DSCI-560 Assignment No. 3 - Part 1

(Due Date: September 17, 2023, 11:59 pm)

Stock Price Analysis & Algorithmic Trading - Part 1

This assignment is the first part of a 2-part assignment that provides you with hands-on experience with building a real-time stock price analysis and algorithmic trading model. Part 1 would focus on data collection, storage, and pre-processing to prepare your team for analysis and algorithmic trading on the stored data in Part 2.

In this lab, you will work with your team to design and implement functions that dynamically fetch stock data using the finance API. Additionally, you'll explore different ways to store and quickly retrieve the data from the database efficiently.

---

1. Database Installation and Configuration

We will work with MySQL to create our databases, store data in these tables using Python scripts, and query the data using SQL queries.

1. Installation

You must follow a list of commands to install MySQL and phpMyAdmin on your Linux machine.

1.1. Install Apache web server. Apache is a web server software that will be used to host phpMyAdmin. Run the following command to install Apache:

```
sudo apt install apache2
```

1.2. Enable Apache on the firewall. The firewall may block access to Apache. To enable Apache on the firewall, run the following command:

```
sudo ufw allow 'Apache Full'
```

1.3. Install MySQL server. MySQL is a database management system that will be used to store data. Run the following command to install MySQL:

```
sudo apt install mysql-server
```

1.4. MySQL security. By default, MySQL is installed with a weak password for the root user. To secure MySQL, run the following command to change the root password:

```
sudo mysql_secure_installation
```

[Note: This step will prompt you for a new password. In case it goes into a recursive loop, which is a known issue for some installation versions, follow the steps below to fix the issue and successfully set a password]

https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04#step-2-configuring-mysql

1.5. Install PHP. PHP is a programming language that is used to create dynamic web pages. Run the following command to install PHP:

```
sudo apt install php
```

1.6. Install phpMyAdmin. phpMyAdmin is a web-based graphical user interface for MySQL. Run the following command to install phpMyAdmin:

```
sudo apt install phpmyadmin
```

1.7. When prompted, choose Apache as the webserver to configure for phpMyAdmin.

1.8. Restart Apache. Finally, restart Apache to make the changes take effect:

```
sudo service apache2 restart
```

1.9. Once you have completed these steps, you should be able to access phpMyAdmin by opening a web browser and navigating to the following address: `http://localhost/phpmyadmin`

1.10. You will be prompted to login with the root username and password set in step 5. Once logged in, you can create databases, tables, and users, as well as manage your MySQL data.

2. **Install Required Libraries**

Run the following command to install the required python libraries.

```
pip install mysql-connector-python pandas numpy yfinance
```

3. **Reading**

Use the following resources to understand the MySQL setup and installation steps and learn about the functionalities they provide and their usage. Follow the documentation to install MySQL and PhpMyAdmin on your machine.

3.1. https://dev.mysql.com/doc/

3.2. https://www.ibm.com/blog/sql-vs-nosql/

[***Note:*** Though MySQL with phpMyAdmin is the preferred way to store data, your team is free to explore other options that you feel might be better alternatives. If you do use other alternatives, make sure you document the tools used and the rationale behind them]

2. Data Collection / Storage

For this task, you will focus on creating the database tables, fetching stock data using the yfinance API, and writing a Python script to populate the database tables from the dataset.

Use this yfinance to fetch real-time stock price data for a set of stocks. You can choose a specific set of companies or sectors you're interested in.

The user should have the option to create a portfolio and define a list of stocks to be included in the portfolio. The script should accordingly fetch the stock price data for an input date range for the list of stocks in the portfolio.

Write an additional script that lets the user manage their portfolio by adding a stock to the portfolio, removing stock from the portfolio, and displaying all their portfolios with the creation date and list of stocks in the portfolio.

Perform a validation check whether or not the given stock is valid and available in the yfinance API before adding it to the portfolio and display appropriate messages to the user such as: "added successfully" or "invalid stock name," etc.

Resources:

1. https://github.com/ranaroussi/yfinance
2. https://towardsdatascience.com/python-how-to-get-live-market-data-less-than-0-1-second-lag-c85ee280ed93

3. Data Preprocessing

Handle missing values in the collected data, considering strategies like forward/backward filling or interpolation.

Transform the data into a suitable format for analysis, such as converting timestamps and calculating relevant metrics (e.g. daily returns, etc). Define the metrics that you feel would help you analyze and predict stock prices in Part 2 of the assignment.

Resources:

1. https://pandas.pydata.org/pandas-docs/stable/user_guide/missing_data.html

4. Team Discussions

Your team is expected to meet in person / virtually each day of the week and discuss the assignment progress & next steps. Document and compile minutes of all meetings in a separate file called ***'meeting_notes_A3_P1_<team_name>.pdf'***

Submission

Make one submission per team. ***You will see a group with your team name created on the blackboard. Use the group to make submissions for group assignments.*** Each team must submit all the code files for the working solution, a readme document containing information for running the code in pdf format, and a document that outlines the minutes of all team meetings in pdf format.

Provide a video per team that demonstrates the entire working solution, explains how the data tables were loaded, demonstrates query results, and talks about the design decisions and reasoning for the same. Also, include details about how your team preprocessed the data. Please include the team name and the names of all three team members in the video.

There will be a 50% penalty for all late submissions.
