

Readme for the real-time stock price analysis and trading model

Team Formation

Team name: HealthEdBot

Names & USC ID:

Cici Chang (8706354957)

Daoyang Li (7168949829)

Yifan Yang (8386626967)

<https://github.com/DSCI560>

Introduction

This Python script is designed for predicting when to buy and sell stocks based on the collected real-time stock price data and evaluating the effectiveness of this trading algorithms.

Requirements:

- Linux ubuntu
- Python 3.x
- MySQL database
- Libraries: yfinance, mysql.connector

Install the necessary Python libraries using pip:

```
pip install yfinance mysql-connector-python datetime
```

Setup

Input trading dates and begin days.

```

daoyang@daoyang-virtual-machine:~/Desktop/daoyang_7168949829/lab3$ python3 mock_trading_env_updated.py
Loaded stocks for portfolio 'tech': ['AAPL', 'MSFT', 'AMZN', 'GOOGL', 'JPM', 'JNJ', 'KO', 'BA', 'NVDA', 'AAPL']
2023-10-31
Bought 10 shares of MSFT at 336.62677001953125 per share.

Bought 10 shares of GOOGL at 124.45999908447266 per share.

AAPL: 0 shares at 170.06593322753906 per share. Total: 0.0
MSFT: 10 shares at 336.62677001953125 per share. Total: 3366.2677001953125
AMZN: 0 shares at 132.7100067138672 per share. Total: 0.0
GOOGL: 10 shares at 124.45999908447266 per share. Total: 1244.5999908447266
JPM: 0 shares at 136.57781982421875 per share. Total: 0.0
JNJ: 0 shares at 145.86192321777344 per share. Total: 0.0
KO: 0 shares at 55.7064323425293 per share. Total: 0.0
BA: 0 shares at 182.35000610351562 per share. Total: 0.0
NVDA: 0 shares at 411.57379150390625 per share. Total: 0.0
Total Portfolio Value: 10000.0
-----
2023-11-01
Bought 10 shares of MSFT at 337.4251403808594 per share.

Bought 10 shares of GOOGL at 124.08000183105469 per share.

AAPL: 0 shares at 170.54531860351562 per share. Total: 0.0
MSFT: 20 shares at 337.4251403808594 per share. Total: 6748.5028076171875
AMZN: 0 shares at 133.08999633789062 per share. Total: 0.0
GOOGL: 20 shares at 124.08000183105469 per share. Total: 2481.6000366210938
JPM: 0 shares at 138.207763671875 per share. Total: 0.0
JNJ: 0 shares at 147.16151428222656 per share. Total: 0.0
KO: 0 shares at 56.04374694824219 per share. Total: 0.0
BA: 0 shares at 186.82000732421875 per share. Total: 0.0
NVDA: 0 shares at 407.7641296386719 per share. Total: 0.0
Total Portfolio Value: 10004.183731079102
-----
Total Return: 15.87%
Annualized Sharpe Ratio: 16.90
-----Final profit-----
AAPL: 0 shares at 185.9199981689453 per share. Total: 0.0
MSFT: 9.0 shares at 388.4700012207031 per share. Total: 3496.230010986328
AMZN: 31.0 shares at 154.6199951171875 per share. Total: 4793.2198486328125
GOOGL: 0 shares at 142.64999389648438 per share. Total: 0.0
JPM: 0 shares at 169.0500030517578 per share. Total: 0.0
JNJ: 9.0 shares at 162.38999938964844 per share. Total: 1461.509994506836
KO: 3.0 shares at 60.38999938964844 per share. Total: 181.1699981689453
BA: 0 shares at 217.6999969482422 per share. Total: 0.0
NVDA: 3.0 shares at 547.0999755859375 per share. Total: 1641.2999267578125
Total Portfolio Value: 11586.54610824585

```

Usage

Run the python script ‘mock_trading_env_updated.py’ directly through the terminal. The script includes a test function demonstrating the functionalities, such as buy and sell stocks, output the total return and annualized sharpe ratio.

Main Functions

- **fetch_stock_data(stock_symbol, start_date, end_date):** Fetches historical data for a specified stock symbol from Yahoo Finance.
- **load_initial_portfolio(self, portfolio_name):** Initialize or load portfolio from the database
- **buy_stock(self, symbol, quantity, date):** Buy stock

- **sell_stock(self, symbol, quantity, date):** Sell stock
- **update_portfolio_value(self, date):** Display stock information
- **calculate_performance_metrics(self, days, date):** Display total return and annualized sharpe ratio
- **execute_trades(self, date):** Execute the trade