

Python para el análisis de datos. Proyecto Final (MEBDS)

Realizado por: *Rodrigo de la Nuez Moraleda y Marcos Castro Cacho*

¿cómo hemos descubierto las columnas que utiliza kraken?

¿cómo hemos descubierto los intervalos de tiempo disponibles?

- **Descripción de cómo se ha afrontado el proyecto y por qué**

En este proyecto se busca ser capaz de obtener y graficar datos actualizados de la cotización de un par de divisas de forma que se pueda realizar un análisis financiero y estudio de las tendencias del mercado. Este proceso ha sido llevado a cabo utilizando la plataforma Kraken, cuyo servicio ofrece información en tiempo real sobre precios y cotizaciones de diversas monedas, incluyendo criptomonedas como Bitcoin y Ethereum, entre otras, así como pares de divisas tradicionales. Con este objetivo se ha utilizado la librería *krakenex* que permite inicializar un cliente que interactúe a través de consultas con la propia API de la plataforma, en nuestro caso hemos tomado la información OHLC para un par de divisas e intervalo (unidad de minutos) seleccionado de entre los que la plataforma permite seleccionar.

- **Descripción detallada de cada uno de los puntos que se evalúan**

- **Descripción de la estructura del código y estructura de ficheros**

- **Descripción de la forma de ejecución del código**

- **Pantallazos de todo Aquel punto que requiera algún tipo de visualización**

- **NADA QUE NO SE PUEDA ENCONTRAR DESCRITO O MOSTRADO EN LA MEMORIA SERÁ EVALUADO, AUNQUE ESTÉ REALIZADO EN EL CÓDIGO**

1. Lectura y representación del movimiento del par de monedas

- **Descargar datos, utilizando la librería Kraken (<https://github.com/veox/python3-krakenex>)**

Para la descarga de datos primero obtenemos la nomenclatura que utiliza la plataforma para almacenar la información de los distintos pares de divisas, de forma que mediante la librería *requests* hacemos webscraping y obtenemos todos los pares disponibles de forma que el usuario pueda seleccionar el par de divisas que desee observar. Además, se da la opción de seleccionar un intervalo de tiempo (en minutos) de hasta un mes para que se muestre la información agregada según dicho intervalo. Con esta información hacemos una consulta a la API de Kraken teniendo en cuenta los intervalos que permite seleccionar en los dos siguientes escenarios:

- 1ª opción: el intervalo seleccionado se encuentra entre los disponibles (<mencionar cuáles son>) y, por tanto, con una consulta directa se pueden obtener los datos deseados para poder transformarlos a un dataframe de pandas y manipularlos para crear un índice temporal acorde al intervalo seleccionado y transformar los tipos necesarios

- 2ª opción: el intervalo seleccionado no se encuentra entre los disponibles, en tal caso calculamos el máximo divisor de entre los posibles para que al realizar la agregación el coste computacional sea el menor posible. Utilizamos los datos como en el caso anterior tomando como intervalo seleccionado el divisor y usando una función de agregación para transformarlos al nuevo intervalo que queremos mostrar en la aplicación creada.

- **Graficar cotizaciones. Graficar el par ETH/USDT. Input de usuario que permita graficar cualquier cotización a una a elegir en el menú.**

En una primera instancia realizamos pruebas con el par ETH/USD a través de las librerías *matplotlib* y *mplfinance* para realizar gráficos de velas y de líneas con el oscilador estocástico y la media móvil pero finalmente utilizamos *plotly.graph_objs* para este objetivo.

<Insertar imágenes de los gráficos con ETH/USD y otro par a seleccionar, yo propongo bitcoin-libra esterlina>

2. Indicadores técnicos.

- **Calcular el estocástico y graficarlo.**

Explicación del estocástico

Explicación de nuestra forma de calcularlo

- **Calcular el estocástico sobre una media móvil.**

Explicación de la media móvil y su utilidad

Explicación de nuestro cálculo

<Insertar imagen con los dos pares anteriores>

- **Graficar el indicador junto con la cotización del par calculado.**

3. Estructuración.

- **Funciones**

- **Utilización de clases**

- **Manejo de errores y excepciones**

Durante todo el proceso hemos buscado evitar casos que puedan dar pie a errores por incongruencia con el tipado de las variables o la no selección de alguno de los atributos necesarios par el cálculo de los distintos tipos de gráficas.