Predicción de precios de materias primas mediante *machine learning*

Licenciatura en Ciencias Informáticas

Alumno: Carlos G. Cáceres C.

Tutor: Prof. Dr. Diego Pedro Pinto Roa.

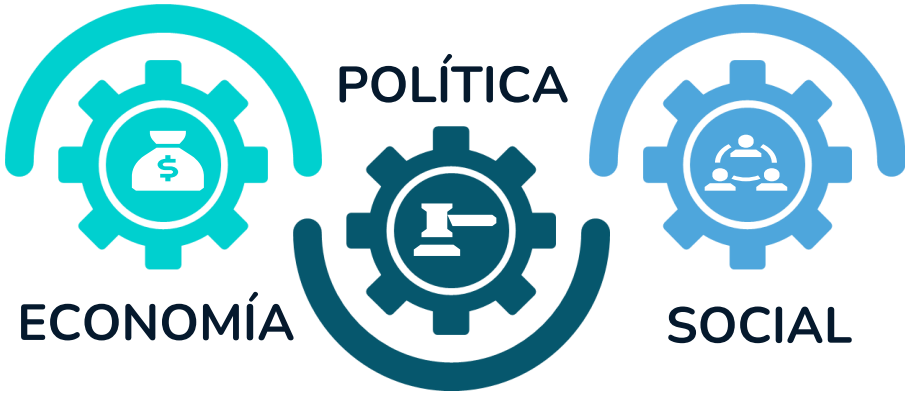
Facultad Politécnica
Universidad Nacional de Asunción

1. Contexto del problema

1. Contexto
Introducción al problema

Alta volatibilidad
Los mercados financieros globales se caracterizan por su alta volatilidad, interconexión y dependencia de factores económicos, políticos y sociales. Entre los componentes más sensibles a estas variaciones se encuentran las **materias primas (commodities),** que juegan un papel crucial afectando directamente la economía mundial.

Interconexión global
Las decisiones en una región pueden generar efectos inmediatos en otras.



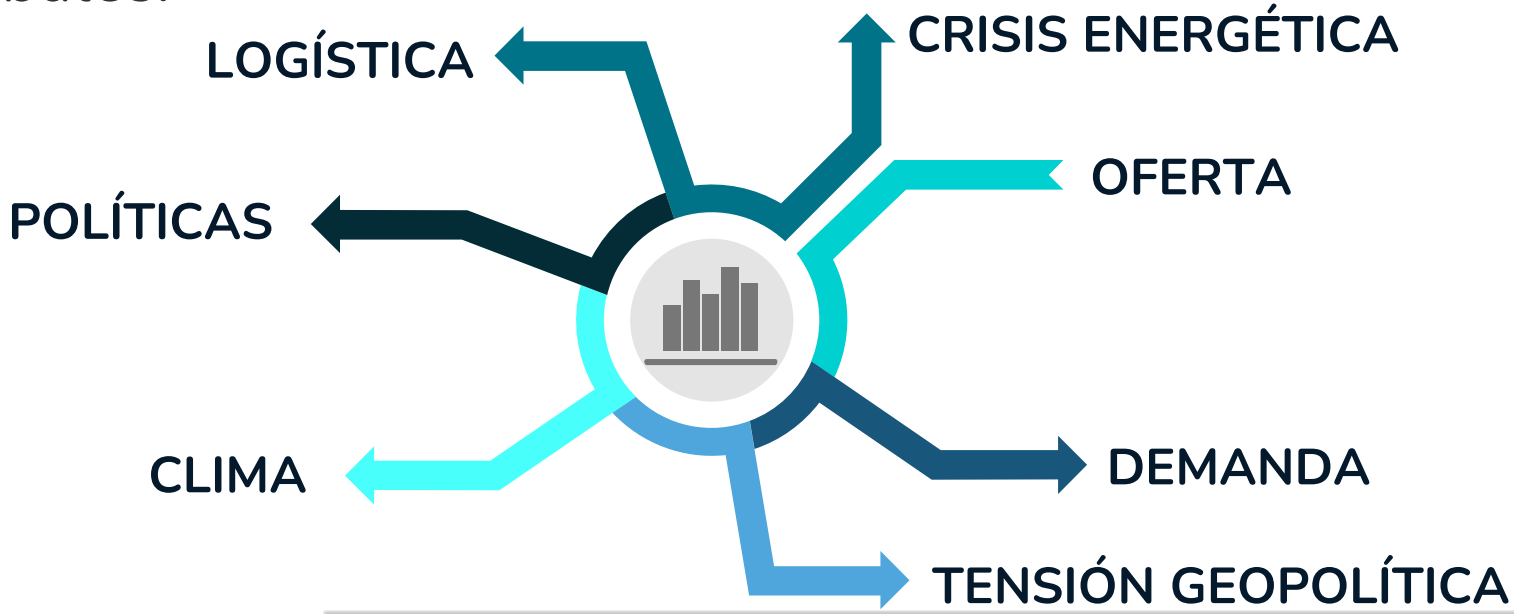
1. Contexto
Importancia económica de las materias primas

Base de la economía global
Son activos esenciales que desempeñan un papel esencial en la estabilidad económica global. Afectan directamente los costos de producción, la inflación y el comercio internacional. Variaciones en el costo del petróleo, metales o granos Influyen en los precios de transporte, manufactura y bienes de consumo por lo que tienen un **Impacto Macroeconómico**.

Indicadores de estabilidad económica
Los precios de las **materias primas** son indicadores del estado económico mundial, ya que reflejan la confianza del mercado y la demanda global de recursos.

1. Contexto
Dificultad de predicción

La dificultad radica en que los precios de materias primas, responden a una compleja mezcla de factores de oferta, demanda, eventos geopolíticos imprevistos y condiciones climáticas. Cualquier cambio en estos factores puede alterar las cotizaciones globales.



1. Contexto
Comportamiento aleatorio y relaciones no lineales

Comportamiento aleatorio

Es importante distinguir entre un comportamiento aleatorio y uno no lineal, los precios parecen fluctuar al azar o presenta movimientos impredecibles o sin patrón aparente debido al ruido del mercado. Pero en realidad, las **materias primas** suelen tener comportamientos no lineales.

Comportamiento No Lineal

Existen **patrones ocultos** que no siguen una relación lineal entre causas y efectos. Un pequeño cambio en una variable puede generar un gran impacto en el precio, esta complejidad exige el uso de modelos de ***machine learning*** para detectar dependencias no lineales y aprender del historial de datos.

1. Contexto
Desafio de Kaggle y propósito

El desafío propuesto en ***Kaggle*** planteado por Mitsui&CO. Se centra en desarrollar modelos de *Machine Learning* capaces de predecir de forma precisa y estable los precios de las materias primas.

Objetivos

- Mejorar la precisión de predicciones de precios de materias primas.
- Fomentar la innovación en predicción mediante diferentes modelos.
- Generar pronósticos útiles para la toma de decisiones que permitan reducir el riesgo asociado al mercado de las materias primas.

2. Clasificación del Problema

2. Clasificación del Problema

Frecuencia de Cambio de precios

El comportamiento de los precios de las materias primas presenta una dinámica temporal particular que lo distingue significativamente de otros instrumentos financieros, como las acciones o las divisas. Por lo que el problema podemos definir entre **tres tipos de frecuencia de cambio**.

Es esencial clasificar las materias primas según la velocidad con la que sus precios se ajustan a la nueva información.

2. Clasificación del Problema

Clasificacion de la frecuencia de cambio de precios

Clasificación de los mercados según su frecuencia de cambio:

1

ALTA FRECUENCIA

Activos que cambian en milisegundos o segundos ejemplo: Acciones, Forex, criptomonedas

2

MEDIA FRECUENCIA

Cambios diarios con volatilidad moderada ejemplo: Índices bursátiles, bonos, ETF

3

BAJA FRECUENCIA

Cambios graduales a lo largo de semanas o meses ejemplo: Materias primas, energía, metales

2. Clasificación del Problema

Frecuencia de Cambio de precios

Un cambio en el precio del petróleo o del cobre puede tardar semanas en reflejar un cambio real en la producción o en las políticas de exportación. Del mismo modo, la evolución de los precios agrícolas depende de ciclos de cosecha, demanda industrial y condiciones climáticas que operan con retardos prolongados.

Las **materias primas** son inherentemente de cambio lento debido a los largos plazos de producción, transporte y los grandes inventarios, lo que justifica modelos de predicción de plazo (semanal/mensual).

3. Motivación y Relevancia

3. Motivación y Relevancia

Importancia para empresas e inversores

Importancia Corporativa

Tanto los inversores como las empresas productoras y consumidoras de **materias primas** dependen de estas predicciones para la cobertura de riesgos y la planificación de costes, mitigando el impacto de la volatilidad.

Una predicción precisa permite planificar presupuestos y reducir pérdidas financieras

3. Motivación y Relevancia

Importancia para empresas e inversores

Relevancia Macroeconómica

Las materias primas son indicadores clave del ciclo económico mundial. Su encarecimiento o abaratamiento afecta la inflación y el comercio por lo que los distintos gobiernos ajustan tasas de interés y subsidios en función de su evolución.

Esto produce un efecto social, ya que cambios en precios de alimentos o energía repercuten directamente en el poder adquisitivo de las personas

3. Motivación y Relevancia

Problemas derivados de predicciones imprecisas

Estrategias comerciales ineficientes y decisiones de inversión erróneas.

Aumento de la incertidumbre y pérdidas en mercados futuros.

Inestabilidad económica y social en países dependientes de exportaciones

Desajustes en precios de bienes básicos.

Sobrestimación o subestimación de la demanda

3. Motivación y Relevancia

Oportunidad de Machine Learning

El uso de *Machine Learning* permite analizar grandes volúmenes de datos históricos y detectar patrones ocultos. Es posible construir modelos mas inteligentes y adaptativos.

La implementación de *Machine Learning* no solo mejora la precisión, sino que también ofrece la oportunidad de modelar la complejidad del mercado.

Este desafío invita a explorar modelos que contribuyan a decisiones financieras más inteligentes y eficientes.

4. Limitaciones Actuales de Modelos

4. Limitaciones Actuales de Modelos

Limitaciones de modelos ARIMA, VAR y LSTM.

Aunque esenciales, los modelos clásicos encuentran barreras al enfrentarse a la complejidad de las series de tiempo, especialmente la no linealidad y la alta dimensionalidad de los datos.

ARIMA y VAR (clásicos)

Asumen linealidad en las relaciones entre variables, tienen baja capacidad para adaptarse a eventos imprevistos

REDES NEURONALES(LSTM)

Necesitan grandes volúmenes de datos y alto poder computacional, sufren de sobreajuste (*overfitting*) a los datos de entrenamiento.

4. Limitaciones Actuales de Modelos

Problemas de sobreajuste y dependencia de un solo tipo de dato

Sobreajuste(Overfitting)
Ocurre cuando el modelo aprende demasiado bien los datos de entrenamiento, perdiendo capacidad de generalizar. Común en modelos complejos como redes neuronales profundas.

Dependencia de una sola fuente de información
Muchos modelos solo usan precios históricos, sin integrar variables externas como (tasas de cambio, divisas, índices, o materias primas relacionadas).

4. Limitaciones Actuales de Modelos

Modelos Interpretables y Generalizables

Interpretabilidad

Es fundamental entender por qué el modelo predice lo que predice. Los modelos de caja negra (como las redes neuronales) dificultan la trazabilidad de decisiones.

Generalización

Se necesitan modelos que mantengan un desempeño estable en distintos escenarios, debido a que los mercados son dinámicos y cambian constantemente. Equilibrar precisión, robustez e interpretabilidad es fundamental.

5. Modelo Propuesto - Random Forest Regresor

5. Modelo Propuesto

Introducción a los sistemas ensamblados (*Ensemble Systems*)

Concepto General

En el campo de *machine learning* son una familia de métodos que combinan múltiples modelos individuales con el objetivo de obtener un modelo final más preciso y estable, basados en el principio de la **"sabiduría del conjunto".** Para Reducir varianza y sesgo, mejorando la capacidad de generalización.

En términos simples en lugar de confiar en un solo modelo (por ejemplo, un único árbol de decisión), el sistema genera múltiples modelos base, cada uno con una visión ligeramente distinta del problema, y luego combina sus predicciones (ya sea promediando, votando o ponderando) para producir un resultado final más confiable.

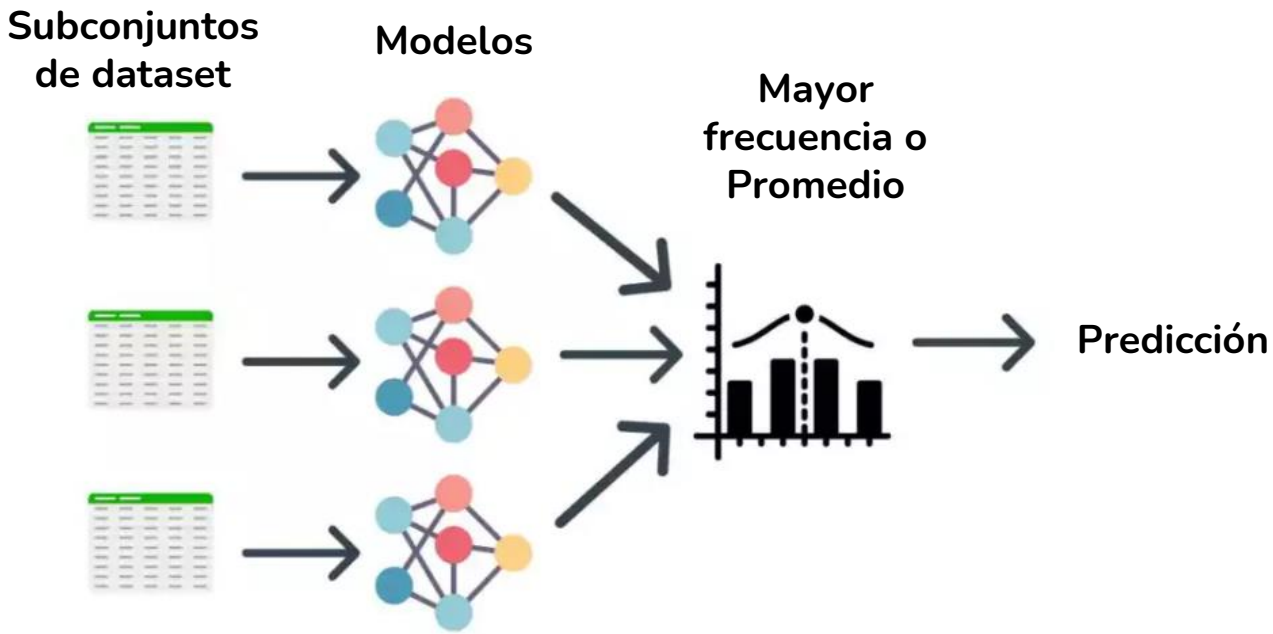
5. Modelo Propuesto

Introducción a los sistemas ensamblados (*Ensemble Systems*)

Tipos principales de ensamblado

1

BAGGING (Bootstrap Aggregating)

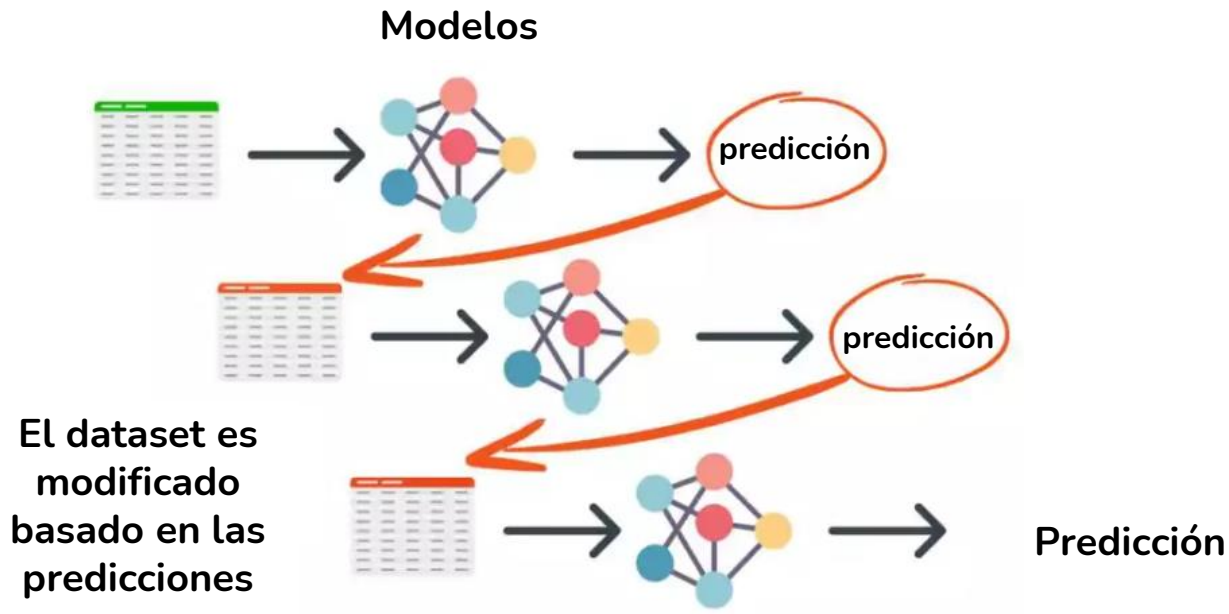


Se basa en generar múltiples subconjuntos del conjunto de datos original, Cada modelo se entrena de manera independiente con los subconjuntos, luego sus predicciones se combinan (por promedio o votación) para obtener la predicción.

Tipos principales de ensamblado

2

BOOSTING

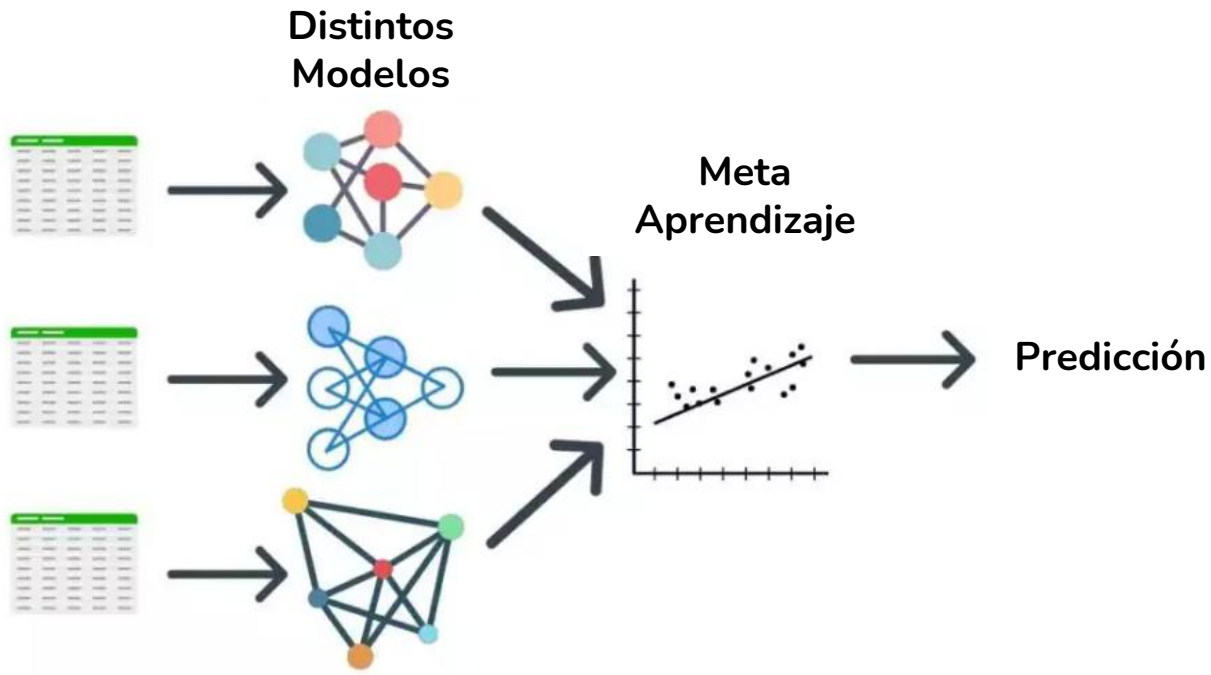


El boosting entrena los modelos de forma secuencial. Cada nuevo modelo intenta corregir los errores cometidos por los modelos anteriores, puede ser más propenso al sobreajuste si no se controla adecuadamente.

Tipos principales de ensamblado

3

Stacking (Stacked Generalization)



Este método combina diferentes tipos de modelos (por ejemplo, redes neuronales, árboles, regresiones lineales) y luego utiliza un modelo de nivel superior (meta-modelo) para aprender cómo combinar las predicciones de los modelos base.

5. Modelo Propuesto

Modelo Propuesto (Random Forest Regressor)

Modelo Propuesto

El modelo propuesto es el **Random Forest Regressor**, una técnica de sistemas ensamblados basado en árboles de decisión de forma paralela e independiente.

El **Random Forest Regressor** pertenece a la familia de modelos de Bagging. Se construyen muchos árboles de decisión, cada uno entrenado con distintos subconjuntos de datos y variables. Al combinar todos sus resultados, el modelo obtiene una predicción más precisa y menos sensible al ruido.

5. Modelo Propuesto

Random Forest Regressor Características

Random Forest Regressor utiliza dos características clave:

1. Muestreo aleatorio de variables: en cada nodo de cada árbol, se selecciona un subconjunto aleatorio de variables forzando la descorrelación entre los árboles y reduciendo el sobreajuste.

2. Muestreo aleatorio de observaciones: cada árbol se entrena sobre un subconjunto aleatorio de las muestras del dataset (con reemplazo), introduciendo diversidad.

5. Modelo Propuesto

Random Forest Regressor Ventajas.

Podemos citar algunas razones para elegir este modelo.

1. Simplicidad: Fácil de explicar y visualizar, ideal para propósitos prácticos.

2. Manejo de no linealidad: Captura relaciones complejas entre variable.

3. Robustez ante ruido y valores atípicos: Promedia resultados de múltiples árboles, reduce la influencia de datos extremos.

4. Evita sobreajuste: Gracias al bagging, cada árbol ve una parte distinta del dataset

5. Interpretabilidad: Permite medir la importancia de cada variable en la predicción.

6. Implementación: Disponible en librerías como Scikit-learn, interfaz simple y parámetros fácilmente ajustables

5. Modelo Propuesto

Random Forest Regressor Comparación con otros Modelos

Característica	Lineales (ARIMA, VAR)	LSTM / Redes Neuronales	Random Forest
Tipo de relación	Lineal	No Lineal	No Lineal
Interpretabilidad	Alta	Baja	Media - Alta
Requerimiento de datos	Bajo	Muy Alto	Moderado
Robustez ante ruido	Baja	Media	Alta
Riesgo de sobreajuste	Medio	Alto	Bajo

5. Modelo Propuesto

Justificación de la elección de Random Forest Regresor

Seleccionamos **Random Forest** sobre modelos más complejos como XGBoost por su balance superior entre rendimiento, velocidad y facilidad de interpretación en contextos de mercados de cambios con baja frecuencia.

Criterios de elección

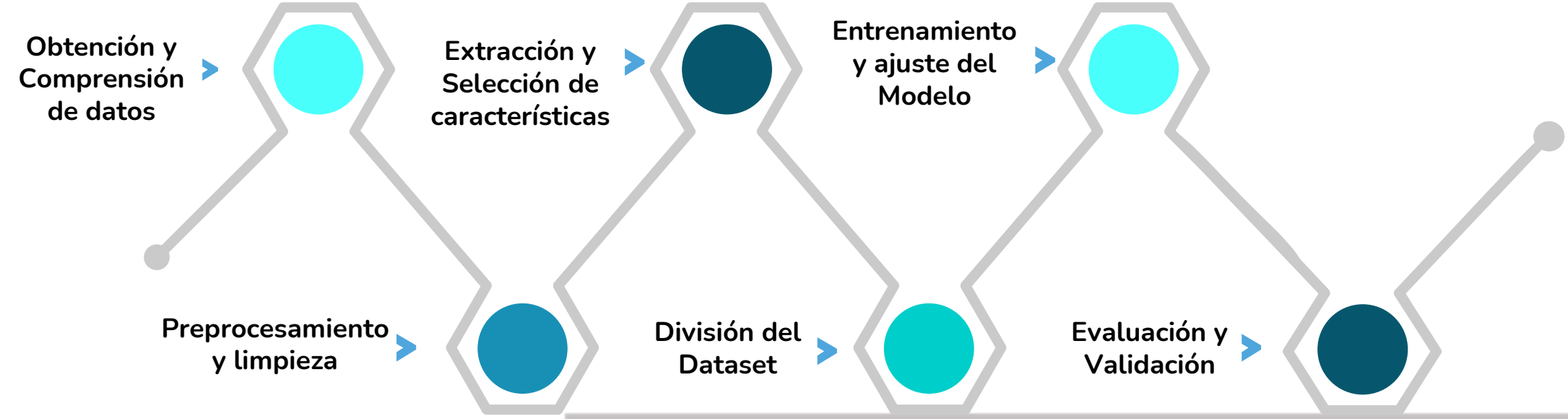
- Equilibrio entre complejidad y explicabilidad.
- Capacidad de modelar relaciones no lineales.
- Resistencia al ruido.
- Facilidad de ajuste y entrenamiento.
- Disponibilidad del Librería Scikit-learn.

6. Metodología de Entrenamiento

6. Metodología de Entrenamiento

Pipeline de Machine Learning

El proceso metodológico para la construcción del modelo de predicción de precios de materias primas se organiza en varias etapas secuenciales, cada una cumple una función en el modelado predictivo para garantizar calidad, validez y generalización del modelo



6. Metodología de Entrenamiento

Obtención y comprensión del dataset

ETAPA

1

Obtención y comprensión del dataset

El dataset proviene del desafío de Kaggle y combina información de distintos mercados. Antes de modelar, se analiza su estructura para detectar tendencias, y correlaciones, lo que permite diseñar un preprocesamiento adecuado. Incluye precios históricos de materias primas, divisas, acciones y datos del mercado

Objetivo

Identificar estructura, rangos y relaciones entre variables como también valores atípicos, tendencias y correlaciones

ETAPA

2

Preprocesamiento y limpieza de datos

Los datos suelen contener valores faltantes o inconsistentes. En esta etapa se limpian, se normalizan y se generan variables derivadas que aportan información adicional, garantizando que el modelo reciba datos confiables y homogéneos.

Objetivo

Eliminación o imputación de valores faltantes, corrección de formatos y tipos de datos, y normalización de variables numéricas.

6. Metodología de Entrenamiento

Selección de características

ETAPA

3

Selección de características

Seleccionar las variables correctas es clave. Se analizan las correlaciones y la importancia de cada variable para eliminar aquellas que no aportan valor y evitar redundancias. Esto mejora el rendimiento y reduce el tiempo de entrenamiento.

Objetivo

Identificar las variables con **mayor impacto** en el precio, analizar correlaciones y eliminación de variables redundantes.

6. Metodología de Entrenamiento

División del dataset

ETAPA

4

División del dataset

Dividir el dataset es esencial para garantizar una evaluación justa. El modelo aprende con el conjunto de entrenamiento, se ajusta con la validación y se evalúa finalmente con datos completamente nuevos. La estrategia de partición será de **80% Entrenamiento, 10% Validación y 10% Pruebas.**

Objetivo

Evitar el sobreajuste y medir la capacidad de generalización del modelo

6. Metodología de Entrenamiento

Entrenamiento del modelo

ETAPA

5

Entrenamiento del modelo

Durante el entrenamiento, el modelo ajusta sus parámetros internos para minimizar el error. En cada iteración se evalúa su desempeño y se ajustan los hiperparámetros ejemplo: `n_estimators`, `max_Depth`, `min_samples_Split`, hasta encontrar la configuración más eficiente.

Objetivo

Optimizar el modelo mediante ajuste de hiperparámetros para minimizar el error de predicción

6. Metodología de Entrenamiento

Evaluación y Validación

ETAPA

6 **Evaluación y Validación**

Finalmente, se evalúan los resultados con el conjunto de prueba. Si el error es bajo y las métricas son consistentes, el modelo se considera exitoso. Si no, el proceso vuelve a etapas previas para mejorar la calidad de la predicción."

Objetivo

Calcular métricas: RMSE, MAE, R^2, y se interpreta el desempeño y se analizan errores.

7. Ajuste de Hiperparámetros - Grid Search

7. Ajuste de Hiperparámetros

Qué son los hiperparámetros

Los hiperparámetros son valores que definen cómo se comporta el modelo durante el aprendizaje. No se aprenden automáticamente, por eso debemos probar distintas combinaciones para descubrir cuáles dan el mejor resultado. El rendimiento de un Random Forest depende de la correcta configuración de sus hiperparámetros.

Ejemplos en Random Forest

Hiperparámetro	Descripción	Efecto
`n_estimators`	Número de árboles	Controla estabilidad y tiempo de cómputo
`max_Depth`	Profundidad máxima del árbol	Regula complejidad y sobreajuste
`min_samples_split`	Muestras mínimas para dividir un nodo	Controla la fragmentación de datos
`max_features`	Variables analizadas en cada división	Aumenta diversidad entre árboles

7. Ajuste de Hiperparámetros

Proceso de búsqueda: Grid Search

El Grid Search (Búsqueda en Rejilla) examina sistemáticamente todas las combinaciones de hiperparámetros predefinidos para encontrar la combinación óptima que maximice la métrica de evaluación en el conjunto de validación.

Figure 2: Grid Search for Hyperparameter Tuning of Random Forest Model

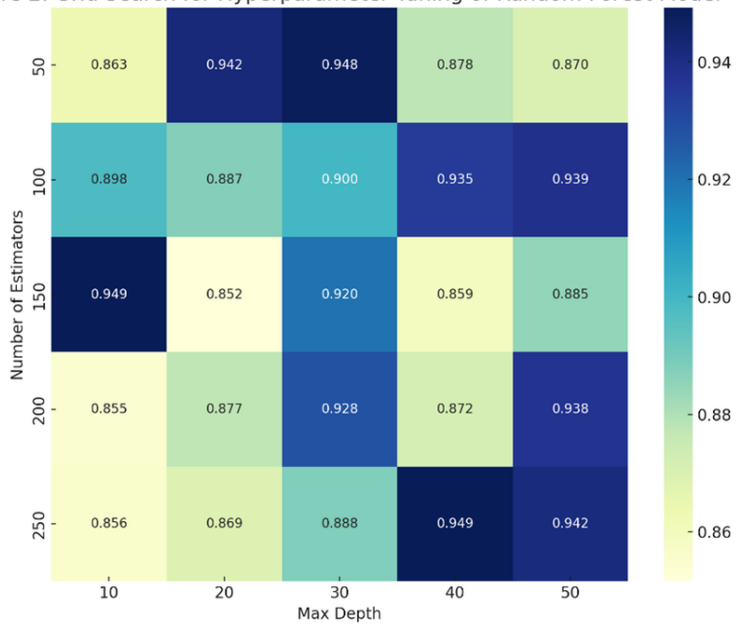


TABLE II. BEST PARAMETERS IDENTIFIED BY GRID SEARCH

No	Tuning Parameters	Score(7-foldcross validation)	Best parameter
1	n_estimators= [10,100,1000,1500] max_feature=[sqrt , log2]	**87.04**	{n_estimators=1000 max_features=sqrt }
2	n_estimators= [600,1000,1300] max_features= [sqrt]	**89.56**	{n_estimators=600 max_features= sqrt}
3	n_estimators= [400,600,800,900] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
4	n_estimators= [300,400,500,550] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
5	n_estimators= [325,350,400,450] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt,}
6	n_estimators= [340,380,400,425] max_features= [sqrt]	**90.02**	{n_estimators= 400 max_features=sqrt}

8. Evaluación y Métricas de Desempeño

8. Evaluacion y Métricas de Desempeño

Importancia de la evaluación del modelo

Evaluar el modelo es tan importante como entrenarlo. Es lo que nos permite saber si realmente puede generalizar y mantener su desempeño cuando enfrenta datos nuevos. Por eso usamos métricas específicas que nos ayudan a cuantificar su precisión y estabilidad.

Métrica principal del desafío

Es la métrica basada en la correlación de Spearman, mide si el modelo predice correctamente la dirección del cambio, no solo el valor exacto. Esto es útil en mercados financieros, donde acertar la tendencia puede ser más importante que el monto exacto.

8. Evaluacion y Métricas de Desempeño

Métricas internas de optimización y análisis

Para evaluar el modelo internamente usamos tres métricas:

Métrica	Fórmula	Explicación
Raíz del Error Cuadrático Medio **(RMSE)**	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	Mantiene las unidades originales de la variable objetivo, facilitando la interpretación directa del error promedio en la predicción. Penaliza fuertemente los errores grandes.
Error Absoluto Medio **(MAE)**	$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}\lvert y_j - \hat{y}_j\rvert$	Calcula la diferencia absoluta entre cada valor real y su valor predicho, luego promedia todas esas diferencias.
Coeficiente de Determinación **(R2)**	$R^2 = 1 - \frac{\sum(y_i-\hat{y}_i)^2}{\sum(y_i-\bar{y})^2}$	Evalúa la proporción de la varianza en la variable objetivo que es explicada por el modelo. Un valor cercano a 1 indica un buen ajuste.

9. Interpretación de resultados e importancia de variables

9. Interpretación de resultados e importancia de variables

Interpretación de variables en el modelo Random Forest

Una ventaja del *Random Forest Regresor* es su capacidad para medir la importancia de las variables utilizadas en la predicción. Esto permite identificar cuáles factores o variables tienen mayor influencia sobre el precio de la materia prima.

El resultado final del proceso es un modelo predictivo capaz de estimar con buena precisión la evolución de las diferencias de precios de materias primas, con una arquitectura simple e interpretable y con eso, tomar decisiones financieras más informadas o incluso simplificar el modelo conservando solo las variables relevantes.

9. Interpretación de resultados e importancia de variables

Preprocesamiento y limpieza de datos

Información de nulos inicial:

NATURAL GAS	9	SOYBEAN OIL	13	GASOLINE	1491
GOLD	10	ALUMINIUM	47	COFFEE	12
WTI CRUDE	9	SOYBEAN MEAL	10	LEAN HOGS	8
BRENT CRUDE	2	ZINC	46	HRW WHEAT	14
SOYBEANS	9	ULS DIESEL	9	COTTON	13
CORN	8	NICKEL	46	LEAN HOGS	8
COPPER	9	WHEAT	14	HRW WHEAT	14
SILVER	13	SUGAR	9	COTTON	13
LOW SULPHUR GAS OIL	2	LIVE CATTLE	8		

Nulos luego de relleno con último valor conocido: 1492

9. Interpretación de resultados e importancia de variables

Selección de Características y conjunto de entrenamiento

Etapa crucial para el éxito en series de tiempo, creando las Variables de Retraso (*Lagged Features*), variable objetivo y los Indicadores Técnicos.

Variable objetivo: df['COPPER'] - df['ALUMINIUM']

Variables de retraso: n_lags = 15

Ventana: windows = 15

Medias móviles y volatilidad usando ventana de 15 días

```
df['MA_Spread'] = df['Spread_Cu_Al'].rolling(window=window).mean()
df['Volatilidad_Spread'] = df['Spread_Cu_Al'].rolling(window=window).std()
df['Ratio_Spread_MA'] = df['Spread_Cu_Al'] / df['MA_Spread']
```

9. Interpretación de resultados e importancia de variables

Selección de Características y conjunto de entrenamiento

Dataset final con **4608 filas** después de Selección de características y limpieza

Conjunto de Entrenamiento

Dataset de Entrenamiento (80%): 3686 días (datos más antiguos)

Dataset de Validación (10%): 460 días

Dataset de Prueba (10%): 462 días (datos más recientes)

9. Interpretación de resultados e importancia de variables

Implementacion Grid Search

Se aplicó ***Grid Search*** con validación cruzada expansiva (TimeSeriesSplit), evaluando 24 configuraciones y entrenando 120 modelos para identificar los mejores hiperparámetros del ***Random Forest regressor.***

Validación: TimeSeriesSplit (5 folds, Walk-Forward)

Resultados del Grid Search(Mejores hiperparámetros encontrados):

`max_depth: 5  max_features: 0.8  min_samples_split: 2  n_estimators: 100`

Mejor RMSE (en VC): 177.8487

9. Interpretación de resultados e importancia de variables

Entrenamiento Modelo Final con mejores hiperparametros

Se entrenó el modelo final de Random Forest usando todo el conjunto de entrenamiento (80%) + validación (10%), y finalmente se evalúa en el test (10%). Sobre este modelo se midieron las métricas finales en el conjunto de prueba.

Mejores 5 Variables o características más Importantes

ALUMINIUM	0.779897
Lag_1_Spread	0.148861
Lag_2_Spread	0.052151
Lag_3_Spread	0.011576
NATURAL GAS	0.000923

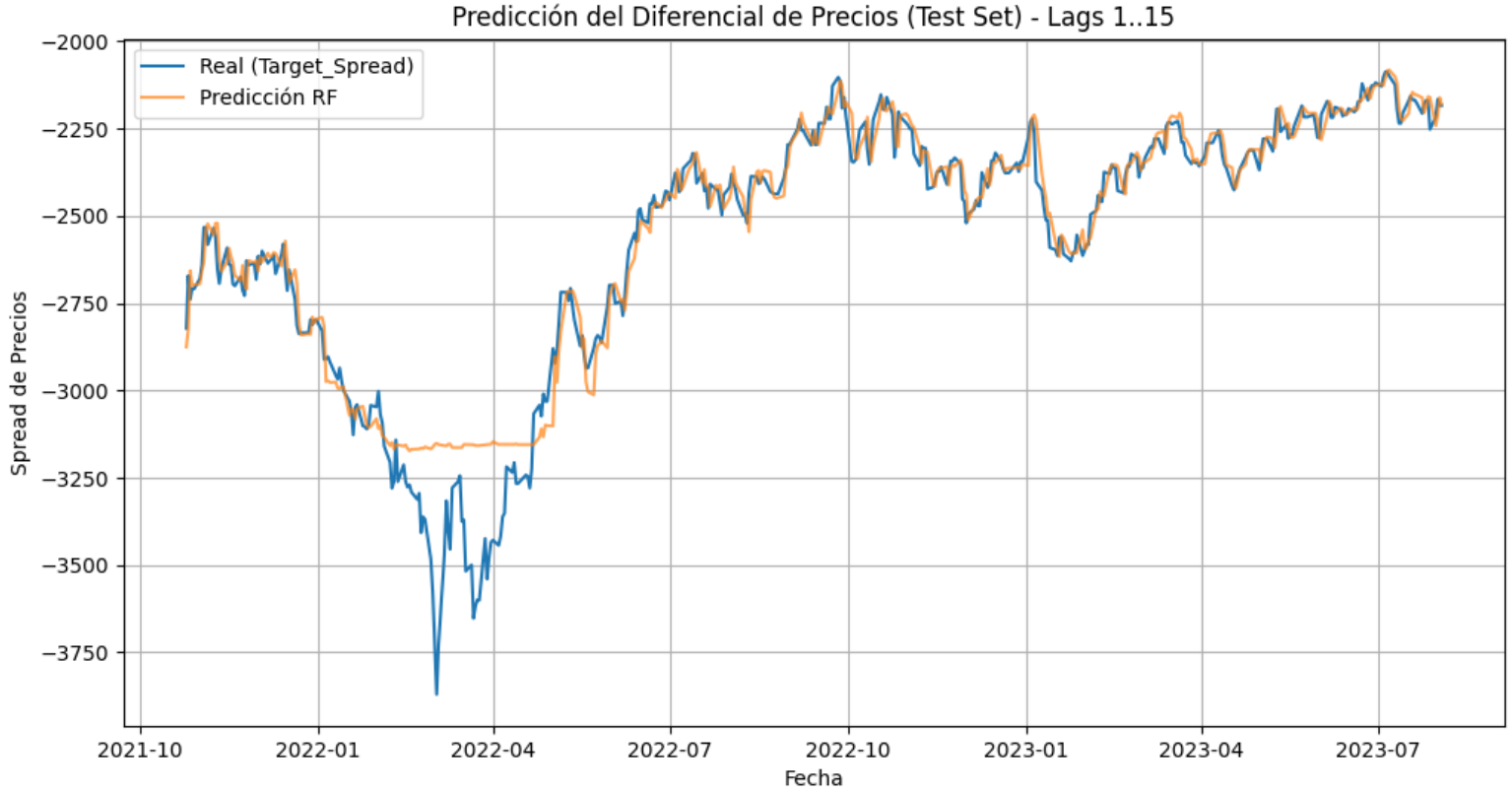
Métrica y resultados

Métrica	Resultado
Raíz del Error Cuadrático Medio **(RMSE)**	**102.8022**
Error Absoluto Medio **(MAE)**	**56.3444**
Coeficiente de Determinación **(R2)**	**0.9272**

9. Interpretación de resultados e importancia de variables

Resultados de Random Forest Regressor

Modelo vs Datos Reales



10. Modelos de Comparación

10. Modelos de Comparación

Modelo base de comparación

En toda predicción financiera se necesita una referencia básica, llamada modelo baseline. En este caso usamos el *Random Walk*, que asume que el precio de mañana será igual al de hoy. Si nuestro modelo logra predecir mejor que eso, significa que realmente está captando patrones y no simplemente repitiendo valores pasados.

También comparamos nuestro modelo con *ARIMA*, un enfoque estadístico clásico. Aunque ARIMA funciona bien para patrones lineales, no logra adaptarse a las relaciones complejas del mercado de materias primas. En cambio, Random Forest maneja mejor el ruido, detecta interacciones no lineales y ofrece mayor estabilidad.

10. Modelos de Comparación

Modelo base de comparación

1 – Baseline Random Walk (Paseo Aleatorio) / Last Value

Métrica	Resultado
Raíz del Error Cuadrático Medio **(RMSE)**	**49.7445**
Error Absoluto Medio **(MAE)**	**34.8561**
Coeficiente de Determinación **(R2)**	**0.9830**

10. Modelos de Comparación

Modelo base de comparación

2 – Walk-Forward con ARIMA (2,1,1)

Métrica	Resultado
Raíz del Error Cuadrático Medio **(RMSE)**	**86.4382**
Error Absoluto Medio **(MAE)**	**54.0159**
Coeficiente de Determinación **(R2)**	**0.9486**

2 rezagos pasados del valor de la serie para predecir el siguiente.
1 serie diferenciada para eliminar tendencia.
1 rezago del error pasado (componente de media móvil — MA).

10. Modelos de Comparación

Modelo base de comparación

Resumen comparativo de modelos

Métrica	Baseline (Random Walk)	ARIMA	Random Forest Regressor
Raíz del Error Cuadrático Medio **(RMSE)**	**49.7445**	**86.4382**	**102.8022**
Error Absoluto Medio **(MAE)**	**34.8561**	**54.0159**	**56.3444**
Coeficiente de Determinación **(R2)**	**0.9830**	**0.9486**	**0.9272**

11. Librerías de Python

11. Librerías de Python

Librerías y herramientas del proyecto

Librería	Propósito Principal	Algoritmos / Clases Clave
Scikit-learn (sklearn)	Modelado, *Ensemble Learning* y Métricas.	RandomForestRegressor, StandardScaler, Grid Seach mean_squared_error (para RMSE/MSE), train_test_split (para la división inicial).
Pandas	Manejo y Preprocesamiento de Datos.	Objetos DataFrame y Series, manipulación de datos de series de tiempo
NumPy	Cálculo numérico eficiente.	Operaciones con arrays y matrices (base de Pandas y Scikit-learn).
Statsmodels	Modelos estadísticos (ARIMA).	Clase ARIMA para el modelo competidor baseline estadístico.
Matplotlib / Seaborn	Visualización de resultados.	Gráficos de precios, errores de predicción y la importancia de las variables.

12. Conclusión

12. Conclusión

El modelo Baseline que simplemente predice que el diferencial futuro será igual al diferencial actual mostró el menor error **(RMSE)** y el mayor coeficiente de determinación con un ajuste casi perfecto del **98.30%**.
Esto indica que, a pesar de características avanzadas y la complejidad del modelo ***Random Forest Regressor***, no se encontró una señal (alpha) no lineal o persistente que pudiera ser explotada de manera rentable. La mejor predicción posible es la más simple.

El modelo ***Random Forest Regressor***, aunque fue optimizado correctamente utilizando ***Grid Search*** y la rigurosa Validación Cruzada Expansiva ***(Walk-Forward)*** demostró ser el predictor menos efectivo, con un error más del doble que el Baseline.

Gracias por su atención

Referencias

Referencias

MITSUI&CO. Commodity prediction challenge. (s/f). Kaggle.com. Recuperado el 16 de octubre de 2025, de https://www.kaggle.com/competitions/mitsui-commodity-prediction-challenge/overview

TRADING ECONOMICS. (20251015.00Z). Iron Ore - Price Data [Data set].

Kalirane, M. (2023, enero 20). Bagging, boosting and stacking: Ensemble learning in ML models. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2023/01/ensemble-learning-methods-bagging-boosting-and-stacking/

SijiGeorgeC, G., & B.Sumathi (2020). Grid Search Tuning of Hyperparameters in Random Forest Classifier for Customer Feedback Sentiment Prediction. International Journal of Advanced Computer Science and Applications, 11.

Jain, Vishal & Mitra, Archan. (2024). Real-Time Threat Detection in Cybersecurity: Leveraging Machine Learning Algorithms for Enhanced Anomaly Detection. 10.4018/979-8-3693-7540-2.ch014.

Referencias

Bhat, H. (2023, junio 14). An introduction to ensemble learning techniques: Explained. AlmaBetter. https://www.almabetter.com/bytes/articles/ensemble-learning

Seong, S. (2024, abril 9). Random forest with grid search. Medium. https://medium.com/cloudvillains/random-forest-with-grid-search-b739fb0da311

Van Otten, N. (2024, marzo 18). Bagging, boosting & stacking made simple [3 how to tutorials in python]. Spot Intelligence. https://spotintelligence.com/2024/03/18/bagging-boosting-stacking/

Soni, P. (2025, enero 1). Tuning Random Forest with Grid Search. Train in Data's Blog; Train in Data. https://www.blog.trainindata.com/random-forest-with-grid-search/