

Assignment 1: Creating data_fetch.py

Task: Implement the `download_historical_data` function to fetch historical price data for cryptocurrencies from yahoo API. Within this module, implement the function `download_historical_data` using Yahoo Finance as the data source. This function should be capable of fetching historical data for a specified stock symbol between given start and end dates. Additionally, it should support an optional parameter for the data timeframe with a default value of '1d' (daily).

Function Specifications:

- **Parameters:**
 - `symbol`: The ticker symbol of the stock (e.g., 'RELIANCE.NS').
 - `start_date`: Start date for the data in 'YYYY-MM-DD' format.
 - `end_date`: End date for the data in 'YYYY-MM-DD' format.
 - `timeframe`: The frequency of the data ('1d', '1wk', '1mo'), default is '1d'.
- **Return:** A pandas DataFrame containing the fetched data.

For Visualization:

- In a separate script, `main.py`, import your `download_historical_data` function.
- Fetch data for 'RELIANCE.NS' from June 1, 2024, to today
- Plot the closing prices of the stock using matplotlib with well labeled image
- Now make this plotting as a function specify parameters you pass and its objective is to show plot.
- Move this function to `performance.py` file, now import this in `main` and call this to plot.

Deliverables

1. A screen recording (maximum 20 minutes) showing:
 - The coding process of your implementation (time lapse)
 - DataFrame being displayed and the plot generated in your `main.py`

Deadline: **June 19, 2024**

This assignment is designed to ensure you grasp basic data fetching and manipulation techniques, which are crucial for the upcoming complex tasks in this course. Take this assignment seriously to prepare for the challenging parts of our project.

Example : `main.py`

From `data_fetch.py` import `download_historical_data`

```
from strategy_builder import buy_func, sell_func

from backtest import ohlc_long_only_backtester

from performance import calculate_performance_metrics

symbol = "BTC-USDT"

timeframe = "4hour"

df = download_historical_data(symbol, start_date, end_date, timeframe)
```