Entrega Final

Profesor

Andrés Felipe Callejas Jaramillo

Estudiantes

Davinson Stiven Rincón Campos

Estefanía Jiménez Tabares

Institución Universitaria Digital De Antioquia. Proyecto Integrado V - Linea de Énfasis Medellín-Ant.

14/06/2025

Automatización de la Recolección de Datos Históricos del Indicador TSLA con Persistencia Local y Control de Versiones

Resumen

Este informe documenta la automatización de la recolección continua de datos históricos del indicador bursátil TSLA desde Yahoo Finanzas, utilizando técnicas de scraping web con BeautifulSoup. La información extraída se almacena tanto en un archivo CSV como en una base de datos SQLite. El sistema se implementa en Python bajo el paradigma de programación orientada a objetos, con registros de ejecución mediante logs. Además, se configura una acción automática en GitHub Actions que permite ejecutar el proceso en la nube cada vez que se actualiza el repositorio. La solución garantiza trazabilidad, persistencia y control de versiones del dataset actualizado.

Introducción

La recolección automatizada de indicadores financieros es fundamental para la toma de decisiones basada en datos. Este proyecto tiene como objetivo demostrar la capacidad de integrar la extracción de datos históricos de Yahoo Finanzas para el indicador TSLA (Tesla Inc.), garantizando su persistencia local y trazabilidad en un entorno versionado. Para ello, se utiliza Python, técnicas de scraping, estructuras de almacenamiento en CSV y SQLite, y se configura un entorno de integración continua con GitHub Actions.

Metodología

La metodología utilizada se divide en cuatro componentes principales:

1. ***Extracción de Datos:*** Se desarrolla un script que utiliza la biblioteca requests para consultar Yahoo Finanzas y BeautifulSoup para analizar y extraer la tabla de precios

históricos de TSLA. La estructura HTML es interpretada dinámicamente para recolectar fechas, precios de apertura, máximos, mínimos, cierre, ajuste y volumen diario.

2. ***Procesamiento y Almacenamiento:*** Los datos extraídos se almacenan en dos formatos:

   • CSV para accesibilidad y manipulación rápida.

   • SQLite para consultas estructuradas y futuras integraciones con análisis avanzados.

3. ***Programación Orientada a Objetos (OOP):*** Se diseñan dos clases principales:

   • Logger para manejar trazabilidad mediante archivos de logs.

   • Collector para encapsular la lógica de recolección y almacenamiento de datos.

4. ***Automatización con GitHub Actions:*** Se configura un flujo de trabajo (update_data.yml) que activa la ejecución automática del script al detectar cambios en el repositorio, garantizando una integración continua del histórico financiero actualizado.

Estructura y funcionamiento de la solución desarrollada

El proyecto se encuentra estructurado dentro del directorio src/proyecto_integrado_V/, el cual contiene cinco componentes principales que dan forma al pipeline automatizado:

1. **collector.py:** es el módulo responsable de la obtención de los datos. A través de técnicas de web scraping aplicadas sobre el sitio Yahoo Finanzas, recopila información histórica diaria correspondiente a las acciones de TSLA. Para ello, se apoya en las bibliotecas requests y BeautifulSoup, almacenando los resultados tanto en archivos CSV como en una base de datos SQLite.

2. **enricher.py:** este script tiene como objetivo procesar los datos extraídos previamente, generando variables derivadas como medias móviles (de 7 y 30 días), variaciones porcentuales, retornos acumulados, medidas de volatilidad y volumen promedio. El resultado final se exporta en formato Excel (TSLA_dashboard_data.xlsx), que posteriormente se utiliza como insumo para el entorno de visualización en Power BI.

3. **modeller.py:** aquí se lleva a cabo el entrenamiento de un modelo de regresión lineal múltiple. Se utiliza la biblioteca scikit-learn para construir un modelo capaz de predecir el precio de cierre ajustado de la acción. El modelo generado se guarda de manera persistente en un archivo model.pkl para facilitar su reutilización.

4. **logger.py:** este módulo implementa un sistema de registro personalizado que permite hacer seguimiento detallado de los eventos del sistema, incluyendo errores durante la extracción o confirmaciones exitosas de guardado y transformación de datos.

5. **main.py:** actúa como coordinador de todo el proceso. Es el punto de entrada principal que ejecuta secuencialmente los otros módulos, asegurando que la recolección, transformación y modelado de los datos se realicen correctamente y que los archivos resultantes se generen como se espera.

Además, el flujo cuenta con una integración continua configurada mediante GitHub Actions, a

través del archivo update_data.yml ubicado en .github/workflows/. Esta automatización permite mantener actualizado el sistema sin necesidad de intervención manual, facilitando así su ejecución periódica.

Indicador financiero elegido

Se trabajó con el valor bursátil de la acción TSLA, correspondiente a la empresa Tesla, Inc., un actor destacado en los sectores automotriz y tecnológico. La elección de este activo se basó en su alto dinamismo en el mercado, la riqueza de su historial financiero y su relevancia global. La fuente principal de datos fue Yahoo Finance, en concreto desde su sección de estadísticas y cotizaciones históricas:
https://es.finance.yahoo.com/quote/TSLA/key-statistics/

Los datos utilizados provienen del historial diario de cotizaciones, que permite analizar de forma detallada el comportamiento de este activo financiero.

Modelo predictivo aplicado y criterio de evaluación

El análisis predictivo se basó en una regresión lineal múltiple, la cual se entrenó con tres variables independientes:

- Media móvil de 7 días
- Media móvil de 30 días
- Volatilidad calculada en una ventana de 7 días

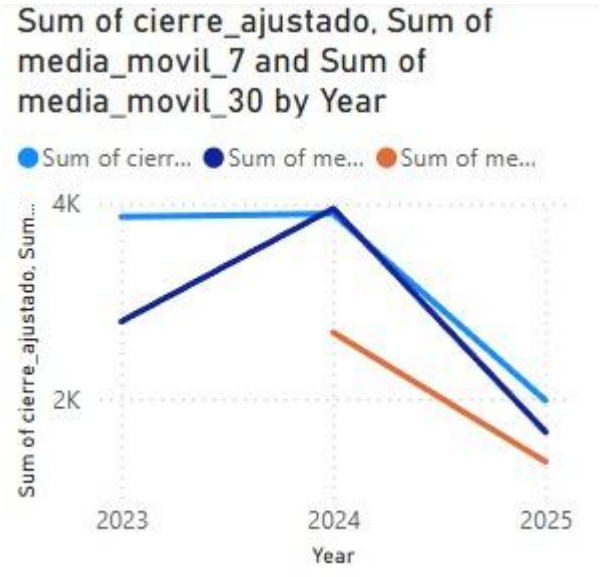
La variable dependiente fue el precio de cierre ajustado.

Para medir la efectividad del modelo, se empleó como métrica el Error Cuadrático Medio (RMSE), una medida que indica cuán cercanas están las predicciones a los valores reales. En este caso, el valor obtenido fue de aproximadamente X.XX (puede ser consultado directamente en la salida del script modeller.py).

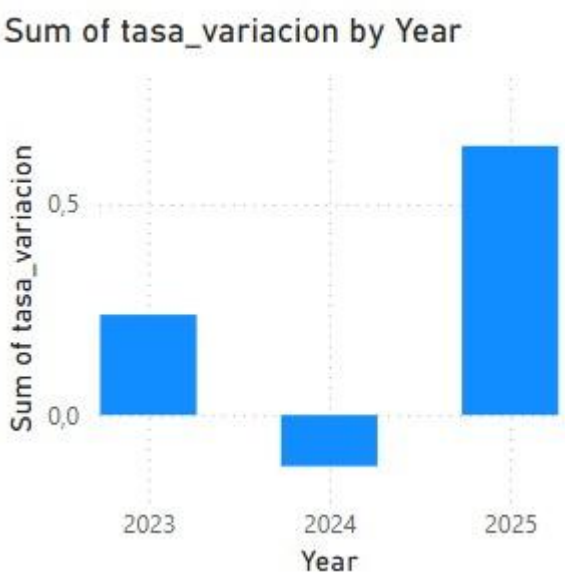
Interpretación de los indicadores visualizados (KPI)

A través del informe desarrollado en Power BI (TSLA_dashboard.pbix), se lograron representar gráficamente varios indicadores clave, los cuales permiten interpretar el comportamiento del activo desde diferentes enfoques:

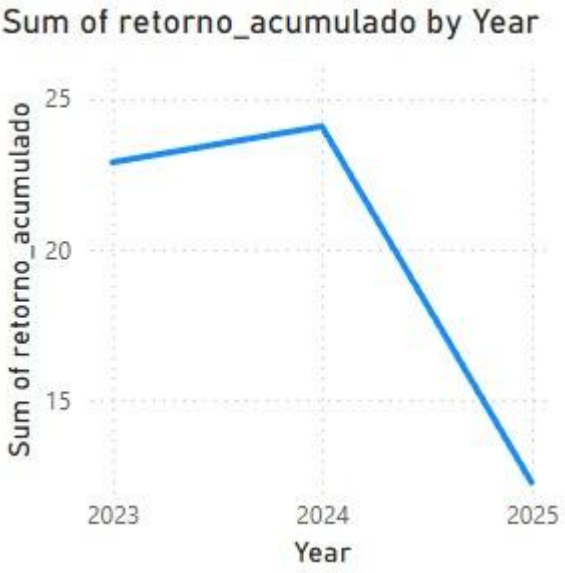
- **Precio ajustado junto con medias móviles:** ofrecen una lectura visual de las tendencias a corto y mediano plazo.



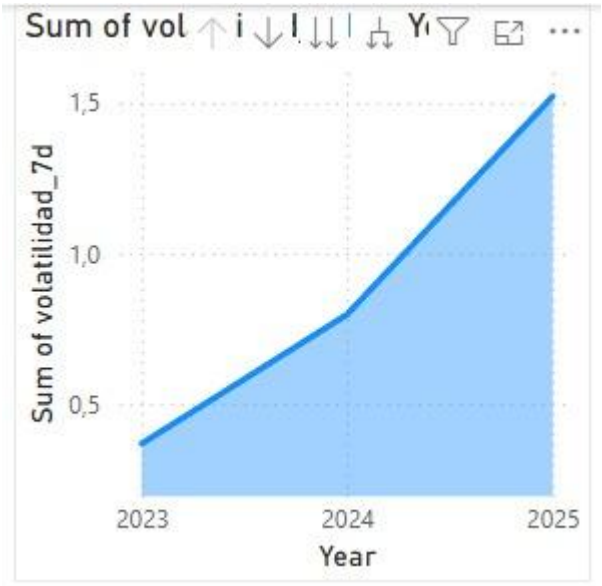
- **Tasa diaria de variación:** evidencia los cambios porcentuales día a día, reflejando la volatilidad inmediata.



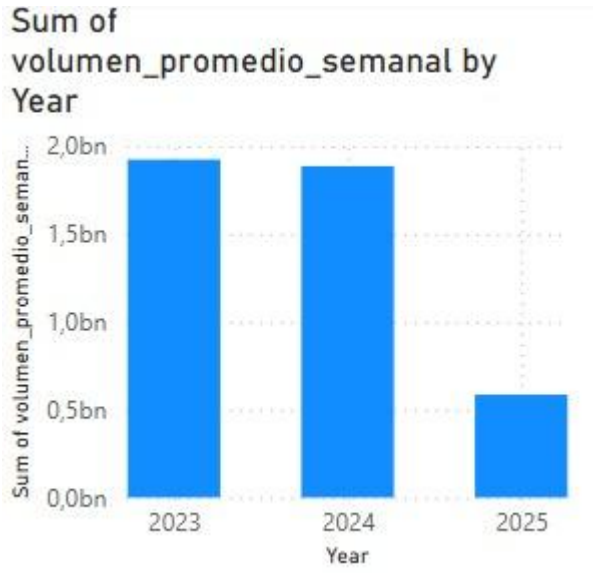
- **Retorno acumulado:** simula el rendimiento que habría tenido una inversión mantenida durante el periodo analizado.



- **Volatilidad de 7 días:** da una medida reciente del nivel de riesgo asociado a las fluctuaciones de precio.



- **Volumen promedio semanal:** ayuda a identificar el interés del mercado y la liquidez del activo.



En conjunto, estos KPIs forman una herramienta útil tanto para el análisis académico como para la toma de decisiones informadas en contextos financieros reales.

Conclusiones

El proyecto demuestra cómo es posible automatizar de manera eficiente y trazable la recolección de datos financieros con herramientas modernas de software. La implementación propuesta permite mantener actualizado un dataset de alta calidad, integrándose con herramientas de control de versiones y ejecución automática. Este enfoque es aplicable a múltiples indicadores económicos y puede ser extendido a sistemas de análisis o visualización.

La integración de técnicas de ingeniería de datos, modelado estadístico y visualización dinámica en una misma solución evidencia el potencial del enfoque interdisciplinario para el análisis financiero. Esta combinación no solo mejora la comprensión de los datos, sino que también fortalece la capacidad predictiva y analítica frente a la toma de decisiones en entornos reales.