Project iTrade

In this project, you will create a program that mimics an application that helps beginner traders manage their own investments. You will design an algorithm and present it in the form of a flowchart, pseudocode, and a Python program. Furthermore, you will report on the design process and write about the broader implications of such an algorithm in the project report.

The system you will implement is called *iTrade*, and its purpose is to recommend stocks to invest in for novice traders. The recommendations are based on the performance of the stocks as well as some preference settings that the user can give. For this, the system can make use of a database that includes information about 500 different stocks, from which the application can choose the recommended ones. A summary of the information available about each of the stocks is shown in Table 1. You can find a more detailed description of the algorithm's workings in the next section.

Table 1 Properties of companies found in the database that iTrade can use.

Name of property	***Description***
ID	The three-letter unique identifier of the company to which the stock belongs.
Performance	The performance (also called return) of the stock, is given as a percentage. For instance, a return of 11.7% means that for every \$100 invested in this stock, the user will earn \$11.7. (A negative return means a loss on the investment.)
Industry	The industry to which the corporation belongs.
Foundation year	The year in which the corporation was founded.
Environmental rating *Social rating* *Governance rating*	These three attributes reflect on how socially responsible a corporation is. On a scale of 1-10, where 10 is the highest, they rate the steps taken towards a more responsible business according to these three categories.

1. Stock selection strategy

The job of the algorithm is to recommend a given number of stocks for the user to invest in. This number is given by the user when they start the application, which can be any number *n* between 1 and 100, and the algorithm will recommend the top *n* highest performing stocks to the user. However, performance is not the only factor the application can take into account: the user can also choose one or more from 3 available preference settings, which will further influence which stocks will be recommended by the system.

You can understand these preference settings as filters: the output of the algorithm is always a ranked list of the stocks based on performance, but only stocks that comply with the active preference settings will be shown.

1.1. Preference settings

1.1.1. Highest in the industry

If this preference setting is active, the algorithm will recommend only stocks that are among the highest performers in their respective industries.

1.1.2. Establishment year

This preference setting helps the novice trader invest only in well-established companies. If the setting is active, the user can give as input a year between 1800 and 2020, and the algorithm will only recommend corporations that were founded **before or in** that year.

1.1.3. ESG criteria

ESG (**e**nvironment, **s**ocial, **g**overnance) criteria help investors to only choose companies with responsible business practices. If this setting is active, only corporations with high ESG scores will be recommended.

You might notice that there are some open questions left in the description of these rules. That is entirely on purpose: while you need to follow the general guidelines above, you have some liberty in designing your own version of *iTrade*. To help with the ambiguity, some of the decisions which you will need to make will be addressed later in this document.

2. Your tasks

Now that you have an idea of how *iTrade* operates, your job will be the following:

1. Think about the broader implications of modern computing technology on the financial world, and briefly express your thoughts on the topic in the *Context Task* described below.

2. Design an algorithm that represents your implementation of *iTrade*.
3. Provide a flowchart for this algorithm.
4. Provide pseudocode for this algorithm.
5. Implement your algorithm in a Python program.

2.1. Context Task

Modern computing and information technology have transformed the world of finance. The accessibility of online trading platforms, the prominence of automated trading agents, and the emergence of cryptocurrencies are some examples of technology's impact on the financial world. Based on your own experience and additional research, express your opinion briefly (in 200-300 words) on the impact modern computing has on finance and on the social consequences that follow.

Since this is a very complex field, you are of course not required to have deep knowledge of it. You can choose to explore one aspect in a bit more depth (e.g., issues around algorithmic trading or cryptocurrencies) or mention more angles in less detail. You will be graded on your ideas and argumentation, not on your financial knowledge.

Please make sure to use academic language and provide references for the claims you make. For instance, you can use the sources listed below, or you can find your own resources if you make sure that they are credible and scientific (Google Scholar might be a good starting point). Include at least three properly cited references in your answer.

A word of advice for reading scientific articles: you do not need to read dozens of articles end-to-end for this task. Always start by reading the abstract of a paper carefully to decide if the study could have the answer to your question and do not spend time on small practical details at this point (e.g., the methods of the study to the last detail). Identify the key points and with a critical mindset, use them to express your thoughts on the topic.

Some inspiration for papers you could use (each of them is freely accessible online):

[1] A. A. Kirilenko and A. W. Lo, 'Moore's Law versus Murphy's Law: Algorithmic Trading and Its Discontents', *Journal of Economic Perspectives*, vol. 27, no. 2, pp. 51–72, Feb. 2013, doi: 10.1257/jep.27.2.51.

[2] P. Howson and A. De Vries, 'Preying on the poor? Opportunities and challenges for tackling the social and environmental threats of cryptocurrencies for vulnerable and low-income communities', *Energy Research & Social Science*, vol. 84, p. 102394, Feb. 2022, doi: 10.1016/j.erss.2021.102394.

[3] S. Borms, K. Boudt, F. Van Holle, and J. Willems, 'Semi-supervised text mining for monitoring the news about the ESG performance of companies', in *Data Science for Economics and Finance: Methodologies and Applications*, Springer International Publishing

Cham, 2021, pp. 217–239.

2.2. Notes on the design process

As mentioned before, you have quite a lot of liberty in designing your own version of *iTrade*, which means that there will be several things that you will need to decide for yourself. These things include, but are not necessarily limited to:

- How you interpret “highest performer” stocks in a particular industry. For instance, you could consider only a given top percentile, or every stock that performs over the industry average, or maybe some other approach that you think is better suited for the task.
- What constitutes a “high ESG rating”. As in the previous point, you can choose to use a top percentile, cut off at a given minimum score, or go with a different approach. You might even decide that not all three categories are equally important and assign different weights to them.
- What happens if the filters return fewer stocks than the number the user specified that they wanted to see? It is possible that the preference settings limit the number of recommended stocks too much (they might even result in 0 returned stocks). You must decide how the algorithm should handle this scenario. Please make sure to be transparent: for instance, if you decide to relax the preference filters in this case, then you should communicate this clearly to the user in a printed message.
- To what extent do you want to include user input? One approach to handling the above questions is to let the user decide: it is great if you can incorporate user input to customize your algorithm, but you do not necessarily need to do that (although you should at least use user input to set the preference settings).

You might run into other open questions during the design process – that is usually not a problem! Feel free to be creative and implement an algorithm that you believe is the most suitable for the task. What is very important, however, is that you explain and justify your choices in the report: you must show that you made thoughtful decisions throughout the design process.

2.3. Notes on the flowchart and pseudocode

The purpose of the flowchart and the pseudocode is to give an easy-to-interpret overview of your algorithm. Therefore, while it is essential that you include every step that is relevant to the stock recommendation process, you do not need to include steps that are specific to the programming language (e.g., the way you read your dataset files into your Python program). In these tasks, the goal is to have a reproducible representation of your algorithm which could be implemented in any programming language.

2.4. Notes on the report

For the project report, you will need to follow the template provided to you on Canvas. In the report, you will complete the *Context Task*, report on the design process, and provide your flowchart, pseudocode, and Python code. There is no strict word count limit. For the *Context Task*, you should aim for at least 200-300 words, while describing the design process will probably be longer. The most important thing is that you give a well-rounded description of your thought process and explain any important decisions that you made along the way. Importantly, please reflect on the work division within the group at the end of the design process section.

3. Practical notes and tips

- The dataset of available stocks, which includes the attributes specified in Table 1 is provided to you in the file called .csv. You can read this file into your Python program as a dictionary, pandas dataframe, or other object which you are the most confident to work with.
- Make sure to present your stock recommendations in an appealing way. For instance, you could print the identifiers of the chosen stocks (in a ranked order!) in a nicely formatted message.
- **Consult the rubric** to see what requirements your project will be graded on and keep these in mind while you work.

Good luck and have fun!