ALGO TRADING IN PYTHON

#4 : ERRORS HANDLING AND Q&A

Long Tran – Snap Innovations Pte Inc

COURSE OUTLINE

- Session/Week 1 : Trading strategy and backtesting in Python
- Session/Week 2 : Connect to the exchange (REST api)
- Session/Week 3 : Real-time data streaming (websocket)
- Session/Week 4 : Errors handling and Q&A

LAYOUT : SESSION #4

1. Exceptions
2. Logical vs Exchange errors
3. Coding routines
4. Algo summary
5. Potential projects

EXCEPTIONS

- Exception Class Hierarchy:
  - BaseException:
    - Exception (user-defined exceptions/errors)
    - GeneratorExit
    - KeyboardInterupt
    - SystemExit
- Syntax:

```
try:

    ### main function ###

except:

    ### error occurs, do something!!! ###
```

LOGICAL VS EXCHANGE ERRORS

- Logical errors:
  - Conflicts among the modules
  - Incorrect requests to the exchange: price/quantity precision, characters case sensitive, etc.
- Exchange errors:
  - Sever shut down
  - Internet issues
  - etc.

-> Solutions:

- Correct logical conflicts in the program (use Python traceback)
- THEN, use try-except to handle exchange errors

CODING ROUTINES

- Script Editors:
  - Notepad++
  - Atom
  - Visual Studio Code
  - etc.
- Linux/Ubuntu Terminal:
  - Windows Subsystem for Linux
  - MacOS
  - Virtual Private Server (Ubuntu)
- Tips:
  - Never leave a script uncommeted
  - Optimize the loops

ALGO SUMMARY

Modules Name	Type	Functions
bbalgo.py	Main module	• build connectivity • build trading model • build portfolio Call: tradingpy.py, wss.py
trading.py	Trading Signal/Model	• process data inside trading model • Find entry, exit, handle trading signals Implement: Portfolio class, TradingModel class, Signal class
binancepy.py	Exchange Connection	• Request data to the exchange • Handle orders/positions • Open/refresh/close data stream Implement: MarketData class, Client class
wss.py	Data Streaming	• Handle data stream session • Pass data to tradingpy classes Call: tradyingpy.py, binancepy.py
indicators.py	Support	Compute technical indicators (MA, ATR, etc.)
ultilities.py	Support	Other support functions: printing, data type converters

POTENTIAL PROJECTS

- bbalgo.py

  Implement a while loop to run the program 24/7
- binancepy.py
  1. Update new requests from Binance
  2. Extend to other exchange using REST
- tradingpy.py
  1. Modify for new trading strategy
  2. Optimize loops to have powerful real-time predictors
- wss.py
  1. Handle new data type from exchanges
  2. Optimize/threading the loop to process signals more effectively
- indicator.py, ultilities.py

  Update on demand