



ITESO, Universidad Jesuita de Guadalajara

Proyecto #02 – Introducción a Trading

Santiago Reyes Castillo

Ingeniería Financiera

Microestructuras de Mercado y Trading

7 de octubre, 2025

Índice

Introducción	3
Metodología.....	4
Preprocesamiento de Datos	4
Estrategia (Indicadores y Parámetros).....	4
Backtesting.....	7
Métricas.....	8
Optimización	8
Train, Test & Validation.....	9
Análisis y Comparación de Resultados	10
Comportamiento del Portafolio	10
Métricos de desempeño	12
Limitaciones	12
Análisis de Riesgo.....	13
Conclusiones	14
Fuentes de Consulta.....	15

Introducción

Conforme pasa el tiempo, el mercado financiero ha ido creciendo y expandiéndose cada vez más. Existen cada vez más tipos de activos financieros, derivados y muchos otros tipos de componentes del mercado en los que uno puede depositar su dinero para especular, hacer arbitraje o cubrirse, dependiendo de los objetivos de cada individuo. Sin embargo, siempre ha existido la incógnita de cómo hacer dinero de manera “infalible” con ellos. Por eso, hemos intentado utilizar todos los medios posibles para intentar captar información del mercado que nos ayude a explicarla, y por ende explotar el mercado para nuestro beneficio.

Particularmente, y el objeto de este proyecto, son los indicadores de análisis técnico tan utilizados para hacer trading de distintos tipos. El análisis técnico comprende toda aquella información que obtenemos a partir de la conducta del mercado, de los precios y lo que vemos que sucede, y cómo intentamos darle un significado y justificarlo con hechos y pruebas.

Usualmente, se dice que es imposible asegurar una ganancia en un mercado como el mercado de acciones, ya que distintos supuestos o hipótesis prueban que el precio de los activos sigue un movimiento estocástico, por ende, aleatorio e imposible de predecir. Pero a pesar de ello, se han derivado estrategias que intentan romper estos supuestos y generar una forma de operar tal que aumentes tus probabilidades de generar ganancias y disminuyas las de tener pérdidas.

En este proyecto se desarrollará una estrategia utilizando indicadores técnicos, para analizar si es que es posible, dada una estrategia y un activo dado, generar operaciones para cumplir este objetivo, utilizando distintos métodos numéricos y de análisis para ello.

Metodología

A continuación, se presenta una descripción de la metodología utilizada en este proyecto. Este proyecto se llevó a cabo exclusivamente en el lenguaje de programación de Python y diversas librerías para éste. A su vez, fue usado Git como software de historial de versiones para ir manejando los avances del proyecto poco a poco. Se describirá el proceso comenzando por el preprocesamiento de los datos y el contexto de estos, para luego definir la estrategia de trading que fue utilizada, sus parámetros y funcionamiento. Posteriormente, se explicará el backtesting que fue diseñado para probar esta estrategia y por qué difiere, por ejemplo, de las ventas en corto que suelen hacerse normalmente en un mercado tradicional.

Además, se describe el proceso por el cual se entrenaron los datos y se probaron en un entorno real simulado con una partición en conjuntos de Train, Test y validation.

Preprocesamiento de Datos

Los datos utilizados para este proyecto son el valor del “BitCoin” en dólares (BTC/USD), con intervalos de tiempo de cada hora, desde agosto del 2017 a septiembre del 2025, aproximadamente 8 años de datos equivalentes a más de 7,000 entradas individuales, lo cual nos da bastante información para utilizar como base.

El archivo de datos original requiere mínimo preprocesamiento contiene datos de fecha, precio de cierre, high, low y volúmenes. Simplemente se debe de invertir el orden de los datos ya que originalmente están de más reciente a más antiguo, al invertirlo queda utilizable.

Para la importación de los datos se creó una función llamada “download_data”. Ésta es muy corta, recibe el nombre del archivo, importa los datos en un DataFrame de la librería Pandas, los reordena, y regresa un DataFrame utilizable.

Estrategia (Indicadores y Parámetros)

Luego de importar los datos, se define la estrategia que se planea utilizar. Esto debido a que se genera la información de los indicadores de análisis técnicos, y se añade al DataFrame original. En este proyecto, fueron utilizados 3 indicadores de análisis técnico distintos:

- RSI (Relative Strength Index)
- MACD (Moving Average Convergence Divergence)
- ADX (Average Directional Index)

El RSI es un indicador de análisis técnico que mide la fuerza y velocidad con la que se mueve el precio de un activo, para identificar momentos de sobrecompra y sobreventa en una escala del 0 al 100. Si el RSI es alto, indica que el activo se encuentra en condiciones de sobrecompra y si es bajo, de sobreventa. Este indicador nos puede dar una idea de en qué momento debemos entrar a una posición y en qué dirección.

Se calcula de la siguiente manera:

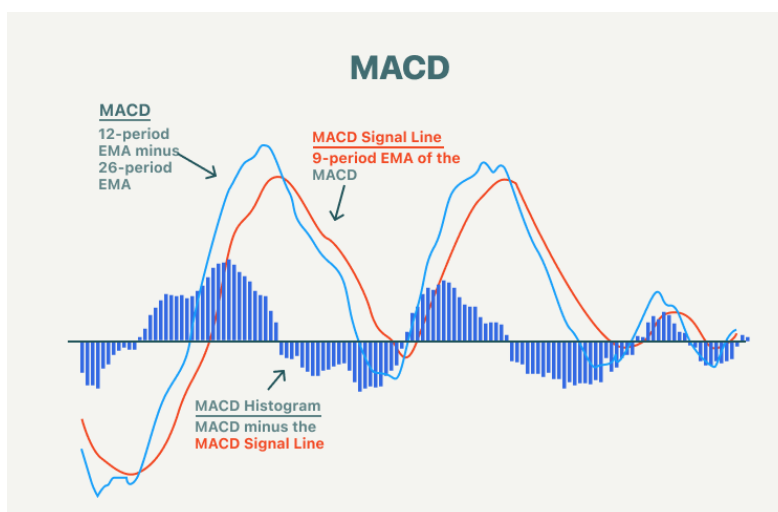
$$RSI = \frac{100}{1 + RS}$$
$$\text{where } RS = \frac{\text{Average Gain}}{\text{Average Loss}}$$

Los parámetros que se utilizaron para modelar la estrategia para este proyecto fueron límites para el RSI, sobre los cuales asumimos que el indicador nos da una entrada de compra y venta, y el periodo de tiempo (ventana) que se utiliza para calcular el indicador.

El MACD otro indicador que nos puede dar información valiosa sobre el estado de un activo. Este indicador se compone de 3 componentes (las tres medias móviles exponenciales), una denominada de ventana corta y otra de ventana larga, y la señal, compuesta por las diferencias entre ambas medias.

Este indicador nos da información valiosa del cambio de tendencia de un activo. Al graficar las primeras dos medias móviles, podemos observar que convergen y divergen cuando el precio cambia de tendencia, por lo que, reaccionando a estos cruces entre las medias, podemos obtener una entrada de compra o de venta.

Un ejemplo de un MACD en una gráfica es la siguiente figura:



En este caso, los parámetros a modelar fueron las ventanas de cada media móvil (larga, corta y señal) para adecuarse a los datos.

El ADX nos describe qué tan fuerte es la tendencia en el precio de un activo dado. Suele considerarse un indicador muy confiable para explicar este comportamiento en específico de un activo. Se puede utilizar con cualquier activo y funciona de la siguiente manera:

Primero se calcula de DMI, que se parte en positivo (+DMI) y negativo (-DMI) con la siguiente fórmula

Plus directional movement (+DM) = Current High - Previous High

Minus directional movement (-DM) = Previous Low - Current Low

La diferencia entre ambos DMI's "suavizados" nos indica en qué dirección va la tendencia, mientras que el ADX nos indica la fuerza de la misma. Entonces el ADX se calcularía de la siguiente manera, donde Smoothed es una media móvil con una ventana dada para suavizar las tendencias y ATR es un rango que normaliza la dirección de los movimientos utilizando su volatilidad:

$$+DI = \left(\frac{\text{Smoothed} + DM}{ATR} \right) \times 100$$

$$-DI = \left(\frac{\text{Smoothed} - DM}{ATR} \right) \times 100$$

$$DX = \left(\frac{|+DI - -DI|}{|+DI + -DI|} \right) \times 100$$

$$ADX = \frac{(\text{Prior ADX} \times 13) + \text{Current ADX}}{14}$$

Para el ADX, utilizamos como parámetro principal la ventana sobre el cual se calcula los datos, y así adecuarla a nuestros datos.

Los indicadores de análisis técnicos fueron calculados utilizando la librería "ta" de Python, por lo que serán funciones de esta las encargadas de ofrecer la información de estos en el proyecto.

Para ello, se declaró una función llamada “calculate_signals”, que recibe los parámetros mencionados para cada indicador, así como el DataFrame de la función anterior. Luego, calcula cada uno de los indicadores y los añade al DataFrame, para luego generar señales en base a la siguiente estrategia por cada indicador:

- RSI

Si el RSI en una dada entrada de la tabla, sobrepasa el límite superior fijado por el parámetro, se emite una señal de venta. Si está por debajo del límite inferior, se emite una señal de compra.

Todas las señales de compra se guardan con un valor de Verdadero o Falso para cada indicador por cada entrada de los datos.

- MACD

Para el MACD se evalúa si la señal del indicador es mayor al indicador mismo. Cuando la señal está por encima, indica un movimiento alcista (compra), y cuando está por debajo, uno bajista (venta).

- ADX

Finalmente, para el ADX, se evalúa primero si éste está por encima de 25 (indica una tendencia clara) y luego dependiendo de la dirección dictada por el +DI y -DI, se decide si la señal es de compra o venta. Si +DI es mayor a -DI, será una compra, y viceversa. Todas estas señales serán las que nos darán la pauta de las operaciones que se harán en el backtesting.

Backtesting

Para el backtesting, simulamos un entorno lo más real posible, tomando en cuenta comisiones y una cantidad de capital inicial dada. No tomamos en cuenta la posibilidad de apalancarnos y, al utilizar BitCoin, tenemos que hay datos todos los días del año, sin días feriados.

Para hacer el backtesting, definimos una clase en Python, donde guardaríamos la cada operación que podemos tener abierta. Luego, ejecutamos un ciclo que itere sobre cada entrada de los datos, en el DataFrame. En este ciclo, lo primero que ocurre es que se evalúa si se debe cerrar una posición dada, long o short, y se obtiene su beneficio o pérdida, agregándolo al efectivo que se tiene. Cabe mencionar que la simulación se llevó a cabo con 1 millón de dólares como inversión inicial. Si el precio del activo sobrepasaba o estaba por debajo del stop loss o take profit de cada operación, se cierra y se añade el efectivo.

Luego, se analiza cada entrada de datos para, con las señales previamente generadas, abrir operaciones de cada tipo. Esta parte es sencilla, puesto que únicamente se “compra” el activo para ambos long y short, y se descuenta del efectivo, restándole claro comisiones, siempre y cuando el valor del precio multiplicado por el parámetro de cuántas acciones se van a comprar por operación sea menor al efectivo disponible.

Posteriormente se valúa el portafolio para analizar el rendimiento histórico del mismo a lo largo de toda la estrategia. Para ello, en el caso del long, simplemente se calcula, por cada posición abierta, cuánto vale el activo en ese momento y se suma al valor del portafolio. Sin embargo, para el short, tomamos como supuesto que como no estamos efectuando una venta en corta del activo, si no una posición larga en su inverso, es decir, USD/BTC. De esta manera la pérdida en BTC/USD, se refleja como una ganancia en el opuesto, y podemos modelar de manera más práctica una posición corta en el activo. Entonces, para valorar cada posición, le restamos valor inicial de la posición que tomamos, el valor actual de las posiciones. A esto le añadimos el efectivo disponible, y así obtenemos un récord histórico de cómo se comportó el portafolio.

Finalmente, al finalizar el backtest, se cierran todas las operaciones con el último precio de cierre disponible, y se les da valor para obtener el valor final del portafolio.

Métricos de Desempeño

Para analizar cada portafolio dado, se utilizaron distintas métricas de evaluación, que se calcularon con el récord histórico producto del backtesting. Fueron utilizados los siguientes indicadores

- Rendimiento Anual: Refleja cómo hubiera crecido, relativo a la inversión inicial, el valor del portafolio, por año.
- Volatilidad Anual: Indica la dispersión posible de los valores del portafolio, a mayor volatilidad, mayores posibilidades de y rangos de variación en el mismo.
- Max Drawdown: Indica cuál fue la pérdida más grande en un movimiento dado del valor del portafolio. Es decir, después de un pico, cuál fue la pérdida más grande observada.
- Calmar Ratio: Se calcula como la razón de cambio del Rendimiento Anual, entre el Max Drawdown, y nos indica qué tan bueno es el rendimiento del portafolio dado su riesgo (su pérdida más grande).
- Sortino Ratio: Una variación del ratio de Sharpe, que, en lugar de utilizar la volatilidad total del portafolio, únicamente utiliza la volatilidad de las pérdidas. Nos ayuda a evaluar el rendimiento de un portafolio dado el riesgo “perjudicial”.
- Win Rate: Nos indica, de todas las operaciones que se ejecutaron, cuántas generaron una ganancia.

Las métricas se calcularon con una función que recibe el histórico del portafolio, y regresa cada una de estas métricas, de forma anualizada.

Optimización

La optimización consiste en obtener los mejores hiperparámetros del modelo, aquellos parámetros de las señales de los indicadores, y del backTest. Fueron los siguientes:

- Ventana, límite inferior y superior del RSI (3)
- Ventana larga, corta y de señal del MACD (3)
- Ventana de cálculo del ADX (1)
- Stop Loss, Take Profit y Número de acciones por operación, para el BackTest (3)

En total, se optimizaron 10 parámetros para esta estrategia.

Para efectuar esta operación, se utilizó el algoritmo de Optuna. Optuna es también una librería de Python que se encarga, a partir de métodos como Optimización Bayesiana y Parzen Tree Estimators, ofrecer un algoritmo eficiente de optimización, que se ajusta a sí mismo conforme ocurren las pruebas entre conjuntos distintos de parámetros. Es un algoritmo eficiente, pero al ser muchos parámetros puede tardar un rato. En este proyecto se usaron 50 pruebas de parámetros para la optimización, es decir, se probaron 50 conjuntos distintos de parámetros para encontrar el mejor. Esto debido a que durante la prueba de optimización, se encontró que de hacer muchas más pruebas no resultaba en un mejor modelo, si no en un sobreajuste de los parámetros a los datos, lo cual no es una estrategia válida para diseñar un modelo que se pretende utilizar. Además, como medida de evasión de sobreajuste, se penalizaron resultados demasiado buenos, justificando que podrían ser un sobreajuste.

La métrica principal que maximizó el algoritmo de optimización fue el ratio de Calmar.

Train, Test & Validation

Para profundizar en evitar un sobreajuste, y además probar la capacidad de la estrategia por funcionar en distintos entornos, se dividió el conjunto de datos en 3, Train, Test y Validation. El conjunto Train es sobre el cual se optimizaron los parámetros, para luego evaluarse en el resto de los conjuntos y verificar sus resultados. Además, el conjunto de datos de Train se optimizó siguiendo un proceso de análisis de Walk-Forward.

Usualmente, en modelos de aprendizaje estadístico, se emplean distintas estrategias para verificar que el entrenamiento de un modelo funcione correctamente en todos los conjuntos de datos. Para ello, se suele partir el conjunto de entrenamiento, y se verifica que la métrica a optimizar sea relativamente uniforme entre la evaluación de cada segmento. Esto normalmente se hace con K-fold o N-folds, en donde se seleccionan secciones al azar del código en orden indistinto. Sin embargo, el precio de un activo se comporta como una serie de tiempo, por lo que eliminar el factor de tiempo de la serie impacta de manera negativa, queremos tener información ordenada que represente correctamente la población total de información.

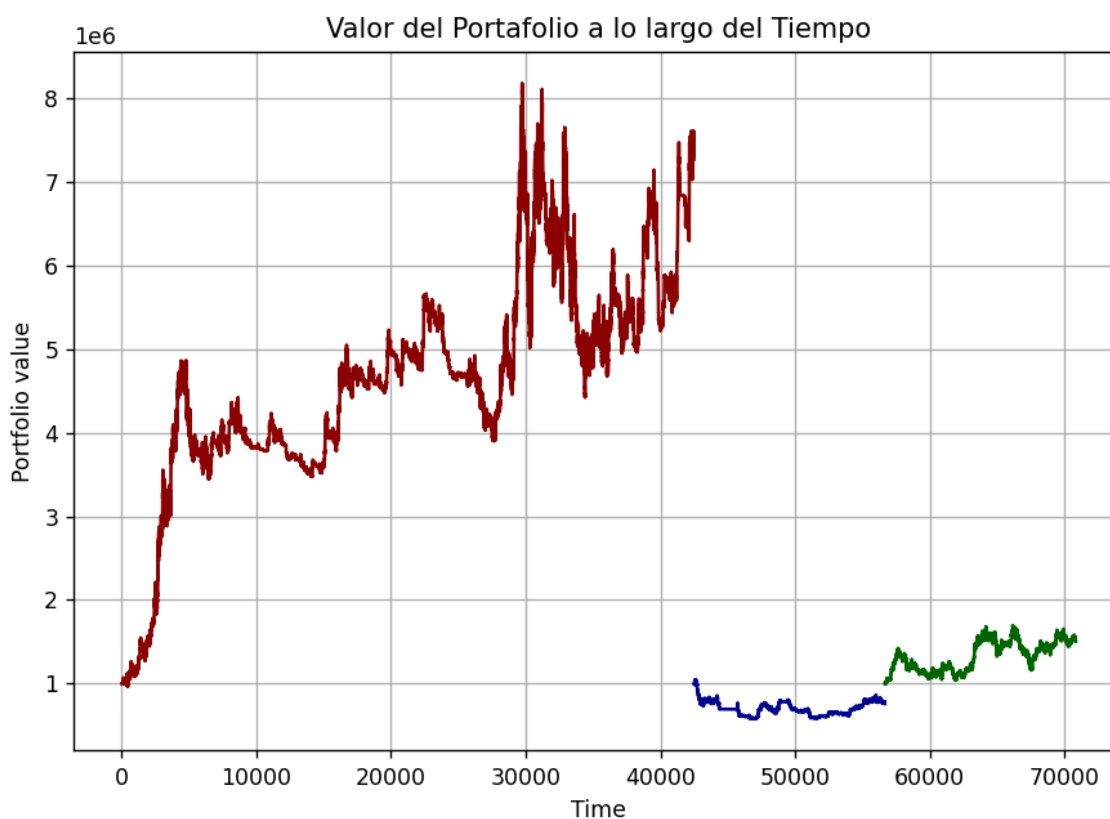
En el walk-forward, ocurre algo similar a en K-Folds, ya que se parte en un número arbitrario de partes el conjunto de datos, sin embargo, se hace en orden. Por ende, cada sección es continuación de la anterior, y así evitamos perder información valiosa.

Entonces, entrenamos el modelo en el conjunto de Train, que a su vez está dividido en secciones consecutivas, de las cuales se extrajo el ratio de Calmar promedio, y con los parámetros que generaron esta métrica, se evaluó la estrategia en el resto de conjuntos, obteniendo así los resultados de la estrategia.

Análisis y Comparación de Resultados

Comportamiento del Portafolio

A continuación se presenta el valor del portafolio a lo largo del tiempo, en los tres conjuntos en los que se probó la estrategia:



Podemos ver que el valor del portafolio creció de manera desmedida en el segmento de Train, a pesar de que se penalizó el sobreajuste de los parámetros y se partió en 10 para hacer el walk-forward. Luego, para el conjunto de datos Test, existió una ligera pérdida, aunque la volatilidad se redujo significativamente, y los movimientos no fueron tan bruscos.

Finalmente, en el conjunto de Validation, obtuvimos una ligera ganancia, por lo que vemos que la estrategia puede funcionar, sin embargo, no siempre lo hace, a pesar de la optimización que hicimos específicamente para el contexto de este activo.

Los parámetros con los que se obtuvieron estos resultados fueron los siguientes:

Parámetro	Valor
Rsi window	15
Rsi lower threshold	24.8881
Rsi upper threshold	78.5113
Macd window slow	22
Macd window fast	14
Macd window sign	7
Adx window	13
Stop loss	0.08924
Take profit	0.10001
N shares	12.8
Best Value (Objetivo)	3.2499

Podemos ver que casi todos los parámetros son conservadores. Por ejemplo, el stop loss y take profit son relativamente conservadores, en 8 y 10 por ciento, por lo que es inteligente tomar pequeñas ganancias y no dejar crecer las pérdidas de manera sustancial.

Además, al crear los rangos sobre los cuales se buscaron parámetros, se buscaron que fueran rangos prácticos que cayeran en ejemplos posibles de la realidad. Es curioso, por ejemplo, que las ventanas del MACD cuadraron casi con 7 periodos de diferencia. También, es importante notar que “n_shares”, la cantidad de acciones que se compraron por operación tampoco fue demasiado alta, siendo de 12.8.

Métricos de desempeño

A continuación, vemos una comparativa de las métricas desempeño de cada conjunto de datos:

Métrica	Train	Test	Validation
Rendimiento Anual	0.5157	-0.1243	0.3231
Volatilidad Anual	0.4614	0.3102	0.3686
Max Drawdown	0.4596	0.4499	0.3165
Calmar Ratio	1.1220	-0.2763	1.0211
Sortino Ratio	1.3227	-0.4357	1.1249
Win Rate	0.4961	0.4243	0.4773

Comparando el rendimiento anual de nuestra estrategia, contra benchmarks como la tasa libre de riesgo estadounidense (4% anual) y una de las inversiones pasivas más famosas, el S&P 500 (10.5% aprox.) vemos que cuando la estrategia funciona, los sobrepasa de manera contundente. Obtuvimos 51.57% en el train, lo cual es esperado, sabiendo la optimización que buscamos hacer con esos datos, y 32.31% en el set de validación, el cual representa un muy buen rendimiento anual.

Sin embargo, si comparamos por ejemplo, el ratio de calmar, vemos que en las circunstancias de ganancia, apenas se coloca por encima de 1 en ambos casos, por lo que a pesar de tener un buen rendimiento, considerando el posible riesgo podría ser una opción no muy segura.

El win rate estuvo por debajo de la mitad en todos los conjuntos de datos, por lo que habría que manipular la manera en que las señales nos dan opciones de compra y salida, sabiendo que en menos de la mitad de las veces son señales verdaderas.

Limitaciones

Las principales limitaciones de este proyecto radican en la manera de diseñar la estrategia, y el entorno en que se diseñó. El mercado de Criptomonedas es conocido por ser muy volátil, y no está sujeto a las mismas condiciones de mercado que las acciones tradicionales, por lo que usar indicadores de análisis técnico tradicionales en trading de acciones, quizás no garantizan funcionamiento en todos los mercados y activos.

Además, existe la posibilidad de hacer más robusta la estrategia añadiendo confirmaciones de otros indicadores, pero manteniéndolo en tres, es difícil de asegurar que las señales correctamente representen la decisión a tomar en cada momento en el tiempo.

Otro aspecto a tomar en cuenta es que esta estrategia se simuló en un entorno muy líquido y sin costos de spread inesperados, lo cual puede sesgar nuestras ganancias a ser mejor de lo que serían tradicionalmente en otros mercados.

También es posible modelar parámetros, de manera que cambien con el tiempo, o que reaccionen a tendencias de mercado. Por ejemplo, podríamos tratar “n shares” como una cantidad que cambia conforme el precio del activo lo hace, ya que no es lo mismo comprar 10 instancias de un activo que vale 100 dólares a 10 instancias de un activo que vale 5000. Esto nos puede dejar sin efectivo suficiente para ejecutar operaciones y por ende sesgar nuestra estrategia.

Análisis de Riesgo

El riesgo es inherente en este tipo de mercados, sobre todo cuando designas una estrategia pre-diseñada para intentar competir con un mercado que se actualiza cada segundo. Por eso es importante analizar los posibles peligros que se tienen al implementar estas estrategias. La volatilidad, por ejemplo, resultado de la estrategia fue relativamente alta, de más del 30% en todos los casos, por lo que es probable que incurra una pérdida sustancial a lo largo de esta. Además, como fue mencionado anteriormente, la mayoría de las operaciones fueron perdedoras, por lo que a largo plazo podría significar un rendimiento negativo. Para este tipo de estrategias, se debe ser cauteloso para generar un modelo robusto, resistente a las pérdidas que no represente un riesgo demasiado alto, también tomando en cuenta que el mercado en el que se llevó a cabo es aún un mercado muy volátil.

Conclusiones

A lo largo de este proyecto, se logró diseñar, implementar y evaluar una estrategia de trading basada en indicadores de análisis técnico, optimizada y probada bajo un esquema de backtesting y validación. Los resultados obtenidos demuestran que, si bien es posible construir estrategias que generen rendimientos en determinados contextos estas no garantizan un desempeño consistente en todos los mercados.

El rendimiento promedio y las métricas de riesgo señalan que la estrategia tiene potencial para ser rentable, alcanzando rendimientos anuales superiores a los de benchmarks tradicionales en ciertos periodos. Sin embargo, su volatilidad elevada y el bajo win rate, por ejemplo, revelan una estructura aún frágil frente a la naturaleza estocástica y cambiante de los mercados financieros, especialmente en un activo tan volátil como el BitCoin.

La optimización con Optuna permitió encontrar combinaciones de parámetros eficientes, aunque también es notable la importancia de evadir el sobreajuste, ya que resultados excesivamente buenos en el conjunto de entrenamiento no necesariamente se replicaron en entornos fuera de muestra.

El proyecto valida que los indicadores técnicos pueden ser herramientas útiles dentro de un marco analítico sólido y con un adecuado control del riesgo. Sin embargo, el éxito en este tipo de estrategias depende de la adaptación dinámica del modelo a las condiciones del entorno.

Como área de oportunidad, sería recomendable incorporar modelos que rediseñen los parámetros a lo largo del tiempo, así como más indicadores o información que nos ayude a tomar decisiones y verificar las tendencias de mercado que nos pueden ayudar a generar más ganancias.

Fuentes de Consulta

- <https://www.google.com/url?sa=i&url=https%3A%2F%2Finvestorplace.com%2F2023%2F11%2Fthe-relative-strength-index-rsi-indicator-does-it-work%2F&psig=AOvVawljtwIDKRPBTXy-dLJu7IHZ&ust=1759900859130000&source=images&cd=vfe&opi=89978449&ved=OCBgQjhxqFwoTCMCF6OKrkZADFQAAAAAdAAAAABAE>
- <https://www.investopedia.com/terms/r/rsi.asp>
- <https://www.investopedia.com/articles/forex/05/macddiverge.asp>
- <https://www.investopedia.com/terms/m/macd.asp>
- <https://www.investopedia.com/terms/m/maximum-drawdown-mdd.asp>
- <https://www.investopedia.com/terms/c/calmarratio.asp>