regresión Lineal	31.43	16.02	70.08 %

La Tabla 4.12 presenta casos representativos del ajuste para opciones PUT.

Tabla 4.12: Ejemplos de ajuste exitoso para opciones PUT

Ticker	Strike	S_0	Días	IV	Last	BS	$\hat{y}$	Ajustado	Error BS	Error Final
QQQ	434.78	603.18	438	0.28	10.56	4.11	6.47	10.58	6.45	-0.02
NVDA	1520	187.62	438	0.00	489.00	1263.62	-777.62	486.00	-774.62	3.00
AMD	175	164.67	193	0.52	13.65	28.46	-14.74	13.72	-14.81	-0.07
SPY	735	669.21	46	0.18	71.68	64.48	7.23	71.71	7.20	-0.03
AAPL	260	258.02	319	0.21	22.05	27.43	-5.42	22.02	-5.38	0.03

Nota: S_0 = Precio actual del subyacente. Todos los valores monetarios en USD.

El caso de NVDA PUT demuestra la capacidad del modelo para corregir sobrevaloraciones extremas de Black-Scholes. Esta opción deep in-the-money tiene un precio teórico de $1,263.62, pero el mercado la cotiza en $489.00. El modelo predice un ajuste negativo de -$777.62, resultando en un precio ajustado de $486.00 con error final de solo $3.00.

Al igual que en opciones CALL, el modelo mejora efectivamente la valoración en solo el 22.9 % de las opciones PUT, indicando que la corrección individual presenta limitaciones significativas a pesar de la mejora en métricas agregadas.

4.2.6. Rendimiento de los modelos

La Tabla 4.13 consolida los resultados de ambos modelos de regresión lineal, facilitando la comparación directa entre tipos de opciones.

Tabla 4.13: Resumen comparativo de los modelos de regresión lineal por tipo de opción.

Métrica	**CALL**	**PUT**
Configuración		
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
Número de predictores	18	18
Desempeño en prueba		
RMSE ($)	73.15	31.43
MAE ($)	30.71	16.02
R^2	0.4388	0.9078
R^2 ajustado (train)	0.4473	0.9018
Comparación con baseline		
RMSE Baseline ($)	97.66	103.54
Mejora RMSE vs Baseline	25.10 %	69.65 %
Mejora MAE vs Baseline	5.01 %	52.97 %
Efectividad en ajuste de precios		
RMSE BS Original ($)	98.42	105.02
RMSE BS + Reg. Lineal ($)	73.15	31.43
Mejora en precio	25.68 %	70.08 %
Opciones mejoradas	28.9 %	22.9 %
Variable más significativa		
Top predictor	Strike_Norm ($t = 47{,}42$)	Strike_Norm ($t = -210{,}51$)

El análisis comparativo revela varios hallazgos relevantes:

Desempeño asimétrico por tipo de opción: El modelo PUT alcanza un $R^2 = 0{,}9078$, explicando más del 90 % de la varianza del error de Black-Scholes, mientras que el modelo CALL logra un $R^2 = 0{,}4388$. Esta diferencia sustancial sugiere que los errores de valoración en opciones CALL tienen componentes más difíciles de capturar mediante relaciones lineales, posiblemente debido a mayor heterogeneidad en el comportamiento especulativo de estos instrumentos.

Patrones opuestos en coeficientes clave: Los coeficientes revelan asimetrías fundamentales entre tipos de opciones. Strike_Normalizado presenta efectos opuestos (CALL: +67,08; PUT: −146,69), al igual que IV (CALL: −11,86; PUT: +38,13). Volatilidad_Historica es no significativa para CALL pero altamente significativa para PUT ($\hat{\beta} = -36{,}82$). Estas divergencias confirman la decisión de entrenar

modelos separados y reflejan mecanismos de error estructuralmente diferentes.

Efectos por Ticker: NVDA muestra los efectos más extremos: +67,94 para CALL (mayor subestimación de BS) y −25,19 para PUT (mayor sobrestimación), reflejando la alta volatilidad característica de este activo tecnológico. META también presenta efectos significativos pero en direcciones opuestas según el tipo de opción. Vol (volumen de negociación) no es significativa en ninguno de los dos modelos ($p > 0{,}3$), indicando que la liquidez intradía no contribuye a explicar el error de BS.

Mejora sustancial vs baseline: Ambos modelos superan significativamente al baseline. El modelo PUT reduce el RMSE en 69.65 % ($103.54 → $31.43), mientras que el modelo CALL lo reduce en 25.10 % ($97.66 → $73.15). Esto demuestra que incorporar características de las opciones aporta valor predictivo considerable, incluso bajo la restricción de linealidad.

Limitaciones del modelo lineal: A pesar de las mejoras observadas en métricas agregadas, el modelo presenta limitaciones importantes que justifican la exploración de alternativas no lineales:

1. **Bajo poder explicativo para CALL:** El $R^2 = 0{,}44$ implica que el 56 % de la variabilidad permanece sin explicar, sugiriendo relaciones no lineales e interacciones entre variables no capturadas.

2. **Violación de supuestos clásicos:** Los diagnósticos revelan heterocedasticidad severa (BP = 3,107 para CALL, 7,599 para PUT; $p < 0{,}001$), no normalidad de residuos (Anderson-Darling > 390; $p < 0{,}001$) y patrones no lineales en los gráficos de residuos. Aunque los coeficientes permanecen insesgados, los errores estándar pueden estar subestimados.

3. **Riesgo de deterioro en casos individuales:** El modelo mejora la valoración en solo el 28.9 % (CALL) y 22.9 % (PUT) de los casos individuales. Para opciones CALL, el error promedio aumenta en $4.75 tras el ajuste, indicando que el modelo introduce errores de mayor magnitud cuando falla.

4. **Asunción de linealidad:** El modelo asume relaciones lineales aditivas entre variables explicativas y el error. Sin embargo, fenómenos como el *volatility smile* documentado en la literatura financiera implican relaciones no lineales entre moneyness, volatilidad implícita y errores de valoración que no pueden capturarse sin transformaciones adicionales.

5. **Ausencia de interacciones automáticas:** Efectos de interacción entre variables (por ejemplo, el impacto de la volatilidad implícita podría variar según el nivel de moneyness) deben especificarse manualmente. El modelo no detecta ni incorpora estas interacciones de forma automática.

Estas limitaciones, particularmente el bajo poder explicativo para opciones CALL y las violaciones sistemáticas de supuestos, motivan la exploración de técnicas de machine learning no lineal en las

secciones subsecuentes.

4.3. Random Forest

4.3.1. Proceso de modelado

La Figura 4.6 presenta el flujo metodológico del proceso de modelado mediante Random Forest, estructurado en cinco fases secuenciales.

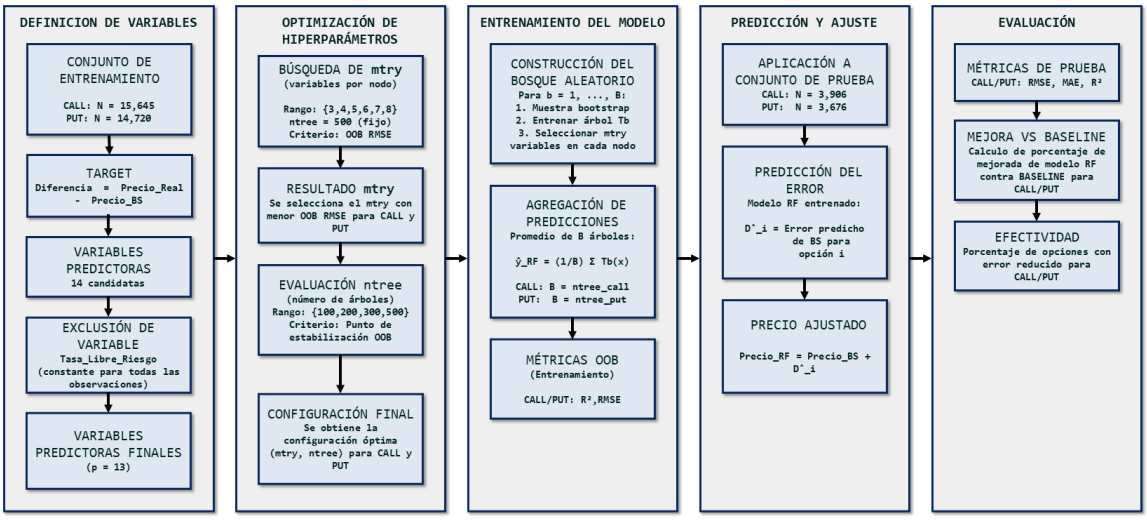


Figura 4.6: Diagrama del proceso de modelado mediante Random Forest para la predicción del error de Black-Scholes. Fuente: Elaboración propia.

El proceso se estructura en cinco fases principales:

1. **Definición de Variables:** Se establece la variable objetivo (Diferencia = Precio_Real − Precio_BS) y se identifican las 14 variables predictoras candidatas. La variable Tasa_Libre_Riesgo se excluye por ser constante (3.858 %) en todas las observaciones, resultando en 13 predictores finales.

2. **Optimización de Hiperparámetros:** Evaluación sistemática del parámetro `mtry` (número de variables por nodo) mediante el criterio de error Out-of-Bag, seguido de la determinación del número óptimo de árboles (`ntree`) basándose en el punto de estabilización del error OOB.

3. **Entrenamiento del Modelo:** Construcción de los bosques aleatorios mediante el proceso iterativo de muestreo bootstrap, entrenamiento de árboles individuales y agregación de predicciones según la fórmula $\hat{y}_{RF} = (1/B)\sum T_b(x)$.

4. **Predicción y Ajuste:** Aplicación de los modelos entrenados al conjunto de prueba para obtener los errores predichos ($\hat{D}_i$) y cálculo de los precios ajustados como Precio_RF = Precio_BS + $\hat{D}_i$.

5. **Evaluación:** Cálculo de métricas de desempeño (RMSE, MAE, R^2) sobre el conjunto de prueba, comparación de resultados contra el baseline y determinación del porcentaje de opciones con error reducido.

4.3.2. Configuración e hiperparámetros

La implementación de Random Forest requiere la especificación de dos hiperparámetros principales que controlan la estructura y comportamiento del modelo: el número de árboles en el bosque (`ntree`) y el número de variables consideradas en cada división (`mtry`). La selección adecuada de estos parámetros permite maximizar la capacidad del modelo para capturar los patrones de error de Black-Scholes.

El parámetro `ntree` determina cuántos árboles de decisión componen el bosque. Un número mayor de árboles generalmente mejora la estabilidad de las predicciones al promediar el ruido individual de cada árbol. Sin embargo, cada árbol adicional incrementa el tiempo de entrenamiento y el uso de memoria, por lo que se analiza la evolución del error OOB en función de `ntree` para seleccionar un valor a partir del cual agregar más árboles no reduce significativamente el error.

El parámetro `mtry` controla la aleatoriedad del modelo al especificar cuántas variables se evalúan como candidatas en cada nodo de división. Para problemas de regresión, la recomendación teórica establece $m = p/3$, donde p es el número total de predictores [16]. Sin embargo, esta heurística puede no ser óptima para todos los conjuntos de datos, por lo que se realizó una búsqueda sistemática sobre un rango de valores.

El proceso de optimización se ejecutó en dos etapas secuenciales. En la primera etapa, se evaluaron diferentes valores de `mtry` manteniendo fijo `ntree` = 500 para garantizar convergencia del error OOB. Los valores evaluados fueron `mtry` $\in \{3,4,5,6,7,8\}$, cubriendo desde configuraciones más aleatorias (mtry bajo) hasta configuraciones más determinísticas (mtry alto). En la segunda etapa, una vez identificado el `mtry` óptimo, se evaluó la convergencia del error OOB para diferentes valores de `ntree` $\in \{100,200,300,500\}$ para determinar el número mínimo de árboles necesario.

La Tabla 4.14 presenta los resultados de la optimización del parámetro `mtry` para ambos tipos de opciones, mostrando el error OOB (RMSE) obtenido con cada configuración.

Tabla 4.14: Optimización del hiperparámetro `mtry` mediante error Out-of-Bag.

mtry	OOB RMSE - CALL ($)	OOB RMSE - PUT ($)
3	16.51	11.05
4	16.31	10.76
5	16.21	10.70
6	16.17	10.77
7	16.22	10.68
8	**16.15**	**10.57**

Para ambos tipos de opciones, el error OOB mínimo se alcanzó con `mtry` = 8, lo que representa aproximadamente el 62 % de los 13 predictores disponibles. Este valor supera la recomendación teórica de $p/3 \approx 4$, sugiriendo que el problema requiere mayor información en cada división para capturar adecuadamente los patrones de error. Se evaluó la convergencia del error Out-of-Bag (OOB) en función del número de árboles (`ntree`). Dado que este parámetro no posee una regla teórica de determinación, su selección se realizó de forma empírica, identificando el punto a partir del cual el error se estabiliza. El modelo CALL alcanzó convergencia con 200 árboles, mientras que el modelo PUT requirió 500 árboles. Los demás hiperparámetros se mantuvieron en sus valores por defecto, como `nodesize` = 5 que es el tamaño mínimo de nodo terminal. Adicionalmente, se habilitó `importance` = TRUE para cuantificar la contribución de cada predictor.

La Tabla 4.15 resume la configuración final seleccionada para cada modelo, incluyendo los hiperparámetros optimizados y las características del conjunto de datos utilizado.

Tabla 4.15: Configuración final de los modelos Random Forest por tipo de opción.

Parámetro	CALL	PUT
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
Número de predictores (p)	13	13
`ntree` (árboles)	200	500
`mtry` (variables/nodo)	8	8
`nodesize` (tamaño mínimo de nodo)	5	5
`importance`	TRUE	TRUE

Es notable que el modelo para opciones PUT requiere una configuración más compleja en términos del número de árboles: 500 frente a 200 para CALL. Esta diferencia sugiere que los patrones de error de Black-Scholes en opciones PUT presentan mayor complejidad estructural, requiriendo mayor

capacidad del modelo para su captura efectiva.

Las 13 variables predictoras utilizadas en ambos modelos incluyen las 12 variables numéricas definidas en el Capítulo 3 (Strike, Precio_Actual, Dias_Vencimiento, Volatilidad_Historica, IV, Vol, OI, Moneyness, Log_Moneyness, Vol_Diff, Strike_Normalizado e In_The_Money) más la variable categórica Ticker, que identifica el activo subyacente y permite al modelo capturar efectos específicos por emisor.

4.3.3. Resultados para opciones CALL

El modelo Random Forest para opciones CALL se entrenó sobre 15,645 observaciones utilizando la configuración optimizada (`ntree` = 200, `mtry` = 8). La Figura 4.7 ilustra la evolución del error OOB conforme se incrementa el número de árboles, evidenciando la estabilización del error aproximadamente a partir de 100 árboles, lo que valida la selección de 200 árboles como configuración final.

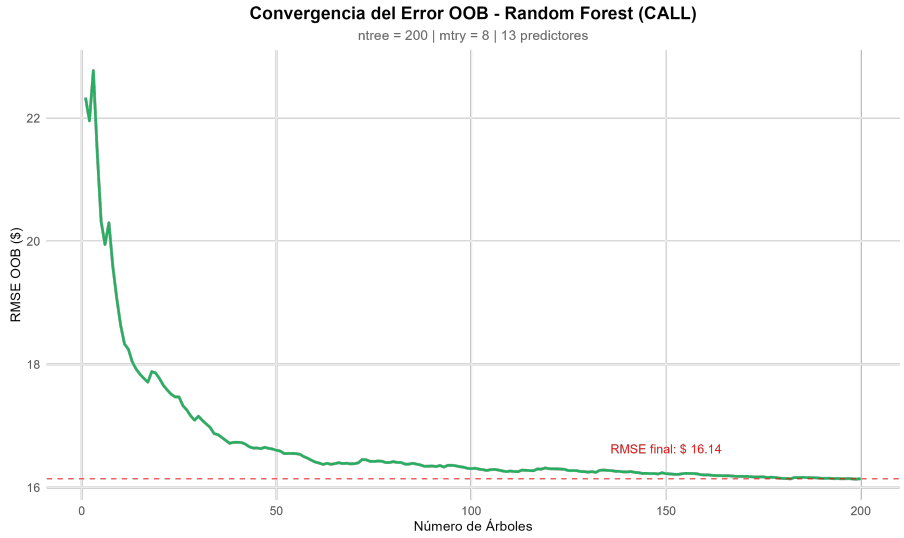


Figura 4.7: Convergencia del error OOB en función del número de árboles para opciones CALL. La estabilización temprana justifica el uso de 200 árboles.

La Tabla 4.16 presenta las métricas de desempeño del modelo evaluadas tanto en el conjunto de entrenamiento como en el conjunto de prueba.

Tabla 4.16: Métricas de desempeño del modelo Random Forest para opciones CALL.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	7.69	2.54	0.9925
Prueba	17.38	5.79	0.9683

El modelo alcanza un RMSE de $17.38 en el conjunto de prueba, representando una reducción del 82.20 % respecto al baseline ($97.66). El coeficiente de determinación $R^2 = 0.9683$ indica que el modelo

explica el 96.83 % de la variabilidad en los errores de Black-Scholes para opciones CALL. La diferencia entre las métricas de entrenamiento y prueba sugiere cierto grado de sobreajuste.
La Figura 4.8 presenta el diagrama de dispersión entre los errores predichos y los errores reales en el conjunto de prueba. La concentración de puntos alrededor de la línea de identidad confirma la capacidad del modelo para capturar la estructura de los errores de Black-Scholes.

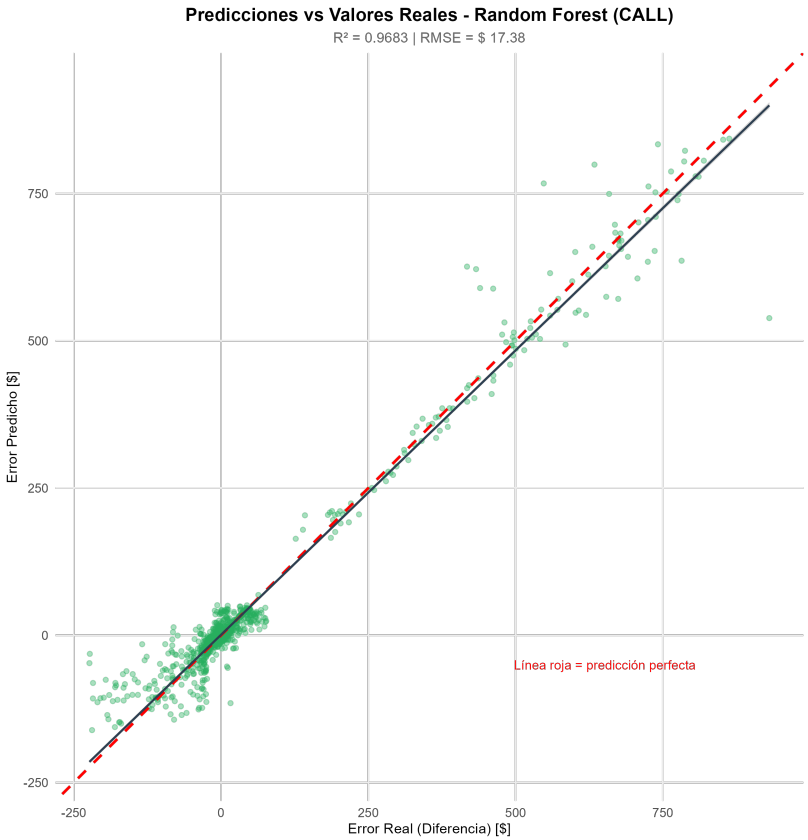


Figura 4.8: Errores predichos vs. errores reales para opciones CALL en el conjunto de prueba.

La Tabla 4.17 compara el desempeño del modelo Random Forest contra los modelos previos, cuantificando la mejora incremental en cada etapa de la progresión metodológica.

Tabla 4.17: Comparación de desempeño: Random Forest vs. modelos anteriores para opciones CALL.

Modelo	**RMSE ($)**	**MAE ($)**	**Mejora vs Baseline**
BS Original	98.42	25.96	—
Baseline (C = +$10.28)	97.66	32.33	0.77 %
Regresión Lineal	73.15	30.71	25.09 %
Random Forest	**17.38**	**5.79**	**82.20 %**

Random Forest representa una mejora respecto a los modelos anteriores. La reducción de RMSE desde $73.15 (regresión lineal) hasta $17.38 evidencia la capacidad del modelo para capturar las no linealidades e interacciones que el modelo lineal no podía representar.

La aplicación práctica del modelo se evalúa mediante el precio ajustado $P_{RF} = P_{BS} + \hat{D}_{RF}$. La Tabla 4.18 presenta la efectividad de esta corrección.

Tabla 4.18: Efectividad del ajuste de precios mediante Random Forest para opciones CALL.

Métrica	**Precio BS**	**Precio RF**
RMSE vs. Precio Real ($)	98.42	17.38
Mejora	—	82.34 %
Opciones con error reducido	—	76.9 %

El modelo mejora la precisión de valoración en el 76.9 % de las opciones CALL del conjunto de prueba, con una reducción global del error de 82.34 %. Este resultado demuestra la efectividad del enfoque de corrección mediante predicción de errores.

Para evaluar la contribución de cada predictor, Random Forest proporciona el incremento porcentual en el error cuadrático medio (%IncMSE) cuando los valores de una variable son permutados aleatoriamente. La Tabla 4.19 presenta las cinco variables más importantes para opciones CALL.

Tabla 4.19: Top 5 variables más importantes según %IncMSE para opciones CALL.

Rank	**Variable**	**Importancia (%)**
1	Dias_Vencimiento	100.0
2	IV	53.2
3	OI	23.9
4	Strike_Normalizado	20.5
5	Ticker	17.3

La variable `Dias_Vencimiento` emerge como el predictor dominante (100 %), seguida por la volatilidad implícita IV (53.2 %). Esto tiene sentido teórico: el supuesto de volatilidad constante de Black-Scholes se vuelve más problemático conforme aumenta el horizonte temporal, mientras que las discrepancias entre la volatilidad implícita de mercado y la volatilidad histórica constituyen una fuente fundamental de error en opciones de compra. La Figura 4.9 visualiza la importancia de las 13 variables.

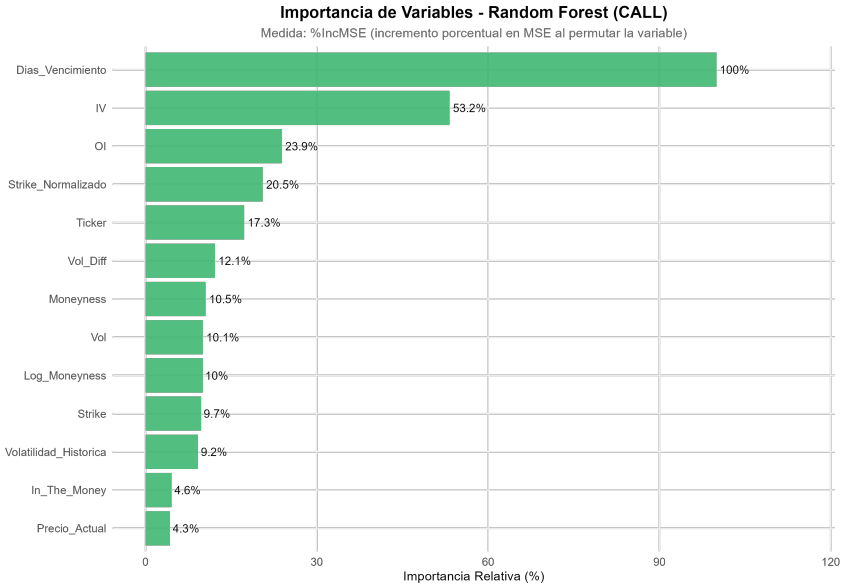


Figura 4.9: Importancia de variables (%IncMSE) para el modelo Random Forest de opciones CALL.

4.3.4. Resultados para opciones PUT

El modelo Random Forest para opciones PUT se entrenó sobre 14,720 observaciones con la configuración optimizada (`ntree` = 500, `mtry` = 8). La mayor complejidad de configuración respecto al modelo CALL refleja la estructura más intrincada de los errores de Black-Scholes en este tipo de opciones. La Figura 4.10 muestra la convergencia del error OOB, donde se observa que la estabilización requiere aproximadamente 300 árboles, justificando la selección de 500 árboles para garantizar convergencia completa.

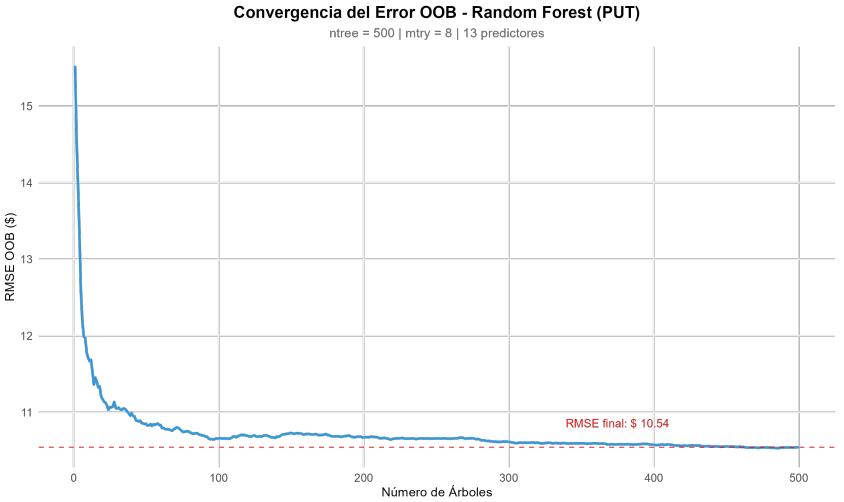


Figura 4.10: Convergencia del error OOB en función del número de árboles para opciones PUT. La estabilización más lenta justifica el uso de 500 árboles.

La Tabla 4.20 presenta las métricas de desempeño del modelo para opciones PUT.

Tabla 4.20: Métricas de desempeño del modelo Random Forest para opciones PUT.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	4.74	1.13	0.9977
Prueba	11.06	2.64	0.9886

El modelo PUT exhibe un desempeño superior al modelo CALL en todas las métricas. El RMSE de \$11.06 representa una reducción del 89.32 % respecto al baseline (\$103.54), mientras que el R^2 = 0.9886 indica que el modelo explica el 98.86 % de la variabilidad en los errores. Este resultado sugiere que, aunque los errores de Black-Scholes en opciones PUT son más complejos, también presentan patrones que Random Forest puede capturar efectivamente. La Figura 4.11 presenta el diagrama de dispersión para opciones PUT, donde se observa una concentración más estrecha alrededor de la línea de identidad comparada con opciones CALL.

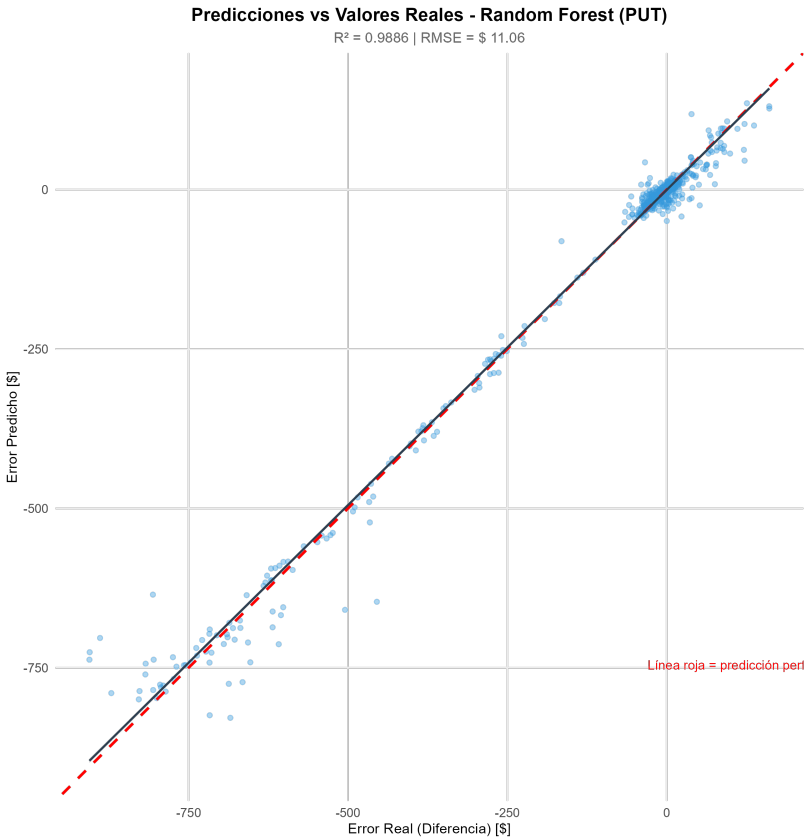


Figura 4.11: Errores predichos vs. errores reales para opciones PUT en el conjunto de prueba.

La Tabla 4.21 presenta la comparación con los modelos anteriores para opciones PUT.

Tabla 4.21: Comparación de desempeño: Random Forest vs. modelos anteriores para opciones PUT.

Modelo	RMSE ($)	MAE ($)	Mejora vs Baseline
BS Original	105.02	22.78	—
Baseline (C = -$16.34)	103.54	34.06	1.41 %
Regresión Lineal	31.43	16.02	69.64 %
Random Forest	**11.06**	**2.64**	**89.32 %**

La mejora de Random Forest sobre regresión lineal es aún más pronunciada en opciones PUT: el RMSE se reduce de $31.43 a $11.06, una disminución del 64.8 % adicional. Este resultado confirma que los errores de Black-Scholes en opciones PUT contienen patrones no lineales significativos que el modelo lineal no podía capturar. La Tabla 4.22 evalúa la efectividad del ajuste de precios para opciones PUT.

Tabla 4.22: Efectividad del ajuste de precios mediante Random Forest para opciones PUT.

Métrica	Precio BS	Precio RF
RMSE vs. Precio Real ($)	105.02	11.06
Mejora	—	89.47 %
Opciones con error reducido	—	89.1 %

El modelo mejora la precisión de valoración en el 89.1 % de las opciones PUT, superando significativamente el 76.9 % alcanzado para opciones CALL. Este resultado sugiere que Random Forest es particularmente efectivo para corregir las sobrestimaciones sistemáticas de Black-Scholes en opciones PUT. La Tabla 4.23 presenta las cinco variables más importantes para opciones PUT según el criterio %IncMSE.

Tabla 4.23: Top 5 variables más importantes según %IncMSE para opciones PUT.

Rank	Variable	Importancia (%)
1	Dias_Vencimiento	100.0
2	OI	72.9
3	Ticker	35.6
4	Volatilidad_Historica	32.8
5	Strike	31.3

Al igual que en CALL, `Dias_Vencimiento` domina con el 100 % de importancia. Sin embargo, los perfiles difieren notablemente: OI es segunda para PUT (72.9 %) pero tercera para CALL (23.9 %), mientras que IV es segunda para CALL (53.2 %) pero presenta apenas 9.1 % de importancia en PUT. Esta divergencia corrobora la decisión de entrenar modelos separados por tipo de opción. La Figura

4.12 visualiza la importancia de las 13 variables.

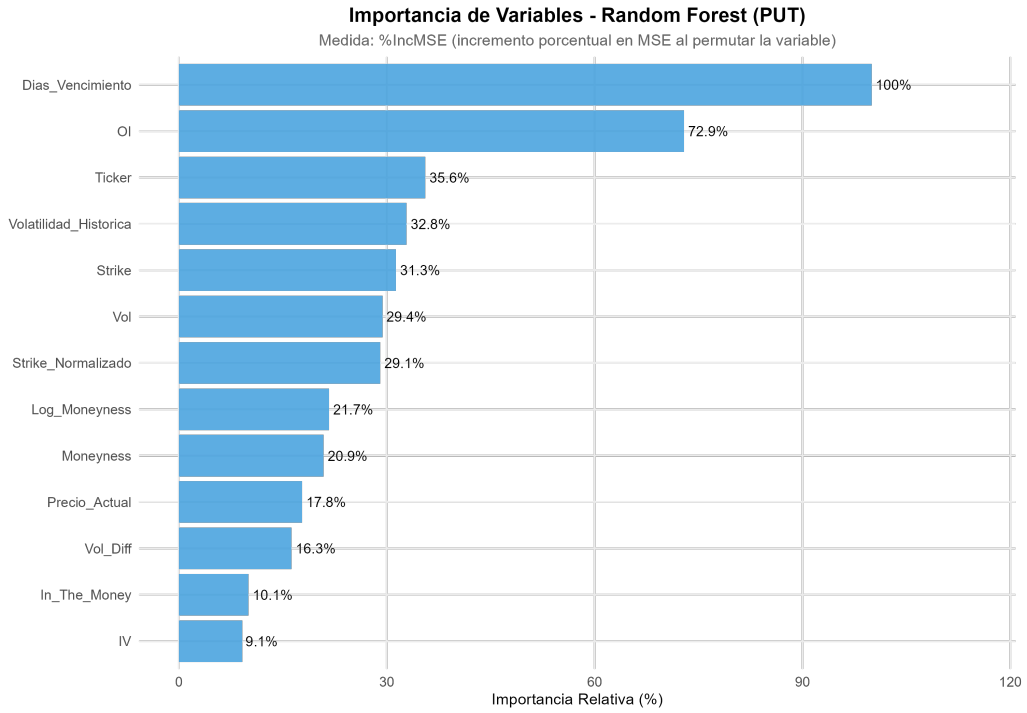


Figura 4.12: Importancia de variables (%IncMSE) para el modelo Random Forest de opciones PUT.

4.3.5. Rendimiento de los modelos

La Tabla 4.24 consolida los resultados de ambos modelos Random Forest, facilitando la comparación directa entre tipos de opciones.

Tabla 4.24: Resumen comparativo de los modelos Random Forest por tipo de opción.

Métrica	**CALL**	**PUT**
Configuración		
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
`ntree` óptimo	200	500
`mtry` óptimo	8	8
Desempeño en prueba		
RMSE ($)	17.38	11.06
MAE ($)	5.79	2.64

Continúa en la siguiente página

Tabla 4.24 – continuación

Métrica	**CALL**	**PUT**
R^2	0.9683	0.9886
Comparación con baseline		
RMSE Baseline ($)	97.66	103.54
Mejora vs Baseline	82.20 %	89.32 %
Efectividad en ajuste de precios		
RMSE Precio BS ($)	98.42	105.02
RMSE Precio RF ($)	17.38	11.06
Mejora en precio	82.34 %	89.47 %
Opciones mejoradas	76.9 %	89.1 %
Variable más importante		
Top predictor	Dias_Vencimiento (100 %)	Dias_Vencimiento (100 %)

El análisis comparativo de los modelos Random Forest revela un desempeño asimétrico, donde el modelo orientado a opciones PUT superó significativamente al de las CALL en todas las métricas evaluadas. Mientras que para las opciones PUT se alcanzó un R^2 de 0,9886 y un RMSE de $11,06, el modelo para CALL obtuvo un R^2 de 0,9683 y un RMSE de $17,38. Esta diferencia sugiere que, si bien las opciones PUT requirieron una configuración más compleja —500 árboles frente a 200 en CALL—, sus errores de valoración presentan patrones más regulares y predecibles para el algoritmo.

En términos de impacto, Random Forest representó un salto cualitativo sobre la regresión lineal, logrando mejoras adicionales en el RMSE del 76,2 % para CALL y del 64,8 % para PUT. Asimismo, los perfiles de importancia de variables confirmaron mecanismos de error divergentes: para las opciones CALL, la volatilidad implícita (IV) fue el segundo predictor más relevante (53,2 %), mientras que para las PUT, el interés abierto (OI) ocupó esa posición (72,9 %). Estos resultados validan la efectividad de los modelos no lineales para capturar las ineficiencias del modelo Black-Scholes y justifican la decisión metodológica de segmentar el análisis por tipo de contrato.

4.4. XGBoost

4.4.1. Proceso de modelado

La Figura 4.13 presenta el flujo metodológico del proceso de modelado mediante XGBoost, desde la preparación de datos hasta la evaluación del desempeño predictivo.

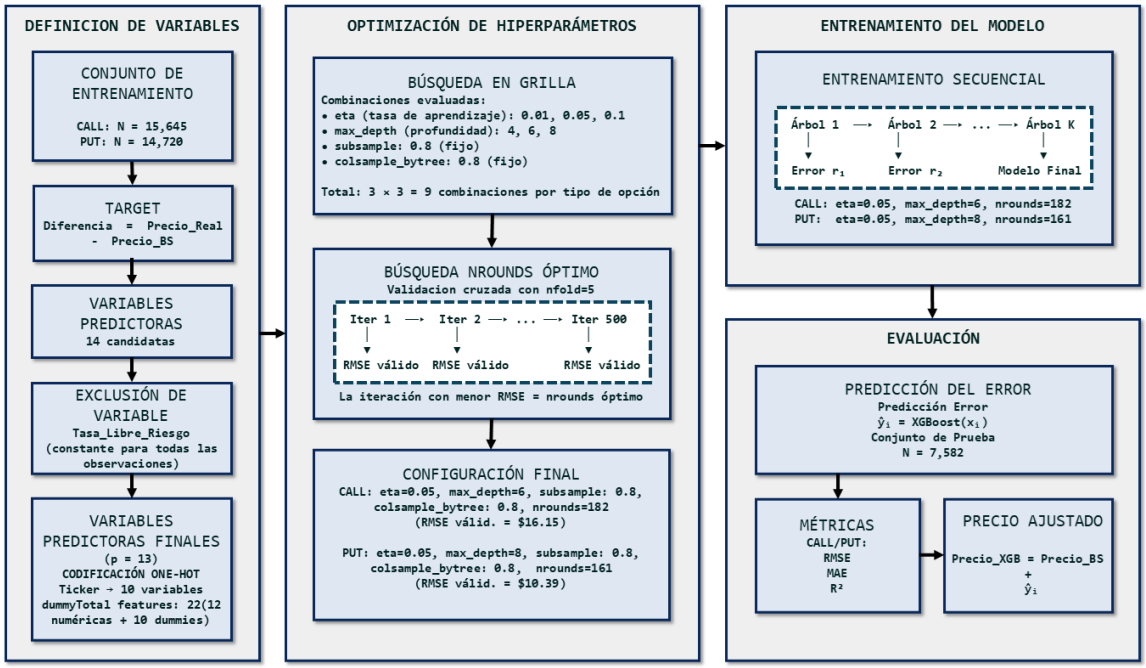


Figura 4.13: Diagrama del proceso de modelado mediante XGBoost para la predicción del error de Black-Scholes. Fuente: Elaboración propia.

El proceso se estructura en cuatro fases principales:

1. **Definición de Variables**: Preparación del conjunto de entrenamiento (30,365 observaciones), definición de la variable objetivo (Diferencia) y codificación de variables categóricas (Ticker) mediante one-hot encoding.

2. **Optimización**: Búsqueda de hiperparámetros óptimos mediante validación cruzada con early stopping, evaluando combinaciones de tasa de aprendizaje, profundidad máxima y tasas de submuestreo.

3. **Entrenamiento**: Construcción del modelo final con los hiperparámetros seleccionados, entrenando árboles secuencialmente hasta alcanzar el número óptimo de iteraciones.

4. **Evaluación**: Cálculo de métricas de desempeño (RMSE, MAE, R^2) sobre el conjunto de prueba y generación del precio ajustado como Precio_BS + $\hat{y}_i$.

4.4.2. Configuración e hiperparámetros

La implementación de XGBoost requiere la especificación de múltiples hiperparámetros que controlan tanto la estructura de los árboles individuales como el proceso de boosting. Los parámetros principales incluyen: `eta` (tasa de aprendizaje), `max_depth` (profundidad máxima de cada árbol), `subsample` (fracción de observaciones por árbol) y `colsample_bytree` (fracción de variables por árbol).

Para cada combinación de hiperparámetros, se entrenó el modelo mediante validación cruzada de 5 pliegues, deteniendo el entrenamiento automáticamente cuando el error de validación no mejorarba (early stopping). Este procedimiento determinó simultáneamente la configuración óptima y el número de iteraciones (`nrounds`) requerido. La Tabla 4.25 presenta los resultados de la búsqueda para opciones CALL.

Tabla 4.25: Búsqueda de hiperparámetros mediante validación cruzada para opciones CALL.

eta	max_depth	subsample	colsample	Best RMSE ($)	Best Iter
0.01	4	0.8	0.8	17.08	500
0.05	4	0.8	0.8	16.32	332
0.10	4	0.8	0.8	17.13	123
0.01	6	0.8	0.8	16.51	494
0.05	**6**	**0.8**	**0.8**	**16.15**	**132**
0.10	6	0.8	0.8	17.33	59
0.01	8	0.8	0.8	16.71	495
0.05	8	0.8	0.8	16.28	101
0.10	8	0.8	0.8	16.25	44

Para opciones CALL, la configuración óptima corresponde a `eta = 0.05`, `max_depth = 6`, con un RMSE de validación de $16.15 alcanzado en 132 iteraciones. La Tabla 4.26 presenta los resultados para opciones PUT.

Tabla 4.26: Búsqueda de hiperparámetros mediante validación cruzada para opciones PUT.

eta	max_depth	subsample	colsample	Best RMSE ($)	Best Iter
0.01	4	0.8	0.8	12.07	500
0.05	4	0.8	0.8	11.87	229
0.10	4	0.8	0.8	11.46	107
0.01	6	0.8	0.8	10.71	500
0.05	6	0.8	0.8	11.27	123
0.10	6	0.8	0.8	10.93	57
0.01	8	0.8	0.8	10.47	500
0.05	**8**	**0.8**	**0.8**	**10.39**	**111**
0.10	8	0.8	0.8	11.18	51

Para opciones PUT, el óptimo se alcanza con `eta = 0.05`, `max_depth = 8`, logrando un RMSE de $10.39 en 111 iteraciones, inferior al obtenido para CALL. La Tabla 4.27 resume la configuración final seleccionada para cada modelo.

Tabla 4.27: Configuración final de los modelos XGBoost por tipo de opción.

Parámetro	CALL	PUT
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
Número de features	22	22
eta (learning rate)	0.05	0.05
max_depth	6	8
subsample	0.8	0.8
colsample_bytree	0.8	0.8
nrounds (iteraciones)	182	161

Las 22 variables predictoras corresponden a las 12 variables numéricas utilizadas en Random Forest más las 10 variables dummy generadas por la codificación one-hot del ticker (con AAPL como categoría de referencia implícita en el modelo).

4.4.3. Resultados para opciones CALL

El modelo XGBoost para opciones CALL se entrenó sobre 15,645 observaciones utilizando la configuración optimizada (`eta = 0.05`, `max_depth = 6`, `nrounds = 182`). La Figura 4.14 ilustra la evolución del error durante el entrenamiento, mostrando las curvas de aprendizaje para los conjuntos de entrenamiento y prueba.

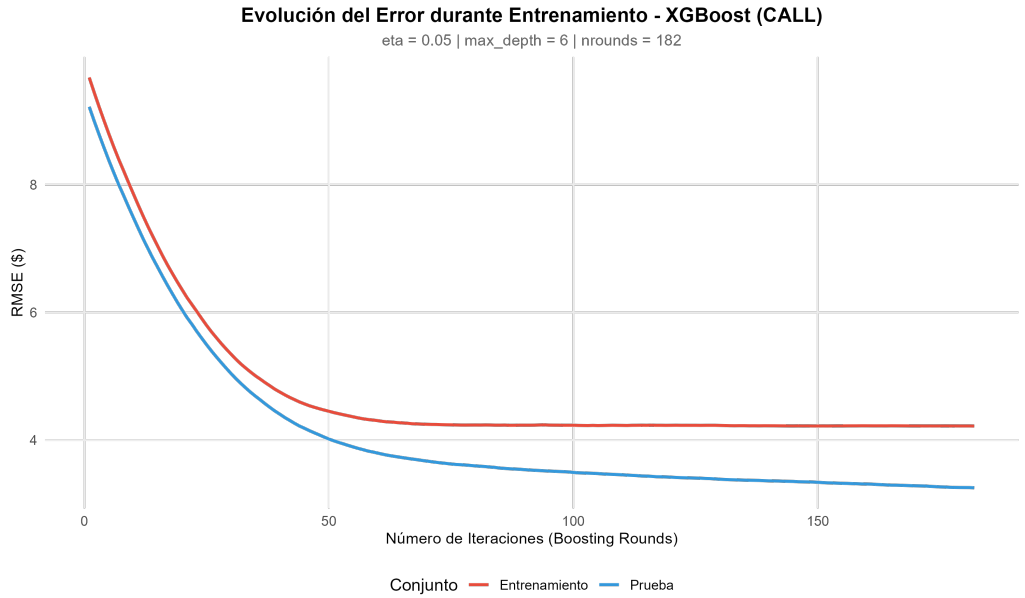


Figura 4.14: Evolución del RMSE durante el entrenamiento de XGBoost para opciones CALL.

Ambas curvas descienden de manera similar durante las primeras 50 iteraciones, tras lo cual el error de entrenamiento continúa disminuyendo gradualmente mientras el error de prueba se estabiliza. La Tabla 4.28 presenta las métricas de desempeño del modelo.

Tabla 4.28: Métricas de desempeño del modelo XGBoost para opciones CALL.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	10.57	4.87	0.9858
Prueba	17.80	6.34	0.9668

El modelo alcanza un RMSE de $17.80 en el conjunto de prueba, representando una mejora del 81.77 % respecto al baseline ($97.66). El coeficiente de determinación $R^2 = 0{,}9668$ indica que el modelo explica el 96.68 % de la variabilidad en los errores de Black-Scholes para opciones CALL. La Figura 4.15 presenta el diagrama de dispersión entre errores predichos y errores reales en el conjunto de prueba.

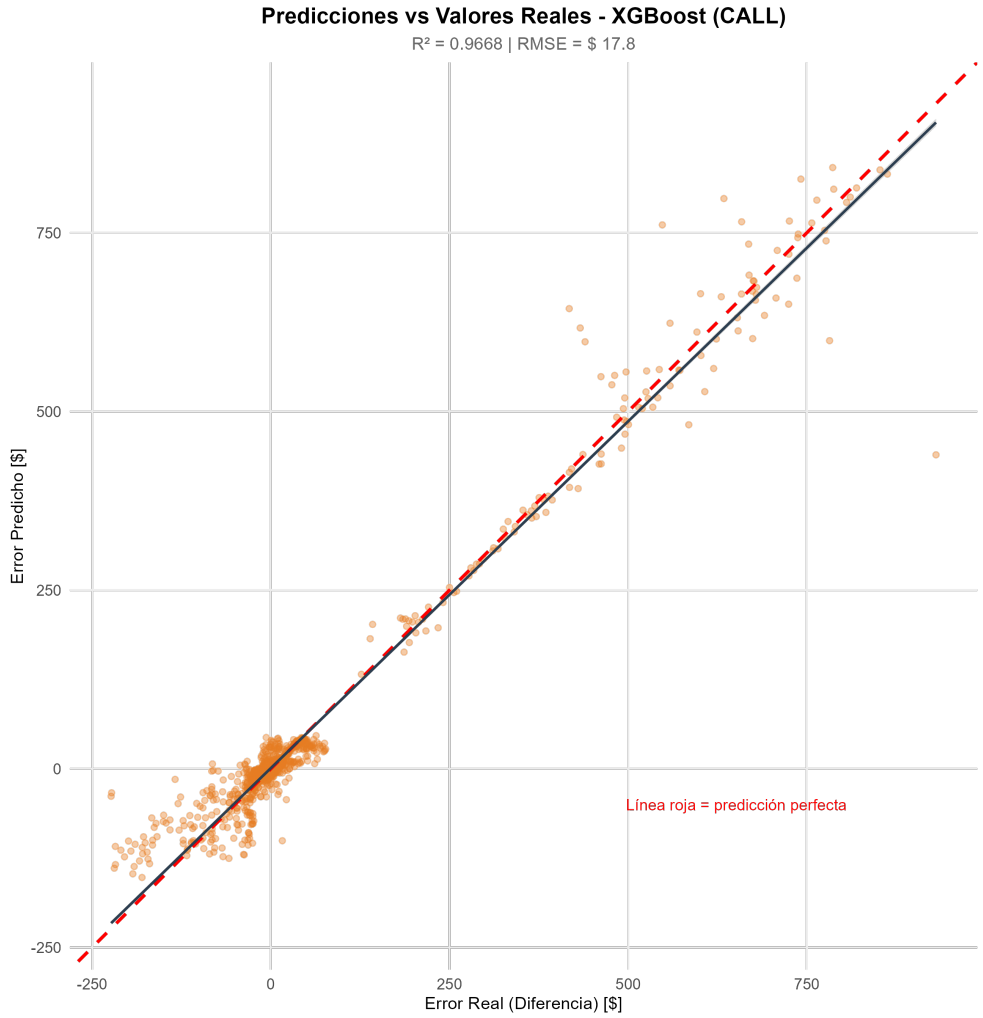


Figura 4.15: Errores predichos vs. errores reales para opciones CALL en el conjunto de prueba.

La concentración de puntos alrededor de la línea de identidad confirma la capacidad del modelo para capturar la estructura de los errores de Black-Scholes, aunque con mayor dispersión en errores de gran magnitud. La Tabla 4.29 compara el desempeño de XGBoost contra los modelos previos.

Tabla 4.29: Comparación de desempeño: XGBoost vs. modelos anteriores para opciones CALL.

Modelo	RMSE ($)	MAE ($)	Mejora vs Baseline
BS Original	98.42	25.96	—
Baseline (C = +$10,28)	97.66	32.33	—
Regresión Lineal	73.15	30.71	25.1 %
Random Forest	17.38	5.79	82.2 %
XGBoost	17.80	6.34	81.77 %

Los resultados revelan que, para opciones CALL, XGBoost presenta un desempeño ligeramente inferior a Random Forest (RMSE de $17.80 vs $17.38, diferencia de $0.42). Esta diferencia representa una reducción del 2.41 % en la mejora respecto al baseline. El modelo mejora la precisión de valoración en el 68.8 % de las opciones CALL del conjunto de prueba. La Figura 4.16 visualiza la reducción progresiva del RMSE desde el modelo Baseline ($97.66) hasta XGBoost ($17.80), evidenciando la mejora obtenida con cada incremento en complejidad del modelo.

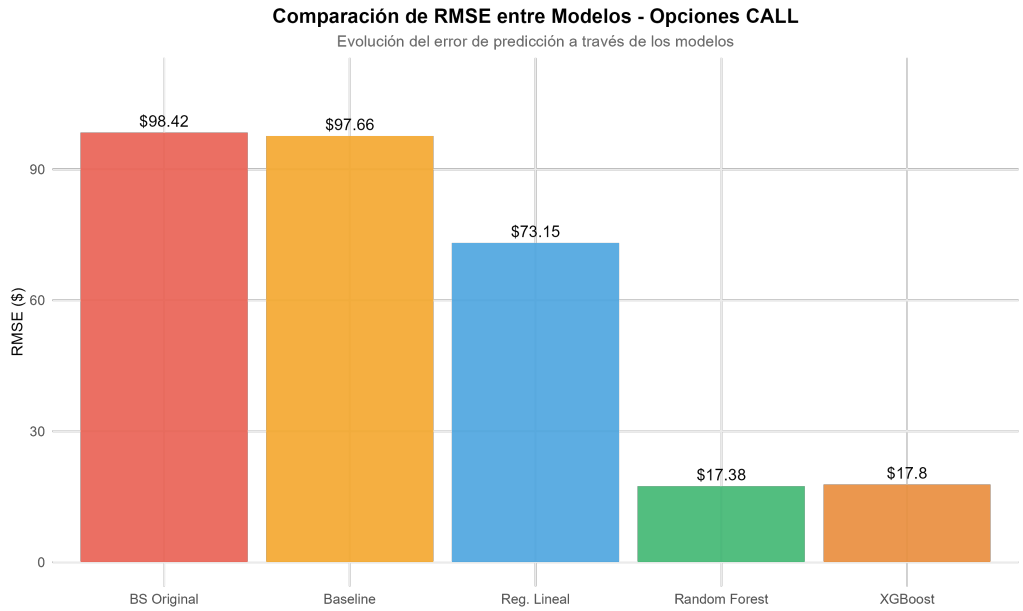


Figura 4.16: Comparación de RMSE entre modelos para opciones CALL.

Para evaluar la relevancia de cada predictor, XGBoost mide cuánto disminuye el error de predicción cuando una variable es utilizada en las divisiones de los árboles. La importancia de variables se calculó mediante la función `xgb.importance()` del paquete `xgboost`, que mide la contribución de cada

variable a la reducción del error durante la construcción de los árboles. La Tabla 4.30 presenta las cinco variables más relevantes.

Tabla 4.30: Las cinco variables más relevantes para el modelo XGBoost de opciones CALL.

Rank	Variable	Importancia Normalizada (%)
1	IV (Volatilidad Implícita)	100.0
2	Moneyness	17.3
3	Strike	13.8
4	Volatilidad_Historica	1.8
5	Precio_Actual	1.8

La volatilidad implícita (IV) emerge como el predictor dominante con el 100 % de importancia normalizada, seguida por Moneyness (17.3 %) y Strike (13.8 %). Este ordenamiento difiere del obtenido con Random Forest, donde `Dias_Vencimiento` ocupaba el primer lugar e IV el segundo. Sin embargo, ambos modelos coinciden en que IV es una de las variables más relevantes para predecir el error de Black-Scholes en opciones CALL. La Figura 4.17 visualiza la importancia relativa de todas las variables.

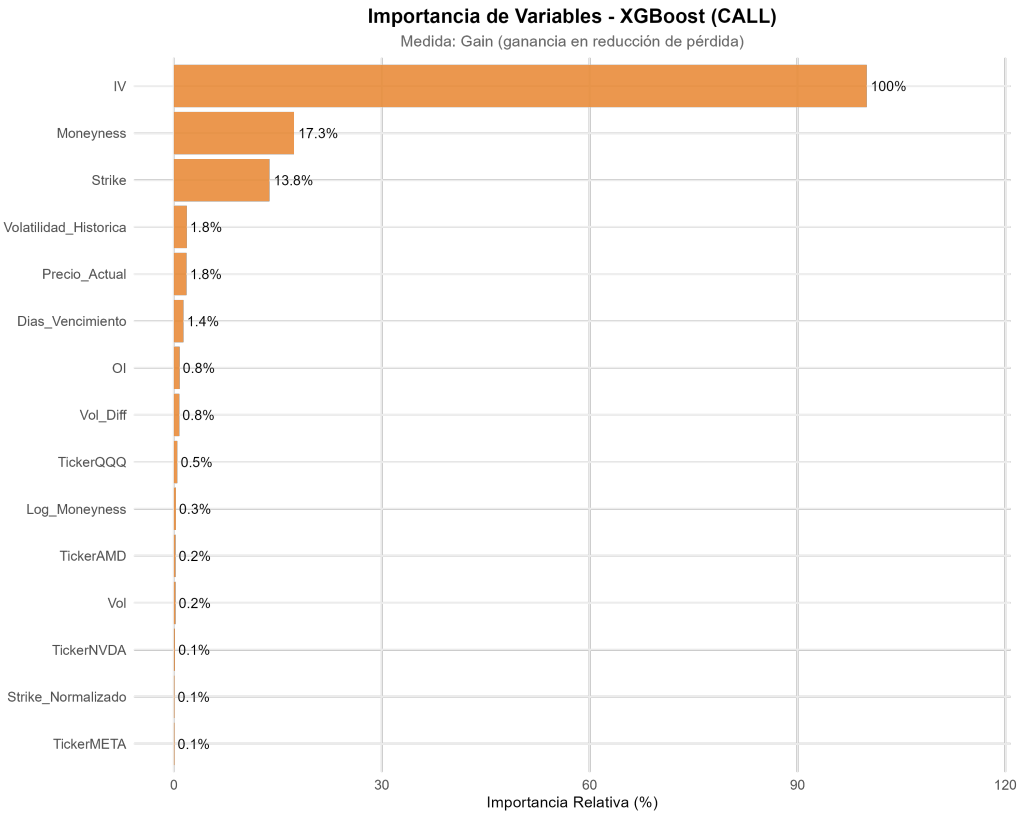


Figura 4.17: Importancia de variables para el modelo XGBoost de opciones CALL.

4.4.4. Resultados para opciones PUT

El modelo XGBoost para opciones PUT se entrenó sobre 14,720 observaciones con la configuración optimizada (`eta = 0.05`, `max_depth = 8`, `nrounds = 161`). La Figura 4.18 presenta las curvas de aprendizaje durante el entrenamiento.

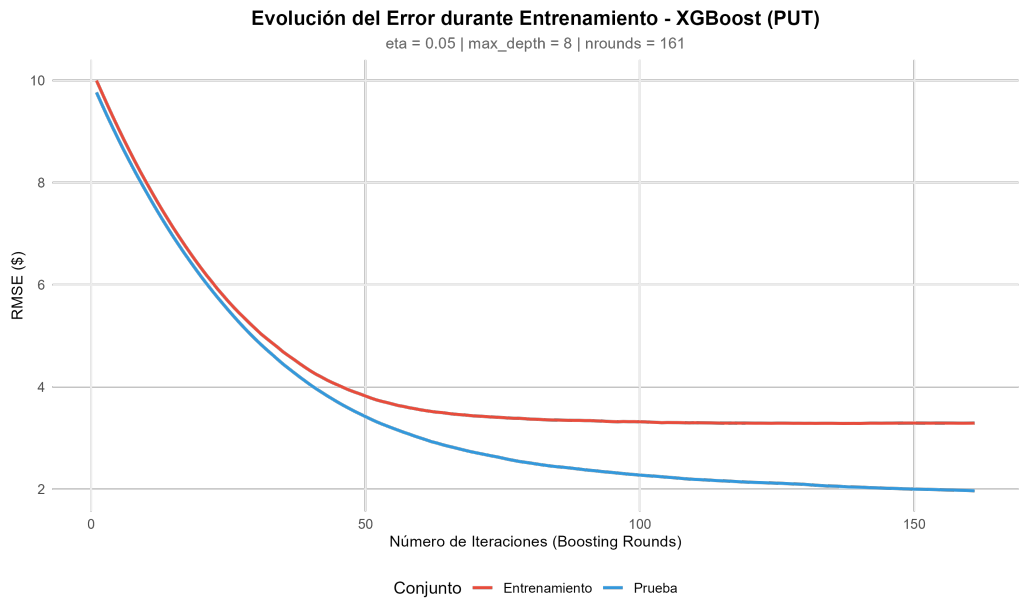


Figura 4.18: Evolución del RMSE durante el entrenamiento de XGBoost para opciones PUT.

Las curvas de aprendizaje para PUT exhiben un patrón similar al de CALL, con convergencia rápida en las primeras 50 iteraciones seguida de estabilización. La Tabla 4.31 presenta las métricas de desempeño.

Tabla 4.31: Métricas de desempeño del modelo XGBoost para opciones PUT.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	3.86	1.71	0.9985
Prueba	10.83	2.95	0.9891

El modelo PUT exhibe un desempeño superior al modelo CALL en todas las métricas. El RMSE de \$10.83 representa una mejora del 89.54 % respecto al baseline (\$103.54), mientras que el $R^2 = 0{,}9891$ indica que el modelo explica el 98.91 % de la variabilidad en los errores de Black-Scholes para opciones PUT. La Figura 4.19 presenta el diagrama de dispersión para opciones PUT.

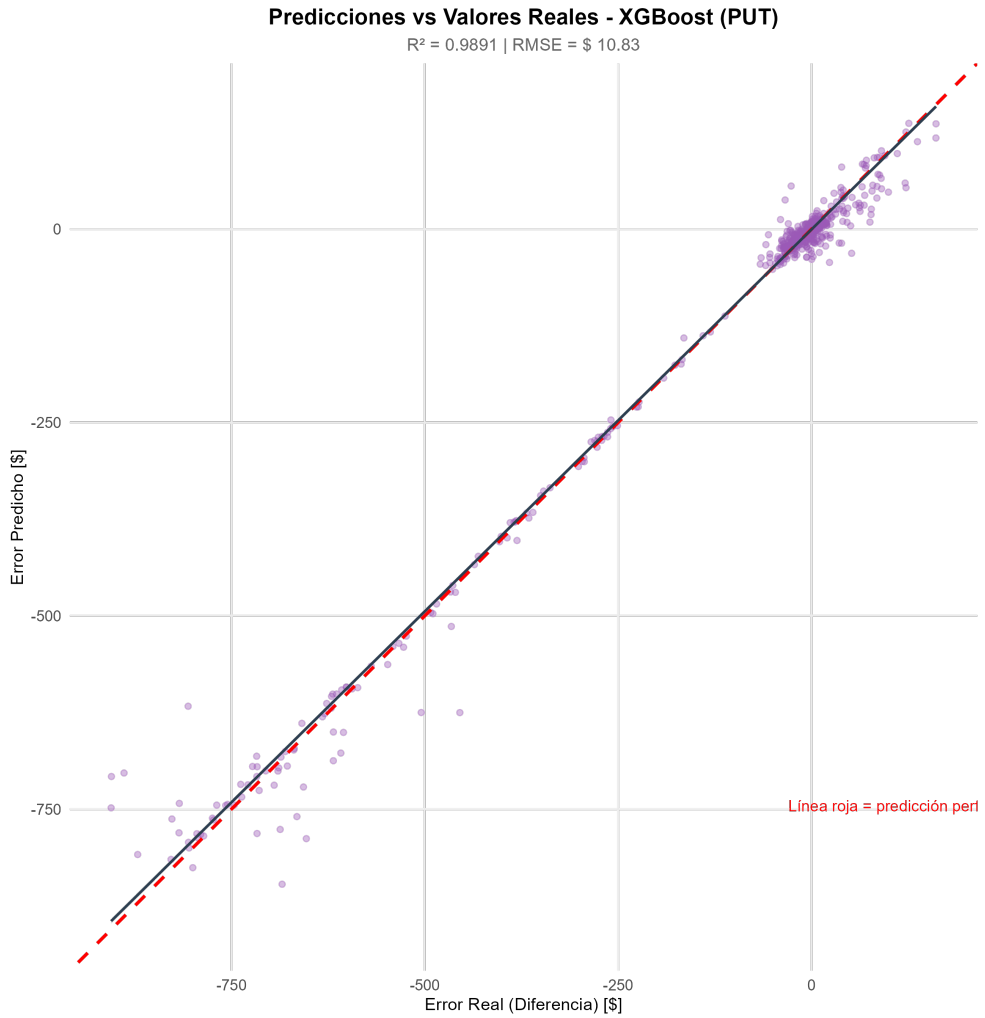


Figura 4.19: Errores predichos vs. errores reales para opciones PUT.

Se observa una concentración más estrecha alrededor de la línea de identidad comparada con opciones CALL, especialmente en errores de gran magnitud. La Tabla 4.32 presenta la comparación con los modelos anteriores.

Tabla 4.32: Comparación de desempeño: XGBoost vs. modelos anteriores para opciones PUT.

Modelo	**RMSE ($)**	**MAE ($)**	**Mejora vs Baseline**
BS Original	105.02	22.78	—
Baseline ($C = -\$16{,}34$)	103.54	34.06	—
Regresión Lineal	31.43	16.02	69.64 %
Random Forest	11.06	2.64	89.32 %
XGBoost	10.83	2.95	89.54 %

A diferencia de las opciones CALL, para opciones PUT XGBoost supera marginalmente a Random

Forest (RMSE de $10.83 vs $11.06, mejora del 2.09 %). Esta diferencia, aunque modesta, sugiere que el enfoque de boosting captura patrones residuales que el bagging de Random Forest no logra modelar completamente en este tipo de opciones. El modelo mejora la precisión de valoración en el 66.6 % de las opciones PUT del conjunto de prueba.

La Figura 4.20 visualiza la evolución del RMSE a través de los modelos.

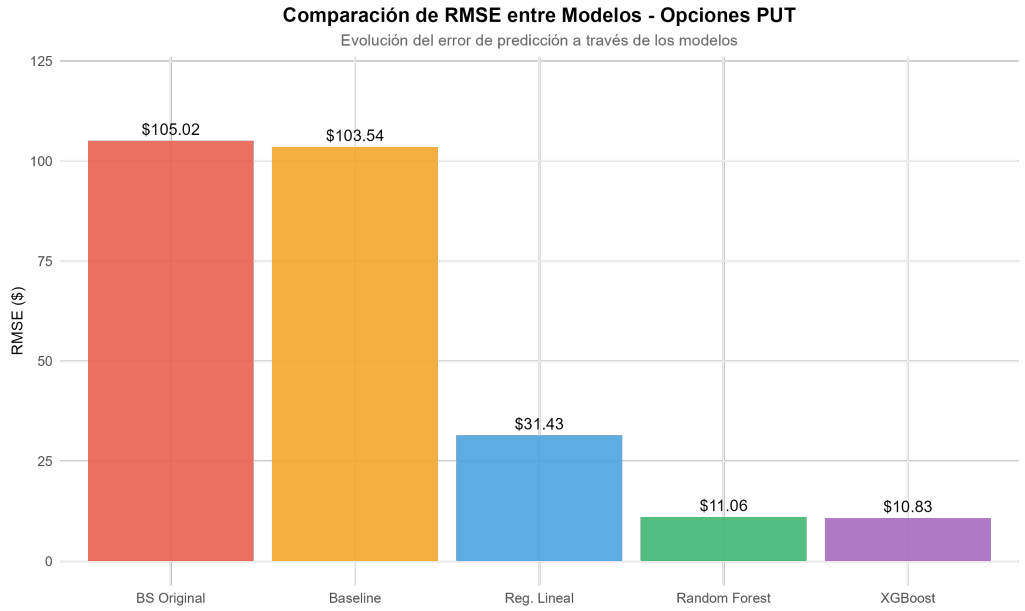


Figura 4.20: Comparación de RMSE entre modelos para opciones PUT.

El análisis de importancia de variables para PUT presenta un perfil marcadamente diferente al de CALL. La Tabla 4.33 muestra las cinco variables más importantes.

Tabla 4.33: Las cinco variables más relevantes para el modelo XGBoost de opciones PUT.

Rank	Variable	Importancia Normalizada (%)
1	Moneyness	100.0
2	Log_Moneyness	14.8
3	Strike_Normalizado	14.6
4	Strike	6.9
5	Vol_Diff	1.9

Para opciones PUT, Moneyness domina con el 100 % de importancia normalizada, seguida por Log_Moneyness y Strike_Normalizado. A diferencia de CALL, la variable IV aparece con apenas 0.4 % de importancia en PUT. Este contraste confirma que los mecanismos que generan errores de Black-Scholes operan de manera diferente según el tipo de opción. La Figura 4.21 visualiza el perfil completo de importancia de variables.

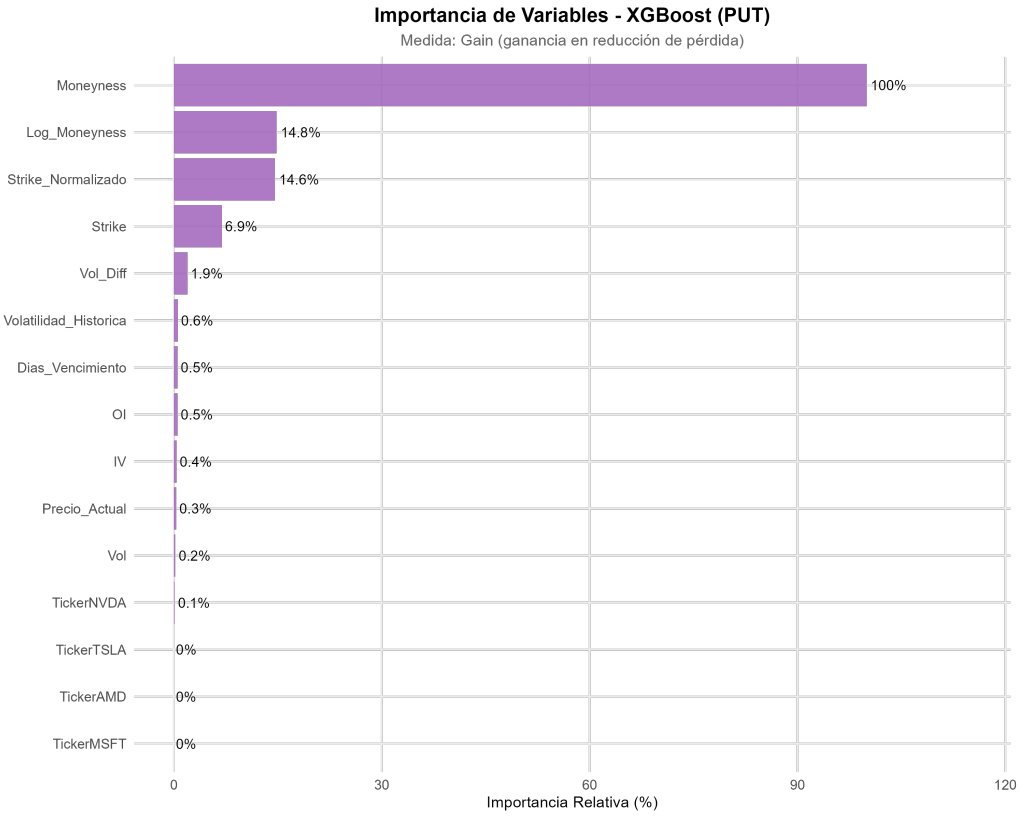


Figura 4.21: Importancia de variables para el modelo XGBoost de opciones PUT.

4.4.5. Rendimiento de los modelos

La Tabla 4.34 consolida los resultados de ambos modelos XGBoost, facilitando la comparación directa entre tipos de opciones y otros modelos.

Tabla 4.34: Resumen comparativo de los modelos XGBoost por tipo de opción.

Métrica	**CALL**	**PUT**
Configuración		
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
eta (learning rate)	0.05	0.05
max_depth	6	8
nrounds (iteraciones)	182	161
Desempeño en prueba		
RMSE ($)	17.80	10.83
MAE ($)	6.34	2.95

Continúa en la siguiente página

Tabla 4.34 – continuación

Métrica	**CALL**	**PUT**
R^2	0.9668	0.9891
Comparación con baseline		
RMSE Baseline ($)	97.66	103.54
Mejora vs Baseline	81.77 %	89.54 %
Comparación con Random Forest		
RMSE Random Forest ($)	17.38	11.06
Diferencia vs RF	+$0.42 (−2.41 %)	−$0.23 (+2.09 %)
Efectividad en ajuste de precios		
Opciones mejoradas	68.8 %	66.6 %
Variable más importante		
Top predictor	IV (100 %)	Moneyness (100 %)

El análisis comparativo confirma que, al igual que en Random Forest, el modelo para opciones PUT predice con mayor precisión que el de CALL, alcanzando un RMSE de $10.83 frente a $17.80. Al comparar ambos algoritmos, ninguno domina de forma clara: Random Forest es ligeramente mejor para CALL ($17.38 vs $17.80) y XGBoost lo es para PUT ($10.83 vs $11.06), con diferencias inferiores a $0.50 en ambos casos. XGBoost logra resultados similares con menos iteraciones (182 vs 200 para CALL; 161 vs 500 para PUT), lo que representa una ventaja en tiempo de cómputo. Respecto a las variables más influyentes, cada tipo de opción depende de factores distintos: en CALL, la volatilidad implícita (IV) es la variable dominante, mientras que en PUT lo es Moneyness, lo que refuerza la necesidad de modelos separados por tipo de contrato.

No obstante, el modelo presenta algunas limitaciones. La brecha entre el RMSE de entrenamiento ($10.57) y el de prueba ($17.80) para CALL sugiere que existe margen de mejora en la capacidad de generalización, aunque el desempeño en prueba sigue siendo significativamente superior al baseline. Además, su desempeño varía según la configuración de hiperparámetros, lo que exige un proceso de ajuste cuidadoso. A nivel individual, el modelo mejora la valoración en el 68.8 % de las opciones CALL y el 66.6 % de las PUT. En conjunto, ambos algoritmos alcanzan un nivel de precisión global similar, y mejoras adicionales requerirían más datos, nuevas variables predictoras o enfoques metodológicos diferentes.

4.5. K-Nearest Neighbors (KNN)

4.5.1. Proceso de modelado

La Figura 4.22 presenta el flujo metodológico del proceso de modelado mediante KNN, desde la preparación de datos hasta la evaluación del desempeño predictivo.

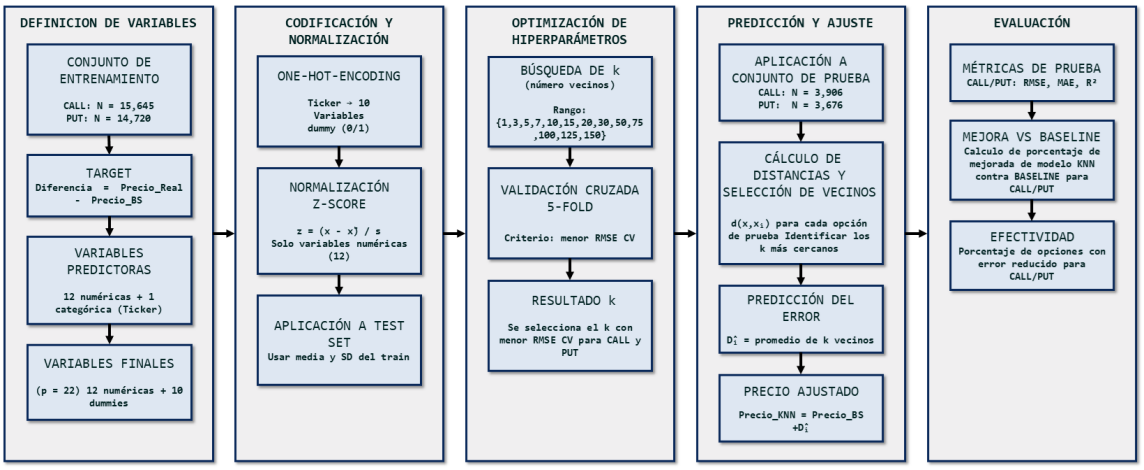


Figura 4.22: Diagrama del proceso de modelado mediante KNN para la predicción del error de Black-Scholes. Fuente: Elaboración propia.

El proceso se estructura en cinco fases principales:

1. **Definición de Variables**: Preparación del conjunto de entrenamiento (30,365 observaciones), definición de la variable objetivo (`Diferencia`) y selección de las 12 variables numéricas predictoras más la variable categórica `Ticker`.

2. **Codificación y Normalización**: Aplicación de one-hot encoding a `Ticker` (generando 10 variables dummy) y transformación Z-score a las variables numéricas utilizando estadísticas calculadas únicamente sobre el conjunto de entrenamiento.

3. **Optimización**: Búsqueda del hiperparámetro óptimo k (número de vecinos) mediante validación cruzada de 5 pliegues, evaluando valores desde $k = 1$ hasta $k = 150$.

4. **Predicción y Ajuste**: Aplicación del modelo entrenado al conjunto de prueba, calculando el precio ajustado como Precio_KNN = Precio_BS + $\hat{D}_i$.

5. **Evaluación**: Cálculo de métricas de desempeño (RMSE, MAE, R^2) y comparación con modelos anteriores.

4.5.2. Configuración e hiperparámetros

La implementación de KNN requiere la especificación de un hiperparámetro principal: el número de vecinos k. Este parámetro controla el balance entre sesgo y varianza del modelo. Valores pequeños de k producen modelos flexibles con bajo sesgo pero alta varianza (sensibles al ruido), mientras que valores grandes producen modelos más suaves con mayor sesgo pero menor varianza.

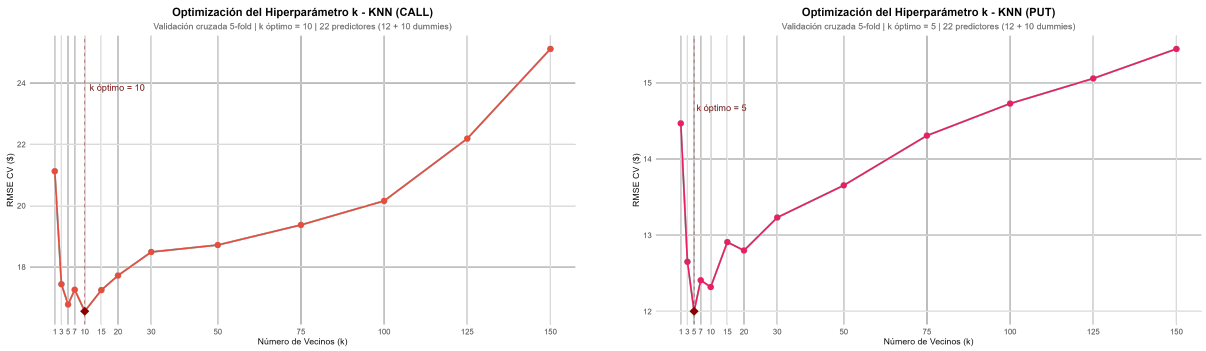
El proceso de optimización evaluó valores de $k \in \{1, 3, 5, 7, 10, 15, 20, 30, 50, 75, 100, 125, 150\}$ mediante validación cruzada de 5 pliegues. La Tabla 4.35 presenta los resultados para ambos tipos de opciones.

Tabla 4.35: Optimización del hiperparámetro k mediante validación cruzada.

k	RMSE CV CALL ($)	RMSE CV PUT ($)
1	21.12	14.47
3	17.44	12.65
5	16.78	**12.00**
7	17.26	12.41
10	**16.55**	12.32
15	17.24	12.91
20	17.72	12.80
30	18.49	13.23
50	18.72	13.65
75	19.37	14.31
100	20.16	14.73
125	22.19	15.06
150	25.12	15.44

Para opciones CALL, el mínimo RMSE de validación cruzada ($16.55) se alcanza con $k = 10$, mientras que para opciones PUT el óptimo ($12.00) se alcanza con $k = 5$. Esta diferencia sugiere que los patrones de error en opciones PUT son más localizados, requiriendo menos vecinos para capturarlos adecuadamente.

La Figura 4.23 visualiza la curva de optimización para ambos tipos de opciones. En ambos casos, el error desciende rápidamente para valores pequeños de K, alcanza un mínimo y posteriormente aumenta de manera gradual conforme K se incrementa.



(a) Opciones CALL **(b)** Opciones PUT

Figura 4.23: Optimización del hiperparámetro k mediante validación cruzada 5-fold. El punto óptimo se indica con un marcador especial.

La Tabla 4.36 resume la configuración final de los modelos KNN.

Tabla 4.36: Configuración final de los modelos KNN por tipo de opción.

Parámetro	**CALL**	**PUT**
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
Número de predictores (total)	22	22
Variables numéricas	12	12
Variables dummy (Ticker)	10	10
k (vecinos)	10	5
Normalización	Z-score	Z-score
Métrica de distancia	Euclidiana	Euclidiana

Las 22 variables predictoras incluyen las 12 variables numéricas definidas en el Capítulo 3 (`Strike`, `Precio_Actual`, `Dias_Vencimiento`, `Volatilidad_Historica`, `IV`, `Vol`, `OI`, `Moneyness`, `Log_Moneyness`, `Vol_Diff`, `Strike_Normalizado` e `In_The_Money`) más las 10 variables dummy generadas por la codificación one-hot del `Ticker`.

4.5.3. Resultados para opciones CALL

El modelo KNN para opciones CALL se configuró con $k = 10$ vecinos utilizando 15,645 observaciones como conjunto de referencia. La Tabla 4.37 presenta las métricas de desempeño.

Tabla 4.37: Métricas de desempeño del modelo KNN para opciones CALL.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	14.86	5.31	0.9719
Prueba	17.42	6.28	0.9682

El modelo alcanza un RMSE de $17.42 en el conjunto de prueba, representando una mejora del 82.16 % respecto al baseline ($97.66). El coeficiente de determinación $R^2 = 0{,}9682$ indica que el modelo explica el 96.82 % de la variabilidad en los errores de Black-Scholes para opciones CALL.

La Figura 4.24 presenta el diagrama de dispersión entre los errores predichos y los errores reales en el conjunto de prueba.

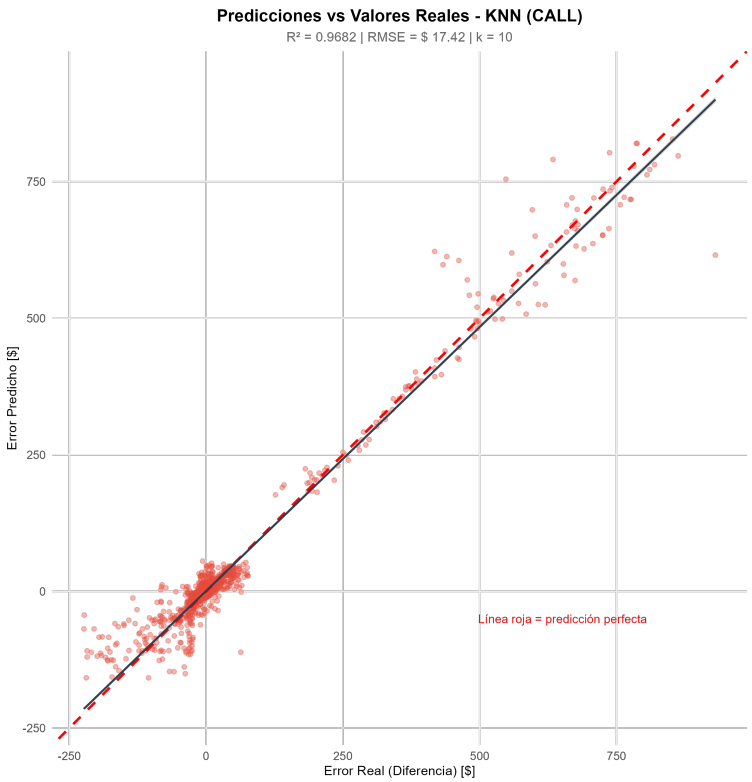


Figura 4.24: Errores predichos vs. errores reales para opciones CALL en el conjunto de prueba. La línea diagonal representa predicción perfecta.

La Tabla 4.38 compara el desempeño de KNN contra los modelos previos.

Tabla 4.38: Comparación de desempeño: KNN vs. modelos anteriores para opciones CALL.

Modelo	RMSE ($)	MAE ($)	Mejora vs Baseline
BS Original	98.42	25.96	—
Baseline (C = +$10.28)	97.66	32.33	—
Regresión Lineal	73.15	30.71	25.10 %
Random Forest	17.38	5.79	82.20 %
KNN	17.42	6.28	82.16 %
XGBoost	17.80	6.34	81.77 %

KNN alcanza un RMSE de $17.42, muy cercano al de Random Forest ($17.38) y ligeramente inferior al de XGBoost ($17.80). Los tres modelos no lineales obtienen resultados comparables para opciones CALL, con diferencias menores a $0.50 entre ellos. El modelo mejora la precisión de valoración en el 73.3 % de las opciones CALL del conjunto de prueba.

La Figura 4.25 visualiza la comparación de RMSE entre modelos.

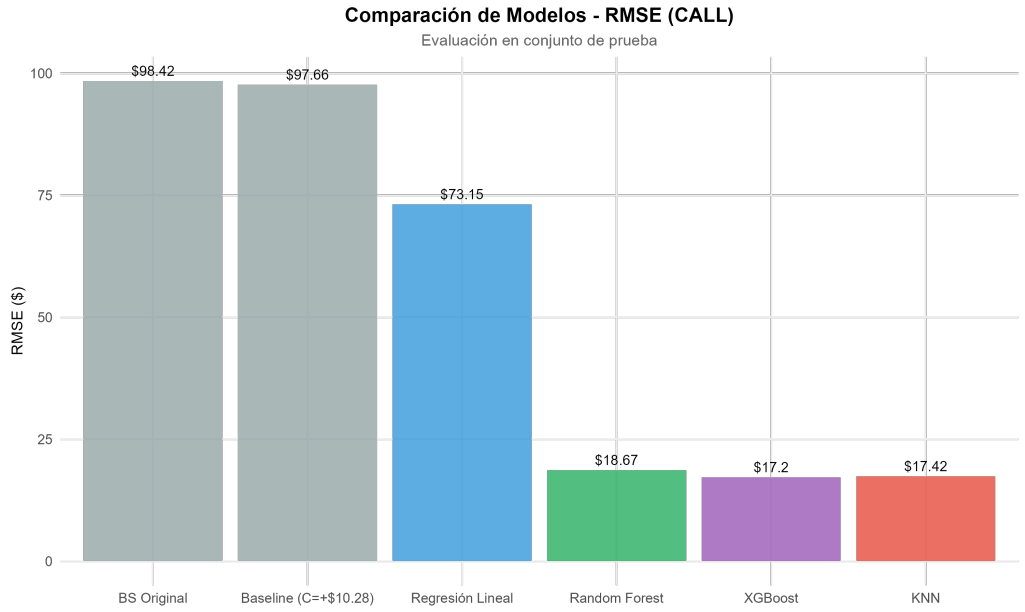


Figura 4.25: Comparación de RMSE entre modelos para opciones CALL.

Para evaluar la relevancia de cada predictor, se utilizó la técnica de importancia por permutación. A diferencia de Random Forest y XGBoost, que proporcionan medidas de importancia incorporadas, KNN no genera esta información de forma nativa, por lo que se requiere una técnica externa. Esta técnica consiste en mezclar aleatoriamente los valores de una variable y medir cuánto aumenta el error de predicción: si el error aumenta considerablemente, la variable es importante; si el error

apenas cambia, la variable aporta poco al modelo. La Tabla 4.39 presenta las cinco variables más relevantes.

Tabla 4.39: Las cinco variables más relevantes para el modelo KNN de opciones CALL.

Rank	Variable	Importancia Normalizada (%)
1	Vol_Diff	100.0
2	Strike_Normalizado	96.0
3	IV	45.3
4	Log_Moneyness	38.3
5	OI	14.6

La diferencia de volatilidades (`Vol_Diff`) y el strike normalizado dominan la predicción, seguidos por la volatilidad implícita (`IV`). Las variables dummy de `Ticker` aparecen al final del ranking con importancias menores al 0.5%, indicando que las características numéricas de las opciones son suficientes para capturar la mayor parte de los patrones de error. La Figura 4.26 visualiza el perfil de importancia de variables.

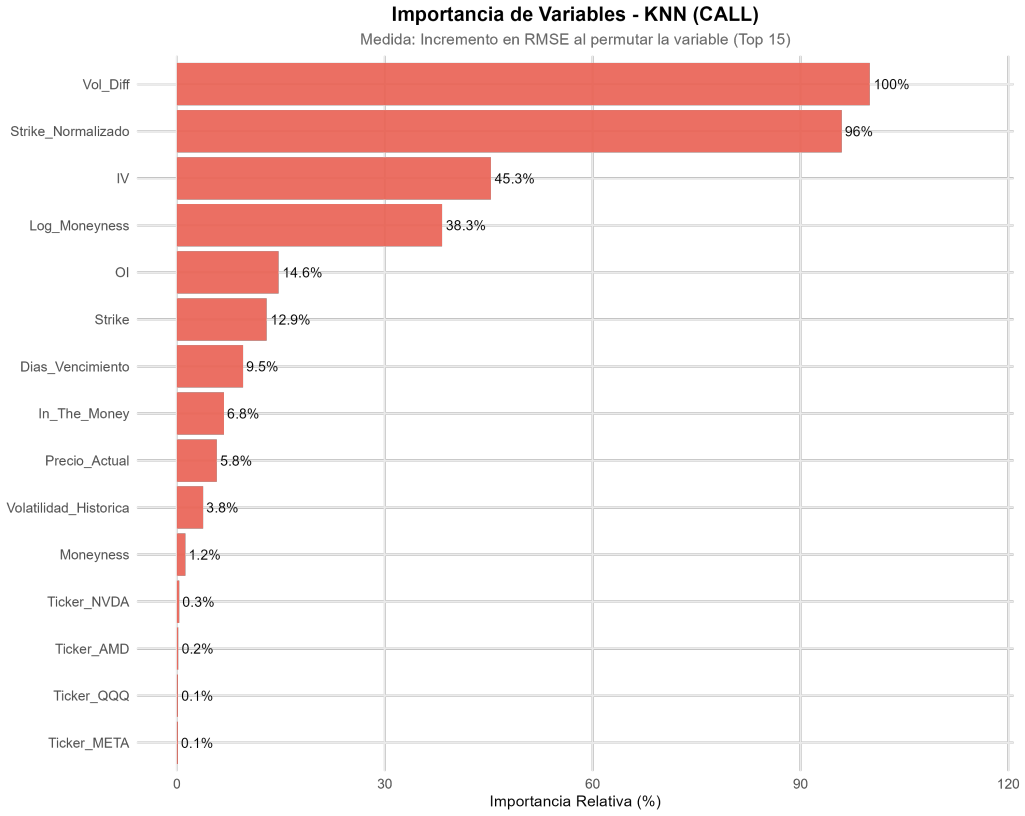


Figura 4.26: Importancia de variables (por permutación) para el modelo KNN de opciones CALL.

4.5.4. Resultados para opciones PUT

El modelo KNN para opciones PUT se construyó con $k = 5$ vecinos sobre 14,720 observaciones. La Tabla 4.40 presenta las métricas de desempeño.

Tabla 4.40: Métricas de desempeño del modelo KNN para opciones PUT.

Conjunto	RMSE ($)	MAE ($)	R^2
Entrenamiento	9.56	2.24	0.9906
Prueba	13.79	3.17	0.9823

Este modelo exhibe un desempeño superior al modelo KNN para opciones CALL en todas las métricas. El RMSE de $13.79 representa una mejora del 86.68 % respecto al baseline ($103.54), mientras que el $R^2 = 0{,}9823$ indica que el modelo explica el 98.23 % de la variabilidad en los errores.

La Figura 4.27 presenta el diagrama de dispersión para opciones PUT.

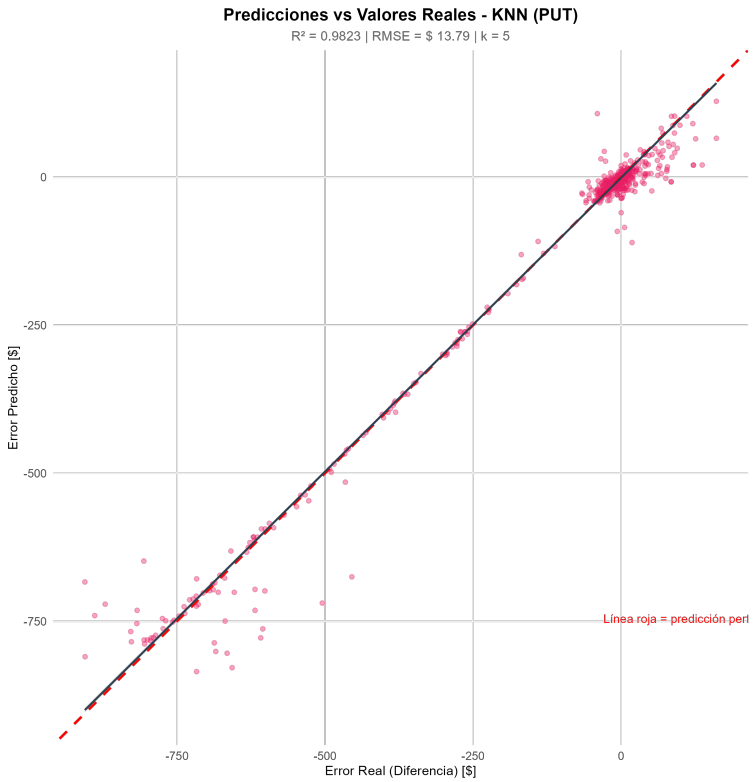


Figura 4.27: Errores predichos vs. errores reales para opciones PUT en el conjunto de prueba.

La Tabla 4.41 presenta la comparación con los modelos anteriores.

Tabla 4.41: Comparación de desempeño: KNN vs. modelos anteriores para opciones PUT.

Modelo	RMSE ($)	MAE ($)	Mejora vs Baseline
BS Original	105.02	22.78	—
Baseline (C = -$16.34)	103.54	34.06	—
Regresión Lineal	31.43	16.02	69.64 %
Random Forest	11.06	2.64	89.32 %
XGBoost	10.83	2.95	89.54 %
KNN	13.79	3.17	86.68 %

Para opciones PUT, KNN presenta un RMSE de $13.79, superior al de Random Forest ($11.06) y XGBoost ($10.83). A diferencia de CALL, donde los tres modelos obtienen resultados similares, en PUT los métodos de ensemble mantienen una ventaja más clara. El modelo mejora la precisión de valoración en el 86.6 % de las opciones PUT del conjunto de prueba.

La Figura 4.28 visualiza la comparación de RMSE entre modelos.

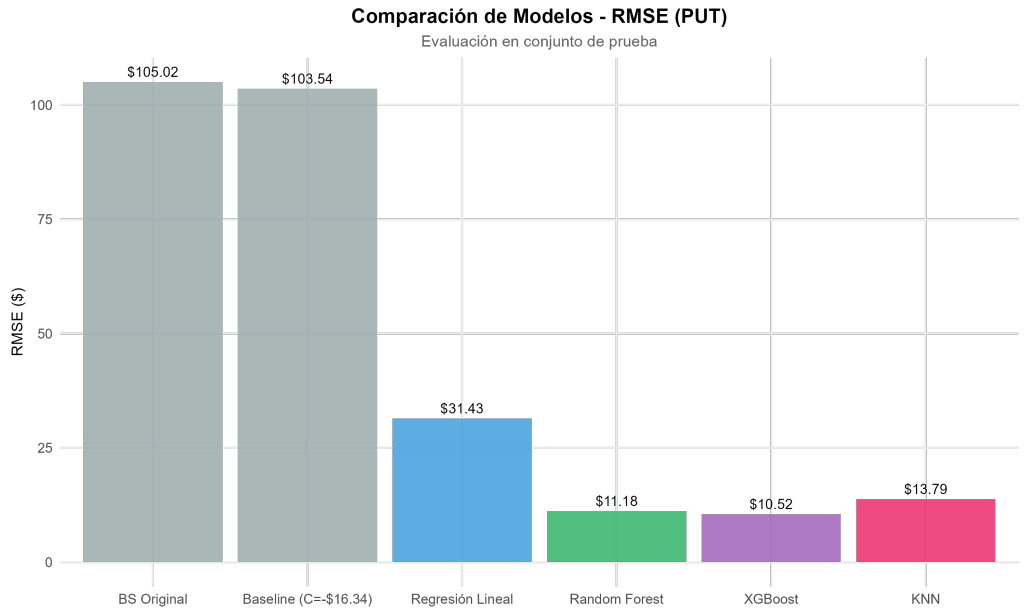


Figura 4.28: Comparación de RMSE entre modelos para opciones PUT.

La Tabla 4.42 presenta las cinco variables más importantes para PUT.

Tabla 4.42: Las cinco variables más relevantes para el modelo KNN de opciones PUT.

Rank	Variable	Importancia Normalizada (%)
1	Strike_Normalizado	100.0
2	Vol_Diff	11.0
3	Strike	7.1
4	IV	6.1
5	Log_Moneyness	6.1

Para opciones PUT, `Strike_Normalizado` domina con el 100 % de importancia normalizada, seguido a distancia por `Vol_Diff` (11.0 %). Este perfil es más concentrado que el de CALL, donde las cuatro primeras variables superaban el 38 % de importancia. Al igual que en CALL, las variables dummy de Ticker muestran importancia cercana a cero.

La Figura 4.29 visualiza el perfil de importancia.

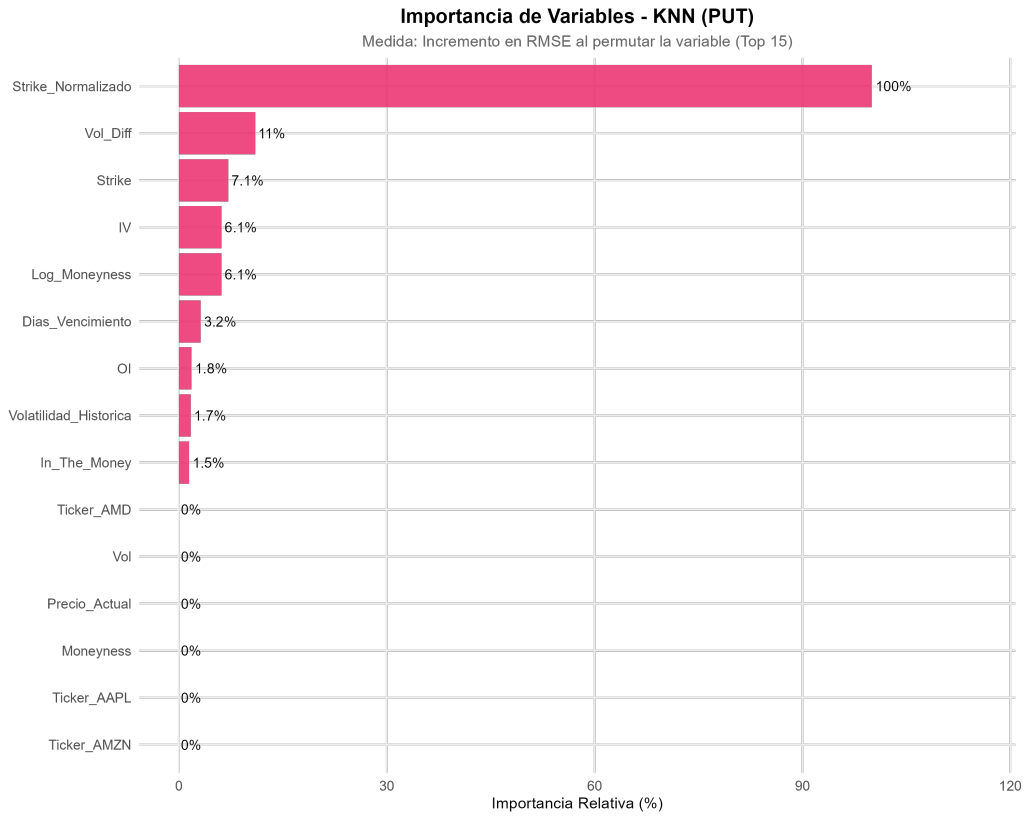


Figura 4.29: Importancia de variables (por permutación) para el modelo KNN de opciones PUT.

4.5.5. Rendimiento de los modelos

La Tabla 4.43 consolida los resultados de ambos modelos KNN.

Tabla 4.43: Resumen comparativo de los modelos KNN por tipo de opción.

Métrica	**CALL**	**PUT**
Configuración		
Observaciones entrenamiento	15,645	14,720
Observaciones prueba	3,906	3,676
k (vecinos)	10	5
Número de predictores	22	22
Desempeño en prueba		
RMSE ($)	17.42	13.79
MAE ($)	6.28	3.17
R^2	0.9682	0.9823
Comparación con baseline		
RMSE Baseline ($)	97.66	103.54
Mejora vs Baseline	82.16 %	86.68 %
Comparación con modelos de ensemble		
RMSE Random Forest ($)	17.38	11.06
RMSE XGBoost ($)	17.80	10.83
Efectividad en ajuste de precios		
Opciones mejoradas	73.3 %	86.6 %
Variable más importante		
Top predictor	Vol_Diff (100 %)	Strike_Normalizado (100 %)

El análisis comparativo muestra que, al igual que en los modelos anteriores, KNN predice con mayor precisión las opciones PUT (RMSE de $13.79) que las CALL ($17.42). Para opciones CALL, los tres modelos no lineales alcanzan un desempeño muy similar: Random Forest ($17.38), KNN ($17.42) y XGBoost ($17.80), con diferencias menores a $0.50. Para opciones PUT, en cambio, los modelos de ensemble mantienen una ventaja más clara, con RMSE de $11.06 (Random Forest) y $10.83 (XGBoost) frente a $13.79 de KNN.

Respecto a las variables más influyentes, los perfiles difieren según el tipo de opción: para CALL, la diferencia entre volatilidad implícita e histórica (`Vol_Diff`) es el factor dominante, mientras que para PUT lo es la posición relativa del strike (`Strike_Normalizado`). Las variables dummy de Ticker

muestran importancia inferior al 0.5 % en ambos modelos, lo que sugiere que los patrones de error de Black-Scholes dependen principalmente de las características de la opción y no del activo subyacente específico.

4.6. Redes Neuronales MLP

4.6.1. Proceso de modelado

La Figura 4.30 presenta el flujo metodológico del proceso de modelado mediante redes neuronales MLP, desde la preparación de datos hasta la evaluación del desempeño predictivo.

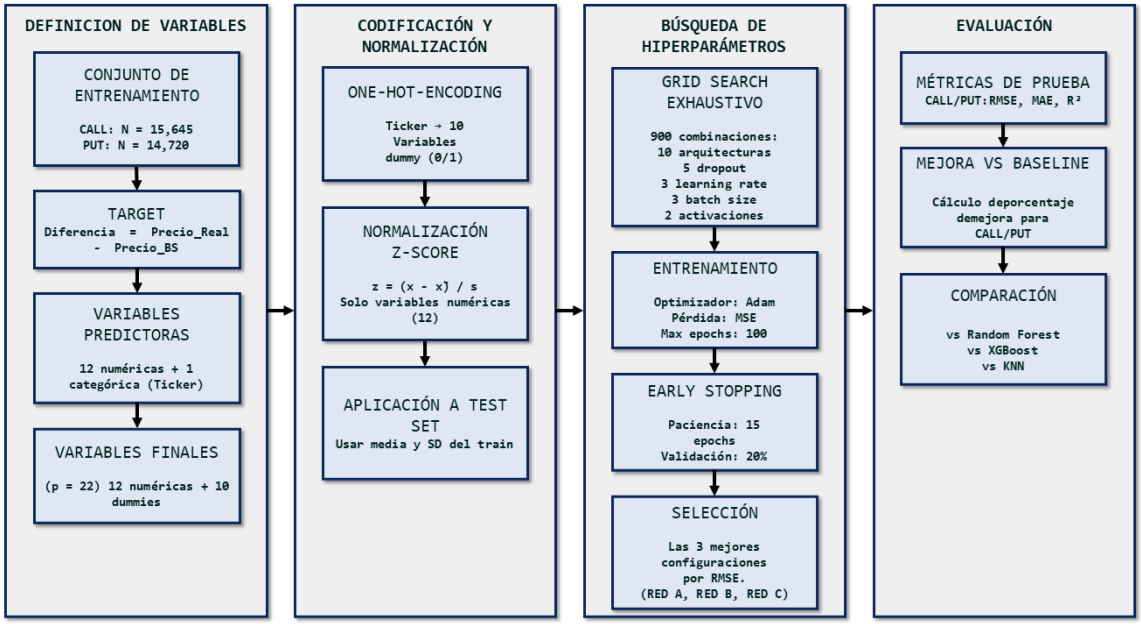


Figura 4.30: Diagrama del proceso de modelado mediante redes neuronales MLP para la predicción del error de Black-Scholes. Fuente: Elaboración propia.

El proceso se estructura en cuatro fases principales:

1. **Definición de Variables**: Preparación del conjunto de entrenamiento (15,645 observaciones CALL y 14,720 PUT), definición de la variable objetivo (`Diferencia`) y selección de las 12 variables numéricas predictoras más la variable categórica `Ticker`.

2. **Codificación y Normalización**: Aplicación de one-hot encoding a `Ticker` (generando 10 variables dummy) y transformación Z-score a las variables numéricas utilizando estadísticas calculadas únicamente sobre el conjunto de entrenamiento.

3. **Búsqueda de Hiperparámetros**: Grid Search exhaustivo evaluando 900 configuraciones por tipo de opción, combinando 10 arquitecturas, 5 valores de dropout, 3 learning rates, 3 batch sizes y 2 funciones de activación. Cada modelo se entrena con optimizador Adam, función de pérdida MSE, máximo 100 epochs y early stopping con paciencia de 15 epochs.

4. **Evaluación**: Selección de las tres mejores redes por RMSE, cálculo de métricas de desempeño (RMSE, MAE, R^2) y comparación con modelos anteriores (Random Forest, XGBoost, KNN).

4.6.2. Configuración y búsqueda de hiperparámetros

Dado que no existe una forma analítica de determinar los hiperparámetros óptimos de una red neuronal, se implementó una búsqueda exhaustiva (Grid Search) evaluando todas las combinaciones posibles dentro de un espacio de búsqueda predefinido. La Tabla 4.44 presenta los valores considerados para cada hiperparámetro.

Tabla 4.44: Espacio de búsqueda de hiperparámetros para redes neuronales MLP.

Hiperparámetro	**Valores**	**Cantidad**
Arquitectura	32, 64, 128, 32-16, 64-32, 128-64, 64-32-16, 128-64-32, 256-128-64, 128-64-32-16	10
Dropout	0.0, 0.1, 0.2, 0.3, 0.4	5
Learning Rate	0.001, 0.005, 0.01	3
Batch Size	32, 64, 128	3
Activación	ReLU, tanh	2
Total combinaciones	$10 \times 5 \times 3 \times 3 \times 2$	**900**

El espacio de búsqueda contempla arquitecturas desde redes simples de una sola capa con 32 neuronas hasta redes profundas de 4 capas con 128-64-32-16 neuronas. Se evaluaron 900 configuraciones diferentes para cada tipo de opción (CALL y PUT), entrenando un total de 1,800 modelos.

Cada modelo se entrenó con un máximo de 100 epochs, implementando *early stopping* con paciencia de 15 epochs para detener el entrenamiento si el error de validación dejaba de mejorar. Se utilizó una partición de validación del 20 % del conjunto de entrenamiento para monitorear el desempeño durante el entrenamiento. La Tabla 4.45 resume la configuración del proceso de entrenamiento.

Tabla 4.45: Configuración del entrenamiento de redes neuronales MLP.

Parámetro	Valor
Variables predictoras	22 (12 numéricas + 10 dummies)
Normalización	Z-score (solo variables numéricas)
Optimizador	Adam
Función de pérdida	MSE (Error Cuadrático Medio)
Epochs máximos	100
Early stopping	Paciencia de 15 epochs
Validación	20 % del conjunto de entrenamiento
Configuraciones evaluadas	900 por tipo de opción

4.6.3. Resultados para opciones CALL y PUT

La distribución de RMSE obtenida en las 900 configuraciones evaluadas para cada tipo de opción (Figura 6.4, Anexo 6.2) evidencia una amplia variabilidad según la configuración seleccionada. Para opciones CALL, el RMSE varía entre $22.49 y $94.52, mientras que para PUT varía entre $13.70 y $104.14. Esta variabilidad confirma la importancia de realizar una búsqueda exhaustiva de hiperparámetros.

El desempeño por tipo de arquitectura (Figura 6.5, Anexo 6.2) muestra que las redes con 2-3 capas ocultas tienden a obtener mejores resultados que las arquitecturas más simples o más profundas.

A partir de las 900 configuraciones evaluadas, se seleccionaron las tres mejores redes neuronales para cada tipo de opción. La Tabla 4.46 presenta las mejores configuraciones para opciones CALL.

Tabla 4.46: Las tres mejores redes neuronales para opciones CALL.

Red	Arquitectura	Dropout	LR	Batch	Activación	RMSE ($)
A	128-64	0.1	0.005	64	ReLU	22.49
B	256-128-64	0.4	0.001	128	ReLU	22.51
C	64-32-16	0.0	0.005	128	ReLU	22.60

La Tabla 4.47 presenta las mejores configuraciones para opciones PUT.

Tabla 4.47: Las tres mejores redes neuronales para opciones PUT.

Red	Arquitectura	Dropout	LR	Batch	Activación	RMSE ($)
A	128-64	0.0	0.005	64	ReLU	13.70
B	256-128-64	0.0	0.005	32	ReLU	14.05
C	128-64	0.0	0.005	32	ReLU	14.09

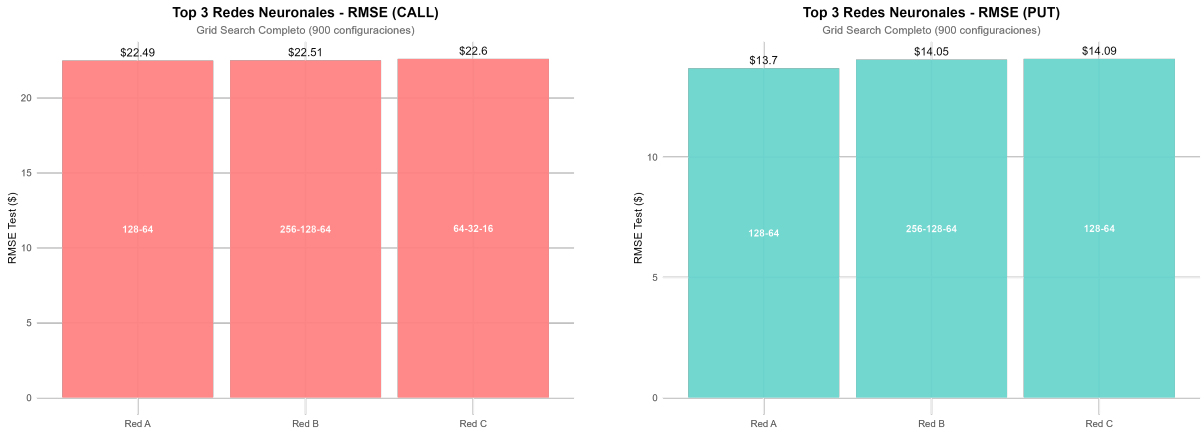
Se observan patrones consistentes en las mejores configuraciones: todas utilizan la función de activación ReLU, learning rates moderados (0.001-0.005), y arquitecturas de 2-3 capas ocultas. Para opciones PUT, las tres mejores redes no utilizan dropout, mientras que para CALL hay mayor variabilidad en este parámetro.

La Tabla 4.48 presenta las métricas completas de las mejores redes seleccionadas.

Tabla 4.48: Métricas de desempeño de las tres mejores redes neuronales.

Tipo	Red	RMSE ($)	MAE ($)	R^2
CALL	Red A (128-64)	22.49	9.19	0.9469
	Red B (256-128-64)	22.51	9.09	0.9469
	Red C (64-32-16)	22.60	9.57	0.9464
PUT	Red A (128-64)	13.70	4.42	0.9825
	Red B (256-128-64)	14.05	4.38	0.9816
	Red C (128-64)	14.09	4.48	0.9815

La Figura 4.31 visualiza el desempeño de las tres mejores redes para cada tipo de opción.



(a) Opciones CALL **(b)** Opciones PUT

Figura 4.31: Top 3 redes neuronales MLP por tipo de opción.

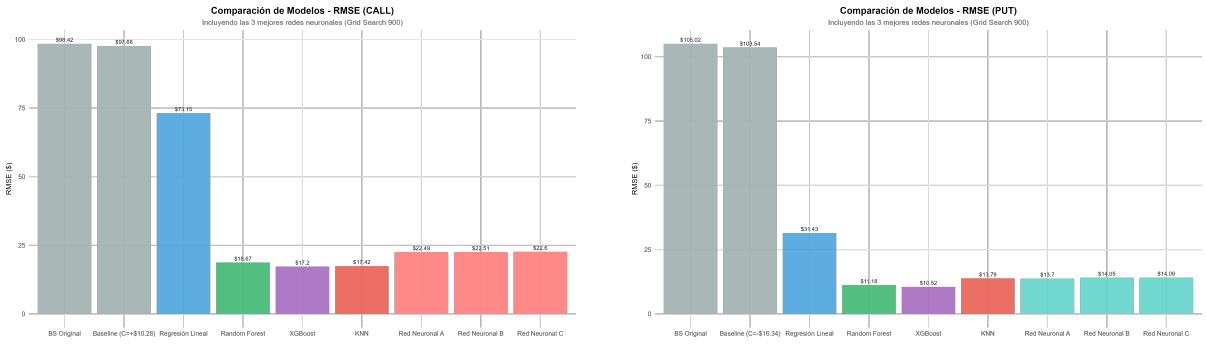
4.6.4. Rendimiento de los modelos

La Tabla 4.49 presenta la comparación de la mejor red neuronal con los modelos previamente desarrollados.

Tabla 4.49: Comparación de la mejor red neuronal MLP con otros modelos.

Modelo	RMSE CALL ($)	RMSE PUT ($)	Mejora CALL (%)	Mejora PUT (%)
BS Original	98.42	105.02	–	–
Baseline	97.66	103.54	0.00	0.00
Regresión Lineal	73.15	31.43	25.10	69.64
Random Forest	17.38	11.06	82.20	89.32
XGBoost	17.80	10.83	81.77	89.54
KNN	17.42	13.79	82.16	86.68
MLP (mejor)	**22.49**	**13.70**	**76.97**	**86.77**

La Figura 4.32 visualiza la comparación de RMSE entre todos los modelos, incluyendo las tres mejores redes neuronales.



(a) Opciones CALL **(b)** Opciones PUT

Figura 4.32: Comparación de RMSE entre todos los modelos incluyendo redes neuronales MLP.

A pesar de la búsqueda exhaustiva de 900 configuraciones, las redes neuronales MLP no lograron superar el desempeño de los modelos de ensemble (Random Forest y XGBoost) ni de KNN. Para opciones CALL, la mejor red neuronal obtiene un RMSE de $22.49, comparado con $17.80 de XGBoost. Para opciones PUT, la diferencia es menor: $13.70 de MLP versus $10.83 de XGBoost.

Este resultado puede atribuirse a varios factores. Primero, el tamaño del conjunto de datos (aproximadamente 15,000-19,000 observaciones de entrenamiento) puede ser insuficiente para que las redes neuronales capturen patrones complejos de manera efectiva, ya que estos modelos típicamente requieren grandes volúmenes de datos para superar a algoritmos más tradicionales. Segundo, los modelos basados en árboles (Random Forest, XGBoost) son particularmente efectivos para datos tabulares estructurados como los utilizados en este estudio. Tercero, la naturaleza del problema de predicción de errores de Black-Scholes puede no beneficiarse de las capacidades de aprendizaje de representaciones profundas que caracterizan a las redes neuronales.

No obstante, las redes neuronales MLP logran una mejora sustancial respecto al modelo Baseline, con reducciones del RMSE del 76.97 % para CALL y 86.77 % para PUT. Además, para opciones PUT, la mejor red neuronal ($13.70) obtiene un desempeño comparable a KNN ($13.79), lo que sugiere que en ciertos contextos las redes neuronales pueden ser competitivas.

El detalle completo de las 900 configuraciones evaluadas para cada tipo de opción se encuentra disponible en el Anexo 6.2.

4.7. Prototipo Evaluador de Corrección de Precios

Como cierre de la fase de modelación y atendiendo a la fase de despliegue (*Deployment*) propuesta por la metodología CRISP-DM [1], se desarrolló un prototipo web interactivo que materializa los

resultados obtenidos en una herramienta de uso práctico. Este prototipo permite visualizar, contrato por contrato, el desempeño del enfoque híbrido Black-Scholes + Machine Learning sobre el conjunto de prueba (7,582 observaciones), facilitando la evaluación cualitativa de la corrección aplicada por los modelos seleccionados.

El prototipo está disponible de forma pública en la siguiente dirección:

https://julian936.github.io/Tesis---Prediccion-BS/

4.7.1. Propósito y alcance

El **Evaluador de Corrección de Precios de Opciones Europeas** fue concebido como una herramienta complementaria al documento de tesis, con tres objetivos principales:

- **Validación visual del modelo:** permitir al usuario seleccionar cualquier contrato del conjunto de prueba y observar de manera inmediata la comparación entre el precio teórico de Black-Scholes, el precio ajustado mediante el modelo de aprendizaje automático correspondiente y el precio real de mercado.
- **Trazabilidad del proceso analítico:** exponer de forma transparente las variables financieras de cada contrato (Strike, Precio Spot, Días al Vencimiento, Volatilidad Implícita, *Moneyness*, entre otras), así como la corrección específica aplicada por el modelo.
- **Comunicación de resultados:** traducir los hallazgos cuantitativos de la tesis (RMSE, MAE, R^2) en un formato accesible para usuarios no técnicos, demostrando empíricamente cómo el enfoque híbrido reduce el error de valoración en cada contrato individual.

4.7.2. Arquitectura y modelos integrados

El prototipo integra los dos modelos ganadores identificados en las secciones anteriores, aplicando el modelo correspondiente según el tipo de opción del contrato seleccionado:

- Para opciones **CALL**: el modelo **Random Forest** con configuración `mtry=8` y `ntree=200`, que alcanzó un RMSE de $17,38 en el conjunto de prueba (sección 4.3.3).
- Para opciones **PUT**: el modelo **XGBoost** con configuración `eta=0,05`, `max_depth=8` y `nrounds=161`, que alcanzó un RMSE de $10,83 en el conjunto de prueba (sección 4.4.4).

Los modelos fueron entrenados sobre los 30,365 contratos del conjunto de entrenamiento y aplicados sobre los 7,582 contratos del conjunto de prueba descargados de Yahoo Finance el 6 de octubre de 2025, cubriendo diez activos representativos del mercado estadounidense: AAPL, AMD, AMZN, GOOGL, META, MSFT, NVDA, QQQ, SPY y TSLA.

4.7.3. Funcionalidades principales

La interfaz del prototipo se estructura en cuatro componentes funcionales, descritos a continuación:

4.7.3.0.1. 1. Selector de contratos. El usuario puede filtrar los 7,582 contratos del conjunto de prueba mediante tres criterios: tipo de opción (CALL, PUT o todos), *ticker* (uno de los diez activos disponibles) y ordenamiento por cualquiera de las variables financieras (Strike, Precio Spot, Días al Vencimiento, Volatilidad Implícita, *Moneyness*, Precio Real, Precio Black-Scholes). Esto facilita la exploración dirigida de casos específicos.

4.7.3.0.2. 2. Comparación de precios. Para cada contrato seleccionado, el panel principal muestra simultáneamente:

- El *Precio de mercado* ($Last$) observado, considerado como valor de referencia.
- El *Precio Black-Scholes* (P_{BS}) calculado mediante las ecuaciones (2.1) y (2.2), junto con su error absoluto respecto al precio real.
- El *Precio ajustado* ($P_{BS+ML} = P_{BS} + \hat{f}_{ML}(\mathbf{x})$), generado al sumar la corrección estimada por el modelo de *Machine Learning*, junto con su error residual.

4.7.3.0.3. 3. Visualización del impacto de la corrección. Se presenta una visualización gráfica comparativa de los errores absolutos antes y después de la corrección, junto con la magnitud de la corrección aplicada por el modelo ($\hat{f}_{ML}(\mathbf{x})$). Esto permite evaluar de forma inmediata si el modelo redujo el error de valoración para el contrato consultado.

4.7.3.0.4. 4. Diagnóstico del modelo por contrato. El prototipo emite un *veredicto* cualitativo sobre el desempeño del modelo en cada caso particular, identificando si la corrección fue efectiva o no, y exponiendo las variables financieras del contrato que pueden explicar dicho comportamiento. Este diagnóstico contribuye a la interpretabilidad del modelo y permite detectar regímenes donde el enfoque híbrido presenta mayores o menores limitaciones.

4.7.4. Stack tecnológico

El prototipo fue desarrollado con tecnologías web estándar (HTML, CSS y JavaScript) y desplegado mediante GitHub Pages, garantizando acceso público y gratuito sin necesidad de infraestructura adicional. Los datos del conjunto de prueba y las predicciones de los modelos se encuentran serializados de forma estática, lo que permite una ejecución *client-side* sin requerimientos de servidor.

4.7.5. Limitaciones del prototipo

Es importante señalar que el prototipo opera exclusivamente sobre los datos descargados el 6 de octubre de 2025 y no implementa la descarga en tiempo real desde Yahoo Finance ni el reentrenamiento dinámico de los modelos. Su propósito es ilustrar y validar el enfoque metodológico propuesto en este trabajo, no servir como herramienta operativa de valoración en producción. Una versión productiva requeriría los componentes adicionales descritos en la sección de Trabajos Futuros (sección 5.2): conexión a APIs de datos en tiempo real, mecanismos de recalibración periódica del modelo y monitoreo continuo del desempeño predictivo.

4.7.6. Aporte al cumplimiento de los objetivos

El desarrollo del prototipo cierra el ciclo CRISP-DM al trasladar los modelos entrenados desde el entorno de investigación hacia un *entregable* consultable, evidenciando empíricamente la aplicabilidad práctica del enfoque híbrido propuesto. Constituye, por tanto, un complemento al cumplimiento del Objetivo Específico 4, demostrando que la corrección del precio teórico mediante aprendizaje automático no solo es estadísticamente significativa, sino también operativamente viable.

Capítulo 5

CONCLUSIONES Y TRABAJOS FUTUROS

Este capítulo presenta las conclusiones derivadas del cumplimiento de los cuatro objetivos específicos del proyecto. A continuación se sintetizan las respuestas a las cinco preguntas de sistematización planteadas en el Capítulo 1: la pregunta 1 (variables explicativas) se aborda en el Objetivo 1; las preguntas 2 y 5 (técnicas de ML y comparación con BS) se abordan en el Objetivo 2; las preguntas 3 y 4 (mejora del ajuste y nivel de error) se abordan en los Objetivos 3 y 4.

5.1. CONCLUSIONES

El modelo de Black-Scholes, ampliamente utilizado en la valoración de opciones europeas, genera desviaciones sistemáticas respecto al precio real de mercado debido a sus supuestos de volatilidad constante y eficiencia perfecta del mercado. Este trabajo demostró que es posible predecir esas desviaciones mediante técnicas de aprendizaje automático y utilizarlas para corregir el precio teórico, reduciendo significativamente el error de valoración en opciones *call* y *put* sobre diez activos del mercado estadounidense.

Sobre el diseño y estructuración de la base de datos (Objetivo 1). Se construyó un dataset de 37,947 contratos de opciones europeas descargados de Yahoo Finance el 6 de octubre de 2025, particionados estratificadamente en 80 % entrenamiento y 20 % prueba. Las pruebas de Kolmogorov-Smirnov confirmaron que ambos conjuntos son estadísticamente comparables en las distribuciones de las variables clave ($p > 0{,}05$), garantizando que las métricas de evaluación reflejan el desempeño real del modelo ante datos no vistos. La ingeniería de variables identificó cinco predictores derivados de alta relevancia: `Moneyness`, `Log_Moneyness`, `Vol_Diff`, `Strike_Normalizado` e `In_The_Money`. El análisis de importancia en los modelos ganadores reveló perfiles diferenciados según el tipo de contrato: para opciones CALL, `Dias_Vencimiento` resultó el predictor dominante en Random Forest, evidenciando que el supuesto de volatilidad constante de Black-Scholes se vuelve más problemático conforme aumenta el horizonte temporal; para opciones PUT, `Moneyness` dominó en XGBoost, indicando que la posición relativa del strike respecto al precio spot es el factor determinante del

error de valoración. Esta divergencia confirma que los mecanismos generadores de error operan de manera fundamentalmente distinta según el tipo de contrato, justificando el entrenamiento de modelos separados.

Sobre la construcción e implementación de los modelos (Objetivo 2). Se implementaron seis modelos de complejidad creciente. La regresión lineal cumplió un rol metodológico relevante más allá de su desempeño predictivo: identificó que `Tasa_Libre_Riesgo` debía excluirse por ser constante en todas las observaciones, y que `Vol_Diff` presentaba dependencia lineal perfecta con `IV` y `Volatilidad_Historica`, orientando así la selección de predictores para los modelos posteriores. Los modelos de ensemble — Random Forest y XGBoost — superaron ampliamente a la regresión lineal y a las redes neuronales MLP, confirmando que los datos tabulares estructurados favorecen métodos basados en árboles. La decisión de entrenar modelos separados por tipo de opción quedó justificada por los perfiles divergentes de importancia de variables: `Dias_Vencimiento` dominó en CALL bajo Random Forest, mientras que `Moneyness` dominó en PUT bajo XGBoost, evidenciando que los mecanismos que generan errores de Black-Scholes operan de forma fundamentalmente distinta según el tipo de contrato.

Sobre la evaluación del rendimiento (Objetivo 3). El RMSE fue la métrica principal de evaluación por penalizar cuadráticamente los errores grandes, más relevantes en el contexto financiero que los errores típicos capturados por el MAE. El R^2 complementó la comparación al cuantificar la proporción de variabilidad explicada. Todos los modelos de aprendizaje automático superaron al baseline en ambos tipos de opciones. Las redes neuronales MLP, a pesar de evaluar 900 configuraciones mediante Grid Search, no lograron superar a los modelos de ensemble, resultado atribuible al volumen de datos disponible — aproximadamente 15,000 observaciones de entrenamiento por tipo de opción — insuficiente para que arquitecturas profundas superen a métodos más tradicionales en datos tabulares de esta escala.

Sobre la mejora en la estimación de precios (Objetivo 4). El enfoque híbrido Black-Scholes + aprendizaje automático demostró reducir sustancialmente el error de valoración. Random Forest obtuvo el mejor desempeño para opciones CALL con un RMSE de $17,38 en el conjunto de prueba, representando una mejora del 82,20 % frente al baseline ($97,66). XGBoost fue el mejor modelo para opciones PUT con un RMSE de $10,83, equivalente a una mejora del 89,54 % frente al baseline ($103,54). El modelo mejora la valoración en el 76,9 % de las opciones CALL y en el 66,6 % de las PUT de forma individual. Las limitaciones observadas se concentran en opciones con volatilidad implícita nula, precios muy cercanos a cero y casos atípicos fuera del rango del conjunto de entrenamiento.

Análisis del comportamiento asimétrico de los modelos ganadores. El hecho de que Random Forest maximice el rendimiento en opciones CALL y XGBoost lo haga en opciones PUT responde directamen-

te a las diferencias en la distribución y naturaleza matemática de los residuos en cada subconjunto de datos.

1. **Robustez de Random Forest (*Bagging*) en opciones CALL.** El error del modelo tradicional de Black-Scholes está dominado por la variable de horizonte temporal (`Dias_Vencimiento`). En activos altamente volátiles, el comportamiento de la variable objetivo se vuelve inestable a mayor plazo, induciendo una distribución de errores con una fuerte asimetría a la derecha y una alta concentración de valores atípicos (*outliers*) extremos positivos. En este entorno caracterizado por una alta varianza y ruido asimétrico en los datos, la arquitectura de *bagging* de Random Forest resulta óptima: al construir múltiples árboles de decisión independientes mediante muestras *bootstrap* y promediar sus salidas, el algoritmo estabiliza la varianza global del sistema y neutraliza de manera efectiva el impacto distorsionador de estas anomalías extremas en la predicción, previniendo el sobreajuste.

2. **Precisión de XGBoost (*Boosting*) en opciones PUT.** Para las opciones PUT, el factor clave que explica el error del modelo clásico es el `Moneyness` (la relación entre el precio actual del activo y el precio de ejercicio). En el mercado real, las fallas de Black-Scholes no ocurren al azar, sino que siguen una tendencia en forma de curva muy marcada y predecible. Esto significa que el error cambia de manera constante y predecible a medida que la opción se aleja o se acerca a su punto de equilibrio. La arquitectura de *boosting* secuencial de XGBoost es ideal para este tipo de datos: en lugar de promediar árboles independientes, el algoritmo construye árboles uno detrás de otro, donde cada nuevo árbol se entrena específicamente para corregir los pequeños fallos que cometió el anterior. Este aprendizaje iterativo paso a paso permite modelar la curvatura del error con una precisión matemática muy alta, superando la promediación simple de Random Forest.

Respuesta a la pregunta de investigación. El trabajo confirma que es posible predecir la diferencia entre el valor teórico de Black-Scholes y el valor real de mercado mediante técnicas de aprendizaje automático, y que dicha predicción permite ajustar el precio teórico reduciendo el error de valoración en más del 82 % para opciones CALL y el 89 % para opciones PUT. Las variables que más influyen en esas diferencias son `Dias_Vencimiento` para contratos CALL y `Moneyness` para contratos PUT. Entre los modelos evaluados, Random Forest y XGBoost ofrecieron el mejor balance entre precisión y generalización. En cuanto a su aplicabilidad, el enfoque es operacionalmente viable: conserva la estructura interpretable del modelo Black-Scholes y solo añade un paso de corrección computacionalmente ligero, lo que facilita su integración en flujos existentes de valoración financiera.
Formalmente, el modelo híbrido propuesto se expresa como:

$$\hat{P}_{\text{BS+ML}} = P_{BS}(S_0, K, T, r, \sigma) + \hat{f}_{ML}(\mathbf{x}) \tag{5.1}$$

donde P_{BS} es el precio teórico de Black-Scholes y $\hat{f}_{ML}(\mathbf{x})$ es la predicción del error de valoración generada por el modelo de aprendizaje automático sobre el vector de características $\mathbf{x}$. La especificación final, diferenciada por tipo de contrato, queda:

$$\hat{P}_{\text{BS+ML}} = \begin{cases} P_{BS} + \hat{f}_{RF}^{CALL}(\mathbf{x}), & \text{RMSE} = \$17{,}38 \quad (R^2 = 0{,}9683) \\ P_{BS} + \hat{f}_{XGB}^{PUT}(\mathbf{x}), & \text{RMSE} = \$10{,}83 \quad (R^2 = 0{,}9891) \end{cases} \tag{5.2}$$

A modo de referencia, el modelo baseline con corrección constante quedó definido por $C_{CALL} = +\$10{,}28$ y $C_{PUT} = -\$16{,}34$, valores que cuantifican el sesgo sistemático del modelo Black-Scholes —subvaloración en CALL y sobrevaloración en PUT— y que constituyen el umbral mínimo superado por todos los modelos de aprendizaje automático implementados.

Es importante precisar que el modelo BS+ML estima el valor justo de la prima — cuánto debería costar el contrato hoy según sus características financieras — pero no predice la dirección futura del activo subyacente. Para una decisión de inversión completa se requiere complementar esta señal de valoración con un modelo de predicción del subyacente, aspecto que constituye una extensión natural de este trabajo.

Limitaciones. Los resultados están condicionados a los datos de una única fecha de mercado (6 de octubre de 2025), lo que limita la generalización a otros regímenes de volatilidad o condiciones macroeconómicas distintas. El modelo puede degradarse ante cambios estructurales del mercado sin recalibración periódica. Adicionalmente, el uso del precio *Last* como referencia del valor real introduce ruido en contratos de bajo volumen donde ese precio puede no reflejar el precio de equilibrio del mercado.

5.2. TRABAJOS FUTUROS

Recalibración periódica del modelo. Al estar entrenado sobre datos de una única fecha, el modelo captura los patrones de error propios de ese momento del mercado. Se recomienda implementar un esquema de reentrenamiento mensual con datos frescos y monitoreo semanal del RMSE en producción para detectar degradación anticipada. El modelo debe recalibrarse ante cambios estructurales sostenidos: variaciones significativas del VIX, ajustes de tasas por parte de la Reserva Federal, o eventos corporativos mayores en los activos subyacentes.

Ampliación del dataset. Incorporar datos de múltiples fechas y condiciones de mercado mejoraría la capacidad de generalización, especialmente para las redes neuronales MLP que requieren mayor volumen de observaciones. La inclusión de períodos de alta volatilidad enriquecería el espacio de entrenamiento y reduciría las limitaciones observadas en opciones atípicas.

Integración con modelos de predicción del subyacente. El enfoque BS+ML resuelve la valoración de la prima, pero no la predicción de la dirección del activo. Un trabajo futuro podría integrar este modelo con modelos de series de tiempo (ARIMA, LSTM) o de predicción de volatilidad (GARCH) para construir un sistema de apoyo a decisiones de inversión que combine la señal de precio justo con la señal de dirección del subyacente.

Extensión a otros mercados e instrumentos. La metodología es transferible a otros mercados financieros y a otros instrumentos derivados — opciones americanas, opciones sobre índices, opciones sobre divisas — donde las desviaciones de Black-Scholes presentan características distintas que podrían beneficiarse del mismo enfoque híbrido.

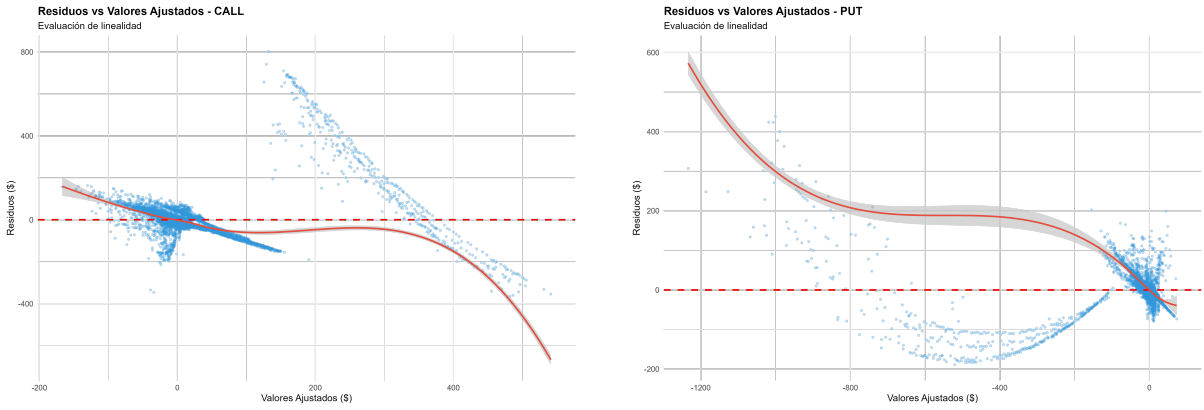
Implementación en tiempo real. Desplegar el modelo con datos en tiempo real mediante APIs de mercado permitiría evaluar su desempeño con precios *bid/ask* en lugar de precios históricos *Last*, acercando el sistema al contexto operativo real de toma de decisiones que motivó este trabajo.

Capítulo 6

Anexos

6.1. Diagnósticos de la Regresión Lineal

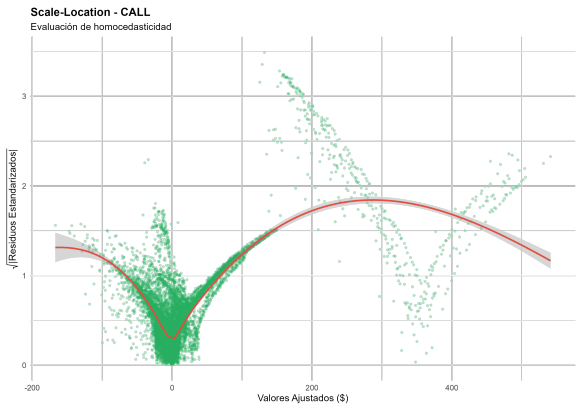
Este anexo presenta los gráficos de diagnóstico utilizados para la validación de los supuestos del modelo de regresión lineal múltiple, tanto para opciones CALL como PUT. La interpretación de estos gráficos se desarrolla en el cuerpo del documento (Sección de Validación de supuestos).



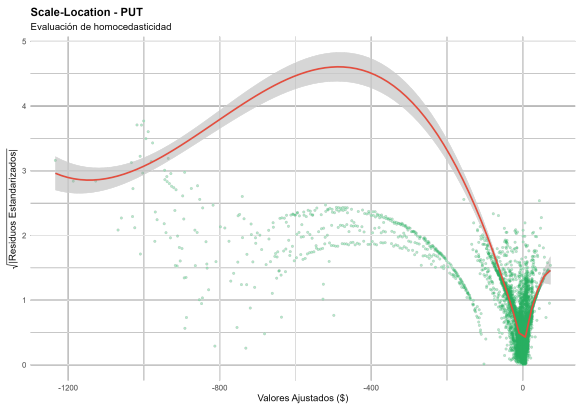
(a) Opciones CALL

(b) Opciones PUT

Figura 6.1: Residuos vs valores ajustados para evaluación de linealidad. La línea roja representa el ajuste LOESS y la línea punteada horizontal indica residuo cero.

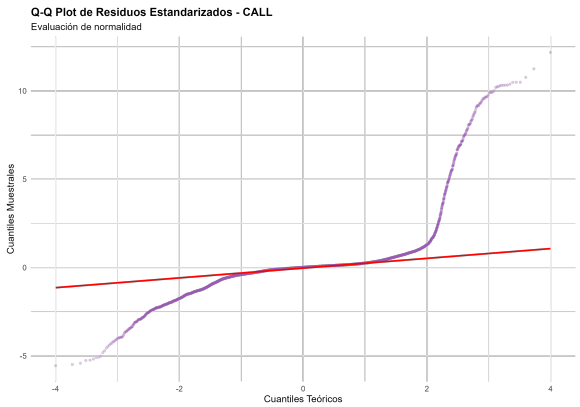


(a) Opciones CALL

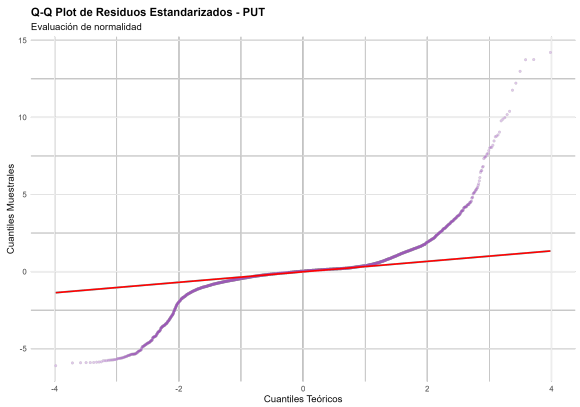


(b) Opciones PUT

Figura 6.2: Gráficos Scale-Location para evaluación de homocedasticidad. Si la varianza fuera constante, la curva LOESS sería aproximadamente horizontal.



(a) Opciones CALL

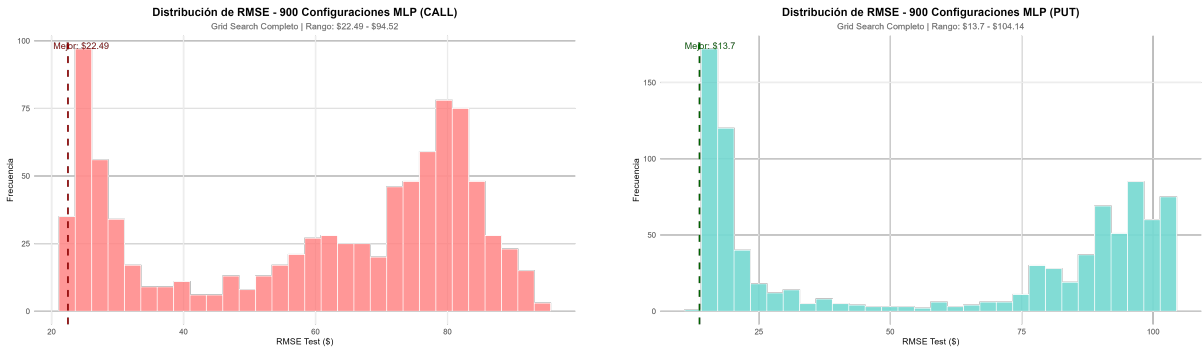


(b) Opciones PUT

Figura 6.3: Q-Q plots de residuos estandarizados. Si los residuos siguieran una distribución normal, los puntos se alinearían sobre la línea roja de referencia.

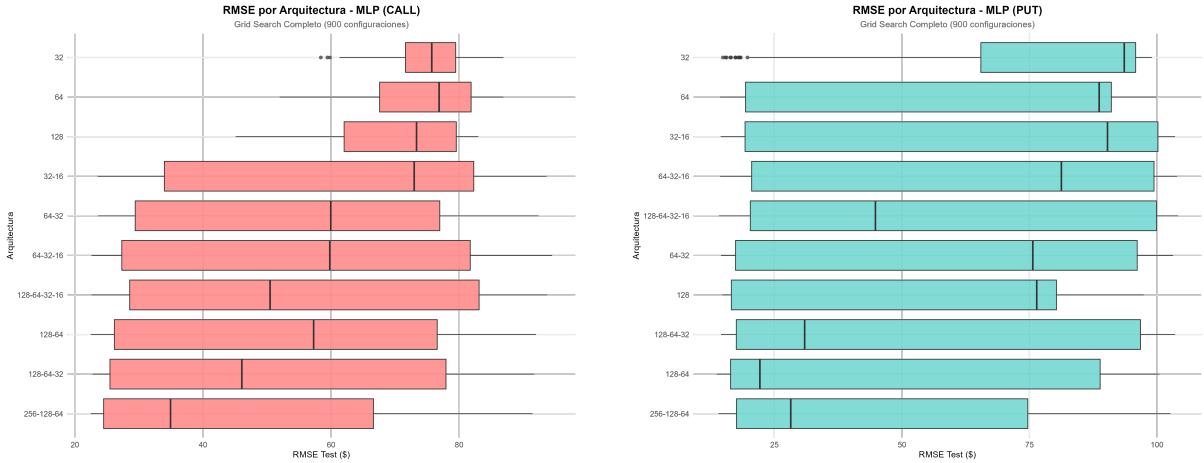
6.2. Configuraciones de Redes Neuronales MLP

Este anexo presenta el análisis detallado de las 900 configuraciones de redes neuronales evaluadas mediante Grid Search para cada tipo de opción: la distribución de RMSE, el desempeño por arquitectura, y las mejores configuraciones ordenadas por RMSE. El archivo completo con las 900 configuraciones está disponible en formato digital: `900_configuraciones_MLP_CALL.xlsx` y `900_configuraciones_MLP_PUT.xlsx`.



(a) Opciones CALL **(b)** Opciones PUT

Figura 6.4: Distribución de RMSE en las 900 configuraciones de redes neuronales MLP.



(a) Opciones CALL **(b)** Opciones PUT

Figura 6.5: RMSE por arquitectura de red neuronal MLP.

6.2.1. Configuraciones para opciones CALL

La Tabla 6.1 presenta las primeras 60 configuraciones ordenadas por RMSE para opciones CALL.

Tabla 6.1: Primeras 60 configuraciones de redes neuronales MLP para opciones CALL (ordenadas por RMSE).

Rank	Arquitectura	Capas	Dropout	LR	Batch	Activ.	RMSE ($)	MAE ($)
1	128-64	2	0.1	0.005	64	relu	22.49	9.19
2	256-128-64	3	0.4	0.001	128	relu	22.51	9.09
3	64-32-16	3	0.0	0.005	128	relu	22.60	9.57
4	128-64-32-16	4	0.0	0.005	32	relu	22.61	9.46
5	64-32-16	3	0.2	0.005	128	relu	22.68	9.13
6	128-64-32	3	0.0	0.001	64	relu	22.77	9.33

Continúa en la siguiente página...

Tabla 6.1 – continuación

Rank	Arquitectura	Capas	Dropout	LR	Batch	Activ.	RMSE ($)	MAE ($)
7	256-128-64	3	0.1	0.001	32	relu	22.78	9.09
8	128-64-32	3	0.3	0.01	32	relu	22.86	9.84
9	128-64	2	0.1	0.005	128	relu	23.05	9.29
10	128-64	2	0.0	0.01	32	relu	23.08	10.48
11	128-64-32	3	0.0	0.001	32	relu	23.10	9.84
12	128-64-32-16	4	0.0	0.01	64	relu	23.10	9.60
13	256-128-64	3	0.1	0.001	128	relu	23.24	9.50
14	256-128-64	3	0.1	0.005	128	relu	23.25	9.52
15	256-128-64	3	0.3	0.001	64	relu	23.26	9.57
16	128-64-32	3	0.1	0.001	64	relu	23.28	9.37
17	128-64-32	3	0.0	0.005	32	relu	23.29	10.91
18	128-64-32-16	4	0.0	0.005	64	relu	23.29	10.12
19	256-128-64	3	0.2	0.01	128	relu	23.30	9.69
20	256-128-64	3	0.1	0.001	64	relu	23.31	9.11
21	256-128-64	3	0.0	0.01	128	relu	23.32	9.88
22	128-64-32	3	0.1	0.01	128	relu	23.37	9.76
23	128-64	2	0.1	0.01	32	relu	23.39	9.18
24	128-64	2	0.2	0.01	128	relu	23.39	9.72
25	64-32-16	3	0.0	0.01	32	relu	23.42	10.51
26	64-32-16	3	0.1	0.01	32	relu	23.46	9.52
27	64-32-16	3	0.2	0.01	128	relu	23.47	9.69
28	128-64-32	3	0.0	0.005	64	relu	23.47	9.99
29	128-64-32-16	4	0.0	0.001	128	relu	23.48	10.59
30	256-128-64	3	0.3	0.005	32	relu	23.49	9.62
31	128-64-32-16	4	0.1	0.001	32	relu	23.49	9.63
32	128-64	2	0.2	0.005	64	relu	23.51	9.40
33	128-64-32	3	0.0	0.01	64	relu	23.54	9.63
34	256-128-64	3	0.0	0.01	32	relu	23.54	9.13
35	32-16	2	0.2	0.001	32	relu	23.57	9.47
36	128-64	2	0.2	0.005	128	relu	23.62	9.87
37	64-32	2	0.0	0.01	32	relu	23.63	10.37
38	256-128-64	3	0.3	0.001	32	relu	23.64	9.64
39	128-64	2	0.2	0.001	64	relu	23.65	9.62
40	64-32-16	3	0.0	0.01	64	relu	23.66	9.84
41	256-128-64	3	0.0	0.001	64	relu	23.67	9.97
42	128-64	2	0.0	0.01	64	relu	23.75	10.36
43	64-32-16	3	0.0	0.005	32	relu	23.75	10.10
44	64-32	2	0.0	0.01	128	relu	23.76	10.44
45	256-128-64	3	0.0	0.005	64	relu	23.86	9.97
46	256-128-64	3	0.2	0.001	64	relu	23.87	9.82
47	128-64-32-16	4	0.0	0.001	32	relu	23.88	9.95
48	128-64-32-16	4	0.0	0.01	128	relu	23.88	10.36
49	128-64-32-16	4	0.2	0.01	64	relu	23.89	9.72

Continúa en la siguiente página...

Tabla 6.1 – continuación

Rank	Arquitectura	Capas	Dropout	LR	Batch	Activ.	RMSE ($)	MAE ($)
50	256-128-64	3	0.4	0.001	64	relu	23.89	9.79
51	256-128-64	3	0.2	0.005	64	relu	23.90	9.75
52	256-128-64	3	0.1	0.01	128	relu	23.91	9.31
53	32-16	2	0.0	0.01	64	relu	23.95	11.61
54	64-32	2	0.2	0.005	64	relu	23.96	9.85
55	128-64-32	3	0.1	0.005	128	relu	24.06	10.06
56	64-32	2	0.2	0.01	128	relu	24.07	10.02
57	128-64-32-16	4	0.2	0.005	64	relu	24.10	9.60
58	128-64-32-16	4	0.0	0.005	128	relu	24.13	10.72
59	256-128-64	3	0.0	0.005	32	relu	24.13	11.76
60	128-64-32-16	4	0.1	0.005	32	relu	24.14	9.72

6.2.2. Configuraciones para opciones PUT

La Tabla 6.2 presenta las primeras 59 configuraciones ordenadas por RMSE para opciones PUT.

Tabla 6.2: Primeras 60 configuraciones de redes neuronales MLP para opciones PUT (ordenadas por RMSE).

Rank	Arquitectura	Capas	Dropout	LR	Batch	Activ.	RMSE ($)	MAE ($)
1	128-64	2	0.0	0.005	64	relu	13.70	4.42
2	256-128-64	3	0.0	0.005	32	relu	14.05	4.38
3	128-64	2	0.0	0.005	32	relu	14.09	4.48
4	128-64-32-16	4	0.0	0.001	64	relu	14.13	5.17
5	128-64	2	0.0	0.001	128	relu	14.21	4.84
6	128-64	2	0.0	0.001	32	relu	14.22	4.77
7	128-64	2	0.0	0.01	32	relu	14.26	4.46
8	256-128-64	3	0.0	0.001	64	relu	14.33	4.67
9	64	1	0.0	0.01	32	relu	14.36	5.33
10	64-32-16	3	0.0	0.005	64	relu	14.37	4.36
11	128-64-32-16	4	0.0	0.005	32	relu	14.41	4.39
12	256-128-64	3	0.0	0.01	128	relu	14.44	4.63
13	64-32-16	3	0.0	0.001	32	relu	14.46	5.01
14	32-16	2	0.0	0.01	32	relu	14.47	4.81
15	64-32	2	0.0	0.01	64	relu	14.51	4.66
16	128-64-32-16	4	0.0	0.01	64	relu	14.55	4.87
17	64-32-16	3	0.0	0.01	64	relu	14.55	4.89
18	128-64-32	3	0.0	0.01	32	relu	14.55	4.40
19	256-128-64	3	0.0	0.001	128	relu	14.56	4.92
20	64-32	2	0.0	0.005	128	relu	14.59	4.90
21	128-64	2	0.0	0.01	64	relu	14.60	5.01
22	128-64-32	3	0.0	0.005	64	relu	14.60	4.72
23	128-64-32	3	0.0	0.001	128	relu	14.61	4.85

Continúa en la siguiente página...

Tabla 6.2 – continuación

Rank	Arquitectura	Capas	Dropout	LR	Batch	Activ.	RMSE ($)	MAE ($)
24	128-64-32-16	4	0.0	0.005	64	relu	14.62	4.71
25	128-64-32-16	4	0.0	0.001	128	relu	14.66	5.19
26	64-32	2	0.0	0.001	64	relu	14.67	5.16
27	128-64	2	0.0	0.001	64	relu	14.67	4.97
28	256-128-64	3	0.0	0.01	64	relu	14.68	4.75
29	32-16	2	0.0	0.01	128	relu	14.69	5.30
30	128-64-32-16	4	0.0	0.001	32	relu	14.74	5.28
31	128-64-32	3	0.0	0.001	32	relu	14.77	5.17
32	128	1	0.1	0.01	64	relu	14.78	5.18
33	256-128-64	3	0.0	0.001	32	relu	14.78	4.89
34	32-16	2	0.0	0.005	128	relu	14.81	5.41
35	128-64-32-16	4	0.1	0.001	32	relu	14.83	4.71
36	128-64-32	3	0.0	0.01	64	relu	14.88	5.00
37	256-128-64	3	0.2	0.001	32	relu	14.90	4.90
38	32	1	0.0	0.01	32	relu	14.91	5.84
39	32-16	2	0.0	0.005	32	relu	14.92	5.35
40	128	1	0.2	0.01	32	relu	14.94	4.97
41	64-32	2	0.1	0.005	32	relu	14.94	4.94
42	128-64-32	3	0.0	0.001	64	relu	14.97	5.59
43	32-16	2	0.0	0.005	64	relu	14.97	4.94
44	256-128-64	3	0.2	0.001	128	relu	14.99	5.06
45	256-128-64	3	0.0	0.005	64	relu	15.01	4.88
46	128	1	0.1	0.01	128	relu	15.03	5.10
47	128	1	0.1	0.005	32	relu	15.07	5.06
48	64-32-16	3	0.0	0.01	32	relu	15.11	4.80
49	128-64	2	0.1	0.005	64	relu	15.19	5.11
50	128	1	0.0	0.01	32	relu	15.20	6.23
51	128	1	0.0	0.01	128	relu	15.21	6.31
52	128-64	2	0.2	0.005	128	relu	15.23	5.22
53	128	1	0.2	0.005	64	relu	15.24	5.11
54	128	1	0.2	0.01	64	relu	15.26	5.00
55	128-64-32	3	0.0	0.005	32	relu	15.26	5.38
56	128-64-32	3	0.1	0.005	64	relu	15.29	4.85
57	64-32-16	3	0.0	0.01	128	relu	15.32	5.00
58	128-64	2	0.1	0.001	64	relu	15.32	5.19
59	128	1	0.0	0.005	128	relu	15.33	6.04

Se observa que las mejores configuraciones para ambos tipos de opciones comparten características comunes: todas utilizan la función de activación ReLU, arquitecturas de 2-3 capas ocultas con estructura decreciente, y learning rates entre 0.001 y 0.01. Para opciones PUT, las mejores configuraciones no utilizan dropout, mientras que para opciones CALL hay mayor variabilidad en este parámetro.

6.3. Matrices de Correlación

Este anexo presenta las matrices de correlación completas entre las variables del modelo, separadas por tipo de opción. Los colores rojos indican correlaciones negativas, los verdes correlaciones positivas, y la intensidad del color refleja la magnitud de la correlación. Su interpretación se desarrolla en el cuerpo del documento.

Figura 6.6: Matriz de correlaciones para opciones CALL. *Fuente:* Elaboración propia.

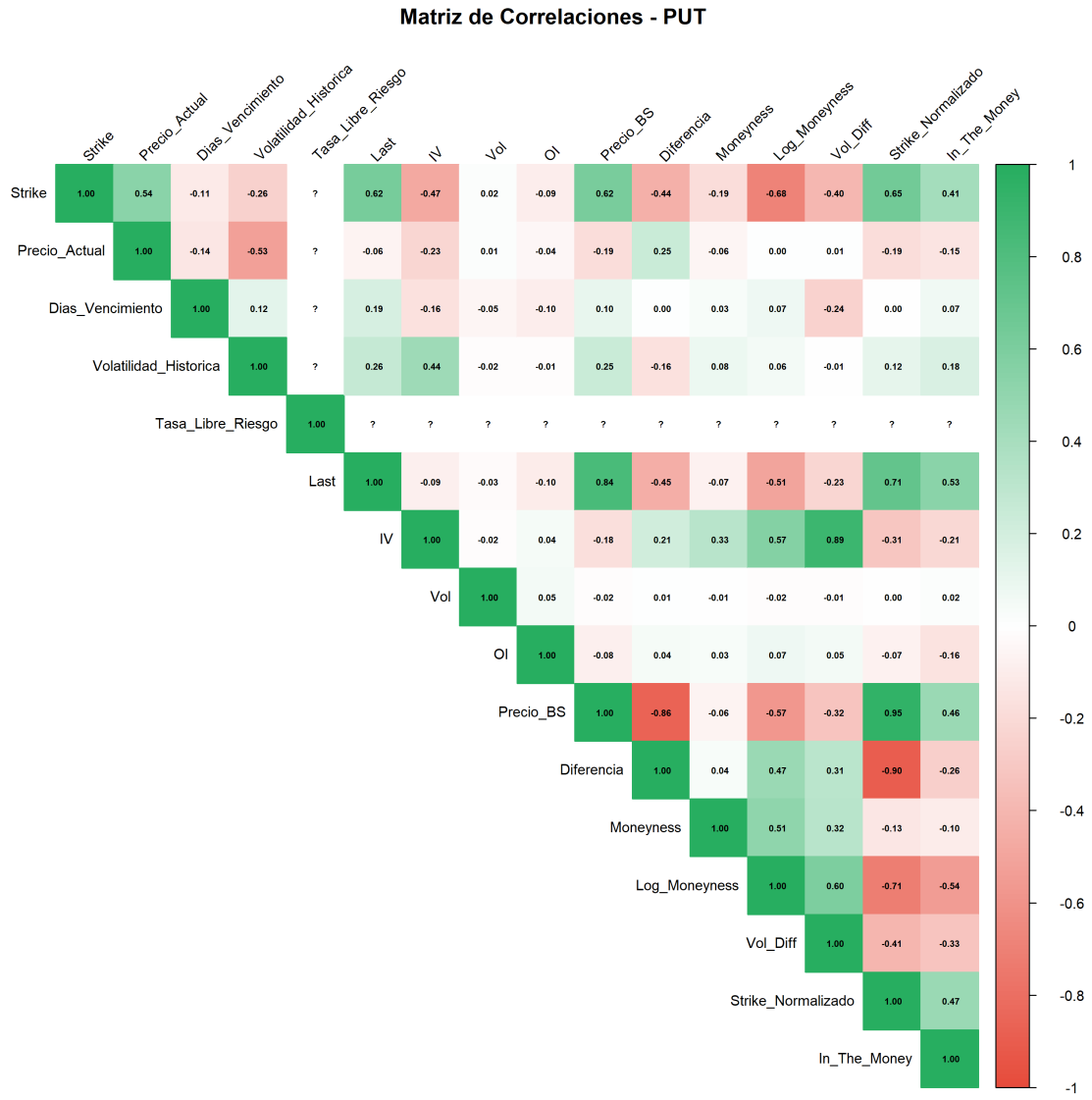


Figura 6.7: Matriz de correlaciones para opciones PUT. *Fuente:* Elaboración propia.

Bibliografía

[1] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining. London, UK: Springer, 2000, pp. 29–39.

[2] J. C. Hull, Options, Futures, and Other Derivatives, 11th ed. Pearson, 2022.

[3] M. Scholes and F. Black, "The pricing of options and corporate liabilities," Journal of Political Economy, vol. 81, no. 3, pp. 637–654, 1973.

[4] R. C. Merton, "Theory of rational option pricing," The Bell Journal of Economics and Management Science, vol. 4, pp. 141–183, 1973.

[5] A. L. Hutchinson and T. P. M. Lo, "A nonparametric approach to pricing and hedging derivative securities via learning networks," Journal of Finance, vol. 49, no. 3, pp. 851–889, 1994.

[6] J. Sirignano and K. Spiliopoulos, "Dgm: A deep learning algorithm for solving partial differential equations," Journal of Computational Physics, vol. 375, pp. 1339–1364, 2018.

[7] M. Capinski and T. Zastawniak, Mathematics for Finance: An Introduction to Financial Engineering. Poland: Springer, 2003.

[8] D. C. Montgomery, E. A. Peck, and G. G. Vining, Introduction to Linear Regression Analysis, 5th ed. New Jersey: Wiley, 2012.

[9] J. Marín Morales and L. A. Carrasco Ribelles, Introducción a los análisis estadísticos en R, 1st ed. Marcombo / Alphaeditorial, 2023.

[10] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[11] ——, "Bagging predictors," Machine Learning, vol. 24, no. 2, pp. 123–140, 1996.

[12] T. K. Ho, "The random subspace method for constructing decision forests," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no. 8, pp. 832–844, 1998.

[13] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA: ACM, 2016, pp. 785–794.

[14] ——, "XGBoost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, CA: ACM, 2016, pp. 785–794.

[15] T. Cover and P. Hart, "Nearest neighbor pattern classification," IEEE Transactions on Information Theory, vol. 13, no. 1, pp. 21–27, 1967.

[16] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed. New York: Springer, 2009.

[17] M. Kuhn and K. Johnson, Applied Predictive Modeling. New York: Springer, 2013.

[18] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, 2015.

[19] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016. [Online]. Available: https://www.deeplearningbook.org

[20] S. Haykin, Neural Networks and Learning Machines, 3rd ed. Upper Saddle River, NJ: Pearson, 2009.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in Proceedings of the 27th International Conference on Machine Learning (ICML), 2010, pp. 807–814.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, no. 6088, pp. 533–536, 1986.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Proceedings of the 3rd International Conference on Learning Representations (ICLR), 2015, arXiv:1412.6980.

[24] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," Journal of Machine Learning Research, vol. 15, no. 56, pp. 1929–1958, 2014.

[25] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," Journal of Machine Learning Research, vol. 13, no. 10, pp. 281–305, 2012.

[26] T. O. Hodson, "Root-mean-square error (rmse) or mean absolute error (mae)," Geoscientific Model Development, no. 15, 2022.

[27] Z. Wang and A. C. Bovik, "Mean squared error: Love it or leave it? a new look at signal fidelity measures," IEEE Signal Processing Magazine, no. 117, 2009.

[28] R. P. Palma, "Análisis crítico del coeficiente de determinación (r^2), como indicador de calidad de modelos lineales y no lineales," ESPOL - FCNM JOURNAL, vol. 20, no. 2, 2022.

[29] A. M. D'Uggento, M. Biancardi, and D. Ciriello, "Predicting option prices: From the black-scholes model to machine learning methods," Big Data Research, vol. 40, no. 100518, 2025.

[30] Y. Chen, X. Xu, T. Lan, and S. Shang, "The predictability of stock price: Empirical study on tick data in chinese stock market," Big Data Research, vol. 35, no. 100414, 2024.

[31] F. Ferreira, A. Gandomi, and R. Cardoso, "Artificial intelligence applied to stock market trading: A review," IEEE Access, vol. 9, 2021.

[32] S. Gu, B. Kelly, and D. Xiu, "Empirical asset pricing via machine learning," The Review of Financial Studies, vol. 33, no. 5, pp. 2223–2273, 2020.