**Vrije Universiteit Amsterdam**

**Computational Thinking**

**Project Assignment: *<I Trade>***

**Group number: <Mykhailo Varha>**

**Members with student numbers:**

*<Mykhailo Varha> <lgm 557>*

***Date: <14/12/2024>***

**General instructions:**

Fill in the template where text between <> indicates it. Follow the instructions given to you in the project description. Rename this file to CT\_REPORT\_*GROUPNUMBER* and submit it as a pdf.

**Context Task**

AI has both negative and positive effects on the financial stock market. Here are the positive impacts. First, AI systems are more efficient in detecting fraudulent activities than most humans. So, security and integrity of the market are in a much better state than before. Also, AI as a technological prospect has increased the demand for tech-oriented stocks and thus increasing the market variety.

On the other hand, with the rise of AI technology appeared the Trading bots, that make split-second decisions in their trading patterns, and trade with monstrous efficiency. Because of these trading bots, the stock market becomes more liquid. After extensive research the liquid stock market is not good in the long term, for a few reasons: insecurity as stocks can be sold in bulk by big companies, susceptibility to flash crashes and high upkeep cost. These weaknesses are not inherently caused by AI technology, as they are mostly caused by human nature itself. The only actual weakness of AI in the stock market is that not all people have equal access to it, and therefore some may have an unfair edge over others.

However, a liquid stock market is very stable and highly competitive, with low entries and low exits so many people can join and leave anytime they wish. This also means that more entrepreneurs may get enough money from selling stocks to enhance their business. Also, AI trading bots are more efficient than most humans thus making the stock market operations much easier and more uniform.

Overall, I think the AI has a positive impact on the stock platform, even though use of it has some downsides. The liquidity of the market is just too big of a benefit no matter how one sees it. It allows for much faster progression for all the population on condition they are willing to risk their money.

**Design process**

<Here you can elaborate on the process of designing your algorithm. Describe important choices you needed to make and reflect on the dilemmas and difficulties you encountered along the way as well as on the work division within your group. Follow the instructions in the project description.>

So for the design of the project, I decided to come up with this approach. First, I decided to create 3 classes. I also used git hub. Just regular practice. My approach is a type of Divide and Conquer, as I separate work to be done modularly.

stock – class that describes a Stock unit with all its parameters

user\_profile – class that describes users input into the algorithm with all the filters

stock\_recommender – class that handles the database of stocks

However, I needed to decide how I was going to handle the database, am I going to use pandas or dask. I have used pandas as a programmer more, so I decided to go with pandas. For reference dask is a module NVIDIA Developer program gives free courses on. Capitalize on that however you wish. I only started to learn dask so I did not use it.

One issue I faced, not major but still an issue – I don’t even need class stock. It is useless as I don’t do transformations on stocks at all. Pandas does everything for me in terms of sorting and transformation as its dataframe class is very useful. I wonder if dask also has that much utility.

Next, I decided that I need to find an approach to solve the problem. I have 3 parameters to limit my search: Top in the industry, year of establishment and ESG score.

Top in the industry parameter is a bit unclear. So I decided to take liberty and define it as the (100 - N)th percentile in the industry in the stock market where N is such number that (100 – N)th percentile in the industry in the stock market will contain as much stocks as user asks for.

ESG score I defined as the some of Environmental, Social and Governance Scores.

Ratio 1 : 1 : 1;

The most limiting factor is year of establishment. If this filter parameter is active no company younger than that year is out. So I decided to make my algorithm first to limit companies on the year of establishment if the parameter is active. However, the next priority parameter was an issue. After a long thought, I came to a decision to make the top in the industry parameter come next, as I would rather help my user achieve more financial success as the my program is targeted for beginners and if my user fails there is a trend for beginners to quit the stock market and that is my worst case scenario. At last the last search parameter is ESG rating. Here I also allowed my user to select which one of the ESG score he want to look at individually.

I also want to add more variety for my user in their stock choices so I decided to make an additional task for myself that will ensure that all industries are mentioned equally as long as the size of the sample of stocks allows that.

So the actual process will look like this:

Ask user for input -> Create User profile -> Access the Database of Stocks -> Create the Dataframe of Stocks -> Exclude the Stocks younger than Users year of establishment parameter -> Select top percentile of the stocks based on the amount of stocks user asks for -> Sort the dataframe based on ESG score performance -> Split the dataframes into smaller dataframes containing one industry and from them include top N stocks equally from each industry -> Output the result.

For the output I used module tabulate as it pairs nicely with pandas dataframes.

**Flowchart**

<Include your flowchart here. Make sure that the image fits inside the margins of a page but is large enough to be readable. You are allowed to split your flowchart into several subcharts (e.g., for different sub-processes), but in this case make sure that you have made it absolutely clear where these charts link into each other.>

**Pseudocode**

<Include your pseudocode here.>

**Python code**

<Include your Python code here. In your submission, do not forget to include your Python code as a .py file as well.>

**Checklist for submission:**

* Your project report as a pdf.
* Your Python code as a .py file.
* Optional: any additional files (such as .csv files) you might have created which are required for your program to run.
* Each of the above included in a .zip file with the name CT\_PROJECT\_*GROUPNUMBER*.zip