



Modelo fundamental de crecimiento en utilidades y *price-to-earnings ratio*, P/E, de los índices accionarios internacionales

Fundamental model of earnings growth and price-to-earnings ratio, P/E, of international stock indices

Por

Daniel Sandino Perdomo¹

Daniel Jiménez Benítez²

Nicolás Navarrete Quintero³

Trabajo de grado presentado como requisito parcial para obtener el título de
Magíster en Administración Financiera – MAF

Asesores

Ph. D. Walter Díaz

Ph. D. María Patricia Durango

Universidad EAFIT

Escuela de Finanzas, Economía y Gobierno

Maestría en Administración Financiera – MAF

Medellín

2025

¹ dsandinop@eafit.edu.co

² djimenezb@eafit.edu.co

³ nnavarret@eafit.edu.co

© 2025 Daniel Sandino, Daniel Jiménez y Nicolás Navarrete
Todos los Derechos Reservados

Resumen

El S&P500, un barómetro de la economía estadounidense, es uno de los índices bursátiles más importantes del mundo. La relación precio-beneficio (*price-to-earnings ratio*, P/E) es una medida de valoración que compara el precio de mercado de una acción con sus ganancias por acción, y se utiliza comúnmente para evaluar si las acciones están sobrevaloradas o subvaloradas. Pronosticar la relación P/E es complejo debido a factores que pueden tener influencia en dicha relación: tasas de interés, crecimiento económico, sentimiento del mercado o proyecciones financieras sobre las empresas, entre otros.

En esta investigación se implementaron dos modelos de redes neuronales recurrentes: la LSTM (*long short-term memory*, memoria a corto plazo y larga duración) y la GRU (*gated recurrent unit*, unidad recurrente cerrada), así como dos modelos de *machine learning*: el XGBoost (*extreme gradient boosting*, aumento de gradiente extremo) y el LigthGBM (*light gradient boosting machine*, motor potenciador de gradiente ligera), para pronosticar la relación P/E del S&P500 utilizando datos históricos entre enero de 1990 y octubre de 2024.

Los resultados muestran que los cuatro modelos tienen un buen rendimiento, aunque el modelo GRU se destaca en términos de precisión y eficiencia computacional, sin dejar a un lado el modelo LigthGBM, un algoritmo de *boosting*, que también muestra resultados competitivos.

La investigación ofrece información valiosa sobre el uso de los cuatro modelos para pronosticar índices de valoración, y puede ser útil como apoyo en la toma de decisiones de inversión.

Palabras claves: LSTM, GRU, XGBoost, LightGBM, *boosting*, S&P500, pronóstico de acciones, relación precio-beneficio (*price-to-earnings ratio*, P/E).

Abstract

The S&P500, a barometer of the U.S. economy, is one of the world's leading stock market indices. The price-to-earnings ratio (P/E) is a valuation measure that compares a stock's market price to its earnings per share, and is commonly used to assess whether stocks are overvalued or undervalued. Forecasting the P/E ratio is complex due to factors that can influence the ratio: interest rates, economic growth, market sentiment, and financial projections for companies, among others.

In this research, two recurrent neural network models were implemented: LSTM (long short-term memory) and GRU (gated recurrent unit), as well as two machine learning models: XGBoost (extreme gradient boosting) and LightGBM (light gradient boosting machine), to forecast the P/E ratio of the S&P500 using historical data between January 1990 and October 2024.

The results show that all four models perform well, although the GRU model stands out in terms of accuracy and computational efficiency, without leaving aside the LightGBM model, a boosting algorithm, which also shows competitive results.

The research offers valuable information on the use of the four models to forecast valuation ratios, and can be useful as support in investment decision making.

Keywords: LSTM, GRU, XGBoost, LightGBM, boosting, S&P500, stock forecasting, P/E.

Contenido

1. Introducción	1
2. Marco teórico	4
2.1 <i>Price-to-earnings</i> (P/E)	4
2.2 Método tradicional de referencia Bai-Perron	4
2.3 Métodos basados en modelos de aprendizaje	6
2.3.1 Redes neuronales recurrentes.....	6
2.3.2 <i>Machine learning</i> (aprendizaje automático)	7
3. Metodología	11
3.1 Datos y preparación	11
3.2 Ejecución de los modelos	11
3.3 Criterio de comparación	12
4. Resultados	13
5. Conclusiones y recomendaciones.....	18
Referencias	21

Índice de tablas

Tabla 1. Estadística descriptiva de los datos de la P/E en el índice S&P500.....	1
Tabla 2. Evaluación de los Modelos mediante el MSE y el RMSE	17

Índice de figuras

Figura 1. P/E del índice S&P500 (01/01/1990-09/11-2022).....	2
Figura 2. Identificación de los regímenes de media utilizando el método Bai-Perron (1992-2024).....	5
Figura 3. Medias asociadas a cada régimen (1992-2024)	5
Figura 4. Comportamiento de una RNN con cuatro capas ocultas y una salida	6
Figura 5. Comparativo del método de crecimiento de los árboles de decisión.....	9
Figura 6. Modelo Treelite.....	10
Figura 7. Iteraciones de entrenamiento del modelo GRU con ajustes de hiper-parámetros	13
Figura 8. Entrenamiento del modelo GRU con los hiper-parámetros seleccionados (<i>epochs</i> : 50 – <i>batch_size</i> : 32).....	14
Figura 9. Iteraciones de entrenamiento del modelo LSTM con ajuste de hiper-parámetros.....	15
Figura 10. Entrenamiento de modelo LSTM con hiper-parámetros seleccionados (<i>epochs</i> : 60 – <i>batch_size</i> : 40).....	16
Figura 11. Datos históricos del P/E en comparación con los modelos LSTM, GRU, XGBoost y LightGBM y su comportamiento predictivo para las siguientes veinticinco semanas.	17

1. Introducción

El índice S&P500, compuesto por las 500 empresas con mayor capitalización de mercado que cotizan en la Bolsa de valores de Estados Unidos, es considerado un indicador clave de la economía norteamericana y, en gran medida, de la economía mundial (Abhyankar *et al.*, 1997). Pronosticar su comportamiento futuro y la rentabilidad esperada influye de forma significativa en la toma de decisiones de inversión a nivel global (Nagy *et al.*, 2024). La relación precio-beneficio (*price-to-earnings ratio*, en adelante la P/E) es una métrica fundamental en el análisis de valores que compara el precio de mercado de una acción con sus ganancias por acción. (Afza & Tahir, 2021)

La Tabla 1 muestra una estadística descriptiva de los datos de la P/E en el índice S&P500.

Tabla 1. Estadística descriptiva de los datos de la P/E en el índice S&P500

	P/E
Conteo	1818,0
Promedio	18,5
Desviación estándar	3,8
Mínimo	10,5
25 %	15,9
50 %	17,8
75 %	20,3
Máximo	28,5

Fuente: elaboración de los autores.

Esta relación es ampliamente utilizada por los inversores para evaluar la valoración del mercado y las perspectivas de crecimiento futuro de una empresa o de un índice bursátil. Históricamente, la relación P/E del S&P500 ha alcanzado un máximo de 28,55x y un mínimo de 10,46x, con una mediana de 17,82. Entre enero de 1990 y octubre de 2024, la relación P/E del S&P500 promedió 18,51x. Para esta investigación, la métrica utilizada fue *trailing price-to-earnings* (trailing P/E, relación precio-beneficio rezagada).

La Figura 1 muestra la P/E del índice S&P500 entre el 1 de enero de 1990 y el 9 de noviembre de 2022.

Figura 1. P/E del índice S&P500 (01/01/1990-09/11-2022)



Fuente: elaboración de los autores.

El análisis financiero y las técnicas de predicción mediante el uso de modelos de *machine learning* son fundamentales para la toma de decisiones de inversión, en tanto son procesos impulsados por avances tecnológicos aplicados al estudio de series de tiempo complejas utilizando datos históricos de índices financieros destacados como el S&P500, y que han demostrado ser más exactos que los métodos tradicionales de predicción. (Nagy *et al.*, 2024)

Siendo la transformación digital un aspecto clave para la optimización de la gestión financiera, en esta investigación se utilizaron dos modelos de predicción basados en redes neuronales de tipo recurrente: la LSTM (*long short-term memory*, memoria a corto plazo y larga duración) y la GRU (*gated recurrent unit*, unidad recurrente cerrada), y dos modelos de *machine learning*: el XGBoost (*extreme gradient boosting*, aumento de gradiente extremo) y el LigthGBM (*light gradient boosting machine*, motor potenciador de gradiente ligera). Al presentar exactitud de la data con márgenes de error muy bajos y adaptarse a las fluctuaciones del mercado, estos modelos son ampliamente utilizados por su alta precisión y eficacia comprobada para la predicción de series temporales en diversas disciplinas. Por tanto, en esta investigación serán aplicados para la predicción de la relación P/E del indicador S&P500. (Andrade Gorjoux *et al.*, 2023)

Los resultados de estos cuatro modelos proporcionan un importante soporte para evaluar las decisiones financieras que los inversionistas puedan tomar y, además, pueden contribuir al mejoramiento económico de un mundo que se encuentra en constante transformación, enfocado en la optimización de los recursos, teniendo en cuenta la importancia de la elección adecuada de las técnicas de predicción.

Para alcanzar este objetivo se implementó inicialmente una metodología que incluye la recopilación de datos históricos del índice S&P500, abarcando el precio de cierre diario y las

ganancias por acción, entre enero de 1990 y octubre de 2024. Seguidamente se realizó una limpieza y una normalización de los datos para asegurar la calidad y la compatibilidad con los cuatro modelos de predicción (LSTM, GRU, XGBoost y LightGBost); estos modelos se entrenaron y evaluaron para determinar su precisión y eficiencia en el pronóstico del índice P/E. Posteriormente se presentó un análisis de los resultados de los modelos para comparar su rendimiento y determinar su eficacia para la toma de decisiones de inversión. Y, finalmente, se expusieron las conclusiones y recomendaciones basadas en los hallazgos de la investigación.

2. Marco teórico

El objetivo de esta investigación es generar una herramienta que permita pronosticar de manera acertada el comportamiento de la P/E haciendo uso de herramientas avanzadas de computación.

El interés de pronosticar esta ratio no es nuevo. De hecho, hoy día existen herramientas menos refinadas fundamentadas en el análisis descriptivo de la data. Estos métodos tradicionales son efectivos en horizontes que pueden alcanzar varios ciclos económicos —por ejemplo, más de cinco años—, y son un buen referente comparativo para «evaluar» si con las nuevas herramientas serían susceptibles de ser remplazados.

2.1 *Price-to-earnings* (P/E)

En el análisis técnico para la colocación de capital en renta variable es común revisar la variación histórica de la relación entre el precio de mercado de una acción y las ganancias unitarias que generó en el último período, puesto que esta relación suele mantener dos características esenciales para valorar un activo:

- Es sectorial: significa que empresas de diferentes sectores conservan una ratio similar, haciendo posible comparar cada empresa contra un rasero del sector e identificar de manera rápida si el activo está sobrevalorado o subvalorado.
- Es estable en el tiempo: esta es, quizás, la aseveración más importante y que debe observarse de manera cuidadosa, pues, si bien es cierto que los datos históricos respaldan este postulado, también lo es que dicha media no es perfectamente estable en el tiempo y, por el contrario, puede desplazarse a nuevos regímenes motivados por cambios fundamentales.

2.2 Método tradicional de referencia Bai-Perron

En 1998, el economista Jushan Bai y el estadístico Pierre Perron observaron que existen datos históricos que presentan comportamientos definidos moderados que migran a un estado o régimen diferente en distintos rangos de tiempo sin que esto necesariamente afecte la naturaleza de su comportamiento. (Bai & Perron, 1998)

Este es el caso de la relación P/E que, se sabe, fluctúa alrededor de una media histórica definida, pero que, impulsado por factores fundamentales, puede migrar a una media diferente —un cambio de régimen— en algún momento del tiempo y conservar la condición de buscar esa nueva media a lo largo del tiempo hasta que algún fenómeno diferente la vuelva a afectar.

Fue así como estos autores presentaron un método para ubicar dichos cambios de régimen en una serie temporal y, en el caso específico de la P/E, calcular, para cada régimen, la media a la que debería regresar. (Carlson *et al.*, 2002)

Considerando la propuesta presentada por estos autores, se aplicó el método Bai-Perron a la base de datos en estudio y se identificaron los regímenes que se muestran en la Figura 2.

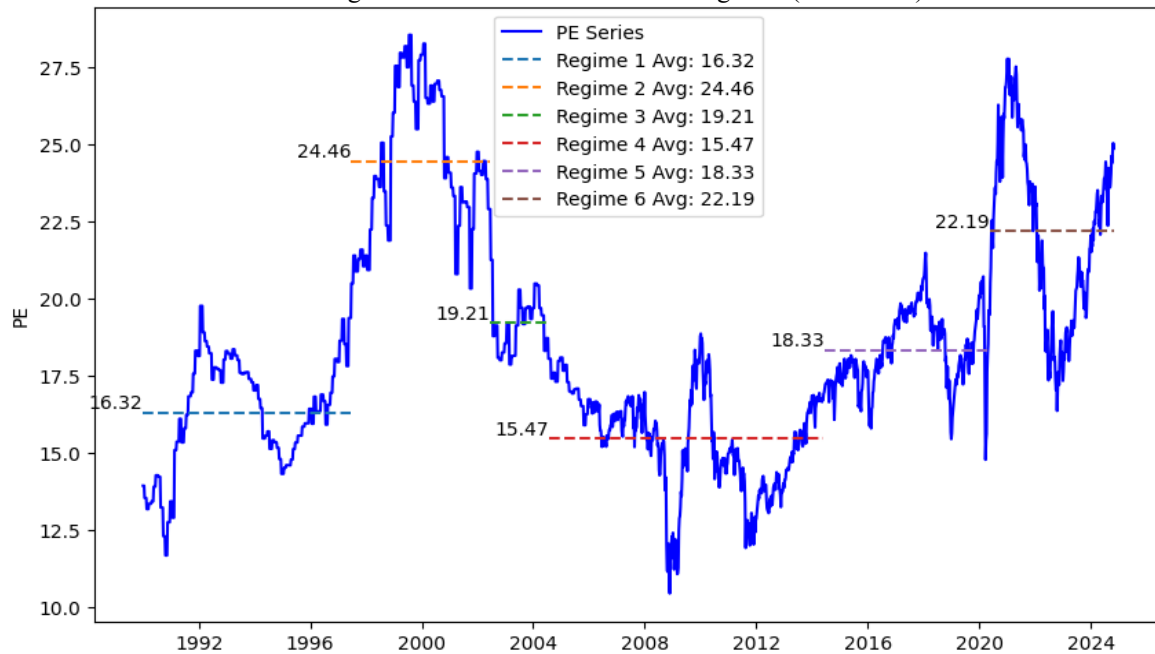
Figura 2. Identificación de los regímenes de media utilizando el método Bai-Perron (1992-2024)



Fuente: elaboración de los autores.

A los regímenes de media de la Figura 2 les corresponden las siguientes medias, que se muestran en la Figura 3.

Figura 3. Medias asociadas a cada régimen (1992-2024)



Fuente: elaboración de los autores.

La forma de interpretar el resultado de la Figura 3 es la siguiente: considerando que el último régimen contempla una media de 22,19, y asumiendo el postulado de «reversión a la media», se puede inferir que, a la fecha de su elaboración (28 de octubre de 2024), el P/E de 24,86 tenderá a bajar para recuperar el nivel de 22,19 del régimen actual, con la salvedad de que dicho régimen irá a su vez buscando, aunque más lentamente, el valor actual. De cualquier manera, la conclusión es que el P/E deberá bajar; pero, la verdadera pregunta es ¿cuándo? Por esta razón, los modelos de predicción objeto de esta investigación son de gran importancia para lograr encontrar estas tendencias y tomar decisiones más acertadas.

2.3 Métodos basados en modelos de aprendizaje

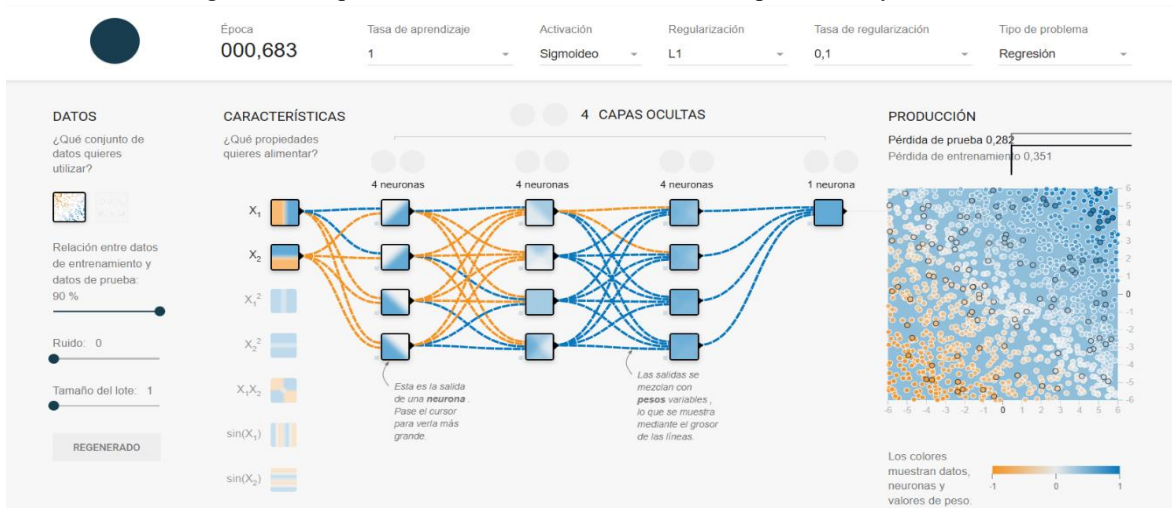
2.3.1 Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN, *recurrent neural networks*) son un tipo de red neuronal artificial diseñada para procesar secuencias de datos. A diferencia de las redes neuronales tradicionales, las RNN tienen conexiones recurrentes que les permiten mantener un estado interno y utilizar información de pasos de tiempo anteriores para influir en la salida actual. Esta propiedad las hace ideales para analizar datos secuenciales como series temporales, donde el orden de los datos es crucial.

Según Sirisha (2022), las RNN son efectivas en tareas de series de tiempo —como las predicciones financieras—, en tanto muestran mejores resultados en comparación con métodos de predicción como el Arima (*autoregressive integrated moving average*, media móvil integrada autorregresiva) y el Sarima (*seasonal autoregressive integrated moving average*, media móvil autorregresiva integrada estacional).

La Figura 4 muestra el comportamiento de una RNN con cuatro capas ocultas y una salida.

Figura 4. Comportamiento de una RNN con cuatro capas ocultas y una salida



Fuente: Smilkov & Carter (s. f.).

LSTM (long short-term memory, memoria a corto plazo y larga duración)

La LSTM es un tipo de red neuronal recurrente (RNN) que aborda el problema de la desaparición del gradiente, que dificultaba el aprendizaje de dependencias a largo plazo en las RNN tradicionales. Cruz Martínez *et al.* (2024) definieron las RNN como una clase de aprendizaje profundo conocidas por su capacidad para procesar y obtener información de datos secuenciales. La LSTM introduce un mecanismo de «compuertas» que controla el flujo de información a través de la red, permitiendo que esta recuerde información durante largos períodos y aprenda patrones complejos en series temporales.

Las redes LSTM tienen una estructura de memoria celular compuesta por tres compuertas principales, así:

- Compuerta de entrada: regula la entrada de nueva información a la memoria celular.
- Compuerta de olvido: controla qué información se elimina de la memoria celular.
- Compuerta de salida: determina qué información de la memoria celular se utiliza para generar la salida.

Estas compuertas permiten que la LSTM aprenda selectivamente qué información es importante recordar y cuál puede descartar, lo que la hace efectiva para modelar series temporales con dependencias a largo plazo y son efectivas en tareas de series de tiempo como la predicción financiera. (Andrade Gorjoux *et al.*, 2023)

La LSTM ha sido aplicada para el diseño de un modelo de predicción de los precios de las acciones y de múltiples índices bursátiles; ejemplos de ellos son la predicción de series de tiempo en la Bolsa de valores de Teherán (Irán) y el desarrollo de un modelo de pronóstico a corto plazo con una alta precisión de las series del mercado de valores de Estados Unidos y China. (Pimentel Ramírez, 2024)

GRU (gated recurrent unit, unidad recurrente cerrada)

Con el transcurrir de los años y la popularidad de las redes LSTM, en 2014 surgió el modelo GRU, otro tipo de red neuronal recurrente (RNN) que también utiliza compuertas para controlar el flujo de información, pero con una estructura más simple y mejorada que la de la LSTM. La GRU combina las compuertas de entrada y olvido de la LSTM en una sola compuerta de «actualización» e introduce una compuerta de «reinicio» que controla la cantidad de información pasada que se retiene.

Esta simplificación en su arquitectura resulta en una mayor eficiencia computacional en comparación con la LSTM, sin sacrificar significativamente el rendimiento en muchas tareas. (Pimentel Ramírez, 2024)

2.3.2 *Machine learning* (aprendizaje automático)

El *machine learning* es una rama de la inteligencia artificial (IA) que está impactando en todas las áreas del conocimiento y ha tenido un crecimiento exponencial en los últimos años. Esta técnica se centra en la creación y evolución de modelos y algoritmos y, específicamente, es un algoritmo de

aprendizaje automático que lleva a cabo un proceso matemático computacional que le permite a un sistema mejorar automáticamente a partir de unos datos de entrada sin la intervención explícita de una persona. (Osorio Buitrón & Rico, 2024)

XGBoost (extreme gradient boosting, aumento de gradiente extremo)

El XGBoost es un modelo de *machine learning* basado en árboles de decisión que opera con un algoritmo de aprendizaje automático, y ha ganado popularidad en los últimos años debido a su alto rendimiento en una variedad de tareas, incluyendo la predicción de series temporales.

Según XGBoost Tutorials (2002), este modelo pertenece a la familia de los algoritmos de *boosting*, que se basan en la idea de combinar varios modelos «débiles» —como los árboles de decisión «simples»— para crear un modelo «fuerte» con mayor precisión. Así, el XGBoost implementa una versión mejorada del algoritmo de *boosting* de gradiente, con las siguientes características:

- Regularización: utiliza técnicas de regularización para evitar el sobreajuste, lo que lo hace más robusto y generalizable.
- Manejo de valores faltantes: puede manejar valores faltantes de forma eficiente, lo que es importante en conjuntos de datos del mundo real.
- Paralelismo: puede aprovechar el paralelismo para acelerar el entrenamiento, lo que lo hace adecuado para conjuntos de datos grandes.

El modelo XGBoost se destaca por su capacidad para modelar relaciones no lineales y complejas en los datos, su eficiencia en el manejo de grandes volúmenes de información, y su potencial para proporcionar resultados predictivos robustos y precisos. (Correa Restrepo, 2024)

LightGBM (light gradient boosting machine, motor potenciador de gradiente ligera)

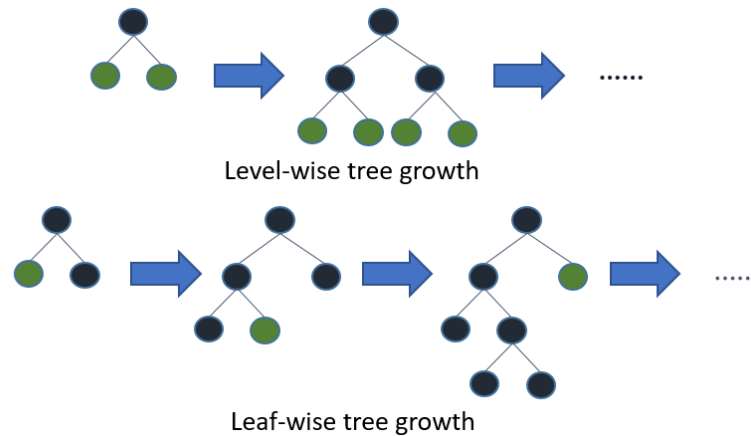
El LightGBM es una estructura *open-source* de *machine learning* de alto rendimiento creada por Microsoft, con el objetivo de optimizar la velocidad y la eficiencia del uso del recurso computacional, lo que lo ha convertido en una de las estructuras más populares en el manejo de datos estructurados.

Al igual que el XGBoost, este modelo hace parte de la familia de los algoritmos *boosting*; sin embargo, se diferencia de este en que utiliza un algoritmo basado en histogramas que almacena los datos en bandejas para reducir el tiempo y el requerimiento de cómputo.

De igual manera, el algoritmo LightGBM supone un crecimiento de árboles por hojas en lugar de un crecimiento por niveles —como lo suelen hacer algoritmos basados en árboles de decisión—, lo que reduce el requerimiento de recursos. (Microsoft Corporation, 2025)

La Figura 5 muestra un comparativo del método de crecimiento de los árboles de decisión.

Figura 5. Comparativo del método de crecimiento de los árboles de decisión



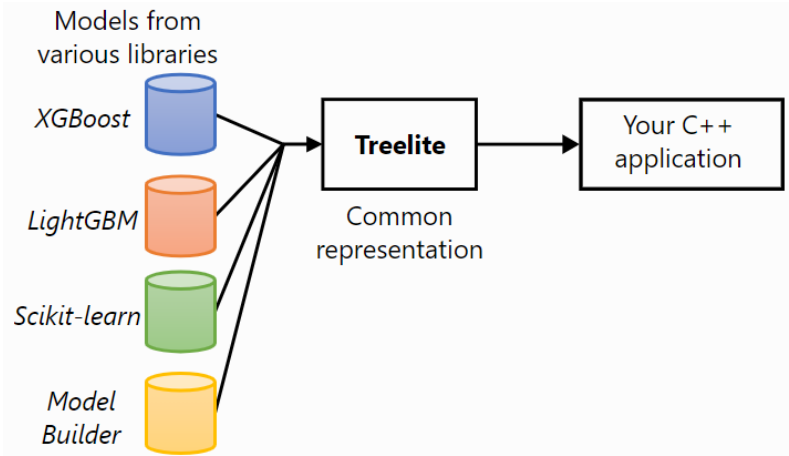
Fuente: Microsoft Corporation (2025).

El algoritmo LightGBM tiene las siguientes características:

- Aprendizaje basado en histogramas: en lugar de evaluar cada punto de datos individualmente, agrupa los datos en *bins* (archivos binarios) discretos, reduciendo el costo computacional.
- División por hoja (*leaf-wise splitting*): a diferencia del crecimiento nivel a nivel —como en el XGBoost y el LightGBM—, crece los árboles de forma hoja a hoja, lo que mejora la eficiencia y la precisión.
- Manejo de grandes volúmenes de datos: está diseñado para conjuntos de datos grandes y de alta dimensionalidad.
- Soporte para GPU (*graphics processing units*, unidades de procesamiento gráfico): puede aprovechar las GPU para un entrenamiento más rápido.
- Manejo nativo de variables categóricas: no es necesario usar *one-hot encoding* (codificación *one-hot*), ya que maneja directamente las variables categóricas.
- Menor consumo de memoria: usa menos memoria en comparación con el XGBoost.
- Una alternativa sería utilizar Treelite, que conjuga XGBoost, LightGBM, Scikit-learn y Model Builder, pero es una opción que, al momento de ejecución de esta investigación no se encuentra disponible para Python, aunque valdría la pena considerar esta opción para el futuro.

La Figura 6 muestra el modelo Treelite.

Figura 6. Modelo Treelite



Fuente: Microsoft Corporation (2025).

3. Metodología

3.1 Datos y preparación

Los datos utilizados en esta investigación incluyen el P/E Trailing, que utiliza los datos reales medidos de las utilidades realizadas durante los últimos doce meses según la información financiera presentada por las empresas que componen el índice S&P500. Cabe recordar que se tomaron los datos históricos del S&P500 desde el 1 de enero de 1990 hasta el 28 de octubre de 2024, en frecuencia semanal.

Para la correcta implementación de los modelos LSTM, GRU, XGBoost y LightGBM, se llevó a cabo un proceso de limpieza y preparación de los datos que se describe a continuación.

- 1) Verificación de fechas y formato: se verificó que la columna de fechas ("Name") estuviera en el formato correcto (`datetime64[ns]`) y se transformó la columna del índice P/E ("`@:USSP500/@:USSP500 (A12TE)`") a valores numéricos (`float64`) utilizando la función "`pd.to_numeric()`".
- 2) Establecimiento del índice: se estableció la columna de fechas como índice del DataFrame para facilitar el análisis y la manipulación de la serie temporal. Además, en el caso de los modelos XGBoost y LightGBM, se descompuso la fecha en día, mes y año, a fin de facilitar la identificación de comportamientos.
- 3) Normalización de datos: se utilizó la clase `MinMaxScaler` de la *library* `Sklearn.preprocessing` para normalizar los valores del índice P/E en un rango de 0 a 1. Este recurso ayuda a mejorar el rendimiento de los modelos de redes neuronales.
- 4) Creación de secuencias: se crearon secuencias de datos con un tamaño de ventana de 30 pasos de tiempo, utilizando la función "`create_sequences()`". Esto significa que cada secuencia de entrada consta de 30 puntos de datos consecutivos, y que el modelo se entrenará para predecir el siguiente punto de datos.
- 5) División en conjuntos de entrenamiento y prueba: se dividió el conjunto de datos en conjuntos de entrenamiento (90 %) y de prueba (10 %) para evaluar el rendimiento de los modelos.

3.2 Ejecución de los modelos

Como se explicó, el objetivo de los modelos es acortar el horizonte del pronóstico y mejorar su resolución; es decir, que ya no basta con suponer que la ratio va a buscar una media que no solo puede desplazarse y cambiar de régimen, sino que, además, dicho pronóstico puede tomar muchos años desperdiciando oportunidades claras de aprovechamiento del mercado. Por esta razón se planteó que los modelos deben ser capaces de ofrecer pronósticos claros y seguros de tendencia a un plazo de seis meses o veinticinco semanas.

De esta manera, se implementaron dos modelos de redes neuronales recurrentes: la LSTM y el GRU, y dos modelos de *machine learning*: el XGBoost y el LightGBM, para pronosticar el índice de valoración P/E. Los modelos se entrenaron utilizando el 90 % de los datos y se probaron utilizando el 10 % restante.

Para el modelo LSTM se utilizó una arquitectura de cuatro capas LSTM con 40 neuronas cada una, y se aplicó un *dropout* del 20 % después de cada capa LSTM para prevenir el sobreajuste. La elección de esta arquitectura se basó en la experimentación y en la búsqueda de un equilibrio entre la capacidad del modelo para capturar patrones complejos y la eficiencia computacional.

Para el modelo GRU se utilizó una arquitectura similar: cuatro capas GRU con 40 neuronas cada una y un *dropout* del 20 %. Se utilizó la función "early stopping" para detener el entrenamiento del modelo cuando dejaba de mejorar en el conjunto de validación y obtener los mejores pesos.

Para el modelo XGBoost se utilizó la función "xgb.XGBRegressor()" de la *library* XGBoost. Se utilizaron 100 estimadores con una tasa de aprendizaje de 0,1 y una profundidad máxima de 5.

Finalmente, para el modelo LightGBM se utilizó la función "LGBMRegressor()" de la *library* LightGBM, con un *random_state* de 42.

3.3 Criterio de comparación

Los modelos se evaluaron utilizando el error cuadrático medio (MSE, *mean square error*), una medida de la diferencia entre los valores pronosticados y los valores reales (Thompson, 1990). Se eligió el MSE como métrica de evaluación debido a su capacidad para penalizar errores grandes y su amplio uso en problemas de regresión.

4. Resultados

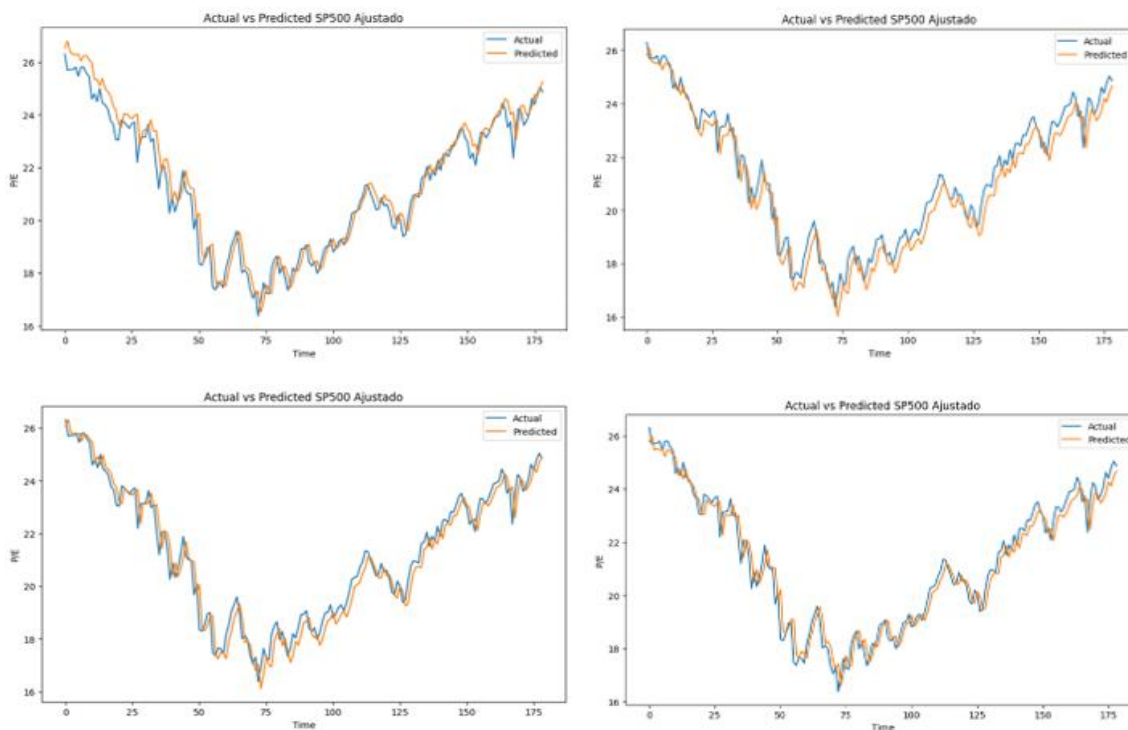
Para lograr los resultados de las predicciones de la ratio P/E, además de la normalización de datos mencionada, se hicieron seis pruebas de ensayo y error con ajuste de hiper-parámetros. En este caso, se ajustaron las *epochs* (épocas) y el *batch size* (tamaño del lote) para entrenar el modelo GRU y así poder obtener la menor pérdida. que se validó mediante el error cuadrático medio (MSE).

Las iteraciones fueron las siguientes: *epochs*: 30 – *batch_size*: 24; *epochs*: 25 – *batch_size*: 24; *epochs*: 25 – *batch_size*: 30; *epochs*: 30 – *batch_size*: 30; *epochs*: 30 – *batch_size*: 25 y *epochs*: 50 – *batch_size*: 32.

El mejor resultado de entrenamiento del modelo de predicción se obtuvo con la iteración *epochs*: 50 – *batch_size*: 32.

La Figura 7 muestra las iteraciones de entrenamiento del modelo GRU con ajuste de hiper-parámetros.

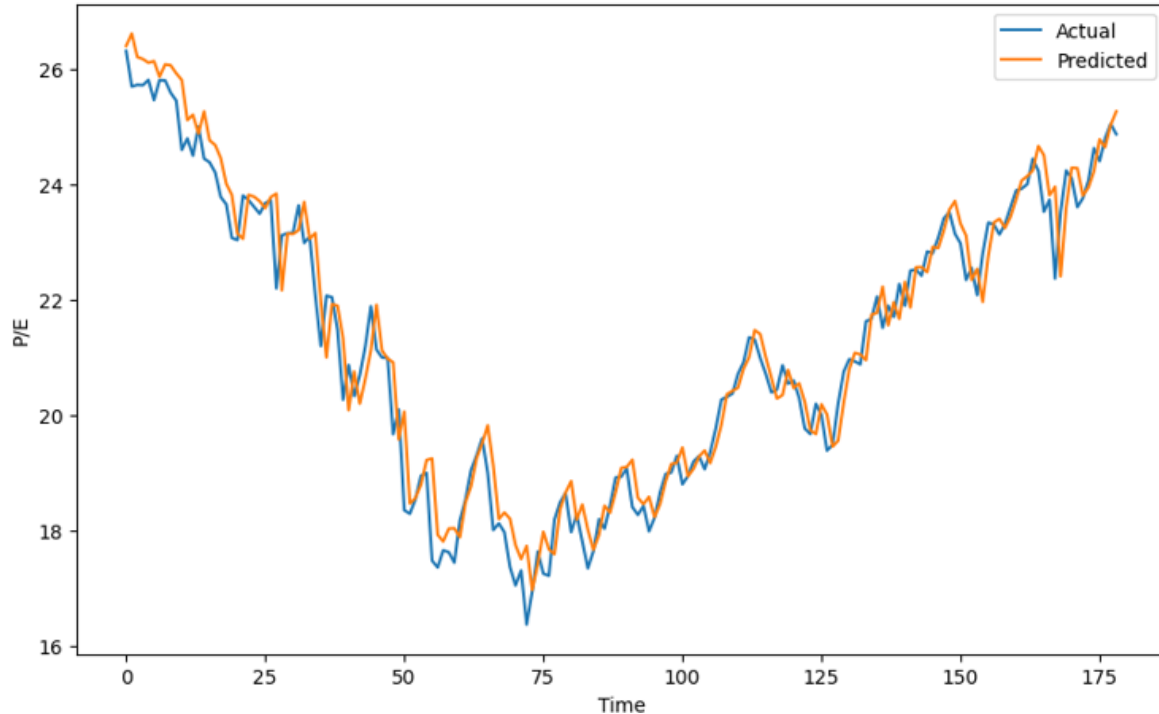
Figura 7. Iteraciones de entrenamiento del modelo GRU con ajustes de hiper-parámetros



Fuente: elaboración de los autores.

La Figura 8 muestra el entrenamiento del modelo GRU con los hiper-parámetros seleccionados (*epochs*: 50 – *batch_size*: 32).

Figura 8. Entrenamiento del modelo GRU con los hiper-parámetros seleccionados (*epochs*: 50 – *batch_size*: 32)
Actual vs Predicted SP500 Ajustado



Fuente: elaboración de los autores.

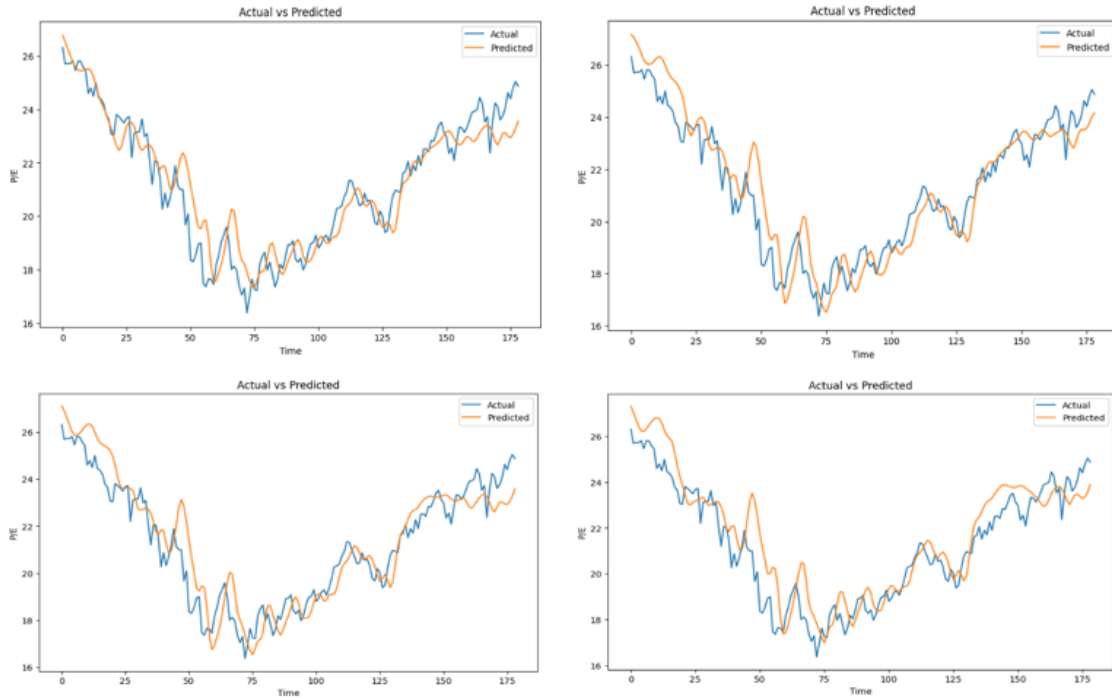
Para el entrenamiento del modelo LSTM se hicieron seis pruebas de ensayo y error con ajuste de hiper-parámetros. En este caso se ajustaron las *epochs* y el *batch_size*, a fin de obtener la menor pérdida, que se validó mediante el error cuadrático medio (MSE).

Las iteraciones fueron las siguientes, *epochs*: 20 – *batch_size*: 30; *epochs*: 24 – *batch_size*: 30; *epochs*: 30 – *batch_size*: 30; *epochs*: 30 – *batch_size*: 25, *epochs*: 50 – *batch_size*: 32, *epochs*: 60 – *batch_size*: 40.

El mejor resultado de entrenamiento del modelo de predicción se obtuvo con la iteración *epochs*: 60 – *batch_size*: 40.

La Figura 9 muestra las iteraciones de entrenamiento del modelo LSTM con ajuste de hiper-parámetros.

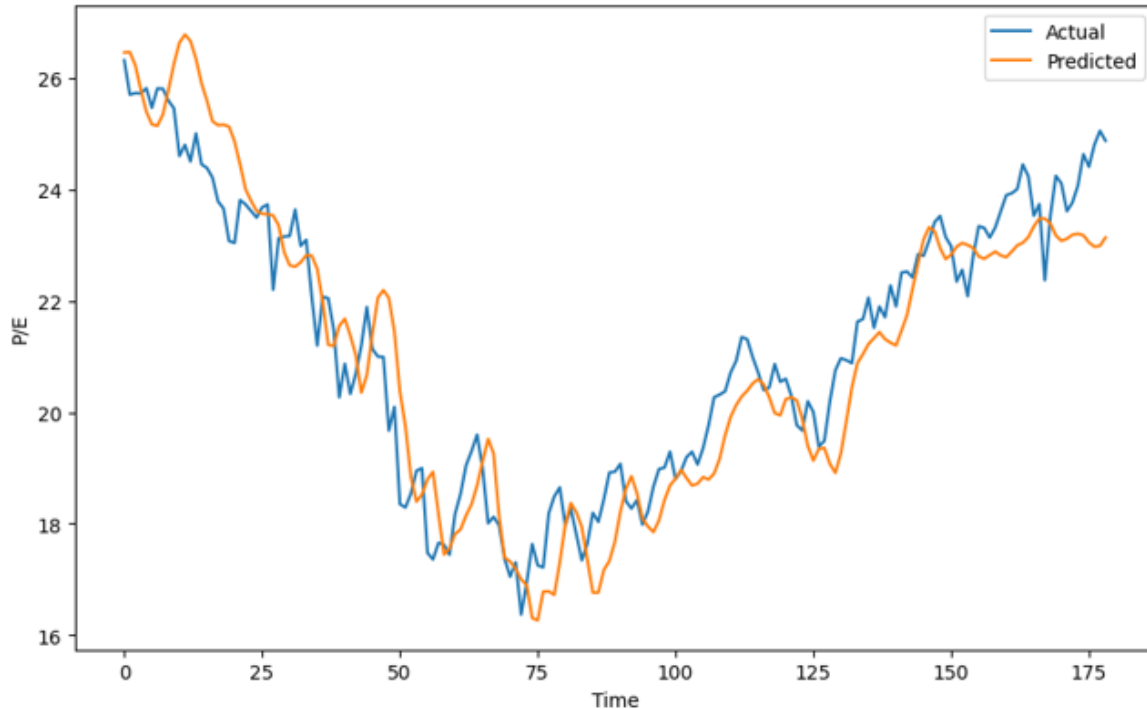
Figura 9. Iteraciones de entrenamiento del modelo LSTM con ajuste de hiper-parámetros



Fuente: elaboración de los autores.

La Figura 10 muestra el entrenamiento de modelo LSTM con hiper-parámetros seleccionados (*epochs: 60 – batch_size: 40*).

Figura 10. Entrenamiento de modelo LSTM con hiper-parámetros seleccionados (*epochs: 60 – batch_size: 40*)
Actual vs Predicted



Fuente: elaboración de los autores.

Para el entrenamiento del modelo XGBoost se escogió el error cuadrático medio (MSE) como medida de ajuste y se utilizaron 100 árboles de decisión, donde cada uno contribuía con el 10 % de corrección sobre el árbol anterior, todo ello con una profundidad máxima de 5 para evitar el sobreajuste.

En el caso del modelo LightGBM se escogió un *random_state* de 42, lo que significa que cada vez que se reinicie, los datos aleatorios serán iguales para garantizar su repetibilidad.

Por defecto, se determinaron los siguientes parámetros:

- *n_estimators* = 100: genera 100 ciclos.
- *Learning_rate* = 0,1: igual que con el XGBoost. es el 10 % de corrección de cada ciclo
- *Max. depth* = -1: significa que no hay límite de profundidad *log*, lo que a su vez significa que el árbol se ramificará hasta que se alcance el criterio seleccionado.
- *Num. leaves* = 31: limita la complejidad de los árboles.
- *Boosting_type* = 'gbdt': usa árboles de decisión del tipo *gradient boosting*.
- *Objective* = 'regression': indica que se trata de una regresión y no de una caracterización.

Los resultados muestran que los cuatro modelos (LSTM, GRU, XGBoost y LightGBM) tienen un buen rendimiento en el pronóstico del índice P/E.

El modelo GRU funciona mejor que el modelo LSTM en términos de precisión y eficiencia computacional, y esto puede deberse a la estructura más simple de la GRU, que requiere menos recursos computacionales para el entrenamiento y la predicción.

El modelo XGBoost, a pesar de ser un modelo de *machine learning*, tuvo un rendimiento similar a los modelos LSTM y GRU, mientras que el modelo LightGBM tuvo un *mean square error* (MSE) menor que los demás modelos, ajustándose de manera más exacta al entrenamiento y al testeo de los datos. En definitiva, su rendimiento resulta ser superior al de las redes neuronales recurrentes para este caso de estudio.

La Tabla 2 muestra un comparativo de la evaluación y el rendimiento de los modelos LSTM, GRU, XGBoost y LightGBM.

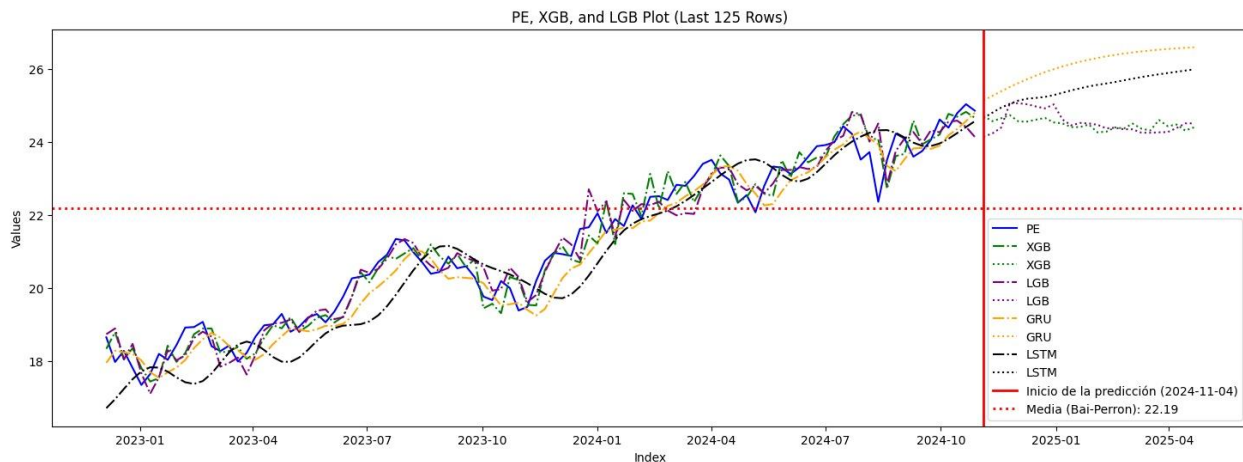
Tabla 2. Evaluación de los modelos mediante el MSE (*mean square error*) y el RMSE (*root mean square error*)

MODELO	MSE	RMSE	RANK
LSTM	1,073111	1,035911	4
GRU	0,355965	0,596628	2
LightGBM	0,078072	0,279414	1
XGBoost	0,385588	0,620957	3

Fuente: elaboración de los autores.

La Figura 11 muestra los datos históricos del P/E en comparación con los modelos LSTM, GRU, XGBoost y LightGBM y su comportamiento predictivo para las siguientes veinticinco semanas.

Figura 11. Datos históricos del P/E en comparación con los modelos LSTM, GRU, XGBoost y LightGBM y su comportamiento predictivo para las siguientes veinticinco semanas.



Fuente: elaboración de los autores.

5. Conclusiones y recomendaciones

La investigación se apalancó en los modelos LSTM, GRU, XGBoost y LightGBM para estimar el comportamiento del P/E, con el objetivo de aportar un fundamento más a la toma de decisiones de inversión. En tanto cada modelo procura un método diferente para presentar dicha estimación, fue importante no solo analizar las estimaciones en grupo y encontrar consenso, sino, además, aplicar el criterio y afianzar la confianza en el uso repetido de las herramientas. En otras palabras, así como los modelos se entrenan con los datos, es responsabilidad del analista evaluar las distintas predicciones y decidir y reevaluar la eficacia de cada uno de los modelos una vez el plazo de la predicción llegue a término.

De igual manera, si bien se considera que modelos como el XGBoost y el Light GBM son modelos de «caja negra», bien vale la pena alentar al usuario a modificar sus hiper-parámetros para afinarlos.

La investigación proporcionó información sobre el uso de modelos LSTM, GRU, XGBoost y LightGBM para pronosticar índices de valoración de las acciones. Los resultados muestran que los cuatro modelos pueden ser utilizados para pronosticar el índice P/E, aunque con diferencias en su comportamiento y precisión. Si bien aquellos basados en *boosting* presentan ventajas en términos de precisión, su interpretabilidad es compleja en comparación con modelos más simples, pues, al ser modelos de «caja negra», son difíciles de interpretar y entender para el conocimiento humano. A pesar de las diferencias en las proyecciones a corto plazo, los modelos tienden a converger en el largo plazo, reforzando su utilidad para la toma de decisiones de inversión.

Como se evidencia en la Tabla 2, la evaluación de los modelos mediante el MSE y el RMSE, el modelo LightGBM presentó un error cuadrático medio de 0,078072 menor en comparación a los demás modelos. Esto demuestra que es el modelo cuya predicción mejor se ajusta a los datos históricos. Lo anterior no garantiza que sea el mejor modelo debido a que existe el riesgo de *sobreajuste*, sin embargo, para mitigar este riesgo se utilizaron técnicas de regularización dentro de los modelos como Dropout, Early Stopping y Regularización L2 (Ridge).

En la Figura 11 se observa que la predicción de los modelos LSTM y GRU tiene una tendencia alcista en el corto plazo mientras que los modelos XGBoost y LightGBM predicen una estabilización del ratio en el corto plazo y convergen en la misma.

Las predicciones se basaron en un horizonte de tiempo de 25 semanas que equivalen a un semestre. Para observar las tendencias de los modelos LSTM, GRU, XGBoost y LightGBM en un horizonte de largo plazo, es posible reconfigurar el número de semanas a predecir.

Aunque en la fase de planteamiento de los modelos se intentó introducir variables como Desempleo, Confianza del consumidor y Precio de los bonos, entre otras —que se consideraron

«explicativas» o correlacionadas al P/E—, se encontró que estas variables, en lugar de mejorar el ajuste del modelo, rápidamente eran purgadas en sus pesos —el nivel de aporte al modelo— para, finalmente, resultar en ruido. De cualquier manera, se reconoce que, sin otras variables en el análisis, el modelo depende exclusivamente de la capacidad de identificar estacionalidades, por lo que se sugiere profundizar un poco más en la incorporación de variables que puedan aportar a la arquitectura de los modelos.

Se recomienda que los inversionistas consideren el uso de estos modelos como una herramienta complementaria en su análisis financiero, sin depender exclusivamente de ellos, ya que la relación P/E es solo una medida de valoración y no debe ser la única que se use a la hora de tomar decisiones de inversión. Además, es importante recordar que el mercado accionario es altamente impredecible y está sujeto a eventos inesperados y fundamentales como cambios en la política monetaria o incertidumbres geopolíticas, que pueden alterar significativamente el comportamiento proyectado por estos modelos.

Para mejorar la precisión y la aplicabilidad de los modelos predictivos, se recomienda utilizar un sistema que se actualice continuamente con datos reales, permitiendo ajustes dinámicos y predicciones más confiables. Adicionalmente, se sugiere revisar constantemente el estado del arte para incorporar nuevas herramientas con modelos diferentes, o que, de manera más elaborada, se conjuguen los modelos existentes, como es el caso de Treelite, un modelo que permite optimizar y acelerar los modelos basados en árboles de decisión, facilitando su integración en sistemas de inversión en tiempo real, o de CatBoost, especializado en series temporales. Explorar modelos híbridos y técnicas avanzadas de aprendizaje automático podría mejorar los resultados en la predicción de los indicadores financieros y la gestión de portafolios.

Analizando financieramente desde un punto de vista netamente técnico, se observó que el P/E, según la predicción, tenderá a estabilizarse el primer semestre del 2025 alrededor de 24,5, lo que implicaría que el precio sigue estando por encima de la media histórica del último régimen (22,19) y, por lo tanto, tiene un gran lastre técnico que lo llevará a bajar eventualmente con la consecuente caída en el precio del índice.

Desde un ángulo más fundamental, es necesario descontar el «efecto Trump» que, desde el inicio de 2025 viene alterando el comportamiento del mercado y de la industria norteamericana con la imposición de aranceles a las importaciones extranjeras y la inevitable retaliación por parte de los países afectados, que encarecen no solo las exportaciones, sino la importación de materias primas para suplir la demanda de la industria local, haciendo menos competitivas las empresas nacionales e impactando directamente las ganancias del índice

Finalmente, se espera una caída significativa del índice por cuenta de la aparición de DeepSeek, la empresa china de inteligencia artificial que llevó por tierra el valor de Nvidia, el segundo activo más importante del índice S&P500, iniciando de manera contundente la etapa de «comoditización»

de la IA con el efecto dominó que esto implica para los proveedores de tecnología, recursos y servicios para el sector.

Referencias

- Abhyankar, A., Copeland, L. S., & Wong, W. (1997). Uncovering nonlinear structure in real-time stock-market indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business & Economic Statistics*, 15(1), 1-14. <https://doi.org/10.1080/07350015.1997.10524681>
- Afza, T., & Tahir S. (2021). Determinants of price-earnings ratio: The case of chemical sector of Pakistan. *International Journal of Academic Research in Business and Social Sciences*, s. d. <http://dx.doi.org/10.6007/IJARBSS/v2-i8/9237>
- Andrade Gorjoux, L. E., González Contreras, J. F., Montiel Pérez, J. Y., & Martínez Castro, J. A. (2023). Comparación entre codificación predictiva lineal (LPC) y red neuronal artificial, long short-term memory (LSTM), para la predicción financiera del índice de precios y cotizaciones (S&P/BMV IPC). *POLIBITS*, 65(1), 19-26. [https://doi.org/10.17562/PB-65\(1\)-3](https://doi.org/10.17562/PB-65(1)-3)
- Arrona Palacios, P. (2024). *Sentimiento de mercado y fundamentales económicos: su impacto en la predicción de los retornos de las acciones de las empresas que cotizan en la bolsa de valores en México* [tesis doctoral, Universidad Autónoma de Nuevo León, San Nicolás de los Garza, Nuevo León, México]. <http://eprints.uanl.mx/id/eprint/28636>
- Bai, J., & Perron, P. (1998). Estimating and testing linear models with multiple structural changes. *Econometrica*, 66(1), 47-78. <https://www.jstor.org/stable/2998540>
- Carlson, J. B., Pelz, E., & Wohar, M. (2002). Will valuation ratios revert to historical means? *The Journal of Portfolio Management*, 28(4), 23-35. DOI 10.26509/frbc-wp-200113
- Carlson, M., Pelz, K., & Wohar, M. E. (2009). The effect of inflation and uncertainty on the relationship between price-earnings ratios and inflation. *Journal of Macroeconomics*, 31(1), 101-112. <https://dergipark.org.tr/tr/download/article-file/363108>
- Correa Restrepo, J. D. (2024). *Desarrollo de un score de riesgo financiero con XGBoost para evaluación de crédito Fintech* [tesis de maestría, Universidad EAFIT, Medellín]. <https://repository.eafit.edu.co/server/api/core/bitstreams/22a1684f-540b-400f-b24f-bffa827189ff/content>
- Cruz Martínez, E. D., Ávila Romero, R., & Chiatchoua, C. (2024). Análisis de la inflación y el crecimiento económico: un enfoque de machine learning. México, 1990-2021. *Revista Universidad y Sociedad*, 16(3), 174-188. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2218-36202024000300174
- Goud, R. G., & Reddy, M. K. (2017). Forecasting of P/E ratio for the Indian equity market stock index NIFTY 50 using artificial neural networks. *International Journal of Economics and Financial Issues*, 7(4), 758-763. <https://www.ijmh.org/wp-content/uploads/papers/v10i5/F1576029623.pdf>
- Jiang, W. (2020). Application of deep learning in stock market valuation: Index forecasting. *International Journal of Advanced Computer Science and Applications*, 11(1), 1-10. <https://doi.org/10.1016/j.eswa.2021.115537>

- Microsoft Corporation. (2025). *LightGBM. Features.*
<https://lightgbm.readthedocs.io/en/stable/Features.html>
- Murillo Ospina, S. (2024). *Inteligencia artificial y autoaprendizaje para la mejora de modelos predictivos: caso de estudio con el índice S&P500* [tesis doctoral, Universidad Nacional de Colombia, Bogotá]. <https://repositorio.unal.edu.co/handle/unal/86576>
- Nagy, M., Valaskova, K., Kovalova, E., & Macura, M. (2024). Drivers of S&P 500's profitability: Implications for investment strategy and risk management. *Economies*, 12(4), 77. <https://doi.org/10.3390/economies12040077>
- Osorio Buitrón, M., & Rico, J. D. (2024). *Optimización de un portafolio de inversión con acciones del Colcap aplicando técnicas de machine learning* [tesis de maestría, Universidad EAFIT, Medellín]. <https://repository.eafit.edu.co/server/api/core/bitstreams/7d5eedd9-0004-45d8-88b4-b031edfd465f/content>
- Pimentel Ramírez, P. I. (2024). *Aplicación de deep learning para predicción de índices bursátiles extranjeros usando modelos multivariados recurrentes y convolucionales con mecanismos de atención* [tesis de maestría, Universidad de Chile, Santiago de Chile]. <https://repositorio.uchile.cl/handle/2250/201498>
- Sirisha, U. M., Belagavi, M. C., & Attigeri, G. (2022). Profit prediction using Arima, Sarima and LSTM models in time series forecasting: A comparison. *IEEE Access*, 10, 124715-124727. DOI 10.1109/ACCESS.2022.3224938
- Smilkov, D., & Carter, S. (s. f.). *Tensor Flow*. <https://playground.tensorflow.org/>
- Thompson, P. A. (1990). An MSE statistic for comparing forecast accuracy across series. *International Journal of Forecasting*, 6(2), 219-227. [https://doi.org/10.1016/0169-2070\(90\)90007-X](https://doi.org/10.1016/0169-2070(90)90007-X)
- XGBoost Tutorials. (2022). *Introduction to boosted trees.*
<https://xgboost.readthedocs.io/en/stable/tutorials/model.html>
- Zorn, D. (2009). P/E changes: Some new results. *Journal of Forecasting*, 28, 358-370. DOI 10.1002/for.1097