Optimización de Portafolios mediante aprendizaje por refuerzo: *Proximal Policy Optimization* (PPO)

García Sánchez Daniel Alfredo

Índice

1. Introducción

El aprendizaje de máquina aplicado a finanzas [1] es una intersección de disciplinas emergentes como el reconocimiento de patrones, econometría, cómputo estadístico y programación dinámica. Debido al incremento de la capacidad computacional y al aumento en el tamaño de los conjuntos de datos, el aprendizaje de máquina ha tenido un crecimiento importante en los últimos años. Una de las primeras áreas financieras en adoptar técnicas de *Machine Learning* (ML) fueron los algoritmos de *trading*.

El aprendizaje por refuerzo o *Reinforcement Learning* (RL) [2] es una fusión de conceptos interdisciplinarios; la ley del efecto en la psicología, la teoría de control óptimo, el uso de estímulos que desvanecen con el paso del tiempo y la programación dinámica, específicamente la ecuación de Bellman [3]. La idea básica de este tipo de aprendizaje es que un agente toma acciones de acuerdo a cierta política, que puede ser determinista o estocástica, con el objetivo de maximizar una función recompensa. En el ámbito de optimización de portafolios, la función recompensa es el retorno de la inversión realizada por el agente.

Se han realizado diversas investigaciones en finanzas utilizando distintos enfoques del *Reinforcement Learning*, por ejemplo, en aprendizaje por refuerzo de solo crítico destaca [4] al introducir una variante del DRQN (*Deep Recurrent Q-Network*) para procesar mejor las series de tiempo de los precios del ETF SPY, en el que el agente tiene tres posibles acciones: comprar, vender o mantener la acción y la recompensa se calcula como la diferencia de precio de cierre del día siguiente y del día actual. Sin embargo, solo buscan maximizar el retorno sin tomar en cuenta el riesgo; además, al considerar un solo activo, no se puede concluir que el método pueda generalizarse.

Utilizando un enfoque de aprendizaje por refuerzo de solo actor, en el que se pueden definir espacios de acciones continuas. En [5] modifican la estructura de una red neuronal recurrente para poder detectar el entorno y la toma de decisiones de forma simultánea, que al combinarse con una variante de retropropagación, logra solucionar el problema del desvanecimiento del gradiente en estructuras complejas. La desventaja en métodos de solo actor es que el tiempo para aprender la política óptima es mayor, y los autores en este caso eligieron únicamente tres activos financieros, dificultando la diversificación de un portafolio.

Finalmente, empleando un enfoque de aprendizaje por refuerzo de actor-crítico, destaca el trabajo en [6], pues combinan tres algoritmos: PPO (*Proximal Policy Optimization*), A2C (*Advantage Actor Critis*) y DDPG (*Deep Deterministic Policy Gradient*) para lograr una mayor robustez a la hora de tomar decisiones y adaptándose mejor a diferentes escenarios, pues cada agente toma decisiones diferentes ante tendencias, caídas y la volatilidad del mercado.

Las redes complejas también han sido utilizadas para la optimización de portafolios de inversión. Por ejemplo, [7] logra caracterizar la estructura compleja del mercado financiero, modelando su evolución a través de redes

temporales y evaluando su estabilidad mediante un análisis de la estructura topológica. Lo que les ayuda a seleccionar las acciones del portafolio. Mientras que los autores de [8] transforman un mercado financiero en una red compleja, en la que los nodos son valores y los enlaces correlaciones de retorno entre los distintos valores, encontrando que los portafolios óptimos están conformados por valores con baja centralidad, pues los de alta centralidad suelen ser acciones antiguas o de gran capitalización, pero también las hay más baratas y riesgosas.

2. Marco Teórico

2.1. Conceptos de finanzas

Un *activo financiero* es un título o anotación contable que otorga a su comprador el derecho a recibir un ingreso futuro por parte del vendedor. Son emitidos por una entidad económica (como empresas y gobiernos) y acreditan la titularidad de ciertos derechos económicos. Los activos financieros tienen las características:

- Liquidez: La capacidad de convertir el activo en efectivo sin sufrir pérdidas.
- Riesgo: La probabilidad de que el vendedor no cumpla con su compromiso y se calcula en función de las garantías que aporte y su solvencia. A mayor riesgo, mayor rentabilidad.
- Rentabilidad: El interés que recibe el comprador del activo en contraprestación al riesgo que asume.

Los activos pueden ser de renta fija o renta variable. La *renta fija* son activos financieros emitidos por empresas, que se comprometen a devolver el dinero prestado en un plazo fijo adicional a una rentabilidad acordada previamente. Por otro lado, los activos de *renta variable* no garantizan la recuperación del dinero invertido ni alguna rentabilidad.

Las *acciones* son títulos representativos de una parte del capital social de una sociedad anónima, se negocian en los mercados bursátiles. Se puede ganar dinero vendiéndolas a un precio mayor al de compra, o a través del reparto de *dividendos* que realizan las sociedades anónimas a sus accionistas.

Un *portafolio* o *cartera de inversiones* es el conjunto total de activos financieros que una persona tiene, puede estar compuesto por instrumentos de renta fija, variable o mixta. Dicho portafolio tiene una serie de características como la tolerancia al riesgo, la rentabilidad y el tiempo dispuesto a esperar para obtener ganancias.

Los *índices bursátiles* internacionales son un conjunto de acciones. Se calculan como la media (simple o ponderada de acuerdo a cierta característica) de precios de un número de acciones seleccionadas pertenecientes a un mercado o un sector determinado. Su rendimiento permite conocer la evolución y tendencia de la parte del mercado a la que representan.

El índice *Standard & Poor's 500* se basa en la capitalización bursátil de 500 empresas que poseen acciones que cotizan en las bolsas NYSE o NASDAQ, y captura aproximadamente el 80 % de toda la capitalización de mercado en Estados Unidos. Las empresas que componen el S&P 500 son seleccionadas por un comité, basándose en ocho criterios primarios: capitalización bursátil, liquidez, domicilio, capital flotante, clasificación del sector, viabilidad financiera, periodo de tiempo durante el cual ha cotizado en bolsa y bolsa de valores. El índice está ponderado de acuerdo a la capitalización de cada empresa. Así, empresas con mayor capital flotante (precio de una acción $\times$ cantidad de acciones que no pertenecen a fundadores), tendrán mayor peso en el índice. El *Standard & Poor's 100* es un subconjunto con las 100 empresas estadounidenses con mayor capitalización bursátil.

2.2. Conceptos de *Reinforcement Learning*

Lo principal al hablar de aprendizaje por refuerzo es el **agente** y el **entorno**. El entorno es el espacio en el que el agente interactúa. En cada paso, el agente ve una versión limitada del entorno y, en base a esa observación, decide qué **acción** tomar dentro de un **espacio de acciones** previamente definido. Una vez que el agente toma una acción, el entorno cambia (aunque también puede cambiar por su cuenta) [3].

Así como el agente interactúa con el entorno, el agente recibe una señal de **recompensa**, la cual es un número que el agente utiliza para determinar qué tan bien o mal está el estado actual del entorno. El objetivo del agente es el de maximizar la recompensa acumulada, la cual se denomina **retorno** [3].

Un **estado** s es una descripción completa del entorno, mientras que una observación es una descripción parcial del estado la cual puede omitir información. Los estados y observaciones se representan mediante vectores, matrices o tensores de valores reales [3].

Cada entorno permite diferentes acciones, al conjunto de acciones válidas en un entorno particular se conoce como **espacio de acciones**; los cuales pueden ser discretos, como comprar, vender o mantener una acción, o continuos como elegir qué porcentaje del portafolio va a cada acción [3].

Una **política** es una regla que utiliza un agente para decidir qué acciones tomar. Puede ser determinista, denotada por:
$\mu : a_t = \mu(s_t)$,
o estocástica, denotada por:
$\pi : a_t \sim \pi(\cdot|s_t)$ [3].

Al pasar a la práctica en aprendizaje por refuerzo, se trabaja con políticas parametrizadas. Son políticas cuyos *outputs* son funciones programables que dependen de un conjunto de parámetros, como los pesos y sesgo de una red neuronal, los cuales se ajustan para cambiar el comportamiento del agente utilizando algoritmos de optimización. Para clarificar que se trata de la política parametrizada y no de la original, se utiliza:

$a_t = \mu_\theta(s_t)$, si es determinista o,
$a_t \sim \pi_\theta(\cdot|s_t)$. si es estocástica [3].

Una **trayectoria** τ es una secuencia de estados y acciones en el mundo $\tau = (s_0, a_0, s_1, a_1, ...)$. El primer estado del entorno s_0 es elegido de forma aleatoria de la distribución de estados inicial $\rho_0 : s_0 \sim \rho_0(\cdot)$. Las transiciones de estado (lo que sucede en el entorno entre el estado a tiempo t, s_t, y el estado a tiempo $t+1$, s_{t+1}). Las transiciones de estado siguen las leyes del entorno y tienen propiedad markoviana, es decir, dependen únicamente de la acción más reciente a_t. De nuevo, las transiciones pueden ser deterministas: $s_{t+1} = f(s_t, a_t)$ o estocásticas
$s_{t+1} \sim P(\cdot|s_t, a_t)$ [3].

La **función recompensa** R depende únicamente del estado actual del entorno, de la acción tomada en ese momento y del siguiente estado del entorno:
$r_t = R(s_t, a_t, s_{t+1})$
aunque suele simplificarse para solo depender del estado actual: $r_t = R(s_t)$, o del par estado-acción $r_t = R(s_t, a_t)$ [3].

El objetivo del agente es maximizar la recompensa acumulada sobre una trayectoria. Existen diferentes tipos de retornos, por ejemplo, la suma de recompensas obtenidas en una ventana de periodo de n pasos (*finite-horizon undiscounted return*):
$R(\tau) = \sum_{t=0}^{T} r_t$.
Otro tipo de retorno es la suma de todas las recompensas obtenidas por el agente, pero con una tasa de descuento que depende de cuántos pasos atrás se obtuvo (*infinite-horizon discounted return*), esta variante incluye un factor de olvido $\gamma \in [0, 1]$, el cual ayuda a que la suma converja:
$R(\tau) = \sum_{t=0}^{\infty} \gamma^t r_t$.
En la práctica, se suele trabajar con algoritmos que busquen optimizar la recompensa finita, para trabajar con tareas finitas y en ventanas de tiempo, pero utilizando el factor de descuento al estimar las funciones de valor [3].

Ya sea que se esté utilizando un *retorno descontado con horizonte infinito* o un *retorno sin descuento con horizonte finito*), e independientemente de la política elegida, el objetivo en *Reinforcement Learning* es seleccionar una política que maximice el retorno esperado cuando el agente actúa de acuerdo a esta. Para hablar del retorno esperado, primero debemos hablar de las distribuciones de probabilidad sobre trayectorias. Si suponemos que tanto las transiciones del entorno como la política son estocásticas. La probabilidad de una trayectoria de longitud T está dada por:

$$P(\tau|\pi) = \rho_0(s_0) \prod_{t=0}^{T-1} P(s_{t+1}|s_t, a_t)\, \pi(a_t|s_t)$$

El retorno esperado ($J(\pi)$), se define como:

$$J(\pi) = \int_{\tau} P(\tau|\pi)R(\tau) = \mathbb{E}_{\tau\sim\pi}[R(\tau)]$$

El problema principal de optimización en Aprendizaje por Refuerzo se puede expresar como:

$$\pi^* = \arg\max_{\pi} J(\pi)$$

donde π^* es la **política óptima** [3].

A menudo es útil conocer el **valor** de un estado, o de un par estado-acción. El valor es el **retorno esperado** si se comienza en ese estado (o par estado-acción) y se actúa siguiendo cierta política para siempre. Las funciones de valor se utilizan en casi todos los algoritmos de Aprendizaje por Refuerzo.

La **función de valor bajo política (*On-Policy Value Function*)** $V^{\pi}(s)$
Es el retorno esperado si se comienza en el estado s y se actúa siempre siguiendo la política π:

$$V^{\pi}(s) = \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid s_0 = s\right]$$

La **función de valor-acción bajo política (*On-Policy Action-Value Function*)** $Q^{\pi}(s,a)$
Es el retorno esperado si, comenzando en el estado s, se toma una acción arbitraria a (que puede no provenir de la política), y luego se actúa siempre siguiendo la política π:

$$Q^{\pi}(s,a) = \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid s_0 = s, a_0 = a\right]$$

La **función de valor óptima (*Optimal Value Function*)** $V^*(s)$
Es el retorno esperado si se comienza en el estado s y se actúa siempre siguiendo la política óptima en el entorno:

$$V^*(s) = \max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid s_0 = s\right]$$

Finalmente, la **función de valor-acción óptima (*Optimal Action-Value Function*)** $Q^*(s,a)$
Es el retorno esperado si se comienza en el estado s, se toma una acción a, y se actúa siempre siguiendo la política óptima en el entorno:

$$Q^*(s,a) = \max_{\pi} \mathbb{E}_{\tau\sim\pi}\left[R(\tau) \mid s_0 = s, a_0 = a\right] [3]$$

Existe una relación entre la función de valor-acción óptima $Q^*(s,a)$ y la acción seleccionada por la política óptima. Por definición, $Q^*(s,a)$ da el retorno esperado de comenzar en el estado s, tomar una acción arbitraria a, y luego actuar según la política óptima de ahí en adelante.
La política óptima en s seleccionará la acción que maximice el retorno esperado al comenzar en s. Como

resultado, si tenemos Q^*, podemos obtener directamente la acción óptima, $a^*(s)$, mediante

$$a^*(s) = \arg\max_a Q^*(s, a).$$

Puede haber múltiples acciones que maximicen $Q^*(s, a)$, en ese caso, todas serían óptimas, y la política óptima puede seleccionar aleatoriamente cualquiera de ellas. Pero siempre existe una política óptima que selecciona de manera determinista una acción [3].

Las cuatro funciones de valor mencionadas anteriormente obedecen ecuaciones autorecursivas llamadas **ecuaciones de Bellman**. La idea básica detrás de las ecuaciones de Bellman es la siguiente:

El valor del punto de inicio es la recompensa esperada al estar ahí, mas el valor de donde se llegue al siguiente tiempo. Las ecuaciones de Bellman para las funciones de valor bajo política son:

$$V^\pi(s) = \mathbb{E}_{a\sim\pi, s'\sim P}[r(s, a) + \gamma V^\pi(s')],$$

y

$$Q^\pi(s, a) = \mathbb{E}_{s'\sim P}[r(s, a) + \gamma \mathbb{E}_{a'\sim\pi}[Q^\pi(s', a')]],$$

con $s' \sim P$ una forma abreviada para $s' \sim P(\cdot|s, a)$, lo que indica que el siguiente estado s' se obtiene de las reglas de transición del entorno; $a \sim \pi$ es una forma abreviada para $a \sim \pi(\cdot|s)$; y $a' \sim \pi$ es una forma abreviada para $a' \sim \pi(\cdot|s')$.
Las ecuaciones de Bellman para las funciones de valor óptimas son

$$V^*(s) = \max_a \mathbb{E}_{s'\sim P}[r(s, a) + \gamma V^*(s')],$$

$$Q^*(s, a) = \mathbb{E}_{s'\sim P}[r(s, a) + \gamma \max_{a'} Q^*(s', a')],$$

La diferencia entre las ecuaciones de Bellman para las funciones de valor bajo política y las funciones de valor óptimas es la presencia o ausencia del máx sobre las acciones. La presencia del máx refleja el hecho de que, siempre que el agente pueda elegir una acción, para actuar de manera óptima, debe seleccionar la acción que lleve al valor más alto [3].

Si no necesitamos describir qué tan buena es una acción en un sentido absoluto, sino en comparación con otras acciones. Es decir, queremos conocer la ventaja relativa de esa acción con respecto a otras, se utiliza la **función de ventaja**.

La función de ventaja $A^\pi(s, a)$ correspondiente a una política π describe cuánto mejor es tomar una acción específica a en el estado s, en lugar de seleccionar una acción al azar según $\pi(\cdot|s)$. La función de ventaja

asume que se actúa de acuerdo con π en todo momento futuro. Formalmente, la función de ventaja se define como:

$$A^{\pi}(s,a) = Q^{\pi}(s,a) - V^{\pi}(s)$$ [3].

2.3. Taxonomía de algoritmos y sus diferencias

Existen diversos algoritmos de *Reinforcement Learning*, los más utilizados pueden clasificarse de acuerdo al siguiente diagrama:

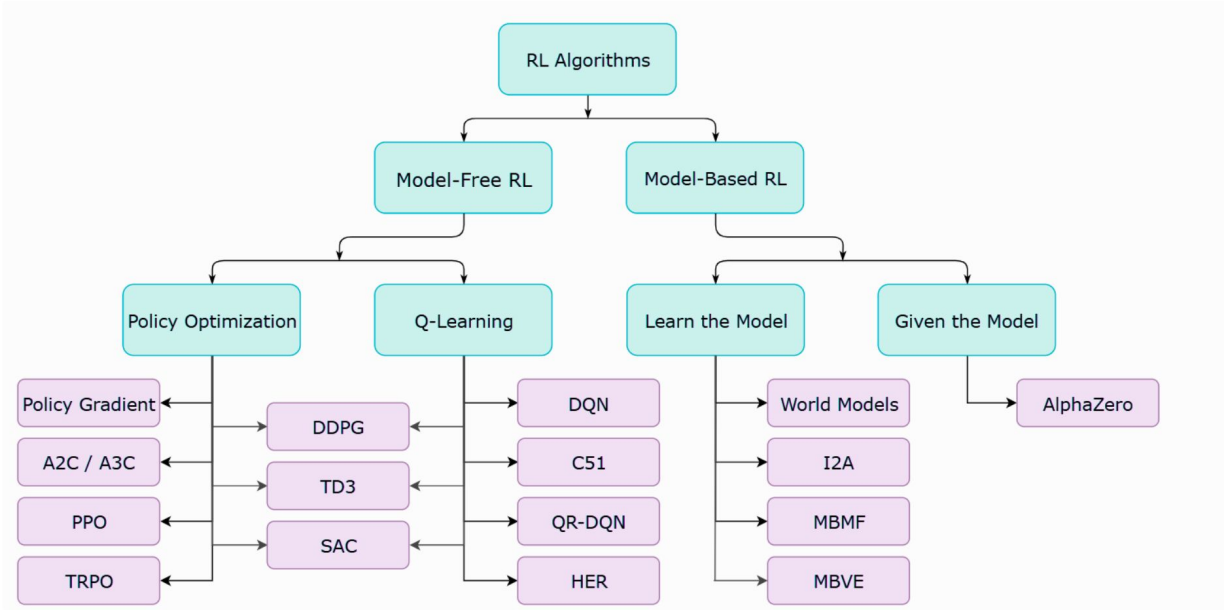

Figura 1: Clasificación taxonómica de los modelos de aprendizaje por refuerzo [9].

Como vemos en la imagen anterior, la primera forma de diferenciar los algoritmos es viendo si son *Model-free* o *Model-Based*. Los algoritmos basados en modelos utilizan una función que predice transiciones de estado y recompensas. La principal ventaja de tener un modelo es que el agente puede planear a futuro observando los resultados de un rango de posibles acciones, decidiendo explícitamente sobre esas opciones. Los agentes pueden filtrar los resultados de dicha planificación anticipada y convertirlos en una política. Los algoritmos basados en modelos suelen tener una ventaja en escenarios donde los datos son escasos [9].

La desventaja principal de los algoritmos basados en modelos es que los agentes no suelen tener acceso a un modelo real del entorno y, si un agente quiere usar un modelo, debe aprenderlo de la experiencia, lo que genera varias complicaciones. Entre las dificultades se encuentra que el agente puede abusar del sesgo, generando que funcione bien respecto a ese modelo, pero tenga comportamientos subóptimos o malos en un entorno real [9].

Los métodos sin modelo no tienen acceso a las posibles mejoras en la eficiencia de muestreo, pero tienden a ser más fáciles de implementar y ajustar. Dentro de los algoritmos sin modelo hay dos grandes grupos; los *Policy Optimization* y los *Q-Learning*, en este trabajo nos centraremos en los primeros [9].

Los métodos de optimización de políticas representan una política $\pi_\theta(a|s)$ explícitamente. Optimizan los parámetros θ de forma directa, utilizando ascenso de gradiente (se utiliza ascenso y no descenso pues buscamos maximizar el retorno esperado $J(\pi_\theta)$, no minimizar una función de pérdida) sobre $J(\pi_\theta)$, o indirectamente, maximizando aproximaciones locales de $J(\pi_\theta)$. La aproximación suele realizarse *on-policy*, es decir, que cada actualización utiliza datos recolectados a partir de la versión más reciente de la política. La optimización de política también requiere aprender un aproximador $V_\phi(s)$ para la función valor $V^\pi(s)$, el cual se utiliza para determinar cómo actualizar la política.

El algoritmo *Proximal Policy Optimization* (PPO), maximiza indirectamente el rendimiento mediante sus actualizaciones. Maximiza una función objetivo sustituta *surrogate objective function* que proporciona una estimación de cuánto cambiará $J(\pi_\theta)$ como resultado de la actualización.

2.4. Procesos de decisión de Markov y ecuación de Bellman

De forma más formal, los algoritmos de aprendizaje por refuerzo pueden verse como *procesos de decisión de Markov* (MDP). Son una quintupla $M = \langle S, A, P, R, \gamma \rangle$ con S el conjunto de estados (que tienen propiedad Markoviana), un conjunto de acciones A, una función de probabilidad de transición P, una función recompensa R y un factor de descuento para recompensas futuras [10].

Las ecuaciones de Bellman son un conjunto de ecuaciones que descomponen la función de valor en la recompensa inmediata más los valores futuros descontados.

$$\begin{aligned} V(s) &= \mathbb{E}[G_t \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma(R_{t+2} + \gamma R_{t+3} + \cdots) \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma G_{t+1} \mid S_t = s] \\ &= \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s] \end{aligned}$$

y de forma igual para la función acción-valor

$$\begin{aligned} Q(s,a) &= \mathbb{E}[R_{t+1} + \gamma V(S_{t+1}) \mid S_t = s, A_t = a] \\ &= \mathbb{E}[R_{t+1} + \gamma \mathbb{E}_{a\sim\pi} Q(S_{t+1}, a) \mid S_t = s, A_t = a] \text{ [10]} \end{aligned}$$

El proceso de se puede descomponer aún más en ecuaciones basadas tanto en las funciones de valor de estado

V como en las funciones de valor de acción Q. A medida que se avanza en la iteración, se extienden V y Q de forma alternante siguiendo la política π.

$$
\begin{aligned}
V_\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) Q_\pi(s,a) \\
Q_\pi(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V_\pi(s') \\
V_\pi(s) &= \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V_\pi(s') \right) \\
Q_\pi(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} \sum_{a' \in \mathcal{A}} \pi(a'|s') Q_\pi(s',a')
\end{aligned}
$$

Si solo nos interesan los valores óptimos de las acciones y estados, en lugar de calcular la esperanza de las recompensas futuras al seguir una política, se asume que se toma la mejor acción en cada paso, obteniendo el máximo mediante actualizaciones alternantes entre la función valor y la función de estado-acción. Los valores óptimos V^* y Q^* son las mejores recompensas acumuladas:

$$
\begin{aligned}
V^*(s) &= \max_{a \in \mathcal{A}} Q^*(s,a) \\
Q^*(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V^*(s') \\
V^*(s) &= \max_{a \in \mathcal{A}} \left(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} V^*(s') \right) \\
Q^*(s,a) &= R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P^a_{ss'} \max_{a' \in \mathcal{A}} Q^*(s',a')
\end{aligned}
$$

Las cuales se parecen a las esperanzas de las ecuaciones de Bellman. Si se tiene la información completa del entorno, el problema se convertiría en uno de planeación, el cual se puede resolver mediante *programación dinámica*; en la mayoría de escenarios no se conoce completamente a $P^a_{ss'}$ o $R(s,a)$, por lo que no se puede resolver aplicando directamente las ecuaciones de Bellman, pero siguen siendo la base teórica del aprendizaje por refuerzo [10].

2.5. Algoritmos de gradiente de políticas

El objetivo de los métodos de gradiente de política es modelar y optimizar la política directamente. La política suele modelarse con una función parametrizada respecto a $\theta, \pi_\theta(a \mid s)$. El valor de la función recompensa depende de la política modelada y varios algoritmos pueden aplicarse para optimizar θ y maximizar la recompensa. La **función recompensa** se define entonces como:

$$
J(\theta) \;=\; \sum_{s \in \mathcal{S}} d^\pi(s)\, V^\pi(s) \;=\; \sum_{s \in \mathcal{S}} d^\pi(s) \sum_{a \in \mathcal{A}} \pi_\theta(a \mid s)\, Q^\pi(s,a)
$$

con $d^\pi(s)$ la distribución estacionaria de una cadena de Markov para π_θ, es decir, la distribución de estados bajo la política π. De manera intuitiva, se puede pensar en π_θ como si al recorrer los estados de la cadena de

Markov, eventualmente se llegue a que la probabilidad de estar en cierto estado se vuelva constante.

$$d^{\pi}(s) \;=\; \lim_{t\to\infty} P\big(s_t = s \mid s_0, \pi_\theta\big)$$

es la probabilidad de que $s_t = s$ cuando se empieza en s_0 y se sigue la política π_θ a lo largo de t pasos.

Se espera entonces que los métodos basados en políticas sean más útiles en espacios continuos. Pues hay infinidad de acciones y estados para estimar. Los métodos basados en valores se vuelven poco viables computacionalmente por ese motivo. Por ejemplo, en *generalized policy iteration* (método basado en valores) el paso para mejorar la política es:

$$\arg\max_{a\in\mathcal{A}} Q^{\pi}(s,a)$$

el cual necesita analizar todo el espacio de acciones, sufriendo de la *maldición de dimensionalidad*. En cambio, si se usa *gradient ascent* (método basado en política), se puede modificar el valor de θ hacia la dirección que indique el gradiente $\nabla_\theta J(\theta)$ para encontrar el θ óptimo para π_θ, obteniendo la recompensa máxima [11].

Teorema del gradiente de política

Aunque sea mejor actualizar el valor de θ hacia la dirección del gradiente, no es sencillo de calcular, pues depende tanto de la selección de acciones como de la distribución estacionaria de los estados, ambas determinadas por π_θ. Como el entorno suele ser desconocido, es difícil estimar el efecto de una actualización de la política sobre la distribución de los estados.

Se puede lidiar con este problema mediante el teorema del gradiente de política (*policy gradient theorem*) [12]. Permite una reformulación de la derivada de la función objetivo para no involucrar la derivada de la distribución de estados $d^{\pi}(\cdot)$ y simplificar mucho el cálculo del gradiente $\nabla_\theta J(\theta)$. El gradiente se convierte en:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \nabla_\theta \sum_{s\in\mathcal{S}} d^{\pi}(s) \sum_{a\in\mathcal{A}} Q^{\pi}(s,a)\pi_\theta(a \mid s) \\
&\propto \sum_{s\in\mathcal{S}} d^{\pi}(s) \sum_{a\in\mathcal{A}} Q^{\pi}(s,a)\nabla_\theta \pi_\theta(a \mid s) \quad [11] \\
&\propto \mathbb{E}_{s\sim d^{\pi},\; a\sim\pi_\theta}\left[Q^{\pi}(s,a)\;\nabla_\theta \log \pi_\theta(a \mid s)\right]
\end{aligned}$$

Permite reformular el gradiente como una expectativa en la que no interviene la derivada de la distribución estacionaria de estados, y que puede estimarse mediante muestreo de trayectorias siguiendo la política. Para una explicación más detallada y la demostración del teorema se puede ver en [12], y para una explicación más intuitiva, se recomienda revisar [13].

3. Procesamiento de datos

3.1. Obtención y limpieza de datos

Se eligieron 10 empresas del índice S&P100 para invertir en ellas mediante la compra de acciones; 3 pertenecientes al sector tecnológico, 3 del sector de salud, 3 del sector financiero y 1 del sector industrial:

1. Nvidia: Empresa de software y semiconductores que diseña unidades de procesamiento de gráficos (GPU) para ciencia de datos y computación de alto rendimiento, así como unidades de sistema en chip (SoC) para la computación móvil y el mercado automotriz. Es líder mundial en hardware y software de inteligencia artificial. En 2022 anunció su chip de grado automotriz de última generación, Drive Thor y en 2023 comenzado a diseñar unidades de procesamiento central (CPU) para el sistema operativo Windows de Microsoft con la intención de empezar su venta en 2025.

2. Amazon: Ofrece servicios de comercio electrónico y servicios de programación en la nube (*Amazon Web Services(AWS)*). En 2025 anunciaron su nuevo chip cuántico Ocelot, el cual utiliza qubits gato.

3. Microsoft: Se considera una de las cinco grandes empresas estadounidenses de tecnología de la información. En 2022, Microsoft anunció un nuevo acuerdo de diez años con la Bolsa de Londres para productos que incluyen Microsoft Azure. Microsoft adquirió aproximadamente el 4 % de esta última empresa como parte del trato. En 2023 anunció un nuevo acuerdo de inversión con OpenAI y adquirió la compañía de videojuegos Activision Blizzard. En 2025 anunciaron su chip cuántico, Majorana 1, el cual usa superconductores topológicos.

4. Eli Lilly: Es una compañía farmaceutica, entre sus productos más famosos se encuentran el Prozac (antidepresivo), la cefalosporina y la eritromicina (antibioticos). Es una de las principales fabricantes de tratamientos médicos para el cáncer, afecciones cardiovasculares, desórdenes del sistema nervioso central y endocrino, la diabetes y enfermedades contagiosas. En Marzo del 2025 anunciaron que invertirán en México y America Latina, tanto financiando investigaciones como expandiendo su producción de medicamentos y fue aprobado un medicamento para la pérdida de peso en la India.

5. Amgen Inc: BusinessWeek la calificó como cuarta de todo el S&P 500 por ser la más orientada hacia el futuro. De acuerdo a analistas de *Yahoo Finance*, es la novena mejor acción del sector de salud en la cual invertir.

6. Bristol Myers Squibb: Es conocida por fabricar productos farmacéuticos de prescripción en varias áreas terapéuticas como cáncer, VIH/SIDA, enfermedades cardiovasculares, diabetes, hepatitis, artritis reumatoide y trastornos psiquiátricos. Desde 2019, con su colaborador ConcertAI, implementaron técnicas de ML e IA con sus investigadores para optimizar el descubrimiento de fármacos; priorizando la predicción computacional para identificar moléculas prometedoras de manera más eficiente.

7. Berkshire Hathaway Inc: Es una sociedad de cartera, es decir, posee partes parciales o totales de otras compañías y administra esos fondos. Son accionistas de empresas como *American Express*, Coca-Cola,

Johnson & Johnson, Apple y *Bank of America*.

8. Black Rock Inc: Es la administradora de activos más grande del mundo, con 9.42 billones de dólares en activos. Opera en 30 países y tiene clientes en 100 países. Es uno de los administradores de fondos indexados *Big Three*, los cuales poseen el 17.5 % de las acciones de todo el mundo. Su software *Aladdin* realiza un seguimiento de las carteras de inversión de las principales instituciones financieras y su división *BlackRock Solutions* proporciona servicios de gestión de riesgos financieros. En 2024 fue aprobado sus ETF para Bitcoin y Ethereum los cuales, en 2025 fueron lanzados en Europa.

9. JP Morgan: Es líder en inversiones bancarias, servicios financieros, gestión de activos financieros e inversiones privadas. Con activos financieros de más de 2.4 billones de dólares. Es actualmente la primera institución bancaria de Estados Unidos. La unidad de fondos de inversión libre es la más grande de Estados Unidos, el fondo JP Morgan US Technology ha tenido ganancias de mas del 30 % en los últimos 3 años.

10. Linde plc: Es la compañía más grande de fabricación y distribución de gases industriales; son utilizados en industrias como la salud, refinerías de petróleo, bebidas, aeroespacial y tratamiento de aguas. También diseñan y construyen plantas químicas a gran escala para la producción de gases industriales o gas natural licuado.

Se eligieron las acciones de acuerdo con los siguientes criterios:

- Hay datos históricos desde el 01-01-2000 o antes, para poder entrenar el modelo con suficientes datos.
- Han tenido una tendencia alcista al comparar su precio del 2020 con el actual.
- Se eligieron tres de las acciones con mayor peso sobre uno de los ETF que siguen al S&P100, el *iShares S&P 100 ETF*. En donde Nvidia tiene un peso de 8.56 %, Amazon de 5.62 % y Microsoft con 8.51 % sobre el capital total del ETF.
- Se analizó las noticias relevantes de los últimos 5 años; compra de otras empresas, firma de contratos importantes, registro de patentes, adquisición de bienes.

Se extrajeron los datos de 24 años de las 10 empresas antes mencionadas utilizando la librería de Python *yfinance*, obteniendo los precios de apertura, precio de cierre, precio mínimo, precio máximo y volumen de transacciones.

Figura 2: Gráfico de velas de Amazon (Enero-Diciembre 2024).

3.2. Matriz de correlación de retornos anuales

Se calculó la correlación de los retornos anuales de las acciones para cada par de acciones del portafolio con el objetivo de evaluar qué tan buena era la diversificación (aún sin considerar los porcentajes que tendría cada una de las acciones). Para esto, primero se calcularon los retornos diarios de cada acción utilizando el método pct_change de la librería *Pandas* sobre el precio de cierre diario. Después se agruparon los datos por año fiscal y se multiplicaron todos los cambios porcentuales para obtener los retornos anuales de cada acción:

Ticker	AMGN	AMZN	BLK	BMY	BRK-B	JPM	LIN	LLY	MSFT	NVDA
Date										
2000-12-31	0.015889	-0.825874	1.526316	0.147430	0.333711	-0.064751	-0.109159	0.419447	-0.627882	0.399863
2001-12-31	-0.117263	-0.304739	-0.007143	-0.275280	0.072642	-0.200000	0.245070	-0.156051	0.527378	3.083561
2002-12-31	-0.143515	0.745841	-0.055156	-0.546078	-0.040396	-0.339752	0.045611	-0.191495	-0.219623	-0.827952
2003-12-31	0.278237	1.785601	0.347970	0.235421	0.161783	0.530417	0.322486	0.107559	0.058801	1.015628
2004-12-31	0.038194	-0.158305	0.454717	-0.104196	0.042984	0.062075	0.155759	-0.193090	-0.023749	0.015517
2005-12-31	0.229306	0.064574	0.404090	-0.103045	-0.000170	0.017431	0.199547	-0.002819	-0.021332	0.551787
2006-12-31	-0.133781	-0.163097	0.400258	0.145344	0.248850	0.216931	0.120280	-0.079343	0.141874	1.024614
2007-12-31	-0.320158	1.347694	0.427255	0.007599	0.291871	-0.096273	0.495196	0.024760	0.192230	0.378817
2008-12-31	0.243540	-0.446459	-0.381227	-0.123303	-0.321368	-0.277663	-0.330853	-0.245739	-0.453933	-0.762787
2009-12-31	-0.020433	1.623245	0.730898	0.086022	0.022402	0.321598	0.352931	-0.113236	0.567901	1.314746
2010-12-31	-0.029521	0.338091	-0.179242	0.048713	0.218959	0.017999	0.188769	-0.018762	-0.084318	-0.175589
2011-12-31	0.169581	-0.038333	-0.064750	0.330816	-0.047560	-0.216172	0.119723	0.186073	-0.069867	-0.100000
2012-12-31	0.342470	0.449278	0.159728	-0.075199	0.175622	0.322406	0.023854	0.186718	0.028891	-0.115440
2013-12-31	0.323434	0.589628	0.530985	0.630868	0.321739	0.329998	0.188031	0.034063	0.400599	0.306688
2014-12-31	0.396301	-0.221771	0.129839	0.110630	0.266447	0.070109	-0.003615	0.352745	0.241647	0.251561
2015-12-31	0.019085	1.177831	-0.047656	0.165340	-0.120613	0.055129	-0.209633	0.221336	0.194403	0.643890
2016-12-31	-0.099304	0.109456	0.117526	-0.150458	0.234323	0.306830	0.144434	-0.127107	0.120043	2.238471
2017-12-31	0.189385	0.559564	0.349950	0.048597	0.216223	0.239309	0.319908	0.148334	0.376569	0.812816
2018-12-31	0.119436	0.284317	-0.235327	-0.151762	0.030068	-0.087152	0.008792	0.370116	0.187398	-0.310078
2019-12-31	0.238352	0.230277	0.279721	0.234898	0.109315	0.427986	0.364394	0.135759	0.552624	0.762547
2020-12-31	-0.046252	0.762561	0.435329	-0.033650	0.023709	-0.088450	0.237717	0.284638	0.410400	1.219295
2021-12-31	-0.021529	0.023768	0.268897	0.005159	0.289516	0.246164	0.314675	0.635987	0.512094	1.252853
2022-12-31	0.167445	-0.496152	-0.226015	0.153970	0.033110	-0.153142	-0.058453	0.324452	-0.286929	-0.503111
2023-12-31	0.096634	0.808810	0.145591	-0.286866	0.154613	0.268456	0.259151	0.593374	0.568009	2.388668
2024-12-31	-0.095063	0.443925	0.262762	0.102319	0.270902	0.409230	0.019381	0.324367	0.120891	1.711724

Figura 3: Retornos anuales de las acciones del portafolio

Con los datos anteriores se realizó el cálculo de correlación de los retornos anuales de las acciones:

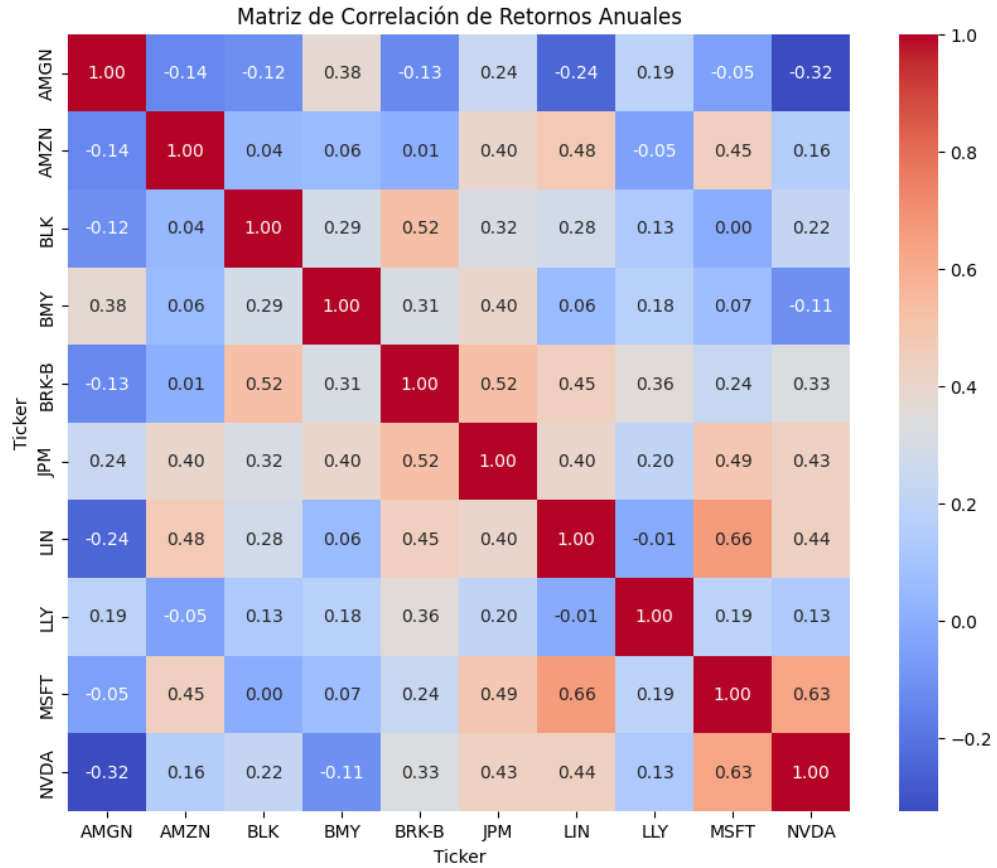

Figura 4: Matriz de correlaciones de las acciones del portafolio

En donde podemos ver que tienen relaciones muy débiles, lo que no indica una relación lineal en la mayoría

de los pares, a excepción de Nvidia y Microsoft.

4. PPO

PPO es una familia de métodos de optimización de políticas que utilizan múltiples épocas de ascenso estocástico por gradiente para realizar cada actualización de la política. Estos métodos tienen la estabilidad y fiabilidad de los métodos de región de confianza, como su predecesor TRPO, pero son mucho más simples de implementar. [14]

El algoritmo destaca por el objetivo sustituto recortado (*Clipped Surrogate Objective*), el cual es un reemplazo directo del objetivo de gradiente de política, diseñado para mejorar la estabilidad del entrenamiento al limitar los cambios que realiza en su política en cada paso. Los métodos de gradiente de política estándar como REINFORCE utilizaban el objetivo $L^{PG}(\theta) = \hat{\mathbb{E}}_t \left[\log \pi_\theta(a_t|s_t)\hat{A}_t\right]$ para optimizar la red neuronal, donde $\hat{A}_t$ puede ser la recompensa con descuento, o la función de ventaja definida anteriormente. Al realizar un paso de ascenso de gradiente sobre esta función de pérdida con respecto a los parámetros de la red, se incentivarán las acciones que condujeron a una mayor recompensa.

Los métodos de gradiente de política estándar utilizan la probabilidad logarítmica de la acción $log_\pi(a|s)$ para medir la influencia de las acciones en las recompensas. Otra función, introducida en [15] utilizan la razón entre la probabilidad de la acción bajo la política actual $\pi(a|s)$ y la probabilidad bajo la política anterior: $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$. Este ratio $r(\theta)$ será mayor que 1 cuando la acción sea más probable bajo la política actual que bajo la política anterior; y estará entre 0 y 1 cuando la acción sea menos probable en la política actual comparada con la anterior. Para construir una función objetivo con $r(\theta)$, se reemplaza por el término logarítmico $log_\pi(a|s)$. Esto es lo que hace el predecesor de PPO, TRPO *(Trust Region Policy Optimization)* $L^{TRPO}(\theta) = \hat{\mathbb{E}}_t \left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t\right] = \hat{\mathbb{E}}_t \left[r_t(\theta)\hat{A}_t\right]$.

TRPO presenta un problema al analizar que, si la acción con la nueva política es mucho más probable, entonces $r(\theta)$ sería muy grande y la actualización de la política sería demasiado brusca. Es por eso que implementan mecanismos adicionales, como restricciones basadas en la Divergencia KL, que limitan la magnitud del cambio en la política y ayudan a garantizar una mejora monótona. ¿Y si, en lugar de añadir los mecanismos adicionales, pudiéramos integrar directamente las propiedades de estabilización en la función objetivo? Es esto mismo lo que realiza el algoritmo PPO. Obtiene los mismos beneficios de rendimiento que TRPO, evitando su complejidad, optimizando este objetivo sustituto recortado:

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

en donde $r_t(\theta)\hat{A}_t$ definen la dirección y magnitud del gradiente en el espacio de parámetros θ durante la optimización como lo hacía en TRPO, pero ahora, si este valor es muy grande, PPO tomará el mínimo

(segundo término), el cual es una versión de $r(\theta)$ en la que está limitada entre (1-ϵ) y (1+ϵ). El valor de ϵ suele ser de 0.2 de acuerdo al artículo original.

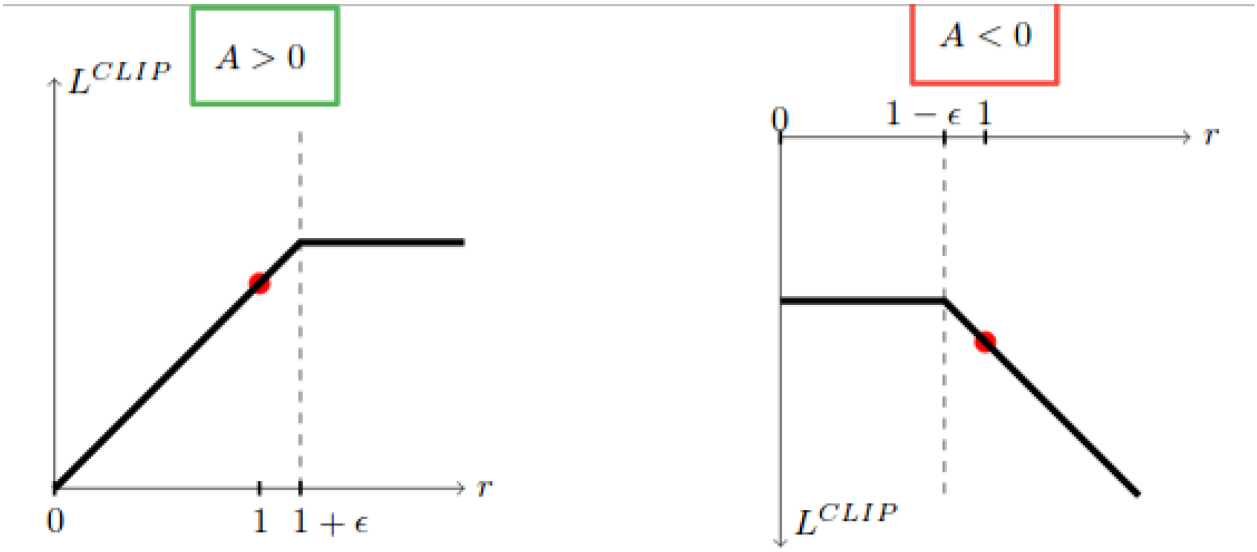

Figura 5: Un único paso de tiempo de la función sustituta L^{CLIP} en función del ratio de probabilidad r, para ventajas positivas (izquierda) y ventajas negativas (derecha). El círculo rojo en cada gráfico indica el punto inicial de la optimización.

En el lado izquierdo, el valor de r se *recorta* cuando es demasiado alto. Esto ocurre cuando una acción se vuelve mucho más probable bajo la política actual que bajo la política anterior. En tales casos, evitamos dar pasos excesivamente grandes (ya que es solo una aproximación local basada en una muestra de nuestra política, y perdería precisión si se avanza demasiado). Así, el *clipping* limita el crecimiento del objetivo. La región plana bloquea el gradiente (es decir, el gradiente se vuelve 0).

En el lado derecho, donde la acción tiene un efecto negativo estimado, el *clipping* se activa cerca de 0, lo que indica que la acción es muy improbable bajo la política actual. Esta región plana previene actualizaciones bruscas que reducirían drásticamente la probabilidad de la acción después de haber dado ya un paso significativo en esa dirección.

En resumen, si la acción fue buena $A > 0$ y se volvió mas probable en el último paso del gradiente, no se debe actualizar demasiado o podría empeorar la política (por lo que el gradiente se hace 0), si la acción se volvió menos probable, se puede deshacer el último paso todo lo que quieras. Y si la acción fue mala $A < 0$ y se volvió menos probable, no se debe hacer mucho menos probable o la política podría empeorar, pero si se volvió mas probable se puede deshacer ese último paso (para arreglar el error).

A diferencia de los métodos clásicos de policy gradient, y gracias a la *Clipped Surrogate Objective*, PPO permite ejecutar múltiples épocas de ascenso de gradiente sobre las mismas muestras sin provocar actualizaciones bruscas en la política. Maximizando el aprovechamiento de los datos y reduciendo la ineficiencia muestral.

PPO tiene N actores paralelos generando trayectorias en el entorno, y los datos se dividen en mini-batches y se entrenan durante K épocas usando la función de *clipping*, permitiendo aprender más de cada muestra.

Para la regularización de la política, el algoritmo PPO estándar utiliza el objetivo con recorte (clipped objective); para la parametrización de la política, el algoritmo PPO estándar utiliza una distribución Gaussiana en espacios de acción continuos y la función Softmax en espacios de acción discretos. Además, sigue un marco clásico de Actor-Critic con cuatro componentes:

- Inicialización: inicializa los atributos y redes relacionados.
- Exploración: explora transiciones mediante la interacción entre la red del Actor y el entorno.
- Cálculo: calcula las variables relacionadas, como el término de razón (ratio term), la recompensa exacta y la ventaja estimada.
- Actualización: actualiza las redes del Actor y el Critic basándose en la función de pérdida y la función objetivo.

Algorithm 1 PPO-Clip

Input: initial policy parameters θ_0, initial value function parameters ϕ_0
for $k = 0, 1, 2, \ldots$ **do**
    Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go $\hat{R}_t$.
    Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based on the current value function V_{ϕ_k}.
    Update the policy by maximizing the PPO-Clip objective:

$$\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} \hat{A}^{\pi_{\theta_k}}(s_t, a_t),\ g(\epsilon, \hat{A}^{\pi_{\theta_k}}(s_t, a_t)) \right),$$

    typically via stochastic gradient ascent with Adam.
    Fit value function by regression on mean-squared error:

$$\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k|T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t \right)^2,$$

    typically via some gradient descent algorithm.
end for

Paso 1: inicializa las redes del Actor y del Crítico, y el parámetro ϵ.

Paso 3: recopila un lote de trayectorias usando la política más reciente del Actor.

Paso 4: calcula la recompensa exacta para cada trayectoria en cada paso.

Paso 5: calcula la ventaja estimada para cada trayectoria usando la red del Crítico más reciente.

Paso 6: actualiza los parámetros del Actor mediante ascenso por gradiente estocástico, maximizando la función objetivo PPO-Clip durante K épocas. Al comenzar la optimización, la política antigua es igual a la política actualizada, lo que implica que el cociente es 1. A medida que se actualiza la política, el cociente comienza a alcanzar los límites del recorte (clipping).

Paso 7: actualiza los parámetros del Crítico mediante descenso por gradiente sobre el error cuadrático medio.

Se utilizó el entorno *env_portfolio_optimization* de la librería *FinRL* para entrenar al agente de PPO. El **estado** se obtiene con la función *_get_state_and_info_from_time_index* la cual obtiene el estado y la información correspondiente a ese tiempo y actualiza los atributos de la *data* con la información del paso actual en la simulación.

El *espacio de acciones* es un *Box* 1-dimensional de longitud $n+1$, con n acciones del portafolio y un espacio para el dinero en efectivo con el que se inicia y el que se obtiene después de cada acción de venta.

La función de transición en el entorno está dada por la función *step* y devuelve $(state, reward, terminal, info)$ con *state* el siguiente estado simulado, *reward* la recompensa de la última acción realizada e $info$, con información acerca del ultimo estado como el ratio de Sharpe, el número de paso, el *max drawdown*, el retorno total hasta el momento, y el porcentaje del capital que está en cada valor y en efectivo. Dentro del mismo *step* se calcula la *recompensa* con *rate_of_return*:

```
# define portfolio return
rate_of_return = (
    self._asset_memory["final"][-1] / self._asset_memory["final"][-2]
)
portfolio_return = rate_of_return - 1
portfolio_reward = np.log(rate_of_return)
```

Es decir, calcula el factor de retorno, después lo transforma en retorno porcentual y calcula la recompensa logarítmica porque es mas estable pues evita divisiones entre números pequeños o grandes, y se acumula linealmente, permitiendo sumar retornos logarítmicos directamente a lo largo del tiempo. Otra ventaja de utilizar el logaritmo es que es simétrico, pues si se pierde el 50 % del portafolio, se necesitaría ganar el 100 % para volverr al estado inicial, pero el logaritmo es simétrico pues $log(\frac{1}{n}) = -log(n)$.

5. Resultados

Se entrenó el modelo de *Proximal Policy Optimization* utilizando el ambiente *env_portfolio_optimization* de la librería *FinRL*, de Python. Se utilizaron datos del 3 de enero de 2017 al 31 de diciembre de 2022 para entrenar al agente y después se probó utilizando los datos del 1 de enero de 2023 al 1 de enero de 2025.

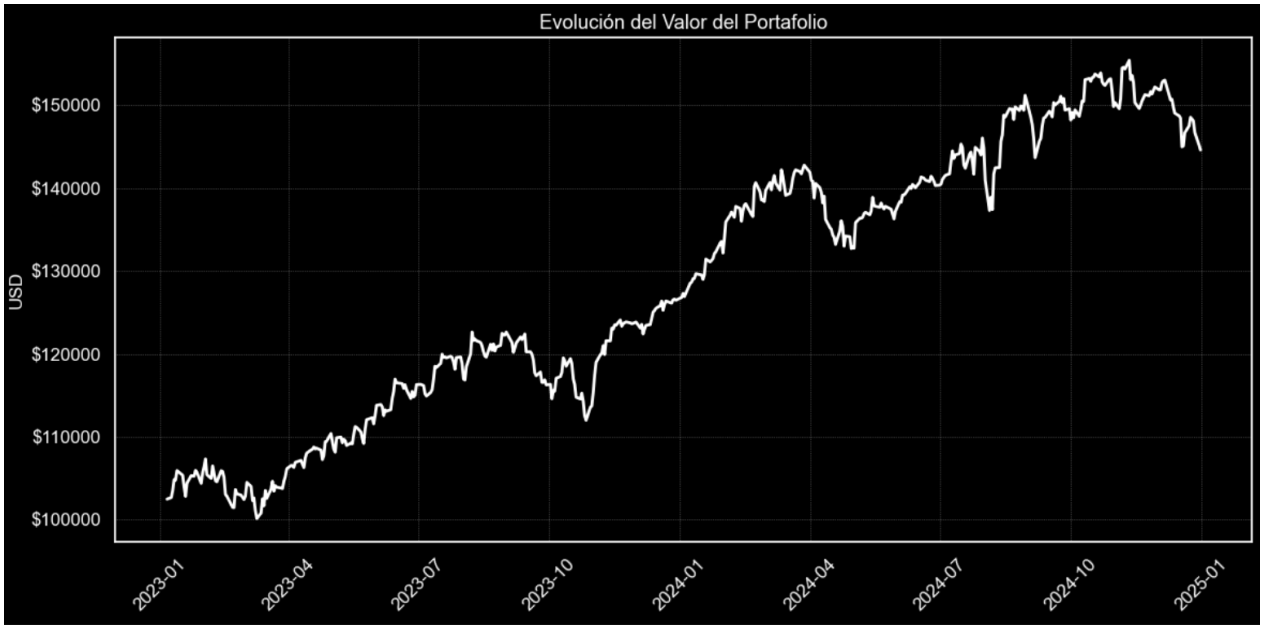

Figura 6: Evolución del rendimiento del portafolio al utilizar PPO.

El retorno final del portafolio, después de 500 dias fue de 144,615.65 iniciando con un capital de 100,000.00 dólares, por lo que se obtuvo un retorno acumulado del 144.6 %. Para poder compararlo con ganancias de otros portafolios y con otros tiempos, es importante anualizar el rendimiento.

Retorno total $= \frac{144{,}615{,}65}{100{,}000{,}00} - 1 = 0{,}4461$ Luego, para obtener el retorno anualizado usando la fórmula de interés compuesto: Retorno anualizado $= (1 + \text{Retorno total})^{\frac{1}{\text{num. años}}} - 1 = (1 + 0{,}4461)^{\frac{1}{2}} - 1 = 0{,}202$. Por lo que el retorno anual de la estrategia implementada por el agente es del 20 %, superior al retorno anual promedio del S&P500, el cual es del 9.81 %.

Se utilizaron métricas de riesgo y desempeño para evaluar la eficiencia y estabilidad del agente, obteniendo los siguientes resultados:

Métrica	**Valor**
Maximum Drawdown	-0.08
Sharpe Ratio	1.50
Sortino Ratio	0.01
Calmar Ratio	2.10
Omega Ratio	1.27
Daily VaR (5 %)	-0.01

- *Maximum Drawdown*: mide la mayor caída porcentual desde un pico histórico hasta un valle, antes de recuperarse. Entonces, en su peor momento, el agente perdió 8 % de su valor desde su máximo anterior.
- *Sharpe Ratio*: mide numéricamente la relación entre la rentabilidad y la volatilidad histórica de un fondo de inversión. Generalmente se considera un Ratio de Sharpe mayor a 1 bueno y uno mayor a 2

excelente.

- *Sortino ratio*: mide el rendimiento ajustado al riesgo de una inversión, activo, cartera de valores o estrategia. El valor de 0.01 sugiere que la estrategia no compensa adecuadamente los momentos de pérdidas bruscas.
- *Calmar Ratio*: es una medida que evalúa los rendimientos ajustados al riesgo de una inversión, calculada dividiendo su Tasa de Crecimiento Anual Compuesta (CAGR) por su máxima caída (drawdown). Un valor de 2.10 indica que la estrategia genera más del doble de rentabilidad en comparación con su peor pérdida histórica.
- *Omega Ratio*: mide la relación ponderada por la probabilidad de ganancias versus pérdidas para una inversión determinada. Un valor de 1.27 sugiere que, aunque la estrategia genera más ganancias que pérdidas, aunque los valores óptimos son mayores a 1.5.
- *Daily VaR (5 %)*: estima la pérdida máxima diaria esperada con un 95 % de confianza. Un valor de 0.01 implica que, en condiciones normales, El 95 % de los días, las pérdidas no superarán el 1 %.

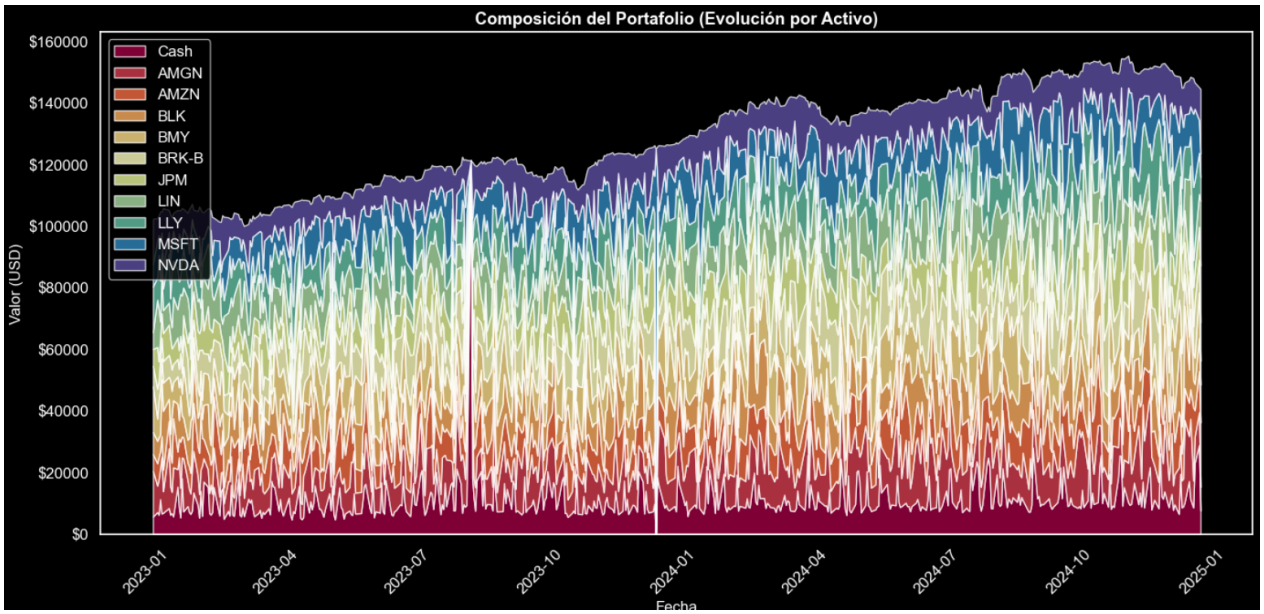

Figura 7: Porcentaje del dinero invertido en el activo a tiempo t.

6. Conlcusiones

En el presente trabajo se implementó un algoritmo de aprendizaje por refuerzo (PPO) y fue capaz de batir al mercado de referencia. Por lo tanto, se ha demostrado que las técnicas derivadas del aprendizaje por refuerzo permiten optimizar los pesos de un portafolio de inversión con éxito.

Como futuro trabajo se podría comparar los resultados de este agente con estrategias como comprar y mantener o con otros algoritmos de aprendizaje por refuerzo como DDPG, A2C o TD3. También se podría combinar con técnicas de selección de acciones, por ejemplo, con redes complejas.

Referencias

1. MATTHEW, F. D. *Machine learning in finance: From theory to practice* (Springer, 2021).
2. Sutton, R. S., Barto, A. G. *et al.* Reinforcement learning. *Journal of Cognitive Neuroscience* **11,** 126-134 (1999).
3. Achiam, J. *Spinning Up in Deep Reinforcement Learning* https://spinningup.openai.com/en/latest/spinningup/rl_intro.html. Accessed: 2025-04-17. 2018.
4. Chen, L. y Gao, Q. *Application of deep reinforcement learning on automated stock trading* en *2019 IEEE 10th International Conference on Software Engineering and Service Science (ICSESS)* (2019), 29-33.
5. Deng, Y., Bao, F., Kong, Y., Ren, Z. y Dai, Q. Deep direct reinforcement learning for financial signal representation and trading. *IEEE transactions on neural networks and learning systems* **28,** 653-664 (2016).
6. Yang, H., Liu, X.-Y., Zhong, S. y Walid, A. *Deep reinforcement learning for automated stock trading: An ensemble strategy* en *Proceedings of the first ACM international conference on AI in finance* (2020), 1-8.
7. Zhao, L. *et al.* Stock market as temporal network. *Physica A: Statistical Mechanics and its Applications* **506,** 1104-1112 (2018).
8. Peralta, G. y Zareei, A. A network approach to portfolio selection. *Journal of Empirical Finance* **38,** 157-180 (2016).
9. Achiam, J. *Spinning Up in Deep Reinforcement Learning* Accedido: 2025-04-17. 2018. https://spinningup.openai.com/en/latest/spinningup/rl_intro2.html.
10. Weng, L. A (Long) Peek into Reinforcement Learning. *lilianweng.github.io.* https://lilianweng.github.io/posts/2018-02-19-rl-overview/ (2018).
11. Weng, L. Policy Gradient Algorithms. *lilianweng.github.io.* https://lilianweng.github.io/posts/2018-04-08-policy-gradient/ (2018).
12. Sutton, R. S. y Barto, A. G. *Reinforcement Learning: An Introduction* 2nd. Available online via Carnegie Mellon University course 10-703. https://www.andrew.cmu.edu/course/10-703/textbook/BartoSutton.pdf (MIT Press, 2018).
13. Nanda, A. *Policy Gradient Theorem Explained: A Hands-On Introduction* DataCamp Tutorial. https://www.datacamp.com/tutorial/policy-gradient-theorem (2025).
14. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. y Klimov, O. *Proximal Policy Optimization Algorithms* 2017. arXiv: 1707.06347 [cs.LG]. https://arxiv.org/abs/1707.06347.
15. Kakade, S. y Langford, J. *Approximately optimal approximate reinforcement learning* en *Proceedings of the nineteenth international conference on machine learning* (2002), 267-274.