Universidad del País Vasco

Euskal Herriko Unibertsitatea

Máster Universitario en Modelización e Investigación Matemática, Estadística y Computación 2023/2024

Trabajo Fin de Máster

Deep Reinforcement Learning en Trading Algorítmico: Una aplicación basada en "Advances in Financial Machine Learning"

Alexander de la Puente González

Director/es

Josu Doncel Vicente

Lugar y fecha de presentación prevista

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

El trading financiero es un campo en constante evolución, impulsado por los avances tecnológicos y el creciente uso de técnicas de *machine learning* y *deep learning*. Entre estas, el *deep reinforcement learning* se ha destacado como una metodología prometedora para desarrollar estrategias de trading que pueden adaptarse dinámicamente a las condiciones cambiantes del mercado. Este proyecto surge de una motivación personal y académica de explorar y aprender a desarrollar estrategias de trading utilizando técnicas avanzadas de *machine learning*.

Una de las principales motivaciones de este trabajo es aprender a implementar un agente de *deep reinforcement learning* para la creación de estrategias de trading. Con el auge de las librerías gratuitas como *Gymnasium* y *Stable Baselines*, se abre una oportunidad única para aprender y aplicar estas herramientas en el diseño, entrenamiento y evaluación de agentes de aprendizaje por refuerzo. Además, este proyecto busca profundizar en las metodologías propuestas en el libro *Advances in Financial Machine Learning* de Marcos López de Prado, un referente en la aplicación de técnicas de *machine learning* al análisis financiero. La implementación y evaluación de estas metodologías permitirán no solo entender mejor cómo funcionan, sino también cómo pueden ayudar en la creación de estrategias de trading más robustas y efectivas.

Otro aspecto clave de esta investigación es evaluar la accesibilidad y aplicabilidad de estas técnicas para personas que no provienen del ámbito financiero. A medida que el trading algorítmico y las técnicas de *machine learning* se vuelven más comunes, es crucial entender cómo alguien con un trasfondo técnico, pero sin experiencia directa en finanzas, puede acercarse a esta disciplina. El objetivo es identificar las barreras de entrada, los desafíos y las oportunidades que presenta este campo, así como evaluar qué tan cerca o lejos está esta disciplina de ser accesible para un público más amplio.

En resumen, este trabajo tiene como objetivo aprender y aplicar técnicas avanzadas de *machine learning*, específicamente *deep reinforcement learning*, en el desarrollo de estrategias de trading, utilizando herramientas accesibles y metodologías de vanguardia. A través de este proceso, se pretende obtener una comprensión más profunda del potencial y las limitaciones de estas técnicas en un contexto financiero, contribuyendo así al conocimiento en esta intersección entre tecnología y finanzas.

1.1. Contextualización del problema

El desarrollo de estrategias de trading efectivas en un entorno financiero altamente dinámico requiere la implementación de técnicas avanzadas que puedan adaptarse a las

complejidades y variabilidades del mercado. En este trabajo, se plantea una estrategia integral que combina varias metodologías innovadoras con el objetivo de evaluar su utilidad en la generación de estrategias de trading robustas y efectivas.

El primer componente clave de esta estrategia es la implementación de barras de desequilibrio de volumen y dólares, una técnica que ofrece una representación más precisa de la actividad del mercado en comparación con las barras de tiempo tradicionales. Estas barras se construyen basándose en la actividad real del mercado, en lugar de intervalos de tiempo fijos, lo que permite capturar mejor la dinámica subyacente de la oferta y la demanda. Este enfoque se evaluará en conjunto con las barras de tiempo convencionales para determinar si proporcionan una ventaja en la formación de estrategias de trading.

El segundo componente es el entrenamiento de un algoritmo de *deep reinforcement learning* (DRL) sobre estas barras construidas y sobre las barras originales. El objetivo aquí es explorar cómo las diferentes representaciones del mercado afectan el rendimiento del algoritmo de DRL en la identificación de oportunidades de trading. Este enfoque permitirá no solo evaluar la efectividad del aprendizaje por refuerzo en estos distintos contextos, sino también determinar si la precisión y la estabilidad del agente de trading pueden mejorar al cambiar la base de datos sobre la que se entrena.

Finalmente, se aplicará la técnica de *Meta-Labeling* sobre las señales generadas por el algoritmo de *deep reinforcement learning*. El *Meta-Labeling* actúa como una capa adicional de análisis, refinando las señales emitidas y potenciando la capacidad del modelo para filtrar las señales más prometedoras. La efectividad de esta técnica se evaluará comparando su rendimiento con el de las señales originales, así como con otros enfoques sin *Meta-Labeling*.

El planteamiento de esta estrategia no solo busca implementar y analizar cada una de estas técnicas de manera individual, sino también explorar cómo funcionan en conjunto. Se realizará una comparativa exhaustiva entre las diferentes configuraciones para determinar cuáles ofrecen los mejores resultados en términos de precisión, estabilidad y rentabilidad en las estrategias de trading. Este enfoque permitirá obtener una visión más completa de cómo estas técnicas pueden contribuir al desarrollo de modelos de trading más eficaces y adaptativos.

1.2. Estructura del Documento

Este documento se organiza en siete capítulos, cada uno de los cuales aborda un aspecto clave del desarrollo de estrategias de trading utilizando técnicas avanzadas de *machine learning*:

- **Capítulo 1: Introducción.** Se presenta la motivación, los objetivos del trabajo y una contextualización inicial del planteamiento de la estrategia.
- **Capítulo 2: Fundamentos de los Datos Financieros.** En este capítulo se introducen los conceptos financieros relevantes que son fundamentales para comprender los datos utilizados en las estrategias de trading. Se abordan temas como la naturaleza de los datos financieros, tipos de barras, y la importancia de la estructura de los datos en el análisis y la toma de decisiones.
- **Capítulo 3: Barras de Desequilibrio.** Este capítulo explora la implementación de barras de desequilibrio de volumen y dólares como una alternativa a las

barras de tiempo tradicionales. Se discuten sus ventajas, su construcción y su impacto en la precisión del análisis de mercado.

- **Capítulo 4: Aprendizaje por Refuerzo.** Aquí se detalla el uso del *deep reinforcement learning* para entrenar agentes de trading sobre las barras de desequilibrio y las barras de tiempo originales. Se analizan las técnicas y algoritmos utilizados, así como los resultados esperados de estos enfoques.

- **Capítulo 5: Meta-Labeling.** En este capítulo se profundiza en la técnica de *Meta-Labeling*, su implementación sobre las señales generadas por los modelos de aprendizaje por refuerzo, y cómo esta técnica puede mejorar la precisión y la gestión del riesgo en las estrategias de trading.

- **Capítulo 6: Resultados.** Se presentan los resultados obtenidos de las implementaciones y evaluaciones realizadas en los capítulos anteriores. Se comparan las diferentes técnicas utilizadas y se discuten sus ventajas y limitaciones.

- **Capítulo 7: Conclusiones.** En el capítulo final se resumen los hallazgos clave del trabajo, se destacan las principales contribuciones, y se proponen posibles líneas de investigación futura para continuar explorando y mejorando las estrategias de trading basadas en *machine learning*.

Esta estructura proporciona una visión clara y coherente del flujo del trabajo, desde la introducción de conceptos fundamentales hasta la implementación y evaluación de técnicas avanzadas, culminando en las conclusiones sobre su efectividad y aplicabilidad.

Capítulo 2

Fundamentos de los datos Financieros

2.1. Descripción de los Datos Financieros

Aunque los datos financieros vienen en muchas formas, este trabajo se centrará en aquellos provenientes de acciones o índices (cestas de instrumentos financieros). Para simplificar, en lugar de pensar en instrumentos financieros individuales, consideraremos la serie temporal subyacente de precios.

2.1.1. Introducción a la Serie Temporal

La serie temporal $\{p_t\}$ representa el precio de una acción o índice en el instante de tiempo discreto t. Es importante notar que t dependerá de la frecuencia de muestreo utilizada para recopilar los datos.

Frecuencia

- **Datos de Baja Frecuencia (LF)**: Diarios, mensuales, trimestrales.
- **Datos de Alta Frecuencia (HF)**: Intradía (30 min., 5 min., etc.).

Al introducir estos conceptos, es pertinente señalar que el modelado en finanzas se realiza con el logaritmo natural del precio (log-precios) en lugar de los precios regulares. Esto se representará como $y_t := \log(p_t)$. Para ilustrar esto, se introduce el simple, pero ampliamente utilizado, modelo de un paseo aleatorio con deriva:

$$y_t = y_{t-1} + \mu + \epsilon_t$$

donde $\epsilon_t \sim i.i.d.\mathcal{N}(0, \sigma^2)$.

2.1.2. Retornos de Activos

El siguiente concepto a introducir son los retornos, una técnica para normalizar los precios y permitir la comparación entre diferentes series temporales, independientemente del valor del precio. Los dos tipos que se van a utilizar son los retornos lineales y logarítmicos.

- **Lineales**: $R_t := \frac{p_t - p_{t-1}}{p_{t-1}} = \frac{p_t}{p_{t-1}} - 1$
- **Logarítmicos**: $r_t := \log\left(\frac{p_t}{p_{t-1}}\right) = \log(p_t) - \log(p_{t-1}) = y_t - y_{t-1}$

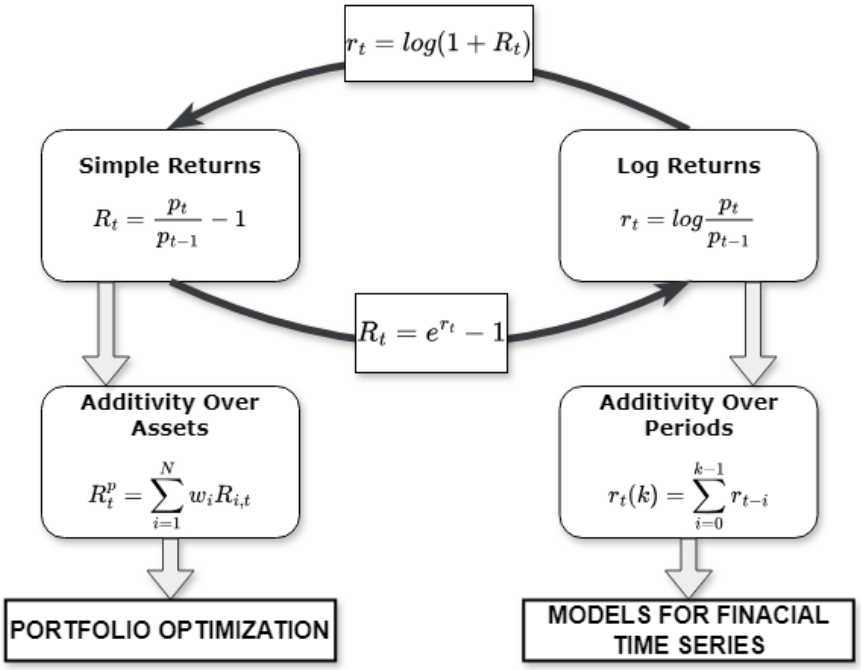

Figura 2.1: Relación entre retornos simples y logarítmicos

La figura 2.1 muestra la relación existente entre los retornos logarítmicos y los retornos simples. Ambos serán utilizados a lo largo de este trabajo tanto para el entrenamiento del agente de aprendizaje por refuerzo en como para el cálculo del los rendimientos generados por las estratégias construídas.

2.2. Hechos Estilizados

Los hechos estilizados son patrones o regularidades observadas empíricamente en los datos financieros que se presentan de manera consistente en diferentes mercados, periodos de tiempo y condiciones económicas. Estos hechos no dependen de modelos teóricos específicos, sino que se derivan directamente de la observación de los datos. La identificación de estos patrones es crucial para el desarrollo y validación de modelos financieros y econométricos. En este trabajo, nos basamos en los hechos estilizados descritos por Rama Cont (2001) en su artículo *Empirical properties of asset returns: stylized facts and statistical issues* [2].

A continuación, se describen los principales hechos estilizados relevantes para este estudio:

1. **Ausencia de Autocorrelaciones**: Los retornos de los activos financieros, especialmente en frecuencias diarias y superiores, tienden a mostrar una autocorrelación lineal insignificante. Esto significa que los retornos pasados

no son buenos predictores de los retornos futuros, lo cual es consistente con la hipótesis del mercado eficiente en su forma débil. Sin embargo, en escalas temporales intradía, pueden observarse autocorrelaciones significativas.

2. **Colas Pesadas**: La distribución de los retornos de los activos financieros presenta colas más gruesas que una distribución normal. Esto implica que los eventos extremos (grandes movimientos de precios) ocurren con mayor frecuencia de lo que se esperaría bajo una distribución normal. Las colas pesadas se modelan mejor con distribuciones como la distribución de Pareto o la distribución t de Student.

3. **Asimetría de Ganancias/Pérdidas**: Los retornos de los activos financieros muestran una asimetría en sus movimientos extremos. Las caídas bruscas en los precios (pérdidas) son más comunes y pronunciadas que los incrementos bruscos (ganancias). Esto se debe a factores como el pánico de los inversores y las ventas masivas en respuesta a malas noticias.

4. **Gaussianidad Agregada**: A medida que se incrementa la escala temporal sobre la cual se calculan los retornos (por ejemplo, pasando de retornos diarios a retornos mensuales), la distribución de los retornos tiende a aproximarse a una distribución normal. Esto es consistente con el teorema central del límite, que establece que la suma de variables aleatorias independientes y con varianza finita tiende hacia una distribución normal.

5. **Intermitencia**: Los retornos financieros muestran una alta variabilidad en todas las escalas temporales. Esto significa que los periodos de alta y baja volatilidad no están distribuidos uniformemente en el tiempo, sino que se alternan de manera impredecible.

6. **Agrupamiento de Volatilidad**: La volatilidad de los retornos tiende a aparecer en clústeres. Periodos de alta volatilidad tienden a ser seguidos por más periodos de alta volatilidad, y lo mismo ocurre con los periodos de baja volatilidad. Este fenómeno se puede modelar mediante modelos de heterocedasticidad condicional, como los modelos GARCH (Generalized Autoregressive Conditional Heteroskedasticity).

Estos hechos estilizados proporcionan una base empírica sobre la cual se pueden construir y validar modelos financieros. La identificación y comprensión de estos patrones ayudan a mejorar la precisión de los modelos de riesgo, a desarrollar estrategias de trading más robustas y a diseñar mejores políticas de regulación financiera.

2.2.1. Datos de Alta Frecuencia

Los datos de alta frecuencia (HF) representan el registro de precios y volúmenes de transacciones de activos financieros en intervalos de tiempo muy cortos, como segundos o milisegundos. Estos datos incluyen información detallada sobre cada transacción, como el precio, la cantidad negociada y el tiempo exacto de la transacción. A continuación, se detallan algunas características clave de los datos de alta frecuencia:

- **Granularidad Temporal**: Los datos se registran en intervalos muy cortos, lo que permite un análisis detallado de la dinámica del mercado en el corto plazo.
- **Volumen de Datos**: La gran cantidad de transacciones que ocurren en cortos periodos de tiempo genera volúmenes masivos de datos que deben ser almacenados y procesados.
- **Precisión**: Incluyen información precisa sobre el precio y el volumen de cada transacción, así como el tiempo exacto en que ocurrieron.
- **Eventos de Mercado**: Los datos de alta frecuencia capturan eventos de mercado que no son visibles en datos de menor frecuencia, como órdenes de compra y venta, cambios en la profundidad del mercado y reacciones instantáneas a noticias.

A pesar de sus ventajas, los datos de alta frecuencia presentan varios desafíos:

- **Costo y Accesibilidad**: Los datos de alta frecuencia son difíciles y costosos de obtener, ya que generalmente requieren suscripciones a servicios de datos financieros especializados y costosos.
- **Procesamiento y Almacenamiento**: El volumen masivo de datos requiere infraestructuras avanzadas para su almacenamiento y procesamiento eficiente.
- **Ruido y Volatilidad**: La alta granularidad temporal de estos datos incluye mucho ruido, lo que puede complicar el análisis y modelado.

Debido a estos desafíos, en este trabajo se utilizarán datos de minuto. Los datos de minuto representan un compromiso entre la granularidad y la manejabilidad, proporcionando suficiente detalle para un análisis robusto sin los costos y complejidades asociados con los datos de alta frecuencia. Estos datos son más accesibles y permiten capturar las tendencias y patrones intradía sin la sobrecarga de procesamiento asociada con datos de mayor frecuencia.

2.2.2. Datos Utilizados

En este trabajo se utilizarán dos conjuntos de datos financieros: los datos de Bitcoin y los datos del SPY (SPDR S&P 500 500 ETF Trust). A continuación, se describen estos datos, se muestran sus series temporales y se analizan sus propiedades estadísticas para verificar si cumplen con los hechos estilizados mencionados anteriormente.

Datos de Bitcoin

Bitcoin es una criptomoneda y un sistema de pago descentralizado que ha ganado popularidad y relevancia en los mercados financieros. Los datos de Bitcoin que utilizaremos incluyen los precios de cierre por minuto. A continuación se muestra la serie temporal de precios de Bitcoin:

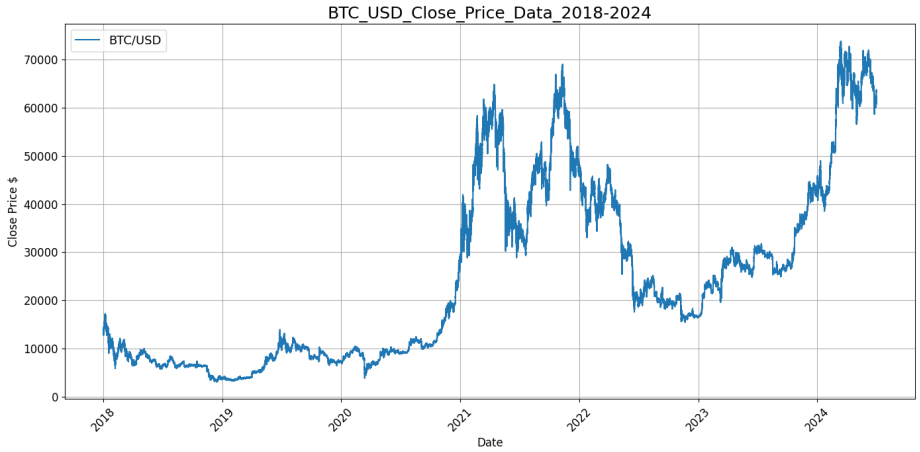

Figura 2.2: Serie Temporal de Precios de Bitcoin

Datos del SPY

El SPY es un ETF que sigue el rendimiento del índice S&P 500. Los datos del SPY que utilizaremos también incluyen los precios de cierre por minuto. A continuación se muestra la serie temporal de precios del SPY:

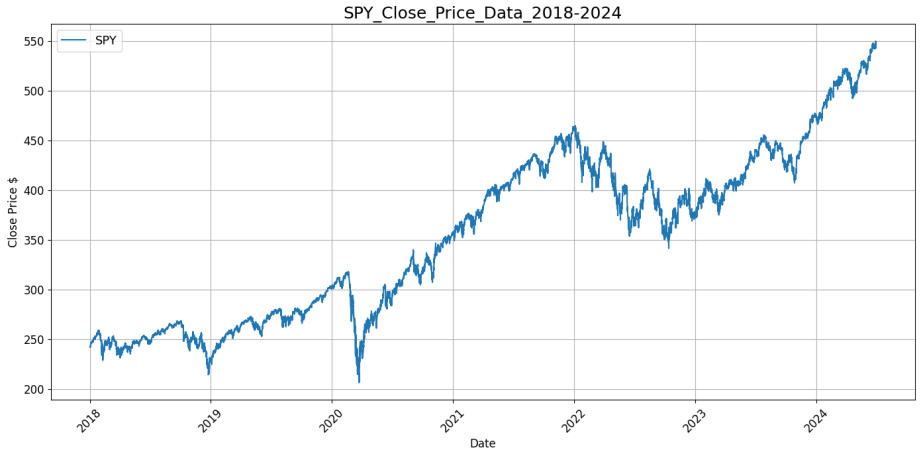

Figura 2.3: Serie Temporal de Precios del SPY

Este análisis permitirá verificar si los datos de Bitcoin y del SPY cumplen con los hechos estilizados descritos por Cont (2001) [2], proporcionando una base sólida para el modelado y la estrategia de trading propuesta en este trabajo.

2.3. Métricas Financieras

Para evaluar el rendimiento de una estrategia de trading, es crucial utilizar métricas financieras adecuadas que permitan comparar y analizar los resultados obtenidos. A continuación, se describen algunas de las métricas financieras más utilizadas junto con sus expresiones matemáticas.

Cumulative Returns

Los *Cumulative Returns* (*Retornos Acumulados*) representan el total de los retornos obtenidos por una inversión desde el inicio hasta un punto en el tiempo. Se calculan como el producto de los rendimientos individuales de cada período.

$$\text{Cumulative Returns} = \prod_{i=1}^{n}(1 + r_i) - 1, \tag{2.1}$$

donde r_i es el retorno en el período i.

Annualized Return

El *Annualized Return* (*Retorno Anualizado*) mide el rendimiento medio anual de una inversión, ajustando el rendimiento acumulado para que se exprese en una base anual.

$$\text{Annualized Return} = (1 + \text{Cumulative Returns})^{\frac{1}{T}} - 1, \tag{2.2}$$

donde T es el número de años.

Sharpe Ratio

El *Sharpe Ratio* mide el rendimiento ajustado al riesgo de una inversión, comparando el exceso de retorno de la inversión sobre la tasa libre de riesgo con la volatilidad de los retornos.

$$\text{Sharpe Ratio} = \frac{E[R_p - R_f]}{\sigma_p}, \tag{2.3}$$

donde R_p es el retorno de la cartera, R_f es la tasa libre de riesgo, y σ_p es la desviación estándar de los retornos de la cartera.

Sortino Ratio

El *Sortino Ratio* es una variación del Sharpe Ratio que solo considera la volatilidad de los retornos negativos (downside risk), proporcionando una medida del rendimiento ajustado al riesgo que penaliza solo los retornos por debajo de un umbral.

$$\text{Sortino Ratio} = \frac{E[R_p - R_f]}{\sigma_d}, \tag{2.4}$$

donde σ_d es la desviación estándar de los retornos negativos.

Maximum Drawdown

El *Maximum Drawdown* (*Máxima Pérdida*) mide la mayor caída en el valor de una cartera desde un máximo anterior hasta un mínimo posterior, lo que indica el peor comportamiento de la estrategia en términos de pérdida de valor.

$$\text{Maximum Drawdown} = \min_{t\in[0,T]}\left(\frac{V_t - V_{\max,t}}{V_{\max,t}}\right), \tag{2.5}$$

donde V_t es el valor de la cartera en el tiempo t, y $V_{\max,t}$ es el valor máximo de la cartera hasta el tiempo t.

Calmar Ratio

El *Calmar Ratio* mide el rendimiento ajustado al riesgo de una inversión, similar al Sharpe Ratio, pero utilizando el *Maximum Drawdown* como medida del riesgo.

$$\text{Calmar Ratio} = \frac{\text{Annualized Return}}{\text{Maximum Drawdown}}. \tag{2.6}$$

Value at Risk (VaR)

El *Value at Risk* (*VaR*) mide la pérdida máxima esperada de una inversión en un período determinado, con un nivel de confianza específico. Es una medida de riesgo que indica el valor en riesgo.

$$\text{VaR}_\alpha = -\text{Quantile}_\alpha(R_p), \tag{2.7}$$

donde α es el nivel de confianza y R_p es la distribución de retornos de la cartera.

Alpha

El *Alpha* mide el rendimiento adicional que una inversión genera en comparación con su benchmark, ajustado por el riesgo. Es un indicador de la habilidad del gestor de la cartera para generar rendimientos superiores.

$$\alpha = R_p - \left(R_f + \beta(R_m - R_f)\right), \tag{2.8}$$

donde R_m es el retorno del mercado, y β es la sensibilidad de la cartera al mercado.

Beta

El *Beta* mide la sensibilidad de los retornos de una cartera en relación con los movimientos del mercado. Un β mayor que 1 indica que la cartera es más volátil que el mercado.

$$\beta = \frac{\text{Cov}(R_p, R_m)}{\sigma_m^2}, \tag{2.9}$$

donde $\text{Cov}(R_p, R_m)$ es la covarianza entre los retornos de la cartera y los del mercado, y σ_m^2 es la varianza de los retornos del mercado.

2.4. Métricas de Machine Learning

Las métricas de evaluación son fundamentales para medir el rendimiento de los modelos de *Machine Learning*, especialmente en el contexto de clasificación. A continuación, se describen algunas de las métricas más utilizadas en la evaluación de modelos de clasificación.

Precision

La *Precision* (Precisión) mide la proporción de verdaderos positivos (TP) entre todas las predicciones positivas realizadas por el modelo. Es una medida de la exactitud del modelo cuando predice la clase positiva.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{2.10}$$

donde TP son los verdaderos positivos y FP son los falsos positivos.

Recall

El *Recall* (Cobertura o Sensibilidad) mide la proporción de verdaderos positivos (TP) entre todos los ejemplos que realmente son positivos. Es una medida de la capacidad del modelo para identificar correctamente las instancias positivas.

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{2.11}$$

donde FN son los falsos negativos.

F1-Score

El *F1-Score* es la media armónica de la *Precision* y el *Recall*, proporcionando un balance entre ambas métricas. Es especialmente útil cuando se desea tener un equilibrio entre la precisión y la cobertura.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.12}$$

Capítulo 3

Barras de desequilibrio

3.1. Barras Basadas en Información

En la industria financiera, es común el uso de barras de tiempo para transformar series de observaciones que llegan a intervalos irregulares en series homogéneas derivadas de un muestreo regular. Las barras de tiempo, obtenidas al muestrear información en intervalos de tiempo fijos (por ejemplo, cada minuto), suelen incluir datos como la marca temporal, el precio de apertura, el precio de cierre, el precio más alto, el más bajo, y el volumen negociado. Estos datos se conocen como datos *ohclv*.

Aunque las barras de tiempo son populares tanto entre los profesionales como entre los académicos, el libro de Marcos López de Prado introduce las limitaciones que presentan este tipo de datos:

- Los mercados no procesan información a intervalos de tiempo constantes. Por ejemplo, la actividad es significativamente mayor en las horas inmediatamente posteriores a la apertura, en comparación con periodos de menor actividad, como el mediodía.

- Las barras de tiempo tienden a sobre-representar la actividad durante periodos tranquilos y sub-representar durante momentos de alta actividad, lo que distorsiona la representatividad de los datos.

- Las series temporales basadas en tiempo suelen exhibir propiedades estadísticas deficientes, como correlación serial, heteroscedasticidad y no-normalidad de los retornos, lo que complica el modelado y análisis de los datos.

Formar barras basadas en la información de mercado, en lugar de intervalos de tiempo fijos, es una alternativa que mejora la representatividad de los datos y las propiedades estadísticas de las series temporales generadas.

Este tipo de barras basadas en información ajustan dinámicamente el tamaño de las barras en respuesta a la llegada de nueva información, lo que permite una representación más precisa de las condiciones de mercado.Este enfoque es útil para identificar y reaccionar ante la presencia de traders informados, quienes pueden provocar desequilibrios en los precios.

3.1.1. Barras de Desequilibrio

Las barras de desequilibrio son un tipo de barra basada en información que ajusta el muestreo según la actividad del mercado en lugar de hacerlo a intervalos de tiempo fijos. Este enfoque permite capturar de manera más efectiva los momentos en que se produce nueva información relevante, mejorando así la capacidad de respuesta ante los cambios del mercado.

La Regla del Tick

En el ámbito de la microestructura del mercado, es esencial comprender cómo se generan y clasifican las operaciones. En un libro de órdenes de subasta doble, se registran cotizaciones para vender (ofertas) y comprar (demandas) un valor a diferentes niveles de precios. Las operaciones ocurren cuando un comprador coincide con una oferta o un vendedor con una demanda. La regla del tick es una herramienta que permite identificar el lado agresor de cada operación. Esta regla clasifica una transacción como iniciada por el comprador si el precio sube ($\Delta p_t > 0$) o por el vendedor si el precio baja ($\Delta p_t < 0$). Si el precio se mantiene igual ($\Delta p_t = 0$), la clasificación se mantiene según el último tick registrado:

$$b_t = \begin{cases} 1 & \text{si } \Delta p_t > 0 \\ -1 & \text{si } \Delta p_t < 0 \\ b_{t-1} & \text{si } \Delta p_t = 0 \end{cases} \tag{3.1}$$

donde p_t es el precio de la operación indexado por $t = 1, \ldots, T$ y b_0 se establece arbitrariamente en 1. La regla del tick, a pesar de su simplicidad, ha demostrado ser efectiva en la clasificación de transacciones, con una alta precisión documentada en varios estudios (Aitken y Frino, 1996).

Tipos de Barras de Desequilibrio

Barras de Desequilibrio de Tick (TIB) Las barras de desequilibrio de tick se basan en la idea de que el desequilibrio en los ticks puede revelar información importante. Se consideran secuencias de ticks donde cada tick tiene un precio p_t y un volumen v_t. La regla del tick se utiliza para generar una secuencia $\{b_t\}$ que clasifica cada tick como compra o venta. El desequilibrio de tick en un intervalo se define como la suma de los ticks clasificados:

$$\theta_T = \sum_{t=1}^{T} b_t \tag{3.2}$$

Para determinar cuándo muestrear una nueva barra, se calcula el desequilibrio esperado θ_T al inicio de la barra. Este se estima como:

$$E_0[\theta_T] = E_0[T](2P[b_t = 1] - 1) \tag{3.3}$$

donde $E_0[T]$ es el tamaño esperado de la barra, y $P[b_t = 1]$ y $P[b_t = -1]$ son las probabilidades de que un tick se clasifique como compra o venta, respectivamente. Una TIB se genera cuando el desequilibrio acumulado excede un umbral basado en estas expectativas:

$$T^* = \arg\min_T \{|\theta_T| \geq E_0[\theta_T]|2P[b_t = 1] - 1|\} \tag{3.4}$$

Este tipo de barras se utiliza principalmente en datos de transacciones, ya que ofrecen una visión detallada de cada tick en el mercado.

Barras de Desequilibrio de Volumen y Dólares (VIB y DIB) Las barras de desequilibrio de volumen y de dólares extienden el concepto de las TIB al considerar el volumen y el valor en dólares de las transacciones, respectivamente. El objetivo es identificar desequilibrios en el volumen de transacciones o el valor monetario que puedan indicar la presencia de información nueva y relevante.

El desequilibrio en un intervalo se define como:

$$\theta_T = \sum_{t=1}^{T} b_t v_t \tag{3.5}$$

donde v_t representa el volumen negociado o la cantidad en dólares intercambiado. El valor esperado de este desequilibrio se calcula como:

$$E_0[\theta_T] = E_0[T](v_+ - v_-) = E_0[T](2v_+ - E_0[v_t]) \tag{3.6}$$

Aquí, v_+ y v_- representan la contribución esperada del volumen de las compras y ventas, respectivamente. En la práctica, $E_0[T]$ y $2v_+ - E_0[v_t]$ se estiman usando promedios móviles ponderados exponencialmente. Una VIB o DIB se define como un subconjunto T^* contiguo de ticks tal que:

$$T^* = \arg\min_T \{|\theta_T| \geq E_0[T]|2v_+ - E_0[v_t]|\} \tag{3.7}$$

Cuando el desequilibrio θ_T excede las expectativas, las barras se generan con mayor frecuencia, reflejando la presencia de traders informados y ajustando dinámicamente el tamaño de las barras según la información disponible.

3.1.2. Adaptación de la Metodología con Datos de Minuto

La metodología presentada en el libro de Marcos López de Prado está diseñada originalmente para trabajar con datos de tick, los cuales proporcionan un registro extremadamente detallado de cada transacción individual en el mercado. Los datos de tick incluyen información específica sobre el precio, el volumen y el tiempo exacto en el que se realiza cada operación, lo que permite una representación precisa y granular de la actividad del mercado. Este nivel de detalle permite capturar la dinámica completa del mercado, especialmente cuando se busca identificar patrones como desequilibrios en la oferta y la demanda que pueden indicar la presencia de traders informados.

Los datos de tick son valiosos porque reflejan cada cambio en el mercado en tiempo real, permitiendo un análisis fino de la microestructura del mercado. Sin embargo, obtener estos datos puede ser complicado y costoso, debido a la gran cantidad de información que se genera y la necesidad de acceder a servicios de datos financieros especializados.

Dada la dificultad para obtener datos de tick, en este estudio se opta por utilizar datos de minuto, debido a su accesibilidad y facilidad de manejo. Los datos de minuto agregan todas las transacciones que ocurren dentro de un minuto, ofreciendo una

visión resumida de la actividad del mercado durante ese intervalo de tiempo. Aunque esta aproximación sacrifica cierta granularidad, sigue siendo adecuada para el análisis de tendencias y patrones en el mercado.

Se han obtenido series temporales de datos de minuto para Bitcoin y el ETF SPY a través de las APIs de Alpha Vantage y Financial Modelling Prep, de forma gratuita. Estos datos fueron procesados para incluir únicamente las transacciones realizadas dentro del horario de mercado: 24 horas al día en el caso de Bitcoin, y el horario estándar del mercado para el ETF SPY. Además, se han interpolado las muestras faltantes propagando hacia adelante el valor del minuto anterior, asegurando así la continuidad de la serie temporal.

Para implementar la metodología de barras de desequilibrio con estos datos, se utilizará el precio de cierre de cada minuto como proxy del precio de transacción. Este enfoque simplifica la aplicación de la metodología, permitiendo su adaptación a los datos disponibles, mientras se mantiene la integridad del análisis basado en la llegada de nueva información relevante al mercado.

Implementación

La implementación de las barras de desequilibrio requiere una cuidadosa consideración de los parámetros iniciales, ya que estos son fundamentales para el cálculo del primer valor esperado de ticks en una barra ($E_0[T]$). Al inicio del proceso, no se dispone de barras previas que puedan proporcionar una estimación de $E_0[T]$, por lo que es necesario hacer una suposición inicial. A medida que se generan más barras, $E_0[T]$ se ajusta dinámicamente utilizando un promedio móvil exponencialmente ponderado (EWMA) basado en los valores de T de las barras anteriores.

Los gráficos presentados en las Figuras 3.2 y 3.1 muestran el número de barras generadas para el ETF SPY y el Bitcoin (BTC) al aplicar la metodología de barras de desequilibrio, utilizando como parámetros iniciales `ewma_window`, `T_init` y `imbalance_init`. Los datos del SPY cuentan con 662,353 muestras iniciales, mientras que los del BTC cuentan con 3,417,119. El parámetro `imbalance_init` se inicializa con la media histórica del volumen, mientras que `ewma_window` y `T_init` se han ajustado a través de un barrido de valores.

Los gráficos reflejan cómo la variación de estos parámetros afecta la frecuencia de barras generadas. Específicamente, con parámetros iniciales bajos, la cantidad de desequilibrio necesario para generar una nueva barra aumenta de manera explosiva, lo que lleva a una rápida reducción en la frecuencia de barras generadas. Por otro lado, con parámetros iniciales altos, el desequilibrio necesario para el sampleo es demasiado bajo, resultando en un número de barras generadas que es prácticamente idéntico al número de barras iniciales.

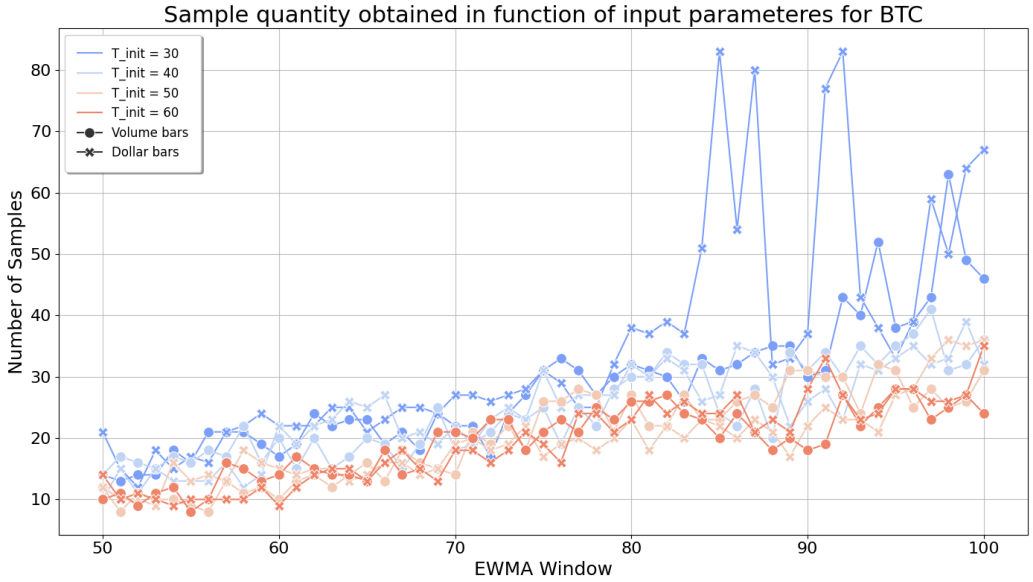

Figura 3.1: Número de barras obtenido en datos de BTC

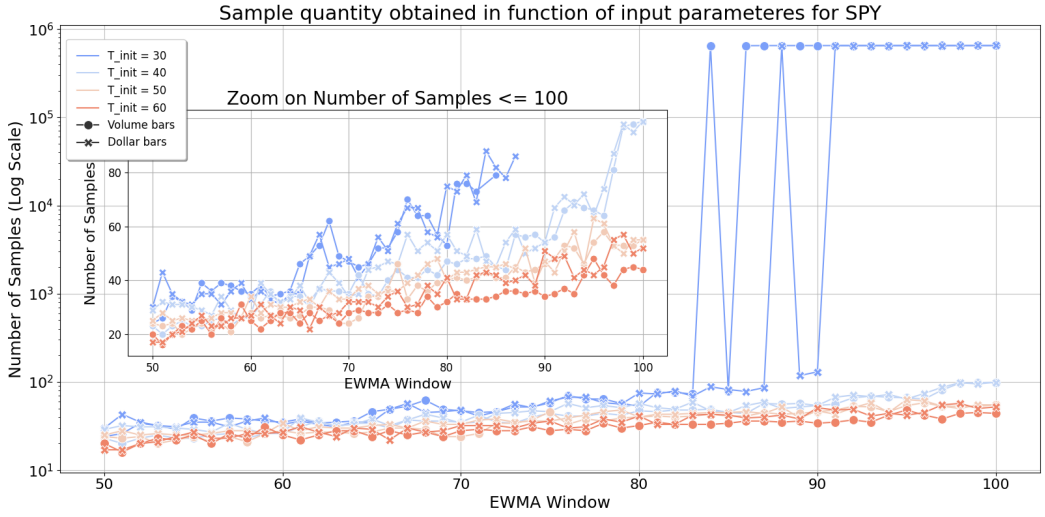

Figura 3.2: Número de barras obtenido en datos del SPY

Este comportamiento muestra un problema crítico en la implementación: la dinámica de generación de barras tiende a explotar, tal y como se muestra en la figura 3.3. A medida que se van generando nuevas barras, el humbral de desequilibrio, es decir, el producto de la multiplicacion del tamaño esperado de la barra por el desequilibrio esperado, provoca que el humbral de desequilibrio sea cada vez mayor.

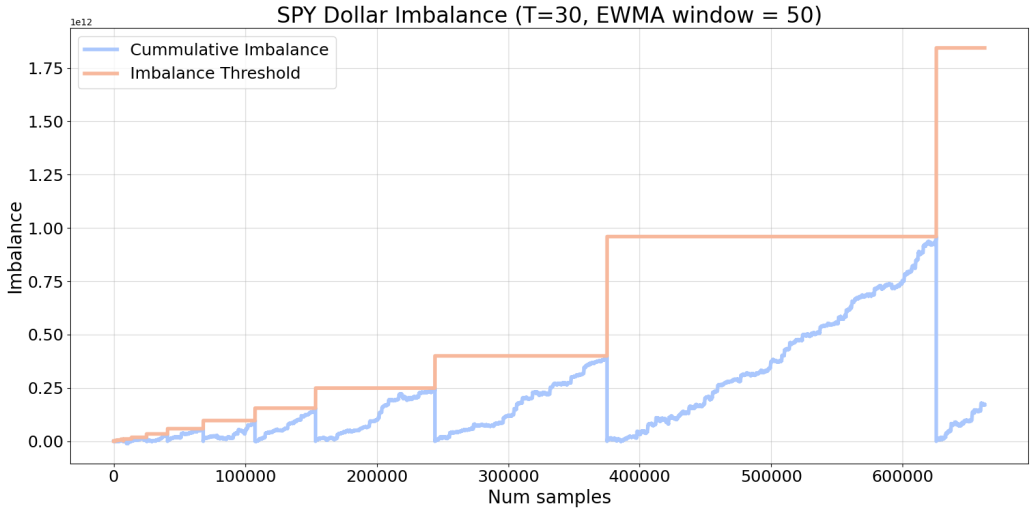

Figura 3.3: Explosión del umbral de desequilibrio en SPY

Para hacer frente a este problema, algunos investigadores han propuesto modificaciones a la implementación sugerida, tales como establecer un límite máximo al número esperado de ticks por barra o seleccionar directamente un umbral fijo de desequilibrio [5].

3.1.3. Resultados

Para este trabajo, se ha optado por aplicar la metodología con umbrales de desequilibrio fijos, definidos experimentalmente. Para cada activo y tipo de desequilibrio se ha seleccionado un umbral distinto, tal y como se muestra en la tabla 3.1.

Tabla 3.1: Umbrales utilizados y numero de muestras obtenido

Activo	**Desequilibrio**	**Umbral**	**Muestras iniciales**	**Muestras finales**	**Reducción (%)**
SPY	Volumen	6×10^5	662,353	55,430	91.63 %
SPY	Dólar	$1{,}5 \times 10^8$	662,353	85,228	87.13 %
BTC	Volumen	5×10^6	3,417,119	110,772	96.76 %
BTC	Dólar	$1{,}5 \times 10^{10}$	3,417,119	1,003,756	70.63 %

Los umbrales de desequilibrio se han definido con el fin de obtener una disminución significativa del numero de muestras de cada activo. Las disminuciones en la cantidad de datos van desde el 70 % para el desequilibrio de dolares de bitcoin, hasta el 96 % para el desequilibrio en volumen del mismo activo.

En la Figura 3.4 se muestra una comparativa de los datos de precio originales de bitcoin y los de desequilibrio de volumen durante durante un segmento perteneciente al período de alta volatilidad ocasionada por la pandemia de COVID-19. Se puede observar cómo, para el mismo período de tiempo, las barras generadas mediante la metodología de desequilibrio son menos numerosas que las barras de minuto originales.

Este fenómeno se debe a la agregación dinámica de los datos basada en la llegada de nueva información relevante, lo que permite una representación más frecuente durante periodos de alta actividad de mercado.

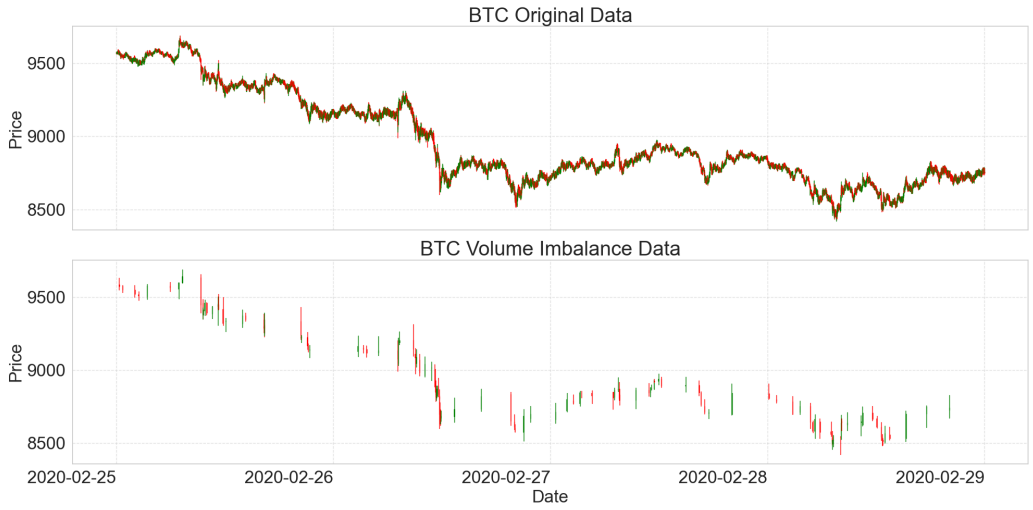

Figura 3.4: Datos BTC originales y de Desequilibrio de Volumen

El artículo de Easley, López de Prado y O'Hara titulado "The Volume Clock: Insights into the High Frequency Paradigm"[4] expone que el uso de un marco temporal basado en volumen, en lugar de un marco cronológico tradicional, ofrece ventajas estadísticas significativas. Entre ellas, se destaca que este enfoque permite una reducción de los efectos estacionales intra-sesión y contribuye a una recuperación parcial de la normalidad en la distribución de los retornos financieros. Para evaluar esta afirmación, se ha aplicado la metodología de desequilibrio propuesta por los autores a los activos SPY y BTC, generando series temporales basadas en volumen y dinero.

Con el fin de verificar la normalidad de los retornos obtenidos mediante esta metodología, se han realizado pruebas estadísticas de normalidad, incluyendo los tests de Kolmogorov-Smirnov, Anderson-Darling y Jarque-Bera, cuyos resultados se resumen en la Tabla 3.2. Los resultados muestran que las series temporales originales de ambos activos presentan una desviación significativa de la normalidad, con valores extremadamente elevados de curtosis y asimetría, especialmente en el caso del SPY.

Tras la aplicación de barras dinámicas basadas en volumen y dolares, se observa una reducción considerable en la curtosis y asimetría de los retornos. En particular, las barras basadas en volumen han demostrado ser más efectivas para aproximar la distribución de retornos a una forma más cercana a la normalidad. Estos resultados confirman en gran medida las hipótesis expuestas en el trabajo de Easley, López de Prado y O'Hara, evidenciando que el uso de un marco temporal basado en volumen contribuye a mejorar las propiedades estadísticas de las series temporales, aunque persisten algunas desviaciones de la normalidad, especialmente en los datos de BTC.

Tabla 3.2: Resultados de pruebas estadísticas de normalidad para SPY y BTC

Asset	Series	K-S Test p-value	Anderson-Darling Statistic	Jarque-Bera p-value	Skewness	Kurtosis
SPY	Original	0.0	46037.02	0.0	-11.83	2981.95
SPY	Volume	0.0	1553.58	0.0	-3.46	249.67
SPY	Dollar	0.0	2947.14	0.0	-4.24	383.73
BTC	Original	0.0	646407.27	0.0	0.012	44.68
BTC	Volume	0.0	3131.35	0.0	-0.026	26.13
BTC	Dollar	0.0	97255.28	0.0	0.070	139.58

Además de las pruebas estadísticas, se han generado cuatro representaciones visuales que comparan las distribuciones de retornos de los activos SPY y BTC antes y después de aplicar la metodología basada en volumen y dinero. La Figura 3.5 muestra una comparativa de los histogramas de los retornos para el SPY y el BTC, donde se observa cómo la transformación mediante barras dinámicas logra atenuar la asimetría y la curtosis extrema presentes en las series temporales originales.

Por otro lado, en la Figura 3.6, se presentan los QQ-plots correspondientes, que permiten evaluar visualmente el grado de ajuste de los retornos a una distribución normal. Los QQ-plots evidencian una mejora en la alineación con la diagonal en las series ajustadas, especialmente para el SPY, lo que sugiere una aproximación más cercana a la normalidad tras la aplicación de la metodología propuesta.

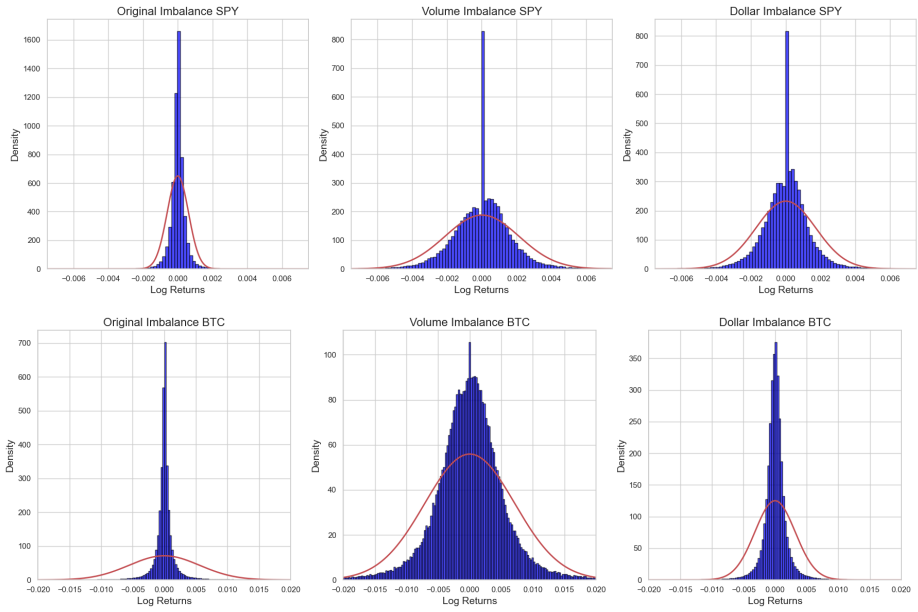

Figura 3.5: Comparativa de histogramas de los retornos del SPY (arriba) y BTC (abajo)

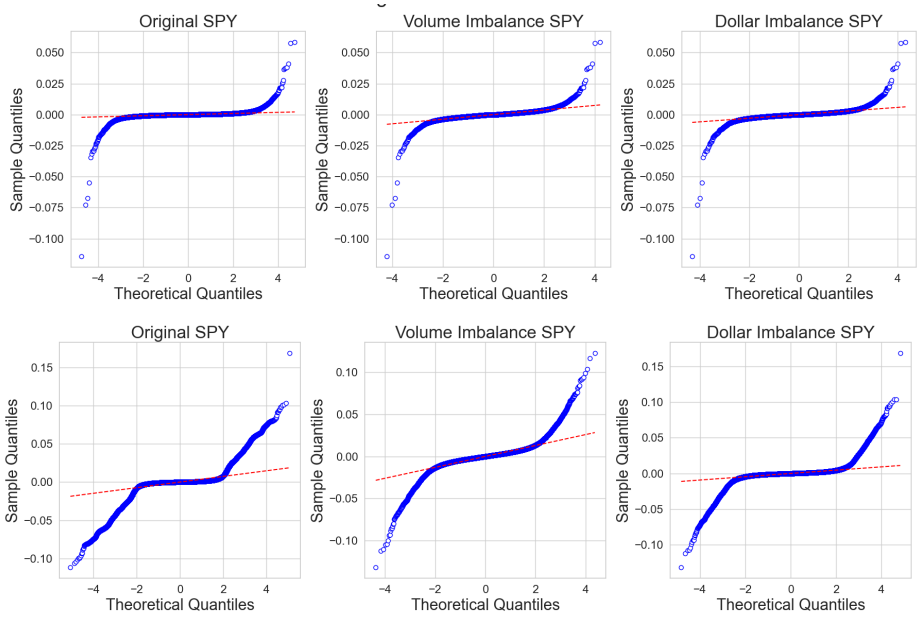

Figura 3.6: Comparativa de QQ-plots de los retornos del SPY (arriba) y BTC (abajo)

Capítulo 4

Deep Reinforcement Learning

4.1. Introducción al Aprendizaje por Refuerzo (Reinforcement Learning)

4.1.1. Definición y Contexto General

El *Aprendizaje por Refuerzo* (Reinforcement Learning, RL) es una rama de la inteligencia artificial que se centra en cómo un agente aprende a tomar decisiones secuenciales optimizadas mediante la interacción con un entorno dinámico. A diferencia de otros tipos de aprendizaje, como el *aprendizaje supervisado*, donde el modelo aprende patrones extraídos a partir de ejemplos previamente etiquetados, o el *aprendizaje no supervisado*, donde el modelo busca patrones en datos sin etiquetas, el RL implica un aprendizaje activo donde el agente recibe retroalimentación en forma de recompensas o penalizaciones por las acciones que toma. Este proceso permite al agente aprender sin ejemplos de comportamiento óptimo, optimizando en su lugar una señal de recompensa.

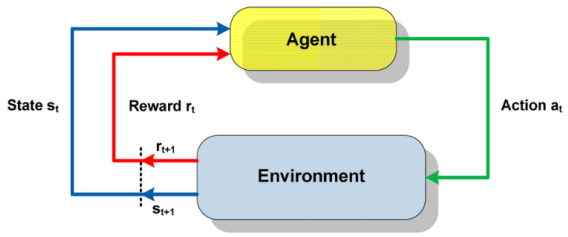

Figura 4.1: Esquema Reiforcement Learning

En cada paso de tiempo t, el agente recibe una una observación O_t y una recompensa R_t. La observación recibida le permite ejecutar una acción A_t, la cual provocará que el entorno cambie y emita una nueva observación O_{t+1} y una nueva recompensa R_{t+1}.

Hipótesis de la Recompensa

Una recompensa R_t es una señal de retroalimentación escalar que indica cómo de buena ha sido la accion tomada por el agente en el paso t.

A partir de esta señal de recopensa se formula la *hipótesis de la recompensa*, que afirma que en un esquema de aprendizaje por refuerzo, cualquier objetivo puede ser formalizado como el resultado de maximizar una recompensa acumulativa. Es decir, el objetivo del agente puede ser expresado como la maximización de una función de recompensa a lo largo del tiempo.

Esta función de recompensa se conoce como *retorno* (G_t), y se define matemáticamente como el sumatorio de las recompensas futuras, descontadas por un factor de descuento γ:

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}$$

El retorno G_t es lo que el agente intenta maximizar, y es la base sobre la cual se definen otros elementos como la función de valor.

Esto proporciona un marco ampliamente aplicable en situaciones donde las decisiones se toman en una serie de pasos, lo que incluye problemas de control robótico, juegos, y *trading* financiero.

En el contexto particular del trading financiero, debido a la naturaleza secuencial y dinámica de los mercados, los agentes deben tomar decisiones basadas en información incompleta y en constante cambio, buscando maximizar una recompensa a largo plazo, como la rentabilidad de una cartera.

4.1.2. Componentes Básicos del Aprendizaje por Refuerzo

El aprendizaje por refuerzo se formaliza mediante un *Proceso de Decisión de Markov* (MDP), que se define por un cuarteto $\langle S, A, P, R \rangle$ el cual describe el problema de decisión.

Un MDP se representa por un conjunto de estados S, un conjunto de acciones A, una función de transición de estados $P(s'|s, a)$, y una función de recompensa $R(s, a)$. El objetivo del agente es aprender una política $\pi(a|s)$ que maximice la recompensa acumulada a lo largo del tiempo.

Un proceso de decisión se dice que es Markov si la probabilidad de transición depende únicamente del estado y acción actuales, no de la historia completa:

$$p(r, s'|S_t, A_t) = p(r, s'|H_t, A_t)$$

Esto significa que el estado contiene toda la información relevante de la historia, siendo la historia la secuencia completa de observaciones, acciones y recompensas. No quiere decir que la historia contenga toda la información, sino más bien que el añadir cualquier información adicional no afectará la decisión.

- **Estados (S)**: Representan todas las posibles situaciones en las que se puede encontrar el agente dentro del entorno.
- **Acciones (A)**: Conjunto de todas las decisiones que el agente puede tomar desde cualquier estado dado.
- **Función de Transición de Estados (P)**: Describe la probabilidad de transitar del estado s al estado s' como resultado de realizar una acción a, es decir, $P(s'|s, a)$.

- **Función de Recompensa (R)**: Proporciona una recompensa r que el agente recibe al transitar del estado s al estado s' al realizar la acción a, es decir, $R(s, a, s')$.

El objetivo del agente en RL es aprender una *política* $\pi(a|s)$ que maximice la recompensa acumulada a lo largo del tiempo. La política puede ser determinista, donde una acción específica es seleccionada para cada estado, o estocástica, donde se asigna una distribución de probabilidad a las acciones en cada estado.

Un concepto fundamental en RL es la *función de valor*, que mide cuán favorable es un estado o una acción en términos de la recompensa esperada:

- **Función de Valor de Estado ($V^\pi(s)$)**: Estima la recompensa total esperada comenzando desde el estado s y siguiendo la política π. Formalmente, se define como:

$$V^\pi(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, \pi\right]$$

  donde $\gamma \in [0, 1]$ es el factor de descuento que determina la importancia de las recompensas futuras. Esta función refleja el valor a largo plazo de estar en un estado particular bajo una política dada.

  La función de valor del estado, $V^\pi(s)$, trata de cuantificar *cómo de bueno* es estar en un estado s cuando se sigue una política π.

- **Función de Valor de Acción ($Q^\pi(s, a)$)**: Extiende la función de valor del estado para considerar no solo el estado s, sino también la acción a tomada en ese estado. Esta función estima la recompensa total esperada al tomar la acción a en el estado s y seguir la política π posteriormente:

$$Q^\pi(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right]$$

  La función de valor de la acción, $Q^\pi(s, a)$, trata de medir *cómo de buena* es una acción a específica en un estado s determinado cuando se sigue la política π después.

En resumen, la función del valor del estado, $V^\pi(s)$, mide cuán bueno es estar en un estado dado bajo una política π, mientras que $Q^\pi(s, a)$ mide cuán buena es una acción específica en ese estado.

Política

La **política** (*policy*) es la función que guía la selección de acciones en función del estado actual del entorno. Existen dos variantes principales:

- **Política determinista:** Representada como $\pi(s)$, esta política asigna una acción específica a a cada estado s. Formalmente, $\pi(s) = a$, donde a es la acción elegida para el estado s.

- **Política estocástica:** Denotada como $\pi(a|s)$, esta política define una distribución de probabilidad sobre las acciones posibles. Es decir, $\pi(a|s) = \mathbb{P}(A = a \mid S = s)$, donde a es seleccionada según la probabilidad correspondiente en el estado s.

La política determinista es útil en entornos estables, mientras que la estocástica es preferible en situaciones con alta incertidumbre o donde la exploración es necesaria.

Exploración vs. Explotación

En el Aprendizaje por Refuerzo, un desafío clave es el equilibrio entre *exploración* y *explotación*. Este balance es fundamental para el aprendizaje efectivo del agente:

- **Exploración**: El agente intenta nuevas acciones para descubrir si existen mejores recompensas disponibles. La exploración es crucial cuando el agente no tiene suficiente conocimiento del entorno y necesita recolectar más información.
- **Explotación**: El agente selecciona la mejor acción conocida basada en su política actual para maximizar la recompensa inmediata. La explotación se basa en la información acumulada y tiene como objetivo maximizar la recompensa a corto plazo.

El *trade-off* entre exploración y explotación es crucial porque un agente que explora demasiado podría no aprovechar las recompensas conocidas, mientras que un agente que explota demasiado podría perder oportunidades para descubrir estrategias más efectivas.

Existen diversas estrategias para manejar este trade-off, como la *política epsilon-greedy*, donde con una probabilidad ϵ, el agente explora una acción al azar, y con una probabilidad $1 - \epsilon$, explota la mejor acción conocida. El valor de ϵ generalmente disminuye a medida que el agente gana más confianza en su política, lo que permite una mayor explotación con el tiempo.

4.1.3. El Problema de la Toma de Decisiones Óptimas en el Tiempo - La ecuación de Bellman

En el contexto del aprendizaje por refuerzo, uno de los problemas fundamentales es **cómo un agente puede tomar decisiones óptimas en situaciones donde las consecuencias de sus acciones no son inmediatas, sino que se extienden en el tiempo**. Las decisiones actuales no solo influyen en la recompensa inmediata, sino también en las recompensas futuras.

Algunas decisiones pueden ofrecer beneficios inmediatos pero desviar al agente de un objetivo más valioso en el futuro, mientras que otras pueden tener un alto coste presente, pero ser cruciales para un éxito posterior.

Este desafío, conocido como el problema de la maximización del retorno total esperado, requiere una estrategia que considere tanto las recompensas inmediatas como las futuras, ponderadas para reflejar su importancia relativa. La ecuación de Bellman se introduce como una herramienta esencial para descomponer este problema global en subproblemas más manejables, **permitiendo al agente evaluar el impacto a largo plazo de sus decisiones actuales de manera eficiente y sistemática**.

La ecuación de Bellman descompone el problema global de maximización del retorno total esperado en subproblemas más manejables.

El valor de un estado $V^{\pi}(s)$ bajo una política π se define como el valor esperado del retorno total que se puede obtener comenzando desde ese estado s y siguiendo la política π. Matemáticamente, se expresa como:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \mid s_t = s\right] \tag{4.1}$$

Esta ecuación describe el valor esperado del retorno acumulado desde el estado s. Para hacerla más práctica, es posible descomponer esta expresión en dos partes: la recompensa inmediata y el valor de los estados futuros. Separando el primer término de la suma, se obtiene:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[R_{t+1} + \gamma \sum_{k=1}^{\infty} \gamma^{k-1} R_{t+k+1} \mid s_t = s\right] \tag{4.2}$$

Aquí, R_{t+1} es la recompensa inmediata obtenida al tomar una acción en el estado s, y el resto de la suma representa el valor descontado de las recompensas futuras, que corresponde al valor de $V^{\pi}(s_{t+1})$. De esta forma, es posible reescribir la ecuación como:

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[R_{t+1} + \gamma V^{\pi}(s_{t+1}) \mid s_t = s\right] \tag{4.3}$$

Para expandir esta expectativa, se consideran todas las posibles acciones a que el agente puede tomar en el estado s, y todas las posibles transiciones a estados futuros s' después de tomar la acción a. Esto lleva a la forma completa de la ecuación de Bellman:

$$V^{\pi}(s) = \sum_{a} \pi(a|s) \sum_{s'} P(s'|s,a)\left[R(s,a,s') + \gamma V^{\pi}(s')\right] \tag{4.4}$$

Aquí:

- $\pi(a|s)$ es la probabilidad de tomar la acción a dado el estado s.
- $P(s'|s,a)$ es la probabilidad de transición al estado s' dado el estado s y la acción a.
- $R(s,a,s')$ es la recompensa inmediata por la transición de s a s' mediante la acción a.
- γ es el factor de descuento, que pondera la importancia relativa de las recompensas futuras.

Esta ecuación establece que el valor de un estado es la suma de la recompensa inmediata y el valor futuro esperado, ponderado por las probabilidades de las acciones y las transiciones de estado. Así, la ecuación de Bellman permite calcular los valores de los estados de manera recursiva, evaluando tanto el impacto inmediato como el impacto a largo plazo de las decisiones actuales.

Cálculo Numérico: Iteración de Valores

Para calcular los valores $V^{\pi}(s)$ numéricamente, se suele utilizar un método iterativo. Inicialmente, se asigna un valor arbitrario a cada estado, por ejemplo, $V_0(s) = 0$ para todos los s. Luego, estos valores se actualizan iterativamente utilizando la ecuación de Bellman:

$$V_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a) \left[R(s,a,s') + \gamma V_k(s') \right] \tag{4.5}$$

El proceso comienza con los estados finales, cuyos valores se pueden calcular directamente. Para un estado final s_f, donde no hay más decisiones ni transiciones posibles, el valor $V^\pi(s_f)$ se define como:

$$V^\pi(s_f) = 0 \quad \text{(sin recompensas futuras)} \tag{4.6}$$

o, si hay una recompensa final al alcanzar s_f:

$$V^\pi(s_f) = R_f \tag{4.7}$$

A partir de estos valores finales, se propaga hacia atrás a través del tiempo, actualizando los valores de los estados anteriores en función de los valores ya calculados de los estados futuros. Este proceso se repite hasta que los valores convergen, es decir, cuando $V_{k+1}(s) \approx V_k(s)$ para todos los estados s.

4.1.4. Métodos Clásicos de Aprendizaje por Refuerzo

Existen varios algoritmos clásicos de RL que permiten a los agentes aprender políticas óptimas. Los algoritmos se clasifican principalmente en dos categorías: *on-policy* y *off-policy*. Estos términos se refieren a la relación entre la política que se está evaluando y mejorando y la política que se está utilizando para generar las acciones y, por lo tanto, las experiencias (es decir, las transiciones estado-acción-recompensa) que se utilizan para actualizar el modelo.

Un algorítmo de tipo *on-policy* es aquel en el que la política que se optimiza es la misma que se utiliza para interactuar con el entorno y generar experiencias.Es decir, se evalúa y mejora la misma política que se sigue durante el proceso de aprendizaje. Mientras que un algorítmo del tipo *off-policy* es quel en el que la política que se intenta optimizar es diferente de la política utilizada para generar las experiencias. Esto permite optimizar una política mientras se sigue otra.

Un ejemplo de algoritmos representativos de cada una de estas categorías son los siguientes:

- **Q-learning**: Un algoritmo *off-policy* que aprende la función de valor de acción $Q(s,a)$ actualizando iterativamente las estimaciones en función de las recompensas recibidas:

  $$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1},a') - Q(s_t,a_t) \right]$$

  donde α es la tasa de aprendizaje. Como algoritmo *off-policy*, Q-learning aprende la política óptima independientemente de la política que el agente sigue durante la exploración. Esto significa que la política utilizada para seleccionar acciones (*behavior policy*) puede ser diferente de la política que se está optimizando (*target policy*).

- **SARSA**: Un algoritmo *on-policy* que, a diferencia de Q-learning, sigue la política actual para actualizar la función $Q(s,a)$. La actualización en SARSA se

realiza utilizando la acción que el agente realmente toma, lo que significa que la política utilizada para seleccionar acciones es la misma que se está optimizando:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t) \right]$$

Como resultado, SARSA tiende a ser más conservador en su aprendizaje, ya que la política sigue siendo consistente entre el aprendizaje y la ejecución.

Los algoritmos *on-policy*, como SARSA, tienen la ventaja de ser coherentes con la política actual, lo que puede resultar en un aprendizaje más seguro y estable en entornos dinámicos. Sin embargo, pueden ser menos eficientes, ya que dependen exclusivamente de las experiencias generadas por la misma política que se está optimizando, lo que puede limitar la exploración. Por otro lado, los algoritmos off-policy, como Q-Learning, son más flexibles y permiten un aprendizaje más eficiente al poder optimizar una política óptima mientras se sigue una política diferente, más exploratoria. Sin embargo, esta flexibilidad puede conllevar complejidad adicional y posibles problemas de estabilidad durante el proceso de aprendizaje.

Estos métodos son fundamentales para entender cómo los agentes pueden aprender a tomar decisiones óptimas en entornos inciertos y son la base sobre la cual se construyen los métodos más avanzados como el *Deep Reinforcement Learning* (DRL).

4.2. Evolución del Deep Reinforcement Learning (DRL)

4.2.1. Combinación con Redes Neuronales

El *Deep Reinforcement Learning* (DRL) surge como una evolución natural del *Aprendizaje por Refuerzo* (RL) tradicional, combinando sus principios con la potencia de las redes neuronales profundas (*Deep Learning*). Mientras que los métodos clásicos de RL, como el *Q-learning*, son eficaces en problemas con espacios de estados discretos y de baja dimensionalidad, presentan limitaciones significativas cuando se aplican a entornos complejos y continuos.

El uso de redes neuronales como aproximadores de funciones en RL permitió superar estas limitaciones. Las redes neuronales profundas son capaces de procesar entradas de alta dimensionalidad y de extraer características complejas directamente de los datos sin la necesidad de diseñar manualmente las características relevantes.

4.2.2. Principales Enfoques y Algoritmos de DRL

El desarrollo de DRL ha dado lugar a diversos enfoques que combinan RL y Deep Learning, dando lugar a algoritmos innovadores que han transformado la forma en que se aborda el aprendizaje por refuerzo en entornos complejos. A continuación, se presentan los enfoques principales en DRL, junto con ejemplos representativos de cada uno.

Métodos Basados en Funciones de Valor

Los métodos basados en funciones de valor son aquellos en los que el objetivo es aprender una función que prediga el valor de tomar una acción en un estado

dado. El ejemplo más destacado de este enfoque es el *Deep Q-Network* (DQN). DQN, introducido por *DeepMind*, utiliza redes neuronales profundas para aproximar la función $Q(s, a)$, lo que permitió manejar entornos con espacios de estados grandes y continuos.

En DQN, una red neuronal recibe como entrada el estado del entorno y produce una estimación de los valores $Q(s, a)$ para cada acción posible. La red se entrena utilizando el algoritmo de *Q-learning* tradicional, pero con varias modificaciones clave para estabilizar el aprendizaje, como el uso de un *buffer* de *replay* para almacenar transiciones de experiencias pasadas y el uso de una red objetivo (*target network*) que se actualiza periódicamente para reducir la inestabilidad en la actualización de los valores Q.

Métodos de Gradiente de Política

Los métodos de gradiente de política representan un enfoque alternativo en DRL, en el cual, en lugar de aprender una función de valor, el agente aprende directamente una política parametrizada $\pi_\theta(a|s)$ que maximiza la recompensa esperada. La principal ventaja de estos métodos es su capacidad para manejar espacios de acción continuos y para aprender políticas estocásticas.

Un algoritmo básico dentro de esta categoría es *REINFORCE*, que utiliza el gradiente de la recompensa esperada para actualizar los parámetros de la política. Aunque REINFORCE es sencillo y conceptualmente atractivo, su principal limitación es la alta varianza en las estimaciones del gradiente, lo que puede llevar a un aprendizaje lento y poco estable.

Métodos Actor-Critic

Los métodos *Actor-Critic* combinan lo mejor de ambos mundos al utilizar tanto una función de valor como una política parametrizada. En este enfoque, el *actor* es responsable de seleccionar las acciones basadas en la política $\pi_\theta(a|s)$, mientras que el *crítico* estima el valor de la política actual utilizando una función de valor $V^\pi(s)$ o $Q^\pi(s, a)$.

Un ejemplo notable es el *Asynchronous Advantage Actor-Critic* (A3C), que introduce el concepto de ventaja (*advantage*), definida como la diferencia entre el valor de la acción y el valor esperado del estado ($A(s, a) = Q(s, a) - V(s)$). A3C mejora la estabilidad del aprendizaje al usar múltiples actores que interactúan con copias del entorno en paralelo, lo que también mejora la eficiencia computacional.

4.3. Proximal Policy Optimization (PPO) y su Implementación

4.3.1. Introducción a Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO) es un algoritmo de *Deep Reinforcement Learning* (DRL) desarrollado por *OpenAI*, que ha demostrado ser exitoso en una amplia variedad de tareas, desde el control robótico hasta videojuegos complejos como *Dota 2*. PPO es un método de gradiente de política (*Policy Gradient*), lo que significa que aprende directamente de la experiencia en tiempo real generada por el agente en

su interacción con el entorno, a diferencia de los métodos basados en *Q-learning*, que pueden aprender de datos almacenados.

PPO fue diseñado para encontrar un equilibrio entre la eficiencia de la muestra, la facilidad de implementación y la estabilidad en el entrenamiento. Esto lo convierte en una herramienta potente y versátil para abordar problemas de aprendizaje por refuerzo en entornos continuos y de alta dimensionalidad.

En este trabajo, PPO se utilizará para la generación de estrategias de trading, aprovechando su capacidad para manejar entornos complejos y dinámicos como los mercados financieros. Este algoritmo ha sido seleccionado debido a su robustez y simplicidad en la implementación, características que lo hacen ideal para desarrollar una estrategia de trading que pueda adaptarse a las condiciones cambiantes del mercado.

PPO fue presentado por primera vez en el paper de *Schulman et al.* titulado "*Proximal Policy Optimization Algorithms*"[7], que ha sido ampliamente citado en la literatura de aprendizaje por refuerzo debido a su enfoque innovador y eficaz.

4.3.2. Descripción Técnica de PPO

PPO introduce varias mejoras clave en la optimización de políticas para garantizar un aprendizaje más estable. A diferencia de otros métodos que pueden permitir grandes cambios en la política, PPO limita estos cambios para evitar desestabilizar el proceso de entrenamiento.

Problemas en el Aprendizaje por Refuerzo

A diferencia del aprendizaje supervisado, donde se trabaja con un conjunto de datos estático, en el aprendizaje por refuerzo el conjunto de datos cambia constantemente porque el agente genera sus propios datos a medida que interactúa con el entorno. Esto provoca que las distribuciones de las observaciones y las recompensas varíen durante el entrenamiento, lo que puede causar inestabilidad.

Además, el aprendizaje por refuerzo es altamente sensible a la configuración de hiperparámetros y la inicialización. Por ejemplo, una tasa de aprendizaje demasiado alta puede llevar a que las actualizaciones de la política desvíen al agente hacia regiones del espacio de parámetros donde su desempeño se deteriora drásticamente, sin posibilidad de recuperación.

Función de Objetivo en PPO

PPO aborda estos problemas proponiendo una función de objetivo que maximiza la recompensa esperada mientras limita las actualizaciones de la política para evitar cambios excesivos. La función de objetivo en PPO se expresa como:

$$L^{PPO}(\theta) = \mathbb{E}_t \left[\text{mín} \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right]$$

Aquí, $r_t(\theta)$ es la razón de probabilidad entre la nueva política $\pi_\theta(a|s)$ y la política antigua $\pi_{\theta_{\text{old}}}(a|s)$:

$$r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$$

El término $\hat{A}_t$ es la estimación de la ventaja, que mide cuánto mejor o peor fue la acción tomada en comparación con lo esperado. El operador *clip* limita el valor de $r_t(\theta)$ dentro del rango $[1-\epsilon, 1+\epsilon]$, con ϵ típicamente igual a 0.2, para prevenir que las actualizaciones de la política sean demasiado agresivas, manteniendo así la estabilidad del entrenamiento.

Ventaja y Estimación del Valor

Para calcular la ventaja $\hat{A}_t$, PPO utiliza dos componentes:

- **Retorno descontado**: Es la suma de las recompensas acumuladas, descontadas por un factor γ que valora más las recompensas inmediatas.
- **Función de valor**: Estima el valor esperado de estar en un estado dado, calculando la recompensa total esperada desde ese estado en adelante.

La ventaja se calcula restando la función de valor del retorno descontado, proporcionando una medida de si la acción tomada fue mejor o peor que el promedio esperado.

Procedimiento de Entrenamiento

PPO recolecta nuevas trayectorias de interacciones con el entorno en cada iteración, utilizando estas trayectorias para actualizar la política. A diferencia de otros métodos, no utiliza un *replay buffer*, lo que significa que las experiencias se utilizan una sola vez para hacer la actualización, y luego se descartan.

Este proceso iterativo continúa hasta que las mejoras en la política se estabilizan, lo que indica que el algoritmo ha convergido. PPO equilibra la simplicidad y la robustez, permitiendo un entrenamiento estable incluso en entornos complejos y dinámicos.

4.4. Implementación del Entorno de Entrenamiento con Gymnasium

4.4.1. Introducción a Gymnasium

Gymnasium es una biblioteca de Python ampliamente utilizada en el campo del *Reinforcement Learning* (RL) para crear, gestionar y probar entornos de simulación. Esta biblioteca es una evolución de OpenAI Gym, diseñada para facilitar la construcción de entornos personalizados y la evaluación de algoritmos de RL en una variedad de dominios, desde juegos sencillos hasta tareas más complejas como la robótica y el trading financiero.

La popularidad de Gymnasium se debe a su simplicidad y flexibilidad. Proporciona una interfaz estándar que permite a los desarrolladores de algoritmos de RL interactuar fácilmente con los entornos. Esta estandarización es clave para comparar el rendimiento de diferentes algoritmos en condiciones controladas.

4.4.2. Creación del Entorno de Trading

Aunque existen entornos predefinidos en *Gymnasium* para el entrenamiento de algoritmos de trading, como *Gym Trading Env* o *FinRL*, la mayoría de estos están diseñados para operar con múltiples activos financieros simultáneamente o introducen variables adicionales como la compra a crédito de activos.

Para este trabajo, se ha optado por un enfoque más sencillo y controlado, desarrollando un entorno personalizado donde el agente opera únicamente con un solo activo en cada sesión de trading. El entorno se define en un espacio de acciones discreto $\mathcal{A} = \{-1, 0, 1\}$, donde:

- -1: Vender, lo que implica cerrar completamente la posición actual.
- 0: Mantener (Hold), es decir, no realizar ninguna acción.
- 1: Comprar, utilizando toda la liquidez disponible en ese momento.

Este diseño simplificado asegura que cada vez que el agente decide comprar, lo hace utilizando la totalidad de los fondos disponibles, y cuando decide vender, liquida completamente la posición. De esta manera, el agente no gestiona el *position sizing*, ni considera la operación con posiciones negativas. Asimismo, no se tienen en cuenta otras variables como los costes de transacción, el *slippage*, o la liquidez disponible.

Desarrollar un entorno desde cero no solo facilita una mejor comprensión del problema, sino que también permite personalizar los aspectos del entorno para que se ajusten mejor a los requisitos del algoritmo y los objetivos específicos de este trabajo.

Para implementar el entorno de trading en *Gymnasium*, se han desarrollado los siguientes scripts:

Scripts del Entorno de Trading

- `history.py`: Este script se encarga de guardar la historia del agente durante el episodio de entrenamiento. Registra información clave como el índice de la posición actual, el paso temporal, la fecha, la posición actual y real del portafolio, los datos asociados al estado actual, la valoración del portafolio, la distribución del portafolio y la recompensa obtenida en cada paso. Esta información se utiliza para analizar y evaluar el comportamiento del agente a lo largo del tiempo, pero no maneja los datos de la serie temporal en sí.

- `reward_functions.py`: Aquí se definen varias funciones de recompensa que pueden utilizarse para evaluar el rendimiento del agente. Para los resultados de este trabajo, se ha utilizado una función de recompensa básica que calcula el retorno logarítmico del portafolio en cada paso temporal (*step*). Esta elección se debe a que los retornos logarítmicos pueden sumarse a lo largo del tiempo, a diferencia de los retornos simples, lo que facilita el análisis acumulativo del rendimiento del agente.

- `simple_portfolio.py`: Se define el portfolio que maneja la simulación del portafolio del agente. Se encarga de actualizar el estado del portafolio según las acciones de compra o venta tomadas por el agente, teniendo en cuenta los costos de transacción. En cada operación de compra o venta, el agente compra o vende todo lo disponible, lo que simplifica la gestión del portafolio. Además, el script

calcula el valor total del portafolio en cada paso del tiempo, proporcionando una representación precisa del rendimiento del agente en el entorno de trading.

- `trading_env.py`: Este es el núcleo del entorno de trading, integrando todos los componentes anteriores. Define el entorno siguiendo la estructura estándar de Gymnasium, implementando los métodos necesarios como `reset()`, `step()`, y `render()`:

  - `reset()`: Este método reinicia el entorno al estado inicial, devolviendo la primera observación. Se llama al inicio de cada episodio de entrenamiento para preparar al agente para un nuevo intento de maximizar su recompensa.
  - `step(action)`: Este método aplica la acción seleccionada por el agente y actualiza el estado del entorno. Devuelve cinco elementos clave:
    1. `observation`: La nueva observación que refleja el estado actualizado del entorno.
    2. `reward`: El retorno de la función de recompensa.
    3. `done`: Un valor booleano que indica si el episodio ha terminado.
    4. `truncated`: Un valor booleano que indica si hubo problemas en el episodio.
    5. `info`: Información extra, como la valoración del portafolio, que puede ser útil para análisis adicionales.
  - `render()`: Aunque es un método necesario en la estructura de un entorno Gymnasium, en esta implementación no se utiliza. Este método opcional se emplea generalmente para visualizar el entorno, permitiendo la representación gráfica del desempeño del agente, pero en este caso, no se ha requerido su uso.

Definición del espacio de observaciones

El espacio de observaciones del entorno de trading se ha construido utilizando una combinación de series temporales originales, indicadores técnicos calculados sobre estas series, y variables dinámicas adicionales que reflejan el estado y las acciones previas del agente.

A continuación, se detalla el proceso seguido para definir este espacio:

En primer lugar, se han utilizado las series originales de precios del SPY y del BTC, así como las series de desequilibrio de volumen y de dólares obtenidas en 3.1.3.

A partir de estas series, se han calculado diversos indicadores técnicos que proporcionan información clave sobre la tendencia, la volatilidad y el impulso del mercado. Los indicadores utilizados son:

- **Medias Móviles (MA 20 y MA 50)**: Se calculan medias móviles simples de 20 y 50 periodos, que suavizan las fluctuaciones de los precios y ayudan a identificar la dirección de la tendencia. La MA 20 se considera una media de corto plazo, mientras que la MA 50 es una media de medio plazo.

- **Media Móvil Exponencial (EMA)**: A diferencia de las medias móviles simples, la EMA otorga mayor peso a los precios recientes, lo que permite reaccionar más rápidamente a los cambios en la dirección del mercado.

- **Bandas de Bollinger**: Este indicador consiste en una media móvil rodeada por dos bandas a una distancia de dos desviaciones estándar. Las Bandas de Bollinger ayudan a identificar periodos de alta y baja volatilidad, así como posibles puntos de reversión de la tendencia.

- **MACD (Moving Average Convergence Divergence)**: El MACD es un oscilador que muestra la relación entre dos medias móviles (usualmente 12 y 26 periodos). Es útil para identificar el impulso del mercado y posibles señales de compra o venta cuando la línea MACD cruza la línea de señal.

- **RSI (Relative Strength Index)**: El RSI es un indicador de momentum que mide la velocidad y el cambio de los movimientos de precios. Se mueve en un rango de 0 a 100, y valores extremos (por encima de 70 o por debajo de 30) indican que un activo puede estar sobrecomprado o sobrevendido.

- **ATR (Average True Range)**: El ATR mide la volatilidad del mercado calculando el rango medio verdadero durante un periodo de tiempo. Es útil para evaluar el riesgo y establecer niveles de stop-loss.

- **CCI (Commodity Channel Index)**: El CCI mide la variación del precio de un activo en relación con su media estadística. Valores altos o bajos extremos indican condiciones de sobrecompra o sobreventa.

Estos indicadores, junto con los datos originales de precios (OHLC) se han normalizado para garantizar que todas las características tengan una escala comparable, lo que facilita el entrenamiento del agente y mejora la estabilidad del aprendizaje.

Además, se han añadido indicadores dinámicos que proporcionan información adicional sobre el estado actual y las acciones previas del agente. Estos indicadores son:

- **Última Acción Tomada (Last Action Taken)**: Refleja la última acción ejecutada por el agente en el paso temporal anterior.

- **Acción Real Tomada (Real Action Taken)**: Muestra la acción real que se llevó a cabo, teniendo en cuenta las posibles restricciones o ajustes automáticos realizados por el entorno.

- **Exposición (Portfolio Exposition)**: Indica el porcentaje de exposición que posee el portfolio sobre el activo que se opera. Es la relación entre el valor del activo y el valor total del portfolio.

Finalmente, para capturar la dinámica temporal de estos datos, se define una ventana de observación de 10 periodos. Esto significa que en cada paso temporal, el agente recibe como observación una matriz que incluye los valores normalizados de los indicadores técnicos, los datos de precios (OHLC) y los indicadores dinámicos correspondientes a los últimos 10 periodos.

Esta configuración proporciona al agente una visión comprensiva del estado actual del mercado y su propio comportamiento reciente, lo que le permite tomar decisiones más informadas.

4.4.3. Función de Recompensa

En los sistemas de aprendizaje por refuerzo, la función de recompensa es un componente fundamental que guía el aprendizaje del agente. Esta función es responsable de proporcionar señales al agente sobre la calidad de las acciones que toma en un entorno dado, permitiéndole ajustar su comportamiento para maximizar la acumulación de recompensas a lo largo del tiempo.

Una función de recompensa bien diseñada es crucial para el éxito de un agente de aprendizaje por refuerzo, ya que define el objetivo que el agente debe perseguir y, por lo tanto, tiene un impacto directo en el comportamiento que el agente desarrollará. Si la función de recompensa no refleja adecuadamente los objetivos deseados, el agente podría aprender estrategias subóptimas que no maximicen los beneficios o que incrementen el riesgo de manera innecesaria.

En este caso, se ha implementado la función de recompensa propuesta por Dantas y Silva [3], que es especialmente adecuada para el trading de acciones en mercados financieros. Esta función se basa en un retorno diferencial diario, definido como:

$$R_t = \alpha \left(\frac{P_t^p - P_{t-1}^p}{P_{t-1}^p} \right) - \left(\frac{P_t^s - P_{t-1}^s}{P_{t-1}^s} \right)$$

donde:

- P_t^p es el valor de la cartera en el tiempo t,
- P_t^s es el precio de la acción en el tiempo t,
- α es una constante que ajusta la penalización o recompensa adicional. En este caso, se ha utilizado un valor de $\alpha = 1{,}5$, el mismo utilizado en el paper original.

La elección de esta función de recompensa se debe a que, previamente, se probaron otras funciones basadas en el retorno inmediato o en el ratio Sharpe. Sin embargo, se observó que el agente terminaba por no operar con estas funciones, ya que buscaba evitar recompensas negativas.

Este comportamiento destaca la importancia de una función de recompensa adecuada, ya que una mala elección puede llevar al agente a desarrollar estrategias que no cumplen con los objetivos del sistema.

La función de recompensa de Dantas y Silva incentiva al agente a tomar decisiones activas, incluso en condiciones de mercado adversas. Por ejemplo, si el valor de la cartera (P_t^p) no cambia pero el precio de la acción (P_t^s) disminuye, el agente recibe una recompensa positiva por evitar una pérdida potencial. Inversamente, si el precio de la acción sube pero el agente no ha comprado, se le penaliza, lo que fomenta una mayor proactividad en sus decisiones de trading.

Este diseño permite al agente aprender a equilibrar entre maximizar retornos y minimizar riesgos.

4.5. Entrenamiento y resultados

El entrenamiento del agente PPO se ha realizado con los parámetros presentados en la Tabla 4.1 sobre las series temporales de los activos SPY y BTC obtenidos en la sección 3.1.3.

DRL Agent	PPO
Learning Rate	1e-3
Gamma	0.99
Number of Epochs	100
Device	cuda

Tabla 4.1: Parámetros de entrenamiento del agente DRL.

El objetivo es comparar el rendimiento de las estrategias obtenidas con el agente PPO al operar sobre los datos originales (barras de tiempo) y los datos de barras de desequilibrio de volúmen y dólares.

En cada caso, el agente se ha entrenado durante 1,500,000 pasos, siendo la longitud máxima del entorno el 20 % de la serie temporal total, lo que permite que el agente pueda operar distintos periodos de la serie. Además, el portafolio inicial se establece en $100,000, lo que proporciona al agente suficiente capital para explorar y aprender sobre las decisiones de compra y venta en el entorno simulado.

Los resultados obtenidos sobre el conjunto de validación out-of-sample se presentan en las Tablas 4.2 y 4.3, así como en las Figuras 4.2 y 4.3.

En términos de rendimiento acumulado y retorno anualizado, las estrategias basadas en PPO no logran superar a los benchmarks, tanto en el caso del SPY como en el del BTC. En particular, el *buy and hold* del BTC muestra una ventaja significativa en cuanto a rendimiento bruto, lo que refleja la alta volatilidad y el potencial de retorno del mercado de criptomonedas.

Si bien las estrategias basadas en PPO presentan un drawdown más bajo y mejores ratios de Sharpe y Sortino en comparación con los benchmarks, estas características no suponen ninguna ventaja real debido a la evidente peor rentabilidad que ofrecen. Esto sugiere que, a pesar de manejar mejor el riesgo relativo, las estrategias basadas en PPO no son competitivas en términos de rendimiento total.

Al comparar las diferentes configuraciones de barras, la estrategia basada en barras de desequilibrio (Dollar Bars) destaca por mostrar un Sharpe Ratio y un Sortino Ratio superiores a los obtenidos con las barras originales y las barras de volumen. Sin embargo, esta mejora en la gestión del riesgo no compensa la inferioridad en términos de rentabilidad absoluta en comparación con los benchmarks.

En resumen, las estrategias de PPO, aunque presentan un menor drawdown y mejores ratios de Sharpe en ciertas configuraciones, no ofrecen una ventaja competitiva frente a los benchmarks en términos de rentabilidad global. Estos hallazgos sugieren que, para maximizar el rendimiento, podría ser necesario explorar otras estrategias o combinaciones que permitan mejorar los resultados obtenidos.

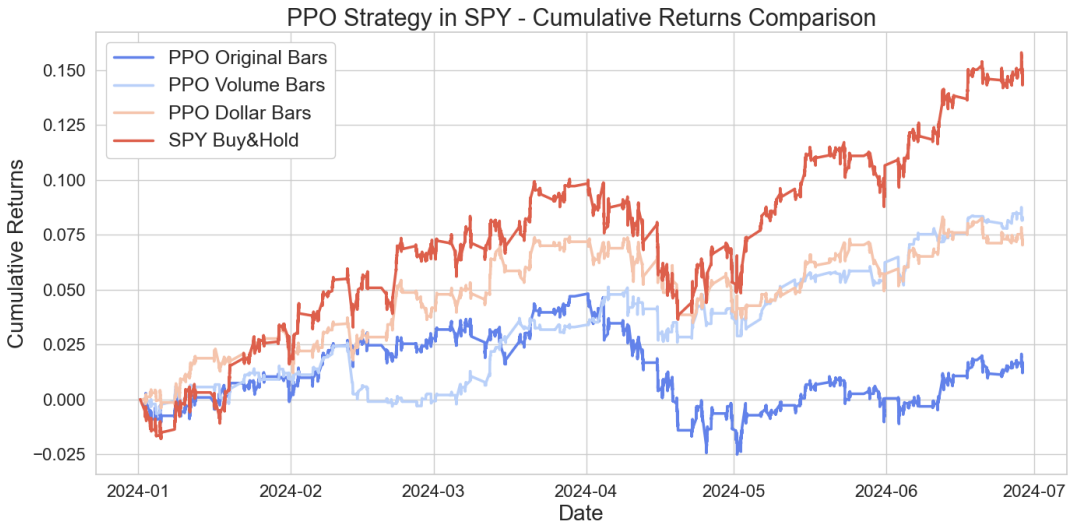

Figura 4.2: Comparativa de retornos acumulados de la estrategia PPO sobre SPY

	Cumulative Returns	Annualized Return	Sharpe Ratio	Sortino Ratio	Maximum Drawdown	Calmar Ratio	Value at Risk (VaR)	Alpha	Beta
Benchmark (S&P 500)	**0.146**	**0.211**	0.130	0.186	-0.058	3.626	-0.001	0.000	1.000
PPO Original Bars	0.015	0.021	0.019	0.028	0.070	0.295	-0.000	-0.000	0.632
PPO Volume Bars	0.082	0.118	**0.473**	0.655	-0.028	4.168	-0.001	-0.002	0.356
PPO Dollar Bars	0.072	0.103	0.219	0.310	-0.037	2.812	-0.001	-0.000	0.454

Tabla 4.2: Resultados de los experimentos de PPO en diferentes tipos de barras comparados con el Benchmark (SPY).

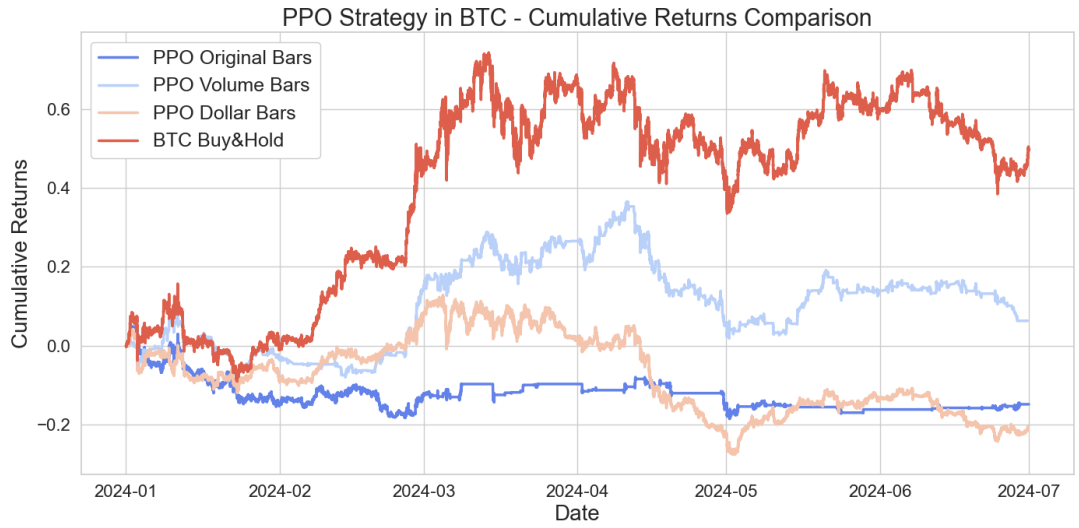

Figura 4.3: Comparativa de retornos acumulados de la estrategia PPO sobre BTC

	Cumulative Returns	Annualized Return	Sharpe Ratio	Sortino Ratio	Maximum Drawdown	Calmar Ratio	Value at Risk (VaR)	Alpha	Beta
Benchmark (BTC)	**0.497**	**0.753**	0.034	0.048	-0.234	3.216	-0.001	0.000	1.000
PPO Original Bars	-0.148	-0.200	-0.020	-0.028	-0.226	-0.888	-0.001	-0.000	0.209
PPO Volume Bars	0.063	0.089	**0.054**	0.075	-0.254	0.351	-0.004	-0.002	0.415
PPO Dollar Bars	-0.213	-0.284	-0.025	-0.034	-0.358	-0.792	-0.001273	-0.001	0.484

Tabla 4.3: Resultados de los experimentos de PPO en diferentes tipos de barras comparados con el Benchmark (SPY).

Capítulo 5

Meta-labeling

5.1. Concepto y Definición de Meta-Labeling

El *Meta-Labeling* es una técnica avanzada que ha demostrado ser particularmente útil en la creación de estrategias de trading robustas, especialmente en el contexto del aprendizaje automático y la gestión de riesgos.

Este método consiste en la superposición de una capa adicional de análisis sobre una decisión de trading ya existente, evaluando la probabilidad de éxito de las señales generadas por un modelo base. En lugar de limitarse a una sola predicción binaria (por ejemplo, comprar o no comprar un activo), el Meta- Labeling aplica una segunda etapa de modelado para determinar si la señal inicial tiene una alta probabilidad de ser correcta. Este enfoque permite refinar la decisión final, mitigando riesgos y mejorando el rendimiento general del modelo de trading.

El proceso típico de Meta-Labeling involucra dos modelos:

1. **Modelo Primario ($\mathcal{M}_1$)**: Este modelo genera la señal de trading.

2. **Meta-Modelo ($\mathcal{M}_2$)**: El meta-modelo se entrena para predecir la fiabilidad de las señales generadas por el modelo primario. A cada señal generada por $\mathcal{M}_1$

se le asigna un *meta-label*, que indica si dicha señal resultó en una operación exitosa o fallida.

Formalmente, para una observación i, el modelo primario $\mathcal{M}_1$ genera una predicción $\hat{y}_i \in \{-1, 0, 1\}$. El meta-label $z_i \in \{0, 1\}$ se define de la siguiente manera:

$$z_i = \begin{cases} 1 & \text{si la operación resultante de la señal } \hat{y}_i \text{ fue exitosa,} \\ 0 & \text{si la operación resultante de la señal } \hat{y}_i \text{ falló.} \end{cases}$$

Aquí, $z_i = 1$ indica que la señal emitida por $\mathcal{M}_1$ fue confiable y resultó en una operación exitosa (por ejemplo, una ganancia), mientras que $z_i = 0$ indica que la señal no fue confiable y resultó en una operación fallida (por ejemplo, una pérdida).

El meta-modelo $\mathcal{M}_2$ se entrena utilizando las mismas características $\mathbf{x}_i$ que alimentan al modelo primario, así como la señal generada por $\mathcal{M}_1$, y otros indicadores adicionales que no fueron utilizados en el modelo primario. Estos indicadores adicionales pueden incluir variables exógenas o nuevas características que ayuden a mejorar la capacidad del meta-modelo para evaluar la fiabilidad de las señales originales.

Esta estructura de dos niveles ofrece una ventaja significativa, ya que permite refinar las decisiones de trading al evaluar de manera más precisa la probabilidad de éxito de las señales iniciales. El Meta-Labeling, por tanto, no solo añade una capa de seguridad al proceso de toma de decisiones, sino que también mejora la precisión y el rendimiento general de la estrategia de trading.

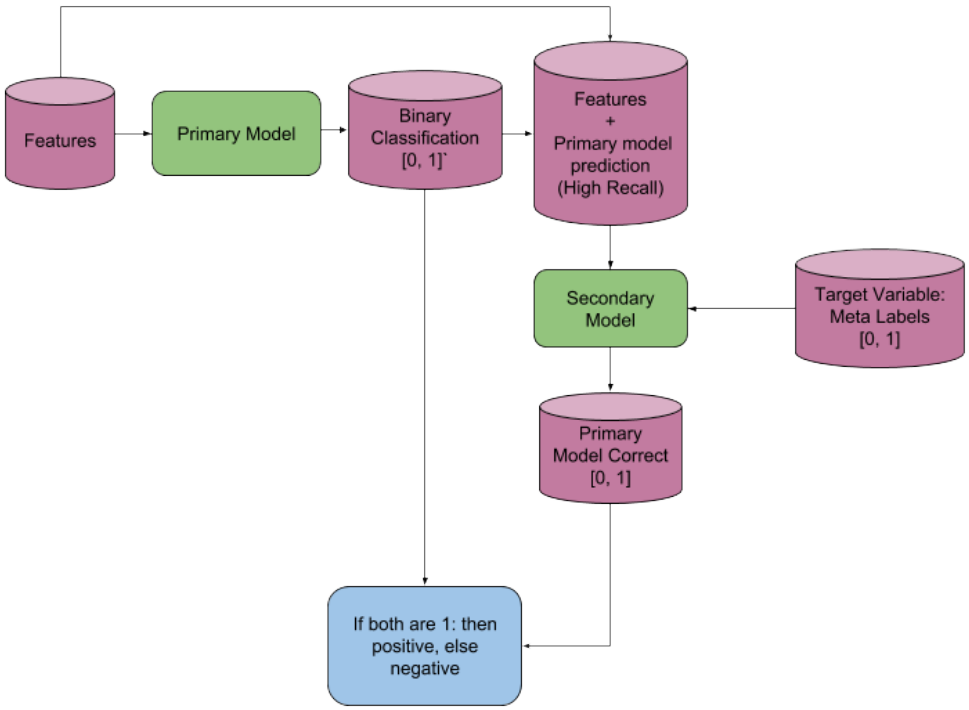

Figura 5.1: Diagrama Meta-Labeling (Adaptación de *Hudson and Thames*)

Uno de los principales beneficios del Meta-Labeling es su capacidad para mejorar el rendimiento de un predictor que, por sí solo, puede no ser suficientemente robusto [1]. Supongamos que $\mathcal{M}_1$ es un modelo primario con una precisión moderada y un

recall bajo, lo que significa que emite muchas señales incorrectas (falsos positivos). Al aplicar Meta-Labeling, se puede ajustar el *recall* y la precisión, ya que $\mathcal{M}_2$ filtra las señales generadas por $\mathcal{M}_1$, reduciendo el número de falsos positivos.

El rendimiento de un modelo de clasificación se evalúa comúnmente mediante métricas como la precisión y el *recall*. Para un modelo primario $\mathcal{M}_1$, estas métricas se definen como:

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}},$$

donde TP, FP y FN son verdaderos positivos, falsos positivos y falsos negativos, respectivamente. Si $\mathcal{M}_1$ genera muchas señales falsas (altos FP), la precisión disminuye, lo que puede llevar a decisiones de trading no óptimas.

El Meta-Labeling se propone como una solución a este problema. Al entrenar $\mathcal{M}_2$ para predecir la probabilidad de éxito de las señales de $\mathcal{M}_1$, es posible aumentar la precisión del sistema general. Esto se debe a que $\mathcal{M}_2$ actúa como un filtro adicional que reduce los FP, manteniendo o incluso mejorando el *recall*:

$$\text{Precisión}_{\text{Meta}} = \frac{\text{TP}_{\text{Meta}}}{\text{TP}_{\text{Meta}} + \text{FP}_{\text{Meta}}}, \quad \text{Recall}_{\text{Meta}} = \frac{\text{TP}_{\text{Meta}}}{\text{TP}_{\text{Meta}} + \text{FN}_{\text{Meta}}},$$

donde TP_{Meta}, FP_{Meta}, y FN_{Meta} son los valores ajustados después de aplicar el Meta-Labeling.

5.1.1. Optimización del Ratio de Sharpe mediante Meta-Labeling

Uno de los aspectos más destacados del Meta-Labeling es su capacidad para optimizar el ratio de Sharpe de una estrategia de trading, mejorando el rendimiento ajustado al riesgo.

En su forma más general, el ratio de Sharpe se define como la razón entre el exceso de retorno de una inversión sobre la tasa libre de riesgo y la desviación estándar de esos retornos. Sin embargo, en el contexto de resultados binarios, como en las estrategias de trading donde las decisiones se reducen a "ganar.º "perder", el ratio de Sharpe puede adaptarse utilizando la siguiente fórmula, tal como se describe en *Advances in Financial Machine Learning* de Marcos López de Prado:

$$\theta(p, n, \pi^-, \pi^+) = \frac{(\pi^+ - \pi^-)p + \pi^-}{(\pi^+ - \pi^-)\sqrt{p \cdot (1-p)}} \cdot \sqrt{n}$$

Donde:

- π^+ es el retorno asociado a un resultado positivo.
- π^- es el retorno asociado a un resultado negativo.
- p es la probabilidad de obtener un resultado positivo, es decir, la probabilidad de que una operación sea exitosa.
- n es el número de resultados o decisiones (como el número de operaciones realizadas en un año).

Esta fórmula captura el rendimiento ajustado al riesgo en situaciones donde las decisiones de trading producen resultados binarios. La diferencia $\pi^+ - \pi^-$ representa el diferencial entre el retorno positivo y el negativo, que se ajusta por la volatilidad de los resultados, medida como $\sqrt{p \cdot (1-p)}$. Esto refleja la incertidumbre inherente al sistema, basada en la probabilidad de un resultado positivo y la frecuencia de las operaciones. Un valor alto de θ indica que la estrategia tiene un buen rendimiento relativo al riesgo asumido.

5.1.2. Impacto del Meta-Labeling en el Ratio de Sharpe

El Meta-Labeling puede jugar un papel crucial en la optimización de θ. Cuando las ganancias (π^+) son significativamente menores que las pérdidas ($\pi^- \ll -\pi^+$), el ratio de Sharpe tiende a ser bajo, lo que refleja una estrategia con un rendimiento ajustado al riesgo desfavorable. En este caso, una forma efectiva de mejorar θ es aumentando la probabilidad de éxito p, aunque a costa de reducir el número de operaciones n. Aquí es donde entra en juego el Meta-Labeling.

- **Modelo Primario ($\mathcal{M}_1$)**: El modelo primario es responsable de identificar las señales iniciales de trading, determinando los posibles retornos π^- y π^+. Sin embargo, este modelo podría generar un número significativo de falsos positivos, lo que reduciría p y, en consecuencia, el ratio de Sharpe.
- **Meta-Modelo ($\mathcal{M}_2$)**: El modelo secundario se encarga de ajustar el umbral de aceptación para filtrar las señales generadas por el modelo primario. Este modelo tiene la capacidad de incrementar p seleccionando solo las señales con mayor probabilidad de éxito, lo que reduce el número total de operaciones n, pero mejora la calidad de las mismas.

Por lo tanto, el Meta-Labeling puede mejorar θ de las siguientes maneras:

- **Mejora de la Probabilidad de Éxito (p)**: Al aplicar Meta-Labeling, se ajusta el umbral de aceptación de las señales generadas por el modelo primario. Al filtrar las señales para seleccionar solo aquellas con una mayor probabilidad de éxito, el Meta-Labeling aumenta p, mejorando así el retorno ponderado promedio en el numerador de la fórmula.
- **Reducción del Riesgo ($\sqrt{p \cdot (1-p)}$)**: Al incrementar p, se reduce la variabilidad de los resultados, lo que disminuye el riesgo relativo (representado por el denominador de la fórmula). Esto significa que las operaciones seleccionadas son menos propensas a ser erráticas, lo que estabiliza el rendimiento de la estrategia.
- **Ajuste del Número de Operaciones (n)**: Aunque el número de operaciones puede disminuir (menor n), las que se realizan tienen una mayor calidad y probabilidad de éxito. Como n aparece fuera de la raíz cuadrada en la fórmula, un menor número de operaciones de alta calidad puede tener un impacto significativo en la mejora del ratio de Sharpe.

Por ejemplo, si una estrategia inicialmente genera muchas señales de trading, pero con un alto riesgo de pérdidas, el Meta-Labeling puede reducir estas señales

seleccionando solo aquellas con alta probabilidad de éxito, lo que resulta en menos operaciones, pero más seguras. Esto no solo mejora la precisión y el *recall* de la estrategia, sino que también optimiza el ratio de Sharpe al evitar grandes pérdidas y mejorar el rendimiento ajustado al riesgo.

En resumen, el Meta-Labeling permite que una estrategia de trading mejore tanto en términos de rendimiento como de estabilidad, maximizando las ganancias relativas al riesgo asumido. Esta técnica es especialmente valiosa en contextos donde es crucial evitar grandes pérdidas y maximizar la probabilidad de éxito de las operaciones [6].

Beneficios del F1-Score y del Control de Tamaño de las Apuestas

El Meta-Labeling es particularmente útil para lograr mayores *F1-scores*. Primero, se construye un modelo que logre un alto *recall*, incluso si la precisión no es particularmente alta. Luego, se corrige la baja precisión aplicando Meta-Labeling a los positivos identificados por el modelo primario. Esto permite obtener un *F1-score* más equilibrado y un modelo más robusto en la identificación de oportunidades de trading.

Además, el Meta-Labeling permite desarrollar un sistema de ML que, en lugar de ser una caja negra, actúa como una caja blanca, en la que se puede entender y ajustar la lógica detrás de las decisiones tomadas. Esto es crucial para limitar los efectos del sobreajuste, ya que el ML no decide el lado de la apuesta, sino solo el tamaño de la misma. En este sentido, el Meta-Labeling permite gestionar adecuadamente el tamaño de las apuestas, evitando situaciones en las que se logre alta precisión en pequeñas apuestas, pero baja precisión en apuestas grandes, lo que podría llevar a la ruina [6].

5.2. Etiquetado de Datos Financieros

Para aplicar Meta-Labeling de manera efectiva, es imprescindible etiquetar correctamente los datos financieros. El etiquetado de datos consiste en asignar una etiqueta a cada señal de trading que indique si la operación resultante fue exitosa o fallida.

Este proceso es crucial, ya que el etiquetado correcto de las señales es lo que permitirá al meta-modelo aprender y predecir la fiabilidad de futuras señales emitidas por el modelo primario.

5.2.1. Método del Horizonte Temporal Fijo

El *método del horizonte temporal fijo* es una técnica ampliamente utilizada en la literatura financiera para etiquetar datos. Este método asigna etiquetas a las observaciones basándose en el retorno del activo durante un período de tiempo predefinido después de que se genera una señal de trading.

Descripción del Método

Consideremos:

- Una matriz de características $\mathbf{X}$ con I filas, donde cada fila representa una observación $\mathbf{X}_i$ correspondiente a un instante temporal t.

- Un umbral fijo τ que define la magnitud mínima del retorno necesaria para considerar una operación como exitosa o fallida.
- Un horizonte temporal fijo h, que representa el número de barras de tiempo que se observan después de la generación de la señal.

El procedimiento de etiquetado se define de la siguiente manera:

1. **Generación de la Señal**: En el tiempo $t_{i,0}$, inmediatamente después de observar $\mathbf{X}_i$, se genera una señal de trading.
2. **Cálculo del Retorno**: Se calcula el retorno del precio del activo desde $t_{i,0}$ hasta $t_{i,0} + h$ mediante la siguiente expresión:

$$r_{t_{i,0},t_{i,0}+h} = \frac{P_{t_{i,0}+h}}{P_{t_{i,0}}} - 1, \tag{5.1}$$

   donde P_t representa el precio del activo en el tiempo t.
3. **Asignación de Etiquetas**: La etiqueta y_i se asigna según el valor de $r_{t_{i,0},t_{i,0}+h}$ comparado con el umbral τ:

$$y_i = \begin{cases} 1 & \text{si } r_{t_{i,0},t_{i,0}+h} > \tau, \\ -1 & \text{si } r_{t_{i,0},t_{i,0}+h} < -\tau, \\ 0 & \text{si } |r_{t_{i,0},t_{i,0}+h}| \leq \tau. \end{cases} \tag{5.2}$$

A pesar de su simplicidad y amplia adopción, este método presenta varias limitaciones importantes:

1. **Ignorancia de la Volatilidad Variable**: El uso de un umbral fijo τ no considera que la volatilidad del mercado puede variar significativamente en diferentes momentos. Por ejemplo, durante períodos de alta volatilidad, un umbral τ puede ser demasiado bajo, etiquetando muchas observaciones como positivas o negativas de manera inapropiada. Contrariamente, durante períodos de baja volatilidad, el mismo umbral puede ser demasiado alto, resultando en la mayoría de las observaciones etiquetadas como neutrales.
2. **Desconsideración de la Ruta de Precios**: El método solo considera el precio inicial y final dentro del horizonte temporal, ignorando el camino seguido por el precio entre estos puntos.Esto es problemático porque en la práctica, las estrategias de trading están sujetas a niveles de *stop-loss* y *take-profit* que pueden activarse antes de que finalice el horizonte temporal. Ignorar estos eventos intermedios puede llevar a etiquetados engañosos que no reflejan las condiciones reales de mercado y gestión de riesgo.
3. **Dependencia de Barras de Tiempo**: El método se basa en barras de tiempo fijas, que pueden no capturar adecuadamente la actividad del mercado, especialmente en activos con volúmenes de negociación variables. Esto puede resultar en una representación inconsistente de la información del mercado y afectar negativamente la calidad del etiquetado.

Estas limitaciones pueden conducir a un etiquetado inexacto y, por lo tanto, afectar la capacidad del modelo para aprender y predecir efectivamente.

5.2.2. Método de las Tres Barreras

El *Método de las Tres Barreras* (*Three-Barrier Method*) es una técnica alternativa de etiquetado que proporciona una mayor precisión que el método del horizonte temporal fijo, especialmente en entornos de trading donde la ruta seguida por el precio, así como los niveles de *stop-loss* y *take-profit*, son fundamentales.

Este método consiste en definir tres barreras que determinan cómo se etiqueta una observación en función de cuál de ellas es alcanzada primero. Las barreras son las siguientes:

- **Barrera Superior (*Take Profit*)**: Representa un nivel de precio predefinido que, si se alcanza, indica que la operación ha sido exitosa. En este caso, se asigna una etiqueta de $y_i = 1$.
- **Barrera Inferior (*Stop-Loss*)**: Es un nivel de precio que, si se toca primero, indica que la operación ha fallado, por lo que se asigna una etiqueta de $y_i = -1$.
- **Barrera Temporal (*Expiration Limit*)**: Define el número máximo de barras de tiempo después de las cuales se fuerza el cierre de la operación, independientemente de si se ha alcanzado alguna de las otras dos barreras. Si se toca esta barrera antes que las otras, se puede asignar una etiqueta de $y_i = 0$, o bien, según la preferencia, se puede etiquetar en función del signo del retorno acumulado hasta ese momento.

Figura 5.2 ilustra cómo funcionan estas tres barreras. En la imagen, se observa un ejemplo de trayectoria del precio donde las barreras determinan el etiquetado de la señal:

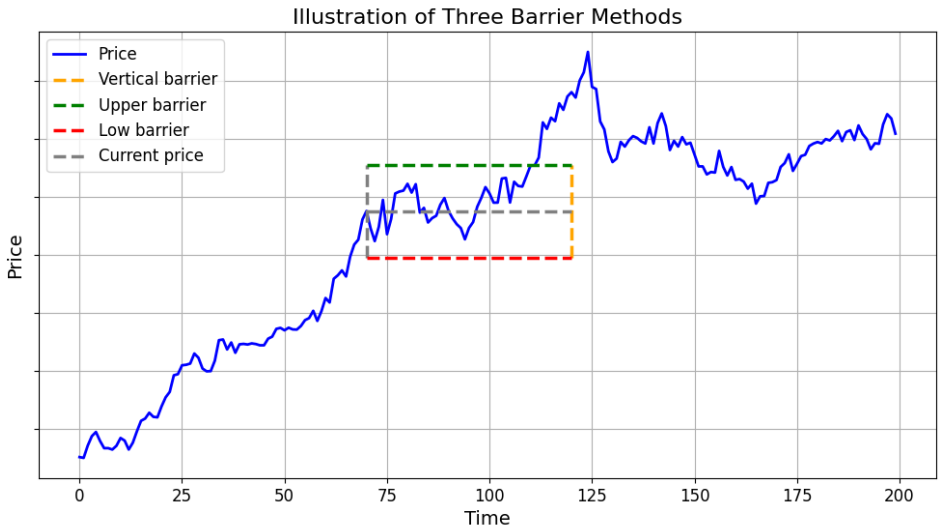

Figura 5.2: Ilustración del método de las trés barreras

A diferencia del método del horizonte temporal fijo, el *Three-Barrier Method* es dependiente de la ruta (*path-dependent*), lo que significa que toma en cuenta toda la

trayectoria del precio desde el momento en que se genera la señal ($t_{i,0}$) hasta que se toca una de las barreras.

Este método permite etiquetar de manera más realista las operaciones, ya que considera tanto los límites de toma de ganancias y pérdidas como un límite temporal, proporcionando una visión más completa del comportamiento del precio.

Este enfoque también ofrece varias ventajas clave. En primer lugar, al considerar dinámicamente los niveles de toma de ganancias y stop-loss, que pueden ajustarse en función de la volatilidad, el método reduce la probabilidad de etiquetar incorrectamente las observaciones. Esto es especialmente importante en mercados volátiles, donde los precios pueden fluctuar considerablemente en cortos períodos de tiempo. Además, la inclusión de una barrera temporal introduce una capa adicional de control, permitiendo cerrar una operación si no se alcanzan ni el nivel de toma de ganancias ni el de pérdidas dentro del tiempo estipulado, lo que ayuda a evitar situaciones en las que se mantenga una posición de manera indefinida sin obtener un beneficio claro.

En resumen, el *Método de las Tres Barreras* no solo mejora la precisión del etiquetado al tener en cuenta toda la ruta del precio, sino que también proporciona un mecanismo más flexible y adaptativo para la gestión de operaciones en estrategias de trading. Al combinar estos elementos, este método contribuye significativamente a la creación de estrategias de trading más robustas y efectivas, optimizando el rendimiento ajustado al riesgo en un entorno financiero incierto.

5.3. Conclusiones

5.3.1. Resumen de Hallazgos Principales

A lo largo de este trabajo, se han implementado y evaluado varias técnicas avanzadas en el contexto de la creación de estrategias de trading, con el objetivo de determinar su efectividad y utilidad en la práctica.

En primer lugar, la implementación de barras de desequilibrio de volumen y dólares ha demostrado ser una alternativa viable a las barras de tiempo tradicionales. Estas barras han ofrecido una representación más precisa de la actividad del mercado, capturando mejor los puntos de desequilibrio entre oferta y demanda. Los resultados muestran que las estrategias basadas en estas barras tienden a ser más sensibles a los cambios en la dinámica del mercado, lo que puede mejorar la identificación de oportunidades de trading.

Por otro lado, el entrenamiento de un algoritmo de *deep reinforcement learning* sobre estas barras de desequilibrio ha producido resultados mixtos. Aunque se observó una mejora en la capacidad del agente para adaptarse a las condiciones del mercado en comparación con las barras de tiempo, la complejidad del modelo y la necesidad de un ajuste fino han presentado desafíos adicionales. Sin embargo, estos resultados sugieren que, con una optimización adecuada, las barras de desequilibrio podrían ofrecer una base sólida para estrategias de trading más adaptativas.

Finalmente, la aplicación de la técnica de *Meta-Labeling* ha mostrado un impacto positivo en la precisión de las señales generadas por el modelo de *deep reinforcement learning*. Al actuar como una capa adicional de validación, el *Meta-Labeling* ha mejorado las métricas de precisión y *recall*, lo que sugiere que esta técnica es especialmente útil para filtrar señales y reducir la incidencia de operaciones fallidas.

Este refinamiento ha permitido optimizar el rendimiento ajustado al riesgo, haciendo que la estrategia de trading sea más robusta en entornos volátiles.

5.3.2. Cumplimiento de los Objetivos

Los objetivos planteados al inicio de este trabajo se han cumplido en gran medida. En primer lugar, se ha logrado desarrollar y evaluar una estrategia de trading utilizando técnicas avanzadas como las barras de desequilibrio, el *deep reinforcement learning*, y el *Meta-Labeling*. Estos esfuerzos han resultado en una comprensión más profunda de cómo estas herramientas pueden integrarse para mejorar el rendimiento de las estrategias de trading.

Además, se ha aprendido a implementar y ajustar algoritmos de *deep reinforcement learning*, aprovechando librerías como *Gymnasium* y *Stable Baselines*. A través de la comparación de diferentes enfoques, se ha demostrado que el uso de barras de desequilibrio y *Meta-Labeling* puede proporcionar ventajas significativas en ciertos contextos. Aunque se encontraron desafíos en la implementación, especialmente en términos de ajuste y optimización, el proceso ha sido valioso para adquirir habilidades prácticas en el uso de estas metodologías.

Finalmente, la comparativa entre técnicas ha revelado insights importantes sobre su efectividad y sus limitaciones. Los resultados obtenidos sugieren que, aunque cada técnica tiene sus propios méritos, la combinación de barras de desequilibrio y *Meta-Labeling* ofrece un enfoque prometedor para desarrollar estrategias de trading más adaptativas y precisas.

5.3.3. Limitaciones del Estudio

A pesar de los resultados positivos, este estudio presenta algunas limitaciones que deben tenerse en cuenta. Una de las principales limitaciones es la complejidad técnica asociada con la implementación y ajuste de los algoritmos de *deep reinforcement learning*. La necesidad de un poder computacional significativo y el tiempo requerido para entrenar estos modelos ha sido un desafío constante durante la investigación.

Otra limitación importante es la generalización de los resultados. Los experimentos realizados se basaron en datos específicos y en un conjunto limitado de escenarios de mercado. Por lo tanto, los resultados obtenidos pueden no ser directamente aplicables a otros mercados o condiciones sin una adecuada recalibración de los modelos.

Además, la dependencia de la calidad de los datos también representa una limitación significativa. La efectividad de las barras de desequilibrio y del *Meta-Labeling* está estrechamente ligada a la precisión y la integridad de los datos utilizados. Cualquier anomalía en los datos podría haber afectado los resultados y, en consecuencia, las conclusiones derivadas de este estudio.

5.3.4. Implicaciones Prácticas

Los hallazgos de este trabajo tienen importantes implicaciones prácticas para el desarrollo de estrategias de trading en el entorno financiero actual. En primer lugar, la adopción de barras de desequilibrio como una alternativa a las barras de tiempo tradicionales ofrece a los traders y desarrolladores de algoritmos una herramienta más afinada para capturar la dinámica del mercado. Esto podría mejorar la precisión de las

estrategias de trading, especialmente en mercados donde la volatilidad y el volumen juegan un papel crucial.

El uso de *deep reinforcement learning* en conjunto con barras de desequilibrio también muestra un gran potencial para desarrollar estrategias más adaptativas, capaces de ajustarse a las condiciones cambiantes del mercado. Sin embargo, la complejidad de su implementación sugiere que, aunque es prometedora, esta técnica requiere un enfoque cuidadoso y una considerable inversión en recursos computacionales.

Por último, la técnica de *Meta-Labeling* podría ser particularmente valiosa para los gestores de riesgo y desarrolladores de estrategias que buscan mejorar la precisión de sus modelos sin sacrificar la flexibilidad. Al refinar las señales generadas por el modelo primario, el *Meta-Labeling* ayuda a reducir la exposición a operaciones fallidas, lo que podría traducirse en una mejor gestión del riesgo y una optimización del rendimiento ajustado al riesgo.

5.3.5. Líneas Futuras de Investigación

Este trabajo abre varias posibles líneas de investigación futura. Una dirección interesante sería explorar mejoras en las técnicas implementadas, particularmente en la optimización de los algoritmos de *deep reinforcement learning*. Ajustes en los hiperparámetros, la incorporación de técnicas de aprendizaje más avanzadas, o la combinación con otros métodos de *machine learning* podrían resultar en mejoras significativas en la efectividad de las estrategias de trading.

Otra área de investigación futura podría centrarse en la aplicación de estas técnicas en diferentes mercados o activos financieros. Esto permitiría evaluar la robustez y generalización de los resultados obtenidos, así como adaptar las estrategias a distintos contextos financieros.

Además, sería valioso investigar la integración de nuevas fuentes de datos o indicadores en las estrategias de trading, como datos alternativos o indicadores de sentimiento, que podrían complementar los enfoques actuales y proporcionar una visión más completa del mercado.

En resumen, aunque este trabajo ha proporcionado resultados prometedores, la investigación en el campo del *deep reinforcement learning* aplicado al trading está lejos de ser concluyente. Existe un vasto potencial para continuar explorando y refinando estas técnicas, y las contribuciones de este estudio representan solo el comienzo de un camino mucho más amplio y complejo en la intersección entre *machine learning* y finanzas.

Bibliografía

[1] J. Joubert A. Singh. Does meta-labeling add to signal efficacy? *Journal Name*, 2019.

[2] Rama Cont. Empirical properties of asset returns: stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236, 2001.

[3] Stefano Giacomazzi Dantas and Daniel Guerreiro e Silva. Equity trading at the brazilian stock market using a q-learning based system. In *2018 7th Brazilian Conference on Intelligent Systems (BRACIS)*, pages 133–138. IEEE, oct 2018.

[4] David Easley, Marcos López de Prado, and Maureen O'Hara. The volume clock: Insights into the high frequency paradigm. *The Journal of Portfolio Management*, 39(1):19–29, 2012.

[5] Quantitative Finance Stack Exchange. Tick imbalance bars - advances in financial machine learning, 2019.

[6] Marcos López de Prado. *Advances in Financial Machine Learning*. John Wiley & Sons, Hoboken, NJ, 2018.

[7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.