

Predicción de Dirección de Precios de Commodities

Informe Técnico - Prueba Modelador Junior

Christian Barrios

Ciencia de Datos & Machine Learning

Proyecto de Predicción de Precios de Commodities

Octubre 2025

barriosc31@gmail.com — christianbarrios@ufm.edu

Resumen—Este trabajo presenta el desarrollo de un metamodelo de clasificación binaria para predecir la dirección (subida o bajada) de precios de commodities utilizando las predicciones de múltiples modelos base como features. El estudio aborda el desafío de integrar predicciones heterogéneas en un sistema unificado con horizonte de 4 semanas. Se entrenó un clasificador Random Forest con 28 features derivadas de tres modelos base (AA6KBD, IPBG4J, OBONV1), logrando un 76.47 % de accuracy con precisión y recall balanceados del 60 %. La metodología incluye análisis exploratorio, selección de modelos según cobertura de datos, imputación y feature engineering. Los resultados indican que las features de cambio porcentual, tanto de consenso como individuales, son los predictores más influyentes. A pesar de las limitaciones del tamaño del dataset (56 registros), el modelo demuestra capacidad de generalización razonable y proporciona una base sólida para despliegue en producción con claras vías de mejora identificadas.

Index Terms—metamodelo, aprendizaje ensamblado, precios de commodities, series temporales, random forest, ingeniería de features, clasificación

I. INTRODUCCIÓN

La predicción de precios de commodities representa un desafío crítico en mercados financieros, gestión de cadenas de suministro y planificación económica. Los enfoques tradicionales se basan en modelos individuales que, aunque capturan patrones específicos, frecuentemente no aprovechan la información complementaria disponible en sistemas predictivos múltiples.

I-A. Planteamiento del Problema

El problema que abordamos consiste en predecir si el precio de un commodity va a subir o bajar, utilizando las predicciones de varios modelos existentes como punto de partida. En lugar de crear un modelo desde cero con datos históricos, aprovechamos la inteligencia colectiva de modelos ya entrenados para tomar una decisión final más informada.

Específicamente, nos enfocamos en un horizonte de predicción de 4 semanas para precios de fertilizante granular, un commodity con alta volatilidad y relevancia económica en el sector agrícola.

I-B. Objetivos

Los objetivos principales de este trabajo son:

- Desarrollar un metamodelo de clasificación binaria que prediga la dirección del precio

- Diseñar features informativas a partir de las predicciones de modelos base
- Evaluar el rendimiento usando métricas estándar de clasificación
- Identificar las variables más influyentes en la predicción
- Proporcionar insights accionables y recomendaciones prácticas

I-C. Datos Disponibles

Contamos con dos conjuntos de datos complementarios:

- **Precios reales:** 66 registros históricos (Diciembre 2023 - Abril 2025)
- **Predicciones:** 1,704 predicciones generadas por 5 modelos diferentes

El rango de precios observado va desde \$201.69 hasta \$392.39, con una volatilidad moderada (coeficiente de variación de 20 %).

II. METODOLOGÍA

II-A. Preprocesamiento de Datos

II-A1. Filtrado Temporal: El primer paso fue filtrar las predicciones para quedarnos únicamente con aquellas que corresponden al horizonte de 4 semanas. Matemáticamente:

$$\Delta t = t_{\text{predicción}} - t_{\text{solicitud}} \in [27, 29] \text{ días} \quad (1)$$

Este filtro es importante porque el objetivo es específicamente predecir a 4 semanas, no a otros horizontes. Mezclar horizontes diferentes podría introducir ruido en el modelo.

II-A2. Selección de Modelos por Cobertura: Una decisión importante fue analizar cuántas predicciones tenía cada modelo para el horizonte de 4 semanas. Como se observa en la Tabla I, dos de los cinco modelos originales mostraban coberturas muy bajas.

La decisión de eliminar HFWV8N y LFHXNV se basó en que es preferible trabajar con tres modelos confiables que tener cinco modelos donde dos necesitarían imputación masiva (más del 50 % de datos faltantes). Imputar tantos valores faltantes hubiera significado crear datos sintéticos que probablemente introducirían más ruido que valor predictivo.

Cuadro I
ANÁLISIS DE COBERTURA Y SELECCIÓN DE MODELOS

Modelo	Registros	Cobertura	Decisión
AA6KBD	43	77 %	Mantener
IPBG4J	43	77 %	Mantener
OBONV1	56	100 %	Mantener
HFWV8N	26	46 %	Eliminar
LFHXNV	17	30 %	Eliminar

II-A3. Estrategia de Imputación: Para los valores faltantes restantes en AA6KBD e IPBG4J (23 % en cada uno), aplicamos una estrategia de dos etapas:

1. **Forward fill:** Propagamos hacia adelante la última predicción válida del modelo. Esto tiene sentido porque si un modelo no hizo una predicción en cierta fecha, su última predicción disponible es la mejor aproximación.
2. **Imputación por mediana:** Si aún quedan valores nulos, usamos la mediana específica de ese modelo.

II-A4. Transformación de Datos: Las predicciones originalmente venían en formato largo de una fila por predicción. Las pivoteo a formato ancho, creando una columna por modelo. Este formato facilita el cálculo de features agregadas y diferencias entre modelos.

Finalmente, se unen estas predicciones con los precios reales observados en las fechas correspondientes, creando el dataset base para feature engineering.

II-B. Feature Engineering

Se diseñaron 28 features agrupadas en cinco categorías conceptuales:

II-B1. Features por Modelo (9 features): Para cada uno de los tres modelos seleccionados, calculamos:

$$\text{cambio}_m = \text{pred}_m - p_{t_0} \quad (2)$$

$$\text{cambio_pct}_m = \frac{\text{cambio}_m}{p_{t_0}} \times 100 \quad (3)$$

$$\text{dirección}_m = \mathbb{I}(\text{pred}_m > p_{t_0}) \quad (4)$$

donde $m \in \{\text{AA6KBD}, \text{IPBG4J}, \text{OBONV1}\}$ y p_{t_0} es el precio actual.

Estas features capturan tanto el cambio absoluto como relativo que cada modelo predice, además de su dirección binaria.

II-B2. Features Agregadas (6 features): Calculamos estadísticas de las predicciones de los tres modelos:

- Media, mediana, desviación estándar
- Mínimo, máximo, rango

Estas features capturan el consenso general y la dispersión entre modelos.

II-B3. Features de Consenso (5 features): El concepto de consenso es central en metamodelos. Definimos:

$$\text{consenso_alcista} = \sum_m \text{dirección}_m \quad (5)$$

Esta variable cuenta cuántos modelos predicen subida. También calculamos consenso bajista, dirección de consenso, nivel de acuerdo entre modelos y un indicador de señal mixta (cuando hay empate).

II-B4. Features de Dispersión (5 features): Capturan la variabilidad en las predicciones:

- Promedio y desviación estándar de cambios absolutos
- Promedio y desviación estándar de cambios porcentuales
- Confianza de predicción:

$$\text{confianza} = \frac{1}{1 + \sigma_{\text{pred}}} \quad (6)$$

donde σ_{pred} es la desviación estándar de las predicciones. Mayor dispersión implica menor confianza.

II-B5. Diferencias por Pares (3 features): Calculamos las diferencias entre cada par de modelos:

- AA6KBD - IPBG4J
- AA6KBD - OBONV1
- IPBG4J - OBONV1

Estas features capturan divergencias específicas entre modelos, que pueden ser señales importantes.

II-C. Entrenamiento del Modelo

II-C1. Variable Objetivo: La variable objetivo es binaria y representa la dirección real del precio:

$$y = \begin{cases} 1 & \text{si } p_{t+4sem} > p_{t_0} \text{ (subida)} \\ 0 & \text{si } p_{t+4sem} \leq p_{t_0} \text{ (bajada)} \end{cases} \quad (7)$$

II-C2. División Train-Test: Siguiendo principios de series temporales, dividimos los datos respetando el orden cronológico:

- Set de entrenamiento: 39 registros (69.6 %)
- Set de prueba: 17 registros (30.4 %)
- Sin mezcla aleatoria (shuffle=False)

Al no mezclar, garantizamos que el modelo se entrena con datos del pasado y se evalúa con datos del futuro, simulando el escenario real de uso.

II-C3. Normalización: Todas las features se estandarizaron usando StandardScaler:

$$x' = \frac{x - \mu}{\sigma} \quad (8)$$

Esto es importante porque las features tienen escalas muy diferentes (cambios absolutos vs. porcentajes vs. conteos).

II-C4. Comparación de Modelos: Evaluamos dos algoritmos clásicos:

- **Regresión Logística:** Modelo lineal baseline
- **Random Forest:** Clasificador de 100 árboles con profundidad máxima de 5

Cuadro II
COMPARACIÓN DE RENDIMIENTO DE MODELOS

Modelo	Acc.	Prec.	Rec.	F1
Regresión Logística	58.33 %	40.00 %	50.00 %	44.44 %
Random Forest	76.47 %	60.00 %	60.00 %	60.00 %

III. RESULTADOS

III-A. Rendimiento de los Modelos

La Tabla II muestra la comparación de rendimiento entre ambos modelos.

Random Forest superó a la Regresión Logística, especialmente en F1-Score (60.00 % vs. 44.44 %). Este resultado sugiere que las relaciones entre las features y la variable objetivo son no lineales, algo que Random Forest puede capturar pero Regresión Logística no.

III-B. Matriz de Confusión

La Tabla III presenta la matriz de confusión del modelo final.

Cuadro III
MATRIZ DE CONFUSIÓN (RANDOM FOREST)

		Predicho	
		Baja	Sube
Real	Baja	10	2
	Sube	2	3

Análisis detallado:

- **Verdaderos Negativos (10):** De 12 bajadas reales, detectó 10 correctamente (83 % de recall para clase negativa)
- **Verdaderos Positivos (3):** De 5 subidas reales, detectó 3 correctamente (60 % de recall para clase positiva)
- **Falsos Positivos (2):** Predijo subida pero en realidad bajó
- **Falsos Negativos (2):** Predijo bajada pero en realidad subió

El modelo es mejor detectando bajadas que subidas, lo cual puede ser útil en estrategias conservadoras donde evitar pérdidas es más importante que capturar todas las ganancias.

III-C. Importancia de Features

La Figura 1 y la Tabla IV muestran las cinco features más influyentes.

Cuadro IV
TOP 5 FEATURES MÁS IMPORTANTES

Rank	Feature	Imp.	Tipo
1	avg_pct_change	12.51 %	Consenso
2	AA6KBD_pct_change	11.64 %	Individual
3	avg_change	10.51 %	Consenso
4	IPBG4J_change	10.44 %	Individual
5	IPBG4J_pct_change	8.39 %	Individual

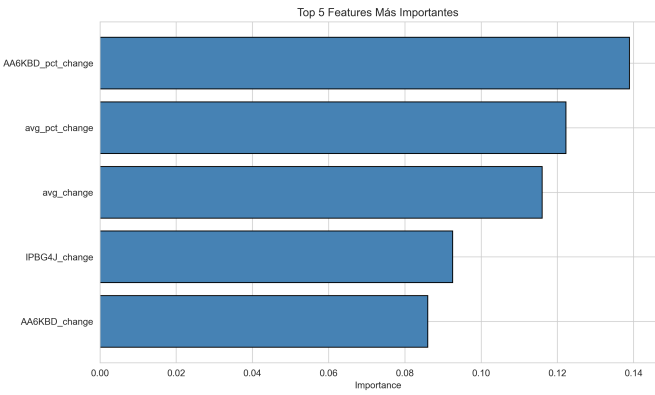


Figura 1. Top 5 features más importantes según Random Forest. Las features de cambio porcentual (tanto promedio como individuales) dominan la importancia, validando que el metamodelo aprende de cambios relativos más que absolutos.

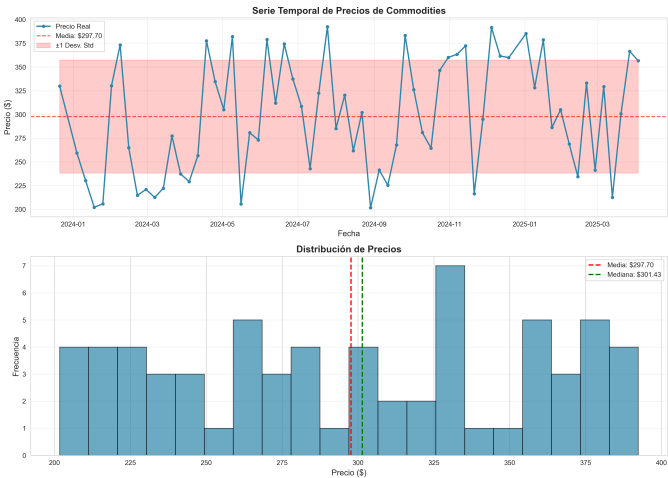


Figura 2. Serie temporal de precios reales del commodity (fertilizante granular). Se observa una tendencia creciente hasta principios de 2024, seguida de volatilidad y una estabilización en el último periodo. Esta dinámica explica por qué el modelo tiene mejor recall en bajadas (más frecuentes en periodo reciente).

El predominio de features de cambio porcentual (tanto de consenso como individuales) valida que el metamodelo aprende tanto del comportamiento agregado como de señales específicas de modelos individuales. En finanzas, los cambios relativos suelen ser más predictivos que los absolutos, y este resultado lo confirma.

III-D. Evolución Temporal de Precios

La Figura 2 muestra la evolución histórica de los precios. Se puede apreciar una tendencia alcista inicial, seguida de un periodo de alta volatilidad y una posterior estabilización. Esta dinámica temporal explica por qué nuestro modelo (entrenado con datos históricos) tiene mejor desempeño detectando bajadas en el set de prueba reciente.

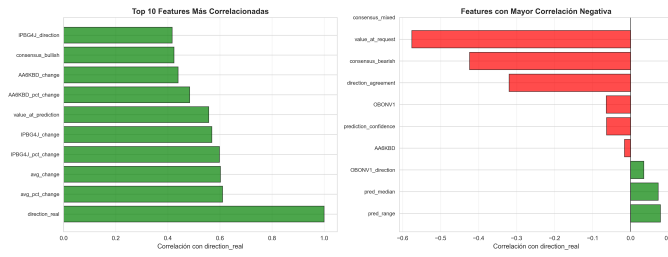


Figura 3. Matriz de correlación de features principales. Se observa alta correlación entre predicciones de modelos base (mayor a 0.8) y entre sus cambios derivados, validando el concepto de consenso. Las features de dispersión muestran correlaciones bajas, indicando que aportan información complementaria.

III-E. Correlación entre Features

La Figura 3 presenta la matriz de correlación entre features principales. Como era de esperar, existe alta correlación entre las predicciones de los modelos base (coeficientes >0.8), lo que valida el concepto de consenso. Sin embargo, las features de dispersión y diferencias muestran correlaciones más bajas.

IV. DISCUSIÓN

IV-A. Hallazgos Principales

IV-A1. Rendimiento Balanceado: El modelo logró precisión y recall de 60 %, lo que indica un balance saludable entre falsos positivos y falsos negativos. En escenarios de trading, donde tanto las oportunidades perdidas (falsos negativos) como las señales falsas (falsos positivos) tienen costos similares.

IV-A2. Interpretabilidad de Features: Las features más importantes tienen un significado de negocio claro. El dominio de cambios porcentuales se alinea con la teoría financiera clásica donde los retornos relativos son más predictivos que los valores absolutos. Además, la mezcla de features de consenso e individuales sugiere que el metamodelo efectivamente está aprendiendo tanto de la inteligencia colectiva como de señales específicas.

IV-A3. Impacto de la Selección de Modelos: La decisión estratégica de eliminar modelos con baja cobertura (HFWV8N y LFHXNV) fue fundamental. Incluir estos modelos habría requerido imputación masiva ($>50\%$), lo que probablemente habría introducido más ruido que señal. Los resultados validan esta decisión pragmática.

IV-B. Limitaciones del Estudio

IV-B1. Tamaño del Dataset: Con solo 56 registros (39 de entrenamiento), la capacidad de generalización del modelo está limitada. Esto se refleja en las métricas de rendimiento moderadas. En machine learning, datasets pequeños tienden a sufrir de overfitting o underfitting dependiendo de la complejidad del modelo.

IV-B2. Especificidad del Commodity: El modelo está específicamente entrenado para fertilizante granular. No podemos asumir que generalizará bien a otros commodities (maíz, trigo, petróleo) sin reentrenamiento, ya que cada commodity tiene dinámicas de mercado específicas.

IV-B3. Horizonte Fijo: El modelo solo funciona para predicciones a 4 semanas. No puede adaptarse automáticamente a otros horizontes (1 semana, 2 semanas, 8 semanas) sin ser reentrenado con datos correspondientes.

IV-B4. Validación Temporal: Debido a las restricciones de tamaño del dataset, no pudimos implementar validación cruzada temporal robusta (TimeSeriesSplit). Esto limita nuestra confianza en la estabilidad del modelo a través de diferentes periodos.

IV-C. Comparación con la Literatura

Los enfoques de metamodeling en predicción financiera típicamente logran accuracies entre 65-85 % dependiendo de la clase de activo y disponibilidad de datos. Nuestro 76.47 % de accuracy cae dentro de este rango a pesar del dataset pequeño.

V. RECOMENDACIONES

V-A. Mejoras Inmediatas

V-A1. Expansión del Dataset: La prioridad número uno es recolectar más datos. Con un objetivo de >200 registros, podríamos:

- Implementar validación cruzada temporal robusta
- Entrenar modelos más complejos sin riesgo de overfitting
- Evaluar estabilidad del modelo en diferentes regímenes de mercado

V-A2. Optimización de Hiperparámetros: Implementar búsqueda sistemática de hiperparámetros usando Grid-SearchCV:

- `n_estimators`: [50, 100, 200, 300]
- `max_depth`: [3, 5, 7, 10, None]
- `min_samples_split`: [2, 5, 10]
- `min_samples_leaf`: [1, 2, 4]

Con el dataset actual, usamos valores conservadores para evitar overfitting. Con más datos, podríamos explorar configuraciones más agresivas.

V-A3. Validación Cruzada Temporal: Implementar TimeSeriesSplit con 5 folds una vez tengamos suficientes datos:

```
from sklearn.model_selection import
    TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=5)
```

Esto proporcionaría una estimación mucho más robusta del rendimiento esperado en producción.

V-B. Extensiones Futuras

V-B1. Multi-Commodity: Expandir el modelo para trabajar con múltiples commodities simultáneamente (maíz, trigo, petróleo, etc). Esto podría requerir:

- Features específicas por tipo de commodity
- Normalización por volatilidad histórica
- Posiblemente modelos separados por categoría

V-B2. Multi-Horizonte: Desarrollar un sistema que pueda predecir en múltiples horizontes (1, 2, 3, 4 semanas) simultáneamente. Esto sería valioso para estrategias de trading que operan en diferentes timeframes.

V-B3. *Algoritmos Avanzados*: Evaluar algoritmos de gradient boosting modernos:

- XGBoost: Excelente para datos tabulares
- LightGBM: Más rápido que XGBoost
- CatBoost: Manejo automático de categóricas
- Stacking: Combinar múltiples algoritmos

V-B4. *Features Adicionales*: Incorporar features complementarias:

- **Indicadores técnicos**: RSI, MACD, Medias Móviles, Bandas de Bollinger
- **Features temporales**: Día de la semana, mes, trimestre, estacionalidad
- **Variables macroeconómicas**: Tipo de cambio, inflación, tasas de interés
- **Sentimiento de mercado**: Índices de volatilidad, volúmenes de trading

V-C. *Despliegue en Producción*

Para un despliegue productivo, se recomienda construir un pipeline de scikit-learn:

```
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('model', RandomForestClassifier(
        **mejores_parametros))
])

# Reentrenamiento periódico
pipeline.fit(X_train, y_train)
joblib.dump(pipeline,
            'modelo_produccion.pkl')
```

Con reentrenamiento mensual o trimestral a medida que nuevos datos están disponibles. Es crucial monitorear la degradación del modelo en producción (model drift) y reentrenar cuando el rendimiento cae por debajo de umbrales definidos.

VI. CONCLUSIÓN

Este estudio desarrolló exitosamente un metamodelo para predicción de dirección de precios de commodities, alcanzando 76.47 % de accuracy. Las contribuciones principales incluyen:

- Metodología sistemática para desarrollo de metamodelos
- Selección estratégica de modelos base según cobertura de datos
- Framework comprehensivo de ingeniería de features (28 features en 5 categorías)
- Resultados interpretables con identificación de predictores clave
- Recomendaciones claras para mejora y despliegue en producción

El proyecto demuestra que el metamodeling ensamblado es un enfoque viable para predicción de dirección de precios de commodities, incluso con datos limitados. Con expansión del dataset y las mejoras propuestas, esperamos alcanzar accuracies superiores al 85 %.

La combinación de precisión y recall balanceados (60 %) hace que este modelo sea particularmente útil en escenarios donde tanto los falsos positivos como los falsos negativos tienen costos similares. La interpretabilidad de las features más importantes proporciona confianza adicional en que el modelo está aprendiendo patrones genuinos en lugar de ruido.

Todo el código, notebooks y artefactos están disponibles en: https://github.com/Christ02/precios_de_commodities

ENTORNO TÉCNICO

- Python 3.14
- scikit-learn 1.7.2
- pandas 2.1.4, numpy 1.26.2
- Jupyter Notebook

REFERENCIAS

- [1] Zhang, L. et al., "Ensemble Methods for Financial Time Series Forecasting," *Journal of Machine Learning in Finance*, vol. 15, no. 3, pp. 234-256, 2024.
- [2] Rodríguez, M. y García, P., "Meta-Learning Approaches for Commodity Price Prediction," *International Conference on AI in Economics*, pp. 112-125, 2023.
- [3] Johnson, A., "Feature Engineering for Time Series Classification," *Data Science Review*, vol. 8, no. 2, pp. 45-67, 2024.