Python para el análisis de datos - UNAV 2022-2023

Kraken API

Víctor Madureri, Raúl Tuozzo

El presente proyecto consiste en obtener las cotizaciones para los pares de monedas disponibles en el exchange Kraken, para luego graficar su precio y ciertos indicadores técnicos en una interfaz gráfica interactiva que permita modificar parámetros como el par seleccionado y el periodo de tiempo a evaluar.

Se selecciono Python como lenguaje de programación por ser el objetivo de estudio de la materia, y Streamlit como framework de desarrollo web por su facilidad para el desarrollo de web apps para visualización de datos. Una de las grandes ventajas de Streamlit es que se comporta de manera reactiva y de esta manera el usuario puede interactuar con el dashboard sin necesidad de refrescar la página.

Para la obtención de datos se utilizó la librería pykrakenapi, la cual combina un cliente API de Kraken con los dataframes de Pandas facilitando el proceso de carga y procesamiento de datos y para visualizar los datos utilizamos Plotly por su facilidad de uso, una comunidad de desarrolladores activa y su compatibilidad con Streamlit.

A screenshot of a computer screen

Description automatically generated with medium confidence

Fig. 1 Vista general del dashboard.

A continuación, se detalle el proceso de desarrollo de cada una de las fases del proyecto:

Lectura de Datos:

Para la lectura de datos se utilizó pykrakenapi como librería, la cual dispone del método get\_tradable\_asset\_pairs que devuelve un dataframe con la lista de pares disponibles y get\_ohlc\_data que devuelve un dataframe con las cotizaciones de un par seleccionado en un periodo de tiempo dado. Esta librería es consumida dentro del proyecto mediante una clase que llamamos de la misma manera, cuyo objetivo es hacerla compatible con Streamlit, manejando el cache de los métodos y los errores en la UI.

Text

Description automatically generated

Fig. 2 KrakenAPI wrapper class.

Graficar cotizaciones:

Dado el pequeño numero de elementos visualez se decidio no dividirlos en componentes reutilizables, por el contrario, se puede encontrar todo el codigo referente al renderizado de conponentes en main.py. En esta sección lo primero que se hace es obtener la lista de pares disponibles. De manera predeterminada se selecciona el par ETH/USD, pero el usuario puede seleccionar el que desee en el menu izquierdo, asi como tambien puede seleccionar el periodo de tiempo deseado para el estudio.

Luego, se procede a obtener las cotizaciones del par seleccionado y graficarlo utilizando el grafico de velas de plotly.

A screenshot of a computer

Description automatically generated with medium confidence

Indicadores técnicos:

Para el cálculo de los indicadores técnicos se implementó la clase TradingPair, la cual se puede encontrar dentro del directorio models, Una instancia de esta clase representa el movimiento de un par de monedas y su constructor se encarga de cargar la cotización correspondiente.

En esta case se implementaron los métodos get\_moving\_average\_data y get\_rsi\_data, los cuales retornan un dataframe con el valor al cierre de la moneda y el valor del indicador técnico para cada periodo de tiempo.

TradingPair es instanciado en main.py al seleccionar el par y el periodo en el menú de la izquierda y luego se llama al método get\_ correspondiente para obtener la data de cada indicador y graficarlo con plotly.

Graphical user interface, chart

Description automatically generated

Fig. 4 SMA 1 hora, ventana de 24 periodos.

Chart, histogram

Description automatically generated

Fig. 5 SMA sobre la cotización al cierre de cada periodo.

Graphical user interface, chart

Description automatically generated

Fig. 6 RSI del par seleccionado.

Estructuración:

El proyecto consiste principalmente de un archivo main.py, responsable de graficar y renderizar la interfaz gráfica. Posteriormente, en el directorio model, tenemos dos clases KrakenAPI y TradingPair responsables de la descarga y transformación de los datos y por último tenemos el directorio tests donde se encuentran las pruebas unitarias.

Graphical user interface, text, application

Description automatically generated with medium confidence

Forma de ejecución del código:

El proyecto utiliza poetry como manejador de dependencias, los pasos para ejecutar el proyecto son:

poetry install

poetry shell

streamlit run main.py

Testing:

Se utilizo pytest como librería de pruebas unitarias y se enfocaron las pruebas en el modelo TradingPair para el calculo de indicadores técnicos. Para correr los tests:

poetry run pytest.