Universidad del País Vasco

Euskal Herriko Unibertsitatea

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

Trabajo Fin de Máster

Título del trabajo

Alexander de la Puente González

Director/es
Josu Doncel Vicente
Nombre Apellido1 Apellido2

Lugar y fecha de presentación prevista

Índice general

Índice de figuras **III**

Índice de tablas **IV**

1. Introducción **1**
1.1. Contextualización del problema 1
1.2. Relación con el contenido del máster 1
1.3. Estructura del documento 1

2. Estado del arte ??? **3**

3. Planteamiento del problema y formulación **4**
3.1. Descripción de los Datos Financieros 4
3.1.1. Introducción a la Serie Temporal 4
3.1.2. Retornos de Activos 4
3.2. Hechos Estilizados 6
3.2.1. Propiedades Estadísticas 7
3.2.2. Datos de Alta Frecuencia 7
3.2.3. Datos Utilizados 8

4. Imbalance bars **10**
4.1. Barras Basadas en Información 10
4.1.1. Barras de Desequilibrio 11
4.1.2. Adaptación de la Metodología con Datos de Minuto 12
4.1.3. Resultados 15

5. Deep Reinforcement Learning **19**
5.1. Introducción al Aprendizaje por Refuerzo (Reinforcement Learning) 19
5.1.1. Definición y Contexto General 19
5.1.2. Componentes Básicos del Aprendizaje por Refuerzo 19
5.1.3. Métodos Clásicos de Aprendizaje por Refuerzo 21
5.2. Introducción al Aprendizaje por Refuerzo (Reinforcement Learning) 22
5.2.1. Definición y Contexto General 22
5.2.2. Componentes Básicos del Aprendizaje por Refuerzo 23
5.2.3. Métodos Clásicos de Aprendizaje por Refuerzo 25
5.2.4. Trade-off Exploración-Explotación 25
5.3. Evolución del Deep Reinforcement Learning (DRL) 26
5.3.1. Combinación con Redes Neuronales 26

5.3.2. Principales Algoritmos de DRL 26
5.3.3. Avances y Limitaciones 27

6. Meta-labeling **29**

7. Conclusiones **30**

A. Demostración del teorema **31**

Bibliografía **32**

Índice de figuras

3.1. ola 5
3.2. Serie Temporal de Precios de Bitcoin 8
3.3. Serie Temporal de Precios del SPY 9

4.1. Número de barras obtenido en datos de BTC 14
4.2. Número de barras obtenido en datos del SPY 14
4.3. Explosión del umbral de desequilibrio en SPY 15
4.4. Datos BTC originales y de Desequilibrio de Volumen 16
4.5. Comparativa de histogramas de los retornos del SPY (arriba) y BTC (abajo) 17
4.6. Comparativa de QQ-plots de los retornos del SPY (arriba) y BTC (abajo) 18

5.1. Esquema Reiforcement Learning 23

Índice de tablas

4.1. Umbrales utilizados y numero de muestras obtenido 15
4.2. Resultados de pruebas estadísticas de normalidad para SPY y BTC 17

Capítulo 1

Introducción

- Contextualización del problema.
- Antecedentes y citas a trabajos (libros, artículos o manuscritos) que sirven de referencia, como por ejemplo [2]
- Relación con el contenido de alguna/s asignatura/s del máster y técnicas o tería utilizadas en el planteamiento o resolución del problema.
- Comentario sobre los capítulos o secciones que componen la memoria.

1.1. Contextualización del problema

En los últimos años, el uso de estrategias de trading automatizadas ha cobrado una gran relevancia en el mundo financiero. La incorporación de técnicas de aprendizaje profundo y métodos avanzados de análisis financiero ha permitido el desarrollo de estrategias más sofisticadas y eficientes. En este contexto, la aplicación de Reinforcement Learning (RL) y metodologías avanzadas, como las propuestas por Marcos López de Prado, ofrecen un enfoque novedoso para abordar los desafíos del trading financiero.

1.2. Relación con el contenido del máster

Este trabajo se relaciona estrechamente con varias asignaturas del máster, como la teoría de machine learning, optimización y finanzas cuantitativas. Las técnicas de deep RL, así como las metodologías avanzadas de análisis financiero, son fundamentales para la formulación y resolución del problema planteado.

1.3. Estructura del documento

El presente trabajo se estructura en los siguientes capítulos:

- **Capítulo 1: Introducción** - Se presenta el contexto del problema, los antecedentes relevantes y la relación con los contenidos del máster.

- **Capítulo 2: Planteamiento del problema y formulación** - Se detalla el problema a resolver, incluyendo la formulación matemática y los objetivos específicos.
- **Capítulo 3: Caso de estudio y simulaciones** - Se presenta un caso de estudio concreto, junto con las simulaciones y resultados obtenidos.
- **Capítulo 4: Conclusiones** - Se resumen los hallazgos principales y se discuten posibles trabajos futuros.

Capítulo 2

Estado del arte ???

Capítulo 3

Planteamiento del problema y formulación

3.1. 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.

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

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

Propiedades destacadas:

1. $r_t = \log(1 + R_t)$
2. La serie de Taylor de $\log(1+x) = \sum_{k=1}^{\infty} \frac{(-1)^{k+1} x^k}{k}$ da como resultado $r_t \approx R_t$ cuando $R_t \approx 0$.
3. Retornos lineales compuestos:

$$R_t(k) := \frac{p_t - p_{t-k}}{p_{t-k}} = \frac{p_t}{p_{t-k}} - 1 = (1 + R_t) \cdot (1 + R_{t-1}) \cdots \left(1 + R_{t-(k-1)}\right) - 1$$

4. Retornos logarítmicos compuestos:

$$r_t(k) := \log\left(\frac{p_t}{p_{t-k}}\right) = y_t - y_{t-k} = (y_t - y_{t-1}) + \cdots + (y_{t-(k-1)} - y_{t-k}) = r_t + \cdots + r_{t-(k-1)}$$

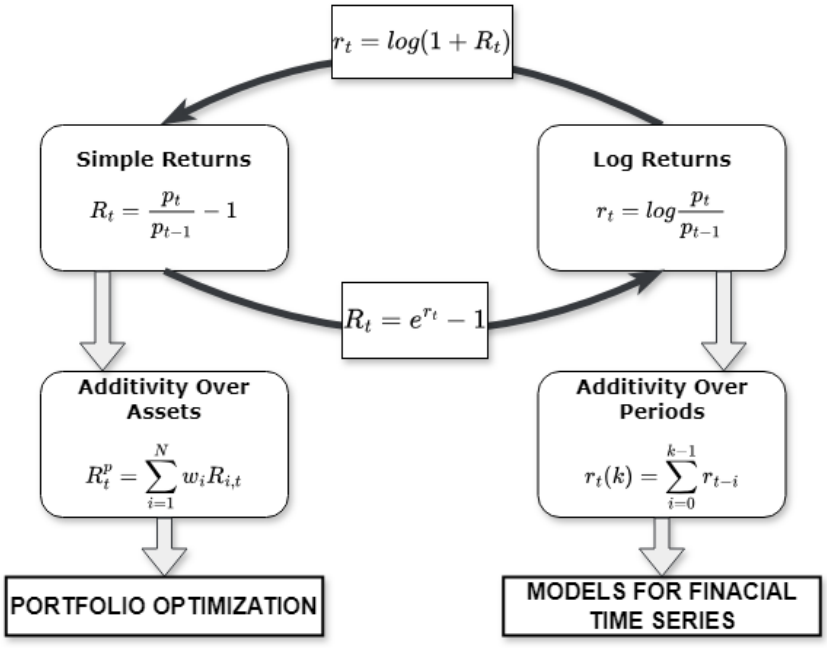

Figura 3.1: ola

3.2. 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* [1].

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.

3.2.1. Propiedades Estadísticas

La autocorrelación de los log-retornos diarios se define como:

$$\rho_k = \frac{\text{Cov}(r_t, r_{t+k})}{\sqrt{\text{Var}(r_t)\text{Var}(r_{t+k})}}$$

donde $\text{Cov}(r_t, r_{t+k})$ es la covarianza de r_t y r_{t+k}, y $\text{Var}(r_t)$ la varianza de r_t.

3.2.2. 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.

3.2.3. 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:

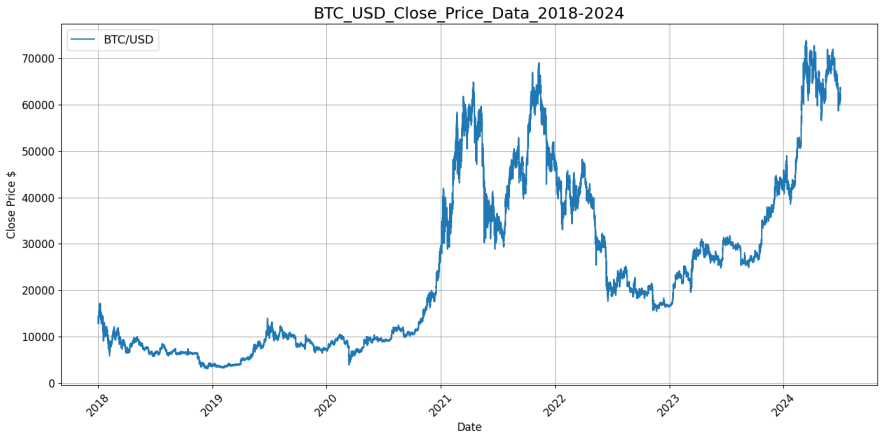

Figura 3.2: Serie Temporal de Precios de Bitcoin

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:

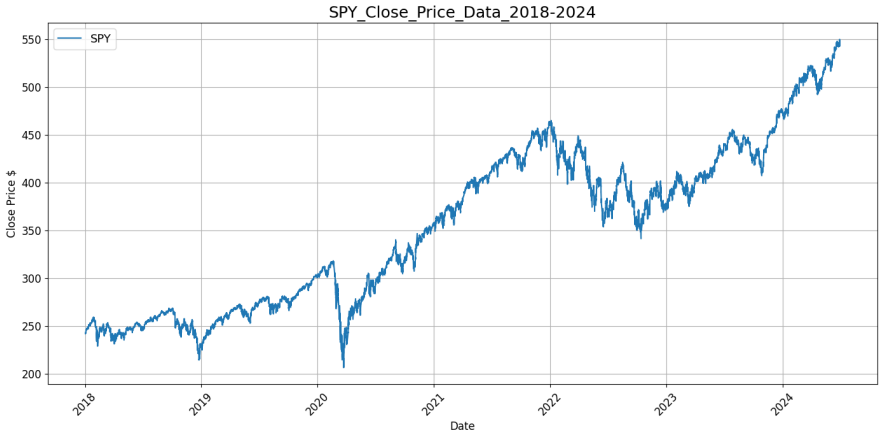

Figura 3.3: Serie Temporal de Precios del SPY

Propiedades Estadísticas de los Datos

Para verificar si los datos de Bitcoin y del SPY cumplen con los hechos estilizados, analizaremos sus propiedades estadísticas. A continuación se presentan los resultados del análisis:

Este análisis permitirá verificar si los datos de Bitcoin y del SPY cumplen con los hechos estilizados descritos por Cont (2001) [1], proporcionando una base sólida para el modelado y la estrategia de trading propuesta en este trabajo.

Capítulo 4

Imbalance bars

4.1. 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.

4.1.1. 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:

$$b_t = \begin{cases} 1 & \text{si } \Delta p_t > 0 \\ -1 & \text{si } \Delta p_t < 0 \\ b_{t-1} & \text{si } \Delta p_t = 0 \end{cases} \tag{4.1}$$

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:

$$\theta_T = \sum_{t=1}^{T} b_t \tag{4.2}$$

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

$$E_0[\theta_T] = E_0[T](2P[b_t = 1] - 1) \tag{4.3}$$

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:

$$T^* = \arg\min_T \{|\theta_T| \geq E_0[\theta_T]|2P[b_t = 1] - 1|\} \tag{4.4}$$

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:

$$\theta_T = \sum_{t=1}^{T} b_t v_t \tag{4.5}$$

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

$$E_0[\theta_T] = E_0[T](v_+ - v_-) = E_0[T](2v_+ - E_0[v_t]) \tag{4.6}$$

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:

$$T^* = \arg\min_T \{|\theta_T| \geq E_0[T]|2v_+ - E_0[v_t]|\} \tag{4.7}$$

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.

4.1.2. 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 4.2 y 4.1 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.

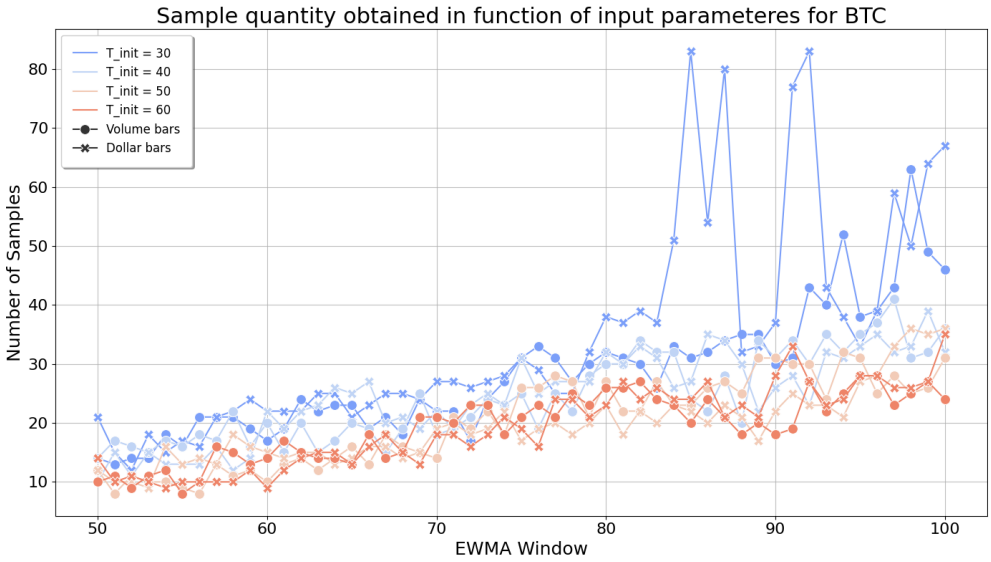

Figura 4.1: Número de barras obtenido en datos de BTC

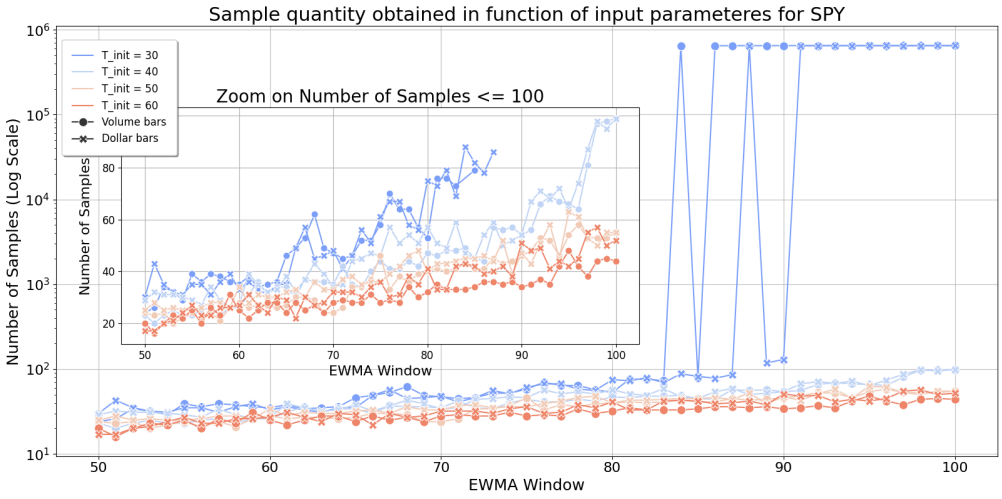

Figura 4.2: Número de barras obtenido en datos del SPY

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 4.3. 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.

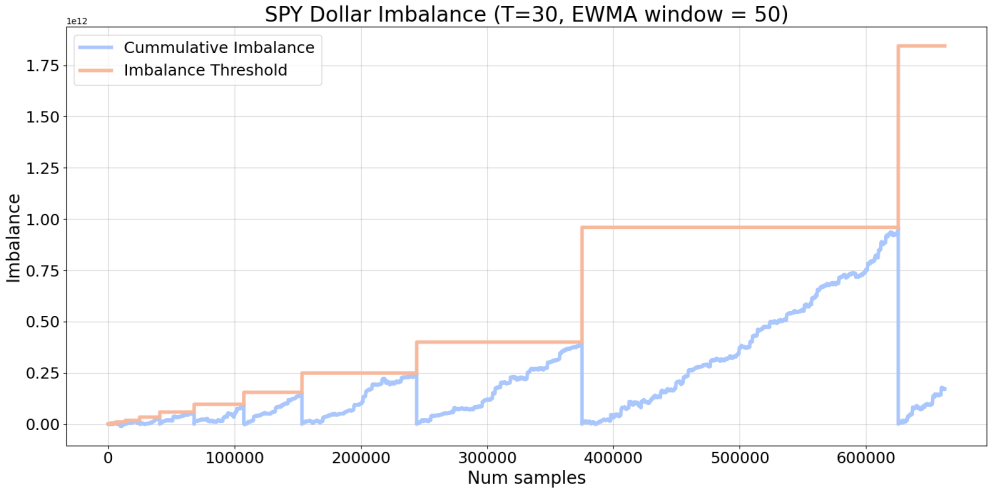

Figura 4.3: Explosión del umbral de desequilibrio en SPY

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 [4].

4.1.3. 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 4.1.

Tabla 4.1: Umbrales utilizados y numero de muestras obtenido

Activo	**Des-equilibrio**	**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 4.4 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.

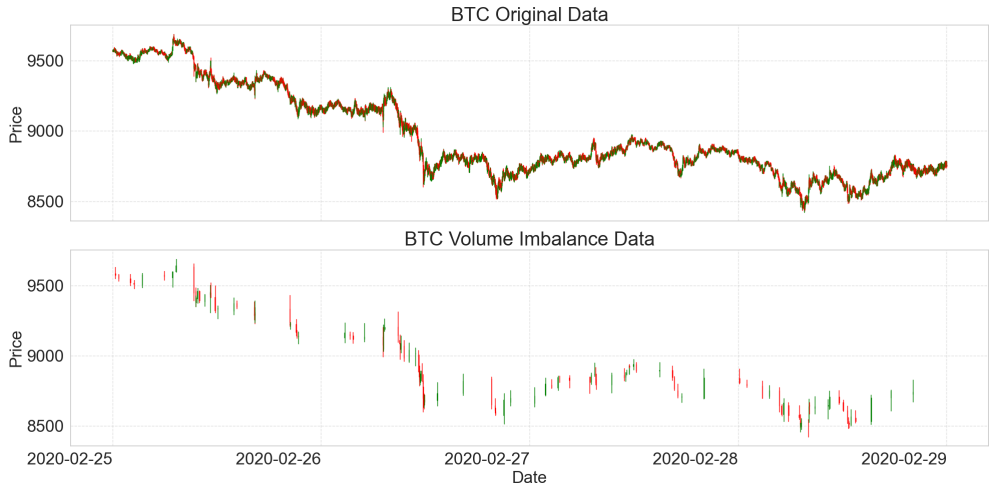

Figura 4.4: Datos BTC originales y de Desequilibrio de Volumen

El artículo de Easley, López de Prado y O'Hara titulado "The Volume Clock: Insights into the High Frequency Paradigm"[3] 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 4.2. 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 4.2: 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 4.5 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 4.6, 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.

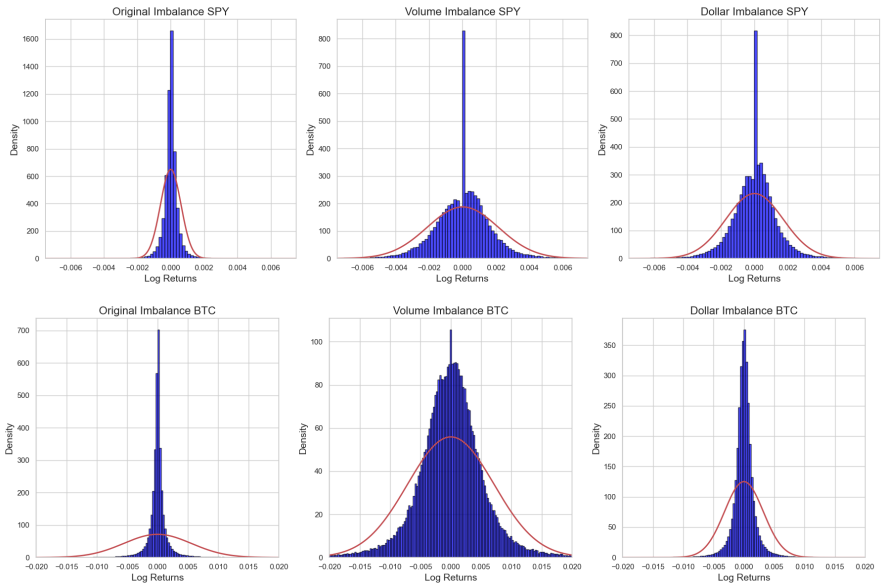

Figura 4.5: Comparativa de histogramas de los retornos del SPY (arriba) y BTC (abajo)

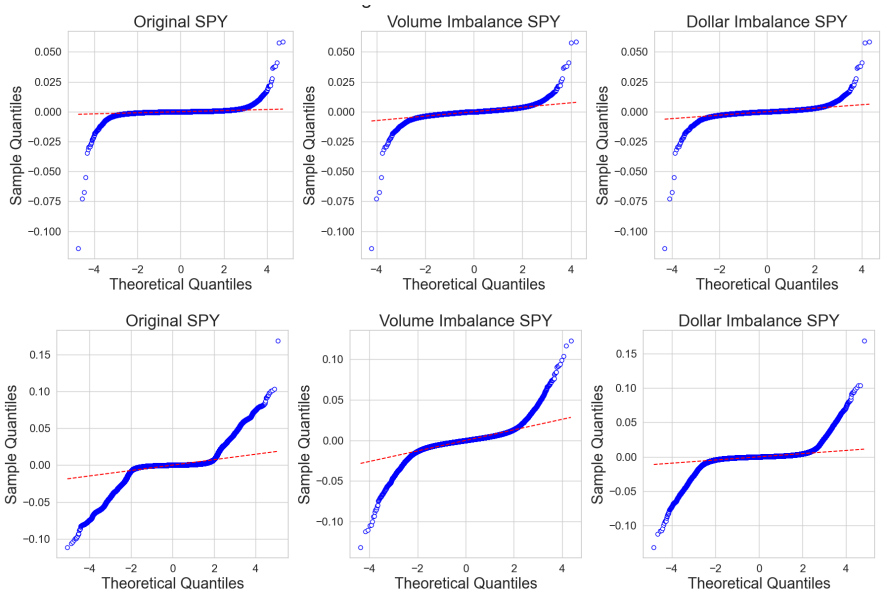

Figura 4.6: Comparativa de QQ-plots de los retornos del SPY (arriba) y BTC (abajo)

Capítulo 5

Deep Reinforcement Learning

5.1. Introducción al Aprendizaje por Refuerzo (Reinforcement Learning)

5.1.1. Definición y Contexto General

El *Aprendizaje por Refuerzo* (Reinforcement Learning, RL) es una rama dentro 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 de ejemplos previamente etiquetados, o el *aprendizaje no supervisado*, donde el modelo busca patrones en datos sin etiquetas, un agente de RL basa su aprendizaje en la retroalimentación recibida por el entorno en forma de recompensas o penalizaciones por las acciones que toma.

En un entorno de RL, se define un *Proceso de Decisión de Markov* (MDP, por sus siglas en inglés), 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. Este marco es 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.

5.1.2. Componentes Básicos del Aprendizaje por Refuerzo

El proceso de RL se puede modelar mediante un *Proceso de Decisión de Markov* (Markov Decision Process, MDP), que se define por un cuarteto $\langle S, A, P, R \rangle$:

- **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.

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.

5.1.3. 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).

5.2. Introducción al Aprendizaje por Refuerzo (Reinforcement Learning)

5.2.1. 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.

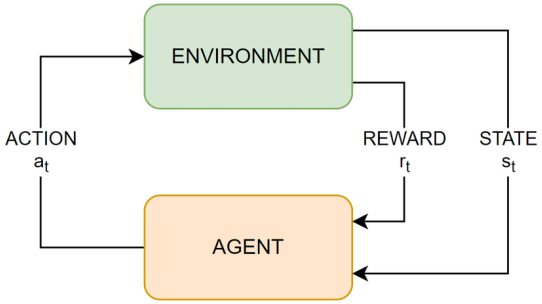

Figura 5.1: Esquema Reiforcement Learning

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.

5.2.2. 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.

Política

Una *política* define el comportamiento del agente, es decir, es un mapeo de estados a acciones. Las políticas pueden ser:

- **Determinísticas**: $A = \pi(S)$.
- **Estocásticas**: $\pi(A|S) = p(A|S)$, donde A es elegido probabilísticamente dado el estado S.

Función de Valor

En el contexto del RL, las funciones de valor son fundamentales para evaluar qué tan "bueno.es estar en un estado o tomar una acción específica en un estado, bajo una política dada π. Existen dos tipos principales de funciones de valor:

- **Función de Valor de Estado ($V^\pi(s)$)**:

  $$V^\pi(s) = \mathbb{E}[G_t|S_t = s, \pi] = \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots |S_t = s, \pi]$$

  Esta función estima el valor esperado del retorno futuro comenzando desde el estado s y siguiendo la política π.

- **Función de Valor de Acción ($Q^\pi(s,a)$)**:

  $$Q^\pi(s,a) = \mathbb{E}[G_t|S_t = s, A_t = a, \pi] = \mathbb{E}[R_{t+1}+\gamma R_{t+2}+\gamma^2 R_{t+3}+\ldots |S_t = s, A_t = a, \pi]$$

  Esta función estima el valor esperado del retorno futuro al tomar la acción a en el estado s y seguir la política π a partir de entonces.

El agente usa estas funciones de valor para seleccionar acciones que maximicen el retorno esperado a largo plazo.

Ecuación de Bellman

La ecuación de Bellman proporciona una forma recursiva de calcular las funciones de valor, aprovechando la estructura de las recompensas futuras. Para la función de valor de estado, la ecuación de Bellman es:

$$V^{\pi}(s) = \mathbb{E}[R_{t+1} + \gamma V^{\pi}(S_{t+1}) | S_t = s, \pi]$$

De manera similar, para la función de valor de acción:

$$Q^{\pi}(s, a) = \mathbb{E}[R_{t+1} + \gamma Q^{\pi}(S_{t+1}, A_{t+1}) | S_t = s, A_t = a, \pi]$$

Estas ecuaciones son fundamentales para el desarrollo de algoritmos de RL, ya que permiten descomponer el problema de optimización en subproblemas más manejables.

5.2.3. Métodos Clásicos de Aprendizaje por Refuerzo

Existen varios algoritmos clásicos de RL que permiten a los agentes aprender políticas óptimas, siendo los más conocidos *Q-learning* y *SARSA*.

- **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.

5.2.4. Trade-off Exploración-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.

—

5.3. Evolución del Deep Reinforcement Learning (DRL)

5.3.1. 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. Esta capacidad ha sido particularmente útil en problemas de RL donde el espacio de estados es continuo o extremadamente grande, como es común en el trading financiero, la robótica y los videojuegos.

5.3.2. Principales Algoritmos de DRL

El desarrollo de DRL ha dado lugar a una serie de algoritmos innovadores que han transformado la forma en que se aborda el aprendizaje por refuerzo en entornos complejos. A continuación, se presentan algunos de los algoritmos más importantes en este campo:

Deep Q-Networks (DQN)

El *Deep Q-Network* (DQN) es uno de los algoritmos más representativos de la combinación entre RL y Deep Learning. DQN fue introducido por *DeepMind* y marcó un hito en la historia del RL al aplicar 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.

Policy Gradient Methods

Los *Policy Gradient Methods* son 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 los métodos de gradiente de política 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 inestable.

Actor-Critic Methods

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.

5.3.3. Avances y Limitaciones

El desarrollo del DRL ha permitido avances significativos en la capacidad de los agentes para aprender en entornos complejos y de alta dimensionalidad. Algunos de los avances más destacados incluyen:

- **Escalabilidad**: El uso de redes neuronales profundas ha permitido que los algoritmos de RL escalen a problemas que antes eran inabordables, como

juegos complejos y mercados financieros.

- **Capacidad de Generalización**: DRL ha mostrado una notable capacidad para generalizar el conocimiento adquirido en un entorno a otros entornos similares, lo que es crucial en aplicaciones como el trading, donde las condiciones del mercado cambian constantemente.
- **Desempeño en Tareas Complejas**: DRL ha superado a los humanos en varias tareas complejas, como juegos de Atari, y ha logrado una eficiencia sin precedentes en aplicaciones como la optimización de portafolios.

Sin embargo, el DRL también presenta limitaciones que deben ser abordadas:

- **Complejidad Computacional**: Entrenar modelos de DRL requiere un poder computacional significativo y grandes volúmenes de datos, lo que puede ser un obstáculo para su aplicación en tiempo real.
- **Estabilidad y Convergencia**: Los algoritmos de DRL pueden ser inestables y propensos a la divergencia durante el entrenamiento, especialmente en entornos con recompensas escasas o en problemas altamente no lineales.
- **Sensibilidad a la Parametrización**: El desempeño de los algoritmos de DRL es altamente sensible a la elección de hiperparámetros, lo que requiere una considerable experimentación y ajuste fino.

En conclusión, el DRL ha ampliado significativamente el alcance de las aplicaciones de RL, permitiendo a los agentes abordar problemas más complejos y realistas. No obstante, sigue siendo un área de investigación activa con desafíos técnicos y prácticos que deben ser resueltos para su adopción generalizada.

PUNTOS A INTRODUCIIIIIIIIIIIIIR

- Explicación de las ecuaciones y metodos (bellman, metodos de diferencias temporales vs monte carlo,)
- Algoritmos utilizados
- Implementación del entorno en gimnasium
- obtención del espacio de observaciones (indicadores...)
- funciones de recompensa
- resultados
- conclusiones

Capítulo 6

Meta-labeling

Consiste en la utilización de un modelo secundario para afinar las predicciones del primario.

- **Generación de señales**: El modelo primario genera señales
- **Asignación de labeles**: Cada señal generada por el modelo primario se etiqueta como correcta o incorrecta en base al rendimiento observado.
- El libro propone varias maneras de hacerlo.
- **Entrenamiento del meta-modelo**: El meta-modelo se entrena utilizando estas etiquetas y otras features adicionales que puedan influir en la precision de la señla.
- **Prediccion de la calidad de la señales**: Una vez entrenado, el meta-modelo se utiliza para predecir la probabilidad de que las futuras señales del modelo primario sean correctas.

Capítulo 7

Conclusiones

Apéndice A

Demostración del teorema

Bibliografía

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

[2] Martin Dindos, Jill Pipher, and David Rule. Boundary value problems for second-order elliptic operators satisfying a carleson condition. *Commun Pure Appl Math*, 70(7):1316–1365, 2017.

[3] 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.

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