Georgia Herron
z5060047

FLUX
Automated Asset Management

Product Design

Overview

Flux is an automated portfolio management product, designed to provide valuable, data-based insights to investors that helps investors more informed investment decisions. Its main feature is portfolio optimisation. The model is dynamic in design; it draws on multiple data streams to provide instantaneous, straightforward advice that helps customers minimise risk and maximise returns. The model provides recommendations provided by the model are based on a broad spectrum of market and sentiment indicators, news and heterogeneous data streams.

Information and resources pooled from a broad spectrum of market and sentiment indicators, news and heterogeneous data streams. Through harnessing the power of algorithms, flux automatically draws on multiple data sources and offers instantaneous, straightforward investment advice.

Interface

The interface of flux will be very user-friendly dashboard, with a simple, engaging design. The customer will be asked to specify three things before their recommendations are generated:

- Risk Tolerance
- Number of Scenarios to Run (Accuracy vs. Computational Time)
- Whether or not to include ML news sentiment analysis

This keeps it nice and simple for the user. There will be concise descriptions of all required inputs and outputs, and links to more information if the customer wants to know more. When the recommendations are generated, the dashboard will display a breakdown of portfolio recommendations, and graphs that display the data visually.

Usage

We recommend flux be used in parallel with research, as there is a limitation to the accuracy of machine learning. Flux is designed to point investors in the right direction and speed up the process of building a solid portfolio, allowing more time to do the things you want to do in life, or the things that machines cannot.

Branding

Our product is named after the only constant in life - *change*. Along with all things in life, equity markets are forever in a state of flux - they are constantly changing. With this fundamental law of the universe in mind, we knew we had to build a dynamic model. Our model can recognise change when it happens & adapts.

Key aims of the logo design: **Simple, Symbolic & Memorable**.

The fluctuating wave in our logo represents market volatility - we expect it.
In the words of Heraclitus (~500 BCE), "flux is life", meaning everything in life is constantly changing. With so many market forces at play, particular in the particularly volatile time we are living in, predicting the future for certain is impossible. However, the beauty of data is that we can tip the scale in your favour. Flux does the tedious, time-consuming work for you.

MAKE SURE YOU CHANGE README FILE LINK FOR THE "REPORT" IN ONEDRIVE!

ALSO, CHANGE PAGE NUMBERS AT THE END!!!

Input Features (continued)

Key data features that Flux's model requires are summarised below; these raw inputs are extracted, loaded and transformed in Station #1.

(i) Historical Equities Data

Equities time-series data stored in reliable online sources (e.g., the ASX) is extracted, loaded and transformed in Python, converting the inputted data into Pandas DataFrame - a format the computer can process and understand. The remaining steps in Station #1 are to ensure the data is clean, labelled and checked before it is passed on to Station #2 for model training. Key features of the historical equities data are as follows.

- **Columns** | Equity prices over time (i.e., tickers).
- **Rows** | Contains the last recorded price for each equity.
- **Index** | Dates the equity prices were recorded.

When considering the model's design, considerable emphasis was placed on the model's potential for scalability. I attempted to program it so the relevant data features were automatically loaded and processed by the model, requiring no manual Excel adjustments; this would be particularly important when more analysing many than 10 equities. One of Flux's main customer appeals is that it cuts down the tedious tasks investors are required to do, so building a model with a smart data loading algorithm was deemed to be a valuable feature.

(ii) Historical News Data

The ML model incorporates Natural Language Processing (NLP), seeking to give the machine some basic skills to interpret and extract meaning from human language. News history is fed to the model as a JSON file, with the format of {“Headline”: {“source”: “headline”}, ...}.

(iii) Client Risk Tolerance (or Preference)

Before generating recommendations, the client is required to provide the integer that represents their level of risk tolerance (or preference), with the scale of inputs ranging between 1 to 5, with 5 being the highest level of risk; this will be displayed as a slider bar on Flux's dashboard interface.

(iv) With/Without NLP

The client has the option to include or not include NLP in the recommendations. Note: enabling NLP heavily slows down the model's speed. This feature needs further optimisation if the model is to be successfully scaled up. Currently, it is not recommended to be used when a high number of scenarios is applied.

(v) Level of Accuracy

The client is asked to provide the level of accuracy that they wish to achieve in the recommendations. This question will be displayed on the dashboard as a scale ranging from "Minimum" (left) to "Maximum" (right), with "Maximum" meaning the model will generate the highest number of random portfolios when making the portfolio recommendations (more compute-heavy). The slider scale will have 10 interval options that the user can choose from, and there will be a small information icon that can provide the user with an explanation of how this will affect their recommendations. An explanation of this will be displayed on the dashboard to inform clients before they make the decision. Note: the specific number of portfolios that the model must generate for each user input (1-10) has not been determined yet. The model could simply request the user specifies the number of random portfolios to generate, but in my opinion, a min/max scale will be easier for clients to understand. Additionally, as all investors will most likely want the highest level of accuracy, this could be a feature accessible according to the fee you pay, with the higher fees enabling the Maximum option because if given the choice of all accuracy options, I expect that majority of customers would select Maximum.

Output Features

After processing the input features, Flux outputs a series of stock weights that allow the user to build an optimal portfolio, based on the data that the model has analysed. The key output features are:

• Historical stock price graphs for 10 equities in the outputted optimal portfolio. If possible, I would like to include the weight, return and volatility for each, however, this was hard for me to achieve in my Python code.
• Optimal weights for the optimised portfolio - depends on the client's inputted risk tolerance/preference.
• Optimal Sharpe Ratio Weightings (i.e. the best portfolio)
• Least risky portfolio

You may notice that news sentiment is not output, instead, this sentiment adjustment is incorporated into the core portfolio analysis and incorporated into the weighting recommendation.

Features Engineering

Feature Engineering refers to using domain knowledge and data exploration to refine our given data to more usable features; this (Station #2) is where the model is trained to learn how to recognise patterns within the data, as well as what to look for.

Input Features

After extracting, loading and transforming data in Station #1, there were two key choices made to improve the data. The first was normalizing price changes. The second was to remove news data related to Twitter as it was largely just adding noise to the model, for instance, tweets containing CBA often referred to the Chinese basketball association rather than Commonwealth bank.

Data Collection & Format Requirements

I put a strong emphasis on my code being reusable and easily extensible. Accordingly, l ensured that all data ingested from the given excels was done so programmatically. I have not specified any stocks when refining back to pricing. As a result, l could increase the stocks provided from 10 to 10,000 and need no changes, providing the column header PX_LAST remained the same. Given the excel stock data is from Bloomberg this could allow for easy linkage to a larger data set, or API. Similarly, the news data is in JSON format and so with some small tweaks could be ingested from an API.

Model Design

Model design involves building a model that can be used to predict equity performance and provide optimal asset allocation. In my product, this involves combining NLP to help predict equity returns and Modern Portfolio Theory to optimise portfolios for a client's risk preference.

The growing rate of technological advancement and FinTech competition means FinTech companies must be able to adapt, fast. The model is a flexible design, so it can be easily updated. A crux of our model design is flexibility - flex is designed to adapt.
Our goal is to help investors make smarter investment decisions by analysing the equity market and providing customers with data-based investment recommendations, helping them optimise their portfolios, minimise risks and maximise returns.

1.1 What model have you designed and aiming to use in the implementation?
Optimising portfolios is done by first generating the number of portfolios specified by the client with randomly allocated weights among the 10 stocks, using returns adjusted for news, and plotting their respective Sharpe ratios, expected returns and volatilities as seen in Figure 4. I then found the maximum Sharpe ratio within these portfolios as the optimal portfolio, assuming the 1-year Australian government bond return as the risk-free rate. The least risky portfolio was similarly found by finding the portfolio with minimum volatility that had been generated.

Client risk profiles are considered when determining the optimal portfolio for a client, with their risk preference corresponding to a range dependent on the volatility of the generated portfolios. Once a valid volatility range for a client is determined, the optimal Sharpe ratio within this range is found. For this reason, a client may select to be in the "Least Risk" category yet be recommended something with higher volatility than the "Minimum Variance" output. However, the Sharpe ratio has been maximised and so the client is still receiving reasonable returns given their risk preference. This is reflected in figure 3, where even though they have selected the least risk preference they have a 0.53% increase in volatility resulting in a 4.47% increase in returns over the least risky portfolio, a desirable choice for almost every investor. This method is quite robust and could foreseeably be used on a production version of the product. News sentiment can provide valuable insights into the more human factors at play in the market. For instance, recently Gamestock (GME) experienced huge price growth, largely driven by investor sentiment in Reddit. The ability to harvest this information, and then either instantly acts on it, or at least consider it, provides the opportunity to gain an information advantage above the rest of the market. In turn, if we can accurately quantify and apply this sentiment, we can expect to achieve superior returns. In this design, NLP is used to generate an adjustment to returns of between -5% and 5%, which is then applied to returns before random portfolios are generated. There is an option to disable NLP in the dashboard which may produce superior returns due to the restrictions outlined below.

Restrictions

Assumptions
• The portfolio can only be made up of the provided assets, there is no shorting or investing at the risk-free rate.
• News headlines directly correspond to price. That is, a positive headline always leads to an increase in price.
Data Spec10x Top 20 ASX stocks by mcap, 5 year history; Only stocks with full history included (e.g. exclude newly listed firms). Warnings: Survivorship bias when analysing historic data of current stock list; Historic performance is not necessarily indicative of future performance.
Raw Bloomberg data: Public holidays and non-trading days are filled with lagged data; Returns are percentages; Includes ASX/S&P200 index return.

Assumptions

Assumptions
• The portfolio can only be made up of the provided assets, there is no shorting or investing at the risk-free rate.
• News headlines directly correspond to price. That is, a positive headline always leads to an increase in price.

Computer Speed

Both portfolio generation and sentiment analysis are compute heavy and have not been optimised.

Accuracy

Modern portfolio theory is not overly accurate. However, I believe the real weakness of this product is the sentiment analysis prediction. This is made virtually impossible due to the highly competitive nature of the industry as well as the data limitations outlined previously. Predicting equity prices is a very difficult problem, but funds that have been able to accurately do so, Two Sigma, Optiver, SIG, etc. have been able to generate abnormal returns for their investors. Given the competitive nature of this industry, I think perhaps this is a problem that should be left until the very end of optimisation as it may not be solved.

1.4 What do you expect to see in model implementation?

1.5 Do you see any boundaries of the model?

1.6 How would you define Station #3?

FLUX

Automated Asset Management

Model Implementation

Model implementation leverages the models designed in station 3 to execute on data and provide a recommendation to our end users. In this product, this implementation is outputting the recommended weighting to hold in each stock, in the form of a web dashboard.

Steps Taken to Implement Flux

To implement this model, I applied the design outlined previously to the provided training data, initially outputting these results in terminal. However, I quickly realised this is an intimidating hurdle for the typical retail investor. As such l built a python dashboard and hosted it on http://python-asset-manager.herokuapp.com/. I further developed alternative wireframes using Figma as seen in figure 5 that convey what a complete product may look like.

Difficulties

The biggest difficulty l had was overcoming the previously mentioned restrictions to deliver a product, which may not be 100% accurate, but that gives a valuable insight into the application of technology to asset management. Additionally, determining how to incorporate client risk tolerance and news sentiment into the modern portfolio theory was a challenge, but ultimately l believe l utilized these inputs correctly. Finally, determining how to best display information in a graph was surprisingly challenging. I ultimately decided to show return and volatility as well as the weightings by hovering over any portfolio in the graph, which l feel helps educate the client.

Recommended Steps when Introducing Flux to Customers

When introducing clients to the tool l would walk them through the basics of finance, what return and volatility are, and the idea behind portfolio diversification. Additionally, l would explain the limitations of NLP and other tools or individuals that claim to have exclusive stock picks as even professional analysts often wrong (Figure 1). Finally, l would walk them through the impact of differing risk profiles and what Sharpe return indicates. Ultimately, this should encourage clients to invest responsibly in a diversified and high performing portfolio as recommended by Stonks.

Appendix

FLUX

Automated Asset Management

References

ASX 2021, *ASX200top10*, Australian Securities Exchange (ASX), viewed 10 August 2021, <https://www2.asx.com.au/>.

Input Features

The main function of the Flux product is Portfolio Optimisation, based on Modern Portfolio Theory. For the model to generate reliable client recommendations, it must be fed and trained with clean, reliable input data; Station #1 is responsible for this. It is where input data is extracted, loaded and transformed into a computable format that the computer can understand - a Pandas DataFrame. In this stage, the input data is cleansed and labelled, with the final stage of Station #1 being to check the input data loaded correctly and ensure it has been cleaned of impurities that may otherwise cause the model to make poor client recommendations. For this reason, the data cleaning stage in Station #1 is of critical importance; the model will can only ever be as good as the information it is fed. Only after passing the "test" can the DataFrame be sent onto Station #2 for model training.

Flux applies statistical and news sentiment analysis on historical stock price data, seeking patterns within the time-series data that help the model provide portfolio recommendations for customers. Supervised machine learning algorithms provide a set of "instructions", specifying what the model should look for within the data; these algorithms train the model and are what give it automation. The model's overarching purpose is to find meaning within input data (e.g., in the form of patterns or trends), which allows it to make educated predictions and informed data-based recommendations for clients.

The first stage (Station #1) is where input data is extracted, loaded and transformed into a computable format that the machine can understand. The model attempts to make sense of large amounts of input data, sourced from multiple online sources. Supervised ML algorithms contain "instructions" that teach the machine what to look for, and how to learn to recognise patterns within the data; this training is what gives the model automation. Flux is able to provide Portfolio Optimisation recommendations (the product's main feature) after analysing the input data. The model looks for patterns over time in both numeric and non-numeric data for equities so it can make an educated prediction of the future. Statistics is used to analyse the numeric data, which is historical stock price data recorded over time; this is what the model uses to make portfolio optimisation recommendations, which is the main feature of the product. An additional feature of the model is news sentiment analysis, which is made optional for the user. NLP on the greater of as well as (linguistic data) regarding the equity companies (that is available on the internet (e.g., news articles published on the companies and industries, etc.). which is drawn from multiple streams. The model converts the data returning it as a cleansed Pandas DataFrame, which has been labelled and sorted appropriately. The DataFrame is passed on to Station #2 only after the data is confirmed "fit" for model training; this is done simply by displaying some basic information. Confirming the model is correctly loading data is very important as recommendations will not be valid without reliable/pure input data.