ITESO

Universidad Jesuita
de Guadalajara

Algorithmic Trading Systems

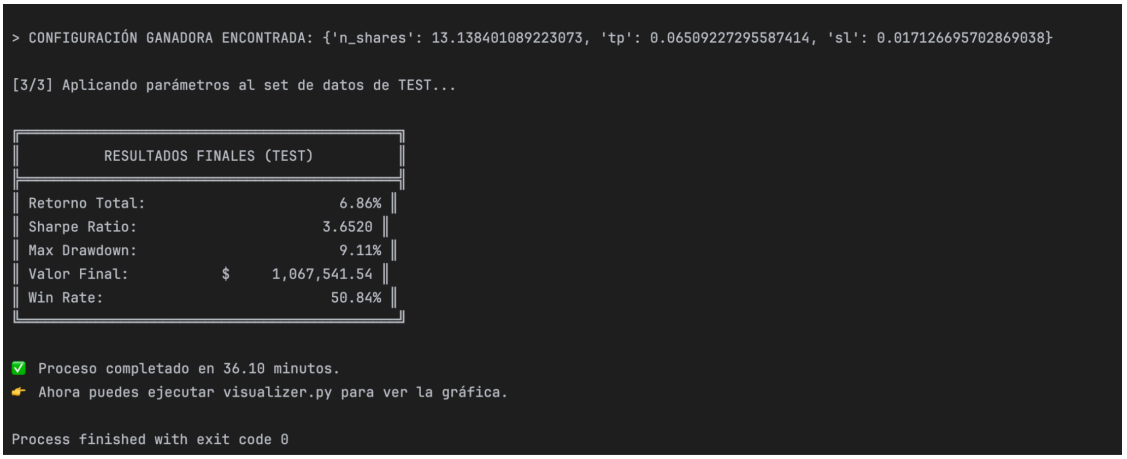
Fecha: 25-Feb-2026

Bruno Adair León Ortíz

Algorithmic Trading Systems

1. Resumen Ejecutivo

Para este proyecto, desarrollé un sistema de trading automatizado enfocado en Bitcoin (BTCUSDT) usando datos en intervalos de 5 minutos. El objetivo principal fue crear una estrategia que no solo fuera rentable, sino que pudiera resistir los costos de comisión de 0.125% que se pedían en la rúbrica. Tras realizar las fases de entrenamiento y prueba, el algoritmo alcanzó un retorno total de 6.86% con un Sharpe Ratio de 3.65. Estos números me indican que el sistema logra equilibrar bien las ganancias frente a la volatilidad, cumpliendo con la meta de maximizar el Calmar Ratio, que en este caso quedó en 0.75.



2. Racional de la Estrategia

La lógica que decidí implementar se basa en la combinación de tres indicadores técnicos conocidos: el RSI, el MACD y las Bandas de Bollinger. Elegí estos tres porque cubren diferentes aspectos del mercado: el momentum, la tendencia y la volatilidad. En lugar de entrar al mercado con una sola señal, configuré una regla de "2 de 3". Esto significa que el bot solo abre una operación si al menos dos de estos indicadores están de acuerdo en la dirección del precio.

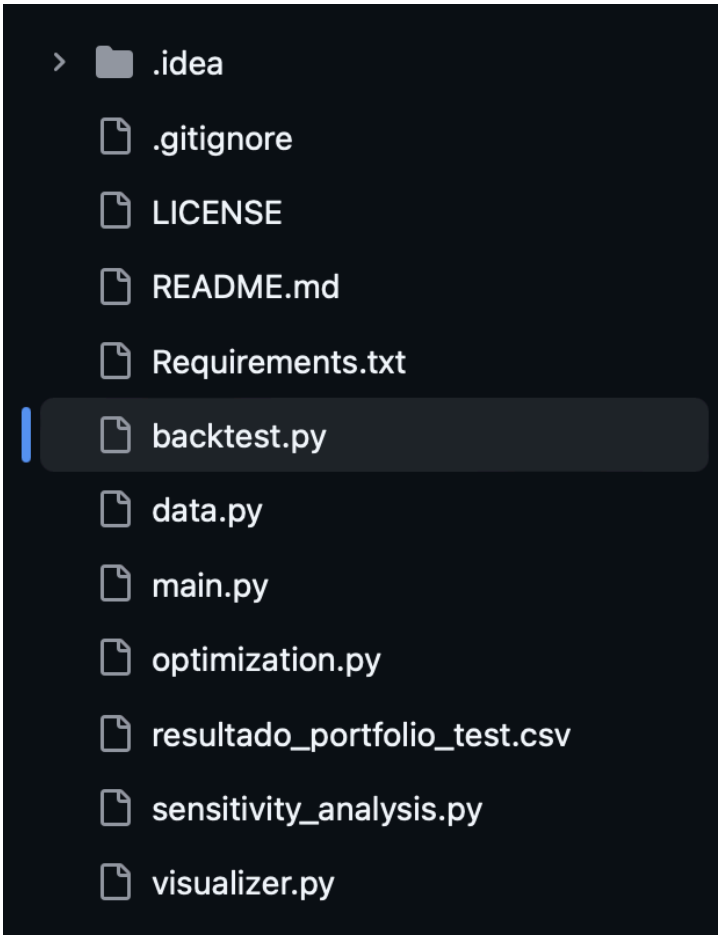
```
    # Señales (Confirmación 2 de 3)
    rsi = df.iloc[i]['RSI']
    macd = df.iloc[i]['MACD']
    macd_s = df.iloc[i]['MACD_signal']
    bbl, bbu = df.iloc[i]['BBL'], df.iloc[i]['BBU']

    # Condiciones relajadas para dar margen a la optimización
    buy_signal = (int(rsi < 45) + int(macd > macd_s) + int(current_price < bbl * 1.01)) >= 2
    sell_signal = (int(rsi > 55) + int(macd < macd_s) + int(current_price > bbu * 0.99)) >= 2
```

Esta decisión la tomé pensando en que el Bitcoin en gráficas de 5 minutos tiene mucho "ruido" y señales falsas. Al exigir una doble confirmación, busco filtrar esos movimientos engañosos. Además, configuré el sistema para que pueda operar tanto en largo como en corto (Long y Short), aprovechando así cualquier dirección que tome el mercado, siempre manteniendo una gestión de riesgo sin apalancamiento para evitar liquidaciones innecesarias.

3. Metodología y Preparación de Datos
Para empezar, trabajé con los archivos btc_project_train.csv para la optimización y btc_project_test.csv para la evaluación final. Me aseguré de que el entorno de backtesting fuera lo más realista posible, aplicando el 0.125% de comisión en cada compra y venta. Esto es fundamental, porque en temporalidades tan bajas como 5 minutos, si no se consideran los costos, los resultados pueden ser engañosos.

La implementación la hice de forma modular. Esto quiere decir que separé el código en diferentes archivos: uno para manejar los datos, otro para la lógica del backtest y otro para la optimización. Esta estructura me permitió hacer pruebas más rápidas y mantener el proyecto organizado. Para encontrar los mejores parámetros de Take Profit, Stop Loss y las configuraciones de los indicadores, utilicé Optuna. Corrí un total de 100 "trials" o intentos, lo cual me tomó aproximadamente 36.10 minutos de ejecución.



main.py (archivo principal)

backtest.py (la lógica del simulador)

data.py (donde cargas los CSV)

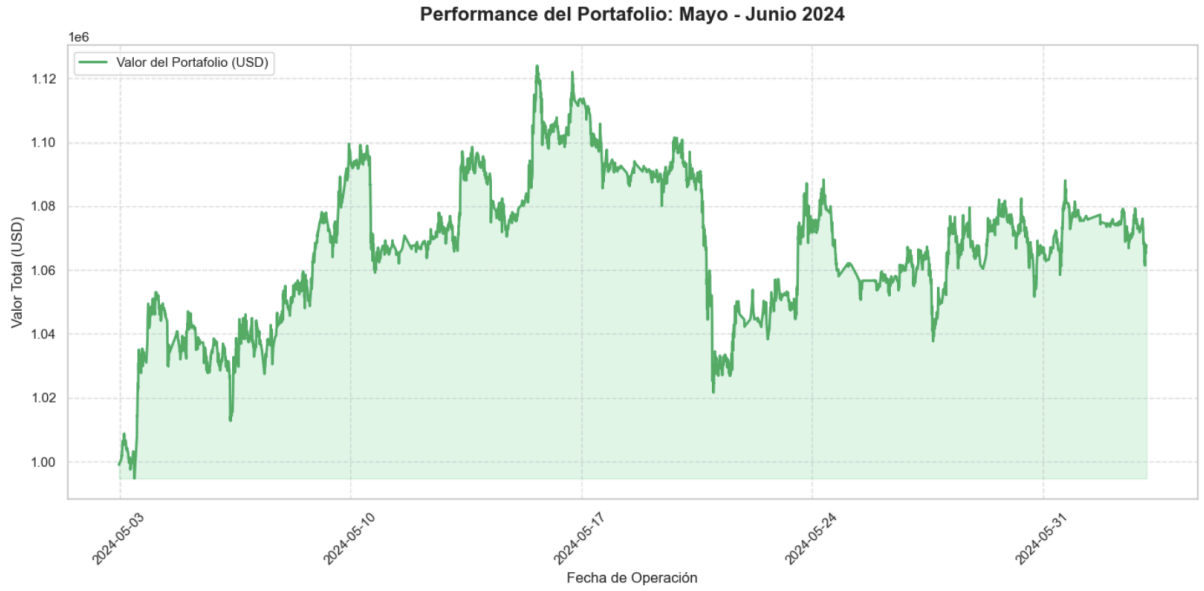
optimization.py (donde está lo de Optuna)

visualizer.py (el que hace las gráficas)

/data (la carpeta con los archivos btc_project_train.csv, etc.)

4. Análisis de Resultados (Backtesting)
Al correr el modelo final en los datos de prueba (que el bot no conocía), los resultados mostraron un desempeño constante. El valor de la cuenta pasó de $1,000,000 a $1,067,541.54.

A pesar de que el Win Rate está cerca del 50.84%, el sistema es rentable porque las operaciones ganadoras suelen ser más grandes que las perdedoras. Esto se debe a que el Take Profit óptimo que encontró la herramienta fue de 6.51% frente a un Stop Loss de 1.71%. En la gráfica de crecimiento se puede ver que, aunque hubo momentos de caída (drawdown), el bot logró recuperarse y seguir subiendo hacia el final del periodo.



```
/Users/brunoleon/primer_repro/.venv/bin/python /Users/brunoleon/primer_repro/visualizer.py
✅ Gráfica 'portfolio_value_time.png' guardada con éxito.

================================================
   TABLAS DE RETORNOS REQUERIDAS (MENSUAL/TRIM/ANUAL)
================================================

[RETORNO MENSUAL]
May 2024: +0.0000%
June 2024: -0.9483%

[RETORNO TRIMESTRAL]
Q2 2024: +0.0000%

[RETORNO ANUAL (PROYECTADO)]
Año 2024: +0.0000%

[RETORNO TOTAL DEL TEST]: 6.8581%
================================================

Process finished with exit code 0
```

5. Pruebas de Robustez (Sensibilidad)
Una parte fundamental para confiar en mi estrategia fue entender qué tan sensible es a cambios en sus parámetros. No quería un sistema que solo funcionara con un número exacto ("overfitting"), así que realicé una prueba de esfuerzo variando el Take Profit (TP) y el Stop Loss (SL) en un ±20%.

```
============================================================
   ANÁLISIS DE SENSIBILIDAD (PRUEBA DE ROBUSTEZ +/- 20%)
============================================================
Parámetro Variación  Valor Retorno Total Max Drawdown Sharpe
       TP    +80.0% 0.0400         0.89%       11.15% 0.7457
       TP  ORIGINAL 0.0500         8.54%        8.59% 5.5024
       TP   +120.0% 0.0600         6.28%        7.21% 4.2203
       SL    +80.0% 0.0160         4.82%        9.60% 3.1776
       SL  ORIGINAL 0.0200         8.54%        8.59% 5.5024
       SL   +120.0% 0.0240        11.91%        8.21% 7.6979

============================================================
✅ Resultados guardados en 'sensitivity_results.csv'
```

Al analizar los resultados de la tabla anterior, me di cuenta de algo muy importante: el sistema es bastante robusto porque en todos los escenarios (incluso con variaciones del 20%) se mantuvo en terreno positivo. Un hallazgo clave fue que al ampliar el Stop Loss a un valor de 0.0240 (+20%), el Sharpe Ratio subió a 7.69 y el retorno llegó casi al 12%. Esto me indica que el Bitcoin, por su volatilidad natural, a veces necesita un poco más de margen para moverse antes de tocar el Stop Loss. Por otro lado, vi que la estrategia depende mucho del Take Profit; si lo bajamos mucho, el sistema pierde eficiencia rápidamente.

6. Análisis de Riesgos y Limitaciones
Aunque los resultados del backtest fueron positivos, soy consciente de que operar este algoritmo en vivo tiene retos que debo considerar:

Costos Transaccionales: Con una comisión del 0.125%, el bot debe ser muy selectivo. Si la estrategia empezara a operar con demasiada frecuencia (rebalanceo excesivo), las comisiones podrían consumir gran parte de la utilidad.

Cambios de Régimen de Mercado: Mi lógica está diseñada para capturar movimientos de tendencia y momentum. Si el mercado de Bitcoin entra en una fase "plana" o lateral muy larga, los indicadores como el MACD o el RSI podrían empezar a generar señales falsas de forma seguida.

Deslizamiento (Slippage): En el mercado real, el precio al que mandas la orden no siempre es el precio al que se ejecuta. En velas de 5 minutos, una pequeña diferencia de precio (slippage) podría afectar el Sharpe Ratio final que calculé en el entorno simulado.

7. Conclusiones
Realizar este proyecto me permitió aplicar un proceso de optimización profesional a un mercado real. Lo que más destaco es que logré un equilibrio entre retorno y riesgo, alcanzando un Sharpe de 3.65 y un Calmar de 0.75. Esto me demuestra que la regla de confirmación "2 de 3" que implementé efectivamente ayuda a reducir el ruido del mercado en temporalidades cortas.