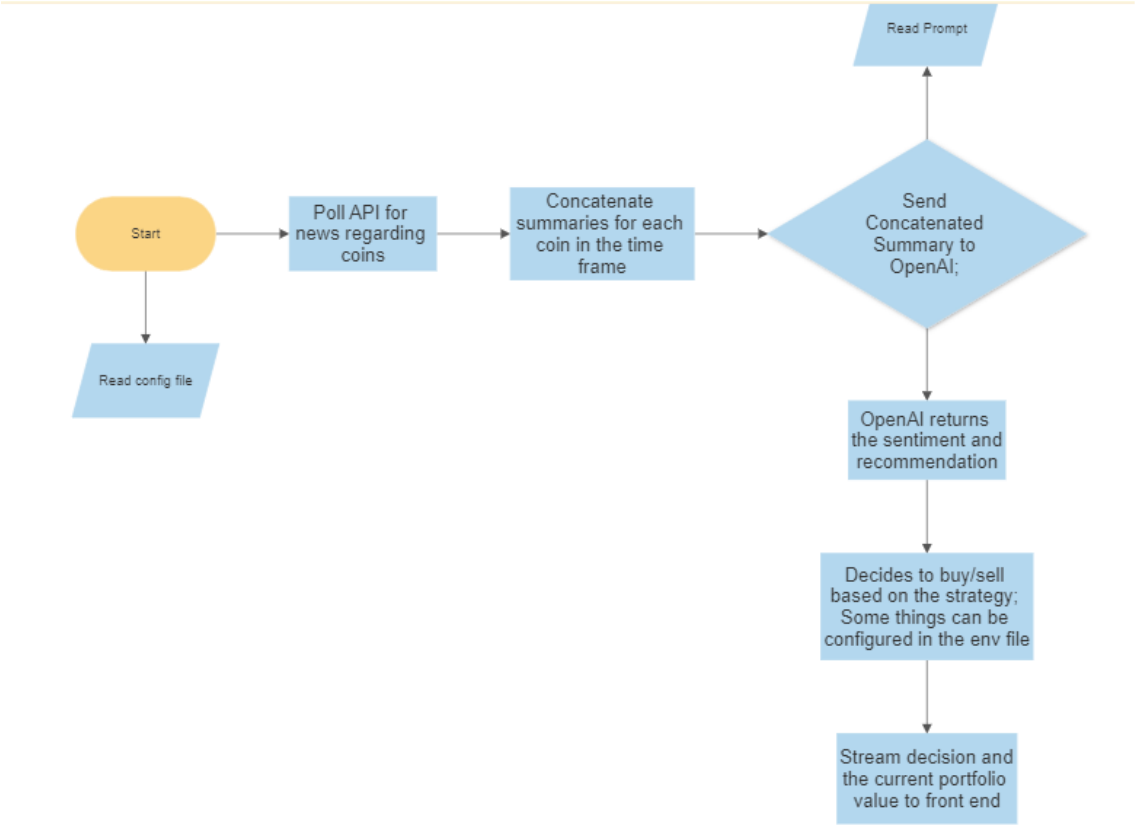
AI portfolio Manager

Team
Wladislaw Strecker
Arun Narayanan

Dependencies
Aylien API for news
Binance API for coin pricing information
OpenAI API for sentiment analysis and decision making

Architecture



Code
https://github.com/WladikC/hackathon_team_9

Summary
It is a python flask application that contacts Aylien, OpenAI and Binance to simulate trading decisions on Crypto Currencies. For the purposes of the project, the decision relied solely on what OpenAI LLM “thought” about the news regarding the crypto currency. In the real world decision will be based on multiple factors

Idea for further improvements

What we have built is a framework that can be expanded to easily test various trading strategies. Two major improvements that can be made to this system are described below. These are things that we would have built ourselves if he had enough time and resources.

Vectorizing Historical Price Data and News

If we suppose that the ultimate goal is to try and build an AI powered trading tool based on news feeds and other factors, then it is important to transform data that is already available (news feeds, pricing information etc) into a form that can be read in an efficient manner by the AI agent. This vector-db should be stored locally. Some of the advantages of possessing such a system is listed below (not exhaustive)

- Improved speed in fetching data
- Enhanced privacy since it is possible to enrich the data with proprietary information
- Possibility to build and test custom models that can potentially find previously unknown correlation rules between financial assets.

Since LLMs are prevalent now, a vector-db like the one described above can be leveraged for day-to-day activities as well. For eg: asking questions on a set of proprietary or interesting documents rather than combing through the documents.

Using AI to generate “automatable” trading strategies

In this project, we observed that new code had to be written to test new trading strategies. Since LLMs can generate code and markup now, we wondered if it is possible to build a generic system that can test new trading strategies at a faster rate.

The idea is for the trader/researcher to express a new strategy in plain english and the LLM will generate either a strategy yml file or actual code that can be plugged into a larger code base that runs the strategy.

With such a system, it will also be possible to discover new trading strategies; The feedback will be taken from the vector-db and the fixed encoding scheme will possibly allow the AI to mutate existing strategies.

There is one screenshot in the next page.

A

i want you to generate a yml based encoding scheme to express a trading strategy. This yml file is read by a python program that buys/sells instruments. The scheme should support checking arbitrary number of conditions with an arbitrary number of variables and the python program should be generic enough to handle any yml file

Sure, I can provide an example of a YAML based encoding scheme and corresponding Python program that should handle it. However, since you've asked for an arbitrary number of conditions with an arbitrary number of variables, we should define some structure to your YAML file. Let's say we have a condition and action pair, where a condition consists of a single or multiple comparisons, and an action is the type of operation to be performed (buy or sell) and the quantity.

YAML Example:

yaml Copy code

```yaml
trading_strategy:
  - condition:
      - variable: 'moving_average'
        comparator: '>'
        value: 50
      - variable: 'rsi'
        comparator: '<'
        value: 30
    action:
      operation: 'buy'
      quantity: 100
```