PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA



TRABAJO PREVIO PARA LA OBTENCIÓN DEL TÍTULO DE MAGISTER EN SISTEMAS DE INFORMACIÓN MENCIÓN DATA SCIENCE

TEMA:

Análisis del comportamiento histórico y clasificación mediante Machine Learning de la variación mensual del precio del tomate riñón en mercados mayoristas del Ecuador utilizando datos públicos del SIPA.

AUTOR:

William Dario Herrera Bravo

TUTOR:

Gustavo Xavier Chafla Altamirano, PhD.

QUITO, AGOSTO 2026

DEDICATORIA

Dedico este proyecto primeramente a Dios, quien me ha dado vida, salud y sabiduría para culminar una etapa más en mi formación profesional.

A mi madre, por estar siempre a mi lado, por apoyarme en cada uno de mis sueños y por ser una guía fundamental en mi vida.

A mis abuelos, por motivarme constantemente a seguir adelante y por inspirarme a crecer en mi vida personal y profesional.

A mi hermano, por brindarme su apoyo incondicional, por acompañarme en este proceso y por ser parte importante de mi vida.

A todos ustedes, les dedico este logro con mucho cariño y gratitud, porque de una u otra manera han sido parte fundamental de este camino.

AGRADECIMIENTO

Agradezco primeramente a Dios, por acompañarme durante este proceso y permitirme culminar una etapa importante en mi vida personal y profesional. Su guía ha sido fundamental durante este proceso y me ha permitido avanzar con constancia hasta alcanzar esta meta.

A mi madre, por su amor, paciencia y apoyo incondicional. Gracias por estar siempre a mi lado, por creer en mí y por acompañarme en cada uno de mis sueños. Su esfuerzo y confianza han sido una motivación constante durante este proceso.

A mis abuelos, por sus consejos, cariño y palabras de aliento. Gracias por motivarme a seguir adelante y por ser parte importante de mi formación como persona y profesional.

A mi hermano, por brindarme su apoyo, compañía y confianza en cada etapa de mi vida. Gracias por estar presente y por formar parte de este logro.

A mi tutor de tesis, por su orientación, tiempo y acompañamiento durante el desarrollo de este trabajo. Sus observaciones y recomendaciones fueron importantes para mejorar y culminar esta investigación.

Finalmente, agradezco a todas las personas que, de una u otra manera, me apoyaron durante este camino. Este logro representa el esfuerzo, la constancia y el respaldo de quienes han estado presentes en mi vida.

TABLA DE CONTENIDO

ÍNDICE DE TABLAS 5
ÍNDICE DE FIGURAS 6
1. CAPÍTULO I: INTRODUCCIÓN 7
1.1. PLANTEAMIENTO DEL PROBLEMA 9
1.1.1. PREGUNTAS DE INVESTIGACIÓN 10
1.2. OBJETIVOS 11
1.2.1. OBJETIVO GENERAL 11
1.2.2. OBJETIVOS ESPECÍFICOS 11
1.3. JUSTIFICACIÓN 11
2. CAPÍTULO II: MARCO TEÓRICO 13
2.1. Machine Learning 13
2.1.1. Aprendizaje Supervisado 14
2.1.2. Clasificación multiclase 16
2.2. Modelos de clasificación utilizados 16
2.2.1. Dummy Classifier 16
2.2.2. Regresión logística 17
2.2.3. Random Forest 17
2.2.4. Gradient Boosting 18
2.3. Comportamiento y variación de los precios agrícolas 18
2.4. Análisis de series temporales 19
2.5. Preprocesamiento de datos 19
2.6. Sistemas de información agrícola en el Ecuador SIPA 20
2.7. Evaluación de modelos de clasificación 21
2.8. Metodología CRISP-DM 23
3. CAPÍTULO III: METODOLOGÍA 24
3.1. Enfoque de la investigación 24
3.2. Fuente y recolección de datos 25
3.3. Aplicación de la metodología CRISP-DM 26
3.4. Desarrollo metodológico de la investigación 26
3.4.1. Comprensión y exploración de los datos. 26

3.4.2. Preparación y transformación de los datos 27
3.4.3. Construcción de la variable objetivo 29
3.4.4. Integración de datasets complementarios 31
3.4.5. Definición de escenarios de modelado 33
3.4.6. Selección de variables predictoras 35
3.4.7. División temporal y preprocesamiento para el modelado 36
3.4.8. Entrenamiento, optimización y evaluación de modelos 38
3.5. Consideraciones éticas 39
4. CAPÍTULO IV: RESULTADOS Y ANÁLISIS 41
4.1. Resultados del análisis exploratorio 41
4.2. Resultados de la construcción de la variable objetivo 47
4.3. Comparación de escenarios de modelado 49
4.4. Comparación de modelos de clasificación 52
4.5. Evaluación del modelo final 54
4.6. Importancia de variables 56
4.7. Análisis de errores del modelo 60
5. CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES 64
5.1. Conclusiones 64
5.2. Recomendaciones 65
6. REFERENCIAS 67
7. ANEXOS 71
7.1. ANEXO 1. Código principal utilizado para el procesamiento y modelado 71

ÍNDICE DE TABLAS

Tabla 1: Técnicas de Aprendizaje Supervisado 15
Tabla 2: Variables consideradas en la preparación de los datos 29
Tabla 3: Categorías de clasificación de la variación mensual del precio 31
Tabla 4: Datasets complementarios considerados en la investigación 32
Tabla 5: Escenarios de modelado definidos en la investigación 34
Tabla 6: Variables predictoras utilizadas en el escenario base 36

Tabla 7: División temporal de los datos para el escenario base 37
Tabla 8: Resumen del procesamiento de la base principal de mercados mayoristas 42
Tabla 9: Estadísticas descriptivas del precio mensual consolidado del tomate riñón 43
Tabla 10: Precio mensual consolidado según tipo de producción..... 44
Tabla 11: Distribución de la variable objetivo 48
Tabla 12: Comparación general de escenarios de modelado..... 50
Tabla 13: Comparación inicial de modelos de clasificación en el escenario base 52
Tabla 14: Comparación del desempeño del modelo final frente a la línea base 54
Tabla 15: Reporte de clasificación del modelo Random Forest optimizado..... 55
Tabla 16: Principales grupos de variables relevantes en el modelo final..... 59
Tabla 17: Resumen general de aciertos y errores del modelo final..... 60
Tabla 18: Porcentaje de acierto del modelo final por categoría 61
Tabla 19: Porcentaje de acierto del modelo final por mes 62

ÍNDICE DE FIGURAS

Figura 1: Representación gráfica de un modelo de Machine Learning..... 13
Figura 2: Representación gráfica del algoritmo Random Forest..... 17
Figura 3: Metodología CRISP-DM..... 23
Figura 4: Evolución mensual del precio del tomate riñón 44
Figura 5: Evolución mensual del precio por tipo de producción 45
Figura 6: Precio mediano del tomate riñón por mes 46
Figura 7: Distribución de la variación mensual del precio del tomate riñón 49
Figura 8: Matriz de confusión del modelo Random Forest optimizado..... 56
Figura 9: Importancia de variables del modelo Random Forest optimizado 57

1. CAPÍTULO I: INTRODUCCIÓN

La variación de los precios agrícolas constituye un fenómeno de interés económico y comercial debido a sus efectos sobre la planificación, la distribución y la comercialización de los productos alimentarios. En los mercados agrícolas, los precios pueden experimentar alzas, bajas o períodos de estabilidad como consecuencia de factores temporales, productivos, territoriales, comerciales y logísticos. Estas variaciones crean incertidumbre para quienes participan en la venta de estos productos y dificultan predecir cómo se comportarán los mercados en el futuro.

En Ecuador, el tomate riñón es un producto hortícola de consumo frecuente que se vende en diversos mercados mayoristas. El valor del tomate puede variar según el período y el lugar de comercialización, debido a las características propias de cada mercado, la disponibilidad del producto, su naturaleza perecedera y otros factores asociados a la dinámica agrícola. De este modo, el estudio histórico de los precios facilita descubrir tendencias, fluctuaciones y diferencias geográficas relevantes para entender los cambios que experimenta el producto.

El Sistema de Información Pública Agropecuaria (SIPA) contiene datos públicos sobre los precios de productos agrícolas de mercados mayoristas del Ecuador. Esta información representa una fuente relevante para la aplicación de métodos estadísticos y el desarrollo de investigaciones científicas. Sin embargo, su empleo implica procesos de integración, limpieza, transformación y organización que permitan construir un conjunto de datos consistente y adecuado para el análisis.

Como antecedente en el país se encuentra el estudio desarrollado por Peñafiel et al. (2025), investigadores del Instituto Superior Tecnológico Alfonso Herrera, quienes aplicaron técnicas de aprendizaje automático para la predicción de precios agrícolas en la provincia del Carchi, enfocándose en productos como el maíz, la papa, el trigo y el frijol. En su investigación se emplearon algoritmos como regresión lineal, árboles de decisión y redes neuronales. Para el desarrollo del estudio, se utilizó un dataset conformado por registros históricos de precios y producción agrícola correspondientes a la provincia del Carchi.

Otro antecedente relevante es el estudio desarrollado por Asencio Beltrán (2022), investigadora de la Universidad de Guayaquil, quien creó un prototipo basado en algoritmos de aprendizaje

automático supervisado para predecir los precios de los insumos más utilizados por los agricultores de la costa ecuatoriana. La investigación aplicó modelos de series de tiempo mediante redes neuronales artificiales MLP y LSTM para pronosticar el comportamiento de los precios de fertilizantes y agroquímicos. Para ello, se utilizó un dataset proporcionado por el SIPA, correspondiente al período de enero de 2013 a abril de 2022.

De manera complementaria, organismos internacionales como la Organización de las Naciones Unidas para la Alimentación y la Agricultura promueven el uso de tecnologías digitales, inteligencia artificial y análisis de datos para fortalecer la eficiencia, sostenibilidad y capacidad de respuesta de los sistemas agroalimentarios (FAO, 2022). Estas iniciativas evidencian la creciente importancia de aplicar herramientas tecnológicas al estudio de fenómenos agrícolas.

Aunque las investigaciones anteriores demuestran la utilidad del Machine Learning para estimar precios agrícolas, la mayor parte de estos trabajos se ha concentrado en la predicción de valores numéricos continuos, en productos distintos al tomate riñón o en ámbitos geográficos específicos. En consecuencia, existe la oportunidad de estudiar el comportamiento del precio desde una perspectiva diferente, orientada a identificar la dirección de su variación mensual mediante categorías interpretables.

En este contexto, esta investigación busca analizar el comportamiento histórico del precio del tomate riñón en los mercados mayoristas del Ecuador y aplicar modelos de clasificación basados en Machine Learning para identificar si su variación mensual corresponde a una baja, estabilidad o alza. Para ello, se utilizan datos públicos del SIPA. Este planteamiento se aborda como un problema de clasificación multiclase, debido a que la variable objetivo corresponde a una categoría y no a un valor numérico continuo.

Este enfoque se diferencia de la predicción de un valor monetario exacto, debido a que busca determinar la dirección del cambio mensual del precio. Para ello, se calcula la variación entre períodos consecutivos y se aplican criterios previamente establecidos para asignar cada observación a una de las tres categorías de análisis.

En la investigación se realizó un análisis exploratorio del comportamiento histórico del precio, la integración y limpieza de los datos, la construcción de una variable objetivo, el entrenamiento de distintos modelos de clasificación y la optimización de sus hiperparámetros. Los modelos

fueron evaluados empleando métricas apropiadas para problemas multiclase, como accuracy, F1 macro, F1 weighted y matriz de confusión.

Como resultado de los modelos entrenados, Random Forest optimizado mostró el mejor desempeño comparativo con un accuracy de 0,5012, un F1 macro de 0,4805 y un F1 weighted de 0,4958. No obstante, los resultados evidenciaron una capacidad moderada para clasificar la variación mensual del precio, lo que refleja la complejidad del fenómeno analizado y las limitaciones de las variables disponibles.

En este sentido, este estudio proporciona una metodología reproducible para analizar datos públicos de precios agrícolas y evaluar la aplicabilidad de modelos de clasificación en la identificación de movimientos mensuales del precio del tomate riñón en los mercados mayoristas del Ecuador.

1.1. PLANTEAMIENTO DEL PROBLEMA

El precio del tomate riñón en los mercados mayoristas del Ecuador presenta variaciones frecuentes a lo largo del tiempo. En determinados períodos puede aumentar, disminuir o mantenerse relativamente estable, debido a factores relacionados con la disponibilidad del producto, las condiciones de comercialización, la estacionalidad, la ubicación geográfica y las características particulares de cada mercado. Estas fluctuaciones dificultan comprender el comportamiento del precio y generan incertidumbre para los actores que participan en la comercialización del producto.

Aunque el Sistema de Información Pública Agropecuaria dispone de registros históricos sobre los precios del tomate riñón en distintos mercados mayoristas del país, esta información requiere procesos de integración, depuración y transformación antes de ser utilizada para el análisis. Además, la revisión directa de los registros o el cálculo de promedios no siempre permite identificar de manera clara los patrones asociados con las variaciones mensuales del precio.

La mayoría de los estudios relacionados con precios agrícolas se ha orientado a predecir valores numéricos mediante modelos de regresión o series temporales. Sin embargo, la estimación de un precio exacto puede resultar compleja debido a la cantidad de factores que intervienen en su comportamiento y a que no todos se encuentran disponibles dentro de los datos históricos del SIPA. Es por ello que resulta pertinente analizar el problema desde una perspectiva de

clasificación, orientada a identificar si la variación mensual del precio disminuye, se mantiene estable o aumenta.

Las técnicas de Machine Learning permiten analizar relaciones no lineales entre las variables y reconocer patrones que pueden no ser evidentes mediante métodos descriptivos tradicionales. No obstante, su desempeño depende de la calidad de los datos, la construcción de la variable objetivo, la distribución de las categorías y la selección de las características utilizadas durante el entrenamiento.

En este contexto, surge la necesidad de analizar el comportamiento histórico del precio del tomate riñón y desarrollar modelos de clasificación que permitan identificar su variación mensual en los mercados mayoristas del Ecuador. La investigación busca determinar en qué medida los datos públicos del SIPA contienen información suficiente para diferenciar las categorías baja, estabilidad y alza, así como evaluar cuál de los algoritmos implementados presenta el mejor desempeño para este problema.

1.1.1. PREGUNTAS DE INVESTIGACIÓN

Pregunta de Investigación Principal

¿En qué medida los modelos de Machine Learning permiten clasificar la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza en los mercados mayoristas del Ecuador, utilizando datos públicos del SIPA?

Preguntas de Investigación Secundarias

- ¿Cuáles son los principales patrones históricos, temporales y geográficos asociados con la variación mensual del precio del tomate riñón en los mercados mayoristas del Ecuador?
- ¿Qué algoritmo de Machine Learning presenta el mejor desempeño para clasificar la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza?
- ¿Qué variables presentan mayor contribución en el modelo de clasificación seleccionado y cuáles son sus principales limitaciones?

1.2. OBJETIVOS

1.2.1. OBJETIVO GENERAL

Analizar el comportamiento histórico y clasificar mediante algoritmos de Machine Learning la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza en los mercados mayoristas del Ecuador, utilizando datos públicos del SIPA.

1.2.2. OBJETIVOS ESPECÍFICOS

- Analizar el comportamiento histórico e identificar los patrones temporales, estacionales y geográficos asociados con la variación mensual del precio del tomate riñón en los mercados mayoristas del Ecuador, utilizando los registros públicos del SIPA.
- Desarrollar y comparar modelos de clasificación mediante algoritmos de Machine Learning para identificar la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza.
- Evaluar e interpretar el modelo de clasificación con mejor desempeño mediante métricas adecuadas, identificando las variables con mayor contribución y sus principales limitaciones.

1.3. JUSTIFICACIÓN

Esta investigación se justifica por la importancia de analizar con mayor detalle cómo ha variado históricamente el precio del tomate riñón en los mercados mayoristas del Ecuador. Las variaciones mensuales de este producto pueden generar incertidumbre en los procesos de comercialización, ya que el precio no siempre sigue una tendencia fija. En algunos períodos disminuye, en otros se mantiene relativamente estable y en otros presenta aumentos.

Desde una perspectiva práctica, la clasificación de la variación mensual del precio en las categorías baja, estabilidad y alza permite representar su comportamiento de una manera comprensible. A diferencia de la estimación de un valor monetario exacto, este enfoque busca identificar la dirección del cambio del precio y proporcionar información que pueda utilizarse como un insumo complementario para el análisis del mercado.

En el ámbito tecnológico, la investigación permite aprovechar los datos públicos disponibles en el Sistema de Información Pública Agropecuaria mediante técnicas de integración, limpieza, transformación, análisis exploratorio y Machine Learning. De esta manera, se demuestra cómo

la información histórica del sector agrícola puede ser procesada para construir modelos de clasificación y analizar patrones que no siempre son evidentes mediante una revisión directa de los registros.

Desde el punto de vista académico, el estudio aporta evidencia sobre la aplicación de algoritmos de Machine Learning en un problema de clasificación multiclase relacionado con los precios agrícolas. Asimismo, permite comparar el desempeño de diferentes modelos, identificar las variables que presentan mayor contribución en la clasificación y reconocer las limitaciones derivadas de utilizar únicamente la información disponible en las bases de datos del SIPA.

La investigación también resulta relevante porque propone una metodología reproducible que puede servir como referencia para futuros estudios relacionados con el comportamiento de precios de otros productos agrícolas. Además, los resultados obtenidos permiten evidenciar que la clasificación de las variaciones de precios constituye un problema complejo, influenciado por factores que no siempre se encuentran incluidos en los datos históricos.

Finalmente, el desarrollo del estudio es viable debido a la disponibilidad pública de los registros del SIPA, al acceso a herramientas de código abierto como Python y sus librerías especializadas, y a los recursos técnicos necesarios para realizar el procesamiento, análisis y modelado de los datos.

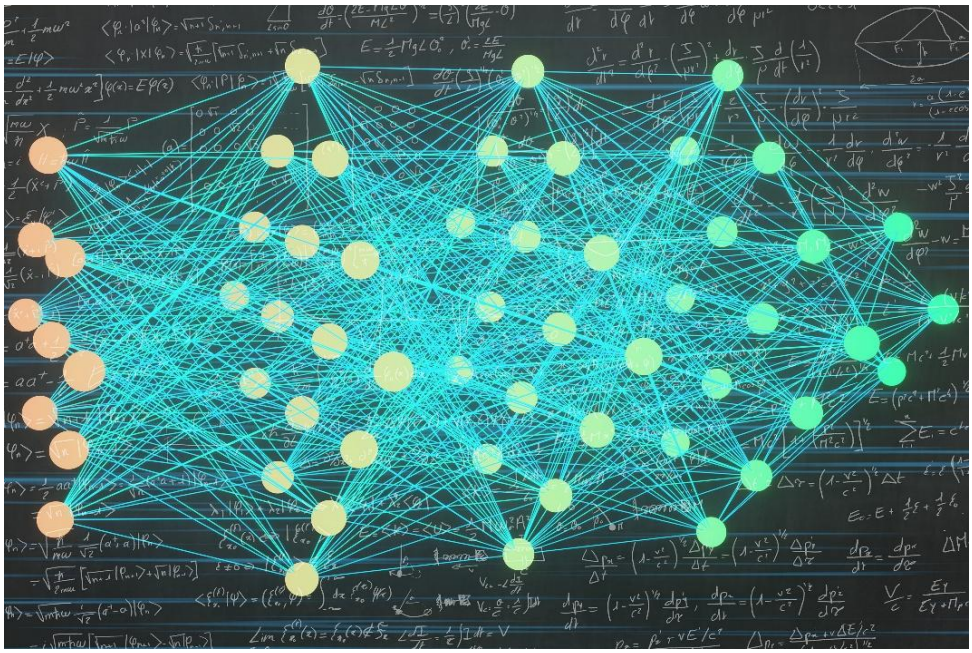
2. CAPÍTULO II: MARCO TEÓRICO

2.1. Machine Learning

El Machine Learning (ML), traducido al español como aprendizaje automático, es una rama de la Inteligencia Artificial (IA), que permite que las computadoras aprendan patrones a partir de datos de entrenamiento y luego generen predicciones a partir de aquello. ML lleva a cabo tareas de análisis de datos sin recibir instrucciones detalladas, esta tecnología sirve para procesar grandes cantidades de datos históricos, identificar patrones y predecir nuevas relaciones entre datos previamente desconocidos (AWS, 2026).

Figura 1:

Representación gráfica de un modelo de Machine Learning



Nota. Tomado de *New algorithms enable efficient machine learning with symmetric data*, por A. Zewe, 2025, MIT News (https://news.mit.edu/2025/new-algorithms-enable-efficient-machine-learning-with-symmetric-data-0730). Crédito de la imagen: iStock y MIT News.

En ML los algoritmos se clasifican de acuerdo con la forma en que aprenden a partir de los datos, lo que determina su aplicación en distintos tipos de problemas. En este contexto, se distinguen generalmente tres tipos de aprendizaje:

- Aprendizaje Supervisado.

- Aprendizaje no Supervisado.
- Aprendizaje por Refuerzo.

2.1.1. Aprendizaje Supervisado

El Aprendizaje Supervisado es un enfoque de ML que utiliza conjuntos de datos etiquetados para entrenar modelos de IA que ayudan a identificar patrones y relaciones entre variables. El objetivo del proceso de aprendizaje es crear un modelo que pueda predecir resultados correctos sobre datos no observados anteriormente (Belcic & Stryker, 2025).

En este tipo de aprendizaje, cada observación del conjunto de entrenamiento se encuentra asociada con una etiqueta o resultado esperado. El modelo utiliza estos ejemplos para ajustar sus parámetros internos y posteriormente es evaluado con datos que no participaron directamente en el entrenamiento. Este procedimiento permite determinar su capacidad para generalizar los patrones aprendidos a observaciones diferentes (James et al., 2021).

Las técnicas de aprendizaje supervisado se dividen principalmente en clasificación y regresión, cuyas características se detallan en la tabla 1:

Tabla 1:

Técnicas de Aprendizaje Supervisado

Técnica	**Definición**	**Algoritmos**
Clasificación	La clasificación es una técnica del aprendizaje supervisado que tiene como finalidad asignar cada observación a una categoría previamente definida, a partir del análisis de sus características y patrones. Se utiliza cuando la variable objetivo es categórica, es decir predicen una clase, el ¿qué es?	Entre los algoritmos utilizados en esta técnica encontramos: • Regresión logística • Árboles de decisión • Random Forest • K-Nearest Neighbors (KNN) • Support Vector Machine (SVM) • Redes neuronales
Regresión	La regresión es una técnica del aprendizaje supervisado utilizada para analizar y modelar la relación entre una variable dependiente y una o más variables independientes, con el objetivo de estimar y predecir valores continuos a partir de datos históricos. Se utiliza cuando la variable objetivo es numérica continua, es decir, predicen una cantidad, el ¿cuánto es?	Entre los algoritmos utilizados en esta técnica encontramos: • Regresión lineal • Regresión polinómica • Random Forest Regressor • Redes neuronales (MLP, LSTM) • SVR (Support Vector Regression)

Fuente: Autor

2.1.2. Clasificación multiclase

La clasificación es una técnica de aprendizaje supervisado que se utiliza cuando la variable objetivo está formada por categorías previamente definidas. Durante el proceso de entrenamiento, el algoritmo analiza las características de las observaciones y aprende patrones que posteriormente emplea para asignar una clase a nuevos registros (James et al., 2021).

Los problemas de clasificación pueden ser binarios o multiclase. La clasificación binaria considera únicamente dos posibles resultados, mientras que la clasificación multiclase trabaja con más de dos categorías. En este último caso, cada observación debe asignarse a una sola de las clases disponibles (Scikit-Learn Developers, 2026).

En este estudio, la variación mensual del precio se clasifica en tres categorías: baja, estabilidad y alza. Como cada registro pertenece únicamente a una de ellas, el problema se aborda como una clasificación multiclase.

2.2. Modelos de clasificación utilizados

Para abordar el problema de clasificación multiclase planteado en la investigación, se utilizaron cuatro modelos con diferentes niveles de complejidad: Dummy Classifier, regresión logística, Random Forest y Gradient Boosting. La selección de estos algoritmos permitió contar con un modelo de referencia, incluir un modelo lineal de fácil interpretación y comparar dos métodos basados en árboles capaces de identificar relaciones no lineales entre las variables.

2.2.1. Dummy Classifier

Dummy Classifier es un modelo de referencia utilizado para establecer un nivel mínimo de desempeño en un problema de clasificación. Este modelo no busca aprender relaciones entre las variables predictoras, sino que genera sus resultados mediante una regla sencilla basada en la distribución de las clases. Una de las estrategias más utilizadas consiste en asignar a todas las observaciones la categoría mayoritaria del conjunto de entrenamiento (Metz et al., 2015).

La utilización de un modelo de referencia permite determinar si los algoritmos de mayor complejidad realmente consiguen identificar patrones útiles en los datos. Según Kuhn y Johnson (2013), si un modelo entrenado no obtiene mejores resultados que una estrategia básica, su desempeño debe interpretarse con cuidado, ya que posiblemente no haya aprendido patrones útiles ni represente una mejora real frente al modelo de referencia.

2.2.2. Regresión logística

La regresión logística es un modelo que se utiliza para resolver problemas de clasificación. Su funcionamiento se basa en estimar la probabilidad de que una observación pertenezca a una categoría determinada, considerando los valores de sus variables predictoras. La función logística transforma el resultado del modelo en una probabilidad comprendida entre cero y uno (James et al., 2021).

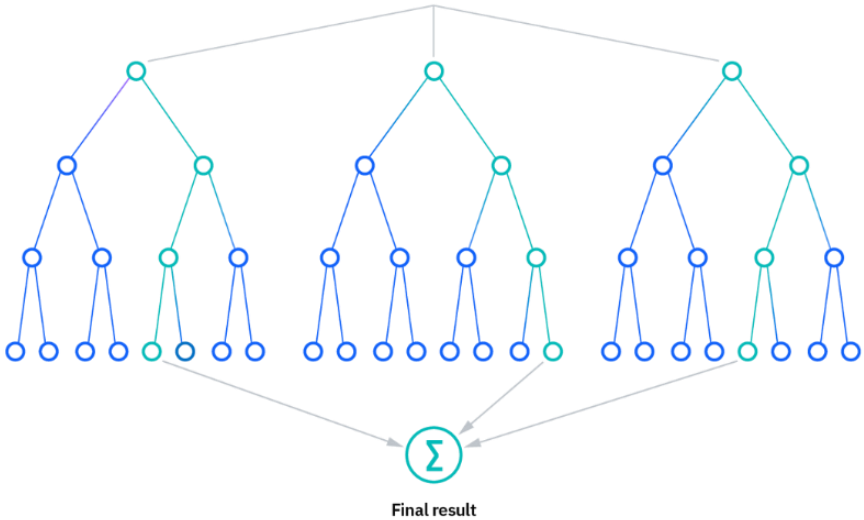
Cuando la variable objetivo contiene más de dos categorías, puede utilizarse la regresión logística multinomial. El modelo calcula qué tan probable es que cada registro pertenezca a cada una de las clases y lo asigna a la clase con la probabilidad más alta (Hastie et al., 2009).

2.2.3. Random Forest

Random Forest es un algoritmo de aprendizaje supervisado basado en la combinación de múltiples árboles de decisión. En lugar de utilizar un solo árbol para realizar la clasificación, el modelo construye varios árboles y reúne sus resultados para obtener una decisión final. En los problemas de clasificación, cada árbol genera una predicción y la categoría que recibe la mayor cantidad de votos se asigna como resultado final (Breiman, 2001).

Figura 2:

Representación gráfica del algoritmo Random Forest



Nota. Tomado de *What is random forest?*, por E. Kavlakoglu, 2021, IBM (https://www.ibm.com/think/topics/random-forest). Crédito de la imagen: IBM.

La combinación de múltiples árboles permite que Random Forest pueda identificar relaciones no lineales e interacciones entre las variables. Además, puede trabajar con conjuntos de datos que contienen muchas características y mostrar cuáles de ellas aportan más al resultado del modelo (Genuer et al., 2010).

2.2.4. Gradient Boosting

Gradient Boosting es un método de aprendizaje supervisado que combina varios árboles de decisión construidos de manera secuencial. En Random Forest, los árboles trabajan de manera independiente; en cambio, en Gradient Boosting, cada árbol nuevo se construye para corregir los errores dejados por los anteriores (Friedman, 2001).

Durante el entrenamiento, el modelo parte de una predicción inicial y va agregando nuevos árboles de forma progresiva. Cada uno se concentra en los casos donde las predicciones anteriores tuvieron más errores o resultaron más difíciles de clasificar. De esta manera, las predicciones se van ajustando progresivamente hasta obtener un resultado final más preciso (Hastie et al., 2009).

Una de las ventajas de Gradient Boosting es que puede identificar relaciones que no siguen un comportamiento lineal y obtiene buenos resultados cuando se trabaja con datos organizados en tablas. Sin embargo, su funcionamiento depende de una adecuada configuración de hiperparámetros, como el número de árboles, la profundidad y la tasa de aprendizaje. Si el modelo se configura con demasiada complejidad, puede provocar sobreajuste y reducir su capacidad para clasificar nuevos registros (Friedman, 2001).

2.3. Comportamiento y variación de los precios agrícolas

Los precios agrícolas pueden presentar aumentos, disminuciones o períodos de estabilidad a lo largo del tiempo. Estos cambios pueden relacionarse con variaciones en la oferta y la demanda, las condiciones de producción, el transporte, el almacenamiento y otros acontecimientos que afectan el funcionamiento de los mercados. Por esta razón, la variación de los precios es una característica frecuente dentro de la comercialización de productos agrícolas (Gilbert & Morgan, 2010).

La volatilidad de precios se refiere al nivel de variación que presenta el valor de un producto entre distintos períodos. Cuando la volatilidad es alta, el precio suele presentar variaciones más

intensas o repetidas. En cambio, una volatilidad baja refleja un comportamiento más estable. Analizarla permite identificar qué tan fuertes son los cambios y en qué períodos se producen las variaciones más relevantes (Gilbert & Morgan, 2010).

Los precios agrícolas también pueden mostrar comportamientos estacionales, es decir, variaciones que tienden a repetirse durante determinados meses o épocas del año. Estos patrones pueden estar relacionados con los ciclos de producción y cosecha, debido a que la cantidad disponible de un producto puede cambiar según el período (Gilbert et al., 2017).

En esta investigación, el comportamiento histórico del precio del tomate riñón se analiza a partir de registros mensuales de mercados mayoristas del Ecuador. Esta información permite observar cambios entre períodos consecutivos y clasificar la variación mensual del precio en las categorías baja, estabilidad o alza, de acuerdo con los criterios definidos en la metodología.

2.4. Análisis de series temporales

Una serie de tiempo es un conjunto de observaciones registradas de manera ordenada en función del tiempo. Estos datos pueden recopilarse en intervalos regulares, como días, meses o años, con el objetivo de analizarlos y posteriormente identificar patrones, comportamientos recurrentes y variaciones a lo largo del tiempo.

El análisis de series temporales facilita la comprensión de tendencias, efectos estacionales y fluctuaciones irregulares, lo que resulta fundamental para interpretar el comportamiento histórico de una variable y realizar proyecciones futuras. Según Wilson (2016), una serie de tiempo está formada por observaciones registradas en períodos sucesivos, cuyo análisis permite describir su estructura mediante modelos matemáticos y realizar pronósticos sobre su comportamiento.

En este sentido, una serie de tiempo no se limita a ordenar datos por fecha; también permite estudiar cómo ha cambiado una variable, reconocer patrones en su comportamiento y utilizar esa información para orientar análisis posteriores.

2.5. Preprocesamiento de datos

El preprocesamiento de datos consiste en transformar y acondicionar los datos para que tengan la forma y calidad adecuadas para entrenar modelos de Machine Learning. Este proceso incluye

estrategias como la limpieza de datos, la partición en conjuntos de entrenamiento y prueba, la normalización o estandarización, el manejo de sesgos y la generación de datos sintéticos, entre otras técnicas que mejoran la utilidad del conjunto de datos para el Machine Learning (Amazon Web Services, 2025).

La limpieza de datos permite identificar y corregir problemas como registros duplicados, valores faltantes, errores de escritura, formatos diferentes e información inconsistente. Estos problemas pueden alterar los resultados obtenidos y reducir la capacidad del modelo para realizar clasificaciones adecuadas, por lo que es necesario corregirlos antes del entrenamiento (García et al., 2015).

Cuando los datos proceden de diferentes archivos o períodos, deben reunirse en una sola base. Para lograrlo, es necesario unificar los nombres de las variables, sus formatos y las unidades utilizadas. Posteriormente, pueden aplicarse transformaciones para adaptar los datos a las necesidades del análisis, como la conversión de fechas, la creación de variables temporales y la codificación de categorías en valores que puedan ser procesados por los algoritmos (García et al., 2015).

También es necesario dividir la información en un conjunto de datos de entrenamiento, utilizado para que el modelo aprenda los patrones presentes en la información, y un conjunto de datos de prueba, empleado para evaluar su desempeño mediante observaciones que no participaron en el proceso de entrenamiento (James et al., 2021).

2.6. Sistemas de información agrícola en el Ecuador SIPA

Los sistemas de información agrícola desempeñan un papel fundamental en la recolección, estructuración y divulgación de información vinculada a la producción, los precios y la comercialización de bienes agrícolas (Aker, 2011). En el Ecuador, el Ministerio de Agricultura y Ganadería dispone de un sistema oficial llamado SIPA, en donde se registra información del sector agroproductivo.

El SIPA del Ecuador constituye una plataforma integrada de información estadística y geográfica orientada a apoyar la toma de decisiones en el sector agropecuario. Este sistema proporciona datos relevantes que permiten tanto a los actores del sector productivo como a investigadores analizar el desempeño del ámbito agropecuario, generar conocimiento científico

y comprender la situación del país en esta área estratégica (Ministerio de Agricultura y Ganadería, s. f.).

La creación de este sistema responde a la necesidad de disponer de información confiable y actualizada sobre el sector agropecuario. En este sentido, la Ley Orgánica de Tierras Rurales y Territorios Ancestrales (LOTRTA), en su artículo 41, establece la conformación del Sistema de Información Pública Agropecuaria con el propósito de recopilar, administrar y poner a disposición información oportuna para los productores y los agentes económicos que intervienen en la producción y comercialización agropecuaria, así como en las actividades vinculadas al uso de la tierra rural (Ley Orgánica de Tierras Rurales y Territorios Ancestrales, 2016).

El funcionamiento del SIPA se basa en la recopilación de información agroproductiva directamente en territorio, mediante el levantamiento de boletas de información. Posteriormente, estos datos son sometidos a procesos de análisis y validación, para finalmente ser difundidos a través de los distintos canales institucionales.

El sistema recopila una amplia variedad de información, entre la cual se incluyen:

- **Indicadores sectoriales:** información de precios en mercados mayoristas, bodegas, camales y ferias; datos de la agroindustria; precios internacionales; e información sobre agroquímicos.
- **Información de comercio exterior agropecuario y agroindustrial:** cifras agroproductivas; informes de rendimiento; estimaciones de superficie cultivada y zonificaciones productivas.
- **Geoportal del Agro Ecuatoriano**: incluye metadatos, servicios geoespaciales y mapas interactivos.

2.7. Evaluación de modelos de clasificación

La evaluación de un modelo de clasificación permite conocer su capacidad para asignar correctamente cada observación a la categoría que le corresponde. Para este propósito se utilizan diferentes métricas obtenidas a partir de la comparación entre las clases reales y las clases predichas por el modelo. El uso de varias métricas permite realizar una valoración más

completa, debido a que cada una representa un aspecto diferente del desempeño del clasificador (Sokolova & Lapalme, 2009).

Una de las principales herramientas de evaluación es la matriz de confusión, en la cual se organizan las predicciones correctas e incorrectas para cada categoría. En un problema multiclase, las filas pueden representar las clases reales y las columnas las clases predichas. Los valores ubicados en la diagonal principal corresponden a las clasificaciones correctas, mientras que los demás valores muestran los casos en los que una categoría fue confundida con otra (Sokolova & Lapalme, 2009).

La exactitud, conocida también como accuracy, indica el porcentaje de registros que el modelo clasificó de forma correcta. Aunque permite obtener una idea general de su desempeño, puede ser poco representativa cuando existe una distribución desigual entre las categorías. Un modelo podría alcanzar una exactitud aparentemente favorable al clasificar correctamente la categoría más frecuente, aunque presente un desempeño bajo en las clases con menor representación (Branco et al., 2016).

La precisión muestra cuántos de los registros que el modelo asignó a una categoría pertenecían realmente a ella. En cambio, el recall o sensibilidad mide cuántos de los registros que formaban parte de una clase fueron reconocidos correctamente por el modelo. El F1-score combina la precisión y el recall mediante una media armónica, por lo que resulta útil cuando se busca mantener un equilibrio entre ambas medidas (Sokolova & Lapalme, 2009).

En una clasificación con varias categorías, el F1-score se obtiene para cada clase y luego puede combinarse mediante distintos promedios. El F1 macro concede el mismo peso a todas las categorías, sin considerar cuántos registros posee cada una, esto permite comprobar si el modelo presenta un desempeño equilibrado entre las categorías. En cambio, el F1 ponderado, conocido como F1 weighted, considera la cantidad de registros de cada clase, por lo que las categorías más frecuentes tienen una mayor influencia en el resultado final. La comparación de ambos valores ayuda a reconocer posibles diferencias de desempeño entre las clases (Sokolova & Lapalme, 2009).

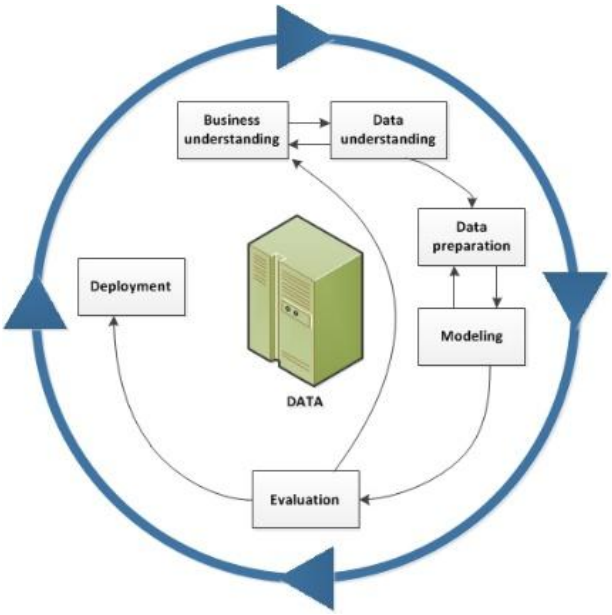
2.8. Metodología CRISP-DM

CRISP-DM (Cross-Industry Standard Process for Data Mining) es una metodología que se utiliza para desarrollar trabajos de minería de datos.

Fue propuesta originalmente por Chapman et al. (1999), como un estándar industrial orientado a guiar de manera sistemática el ciclo de vida de un proyecto de análisis de datos, desde la comprensión del problema hasta la implementación del modelo.

Figura 3:

Metodología CRISP-DM



Nota. Tomado de *Conceptos básicos de ayuda de CRISP-DM*, por IBM, 2021, IBM Documentation (https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview). Crédito de la imagen: IBM.

CRISP-DM se compone de seis fases interrelacionadas: comprensión del negocio, comprensión de los datos, preparación de los datos, modelado, evaluación e implementación. La ventaja de utilizar esta metodología es que las fases no siguen necesariamente un proceso estrictamente lineal, sino que permiten iteraciones y retroalimentación entre etapas, favoreciendo un enfoque flexible y adaptativo (Chapman et al., 1999).

3. CAPÍTULO III: METODOLOGÍA

En esta investigación se analizaron datos históricos sobre el precio del tomate riñón en mercados mayoristas del Ecuador, obtenidos del SIPA. El análisis incluyó la exploración inicial de los datos, la preparación de variables y la aplicación de algoritmos de Machine Learning para clasificar la variación mensual del precio en tres categorías: baja, estabilidad y alza.

El desarrollo del trabajo se organizó mediante la metodología CRISP-DM, la cual sirvió como guía para ordenar el proceso desde la comprensión del problema y de los datos, hasta la preparación de la base de modelado, el entrenamiento, la optimización y la evaluación de los modelos de clasificación.

3.1. Enfoque de la investigación

La investigación tuvo un enfoque cuantitativo, ya que trabajó con datos numéricos correspondientes a los precios históricos del tomate riñón registrados en mercados mayoristas del Ecuador. Esta información fue procesada mediante técnicas estadísticas, análisis exploratorio y algoritmos de Machine Learning, con el fin de analizar el comportamiento histórico del precio y clasificar su variación mensual en las categorías baja, estabilidad y alza.

El diseño de la investigación fue no experimental, ya que las variables no fueron manipuladas por el investigador. En este tipo de diseño, los fenómenos se estudian en su contexto original sin que el investigador intervenga directamente en las condiciones en las que se produjeron los datos (Hernández Sampieri & Mendoza Torres, 2018). En este caso, los registros fueron analizados tal como se encontraban disponibles en las bases de datos públicas del SIPA.

Asimismo, la investigación tuvo un carácter longitudinal y retrospectivo, debido a que se trabajó con datos registrados a lo largo del tiempo, correspondientes al período comprendido entre enero de 2012 y abril de 2026. Esto permitió observar la evolución del precio del tomate riñón en distintos meses, años, mercados y provincias.

El alcance del estudio fue descriptivo y comparativo. Fue descriptivo porque se analizaron tendencias, patrones temporales, diferencias geográficas y variaciones del precio. Fue comparativo porque se entrenaron y evaluaron distintos modelos de clasificación, con el

objetivo de identificar cuál presentaba el mejor desempeño para clasificar la variación mensual del precio del tomate riñón.

3.2. Fuente y recolección de datos

La información principal utilizada en esta investigación provino del SIPA, plataforma oficial administrada por el Ministerio de Agricultura y Ganadería del Ecuador. De esta fuente se obtuvieron los registros públicos de precios del tomate riñón en mercados mayoristas del Ecuador, correspondientes al período entre enero de 2012 y abril de 2026.

La información se obtuvo mediante la revisión documental de archivos digitales disponibles en fuentes públicas. Por tratarse de registros previamente recopilados y publicados por instituciones oficiales, el estudio trabajó con datos secundarios y no requirió la aplicación de encuestas, entrevistas o experimentos.

El dataset principal contenía información sobre el año, mes, tipo de mercado, mercado, provincia, cantón, producto, presentación, cantidad, unidad de medida, precio promedio y precio promedio por kilogramo. A partir de estos registros se seleccionaron únicamente las observaciones correspondientes al tomate riñón, diferenciando entre tomate riñón de invernadero y tomate riñón a campo abierto.

Además de la base principal, se revisaron e integraron otros conjuntos de datos relacionados con precios al productor, insumos agrícolas, producción, superficie, rendimiento y variables climáticas. Esta información se incorporó para evaluar si los datos adicionales podían mejorar la clasificación de la variación mensual del precio. No obstante, su inclusión en el modelo final estuvo condicionada por la cobertura de cada fuente, la disponibilidad temporal de los registros y el aporte observado en el desempeño de los modelos.

Antes del análisis, todos los archivos fueron organizados, depurados y transformados para asegurar una estructura adecuada de trabajo. Luego, la información se consolidó en una base mensual por provincia, mercado y tipo de producción. Esta organización permitió calcular la variación del precio entre meses consecutivos y construir la variable objetivo utilizada en la clasificación.

3.3. Aplicación de la metodología CRISP-DM

La metodología CRISP-DM fue utilizada para organizar las etapas de la investigación, ya que se adapta al desarrollo de proyectos de análisis de datos y Machine Learning. En este estudio, su aplicación ayudó a ordenar el trabajo desde la comprensión del problema hasta la evaluación de los modelos de clasificación.

En la fase de comprensión de los datos se revisaron los archivos obtenidos del SIPA para conocer su estructura, las variables disponibles, el período de cobertura, los mercados registrados, los tipos de producción y posibles problemas de calidad. Esta etapa permitió tener una visión general de la información antes de continuar con su transformación.

En la fase de preparación de los datos se aplicaron tareas de limpieza, estandarización, integración y consolidación mensual de los registros. También se generaron variables temporales, rezagos de precio y la variable objetivo, definida a partir de la variación porcentual entre meses consecutivos.

En la fase de modelado se trabajó con distintos escenarios de análisis y se entrenaron modelos de clasificación supervisada, como Dummy Classifier, regresión logística, Random Forest y Gradient Boosting. Luego de comparar sus resultados, el modelo con mejor desempeño fue seleccionado para el ajuste de hiperparámetros.

Finalmente, en la fase de evaluación se analizaron los resultados mediante métricas de clasificación como accuracy, precisión, recall, F1-score, F1 macro, F1 weighted y matriz de confusión. La fase de implementación no contempló el despliegue de una aplicación, sino la presentación e interpretación de los resultados obtenidos dentro del documento de investigación.

3.4. Desarrollo metodológico de la investigación

3.4.1. Comprensión y exploración de los datos

La comprensión de los datos comenzó con la revisión del dataset principal de precios de mercados mayoristas obtenido del SIPA. Se examinó la estructura general del archivo, el número de registros, las columnas disponibles, los tipos de datos y la presencia de valores faltantes. Esta base incluía información sobre año, mes, tipo de mercado, mercado, provincia, cantón, producto, presentación, cantidad, unidad de medida, precio promedio y precio promedio por kilogramo.

Luego se revisaron los productos registrados en la base para identificar las observaciones correspondientes al tomate riñón. En este proceso se encontraron distintas denominaciones del producto, principalmente tomate riñón de invernadero y tomate riñón a campo abierto. Esta distinción fue importante porque permitió asociar cada registro con su respectivo tipo de producción.

También se revisó la cobertura temporal y geográfica de la información disponible. Para ello, se verificaron los años y meses registrados, junto con las provincias, cantones y mercados incluidos en la base. Esta revisión mostró que no todas las series de mercado contaban con información completa durante todo el período de estudio. Por esta razón, para calcular la variación del precio se trabajó únicamente con transiciones mensuales consecutivas.

En el análisis exploratorio inicial se revisó cómo se distribuía el precio del tomate riñón en dólares por kilogramo, junto con las diferencias entre tipos de producción, la evolución temporal del precio y la presencia de posibles valores atípicos. También se observaron patrones estacionales y diferencias entre mercados y provincias, lo que ayudó a comprender mejor el comportamiento histórico del precio antes de construir la base de modelado.

Los hallazgos de esta etapa sirvieron como base para las decisiones de preparación de datos, entre ellas la estandarización de nombres, la consolidación mensual de precios, la creación de variables temporales y la construcción de la variable objetivo de clasificación.

3.4.2. Preparación y transformación de los datos

Una vez revisada la información inicial, se prepararon y transformaron los datos para construir una base mensual consistente, adecuada para el análisis y el modelado. Para ello, se verificaron los formatos de las variables originales y se convirtieron a tipo numérico las columnas que contenían valores registrados como texto, principalmente aquellas relacionadas con la cantidad, el precio promedio y el precio promedio por kilogramo.

Luego se filtraron los registros correspondientes al tomate riñón. En esta revisión se encontraron distintas formas de escritura del producto, por lo que se normalizó su nombre para agrupar las variables de tomate riñón de invernadero y tomate riñón a campo abierto. Con esta clasificación se creó la variable Tipo_Produccion, que permitió diferenciar los registros entre invernadero y campo abierto.

También se transformó la variable mes a un formato numérico y se construyó una variable de fecha mensual, utilizando el primer día de cada mes como referencia. Esta transformación permitió ordenar cronológicamente los registros y facilitar el análisis de la evolución temporal del precio.

Como podían existir varios registros para un mismo mes, mercado, provincia y tipo de producción, principalmente por diferencias en las presentaciones comerciales del producto, se aplicó una consolidación mensual. La información se agrupó por fecha, año, mes, tipo de mercado, mercado, provincia, cantón y tipo de producción. El precio mensual consolidado se calculó mediante la mediana del precio en dólares por kilogramo, con el fin de reducir el efecto de posibles valores extremos o diferencias entre presentaciones.

Además, se revisaron inconsistencias en los nombres de provincias, mercados y cantones. Se encontraron diferencias relacionadas con el uso de mayúsculas, minúsculas y formas de escritura, por lo que se aplicó un proceso de estandarización para unificar las categorías. Esta etapa permitió evitar que un mismo mercado o provincia fuera tratado como categorías diferentes durante el análisis y el modelado.

Como resultado de este proceso, se obtuvo una base organizada a nivel mensual, en la que cada registro representó el precio consolidado del tomate riñón para una combinación específica de fecha, provincia, mercado y tipo de producción. Esta base sirvió como insumo principal para calcular la variación del precio entre meses consecutivos y construir posteriormente la variable objetivo de clasificación.

En la Tabla 2 se resumen las principales variables utilizadas durante la preparación de los datos, incluyendo tanto las variables originales como aquellas construidas para organizar la base mensual de análisis.

Tabla 2:

Variables consideradas en la preparación de los datos

Variable	Descripción	Tipo de dato	Tratamiento o función
Año	Año correspondiente al registro	Numérica temporal	Utilizada para construir la fecha mensual
Mes	Mes correspondiente al registro	Temporal	Transformado a número de mes
Fecha	Fecha mensual del registro	Temporal	Construida a partir del año y mes
Provincia	Provincia donde se encuentra el mercado	Categórica	Estandarizada para corregir diferencias de escritura
Cantón	Cantón correspondiente al mercado	Categórica	Revisado y estandarizado cuando fue necesario
Mercado	Mercado mayorista del registro	Categórica	Estandarizado y utilizado como variable geográfica y comercial
Tipo Mercado	Clasificación del mercado registrado	Categórica	Conservado como variable descriptiva y predictora
Producto	Nombre del producto agrícola	Categórica	Filtrado para conservar únicamente tomate riñón
Tipo_Produccion	Tipo de producción del tomate riñón	Categórica	Construida como invernadero o campo abierto
Presentación	Forma de comercialización del producto	Categórica	Utilizada para revisar diferencias entre registros
Unidad de medida	Unidad en la que se registró el producto	Categórica	Revisada para trabajar con precios comparables
Precio_USD_Kg	Precio mensual consolidado en dólares por kilogramo	Numérica continua	Calculado mediante la mediana por fecha, provincia, mercado y tipo de producción

Fuente: Autor

Estas variables permitieron estructurar la información en una unidad de análisis mensual, diferenciada por fecha, provincia, mercado y tipo de producción. A partir de esta base preparada se continuó con el cálculo de la variación mensual del precio y la construcción de la variable objetivo utilizada en los modelos de clasificación.

3.4.3. Construcción de la variable objetivo

La variable objetivo se construyó a partir de la variación porcentual mensual del precio del tomate riñón. Para calcularla, se comparó el precio mensual consolidado de cada observación

con el precio registrado en el mes siguiente dentro de la misma serie temporal. Cada serie se definió por la combinación de provincia, mercado y tipo de producción, de modo que no se compararan registros pertenecientes a contextos comerciales o productivos diferentes.

Antes de calcular la variación, los registros fueron ordenados cronológicamente dentro de cada serie. Además, se verificó que el registro siguiente correspondiera realmente al mes consecutivo. Esta condición fue necesaria debido a que algunas series no presentaban información completa durante todo el período analizado. Es por ello que solo se utilizaron transiciones mensuales consecutivas para construir la variable objetivo.

La variación porcentual mensual se obtuvo al comparar el precio del mes actual con el precio del mes siguiente. Con ese valor, cada observación se clasificó en una de tres categorías: baja, estabilidad o alza. Para separar estas clases se aplicó un umbral de ±10 %, utilizado para diferenciar cambios relevantes del precio frente a variaciones menores.

La selección de este umbral se realizó después de revisar diferentes alternativas de clasificación. Umbrales más bajos, como ±5 % o ±7,5 %, generaban una mayor sensibilidad ante cambios pequeños del precio, mientras que umbrales más altos, como ±12,5 % o ±15 %, concentraban demasiadas observaciones en la categoría estabilidad y reducían la representación de las clases baja y alza. Por esta razón, el umbral de ±10 % fue considerado adecuado, ya que permitió mantener una distribución aceptable entre las tres categorías y, al mismo tiempo, representar como baja o alza únicamente los cambios de mayor relevancia.

Los criterios aplicados para construir la variable objetivo se presentan en la Tabla 3.

Tabla 3:

Categorías de clasificación de la variación mensual del precio

Categoría	Criterio de clasificación	Interpretación
Baja	Variación menor o igual a -10 %	El precio del mes siguiente disminuye de forma relevante frente al mes actual
Estabilidad	Variación mayor a -10 % y menor a 10 %	El precio del mes siguiente se mantiene dentro de un rango moderado de cambio
Alza	Variación mayor o igual a 10 %	El precio del mes siguiente aumenta de forma relevante frente al mes actual

Fuente: Autor

3.4.4. Integración de datasets complementarios

Además del dataset principal de precios de mercados mayoristas, se integraron conjuntos de datos complementarios relacionados con factores que podían influir en la variación mensual del precio del tomate riñón. Con esta información se buscó evaluar si variables externas al precio mayorista aportaban mejoras al desempeño de los modelos de clasificación.

Los datasets complementarios incluyeron información sobre precios al productor, precios de insumos agrícolas, superficie, producción, rendimiento y variables climáticas. Antes de integrarlos con la base principal, cada conjunto de datos fue revisado, depurado y transformado. En este proceso se estandarizaron los nombres de provincias, se construyeron fechas mensuales o anuales según el tipo de información disponible y se generaron variables rezagadas para evitar el uso de información futura dentro del modelo.

La integración de los datasets se realizó utilizando variables comunes de tiempo y ubicación. Para los precios al productor se consideraron la fecha, la provincia y el tipo de producción, mientras que los insumos agrícolas y las variables climáticas se unieron principalmente por fecha y provincia. En el caso de las variables de superficie, producción y rendimiento, al estar disponibles con frecuencia anual, se incorporaron mediante rezago anual, de modo que la información productiva del año previo pudiera relacionarse con los registros mensuales del año siguiente.

Los datasets complementarios considerados se resumen en la Tabla 4.

Tabla 4:

Datasets complementarios considerados en la investigación

Dataset complementario	Variables consideradas	Forma de integración	Finalidad
Precios al productor	Precio al productor, rezagos y variación del precio productor	Fecha, provincia y tipo de producción	Evaluar la relación entre el precio recibido por el productor y la variación posterior del precio mayorista
Insumos agrícolas	Índices de fertilizantes, fungicidas, herbicidas e insecticidas	Fecha y provincia	Incorporar información relacionada con costos de producción agrícola
Superficie, producción y rendimiento	Superficie cosechada, producción total y rendimiento	Provincia y año rezagado	Representar condiciones productivas anuales asociadas al tomate riñón
Variables climáticas	Precipitación, temperatura y variables rezagadas	Fecha y provincia	Evaluar posibles relaciones entre condiciones climáticas y variaciones del precio

Fuente: Autor

Al integrar los datasets complementarios, se obtuvo una base de modelado más amplia, con variables provenientes de diferentes fuentes. Aun así, varias de estas variables no se incorporaron al modelo final, principalmente por limitaciones de cobertura temporal, disponibilidad geográfica y valores faltantes. Por ello, se plantearon distintos escenarios de

modelado, con el fin de comparar el desempeño de los modelos al trabajar con diferentes grupos de variables.

3.4.5. Definición de escenarios de modelado

Después de integrar la base principal con los datasets complementarios, se definieron distintos escenarios de modelado para analizar el aporte de diferentes grupos de variables en la clasificación de la variación mensual del precio del tomate riñón. Esta decisión respondió a que no todas las variables tenían la misma cobertura temporal y geográfica, situación que podía reducir la cantidad de registros disponibles para el entrenamiento de los modelos.

La base final integrada incluyó variables temporales, geográficas, comerciales e históricas del precio, junto con variables complementarias relacionadas con precios al productor, insumos agrícolas, producción, rendimiento y clima. No obstante, al incorporar más fuentes de información también aumentó la presencia de valores faltantes, sobre todo en aquellos datasets cuya cobertura no coincidía completamente con la base principal de mercados mayoristas.

Debido a estas diferencias en la disponibilidad de los datos, el primer escenario trabajó solo con las variables base del precio mayorista y sus características temporales, mientras que los demás incorporaron gradualmente variables complementarias. Esta comparación permitió analizar si dichas variables mejoraban el desempeño de los modelos o si la reducción de registros completos limitaba su utilidad.

Los escenarios definidos para el modelado se presentan en la Tabla 5.

Tabla 5:

Escenarios de modelado definidos en la investigación

Escenario	Variables consideradas	Número de predictoras	Registros completos	Propósito
Escenario 1: Base	Variables temporales, geográficas, comerciales y rezagos del precio mayorista	15	1831	Servir como modelo principal con mayor cobertura de datos
Escenario 2: Base + insumos + producción	Variables base, índices de insumos agrícolas y variables productivas	26	1373	Evaluar el aporte de costos de producción y condiciones productivas
Escenario 3: Base + clima	Variables base y variables climáticas rezagadas	21	995	Analizar si la información climática aportaba al desempeño del modelo
Escenario 4: Completo sin productor	Variables base, insumos, producción y clima	32	784	Evaluar una base integrada amplia sin precios al productor
Escenario 5: Completo con productor	Variables base, insumos, producción, clima y precios al productor	37	192	Probar el escenario más completo, considerando todas las fuentes disponibles

Fuente: Autor

La comparación de escenarios permitió seleccionar una estrategia de modelado adecuada, considerando no solo el número de variables disponibles, sino también la cantidad de registros completos y el desempeño obtenido por los modelos. De esta manera, el modelo final no fue elegido únicamente por incluir más variables, sino por presentar un mejor equilibrio entre cobertura de datos, capacidad predictiva e interpretación de resultados.

3.4.6. Selección de variables predictoras

La selección de variables predictoras tomó en cuenta la disponibilidad de información, su relación temporal con la variable objetivo y la necesidad de evitar el uso de datos futuros durante el entrenamiento de los modelos. Como la investigación buscó clasificar la variación mensual del precio para el mes siguiente, se utilizaron solo variables disponibles hasta el mes actual o provenientes de períodos anteriores.

En el escenario base se seleccionaron variables temporales, geográficas, comerciales e históricas del precio mayorista. Estas variables permitieron representar el comportamiento reciente del precio del tomate riñón, así como las diferencias asociadas al mes, trimestre, provincia, mercado, tipo de mercado y tipo de producción.

Las variables relacionadas con el precio futuro, como el precio del mes siguiente y la variación porcentual utilizada para construir la variable objetivo, no fueron incluidas como predictoras. Esta decisión permitió evitar fuga de información, es decir, impedir que el modelo tuviera acceso a datos que no estarían disponibles en un contexto real de clasificación.

Las variables predictoras utilizadas en el escenario base se presentan en la Tabla 6.

Tabla 6:

Variables predictoras utilizadas en el escenario base

Grupo de variables	Variables incluidas	Descripción
Variables geográficas y comerciales	Provincia, Mercado, Tipo Mercado, Tipo_Produccion	Representan la ubicación, el mercado de comercialización y el tipo de producción del tomate riñón
Variables temporales	Trimestre, Mes_numero, Mes_sin, Mes_cos	Representan el comportamiento temporal y estacional del precio
Precio actual	Precio_USD_Kg	Precio mensual consolidado del tomate riñón en dólares por kilogramo
Rezagos de precio	Precio_lag_1, Precio_lag_2, Precio_lag_3	Precios registrados uno, dos y tres meses antes dentro de la misma serie temporal
Variación histórica	Variacion_pct_lag_1	Variación porcentual del precio respecto al mes anterior
Tendencia reciente	Promedio_movil_3	Promedio móvil de tres meses del precio
Control de historial	Historial_3_meses_completo	Variable que identifica si la serie cuenta con información completa de los tres meses anteriores

Fuente: Autor

Además de las variables del escenario base, en los escenarios complementarios se evaluaron variables relacionadas con precios al productor, índices de insumos agrícolas, superficie, producción, rendimiento y clima. Sin embargo, estas variables fueron utilizadas de forma comparativa, debido a que presentaban diferencias de cobertura y disponibilidad según la fuente de datos. Por esta razón, la selección final del modelo consideró tanto el desempeño obtenido como la cantidad de registros disponibles para el entrenamiento y la evaluación.

3.4.7. División temporal y preprocesamiento para el modelado

Para el entrenamiento y evaluación de los modelos se utilizó una división temporal de los datos, debido a que la información analizada corresponde a una serie histórica mensual. Esta decisión

permitió evaluar los modelos en un contexto más cercano a una situación real, en la que se entrena con información pasada y se prueba con registros posteriores.

En el escenario base, la información disponible para el modelado estuvo conformada por 1831 observaciones completas. Los registros anteriores al año 2024 fueron utilizados para el entrenamiento de los modelos, mientras que los registros correspondientes al período 2024-2026 fueron reservados para la prueba final. De esta manera, el modelo fue evaluado con datos que no participaron en el proceso de entrenamiento.

La división de los datos se presenta en la Tabla 7.

Tabla 7:

División temporal de los datos para el escenario base

Conjunto de datos	Período considerado	Número de registros	Finalidad
Entrenamiento	Registros anteriores a 2024	1398	Entrenar los modelos de clasificación
Prueba	Registros desde 2024 hasta 2026	433	Evaluar el desempeño final de los modelos
Total	2012-2026	1831	Base completa del escenario base

Fuente: Autor

Antes del entrenamiento, las variables fueron separadas según su tipo de dato. Las variables numéricas fueron procesadas mediante imputación de valores faltantes con la mediana y posteriormente estandarizadas. Este procedimiento permitió que las variables continuas estuvieran en una escala comparable, especialmente para modelos sensibles a la escala, como la regresión logística.

Por otra parte, las variables categóricas fueron procesadas mediante imputación con el valor más frecuente y codificación One-Hot Encoding. Esta técnica permitió transformar variables como provincia, mercado, tipo de mercado y tipo de producción en un formato numérico que pudiera ser interpretado por los algoritmos de Machine Learning.

El preprocesamiento fue aplicado dentro de un flujo de trabajo o pipeline, con el objetivo de asegurar que las transformaciones se ajustaran únicamente con los datos de entrenamiento y luego fueran aplicadas al conjunto de prueba. Esta decisión permitió evitar fuga de información y garantizar una evaluación más adecuada del desempeño de los modelos.

3.4.8. Entrenamiento, optimización y evaluación de modelos

Una vez preparada la base de modelado y definido el conjunto de variables predictoras, se entrenaron diferentes algoritmos de clasificación supervisada con el objetivo de identificar la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza. Los modelos utilizados fueron Dummy Classifier, regresión logística, Random Forest y Gradient Boosting.

El Dummy Classifier se utilizó como modelo de referencia o línea base, para contar con un punto mínimo de comparación. Su inclusión permitió verificar si los algoritmos de mayor complejidad lograban superar una estrategia simple basada en la clase más frecuente. Luego, se entrenó una regresión logística como modelo lineal de comparación, junto con dos modelos basados en árboles de decisión: Random Forest y Gradient Boosting.

Cada modelo se incorporó dentro de un pipeline de preprocesamiento y clasificación, para aplicar de forma ordenada las transformaciones requeridas en las variables numéricas y categóricas. De esta manera, la imputación, estandarización y codificación se ajustaron solo con el conjunto de entrenamiento, evitando que la información del conjunto de prueba influyera en el entrenamiento de los modelos.

La evaluación inicial se realizó sobre el conjunto de prueba definido mediante la división temporal. Para comparar el desempeño de los modelos se utilizaron métricas adecuadas para clasificación multiclase, entre ellas accuracy, precisión, recall, F1-score, F1 macro, F1 weighted y matriz de confusión. El F1 macro fue considerado una métrica relevante debido a que permite evaluar el desempeño promedio entre las tres clases, sin otorgar mayor peso a la clase con más registros.

Después de comparar los modelos iniciales, se seleccionó el algoritmo con mejor desempeño para realizar una optimización de hiperparámetros. En este proceso se evaluaron distintas combinaciones de parámetros del modelo Random Forest, tales como el número de árboles,

profundidad máxima, número mínimo de muestras por hoja, número mínimo de muestras para dividir un nodo, número de variables consideradas en cada división y ponderación de clases.

Para evitar que la optimización dependiera directamente del conjunto de prueba final, el conjunto de entrenamiento fue dividido internamente en un subconjunto de entrenamiento y otro de validación. De esta forma, la selección de hiperparámetros se realizó utilizando datos anteriores al período de prueba, mientras que la evaluación final se mantuvo reservada para los registros correspondientes al período 2024-2026.

Finalmente, el modelo optimizado fue entrenado nuevamente con todo el conjunto de entrenamiento y evaluado sobre el conjunto de prueba final. Este procedimiento permitió comparar el desempeño del modelo optimizado frente a la línea base y los modelos iniciales, además de analizar sus aciertos, errores, importancia de variables y capacidad para clasificar cada una de las categorías de variación mensual del precio.

3.5. Consideraciones éticas

Para el desarrollo de esta investigación se utilizaron datos secundarios y de acceso público, provenientes del SIPA. Al trabajar con este tipo de información, no se utilizaron datos personales, sensibles ni registros que permitan identificar a personas naturales. Por esta razón, el estudio no requirió la aplicación de encuestas o entrevistas.

Durante el desarrollo del estudio se procuró mantener la integridad de los datos mediante procesos de limpieza, estandarización y transformación documentados. Las modificaciones realizadas tuvieron como finalidad corregir formatos, organizar la información y construir variables útiles para el análisis, sin alterar intencionalmente el sentido original de los registros.

También se consideró importante evitar interpretaciones absolutas sobre los resultados obtenidos. Los modelos de Machine Learning desarrollados en esta investigación fueron utilizados con fines académicos y exploratorios, orientados a clasificar la variación mensual del precio del tomate riñón a partir de la información disponible. Por tanto, sus resultados no deben entenderse como una predicción definitiva del comportamiento futuro del mercado ni como una herramienta oficial para la toma de decisiones comerciales.

Finalmente, se reconoce que el desempeño de los modelos depende de la calidad, cobertura y disponibilidad de los datos utilizados. En consecuencia, las conclusiones fueron formuladas considerando las limitaciones de las fuentes de información y evitando atribuir al modelo una capacidad mayor a la evidenciada por sus métricas de evaluación.

4. CAPÍTULO IV: RESULTADOS Y ANÁLISIS

En esta sección se presentan los principales resultados obtenidos durante el análisis de los datos, la construcción de la variable objetivo y el entrenamiento de los modelos de clasificación. Los resultados se organizan de acuerdo con las etapas desarrolladas en la metodología, iniciando con el análisis exploratorio del precio del tomate riñón, seguido de la distribución de las clases de variación mensual, la comparación de escenarios de modelado, la evaluación de los algoritmos y el análisis del modelo final seleccionado.

4.1. Resultados del análisis exploratorio

El análisis exploratorio permitió conocer la estructura general de la información disponible y comprender el comportamiento histórico del precio del tomate riñón antes de construir los modelos de clasificación. Inicialmente, el dataset principal de mercados mayoristas contenía 56.297 registros correspondientes a diferentes productos agrícolas. Luego del proceso de filtrado, se conservaron únicamente los registros relacionados con tomate riñón de invernadero y tomate riñón a campo abierto.

Después de la limpieza, estandarización y consolidación mensual, la base principal quedó organizada a nivel de fecha, provincia, mercado y tipo de producción. Esta estructura permitió analizar el precio mensual consolidado del tomate riñón en dólares por kilogramo y construir posteriormente la variable objetivo de clasificación.

Con el propósito de resumir el proceso aplicado a la base principal, en la Tabla 8 se presenta la reducción progresiva de registros desde el dataset original hasta la obtención de las transiciones mensuales consecutivas utilizadas para construir la variable objetivo.

Tabla 8:

Resumen del procesamiento de la base principal de mercados mayoristas

Etapa del procesamiento	Número de registros	Descripción
Dataset original de mercados mayoristas	56.297	Registros iniciales de diferentes productos agrícolas
Registros filtrados de tomate riñón	2.303	Observaciones correspondientes a tomate riñón de invernadero y campo abierto
Base mensual consolidada	2.117	Registros consolidados por fecha, provincia, mercado y tipo de producción
Transiciones mensuales consecutivas válidas	2.032	Observaciones utilizadas para construir la variable objetivo

Fuente: Autor

A partir de este procesamiento, la base mensual consolidada quedó conformada por 2.117 registros. De estos, 2.032 correspondieron a transiciones mensuales consecutivas válidas, las cuales fueron utilizadas posteriormente para calcular la variación porcentual del precio hacia el mes siguiente.

Para analizar el comportamiento general del precio mensual consolidado del tomate riñón, se calcularon medidas descriptivas como la media, mediana, desviación estándar, valores mínimos, máximos y cuartiles. Estos resultados se presentan en la Tabla 9.

Tabla 9:

Estadísticas descriptivas del precio mensual consolidado del tomate riñón

Indicador	Valor
Número de observaciones	2.117
Media	0,616 USD/kg
Mediana	0,600 USD/kg
Desviación estándar	0,172
Valor mínimo	0,220 USD/kg
Primer cuartil	0,495 USD/kg
Tercer cuartil	0,710 USD/kg
Valor máximo	1,590 USD/kg

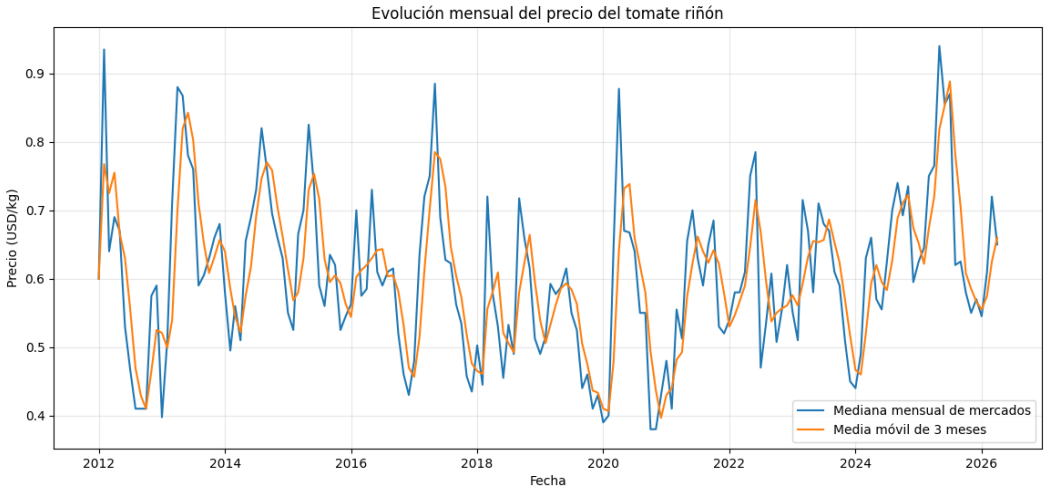
Fuente: Autor

El precio mensual consolidado del tomate riñón registró una mediana de 0,600 USD/kg y una media de 0,616 USD/kg. La cercanía entre estos valores muestra que la mayor parte de los precios se ubicó alrededor de ese rango. Sin embargo, también se observaron diferencias importantes dentro de la base, con valores que fueron desde 0,220 USD/kg hasta 1,590 USD/kg, lo que refleja variaciones entre distintos períodos, mercados y tipos de producción.

La evolución temporal del precio mensual consolidado se presenta en la Figura 4.

Figura 4:

Evolución mensual del precio del tomate riñón



Fuente: Autor

Esta figura permite observar que el precio del tomate riñón no mantuvo un comportamiento uniforme durante el período analizado, sino que presentó fluctuaciones a lo largo del tiempo. Además, la media móvil de tres meses permite visualizar con mayor claridad la tendencia general del precio, reduciendo el efecto de variaciones mensuales puntuales.

Debido a que la base diferenciaba entre tomate riñón de invernadero y tomate riñón a campo abierto, se comparó el precio mensual consolidado según el tipo de producción. Los resultados de esta comparación se muestran en la Tabla 10.

Tabla 10:

Precio mensual consolidado según tipo de producción

Tipo de producción	Número de registros	Media	Mediana	Mínimo	Máximo
Campo abierto	576	0,513	0,490	0,220	1,330
Invernadero	1.541	0,654	0,640	0,300	1,590

Fuente: Autor

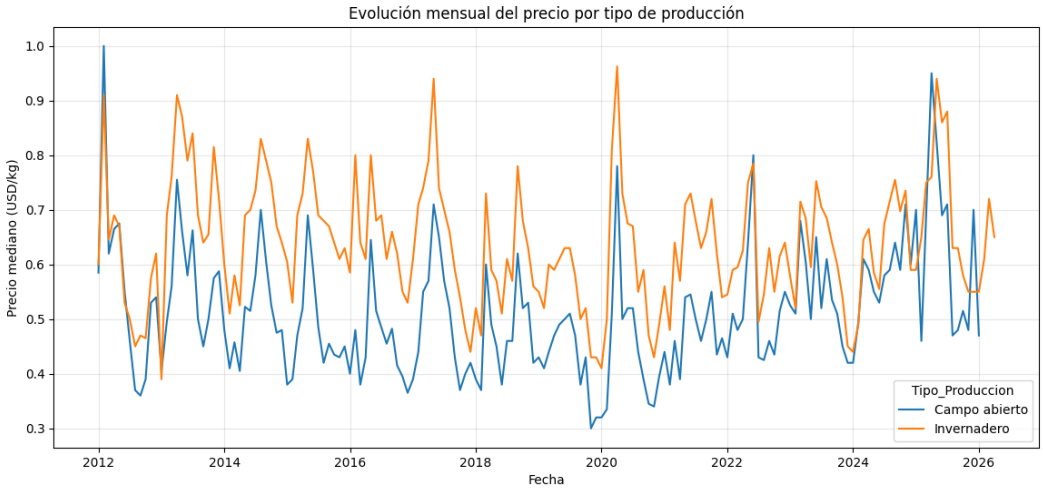
La comparación por tipo de producción evidencia que el tomate riñón de invernadero presentó precios superiores al tomate riñón a campo abierto. Mientras que el tomate de invernadero registró una media de 0,654 USD/kg y una mediana de 0,640 USD/kg, el tomate a campo abierto

presentó una media de 0,513 USD/kg y una mediana de 0,490 USD/kg. Esta diferencia puede estar asociada con las condiciones de producción, la disponibilidad del producto, la calidad percibida y las dinámicas de comercialización en los mercados mayoristas.

La Figura 5 muestra la evolución mensual del precio según el tipo de producción.

Figura 5:

Evolución mensual del precio por tipo de producción



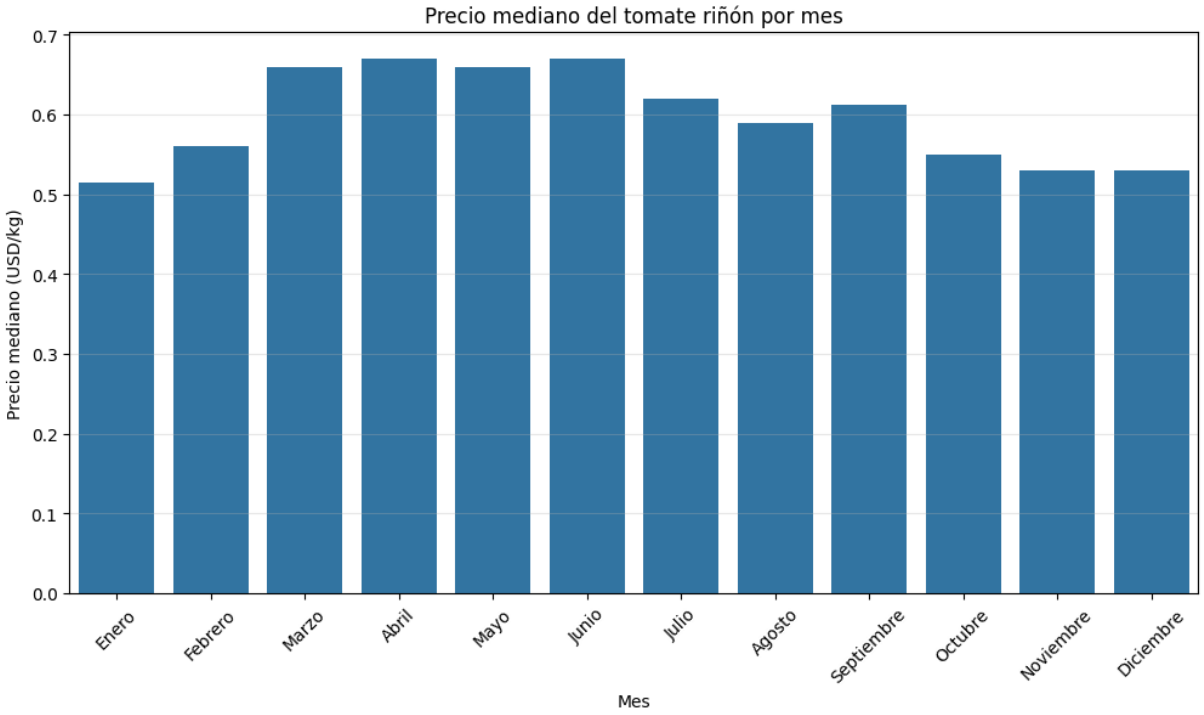
Fuente: Autor

En términos generales, el tomate riñón de invernadero presentó precios superiores al tomate riñón a campo abierto, aunque ambos tipos de producción evidenciaron fluctuaciones durante el período de análisis.

También se analizó el comportamiento estacional del precio, con el propósito de identificar diferencias entre los meses del año. Este análisis permite observar que el precio mediano no fue igual en todos los meses, sino que presentó variaciones según el período mensual.

La Figura 6 muestra el precio mediano del tomate riñón por mes.

Figura 6:
Precio mediano del tomate riñón por mes



Fuente: Autor

Se observa que los meses de marzo, abril, mayo y junio registraron valores medianos más altos, mientras que enero, noviembre y diciembre presentaron precios relativamente menores. Este comportamiento sugiere que el precio del tomate riñón puede estar influenciado por factores temporales relacionados con la oferta, la demanda, los ciclos de producción y las condiciones propias de comercialización en los mercados mayoristas.

Desde el punto de vista geográfico, la información estuvo distribuida en diferentes provincias y mercados del país. Las provincias con mayor número de registros fueron Guayas, Pichincha, Azuay, Imbabura, Cotopaxi, Manabí, Chimborazo y Tungurahua. A nivel de mercados, se destacaron Quito MMQ-EP, Guayaquil - TTV, Ibarra - COMERCIBARRA y Cuenca - EL ARENAL, debido a que concentraron una mayor cantidad de observaciones dentro de la base consolidada.

Finalmente, se revisó la presencia de posibles valores atípicos en el precio mensual consolidado. Aunque se identificaron algunos registros con valores superiores al rango esperado, estos representaron una proporción reducida de la base. Por esta razón, no fueron eliminados de forma

automática, ya que podían reflejar condiciones reales de variación del precio en determinados mercados o períodos. En su lugar, se utilizó la mediana durante la consolidación mensual para reducir el efecto de valores extremos asociados a diferencias entre presentaciones comerciales.

El análisis exploratorio permitió identificar variaciones importantes en el precio del tomate riñón, tanto a nivel temporal como geográfico y productivo. Estos resultados respaldan la decisión de construir una variable objetivo basada en la variación mensual del precio, así como la aplicación de modelos de clasificación supervisada para reconocer patrones asociados con baja, estabilidad o alza.

4.2. Resultados de la construcción de la variable objetivo

Después de consolidar la base mensual de precios, se construyó la variable objetivo a partir de la variación porcentual del precio hacia el mes siguiente. Para ello, cada registro fue comparado con el precio del mes posterior dentro de la misma serie temporal, definida por la combinación de provincia, mercado y tipo de producción.

Debido a que no todas las series tenían información completa en todos los meses, se utilizaron únicamente las transiciones mensuales consecutivas válidas. Como resultado, de los 2.117 registros de la base mensual consolidada, se obtuvieron 2.032 observaciones útiles para construir la variable objetivo.

La variación porcentual mensual fue clasificada mediante un umbral de ±10 %, de acuerdo con los criterios definidos en la metodología. De esta manera, las observaciones con una disminución igual o superior al 10 % fueron clasificadas como baja; aquellas con variaciones entre -10 % y 10 % fueron clasificadas como estabilidad; y las observaciones con un aumento igual o superior al 10 % fueron clasificadas como alza.

La distribución final de la variable objetivo se presenta en la Tabla 11.

Tabla 11:

Distribución de la variable objetivo

Categoría	Número de registros	Porcentaje
Baja	551	27,12 %
Estabilidad	918	45,18 %
Alza	563	27,71 %
Total	**2.032**	**100,00 %**

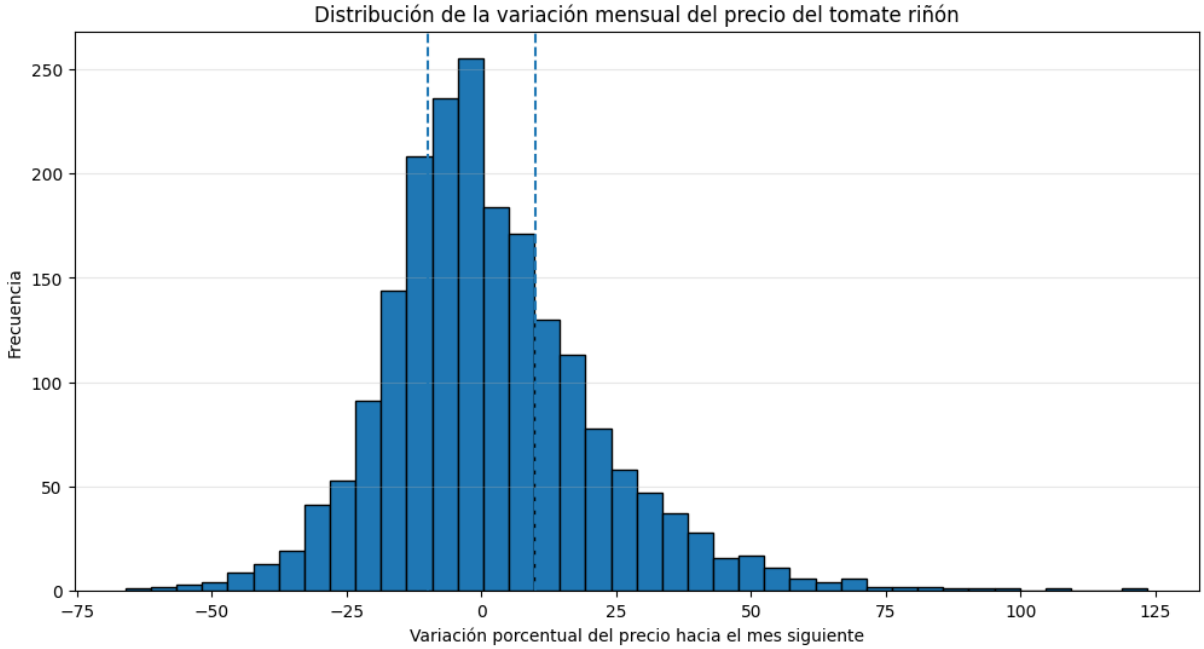
Fuente: Autor

La categoría con mayor presencia fue estabilidad, con 918 registros, lo que representa el 45,18 % del total. Por su parte, las categorías baja y alza tuvieron una distribución bastante cercana, con 551 y 563 registros, respectivamente. Aunque la estabilidad fue la clase más frecuente, las variaciones asociadas con disminuciones y aumentos relevantes del precio también conservaron una cantidad suficiente de observaciones para ser consideradas en el entrenamiento de los modelos de clasificación.

La distribución de la variación porcentual mensual del precio se presenta en la Figura 7. En esta visualización, las líneas verticales representan los umbrales de -10 % y 10 %, utilizados para separar las categorías baja, estabilidad y alza.

Figura 7:

Distribución de la variación mensual del precio del tomate riñón



Fuente: Autor

La Figura 7 permite observar que la mayor concentración de observaciones se ubicó alrededor de valores cercanos a cero, lo que indica que una parte importante de los registros presentó variaciones moderadas entre un mes y el siguiente. Sin embargo, también se identificaron observaciones con cambios negativos y positivos más amplios, las cuales justifican la separación de la variable objetivo en tres categorías.

La construcción de esta variable permitió replantear el análisis del precio mensual del tomate riñón como un problema de clasificación supervisada multiclase. A partir de ella, los modelos de Machine Learning fueron entrenados y evaluados con el propósito de identificar si la variación mensual del precio correspondía a una baja, estabilidad o alza.

4.3. Comparación de escenarios de modelado

Luego de construir la base principal y de integrar los datasets complementarios, se compararon los escenarios de modelado definidos en la metodología. El propósito de esta comparación fue evaluar si la incorporación de variables adicionales, como insumos agrícolas, producción, rendimiento, clima y precios al productor, mejoraba el desempeño de los modelos de clasificación o si, por el contrario, reducía demasiado la cantidad de registros disponibles.

La comparación permitió observar que, a medida que se incorporaban más fuentes de información, disminuía la cantidad de observaciones completas disponibles para el entrenamiento y evaluación de los modelos. Esta reducción se debió principalmente a diferencias de cobertura temporal y geográfica entre los datasets complementarios y la base principal de mercados mayoristas.

Los resultados generales de la comparación de escenarios se presentan en la Tabla 12.

Tabla 12:

Comparación general de escenarios de modelado

Escenario	Registros completos	Mejor modelo identificado	Accuracy	F1 macro	Interpretación
Escenario 1: Base	1.831	Random Forest	0,4758	0,4670	Presentó el mejor equilibrio entre cobertura y desempeño inicial
Escenario 2: Base + insumos + producción	1.373	Regresión logística	0,4473	0,4394	No superó el desempeño del escenario base
Escenario 3: Base + clima	995	No seleccionado como principal	—	—	La incorporación de clima redujo considerablemente la cantidad de registros completos
Escenario 4: Completo sin productor	784	No seleccionado como principal	—	—	La integración amplia redujo la base disponible para el modelado
Escenario 5: Completo con productor	192	No seleccionado como principal	—	—	La cobertura fue insuficiente para considerarlo como escenario principal

Fuente: Autor

Las métricas de desempeño se reportan únicamente para los escenarios que fueron entrenados y comparados directamente como alternativas viables de modelado. En los escenarios restantes, la reducción de registros completos limitó su uso como escenarios principales.

El Escenario 1, integrado por variables del precio mayorista, variables temporales, geográficas, comerciales y rezagos del precio, mostró el mejor balance entre la cantidad de registros disponibles y el desempeño inicial del modelo. Dentro de este escenario, Random Forest obtuvo el mejor resultado, con un accuracy de 0,4758 y un F1 macro de 0,4670.

Por otra parte, el Escenario 2 incorporó variables complementarias relacionadas con insumos agrícolas y producción. Aunque este escenario permitió evaluar información adicional vinculada con posibles costos y condiciones productivas, su desempeño no superó al escenario base. En este caso, el mejor modelo identificado fue la regresión logística, con un accuracy de 0,4473 y un F1 macro de 0,4394.

Los escenarios 3, 4 y 5 presentaron una reducción considerable en el número de registros completos. Esta situación limitó su utilidad como escenarios principales de modelado, debido a que una menor cantidad de observaciones podía afectar la capacidad de entrenamiento y evaluación de los modelos. En particular, el Escenario 5, que incorporaba precios al productor junto con todas las variables complementarias, quedó reducido a 192 registros completos, lo que resultó insuficiente para considerarlo como base principal.

Además, al comparar el Escenario 1 y el Escenario 2 bajo una muestra común, el escenario base continuó mostrando un mejor desempeño. Esto permitió confirmar que la incorporación de variables complementarias no necesariamente mejoró la clasificación de la variación mensual del precio, principalmente por la pérdida de observaciones completas y por el aporte limitado de dichas variables al desempeño del modelo.

En consecuencia, se seleccionó el Escenario 1 como escenario principal de modelado. Esta decisión no se basó únicamente en el valor de las métricas, sino también en la cobertura de datos, la consistencia temporal de la información y la posibilidad de interpretar los resultados de manera más clara. Los escenarios complementarios fueron útiles para evaluar el aporte potencial de otras fuentes de información, pero no fueron considerados como base final debido a sus limitaciones de cobertura y desempeño.

4.4. Comparación de modelos de clasificación

Una vez seleccionado el Escenario 1 como escenario principal de modelado, se entrenaron y compararon diferentes algoritmos de clasificación supervisada. El propósito de esta comparación fue identificar qué modelo presentaba el mejor desempeño inicial para clasificar la variación mensual del precio del tomate riñón en las categorías baja, estabilidad y alza.

Los modelos evaluados fueron Dummy Classifier, regresión logística, Random Forest y Gradient Boosting. El Dummy Classifier fue utilizado como modelo de referencia, debido a que permite establecer un desempeño mínimo basado en una estrategia simple. Los demás modelos fueron comparados frente a esta línea base con el fin de determinar si lograban aprender patrones útiles a partir de las variables predictoras.

La comparación inicial de los modelos se realizó utilizando el conjunto de prueba correspondiente al período 2024-2026. Para evaluar el desempeño se consideraron principalmente el accuracy y el F1 macro. Esta última métrica fue considerada especialmente relevante, debido a que permite analizar el rendimiento promedio entre las tres clases sin favorecer únicamente a la categoría con mayor número de registros.

Los resultados obtenidos se presentan en la Tabla 13.

Tabla 13:

Comparación inicial de modelos de clasificación en el escenario base

Modelo	Accuracy	F1 macro	Interpretación
Dummy Classifier	0,4365	0,2026	Modelo de referencia basado en la clase mayoritaria
Regresión logística	0,4270	0,4280	Mejoró claramente el F1 macro frente a la línea base
Random Forest	0,4758	0,4670	Presentó el mejor F1 macro entre los modelos iniciales
Gradient Boosting	0,4873	0,4553	Obtuvo el mayor accuracy, pero menor F1 macro que Random Forest

Fuente: Autor

El Dummy Classifier alcanzó un accuracy de 0,4365 y un F1 macro de 0,2026. Si bien su exactitud fue parecida a la de algunos modelos entrenados, el valor bajo del F1 macro muestra que no consiguió clasificar de manera equilibrada las tres categorías. Este comportamiento era previsible, ya que se trata de un modelo de referencia que no aprende patrones entre las variables predictoras, sino que asigna las clases siguiendo una regla básica basada en la categoría más frecuente.

La regresión logística alcanzó un accuracy de 0,4270 y un F1 macro de 0,4280. Aunque su exactitud fue ligeramente menor que la del modelo de referencia, su F1 macro fue considerablemente superior. Esto sugiere que la regresión logística logró distribuir mejor sus predicciones entre las categorías baja, estabilidad y alza, en lugar de concentrarse principalmente en la clase más frecuente.

El modelo Random Forest obtuvo un accuracy de 0,4758 y un F1 macro de 0,4670, convirtiéndose en el modelo con mejor desempeño inicial según la métrica F1 macro. Este resultado sugiere que el algoritmo logró identificar patrones más útiles en las variables temporales, geográficas, comerciales y en los rezagos del precio mayorista. Además, su capacidad para capturar relaciones no lineales pudo contribuir a un comportamiento más equilibrado entre las clases.

Por su parte, Gradient Boosting alcanzó el mayor accuracy, con un valor de 0,4873. Sin embargo, su F1 macro fue de 0,4553, inferior al obtenido por Random Forest. Esto indica que, aunque Gradient Boosting clasificó correctamente una mayor proporción total de observaciones, Random Forest presentó un mejor equilibrio promedio entre las tres categorías de clasificación.

En consecuencia, Random Forest fue seleccionado como el modelo inicial con mejor desempeño para continuar con el proceso de optimización. Esta decisión se basó principalmente en el F1 macro, debido a que el problema analizado corresponde a una clasificación multiclase y se buscó evitar que la evaluación estuviera dominada únicamente por la clase con mayor frecuencia.

4.5. Evaluación del modelo final

Después de la comparación inicial de modelos, se seleccionó Random Forest como el algoritmo con mejor desempeño según la métrica F1 macro. A partir de esta selección, se realizó un proceso de optimización de hiperparámetros con el propósito de mejorar su capacidad de clasificación y obtener una configuración más adecuada para el problema analizado.

La optimización se llevó a cabo utilizando una división interna del conjunto de entrenamiento. Para ello, los registros anteriores al año 2024 fueron separados en un subconjunto de entrenamiento interno y un subconjunto de validación. Esta estrategia permitió seleccionar los mejores hiperparámetros sin utilizar el conjunto de prueba final, el cual se mantuvo reservado para evaluar el desempeño definitivo del modelo.

Los hiperparámetros ajustados incluyeron el número de árboles, la profundidad máxima, el número mínimo de muestras por hoja, el número mínimo de muestras para dividir un nodo, el número de variables consideradas en cada división y la ponderación de clases. Como resultado de este proceso, se obtuvo una configuración optimizada del modelo Random Forest.

La comparación entre el modelo de referencia, el Random Forest base y el Random Forest optimizado se presenta en la Tabla 14.

Tabla 14:

Comparación del desempeño del modelo final frente a la línea base

Modelo	Accuracy	F1 macro	F1 weighted
Modelo ingenuo	0,4365	0,2026	0,2653
Random Forest base	0,4758	0,4670	0,4790
Random Forest optimizado	0,5012	0,4805	0,4958

Fuente: Autor

El Random Forest optimizado fue el modelo con mejor desempeño general dentro de la comparación realizada. Alcanzó un accuracy de 0,5012, un F1 macro de 0,4805 y un F1 weighted de 0,4958. Frente al modelo ingenuo, la mejora más relevante se observó en el F1 macro, lo que refleja una mayor capacidad para clasificar la variación mensual del precio entre las tres categorías definidas.

Al comparar el Random Forest optimizado con el Random Forest base, también se observa una mejora en las métricas principales. El accuracy aumentó de 0,4758 a 0,5012, mientras que el F1 macro pasó de 0,4670 a 0,4805. Aunque la mejora no fue elevada, sí permitió obtener un modelo con mejor desempeño final y mayor equilibrio entre las clases.

El desempeño detallado del modelo optimizado por categoría se presenta en la Tabla 15.

Tabla 15:

Reporte de clasificación del modelo Random Forest optimizado

Categoría	Precisión	Recall	F1-score	Soporte
Baja	0,396	0,411	0,404	107
Estabilidad	0,500	0,646	0,564	189
Alza	0,654	0,372	0,474	137
Promedio macro	0,517	0,476	0,481	433
Promedio ponderado	0,523	0,501	0,496	433

Fuente: Autor

La categoría estabilidad presentó el mejor desempeño del modelo, con un recall de 0,646 y un F1-score de 0,564. Este resultado refleja que el modelo logró reconocer con mayor facilidad los casos en los que el precio se mantuvo dentro del rango de estabilidad definido por el umbral de ±10 %.

En la categoría baja, el modelo alcanzó una precisión de 0,396, un recall de 0,411 y un F1-score de 0,404. Estos valores reflejan que la clasificación de disminuciones relevantes del precio presentó mayor dificultad. De forma similar, en la categoría alza se obtuvo una precisión de 0,654, pero un recall de 0,372, lo que indica que, aunque varias predicciones de alza fueron correctas, el modelo no logró identificar todos los casos reales de aumento relevante del precio.

La matriz de confusión del modelo optimizado se presenta en la Figura 8.

Figura 8:

Matriz de confusión del modelo Random Forest optimizado

Matriz de confusión del Random Forest optimizado:

	Predicho_Baja	Predicho_Estabilidad	Predicho_Alza
Real_Baja	44	57	6
Real_Estabilidad	46	122	21
Real_Alza	21	65	51

Fuente: Autor

La matriz de confusión evidencia que el modelo clasificó correctamente 44 casos de baja, 122 casos de estabilidad y 51 casos de alza. La mayor cantidad de aciertos se concentró en la categoría estabilidad, lo cual coincide con los resultados del reporte de clasificación. Sin embargo, también se observa que una parte importante de los casos reales de baja y alza fueron clasificados como estabilidad.

Este comportamiento sugiere que el modelo tuvo mayor facilidad para identificar períodos de variación moderada del precio, pero presentó mayor dificultad para distinguir correctamente los movimientos más fuertes de disminución o aumento. Esto puede estar relacionado con la naturaleza del problema, la variabilidad del precio agrícola y la disponibilidad limitada de variables externas que expliquen completamente los cambios mensuales del mercado.

En términos generales, el Random Forest optimizado fue el modelo que alcanzó el mejor desempeño final entre los algoritmos evaluados. Sin embargo, las métricas obtenidas muestran que su capacidad de clasificación fue moderada. Por ello, los resultados deben entenderse como un apoyo exploratorio para analizar el comportamiento mensual del precio, y no como una estimación definitiva de la variación mensual del precio.

4.6. Importancia de variables

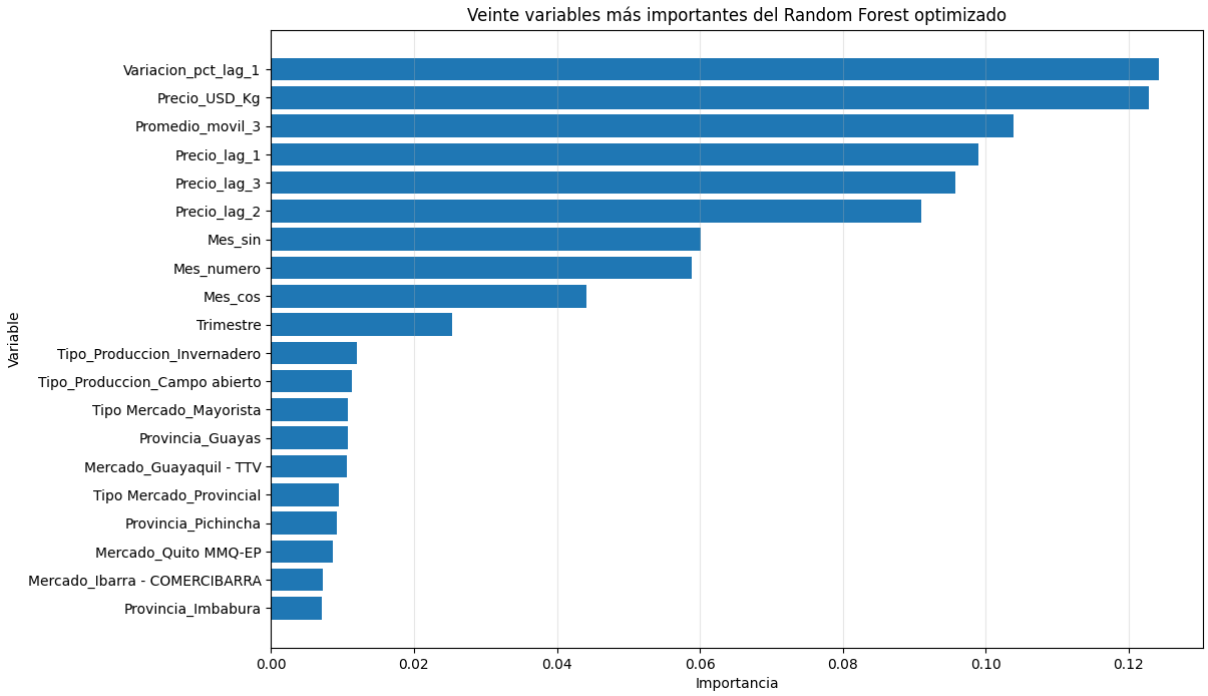
Después de evaluar el modelo final, se analizó la importancia de las variables utilizadas por el Random Forest optimizado. Este análisis permitió identificar qué características tuvieron mayor contribución en el proceso de clasificación de la variación mensual del precio del tomate riñón.

La importancia de variables resulta útil para interpretar el comportamiento del modelo, ya que permite reconocer cuáles fueron los atributos que más aportaron a la separación de las categorías baja, estabilidad y alza. En este caso, las variables con mayor contribución estuvieron

relacionadas principalmente con el comportamiento histórico del precio, los rezagos mensuales y las características temporales del registro.

La Figura 9 presenta las variables con mayor importancia dentro del modelo Random Forest optimizado.

Figura 9:
Importancia de variables del modelo Random Forest optimizado



Fuente: Autor

La variable Variacion_pct_lag_1 se ubicó entre las más importantes del modelo, lo que muestra que el cambio porcentual observado en el mes anterior aportó información útil para clasificar la variación del precio en el mes siguiente. Este comportamiento resulta coherente con el enfoque temporal del estudio, ya que los movimientos recientes del precio pueden influir en la tendencia posterior o guardar relación con ella.

También se identificó una contribución importante de variables asociadas al precio actual y a sus valores rezagados, como Precio_USD_Kg, Precio_lag_1, Precio_lag_2, Precio_lag_3 y Promedio_movil_3. Estas variables permitieron al modelo incorporar información sobre el nivel actual del precio, su comportamiento reciente y la tendencia de corto plazo dentro de cada serie temporal.

Además, las variables temporales, como Mes_numero, Mes_sin, Mes_cos y Trimestre, aportaron información relacionada con posibles patrones estacionales. Esto resulta relevante porque el análisis exploratorio mostró que el precio del tomate riñón no presentó el mismo comportamiento durante todos los meses del año, sino que existieron diferencias estacionales en determinados períodos.

Las variables geográficas y comerciales, como Provincia, Mercado, Tipo Mercado y Tipo_Produccion, también contribuyeron al proceso de clasificación. Esto sugiere que el comportamiento del precio no fue completamente homogéneo entre mercados, provincias y tipos de producción. En particular, la diferenciación entre tomate riñón de invernadero y tomate riñón a campo abierto permitió representar diferencias productivas que también se observaron durante el análisis exploratorio.

Para resumir la interpretación de las principales variables utilizadas por el modelo, en la Tabla 16 se presenta una agrupación de las variables con mayor contribución.

Tabla 16:

Principales grupos de variables relevantes en el modelo final

Grupo de variables	Variables destacadas	Interpretación
Variación histórica	Variacion_pct_lag_1	Representa el cambio porcentual reciente del precio y aportó información sobre movimientos previos
Precio actual y rezagos	Precio_USD_Kg, Precio_lag_1, Precio_lag_2, Precio_lag_3	Permiten representar el nivel actual del precio y su comportamiento en meses anteriores
Tendencia de corto plazo	Promedio_movil_3	Resume el comportamiento promedio del precio durante los tres meses anteriores
Variables temporales	Mes_numero, Mes_sin, Mes_cos, Trimestre	Capturan posibles patrones estacionales asociados con los meses del año
Variables geográficas y comerciales	Provincia, Mercado, Tipo Mercado, Tipo_Produccion	Representan diferencias territoriales, comerciales y productivas; en el modelo fueron transformadas mediante One-Hot Encoding

Fuente: Autor

El análisis de importancia de variables muestra que el modelo se basó principalmente en información histórica del precio, junto con características temporales y geográficas. Esto sugiere que el modelo se apoyó principalmente en el comportamiento reciente de cada serie y en las características propias de los mercados analizados para realizar la clasificación.

No obstante, la importancia de variables debe interpretarse con cautela, ya que indica la contribución relativa de cada característica dentro del modelo, pero no necesariamente representa una relación causal directa. Por esta razón, los resultados deben entenderse como una herramienta de interpretación del modelo y no como una explicación definitiva de los factores que determinan las variaciones del precio.

4.7. Análisis de errores del modelo

Finalmente, se realizó un análisis de errores del modelo Random Forest optimizado, con el propósito de comprender con mayor detalle su comportamiento durante la clasificación de la variación mensual del precio del tomate riñón. Este análisis permitió identificar en qué categorías el modelo presentó mayor cantidad de aciertos y en cuáles tuvo mayores dificultades.

En el conjunto de prueba, conformado por 433 observaciones correspondientes al período 2024-2026, el modelo clasificó correctamente 217 registros y presentó errores en 216 observaciones. Estos resultados son coherentes con el accuracy final de 0,5012, equivalente al 50,12 % de clasificaciones correctas.

Los resultados generales del análisis de aciertos y errores se presentan en la Tabla 17.

Tabla 17:

Resumen general de aciertos y errores del modelo final

Resultado	Número de registros	Porcentaje
Aciertos	217	50,12 %
Errores	216	49,88 %
Total	433	100,00 %

Fuente: Autor

El modelo presentó un desempeño moderado, con una cantidad de aciertos y errores bastante similar. Aunque el Random Forest optimizado fue el algoritmo con mejores resultados dentro de los modelos evaluados, estos valores muestran que clasificar la variación mensual del precio continúa siendo una tarea compleja.

El análisis por categoría permitió identificar diferencias importantes en el desempeño del modelo. La clase estabilidad fue la que presentó el mayor porcentaje de acierto, mientras que las categorías baja y alza mostraron mayores dificultades de clasificación. Estos resultados se presentan en la Tabla 18.

Tabla 18:

Porcentaje de acierto del modelo final por categoría

Categoría	Porcentaje de acierto
Baja	41,12 %
Estabilidad	64,55 %
Alza	37,23 %

Fuente: Autor

La categoría estabilidad registró el mayor porcentaje de acierto, con 64,55 %. Este resultado muestra que el modelo tuvo mayor facilidad para reconocer los casos en los que el precio se mantuvo dentro del rango de variación moderada establecido por el umbral de ±10 %. Esto también puede explicarse porque estabilidad fue la clase con mayor cantidad de observaciones dentro de la variable objetivo.

En cambio, las categorías baja y alza presentaron menores porcentajes de acierto, con 41,12 % y 37,23 %, respectivamente. Esto evidencia que el modelo tuvo más dificultad para reconocer disminuciones o aumentos relevantes del precio. Una posible explicación es que las variaciones fuertes pueden depender de factores externos no incluidos completamente en la base de datos, como cambios repentinos en la oferta, condiciones logísticas, eventos climáticos específicos, decisiones comerciales o variaciones locales de mercado.

También se revisó el desempeño del modelo según el mes del año. Este análisis permitió observar que el modelo no tuvo el mismo comportamiento en todos los meses, sino que presentó diferencias en su porcentaje de acierto mensual. Los resultados se presentan en la Tabla 19.

Tabla 19:

Porcentaje de acierto del modelo final por mes

Mes número	Mes	Registros	Aciertos	Porcentaje de acierto
1	Enero	46	22	47,83 %
2	Febrero	45	31	68,89 %
3	Marzo	46	22	47,83 %
4	Abril	30	7	23,33 %
5	Mayo	31	16	51,61 %
6	Junio	32	23	71,88 %
7	Julio	32	17	53,12 %
8	Agosto	33	18	54,55 %
9	Septiembre	34	16	47,06 %
10	Octubre	36	12	33,33 %
11	Noviembre	35	18	51,43 %
12	Diciembre	33	15	45,45 %

Fuente: Autor

Al revisar el desempeño mensual, junio fue el mes con mayor porcentaje de acierto, con 71,88 %, seguido de febrero, con 68,89 %. También se obtuvieron resultados superiores al 50 % en agosto, julio, mayo y noviembre. Esto muestra que, en esos períodos, el modelo logró identificar con mayor precisión la variación mensual del precio del tomate riñón.

En cambio, abril presentó el menor porcentaje de acierto, con 23,33 %, seguido de octubre, con 33,33 %. Estos meses representaron los períodos de mayor dificultad para el modelo. Por su parte, diciembre, septiembre, enero y marzo presentaron porcentajes de acierto intermedios, cercanos al 45 % y 48 %.

En conjunto, el análisis mensual evidencia que el desempeño del modelo varió según el mes evaluado. Esto puede estar relacionado con diferencias estacionales, cambios en la dinámica de oferta y demanda, o comportamientos particulares del precio en determinados períodos del año. Por esta razón, los resultados mensuales deben interpretarse como una referencia exploratoria del comportamiento del modelo y no como una regla fija para todos los períodos.

En consecuencia, los errores observados reflejan la complejidad del comportamiento del precio del tomate riñón en los mercados mayoristas del Ecuador. La incorporación de variables adicionales con mayor cobertura temporal y geográfica podría contribuir, en futuras

investigaciones, a mejorar la capacidad del modelo para diferenciar con mayor precisión las categorías baja, estabilidad y alza.

5. CAPÍTULO V: CONCLUSIONES Y RECOMENDACIONES

5.1.Conclusiones

Los resultados obtenidos en esta investigación permiten concluir lo siguiente:

- La investigación permitió analizar el comportamiento histórico del precio del tomate riñón en los mercados mayoristas del Ecuador, utilizando datos públicos del SIPA. A partir del procesamiento de la información se construyó una base mensual consolidada por fecha, provincia, mercado y tipo de producción, lo que permitió identificar variaciones temporales, estacionales, geográficas y productivas en el precio del producto. Sin embargo, no todas las series presentaron información completa durante todo el período de estudio, por lo que se trabajó únicamente con transiciones mensuales consecutivas válidas.
- El análisis exploratorio evidenció que el precio del tomate riñón no presentó un comportamiento uniforme durante el período estudiado. Se observaron diferencias entre el tomate riñón de invernadero y el tomate riñón a campo abierto, así como variaciones mensuales, con precios medianos más altos en marzo, abril, mayo y junio, y valores relativamente menores en enero, noviembre y diciembre.
- La construcción de la variable objetivo permitió abordar el estudio como un problema de clasificación supervisada multiclase. Mediante el umbral de ±10 %, la variación mensual del precio fue clasificada en tres categorías: baja, estabilidad y alza. Esta clasificación permitió representar la dirección del cambio mensual del precio de una forma más comprensible que la estimación de un valor monetario exacto.
- La comparación de escenarios mostró que la incorporación de variables complementarias no necesariamente mejoró el desempeño de los modelos. Las variables relacionadas con precios al productor, insumos agrícolas, producción, rendimiento y clima presentaron diferencias de cobertura temporal y geográfica, lo que redujo la cantidad de registros completos disponibles para el modelado.
- Entre los modelos evaluados, Random Forest presentó el mejor desempeño general. Luego de la optimización de hiperparámetros, el modelo alcanzó un accuracy de 0,5012, un F1 macro de 0,4805 y un F1 weighted de 0,4958. Estos resultados superaron al

modelo de referencia, aunque evidencian que la capacidad de clasificación fue moderada.

- El análisis del modelo final evidenció que las variables con mayor aporte estuvieron asociadas principalmente al comportamiento previo del precio. Entre ellas destacaron la variación porcentual del mes anterior, el precio actual, los rezagos de precio y el promedio móvil de tres meses. Además, las variables temporales, geográficas y comerciales también contribuyeron al proceso de clasificación, lo que sugiere que el modelo utilizó principalmente información relacionada con la evolución reciente de cada serie y con las características propias de los mercados analizados.
- En general, los modelos de Machine Learning permitieron clasificar la variación mensual del precio del tomate riñón con un desempeño superior al modelo base, pero no alcanzaron una capacidad predictiva alta. Por ello, los resultados deben interpretarse como una herramienta de apoyo para el análisis exploratorio del comportamiento del precio, y no como una predicción definitiva del mercado futuro.

5.2.Recomendaciones

A partir de las conclusiones alcanzadas, se plantean las siguientes recomendaciones:

- Se recomienda que futuras investigaciones incorporen variables adicionales relacionadas con la dinámica comercial y productiva del tomate riñón, como volumen comercializado, cantidad ofertada por mercado, costos de transporte, disponibilidad del producto, demanda local y precios diarios o semanales, siempre que esta información se encuentre disponible.
- También se recomienda fortalecer la integración de datos complementarios con mayor cobertura temporal y geográfica. Esto permitiría evaluar con mayor precisión el aporte de variables como clima, producción, rendimiento, insumos agrícolas y precios al productor, sin reducir de forma considerable la cantidad de registros disponibles para el modelado.
- Se sugiere evaluar diferentes formas de construir la variable objetivo. Aunque en esta investigación se utilizó un umbral de ±10 %, futuros estudios podrían probar otros criterios de clasificación según la volatilidad del producto, las características de cada mercado o la magnitud histórica de las variaciones observadas.

- Finalmente, se recomienda probar otros algoritmos y enfoques de modelado que puedan capturar mejor patrones temporales, como XGBoost, LightGBM, CatBoost, redes neuronales o modelos específicos para series de tiempo. No obstante, los resultados de estos modelos deben seguir interpretándose como herramientas de apoyo al análisis del mercado y no como predicciones definitivas.

6. REFERENCIAS

Aker, J. (2011). Dial “A” for Agriculture: A Review of Information and Communication Technologies for Agricultural Extension in Developing Countries - Working Paper 269. *Agricultural Economics*, *42*, 631-647. https://doi.org/10.1111/j.1574-0862.2011.00545.x

Amazon Web Services. (2025, noviembre 19). *Data preprocessing—Machine Learning Lens*. https://docs.aws.amazon.com/wellarchitected/latest/machine-learning-lens/data-preprocessing.html

Asencio Beltrán, L. S. (2022). *Diseño de un modelo predictivo de aprendizaje automático supervisado para determinar precios dinámicos de insumos Agrícolas de la región costa del Ecuador*. http://repositorio.ug.edu.ec/handle/redug/63093

AWS. (2026). *¿Qué es el machine learning? - Explicación de la tecnología ML - AWS*. Amazon Web Services, Inc. https://aws.amazon.com/es/what-is/machine-learning/

Belcic, I., & Stryker, C. (2025, septiembre 12). ¿Qué es el aprendizaje supervisado? IBM. https://www.ibm.com/es-es/think/topics/supervised-learning

Branco, P., Torgo, L., & Ribeiro, R. P. (2016). A Survey of Predictive Modeling on Imbalanced Domains. *ACM Computing Surveys (CSUR)*, *49*(2), 31:1-31:50. https://doi.org/10.1145/2907070

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999). CRISP-DM 1.0 step-by-step data mining guide. En *Springer*.

FAO. (2022). *Digital Agriculture and AI Innovation | Innovation at FAO | FAO*. OpenInnovationNetwork. https://www.fao.org/innovation/home/digital-agriculture-and-ai-innovation/en

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, *29*(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

García, S., Luengo, J., & Herrera, F. (2015). *Data Preprocessing in Data Mining* (Vol. 72). Springer International Publishing. https://doi.org/10.1007/978-3-319-10247-4

Genuer, R., Poggi, J.-M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern Recognition Letters*, *31*(14), 2225-2236. https://doi.org/10.1016/j.patrec.2010.03.014

Gilbert, C. L., Christiaensen, L., & Kaminski, J. (2017). Food price seasonality in Africa: Measurement and extent. *Food Policy*, *67*(C), 119-132.

Gilbert, C. L., & Morgan, C. W. (2010). Food price volatility. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *365*(1554), 3023-3034. https://doi.org/10.1098/rstb.2010.0139

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. Springer. https://doi.org/10.1007/978-0-387-84858-7

Hernández Sampieri, R., & Mendoza Torres, C. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta*. https://bibliotecadigital.uce.edu.ec/s/L-D/item/793

IBM. (2021, agosto 17). *IBM SPSS Modeler Subscription*. https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). Statistical Learning. En G. James, D. Witten, T. Hastie, & R. Tibshirani (Eds.), *An Introduction to Statistical Learning: With Applications in R* (pp. 15-57). Springer US. https://doi.org/10.1007/978-1-0716-1418-1_2

Kavlakoglu, E. (2021, octubre 20). *What Is Random Forest? | IBM*. https://www.ibm.com/think/topics/random-forest

Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling*. Springer. https://doi.org/10.1007/978-1-4614-6849-3

Ley Orgánica de Tierras Rurales y Territorios Ancestrales. (2016, marzo 14). *Asambleanacional.gob.ec/es/multimedios-legislativos/37677-ley-organica-de-tierras-rurales-y*. https://www.asambleanacional.gob.ec/es/multimedios-legislativos/37677-ley-organica-de-tierras-rurales-y

Ministerio de Agricultura y Ganadería. (s. f.). *¿Qué es el SIPA?* Sistema de Información Pública Agropecuaria [MAG]. Recuperado 9 de febrero de 2026, de https://sipa.agricultura.gob.ec/index.php/acerca-del-sipa/sistema-informacion-publica-agropecuaria

Peñafiel, J. G. S., Quilsimba, V. M. O., & Carlozama, A. N. O. (2025). Desarrollo de un Modelo para la Predicción de Precios de Productos Agrícolas de la Provincia de Carchi, Mediante Técnicas de Aprendizaje Automático. *Tierra Infinita*, *11*(1), 71-89. https://doi.org/10.32645/26028131.1414

Scikit-Learn Developers. (2026). *1.12. Multiclass and multioutput algorithms*. Scikit-Learn. Recuperado 18 de julio de 2026, de https://scikit-learn.org/stable/modules/multiclass.html

Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427-437. https://doi.org/10.1016/j.ipm.2009.03.002

Wilson, G. T. (2016). Time Series Analysis: Forecasting and Control, 5th Edition , by George E. P. Box , Gwilym M. Jenkins , Gregory C. Reinsel and Greta M. Ljung , 2015 . Published by John Wiley and Sons Inc. , Hoboken, New Jersey , pp. 712. ISBN: 978-1-118-67502-1. *Journal of Time Series Analysis*, *37*(5), 709-711.

Zewe, A. (2025, julio 30). *New algorithms enable efficient machine learning with symmetric data*. MIT News | Massachusetts Institute of Technology. https://news.mit.edu/2025/new-algorithms-enable-efficient-machine-learning-with-symmetric-data-0730

7. ANEXOS

7.1. ANEXO 1. Código principal utilizado para el procesamiento y modelado

Carga del dataset y conversión de variables numéricas:

```
# ========================================
# PASO 1: CARGA Y REVISIÓN INICIAL DEL DATASET
# ========================================

# Importar librerías necesarias
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import io

from google.colab import files
from IPython.display import display

# Subir el archivo CSV desde tu computador
uploaded = files.upload()

# Obtener automáticamente el nombre del archivo cargado
nombre_archivo = next(iter(uploaded))

# Leer el archivo CSV
df_mercados = pd.read_csv(
    io.BytesIO(uploaded[nombre_archivo]),
    sep=";",
    encoding="latin1"
)

# Estandarizar los nombres de las columnas
df_mercados.columns = (
    df_mercados.columns
    .str.strip()
    .str.replace(r"\s+", " ", regex=True)
)

# Mostrar dimensiones del dataset
print("Número de filas y columnas:", df_mercados.shape)

# Mostrar nombres de las columnas
```

```
print("\nColumnas disponibles:")
print(df_mercados.columns.tolist())

# Mostrar las primeras cinco filas
print("\nPrimeras cinco filas:")
display(df_mercados.head())

# Mostrar tipos de datos y valores no nulos
print("\nInformación general del dataset:")
df_mercados.info()
# ========================================
# PASO 2: CONVERSIÓN DE VARIABLES NUMÉRICAS
# ========================================

# Crear una copia de seguridad del dataset original
df = df_mercados.copy()

# Definir las columnas que deben ser numéricas
columnas_numericas = [
    "Cant.",
    "Promedio de Precio (USD)",
    "Precio Promedio (USD/Kg)"
]

# Convertir coma decimal a punto y transformar a número
for columna in columnas_numericas:
    df[columna] = (
        df[columna]
        .astype(str)
        .str.strip()
        .str.replace(",", ".", regex=False)
    )

    df[columna] = pd.to_numeric(df[columna], errors="coerce")

# Verificar los tipos de datos después de la conversión
print("Tipos de datos después de la conversión:")
print(df.dtypes)

# Revisar valores nulos generados durante la conversión
print("\nValores nulos por columna:")
print(df[columnas_numericas].isnull().sum())

# Mostrar estadísticas descriptivas de las variables numéricas
```

```
print("\nEstadísticas descriptivas:")
display(df[columnas_numericas].describe())

# Mostrar cinco registros para verificar el resultado
print("\nPrimeros cinco registros después de la conversión:")
display(df.head())
```

Filtrado del tomate riñón y creación del tipo de producción:

```
# ========================================
# PASO 3: IDENTIFICACIÓN DEL TOMATE RIÑÓN
# ========================================

# Crear una versión estandarizada de la columna Producto
df["Producto_normalizado"] = (
    df["Producto"]
    .astype(str)
    .str.strip()
    .str.upper()
    .str.normalize("NFKD")
    .str.encode("ascii", errors="ignore")
    .str.decode("utf-8")
)

# Buscar todos los productos que contienen la palabra TOMATE
productos_tomate = (
    df.loc[
        df["Producto_normalizado"].str.contains("TOMATE", na=False),
        "Producto"
    ]
    .value_counts()
    .reset_index()
)

# Cambiar nombres de columnas para que sean más claros
productos_tomate.columns = ["Producto registrado", "Número de
registros"]

# Mostrar las variedades de tomate encontradas
print("Variedades de tomate encontradas en el dataset:")
display(productos_tomate)

# Filtrar únicamente los registros de tomate riñón
df_tomate = df[
```

```
    df["Producto_normalizado"].str.contains("TOMATE RINON", na=False)
].copy()

# Revisar tamaño del nuevo dataset
print(f"Número de registros de tomate riñón: {df_tomate.shape[0]}")
print(f"Número de columnas: {df_tomate.shape[1]}")

# Mostrar las variedades incluidas en el filtro
print("\nVariedades de tomate riñón incluidas:")
display(df_tomate["Producto"].value_counts().reset_index())

# Mostrar los primeros registros del dataset filtrado
print("\nPrimeros registros de tomate riñón:")
display(df_tomate.head())
# ========================================
# PASO 4: ESTANDARIZACIÓN Y VARIABLE FECHA
# ========================================

# Crear una variable uniforme para el tipo de producción
df_tomate["Tipo_Produccion"] = np.where(
    df_tomate["Producto_normalizado"].str.contains("INVERNADERO",
na=False),
    "Invernadero",
    "Campo abierto"
)

# Crear diccionario para transformar los meses de español a número
meses_numero = {
    "Enero": 1,
    "Febrero": 2,
    "Marzo": 3,
    "Abril": 4,
    "Mayo": 5,
    "Junio": 6,
    "Julio": 7,
    "Agosto": 8,
    "Septiembre": 9,
    "Octubre": 10,
    "Noviembre": 11,
    "Diciembre": 12
}

# Crear columna numérica del mes
df_tomate["Mes_numero"] = df_tomate["Mes"].map(meses_numero)
```

```
# Verificar que todos los meses fueron reconocidos
print("Meses no reconocidos:")
print(df_tomate.loc[df_tomate["Mes_numero"].isna(), "Mes"].unique())

# Crear una fecha mensual usando año, mes y día 1
df_tomate["Fecha"] = pd.to_datetime(
    {
        "year": df_tomate["Año"],
        "month": df_tomate["Mes_numero"],
        "day": 1
    }
)

# Revisar la distribución por tipo de producción
print("\nRegistros por tipo de producción:")
display(
    df_tomate["Tipo_Produccion"]
    .value_counts()
    .rename_axis("Tipo de producción")
    .reset_index(name="Número de registros")
)

# Mostrar cobertura temporal
print("\nCobertura temporal:")
print("Fecha mínima:", df_tomate["Fecha"].min().strftime("%Y-%m-%d"))
print("Fecha máxima:", df_tomate["Fecha"].max().strftime("%Y-%m-%d"))

# Revisar cobertura geográfica
print("\nCobertura geográfica:")
print("Número de mercados:", df_tomate["Mercado"].nunique())
print("Número de provincias:", df_tomate["Provincia"].nunique())
print("Número de cantones:", df_tomate["Cantón"].nunique())

# Mostrar las primeras filas con las nuevas columnas
display(
    df_tomate[
        [
            "Fecha", "Año", "Mes", "Provincia", "Mercado",
            "Producto", "Tipo_Produccion",
            "Precio Promedio (USD/Kg)"
        ]
    ].head(10)
)
```

Consolidación mensual del precio:

```
# ========================================
# PASO 6: CONSOLIDACIÓN A NIVEL MERCADO-MES
# ========================================

# Definir la unidad final de análisis
claves_modelo = [
    "Fecha",
    "Año",
    "Mes",
    "Mes_numero",
    "Tipo Mercado",
    "Mercado",
    "Provincia",
    "Cantón",
    "Tipo_Produccion"
]

# Consolidar los registros de distintas presentaciones
df_tomate_mensual = (
    df_tomate
    .groupby(claves_modelo, as_index=False)
    .agg(
        Precio_USD_Kg=("Precio Promedio (USD/Kg)", "median"),
        Precio_USD_Kg_promedio=("Precio Promedio (USD/Kg)", "mean"),
        Precio_USD_Kg_min=("Precio Promedio (USD/Kg)", "min"),
        Precio_USD_Kg_max=("Precio Promedio (USD/Kg)", "max"),
        Registros_presentacion=("Precio Promedio (USD/Kg)", "size"),
        Numero_presentaciones=("Pres.", "nunique")
    )
)

# Calcular la diferencia entre el precio máximo y mínimo
# dentro de una misma unidad mercado-mes-tipo de producción
df_tomate_mensual["Rango_precio_USD_Kg"] = (
    df_tomate_mensual["Precio_USD_Kg_max"]
    - df_tomate_mensual["Precio_USD_Kg_min"]
)

# Ordenar cronológicamente
df_tomate_mensual = df_tomate_mensual.sort_values(
    ["Mercado", "Tipo_Produccion", "Fecha"]
).reset_index(drop=True)
```

```
# Mostrar dimensiones antes y después de consolidar
print("Registros antes de consolidar:", df_tomate.shape[0])
print("Registros después de consolidar:", df_tomate_mensual.shape[0])

# Revisar cuántas unidades tuvieron más de una presentación
print("\nUnidades con más de una presentación:")
print(
    df_tomate_mensual["Registros_presentacion"]
    .value_counts()
    .sort_index()
)

# Estadísticas del rango de precios entre presentaciones
print("\nEstadísticas del rango de precio dentro del mismo mercado-
mes:")
display(df_tomate_mensual["Rango_precio_USD_Kg"].describe())

# Mostrar ejemplos donde la diferencia entre presentaciones fue mayor
print("\nEjemplos con mayor diferencia de precio entre presentaciones:")
display(
    df_tomate_mensual
    .sort_values("Rango_precio_USD_Kg", ascending=False)
    [
        [
            "Fecha", "Provincia", "Mercado", "Tipo_Produccion",
            "Precio_USD_Kg_min", "Precio_USD_Kg",
            "Precio_USD_Kg_max", "Rango_precio_USD_Kg",
            "Registros_presentacion", "Numero_presentaciones"
        ]
    ]
    .head(10)
)

# Mostrar las primeras filas de la nueva base mensual
print("\nBase consolidada:")
display(df_tomate_mensual.head())
```

Estandarización de nombres y reconstrucción de la base limpia:

```
# ========================================
# PASO 13: DETECCIÓN DE INCONSISTENCIAS
# EN VARIABLES CATEGÓRICAS
# ========================================
```

```
import re
import unicodedata

# Crear función para normalizar textos
def normalizar_texto(valor):

    # Convertir a texto y eliminar espacios extremos
    texto = str(valor).strip()

    # Eliminar tildes y caracteres especiales
    texto = unicodedata.normalize("NFKD", texto)
    texto = "".join(
        caracter
        for caracter in texto
        if not unicodedata.combining(caracter)
    )

    # Reemplazar espacios repetidos por un solo espacio
    texto = re.sub(r"\s+", " ", texto)

    # Convertir todo a mayúsculas
    texto = texto.upper()

    return texto


# Crear una copia para revisar categorías
df_revision = df_tomate_mensual.copy()

# Columnas categóricas que debemos revisar
columnas_categoricas = [
    "Provincia",
    "Mercado",
    "Cantón",
    "Tipo Mercado"
]

# Crear columnas normalizadas
for columna in columnas_categoricas:
    df_revision[columna + "_normalizado"] = (
        df_revision[columna]
        .apply(normalizar_texto)
    )
```

```
# Revisar diferencias en provincias, mercados, cantones y tipo de
mercado
for columna in columnas_categoricas:

    columna_normalizada = columna + "_normalizado"

    reporte = (
        df_revision
        .groupby(columna_normalizada)
        .agg(
            Variantes_originales=(
                columna,
                lambda x: x.nunique()
            ),
            Nombres_originales=(
                columna,
                lambda x: " | ".join(
                    sorted(x.dropna().astype(str).unique())
                )
            ),
            Registros=(columna, "size")
        )
        .reset_index()
    )

    inconsistencias = reporte[
        reporte["Variantes_originales"] > 1
    ].sort_values(
        "Registros",
        ascending=False
    )

    print("\n" + "=" * 70)
    print(f"Inconsistencias detectadas en: {columna}")
    print("=" * 70)

    if inconsistencias.shape[0] == 0:
        print("No se detectaron variantes de escritura.")
    else:
        display(inconsistencias)

# Mostrar todas las versiones originales de mercado
print("\nListado de mercados registrados:")
display(
```

```
    df_revision["Mercado"]
    .value_counts()
    .reset_index()
    .rename(
        columns={
            "Mercado": "Número de registros",
            "index": "Mercado"
        }
    )
)
# ========================================
# PASO 14: ESTANDARIZACIÓN DEFINITIVA DE
# PROVINCIAS, MERCADOS Y CANTONES
# ========================================

import re
import unicodedata

# Función para crear una clave comparable:
# elimina tildes, espacios repetidos y diferencias de mayúsculas
def normalizar_texto(valor):

    texto = str(valor).strip()

    texto = unicodedata.normalize("NFKD", texto)
    texto = "".join(
        caracter
        for caracter in texto
        if not unicodedata.combining(caracter)
    )

    texto = re.sub(r"\s+", " ", texto)
    texto = texto.upper()

    return texto


# Crear copia del dataset de tomate antes de modificarlo
df_tomate_limpio = df_tomate.copy()

# Guardar número de categorías antes de la limpieza
conteo_antes = {
    "Provincias": df_tomate_limpio["Provincia"].nunique(),
    "Mercados": df_tomate_limpio["Mercado"].nunique(),
```

```
    "Cantones": df_tomate_limpio["Cantón"].nunique()
}

# Diccionario de nombres oficiales para provincias
mapa_provincias = {
    "TUNGURAHUA": "Tungurahua"
}

# Diccionario de nombres oficiales para mercados
mapa_mercados = {
    "QUITO MMQ-EP": "Quito MMQ-EP",
    "GUAYAQUIL - TTV": "Guayaquil - TTV",
    "IBARRA - COMERCIBARRA": "Ibarra - COMERCIBARRA",
    "CUENCA - EL ARENAL": "Cuenca - EL ARENAL",
    "RIOBAMBA - EP-EMMPA": "Riobamba - EP-EMMPA",
    "AMBATO EP-EMA": "Ambato EP-EMA",
    "LA LIBERTAD - ASOPROCOMPRA": "La Libertad - Asoprocompra",
    "MILAGRO - MERCADO DE TRANSFERENCIA": "Milagro - Mercado de
Transferencia",
    "BABAHOYO - 4 DE MAYO": "Babahoyo - 4 de Mayo",
    "GUARANDA - 24 DE MAYO": "Guaranda - 24 de Mayo"
}

# Diccionario de nombres oficiales para cantones
mapa_cantones = {
    "AMBATO": "Ambato"
}

# Estandarizar provincia
clave_provincia = df_tomate_limpio["Provincia"].apply(normalizar_texto)

df_tomate_limpio["Provincia"] = (
    clave_provincia
    .map(mapa_provincias)
    .fillna(df_tomate_limpio["Provincia"].astype(str).str.strip())
)

# Estandarizar mercado
clave_mercado = df_tomate_limpio["Mercado"].apply(normalizar_texto)

df_tomate_limpio["Mercado"] = (
    clave_mercado
    .map(mapa_mercados)
    .fillna(df_tomate_limpio["Mercado"].astype(str).str.strip())
```

```
)

# Estandarizar cantón
clave_canton = df_tomate_limpio["Cantón"].apply(normalizar_texto)

df_tomate_limpio["Cantón"] = (
    clave_canton
    .map(mapa_cantones)
    .fillna(df_tomate_limpio["Cantón"].astype(str).str.strip())
)

# Revisar número de categorías después de la limpieza
conteo_despues = {
    "Provincias": df_tomate_limpio["Provincia"].nunique(),
    "Mercados": df_tomate_limpio["Mercado"].nunique(),
    "Cantones": df_tomate_limpio["Cantón"].nunique()
}

print("Categorías antes y después de la estandarización:")

comparacion_categorias = pd.DataFrame(
    {
        "Antes": conteo_antes,
        "Después": conteo_despues
    }
)

display(comparacion_categorias)

# Definir nuevamente la unidad de análisis mensual
claves_modelo = [
    "Fecha",
    "Año",
    "Mes",
    "Mes_numero",
    "Tipo Mercado",
    "Mercado",
    "Provincia",
    "Cantón",
    "Tipo_Produccion"
]

# Reconstruir la base mensual con nombres ya corregidos
df_tomate_mensual_limpio = (
```

```
    df_tomate_limpio
    .groupby(claves_modelo, as_index=False)
    .agg(
        Precio_USD_Kg=("Precio Promedio (USD/Kg)", "median"),
        Precio_USD_Kg_promedio=("Precio Promedio (USD/Kg)", "mean"),
        Precio_USD_Kg_min=("Precio Promedio (USD/Kg)", "min"),
        Precio_USD_Kg_max=("Precio Promedio (USD/Kg)", "max"),
        Registros_presentacion=("Precio Promedio (USD/Kg)", "size"),
        Numero_presentaciones=("Pres.", "nunique")
    )
)

# Calcular rango de precios entre presentaciones
df_tomate_mensual_limpio["Rango_precio_USD_Kg"] = (
    df_tomate_mensual_limpio["Precio_USD_Kg_max"]
    - df_tomate_mensual_limpio["Precio_USD_Kg_min"]
)

# Ordenar las series cronológicamente
df_tomate_mensual_limpio = df_tomate_mensual_limpio.sort_values(
    ["Provincia", "Mercado", "Tipo_Produccion", "Fecha"]
).reset_index(drop=True)

# Comparar tamaño de la base mensual antes y después
print("Registros mensuales antes de estandarizar:",
df_tomate_mensual.shape[0])
print("Registros mensuales después de estandarizar:",
df_tomate_mensual_limpio.shape[0])

# Confirmar mercados y provincias finales
print("\nMercados finales:")
display(
    df_tomate_mensual_limpio["Mercado"]
    .value_counts()
    .rename_axis("Mercado")
    .reset_index(name="Número de registros")
)

print("\nProvincias finales:")
display(
    df_tomate_mensual_limpio["Provincia"]
    .value_counts()
    .rename_axis("Provincia")
    .reset_index(name="Número de registros")
```

```
)

# Desde ahora, esta será la base usada para el EDA y el modelo
df_eda = df_tomate_mensual_limpio.copy()
```

Construcción de la variable objetivo:

```
# ========================================
# PASO 15: RECONSTRUIR TRANSICIONES Y CLASES
# USANDO LA BASE LIMPIA
# ========================================

# Crear una copia de la base limpia
df_transiciones_limpio = df_eda.copy()

# Definir cada serie temporal
claves_serie = [
    "Provincia",
    "Mercado",
    "Tipo_Produccion"
]

# Ordenar cada serie por fecha
df_transiciones_limpio = df_transiciones_limpio.sort_values(
    claves_serie + ["Fecha"]
).reset_index(drop=True)

# Obtener fecha y precio del registro siguiente
df_transiciones_limpio["Fecha_siguiente"] = (
    df_transiciones_limpio
    .groupby(claves_serie)["Fecha"]
    .shift(-1)
)

df_transiciones_limpio["Precio_mes_siguiente"] = (
    df_transiciones_limpio
    .groupby(claves_serie)["Precio_USD_Kg"]
    .shift(-1)
)

# Calcular cuál debería ser el siguiente mes
df_transiciones_limpio["Fecha_esperada_siguiente"] = (
    df_transiciones_limpio["Fecha"] + pd.offsets.MonthBegin(1)
)

# Identificar transiciones consecutivas
```

```
df_transiciones_limpio["Transicion_consecutiva"] = (
    df_transiciones_limpio["Fecha_siguiente"]
    == df_transiciones_limpio["Fecha_esperada_siguiente"]
)

# Calcular variación porcentual del precio
df_transiciones_limpio["Variacion_pct_siguiente"] = (
    (
        df_transiciones_limpio["Precio_mes_siguiente"]
        - df_transiciones_limpio["Precio_USD_Kg"]
    )
    / df_transiciones_limpio["Precio_USD_Kg"]
) * 100

# Conservar solo transiciones válidas
df_transiciones_validas_limpio = df_transiciones_limpio[
    df_transiciones_limpio["Transicion_consecutiva"]
].copy()

# Definir umbral provisional de clasificación
umbral_seleccionado = 10

# Crear clases: Baja, Estabilidad y Alza
df_transiciones_validas_limpio["Clase_variacion"] = np.select(
    [
        df_transiciones_validas_limpio["Variacion_pct_siguiente"] <= -
umbral_seleccionado,
        df_transiciones_validas_limpio["Variacion_pct_siguiente"] >=
umbral_seleccionado
    ],
    [
        "Baja",
        "Alza"
    ],
    default="Estabilidad"
)

# Mostrar resultados
print("Observaciones mensuales limpias:",
df_transiciones_limpio.shape[0])
print("Transiciones consecutivas válidas:",
df_transiciones_validas_limpio.shape[0])

print("\nPorcentaje de observaciones utilizables:")
```

```
print(
    round(
        (
            df_transiciones_validas_limpio.shape[0]
            / df_transiciones_limpio.shape[0]
        ) * 100,
        2
    ),
    "%"
)

print("\nDistribución de clases con umbral de ±10 %:")
display(
    df_transiciones_validas_limpio["Clase_variacion"]
    .value_counts()
    .reindex(["Baja", "Estabilidad", "Alza"])
    .to_frame(name="Número de observaciones")
)

print("\nEstadísticas de la variación porcentual:")
display(
    df_transiciones_validas_limpio["Variacion_pct_siguiente"]
    .describe()
    .to_frame(name="Variación porcentual")
)

# Guardar las bases definitivas para los siguientes pasos
df_modelo_base = df_transiciones_validas_limpio.copy()
df_mercados_limpio = df_eda.copy()
```

Creación de variables temporales y base de modelado:

```
# ========================================
# PASO 19: CREACIÓN DE VARIABLES TEMPORALES
# ========================================

# Crear una copia de la base con todas las observaciones mensuales
df_features = df_transiciones_limpio.copy()

# Definir las columnas que identifican cada serie temporal
claves_serie = [
    "Provincia",
    "Mercado",
    "Tipo_Produccion"
```

```
]

# Ordenar cada serie por fecha
df_features = df_features.sort_values(
    claves_serie + ["Fecha"]
).reset_index(drop=True)

# Crear fecha y precio de uno, dos y tres meses anteriores
for rezago in [1, 2, 3]:

    df_features[f"Fecha_lag_{rezago}"] = (
        df_features
        .groupby(claves_serie)["Fecha"]
        .shift(rezago)
    )

    df_features[f"Precio_lag_{rezago}"] = (
        df_features
        .groupby(claves_serie)["Precio_USD_Kg"]
        .shift(rezago)
    )

# Verificar que cada rezago corresponda realmente a meses consecutivos
for rezago in [1, 2, 3]:

    fecha_esperada = (
        df_features["Fecha"]
        - pd.DateOffset(months=rezago)
    )

    df_features[f"Lag_{rezago}_valido"] = (
        df_features[f"Fecha_lag_{rezago}"]
        == fecha_esperada
    )

    # Convertir en nulo el precio del rezago si existe una interrupción
mensual
    df_features.loc[
        ~df_features[f"Lag_{rezago}_valido"],
        f"Precio_lag_{rezago}"
    ] = np.nan

# Calcular la variación porcentual respecto al mes anterior
df_features["Variacion_pct_lag_1"] = (
```

```
    (
        df_features["Precio_USD_Kg"]
        - df_features["Precio_lag_1"]
    )
    / df_features["Precio_lag_1"]
) * 100

# Crear promedio móvil de tres meses:
# mes actual + dos meses anteriores
df_features["Promedio_movil_3"] = (
    df_features[
        [
            "Precio_USD_Kg",
            "Precio_lag_1",
            "Precio_lag_2"
        ]
    ]
    .mean(axis=1)
)

# Exigir tres valores disponibles para conservar el promedio móvil
df_features.loc[
    df_features[
        [
            "Precio_USD_Kg",
            "Precio_lag_1",
            "Precio_lag_2"
        ]
    ].isna().any(axis=1),
    "Promedio_movil_3"
] = np.nan

# Conservar solo observaciones que tienen transición válida hacia el mes
siguiente
df_features_modelo = df_features[
    df_features["Transicion_consecutiva"]
].copy()

# Crear nuevamente la variable objetivo
umbral_seleccionado = 10

df_features_modelo["Clase_variacion"] = np.select(
    [
```

```
        df_features_modelo["Variacion_pct_siguiente"] <= -
umbral_seleccionado,
        df_features_modelo["Variacion_pct_siguiente"] >=
umbral_seleccionado
    ],
    [
        "Baja",
        "Alza"
    ],
    default="Estabilidad"
)

# Revisar disponibilidad de las nuevas variables
variables_temporales = [
    "Precio_USD_Kg",
    "Precio_lag_1",
    "Precio_lag_2",
    "Precio_lag_3",
    "Variacion_pct_lag_1",
    "Promedio_movil_3"
]

print("Valores disponibles por variable temporal:")
display(
    pd.DataFrame({
        "Valores_disponibles":
df_features_modelo[variables_temporales].notna().sum(),
        "Valores_faltantes":
df_features_modelo[variables_temporales].isna().sum(),
        "Porcentaje_disponible": (
            df_features_modelo[variables_temporales]
            .notna()
            .mean() * 100
        ).round(2)
    })
)

# Mostrar ejemplos de las variables creadas
print("\nEjemplos de variables temporales:")
display(
    df_features_modelo[
        [
            "Fecha",
            "Provincia",
```

```
            "Mercado",
            "Tipo_Produccion",
            "Precio_USD_Kg",
            "Precio_lag_1",
            "Precio_lag_2",
            "Precio_lag_3",
            "Variacion_pct_lag_1",
            "Promedio_movil_3",
            "Clase_variacion"
        ]
    ].head(15)
)

# Guardar esta base para las siguientes etapas
df_modelo_temporal = df_features_modelo.copy()
# ========================================
# PASO 21: VARIABLES DE CALENDARIO Y
# BASE PRINCIPAL PARA MODELADO
# ========================================

# Crear copia de la base temporal ya preparada
df_modelo_principal = df_modelo_temporal.copy()

# Crear trimestre del año
df_modelo_principal["Trimestre"] = (
    ((df_modelo_principal["Mes_numero"] - 1) // 3) + 1
)

# Crear representación cíclica del mes
df_modelo_principal["Mes_sin"] = np.sin(
    2 * np.pi * df_modelo_principal["Mes_numero"] / 12
)

df_modelo_principal["Mes_cos"] = np.cos(
    2 * np.pi * df_modelo_principal["Mes_numero"] / 12
)

# Crear identificador de cada serie temporal
df_modelo_principal["Serie_mercado"] = (
    df_modelo_principal["Provincia"].astype(str)
    + " | "
    + df_modelo_principal["Mercado"].astype(str)
    + " | "
    + df_modelo_principal["Tipo_Produccion"].astype(str)
```

```
)

# Crear una variable que indica si existen tres meses consecutivos
previos
df_modelo_principal["Historial_3_meses_completo"] = (
    df_modelo_principal[
        [
            "Precio_lag_1",
            "Precio_lag_2",
            "Precio_lag_3"
        ]
    ]
    .notna()
    .all(axis=1)
)

# Mostrar distribución de trimestres
print("Distribución de observaciones por trimestre:")
display(
    df_modelo_principal["Trimestre"]
    .value_counts()
    .sort_index()
    .rename_axis("Trimestre")
    .reset_index(name="Número de observaciones")
)

# Revisar cantidad de observaciones con historial suficiente
print("\nObservaciones con tres meses previos consecutivos:")
print(
    df_modelo_principal["Historial_3_meses_completo"]
    .value_counts()
    .rename(
        {
            True: "Con historial completo",
            False: "Con historial incompleto"
        }
    )
)

# Definir las columnas que estarán disponibles antes de predecir
columnas_modelo_principal = [
    "Fecha",
    "Fecha_siguiente",
    "Año",
```

```
    "Mes",
    "Mes_numero",
    "Trimestre",
    "Mes_sin",
    "Mes_cos",
    "Provincia",
    "Mercado",
    "Cantón",
    "Tipo Mercado",
    "Tipo_Produccion",
    "Serie_mercado",
    "Precio_USD_Kg",
    "Precio_lag_1",
    "Precio_lag_2",
    "Precio_lag_3",
    "Variacion_pct_lag_1",
    "Promedio_movil_3",
    "Historial_3_meses_completo",
    "Variacion_pct_siguiente",
    "Clase_variacion"
]

# Conservar únicamente las columnas necesarias
df_modelo_principal = df_modelo_principal[
    columnas_modelo_principal
].copy()

# Ordenar cronológicamente
df_modelo_principal = df_modelo_principal.sort_values(
    ["Fecha", "Provincia", "Mercado", "Tipo_Produccion"]
).reset_index(drop=True)

# Mostrar estructura final
print("\nDimensiones de la base principal:")
print(df_modelo_principal.shape)

print("\nPrimeros registros de la base principal:")
display(df_modelo_principal.head(10))

# Guardar una copia en CSV para no perder el avance
df_modelo_principal.to_csv(
    "base_modelo_tomate_mercados.csv",
    index=False,
    encoding="utf-8-sig"
```

```
)

print("\nArchivo guardado: base_modelo_tomate_mercados.csv")
```

División temporal, preprocesamiento y entrenamiento de modelos base:

```
# ========================================
# PASO 47: PREPARACIÓN DEL MODELO BASE
# CON DIVISIÓN TEMPORAL
# ========================================

from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier

# Crear copia de la base final
df_modelado_base = df_modelado.copy()

# Seleccionar escenario base
nombre_escenario = "Escenario_1_base"
variables_modelo = escenarios_modelado[nombre_escenario]

# Conservar columnas necesarias
columnas_necesarias = variables_modelo + [variable_objetivo, "Fecha"]

df_escenario_base = df_modelado_base[columnas_necesarias].copy()

# Eliminar registros sin variable objetivo
df_escenario_base = df_escenario_base.dropna(subset=[variable_objetivo])

# Para el modelo base, eliminamos filas con faltantes en variables
predictoras
df_escenario_base = df_escenario_base.dropna(subset=variables_modelo)

# Ordenar por fecha
df_escenario_base =
df_escenario_base.sort_values("Fecha").reset_index(drop=True)
```

```
print("Dimensiones del escenario base:")
print(df_escenario_base.shape)

print("\nRango temporal del escenario base:")
print("Fecha mínima:", df_escenario_base["Fecha"].min())
print("Fecha máxima:", df_escenario_base["Fecha"].max())

print("\nDistribución de clases en escenario base:")
display(
    df_escenario_base[variable_objetivo]
    .value_counts()
    .reindex(["Baja", "Estabilidad", "Alza"])
    .reset_index(name="Registros")
    .rename(columns={"index": "Clase"})
)

# División temporal:
# entrenamiento hasta 2023
# prueba desde 2024
fecha_corte = pd.to_datetime("2024-01-01")

df_train = df_escenario_base[
    df_escenario_base["Fecha"] < fecha_corte
].copy()

df_test = df_escenario_base[
    df_escenario_base["Fecha"] >= fecha_corte
].copy()

print("\nRegistros de entrenamiento:")
print(df_train.shape[0])

print("\nRegistros de prueba:")
print(df_test.shape[0])

print("\nDistribución de clases en entrenamiento:")
display(
    df_train[variable_objetivo]
    .value_counts()
    .reindex(["Baja", "Estabilidad", "Alza"])
    .reset_index(name="Registros")
    .rename(columns={"index": "Clase"})
)
```

```
print("\nDistribución de clases en prueba:")
display(
    df_test[variable_objetivo]
    .value_counts()
    .reindex(["Baja", "Estabilidad", "Alza"])
    .reset_index(name="Registros")
    .rename(columns={"index": "Clase"})
)

# Separar X e y
X_train = df_train[variables_modelo]
y_train = df_train[variable_objetivo]

X_test = df_test[variables_modelo]
y_test = df_test[variable_objetivo]

# Identificar variables categóricas y numéricas
variables_categoricas = [
    columna for columna in variables_modelo
    if df_escenario_base[columna].dtype == "object"
]

variables_numericas = [
    columna for columna in variables_modelo
    if columna not in variables_categoricas
]

print("\nVariables categóricas:")
print(variables_categoricas)

print("\nVariables numéricas:")
print(variables_numericas)

# Preprocesamiento para variables numéricas
transformador_numerico = Pipeline(
    steps=[
        ("imputador", SimpleImputer(strategy="median")),
        ("escalador", StandardScaler())
    ]
)

# Preprocesamiento para variables categóricas
transformador_categorico = Pipeline(
```

```
    steps=[
        ("imputador", SimpleImputer(strategy="most_frequent")),
        ("onehot", OneHotEncoder(handle_unknown="ignore"))
    ]
)

# Unir transformaciones
preprocesador = ColumnTransformer(
    transformers=[
        ("numericas", transformador_numerico, variables_numericas),
        ("categoricas", transformador_categorico, variables_categoricas)
    ]
)
# ========================================
# PASO 48: ENTRENAMIENTO Y EVALUACIÓN
# DE MODELOS BASE
# ========================================

from sklearn.dummy import DummyClassifier
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score, f1_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier

# Definir modelos a comparar
modelos_base = {

    "Modelo_naive_mayoria": DummyClassifier(
        strategy="most_frequent"
    ),

    "Regresion_logistica": LogisticRegression(
        max_iter=2000,
        class_weight="balanced",
        random_state=42
    ),

    "Random_forest": RandomForestClassifier(
        n_estimators=500,
        max_depth=None,
        min_samples_leaf=3,
        class_weight="balanced",
```

```
        random_state=42,
        n_jobs=-1
    ),

    "Gradient_boosting": GradientBoostingClassifier(
        n_estimators=200,
        learning_rate=0.05,
        max_depth=3,
        random_state=42
    )
}

# Lista para guardar resultados generales
resultados_modelos_base = []

# Diccionario para guardar modelos entrenados
modelos_entrenados_base = {}

# Orden de clases para matrices de confusión
orden_clases = ["Baja", "Estabilidad", "Alza"]

# Entrenar y evaluar cada modelo
for nombre_modelo, modelo in modelos_base.items():

    print("\n" + "=" * 80)
    print(f"Modelo: {nombre_modelo}")
    print("=" * 80)

    # Crear pipeline completo: preprocesamiento + modelo
    pipeline_modelo = Pipeline(
        steps=[
            ("preprocesador", preprocesador),
            ("modelo", modelo)
        ]
    )

    # Entrenar modelo
    pipeline_modelo.fit(X_train, y_train)

    # Predecir en prueba
    y_pred = pipeline_modelo.predict(X_test)

    # Calcular métricas
    accuracy = accuracy_score(y_test, y_pred)
```

```
    f1_macro = f1_score(y_test, y_pred, average="macro")
    f1_weighted = f1_score(y_test, y_pred, average="weighted")

    # Guardar resultados
    resultados_modelos_base.append({
        "Modelo": nombre_modelo,
        "Accuracy": accuracy,
        "F1_macro": f1_macro,
        "F1_weighted": f1_weighted
    })

    # Guardar modelo entrenado
    modelos_entrenados_base[nombre_modelo] = pipeline_modelo

    # Mostrar reporte de clasificación
    print("\nReporte de clasificación:")
    print(
        classification_report(
            y_test,
            y_pred,
            labels=orden_clases,
            digits=3
        )
    )

    # Mostrar matriz de confusión
    matriz = confusion_matrix(
        y_test,
        y_pred,
        labels=orden_clases
    )

    matriz_df = pd.DataFrame(
        matriz,
        index=[f"Real_{clase}" for clase in orden_clases],
        columns=[f"Predicho_{clase}" for clase in orden_clases]
    )

    print("\nMatriz de confusión:")
    display(matriz_df)

# Crear tabla resumen de resultados
resultados_modelos_base = pd.DataFrame(resultados_modelos_base)
```

```
# Ordenar por F1 macro
resultados_modelos_base = resultados_modelos_base.sort_values(
    "F1_macro",
    ascending=False
).reset_index(drop=True)

print("\n" + "=" * 80)
print("Resumen comparativo de modelos base")
print("=" * 80)

display(resultados_modelos_base.round(4))

# Identificar mejor modelo según F1 macro
mejor_modelo_base_nombre = resultados_modelos_base.loc[0, "Modelo"]
mejor_modelo_base = modelos_entrenados_base[mejor_modelo_base_nombre]

print("\nMejor modelo base según F1 macro:")
print(mejor_modelo_base_nombre)
```

Optimización del Random Forest y evaluación final:

```
# ========================================
# PASO 51: OPTIMIZACIÓN DEL RANDOM FOREST BASE
# ========================================

from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import ParameterGrid
from sklearn.pipeline import Pipeline
from sklearn.metrics import f1_score, accuracy_score,
classification_report, confusion_matrix
import pandas as pd
import numpy as np

# Usaremos el escenario base ya preparado en el Paso 47
# df_train y df_test corresponden a la división temporal:
# entrenamiento antes de 2024 y prueba desde 2024

X_train = df_train[variables_modelo]
y_train = df_train[variable_objetivo]

X_test = df_test[variables_modelo]
y_test = df_test[variable_objetivo]

# Separar una parte final del entrenamiento como validación temporal
```

```
# Entrenamiento interno: antes de 2022
# Validación interna: 2022-2023

fecha_validacion = pd.to_datetime("2022-01-01")

df_train_interno = df_train[
    df_train["Fecha"] < fecha_validacion
].copy()

df_validacion = df_train[
    df_train["Fecha"] >= fecha_validacion
].copy()

X_train_interno = df_train_interno[variables_modelo]
y_train_interno = df_train_interno[variable_objetivo]

X_validacion = df_validacion[variables_modelo]
y_validacion = df_validacion[variable_objetivo]

print("Registros entrenamiento interno:", X_train_interno.shape[0])
print("Registros validación:", X_validacion.shape[0])
print("Registros prueba final:", X_test.shape[0])

# Definir combinaciones de hiperparámetros
parametros_rf = {
    "n_estimators": [300, 500, 700],
    "max_depth": [4, 6, 8, None],
    "min_samples_leaf": [2, 3, 5, 8],
    "min_samples_split": [2, 5, 10],
    "max_features": ["sqrt", "log2"],
    "class_weight": ["balanced"]
}

grid_parametros = list(ParameterGrid(parametros_rf))

print("\nNúmero de combinaciones a evaluar:")
print(len(grid_parametros))

resultados_optimizacion = []
mejor_f1 = -1
mejor_pipeline_rf = None
mejores_parametros_rf = None

# Evaluar combinaciones
```

```
for i, parametros in enumerate(grid_parametros, start=1):

    modelo_rf = RandomForestClassifier(
        random_state=42,
        n_jobs=-1,
        **parametros
    )

    pipeline_rf = Pipeline(
        steps=[
            ("preprocesador", preprocesador),
            ("modelo", modelo_rf)
        ]
    )

    pipeline_rf.fit(X_train_interno, y_train_interno)

    y_pred_val = pipeline_rf.predict(X_validacion)

    f1_macro_val = f1_score(
        y_validacion,
        y_pred_val,
        average="macro"
    )

    accuracy_val = accuracy_score(
        y_validacion,
        y_pred_val
    )

    resultados_optimizacion.append({
        "Combinacion": i,
        "F1_macro_validacion": f1_macro_val,
        "Accuracy_validacion": accuracy_val,
        **parametros
    })

    if f1_macro_val > mejor_f1:
        mejor_f1 = f1_macro_val
        mejor_pipeline_rf = pipeline_rf
        mejores_parametros_rf = parametros

# Convertir resultados a DataFrame
resultados_optimizacion = pd.DataFrame(resultados_optimizacion)
```

```
resultados_optimizacion = resultados_optimizacion.sort_values(
    "F1_macro_validacion",
    ascending=False
).reset_index(drop=True)

print("\nMejores combinaciones en validación:")
display(resultados_optimizacion.head(10).round(4))

print("\nMejores parámetros encontrados:")
print(mejores_parametros_rf)

print("\nMejor F1 macro en validación:")
print(round(mejor_f1, 4))

# Entrenar modelo final optimizado usando todo el entrenamiento 2012-
2023
modelo_rf_optimizado = RandomForestClassifier(
    random_state=42,
    n_jobs=-1,
    **mejores_parametros_rf
)

pipeline_rf_optimizado = Pipeline(
    steps=[
        ("preprocesador", preprocesador),
        ("modelo", modelo_rf_optimizado)
    ]
)

pipeline_rf_optimizado.fit(X_train, y_train)

# Evaluar en prueba final 2024-2026
y_pred_rf_optimizado = pipeline_rf_optimizado.predict(X_test)

accuracy_rf_optimizado = accuracy_score(
    y_test,
    y_pred_rf_optimizado
)

f1_macro_rf_optimizado = f1_score(
    y_test,
    y_pred_rf_optimizado,
    average="macro"
```

```
)

f1_weighted_rf_optimizado = f1_score(
    y_test,
    y_pred_rf_optimizado,
    average="weighted"
)

print("\nResultados del Random Forest optimizado en prueba final:")
print("Accuracy:", round(accuracy_rf_optimizado, 4))
print("F1 macro:", round(f1_macro_rf_optimizado, 4))
print("F1 weighted:", round(f1_weighted_rf_optimizado, 4))

orden_clases = ["Baja", "Estabilidad", "Alza"]

print("\nReporte de clasificación del Random Forest optimizado:")
print(
    classification_report(
        y_test,
        y_pred_rf_optimizado,
        labels=orden_clases,
        digits=3,
        zero_division=0
    )
)

matriz_rf_optimizado = confusion_matrix(
    y_test,
    y_pred_rf_optimizado,
    labels=orden_clases
)

matriz_rf_optimizado_df = pd.DataFrame(
    matriz_rf_optimizado,
    index=[f"Real_{clase}" for clase in orden_clases],
    columns=[f"Predicho_{clase}" for clase in orden_clases]
)

print("\nMatriz de confusión del Random Forest optimizado:")
display(matriz_rf_optimizado_df)
```

Importancia de variables:

```
# ========================================
```

```
# PASO 52: IMPORTANCIA DE VARIABLES
# DEL RANDOM FOREST OPTIMIZADO
# ========================================

import pandas as pd
import matplotlib.pyplot as plt

# Extraer el preprocesador y el modelo entrenado desde el pipeline
preprocesador_entrenado =
pipeline_rf_optimizado.named_steps["preprocesador"]
modelo_rf_entrenado = pipeline_rf_optimizado.named_steps["modelo"]

# Obtener los nombres de las variables después del preprocesamiento
nombres_variables_transformadas =
preprocesador_entrenado.get_feature_names_out()

# Limpiar nombres para que sean más legibles
nombres_variables_limpias = [
    nombre
    .replace("numericas__", "")
    .replace("categoricas__", "")
    for nombre in nombres_variables_transformadas
]

# Obtener importancia de variables del Random Forest
importancias = modelo_rf_entrenado.feature_importances_

# Crear tabla de importancia
df_importancia_variables = pd.DataFrame({
    "Variable": nombres_variables_limpias,
    "Importancia": importancias
})

# Ordenar de mayor a menor importancia
df_importancia_variables = df_importancia_variables.sort_values(
    "Importancia",
    ascending=False
).reset_index(drop=True)

print("Veinte variables más importantes del modelo:")
display(df_importancia_variables.head(20).round(4))

# Graficar las 20 variables más importantes
top_20_importancias = df_importancia_variables.head(20).copy()
```

```
plt.figure(figsize=(12, 8))

plt.barh(
    top_20_importancias["Variable"][::-1],
    top_20_importancias["Importancia"][::-1]
)

plt.title("Veinte variables más importantes del Random Forest
optimizado")
plt.xlabel("Importancia")
plt.ylabel("Variable")
plt.grid(axis="x", alpha=0.3)

plt.show()

# Guardar tabla de importancia
df_importancia_variables.to_csv(
    "importancia_variables_random_forest_optimizado.csv",
    index=False,
    encoding="utf-8-sig"
)

print("\nArchivo guardado:
importancia_variables_random_forest_optimizado.csv")
```