Universidad de Valparaíso
CHILE

Detección de Manipulación en Mercados Bursátiles

Memoria para optar al título profesional de Ingeniero Civil Matemático

JOHNNY HUINCAHUE DUARTE

Profesor Guía

Héctor Olivero Q.
Instituto de Ingeniería Matemática
Universidad de Valparaíso

Miembros de la comisión:

Karine Bertin
Instituto de Ingeniería Matemática
Universidad de Valparaíso

Cristian Meza
Instituto de Ingeniería Matemática
Universidad de Valparaíso

INSTITUTO DE INGENIERÍA MATEMÁTICA
VALPARAÍSO, DICIEMBRE 2025.

Resumen

En este trabajo se evalúa un enfoque para detectar manipulación en mercados bursátiles combinando aprendizaje automático supervisado y, cuando la resolución temporal lo permite, Análisis de Datos Funcionales. Se estudian dos escenarios. En datos reales diarios (caso FLC Group), las observaciones se representan con variables tabulares derivadas de retornos, volumen y volatilidad, y se comparan clasificadores clásicos bajo desbalance severo, usando métricas centradas en la clase minoritaria (principalmente F_2-Score, y el área bajo la curva de Precision-Recall). En paralelo, se generan datos sintéticos intradía en una grilla fija de 5 segundos con tres modelos (MBG, Heston y Cont-Müller), incorporando episodios de manipulación de *pump and dump* con distinta intensidad y distintos niveles de desbalance, lo que permite aplicar suavizado funcional (B-splines/wavelets), Análisis de Componentes Principales Funcionales y clasificación, además del algoritmo k-Nearest Neighbors funcional.

Los resultados muestran que el tratamiento explícito del desbalance de clases es determinante para evitar el colapso hacia la clase mayoritaria, y que la detectabilidad aumenta cuando la manipulación es más intensa o cuando se dispone de variables con señal directa del mecanismo (como profundidades en Cont-Müller). Como limitación central, el contraste funcional en datos reales queda restringido por la falta de información intradía, por lo que se propone como trabajo futuro incorporar datos de alta frecuencia reales y calibrar las simulaciones para aproximar condiciones de mercado más realistas.

Agradecimientos

En primer lugar, deseo agradecer a la comisión evaluadora por su disposición, comprensión y tiempo dedicado a la revisión de este trabajo. Asimismo, agradezco a todos los profesores que formaron parte de mi proceso universitario, quienes, con paciencia y vocación, estuvieron siempre dispuestos a resolver dudas y orientar mi aprendizaje a lo largo de estos años.

Un agradecimiento especial merece el Profesor Héctor Olivero, quien no solo cumplió el rol de profesor guía de esta memoria, sino que fue un apoyo constante tanto en el ámbito académico como personal. Su exigencia, claridad y compromiso marcaron profundamente mi formación, y su orientación fue clave para que este trabajo pudiera concretarse. Sin su guía y confianza, este camino habría sido considerablemente más difícil, y gran parte de lo logrado no habría sido posible.

A mi familia, gracias por el apoyo incondicional durante toda mi vida universitaria. A mis hermanas, por ser una fuente constante de inspiración y motivación para ser una mejor persona cada día. A mi padre, por su ayuda diaria y su presencia permanente en cada etapa de este proceso. A mis gatos, por su silenciosa compañía durante largas jornadas de estudio, que muchas veces hicieron más llevaderos los momentos difíciles.

Un agradecimiento profundo y especial a la persona más importante de mi vida, mi madre, Sally Duarte. Este logro es, en gran medida, suyo. Su esfuerzo, sacrificio, amor y apoyo incondicional a lo largo de todos estos años fueron el pilar fundamental que me permitió llegar hasta aquí. Dedicarle este trabajo es un honor y una forma, aunque insuficiente, de reconocer todo lo que ha hecho por mí. También quiero recordar con cariño a mis abuelas, quienes, a pesar de no estar físicamente presentes, siguen acompañándome y siendo parte de mi vida cada día.

Finalmente, agradezco a todas las personas que me acompañaron durante este proceso. A mis amigos de toda la vida, Luis Toro, Patricio Martínez, Nicolás Toro y entre otros, por su apoyo constante. A mis amigos que conocí en la universidad, Marco Barraza, Alonso Lagos, Felipe Córdova, entre otros, con quienes compartí desafíos, aprendizajes y momentos inolvidables. También a quienes conocí durante mi estancia en Valparaíso fuera de la universidad, que hicieron de esa etapa una experiencia significativa.

Por último, pero no menos importante, agradezco a mi novia, María José Stöwhas, por su compañía, paciencia y apoyo durante todo este largo proceso. Su presencia hizo este camino más llevadero y humano, y sin ella, sin duda, todo habría sido más difícil.

Índice general

Capítulo 1

Preliminares

1.1 Introducción y motivación

Los mercados financieros cumplen un rol central en la economía al facilitar la formación de capital y la asignación de recursos. Su correcto funcionamiento depende de la confianza de los participantes y de que los precios de los activos reflejen, en la medida de lo posible, la información disponible de manera veraz. Esta dinámica se manifiesta a través de variables observables como precios, volúmenes y medidas de liquidez. No obstante, la integridad de los mercados se ve amenazada por prácticas de manipulación, entendidas como acciones deliberadas orientadas a interferir con el proceso normal de formación de precios y a generar señales artificiales en el mercado.

Entre los distintos esquemas de manipulación, el *pump and dump* constituye uno de los casos más estudiados y recurrentes. Este esquema se caracteriza por una fase inicial de presión compradora artificial que eleva el precio del activo (*pump*), seguida por una liquidación rápida de posiciones que provoca una caída abrupta (*dump*). Estas dinámicas distorsionan el proceso de descubrimiento de precios y afectan principalmente a inversionistas menos informados, que suelen participar cuando el precio ya se encuentra inflado.

Tradicionalmente, la vigilancia de los mercados financieros ha dependido de la supervisión humana y de sistemas basados en reglas fijas. Sin embargo, el aumento del volumen de transacciones, la automatización del trading y la mayor velocidad de los mercados han reducido la efectividad de estos enfoques. La identificación de patrones anómalos sutiles y transitorios en grandes volúmenes de datos requiere métodos computacionales capaces de adaptarse a estructuras complejas y cambiantes. En este contexto, el aprendizaje automático se ha consolidado como una herramienta relevante para la detección de manipulación de mercado.

Una limitación común de muchos enfoques clásicos de aprendizaje automático en finanzas es el tratamiento de las series temporales como datos tabulares. Aunque este enfoque es válido, suele perder información asociada a la estructura temporal de los datos. Un episodio de manipulación no corresponde a una observación aislada, sino a una trayectoria que se desarrolla a lo largo del tiempo, con una forma característica que resulta relevante para su identificación.

Desde esta perspectiva, el Análisis de Datos Funcionales ofrece una alternativa para representar las series financieras. En lugar de trabajar con observaciones puntuales, cada trayectoria se modela como una función definida sobre un dominio temporal. Esta representación permite capturar la forma y la dinámica global de los movimientos del mercado de manera más estructurada. En esta memoria se propone combinar esta representación funcional con clasificadores clásicos de aprendizaje automático. En particular, el problema de detección de manipulación se formula como una tarea de clasificación binaria, donde cada trayectoria financiera se asigna a una de dos clases, trayectorias con manipulación

y trayectorias sin manipulación. Bajo este marco, el objetivo de evaluar si la representación funcional mejora la capidad de discriminación entre ambos tipos de comportamientos de mercado.

El desarrollo y evaluación de modelos de detección enfrenta además una dificultad práctica importante: la escasez de datos reales con etiquetas verificadas de manipulación. Los eventos confirmados de fraude son poco frecuentes y la información detallada suele ser confidencial. Para abordar esta limitación, este trabajo se apoya tanto en un caso de estudio real, correspondiente al activo FLC Group, como en datos sintéticos generados mediante modelos estocásticos. En particular, se utilizan el Movimiento Browniano Geométrico, el modelo de Heston y el modelo de microestructura de Cont-Müller para simular trayectorias con y sin manipulación bajo condiciones controladas. Este enfoque permite entrenar y evaluar los modelos y mitigar, de forma experimental, el problema de desbalance severo entre clases.

En síntesis, el objetivo de esta memoria es evaluar la viabilidad técnica de incorporar el Análisis de Datos Funcionales en la detección de manipulación de mercado. A través de experimentos con datos reales y sintéticos, se analiza si la representación funcional de las series financieras aporta ventajas frente a enfoques tradicionales basados en datos tabulares. Este trabajo no pretende desarrollar un sistema regulatorio operativo ni establecer relaciones causales económicas, sino comparar enfoques de representación y clasificación desde un punto de vista metodológico y computacional.

1.2 Objetivos

1.2.1 Objetivo General

Evaluar el desempeño de un enfoque que integre técnicas de Análisis Datos Funcionales y modelos de Aprendizaje Automático Supervisado para la detección de manipulación de precios en mercados financieros en esquemas de *pump and dump*.

1.2.2 Objetivos Específicos

- Entrenar un conjunto de modelos de Aprendizaje Automático Supervisado para la clasificación de datos etiquetados como manipulados o no manipulados.
- Aplicar algoritmos de clasificación automática para datos funcionales a series de precio sintéticas.
- Realizar una comparación de ambos enfoques a través de datos reales y simulados, analizando las métricas de rendimiento predictivo.

1.3 Contexto del Problema

El mercado de valores es un pilar fundamental del sistema financiero global, actuando como una fuente crucial de fondos para el desarrollo económico. Sin embargo, su integridad se ve constantemente amenazada por diversas formas de fraude, entre las cuales la manipulación de precios es una de las más perjudiciales según Alfajeer, Altaweel, Bouridane et al. [2]. La manipulación de mercado se define como cualquier acto deliberado que interfiere con el libre y justo funcionamiento de los mercados para crear situaciones artificiales o engañosas relativas al precio o al volumen de transacción. Estas prácticas no solo distorsionan el proceso de descubrimiento de precios, sino que también erosionan la confianza de los inversores, afectan la asignación eficiente de recursos y pueden provocar pérdidas económicas significativas ([40]).

Allen y Gale (1992) proporcionaron una clasificación fundamental de la manipulación en tres categorías: Basada en acciones, basada en información y basada en volumen de negociación. El esquema

pump and dump basado en información, el esquema *marking the close* basado en acciones, y los esquemas *wash and trades* y *spoofing* basados en volúmenes son los esquemas más comunes y difíciles de detectar ([3];[14];[30];[7]).

Tradicionalmente, la vigilancia de los mercados se ha basado en sistemas de reglas predefinidas y en el conocimiento de expertos ([22]). Sin embargo, la creciente complejidad, el volumen masivo de los datos y la velocidad de las transacciones, especialmente con el auge del trading de alta frecuencia, han hecho que estos métodos tradicionales sean insuficientes. Este escenario ha impulsado la adopción de enfoques más sofisticados y adaptativos, donde el aprendizaje automático se ha posicionado como una herramienta clave [35].

El aprendizaje automático ofrece un paradigma alternativo al modelado estadístico tradicional, ya que no presupone un proceso generador de datos conocido, sino que aprende patrones directamente de los datos. Las metodologías de aprendizaje automático para la detección de manipulación se pueden clasificar principalmente en aprendizaje supervisado, no supervisado y semi-supervisado ([2])

1.3.1 Aprendizaje Supervisado

El aprendizaje supervisado es el enfoque más extendido en la literatura, donde los algoritmos aprenden a partir de un conjunto de datos previamente etiquetado que contiene tanto instancias de operaciones normales como de manipulación. La manipulación, en este contexto, se convierte en un problema de clasificación. Los datos etiquetados suelen provenir de casos de litigio y enjuiciamiento documentados por organismos reguladores como la SEC de EE.UU o La CSRC de China ([45]).

Varios clasificadores han sido evaluados extensamente:

- Naive Bayes: A pesar de su supuesto "ingenuo" de independencia condicional de las características, el clasificador ha demostrado un rendimiento sólido. Golmohammadi, Díaz y Zaiane (2014) encontraron que superó a otros algoritmos como árboles de decisión, redes neuronales y Maquinas de vectores de soporte, lograron una alta sensibilidad (o recall) (89 %) en la detección de transacciones sospechosas. Uslu y Akal (2022) también reportaron un rendimiento superior en la detección de manipulación basada en transacciones en la Bolsa de Estambul.

- Maquinas de vectores de soporte (SVM): Es otro modelo popular que busca encontrar el hiperplano óptimo para separar las clases. Öğüt, Mete Doğanay y Aktaş (2009) encontraron que SVM era el más adecuado para la detección de manipulación que las técnicas estadísticas multivariadas, con SVM mostraron la mejor tasa de detección. Varios estudios han comparado Naive Bayes y SVM, llegando a conclusiones dispares que dependen en gran medida del conjunto de datos y las características de entrada utilizadas ([35]).

- Bosques aleatorios: Como método de ensamble, los bosques aleatorios combinan múltiples árboles de desición para mejorar la precisión y controlar el sobreajuste. Zhang et al. (2017) demostraron que los bosques aleatorios superan a SVM en la detección de manipulación en el mercado chino, atribuyendo su éxito a la capacidad de manejar ruido y un gran número de características. Los bosques aleatorios también son útiles para obtener una rápida comprensión de la importancia de las características.

- Redes Neuronales y Deep Learning (DL): Tanto las redes neuronales profundas (DNNs), como las Redes Neuronales Recurrentes (RNN) y las Long Short-Term Memory (LSTM), son especialmente adecuadas para datos de series temporales como las financieras. Estas arquitecturas pueden capturar dependencias temporales complejas en los datos de negociación. Por ejemplo, Wang et al. (2019) propusieron un modelo híbrido que combina RNN y aprendizaje por ensamble,

mostrando un rendimiento superior al considerar conjuntamente características de negociación y características estáticas de las acciones.

El principal desafío del aprendizaje supervisado es la escasez de datos etiquetados, ya que los casos de manipulación enjuiciados son raros en comparación con el volumen total de transacciones ([22]).

1.3.2 Aprendizaje No Supervisado

El aprendizaje no supervisado busca patrones en datos no etiquetados, lo que lo hace ideal para detectar anomalías o esquemas de manipulación desconocidos. Este se utiliza según la literatura para:

- Detección de Anomalías/Outliers: Este enfoque modela el comportamiento "normal" de las operaciones y señala cualquier desviación significativa como una posible manipulación. Técnicas como el *Peer Group Analysis* comparan el comportamiento de un activo con el de sus pares para detectar desviaciones ([27]). Otros métodos utilizan clustering (como DBSCAN o K-Means) para agrupar transacciones y considerar como outliers aquellas que no pertenecen a ningún clúster denso (Donoho, 2004, citado en [2]; Rukmi et al., 2019, citado en [2])
- Modelos Generativos Profundos: Técnicas como los Autoencoders (AE) y las Redes Generativas Adversarias (GANs) se entrenan para aprender la distribución de los datos de negociación normales. Un autoencoder aprende a reconstruir la entrada; cuando se presenta una operación manipulada, el error de reconstrucción será alto (Rizvi et al., 2020, citado en [2]). Leangarun, Tangamchit y Thajchayapong (2021) utilizaron LSTM-EA y LSTM-GANs para detectar manipulaciones desconocidas en la Bolsa de Tailandia, logrando identificar 5 de 6 casos reales enjuiciados con una baja tasa de falsos positivos.
- Análisis de Redes y Grafos: Este método modela las interacciones de los operadores como un grafo, donde los nodos son los operadores y las aristas son las transacciones. El análisis de la topología de la red puede revelar patrones colusorios como los *wash trades* (donde no hay cambio real de propiedad) o los operadores que comercian intesamente entre sí [7].

1.3.3 Aprendizaje Semi-Supervisado

Este enfoque utiliza una pequeña cantidad de datos etiqutados junto con una gran cantidad de datos no etiquetados. Diaz, Theodoulidis y Sampaio [14] aplicaron una estrategia de dos pasos: primero utilizaron clustering (no supervisado) para generar etiquetas iniciales de "sospechoso" o "normal" en datos intradía, y luego utilizaron estas etiquetas para entrenar un clasificador de árbol de decisión (supervisado). Este enfoque es prometedor dado el problema de la escasez de bases de datos etiquetadas.

1.3.4 Datos, Características y Desafíos

La calidad y granularidad de los datos son cruciales. En este contexto, la granularidad se refiere principalmente a la resolución temporal de las observaciones (por ejemplo, datos diarios, intradía o a nivel de transacción) y al nivel de detalle de las variables disponibles, como precios, volúmenes, spreads o información del libro de órdenes. Los estudios utilizan principalmente datos de mercados como NASDAQ, NYSE y bolsas asiáticas [2]. Los datos pueden ser:

- Datos de Nivel 1: Contienen información básica de precios y volumen. Son más accesibles pero menos informativos para detectar ciertos tipos de manipulación como el "*spoofing*" [28].

- Datos de Nivel 2 o 3 (Libro de Órdenes - LOB): Ofrecen una visión detallada de la oferta y la demanda a diferentes niveles de precios, incluyendo órdenes límite y cancelaciones [30].
- Datos Textuales: Noticias financieras, informes y discusiones de foros se utilizan para detectar manipulación basada en información [35].
- Datos Sintéticos: Dada la escasez de casos de manipulación reales, muchos estudios inyectan patrones de manipulación sintéticos en flujos de datos normales para entrenar y evaluar sus modelos [7].

En el contexto del aprendizaje automático, la ingeniería de características corresponde al proceso de transformar los datos financieros originales en un conjunto de variables de entrada más informativas para los modelos. Estas variables resumen distintos aspectos del comportamiento del mercado con el objetivo de facilitar la distinción entre dinámicas normales y potencialmente manipuladas. Entre las categorías de características más utilizadas en la literatura se encuentran:

- Basadas en Precio: Retornos diarios, volatilidad y medidas de reversión de precios [1].
- Basadas en Volumen: Volumen de negociación, frecuencia de transacciones y desequilibrio de órdenes (compra versus venta) [1].
- Indicadores de Mercado: Margen de compra-venta (*bid-ask spread*), capitalización de mercado y medidas de liquidez.
- Características del Libro de Órdenes: Profundidad del libro y volumen de cancelaciones. La profundidad del libro de órdenes se define como la cantidad de órdenes de compra y venta disponibles a distintos niveles de precios, y se asocia a la liquidez del mercado y al impacto potencial de nuevas órdenes.

A pesar de los avances, según lo planteado en Alfajeer, Altaweel, Bouridane et al. [2] persisten importantes desafíos:

1. Generalización y Adaptabilidad: Los manipuladores adaptan constantemente sus estrategias. Los modelos, especialmente los supervisados, entrenados con patrones históricos, pueden volverse obsoletos. Se necesitan modelos que puedan adaptarse a nuevos patrones de manipulación.
2. Calidad y Disponibilidad de Datos: La falta de un conjunto de datos público y estandarizado de casos de manipulación dificulta la comparación y validación de diferentes métodos. La dependencia de datos sintéticos implica que los patrones de manipulación utilizados para entrenar los modelos están condicionados por los supuestos del modelo generador, lo que puede limitar su capacidad de generalización cuando se aplican a datos reales.
3. Detección en Tiempo Real y Complejidad: El Trading de Alta Frecuencia exige sistemas de detección que operen en milisegundos o microsegundos. La complejidad computacional de los modelos de Deep Learning avanzados es un obstáculo para la implementación en tiempo real. Interpretabilidad de Modelos: Los modelos complejos, como las redes neuronales profundas, suelen considerarse "cajas negras", en el sentido de que el proceso interno que conduce a una decisión no es fácilmente interpretable, lo que dificulta que reguladores y analistas comprendan qué variables influyen en la clasificación de una observación como manipulada.

Capítulo 2

Activos Financieros, Mercados Financieros y Manipulación de Mercado

2.1 Activos Financieros

Según Fabozzi, Modigliani y Jones [15], en el ámbito de la teoría financiera, un activo es cualquier posesión que tenga valor de ser intercambiado. Estos pueden ser tangibles o intangibles. Un activo tangible es aquel cuyo valor depende particularmente de sus propiedades físicas, por ejemplo, las construcciones, terrenos o maquinarias. Por otro lado, los activos intangibles representan un derecho legal sobre un beneficio futuro; su valor no está relacionado con la forma física o de otra índole.

Los activos financieros son activos intangibles, cuyo valor intrínseco no reside en sus propiedades físicas, sino en la reclamación legal que representan sobre un beneficio futuro, típicamente un flujo de caja. Dicho de otro modo, es un instrumento que formaliza una relación crediticia o de propiedad, permitiendo la transferencia de fondos entre agentes económicos.

La formalización de esta transferencia se materializa a través de un instrumento financiero, término que en este contexto se considera intercambiable con el de activo financiero. La entidad que se compromete a efectuar los pagos futuros se denomina emisor, mientras que el propietario del activo es el inversor. A continuación, se presentan ejemplos de activos financieros:

- Un bono emitido por la Tesorería General de la República.
- Una acción ordinaria emitida por Latam Airlines.

En las acciones ordinarias de, por ejemplo, Latam Airlines, al inversor se le otorga el derecho de recibir los dividendos distribuidos por la compañía. En este caso, el inversor también tiene derecho a una parte proporcional del valor de liquidación de la empresa en caso de que ésta sea liquidada.

Los activos financieros se pueden clasificar según la naturaleza de la reclamación que otorgan a su titular:

1. Instrumentos de Deuda (o de Renta Fija): Estos activos, como los bonos o préstamos, representan una reclamación sobre un importe monetario fijo y predeterminado. El emisor se obliga a realizar pagos periódicos de interés y devolver el principal en una fecha de vencimiento establecida.
2. Instrumentos de Patrimonio (o de Renta Variable): El emisor está obligado a pagar un monto basado en las ganancias, si las hubiera al titular del activo, después de que se haya pagado a los propietarios de instrumentos de deuda. Las acciones ordinarias son un ejemplo de instrumento de patrimonio. Estos también son conocidos por tener mayor riesgo y volatilidad, pero capaces de generar mayores ganancias de capital.

3. Instrumentos Híbridos: Existen valores que combinan características de ambos, como las acciones preferentes (que ofrecen un pago fijo pero supeditado[1] al pago de la deuda) o los bonos convertibles (que permiten convertir la deuda en patrimonio bajo ciertas condiciones).

4. Instrumentos Derivados: Son contratos cuyo valor se deriva del valor de un activo subyacente, que puede ser otro activo financiero, un índice o una tasa de interés. Los tipos básicos son los contratos de futuros y las opciones. Su función económica principal es ofrecer un medio eficiente para controlar y gestionar riesgos y la especulación[2].

Además, los activos financieros se caracterizan por una serie de propiedades que determinan su valoración y atractivo para distintos inversores:

- Divisibilidad y Denominación: capacidad de ser fraccionados en unidades más pequeñas.
- Reversibilidad: coste asociado a la compra y posterior venta del activo para volver a efectivo.
- Flujo de caja: pagos que el activo generará para su propietario, como cupones o dividendos.
- Liquidez: facilidad y coste para vender un activo de forma inmediata sin afectar significativamente su precio de mercado.
- Riesgo: incertidumbre asociada a la predictibilidad de su rendimiento.

Desde la perspectiva de la microestructura del mercado [33], la existencia de los activos financieros cumple dos funciones económicas primordiales:

1. Transferencia de Fondos: canalizan recursos de las unidades de superávit (agentes con exceso de fondos, como ahorradores) hacia las unidades de déficit (agentes con necesidad de fondos para invertir activos tangibles, como empresas o gobiernos)

2. Redistribución del Riesgo: permiten transferir y redistribuir el riesgo al flujo de caja generado por los activos tangibles entre aquellos que buscan y aquellos que proveen fondos.

2.2 Mercados financieros

Un mercado financiero es una estructura o mecanismo donde se intercambian activos financieros. Estos mercados son cruciales para canalizar fondos de las unidades en superávit a las unidades de déficit. Este proceso puede ocurrir de dos maneras:

1. Financiamiento Directo: los prestatarios obtienen fondos directamente de los prestamistas en los mercados financieros mediante la venta de valores (como bonos o acciones). Aunque mediáticamente prominente, este método representa una fracción menor del financiamiento corporativo total.

2. Financiamiento Indirecto: Involucra a intermediarios financieros (como bancos, fondos de inversión y compañías de seguros) que actúan como puente entre prestamistas y prestatarios. Estos intermediarios emiten sus propios pasivos (depósitos, pólizas) para adquirir los activos (préstamos, bonos) de las unidades déficit. Este es el canal principal para movilizar fondos en la mayoría de las economías.

[1] En este contexto, "supeditado" significa que el pago de dividendos de las acciones preferentes está condicionado a que previamente se haya cumplido con el pago de la deuda de la empresa.

[2] Es el acto de apostar sobre la dirección futura del precio de un activo subyacente con el objetivo de obtener un beneficio

También, debemos considerar que la existencia de los mercados financieros no es una condición necesaria para la creación e intercambio de activos financieros, en la mayoría de las economías los activos se crean y posteriormente se negocian en algún tipo de mercado. Sin embargo, la existencia de estos es fundamental para el crecimiento económico y la eficiencia en la asignación de capital. Su necesidad y rol se manifiestan a través de varias funciones económicas esenciales [15, 31]:

1. Descubrimiento de Precios: la interacción entre compradores y vendedores determina el precio de los activos, lo cual, a su vez, establece el rendimiento requerido por los inversionistas. Estos precios actúan como señales que dirigen la asignación de recursos a sus usos más productivos. En mercados eficientes, se espera que los precios reflejen toda la información disponible [17].

2. Provisión de Liquidez: ofrecen un mecanismo para que los inversionista vendan sus activos financieros y convierten en efectivo. La liquidez, entendida como la facilidad realizar transacciones de forma inmediata a bajo coste, es una característica deseable que reduce el riesgo para los inversionistas y, por ende, el costo de capital para los emisores.

3. Reducción de Costos de Transacción: los mercados organizados disminuyen significativamente los costos de búsqueda e información. La presencia de intermediarios y la estandarización de contratos reducen los costos asociados a la búsqueda de contrapartes y a la evaluación de la calidad de los activos.

La ausencia de mercados financieros funcionales limita el desarrollo económico, como se observa en muchos países emergentes donde la debilidad de estas estructuras obstaculiza el crecimiento [31].

2.2.1 Microestructura del Mercado

El término microestructura de mercado fue mencionado por primera vez por [20], este es el estudio del proceso y los resultados de negociación de activos bajo un conjunto de reglas explícitas. Analiza cómo los mecanismos de negociación específicos afectan la formación de precios, la liquidez y la eficiencia del mercado. La microestructura, por tanto, abre la caja negra de la formación de precios, reconociendo que esta no es un proceso abstracto de oferta y demanda, sino el resultado de las interacciones estratégicas de diversos agentes bajo reglas institucionales concretas.

Los elementos clave que definen la microestructura de un mercado son [33]:

1. Participantes del Mercado: Los mercados están compuestos por una variedad de agentes con diferentes motivaciones y niveles de información. Bagehot [4] distinguió entre traders informados (que poseen información privada, también llamados insiders) y traders por liquidez o no informados (que negocian por necesidades de consumo o gestión de cartera). Además, existen intermediarios como:

   - Brokers (Corredores): actúan como agentes, ejecutando órdenes en nombre de sus clientes.
   - Dealers (Creadores de Mercado): actúan como principales, comprando y vendiendo por cuenta propia y obteniendo ganancias del diferencial entre precio de compra (bid) y el de venta (ask). Su función es crucial para proveer inmediatez y liquidez al mercado.
   - Especialistas: En mercados organizados como la Bolsa de Nueva York (NYSE), combina roles de bróker y dealer, con la obligación de mantener un mercado "justo y ordenado".

2. Mecanismos de Negociación: Se refieren a las reglas sobre cómo se envían, priorizan y ejecutan las órdenes. Los mercados pueden ser:

- Dirigidos por Órdenes (Orden-Driven): Los precios se forman por la interacción de las órdenes de compra y venta del público, generalmente a través de un libro de órdenes.
- Dirigidos por Cotizaciones (Quote-Driven): Los dealers publican continuamente los precios a los que están dispuestos a comprar y vender, y la negociación se realiza contra estas cotizaciones.

3. Transparencia del Mercado: En la práctica, la transparencia se refiere a información agregada y anonimizada sobre precios, órdenes y profundidades, y no a la identificación individual de los participantes.

4. Fricciones del Mercado: Costos que impiden un funcionamiento perfecto, como comisiones de corretaje, el *spread bid-ask*[3], impuestos y costos de adquirir y procesar información. Estos costos generan desviaciones entre el precio de transacción y el valor fundamental del activo

En mercados organizados basados en libros de órdenes, la profundidad del lado comprador (bid depth) y del lado vendedor (ask depth) corresponde a la cantidad de órdenes disponibles a distintos niveles de precio en cada lado del libro. Estas profundidades reflejan liquidez latente, es decir, la capacidad del mercado para absorber órdenes sin generar variaciones significativas en el precio. En esta memoria, la profundidad bid y ask se utiliza como una variable microestructural observable en modelos donde el volumen transado no está explícitamente modelado, en particular en el marco de simulación basado en Cont–Müller.

La microestructura, por lo tanto, es el conjunto de reglas e interacciones que determinan cómo la demanda latente de los inversores se traduce en precios y variables microestructurales observables, afectando la liquidez y la eficiencia informativa del mercado. En este contexto, la liquidez puede manifestarse a través de distintas variables observables, como el volumen transado o la profundidad del libro de órdenes. Si bien ambas capturan aspectos diferentes del proceso de negociación, en esta memoria se utilizan de manera complementaria según la disponibilidad de datos y el modelo considerado.

2.2.2 ¿Cómo se Clasifican los Mercados Financieros?

Los mercados financieros se pueden clasificar según diversos criterios, lo que permite un análisis más detallado de sus funciones y características [15]:

1. Por Tipo de Activo (Claim):
   - Mercado de Deuda (Debt Market): se negocian instrumentos de deuda como bonos y pagarés, que representan un préstamo que debe ser reembolsado.
   - Mercado de Renta Variable (Equity Market): se negocian acciones, que representan una participación en la propiedad y en los beneficios de una empresa.

2. Por Vencimiento del Activo:
   - Mercado Monetario (Money Market): se negocian activos de deuda a corto plazo (generalmente con vencimiento inferior a un año), como letras del Tesoro y papel comercial. Se caracterizan por su alta liquidez y bajo riesgo.
   - Mercado de Capitales (Capital Market): se negocian activos de deuda a largo plazo y acciones. Estos instrumentos se utilizan para financiar inversiones a largo plazo.

[3] Diferencia entre el precio más alto que un comprador está dispuesto a pagar y el precio más bajo que un vendedor está dispuesto a aceptar por un instrumento financiero.

3. Por la Novedad de la Emisión (Seasoning of Claim):
    - Mercado Primario (Primary Market): donde se emiten por primera vez los valores. Es el mecanismo a través del cual las empresas y gobiernos obtienen nuevo financiamiento.
    - Mercado Secundario (Secondary Market): donde se negocian los valores ya emitidos. Proporciona liquidez a los inversionistas y es crucial para la determinación de precios en el mercado primario.
4. Por la Estructura Organizativa:
    - Mercados Organizados (Exchanges): Tienen una ubicación física o una plataforma electrónica centralizada donde se realizan las transacciones bajo un conjunto de reglas estandarizadas, como la Bolsa de Nueva York o el Nasdaq.
    - Mercados Extrabursátiles (Over-the-Counter, OTC): Es una red descentralizada de dealers que negocian entre sí, generalmente por teléfono o medios electrónicos. Es el principal mercado para bonos y derivados.
5. Por Momento de Entrega:
    - Mercado al Contado (Spot Market): La entrega del activo y el pago se realizan de forma inmediata o en un plazo muy corto (típicamente dos días hábiles).
    - Mercado de Derivados (Derivative Market): se negocian contratos cuyo valor se deriva de un activo subyacente (acciones, bonos, materias primas). La liquidación se realiza en una fecha futura. Incluye futuros, opciones y swaps.
6. Por Perspectiva Geográfica:
    - Mercado Interno (National Market): incluye el mercado doméstico (emisores locales) y el mercado extranjero (emisores no domiciliados en el país).
    - Mercado Externo (International Market): También conocido como Euromercado, donde los valores se ofrecen simultáneamente a inversionistas de varios países y fuera de la jurisdicción de un solo país.

Aunque existen múltiples tipos de mercados financieros, esta memoria se centra en el mercado secundario de renta variable. La razón es que la manipulación de mercado se manifiesta durante la negociación de activos ya emitidos, donde los precios y volúmenes se forman a partir de la interacción continua entre órdenes de compra y venta.

El análisis se enfoca principalmente en mercados organizados y electrónicos, donde la negociación se realiza mediante mecanismos basados en libros de órdenes. En este tipo de mercados es posible observar directamente variables como precios, volúmenes y cambios en la liquidez, que son las entradas utilizadas posteriormente en los modelos de detección.

Otros segmentos, como el mercado primario o ciertas transacciones extrabursátiles, quedan fuera del alcance del estudio, ya que no presentan un proceso de formación de precios continuo ni generan información comparable para la detección algorítmica de manipulación.

Este alcance fija el contexto institucional del trabajo y permite definir de manera clara qué se entiende por comportamiento normal y comportamiento anómalo en los capítulos posteriores.

2.3 Representación de Activos Financieros

En el análisis cuantitativo, una serie de tiempo financiera es una secuencia de observaciones de una o más variables, indexadas cronológicamente. Estas pueden ser diaria, intradaria o de alta frecuencia.

Formalmente, consideramos un proceso estocástico $\{P(t)\}_{t\in\mathcal{T}}$ definido en un espacio de probabilidad $(\Omega, \mathcal{F}, \mathbb{P})$, donde $\mathcal{T}$ es un conjunto de índices de tiempo. En la práctica, no se observa la trayectoria completa del proceso, sino realizaciones muestreadas en instantes discretos de tiempo. Es decir, los datos financieros corresponden a una discretización de un proceso estocástico subyacente, observada en una grilla temporal finita. Para un activo específico a y un período de N días de negociación, una serie de tiempo de precios de cierre se denota como una secuencia $\{P_{a,t}\}_{t=1}^{N}$, donde $P_{a,t}$ es el precio del activo a en el día t. Si trabajamos con datos intradiarios, por ejemplo, horarios, la serie se convierte en $\{P_{a,t,h}\}_{t=1,h=1}^{N,H}$, donde H es el número de intervalos horarios en una jornada bursátil. Cada una de estas secuencias es un vector de alta dimensionalidad que, en su forma cruda, puede no ser directamente apto para muchos algoritmos de machine learning [32]. La ingeniería de características busca transformar estos datos brutos en un conjunto de predictores informativos que capturen la dinámica relevante.

Si bien la ingeniería de características es el enfoque predominante en el aprendizaje automático clásico, esta memoria combina representaciones basadas en características con una representación funcional de las series, la cual permite preservar su estructura temporal completa.

Precios

Los precios son la variable más fundamental y su tratamiento depende de los objetivos del estudio. Tenemos distintos tipos de precios:

- Precio de Cierre: es el precio de la última transacción del activo durante el día de negociación.
- Precio de Apertura: es el precio al que se negocio el activo al comenzar el día de negociación.
- Precio Máximo: es el precio máximo que alcanzo el activo durante el día de negociación.
- Precio Mínimo: es el precio mínimo que alcanzo el activo durante el día de negociación.
- Precios de Oferta y Demanda: Representan el precio máximo que un comprador está dispuesto a pagar y el precio mínimo que un vendedor está dispuesto a aceptar, respectivamente, durante el el momento de negociación. La diferencia entre ambos, es conocida como diferencial de precios (bid-ask spread), es una medida primordial de liquidez. Un spread amplio indica baja liquidez, mientras que una estrecha sugiere alta liquidez [9].

Volúmenes

El volumen de negociación es un indicador crucial de la actividad y el interés del mercado en un activo. Dentro de este, tenemos los siguientes tipos:

- Volumen de Transacciones: Usualmente denotado como $V_{a,t}$, corresponde al número total de acciones del activo a negociadas durante el período t.
- Volumen Ponderado Por Precio: Es el valor monetario total de las transacciones, calculado como la suma del producto de precio y volumen para cada transacción en un período. Esta medida captura mejor el impacto económico de la actividad de negociación.
- Volumen en Libros de Órdenes: En datos de alta frecuencia, se analiza el volumen de acciones disponibles en los distintos niveles de precios de compra y venta en libro de órdenes.

En esta memoria, el volumen se considera una de varias medidas posibles de liquidez, y su utilización depende de la disponibilidad de datos y del modelo empleado.

Retornos

Los retornos estandarizan las variaciones de precios, transformando series de tiempo a menudo en procesos más manejables estadísticamente:

- Retorno Simple: representa el cambio porcentual en el precio:

$$R_{i,t} = \frac{P_{a,t} - P_{a,t-1}}{P_{a,t-1}}.$$

- Retorno Logarítmico: esta variable es preferida en los estudios econométricos debido a sus propiedades estadísticas, como la aditividad temporal. Se define como:

$$r_{a,t} = \log\left(\frac{P_{a,t}}{P_{a,t-1}}\right).$$

Volatilidad

La volatilidad mide la dispersión de los retornos y es un indicador de riesgo y la incertidumbre del mercado. Esta suele ser definida la desviación estándar de los retornos logarítmicos $r_{a,t}$ sobre una ventana de k períodos. Para el día t, se calcula como:

$$\sigma_{a,t,k} = \sqrt{\frac{1}{k}\sum_{j=1}^{k}(r_{a,\,t-j} - \bar{r}_a)^2}$$

donde $\bar{r} = \frac{1}{k}\sum_{j=1}^{k} r_{a,t-j}$ corresponde al retorno medio muestral en la ventana considerada.

En este contexto, la volatilidad se emplea como una medida empírica local en el tiempo de la dispersión de los retornos, utilizada como variable observable en tareas de clasificación, y no como un modelo estructural de la dinámica de la volatilidad.

2.4 Mercados Eficientes

El concepto de eficiencia del mercado se formaliza en la Hipótesis del Mercado Eficiente (HME), desarrollada por [17]. La idea central es que los precios de los activos financieros reflejan plenamente toda la información disponible en un momento dado. En un mercado así, los precios son señales precisas para la asignación de capital, y los participantes no pueden obtener de manera consistente rendimientos anormales (es decir, superiores a los justificados por el riesgo asumido) utilizando un conjunto de información específico. La HME no implica que los mercados sean perfectos; de hecho, asume la existencia de costos de transacción e información[16]. En su versión "más débil y económicamente sensata", la eficiencia se mantiene hasta el punto en que los beneficios marginales de actuar sobre la información no superan los costos marginales de adquirirla y procesarla.

En esta memoria, la Hipótesis de Mercado Eficiente no se utiliza como una descripción literal del funcionamiento real de los mercados financieros, sino como un marco de referencia. Bajo este enfoque, la HME define un comportamiento base o "normal" de precios y volúmenes en ausencia de intervenciones estratégicas.

La detección de manipulación se plantea entonces como la identificación de desviaciones sistemáticas respecto de este comportamiento de referencia. Estas desviaciones se interpretan como anomalías en las series temporales financieras, sin que ello implique que los mercados sean perfectamente eficientes en la práctica.

Este uso de la HME permite formalizar el problema de detección desde un punto de vista estadístico, sin asumir que la hipótesis se cumple de manera estricta.

La Hipótesis del Mercado Eficiente se clasifica en tres formas, según el conjunto de información que se considera reflejado en los precios [17]:

1. Eficiencia en Forma Débil (Weak-Form-Efficiency): Los precios actuales reflejan toda la información contenida en el historial de precios y volúmenes pasados. Si se cumple, el análisis técnico[4] sería inútil para generar rendimientos anormales.

2. Eficiencia en Forma Semifuerte (Semi-Strong-Form-Efficiency): Los precios incorporan no solo la información histórica, sino también toda la información públicamente disponible. Esto incluye informes anuales, noticias económicas, anuncios de dividendos, etc. Si esta forma se sostiene, el análisis fundamental basado en datos públicos no permitiría obtener rendimientos superiores de forma consistente.

3. Eficiencia en Forma Fuerte (Strong-Form Efficiency): Los precios reflejan toda la información disponible, tanto pública como privada. Si fuera cierta, ni siquiera los insiders (como directivos de empresas) podrían beneficiarse de su información privilegiada.

Un concepto clave asociado a la HME es que los precios de los activos deberían seguir un "paseo aleatorio", lo que significa que los cambios futuros de precios, son en gran medida, impredecibles a partir de la información pasada. Esto no implica que los precios no cambien, sino que responden a la llegada de nueva información, lo cual es, por definición, impredecible [11, 31].

Matemáticamente, la HME, se formaliza a través de un modelo de "juego justo".

Sea $P_{a,t}$ el precio del activo a en el momento t, y sea $r_{a,t+1}$ el rendimiento de dicho activo en el período que va desde t a $t+1$. Sea Φ_t el conjunto de toda la información disponible en el momento t. El rendimiento del activo a para el período $t+1$, condicionado a la información disponible en t, se denota como $\mathrm{E}[r_{a,t+1}|\Phi_t]$.

El exceso de rendimiento (también denominado rendimiento anormal en la literatura), $\epsilon_{a,t+1}$, se define como la diferencia entre el rendimiento real y el rendimiento esperado, condicionado a la información disponible en t:

$$\epsilon_{a,t+1} = r_{a,t+1} - \mathrm{E}[r_{a,t+1}|\Phi_t] \tag{2.1}$$

Bajo la formulación de "juego justo", el exceso de rendimiento satisface la propiedad [17]:

$$\mathrm{E}[\epsilon_{a,t+1}|\Phi_t] = 0$$

Esto implica que, en promedio, no es posible utilizar la información disponible en el tiempo t para predecir el rendimiento anormal en $t+1$. En otras palabras, la información Φ_t no tiene poder predictivo sobre las desviaciones futuras de los rendimientos respecto a sus expectativas racionales.

El problema fundamental aquí es definir el rendimiento esperado. Esto requiere un modelo de equilibrio de precios de activos. Por ejemplo, utilizando el Modelo de Valoración de Activos de Capital (CAPM), el rendimiento esperado se define como [11]:

[4] El análisis técnico busca patrones en los precios históricos para predecir movimientos futuros.

$$\mathrm{E}[r_{a,t+1}|\Phi_t] = r_{f,t+1} + \beta_a(\mathrm{E}[r_{m,t+1}|\Phi_t] - r_{f,t+1}) \tag{2.2}$$

donde $r_{a,t+1}$ es el rendimiento total del activo a, $r_{f,t+1}$ es el rendimiento del activo libre de riesgo, $r_{a,t+1}$ es el rendimiento del portafolio de mercado y β_a es el riesgo sistemático del activo. En este caso, el conjunto de información Φ_t incluiría todos los datos necesarios para estimar los parámetros del CAPM. Cualquier prueba de la HME es, por tanto, una prueba de hipótesis conjunta: se prueba simultáneamente la eficiencia del mercado y la validez del modelo de precios de activos utilizado [16].

Las tres formas de eficiencia anteriormente mencionadas matemáticamente se definen como Φ_t [17]:

1. Forma Débil: Φ_t contiene únicamente el historial de precios pasados, $\{P_{a,t},\, P_{a,t-1}, \ldots\}$. La HME débil implica que $\mathrm{E}[r_{a,t+1}|P_{a,t},\, P_{a,t-1}, \ldots] = \mathrm{E}[r_{a,t+1}]$.

2. Forma Semifuerte: Φ_t incluye toda la información públicamente disponible (historial de precios, noticias, informes financieros, etc.).

3. Forma Fuerte: Φ_t incluye toda la información, tanto pública como privada.

La evidencia empírica desafía la HME en su forma fuerte, ya que estudios demuestran que los insiders obtienen beneficios anormales de sus transacciones [26].

Una implicación más específica, aunque menos general de la HME es que los precios de los activos siguen un "paseo aleatorio". Esto ocurre si los sucesivos cambios de precios son independientes y están idénticamente distribuidos (i.i.d.).

Matemáticamente, para los logaritmos de los precios $\ln(P_{a,t})$, la hipótesis del paseo aleatorio establece que:

$$\ln(P_{a,t+1}) = \ln(P_{a,t}) + \mu + \varepsilon_{t+1},$$

donde μ es el rendimiento esperado (o drift) y ε_{t+1} es un término de ruido blanco con media cero, varianza constante σ^2, y que es independiente de los términos de ruido pasados (i.e., $\mathrm{Cov}(\varepsilon_t, \varepsilon_{t+k}) = 0$ para $k \neq 0$).

Esta formulación implica que el mejor pronóstico del precio de mañana es el precio de hoy, más un drift esperado, y que cualquier desviación de este pronóstico es impredecible. El modelo del paseo aleatorio es consistente con la HME débil, ya que si los precios siguen un paseo aleatorio, el análisis técnico, no puede generar beneficios anormales [31]. De hecho, la competencia en el mercado asegura que los precios reaccionen inmediatamente a cualquier patrón predecible, eliminado así la oportunidad de beneficio [33].

2.4.1 Evidencias, Anomalías y Críticas

Dentro de la literatura, el término anomalía se utiliza para describir patrones de comportamiento de los precios que no son consistentes con las versiones estándar de la Hipótesis de Mercado Eficiente. Sin embargo, no todas las anomalías tienen el mismo origen ni la misma interpretación.

Por un lado, existen anomalías de carácter endógeno, como los efectos de calendario, el efecto tamaño o fenómenos de sobrerreacción y subreacción. Estos patrones se asocian a comportamientos agregados del mercado o a fricciones sistemáticas, sin requerir necesariamente la acción deliberada de agentes individuales.

Por otro lado, se encuentran las anomalías estratégicas, como la manipulación de mercado y el uso de información privilegiada. En estos casos, las desviaciones respecto del comportamiento esperado surgen de acciones intencionales orientadas a obtener beneficios económicos.

Esta distinción es relevante para esta memoria, ya que el foco del análisis y de los métodos de detección se concentra exclusivamente en anomalías estratégicas, que son las que corresponden a conductas potencialmente sancionables y observables a través de patrones anómalos en los datos de mercado.

En este contexto, cabe cuestionarse si "¿pueden ser los mercados eficientes?". La literatura respalda, al menos, las formas débil y semifuerte de la HME:

- Paseo Aleatorio de Precios: la evidencia empírica generalmente sugiere que los precios de las acciones siguen un paseo aleatorio, lo que hace que los cambios futuros sean impredecibles a partir de datos pasados [31]. Esto debilita la premisa del análisis técnico.
- Desempeño de Gestores Profesionales: los estudios sobre el rendimiento de analistas de inversión y gestores de fondos de inversión, en su mayoría, no han logrado demostrar que estos profesionales puedan superar consistentemente al mercado después de ajustar por riesgo y costos [16]. Este hallazgo es consistente con un mercado donde la información pública se refleja rápidamente en los precios, eliminando oportunidades de arbitraje[5].
- Estudios de Eventos: los estudios que analizan la reacción de los precios a anuncios públicos específicos (como fusiones, divisiones de acciones o informes de ganancias) generalmente encuentran que los precios se ajustan de manera muy rápida, a menudo en cuestión de minutos o en un día [16]. Esta rápida incorporación de nueva información es una fuerte evidencia a favor de la eficiencia semifuerte.

A pesar del apoyo general, la investigación ha documentado numerosas "anomalías" o patrones que desafían la HME. Estas anomalías sugieren que, en ciertas circunstancias, los mercados no son perfectamente eficientes, tenemos como casos:

- Manipulación del Mercado: la existencia de manipulación de precios es, por definición, una violación de los supuestos de la eficiencia del mercado, ya que implica la creación deliberada de precios artificiales que no reflejan el valor fundamental. La efectividad de las prácticas de manipulación, a menudo dirigidas a acciones ilíquidas o en mercados emergentes [1, 18], desafía la idea de que los precios reflejan siempre toda la información disponible. La posibilidad de manipulación se ve exacerbada por la asimetría de información. Este tipo de anomalía se verá a fondo más adelante, ya que es el principal foco de esta memoria.
- Insider Trading: el uso de información privilegiada (insider trading) para obtener beneficios anormales es una clara violación de la eficiencia en su forma fuerte [35]. Estudios empíricos han confirmado que los insiders efectivamente obtienen beneficios de sus transacciones, lo que contradice la HME fuerte, indicando que operan basándose en información privada. Esto se ha observado incluso en mercados regulados como el español[13].
- Anomalías de Calendario y de Tamaño: se han identificado patrones persistentes en los rendimientos, como el "efecto enero" (rendimientos anormalmente altos en enero, especialmente para empresas pequeñas) o el "efecto fin de semana" (rendimientos promedio negativos los lunes) [16]. El "efecto de la empresa pequeña" muestra que las empresas de menor capitalización han obtenido históricamente rendimientos anormalmente altos, incluso después de ajustar por el riesgo [31].

[5] El arbitraje es el acto de obtener beneficios sistemáticos sin riesgo (o ajustados por el riesgo) mediante la explotación de precios temporalmente mal valorados. Si tales oportunidades persistieran, agentes podrían capturarlas de forma recurrente, lo que se reflejaría en rendimientos superiores al mercado.

- Sobrerreacción y Subreacción del Mercado: Existe evidencia de que los inversores tienen a sobrerreaccionar a noticias inesperadas y dramáticas, lo que lleva a movimientos de precios predecibles a largo plazo [17]. Por otro lado, los mercados a veces subreaccionan a anuncios, como los de beneficios, lo que resulta en una deriva post-anuncio en los rendimientos [16].

- Burbujas y Caídas Abruptas: la existencia de burbujas especulativas (donde los precios se desvían masivamente de su valor fundamental) y caídas bursátiles como la de 1987, que son difíciles de explicar por cambios en los fundamentales económicos, pone en duda la versión fuerte de la HME que postula que los precios siempre reflejan el valor intrínseco [16]. La finanza conductual argumenta que estos fenómenos se explican por sesgos psicológicos de los inversores, como el exceso de confianza o el comportamiento gregario.

Más allá de la evidencia empírica, existen críticas conceptuales a los fundamentos de la HME:

- El Problema de la Hipótesis Conjunta: Como [16] reconoce, la HME no es comprobable por sí misma. Cualquier prueba de eficiencia de mercado es, inevitablemente, una prueba conjunta de la eficiencia y del modelo de equilibrio de precios de activos utilizados para definir los retornos "normales" . Por lo tanto, cuando se encuentra una anomalía, es imposible determinar si se debe a la ineficiencia del mercado o a que el modelo de riesgo es incorrecto. Esto crea una ambigüedad fundamental que limita las inferencias precisas sobre el grado de eficiencia del mercado.

En respuesta a la pregunta planteada "¿pueden ser los mercados eficientes?", la respuesta es que la eficiencia es una aproximación útil, pero no una descripción perfecta de la realidad [16]. La evidencia sugiere que los mercados son notablemente rápidos en procesar información, especialmente la pública, lo que hace muy difícil para la mayoría de los inversores "batir al mercado" de forma consistente. Esto se alinea con las formas débil y semifuerte de la HME.

Sin embargo, las anomalías documentadas, y en particular la existencia probada de prácticas como el insider trading y la manipulación de mercado, demuestran que los precios pueden desviarse de su valor fundamental debido a asimetrías de información y acciones deliberadas. La eficiencia en forma fuerte es ampliamente rechazada.

2.5 Manipulación de Mercados

La existencia de prácticas de manipulación constituye uno de los desafío más directo a la validez de la Hipótesis de Mercado Eficiente. Mientras que la teoría asume que los precios se forman libremente, la manipulación crea distorsiones que vulneran los tres niveles de eficiencia definidos anteriormente:

1. Forma Débil: Al crear patrones artificiales de precio y variables de liquidez, busca explotar a los agentes que creen en la predictibilidad de estos patrones, desafiando directamente esta forma de eficiencia

2. Forma Semi-Fuerte: Cuando se realiza un anuncio público, los precios se ajustan de manera rápida e insesgada; al introducir información falsa en el dominio público, se provoca que los precios reflejen no la realidad, sino la desinformación, atentando contra esta forma de eficiencia.

3. Forma Fuerte: la existencia de retornos anormales por parte de los insiders corporativos, que poseen información privada no divulgada, contradice esta forma de eficiencia.

Para analizar estas vulneraciones, la literatura sobre microestructura de mercados ofrece una taxonomía fundamental para el estudio de la manipulación. Allen y Gale [3] propusieron una clasificación que

distingue las estrategias de manipulación en función de los medios empleados. Esta clasificación es útil para el modelado matemático, ya que cada categoría deja huellas estadísticas distintas en las variables observadas, que son potencialmente detectables mediante machine learning [35]. Las categorías son según [3]:

1. Manipulación Basada en Acciones (Action-Based): ocurre cuando un agente, usualmente con control sobre una corporación (como un ejecutivo), toma decisiones empresariales que alteran el valor real o percibido de la firma con la intención de generar ganancias a partir de una posición preexistente en sus valores. Un ejemplo histórico es el cierre de plantas de producción por parte de ejecutivos de American Steel para despreciar el valor de las acciones que habían vendido en corto. Este tipo de manipulación podría manifestarse como una anomalía estructural en los rendimientos que preceden a un anuncio importante, identificable mediante técnicas de detección de outliers.

2. Manipulación Basada en Información (Informative-based): consiste en difundir información falsa o rumores con el objetivo de inducir a error a los participantes del mercado y provocar movimientos de precios artificiales. Estrategias como el *pump and dump* son ejemplos. Un manipulador adquiere una posición larga en un activo, difunde noticias positivas fraudulentas para atraer a otros inversores e inflar el precio, y finalmente liquida su posición con ganancias sustanciales. El auge de internet y las redes sociales ha amplificado la viabilidad de estas estrategias.

3. Manipulación Basada en Transacciones (Trade-based): esta categoría, la más difícil de erradicar según [3], no involucra acciones corporativas ni información falsa, sino que se basa únicamente en la ejecución de órdenes de compra y venta para crear apariencia engañosa de actividad o para influir directamente en el precio. Tácticas como las *wash trades* (operaciones ficticias sin cambio de beneficiario real) o matched orders (órdenes cruzadas coordinadas) buscan inflar artificialmente el volumen de transacciones. Otras tácticas incluyen el *spoofing* y *layering*, donde se ingresan grandes órdenes con la intención de cancelarlas antes de su ejecución para engañar a otros sobre la profundidad del mercado y la dirección del precio. Estas actividades dejan patrones anómalos en variables microestructurales observables, como volumen transado, profundidad del libro de órdenes y volatilidad local, que son susceptibles de ser identificados mediante modelos de machine learning.

2.5.1 Esquema pump and dump

El *pump and dump* es un esquema de manipulación basado en información en el que uno o más agentes coordinan compras y difusión de señales (por ejemplo, rumores o mensajes) para elevar artificialmente el precio de un activo, con el objetivo de vender posteriormente a precios inflados. La ganancia del manipulador proviene de comprar antes de la subida inducida y liquidar durante la fase de sobreprecio, transfiriendo el riesgo a participantes tardíos que compran cuando el precio ya se encuentra inflado. Este patrón se observa con mayor frecuencia en activos de baja liquidez, donde un flujo de órdenes relativamente pequeño puede mover el precio de manera significativa [1, 34].

Desde un punto de vista temporal, el evento puede describirse mediante tres tiempos críticos: t_0 (inicio del evento), t_1 (fin del *pump* e inicio del *dump*) y t_2 (fin del *dump* y transición al régimen post-evento). La Figura 2.1 resume este comportamiento de forma esquemática. En el intervalo $[t_0, t_1)$, el manipulador (y/o un grupo coordinado) ejerce presión compradora y busca atraer demanda externa, generando una trayectoria de precio al alza que no se explica por cambios en el valor fundamental del activo. En el intervalo $[t_1, t_2)$ se produce la fase de *dump*, caracterizada por ventas agresivas (realización de ganancias) y una corrección abrupta del precio. Finalmente, para $t \geq t_2$ suele observarse un régimen post-evento con estabilización parcial del precio, aunque frecuentemente en niveles inferiores a los observados durante el *pump*, consistente con la reversión del componente artificial del movimiento.

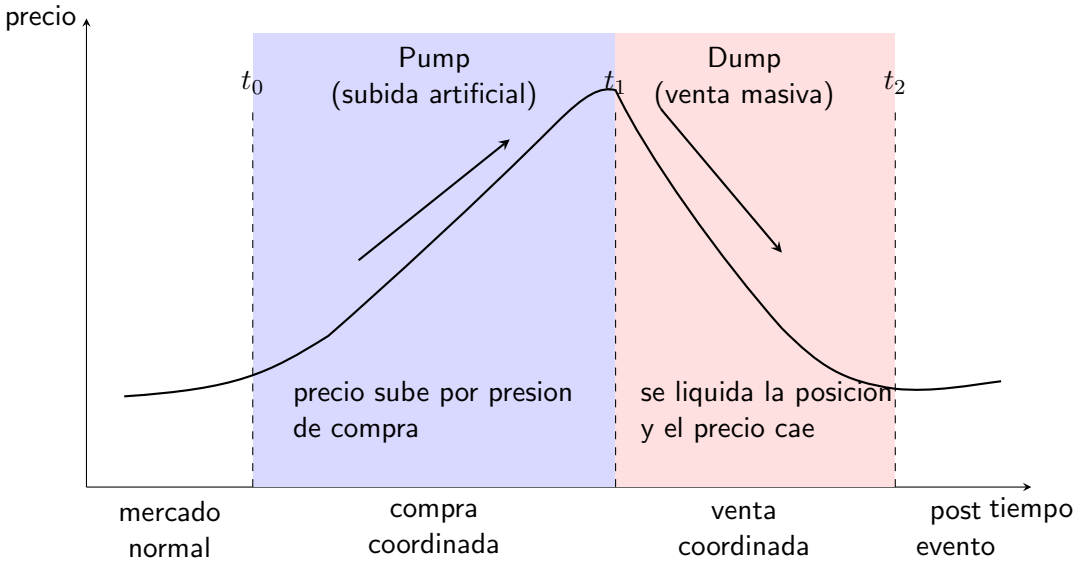


Figura 2.1: Esquema conceptual del *pump and dump*: fase de *pump* (subida artificial) seguida de *dump* (venta masiva y caída).

Desde la perspectiva de variables observables, el *pump and dump* tiende a dejar una firma conjunta en precio, retornos y variables de liquidez. Primero, durante el *pump* se observa un tramo con retornos positivos inusuales y un aumento del volumen, consistente con presión compradora y entrada de participantes no informados. Segundo, durante el *dump* suele aparecer una reversión rápida con retornos negativos grandes y un volumen elevado asociado a ventas. Tercero, es común observar un incremento transitorio de la volatilidad alrededor del evento, reflejando una dinámica más inestable que el comportamiento base del activo [1, 34]. Estas regularidades motivan que, en los capítulos posteriores, la detección se formula como un problema de clasificación basado en características derivadas de retornos, volatilidad y volumen, con énfasis en métricas robustas al desbalance de clases.

En esta memoria, el *pump and dump* se utiliza como caso central por dos razones: primero, es un ejemplo representativo de manipulación basada en información dentro de la taxonomía de Allen y Gale [3], y segundo, genera patrones observables en series de tiempo que permiten una evaluación consistente tanto en datos reales etiquetados como en datos sintéticos bajo supuestos controlados.

2.5.2 Reguladores del Mercado

La discusión regulatoria presentada a continuación tiene como objetivo establecer el marco general bajo el cual la manipulación de mercado es definida, prohibida y sancionada en mercados financieros organizados. Los casos de Estados Unidos y Chile se utilizan como ejemplos representativos de marcos regulatorios formales, y no pretenden describir de manera específica el contexto institucional del mercado a partir del cual provienen los datos analizados en esta memoria.

Bajo este alcance general, la regulación financiera se justifica por la necesidad de corregir fallos de mercado, como la asimetría de información. El objetivo fundamental de un regulador como la *Securities and Exchange Commission* (SEC) en Estados Unidos o la Comisión para el Mercado Financiero (CMF) en Chile es garantizar la integridad, eficiencia y transparencia del mercado.

La regulación busca promover la competencia y la equidad en la negociación, restringir el insider trading y prevenir que los emisores defrauden a los inversores ocultando información [15]. Para ello, establecen un marco normativo sancionatorio, utilizando sistemas de vigilancia cada vez más basados

en algoritmos y análisis de datos para detectar patrones sospechosos.

Tanto Estados Unidos como Chile han establecido marcos legales robustos, aunque con diferencias en su especificidad y alcance.

- Estados Unidos (SEC): La regulación estadounidense, materializada en la *Securities Exchange Act* de 1934, confiere a la Securities and Exchange Commission (SEC) la autoridad para supervisar los mercados, prohibiendo explícitamente la manipulación. Su marco normativo es extenso y detallado, con reglas específicas contra prácticas como *insider trading, wash trades* y *spoofing*. El marco sancionatorio es severo, incluyendo multas millonarias y penas de prisión. La SEC ha sido pionera en el uso de tecnología para la fiscalización, empleando sistemas de vigilancia algorítmica basados en análisis de datos y reconocimiento de patrones.
- Chile: La Ley N°18.045 de Mercado de Valores es el cuerpo normativo principal. El Artículo 52 define y prohíbe la manipulación de precios como toda acción destinada a *"estabilizar, fijar o hacer variar artificialmente los precios de valores de oferta pública"*. El Artículo 53 prohíbe las transacciones ficticias. La Comisión para el Mercado Financiero (CMF) tiene facultades para supervisar, investigar y sancionar. El marco sancionatorio incluye multas administrativas y la posibilidad de acciones penales que pueden resultar en penas de presidio, según los artículos 59 y 60.

Ambos países prohíben la manipulación y el uso de información privilegiada. Ambos reguladores exigen altos estándares de divulgación de información. Sin embargo, la regulación de la SEC ha sido históricamente más granular y prescriptiva. La jurisprudencia de EE.UU. es más abundante, y los recursos tecnológicos de la SEC para la vigilancia algorítmica han sido tradicionalmente superiores[40].

2.6 Detección Algorítmica

La elevada frecuencia y el volumen de transacciones en los mercados financieros modernos limitan la viabilidad de una vigilancia manual continua. En este contexto, las estrategias de manipulación de mercado, en particular aquellas de tipo *trade-based*, pueden generar distorsiones observables en la microestructura del mercado, lo que motiva el uso de métodos de detección algorítmica.

Desde un enfoque cuantitativo, la manipulación de mercado puede abordarse como un problema de detección de anomalías en series temporales financieras. En ausencia de intervenciones externas, los log-precios presentan un comportamiento cercano a un paseo aleatorio y los retornos exhiben propiedades estadísticas relativamente estables. La manipulación introduce desviaciones sistemáticas respecto de este comportamiento de referencia.

En consecuencia, la detección de manipulación se formula como un problema de reconocimiento de patrones. Los enfoques econométricos paramétricos con supuestos de estacionariedad global pueden resultar poco flexibles para identificar patrones transitorios y localizados en el tiempo, característicos de esquemas como el *pump and dump*.

Esto motiva el uso de modelos de aprendizaje automático supervisado, capaces de identificar fronteras de decisión no lineales a partir de variables financieras observables. En este trabajo, dichas variables incluyen precios, retornos y medidas de liquidez, tales como el volumen transado o la profundidad del libro de órdenes.

El análisis combina un estudio de caso con un activo real y experimentos controlados sobre datos sintéticos. El uso de un único activo real responde a la escasez de bases de datos públicas con etiquetas verificadas de manipulación [22]. Los datos sintéticos se generan mediante modelos estocásticos con el objetivo de evaluar, bajo supuestos controlados, la capacidad de distintos enfoques de clasificación para identificar patrones anómalos.

Capítulo 3

Herramientas matemáticas, estadísticas y computacionales

Las herramientas presentadas en este capítulo se utilizan para formular y resolver el problema de detección de manipulación de mercado definido en el capítulo anterior, a partir de variables observables como precios y volúmenes.

3.1 Aprendizaje automático

Como se estableció en el capítulo anterior, los mercados financieros generan series de tiempo de precios y variables de negociación donde episodios de manipulación pueden manifestarse como patrones anómalos. En este trabajo, el aprendizaje automático se utiliza como una herramienta para aprender una función de clasificación que asocia un conjunto de variables observables a una etiqueta que indica la presencia o ausencia de manipulación a partir de variables observables y clasificar intervalos temporales etiquetados como manipuladas o no manipuladas.

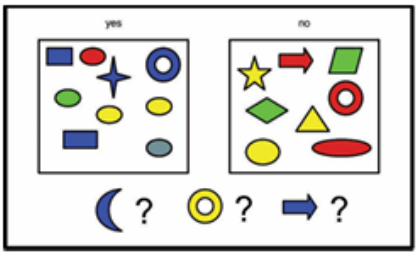
Según Murphy [32], el aprendizaje automático estudia métodos que identifican patrones en los datos para realizar predicciones o tomar decisiones a partir de observaciones.

El aprendizaje automático está categorizado en dos tipos principales. En primer lugar, está el **aprendizaje supervisado**, que se encuentra centrado en aprender relaciones entre las n-variables de entrada $\mathbf{x}$ y la variable de salida y. Esto se logra utilizando un conjunto de datos etiquetados, denotados como $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^{N}$, donde $\mathcal{D}$ es llamado el conjunto de entrenamiento, N es el número de observaciones y el subíndice i denota la i-ésima observación del conjunto de datos. Cada entrada de entrenamiento $\mathbf{x}_i$ representa un vector de n-variables, mientras que cada salida y_i puede ser una variable categórica correspondiente a un conjunto finito, $y_i \in \{1, \ldots, C\}$, o bien una variable numérica. Cuando y_i es categórica, el problema es llamado de clasificación. Por otro lado, cuando y_i es un valor real, el problema es llamado de regresión.

En la figura 3.1 se observa un ejemplo de problema de aprendizaje supervisado, donde se representa un conjunto de datos de entrenamiento de $N \times D$. Cada fila representa un vector de características $\mathbf{x}_i$. La última columna es la etiqueta, $y_i \in \{0, 1\}$

En este contexto, una característica (o *feature*) corresponde a una variable cuantitativa utilizada como entrada por los modelos de aprendizaje automático. En este trabajo, cada observación $\mathbf{x}_i$ representa un vector de características construido a partir de los datos financieros originales, donde cada componente captura información relevante sobre el comportamiento del activo en un período determinado, como retornos, volúmenes o medidas de volatilidad. Esta representación permite transformar series temporales financieras en una forma adecuada para su procesamiento mediante modelos de clasificación.

La segunda categoría de aprendizaje automático es el **aprendizaje no supervisado**. Aquí solo tenemos las entradas, $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$ y nuestro objetivo es encontrar patrones de interés en nuestro conjunto de datos. En contraste con el anterior, este es un problema menos bien definido, dado que no sabemos que tipo de patrones estaremos viendo, además, no existe una métrica de error directa para evaluar que tan bien rinde el modelo. En la Figura 3.1 se observa un problema de aprendizaje no supervisado, con un conjunto de entrenamiento de formas coloreadas, con 3 casos sin clasificar.



D features (attributes)

N cases

Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

Figura 3.1: Figura recuperada de [32].

h

3.1.1 Problema de clasificación

La detección de manipulación de mercado se formula como un problema de clasificación binaria. Cada observación se representa mediante un vector de características $\mathbf{x}_i \in \mathbb{R}^n$, construido a partir de variables financieras observables.

Sea Φ_t el conjunto de información disponible en el instante temporal t. El vector $\mathbf{x}_i$ corresponde a una representación discretizada de Φ_t e incluye, entre otras, las siguientes variables:

- Variables base: precio de cierre $P_{a,i}$ y volumen transado $V_{a,i}$.
- Variables derivadas: retornos logarítmicos $r_{a,i}$ y medidas de volatilidad $\sigma_{a,i,k}$.

El objetivo consiste en aproximar una función de clasificación $\hat{f} : \mathbb{R}^n \to \{0, 1\}$ tal que, dado un vector $\mathbf{x}_i$, permita identificar si la observación corresponde a una dinámica de mercado no manipulada $(y_i = 0)$ o a un escenario asociado a manipulación de mercado $(y_i = 1)$.

La variable objetivo y_i se define de la siguiente forma:

$$y_i = \begin{cases} 0, & \text{si la observación no se encuentra manipulada,} \\ 1, & \text{si la observación se encuentra manipulada.} \end{cases} \tag{3.1}$$

Las variables consideradas dependen del conjunto de datos y del modelo. En particular, además de precios y volúmenes, se incorporan variables de liquidez, como la profundidad bid/ask, cuando esta información se encuentra disponible.

El objetivo es discriminar entre observaciones asociadas a un comportamiento normal del mercado y aquellas compatibles con manipulación. Esta formulación es coherente con el enfoque empírico del trabajo, dado que las bases de datos utilizadas cuentan con etiquetas discretas que indican la presencia o ausencia de manipulación en cada período.

Alternativas como el aprendizaje no supervisado no se ajustan directamente a este contexto, ya que no responden al objetivo de identificar eventos etiquetados ni permiten evaluar el desempeño del modelo respecto de un criterio de referencia bien definido.

3.1.2 Protocolo de Validación

Un aspecto central al entrenar clasificadores es evitar el sobreajuste. El sobreajuste ocurre cuando el modelo aprende patrones específicos del conjunto de entrenamiento, incluyendo ruido o regularidades accidentales, que no se mantienen en datos no observados. En ese caso, el desempeño evaluado sobre el propio conjunto de entrenamiento (por ejemplo, medido mediante métricas de clasificación) puede ser alto, pero el rendimiento disminuye al evaluar con observaciones nuevas. Por esta razón, el desempeño de los modelos se analiza mediante evaluación sobre datos no vistos y métricas que reflejan su capacidad de generalización, entendida como la capacidad del modelo para mantener su rendimiento cuando se aplica a datos que no fueron utilizados en su ajuste, especialmente en un escenario desbalanceado como el de manipulación.

Para evaluar la capacidad de generalización, los datos se dividen en un conjunto de entrenamiento (train), usado para ajustar los parámetros del modelo, y un conjunto de prueba (test), que se mantiene separado y se utiliza solo para estimar el desempeño en datos no vistos. Esta separación permite detectar sobreajuste y comparar modelos de manera imparcial.

Cuando las observaciones provienen de ventanas temporales potencialmente correlacionadas, la partición se realiza a nivel de unidad temporal (por ejemplo, por día) para evitar dependencia entre train y test. La selección de hiperparámetros se realiza mediante validación cruzada estratificada dentro del conjunto de entrenamiento. En el enfoque funcional, el ajuste de la base/suavizado y de FPCA se realiza únicamente con curvas del entrenamiento, y luego se proyecta el conjunto de prueba utilizando esos mismos objetos ajustados.

La validación cruzada estratificada es una variante de la validación cruzada en la que cada partición (fold) conserva, lo mejor posible, la misma proporción de clases que existe en el conjunto de datos original.

En un problema de clasificación, esto significa que si, por ejemplo, el 30 % de las observaciones pertenece a la clase minoritaria y el 70 % a la mayoritaria, cada fold tendrá aproximadamente ese mismo 30/70. El objetivo es evitar que algunos folds queden desbalanceados o incluso sin observaciones de una clase, lo que produciría estimaciones sesgadas o inestables del rendimiento del modelo.

3.2 Algoritmos de clasificación

Los algoritmos de clasificación buscan aproximar f para asignar una etiqueta de clase a nuevas observaciones a partir de sus características, aprendiendo una regla de decisión a partir de datos etiquetados.

Es importante notar que los algoritmos considerados no se seleccionan con el objetivo de identificar un modelo óptimo en abstracto, sino para comparar distintos enfoques de clasificación que representan compromisos diferentes entre interpretabilidad, flexibilidad y capacidad para manejar desbalance severo.

3.2.1 Árboles de Regresión y Clasificación

Los árboles de regresión y clasificación o también conocidos como modelos CART (por sus siglas en inglés Classification and regression trees), también llamados árboles de decisión, se definen particionando recursivamente el espacio de entrada y definiendo un modelo local en cada región resultante de dicho espacio. Esto puede representarse mediante un árbol, con una hoja por cada región.

Este algoritmo fue propuesto por Breiman et al. [6], es un método no paramétrico y no lineal para entrenar árboles de decisión. Estos se basan en la partición recursiva del espacio de características en un conjunto de regiones rectangulares, ajustando un modelo simple en cada una de ellas.

Este generará exclusivamente árboles binarios, donde cada nodo interno se divide en exactamente dos nodos hijos, utilizando una pregunta de "si/no".

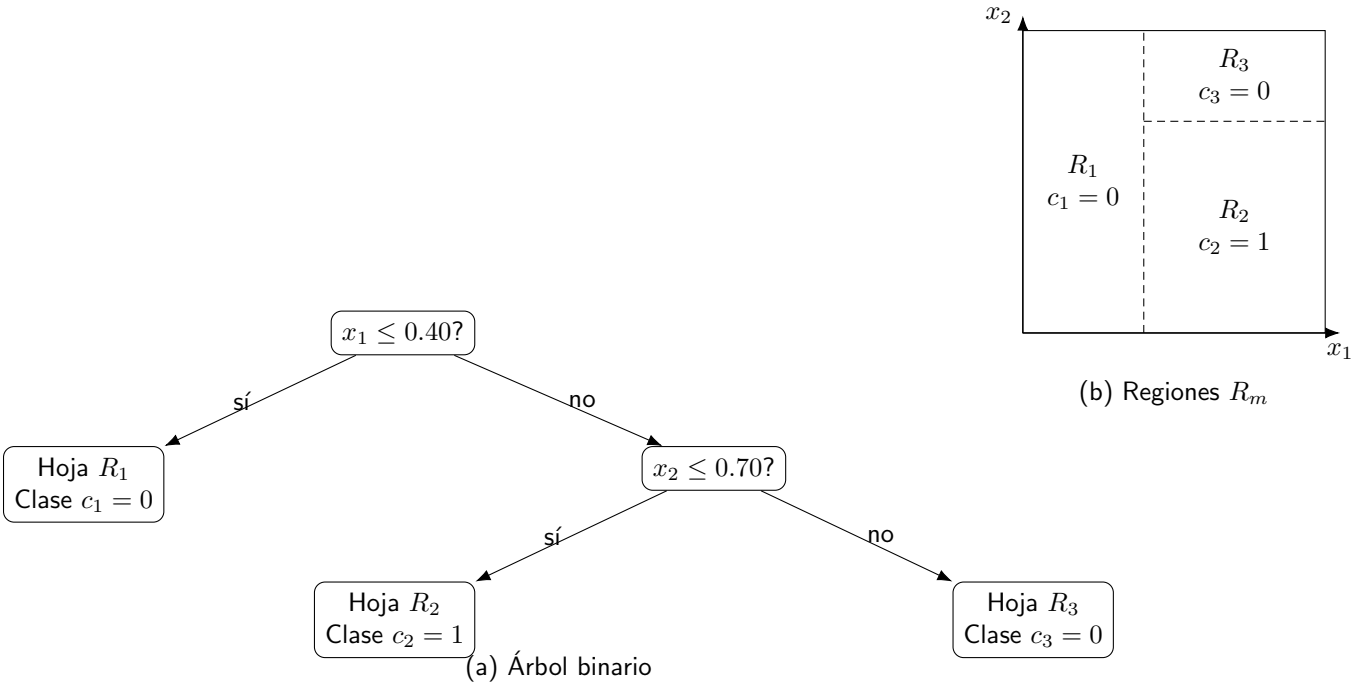
El modelo final $\hat{f}(\mathbf{x})$ particiona el espacio de características $\mathcal{X} \subseteq \mathbb{R}^l$ en M regiones rectangulares disjuntas, $R_1, R_2, \ldots, R_M$, tal que $\mathcal{X} = \bigcup_{m=1}^{M} R_m$.

La función predictiva $\hat{f}(\mathbf{x})$ se define como la clase dominante c_m en la región R_m a la que pertenece $\mathbf{x}$:

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} c_m I(\mathbf{x} \in R_m),\ c_m \in \{0, 1\} \tag{3.2}$$

donde c_m es la clase mayoritaria en la región terminal R_m, y $I(\cdot)$ es la función indicatriz.

Podemos ver la Figura 3.2 como ejemplo, donde el árbol (a) aplica dos reglas binarias: primero separa por $x_1 \leq 0.40$ y, si $x_1 > 0.40$, separa por $x_2 \leq 0.70$. Eso induce la partición (b) en tres regiones rectangulares disjuntas R_1, R_2, R_3. En cada región terminal, la predicción es constante e igual a la clase mayoritaria c_m, como en $\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} c_m\, I(\mathbf{x} \in R_m)$.



(a) Árbol binario

(b) Regiones R_m

Figura 3.2: Ejemplo de CART con dos variables (x_1, x_2).

3.2.2 Bosques Aleatorios

Los Bosques Aleatorios (Random Forest) es un método de *ensemble learning* que combina las predicciones de múltiples árboles de decisión para producir una predicción final Breiman [5]. Esta técnica se basa en dos mecanismos clave para aumentar la diversidad del ensemble: Bagging (Bootstrap Aggregating) y la selección de características aleatoria en cada división de nodo.

Los Bosques Aleatorios, $\hat{H}(\mathbf{x})$, es un predictor agregado que resulta de la combinación de B árboles de decisión individuales, $T_b(\mathbf{x})$, donde cada árbol $b \in \{1, \ldots B\}$ se entrena sobre una muestra bootstrap Z^{*b} y utilizando un subconjunto aleatorio de características en cada división de nodo.

Sea $\mathcal{D}$ el conjunto de entrenamiento, y n el número de características. La función de predicción final del conjunto, $\hat{H}(\mathbf{x})$, se obtiene por votación mayoritaria:

$$\hat{H}(\mathbf{x}) = \underset{c \in \{0,1\}}{\arg\max} \sum_{b=1}^{B} I(T_b(\mathbf{x}) = c), \tag{3.3}$$

donde $I(\cdot)$ denota la función indicatriz.

El procedimiento de construcción del Bosque Aleatorio es el siguiente.

La construcción de cada árbol T_b sigue un proceso recursivo que garantiza su baja correlación con el resto del bosque aleatorio. El algoritmo se basa en la metodología CART.

- Paso 1: Muestreo Bootstrap (Bagging)

  Para cada árbol b, se genera un subconjunto de entrenamiento Z^{*b} del tamaño N mediante muestreo con reemplazo del conjunto de datos original $\mathcal{D}$. Este paso asegura cada árbol se entrene sobre una muestra ligeramente diferente, lo que ayuda a reducir la varianza y mitiga el sobreajuste.

- Paso 2: Crecimiento de Árboles con aleatoriedad de características

  Cada árbol T_b hace crecer completamente sobre su muestra Z^{*b}. La clave reside en la aleatoriedad introducida en cada división de nodo:

  1. Selección Aleatoria de Características: En cada nodo j del árbol T_b, solo se consideran m características seleccionadas aleatoriamente del total de n características disponibles, donde $m \leq n$. Para el problema de clasificación, el valor común para m es $m = \lfloor \sqrt{n} \rfloor$.
  2. Búsqueda de la Mejor División: El algoritmo busca la mejor característica k^* y el mejor punto de división t_k^* dentro de las m características seleccionadas que minimicen la función de costo o impureza.

- Paso 3: Criterio de Impureza para Clasificación Binaria:

  Para un problema de clasificación binaria, la medida de impureza más utilizada es el Índice de Gini. Sea N_j el número de muestras que caen en el nodo j, y sea $p_{j,k}$ la proporción de muestras de la clase $k \in \{0, 1\}$ en el nodo j. El Índice de Gini G_j para el nodo j se define como:

  $$G_j = 1 - \sum_{k=0}^{1} p_{j,k}^2 \tag{3.4}$$

  El valor de $p_{j,k}$ corresponde a la proporción de observaciones de la clase k entre todas las observaciones de entrenamiento nodo j. Si un nodo es puro (todas las instancias pertenecen a la misma clase), entonces $G_j = 0$.

  El algoritmo selecciona la característica k^* y el umbral t_k^* que resultan en la mayor reducción de impureza, lo cual equivale a minimizar la impureza de Gini ponderada de los nodos hijos resultantes, R_L (izquierdo) y R_R (derecho):

  $$\underset{k,t_k}{\text{mín}} \left(\frac{N_L}{N_j} G_L + \frac{N_R}{N_j} G_R \right) \tag{3.5}$$

  Donde N_L y N_R son los tamaños de las muestras en los nodos hijos, respectivamente.

Podemos ver en la Figura 3.3, donde cada árbol T_b se entrena de forma independiente sobre una muestra bootstrap distinta y produce una predicción individual. La salida del bosque se obtiene mediante votación mayoritaria entre los árboles. En este ejemplo, dos de los tres árboles predicen la clase 1, por lo que la predicción final del conjunto es $\hat{H}(\mathbf{x}) = 1$.

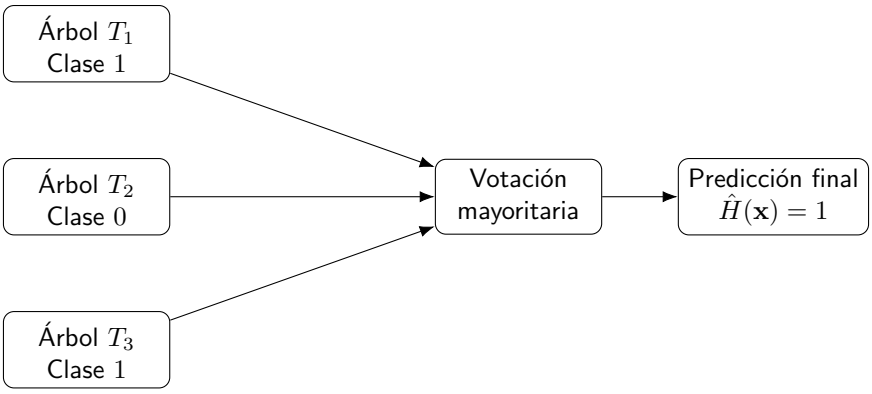


Figura 3.3: Ejemplo de un Bosque Aleatorio para clasificación binaria.

3.2.3 Naïve Bayes

El clasificador de Naïve Bayes es un algoritmo probabilístico basado en el Teorema de Bayes. Este teorema describe la probabilidad de un evento basándose en el conocimiento previo de condiciones que podrían estar relacionadas con el evento. La fórmula del Teorema de Bayes:

$$\mathbb{P}(y = c|\mathbf{x}) = \frac{\mathbb{P}(\mathbf{x}|y = c)\mathbb{P}(y = c)}{\mathbb{P}(\mathbf{x})} \tag{3.6}$$

Donde para el caso de clasificación binaria, $c \in \{0, 1\}$, y Además,

- $\mathbb{P}(y = c|\mathbf{x})$ la probabilidad de la clase $y = c$ dadas las características $\mathbf{x}$, o también conocida como la probabilidad a posteriori.
- $\mathbb{P}(\mathbf{x}|y = c)$ es la verosimilitud de observar el vector de características $\mathbf{x}$ dado que la variable objetivo es c.
- $\mathbb{P}(y = c)$ es la probabilidad a priori de que la variable objetivo sea c.
- $\mathbb{P}(\mathbf{x})$ es la probabilidad marginal de observar el vector de características $\mathbf{x}$.

Así, el clasificador Naïve Bayes consiste en asignar una muestra a la clase que maximice la probabilidad posterior. Para una clasificación binaria, la regla es:

$$\hat{f(x)} = \underset{c\in\{0,1\}}{\arg\max} \left(\mathbb{P}(\mathbf{x}|y = c)\mathbb{P}(y = c)\right) \tag{3.7}$$

Por otro lado, el clasificador asume la independencia condicional entre las características, dada la clase. Matemáticamente, esto se expresa como:

$$\mathbb{P}(\mathbf{x}|y = c) = \prod_{i=1}^{n} \mathbb{P}(x_i|y = c) \tag{3.8}$$

Este supuesto se denomina "ingenuo" porque en mercados financieros las variables como precio, volumen y volatilidad rara vez son condicionalmente independientes. No obstante, pese a esta simplificación, el modelo puede resultar competitivo en ciertos problemas prácticos, especialmente cuando la dimensionalidad es alta y se requiere un clasificador de referencia de baja complejidad. Aplicando esto a 3.7, obtenemos la siguiente estimación para $\hat{y}$:

$$\hat{y} = \underset{c\in\{0,1\}}{\arg\max} \left(\mathbb{P}(y = c) \prod_{i=1}^{n} \mathbb{P}(x_i|y = c)\right) \tag{3.9}$$

Los términos $\mathbb{P}(y = c)$ y $\mathbb{P}(x_i|y = c)$ deben ser estimados a partir de un conjunto de datos de entrenamiento etiquetados. Generalmente para las verosimilitudes condicionales se utiliza la distribución gaussiana:

$$\mathbb{P}(x_i|y = c) = \frac{1}{\sqrt{2\pi\sigma_{ic}^2}} e^{\left(-\frac{(x_i-\mu_{ic})^2}{2\sigma_{ic}^2}\right)} \tag{3.10}$$

Donde se suele estimar la media (μ_{ic}) y la varianza (σ_{ic}^2) para cada característica i y cada clase $c \in \{0, 1\}$ a partir de los datos de entrenamiento.

3.2.4 Máquinas de Vectores de Soporte

Las Máquinas de Vectores de Soporte (Support Vector Machine, SVM), propuesta por Cortes y Vapnik [12], es un clasificador binario que busca determinar un hiperplano que separe dos clases en el espacio de características maximizando el margen entre ellas. En esta sección, para la formulación matemática se re-etiquetan las clases como $y_i \in \{-1, 1\}$[1].

Formalmente, dado un conjunto de entrenamiento $\mathcal{D}$, el hiperplano separador se define mediante el vector de peso $\mathbf{w}$ y el término de sesgo b. El problema de optimización primal para el caso linealmente separable se establece como la minimización de la norma del vector de pesos, que es inversamente proporcional al margen:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2$$

sujeto a las restricciones que aseguran la correcta clasificación de todos los puntos con un margen de al menos 1:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \quad i = 1, \ldots, N$$

Sin embargo, dado que los datos financieros, como las series temporales de precios o volúmenes, raramente son perfectamente separables debido a la naturaleza estocástica y ruidosa de los mercados, la formulación de margen estricto resulta excesivamente restrictiva y vulnerable a valores atípicos. Para abordar la superposición de clases y la presencia de "ruido" inherente de las series financieras, se introducen variables de holgura $\xi_i \geq 0$, reguladas por el hiperparámetro C, que equilibra la maximización del margen con la penalización de errores de clasificación.

$$\min_{\mathbf{w},b,\xi} \quad \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{N}\xi_i \tag{3.11}$$

$$\text{s.t.} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad i = 1, \ldots, N \tag{3.12}$$

Para capturar las relaciones intrínsecamente no lineales y complejas presentes en los datos de curvas de precios y volúmenes, la SVM emplea el truco del kernel. Este mecanismo permite mapear implícitamente los datos originales a un espacio de características de alta o incluso de dimensión infinita, $\mathcal{H}$, mediante la función $\phi(\cdot)$, donde la separación lineal puede ser factible. Al trabajar con la formulación dual del problema de optimización, el cálculo directo de $\phi(\mathbf{x})$ se evita, sustituyendo todos los productos internos $\langle \mathbf{x}_i, \mathbf{x}_j \rangle$ por una función de kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$. En este trabajo utilizaremos la función kernel radial (*RBF*) definida como

[1] Para la formulación matemática de SVM, re-etiquetamos las clases tal que $y_i \in \{-1, 1\}$.

$$K(\mathbf{x_i}, \mathbf{x_j}) = \exp\left(-\gamma \left\|\mathbf{x_i} - \mathbf{x_j}\right\|_2^2\right), \gamma > 0.$$

Así, la SVM puede capturar relaciones no lineales a través de $K(\mathbf{x}_i, \mathbf{x}_j)$ sin construir explícitamente el mapeo $\phi(\cdot)$.

3.2.5 K-Vecinos más Cercanos

El algoritmo de *K*-Vecinos Más Cercanos (*K*-Nearest Neighbors, KNN), introducido por Fix y Hodges [19], se clasifica como un método no paramétrico y basado en instancias, también denominado algoritmo de *aprendizaje perezoso* (*lazy learning*). Su rasgo distintivo consiste en que no construye un modelo funcional explícito durante la fase de entrenamiento, sino que almacena íntegramente el conjunto de observaciones disponibles. En consecuencia, la mayor carga computacional del método se concentra en la etapa de predicción, donde la clasificación de una nueva observación se realiza mediante la consulta directa al conjunto de entrenamiento.

Sea $\mathcal{D} = (\mathbf{x}_i, y_i)_{i=1}^N$ el conjunto de entrenamiento, donde $\mathbf{x}_i \in \mathbb{R}^n$ representa el vector de características de la i-ésima observación y $y_i \in \mathcal{C}$ su correspondiente etiqueta de clase, con $\mathcal{C} = \{0, 1\}$ en el caso de clasificación binaria. Dada una nueva observación $\mathbf{x}_q \in \mathbb{R}^n$, el algoritmo identifica el conjunto de sus K vecinos más cercanos, denotado por $\mathcal{N}_K(\mathbf{x}_q)$, mediante la evaluación de una métrica de distancia definida en el espacio de características.

En este trabajo se utilizaran dos métricas de distancia: la distancia euclídea, correspondiente a la norma ℓ_2, definida por

$$d(\mathbf{x}_i, \mathbf{x}_q) = \|\mathbf{x}_i - \mathbf{x}_q\|_2 = \sqrt{\sum_{j=1}^{n} (x_{ij} - x_{qj})^2}. \tag{3.13}$$

Y, la distancia uniforme, asociada a la norma ℓ_∞, definida como

$$d_\infty(\mathbf{x}_i, \mathbf{x}_q) = \|\mathbf{x}_i - \mathbf{x}_q\|_\infty = \max_{1 \le j \le n} |x_{ij} - x_{qj}|. \tag{3.14}$$

Una vez determinado el vecindario $\mathcal{N}_K(\mathbf{x}_q)$, la predicción de la etiqueta asociada a $\mathbf{x}_q$ se obtiene mediante un mecanismo de votación mayoritaria. De manera general, la regla de clasificación se define como

$$\hat{f}(x_q) = \arg\max_{c \in \mathcal{C}} \sum_{i \in \mathcal{N}_K(\mathbf{x}_q)} I(y_i = c). \tag{3.15}$$

En el contexto específico de esta tesis, orientada a la detección de manipulación en mercados financieros, se considera el problema de clasificación binaria, donde $y_i = 1$ representa una observación manipulada y $y_i = 0$ un comportamiento normal del mercado. Bajo esta formulación, el clasificador KNN puede interpretarse como un estimador empírico de la probabilidad condicional

$$\mathbb{P}(Y = 1 \mid X = \mathbf{x}_q), \tag{3.16}$$

aproximada por la fracción de vecinos pertenecientes a la clase positiva dentro del vecindario $\mathcal{N}_K(\mathbf{x}_q)$. En consecuencia, la regla de decisión puede expresarse como

$$\hat{y} = I\left(\frac{1}{K} \sum_{i \in \mathcal{N}_K(\mathbf{x}_q)} y_i > \tau\right), \tag{3.17}$$

donde $\tau \in (0, 1)$ es un umbral de decisión, que en la práctica suele fijarse en $\tau = 0.5$.

La elección del hiperparámetro K resulta crucial, ya que determina el compromiso fundamental entre sesgo y varianza del estimador. Valores pequeños de K generan fronteras de decisión altamente irregulares y sensibles al ruido presente en los datos financieros, incrementando la varianza y el riesgo de sobreajuste. Por el contrario, valores grandes de K inducen un suavizado excesivo de la frontera de decisión, lo que incrementa el sesgo y reduce la capacidad del modelo para detectar patrones locales y anómalos característicos de esquemas de manipulación.

En la Figura 3.4 podemos ver un ejemplo del algoritmo, donde el punto de consulta $\mathbf{x}_q$ (estrella) se clasifica identificando sus K vecinos más cercanos según la distancia euclídea, representados de forma esquemática por el círculo punteado. La etiqueta asignada corresponde a la clase mayoritaria dentro del vecindario.

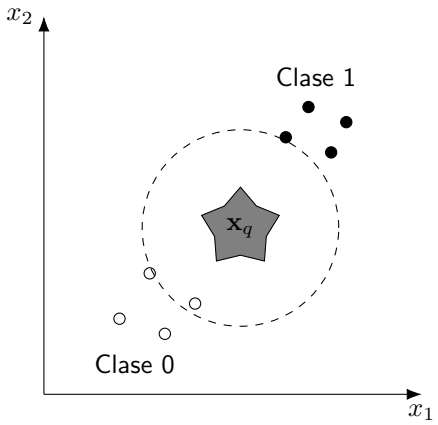


Figura 3.4: Ejemplo del algoritmo KNN con $K = 5$ en un espacio bidimensional.

3.2.6 Redes Neuronales Artificiales

La Red Neuronal Artificial (ANN), en su arquitectura de Perceptrón Multicapa (Multilayer Perceptron, MLP), es un modelo de aprendizaje automático basado en la composición de transformaciones afines y funciones de activación no lineales. Esta estructura permite representar relaciones no lineales entre variables de entrada y una salida, lo que resulta pertinente para tareas de clasificación en datos financieros.

La arquitectura del MLP consiste en una capa de entrada, una o más capas ocultas compuestas por unidades (neuronas) de procesamiento y una capa de salida, configurándose como un clasificador o regresor de dos etapas. El proceso de transformación de la información a través de la red se denomina propagación hacia adelante (forward propagation), donde la entrada de una capa previa se transforma secuencialmente en la activación de la capa siguiente. Formalmente, la activación de una neurona en una capa oculta l, denotada por el vector de activaciones $\mathbf{a}^{(l)}$, se calcula como una composición de una transformación afín y una función de activación no lineal ϕ, tal que:

$$\mathbf{a}^{(l)} = \phi(\mathbf{W}^{(l)}\mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}) \tag{3.18}$$

En lo que sigue, $\mathbf{W}^{(l)}$ y $\mathbf{b}^{(l)}$ denotan, respectivamente, los pesos y sesgos de la capa l, y $\phi(\cdot)$ es una activación no lineal aplicada componente a componente.

Donde $\mathbf{a}^{(l-1)}$ representa el vector de activaciones de la capa anterior (o la capa de entrada $\mathbf{X}$ si $l = 1$), $\mathbf{W}^{(l)}$ es la matriz de pesos que conecta la capa anterior con la capa l, y $\mathbf{b}^{(l)}$ es el vector de sesgos (bias) asociado a esa capa,. Es crucial que $\phi(\cdot)$ sea una función no lineal, como la Unidad de Activación Lineal Rectificada (Rectified Linear Unit, ReLU), definida como $\phi(\mathbf{x}) = \max(\mathbf{x}, 0)$, o la

tangente hiperbólica, definida como $\phi(\mathbf{x}) = \tanh(\mathbf{x})$, ya que es esta no-linealidad la que permite a la red modelar interacciones complejas en los datos de entrada.

Para el propósito de la clasificación binaria, donde se busca distinguir entre el comportamiento de mercado normal y el manipulado, la capa de salida del MLP se diseña con una única neurona que emplea la función logística o sigmoide, $\sigma(z) = 1/(1+\exp(-z))$. El valor de salida, $\hat{y}$, representa la probabilidad estimada de que una instancia dada pertenezca a la clase positiva (manipulación):

$$\hat{y} = \sigma(\mathbf{z}_{out}) \tag{3.19}$$

El entrenamiento del MLP se rige por el principio de minimización del riesgo empírico (Empirical Risk Minimization, ERM), ajustando los parámetros internos (pesos $\mathbf{W}$ y sesgos $\mathbf{b}$) para reducir el error de predicción sobre el conjunto de entrenamiento. Dada la naturaleza probabilística de la salida para la clasificación binaria, la función de pérdida típicamente empleada es la entropía cruzada binaria (Binary Cross-Entropy) o log-pérdida (log loss), si bien la entropía cruzada binaria es convexa respecto a la salida del modelo, el problema de optimización global del MLP es no convexo debido a la composición de múltiples capas no lineales. La minimización de esta función de coste se lleva a cabo de forma iterativa mediante el algoritmo de optimización de Descenso de Gradiente (Gradient Descent, GD). La eficiencia en el cálculo de los gradientes para todas las capas se logra a través del algoritmo de retropropagación (backpropagation), que aplica la regla de la cadena de forma recursiva desde la capa de salida hasta la capa de entrada.

En la Figura 3.5 podemos ver un ejemplo de MLP, donde la entrada $\mathbf{x}$ se transforma mediante una capa oculta aplicando una combinación afín (pesos $\mathbf{W}$ y sesgos $\mathbf{b}$) seguida de una activación no lineal ϕ (p.ej. ReLU). La neurona de salida usa sigmoide σ para entregar $\hat{y} \in [0, 1]$ como probabilidad estimada de la clase positiva. En el entrenamiento, la propagación hacia adelante calcula $\hat{y}$ y la retropropagación (línea punteada) actualiza los parámetros minimizando la pérdida mediante descenso de gradiente.

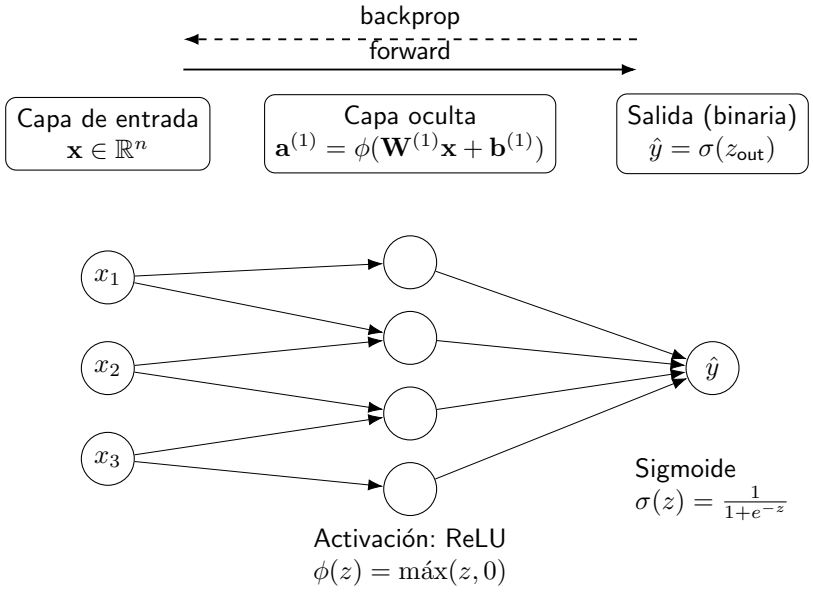


Figura 3.5: Esquema de un Perceptrón Multicapa (MLP) para clasificación binaria.

3.3 Métricas de evaluación de modelos

La evaluación del desempeño de un clasificador binario es un componente fundamental en la detección de manipulación en los mercados financieros, ya que permite discriminar entre un comportamiento financiero normal y uno que presente prácticas manipulativas. En este contexto, dicha evaluación se sustenta en un conjunto de métricas cuantitativas que permiten medir, desde distintas perspectivas, el desempeño y la capacidad discriminativa de los modelos propuestos.

3.3.1 Matriz de Confusión

La **matriz de confusión** tabulariza los cuatro resultados posibles de la clasificación, contrastando las etiquetas predichas con las etiquetas reales . En el contexto de la detección de manipulación de mercado, donde se busca identificar anomalías, estos resultados se interpretan de la siguiente manera: un Verdadero Positivo (TP) ocurre cuando una instancia de manipulación (clase positiva) es clasificada correctamente; un Falso Negativo (FN) representa una manipulación real que el modelo no detecta, clasificándola incorrectamente como normal; un Verdadero Negativo (TN) indica que una operación normal (clase negativa) ha sido clasificada correctamente; y finalmente, un Falso Positivo (FP) constituye una falsa alarma, donde una operación normal es clasificada erróneamente como manipulada.

Formalmente, la matriz de confusión puede representarse mediante la Tabla 3.1, donde se cruzan las etiquetas reales con las etiquetas predichas por el modelo.

Etiqueta Real	**Predicción: Normal**	**Predicción: Manipulación**
Normal	TN	FP
Manipulación	FN	TP

Cuadro 3.1: Matriz de confusión para un clasificador binario en detección de manipulación de mercado

3.3.2 Exactitud

Si bien la Exactitud (*Accuracy*) es una métrica común para evaluar modelos de clasificación binaria, definida como la proporción de clasificaciones correctas sobre el total de predicciones, su aplicación en la detección de manipulación en mercados puede ser engañosa. La Exactitud se calcula formalmente como:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.20}$$

El principal problema con la Exactitud radica en la naturaleza desbalanceada de los datos de mercado en este dominio. Dado que las actividades de manipulación son eventos raros (clase positiva minoritaria), un clasificador que sistemáticamente prediga que todas las instancias son normales (clase negativa) puede lograr un alto porcentaje de exactitud, pero resulta inútil para la tarea de detección. Además, en la vigilancia de mercados los costes de error son asimétricos: no detectar una manipulación real (Falso Negativo) suele ser más costoso que generar una falsa alarma (Falso Positivo). Por lo tanto, se necesitan métricas que enfaticen el rendimiento del modelo en la identificación de la clase minoritaria (manipulación).

Para lograr una evaluación robusta, se prioriza el uso de métricas que se centran en la proporción de aciertos y errores relativos a la clase positiva.

3.3.3 Precisión

La **Precisión** se define como la razón de verdaderos positivos sobre el total de predicciones positivas realizadas, midiendo la exactitud de las alertas emitidas por el modelo:

$$\text{Precisión} = \frac{TP}{TP + FP} \tag{3.21}$$

Una alta precisión es deseable porque minimiza el número de Falsos Positivos, lo cual es crucial para reducir los costes de investigación regulatoria innecesaria.

3.3.4 Sensibilidad

La **Sensibilidad**, también conocida como *Recall* o Tasa de Verdaderos Positivos (TPR), cuantifica la capacidad del modelo para identificar correctamente todas las instancias de manipulación que realmente ocurrieron:

$$\text{Sensibilidad} = \frac{TP}{TP + FN} \tag{3.22}$$

Maximizar la sensibilidad es vital en la detección de fraudes, ya que los Falsos Negativos (manipulaciones perdidas) son intrínsecamente más problemáticos y costosos. Sin embargo, existe un compromiso inherente entre estas dos métricas, conocido como el *precision–recall tradeoff*, donde generalmente aumentar la precisión tiende a reducir la sensibilidad, y viceversa. Para gestionar este equilibrio, se emplea la métrica general F_β-Score.

3.3.5 F_β-Score

El F_β-Score es la media armónica ponderada de la precisión y la sensibilidad, diseñada para proporcionar una medida única de rendimiento que refleje la importancia relativa asignada a cada métrica a través del parámetro β. La formulación general es:

$$F_\beta = \left(1 + \beta^2\right) \times \frac{\text{Precisión} \cdot \text{Sensibilidad}}{(\beta^2 \text{Precisión}) + \text{Sensibilidad}} \tag{3.23}$$

Cuando $\beta = 1$, se obtiene el $F1$-Score, que trata la precisión y la sensibilidad con igual peso y es una métrica de compromiso estándar. No obstante, debido a la alta sensibilidad a costes de los Falsos Negativos en la detección de manipulación (sensibilidad al coste), se puede optar por un valor de $\beta > 1$ (como el $F2$-Score) para otorgar mayor importancia a la sensibilidad, penalizando fuertemente las omisiones de detección de fraude [22].

3.3.6 Curva ROC

La evaluación de la capacidad discriminativa intrínseca de un modelo de clasificación, independientemente del punto de corte operativo seleccionado, se aborda mediante el análisis de la Curva Característica Operativa del Receptor (Receiver Operating Characteristic, ROC) y su área bajo la curva (Area Under the Curve, AUC). Los algoritmos de clasificación supervisada, no generan directamente una etiqueta de clase fija, sino que producen un puntaje o una probabilidad estimada $\hat{y} \in [0, 1]$ de que una instancia pertenezca a la clase positiva (manipulación). La conversión de esta probabilidad en una etiqueta binaria (manipulado o normal) requiere la aplicación de un umbral de decisión τ [21].

Al variar este umbral τ desde $-\infty$ hasta $+\infty$, se altera el balance entre la Tasa de Verdaderos Positivos (TPR) y la Tasa de Falsos Positivos (FPR). La FPR mide la proporción de instancias

normales que se clasifican incorrectamente como manipuladas. Formalmente, la Tasa de Falsos Positivos se calcula como:

$$FPR = \frac{FP}{FP + TN} \tag{3.24}$$

Es fundamental notar que el FPR es inversamente proporcional a la Especificidad, definida como Especificidad $= \frac{TN}{FP+TN}$, cumpliéndose la relación

$$FPR = 1 - \text{Specificity}. \tag{3.25}$$

La Curva ROC se define entonces como la representación gráfica de la TPR (eje Y) en función de la FPR (eje X) a medida que el umbral de decisión τ es ajustado. Una curva ideal se situaría en la esquina superior izquierda del gráfico como se muestra en la Figura 3.6.

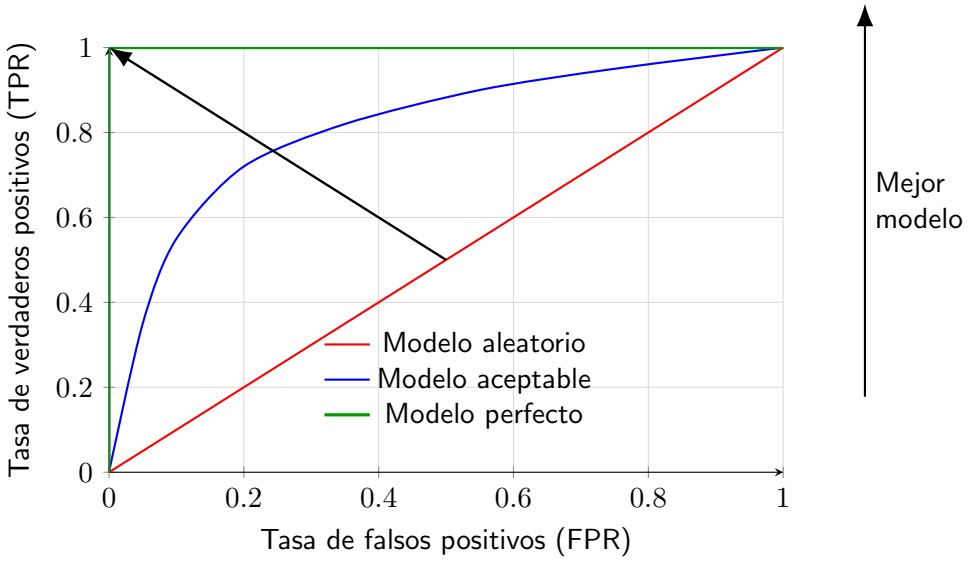


Figura 3.6: Ejemplo de curvas ROC. La curva verde representa un clasificador perfecto (sube verticalmente en $x = 0$ y sigue horizontalmente en $y = 1$), la azul un modelo aceptable y la roja un clasificador aleatorio. La flecha interna indica la dirección de mejora.

3.3.7 Área Bajo la Curva ROC

La calidad global de un clasificador se resume mediante el Área Bajo la Curva (AUC), un valor escalar que cuantifica la superficie total bajo la Curva ROC. Matemáticamente, el AUC se expresa como la integral de la curva ROC:

$$AUC = \int_0^1 TPR(FPR)\, d(FPR) \tag{3.26}$$

El valor del AUC oscila entre 0.5, indicando que el modelo no presenta capacidad discriminativa superior a una elección aleatoria, y 1.0, correspondiente a una clasificación perfecta. El AUC posee una interpretación probabilística directa: es la probabilidad de que el clasificador asigne un puntaje mayor a una observación positiva (manipulación) que a una observación negativa (normal) elegidas al azar; en ese sentido, resume la capacidad discriminativa del modelo como medida global de ordenamiento, independiente del umbral de decisión.

En la Figura 3.7 podemos ver el área sombreada que representa $\int_0^1 TPR(FPR)\, d(FPR)$ (ecuación (3.26)).

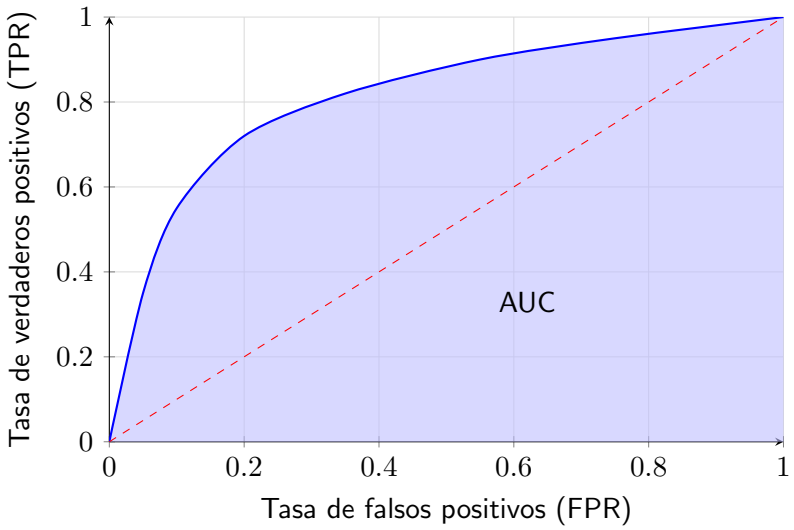


Figura 3.7: Ejemplo del Área Bajo la Curva ROC (AUC).

En este trabajo, el ROC AUC se reporta como una medida global complementaria de capacidad de ordenamiento (ranking) del clasificador. Sin embargo, dado el desbalance severo y la relevancia operativa de la clase positiva, el análisis se centra principalmente en métricas orientadas a la detección de manipulaciones, en particular Sensibilidad, F_2-Score y PR AUC [22, 38]. La Exactitud se reporta solo con fines descriptivos, ya que puede resultar engañosa cuando la clase positiva es rara.

3.3.8 Curva Precision–Recall

En problemas de clasificación binaria con desbalance severo, la Curva *Precision–Recall* (PR) constituye una herramienta más informativa que la Curva ROC, ya que evalúa el desempeño del clasificador enfocándose en la clase positiva. Sea $\hat{s}(\mathbf{x}) \in \mathbb{R}$ un puntaje o $\hat{p}(\mathbf{x}) \in [0, 1]$ una probabilidad estimada para la clase positiva. Para un umbral τ, se define la predicción binaria inducida por umbral como

$$\hat{y}_\tau(\mathbf{x}) = I\left(\hat{p}(\mathbf{x}) > \tau\right), \tag{3.27}$$

El parámetro τ se denomina umbral de decisión y controla el criterio con el cual una observación se asigna a la clase positiva. Valores bajos de τ tienden a clasificar más observaciones como positivas, aumentando la Sensibilidad pero también el número de falsos positivos. En cambio, valores altos de τ producen un comportamiento más conservador, reduciendo falsos positivos a costa de disminuir la Sensibilidad. Al variar τ (por ejemplo, en el intervalo $[0, 1]$ cuando se utilizan probabilidades estimadas), se obtiene una familia de clasificadores derivados del mismo modelo, cada uno asociado a una matriz de confusión distinta.

lo que determina, para cada τ, una matriz de confusión y por tanto valores de Precisión y Sensibilidad dados por

$$\text{Precisión}(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau)}, \qquad \text{Sensibilidad}(\tau) = \frac{TP(\tau)}{TP(\tau) + FN(\tau)}. \tag{3.28}$$

La Curva PR se define como el conjunto de puntos $(\text{Sensibilidad}(\tau), \text{Precisión}(\tau))$ al variar τ en un rango suficientemente amplio.

Un aspecto clave en contextos desbalanceados es que el nivel base de Precisión (clasificador aleatorio sin capacidad discriminativa) coincide con la prevalencia de la clase positiva,

$$\pi = \mathbb{P}(Y = 1) \approx \frac{N_1}{N}, \tag{3.29}$$

por lo que valores de Precisión persistentemente cercanos a π indican escasa utilidad práctica, aun cuando el ROC AUC pueda parecer elevado.

En la Figura 3.8 podemos ver la curva azul que se obtiene al variar el umbral τ y graficar Precisión vs. Sensibilidad. En escenarios desbalanceados, el desempeño base de un clasificador aleatorio corresponde a una Precisión igual a la prevalencia $\pi = \mathbb{P}(Y = 1)$ (línea roja horizontal). Curvas que se mantienen significativamente por sobre π indican utilidad práctica para detectar la clase positiva.

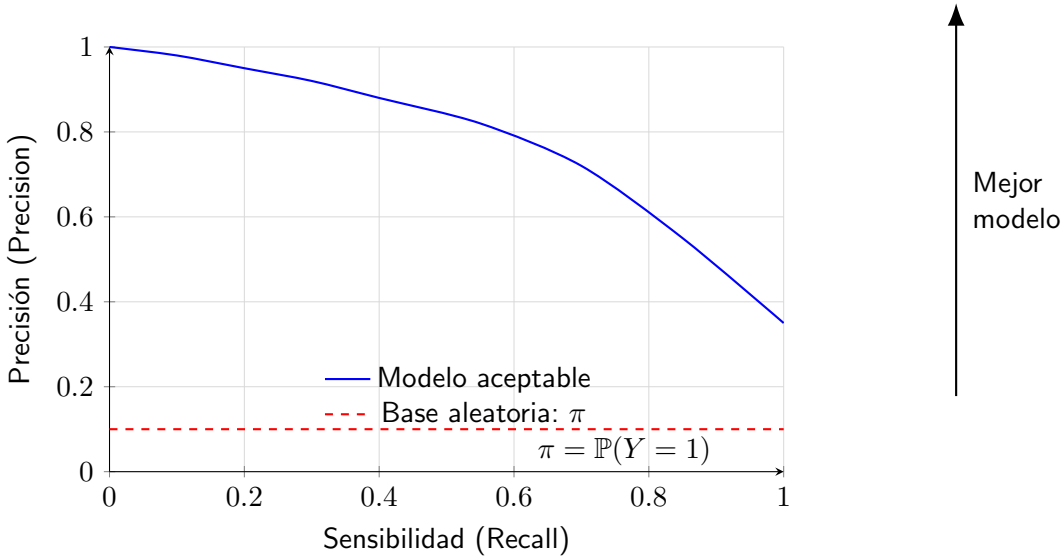


Figura 3.8: Ejemplo de curva Precision–Recall (PR).

3.3.9 Área Bajo la Curva PR

La calidad global del clasificador desde la perspectiva PR se resume mediante el área bajo la curva PR (PR AUC). En la práctica, esta cantidad suele aproximarse de forma discreta a partir de umbrales ordenados que inducen pares (R_k, P_k), donde R_k y P_k denotan Sensibilidad (recall) y Precisión en el punto k. Una aproximación estándar utilizada en la literatura y en implementaciones computacionales corresponde al *Average Precision* (AP):

$$\text{AP} = \sum_{k=1}^{K} (R_k - R_{k-1})\, P_k, \tag{3.30}$$

con $R_0 = 0$ y donde los puntos se ordenan por recall creciente. En este trabajo, se reporta PR AUC como medida complementaria al ROC AUC, dado que penaliza de manera directa la emisión de falsas alarmas (Falsos Positivos) en un escenario donde la clase positiva es rara.

Por esta razón, el análisis se centra en métricas orientadas a la detección de la clase positiva, como Sensibilidad, F_2-Score y PR AUC, relegando la Exactitud a un rol descriptivo.

3.4 Análisis de Datos Funcionales

El análisis de datos funcionales no introduce un problema distinto al planteado previamente, sino que propone una representación alternativa de las mismas variables de mercado utilizadas en el enfoque clásico. En lugar de trabajar con observaciones puntuales, las series temporales se modelan como funciones continuas en el tiempo, lo que permite capturar de manera más estructurada la dinámica temporal de precios, volúmenes y medidas derivadas. Bajo esta formulación, el objetivo sigue siendo la detección de patrones compatibles con manipulación, manteniendo la misma variable objetivo y_i, que indica manipulación cuando $y_i = 1$, o no cuando $y_i = 0$, y el mismo esquema de clasificación binaria.

En este trabajo, la detección de manipulación se formula como un problema de clasificación binaria supervisada, donde se busca aprender una regla de decisión a partir de datos etiquetados. En el enfoque clásico, cada observación se describe mediante un vector de características (datos tabulares). Sin embargo, muchas señales relevantes en mercados bursátiles se manifiestan como *patrones en el tiempo*: formas intradía, cambios de régimen, picos de actividad cerca del cierre, o dinámicas que no se resumen bien con pocos estadísticos.

El Análisis de Datos Funcionales (Functional Data Analysis, FDA) aborda este problema representando cada observación como curva (o función) $p_i(t)$ definida sobre un dominio continuo $t \in \mathcal{T}$, donde $\mathcal{T}$ suele ser un intervalo de tiempo. En términos prácticos, no observamos la función completa, sino mediciones discretas y ruidosas en una grilla:

$$P(t_j) + \varepsilon_{ij}, \quad j = 1, \ldots, m.$$

donde ε_{ij} representa el ruido de la medición. Para la realización correspondiente al día i, estas observaciones se denotan como

$$p_i(t_j) + \varepsilon_{ij}, \quad j = 1, \ldots, m.$$

La idea central es tratar la serie como una realización de un proceso aleatorio funcional y analizarla a nivel de forma (tendencias, curvatura, picos), no solo como una secuencia de puntos. Esta perspectiva es estándar en FDA y se desarrolla en textos como Ramsay y Silverman [37].

En finanzas, una forma natural de construir datos funcionales es representar cada día i como una curva intradía: precio, volumen, volatilidad intradía, u otra variable medida en intervalos regulares. Alternativamente, si no se dispone de intradía, también es posible definir funciones a partir de ventanas móviles (por ejemplo, una curva de retornos en una ventana de L días), aunque este caso requiere justificar bien el dominio temporal elegido.

3.4.1 Construcción de curvas a partir de datos discretos

Para trabajar con funciones, normalmente se realiza un paso de suavizado o reconstrucción funcional. La motivación es separar primero la estructura sistemática de la curva y segundo el ruido de medición. Un enfoque común es aproximar cada curva mediante una expansión en base:

$$p_i(t) \approx \sum_{k=1}^{p} c_{i,k}\, \phi_k(t),$$

donde $\{\phi_k(t)\}$ es una base de funciones y $\{c_{i,k}\}$ los coeficientes a estimar. La elección de la base funcional no es arbitraria, depende del tipo de dinámica observada y del nivel de suavidad esperado, buscando que las funciones de base posean rasgos similares a las trayectorias observadas para lograr una aproximación eficientes con un número reducido de parámetros [37]. A continuación, se describen las bases más utilizadas en la literatura.

B-Splines

Las B-splines buscan representar trayectorias no periódicas y con variaciones locales. Un spline corresponde a una función definida por tramos polinómicos que se unen en un conjunto ordenado de nodos, manteniendo un determinado grado de suavidad en dichos puntos de unión.

Sea $\{t_0 < t_1 < \cdots < t_M\}$ un conjunto ordenado de nodos y sea $p \geq 0$ el orden del spline, equivalente a polinomios de grado p. Las funciones base B-spline se definen de forma recursiva. Para orden cero, se obtienen funciones indicadoras asociadas a cada intervalo:

$$B_{l,0}(t) = \begin{cases} 1, & \text{si } t_l \leq t < t_{l+1}, \\ 0, & \text{en otro caso.} \end{cases}$$

Para órdenes superiores, las funciones B-spline se construyen mediante la relación recursiva:

$$B_{l,p}(t) = \frac{t - t_i}{t_{l+p} - t_i} B_{l,p-1}(t) + \frac{t_{l+p+1} - t}{t_{l+p+1} - t_{l+1}} B_{l+1,p-1}(t),$$

entendiendo que los cocientes se definen como cero cuando el denominador es nulo. Esta definición garantiza que cada $B_{l,p}(t)$ sea un polinomio por tramos de grado p, con continuidad de las derivadas hasta orden $p - 1$.

Un dato funcional $p(t)$ puede representarse mediante una combinación lineal de estas funciones base:

$$p_i(t) \approx \sum_{l=1}^{K} c_{i,l} B_{l,p}(t),$$

donde los coeficientes c_k se estiman a partir de los datos y el número total de funciones base K depende del orden p y del número de nodos considerados.

Desde un punto de vista técnico, una de las principales ventajas de las B-splines es su propiedad de soporte compacto. Cada función base es distinta de cero solo en un número limitado de intervalos adyacentes, lo que induce matrices de diseño y de productos internos con estructura bandeada. Esta característica mejora la estabilidad numérica y reduce el coste computacional de los procedimientos de estimación, haciendo que el tiempo de cálculo crezca aproximadamente de forma lineal con el número de observaciones [37]. Además, la localización temporal de las funciones base permite capturar variaciones locales del proceso mediante una elección adecuada de los nodos, evitando las oscilaciones globales típicas de las bases polinomiales.

No obstante, el uso de B-splines requiere tomar decisiones metodológicas relevantes. La elección del número y la ubicación de los nodos influye directamente en la calidad del ajuste: un número excesivo puede producir sobreajuste, mientras que un número reducido puede generar subajuste. El uso de nodos equiespaciados, aunque frecuente, puede ser ineficiente si la variabilidad del proceso no es homogénea en el dominio.

Wavelets

Las wavelets se basan en la expansión de una trayectoria a partir de traslaciones y dilataciones de una única función generadora, denominada wavelet madre y denotada por ψ. A diferencia de las bases de Fourier, que utilizan funciones con soporte global, las wavelets están diseñadas para proporcionar una representación localizada tanto en el tiempo como en la frecuencia, lo que permite un análisis de resolución múltiple.

Sea ψ una wavelet madre. El sistema de wavelets se define mediante dilataciones y traslaciones de la forma

$$\psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k), \quad j, k \in \mathbb{Z},$$

donde el índice j controla la escala o nivel de resolución, y el índice k determina la localización temporal. La normalización $2^{j/2}$ asegura que las funciones $\psi_{j,k}$ tengan norma unitaria en $L^2(\mathbb{R})$. Esta construcción permite descomponer una función en componentes asociados a distintas escalas, separando la información de baja frecuencia (estructura global) de la información de alta frecuencia (detalles locales).

Un dato funcional $p(t)$ puede representarse mediante una expansión en la base de wavelets:

$$p(t) \approx \sum_j \sum_k c_{j,k}\, \psi_{j,k}(t),$$

donde los coeficientes $c_{j,k}$ capturan la contribución de la señal en la escala j y la posición k.

Las wavelets ofrecen una representación localizada en el tiempo y la frecuencia, lo que permite capturar de forma eficiente irregularidades locales junto con estructuras suaves, con un coste computacional lineal mediante la transformada discreta de wavelet y la posibilidad de obtener representaciones dispersas mediante umbralización de coeficientes [37]. Sin embargo, su desempeño depende fuertemente de la elección de la wavelet madre y de los niveles de descomposición, y ciertas familias presentan dificultades para modelar derivadas suaves o pueden introducir variabilidad innecesaria, lo que limita su integración en modelos que requieren alta regularidad y puede favorecer el sobreajuste en entornos financieros ruidosos.

Si bien, existen otras bases utilizadas comúnmente como las de Fourier, no se utilizarán dada la naturaleza del estudio, ya que suelen pedir periodicidad en la muestra.

En la figura 3.9 podemos observar el ajuste de log retornos para curvas manipuladas y no manipuladas a través de la representación de B-splines y wavelets.

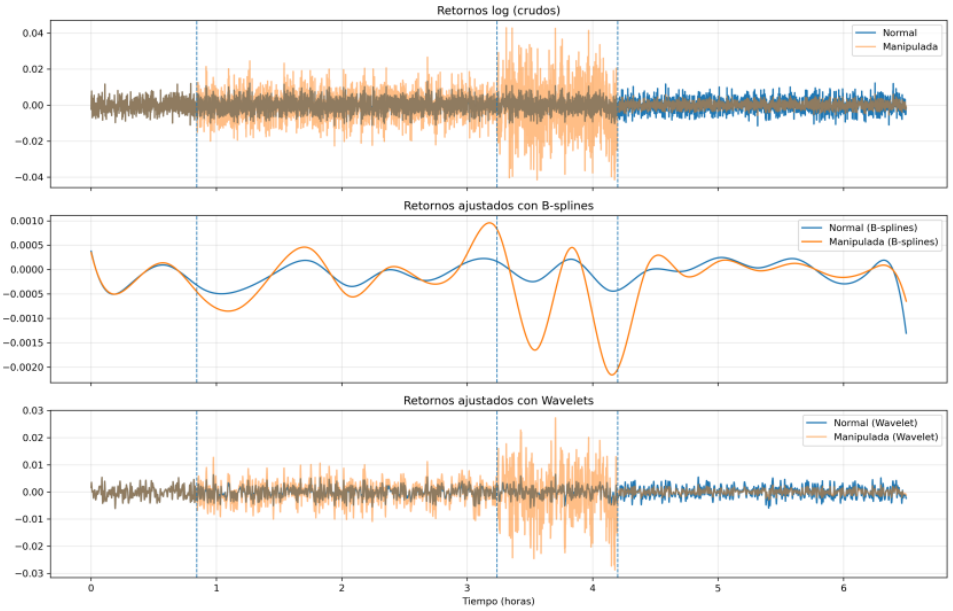


Figura 3.9: Log retornos manipulados y no manipulados representados a través de bases funcionales.

3.4.2 Análisis de Componentes Principales Funcionales

Una vez que los datos están representados como funciones, un objetivo central es reducir dimensión de manera interpretable. La herramienta estándar es el Análisis de Componentes Principales Funcionales (FPCA). Conceptualmente, FPCA busca direcciones funcionales (autofunciones) que expliquen la mayor variabilidad entre curvas, análogo a PCA multivariado.

Sea $\mu(t)$ la media funcional y $G(s,t)$ la función de covarianza:

$$\mu(t) = \mathbb{E}[p(t)], \qquad G(s,t) = \mathrm{Cov}(p(s), p(t)).$$

FPCA se basa en la descomposición espectral de G, entregando autofunciones $\{\phi_k(t)\}$ y autovalores $\{\lambda_k\}$:

$$\int_{\mathcal{T}} G(s,t)\phi_k(s)\, ds = \lambda_k \phi_k(t).$$

Cada curva se aproxima entonces como:

$$p_i(t) \approx \mu(t) + \sum_{k=1}^{K} z_{ik}\, \phi_k(t),$$

donde $z_{ik} = \int_{t\in\mathcal{T}} (p_i - \mu(t))\phi_k dt$ son los scores funcionales (coordenadas) de la curva i en el componente k. En la práctica, K se elige para capturar un porcentaje alto de varianza explicada o mediante validación cruzada.

En este trabajo, FPCA cumple dos roles:

1. Compresión: reducir la dimensión de la representación funcional. Si cada curva se representa inicialmente mediante M coeficientes de base, $\mathbf{c}_i = (c_{i1}, \dots, c_{iM})^\top$, FPCA transforma esta representación en un vector de \textit{scores} $\mathbf{z}_i = (z_{i1}, \dots, z_{iK})$, con $K \ll M$, conservando la mayor parte de la variabilidad entre curvas.

2. Interpretabilidad: las autofunciones $\phi_k(t)$ describen patrones típicos de variación (por ejemplo, aumento temprano vs. aumento tardío, picos cerca del cierre, etc.).

3.4.3 Problema de Clasificación Funcional

En clasificación funcional, el objetivo es predecir una etiqueta $y_i \in \{0,1\}$ a partir de una entrada funcional $p_i(\cdot)$. Formalmente, se busca una regla

$$\hat{f} : \mathcal{X} \to \{0,1\},$$

donde $\mathcal{X}$ es un espacio de funciones. Existen enfoques basados en reducción de dimensión como FPCA o bases funcionales seguidos de un clasificador multivariado "clásico" como los visto anteriormente en este capítulo, enfoques que trabajan directamente en espacios funcionales sin reducir la representación [44]. Un ejemplo de esto puede ser el kNN funcional.

El enfoque propuesto en este trabajo es un método de clasificación funcional basado en reducción de dimensión, y a efectos comparativos el kNN funcional. A continuación, se presenta el enfoque para el método basado en reducción:

1. Construcción de curvas: cada observación i se representa como una función $x_i(t)$ (por ejemplo, volumen intradía, retornos intradía, volatilidad intradía, etc.).

2. Preprocesamiento funcional: suavizado y/o representación en una base, para obtener funciones comparables.

3. FPCA: se estima $\mu(t)$ y las autofunciones $\phi_k(t)$, y se calcula el vector de scores $\mathbf{z}_i = (z_{i1}, \dots, z_{iK})$.

4. Clasificación multivariada: se entrena un clasificador clásico sobre $\mathbf{z}_i$, tal como se hace en el enfoque supervisado estándar.

Este esquema se puede interpretar como un "pseudo-clasificador funcional" porque el clasificador final no opera directamente sobre las curvas, sino sobre una representación finita de las funciones. Sin embargo, la información de entrada sí proviene de la curva completa, ya que los scores FPCA resumen la forma global de $p_i(t)$. En la práctica, esta estrategia es común en FDA porque permite reutilizar clasificadores robustos y bien estudiados, manteniendo el beneficio de representar la dinámica temporal mediante funciones Wang, Huang y Cao [44].

3.4.4 k-Vecinos más Cercanos Funcional

El clasificador de k-Vecinos Más Cercanos funcional representa una extensión directa del algoritmo multivariante clásico al espacio de dimensión infinita de las funciones. Formalmente, dado un conjunto de entrenamiento $\mathcal{D} = \{(p_i(t), y_i)\}_{i=1}^{N}$, donde $p_i(t) \in L^2(\mathcal{T})$ es una función de cuadrado integrable definida en un dominio compacto $\mathcal{T}$ y y_i es la etiqueta de clase asociada donde será $y_i = 1$, en caso que la curva sea manipulada, y $y_i = 0$ en caso que no lo sea; el objetivo es predecir la etiqueta y_{N+1} para una nueva curva observada $p_{N+1}(t)$.

Para la implementación computacional de este algoritmo, es necesario transformar las observaciones discretas del mercado en objetos funcionales matemáticamente tratables. En este trabajo, se opta por una representación mediante expansión de bases, asumiendo que cada trayectoria $p_i(t)$ puede aproximarse mediante una combinación lineal de funciones base $\{\phi_m(t)\}_{m=1}^{M}$, tal que $p_i(t) \approx \sum_{m=1}^{M} c_{im}\phi_m(t)$, donde $\mathbf{c}_i = (c_{i1}, \ldots, c_{iM})^\top$ es el vector de coeficientes. Esta representación permite trasladar el cálculo de distancias funcionales al espacio de coeficientes. La métrica utilizada para determinar la vecindad es la distancia L^2, que mide la disimilitud global entre dos trayectorias funcionales p_i y p_j. Utilizando la expansión de bases, la distancia al cuadrado se define como:

$$d^2(p_i, p_j) = \int_{\mathcal{T}} (p_i(t) - p_j(t))^2 dt = (\mathbf{c}_i - \mathbf{c}_j)^\top \mathbf{W} (\mathbf{c}_i - \mathbf{c}_j) \tag{3.31}$$

donde $\mathbf{W}$ es la matriz de productos interiores de las funciones base, con elementos $W_{mn} = \int \phi_m(t)\phi_n(t)dt$ [44].

3.5 Modelos Estocásticos

La implementación de sistemas robustos para la detección de manipulación de mercado enfrenta un obstáculo crítico derivado de la escasez y la naturaleza confidencial de los conjuntos de datos transaccionales reales debidamente etiquetados [22]. Dado que las actividades fraudulentas representan una fracción minúscula del volumen total de negociación en los mercados secundarios y que su identificación formal suele requerir investigaciones regulatorias exhaustivas, el acceso a datos históricos con una "verdad fundamental" verificable es extremadamente limitado para el investigador independiente. En este escenario, el recurso al modelado estocástico para la generación de datos sintéticos se justifica no solo como una solución ante la falta de muestras, sino como una ventaja técnica que permite el entrenamiento y validación de algoritmos de machine learning bajo entornos controlados donde los parámetros de manipulación son conocidos con precisión. Esta metodología constituye una práctica estándar aceptada tanto en la industria financiera como en el ámbito académico para simular patrones característicos de abuso de mercado e inyectarlos en flujos de datos normales [2].

3.5.1 Movimiento Browniano Geométrico

El Movimiento Browniano Geométrico (GBM) es uno de los modelos más utilizados en ingeniería financiera para modelar la dinámica de precios de activos, consolidado tras el trabajo de Samuelson [39].

Se prefiere este modelo frente al movimiento browniano aritmético porque asegura que los precios sean siempre no negativos. La ecuación diferencial estocástica que describe la evolución del precio S_t es:

$$dS_t = \mu S_t\, dt + \sigma S_t\, dW_t. \, t \in [0, T]. \tag{3.32}$$

En esta ecuación, μ representa la tendencia esperada (deriva) y σ la volatilidad porcentual. El término dW_t es el incremento de un proceso de Wiener, que introduce la aleatoriedad del mercado.

El proceso de Wiener W_t se caracteriza por iniciar en $W_0 = 0$, tener trayectorias continuas e incrementos independientes. Para cualquier intervalo de tiempo, el incremento $W_t - W_s$ sigue una distribución normal con media cero y varianza proporcional al tiempo transcurrido $(t - s)$ [41]. Esta propiedad refleja la hipótesis de que los cambios de precio dependen solo de la información actual.

Mediante el Lema de Itô, la solución analítica de la ecuación diferencial permite expresar el precio en el tiempo t como:

$$S_t = S_0 \exp\left(\left(\mu - \frac{1}{2}\sigma^2\right) t + \sigma W_t\right). \tag{3.33}$$

El término $-\frac{1}{2}\sigma^2$ es un ajuste matemático necesario para que la media del proceso logarítmico sea consistente. Esta fórmula es la base para generar trayectorias de precios simuladas.

Para modelar la manipulación de mercado tipo *pump and dump*, dependiendo de en que etapa nos encontremos. La manipulación ocurrirá en el intervalo $[t_{\text{pump, start}}, t_{\text{pump, end}}]$ y tiene dos fases [28]. En la fase de subida (*pump*), $\mu_{\text{manip}}(t)$ toma un valor positivo alto para inflar el precio. Luego, en la fase de caída (*dump*), el precio desciende rápidamente cuando el manipulador vende sus activos. De esta manera, en el intervalo $[0, t_{\text{pump, start}})$ el activo se encuentra en una etapa previa a la manipulación y sigue un comportamiento normal de mercado; en $[t_{\text{pump, start}}, t_{\text{pump, end}})$ ocurre el *pump*, donde aparece presión de compra artificial que empuja el precio al alza; en $[t_{\text{pump, end}}, t_{\text{dump, end}})$ se produce el *dump*, los manipuladores venden sus posiciones y domina la presión de venta, provocando caídas de precio; y en $(t_{\text{dump, end}}, L]$ tiene lugar el periodo post-dump, donde termina la intervención y el activo vuelve gradualmente a condiciones más estables.

Se define ls deriva $\mu(t)$ como:

$$\mu(t) = \begin{cases} \mu_{inicial}, & t \in [0, t_{\text{pump, start}}), \\ \mu_{\text{pump}}, & t \in [t_{\text{pump, start}}, t_{\text{pump, end}}), \\ \mu_{\text{dump}}, & t \in [t_{\text{pump, end}}, t_{\text{dump, end}}), \\ \mu_{\text{post dump}} & t \in (t_{\text{dump, end}}, L] \end{cases} \tag{3.34}$$

representando la interferencia artificial, mientras que la volatilidad σ se utilizará la configuración planteada por Aggarwal y Wu [1]. Esto simulará a un agente que genera un desequilibrio para desviar el precio de su valor fundamental a través del esquema *pump and dump*.

Luego, σ se define como a continuación:

$$\sigma(t) = \begin{cases} \sigma_{inicial}, & t \in [0, t_{\text{pump, start}}), \\ \sigma_{\text{pump}}, & t \in [t_{\text{pump, start}}, t_{\text{pump, end}}), \\ \sigma_{\text{dump}}, & t \in [t_{\text{pump, end}}, t_{\text{dump, end}}), \\ \sigma_{\text{post dump}} & t \in (t_{\text{dump, end}}, L] \end{cases}$$

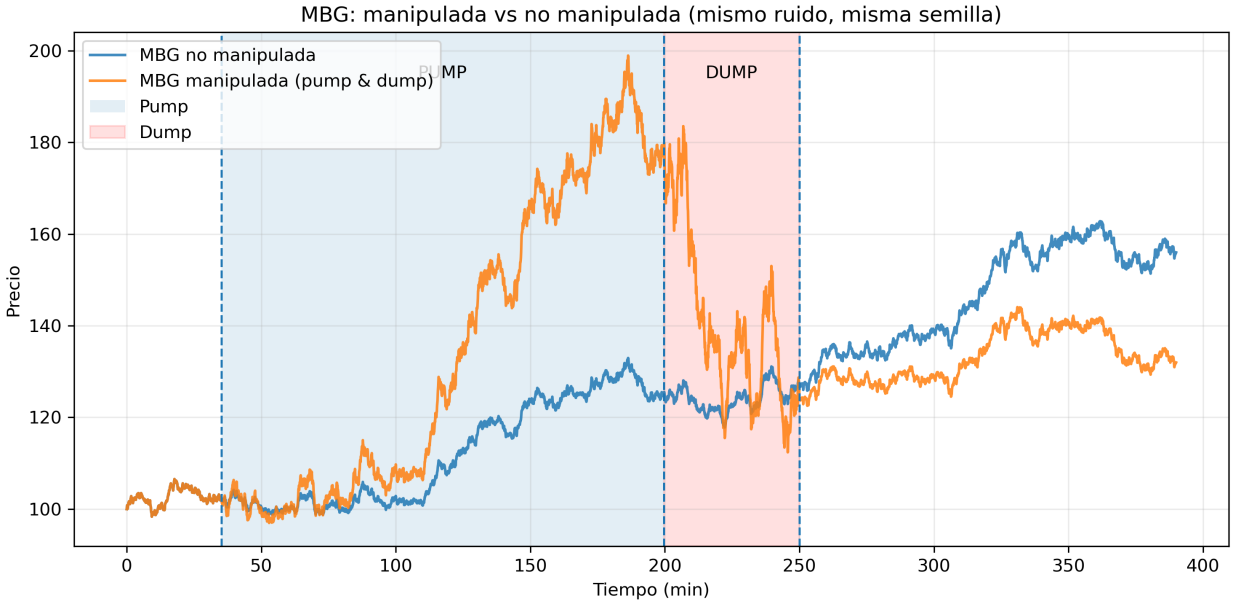


Figura 3.10: Trayectorias simuladas bajo MBG con y sin manipulación del tipo pump-and-dump usando el mismo ruido browniano (misma semilla). Las franjas indican el intervalo de pump y el de dump.

Este enfoque permite etiquetar los datos con el enfoque funcional de manera precisa para el entrenamiento: se asigna la clase positiva ($Y = 1$) a las curvas donde el manipulador se encuentre haya actuado y la clase negativa ($Y = 0$) a las trayectorias normales. El uso de datos sintéticos resuelve la escasez de registros reales de fraude [36] y permite evaluar la eficacia de los algoritmos de detección bajo condiciones controladas [42].

Como se mostrará en el capítulo de experimentos computacionales, el uso de trayectorias generadas mediante GBM presenta limitaciones cuando se evalúa el desempeño de los algoritmos en la detección de escenarios de *pump and dump*.

El uso de modelos estocásticos para la generación de datos sintéticos responde a la escasez de bases de datos reales que cuenten con etiquetas verificadas de manipulación. En este contexto, los datos sintéticos permiten evaluar el comportamiento de los modelos de clasificación bajo escenarios controlados, donde la presencia o ausencia de manipulación es conocida por construcción. Este enfoque no pretende sustituir el análisis con datos reales, sino complementarlo, proporcionando un marco experimental adicional para el estudio de la detección de manipulación.

3.5.2 Modelo de Heston

El modelo de Heston extiende el Movimiento Browniano Geométrico al permitir que la volatilidad del activo sea estocástica en lugar de constante. Fue propuesto por Heston [25] y es ampliamente utilizado porque reproduce características observadas en los mercados financieros, como la variación temporal de la volatilidad y la correlación entre retornos y volatilidad.

El modelo está definido por el siguiente sistema de ecuaciones diferenciales estocásticas:

$$dS_t = \mu S_t\, dt + \sqrt{v_t}\, S_t\, dW_t^{(1)}, \tag{3.35}$$

$$dv_t = \kappa(\theta - v_t)\, dt + \xi\sqrt{v_t}\, dW_t^{(2)}.\, t \in [0, T]. \tag{3.36}$$

En este sistema, S_t representa el precio del activo y v_t su varianza instantánea. El parámetro μ corresponde a la deriva del precio, κ controla la velocidad de reversión a la media de la varianza, θ es el nivel de varianza de largo plazo y ξ determina la volatilidad de la varianza. Los procesos de Wiener $W_t^{(1)}$ y $W_t^{(2)}$ están correlacionados con coeficiente ρ.

La ecuación de la varianza corresponde a un proceso llamado de reversión a la media. Bajo condiciones estándar, este proceso asegura que la varianza permanezca no negativa. En la implementación numérica, esta propiedad se refuerza imponiendo explícitamente $v_t \geq 0$ en cada paso de simulación.

A diferencia del GBM, la dinámica del precio depende directamente de la varianza instantánea, lo que genera trayectorias con volatilidad cambiante en el tiempo y un comportamiento más realista del precio del activo.

Para modelar manipulación de mercado tipo *pump and dump*, se introducen perturbaciones tanto en la deriva del precio como en el nivel de varianza de largo plazo. Se define una deriva dependiente del tiempo $\mu(t)$ y un nivel objetivo de varianza $\theta(t)$:

$$\mu(t) = \begin{cases} \mu_{inicial}, & t \in [0, t_{\text{pump, start}}), \\ \mu_{\text{pump}}, & t \in [t_{\text{pump, start}}, t_{\text{pump, end}}), \\ \mu_{\text{dump}}, & t \in [t_{\text{pump, end}}, t_{\text{dump, end}}), \\ \mu_{\text{post dump}} & t \in (t_{\text{dump, end}}, L] \end{cases} \tag{3.37}$$

$$\theta(t) = \begin{cases} \theta_{inicial}, & t \in [0, t_{\text{pump, start}}), \\ \theta_{\text{pump}}, & t \in [t_{\text{pump, start}}, t_{\text{pump, end}}), \\ \theta_{\text{dump}}, & t \in [t_{\text{pump, end}}, t_{\text{dump, end}}), \\ \theta_{\text{post dump}} & t \in (t_{\text{dump, end}}, L] \end{cases} \tag{3.38}$$

Durante la fase de *pump*, el manipulador introduce una deriva positiva adicional que fuerza una subida artificial del precio, acompañada de un aumento en el nivel de varianza de largo plazo para reflejar un incremento en la actividad del mercado. En la fase de *dump*, la deriva se vuelve negativa, generando una caída rápida del precio, mientras que la varianza se reduce para representar la pérdida de interés posterior a la liquidación de posiciones.

El intervalo de manipulación $[t_0, t_2]$, así como las duraciones relativas de las fases de *pump* y *dump*, se generan de forma aleatoria dentro de rangos predefinidos. Esto introduce heterogeneidad en las trayectorias manipuladas y evita patrones deterministas.

Este enfoque permite etiquetar las trayectorias de manera precisa para el entrenamiento. Las curvas donde el manipulador actúa se asignan a la clase positiva ($Y = 1$), mientras que las trayectorias generadas sin perturbaciones corresponden a la clase negativa ($Y = 0$). El uso del modelo de Heston permite evaluar los algoritmos de detección en un escenario más exigente que el GBM, debido a la presencia de volatilidad estocástica.

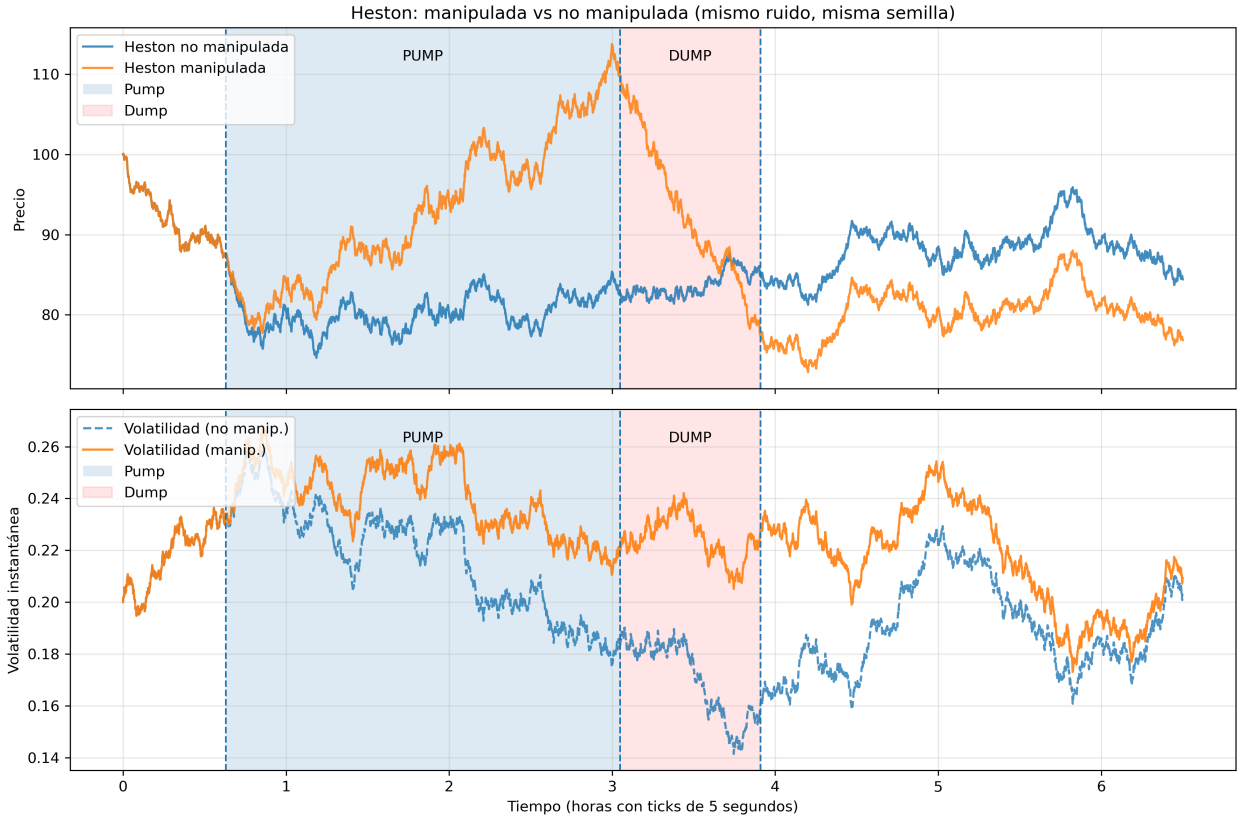


Figura 3.11: Trayectorias simuladas con modelo de Heston, con y sin manipulación del tipo pump-and-dump usando el mismo ruido browniano (misma semilla). Las franjas indican el intervalo de pump y el de dump.

3.5.3 Modelo de Microestructura de Cont-Müller

A diferencia de los modelos de movimiento browniano (geométrico o con volatilidad estocástica) que asumen una dinámica exógena para el precio, el modelo propuesto por Cont y Mueller [10] deriva la evolución del precio a partir de la dinámica interna del Libro de Órdenes Límite (LOB). Este enfoque, basado en Ecuaciones Diferenciales Parciales Estocásticas (SPDE), permite capturar propiedades de la microestructura del mercado, vinculando directamente el flujo de órdenes con la formación de precios.

Recordemos que la profundidad del libro de órdenes se define como la cantidad de órdenes de compra y venta disponibles a distintos niveles de precios, y se asocia a la liquidez del mercado y al impacto potencial de nuevas órdenes. Con esto en mente, el esquema asume que la profundidad del mercado exhibe un comportamiento de reversión a la media. Así, la dinámica conjunta de las profundidades en el *bid* (D_t^b) y en el *ask* (D_t^a), junto con el precio medio (S_t), se describe mediante el siguiente sistema de ecuaciones diferenciales estocásticas:

$$\begin{aligned} dD_t^b &= \nu_b(\overline{D}_b - D_t^b)dt + \sigma_b D_t^b dW_t^1, \\ dD_t^a &= \nu_a(\overline{D}_a - D_t^a)dt + \sigma_a D_t^a dW_t^2, \\ dS_t &= \theta\left[\frac{\nu_b(\overline{D}_b - D_t^b)}{D_t^b} - \frac{\nu_a(\overline{D}_a - D_t^a)}{D_t^a} - (\nu_b - \nu_a)\right]dt \\ &\quad + \theta(\sigma_b - \varrho_{a,b}\sigma_a)dW_t^1 - \theta\sqrt{1-\varrho_{a,b}^2}\sigma_a dW_t^2. t \in [0,T]. \end{aligned} \tag{3.39}$$

Donde $\nu_{a,b}$ representan las velocidades de reversión a la media de la profundidad hacia los niveles de equilibrio $\overline{D}_{a,b}$, y $\sigma_{a,b}$ son las volatilidades del volumen de órdenes. Además, W_t^1 y W_t^2 son dos movimientos brownianos independientes, y para introducir correlación $\varrho_{a,b}$ entre los shocks que afectan a las profundidades bid y ask, se define $d\widetilde{W}_t^a = \varrho_{a,b}\, dW_t^1 + \sqrt{1-\varrho_{a,b}^2}\, dW_t^2$. Este ruido se utiliza en la dinámica de D_t^a.

Un aspecto crucial de este modelo es que la volatilidad del precio es endógena. Como se observa en la tercera ecuación del sistema (3.39), el precio no posee una fuente de ruido independiente; su componente estocástico es una combinación lineal de los mismos ruidos que afectan a la profundidad del libro (dW_t^1, dW_t^2), ponderados por el coeficiente de impacto de mercado θ y las volatilidades de la profundidad. Esto implica que periodos de alta volatilidad en el flujo de órdenes se traducen mecánicamente en alta volatilidad de precios.

Para simular escenarios de manipulación tipo *pump and dump* bajo este marco, se altera el equilibrio natural del sistema mediante la introducción de perturbaciones en los parámetros estructurales durante las ventanas de manipulación:

1. **Deriva Exógena:** Se añade un término de deriva adicional $\mu_{\text{manip}}(t)$ a las ecuaciones de la profundidad del libro D_a y D_b. Durante la fase de *pump*, este término es positivo ($\mu_{\text{pump}} > 0$), forzando una subida artificial que simula una presión de compra agresiva. Durante el *dump*, se invierte el signo ($\mu_{\text{dump}} < 0$).

2. **Amplificación de Volatilidad:** Dado que la evidencia empírica sugiere un aumento de la actividad y volatilidad durante estos esquemas [1], se aplica un multiplicador $\lambda_{\text{vol}} > 1$ a las volatilidades base de la profundidad (σ_a, σ_b) durante la fase de *pump*. Debido a la naturaleza endógena del modelo, este incremento en $\sigma_{a,b}$ se propaga automáticamente a la ecuación del precio, incrementando la volatilidad de S_t.

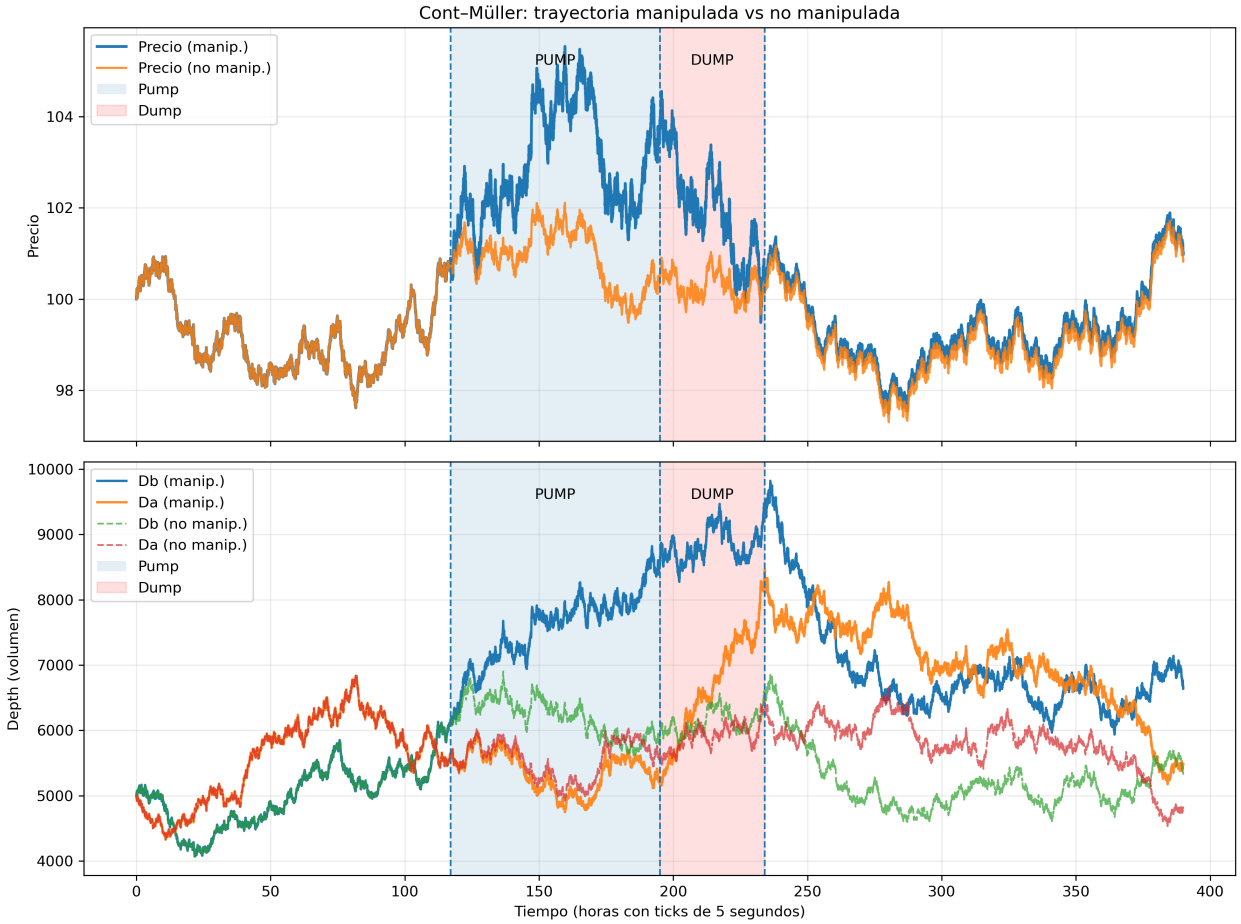


Figura 3.12: Trayectorias simuladas con Modelo de Cont-Müller, con y sin manipulación tipo pump-and-dump usando el mismo ruido browniano (misma semilla). Las franjas indican el intervalo de pump y el de dump.

De esta manera, el modelo genera trayectorias donde el precio no es un paseo aleatorio puro, sino la consecuencia de cambios estructurales en la liquidez disponible y presiones exógenas, proporcionando al clasificador funcional características más ricas y complejas vinculadas a la microestructura del mercado.

3.5.4 Comparativa de los Modelos Estocásticos

En este trabajo se emplearán los tres modelos estocásticos mencionados anteriormente con el objetivo de analizar la detección de esquemas de manipulación tipo *pump and dump* bajo supuestos estructurales progresivamente más complejos. El uso de más de un modelo no responde a un interés meramente comparativo, sino a la necesidad de evaluar cómo cambia la señal de manipulación según el mecanismo mediante el cual esta se introduce en el mercado.

El modelo MBG corresponde al caso más simple. En este marco, la manipulación se representa como un cambio exógeno en la deriva y la volatilidad del precio, sin interacción interna entre ambas magnitudes. Este enfoque permite capturar la forma más directa de un evento *pump and dump*, donde el precio presenta subidas y caídas abruptas claramente visibles. Su principal ventaja es la simplicidad y la facilidad de interpretación. No obstante, el modelo no incorpora volatilidad estocástica ni efectos de memoria, por lo que su capacidad para representar dinámicas reales de mercado es limitada.

El modelo de Heston introduce volatilidad estocástica y correlación entre retornos y varianza. Bajo este esquema, la manipulación no solo afecta la trayectoria del precio, sino que también genera cambios persistentes en la volatilidad. Esto permite estudiar escenarios en los que el evento manipulativo deja una señal más rica, observable tanto en los retornos como en la estructura temporal de la varianza. Sin embargo, el modelo de Heston sigue operando a nivel agregado del precio y no incluye de forma explícita variables de liquidez ni mecanismos de microestructura.

Por último, el modelo de Cont-Müller se basa en la microestructura del mercado y modela la manipulación como una distorsión en la liquidez, a través de cambios en las profundidades bid y ask y en sus dinámicas de reversión. En este caso, la señal de manipulación aparece en el precio de forma indirecta, como consecuencia de desequilibrios persistentes en el libro de órdenes. Este enfoque permite representar un mecanismo más cercano a la operativa real de los esquemas *pump and dump*. No obstante, su mayor complejidad y sensibilidad a los parámetros podría generar señales de precio menos claras, especialmente en escenarios de desbalance severo.

En conjunto, los tres modelos permiten estudiar la detección de manipulación desde perspectivas complementarias: el MBG captura cambios exógenos simples en el precio, Heston incorpora efectos conjuntos en precio y volatilidad, y Cont-Müller modela la manipulación como una alteración estructural de la liquidez. Esta comparación proporciona un marco claro para interpretar las diferencias observadas en el desempeño de los clasificadores a lo largo de los experimentos.

Capítulo 4

Experimentos Computacionales

En este capítulo se evalúan dos fuentes de datos con distinta resolución temporal. La base de datos real de FLC se encuentra disponible a frecuencia diaria (OHLCV), por lo que cada observación corresponde a un día de negociación y se describe de manera natural mediante un vector de características tabulares, construido a partir de retornos, ratios de volumen y medidas de volatilidad diaria. En este contexto, no es posible construir curvas intradía comparables sin introducir supuestos adicionales o acceder a datos de alta frecuencia no disponibles.

Por otro lado, los datos sintéticos se generan directamente sobre una grilla intradía fija (ticks de 5 segundos), lo que permite representar cada trayectoria como una función $p_i(t)$. Esto habilita el uso de técnicas de análisis de datos funcionales, incluyendo suavizado mediante B-splines o wavelets y reducción de dimensión a través de FPCA para obtener un vector de *scores* $\mathbf{z}_i$.

En ambos escenarios, la etapa final de decisión se realiza mediante clasificadores supervisados clásicos. La diferencia fundamental radica en la representación de entrada: variables tabulares en el caso de datos reales y representaciones funcionales (scores FPCA) en el caso de datos sintéticos.

4.1 Descripción de la base de datos real

Para los experimentos con datos reales se utiliza la base de datos del activo FLC Group JSC (FLC), que cotiza en la bolsa de Ho Chi Minh, la base de datos contiene el registro del precio diario en la moneda local, y contiene datos desde el $06-08-2013$ hastas el $04-05-2022$, y los días marcados como manipulados desde el $01-12-2021$ hasta el $10-01-2022$, por la manipulación del stock el expresidente de FLC Group fue condenado por fraude y manipulación bursátil, como parte de una campaña anticorrupción en Vietnam, afectando a la empresa y sus líderes. Esta base de datos contiene las columnas:

1. Open: Precio de apertura del día.
2. High: Precio más alto del día.
3. Low: Precio más bajo del día.
4. Close: Precio de cierre del día.
5. Volume: Volumen transado diario.
6. Manipulated: Etiqueta si el día fue manipulado.

A partir de estas, se realizará ingeniería de características para enriquecer la información de los modelos. Se incluirán:

1. Log-Retornos diarios.
2. Volatilidad calculada sobre una ventana móvil de retornos.
3. Razón del volumen respecto a su referencia histórica.

Podemos ver explicitamente los días marcados como manipulados en la serie de tiempo del precio de apertura.

Figura 4.1: Serie temporal del precio de apertura entre 2021 y 2022, con los días marcados como manipulados destacados mediante puntos rojos del activo FLC.

Podemos observar también la distribución del volumen transado por día.

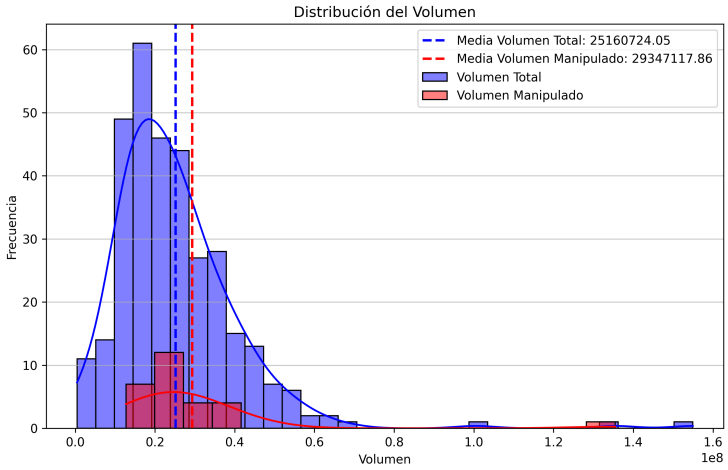


Figura 4.2: Distribución del volumen diario transado. En azul se muestra el volumen total y en rojo el volumen correspondiente a los días etiquetados como manipulados. Las líneas verticales indican las medias respectivas.

4.1.1 Partición de la base de datos

Si bien la base de datos cuenta con 2170 observaciones inicialmente, tras la ingeniería de características, se generan ciertos valores, que se omitiran, quedando un total de 2151 observaciones. Para desarrollar los modelos, se optó por asignar aleatoriamente un 30 % de datos de prueba, y un 70 % para entrenar los modelos, quedando de esta forma 1505 datos para el entrenamiento, y 646 datos de prueba.

4.1.2 Desbalance de clases

El desafío del desbalance de clases es importante en la detección de manipulación. En el contexto del aprendizaje supervisado, se dice que un conjunto de datos $\mathcal{D}$ está desbalanceado cuando las frecuencias de las clases que conforman la variable objetivo y presentan una disparidad significativa.

Para el problema de detección de manipulación bursátil tratado en este trabajo, definimos la clase minoritaria (casos manipulados) como la clase positiva ($y = 1$) y la clase mayoritaria (comportamiento normal) como la clase negativa ($y = 0$). Sean N_1 y N_0 el número de muestras de la clase positiva y negativa respectivamente, tal que $N = N_0 + N_1$. En escenarios de manipulación de mercado, se cumple típicamente que $N_0 \gg N_1$.

El grado de desbalance se puede cuantificar mediante el ratio de desbalance (ρ), definido como:

$$\rho = \frac{N_0}{N_1} \tag{4.1}$$

Cuando ρ es elevado, los algoritmos de clasificación estándar tienden a exhibir un sesgo hacia la clase mayoritaria. Esto ocurre porque la mayoría de las funciones de pérdida estándar, $L(\theta)$, buscan minimizar el error global promedio, tratando a todas las observaciones con la misma importancia.

Matemáticamente, si un modelo trivial predice $\hat{y} = 0$ para todas las observaciones, obtendrá una exactitud de $1-1/\rho$, lo cual puede parecer un rendimiento alto aunque el modelo sea inútil para detectar la manipulación, como sugieren Golmohammadi, Díaz y Zaiane [22], esta es la razón principal para la utilización del F_2-score.

Para mitigar este efecto, existen dos estrategias principales:

1. **Técnicas de remuestreo:** Consisten en modificar la distribución del conjunto de entrenamiento $\mathcal{D}$ antes de entrenar el modelo. Esto incluye el *undersampling* (eliminar muestras de la clase mayoritaria) y el *oversampling* (generar nuevas muestras sintéticas de la clase minoritaria), siendo SMOTE (Synthetic Minority Over-sampling Technique) uno de los algoritmos más utilizados [8].

2. **Aprendizaje sensible al costo:** Esta técnica modifica la función de objetivo del algoritmo para penalizar más severamente los errores de clasificación cometidos en la clase minoritaria. Esto se logra introduciendo un vector de pesos de clase w_y, inversamente proporcional a la frecuencia de la clase:

   $$w_j = \frac{N}{C \cdot N_j}, \quad \text{para } j \in \{0, 1\} \tag{4.2}$$

   Donde C es el número de clases. Integrando esto en una función de riesgo empírico, como la Entropía Cruzada ponderada, el objetivo de optimización se transforma en:

   $$L_{ponderada}(\theta) = -\sum_{i=1}^{N} w_{y_i} \cdot \log(P(y_i|\mathbf{x}_i; \theta)) \tag{4.3}$$

   De esta forma, el gradiente de la función de pérdida se ve forzado a priorizar la correcta clasificación de los eventos de manipulación ($y = 1$), compensando su baja representatividad numérica en el conjunto de datos [24].

Dado que los datos presentan un desbalance de clases severo con un ratio de 75.82 (aproximadamente $1:76$), se abordará el problema mediante dos enfoques complementarios. En primer lugar, se aplicará la técnica de remuestreo SMOTE a todos los algoritmos seleccionados, con el objetivo de evaluar su rendimiento bajo condiciones de entrenamiento equilibradas sintéticamente. En segundo lugar, se implementará durante la fase de entrenamiento el aprendizaje sensible al costo específicamente en aquellos modelos cuya formulación matemática admita la ponderación directa de la función de pérdida (tales como SVM, Árboles de Decisión y Redes Neuronales), permitiendo así contrastar la eficacia de la penalización algorítmica frente a la generación de datos sintéticos. El diagrama de flujo de trabajo, quedará ilustrado en la Figura 4.3.

4.1.3 Diagrama del Flujo de trabajo

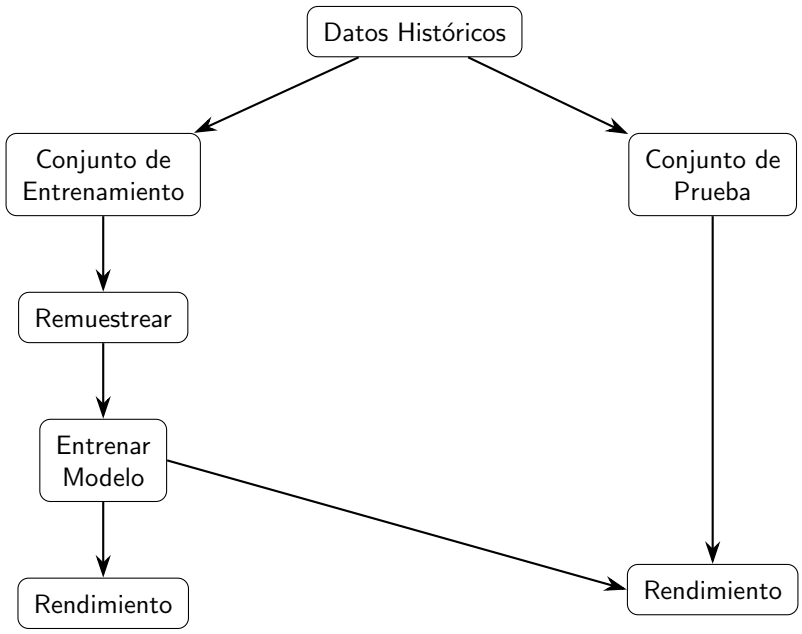


Figura 4.3: Diagrama del flujo de trabajo experimental. Se observa la partición de los datos históricos en conjuntos de entrenamiento y prueba. Es crucial notar que la técnica de remuestreo (SMOTE) se aplica exclusivamente sobre el conjunto de entrenamiento para evitar la filtración de datos[1], asegurando así una evaluación imparcial del rendimiento del modelo sobre datos reales no vistos.

4.2 Resultados Experimentales en Datos reales

Para mantener la sección legible, en el cuerpo se reporta solo la comparación global de métricas. El Anexo A incluye, para cada clasificador, la configuración final de entrenamiento, su matriz de confusión y el detalle de métricas en el conjunto de prueba.

Previo a la evaluación de los distintos modelos de aprendizaje automático, se definió una configuración experimental común, tanto en términos del conjunto de variables como de la estrategia de entrenamiento y evaluación. Una decisión metodológica central en esta etapa fue la exclusión del precio de los activos en su forma cruda, privilegiando en su lugar variables transformadas y estandarizadas que capturan de manera más adecuada la dinámica relevante para la detección de manipulación de mercado.

[1]En este contexto, la filtración de datos hace referencia a cuando el modelo tiene acceso, directa o indirectamente, a información del conjunto de prueba durante la fase de entrenamiento.

En particular, los modelos fueron entrenados exclusivamente utilizando las siguientes variables:

- *Volume*: Volumen transado.
- *Volume MA*: Media móvil del volumen.
- *log_ret*: Retorno logarítmico del precio.
- *rolling_vol*: Volatilidad calculada sobre una ventana móvil de retornos.
- *vol_ratio*: Razón del volumen respecto a su referencia histórica.

La exclusión del precio en niveles se encuentra alineada con lo propuesto por Golmohammadi, Díaz y Zaiane [22], quienes señalan que, si bien el precio constituye una variable central para el monitoreo del mercado, no debe ser utilizado en su forma cruda como variable de entrada en modelos estadísticos o de aprendizaje automático. En particular, los autores destacan que el amplio rango de valores de los precios y su dependencia de la escala del activo dificultan el aprendizaje y la comparabilidad entre observaciones.

En su lugar, se recomienda emplear retornos como una forma de normalización inherente del precio, eliminando explícitamente la variable de precio en niveles del conjunto de datos. Siguiendo esta recomendación, en este trabajo se utilizaron retornos logarítmicos y se descartaron las variables de precio crudo, manteniendo únicamente información derivada de su dinámica temporal.

Adicionalmente, todas las variables utilizadas fueron estandarizadas mediante *StandardScaler*, asegurando media cero y varianza unitaria en el conjunto de entrenamiento. Este paso resulta fundamental para modelos sensibles a la escala de las variables, tales como SVM, KNN y redes neuronales, y contribuye a una comparación justa y consistente entre los distintos clasificadores evaluados.

Finalmente, el énfasis en variables asociadas al volumen y a medidas de volatilidad responde a su capacidad para capturar patrones de actividad anómala frecuentemente asociados a comportamientos manipulativos. Sobre esta base común de variables preprocesadas se entrenaron y evaluaron todos los modelos presentados en las subsecciones siguientes, garantizando coherencia metodológica y comparabilidad directa de los resultados obtenidos.

4.2.1 Discusión y comparación de modelos

Con el objetivo de evaluar de manera integral el desempeño de los distintos clasificadores implementados, se presenta en la Tabla 4.1 una comparación directa de las métricas obtenidas sobre el conjunto de prueba. Dado el carácter altamente desbalanceado del problema ($\rho \approx 76$), la discusión se centra principalmente en métricas sensibles a la clase minoritaria, tales como la Sensibilidad, el F_2-Score, el ROC AUC y el PR AUC, relegando la Exactitud a un rol secundario. Además, se pueden encontrar las configuraciones para cada algoritmo en A.

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9644	0.2222	0.7500	0.5085	0.8585	0.1698
Random Forest	0.9721	0.2222	0.5000	0.4000	0.9781	0.2589
KNN	0.9551	0.1818	0.7500	0.4615	0.8488	0.1425
SVM (RBF)	0.9334	0.1277	0.7500	0.3797	0.9624	0.2731
Naive Bayes	0.9427	0.1628	0.8750	0.4667	0.9726	0.3117
MLP	0.8839	0.0864	0.8750	0.3097	0.9520	0.1362

Cuadro 4.1: Comparación de métricas de desempeño de los modelos evaluados sobre el conjunto de prueba.

A partir de los resultados presentados en la Tabla 4.1, se observa que los clasificadores exhiben diferencias claras en el compromiso entre detección de la clase minoritaria y control de falsas alarmas. Ningún modelo alcanza Sensibilidad perfecta, lo que indica un comportamiento menos extremo y más equilibrado en términos operativos.

En términos de Sensibilidad, los valores más altos corresponden a Naive Bayes y MLP, ambos con *Recall* igual a 0.8750, seguidos por CART, KNN y SVM con valores de 0.7500. Este comportamiento indica que dichos modelos logran recuperar una fracción significativa de los eventos de manipulación. Sin embargo, este aumento en detección se acompaña de niveles bajos de Precisión, especialmente en MLP y SVM, reflejando una mayor proporción de falsas alarmas.

Al considerar el F_2-Score, que prioriza la clase minoritaria, CART presenta el mejor desempeño global ($F_2 = 0.5085$), seguido por Naive Bayes (0.4667) y KNN (0.4615). Estos resultados sugieren que, bajo la configuración evaluada, CART logra el mejor equilibrio entre Sensibilidad y control de falsas alarmas desde una perspectiva orientada a detección. En contraste, SVM y MLP exhiben valores de F_2 inferiores, lo que evidencia que su alta Sensibilidad no se traduce en un desempeño operativo robusto.

Desde el punto de vista discriminativo, Random Forest y Naive Bayes destacan por sus altos valores de ROC AUC y PR AUC. En particular, Naive Bayes alcanza el mayor PR AUC (0.3117), indicando una buena capacidad para priorizar observaciones de la clase minoritaria. No obstante, su menor Precisión limita su utilidad práctica si no se ajustan los umbrales de decisión. Random Forest, por su parte, muestra un comportamiento más equilibrado, con valores altos de ROC AUC (0.9781) y PR AUC (0.2589), aunque con un F_2 inferior al de CART.

En conjunto, los resultados muestran que ningún clasificador domina simultáneamente todas las métricas relevantes. La elección del modelo depende del criterio operativo: si se prioriza la detección de eventos manipulativos, CART y Naive Bayes resultan competitivos; si se privilegia capacidad discriminativa global, Random Forest presenta ventajas claras.

Para evaluar el impacto del desbalance de clases sobre el desempeño de los modelos, se comparan a continuación los resultados obtenidos con y sin la aplicación de técnicas de balanceo en el conjunto de entrenamiento. La Tabla 4.2 presenta los resultados obtenidos sin aplicar SMOTE ni ponderación de clases.

Modelo	**Accuracy**	**Precision**	**Recall**	$\boldsymbol{F_2}$	**ROC AUC**	**PR AUC**
CART	0.9861	0.0000	0.0000	0.0000	0.9234	0.2527
Random Forest	0.9861	0.0000	0.0000	0.0000	0.9825	0.3748
KNN	0.9830	0.2000	0.1250	0.1351	0.8554	0.1961
SVM (RBF)	0.9876	0.0000	0.0000	0.0000	0.9606	0.1836
Naive Bayes	0.9768	0.2308	0.3750	0.3333	0.9685	0.2074
MLP	0.9876	0.0000	0.0000	0.0000	0.5096	0.0135

Cuadro 4.2: Comparación de métricas de desempeño de los modelos sin técnicas de balanceo de clases en el conjunto de entrenamiento evaluados sobre el conjunto de prueba.

Como se muestra en la Tabla 4.2, sin técnicas de balanceo la mayoría de los clasificadores colapsa hacia la clase mayoritaria, obteniendo Sensibilidad y F_2 nulos. Este comportamiento se observa claramente en CART, Random Forest, SVM y MLP, pese a sus altos valores de *Accuracy*.

La Tabla 4.2 evidencia de forma explícita el impacto del desbalance de clases cuando no se aplican técnicas de balanceo en el conjunto de entrenamiento. En este escenario, CART, Random Forest, SVM y MLP colapsan completamente hacia la clase mayoritaria, obteniendo valores nulos de *Recall* y F_2.

KNN y Naive Bayes constituyen excepciones parciales, al recuperar una fracción limitada de la clase minoritaria. Sin embargo, sus valores de F_2 permanecen bajos, lo que indica un desempeño insuficiente

bajo desbalance severo. Aun así, los valores relativamente altos de ROC AUC y PR AUC en modelos como Random Forest y SVM sugieren que la señal discriminativa está presente, pero no se traduce en decisiones efectivas bajo umbrales estándar.

4.3 Descripción de los Datos Sintéticos

Una vez definidos los fundamentos teóricos de los modelos estocásticos (MBG, Heston y Cont-Müller), es necesario establecer el marco experimental bajo el cual se generaron los datos sintéticos y se entrenaron los clasificadores. El objetivo de esta etapa es crear un conjunto de datos controlado que emule las características de alta frecuencia de un mercado financiero, permitiendo la inyección de patrones de manipulación del tipo *pump and dump* para evaluar la robustez de los algoritmos de detección. Para garantizar la comparabilidad entre los distintos modelos y el realismo de las series temporales, se estableció una configuración temporal fija equivalente a una jornada bursátil estándar. Cada trayectoria simulada representa un día de negociación con ticks de 5 segundos[2].

4.3.1 Parámetros de Simulación Modelo MBG

Los parámetros base para cada esquema se escogeran de acuerdo a al estudio realizado por Aggarwal y Wu [1].

Los parámetros usados para la simulación del MBG, junto con los parámetros del evento de manipulación, se resumen en la Tabla 4.3.

[2]Un *tick* corresponde a un instante discreto de observación en la serie temporal. En este trabajo, un tick de 5 segundos implica que el precio del activo se registra y actualiza cada 5 segundos a lo largo de la jornada bursátil simulada, fijando así la resolución temporal de las trayectorias.

Parámetro	Símbolo	Valor / Rango
Configuración Temporal		
Tiempo de simulación	T	6.5 horas (jornada bursátil)
Paso de tiempo	Δt	5 segundos
Puntos por curva	N	$\approx 4{,}680$ observaciones
Parámetros Estocásticos (Base MBG)		
Precio inicial	S_0	$U(2, 2000)$
Deriva base	μ_{base}	$0.0169 + U(-0.4880, 0.4880)$
Volatilidad base	σ_{base}	$0.2431 + U(0, 0.4564)$
Parámetros de Manipulación (Pump & Dump)		
Fracción de inicio del evento	–	$[0.05, 0.25] \cdot N$
Duración del pump	–	$[0.30, 0.40] \cdot N$
Duración del dump	–	$[0.10, 0.20] \cdot N$
Deriva en pump	μ_{pump}	$\lvert\mu_{\text{base}} + U(0, 0.8933)\rvert \cdot m$
Deriva en dump	μ_{dump}	$-\exp(\lvert\mu_{\text{pump}}\rvert)$
Deriva post-dump	μ_{post}	$-\lvert\mu_{\text{base}}\rvert$
Volatilidad en pump	σ_{pump}	$2 \cdot \sigma_{\text{base}}$
Volatilidad en dump	σ_{dump}	$2 \cdot \sigma_{\text{pump}}$
Volatilidad post-dump	σ_{post}	$\sigma_{\text{base}}/2$
Factor de manipulación	m	1.0 (sutil), 4.0 (grosera)

Cuadro 4.3: Resumen de parámetros utilizados para la simulación sintética con MBG. Los términos $U(\cdot)$ representan variables aleatorias uniformes independientes, introduciendo heterogeneidad entre trayectorias.

El evento de manipulación se implementa como un cambio por tramos en los parámetros (μ_t, σ_t): antes del evento se utiliza $(\mu_{\text{base}}, \sigma_{\text{base}})$, durante la fase *pump* se incrementa la deriva y la volatilidad, durante la fase *dump* se impone una deriva fuertemente negativa y mayor volatilidad, y finalmente se aplica un régimen post-evento con deriva negativa moderada y menor volatilidad. La intensidad del evento se controla mediante el multiplicador m, permitiendo construir escenarios de manipulación sutil y grosera.

4.3.2 Parámetros de Simulación Modelo de Heston

Los parámetros base para cada esquema se escogen de acuerdo a al estudio realizado por Aggarwal y Wu [1] para μ y para el resto de parámetros nos basaremos en lo aportado por Guterding y Boenkost [23].

Los parámetros estructurales generales y específicos para el modelo de Heston se detallan en la Tabla 4.4.

Parámetro	Símbolo	Valor / Rango
Configuración Temporal		
Tiempo de Simulación	T	6.5 horas (Jornada bursátil)
Paso de tiempo	Δt	5 segundos
Puntos por curva	N	$\approx 4,680$ observaciones
Parámetros Estocásticos (Base Heston)		
Precio Inicial	S_0	$U(1, 200)$
Deriva base	μ_{base}	$0.0169 + U(-0.4880, 0.4880)$
Varianza base	θ_{base}	$0.05 + U(0, 3)$
Velocidad de reversión	κ	$U(1, 6)$
Correlación precio-var	ρ	$U(-1, -0.06)$
Volatilidad de la var.	ξ	$U(0.2, 0.8)$
Parámetros de Manipulación (Pump & Dump)		
Fracción de inicio del evento	–	$[0.05, 0.25] \cdot N$
Duración del pump	–	$[0.30, 0.40] \cdot N$
Duración del dump	–	$[0.10, 0.20] \cdot N$
Deriva en pump	μ_{pump}	$\lvert\mu_{\text{base}} + U(0, 0.8933)\rvert \cdot m$
Deriva en dump	μ_{dump}	$-\exp(\lvert\mu_{\text{pump}}\rvert)$
Deriva post-dump	μ_{post}	$-\lvert\mu_{\text{base}}\rvert$
Varianza en pump	θ_{pump}	$2 \cdot \theta_{\text{base}}$
Varianza en dump	θ_{dump}	$2 \cdot \theta_{\text{pump}}$
Varianza post-dump	θ_{post}	$\theta_{\text{base}}/2$
Factor de manipulación	m	1.0 (sutil), 4.0 (grosera)

Cuadro 4.4: Resumen de los parámetros utilizados para la generación de trayectorias sintéticas.

4.3.3 Parámetros de Simulación Modelo de Cont–Müller

Los parámetros del modelo de microestructura de Cont–Müller se fijan de forma similar a MBG y Heston: se usa una configuración temporal común (una jornada bursátil) y luego se muestrean parámetros dentro de rangos para que no todas las trayectorias sean iguales. La diferencia es que aquí no solo se simula el precio, sino también las profundidades bid y ask (D_t^b, D_t^a), que afectan directamente la dinámica del precio.

La Tabla 4.5 resume la configuración temporal, los rangos base (no manipulados) y los parámetros del esquema de manipulación tipo *pump & dump*. En el código, la manipulación se introduce cambiando por tramos los niveles de equilibrio de profundidad $(\overline{D}_b, \overline{D}_a)$ y las velocidades de reversión (ν_b, ν_a) durante las ventanas de *pump* y *dump*. Además, se multiplica la parte estocástica del precio por un factor fijo (en el código, 2.0) durante *pump* y *dump* para representar mayor ruido intradía en esos periodos.

Parámetro	**Símbolo**	**Valor / Rango**
Configuración Temporal		
Tiempo de simulación	T	6.5 horas (jornada bursátil)
Paso de tiempo	Δt	5 segundos
Puntos por trayectoria	N	$\approx 4{,}680$ observaciones
Parámetros Base (Cont–Müller, no manipulado)		
Precio inicial	S_0	$U(1, 200)$
Profundidad de equilibrio (base)	$\overline{D}$	$U(2800, 6500)$
Reversión a la media (base)	ν	$U(0.1, 2.0)$
Volatilidad de profundidad	σ_d	$U(0.1, 0.8)$
Impacto de mercado	θ	$U(7, 9)$
Correlación bid–ask	ρ	$U(-0.3, -0.01)$
Parámetros de Manipulación (Pump & Dump estructural)		
Fracción de inicio del evento	–	$U(0.15, 0.35) \cdot N$
Duración del pump	–	$U(0.10, 0.20) \cdot N$
Duración del dump	–	$U(0.05, 0.15) \cdot N$
Factor de intensidad por escenario	m	1.0 (sutil), 4.0 (grosera)
Factor interno de intensidad	η	$U(1, 3)$
Incremento de reversión en pump (bid/ask)	–	$U(0, 0.5)$
Incremento de reversión en dump (bid/ask)	–	$U(0, 0.5)$
Multiplicador del ruido en el precio (pump/dump)	–	2.0
Regímenes aplicados sobre $(\overline{D}_b, \overline{D}_a, \nu_b, \nu_a)$		
Pump: equilibrio bid	$\overline{D}_b(t)$	$\overline{D} \cdot \exp(\eta) \cdot m$
Pump: equilibrio ask	$\overline{D}_a(t)$	$\overline{D}/(\eta \cdot m)$
Dump: equilibrio bid	$\overline{D}_b(t)$	$\overline{D}/(\eta \cdot m)$
Dump: equilibrio ask	$\overline{D}_a(t)$	$\overline{D} \cdot \exp(\eta \cdot m)$
Post: reversión (regularización)	$\nu_b(t), \nu_a(t)$	$\exp(\lvert\nu\rvert)$

Cuadro 4.5: Resumen de parámetros para la generación de trayectorias sintéticas con el modelo de Cont–Müller y manipulación estructural tipo pump-and-dump.

La elección del rango $\theta \in [7, 9]$ se justifica por escala. En este modelo, θ controla el tamaño del impacto en precio asociado a desequilibrios en las profundidades. Con valores pequeños, el precio se mueve solo centavos o micro-centavos incluso en eventos manipulados, lo que vuelve la señal difícil de detectar y además no queda comparable con MBG y Heston (donde se observan movimientos de mayor magnitud). Con valores demasiado altos el precio se vuelve inestable. Por eso se fija ese rango, buscando variaciones razonables y comparables entre modelos.

Varios parámetros del esquema de manipulación (por ejemplo, los multiplicadores sobre $\overline{D}_b$ y $\overline{D}_a$, los incrementos en ν_b y ν_a, y el multiplicador del ruido del precio) se calibran por prueba y error. El objetivo no es ajustar el modelo a un mercado real específico, sino generar episodios de *pump & dump* que sean visibles en precio y en las curvas de liquidez, pero sin producir trayectorias explosivas o irreales. Esta misma lógica se aplica al impacto de mercado: se buscan magnitudes coherentes con el resto de la

memoria y con la interpretación del evento (sutil vs grosero).

Para mantener estabilidad numérica, en cada paso se impone un mínimo a las profundidades ($D_t^b, D_t^a \geq 10$) y al precio ($S_t \geq 0.01$).

Finalmente, en el caso Cont–Müller no se realiza una búsqueda exhaustiva de parámetros de base funcional (número de B-splines, orden, familia/nivel de wavelets). La razón es computacional: cada trayectoria tiene tres curvas (retornos, D_t^b, D_t^a), y una optimización por validación cruzada para bases incrementa fuertemente el tiempo total, especialmente con datasets grandes y múltiples escenarios (desbalance e intensidad). Por consistencia experimental se fijan valores estándar de base (por ejemplo, n_{basis} y nivel wavelet) y se concentra la búsqueda de hiperparámetros en los clasificadores, manteniendo el flujo comparable con MBG y Heston.

Es importante notar que para la generación de los conjuntos de datos, se utilizaron semillas aleatorias independientes ('RandomState' y 'SeedSequence') para asegurar la reproducibilidad de los experimentos y evitar la filtración de datos entre conjuntos.

4.3.4 Escenarios de Manipulación y Desbalance

Dado que el fraude financiero es un evento poco frecuente, se diseñaron experimentos considerando distintos niveles de desbalance de clases y severidad de la manipulación. Se generaron un total de $N_{total} = 10,000$ curvas para cada escenario experimental, variando los siguientes factores:

1. Intensidad de la Manipulación:
   - *Sutil:* El multiplicador de deriva ($\mu_{\text{multiplier}}$) es 1.0, generando patrones que se mezclan con el ruido natural del mercado.
   - *Grosera:* El multiplicador es 4.0, creando picos y caídas más pronunciados y evidentes.
2. Ratio de Desbalance (ρ): Se evaluaron escenarios donde la clase minoritaria (manipulada) representa el $1\,\%$, $5\,\%$ y $10\,\%$ del total de los datos.

Las ventanas de tiempo para el esquema *pump-and-dump* (inicio, duración del pump, duración del dump) fueron aleatorizadas para evitar que los clasificadores aprendieran una ubicación temporal fija del fraude.

4.3.5 Flujo de Trabajo

Como se mencionó en el capítulo 3. El proceso se divide en etapas secuenciales, ilustradas en la Figura 4.4:

1. Generación de datos: simulación de trayectorias de precios u otras características asociadas al modelo, y la conversión a retornos logarítmicos $r_t = \ln(S_t/S_{t-1})$.
2. Representación y preprocesamiento funcional: transformación de los datos discretos a funciones continuas. La representación funcional (B-splines o wavelets) se ajusta exclusivamente utilizando el conjunto de entrenamiento. Dependiendo del escenario y del costo computacional, los parámetros de la base (por ejemplo, número de funciones base o nivel de descomposición) se fijan a valores estándar o se seleccionan mediante validación cruzada dentro del conjunto de entrenamiento.
3. Reducción de dimensión (FPCA, cuando aplica): extracción de las primeras $K = 5$ componentes principales funcionales. En el caso multivariado, se ajusta FPCA por coordenada usando solo entrenamiento y se concatenan los *scores* para obtener $\mathbf{z}_i \in \mathbb{R}^{3K}$; esta formulación también se adapta a las capacidades prácticas de la librería utilizada.

4. Clasificación supervisada: a partir de la representación funcional se consideran dos estrategias complementarias:

   - Enfoque basado en reducción de dimensión: los scores FPCA $\mathbf{z}_i$ se utilizan como variables de entrada para clasificadores multivariados clásicos (Random Forest, SVM, KNN, Naive Bayes y MLP).
   - Enfoque funcional directo: las curvas suavizadas se comparan directamente en el espacio funcional mediante una métrica L^2, utilizando el clasificador k-Vecinos Más Cercanos funcional implementado en `skfda`, sin reducción previa de dimensión. Esto último se ve de manera más explicita en la Figura 4.4.

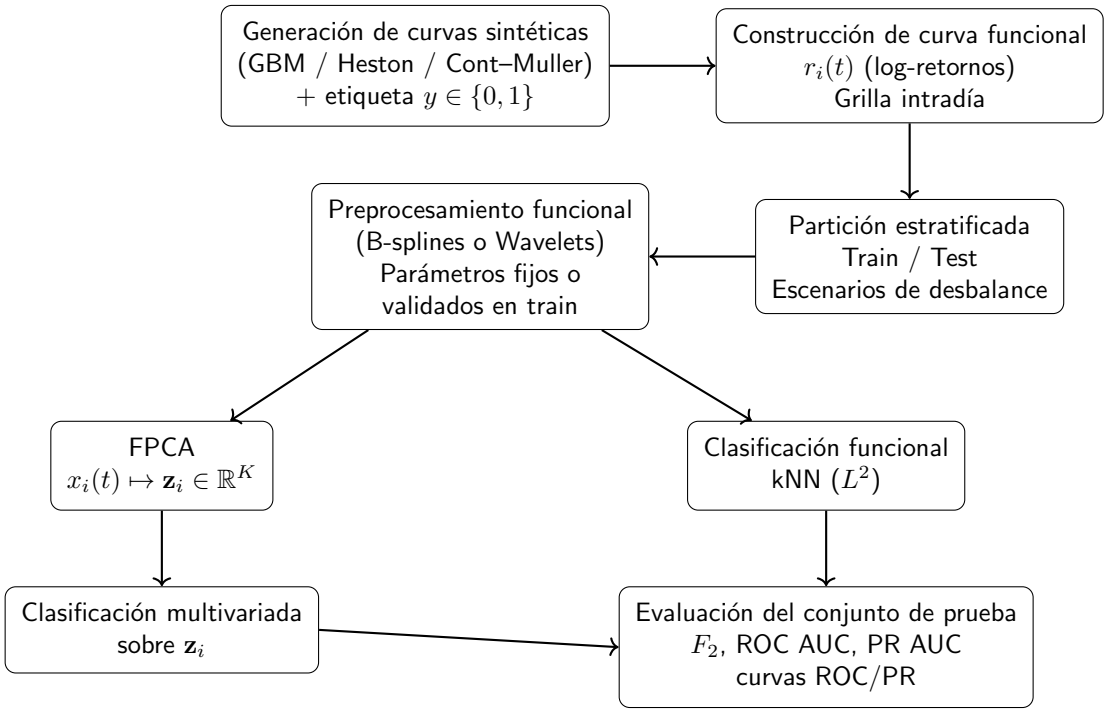


Figura 4.4: Flujo de trabajo experimental para clasificación funcional con datos sintéticos.

Cabe destacar que el uso de FPCA no es obligatorio: mientras que los clasificadores multivariados operan sobre los scores funcionales, el clasificador kNN funcional trabaja directamente sobre las curvas suavizadas, utilizando distancias funcionales sin reducción previa de dimensión.

4.4 Resultados Experimentales en Curvas Sintéticas

El Anexo B reúne los resultados completos de clasificación para los tres modelos de simulación (Movimiento Browniano Geométrico, Heston y Cont–Müller). Para cada modelo se reportan tablas comparativas con *Accuracy*, *Precision*, *Recall*, F_2, ROC AUC y PR AUC, además de las curvas ROC/PR asociadas. Los experimentos se organizaron por nivel de desbalance (1 %, 5 % y 10 %), dos intensidades de manipulación (Sutil y Grosera) y dos representaciones funcionales (B-Splines y Wavelets) como se mencionó anteriormente.

4.4.1 Movimiento Browniano Geométrico

En MBG el desempeño aumenta de manera clara cuando el desbalance deja de ser extremo. Con 1 % de clase minoritaria, el F_2 es bajo o mediano, lo que indica que la detección efectiva de la clase minoritaria sigue siendo limitada (Tablas B.1, B.2, B.3, B.4). En este régimen, Wavelets suele entregar los mejores valores de F_2 y PR AUC de manera conjunta frente a B-Splines. Al pasar a 5 % y 10 %, F_2 sube de forma sostenida y se observan combinaciones más estables, con SVM y Random Forest como alternativas competitivas en ambos tipos de base (Tablas B.5–B.8 y B.9–B.12). Esto es coherente con las curvas ROC/PR, donde las curvas PR se separan más claramente de la línea base a medida que aumenta la prevalencia de la clase minoritaria (Figuras B.6–B.12). Y para el kNN funcional se observa una gran mejora en escenarios de menor desbalance.

Podemos resumir el F_2-score global como en la Figura 4.5 que consolidan el F_2 de todos los escenarios en una sola vista: el color representa el desbalance (1 %, 5 %, 10 %), si esta vacío o lleno (Sutil/Grosera) y la forma la base (círculo: B-Splines, triángulo: Wavelets). Estos gráficos facilitan ver rápidamente qué combinaciones de clasificador/base son más estables al cambiar el desbalance y la intensidad de manipulación. En la siguiente figura 4.5 se observa que la métrica es mayor en los escenarios groseros y también en los menos desbalanceados. Además, el kNN funcional muestra un rendimiento bajo en este caso.

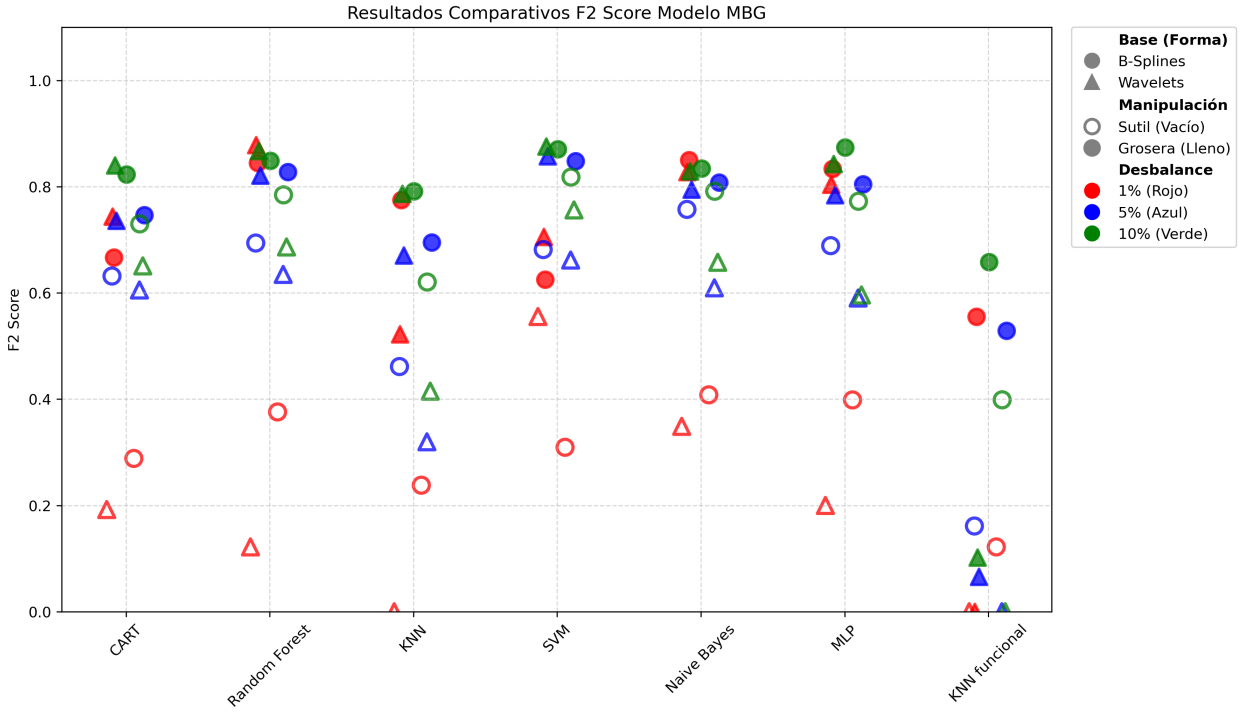


Figura 4.5: Resumen global de F_2 para MBG: desbalance (color), manipulación (transparencia) y base (forma).

4.4.2 Modelo de Heston

En Heston el efecto dominante no es solo el desbalance, sino también la intensidad de manipulación. Con 1 % y manipulación sutil, el rendimiento sobre la minoritaria es bajo en términos de F_2, al pasar a manipulación grosera en el mismo desbalance, varios clasificadores alcanzan mejoras sustanciales en *Recall* y F_2, reflejando que la manipulación grosera induce una señal mucho más separable (Tablas B.13–

B.16). Para 5 % y 10 %, las configuraciones groseras tienden a concentrar los mejores resultados globales (incluyendo PR AUC alta), destacando SVM, Random Forest y Naive Bayes según la representación, mientras que en escenarios sutiles las mejoras son más graduales y dependen más del clasificador (Tablas B.17–B.20 y B.21–B.24). Un punto relevante es que el KNN funcional no muestra estabilidad comparable en Heston: en varios escenarios de desbalance severo su F_2 es cercano a cero, lo que sugiere que, bajo esta parametrización, la distancia funcional utilizada por el método no capta bien la señal de la clase minoritaria (Tablas B.13, B.14, B.16, B.18, B.22). Las curvas ROC/PR son consistentes con esta lectura: la separación en PR se fortalece especialmente en los casos groseros y con desbalances menos extremos (Figuras B.18–B.24).

Podemos resumir el F_2-score global en la Figura 4.6. Se observa que la métrica es mayor en los escenarios groseros y también en los menos desbalanceados. Además, el kNN funcional muestra un rendimiento bajo en este caso.

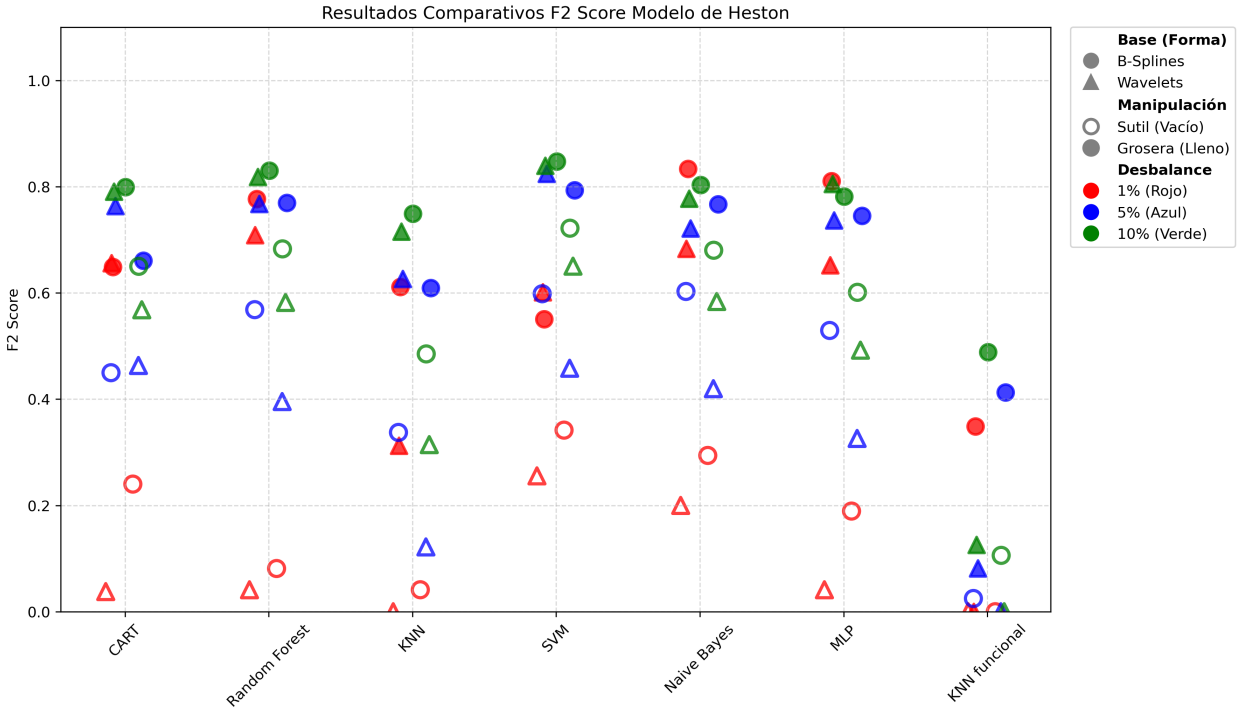


Figura 4.6: Resumen global de F_2 para Heston: desbalance (color), manipulación (transparencia) y base (forma).

4.4.3 Modelo Cont–Müller

En Cont–Müller los resultados no son más débiles; de hecho, en esta batería de experimentos el modelo tiende a ser el más detectable, especialmente bajo manipulación grosera. Incluso con 1 % y manipulación sutil, ya se observan F_2 moderados en varias combinaciones y una ventaja relativa de Wavelets en el mejor desempeño (Tablas B.25, B.26). Al pasar a manipulación grosera, las métricas se acercan a valores casi perfectos incluso con 1 % de minoritaria, y en 5 % aparecen resultados esencialmente saturados (valores de F_2 y AUC cercanos a 1 en múltiples clasificadores y también en KNN funcional), lo que indica una separación extremadamente marcada entre clases bajo la parametrización utilizada (Tablas B.27, B.28, B.31, B.32). En 10 % la lectura se mantiene: en sutil se alcanzan desempeños altos y en grosera se conserva una separación casi total (Tablas B.33–B.36). Este patrón es

coherente con la construcción multivariada del dataset (retornos, D^b y D^a): al incorporar profundidades del libro, el clasificador dispone de variables directamente afectadas por el mecanismo de manipulación, lo que vuelve el problema más separable que cuando se usan solo retornos. Las curvas ROC/PR acompañan esta conclusión, con separación clara en PR incluso en escenarios de baja prevalencia cuando la manipulación es grosera (Figuras B.26–B.36). Como limitación, la presencia de métricas casi perfectas en varios escenarios groseros sugiere un posible efecto de "techo": el caso grosero, tal como fue parametrizado, podría representar un escenario muy favorable (y potencialmente más simple que la realidad), por lo que estos resultados deben interpretarse como un límite superior de desempeño bajo simulación.

Podemos resumir el F_2-score global en la Figura 4.7. Se observa que la métrica es casi perfecta en los escenarios groseros y también en los menos desbalanceados. Además, el kNN funcional en este caso muestra una clara mejora a la hora de clasificar activos como manipulados, superando incluso escenarios groseros de modelos anteriores.

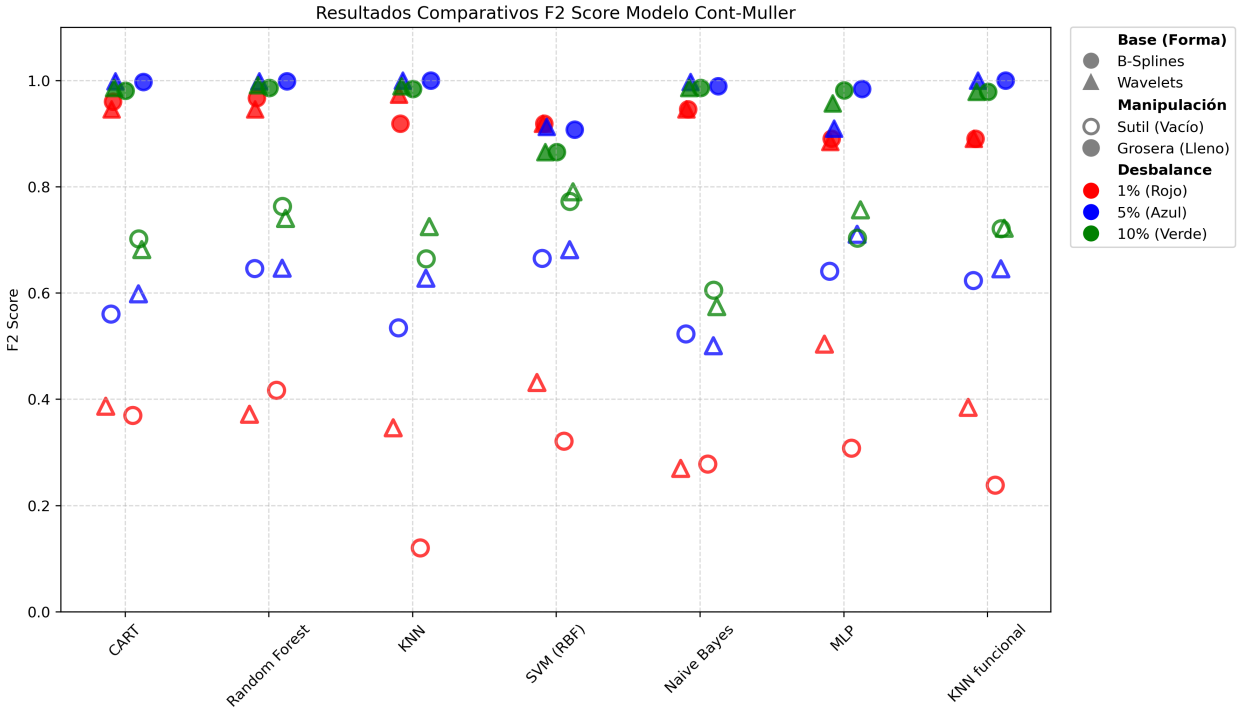


Figura 4.7: Resumen global de F_2 para Cont-Müller: desbalance (color), manipulación (transparencia) y base (forma).

En conjunto, los resultados resumidos en las Figuras 4.5, 4.6 y 4.7 muestran patrones robustos, aunque no necesariamente monótonos en todos los escenarios. En general, el desbalance reduce la estabilidad de la detección y vuelve insuficiente la *Accuracy*, por lo que F_2 y las curvas PR resultan más informativas para comparar modelos. No obstante, la intensidad de la manipulación puede dominar el efecto del desbalance: incluso con prevalencia muy baja (1 %), los escenarios groseros producen separabilidad alta y métricas elevadas.

A nivel comparativo, MBG tiende a ser el caso más exigente cuando la manipulación es sutil, mientras que Heston y, sobre todo, Cont-Müller generan señales más detectables bajo manipulación grosera. En MBG, Wavelets obtiene el mejor F_2 en 5 de 6 combinaciones (la excepción corresponde a desbalance 5 % y manipulación sutil, donde B-Splines + Naive Bayes resulta superior). Finalmente, bajo la parametrización utilizada, Cont-Müller puede inducir escenarios con separabilidad cercana a uno,

lo que debe interpretarse como un caso favorable dentro del marco de simulación (límite superior de desempeño) y no como garantía directa de resultados equivalentes en datos reales.

4.4.4 Discusión general de resultados sintéticos

En conjunto, los experimentos con curvas sintéticas muestran tres regularidades transversales. Primero, el desbalance domina el desempeño del modelo: cuando la clase manipulada es muy rara (1 %), la *Accuracy* deja de ser informativa y las métricas orientadas a la clase minoritaria (en particular F_2 y las curvas PR) son las que reflejan mejor la capacidad real de detección. Este patrón se observa en MBG y también en Heston bajo manipulación sutil, donde es posible mantener una *Accuracy* alta incluso cuando el *Recall* de la minoritaria es bajo o nulo en algunos métodos. En segundo lugar, la intensidad del evento afecta directamente la separabilidad: al pasar de manipulación sutil a grosera, típicamente aumentan el *Recall* y F_2, y las curvas PR se separan más claramente de la línea base, especialmente en Heston y Cont-Müller. En tercer lugar, la estructura del generador de datos determina cuánta señal está disponible: MBG, al depender solo de cambios por tramos en (μ_t, σ_t) sobre una dinámica de precio relativamente simple, tiende a constituir el escenario más exigente cuando la manipulación es sutil; Heston agrega dinámica estocástica de varianza y, en consecuencia, puede amplificar la señal cuando la manipulación es grosera; mientras que Cont-Müller, al incorporar explícitamente variables de microestructura (D_t^b y D_t^a) directamente intervenidas por el mecanismo de manipulación, produce escenarios donde la detección se vuelve significativamente más fácil.

En cuanto a la representación funcional, Wavelets suele entregar ventajas en escenarios difíciles (desbalance extremo y/o manipulación sutil), lo que es consistente con un suavizado más apto para separar señal localizada de ruido intradía. Sin embargo, esta ventaja no es universal y depende del clasificador: métodos como SVM y Random Forest tienden a mantenerse competitivos en ambos tipos de base cuando la prevalencia aumenta. Respecto al enfoque funcional directo, el kNN funcional presenta un comportamiento dispar: en Heston no muestra estabilidad (incluyendo casos con F_2 cercano a cero), lo que sugiere que, bajo la parametrización utilizada y con la métrica funcional adoptada, la distancia L^2 no captura de manera robusta la diferencia entre clases; en cambio, en Cont–Müller el mismo enfoque mejora, coherente con que el espacio funcional ahora incluye variables con intervención directa y, por tanto, mayores diferencias geométricas entre trayectorias manipuladas y no manipuladas.

Finalmente, los resultados casi perfectos observados en varios escenarios groseros de Cont-Müller deben interpretarse con cautela. Dado que la manipulación altera de forma explícita regímenes de profundidad y se refuerza el ruido del precio por tramos, el experimento puede inducir un efecto de "techo" en el que múltiples clasificadores alcanzan métricas cercanas a 1. En este caso, el valor principal del ejercicio no es concluir que el problema real es "fácil", sino identificar un límite superior de desempeño bajo un generador que contiene señal fuerte y controlada. Esta observación motiva, como puente hacia los datos reales, que la comparación se apoye en PR AUC y F_2 y que se discuta explícitamente la brecha entre un mecanismo de simulación con intervención observable y un mercado donde la manipulación se manifiesta de forma más heterogénea y con ruido estructural adicional.

Capítulo 5

Conclusiones

5.1 Conclusiones Generales

El objetivo central de este trabajo fue evaluar el desempeño de un enfoque que integre técnicas de Análisis de Datos Funcionales y de Aprendizaje Supervisado para la detección de manipulación tipo *pump & dump* bajo dos escenarios con distinta resolución temporal: (i) datos reales diarios (OHLCV) del activo FLC, representados como un vector tabular de variables derivadas (retornos, razones de volumen y volatilidad *rolling*), y (ii) datos sintéticos intradía generados en una grilla fija (ticks de 5 segundos), que permiten una representación funcional y el uso de suavizado (B-splines/Wavelets), reducción de dimensión mediante FPCA y, adicionalmente, un enfoque funcional directo con kNN basado en distancia L^2. Esta separación dejó en evidencia que la representación de entrada condiciona qué métodos son aplicables: con datos diarios no se pueden construir curvas intradía comparables sin supuestos extra, mientras que en simulación sí es posible explotar la estructura funcional.

Los experimentos con datos reales mostraron que el desbalance severo es el factor dominante. Sin balanceo (SMOTE y/o ponderación de clases), la mayoría de los clasificadores termina prediciendo casi todo como "no manipulado", obteniendo *Accuracy* alta pero con *Recall* y F_2 cercanos a cero. Al aplicar balanceo en el entrenamiento y estandarizar las variables, los modelos empiezan a detectar parte de la clase manipulada, pero a cambio aparece un aumento de falsas alarmas. En esta configuración, Random Forest fue el modelo con mejor desempeño global en FLC según F_2, con resultados razonables en *Precision* y *Recall*. Por lo mismo, usar F_2 y curvas PR fue clave para una evaluación realista, ya que la exactitud por sí sola no representa bien el desempeño cuando los eventos positivos son raros.

En datos sintéticos, los resultados mostraron patrones consistentes: al disminuir el desbalance (de 1 % a 10 %) aumentan de forma sistemática F_2 y PR AUC; además, al incrementar la intensidad de manipulación la separabilidad crece, especialmente en Heston y con mayor fuerza en Cont-Müller. En MBG el escenario es más difícil bajo desbalance extremo, y el preprocesamiento (en particular Wavelets en varios casos) puede aportar mejoras cuando la señal es débil. En Cont–Müller se obtuvieron desempeños muy altos (incluso cercanos al techo) porque la construcción multivariada incorpora variables de microestructura (D_t^b y D_t^a) que están directamente intervenidas por el mecanismo de manipulación, lo que vuelve el problema mucho más separable que usando solo retornos. Por lo mismo, estos resultados deben interpretarse como un límite superior favorable dentro del marco de simulación y no como una garantía directa de desempeño en mercados reales.

En conjunto, la evidencia del capítulo apoya tres conclusiones prácticas: Primero, el balanceo de clases y la evaluación con métricas orientadas a la minoritaria son condiciones necesarias para que el problema sea abordable; segundo, con datos reales diarios la detección está fuertemente limitada por la resolución temporal y por la escasez de etiquetas positivas, por lo que el desempeño debe entenderse

como detección parcial bajo incertidumbre; y por último, en alta frecuencia, una representación funcional puede capturar mejor la dinámica intradía, pero su utilidad en la práctica depende de contar con datos reales intradía y de calibrar simulaciones que no induzcan separabilidad artificialmente alta.

También para el trabajo, debemos tomar en cuenta que la principal limitación del estudio proviene de la disponibilidad y resolución de los datos reales. En el caso FLC, la información está a frecuencia diaria (OHLCV), por lo que cada muestra resume un día completo y no permite reconstruir ni comparar formas intradía del tipo *pump and dump* sin introducir supuestos adicionales. Esto restringe el alcance del enfoque funcional en datos reales: el análisis de curvas se evalúa en condiciones controladas mediante simulación, pero no puede validarse directamente con series intradía reales del mismo activo y periodo.

Una segunda limitación importante es la escasez en las etiquetas positivas. Los eventos confirmados de manipulación son pocos en relación con el total, lo que induce un desbalance extremo y vuelve los resultados sensibles a la partición entrenamiento/prueba. Además, la etiqueta "manipulado" a nivel diario no necesariamente coincide con un patrón único o uniforme: un mismo día puede contener señales mixtas, o la manipulación puede extenderse en varios días, lo que introduce ruido de etiquetado y reduce el máximo desempeño esperable.

En lo metodológico, se aplicaron técnicas de balanceo como SMOTE y/o ponderación de clases, pero estas no garantizan representar correctamente la distribución real de fraudes. En particular, SMOTE genera ejemplos sintéticos interpolados que pueden crear patrones poco realistas o suavizar en exceso la frontera entre clases; por ello, los resultados deben interpretarse como una evaluación comparativa de clasificadores bajo un esquema estándar de mitigación del desbalance, y no como una estimación exacta de desempeño operativo en producción.

Respecto a los datos sintéticos, las conclusiones dependen de las decisiones de parametrización y de cómo se implementó la manipulación. Los modelos estocásticos (MBG, Heston y Cont-Müller) son aproximaciones y no capturan toda la complejidad del mercado (microestructura real, cambios de régimen, noticias, saltos, etc.). Además, algunos parámetros del esquema de manipulación se calibraron por prueba y error para producir episodios visibles y estables, lo que puede inducir separabilidad artificialmente alta en ciertos escenarios (especialmente en Cont-Müller, donde se intervienen variables directamente ligadas al mecanismo). En consecuencia, los resultados en simulación deben leerse como límites dentro del marco propuesto, no como garantía de generalización a datos reales.

Finalmente, existe una limitación computacional y de diseño experimental: no se realizó una optimización exhaustiva de todos los hiperparámetros del preprocesamiento funcional (por ejemplo, elección fina de bases, niveles de wavelets o número de componentes), y se fijaron configuraciones estándar para mantener el costo razonable y la comparabilidad entre escenarios. Esto implica que el desempeño reportado para B-splines/Wavelets y FPCA podría mejorar con una búsqueda más profunda, pero a costa de mayor complejidad y tiempo de cómputo. En conjunto, estas limitaciones acotan la interpretación del trabajo a un estudio metodológico comparativo, útil para entender condiciones de aplicabilidad y sensibilidad del enfoque, más que para proponer un sistema listo para despliegue regulatorio.

5.2 Trabajos Futuros

Como líneas de extensión natural de este trabajo, se proponen las siguientes:

1. Incorporar datos reales intradía (ticks o barras de alta frecuencia) para evaluar el enfoque funcional en un escenario realista y comparable con la simulación.

2. Refinar la calibración de los modelos sintéticos para reproducir hechos estilizados observados en mercados reales, reduciendo el riesgo de escenarios con separabilidad "demasiado fácil".

3. Explorar modelos de clasificación adicionales y comparables: *gradient boosting* (e.g., XGBoost/LightGBM).

4. Profundizar el análisis funcional multivariado: FPCA conjunta (no por coordenada), métricas funcionales alternativas, kernels funcionales, y técnicas que separen forma vs amplitud para capturar mejor patrones tipo *pump and dump*.

Capítulo A

Anexo 1

En este Anexo, se presentará resultados computacionales y configuraciones para la base datos real (FLC Group) utilizada.

A.1 Resultados Computacionales y Configuraciones

A.1.1 Árbol de Decisión

En esta sección se describe la implementación del modelo *Classification and Regression Tree (CART)*, utilizando el modelo *DecisionTreeClassifier* de la librería *scikit-learn*. La fase de entrenamiento se realizó empleando el criterio de impureza *Gini*, junto con la estrategia de ponderación de clases (*class_weight='balanced'*) para mitigar el desbalance al problema. Los hiperparámetros utilizados se detallan en la Tabla A.1. Posteriormente, se presentan la matriz de confusión (Tabla A.2) y las métricas de desempeño obtenidas sobre el conjunto de prueba (Tabla A.3).

Parámetro	**Valor**
criterion	gini
class_weight	balanced
max_depth	5
min_samples_leaf	4
random_state	42

Cuadro A.1: Parámetros de entrenamiento del clasificador CART.

Etiqueta Real	**Predicción: Normal**	**Predicción: Manipulación**
Normal	615	23
Manipulación	1	7

Cuadro A.2: Matriz de confusión del clasificador CART sobre el conjunto de prueba.

Métrica	Valor
Exactitud	0.9644
Precisión	0.2222
Sensibilidad	0.7500
F_2-Score	0.5085
ROC AUC	0.8585
PR AUC	0.1698

Cuadro A.3: Métricas del clasificador CART sobre el conjunto de prueba.

A.1.2 Bosques Aleatorios

En esta sección, se detalla la implementación del modelo *RandomForestClassifier* en Python, utilizando la librería *scikit-learn*, utilizando los parámetros de la tabla A.4. A continuación, se muestra a tabla de métricas A.6 y la tabla de matriz de confusión A.5 obtenidas luego de evaluar con los datos de prueba.

Parámetro	Valor
n_estimators	200
class_weight	balanced_subsample
max_depth	5
min_samples_leaf	4

Cuadro A.4: Parámetros de entrenamiento RandomForestClassifier.

Etiqueta Real	Predicción: Normal	Predicción: Manipulación
Normal	618	20
Manipulación	1	7

Cuadro A.5: Matriz de confusión del clasificador Random Forest sobre el conjunto de prueba.

Métrica	Valor
Exactitud	0.9721
Precisión	0.2222
Sensibilidad	0.5000
F_2-Score	0.4000
ROC AUC	0.9781
PR AUC	0.2589

Cuadro A.6: Métricas del clasificador Random Forest sobre el conjunto de prueba.

Adicionalmente, se adjunta la tabla A.7, que presenta la importancia que se determino mediante el algoritmo. Esta refleja las variables más relevantes dentro de la clasificación.

Característica	Importancia
Volume MA	0.5515
Volume	0.2155
rolling_vol	0.1761
log_ret	0.0351
vol_ratio	0.0206

Cuadro A.7: Variables con mayor importancia según Random Forest.

A.1.3 K-Vecinos más Cercanos

En esta sección se describe la implementación del clasificador *K-Nearest Neighbors (KNN)*, utilizando el modelo *KNeighborsClassifier* de la librería *scikit-learn*. El modelo fue entrenado considerando $k = 5$ vecinos más cercanos, empleando un esquema de ponderación por distancia (*weights='distance'*) y la métrica euclidiana como medida de similitud. Los hiperparámetros utilizados se detallan en la Tabla A.8. Posteriormente, se presentan la matriz de confusión (Tabla A.9) y las métricas de desempeño obtenidas sobre el conjunto de prueba (Tabla A.10).

Parámetro	Valor
n_neighbors	5
weights	distance
metric	euclidean

Cuadro A.8: Parámetros de entrenamiento del clasificador KNN.

Etiqueta Real	Predicción: Normal	Predicción: Manipulación
Normal	611	27
Manipulación	2	6

Cuadro A.9: Matriz de confusión del clasificador KNN sobre el conjunto de prueba.

Métrica	Valor
Exactitud	0.9551
Precisión	0.1818
Sensibilidad	0.7500
F_2-Score	0.4615
ROC AUC	0.8488
PR AUC	0.1425

Cuadro A.10: Métricas del clasificador KNN sobre el conjunto de prueba.

A.1.4 Máquinas de Vectores de Soporte

En esta sección se describe la implementación del clasificador *Support Vector Machine (SVM)* con kernel radial (*RBF*), utilizando el modelo *SVC* de la librería *scikit-learn*. Con el fin de abordar el marcado desbalance entre clases, el modelo fue entrenado incorporando ponderación de clases

(*class_weight='balanced'*). Adicionalmente, se habilitó la estimación probabilística mediante *probability=True*, lo que permite evaluar el desempeño del clasificador a través de métricas basadas en umbrales. Los hiperparámetros utilizados se detallan en la Tabla A.11. A continuación, se presentan la matriz de confusión (Tabla A.12) y las métricas obtenidas sobre el conjunto de prueba (Tabla A.13).

Parámetro	**Valor**
kernel	rbf
class_weight	balanced
probability	True
random_state	1

Cuadro A.11: Parámetros de entrenamiento del clasificador SVM con kernel RBF.

Etiqueta Real	**Predicción: Normal**	**Predicción: Manipulación**
Normal	597	41
Manipulación	2	4

Cuadro A.12: Matriz de confusión del clasificador SVM sobre el conjunto de prueba.

Métrica	**Valor**
Exactitud	0.9334
Precisión	0.1277
Sensibilidad	0.7500
F_2-Score	0.3797
ROC AUC	0.9624
PR AUC	0.2731

Cuadro A.13: Métricas del clasificador SVM (RBF) sobre el conjunto de prueba.

A.1.5 Naive Bayes

En esta sección se presenta la implementación del clasificador *Gaussian Naive Bayes (GaussianNB)* en Python, utilizando la librería *scikit-learn*. Dado el carácter paramétrico y cerrado del modelo, este fue entrenado utilizando su configuración por defecto, sin ajuste explícito de hiperparámetros. A continuación, se muestran la matriz de confusión (Tabla A.14) y las métricas de desempeño obtenidas sobre el conjunto de prueba (Tabla A.15).

Etiqueta Real	**Predicción: Normal**	**Predicción: Manipulación**
Normal	602	36
Manipulación	1	7

Cuadro A.14: Matriz de confusión del clasificador Naive Bayes sobre el conjunto de prueba.

Métrica	Valor
Exactitud	0.9427
Precisión	0.1628
Sensibilidad	0.8750
F_2-Score	0.4667
ROC AUC	0.9726
PR AUC	0.3117

Cuadro A.15: Métricas del clasificador Naive Bayes sobre el conjunto de prueba.

A.1.6 Red Neuronal Artificial (MLP)

En esta sección se describe la implementación del clasificador *Multi-Layer Perceptron (MLP)*, utilizando la clase *MLPClassifier* de la librería *scikit-learn*. El modelo fue entrenado empleando una arquitectura de dos capas ocultas, con 32 y 8 neuronas respectivamente, función de activación *ReLU* y el optimizador *Adam*. Con el objetivo de mejorar la estabilidad del entrenamiento y prevenir sobreajuste, se incorporó un esquema de parada temprana (*early_stopping=True*). Los hiperparámetros utilizados se detallan en la Tabla A.16. A continuación, se presentan la matriz de confusión (Tabla A.17) y las métricas obtenidas sobre el conjunto de prueba (Tabla A.18).

Parámetro	Valor
hidden_layer_sizes	(32, 8)
activation	relu
solver	adam
max_iter	1000
early_stopping	True
random_state	42

Cuadro A.16: Parámetros de entrenamiento del clasificador MLP.

Etiqueta Real	Predicción: Normal	Predicción: Manipulación
Normal	564	74
Manipulación	1	7

Cuadro A.17: Matriz de confusión del clasificador MLP sobre el conjunto de prueba.

Métrica	Valor
Exactitud	0.8839
Precisión	0.0864
Sensibilidad	0.8750
F_2-Score	0.3097
ROC AUC	0.9520
PR AUC	0.1362

Cuadro A.18: Métricas del clasificador MLP sobre el conjunto de prueba.

Capítulo B

Anexo 2

En este Anexo se presentarán las distintas tablas comparativas para los datos sínteticos de los modelos estocásticos junto a sus curvas ROC y PR.

B.1 Movimiento Browniano Geométrico

B.1.1 Tablas Comparativas

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9687	0.1364	0.4000	0.2885	0.6905	0.2207
Random Forest	0.9923	0.7692	0.3333	0.3759	0.7878	0.4064
KNN	0.9920	1.0000	0.2000	0.2381	0.7265	0.2992
SVM	0.9403	0.1016	0.6333	0.3094	0.8295	0.1055
Naive Bayes	0.9890	0.4444	0.4000	0.4082	0.8728	0.4728
MLP	0.9913	0.6111	0.3667	0.3986	0.7819	0.3931
KNN funcional	0.9910	1.0000	0.1000	0.1220	0.7314	0.3159

Cuadro B.1: Resultados Modelo MBG B-Splines (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9900	0.5000	0.1667	0.1923	0.5825	0.0917
Random Forest	0.9910	1.0000	0.1000	0.1220	0.8587	0.5145
KNN	0.9900	0.0000	0.0000	0.0000	0.5813	0.1204
SVM	0.9857	0.3725	0.6333	0.5556	0.9324	0.4688
Naive Bayes	0.9930	1.0000	0.3000	0.3488	0.9667	0.6372
MLP	0.9917	1.0000	0.1667	0.2000	0.9058	0.5255
KNN funcional	0.9900	0.0000	0.0000	0.0000	0.5000	0.0100

Cuadro B.2: Resultados Modelo MBG Wavelets (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9860	0.4000	0.8000	0.6667	0.8982	0.6959
Random Forest	0.9973	0.8929	0.8333	0.8446	0.9782	0.8965
KNN	0.9973	1.0000	0.7333	0.7746	0.9159	0.8251
SVM	0.9740	0.2692	0.9333	0.6250	0.9940	0.6316
Naive Bayes	0.9977	0.9259	0.8333	0.8503	0.9898	0.8941
MLP	0.9980	1.0000	0.8000	0.8333	0.9926	0.8888
KNN funcional	0.9950	1.0000	0.5000	0.5556	0.9328	0.8416

Cuadro B.3: Resultados Modelo MBG B-Splines (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9907	0.5208	0.8333	0.7440	0.9156	0.7813
Random Forest	0.9980	0.9286	0.8667	0.8784	0.9764	0.9137
KNN	0.9947	1.0000	0.4667	0.5224	0.8662	0.7119
SVM	0.9910	0.5349	0.7667	0.7055	0.9464	0.6214
Naive Bayes	0.9977	0.9600	0.8000	0.8276	0.9824	0.9149
MLP	0.9977	1.0000	0.7667	0.8042	0.9743	0.8712
KNN funcional	0.9900	0.0000	0.0000	0.0000	0.5000	0.0100

Cuadro B.4: Resultados Modelo MBG Wavelets (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9193	0.3589	0.7800	0.6317	0.8565	0.6498
Random Forest	0.9730	0.7556	0.6800	0.6939	0.9467	0.7673
KNN	0.9703	1.0000	0.4067	0.4614	0.8380	0.6442
SVM	0.9277	0.3943	0.8333	0.6816	0.9534	0.7607
Naive Bayes	0.9797	0.8346	0.7400	0.7572	0.9575	0.8355
MLP	0.9800	0.9327	0.6467	0.6889	0.9473	0.7952
KNN funcional	0.9567	1.0000	0.1333	0.1613	0.8923	0.7195

Cuadro B.5: Resultados Modelo MBG B-Splines (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9393	0.4316	0.6733	0.6055	0.7970	0.6142
Random Forest	0.9757	0.8812	0.5933	0.6348	0.9384	0.7575
KNN	0.9637	1.0000	0.2733	0.3198	0.7747	0.5287
SVM	0.9630	0.6196	0.6733	0.6619	0.9165	0.7227
Naive Bayes	0.9763	0.9438	0.5600	0.6096	0.9465	0.7848
MLP	0.9753	0.9419	0.5400	0.5904	0.9401	0.7627
KNN funcional	0.9500	0.0000	0.0000	0.0000	0.5000	0.0500

Cuadro B.6: Resultados Modelo MBG Wavelets (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9497	0.4981	0.8533	0.7468	0.9130	0.8065
Random Forest	0.9857	0.8905	0.8133	0.8277	0.9644	0.8989
KNN	0.9820	0.9898	0.6467	0.6948	0.8891	0.7743
SVM	0.9773	0.7228	0.8867	0.8482	0.9775	0.9114
Naive Bayes	0.9880	0.9831	0.7733	0.8078	0.9742	0.9034
MLP	0.9870	0.9587	0.7733	0.8044	0.9744	0.9078
KNN funcional	0.9737	1.0000	0.4733	0.5291	0.9280	0.8375

Cuadro B.7: Resultados Modelo MBG B-Splines (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9727	0.7208	0.7400	0.7361	0.8625	0.5464
Random Forest	0.9850	0.8832	0.8067	0.8209	0.9791	0.9048
KNN	0.9810	1.0000	0.6200	0.6710	0.8736	0.7461
SVM	0.9840	0.8228	0.8667	0.8575	0.9854	0.9210
Naive Bayes	0.9870	0.9744	0.7600	0.7950	0.9833	0.9208
MLP	0.9867	0.9825	0.7467	0.7843	0.9828	0.9215
KNN funcional	0.9527	1.0000	0.0533	0.0658	0.5400	0.1260

Cuadro B.8: Resultados Modelo MBG Wavelets (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9103	0.5344	0.8033	0.7299	0.8851	0.7657
Random Forest	0.9603	0.8175	0.7767	0.7845	0.9637	0.8734
KNN	0.9547	0.9607	0.5700	0.6205	0.8890	0.7713
SVM	0.9477	0.6927	0.8567	0.8180	0.9570	0.8635
Naive Bayes	0.9660	0.8722	0.7733	0.7913	0.9648	0.8846
MLP	0.9660	0.8960	0.7467	0.7724	0.9648	0.8796
KNN funcional	0.9347	1.0000	0.3467	0.3988	0.9278	0.8259

Cuadro B.9: Resultados Modelo MBG B-Splines (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9117	0.5467	0.6833	0.6508	0.8071	0.6258
Random Forest	0.9460	0.7614	0.6700	0.6865	0.9292	0.7850
KNN	0.9350	0.9646	0.3633	0.4151	0.8100	0.6206
SVM	0.9433	0.6946	0.7733	0.7562	0.9391	0.8112
Naive Bayes	0.9563	0.9246	0.6133	0.6576	0.9393	0.8251
MLP	0.9487	0.8967	0.5500	0.5961	0.9193	0.7818
KNN funcional	0.9000	0.0000	0.0000	0.0000	0.5000	0.1000

Cuadro B.10: Resultados Modelo MBG Wavelets (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9373	0.6327	0.8900	0.8231	0.9263	0.8650
Random Forest	0.9743	0.8996	0.8367	0.8485	0.9715	0.9243
KNN	0.9747	0.9912	0.7533	0.7913	0.9305	0.8695
SVM	0.9630	0.7692	0.9000	0.8704	0.9730	0.9333
Naive Bayes	0.9793	0.9877	0.8033	0.8345	0.9712	0.9320
MLP	0.9810	0.9483	0.8567	0.8736	0.9717	0.9306
KNN funcional	0.9607	1.0000	0.6067	0.6585	0.9525	0.9029

Cuadro B.11: Resultados Modelo MBG B-Splines (Desbalance 0.10, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9497	0.6945	0.8867	0.8402	0.9323	0.8555
Random Forest	0.9763	0.8990	0.8600	0.8675	0.9745	0.9288
KNN	0.9733	0.9783	0.7500	0.7867	0.9321	0.8687
SVM	0.9687	0.8121	0.8933	0.8758	0.9786	0.9379
Naive Bayes	0.9790	0.9917	0.7967	0.8293	0.9768	0.9341
MLP	0.9800	0.9839	0.8133	0.8425	0.9755	0.9334
KNN funcional	0.9083	1.0000	0.0833	0.1020	0.5617	0.2110

Cuadro B.12: Resultados Modelo MBG Wavelets (Desbalance 0.10, Grosera)

B.1.2 Curvas ROC/PR

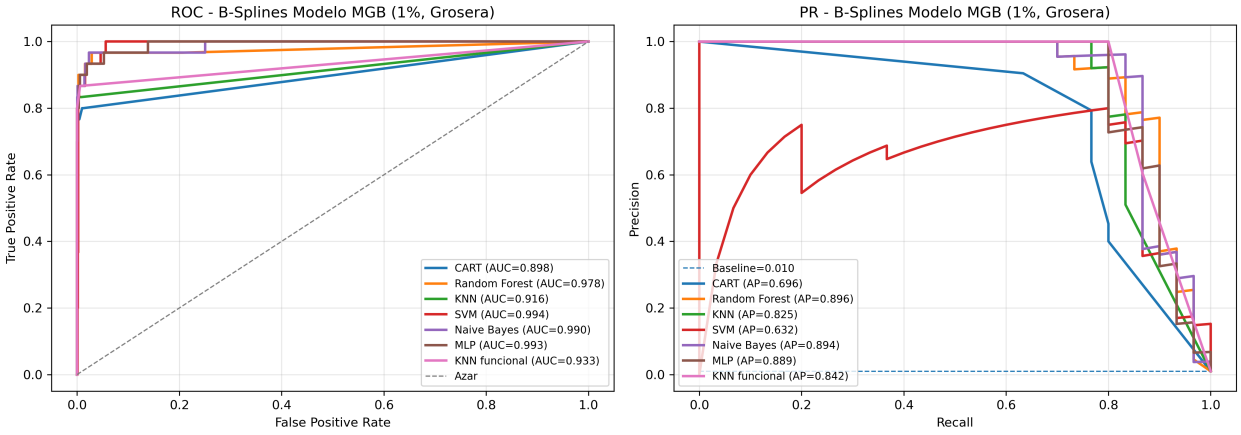


Figura B.1: ROC/PR - B-Splines (Grosera, 1 %)

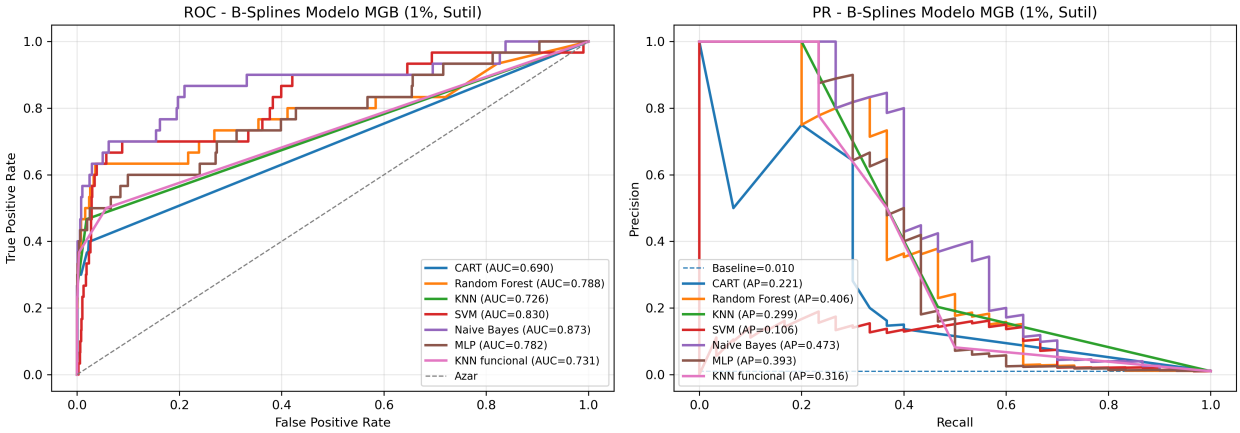


Figura B.2: ROC/PR - B-Splines (Sutil, 1 %)

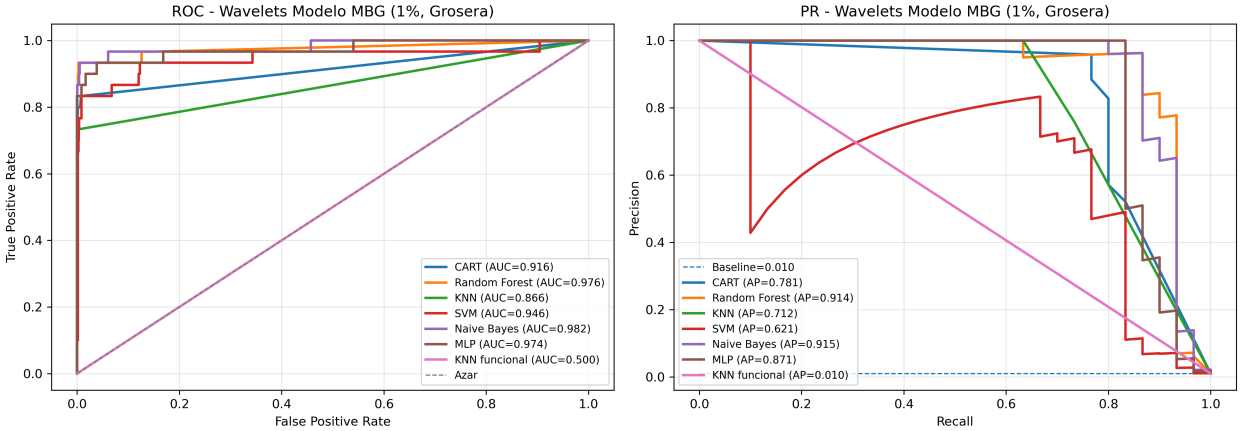


Figura B.3: ROC/PR - Wavelets (Grosera, 1 %)

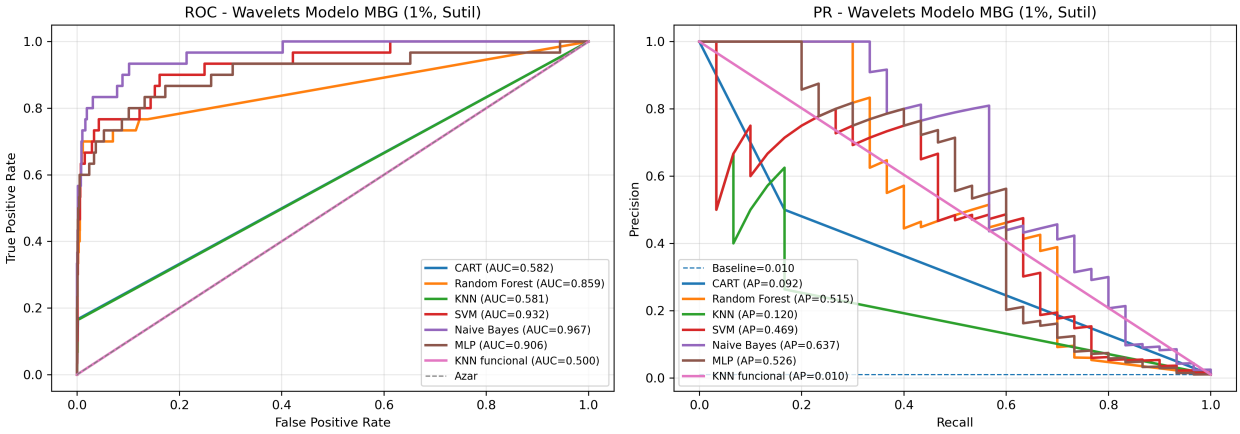


Figura B.4: ROC/PR - Wavelets (Sutil, 1 %)

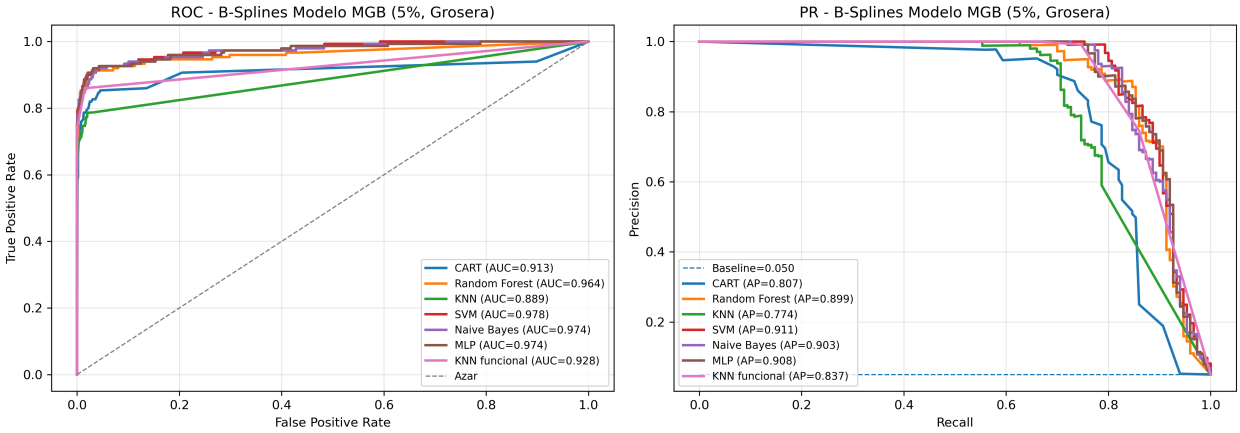


Figura B.5: ROC/PR - B-Splines (Grosera, 5 %)

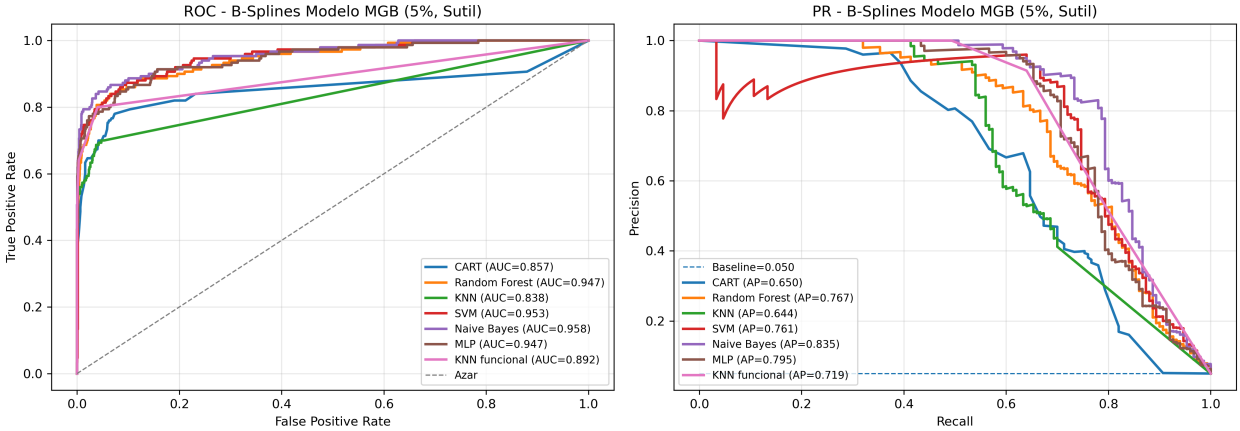


Figura B.6: ROC/PR - B-Splines (Sutil, 5 %)

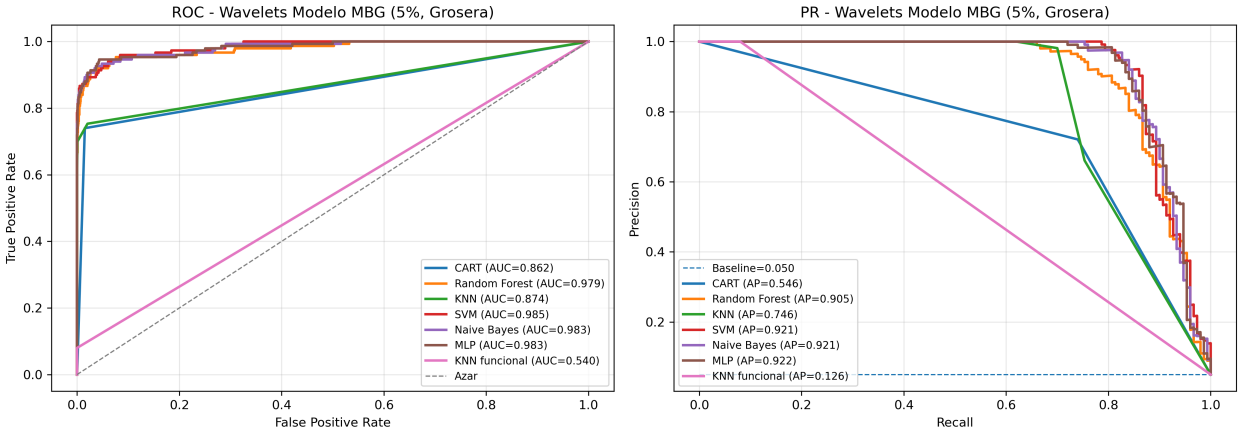


Figura B.7: ROC/PR - Wavelets (Grosera, 5 %)

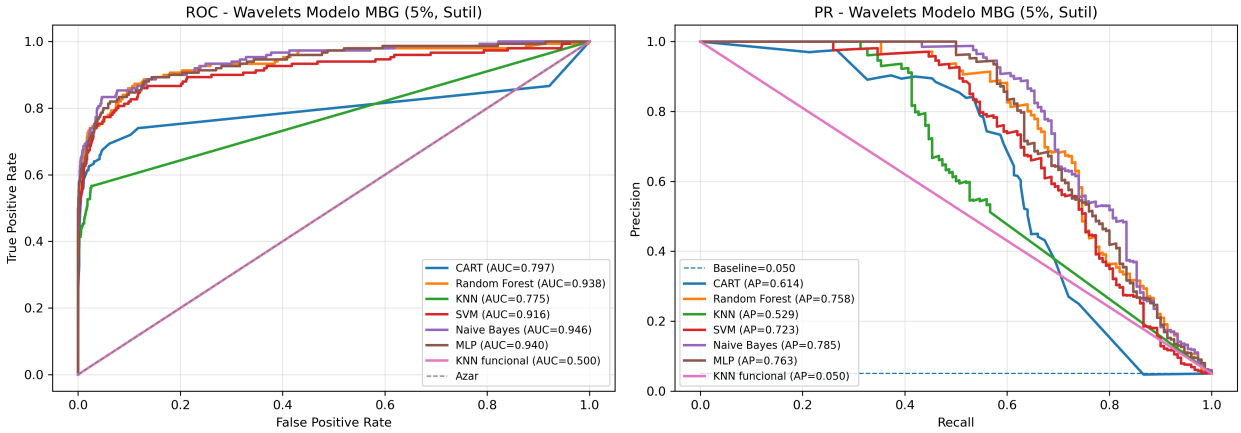


Figura B.8: ROC/PR - Wavelets (Sutil, 5 %)

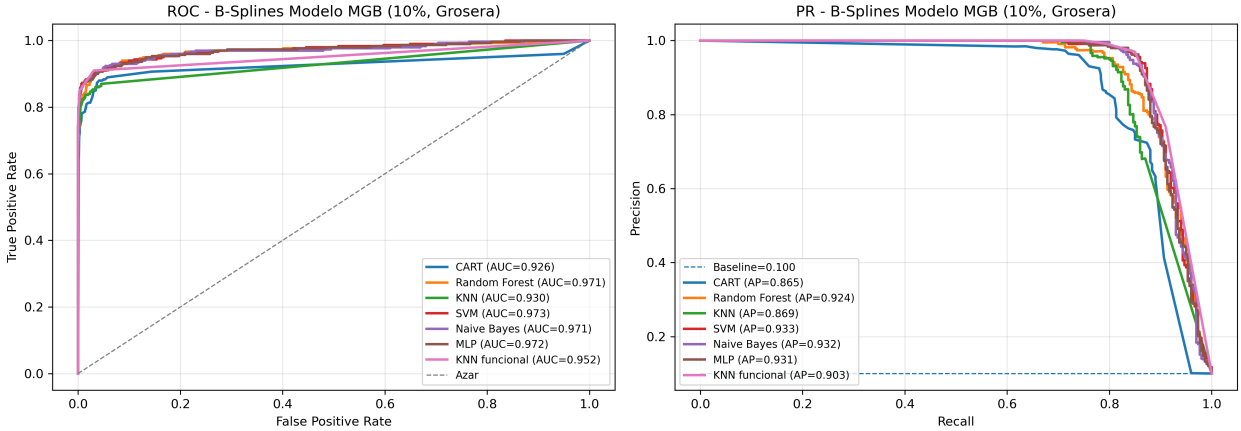


Figura B.9: ROC/PR - B-Splines (Grosera, 10 %)

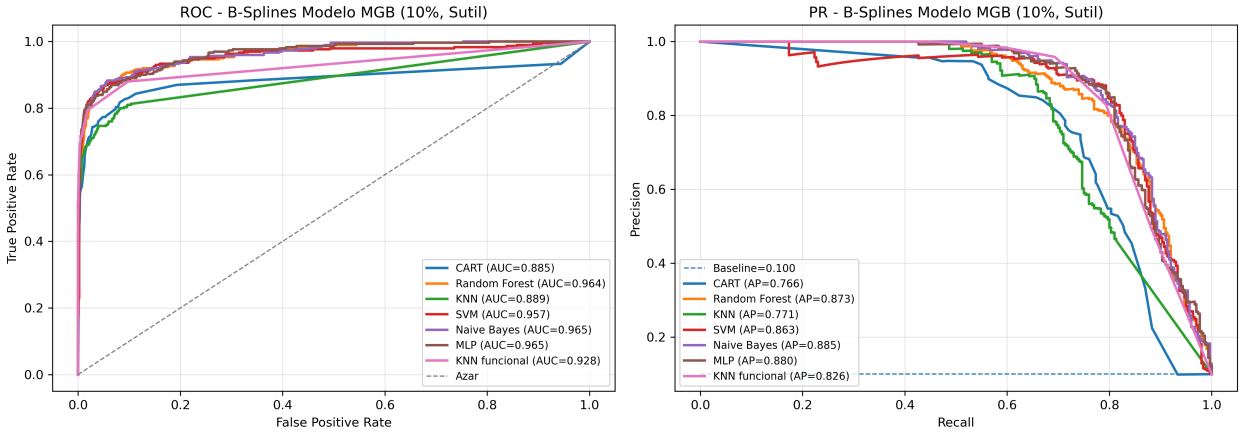


Figura B.10: ROC/PR - B-Splines (Sutil, 10 %)

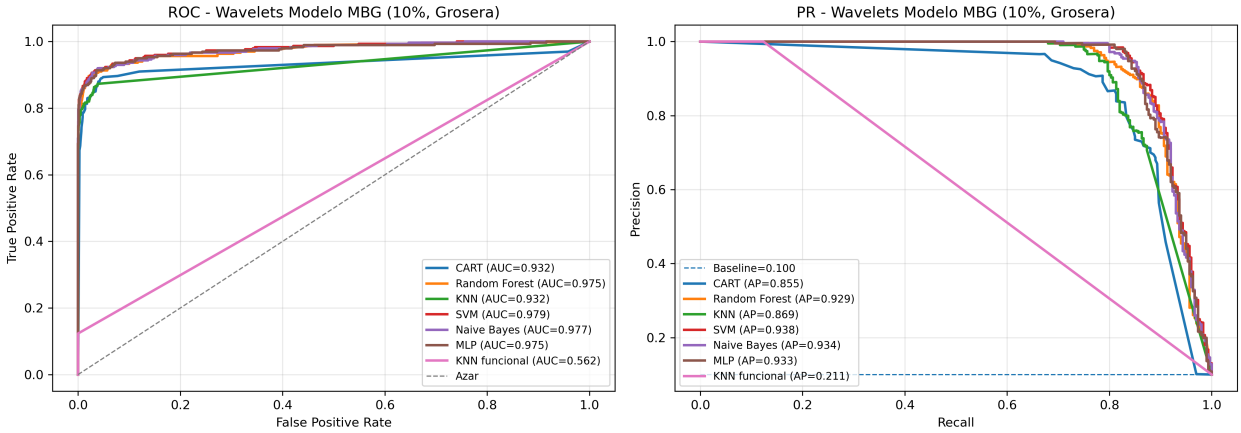


Figura B.11: ROC/PR - Wavelets (Grosera, 10 %)

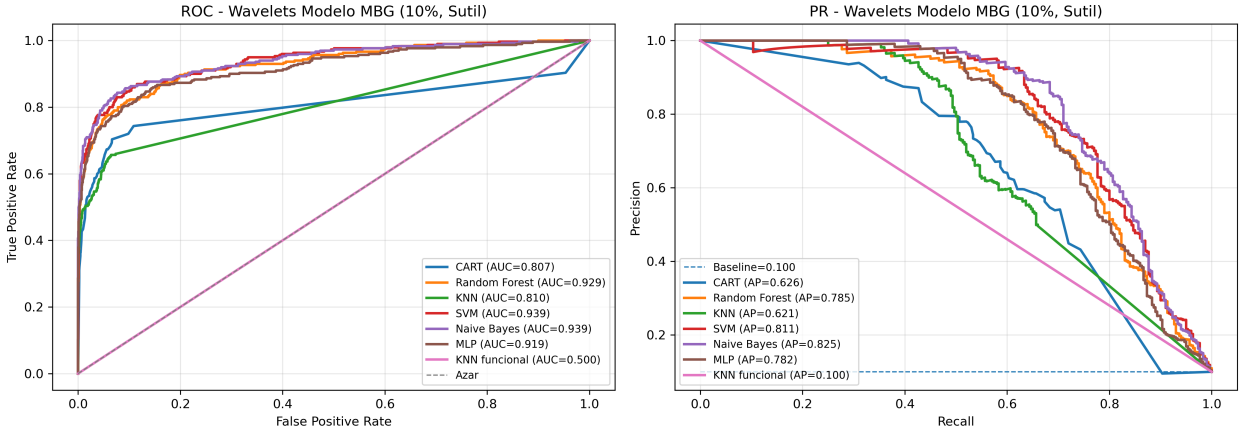


Figura B.12: ROC/PR - Wavelets (Sutil, 10 %)

B.2 Modelo Heston

B.2.1 Tablas Comparativas

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9610	0.1009	0.3667	0.2402	0.6704	0.1230
Random Forest	0.9903	0.6667	0.0667	0.0813	0.9171	0.2333
KNN	0.9903	1.0000	0.0333	0.0413	0.6111	0.1366
SVM	0.9500	0.1203	0.6333	0.3417	0.9005	0.1279
Naive Bayes	0.9850	0.2727	0.3000	0.2941	0.9426	0.3092
MLP	0.9893	0.4167	0.1667	0.1894	0.9131	0.3038
KNN funcional	0.9900	0.0000	0.0000	0.0000	0.7065	0.1538

Cuadro B.13: Resultados Modelo de Heston con B-Splines (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9867	0.0833	0.0333	0.0379	0.5148	0.0124
Random Forest	0.9903	1.0000	0.0333	0.0413	0.7222	0.1300
KNN	0.9900	0.0000	0.0000	0.0000	0.5107	0.0108
SVM	0.9773	0.1607	0.3000	0.2557	0.7899	0.1698
Naive Bayes	0.9917	1.0000	0.1667	0.2000	0.8848	0.3673
MLP	0.9903	1.0000	0.0333	0.0413	0.7903	0.2973
KNN funcional	0.9900	0.0000	0.0000	0.0000	0.5000	0.0100

Cuadro B.14: Resultados Modelo de Heston con Wavelets (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9843	0.3692	0.8000	0.6486	0.8979	0.7051
Random Forest	0.9960	0.8214	0.7667	0.7770	0.9917	0.8072
KNN	0.9950	0.8947	0.5667	0.6115	0.8487	0.6785
SVM	0.9733	0.2449	0.8000	0.5505	0.9450	0.3160
Naive Bayes	0.9967	0.8333	0.8333	0.8333	0.9861	0.8732
MLP	0.9967	0.8571	0.8000	0.8108	0.9790	0.8532
KNN funcional	0.9930	1.0000	0.3000	0.3488	0.7996	0.5957

Cuadro B.15: Resultados Modelo de Heston con B-Splines (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9907	0.5250	0.7000	0.6562	0.8490	0.6999
Random Forest	0.9963	0.9524	0.6667	0.7092	0.9670	0.7534
KNN	0.9927	1.0000	0.2667	0.3125	0.7654	0.4782
SVM	0.9900	0.5000	0.6333	0.6013	0.9047	0.6603
Naive Bayes	0.9963	1.0000	0.6333	0.6835	0.9946	0.8386
MLP	0.9960	1.0000	0.6000	0.6522	0.9612	0.7500
KNN funcional	0.9900	0.0000	0.0000	0.0000	0.5000	0.0100

Cuadro B.16: Resultados Modelo de Heston con Wavelets (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.8857	0.2371	0.5800	0.4498	0.7165	0.4174
Random Forest	0.9620	0.6385	0.5533	0.5685	0.9188	0.6220
KNN	0.9620	0.8462	0.2933	0.3374	0.7820	0.4960
SVM	0.9000	0.3057	0.7867	0.5984	0.9216	0.5122
Naive Bayes	0.9653	0.6769	0.5867	0.6027	0.9264	0.6866
MLP	0.9663	0.7475	0.4933	0.5293	0.9202	0.6649
KNN funcional	0.9510	1.0000	0.0200	0.0249	0.7865	0.4864

Cuadro B.17: Resultados Modelo de Heston con B-Splines (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9273	0.3455	0.5067	0.4634	0.6532	0.3903
Random Forest	0.9620	0.7571	0.3533	0.3955	0.8471	0.5440
KNN	0.9550	1.0000	0.1000	0.1220	0.6943	0.3492
SVM	0.9480	0.4789	0.4533	0.4582	0.8155	0.4367
Naive Bayes	0.9650	0.8358	0.3733	0.4198	0.9075	0.6093
MLP	0.9633	0.9545	0.2800	0.3261	0.9037	0.5906
KNN funcional	0.9500	0.0000	0.0000	0.0000	0.5000	0.0500

Cuadro B.18: Resultados Modelo de Heston con Wavelets (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9700	0.7239	0.6467	0.6608	0.8168	0.4858
Random Forest	0.9807	0.8433	0.7533	0.7698	0.9660	0.8649
KNN	0.9763	0.9438	0.5600	0.6096	0.8898	0.7462
SVM	0.9650	0.6056	0.8600	0.7934	0.9783	0.8973
Naive Bayes	0.9843	0.9402	0.7333	0.7671	0.9749	0.8906
MLP	0.9820	0.9068	0.7133	0.7451	0.9728	0.8737
KNN funcional	0.9680	1.0000	0.3600	0.4128	0.8868	0.7430

Cuadro B.19: Resultados Modelo de Heston con B-Splines (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9680	0.6452	0.8000	0.7634	0.8501	0.7291
Random Forest	0.9830	0.9024	0.7400	0.7676	0.9594	0.8614
KNN	0.9787	1.0000	0.5733	0.6268	0.8739	0.7068
SVM	0.9780	0.7471	0.8467	0.8247	0.9739	0.8820
Naive Bayes	0.9823	0.9533	0.6800	0.7214	0.9730	0.8825
MLP	0.9840	0.9811	0.6933	0.7365	0.9675	0.8812
KNN funcional	0.9533	1.0000	0.0667	0.0820	0.5633	0.1703

Cuadro B.20: Resultados Modelo de Heston con Wavelets (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.8997	0.4988	0.7033	0.6500	0.7866	0.5431
Random Forest	0.9310	0.6440	0.6933	0.6829	0.9295	0.7525
KNN	0.9373	0.8733	0.4367	0.4852	0.8502	0.6527
SVM	0.8943	0.4834	0.8233	0.7218	0.9369	0.7605
Naive Bayes	0.9433	0.7407	0.6667	0.6803	0.9374	0.7823
MLP	0.9357	0.7238	0.5767	0.6011	0.9259	0.7507
KNN funcional	0.9087	1.0000	0.0867	0.1060	0.8429	0.6025

Cuadro B.21: Resultados Modelo de Heston con B-Splines (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.8830	0.4391	0.6133	0.5683	0.7687	0.5578
Random Forest	0.9353	0.7345	0.5533	0.5820	0.9048	0.7034
KNN	0.9247	0.9205	0.2700	0.3144	0.7633	0.5048
SVM	0.9227	0.6030	0.6633	0.6503	0.8928	0.6876
Naive Bayes	0.9433	0.8316	0.5433	0.5838	0.9156	0.7444
MLP	0.9383	0.8808	0.4433	0.4922	0.9052	0.7181
KNN funcional	0.9000	0.0000	0.0000	0.0000	0.5017	0.1030

Cuadro B.22: Resultados Modelo de Heston con Wavelets (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9470	0.6975	0.8300	0.7996	0.8921	0.8143
Random Forest	0.9703	0.8754	0.8200	0.8305	0.9645	0.9037
KNN	0.9697	0.9860	0.7067	0.7491	0.9119	0.8251
SVM	0.9673	0.8258	0.8533	0.8477	0.9746	0.9165
Naive Bayes	0.9733	0.9508	0.7733	0.8033	0.9732	0.9161
MLP	0.9717	0.9614	0.7467	0.7816	0.9718	0.9070
KNN funcional	0.9433	1.0000	0.4333	0.4887	0.9124	0.8238

Cuadro B.23: Resultados Modelo de Heston con B-Splines (Desbalance 0.10, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9383	0.6486	0.8367	0.7908	0.9363	0.8456
Random Forest	0.9700	0.8860	0.8033	0.8186	0.9647	0.9013
KNN	0.9660	0.9853	0.6700	0.7158	0.9077	0.8067
SVM	0.9703	0.8651	0.8333	0.8395	0.9756	0.9147
Naive Bayes	0.9707	0.9530	0.7433	0.7775	0.9738	0.9146
MLP	0.9743	0.9627	0.7733	0.8050	0.9719	0.9116
KNN funcional	0.9103	1.0000	0.1033	0.1259	0.5817	0.2470

Cuadro B.24: Resultados Modelo de Heston con Wavelets (Desbalance 0.10, Grosera)

B.2.2 Curvas ROC/PR

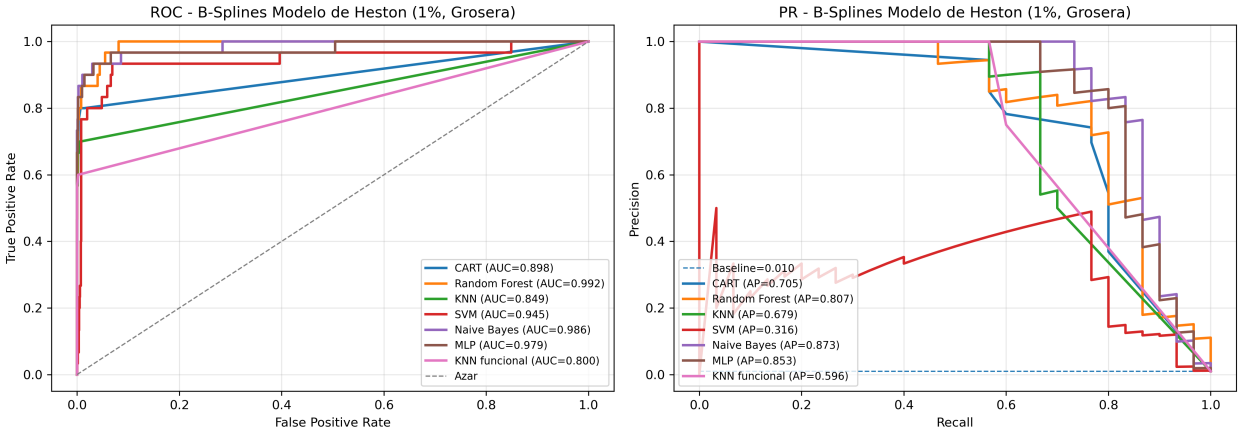


Figura B.13: ROC/PR - B-Splines (Grosera, 1 %)

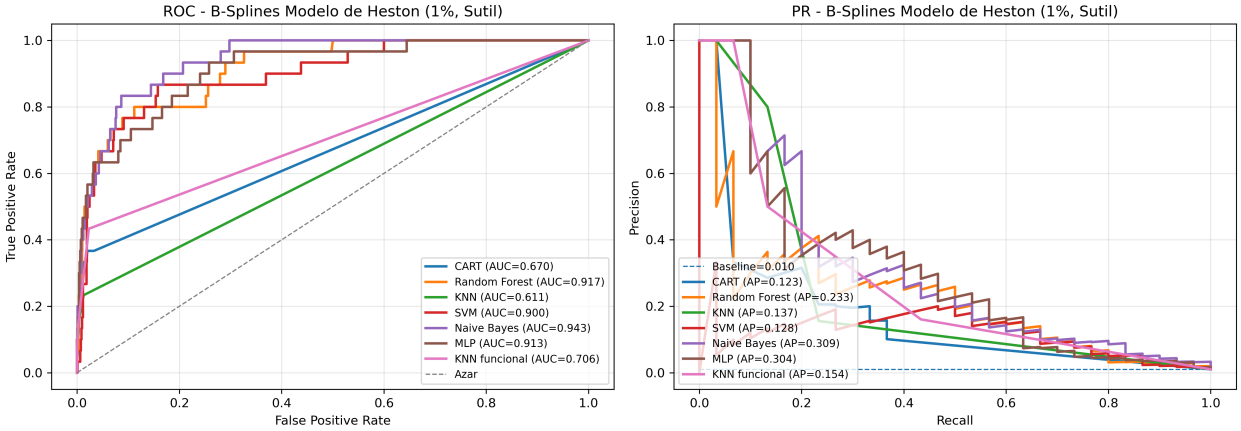


Figura B.14: ROC/PR - B-Splines (Sutil, 1 %)

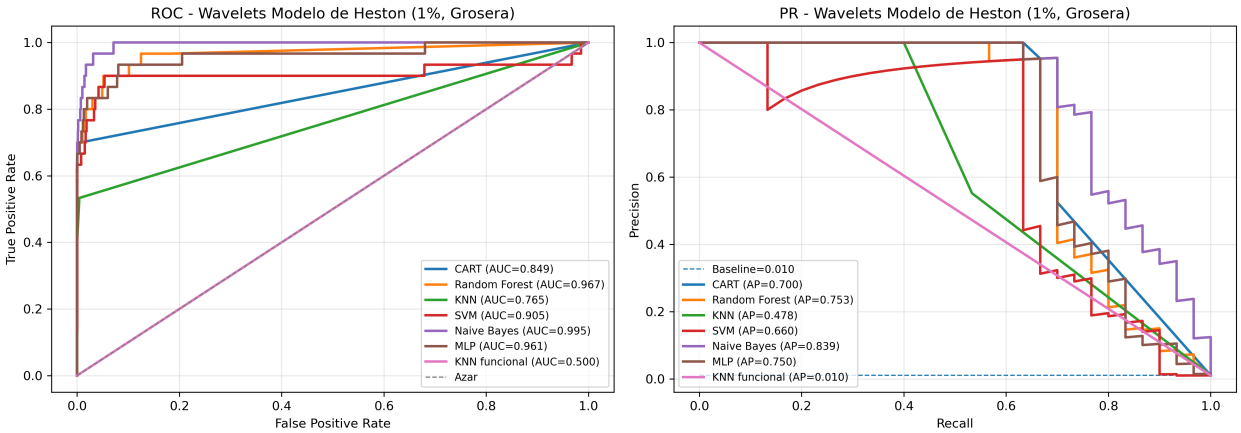


Figura B.15: ROC/PR - Wavelets (Grosera, 1 %)

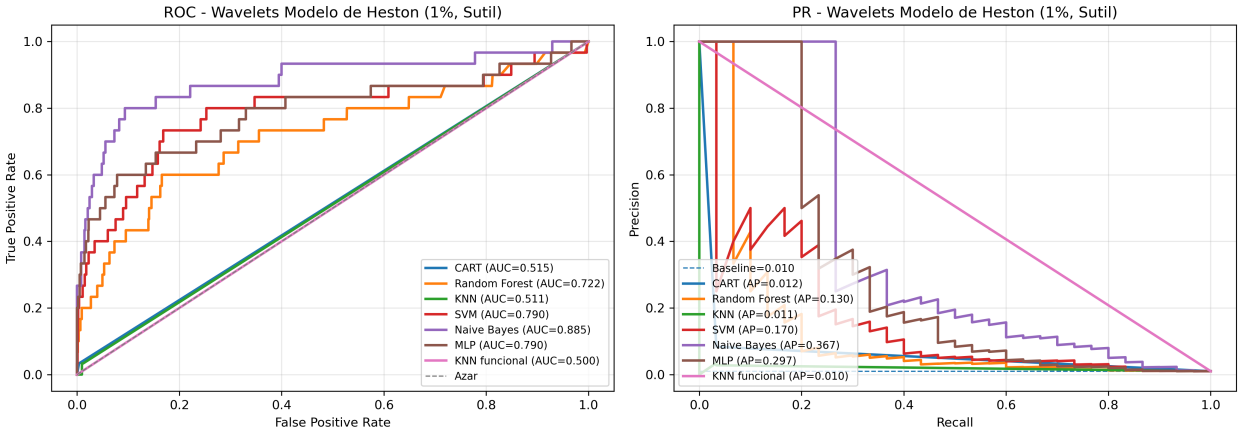


Figura B.16: ROC/PR - Wavelets (Sutil, 1 %)

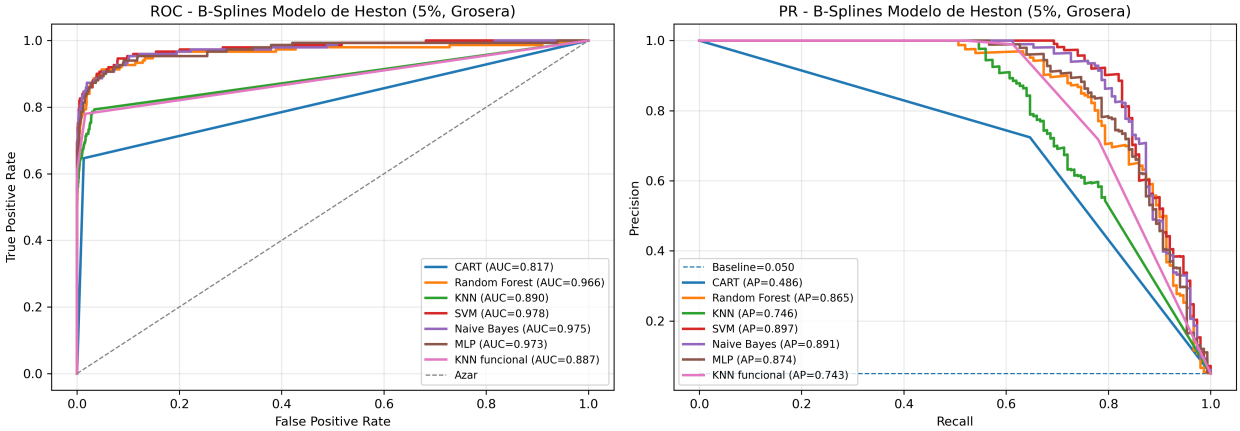


Figura B.17: ROC/PR - B-Splines (Grosera, 5 %)

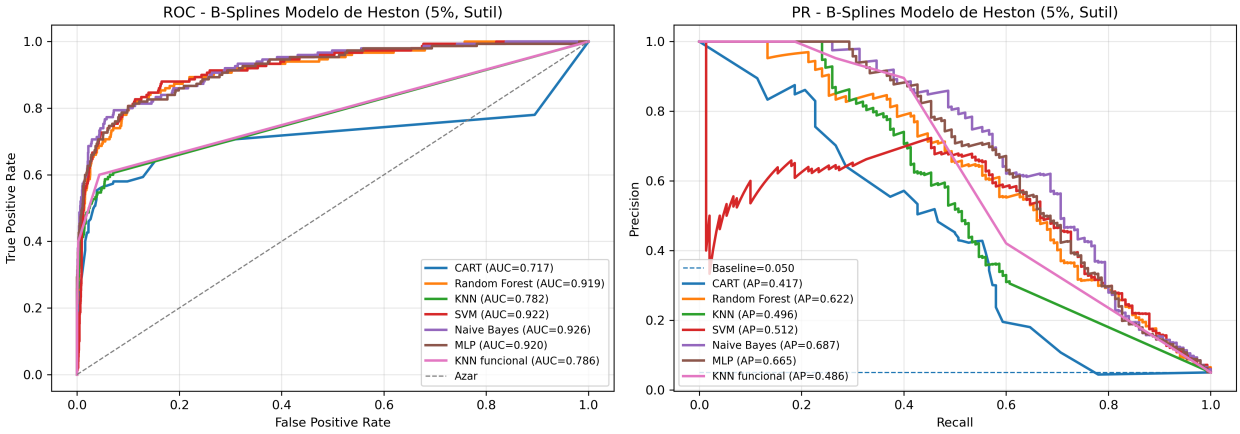


Figura B.18: ROC/PR - B-Splines (Sutil, 5 %)

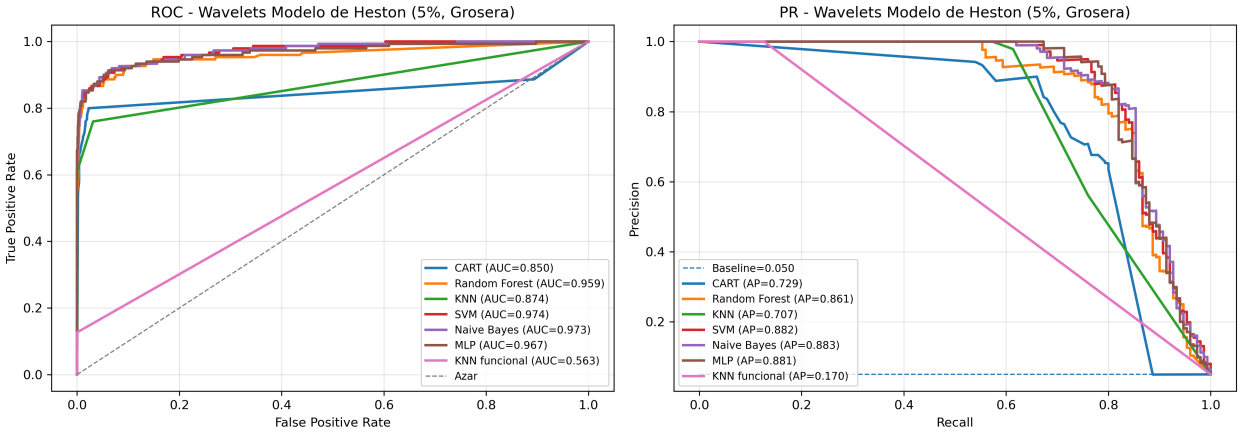


Figura B.19: ROC/PR - Wavelets (Grosera, 5 %)

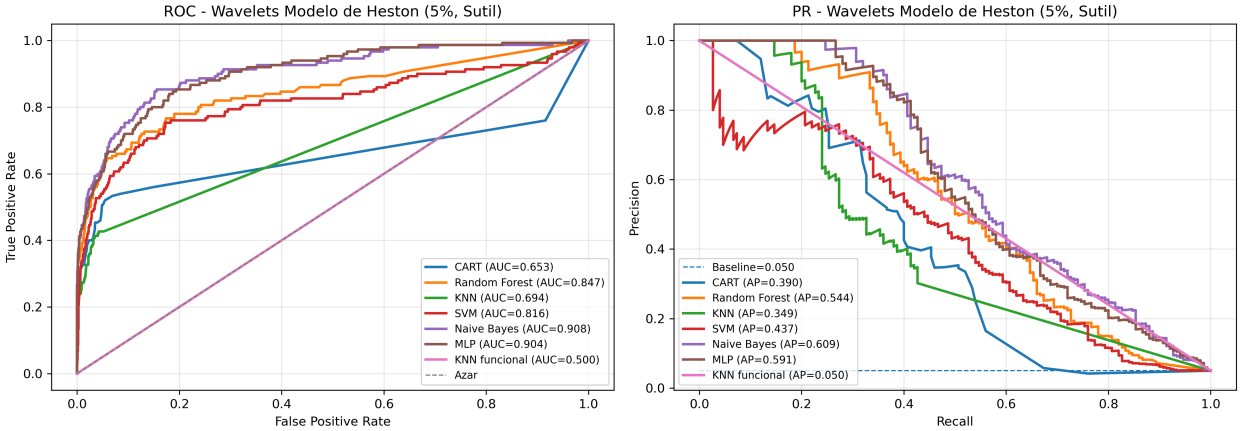


Figura B.20: ROC/PR - Wavelets (Sutil, 5 %)

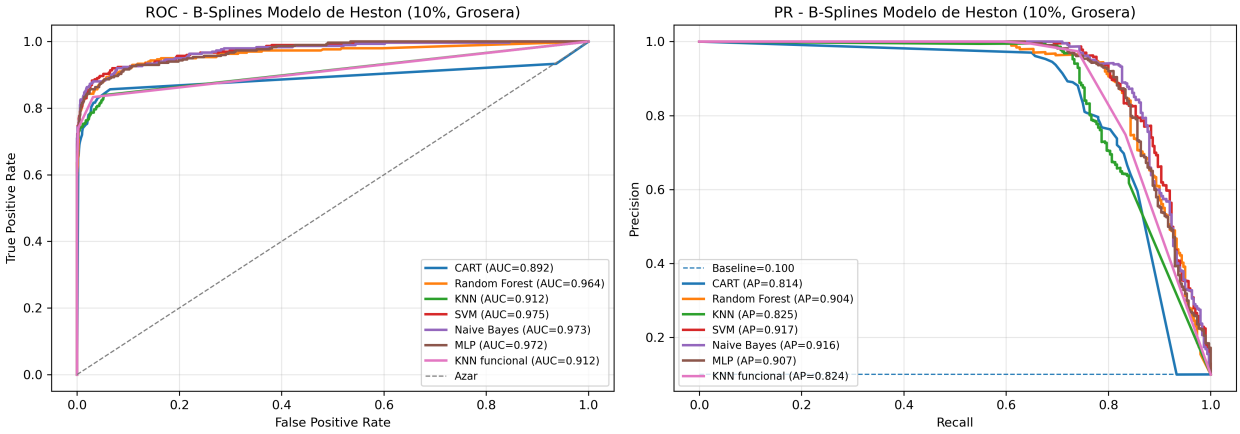


Figura B.21: ROC/PR - B-Splines (Grosera, 10 %)

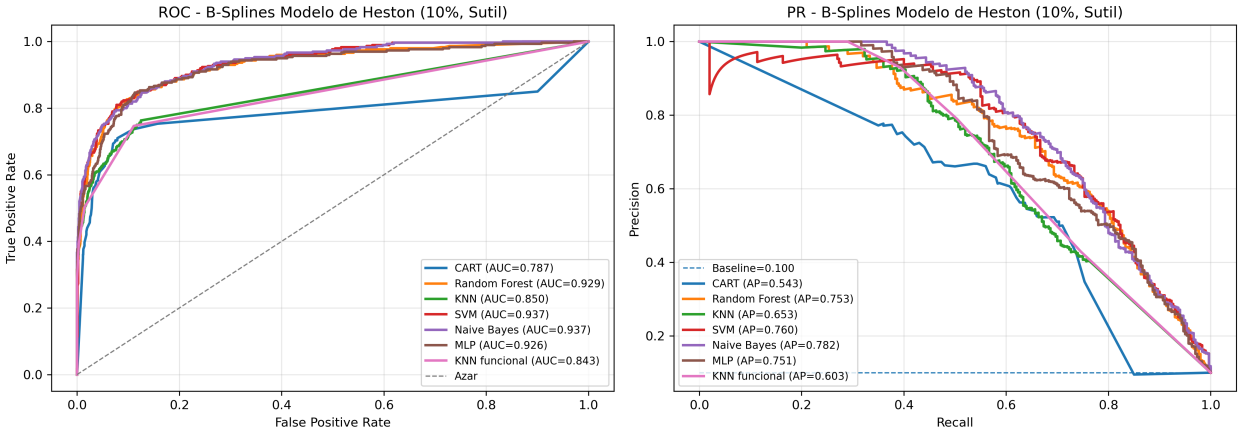


Figura B.22: ROC/PR - B-Splines (Sutil, 10 %)

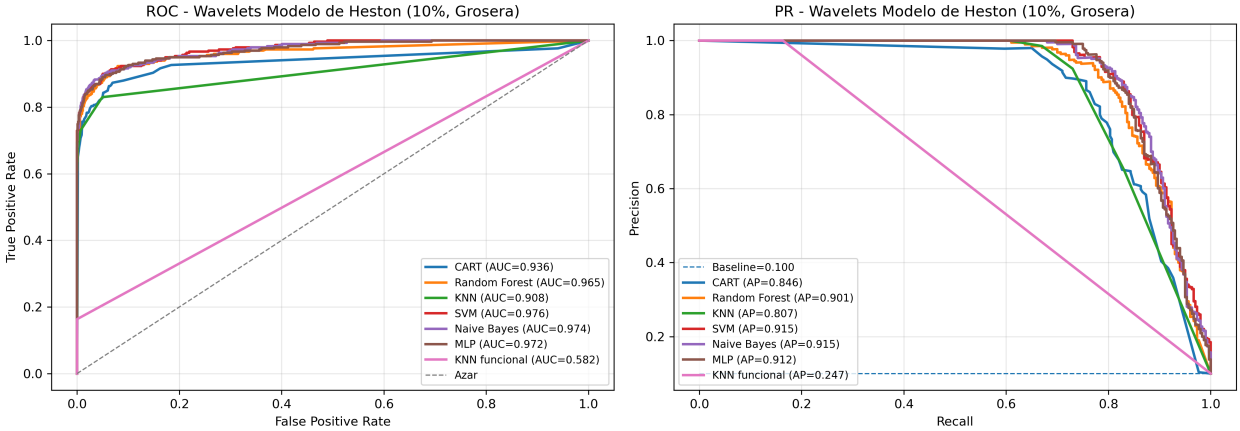


Figura B.23: ROC/PR - Wavelets (Grosera, 10 %)

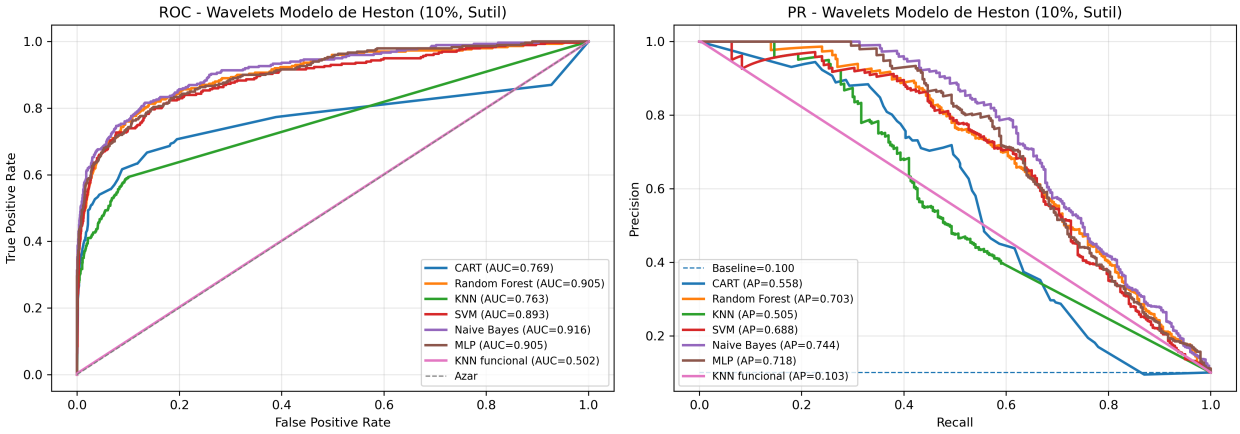


Figura B.24: ROC/PR - Wavelets (Sutil, 10 %)

B.3 Modelo Cont-Müller

B.3.1 Tablas Comparativas

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9800	0.2321	0.4333	0.3693	0.7114	0.2533
Random Forest	0.9900	0.5000	0.4000	0.4167	0.8780	0.3488
KNN	0.9903	0.6000	0.1000	0.1200	0.7741	0.2837
SVM (RBF)	0.9213	0.0945	0.8000	0.3209	0.8808	0.2022
Naive Bayes	0.9447	0.0952	0.5333	0.2778	0.8729	0.1059
MLP	0.9920	0.8000	0.2667	0.3077	0.8928	0.5020
KNN funcional	0.9920	1.0000	0.2000	0.2381	0.8903	0.5881

Cuadro B.25: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9790	0.2295	0.4667	0.3867	0.7259	0.1310
Random Forest	0.9880	0.3929	0.3667	0.3716	0.8266	0.4175
KNN	0.9927	0.9000	0.3000	0.3462	0.8284	0.5537
SVM (RBF)	0.9597	0.1630	0.7333	0.4314	0.9211	0.3117
Naive Bayes	0.9583	0.1074	0.4333	0.2697	0.8940	0.1254
MLP	0.9930	0.7368	0.4667	0.5036	0.8640	0.5874
KNN funcional	0.9933	1.0000	0.3333	0.3846	0.8902	0.5874

Cuadro B.26: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.01, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9990	0.9355	0.9667	0.9603	0.9833	0.9648
Random Forest	0.9993	0.9667	0.9667	0.9667	0.9999	0.9937
KNN	0.9990	1.0000	0.9000	0.9184	0.9833	0.9670
SVM (RBF)	0.9990	1.0000	0.9000	0.9184	1.0000	0.9970
Naive Bayes	0.9993	1.0000	0.9333	0.9459	0.9999	0.9895
MLP	0.9987	1.0000	0.8667	0.8904	0.9983	0.9618
KNN funcional	0.9987	1.0000	0.8667	0.8904	0.9833	0.9670

Cuadro B.27: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9993	1.0000	0.9333	0.9459	0.9667	0.9340
Random Forest	0.9993	1.0000	0.9333	0.9459	0.9833	0.9670
KNN	0.9997	1.0000	0.9667	0.9732	0.9833	0.9670
SVM (RBF)	0.9990	1.0000	0.9000	0.9184	1.0000	0.9979
Naive Bayes	0.9993	1.0000	0.9333	0.9459	0.9998	0.9889
MLP	0.9983	0.9630	0.8667	0.8844	0.9977	0.9327
KNN funcional	0.9987	1.0000	0.8667	0.8904	0.9833	0.9670

Cuadro B.28: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.01, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9283	0.3735	0.6400	0.5601	0.8045	0.4947
Random Forest	0.9567	0.5549	0.6733	0.6458	0.9296	0.6866
KNN	0.9660	0.7353	0.5000	0.5342	0.8584	0.6298
SVM (RBF)	0.9237	0.3785	0.8200	0.6649	0.9426	0.4424
Naive Bayes	0.9180	0.3286	0.6133	0.5227	0.8793	0.4113
MLP	0.9720	0.7797	0.6133	0.6407	0.9551	0.7443
KNN funcional	0.9753	0.8878	0.5800	0.6232	0.9119	0.7391

Cuadro B.29: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9190	0.3505	0.7267	0.5982	0.8478	0.6162
Random Forest	0.9703	0.7402	0.6267	0.6465	0.9335	0.7233
KNN	0.9730	0.8165	0.5933	0.6276	0.8744	0.6855
SVM (RBF)	0.9347	0.4201	0.8067	0.6813	0.9481	0.5822
Naive Bayes	0.9237	0.3414	0.5667	0.5006	0.8688	0.3911
MLP	0.9770	0.8240	0.6867	0.7103	0.9429	0.7783
KNN funcional	0.9757	0.8667	0.6067	0.6454	0.9152	0.7423

Cuadro B.30: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.05, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9993	0.9868	1.0000	0.9973	0.9996	0.9868
Random Forest	0.9997	0.9934	1.0000	0.9987	1.0000	0.9998
KNN	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
SVM (RBF)	0.9943	1.0000	0.8867	0.9072	1.0000	1.0000
Naive Bayes	0.9973	0.9494	1.0000	0.9894	0.9989	0.9615
MLP	0.9990	1.0000	0.9800	0.9839	1.0000	1.0000
KNN funcional	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Cuadro B.31: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9997	0.9934	1.0000	0.9987	0.9998	0.9934
Random Forest	0.9997	0.9934	1.0000	0.9987	1.0000	1.0000
KNN	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
SVM (RBF)	0.9947	1.0000	0.8933	0.9128	1.0000	1.0000
Naive Bayes	0.9993	0.9868	1.0000	0.9973	1.0000	1.0000
MLP	0.9927	0.9507	0.9000	0.9097	0.9984	0.9782
KNN funcional	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Cuadro B.32: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.05, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9223	0.5893	0.7367	0.7016	0.8487	0.5274
Random Forest	0.9447	0.7006	0.7800	0.7627	0.9462	0.8189
KNN	0.9517	0.8475	0.6300	0.6641	0.9066	0.7516
SVM (RBF)	0.9270	0.5967	0.8333	0.7721	0.9443	0.7074
Naive Bayes	0.8970	0.4886	0.6433	0.6050	0.8925	0.5913
MLP	0.9540	0.8320	0.6767	0.7029	0.9480	0.8328
KNN funcional	0.9593	0.8771	0.6900	0.7208	0.9225	0.8029

Cuadro B.33: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9087	0.5314	0.7333	0.6815	0.8482	0.6246
Random Forest	0.9560	0.8158	0.7233	0.7401	0.9559	0.8489
KNN	0.9603	0.8851	0.6933	0.7247	0.9089	0.7709
SVM (RBF)	0.9337	0.6241	0.8467	0.7903	0.9510	0.7780
Naive Bayes	0.9020	0.5086	0.5933	0.5742	0.8837	0.5827
MLP	0.9603	0.8467	0.7367	0.7563	0.9459	0.8525
KNN funcional	0.9583	0.8631	0.6933	0.7217	0.9245	0.8054

Cuadro B.34: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.10, Sutil)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9973	0.9966	0.9767	0.9806	0.9881	0.9757
Random Forest	0.9980	0.9966	0.9833	0.9860	1.0000	0.9997
KNN	0.9980	1.0000	0.9800	0.9839	0.9967	0.9940
SVM (RBF)	0.9837	1.0000	0.8367	0.8649	1.0000	0.9997
Naive Bayes	0.9960	0.9706	0.9900	0.9861	0.9986	0.9763
MLP	0.9977	1.0000	0.9767	0.9812	1.0000	0.9997
KNN funcional	0.9973	1.0000	0.9733	0.9786	1.0000	1.0000

Cuadro B.35: Resultados Modelo Cont-Muller con B-Splines 3-curvas (Desbalance 0.10, Grosera)

Modelo	Accuracy	Precision	Recall	F_2	ROC AUC	PR AUC
CART	0.9980	0.9966	0.9833	0.9860	0.9915	0.9817
Random Forest	0.9987	0.9966	0.9900	0.9913	1.0000	0.9998
KNN	0.9987	1.0000	0.9867	0.9893	0.9983	0.9970
SVM (RBF)	0.9837	1.0000	0.8367	0.8649	1.0000	0.9997
Naive Bayes	0.9960	0.9706	0.9900	0.9861	0.9987	0.9791
MLP	0.9927	0.9728	0.9533	0.9572	0.9970	0.9871
KNN funcional	0.9973	1.0000	0.9733	0.9786	1.0000	1.0000

Cuadro B.36: Resultados Modelo Cont-Muller con Wavelets 3-curvas (Desbalance 0.10, Grosera)

B.3.2 Curvas ROC/PR

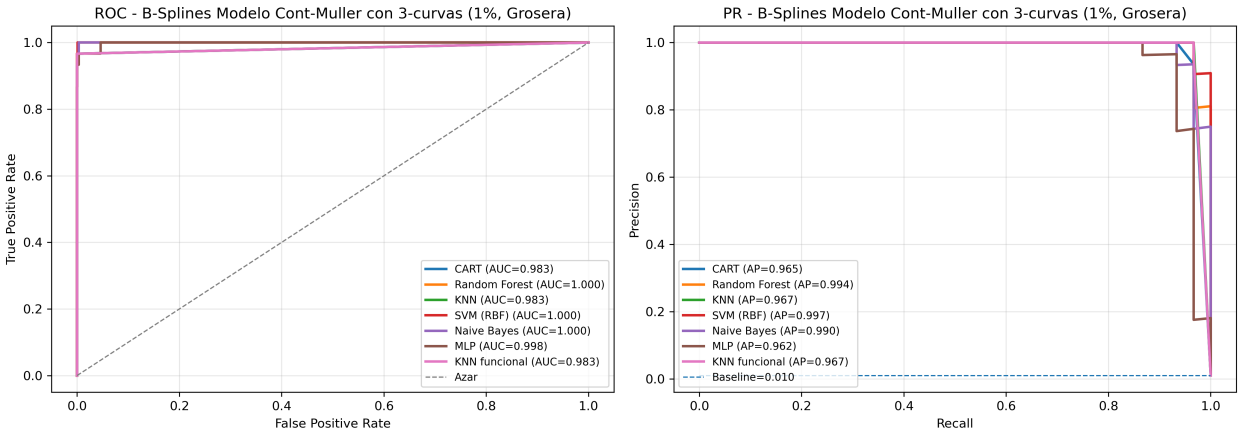


Figura B.25: ROC/PR - B-Splines (Grosera, 1 %)

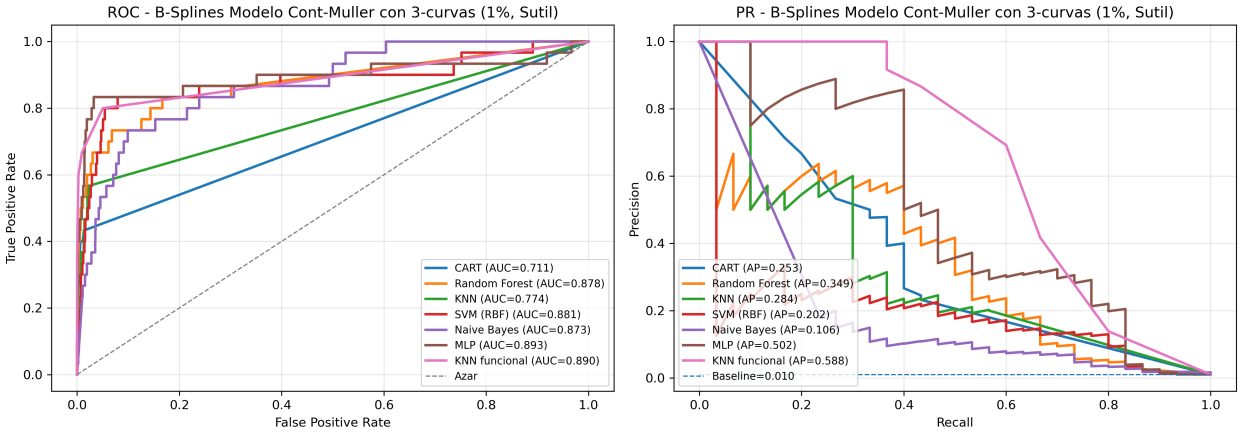


Figura B.26: ROC/PR - B-Splines (Sutil, 1 %)

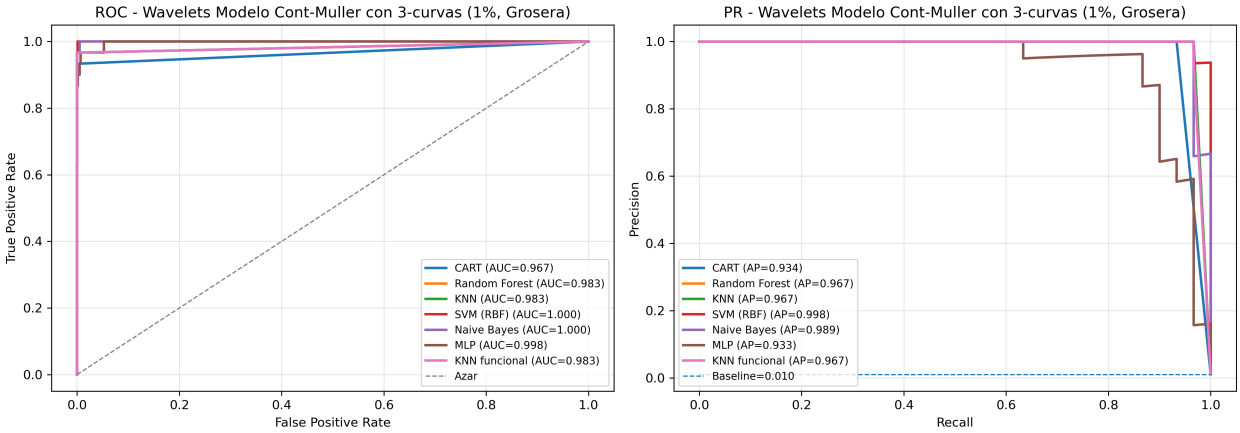


Figura B.27: ROC/PR - Wavelets (Grosera, 1 %)

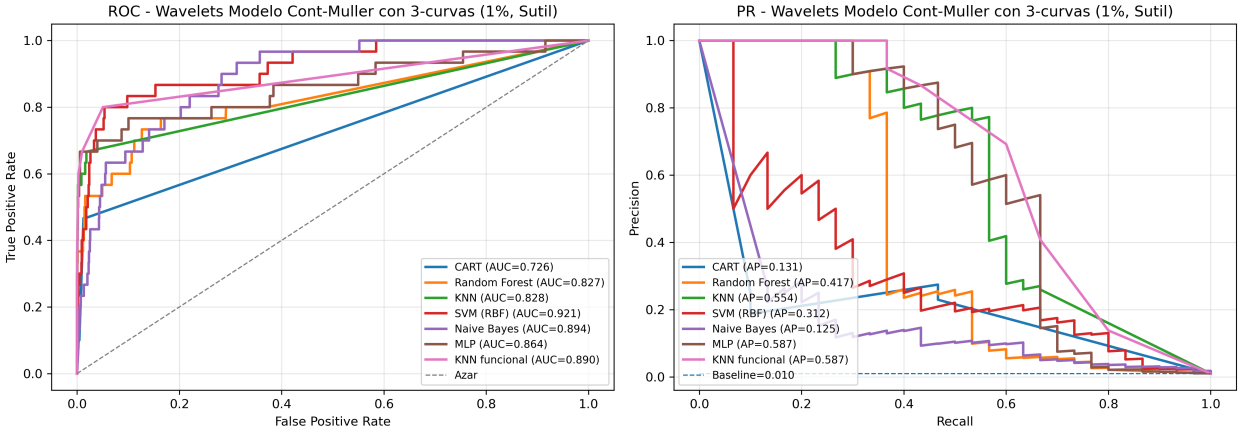


Figura B.28: ROC/PR - Wavelets (Sutil, 1 %)

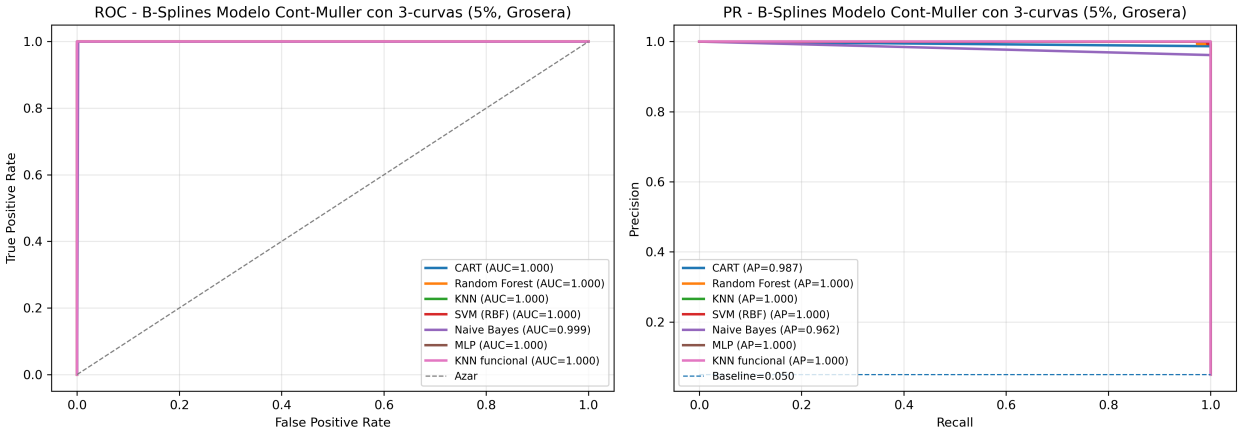


Figura B.29: ROC/PR - B-Splines (Grosera, 5 %)

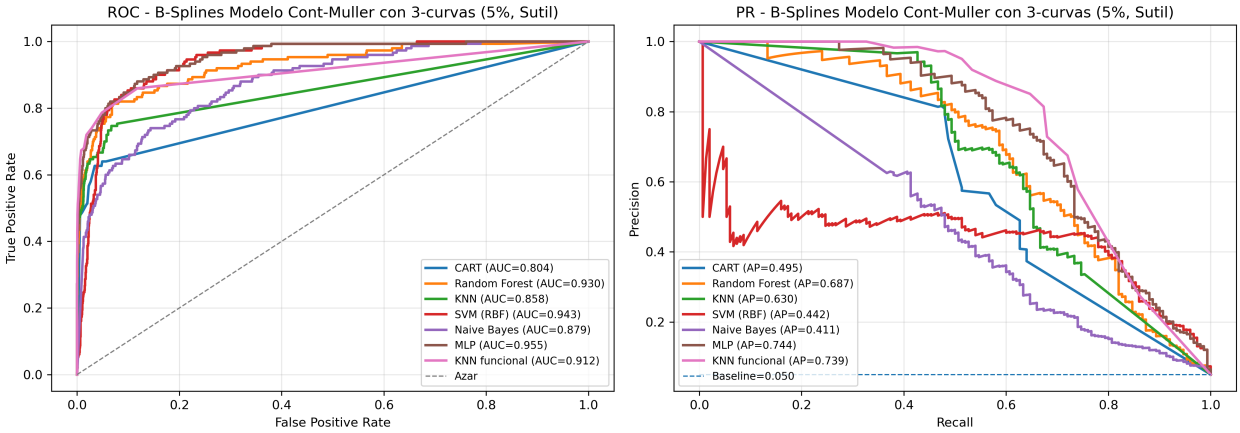


Figura B.30: ROC/PR - B-Splines (Sutil, 5 %)

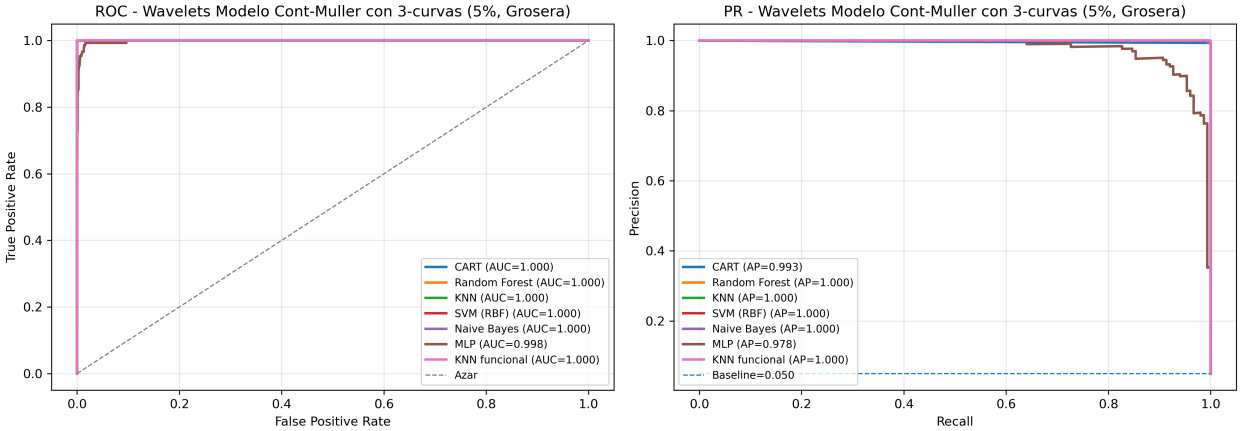


Figura B.31: ROC/PR - Wavelets (Grosera, 5 %)

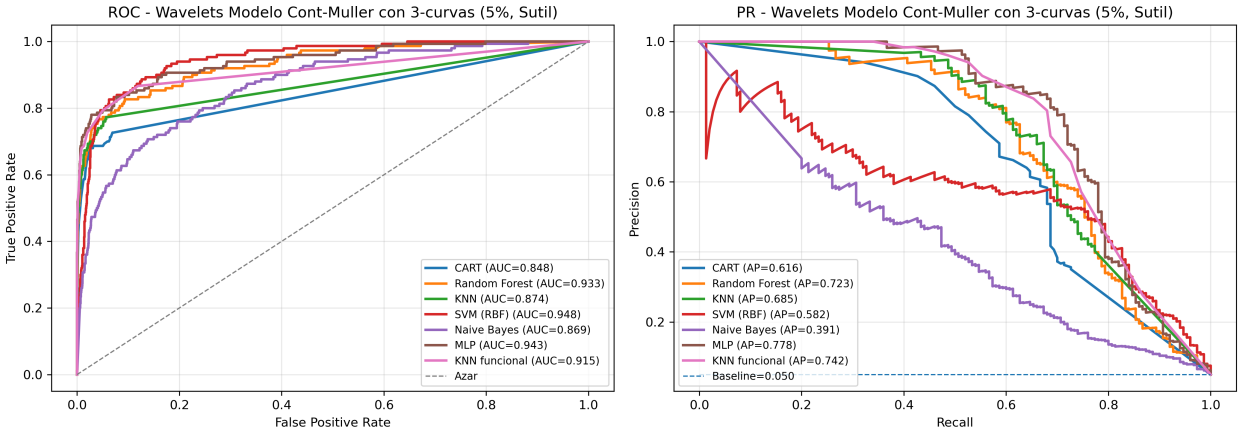


Figura B.32: ROC/PR - Wavelets (Sutil, 5 %)

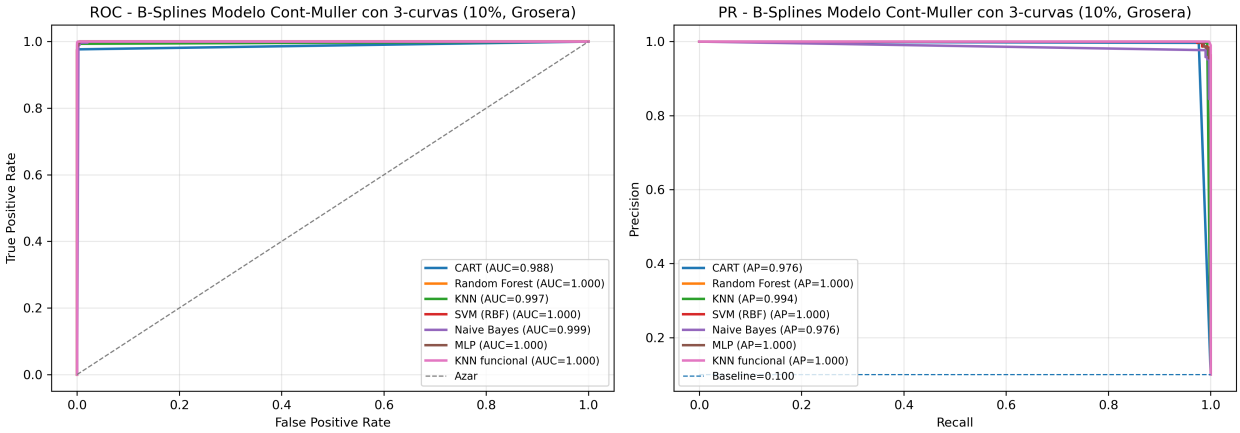


Figura B.33: ROC/PR - B-Splines (Grosera, 10 %)

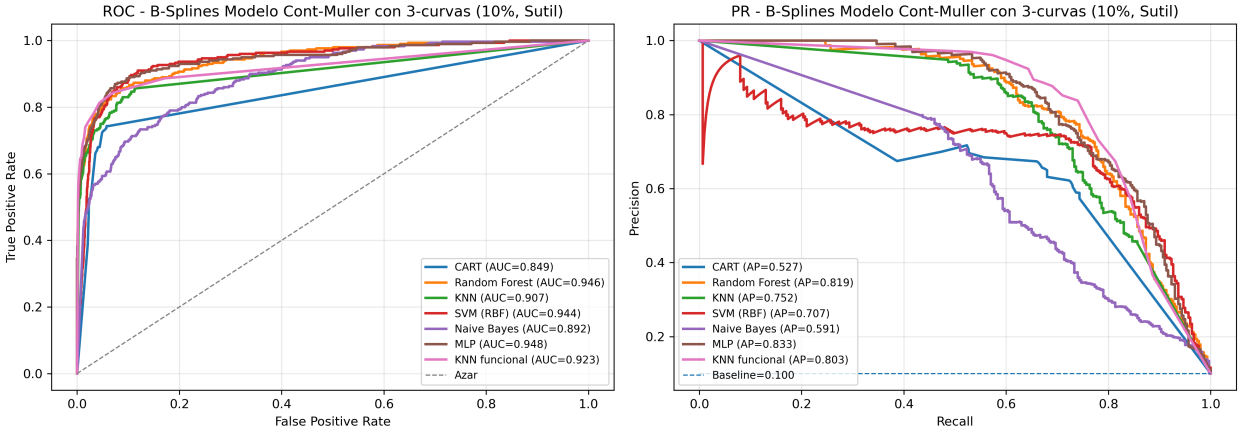


Figura B.34: ROC/PR - B-Splines (Sutil, 10 %)

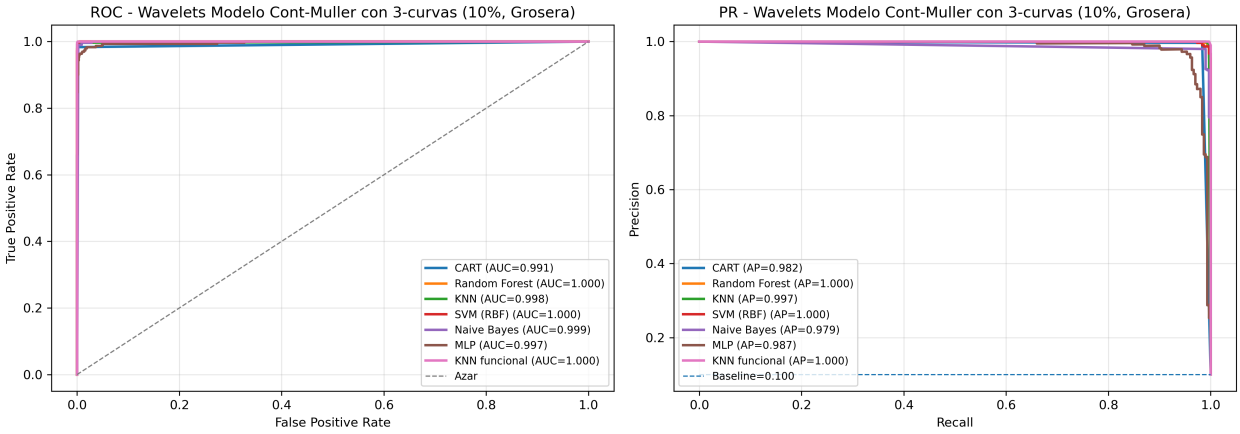


Figura B.35: ROC/PR - Wavelets (Grosera, 10 %)

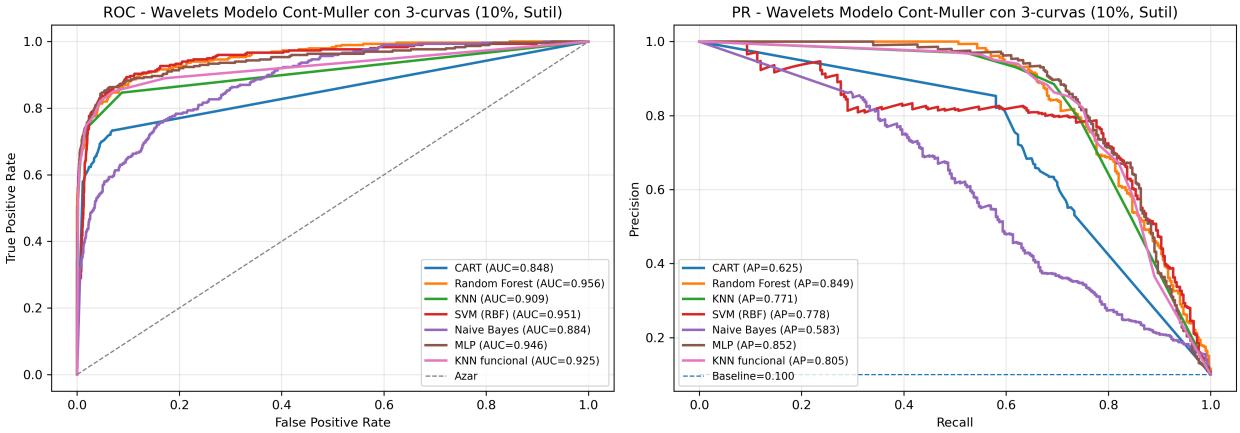


Figura B.36: ROC/PR - Wavelets (Sutil, 10 %)

Capítulo C

Anexo 3

Los códigos desarrollados en este trabajo fueron implmentados mediante Jupyter Notebook. El conjunto completo completo de los notebooks se encuentra disponible en el siguiente repositorio de GitHub:

https://github.com/johnnyhuincahue/Codigo-Trabajo-de-Titulo/tree/main

Bibliografía

[1] Rajesh K. Aggarwal y Guojun Wu. «Stock Market Manipulations». En: *The Journal of Business* 79.4 (2006), págs. 1915-1953. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/10.1086/503652 (visitado 28-09-2025).

[2] A. Alfajeer, A. Altaweel, A. Bouridane et al. «On detecting stock price manipulation attacks: a comprehensive systematic literature review». En: *Multimedia Tools and Applications* 84 (2025), págs. 35793-35870. DOI: 10.1007/s11042-025-20641-4. URL: https://doi.org/10.1007/s11042-025-20641-4.

[3] Franklin Allen y Douglas Gale. «Stock-Price Manipulation». En: *The Review of Financial Studies* 5.3 (mayo de 1992), págs. 503-529. ISSN: 0893-9454. DOI: 10.1093/rfs/5.3.503. eprint: https://academic.oup.com/rfs/article-pdf/5/3/503/24417344/050503.pdf. URL: https://doi.org/10.1093/rfs/5.3.503.

[4] Walter Bagehot. «The Only Game in Town». En: *Financial Analysts Journal* 27.2 (1971), págs. 12-14. DOI: 10.2469/faj.v27.n2.12.

[5] Leo Breiman. «Random Forests». En: *Machine Learning* 45 (2001), págs. 5-32. DOI: 10.1023/A:1010933404324. URL: https://doi.org/10.1023/A:1010933404324.

[6] Leo Breiman et al. *Classification and Regression Trees*. 1st. Chapman y Hall/CRC, 1984. DOI: 10.1201/9781315139470.

[7] Yi Cao et al. «Detecting Wash Trade in Financial Market Using Digraphs and Dynamic Programming». En: *IEEE transactions on neural networks and learning systems* (oct. de 2015). DOI: 10.1109/TNNLS.2015.2480959.

[8] N. V. Chawla et al. «SMOTE: Synthetic Minority Over-sampling Technique». En: *Journal of Artificial Intelligence Research* 16 (jun. de 2002), págs. 321-357. ISSN: 1076-9757. DOI: 10.1613/jair.953. URL: http://dx.doi.org/10.1613/jair.953.

[9] Carole Comerton-Forde y Talis J. Putnins. «Measuring closing price manipulation». En: *Journal of Financial Intermediation* 20.2 (abr. de 2011), págs. 135-158. DOI: None. URL: https://ideas.repec.org/a/eee/jfinin/v20y2011i2p135-158.html.

[10] Rama Cont y Marvin S. Mueller. *A stochastic partial differential equation model for limit order book dynamics*. 2021. arXiv: 1904.03058 [q-fin.TR]. URL: https://arxiv.org/abs/1904.03058.

[11] Thomas Copeland, J. Weston y Kuldeep Shastri. *Financial Theory and Corporate Policy*. Pearson Deutschland, 2013, pág. 924. ISBN: 9781292021584. URL: https://elibrary.pearson.de/book/99.150005/9781292034812.

[12] Corinna Cortes y Vladimir Vapnik. «Support-vector networks». En: *Machine Learning* 20.3 (1995), págs. 273-297. DOI: 10.1007/BF00994018.

[13] Esther B. Del Brio, Alberto Miguel y Javier Perote. «An investigation of insider trading profits in the Spanish stock market». En: *The Quarterly Review of Economics and Finance* 42.1 (2002), págs. 73-94. ISSN: 1062-9769. DOI: https://doi.org/10.1016/S1062-9769(01)00103-X. URL: https://www.sciencedirect.com/science/article/pii/S106297690100103X.

[14] David Diaz, Babis Theodoulidis y Pedro Sampaio. «Analysis of stock market manipulations using knowledge discovery techniques applied to intraday trade prices». En: *Expert Syst. Appl.* 38.10 (sep. de 2011), págs. 12757-12771. ISSN: 0957-4174. DOI: 10.1016/j.eswa.2011.04.066. URL: https://doi.org/10.1016/j.eswa.2011.04.066.

[15] F.J. Fabozzi, F.P. Modigliani y F.J. Jones. *Foundations of Financial Markets and Institutions: Pearson New International Edition*. Pearson Education, 2013. ISBN: 9781292034997. URL: https://books.google.cl/books?id=Gv-pBwAAQBAJ.

[16] EUGENE F. FAMA. «Efficient Capital Markets: II». En: *The Journal of Finance* 46.5 (1991), págs. 1575-1617. DOI: https://doi.org/10.1111/j.1540-6261.1991.tb04636.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1540-6261.1991.tb04636.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.1991.tb04636.x.

[17] Eugene F. Fama. «Efficient Capital Markets: A Review of Theory and Empirical Work». En: *The Journal of Finance* 25.2 (1970), págs. 383-417. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2325486 (visitado 27-09-2025).

[18] Mahmudah Fatluchi y Rofikoh Rokhim. «Closing Price Manipulation in Indonesia Stock Exchange». En: *Proceedings of the International Conference on Business and Management Research (ICBMR 2017)*. Atlantis Press, 2017, págs. 148-157. ISBN: 978-94-6252-431-6. DOI: 10.2991/icbmr-17.2017.14. URL: https://doi.org/10.2991/icbmr-17.2017.14.

[19] Evelyn Fix y Joseph L. Hodges. *Discriminatory Analysis, Nonparametric Discrimination: Consistency Properties*. Inf. téc. Technical Report No. 4. Randolph Field: USAF School of Aviation Medicine, 1951.

[20] Mark B. Garman. «Market microstructure». En: *Journal of Financial Economics* 3.3 (1976), págs. 257-275. ISSN: 0304-405X. DOI: https://doi.org/10.1016/0304-405X(76)90006-4. URL: https://www.sciencedirect.com/science/article/pii/0304405X76900064.

[21] Aurelien Geron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. 2nd. O'Reilly Media, Inc., 2019. ISBN: 1492032646.

[22] Koosha Golmohammadi, David Díaz y Osmar R. Zaiane. «Detecting stock market manipulation using supervised learning algorithms». En: *2014 International Conference on Data Science and Advanced Analytics (DSAA)* (2014). DOI: 10.1109/DSAA.2014.7058109.

[23] Daniel Guterding y Wolfram Boenkost. «The Heston stochastic volatility model with piecewise constant parameters — efficient calibration and pricing of window barrier options». En: *Journal of Computational and Applied Mathematics* 343 (2018), págs. 353-362. ISSN: 0377-0427. DOI: https://doi.org/10.1016/j.cam.2018.04.054. URL: https://www.sciencedirect.com/science/article/pii/S0377042718302498.

[24] Haibo He y Edwardo A. Garcia. «Learning from Imbalanced Data». En: *IEEE Transactions on Knowledge and Data Engineering* 21.9 (2009), págs. 1263-1284. DOI: 10.1109/TKDE.2008.239.

[25] Steven L. Heston. «A Closed Solution for Options with Stochastic Volatility, with Application to Bond and Currency Options». En: *Review of Financial Studies* 6.2 (1993), págs. 327-343. DOI: 10.1093/rfs/6.2.327. URL: https://doi.org/10.1093/rfs/6.2.327.

[26] Jeffrey F. Jaffe. «Special Information and Insider Trading». En: *The Journal of Business* 47.3 (1974), págs. 410-428. ISSN: 00219398, 15375374. URL: http://www.jstor.org/stable/2352458 (visitado 28-09-2025).

[27] Yoonseong Kim y So Young Sohn. «Stock fraud detection using peer group analysis». En: *Expert Systems with Applications* 39.10 (2012), págs. 8986-8992. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.02.025. URL: https://www.sciencedirect.com/science/article/pii/S0957417412002692.

[28] Teema Leangarun, Poj Tangamchit y Suttipong Thajchayapong. «Stock Price Manipulation Detection Based on Mathematical Models». En: *International Journal of Trade, Economics and Finance* 7.3 (2016), págs. 81-88.

[29] Teema Leangarun, Poj Tangamchit y Suttipong Thajchayapong. «Stock Price Manipulation Detection Using Deep Unsupervised Learning: The Case of Thailand». En: *IEEE Access* 9 (2021), págs. 106824-106838.

[30] Haochen Li, Maria Polukarov y Carmine Ventre. «Detecting Financial Market Manipulation with Statistical Physics Tools». En: *Proceedings of the Fourth ACM International Conference on AI in Finance*. ICAIF '23. Brooklyn, NY, USA: Association for Computing Machinery, 2023, pág. 1. ISBN: 9798400702402. DOI: 10.1145/3604237.3626871. URL: https://doi.org/10.1145/3604237.3626871.

[31] F.S. Mishkin y S.G. Eakins. *Financial Markets and Institutions*. Pearson series in finance. Pearson, 2018. ISBN: 9781292215006. URL: https://books.google.cl/books?id=MuURtAEACAAJ.

[32] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. The MIT Press, 2012. ISBN: 0262018020.

[33] Maureen O'Hara. *Market Microstructure Theory*. Cambridge, Mass.: Blackwell Publishers, 1995.

[34] Hulisi Öğüt, M. Mete Doğanay y Ramazan Aktaş. «Detecting stock-price manipulation in an emerging market: The case of Turkey». En: *Expert Systems with Applications* 36.9 (2009), págs. 11944-11949. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2009.03.065. URL: https://www.sciencedirect.com/science/article/pii/S0957417409003200.

[35] Prashant Priyadarshi y Prabhat Kumar. «A comprehensive review on insider trading detection using artificial intelligence». En: *Journal of Computational Social Science* 7.2 (oct. de 2024), págs. 1645-1664. DOI: 10.1007/s42001-024-00284-5. URL: https://ideas.repec.org/a/spr/jcsosc/v7y2024i2d10.1007_s42001-024-00284-5.html.

[36] Tālis Putniņš. «Market Manipulation: A Survey». English. En: *Journal of Economic Surveys* 26.5 (jun. de 2011), págs. 952-967. ISSN: 0950-0804. DOI: 10.1111/joes.2012.26.issue-5.

[37] J.O. Ramsay y B.W. Silverman. *Functional Data Analysis*. 2nd. New York: Springer, 2005.

[38] Takaya Saito y Marc Rehmsmeier. «The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets». En: *PLOS ONE* 10.3 (mar. de 2015), págs. 1-21. DOI: 10.1371/journal.pone.0118432. URL: https://doi.org/10.1371/journal.pone.0118432.

[39] Paul A. Samuelson. «Rational Theory of Warrant Pricing». En: *Industrial Management Review* 6 (1965), págs. 13-32.

[40] Jose Joy Thoppan et al. *Developing an Effective Model for Detecting Trade-based Market Manipulation*. Emerald Publishing Limited, mayo de 2021. ISBN: 978-1-80117-397-1. DOI: 10.1108/9781801173964. eprint: https://www.emerald.com/book-pdf/9007048/9781801173971.pdf. URL: https://doi.org/10.1108/9781801173964.

[41] Ruey S. Tsay. *Analysis of Financial Time Series*. 3.ª ed. Hoboken: John Wiley & Sons, 2010. DOI: 10.1002/9780470644560.

[42] N.C. Uslu y F. Akal. «A Machine Learning Approach to Detection of Trade-Based Manipulations in Borsa Istanbul». En: *Computational Economics* 60 (2022), págs. 25-45. DOI: 10.1007/s10614-021-10131-8. URL: https://doi.org/10.1007/s10614-021-10131-8.

[43] Qili Wang et al. «Enhancing intraday stock price manipulation detection by leveraging recurrent neural networks with ensemble learning». En: *Neurocomputing* 347 (2019), págs. 46-58. ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2019.03.006. URL: https://www.sciencedirect.com/science/article/pii/S0925231219303005.

[44] Shuoyang Wang, Yuan Huang y Guanqun Cao. «Review on functional data classification». En: *WIREs Computational Statistics* 16.1 (2024), e1638. DOI: https://doi.org/10.1002/wics.1638. eprint: https://wires.onlinelibrary.wiley.com/doi/pdf/10.1002/wics.1638. URL: https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/wics.1638.

[45] Jiangyun Zhang et al. «Stock Price Manipulation Detection Based on Machine Learning Technology: Evidence in China». En: *Geo-Spatial Knowledge and Intelligence*. Ed. por Hanning Yuan, Jing Geng y Fuling Bian. Singapore: Springer Singapore, 2017, págs. 150-158.