

Running the code:

please run the pip install -r requirements.txt first to install the necessary packages

Please change the root password in add_stock.py file to your mysql root password(line 8 and line 34)

These are the commands in our command line interface:

-- Command to help user add stocks in the portfolio

python3 command_line.py ***argv

-h, --help show this help message and exit

- **Create a new portfolio:** --create <PORTFOLIO_NAME>
- **Add a stock to a portfolio:** --add <PORTFOLIO_NAME> <STOCK_NAME>
- **Remove a stock from a portfolio:** --remove <PORTFOLIO_NAME> <STOCK_NAME>
- **Display portfolios:** --display Display all portfolios
- **Fetch stock price data for a portfolio for a date range [YYYY-MM-DD]:**
--fetch <PORTFOLIO_NAME> <START_DATE> <END_DATE>
- **Use the model to run the model given a start and end date:**
--predict_signals <PORTFOLIO_NAME> <START_DATE> <END_DATE>
<BUDGET_IN_USD>

Below are examples of how to run the commands

Step 3: Run Your Python Program

- Before running the program, ensure Python is installed on your Mac. You can check by typing `python3 --version` in the Terminal. If Python is installed, you'll see the version number. If not, you'll need to install Python first.
- To run your program, you would use the Python command followed by the name of your script file. For example, if our script is named `stock_cli.py`, you would type:

bash

python3 stock_cli.py [options]

↓

Replace [options] with the specific commands you want to use based on the arguments defined in your script.

Using the Program's Features

Here's how to use each feature based on the arguments you've defined:

Create a New Portfolio

- To create a new portfolio named "MyPortfolio", the command would be:

```
python3 stock_cli.py --create MyPortfolio
```



Add a Stock to a Portfolio

- To add a stock symbol "AAPL" to "MyPortfolio", the command would be:

```
python3 stock_cli.py --add MyPortfolio AAPL
```



Remove a Stock from a Portfolio

- To remove "AAPL" from "MyPortfolio", the command would be:

```
python3 stock_cli.py --remove MyPortfolio AAPL
```



Display All Portfolios

- To display all portfolios, the command would be:

```
python3 stock_cli.py --display
```



Fetch Stock Price Data for a Portfolio

- To fetch data for "MyPortfolio" from "2014-01-01" to "2020-01-01", the command would be:

```
python3 stock_cli.py --fetch MyPortfolio 2014-01-01 2020-01-01
```



Predict Buy/Sell Signals for a Stock

- If the feature to predict buy/sell signals for a stock within a certain date range with the initial fund is implemented and connected to the command `--predict_signals`, to predict signals from "2014-01-01" to "2020-01-01" with initial funds 1000 the command would be:

```
python3 stock_cli.py --predict_signals 2014-01-01 2020-01-01 1000
```



Make sure to replace `python3` with `python` if that's how Python is invoked on your computer. This might vary depending on the setup.

Remember, each time you want to run a command, ensure you're in the directory where your Python script is located, or provide the full path to the script.

Finally, we can only add and delete one stock at a time with the commands. So you need to add stock one by one in the portfolio.

From our implemented models, the LSTM performed better than Arima and moving averages

After running the code the data will be saved into the two csv files `trade_history.csv` and `portfolio_history.csv`

We create a LSTM model for every stock in the portfolio

These are the performance metrics used:

RMSE (Root Mean Square Error) measures the square root of the average squared differences between predicted and actual values, emphasizing larger errors, while MAE (Mean Absolute Error) calculates the average absolute differences between predicted and actual values, representing the average prediction error.

These are our best or lowest model performance:

Mean Absolute Error (MAE): 0.4978713148424343

Root Mean Square Error (RMSE): 0.49836931196550915

Erkang Chen, USC ID : 8906487291, email: erkangch@usc.edu

Jason Sutanto, USC ID: 9111627530, email: jsutanto@usc.edu

Riten Bhagra, USC ID: 1624453305, email: bhagra@usc.edu