Cryptocurrency Investment Advisor

Overview:

This program is meant to be used as a helper for investors when making decisions related to investments into cryptocurrencies. This program relies heavily on the current trends of cryptocurrencies and gives advice on whether to invest or withdraw cryptocurrencies into flat currency. The method used to measure whether the trend is going upwards or downwards is by instantaneous gradient as well as best line fit of the past 24 hours of data. matplotlib along with scikit-learn python modules will be used for this very task, and an extra GDAX python library will also be used to access the latest currency exchange rates via the GDAX API. The cryptocurrencies supported will depend on what GDAX has to offer, but the methods will work effectively on any cryptocurrency. This program is not meant to give 100% reliable results, but it should act as a helper/tool for anyone's use. Losses and unforeseen crashes may happen, and this program will not be able to predict them beforehand. Currently, GDAX supports Bitcoin (BTC), Ethereum (ETH) and Litecoin (LTC) cryptocurrencies. The GDAX API will be used to retrieve historical data to be used in testing the application prior to final submission for reliability assessment and improvements thereof.

Note that flat currency in this case will be US Dollars (USD).

As a summary, the program should perform the following basic functions:

- Retrieve and plot cryptocurrency data of interest.
- Indicate whether the cryptocurrency in concern is currently on uptrend, downtrend or stable.
- The user can deposit a virtual amount of money and the program should display a potential profit/loss figure to be made within a week of investment, given that the trend is to continue at about the same pace. The range will not be increased due to the volatile nature of cryptocurrencies in general. That is, they have demonstrated a history of unpredictable fluctuations in their history. This investment may be kept for more than a week as further earnings/loss accumulate or it may be cashed out back to flat currency at the user's discretion.
- When the user closes the application, the current data is saved in a simple file so that when the application is re-initialized, the data saved can be used to continue tracking the investment and to update balances from new data. Multiple users can have each save file on their own, given that they do not share the same username.
- As a simple method of protecting a user from account abuse by unauthorized people, a password can be used to protect the user's own account. This password will be hashed using the bcrypt module, a predefined salt and a master key (defined by myself) that will be used to compare the hashed password of the user attempting to access an account to the hashed password stored within the owner's save data file. Refer to bcrypt for more details on how hashed passwords are produced using raw password input, salt and master key.

Promised features:

- GUI support using Tkinter.
- Can demonstrate effectiveness by letting the user input a starting figure (in USD) and invest into multiple cryptocurrencies with the current trend in mind, refreshing the rates should reflect the profits/losses made within the period of investment and may also update the current trend from/to uptrend/downtrend/stable.
- The user can cashout the investment back to flat currency to re-invest and top up their balance in case they lost a lot of money due to unpredicted crashes.
- Data save ability, multiple users can save their progress and continue their adventure in the cryptocurrency market. The program should be able to handle multiple password-protected accounts.
- The program will display the potential profit/losses that is to be made within a period of one week into the investment. This potential figure would be based on extrapolating the current and past data. scikit-learn python module will be used for this as a data analysis tool.
- Fetch and display current exchange rates, updated on demand.
- Make a rough graph showing the trends of all cryptocurrencies using matplotlib.

Libraries used

The following modules can be installed via pip if needed:

- matplotlib
- scikit-learn
- gdax
- bcrypt

First checkpoint predicted progress:

The basic Tkinter windows and data save file features should be ready along with the APIs being set up and ready to retrieve data. The main window should display information such as flat currency balance as well as current earnings/loss from current investments, if any. The window should also display a graph showing the cryptocurrency prices over the past 24 hours and a current rate (in flat currency) should be displayed as well. Data analysis algorithm may not be ready by then, and if it was, then it will most likely be at beta-stage sort of reliability and could use a lot of improvements and tweaking.

Conclusion:

I believe that my project idea is unique and is something that I'd use for my own benefit as a virtual asset trader who is involved in the cryptocurrency market. It makes use of concepts such as Object-Oriented Programming, authentication, data analysis, GUI and APIs. It will be an application that I'd be proud to present to potential employers once completed and would be useful for other traders like myself to use. Using APIs (or Requests library in general) is something that we haven't learnt in class, so it is a plus to have incorporated something outside what we learned in the 15-112 course.