International Journal of Web & Semantic Technology (IJWesT) Vol.16, No.1, January 2025

The Future of Financial Assistance: Leveraging LLMs for Personalized, Human-Like Interactions

Hamza Landolsi[a], Ines Abdeljaoued-Tej[a,b]

[a] *University of Carthage, Engineering School of Statistics and Information Analysis, Ariana, Tunisia*
[b] *Laboratory of BioInformatics bioMathematics, and bioStatistics (LR16IPT09), Institut Pasteur de Tunis, University of Tunis El Manar, 13, place Pasteur, B.P. 74, Belvédère, 1002, Tunis, Tunisia*

Abstract

Generative Artificial Intelligence (GenAI) is transforming the business landscape by enhancing accessibility, efficiency, cost-effectiveness, and innovation. This paper investigates the application of Large Language Models (LLMs) and GenAI in the financial sector, proposing a novel framework to reimagine robo-advisory systems. The framework shifts from traditional, rigid platforms to a more humanized approach that actively engages investors in a personalized asset selection process while leveraging LLMs to better understand their goals and profiles.

We present an end-to-end solution designed to address key limitations of conventional robo-advisors, such as inflexibility, restricted asset type offerings (typically limited to equities), and challenges in accessing high-quality, real-time data. The proposed architecture incorporates dynamic client profiling, risk aversion estimation, and portfolio optimization. A tailored asset selector agent, supported by robust data pipelines, ensures the curation of up-to-date market information. Through iterative development, we utilized prompt engineering and multi-agent workflows to refine user interactions and deliver actionable insights.

By implementing an innovative chatbot platform, we demonstrate the potential of LLMs to revolutionize customer service, enhance investor engagement, and provide strategic financial guidance. This study highlights the transformative impact of GenAI in creating more adaptive, personalized, and effective financial advisory solutions.

Keywords: Generative AI, Large Language Models (LLM), Big Data, Practical Applications, Agentic Design Patterns, Finance, Investment analysis, Portfolio Optimization

DOI: 10.5121/ijwest.2025.16102

1. Introduction

Beyond these immediate benefits, financial chatbots hold immense potential to revolutionize the banking landscape. By harnessing the capabilities of Large Language Models (LLMs), financial institutions can unlock new avenues for innovation and customer engagement. The advanced natural language generation provided by LLMs enables chatbots to make hyper-personalized recommendations, anticipate customer needs, and facilitate more nuanced conversations, allowing people to think outside the box and unleash their creative potential to solve more problems and increase the return on investment (ROI) of chatbots. Banks gain a strategic advantage by utilizing AI chatbots, powered by LLMs, to meet evolving customer preferences and competitive market demands. In addition to streamlining routine transactions, these chatbots enable banks to provide customized financial advice, predictive analytics, and seamless integration with other banking services. In addition, by prioritizing a customer-centric approach, banks can foster deeper connections with customers, building long-term loyalty and trust. As the banking industry continues to embrace digital transformation, the potential applications of financial chatbots with LLM are limitless presenting an exciting opportunity for banks to innovate and differentiate themselves in a rapidly changing market [1, 2, 3].

In the quest for more efficient and dynamic conversational interfaces, there is an urgent need to explore the potential of Generative Artificial Intelligence (GenAI), particularly in the area of text generation. Traditionally, manual design of conversational flows has been the norm, but with the advent of GenAI models, such as those recently added to Copilot Studio (formerly Microsoft PVA), there is an opportunity to completely rethink this process [4]. As a result, financial institutions, such as banks, have begun to explore the power of GenAI to solve their business use cases and work on solutions that align with their strategies and needs, such as revolutionizing the robo-advisor to provide more flexibility to their customers [5].

In this work, we took a hypothetical customer scenario of a real bank, as it is our clients. And we explored the opportunities offered by Copilot Studio, one of the powerful solutions offered by the Microsoft Power Platform. We have shifted our focus to how LLMs can evolve financial advisory applications from rigid robo-advisors to more interactive, flexible and customized solutions that meet the growing expectations of clients, while balancing asset performance and suitability with clients preferences and profile. Using prompt engineering techniques, we designed a real-time, robust, and scalable data collection and curation pipeline that allows our application to access the latest market data in a high-quality format. We can see the technical implementations in Appendix A. We then created a machine learning (ML) model to estimate the client's risk profile. We streamlined the customer profiling process with agentic workflow to ensure that we take the investor's profile into account in our interactive asset selection. After identifying the best potential performing assets that match the client's profile, we feed the data into our portfolio optimization engine.

2. Background

Artificial intelligence (AI) is a branch of computer science dedicated to developing systems that mimic human cognitive functions and behaviors. Deep Learning (DL) algorithms mimic the structure of the human brain, are effective at solving complex problems, as seen in image recognition and language translation technologies [6]. Generative AI (GenAI), a specialization of DL, can create diverse content based on learned patterns, while Large Language Models (LLMs) generate text that resembles natural language.

GenAI is not a new concept. Early examples date back to the pioneering work of Russian mathematician Andrey Markov in 1906 [7]. Markov chains, recognized as one of the foundational generative models, were originally employed for tasks such as next word prediction. However, their simple nature limited their ability to produce coherent text. The landscape has changed significantly over the years, with the advent of more powerful architectures and larger datasets. In 2014, generative adversarial networks (GANs) emerged [8], which use two models working together - one to generate output and the other to distinguish real data from the generated output. This approach, exemplified by models such as StyleGAN, significantly improved the realism of generated content [9]. A year later, diffusion models were introduced, which iteratively refine their output to generate new data samples that resemble the training data set. The development led by Stable Diffusion has played a crucial role in the generation of images that closely mimic reality.

In 2017, Google introduced the Transformer architecture [10], marking a significant advancement in natural language processing. Transformers represent each word as a token and create an attention map that illustrates the relationships between these tokens. This contextual awareness enhances the model's ability to generate coherent text, as demonstrated by large language models. Unlike discriminative models that focus on classification, generative AI models create new content by learning patterns and relationships from datasets generated by humans [6].

Generative AI is transforming various fields, with foundational models leading the charge. Models like ChatGPT exhibit impressive multitasking abilities, seamlessly performing tasks such as summarization, question answering, and classification [11]. Their versatility allows for easy integration into specific applications with minimal training and example data requirements.

Complementing this innovation is MFTcoder, an advanced multi-task fine-tuning framework designed specifically for code LLMs. By enabling simultaneous fine-tuning across multiple tasks, MFTcoder effectively addresses common challenges in model training, resulting in enhanced performance and efficiency compared to traditional methods. Seamlessly integrated with popular LLMs such as CodeLLama and Qwen, MFTcoder establishes a new benchmark in code generation, outperforming GPT-4 on benchmarks like HumaneEval [12, 13, 14].

2.1. Problematic

A financial advisor is typically someone who guides investment clients to provide them with insightful metrics, help them clearly identify their goals and set realistic expectations, and make recommendations on where to invest their money. This work requires a strong knowledge of financial markets, financial engineering skills, mathematics and statistics, AI and big data skills. Since the market is very volatile and always changing, automating this process is like a myth and can lead to losing money if AI is not used properly.

We focused on exploring LLM in prototyping a financial advisor application that helps investors build their portfolios based on their preferences, market trends, and asset analysis. First, the AI agent interacts with customers to perform customer profiling and understand their preferences. It sends data to the machine learning model used to get an estimate of the risk aversion (tolerance) coefficient. Then, based on natural language prompts, it selects investments from the investment data available in the database. In the context of a banking customer service chatbot, we may need to implement a custom RAG if:

- We require integration with our bank's proprietary systems or databases to retrieve customer information or perform transactions.
- We seek domain-specific knowledge or expertise to handle complex banking-related queries or tasks.
- We have to reach high accuracy or performance metrics for specific banking-related tasks, such as validating fraud detection before processing a transaction.

In this case, we have created a tailored and accurate conversation experience for bank customers.

2.2. Customer profiling

The client profiling part is essential for every new investor customer meeting with the financial advisor. In this first meeting, the financial advisor tries to understand the client's profile, savings, budget, assets already held, and many other relevant information that can give a better estimate of risk profile (for this part we used a machine learning model to assess the client's risk profile). Next, the financial advisor helps the investor clarify the main goal and make sure it is SMART (asks about any preferences the client would like the financial advisor to consider).[1]

[1] The majority of investors don't specify their goals, don't understand their investment objectives, or don't have clear expectations.

Defining a SMART goal is critical to setting realistic expectations, measuring the success of the portfolio, and intelligently selecting the assets that match the client's preferences, profile, and goal. To ensure the goal is SMART, we first present our proposed agentic workflow for guiding the investor client to define a clear and SMART goal.

(S) The goal should be **Specific**: The investor should clearly define the goal in a specific way. For exampl, planning for retirement, for children's education, for buying a car, etc. We prompted the LLM to extract the goal and based on that decide whether the goal is specific or not.

(M) The goal should be **Measurable**: By measurable, we mean first, a well-defined quantitative target (`I want to have 500 000$ for my retirement`, `The car costs 1 700 000$`, etc.). Second, the amount of money to be invested, this can be an initial investment (`I am willing to invest 8000 $ initially...`) or a monthly/annual investment (or contribution) like `I am able to contribute 500$ monthly to my investments`. The investor must define the target value and should clearly define at least one of the following: initial investment and monthly/annual contribution. We can accept both (the investor can invest with initial investment and/or monthly/annual contribution). The LLM should extract these values and decide whether the goal is measurable or not based on the presence of the specified values with if/else statements.

(T) The goal should be **Time bound**. This means that the investor should set a clear time frame (time horizon) for his investment goal. The time after that he should achieve his specific goal. Therefore, the LLM should clearly outline the time frame, if specified, and adjust the is-time value accordingly to either True/False.

(A) The goal should be **Achievable**! For example, the investor cannot invest 1000$ and expect to have a fortune of 1 000 000$ in a year. For this example, the goal is considered as not achievable.

For this task, we considered that the (R) which stands for **Relevant** in SMART is verified since we assume that the investor is clarifying a goal that aligns with his ambitions. But how can we decide if the future value of the investment meets the goal value? One of the limitations of LLMs is their mathematical reasoning abilities. LLMs are prone to errors when it comes to extracting complex financial values and independently calculating the results. We addressed this limitation by creating a ReAct framework [15] unit equipped with a pythonic function that acts like a calculator used by the LLM with a tool calling.

3. Data and Methods

In order to keep our database up-to-date and represent the actual market state and asset related information, we designed a near real world pipeline: data collection microservice that collects real-time data from financial data sources. We ensure that only new data is processed by the feature selection engine. We compute some important metrics, do some data validation checks to get the final gold layer [16]. Finally, the selected assets and the relevant information of the investor such as risk aversion coefficient are sent to the portfolio optimization microservice. This microservice provides an optimal portfolio and writes a detailed report explaining the choices and some useful visualizations via a friendly user interface.

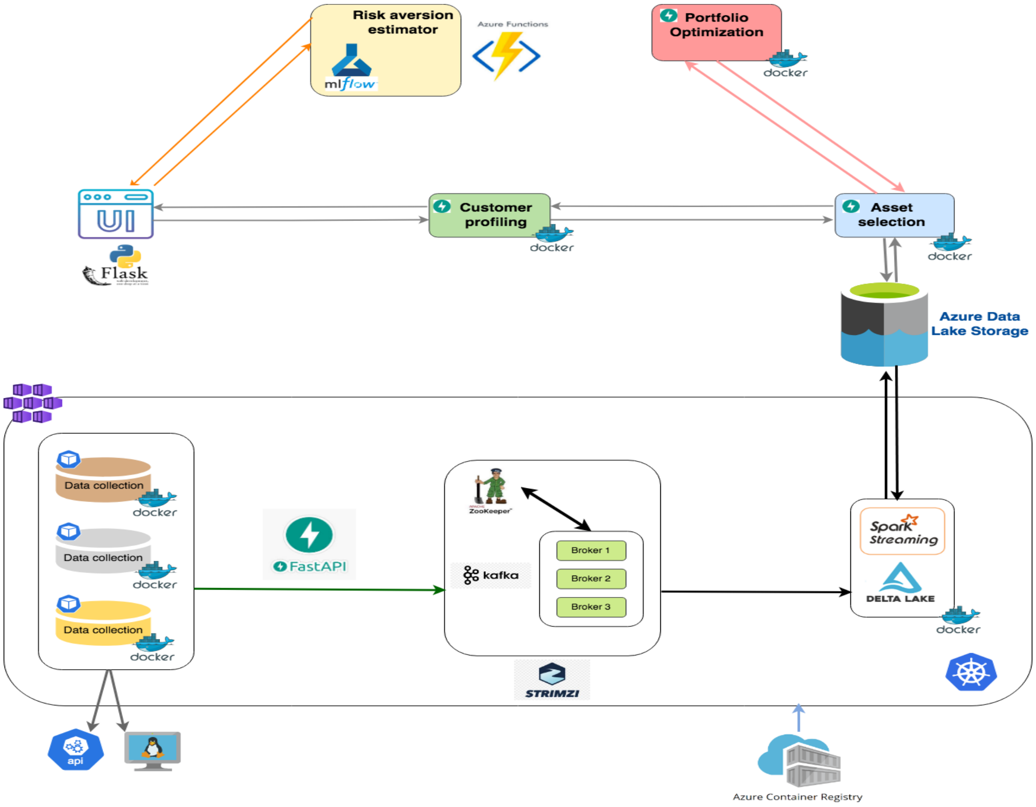

Figure 1: Financial advisor workflow of the proposed solution [17]

Designing the architecture for this solution requires a deep understanding of the workflows of quantitative development and financial engineering, as well as expertise in asset portfolio optimization. It also requires a strong understanding of big data technologies, software development, system design, networking, and cloud engineering. To ensure that asset data accurately reflects real-time market dynamics, the system includes a near real-time data pipeline. The data collection microservice is tasked with collecting information from diverse financial sources through more than 20 exposed endpoints that are activated by scheduled cron jobs every five minutes. This microservice scrapes real-time market data and passes it to Kafka - our message broker - with each data point routed to its corresponding topic. A high-level system architecture diagram, as shown in Figure 1, includes several microservices, including customer profiling, data collection, the Kafka message broker, data preparation, risk assessment, portfolio optimization, and asset selection.

3.1. Data

The proposed approach involves scraping financial data from Yahoo Finance and using state-of-the-art LLMs, including OpenAI's models and BloombergGPT, within a RAG framework. The data is pre-processed and analyzed to identify key financial indicators. Different LLMs are compared to determine the most effective model for generating investment scenarios. Optimization techniques are then applied to refine scenarios for different investor profiles, balancing potential ROI with risk. The nature of financial data is complex, influenced by a wide range of factors such as market sentiment, economic indicators, geopolitical events, and more. Building a unified pipeline that captures all of these nuances can be daunting. Indeed, the industry typically has teams focused on narrow areas, like a specific market, asset class or industry, like finance or health. To address the multiple challenges of financial data (volume, velocity, variety, data quality and integrity, etc.), we implemented a structured, tiered approach that follows the **Medallion Architecture**. It leverages a tiered data transformation process that moves from raw data (Bronze layer) to refined, enriched data (Silver layer) and finally to fully optimized and analytics-ready data (Gold layer).

3.1.1. Bronze Layer: Raw Data Ingestion and Storage

The bronze layer acts as the foundational layer of the architecture, where raw, unfiltered data is ingested and stored. At this stage, there is no cleaning or processing of the data, and the data is preserved in its original schema and format. This layer is critical for auditability, allowing us to trace any analysis back to the original data source. In our case, data is ingested via Kafka streams. Raw data can include price tickers, economic indicators, or market news articles.

At this stage, the data is stored in Azure Data Lake Storage (ADLS), which provides scalable, secure storage for large volumes of raw data. Partitioning is critical here, typically by asset type, to enable efficient retrieval and parallel processing of data.

3.1.2. Silver Layer: Data Cleansing and Enrichment

The silver layer is where the raw data is refined, cleaned, and enriched. This stage focuses on removing outliers, filling in missing data, and transforming the data into a more consistent and usable format. For example, corporate action events such as stock splits or dividend payments must be taken into account to avoid distorting historical pricing data.
Common transformations include:

- **Data Validation:** Financial data are sometimes inaccurate. We perform checks to remove any erroneous trades (e.g., price anomalies or bad ticks) using pre-defined rules based on market conditions.

- **Normalization:** This stage involves standardizing data from different sources and formats. For instance, currency conversions may be applied to normalize asset prices, or time zones may be aligned across global data feeds.

- **Feature Engineering:** Enrichment processes include the creation of technical indicators (e.g., moving averages, RSI), the calculation of financial ratios (e.g., P/E ratio), or even sentiment analysis based on news data. These features are critical inputs to risk models or trading algorithms.

3.1.3. Gold Layer: Analytics and Aggregation

The gold layer is the final stage where the refined data is aggregated, optimized, and made ready for high-value analytics, reporting, and machine learning models. At this level, the data is structured for our asset selector microservice, ready to be interacted with by LLM.

3.2. Asset selection

Once the client profiling phase has been successfully completed and the investor has confirmed their final preferences, the asset selector takes all the relevant profile information and natural language preferences and incorporates them into the asset selection process. We start with the manager agent. This is the one that distributes the tasks to each sub-manager or (asset level manager), see Figure 2. Here the asset manager decides on an execution plan based on the template provided by the `Manager`, taking into account the other agents. For the `Stocks manager`, we have introduced a more complex workflow that allows agents to interact with the stock search agent tool. First, the stock manager divides the preferences into CSV agent `Stocks CSV manager` tasks and `Stocks searcher` tasks. The stock manager can decide if the preference can be checked from the available dataset in the gold layer (this is done by passing the schema of the dataset in the prompt so that the agent know what features are available in the dataset), else these preferences should be verified through web-search tools.

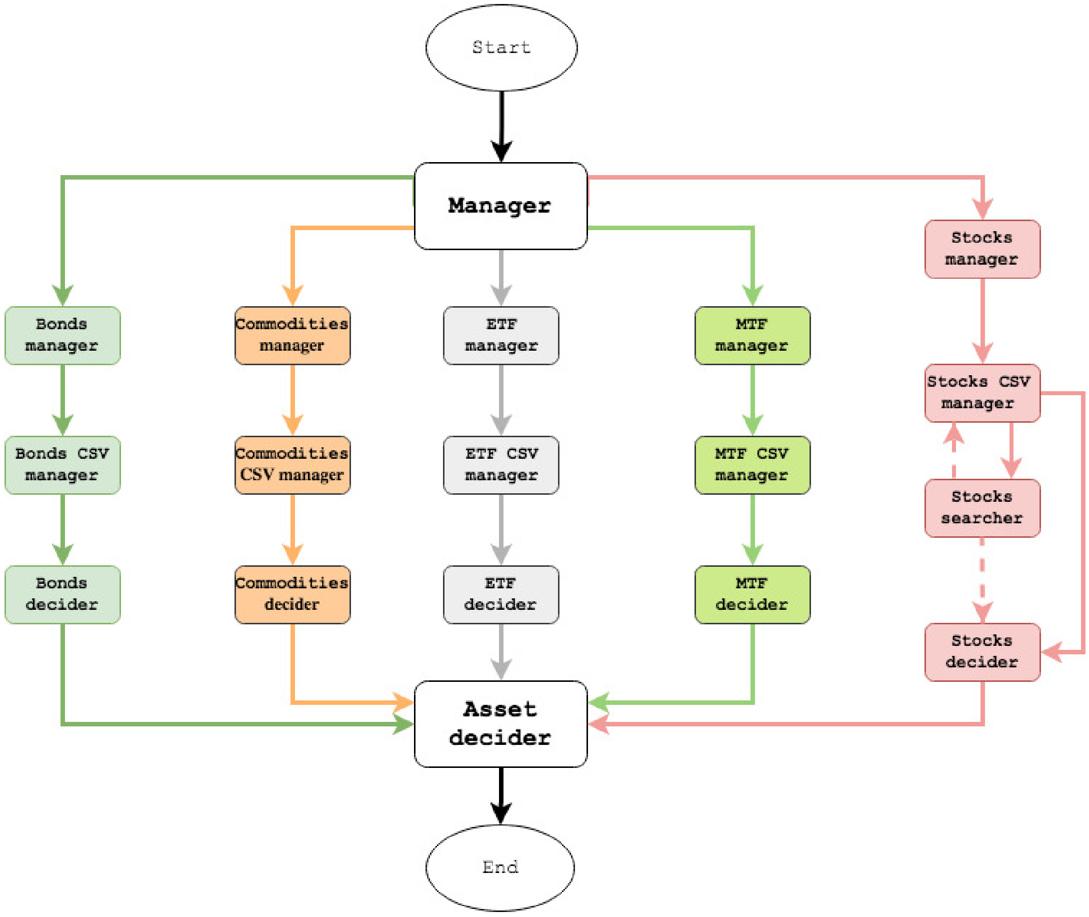

Figure 2: Asset selector agentic workflow (Source [17])

Manager Agent. The asset selection workflow takes as input the asset allocation proportions for each asset type, the preferences, and the risk aversion coefficient. First, the manager agent `Manager` checks if the preferences have some company granular level requirements, then it will reduce the mutual funds (MTFs) and ETFs proportions to 0. In other words the `Manager` won't route the assets to their respective asset managers.

Asset Manager Agent. This manager using ReAct framework plans and decides which tools it is going to use to pick assets with the given constraints. For instance, for the `Stocks manager`:

- The number of stocks to be selected should correspond to the investor's risk level (for example, if the client has an aggressive profile, he may be interested in small-cap companies, i.e. companies with a market capitalization of less than 300 million $).

- Parse preferences and decide whether to check them using search tools from external sources (basically the Internet) or using the internal data source (the gold layer obtained from the data curation pipeline).

3.3. Portfolio optimization

Portfolio optimization involves selecting the optimal portfolio — a collection of assets — that maximizes expected returns for a specified level of risk, or minimizes risk for a given expected return. Various methodologies can be employed to achieve this objective, including Mean-Variance Optimization, the Treynor-Black Model, and the Black-Litterman Model [18].

Developed by Harry Markowitz in the 1950s, **Mean-Variance Optimization** is a fundamental approach to portfolio construction [19]. It aims to create a portfolio that minimizes variance (risk) for a given expected return, or maximizes expected return for a given level of risk. The expected return of a portfolio (with n assets) is given by:

$$E(R_p) = \sum_{i=1}^{n} w_i E(R_i) \tag{1}$$

where $E(R_p)$ is the expected return of the portfolio, w_i is the weight of asset i in the portfolio, and $E(R_i)$ is the expected return of asset i. The variance of the portfolio is defined as:

$$\sigma_p^2 = \sum_{i=1}^{n} w_i^2 \sigma_i^2 + \sum_{i=1}^{n} \sum_{j \neq i} w_i w_j \sigma_{ij} \tag{2}$$

where σ_p^2 is the portfolio variance, σ_i^2 is the variance of asset i, and σ_{ij} is the covariance between assets i and j. The optimization problem can be framed as follows:

$$\min_{w} \sigma_p^2 \quad \text{s.t.} \quad E(R_p) \geq E(R_{target}) \tag{3}$$

where $E(R_{target})$ is the target expected return.

The **Treynor-Black Model** integrates active management with a passive portfolio approach. It provides a framework for combining the results of an active manager with a market portfolio. The model starts with two components: an active portfolio and a market portfolio. The expected return of an active portfolio can be expressed as:

$$E(R_a) = R_f + \frac{E(R_p) - R_f}{\sigma_p} \cdot \sigma_a \tag{4}$$

where R_f is the risk-free rate, $E(R_a)$ is the expected return of the active portfolio, and σ_a is the active risk (standard deviation of the active portfolio returns). The weights in the combined portfolio are determined using the following equations:

$$w_a = \frac{\frac{E(R_a) - R_f}{\sigma_a^2}}{\frac{E(R_a) - R_f}{\sigma_a^2} + \frac{E(R_m) - R_f}{\sigma_m^2}} \tag{5}$$

where $E(R_m)$ and σ_m are the expected return and standard deviation of the market portfolio. In this way, the Treynor-Black model helps to determine the optimal allocation between the active and passive portfolios on the basis of their respective risk-return profiles.

The **Black-Litterman Model** extends the Mean-Variance Optimization by incorporating subjective views into the asset return expectations. This model allows investors to adjust expected returns based on their beliefs while maintaining the equilibrium provided by the Capital Asset Pricing Model (CAPM). The key components of the Black-Litterman model are:

1. Equilibrium Returns. The model starts with a market equilibrium return vector:

   $$\mu = \tau \, \Sigma \, w_m \tag{6}$$

   where μ is the expected return vector, τ is a scalar that reflects the uncertainty of the prior, Σ is the covariance matrix, and w_m is the market weights.
2. Adjusting for Views. The investor's views on certain assets are incorporated as follows:

   $$\begin{bmatrix} \mu_{BL} \\ \mu_P \end{bmatrix} = \begin{bmatrix} \tau\Sigma w_m & 0 \\ 0 & \Omega \end{bmatrix}^{-1} \begin{bmatrix} \tau\Sigma w_m \\ \Pi \end{bmatrix} \tag{7}$$

   where μ_{BL} represents the adjusted expected returns, and Ω is the uncertainty associated with the investor's views.

The Black-Litterman model offers a flexible approach to portfolio optimization by combining market equilibrium with the investor's insights, resulting in a stable and reliable portfolio weights.

Additionally, advanced clustering techniques [20] improve portfolio diversification, while Monte Carlo simulations allow for testing portfolio performance under various future scenarios. This approach opens up the possibility of developing trading algorithms, moving beyond a sole focus on asset allocation.

4. Results

Building a robust financial data pipeline is both a technical and conceptual challenge. In the real world, unforeseen issues often arise, such as sudden regulatory changes that require modifications to the pipeline, or data vendor outages that disrupt the flow of critical market data. One of the most challenging issues is ensuring data consistency across asset classes and time zones, especially when dealing with global markets. Even a small mismatch in timestamps or missing corporate actions can lead to incorrect portfolio calculations and ultimately poor investment decisions.

By leveraging the Medallion architecture, we mitigate many of these challenges by isolating raw data, applying transformations, and producing analytically-ready data sets in a structured, auditable manner. This architecture not only provides scalability and flexibility, but also ensures the accuracy and reliability of the data consumed by financial models and reports.

The dataset encompasses 19 countries: France, Germany, Italy, Austria, China, the UK, India, Australia, Brazil, Canada, Japan, South Korea, Malaysia, the Netherlands, Sweden, Egypt, Qatar, Saudi Arabia, and South Africa. It places special emphasis on the US stock market while also including a diverse range of assets from various regions worldwide. This global representation is a key strength of our data, as it is essential for mitigating systemic risk by ensuring a comprehensive array of financial assets across different markets [21].

We also covered different types of financial instruments. This is not an exhaustive dataset for all types of financial assets - for example, we do not directly include cryptocurrencies here, although there are some mutual funds (MTFs) that create a portfolio of digital assets - but the idea is to deal with different types of assets as much as possible and try to mix and choose the best combination with different metrics for each asset to meet the needs and specific requirements of the investor. Tables Appendix B and Appendix C presented the schema for the key datasets stored in the gold layer. Each dataset contains multiple attributes that capture both raw and derived metrics relevant to financial markets. We evaluated various portfolio optimization techniques. We used the tickers symbols detailed in Table 1. In Table 2 we can find the allocated weights to each portfolio optimization model.

We can notice that in Table 2 the KO ticker for instance in Black-Litterman model has 0 as a weight. The Treynor-Black has a weight equal to 0.11, which is the third most important weight given by this method.

However, the DPZ ticker has a weight larger than 0.1 for all of the three models. This can be a good indicator of the importance of this asset given the profile of the customer (because we do not take into account only the metrics and the performance of the asset, we include some customer related constraints like his risk aversion coefficient).

Ticker	Description
AMZN	Amazon.com, Inc.: A multinational technology company focusing on e-commerce, cloud computing, and AI.
BAC	Bank of America Corporation: A leading multinational investment bank and financial services company.
COST	Costco Wholesale Corporation: A world leader in retail known for its members-only warehouse clubs.
DIS	The Walt Disney Company: A global entertainment and media conglomerate, specializing in film, TV, and theme parks.
DPZ	Domino's Pizza, Inc.: A multinational pizza restaurant chain known for its delivery and carryout services.
KO	The Coca-Cola Company: A leading beverage corporation, best known for its soft drink brand Coca-Cola.
MCD	McDonald's Corporation: A global fast-food company with operations in over 100 countries.
MSFT	Microsoft Corporation: A technology giant focusing on software, hardware, and cloud solutions.
NAT	Nordic American Tankers Limited: A shipping company specializing in crude oil tankers.
SBUX	Starbucks Corporation: A multinational chain of coffeehouses and roastery reserves.

Table 1: Descriptions of Tickers Utilized in the Portfolio

Ticker	Black-Litterman	Mean-Variance	Treynor-Black
AMZN	0.14138	0.16438	0.019488
BAC	0.15031	0.02390	0.006048
COST	0.05212	0.16444	0.020219
DIS	0.05504	0.03166	0.047210
DPZ	0.15471	0.22786	0.151021
KO	0.00000	0.05204	0.119128
MCD	0.09987	0.09086	0.086184
MSFT	0.13623	0.14015	0.077446
NAT	0.11765	0.00000	0.206404
SBUX	0.09270	0.10472	0.052783

Table 2: Returned weights of a portfolio: Black-Litterman, Mean-Variance, and Treynor-Black

5. Discussion and challenges

The financial industry continues to look for ways to innovate and improve investment decisions. The integration of RAG and LLM offers a promising approach to analyze large amounts of financial data and generate tailored investment scenarios. The aim of this study is to use these advanced techniques to improve the formulation of investment strategies and provide clear insights into the return on investment (ROI) and risk for different investor profiles.

The findings of our study underscore the critical role of accurately estimating the risk aversion coefficient in financial advisory and asset allocation. By leveraging machine learning techniques, financial advisors can obtain a more nuanced understanding of individual client preferences, leading to tailored investment strategies that align with each client's risk profile. Our approach is a solution when time matters and we need real time inference of the model, by incorporating a broader range of predictive variables and allowing for the dynamic modeling of risk aversion. This not only enhances the reliability of risk assessments but also empowers financial advisors to make informed decisions that can significantly impact their clients' investment outcomes in an agile way.

Future research should explore the integration of behavioral finance principles, market news sentiment and economic indicators into machine learning models to further enhance the accuracy of risk aversion predictions (available data is always a limitation to start exploring these avenues). In addition, the incorporation of real-time market data could provide a more comprehensive framework for assessing investor risk tolerance in an ever-evolving financial landscape. Data bias refers to the presence of biases and imbalances in the training data that lead to skewed model outputs. Limited world knowledge and hallucination occur when language models lack a comprehensive understanding of real world events and tend to fabricate information. In general, the performance of language models is highly dependent on the quality and representativeness of the training data. Ethical concerns revolve around the responsible and ethical use of language models, especially in sensitive contexts. The issue of bias amplification is prominent, as biases present in the training data may be exacerbated, leading to unfair or discriminatory results. There are potential legal and copyright issues related to generated content that may infringe on copyrights or violate laws. User privacy is a key concern here, particularly the risks of generating text from user input which might contain confidential or sensitive information.

Computational resources are a major concern due to the significant computational power required. There are interpretability challenges in understanding and explaining the decision process of complex models. Evaluating models across tasks and domains is poorly designed, adding further complications. In addition, there are fine-tuning challenges in adapting pretrained models to specific tasks or domains. Maintaining coherent contextual understanding over longer passages or conversations is another difficulty. Language models are also vulnerable to adversarial attacks, where deliberate manipulation of input data can lead to incorrect outputs [22]. Finally, maintaining long-term context and coherence over extended text or conversations is an ongoing struggle.

There are several key issues to consider when deploying language models in production environments. First, scalability is critical to ensure that the model can efficiently handle increased workloads and demand. Latency must be minimized to ensure fast and efficient interactions. Robust monitoring and maintenance systems must be implemented to track model performance and perform regular maintenance. Smooth integration with existing systems such as software, databases, and infrastructure is essential. Cost management strategies are important to optimize the costs associated with deploying and maintaining large language models. Addressing security concerns by mitigating potential vulnerabilities and risks is critical. Ensuring interoperability with other tools, frameworks, or systems is another important consideration.

Developing mechanisms to incorporate user feedback for continuous improvement is important to the effectiveness of the model. Adherence to regulatory compliance requirements and standards is mandatory.

Finally, effectively managing the generation of text in dynamic content-handling environments is also an important consideration. Corrective RAG [23] and RAPTOR [24] are some of the very different proposed solutions to adapt RAG frameworks. For the asset selection process, we could potentially use Variational Auto-encoders (VAE) [25]. Another direction in this study could potentially benefit from real-time data signals, using advanced time series analysis or using Kalman filters to extract powerful insights from market fluctuations [26].

6. Conclusion

We designed a near real-time data collection pipeline, refined the large amounts of data collected, considering the importance of time in leveraging insights with current data. We improved the process of completing questionnaires and conducting face-to-face meetings to clearly define risk aversion, client profile and understand investor preferences, which a financial advisor used to do manually. In addition, we tested another agent-based workflow that is able to select investments based on the client's natural language preferences, giving the customer full control over the selection of assets. This manual selection process is always a challenge for investors with values, ethical and environmental concerns.

This work can be further enhanced by using the latest RAG solutions, Agentic RAG, Agentic Chunking, to further refine our LLM used for generation on financial and banking corpus and give it a more customized persona. This solution has the potential to significantly impact the future of finance by addressing concerns around scalability, privacy and customer trust. It could reshape the way customers interact with financial services by offering a more engaging, dynamic and responsive alternative to traditional robo-advisors.

Acknowledgements

The authors are grateful to Kais Letaief and Nizar Taghuothi for their valuable suggestions and enlightening comments. Their contributions have been a great asset to the quality of this work. Additionally, the authors wish to express their sincere gratitude to the anonymous reviewers for their constructive comments and thorough evaluation.

Appendix A.

To give an appealing User Interface serving as demo purpose only, we orchestrated all the microservices discussed above in a final solution. It can be accessed at `https://github.com/Andolsi-Hamza05/financial-advisor`.

Appendix B. Data preparation

We cannot rely on assumptions because we are dealing with highly dynamic data sets. Everything is volatile and changing rapidly. Asset names per instance can change from day to day. Many companies can open or crash every day. After the data is curated, we extract the symbol (unique identifier for each asset) and scrape its historical data using the `yfinance` library to compute more relevant metrics and check the integrity of the asset from its past behavior.

Metric	Definition	Equation	Asset Types
Yld	Yield is defined as the income generated from an investment, typically expressed annually as a percentage of the investment's cost or market value.	$\text{Yield} = \frac{\text{Annual Income}}{\text{Investment Cost or Price}}$, where Annual Income is equal to Coupon Payment; Investment Price is equal to Bond Price	Bonds
YoY	A method of evaluating the performance or growth of an asset by comparing data from one year to the next.	$= \frac{100 \times (\text{Current Value} - \text{Previous Value})}{\text{Previous Value}}$, where Current Value is equal to Current Price; Previous Value is equal to Price a Year Ago	Bonds, Commodities
YTM	Yield to Maturity is the total return expected if the bond is held to maturity, including interest and capital gains.	$YTM = \frac{C + \frac{F-P}{t}}{\frac{F+P}{2}}$, where C = Annual Coupon; F = Face Value; P = Bond Price; t = Years to Maturity	Bonds
MD	Max Drawdown is the maximum observed loss from a peak to a trough before a new peak is attained.	$\text{MD} = \frac{\text{Trough Value} - \text{Peak Value}}{\text{Peak Value}} \times 100$, where Peak Value = Highest Value; Trough Value = Lowest Value	Commodities
SMA	Short-Term Moving Average: The average of an asset's price over a short period of time.	$\text{SMA} = \frac{P_1 + P_2 + \cdots + P_n}{n}$, where P_i = Price on Day i; n = Short Time Frame (e.g., 50 days)	Commodities, Stocks
LMA	Long-Term Moving Average: The average of an asset's price over a long period of time.	$\text{LMA} = \frac{P_1 + P_2 + \cdots + P_n}{n}$, where P_i = Price on Day i; n = Long Time Frame (e.g., 200 days)	Commodities, Stocks
MAR	Monthly Average Return: The average return on an asset per month.	$\frac{R_1 + R_2 + \cdots + R_m}{m}$, where R_i = Return in Month i; m = Number of Months	Commodities

Continued on the next page

Metric	**Definition**	**Equation**	**Asset Types**
3YAR	The average return of an asset over a 3-year period.	3-Year Avg Return $= \frac{R_1 + R_2 + \cdots + R_{36}}{36}$, where R_i = Return in Month i; 36 months	Commodities
P/E	Price-to-Earnings Ratio compares the price of a stock to its earnings per share (EPS).	$P/E = \frac{P}{E}$, where P = Price per Share; E = Earnings per Share	Stocks
P/B	Price-to-Book Ratio compares the market value of a stock to its book value.	$P/B = \frac{\text{Market Price}}{\text{Book Value}}$, where Market Price = Current Stock Price; Book Value = Net Asset Value	Stocks
SR	Sharpe Ratio measures risk-adjusted return, comparing excess return to the volatility of an asset.	$SR = \frac{R - R_f}{\sigma}$, where R = Asset Return; R_f = Risk-Free Return; σ = Standard Deviation	Stocks
ER	The Expense Ratio is the total cost of managing an MTF or ETF, expressed as a percentage of the average assets under management.	$ER = \frac{\text{Annual Operating Expenses} \times 100}{\text{Average Fund Assets}}$,	MTFs, ETFs

Appendix C. Gold Layer

Dataset	**Field Name**	**Data Type**	**Description**
Stocks	Symbol	String	Unique identifier for the stock.
	Company Name	String	Full name of the company issuing the stock.
	Real-time Price	Float	Current market price of the stock.
	Change	Float	Percentage change in stock price over a given period.
	Volume	Integer	Number of shares traded during the day.
	Avg Volume	Integer	Average daily volume over the last 3 months.
	Market Capitalization	Float	Market value of the company in USD.
	P/E Ratio	Float	Price-to-earnings ratio.
	Country	String	Country where the company is headquartered.
	Sector	String	Industry sector of the company.
	SMA	Float	Simple moving average over a selected window.
	P/B	Float	Price-to-book ratio.
Bond funds	Name	String	Fund's name.
	Symbol	String	Unique identifier for the bond fund.
	SEC 30 Day Yield	Float	Fund's yield based on the SEC-defined methodology.
	TTM Yield	Float	Fund's trailing twelve-month yield.
	Avg Effective Duration	Float	Sensitivity of the fund to interest rate changes.
	Total Return (1 Year)	Float	Fund's annual return over the last year.
	Total Return (3 Years)	Float	Fund's average annual return over 3 years.
	Adjusted Expense Ratio	Float	Cost of managing the fund.
	AUM	Float	Total assets under management in USD.

Continued on the next page

Dataset	**Field Name**	**Data Type**	**Description**
	Category	String	Fund's investment category.
Equity funds	Company Name	String	Name of the issuing company.
	Ticker	String	Unique identifier for the equity fund.
	Total Return (1 Year)	Float	Annual return over the last year.
	Total Return (3 Years)	Float	Average annual return over 3 years.
	Adjusted Expense Ratio	Float	Fund's management fee.
	AUM	Float	Assets under management in USD.
	Category	String	Investment category of the fund.
	Fund Type	String	Type of the fund (e.g., growth, income).
	Region	String	Region of investment focus.
Alternative funds	Name	String	Fund's name.
	Symbol	String	Fund's symbol.
	Medalist Rating	Float	Fund's rating by analysts.
	Morningstar Rating	Float	Star rating assigned by Morningstar.
	Total Return (1 Year)	Float	Fund's 1-year return.
	Total Return (3 Years)	Float	3-year return.
	Total Return (5 Years)	Float	5-year return.
	Adjusted Expense Ratio	Float	Fund's management fee.
Commodities	Commodity	String	Name of the commodity.
	Total Return	Float	Annual return of the commodity.
	Monthly Avg Return	Float	Average monthly return.
	Volatility	Float	Standard deviation of returns.
	Max Drawdown	Float	Maximum observed decline from peak value.
	Short-Term MA	Float	Short-term moving average.
Bonds	Country	String	Country of issuance.
	Yield	Float	Bond yield in percentage.
	YTM	Float	Yield to maturity.
	YoY	Float	Year-over-year change in yield.
	Maturity Date	Date Time	Bond's maturity date.

References

[1] A. L. Eisfeldt, G. Schubert, M. B. Zhang, Generative ai and firm values, Tech. rep., National Bureau of Economic Research (2023).

[2] K.-B. Ooi, G. W.-H. Tan, M. Al-Emran, M. A. Al-Sharafi, A. Capatina, A. Chakraborty, Y. K. Dwivedi, T.-L. Huang, A. K. Kar, V.-H. Lee, et al., The potential of generative artificial intelligence across disciplines: Perspectives and future directions, Journal of Computer Information Systems (2023) 1–32.

[3] H. Ali, A. F. Aysan, What will chatgpt revolutionize in the financial industry?, Modern Finance 1 (1) (2023) 116–129, DOI: 10.61351/mf.v1i1.67.

[4] Microsoft Corporation, Reinvent your AI assistants with generative answers, actions, and more in Microsoft Power Virtual Agents, Microsoft Copilot Studio (2024).

[5] G. Cardillo, H. Chiappini, Robo-advisors: A systematic literature review, Finance Research Letters (2024) 105119.

[6] B. Decardi-Nelson, A. S. Alshehri, A. Ajagekar, F. You, Generative ai and process systems engineering: The next frontier, arXiv:2402.10977 (2024).

[7] H. K. Hamarashid, S. A. Saeed, T. A. Rashid, A comprehensive review and evaluation on text predictive and entertainment systems, Soft Computing (2022) 1–22.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Communications of the ACM 63 (11) (2020) 139–144.

[9] T. Karras, S. Laine, T. Aila, A style-based generator architecture for generative adversarial networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 4401–4410.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30, curran Associates, Inc. (2017).

[11] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung, et al., A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity, arXiv:2302.04023 (2023).

[12] B. Liu, C. Chen, Z. Gong, C. Liao, H. Wang, Z. Lei, M. Liang, D. Chen, M. Shen, H. Zhou, et al., Mftcoder: Boosting code llms with multitask fine-tuning, in: Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2024, pp. 5430–5441.

[13] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, et al., Code Llama: Open foundation models for code, Preprint arXiv:2308.12950 (2023).

[14] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al., Qwen technical report, Preprint arXiv:2309.16609 (2023).

[15] S. Yao, J. Zhao, D. Yu, I. Shafran, K. R. Narasimhan, Y. Cao, React: Synergizing reasoning and acting in language models, in: NeurIPS'2022 Foundation Models for Decision Making Workshop, 2022, pp. 1–33.

[16] I. F. Jephte, Extract, transform, and load data from legacy systems to azure cloud, Master's thesis, Universidade NOVA de Lisboa (Portugal) (2021).

[17] H. Landolsi, I. Abdeljaoued-Tej, From rigid robo-advisors to human-like interactions: Revolutionizing financial assistance with llm-powered solutions, in: Computer Science & Information Technology (CS & IT), ISSN : 2231 - 5403, Vol. 15 (02), January 2025, pp. 157–168.

[18] J. Sen, A. Dutta, Portfolio optimization for the indian stock market, in: Encyclopedia of Data Science and Machine Learning, IGI Global, 2023, pp. 1904–1951, DOI: 10.4018/978-1-7998-9220-5.

[19] H. M. Markowitz, G. P. Todd, Mean-variance analysis in portfolio choice and capital markets, Vol. 66, John Wiley & Sons, 2000.

[20] W. Tang, X. Xu, X. Y. Zhou, Asset selection via correlation block-model clustering, Expert Systems with Applications 195 (2022) 116558, https://doi.org/10.1016/j.eswa.2022.116558.

[21] N. Remolina, Generative ai in finance: Risks and potential solutions, Finance: Risks and Potential Solutions (November 9, 2023). Singapore Management University School of Law Research Paper Forthcoming, SMU Centre for AI & Data Governance Research Paper Forthcoming (2023).

[22] M. Bartolo, A. Roberts, J. Welbl, S. Riedel, P. Stenetorp, Beat the ai: Investigating adversarial human annotation for reading comprehension, Transactions of the Association for Computational Linguistics 8 (2020) 662–678.

[23] S.-Q. Yan, J.-C. Gu, Y. Zhu, Z.-H. Ling, Corrective retrieval augmented generation, Preprint arXiv:2401.15884 (2024).

[24] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, C. D. Manning, Raptor: Recursive abstractive processing for tree-organized retrieval, Preprint arXiv:2401.18059 (2024).

[25] S. A. Hosseini, S. A. Hosseini, M. Houshmand, Variational autoencoder-based dimension reduction of ichimoku features for improved financial market analysis, Franklin Open 8 (2024) 100135, https://doi.org/10.1016/j.fraope.2024.100135.

[26] Y. Kim, H. Bang, et al., Introduction to kalman filter and its applications, Introduction and Implementations of the Kalman Filter 1 (2018) 1–16.