

Chapter 1

Project and Goal

1.1 Executive Summary

The primary objective of this project is to provide users with a tool for studying and visualizing seasonality in financial markets, complemented by educational resources to enhance their understanding of the graphs and data. We will focus on financial assets exhibiting robust seasonal patterns and possessing substantial historical data, including indexes, commodities, forex, and handpicked stocks. Users will be warned against highly volatile assets, and recently listed assets will be excluded.

It's important to note that this project does not aim to offer investment or trading advice. Consequently, no tools for technical analysis or trend prediction will be provided.

1.2 Output

The final output will be a interactive web application featuring a search bar equipped with an efficient searching algorithm. This algorithm, given inputs such as the name, ISIN, or ticker symbol, will generate a price graph illustrating seasonality over a specified time frame in years. The web app will hide the graph if the selected ticker has been recently listed on the market. Additionally, it will alert users if the stock exhibits weak seasonality patterns due to volatility or irregularities over the years.

Furthermore, a second page, accessible via a link from the landing page, will host an educational section. This section will include a brief explanation of financial phenomena, a simple guide on interpreting the data and additional resources.

1.3 Motivations

This project originated from a need we frequently encountered, coupled with a interest in exploring seasonality patterns observed in various assets. Given the absence of comparable open-source tools enabling users to explore seasonality across diverse assets, we opted to conceptualize and develop our own web application.

Moreover, this project offers an opportunity to gain deeper insights into retrieving, manipulating, and visualizing financial data at the frontend level. It also serves as a learning platform for developing web applications using various programming languages, emphasizing the importance of

maintaining a distinct segregation between frontend and backend components, which are interconnected through APIs.

Chapter 2

Tasks e Resources

2.1 Method identification

Initially, we invested time in studying seasonality to discern which asset classes are more frequently influenced by this phenomenon. The project's key stages can be delineated into three main phases: downloading and manipulating the data, rendering the manipulated data into JSON format using FastAPI, and designing the frontend of the web application with FastAPI.

For detailed steps, please refer to the roadmap section.

2.2 Prerequisites

A fundamental understanding of financial markets, including the distinct characteristics among asset classes, their varying levels of volatility, and their responses to economic cycles, is crucial for this project. From a mathematical perspective, advanced skills are not required. The project involves basic manipulation of returns over different periods and calculating standard deviation to evaluate the strength of seasonality in given assets.

Proficiency in Python, particularly in manipulating dataframes using libraries such as NumPy and Pandas, is essential. Additionally, the ability to generate plots for the frontend of the web application is necessary. Familiarity with tools such as Matplotlib, FastAPI, HighCharts, as well as frontend development skills in JavaScript and ReactJS, are also required.

For the seasonality literature, we refer to this articles:

- [Seasonality Charts](#)
- [Seasonality on Financial Markets](#)

2.3 Frameworks, APIs and languages

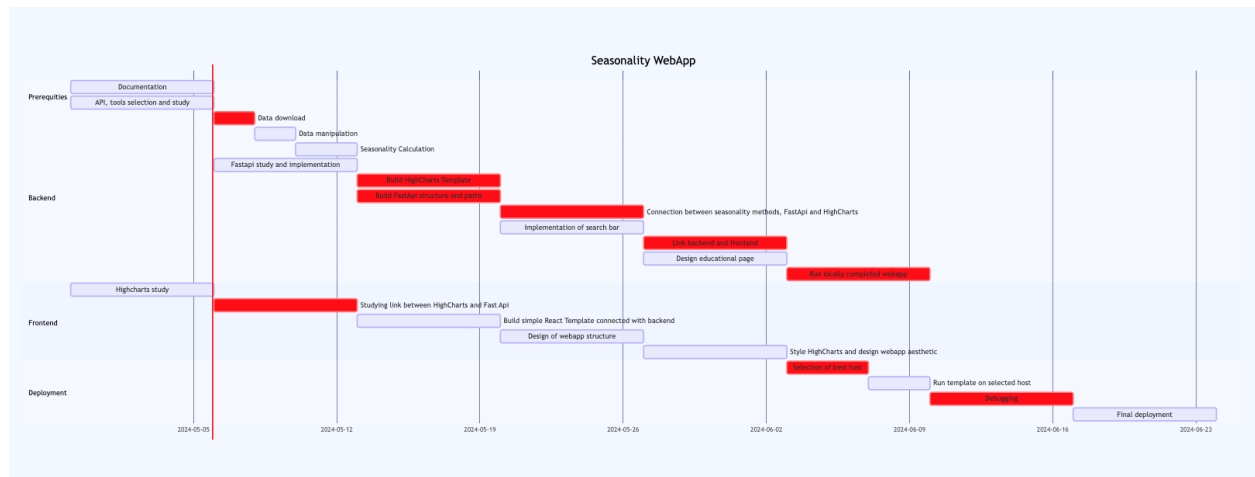
- Programming Language: Python, Javascript
- Source code editor: [Vscode](#).
- Libraries: Pandas, Numpy, Matplotlib, FastApi, ReactJs, High Charts.
- Paper writing platform: Overleaf & Zotero.

- Financial API: Twelvedata.

Chapter 3

Timeline

We created a Gantt Chart to provide details details on each step of the project. This tool will assist us in planning and monitoring activities over time. Every team member can refer to the chart to understand project phases in detail. Additionally, we will discuss resource and personnel allocation in each phase, identifying dependencies between activities and potential risk areas.



The project aims to launch the web app online before the conclusion of the academic semester at Politecnico di Milano and prior to the beginning of the exam session. Weekly meetings will be conducted to monitor project progress and task allocation.

3.1 Roadmap

We've meticulously outlined the project roadmap under the assumption that three team members will consistently work in parallel on various tasks, dividing the timeline into weekly phases. At the conclusion of each week, a meeting will convene to assess project progress and troubleshoot any issues.

During the initial week, our focus will be on writing the documentation, task allocation, and familiarizing ourselves with the tools and libraries integral to the project.

Subsequently, we will advance to developing Python methods for downloading and manipulating financial data. Concurrently, we'll use pre-existing modules developed within our organization to expedite progress. In this phase, we aim to write methods that can be repurposed for future projects, focusing on the importance of script modularity. Meanwhile, efforts will be dedicated to

exploring optimal strategies for implementing FastAPI and interfacing with ReactJS.

The creation of the HighCharts template holds paramount importance; by the third week's conclusion, a functional HighCharts graph must be in place to integrate into the frontend and commence template testing. During this phase, another team member will delineate the web application's structure concerning API calls and routing. This phase culminates in the integration of previously developed seasonality methods and API calls, resulting in a working prototype capable of locally processing requests and retrieving seasonal data.

The subsequent milestone consists in integrating the refined prototype with the frontend, previously designed using ReactJS. Meanwhile, another team member will focus on writing educational content and enhancing the user interface.

With these components in place, we transition into the deployment and testing phase. Collaboration among team members becomes more important as we navigate the complexities of merging various components. Local testing of the entire web application begins to identify bugs and address aesthetic concerns. Simultaneously, the selection of a hosting platform begins, with two members dedicated to local debugging while the third initiates the process of hosting a basic template online.

With the completion of these steps, the entire webpage is ready for online hosting and subsequent troubleshooting.

3.2 Report: Example of prototype

In the following link, you can explore a webpage featuring a possible prototype. Please note that we won't utilize the same financial APIs, libraries, and methods in the final project. [Prototype example](#)