

Proyecto '002 - Introducción al Trading'

Mauricio Martínez Ulloa

Microestructura y Sistemas de Trading

Introducción:

Este proyecto se desarrolló buscando lograr una estrategia de trading profesional, basada en indicadores técnicos que entregaban señales de entrada y de salida, a través de un proceso de backtesting cuya métrica a maximizar es el Calmar Ratio.

Los datos utilizados fueron del activo BTCUSD en temporalidad de una hora, con cerca de 6 años de información, y condiciones específicas utilizadas: Comisión por transacción = 0.125%, sin apalancamiento y posiciones Largas+Cortas. Aparentemente, estos datos fueron obtenidos de Binance, y se reciben a través de un CSV.

Para evaluar el desempeño de nuestro código, utilizamos las métricas: Sharpe Ratio, Sortino Ratio, Calmar Ratio, Maximum Drawdown y Win Rate.

El proyecto requería de la obtención de señales por 3 indicadores como mínimo, aunque, después del desarrollo, opté por incluir adicionales para que la estrategia fuera funcional.

Los parámetros para optimizar (por medio de Optuna) eran Stop Loss, Take Profit, N Shares, más todos aquellos necesarios para la obtención de las señales (dadas un rango definido que conviniera).

El acceso donde se puede consultar el repositorio de GitHub para visualizar todo el progreso del código, y visualizar los resultados mostrados ejecutando *prueba_bestparams.py* es el siguiente: <https://github.com/ulloa09/Proyecto2>

Estrategia planteada y razonamiento:

Buscando en foros y sitios Web, pude ver que muchas personas utilizan sus propios parámetros y estrategias adaptándolo a lo que puede realizar típicamente su activo, lo cual me pareció inteligente, pues al final los datos previos que tenemos nos dan mucha información y podemos observar tendencias, la velocidad de los cruces, y qué tan sensibles son a falsas salidas, que en el caso del BTCUSD para temporalidad de una hora se observan movimientos bruscos que pueden estar dañando el funcionamiento de la estrategia que planteas si eres muy restrictivo con las salidas.

Lo que se busca a través de esta estrategia, tras pasar por un largo proceso, es apoyado de los indicadores que proporcionan las señales, obtener confirmaciones de 2 o más de estos indicadores, los cuales se manejan dentro de un rango obtenido por lo que pude investigar, pero también por la prueba y error, para analizar el comportamiento del portafolio. Estas señales son únicamente de indicadores técnicos, y se proporcionan y procesan únicamente durante el conjunto de entrenamiento (60%), es decir, el restante conjunto de datos se utiliza únicamente como prueba y validación, para observar cómo funciona la estrategia y los parámetros que se obtienen (de los cuales se explica el funcionamiento más adelante) ante los nuevos datos, los cuales no formaron parte del entrenamiento, y por lo tanto son un desafío para la estrategia.

Tras haber analizado los datos de entrenamiento que se obtuvieron, podemos ver una clara tendencia alcista, con caídas no consistentes, aunque existen, y no por periodos muy prolongados. Los datos al ser de temporalidad de una hora, tenían movimientos fuertes de una vela a otra y recibían muchas señales de salida que innecesariamente cerraban la posición. El problema no se daba por parámetros de Stop Loss mal asignados, ni tampoco por una cuestión de mal cálculo en la señal, sino que, al ser una temporalidad tan corta, había muchas señales de salida que se tomaban en cuenta, y no se justificaban, pues la operación no podía cumplir con su objetivo si se interrumpía de manera tan repetida. Este tipo de temporalidad horaria complicó un poco este ejercicio dado que el activo a lo largo del tiempo terminaba por crear una ganancia manteniendo la posición por grandes intervalos de tiempo, cosa que dado las señales y estructura del código, no permitía mantener, sino que buscaba crear ganancias en intervalos muy pequeños, ya que el pago constante de comisiones al abrir y cerrar posiciones, poco a poco iban perjudicando el rendimiento y desempeño de las posiciones a lo largo del tiempo. Es decir, estas operaciones ‘rápidas’ no estaban compensando con ganancias el pago de las comisiones.

No se realizó pre-procesamiento de datos, a excepción de la limpieza y carga de datos básica, así como de convertir las columnas de fechas y de revisar los índices para que el procedimiento coincidiera y fuera consistente a lo largo del proceso, a pesar de que ciertas funciones requieren un re-procesamiento de índice o conversión de datos para facilitar su manejo.

Para el manejo del proceso completo, se separaron en distintos archivos las funciones, quedando una lista de 10 documentos de formato Python Script (.py), los cuales son: *{backtest.py, comparacion.py, main.py, metrics.py, models.py, prueba_bestparams.py, results.py, signals.py, split.py, walk_forward_objective.py}*.

Realizar esta separación de archivos resultó de gran utilidad para el manejo de datos y funciones, pues ha sido más sencillo revisar el proceso que estamos realizando, así como el proceso de depuración ('debugging') a comparación de hacerlo en un solo archivo Python Script. Además, cada archivo fue documentado paso a paso para lo que realiza, por lo que se puede observar el proceso en un flujo como el siguiente:

1. Desde el *main.py* se llama a todas las funciones (algunas implícitamente), desde comenzar leyendo los datos originales y tratando las fechas
 - a. En este mismo archivo, se dividen los datos utilizando la función de *split_dfs* en tramos de 60% entrenamiento, 20% prueba y 20% validación
2. Después de tener los datos preparados, se realiza la optimización para los datos que se proporcionan (el entrenamiento con 60% de los originales) llamando a la función *walk_forward_objective*, que a su vez, realiza la separación y validación cruzada de los datos en los splits (partición en bloques) seleccionados desde *main.py*.
 - a. Esta función, a su vez, llama a *backtest* para cada uno de los splits realizados y va promediando el ratio de calmar que se va obteniendo por bloque y retorna el promedio para evaluar los mejores resultados.
3. La función de *backtesting* internamente realiza una optimización de parámetros que se va evaluando por sugerencias de Optuna para cada uno de ellos, dentro de un rango definido que se implementó a base de prueba y error, evaluando cuales eran los rangos que mejor se ajustaban a este activo en la temporalidad proporcionada.

Esta función, una vez haciendo cada simulación, está comparando con 5 señales de entrada y 2 de salida, para poder abrir y cerrar operaciones, las cuales, generan comisiones en ambos casos. Para contar una señal de entrada o salida, deben coincidir 2 o más del mismo tipo, si no, se ignora la señal. Para el caso de las ventas, ambas deben coincidir, ya que solo hay 2 posibles, dando una mayor robustez a las operaciones que sí necesitan una venta. Ejemplo gráfico del flujo de señales:

Para lograr ir almacenando estas operaciones y sus resultados, se llama a la función *get_portfolio_value* la cual se encarga de actualizar cada uno de los resultados de las operaciones y sus condiciones.

4. A *main.py* regresan los mejores parámetros, es decir, los óptimos evaluados en el entrenamiento, buscando en todo caso la maximización del Calmar Ratio (para el cual Optuna intenta optimizar con todo el proceso previo de *backtesting*).

5. Una vez teniendo nuestros resultados, realizamos la prueba en todos los conjuntos de datos disponibles, pero ahora sin buscar optimizar, simplemente evaluando en el *backtesting* cómo se comportan los resultados obtenidos con datos diferentes, al ser nuevos y posteriores. Dado el modo en el que se programó, si a *backtest* se le proporcionan parámetros, omite la parte de la optimización y solamente evalúa y realiza la evaluación del portafolio con los parámetros dados, regresando el Calmar Ratio (objetivo maximizar), el resultado del portafolio al final del tiempo proporcionado, y las métricas de rendimiento (las cuales son obtenidas desde el archivo *metrics.py*) para poder comparar el desempeño de cada conjunto de datos.
6. El resultado de estos 3 conjuntos de datos se grafica desde *main.py* utilizando la librería de *matplotlib.pyplot*, ajustando los índices y mostrando el comportamiento de cada uno de los conjuntos de datos empezando con un presupuesto de \$1,000,000 USD y su desempeño a lo largo del tiempo para cada uno. Se realiza en una sola gráfica para poder apreciar los movimientos y resultados finales.

El archivo de backtesting es la base de este proyecto, pues es donde se busca la optimización del Calmar Ratio, y se genera toda la lógica de apertura y cierre de operaciones, pero también del cálculo del portafolio, por lo que es importante cuidar que este archivo esté funcionando correctamente y sin cambios en índices, temporalidades, columnas, etcétera.

7. Finalmente, en el archivo principal, se realiza una comparación gráfica (llamando a la función *compare_btc_vs_portfolio*) de estos desempeños normalizados y concatenados, contra el desempeño del BTCUSD si solamente se hubiera comprado al inicio del tiempo (2017) y se hubiera mantenido durante todo el plazo analizado.

Opté por incorporar algunos indicadores adicionales, los que mejor funcionaron y se implementaron al código fueron: *{RSI, MACD, Bandas de Bollinger, OBV, ADX y ATR}* de los cuales se utilizan todos para las señales de entrada, pero solamente se conservaron las salidas por medio de *RSI* y *Bandas de Bollinger*, para que el código conservara las posiciones hasta que fueran rentables, de lo contrario, no lograba hacer dinero previo al cierre. Obtener menos señales de salida logró ser rentable ya que es donde más operaciones se producían, y no dejaban trabajar las posiciones durante periodos prolongados (ni ajustando el Stop Loss y Take Profit, que también se probó), noté una tendencia de en ciertos puntos de un mismo día, que se producían fuertes caídas, probablemente en aperturas o cierres de bolsas o cuando surgía algún evento atípico que fuera noticia (parte de análisis fundamental).

La idea de obtener dicho aumento de señales de entrada surgió después de haber tenido problemas para que realizara operaciones el código, pues durante las pruebas, con 3 o 4 indicadores, no recibía señales para poder realizar entradas (aunque no se consideraran las operaciones cortas), por lo que se fueron adaptando los parámetros, pero al estirar demasiado los rangos, comenzaba a realizar operaciones irracionales, por lo que preferí añadir indicadores. Después de agregar un indicador, realizaba la prueba para ver qué me generaba un mejor resultado, si las entradas o salidas de dicho indicador, y así para cada uno, hasta que llegué a la conclusión de que las salidas estaban eliminando toda la utilidad que lograba ganar por cierres inoportunos.

Al realizar estos cambios, el desempeño con las métricas aumentó significativamente, además de que al tener un respaldo gracias al análisis Walk-Forward (que busca evitar el sobreajuste de los datos), los resultados se vieron beneficiados, generando ganancias y los mejores resultados con 3 o 4 divisiones para los datos dentro del WF.

Después de haber realizado todo el proceso necesario, de simular y optimizar, así como de definir los mejores resultados para el conjunto de entrenamiento, se tomaron en cuenta los mejores parámetros obtenidos (maximizando Calmar Ratio) por Optuna, y estos mismos fueron evaluados en los conjuntos de prueba y validación para analizar su funcionamiento con datos que no conocían.

A lo largo de las pruebas, pude darme cuenta de que en realidad, el funcionamiento del código a pesar de implementar el Walk-Forward, en este ejercicio, no evita el sobreajuste, pues si graficamos la comparativa entre la gráfica original y el movimiento del portafolio (independientemente de qué indicadores se usen, eso determinó más el volumen de operaciones y la velocidad, no el resultado final), podemos observar que se está haciendo un sobreajuste en la mayoría de las operaciones, incluso para los conjuntos de prueba y validación, parece que sigue el movimiento del activo en gran medida, aunque con ciertas diferencias por las señales que llegan a estar a destiempo, el cobro de las comisiones, y el hecho de que las operaciones se abrían al inicio de la vela, y se cerraban terminando esta misma.

El resultado del modelo tuvo su mejor comportamiento utilizando 3 divisiones para la validación cruzada del *Walk-Forward*, con 500 o más iteraciones, y utilizando 6 señales de entrada como posibles operaciones, y solamente 2 de salida (*RSI* y *Bandas de Bollinger*), dando resultados en la mayoría de sus simulaciones con Calmar Ratios por encima de 1.5 para el conjunto de validación, dando en el óptimo un Calmar Ratio de 2 y una ganancia de +\$1,000,000.00 USD.

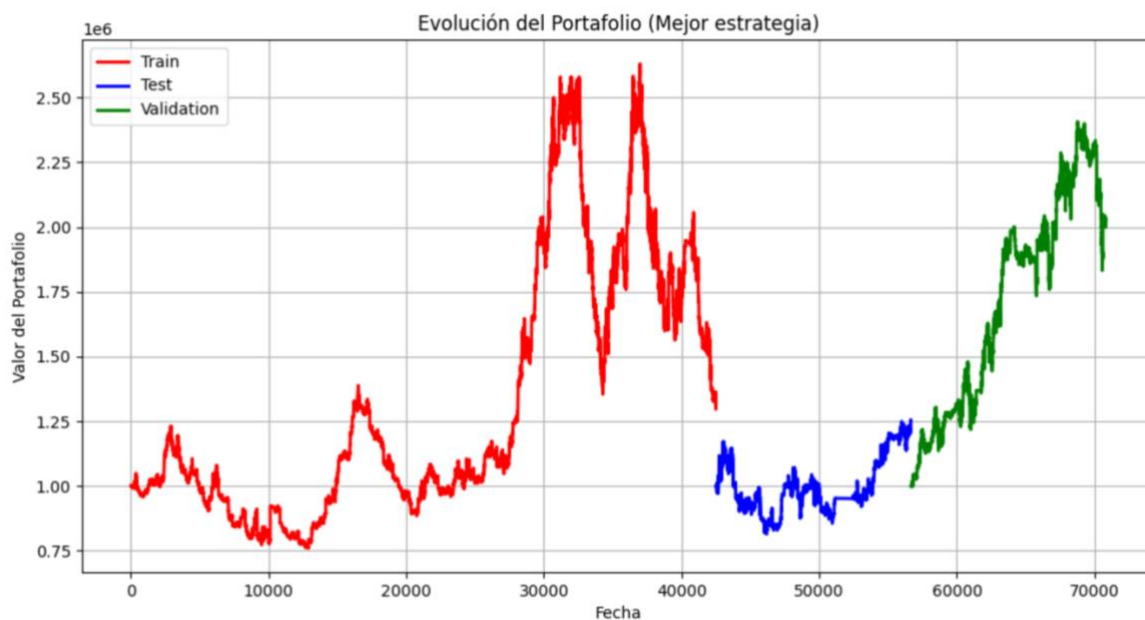
El resultado de las métricas para los parámetros óptimos fue el siguiente:

Parámetro	Valor	Parámetro	Valor
stop_loss	0.0458	bb_window	36
take_profit	0.1476	bb_std	3
rsi_window	12	obv_window	38
rsi_lower	29	atr_window	10
rsi_upper	75	atr_mult	1.0528
macd_fast	8	adx_window	22
macd_slow	40	adx_tresh	22
macd_signal	17	n_shares	4.7685

Los resultados obtenidos para las métricas descritas en las instrucciones, después de realizar el proceso de *backtest* con estos parámetros, fueron los siguientes:

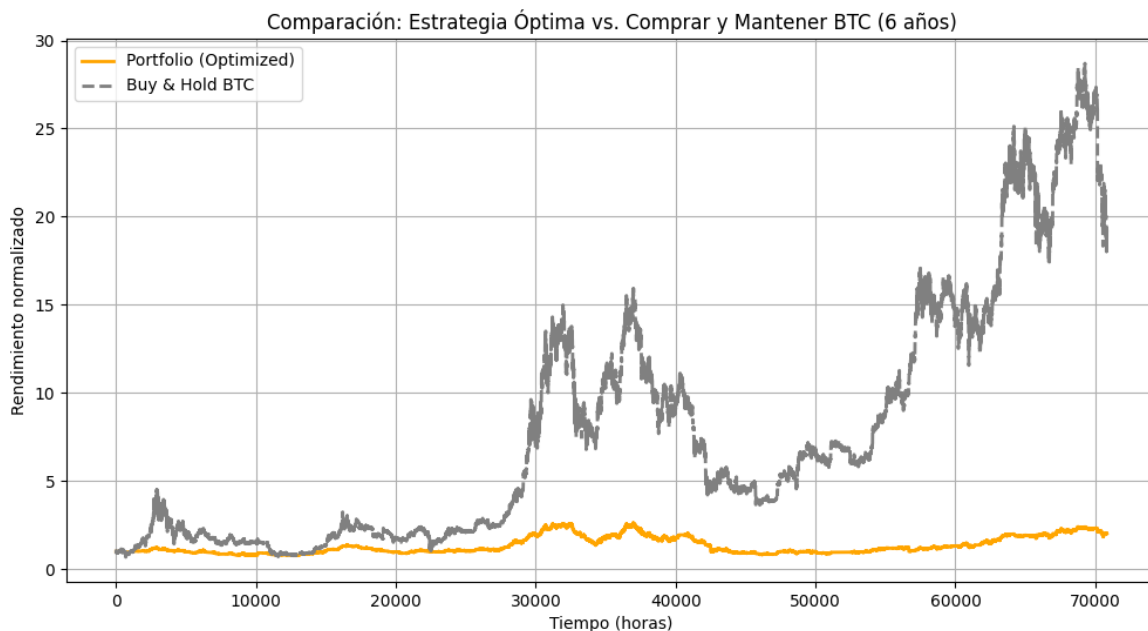
Observaciones	Portafolio	Sharpe	Calmar	Sortino	Win Rate
42496	1.298039e+06	0.3295	0.191733	0.325105	0.489988
14165	1.252164e+06	0.616837	0.603115	0.606606	0.497812
14166	2.029516e+06	1.630541	2.013114	1.645983	0.497494

El desempeño del portafolio fue bueno para el Calmar Ratio que buscábamos optimizar en la validación, además de que las ganancias acompañan en los 3 conjuntos de datos (aumentando significativamente en la validación) y manteniendo un constante Win Rate de >0.48 en prácticamente cualquier simulación por encima de las 100 iteraciones



Como parte de los resultados, se puede hacer una comparativa del rendimiento anual promedio que mantiene el S&P 500, el cual se sitúa cerca del 10%, y durante el periodo en el que se realizó la evaluación (de 2017 a 2025) se mantiene cerca del 14.5% anual. El portafolio tuvo un rendimiento superior, con un 17.85% rendimiento promedio anual en el mismo intervalo de tiempo, siendo un 3.35% más redituable que el famoso índice (que suele considerarse de bajo riesgo).

En teoría, después de analizar los rendimientos otorgados con los parámetros óptimos, podemos observar que el haber comprado una unidad de BTCUSD hubiera sido mucho mejor inversión que todo el proceso que se realizó para optimizar los parámetros de un portafolio a través del análisis técnico. Se presenta una gráfica con el rendimiento normalizado de ambas situaciones, tanto el desarrollo y optimización del portafolio, como la decisión de comprar un BTCUSD y haberlo mantenido durante el mismo intervalo de tiempo.



Podemos observar que la gráfica de la línea gris, que representa el rendimiento del BTC (considerando las dimensiones de ambas gráficas) son muy similares al comportamiento que tiene el portafolio optimizado a lo largo del tiempo, al menos por la parte de entrenamiento, y presentando ligeros cambios al rebasar las 45,000 horas, es decir, pasando de la mitad del intervalo de tiempo analizado. A pesar de terminar con un rendimiento alto o una ganancia elevada, hay alternativas de inversión que otorgan rendimientos similares como los Certificados de Tesorería de México, con un muy bajo riesgo, menor que el previsto en esta alternativa del portafolio, o para

inversores con una mayor aversión al riesgo, existen también fondos que prometen rendimientos incluso superiores al del portafolio.

Se extrajeron los resultados del rendimiento agrupado por temporalidades mensuales, cuatrimestrales y anuales:

Rendimientos Mensuales (últimos 4 años para visualización)

Periodo	Retorno %	Periodo	Retorno %
2022-01	-8.64	2023-12	2.61
2022-02	2.20	2024-01	1.72
2022-03	12.25	2024-02	-3.77
2022-04	-7.80	2024-03	5.91
2022-05	-9.81	2024-04	3.97
2022-06	-38.03	2024-05	1.75
2022-07	7.66	2024-06	11.60
2022-08	-11.62	2024-07	-5.64
2022-09	-3.05	2024-08	18.21
2022-10	3.54	2024-09	4.40
2022-11	-9.59	2024-10	13.43
2022-12	-3.42	2024-11	-1.72
2023-01	20.31	2024-12	0.79
2023-02	-0.71	2025-01	2.41
2023-03	0.10	2025-02	-0.33
2023-04	-0.47	2025-03	1.73
2023-05	-8.32	2025-04	10.36
2023-06	4.50	2025-05	1.54
2023-07	-0.04	2025-06	4.79

2023-08	-0.07	2025-07	-1.84
2023-09	2.53	2025-08	-7.44
2023-10	11.96	2025-09	-3.53
2023-11	5.80		

Rendimientos Trimestrales (2017 → 2025)

Periodo	Retorno %	Periodo	Retorno %
2017Q4	14.12	2022Q1	4.80
2018Q1	-16.69	2022Q2	-48.47
2018Q2	-8.72	2022Q3	-7.75
2018Q3	-3.55	2022Q4	-9.60
2018Q4	-2.75	2023Q1	19.56
2019Q1	11.08	2023Q2	-4.65
2019Q2	47.18	2023Q3	2.42
2019Q3	-16.51	2023Q4	21.54
2019Q4	-17.79	2024Q1	3.67
2020Q1	8.53	2024Q2	18.06
2020Q2	4.54	2024Q3	16.45
2020Q3	8.16	2024Q4	12.36
2020Q4	66.14	2025Q1	3.85
2021Q1	35.48	2025Q2	17.43
2021Q2	-32.94	2025Q3	-12.36
2021Q3	12.16		
2021Q4	-0.54		

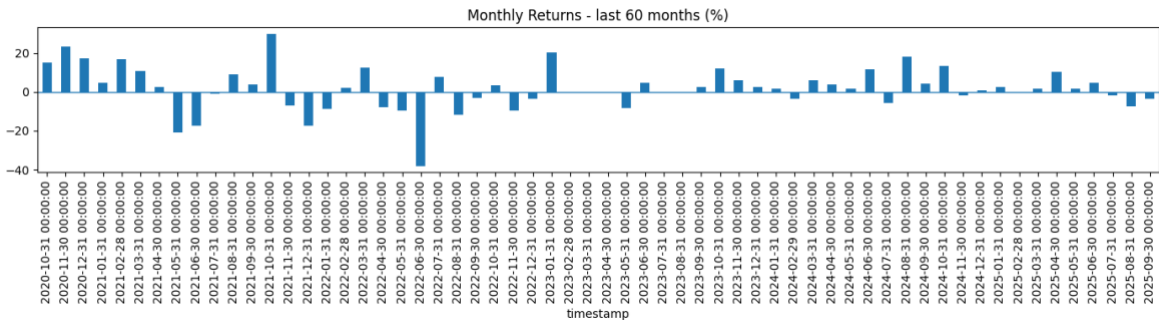
Rendimientos Anuales (2017 → 2025)

Periodo	Retorno %
2018	-28.66
2019	12.22
2020	103.90
2021	1.35
2022	-54.97
2023	41.92
2024	60.15
2025	6.88

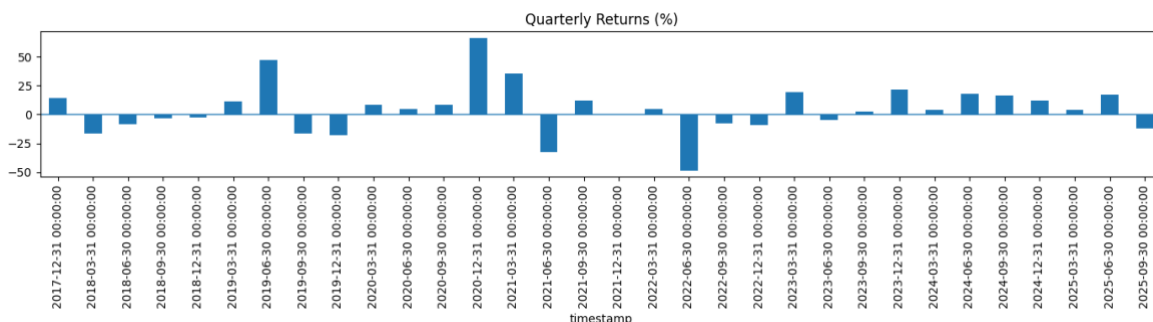
Podemos analizar que solamente hubo 2 años con un resultado negativo (aunque claramente alto) en el rendimiento anual, que promediado a lo largo del tiempo, da un **rendimiento promedio anual** de **18.75%**, siendo un rendimiento respetable, pero que conlleva un riesgo bastante alto.

Para complementar el análisis del comportamiento del portafolio, se presentan las gráficas de rendimiento por temporalidades, considerando agrupación mensual, trimestral y anual:

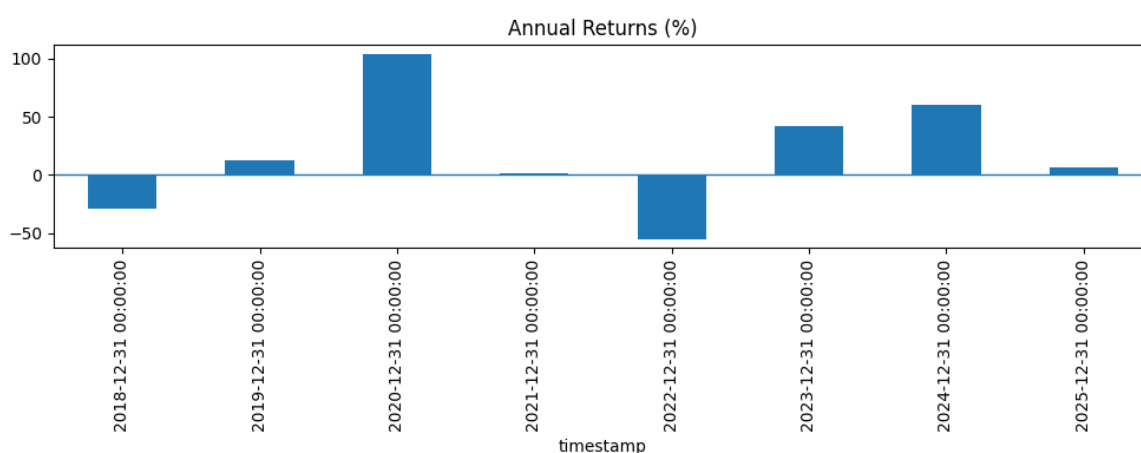
Mensual (2020 → 2025): acotando de 2020 a 2025 solamente, para apreciar el comportamiento de los últimos 5 años, dejando un poco de lado el entrenamiento inicial, para poder ver la gráfica completa.



Trimestral (2017 → 2025):



Anual (2017 → 2025):



Como podemos apreciar, el mejor desempeño del rendimiento se presenta en la validación, a pesar de ser el 2020 el mejor año, superando el 100% de rendimiento anual, valor muy atípico que está sesgado dadas las condiciones que llevaron a una situación económica crítica, pues se presentó la pandemia global de COVID-19, en la cual este activo se viralizó y llegó a altos volúmenes de operación este activo. De 2019 a 2025 se obtienen 6 años de rendimientos positivos, siendo 2021 el más bajo.

Como comparación, el bono del tesoro a 3 años de Estados Unidos en 2022, hubiera pagado tan solo un aproximado de 4.05% de rendimiento, mientras que el portafolio obtuvo, en promedio un 36.31%, siendo el periodo con un mejor desempeño de todo el intervalo de tiempo analizado en este proyecto.

Las limitaciones que considero más grandes para este tipo de ejercicios es que el realizar el backtesting puede otorgar unos parámetros óptimos que se estarán modificando con el paso del tiempo, pero que necesitan un ajuste sobre la marcha (al menos con esta estructura y lógica) muy consistente y que de no realizarse, puede estar generando, como en el proyecto realizado, un sobreajuste de los datos originales,

que al final, podemos lograr obtener un buen resultado posterior que no considero compense el alto riesgo que conlleva, y que basarse en señales de análisis técnico puede ser algo muy arriesgado.

Se pudiera utilizar en procesos más complejos los filtros de volatilidad, o utilizar diferentes temporalidades en los datos, no solo probar con temporalidad de una hora. Un comentario que también se menciona mucho entre los participantes de foros en línea, es buscar medir o cuantificar el sentimiento del mercado, pues suele marcar las tendencias del mercado o las reacciones fuertes.

Uno de los riesgos que podemos observar en este caso, es que si no tienes algún proceso bien estipulado, o eres muy permisivo con el manejo de las pérdidas, puedes terminar perdiendo (en un contexto real) más dinero del que tienes para lograr operar, ya sea porque abres operaciones simultáneas que terminan sin permitirte operar y recuperarte, o porque los parámetros no están correctamente definidos.

Otro de los altos riesgos, es que, al conectarte a un API, es importante el tiempo de retraso con el que se están obteniendo los datos originales, ya que un desfase puede estar provocando pérdidas de dinero sin que quien opera lo note. Conseguir aprovechar pequeñas brechas de precio parece algo imposible en estos casos, tanto por esa limitante, como también porque la mayoría de los traders “desinformados” o “ruidosos”, se mencionó en clase, son personas que creen saber cómo y por qué se mueve el mercado, pero en realidad están fuera de tiempo, quienes logran aprovechar en realidad las diferencias de este tipo son los bots de trading de alta frecuencia, por lo que no son un objetivo alcanzable, al menos con las herramientas que todo el mundo cuenta.

Al entrenar un modelo, como en este caso, si caes en el sobreajuste, tienes una desventaja desde antes de comenzar a operar, pues no tienes un modelo o una estrategia que podría generar dinero al recibir nuevos datos, a excepción de que sean casualidades, o el activo se moviera exactamente de la misma forma que en tus datos de entrenamiento, lo cual es básicamente imposible.

El análisis técnico, puede no ser confiable, pues no solamente necesita una alta precisión matemática, una obtención de datos en tiempo real (con el menor retraso posible, ya que la fuga de información se puede reflejar inmediatamente en pérdida de dinero), sino que también necesitaría para robustecer las estrategias incluir el factor emocional del mercado, pues a pesar de no operar por “emociones”, el mercado sí lo hace, y puede influir en el rendimiento de la estrategia. Además, el futuro incierto y las condiciones atípicas pueden lograr romper el mercado, y con ello, nuestros modelos o estrategias de trading basados únicamente en indicadores técnicos.

Las decisiones que toma este código están basadas meramente en indicadores de análisis técnico, los cuales, a pesar de tener varias opciones para señales tanto de compra, como de venta, está basándose en los rangos definidos y las condiciones que la librería *Optuna* encontró como mejores a lo largo de las pruebas. Este periodo analizado, además, fue un periodo histórico para las criptomonedas, encabezadas por Bitcoin, por lo que es evidente que lograr superarlo era algo muy complejo.

Otro tema interesante que noté es que, moviendo los intervalos de división de datos, podemos llegar a resultados diferentes, es decir, si entrenáramos con menos datos (si seleccionamos, por ejemplo, el 45% de entrenamiento y distribuimos el resto entre los otros conjuntos de datos), se captura de una mejor forma el comportamiento alcista y replica mejores resultados para los demás conjuntos de datos. Esto demuestra que se está realizando un sobreajuste, y que el solamente imitar tendencias alcistas daría un mejor resultado que un entrenamiento más preciso y con más información por los datos entregados.

A manera de conclusión, considero importante señalar que el proyecto implicó constante búsqueda de información, pruebas de parámetros y ajustes en los rangos, entre otros intentos de mejorar los resultados, pero finalmente, se obtuvo un resultado positivo, que a pesar de no ser tan alto en comparación del crecimiento que tuvo el Bitcoin, se obtuvieron resultados interesantes a diferencia de lo que el mercado libre de riesgo ofrece, o de lo que algunos índices han logrado manejar a lo largo del tiempo.

Personalmente, por mi perfil y nivel de aversión al riesgo, no utilizaría este tipo de estrategias, o en general el “trading” como una inversión fija y constante, sino que si llegara a utilizarla, sería con una cantidad pequeña de mi capital, buscando obtener rendimientos interesantes, ya que el riesgo es alto, y en este caso, no se ve compensado por ganancias extraordinarias.

ANEXO - Links visitados para investigar:

<https://www.ig.com/es/estrategias-de-trading/los-10-mejores-indicadores-de-trading-para-inversores-190830>

<https://www.ifcmarkets.mx/about-forex/best-currency-trading-indicators>

https://www.reddit.com/r/TradingView/comments/1gqbysy/whats_your_goto_approach_for_building_algo/?tl=es-419

<https://blog.traderspost.io/article/understanding-overfitting-in-trading-strategy-development>

<https://www.altrady.com/blog/crypto-bots/7-hidden-risks>

<https://www.rankia.co/blog/trading-desde-cero/5900704-bot-trading-ventajas-riesgos>

https://www.reddit.com/r/MexicoFinanciero/comments/18mjd9s/bot_para_stock_trading/

https://www.reddit.com/r/algotrading/comments/1hi4foh/best_methodplatform_for_automated_backtesting/?tl=es-419