Memoria: Proyecto final Python

En este documento se describirá el programa realizado para cumplir con los objetivos de la entrega final de la asignatura ¨Python para análisis de datos”. Dicho programa fue realizado mediante el uso del IDE PyCharm y está compuesto por 4 archivos: *api\_connect.py*, *data\_handling\_service.py*, *plotting\_service.py* y *deploy\_service.py*. A su vez dentro de la carpeta del proyecto se encuentran los archivos del ambiente virtual (carpeta *.env*), un “README.txt” para que el usuario pueda ejecutar el proyecto y un *requirements.txt* para pueda instalar todas las dependencias necesarias para correr el programa.

A continuación, se describirán los archivos y sus contenidos:

* *api\_connect.py*: dentro de este archivo se definió una función llamada “cripto\_download” que utiliza las siguientes librerías de Kraken: “krakenex” y “pykrakenapi”. Esta función descarga los datos de open, high, low y close para la criptomoneda (único argumento de la función) que el usuario va a seleccionar luego dentro de la interfaz de streamlit. La función devuelve los datos en forma de dataframe de pandas.
* *data\_handling\_service.py*: este archivo contiene dos funciones sencillas llamadas rsi y sma. Ambas funciones aceptan un dataframe como argumento. La primera calcula el “realtive strength index” y la segunda calcula la “simple moving average” o media móvil por su nombre en español. Ambas funciones retornan el mismo dataframe con el agregado de una columna que contiene la información de los índices mencionados anteriormente.
* *plotting\_service.py*: este archivo es el encargado utilizar el dataframe descargado y graficar tanto la media móvil como el RSI. Para ello se importaron dos librerías de plotly, plotly express y graph objetcs, las funciones del archivo data\_handling\_service.py y streamlit.

Todo el código está contenido dentro de una función llamada *st\_plotter* que acepta como argumento un dataframe. En un primer paso a ese dataframe se lo modifica utilizando las funciones *rsi* y *sma* para agregar las columnas con la información del RSI y la media móvil. Esas columnas son luego utilizadas para graficar los índices con la posibilidad de interactuar con los mismos. El gráfico es de tipo “vela” para poder representar el open, high, low y close del día. Dichos gráficos son desplegados al final de la función mediante el método de *streamlit* llamada *plotly\_chart.*

* *deploy\_serice.py*: este archivo da formato a la página de streamlit y la despliega. Para ello se crearon los títulos y leyendas y una lista de las criptomonedas que el usuario va a poder seleccionar interactuando con un “drop down list”. La moneda seleccionada en esa lista es tomada por la función “cripto\_download” del archivo *api\_connect.py* que genera un dataframe. Ese dataframe, luego pasa por las funciones del *plotting\_service.py* y *data\_handling\_service.py* y como resultado genera un nuevo par de plots (de RSI y media móvil).

Las funciones utilizadas cuentan con *try* y *except* para devolver mensajes más precisos en cuanto a los errores que uno se puede encontrar al correr el programa.

