Project Report: GenAI Advisor, based on the Financial Advisor Bot template

Andreas Mallios
BSc Computer Science (ML and AI)
Student number: 200128357

June 2025

Contents

1 Introduction

Artificial Intelligence (AI) continues to reshape global industries, with Generative AI (GenAI) standing out as one of the most influential recent developments. GenAI models, such as large language models (LLMs) and diffusion architectures, have driven widespread adoption and significant investment, particularly in the technology sector. Yet, for retail investors, the tools available to access this growth remain largely generic and inaccessible, both in terms of domain focus and interpretability.

Private equity and venture capital investments in GenAI startups are becoming larger and more targeted as investor strategies evolve with the technology. Funding in GenAI exceeded $56 billion in 2024, nearly double the $29 billion recorded in 2023 (14).

As shown in Figure 1 below, while the number of funding rounds declined in 2024, the overall deal value surged. This suggests a consolidation of capital into larger, infrastructure-oriented bets, aligning with investor confidence in scalable GenAI deployments. These trends highlight the need for tools that help investors evaluate publicly listed companies best positioned to benefit from this structural shift.

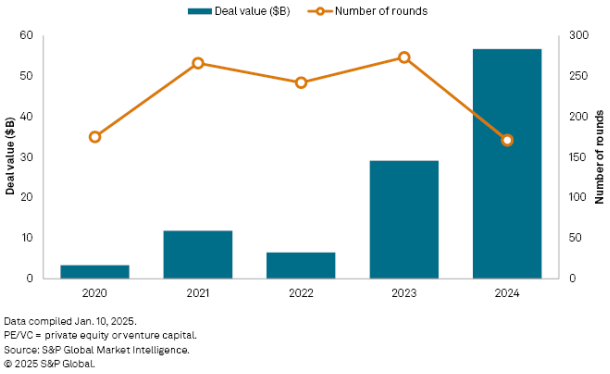

Figure 1: PE/VC investments in GenAI, 2020–2024. Source: S&P Global Market Intelligence (14).

This project responds to that gap by proposing the **GenAI Advisor**, a system that delivers explainable stock recommendations specifically for companies driving or enabling the GenAI revolution. Based on the *Financial Advisor Bot* template, the system is designed for non-technical users, providing accessible, transparent investment signals that go beyond traditional model outputs.

1.1 Concept

The GenAI Advisor is conceived as a personalised financial guidance tool for individual investors interested in the GenAI sector. It provides interpretable recommendations, whether to buy, hold, or sell specific stocks, alongside human-readable explanations for these decisions. Unlike traditional investment tools that obscure decision logic behind opaque algorithms or technical jargon, this system foregrounds explainability and domain relevance.

The intended users of this system are individual retail investors interested in emerging technology themes but lacking the technical background to interpret raw financial signals or operate sophisticated trading tools. These users are typically non-specialists with limited exposure to algorithmic finance or machine learning models. For this group, the GenAI Advisor offers an accessible tool to support their decisions, that reduces complexity while preserving analytical rigour. It is also suitable for students, educators, or early-stage investors exploring explainable AI in financial contexts.

At its core, the GenAI Advisor is built on the belief that financial tools should be not only accurate but also comprehensible. This is particularly important in thematic investing, where

users are drawn not only by quantitative performance but by interest in specific technological trends. In this context, the system acts as a digital intermediary between complex market signals and users' intuitive understanding, offering advice that is grounded and traceable, with a clearly scoped set of companies.

The conceptual novelty lies in its hybrid orientation; it combines evidence-driven financial analysis with a commitment to interpretability. The Advisor does not aim to outperform the market through high-frequency trading (HFT) or proprietary forecasting. Instead, it prioritises accessibility, domain focus, and trustworthiness, delivering insights users can understand and apply in their decision-making. In doing so, it invites users into the analytical process rather than substituting for it.

Furthermore, the GenAI Advisor embraces thematic integrity by carefully curating its stock universe. It avoids general-purpose technology firms in favour of those with demonstrable involvement in the GenAI ecosystem, whether through infrastructure, tooling, model development, or deployment. The rationale for company inclusion and exclusion is detailed in the project scope (Section 1.3). This conceptual grounding ensures that recommendations are not only technically derived but also strategically aligned with the user's thematic intent.

1.2 Motivation

The motivation for this project is both technical and user-oriented. GenAI represents an emerging thematic investment opportunity that remains underserved by existing tools. Simultaneously, there is increasing demand for transparent and accessible AI systems tailored to the needs of retail investors in the financial domain.

Traditional robo-advisors, such as Nutmeg and Moneyfarm, provide automated portfolio allocation but are domain-agnostic and opaque in how decisions are made. While open-source trading bots like Freqtrade offer flexibility, they require technical expertise and lack natural language interfaces. StockGPT (9) demonstrates the potential of generative models in stock prediction, while recent work on XAI-integrated forecasting (10) highlights the importance of trust and interpretability in financial AI systems.

By combining simple rule-based logic with predictive models and natural language explanations, the GenAI Advisor aims to deliver recommendations that are not only accurate but also intelligible. A local LLM serves as an interpretive layer rather than a decision engine, ensuring that outputs remain grounded in logic and statistical inference, while still being comprehensible to the user.

Beyond usability, the project contributes to responsible AI by ensuring data privacy and reproducibility. No personal or sensitive data is collected, and all sources are public. The LLM is hosted locally to avoid transmitting user inputs externally. The system is explicitly intended for educational and demonstrative use, and includes appropriate disclaimers to distinguish it from regulated financial services.

1.3 Scope

The GenAI Advisor focuses on a curated set of publicly traded companies whose business models are materially driven by Generative AI. The portfolio includes firms across three categories: (1) software developers and strategic investors (e.g., Microsoft, Alphabet, Meta, Amazon); (2) hardware providers (e.g., Nvidia, AMD, Intel, Marvell); and (3) infrastructure suppliers (e.g., TSMC, SK hynix, Broadcom, Arista Networks). These companies were selected for their direct contributions to GenAI model development, deployment, or enablement.

To maintain thematic focus and ensure data availability, the system excludes private companies, indirect holdings, and general technology firms lacking core GenAI offerings (e.g., Apple, Netflix). Chinese firms (e.g., Alibaba, Tencent, Baidu) are also excluded due to market access

limitations (16) and restricted financial transparency due to the Variable Interest Entity (VIE) structure of the BATX (Baidu, Alibaba, Tencent, and Xiaomi) stocks (4).

This scope ensures the system operates within a clearly bounded, analytically defensible domain. It enhances the relevance of recommendations by aligning them with user intent and facilitates reproducible evaluation by focusing on transparent and publicly accessible data.

2 Literature Review

The rapid advancement of Generative AI (GenAI) and its recent commercialisation in late 2022, has sparked significant interest in its application to financial forecasting and investment support systems. Despite notable progress in predictive modelling, current financial advisory tools often prioritise performance metrics at the expense of transparency and interpretability. This trade-off presents a particular challenge for non-technical retail investors, who require not only accurate but also comprehensible recommendations in order to make informed decisions.

This literature review examines recent developments at the intersection of GenAI, explainable AI (XAI), and algorithmic trading. It explores generative models for financial prediction, the integration of XAI techniques in forecasting systems, and the implications of user-centred design in FinTech. The aim is to critically evaluate existing approaches and identify a gap in the current landscape: the lack of accessible, domain-specific, and explainable investment tools. The review concludes by outlining how the *GenAI Advisor* project is designed to address this gap through a hybrid architecture that integrates transparent rule-based logic, supervised machine learning, and natural language explanation layers.

2.1 Generative AI in Financial Forecasting

The application of Generative AI (GenAI) in financial domains has emerged as a frontier area of research, particularly in the context of algorithmic trading and decision support. While traditional predictive systems rely on time-series models and supervised learning techniques, recent work has demonstrated the viability of large language models (LLMs) and generative transformers for tasks such as stock sentiment analysis, earnings prediction, and trade signal generation.

A notable example is *StockGPT* by Mai (9), which applies a fine-tuned generative language model to extract predictive signals from financial text and historical data. The system shows promising results in backtesting but offers limited transparency in terms of feature attribution and decision rationale. This highlights an emerging trade-off in GenAI systems between performance and interpretability, particularly in high-stakes domains like finance.

Generative models have also been used to simulate synthetic market data (15), enabling improved model training under data scarcity. However, these approaches often lack integration with rule-based or statistical baselines, making it difficult to assess the marginal utility of the generative component. Furthermore, the absence of human-centred explanation interfaces in these systems poses a barrier to adoption among non-specialist users.

These limitations underline the need for hybrid architectures that balance the predictive power of generative models with the interpretability of traditional techniques. The *GenAI Advisor* addresses this by separating signal generation from explanation, thereby ensuring that the core decision logic remains auditable while leveraging GenAI to enhance accessibility and user trust.

2.2 Explainable AI in Financial Systems

As AI-driven systems increasingly influence financial decision-making, the demand for explainable artificial intelligence (XAI) has become a critical concern. In domains characterised by uncertainty, regulatory oversight, and end-user scepticism, opaque "black-box" models are often ill-suited for practical deployment (6). This is particularly true for retail investors who may lack technical backgrounds yet require justifiable, transparent investment reasoning.

Marey et al. (10) propose an empirical framework that integrates deep learning with local explanation techniques, such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-Agnostic Explanations). Their study demonstrates that user trust and engagement are positively correlated with the presence of intelligible model rationales, especially

when users are allowed to compare multiple explanation formats. However, these explanation mechanisms are typically appended to existing models rather than embedded in the design, leading to potential inconsistencies between what the model decides and what is communicated.

Other work, such as Ribeiro et al. (12), highlights the trade-off between global accuracy and local interpretability. While LIME-style approximations offer useful insight into model behaviour, they can misrepresent model logic under certain conditions, particularly in non-linear financial regimes.

To address these issues, the *GenAI Advisor* integrates XAI at both the structural and presentation layers. Rule-based logic provides a baseline of transparent, auditable recommendations, while machine learning classifiers offer enhanced signal detection. A local language model translates system outputs into natural-language explanations, preserving alignment between system behaviour and user-facing justifications. This layered approach ensures both decision accuracy and explanatory coherence.

2.3 Rule-based vs Machine Learning Investment Strategies

Investment recommendation systems have traditionally relied on rule-based logic grounded in technical indicators such as moving averages, momentum oscillators, and price-volume trends. These heuristics offer a high degree of interpretability, making them attractive to both practitioners and novice investors (11). Strategies based on conditions like moving average crossovers or Relative Strength Index (RSI) thresholds are easily understood and lend themselves to direct explanation.

However, rule-based systems often fail to capture the non-linearities and temporal dependencies inherent in financial time series. This has led to a growing interest in machine learning (ML) models for stock selection and signal generation, such as Random Forests, Gradient Boosting Machines (e.g. XGBoost), and neural networks (5). These models can discover subtle patterns and adapt to changing market regimes, offering superior predictive performance in some contexts.

Yet, the adoption of ML in retail investment contexts is hindered by concerns over transparency and robustness. Ryll and Seidens (13) highlight that black-box models frequently suffer from overfitting, poor generalisation, and limited interpretability when applied to financial data. Chakraborty and Joseph (3) similarly observe that financial institutions remain cautious about deploying ML in high-stakes environments due to its opacity and the risk of unpredictable behaviour under changing market conditions. These issues are especially problematic for non-expert users, who rely on trustworthy and intelligible decision support systems.

The *GenAI Advisor* bridges this gap by employing a hybrid strategy engine. Transparent rules serve as a reliable baseline, while supervised learning models (Random Forest and XGBoost) are used to enhance signal strength by capturing complex interactions. This dual approach enables comparative evaluation and supports user trust, as each recommendation is either traceable to a logical rule or supported by model-driven evidence.

2.4 Designing for Non-Technical Users in FinTech

The usability and accessibility of financial technologies remain a significant challenge, particularly for non-technical users who may lack domain expertise in data science or quantitative trading. FinTech interfaces that prioritise raw performance often neglect explainability and user-centred design, which are essential for effective decision support and adoption among retail investors. Ben David et al. (2) demonstrate through experimental evidence that the inclusion of explainable AI elements significantly improves trust and adoption rates among users of financial algorithmic advisors. Their findings underscore the critical role of intelligibility and transparency in interface design for enhancing user engagement.

Studies have shown that users without a financial background exhibit higher levels of anxiety and lower confidence when interacting with complex or opaque recommendation systems. Without clear explanations, users may disregard otherwise effective recommendations, especially in high-stakes settings involving personal investments. This issue is compounded by the fact that many retail-focused platforms do not disclose the reasoning behind their advice, instead presenting deterministic outcomes with minimal context (17).

Explainable user interfaces (XUIs) have been proposed as a solution, incorporating elements such as visual signal annotation, confidence metrics, and natural-language summaries (2). These approaches have demonstrated success in increasing perceived trust, interpretability, and willingness to act upon AI-generated recommendations. However, implementation of such systems is often inconsistent, and few are tailored to specific thematic portfolios like those in GenAI-focused sectors.

The *GenAI Advisor* responds to this gap by embedding a natural language explanation layer that interprets rule-based and ML-driven signals into plain-English rationales. By adopting a language-first, model-agnostic approach, the system enables users to engage meaningfully with its outputs, fostering transparency and informed decision-making without requiring technical fluency.

2.5 Backtesting and Evaluation in FinTech

Robust evaluation is a critical component of financial algorithm design, ensuring that investment strategies are not only theoretically sound but also empirically validated under realistic market conditions. Backtesting a strategy using historical data—remains the standard approach for evaluating financial models. However, its reliability depends on careful control for biases such as lookahead error, data snooping, and overfitting (1).

Evaluation metrics in the FinTech domain extend beyond accuracy to include financial risk/return measures. The Sharpe Ratio, for instance, assesses risk-adjusted return by penalising volatility, while maximum drawdown quantifies peak-to-trough losses, both essential for comparing the robustness of competing strategies (8). Other common metrics include total return, alpha and beta coefficients, and classification-based indicators such as precision, recall, and F1-score when using ML classifiers for signal prediction.

Tools such as `backtrader` and `pandas`-based frameworks facilitate integration of financial indicators, portfolio simulation, and metric reporting, making them essential in contemporary algorithmic finance development. Nevertheless, evaluation practices remain inconsistent, with many academic and commercial models reporting high in-sample performance without adequate out-of-sample or cross-validation testing (7).

The *GenAI Advisor* explicitly incorporates backtesting into its pipeline using reproducible Python-based workflows. It compares rule-based and machine learning strategies across multiple indicators and time frames, reporting both financial and statistical performance metrics. This dual-layered evaluation, provides transparency and supports empirical justification for investment recommendations.

2.6 Synthesis and Research Gap

The review of existing literature reveals a rich landscape of innovation at the intersection of AI, finance, and user-centred design. However, it also exposes clear limitations in current approaches when evaluated through the lens of accessibility, explainability, and domain specificity, three attributes that are critical for supporting non-technical investors seeking exposure to GenAI-focused equities.

Generative models such as *StockGPT* demonstrate that large language models (LLMs) can be fine-tuned to extract financial signals from unstructured data, yet their outputs often lack interpretability and alignment with human-understandable trading logic (9). While this presents

a promising avenue for performance gains, it simultaneously creates a transparency deficit. Similarly, Takahashi and Mizuno (15) propose using diffusion models to generate synthetic financial time series that replicate complex market behaviours. Although valuable for addressing data scarcity and irregularities, these techniques rarely incorporate explanatory mechanisms or support integration with domain-informed baselines, limiting their applicability in systems designed for human-in-the-loop decision making.

Research on XAI has provided a broad array of model-agnostic tools such as SHAP and LIME (12), which can help reveal internal model dynamics. Yet, as highlighted by Marey et al. (10), these methods are frequently bolted onto systems post hoc and suffer from a mismatch between model behaviour and user comprehension. Additionally, the form and delivery of explanations have not been standardised across FinTech applications, leaving a usability gap that affects trust and adoption.

This concern is amplified when considering ML-only investment systems. Despite their performance advantages, such models are often developed without sufficient consideration for interpretability or operational auditability. Ryll and Seidens (13) highlight that machine learning models in financial forecasting frequently suffer from overfitting and poor generalisation, particularly under changing market regimes. Similarly, Chakraborty and Joseph (3) caution that opaque model architectures pose a challenge for real-world deployment in financial institutions, where accountability and stability are essential. Conversely, rule-based systems offer clarity and transparency but often lack the adaptability required to navigate complex, non-linear market dynamics.

From a usability perspective, research indicates that FinTech platforms frequently underestimate the cognitive demands placed on users, often failing to present decision rationales in a digestible and actionable format. Ben David et al. (2) demonstrate that the inclusion of explainable elements—such as natural-language justifications and transparency cues, significantly improves user trust and the likelihood of adopting AI-generated financial advice. While some interfaces include features like confidence bands or risk scores, few extend to personalised, intelligible explanations grounded in model logic and user context. Moreover, no existing platform explicitly addresses the needs of investors seeking targeted exposure to the GenAI sector, a growing interest area underserved by conventional robo-advisors.

In terms of evaluation, there is a consistent pattern of insufficient out-of-sample validation and limited comparative benchmarking across strategy types (1; 7). Financial metrics such as Sharpe Ratio or drawdown are inconsistently applied, and many studies prioritise classification accuracy without accounting for investor risk tolerance or decision thresholds.

Taken together, these findings point to a clear research and development gap: there is currently no publicly available tool that offers GenAI-specific stock recommendations through a hybrid system combining auditable rules, machine learning models, and explainable outputs tailored to non-expert users. Moreover, there is a lack of reproducible, modular pipelines that support comparative evaluation of rule-based and ML-based strategies in the context of GenAI sector investment.

The *GenAI Advisor* aims to fill this gap by providing an integrated, open-source prototype that embodies the insights drawn from this literature. It does so through: (1) a hybrid strategy engine that combines clarity and complexity; (2) a local LLM explanation layer to generate user-friendly rationales; (3) focused stock screening based on GenAI sector relevance; and (4) a rigorous evaluation framework using both financial and statistical metrics. By unifying these components, the system advances the current state of practice in explainable FinTech design for thematic investing.

3 System Design

This chapter presents a detailed account of the design of the GenAI Advisor system, designed to deliver explainable, user-friendly financial insights for retail investors with an interest in Generative AI equities.

3.1 System Overview & Design Rationale

The architecture of the GenAI Advisor system is composed of five distinct yet interdependent layers:

1. **Data Pipeline**, responsible for sourcing, cleaning, and transforming financial data into formats suitable for analysis.
2. **Strategy Engine**, the analytical core that integrates retrieved knowledge to generate structured investment insights.
3. **Explanation Generator**, producing justifications and rationales for each recommendation.
4. **User Interface**, delivers an intuitive interaction point for retail users, visualising results in a digestible and accessible format.
5. **Backtesting & Evaluation**, supports validation of generated recommendations through historical simulations, user feedback loops and performance scoring.

Each layer encapsulates a single responsibility while remaining interoperable with adjacent layers, reflecting a separation-of-concerns design philosophy. This modular approach enables independent development and testing of each component, reducing coupling and simplifying future iterations or extensions.

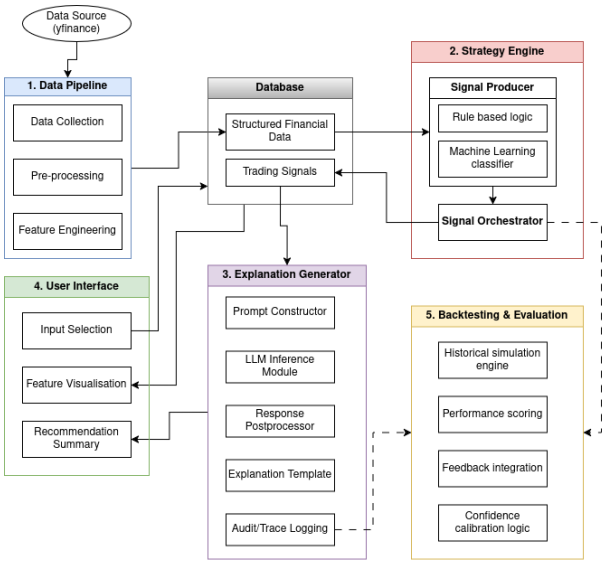

Figure 2: Five-layer Architecture of GenAI Advisor

The design rationale also reflects broader system requirements, particularly the need for explainability, adaptability, and integration with real-time financial data streams. Embedding traceability into the architecture ensures that every recommendation is grounded in verifiable data sources, supporting responsible AI usage. Furthermore, by situating the explanation generator as a distinct layer, the architecture highlights the importance of user trust and transparency as core design principles.

Scalability is not a goal of the current system, and no scaling mechanisms will be implemented in either the prototype or the final version. However, scalability remains an important consideration in the broader context of deploying AI systems in production environments. Should the system be developed further, future work could explore stateless service design, containerised deployment, and asynchronous processing to support larger data volumes and concurrent users. Flexibility across infrastructure backends and compatibility with different classes of language models could also be considered to enable wider applicability.

3.2 Data Pipeline

This foundational layer is responsible for acquiring, preparing, and transforming financial data into structured formats that support both retrieval and modelling. It ensures that the inputs to the system are consistent and semantically rich.

Data Sources: Financial data is sourced via the `yfinance` Python library, which extracts structured data from Yahoo Finance. This includes:

- Daily price data
- Adjusted close prices accounting for dividends and splits
- Market fundamentals including P/E ratio, EPS, market capitalisation, and beta
- Dividend history, earnings results, and analyst forecasts
- Corporate actions such as stock splits and ex-dividend dates

Pre-processing: All raw data is immediately pre-processed in memory using `pandas` and associated libraries before being persisted. This includes:

- Handling of missing values and filtering of anomalous entries
- Normalisation and transformation of numerical columns
- Resampling and forward-filling to align time indices across features
- Annotating rows with relevant metadata

This pre-processing ensures consistency across all instruments and time periods and reduces computational load in downstream stages.

Data Storage: Once cleaned and transformed, the processed dataset is stored in a relational MariaDB database. This allows for fast, structured access by subsequent system components and avoids repeated API calls to Yahoo Finance. Tables are designed to support efficient filtering by date, ticker, and feature type. Indexes are maintained on primary keys and time columns to optimise query latency.

Feature Engineering: This stage derives higher-order signals and representations from the cleaned data:

- **Technical indicators:** including moving averages, MACD, RSI, and volatility bands
- **Fundamental deltas:** e.g. growth rates in earnings, changes in valuation multiples
- **Risk signals:** calculated from rolling variance, beta, and return dispersion
- **Embeddings:** tabular records are converted into dense vector representations using Sentence-BERT for semantic similarity and retrieval

Design Justification: The choice of MariaDB for structured data storage reflects a balance between transparency, performance, and system complexity. Relational databases offer well-defined schema enforcement, support for indexed queries, and seamless integration with Python data analysis libraries. This aligns with the project's emphasis on explainability, traceability, and reproducibility. Alternative solutions were considered, including DuckDB for embedded analytics or Parquet-backed storage for columnar efficiency. However, these options introduce either added complexity or are optimised for larger-scale, concurrent systems. For the scope of this project, focusing on iterative prototyping, backtesting, and clarity, MariaDB provides a robust and pedagogically sound foundation.

3.3 Strategy Engine

The strategy engine forms the analytical core of the GenAI Advisor system, responsible for evaluating investment opportunities using structured signals and domain-specific rules. It interprets features derived from historical data and generates intermediate outputs such as buy/sell/hold flags, risk scores, and confidence levels. This layer operates independently of any generative model.

Signal Retrieval: At runtime, the strategy engine queries the MariaDB backend for time-aligned features such as technical indicators, valuation metrics, volatility profiles, and momentum scores. Data is filtered by ticker, date, and sector using predefined criteria. Missing values are handled via forward-filling or exclusion depending on the indicator's sensitivity.

Rule-based Scoring: Initial decision logic is implemented using deterministic rules. For example:

- *Momentum signal:* Generate a buy signal if the closing price exceeds the 50-day simple moving average and RSI is in the range 50–70.
- *Valuation screen:* Flag as overvalued if the P/E ratio is greater than the sector median by more than one standard deviation.
- *Risk classification:* Assign a "high risk" label if beta exceeds 1.5 or if recent earnings variance is above a defined threshold.

These rule sets are configurable and grounded in established financial heuristics.

Machine Learning Models: Where applicable, supervised models such as logistic regression or gradient-boosted decision trees may be used to estimate directional signals. Input features include moving averages, volatility, return profiles, and sector-adjusted valuation ratios. Outputs are binary or probabilistic classifications of investment suitability.

Strategy Output: The final output is a structured intermediate recommendation, for example:

- *Ticker:* MSFT
- *Action:* HOLD
- *Confidence:* 72%
- *Tags:* "Moderate Risk", "Overvalued"

This output is passed to the explanation layer, where it is contextualised and communicated to the end-user through a natural language interface.

The strict separation of decision logic from explanation ensures that recommendations are reproducible and auditable across time.

3.4 Explanation Generator

The explanation generator is responsible for converting structured strategy outputs into human-readable narratives. This layer interfaces with a large language model (LLM) to articulate the rationale behind each recommendation, ensuring the system remains explainable, and transparent.

Prompt Construction: Each recommendation generated by the strategy engine is converted into a structured prompt. This prompt contains:

- The recommended action (e.g. BUY, HOLD, SELL)
- Associated features (e.g. P/E ratio, RSI, volatility)
- Descriptive tags (e.g. "Overvalued", "Moderate Risk")
- Ticker metadata (e.g. sector, market cap, recent price movement)

The prompt template enforces a consistent narrative structure and ensures that only relevant, pre-computed data is surfaced to the LLM. No raw retrieval or document inference is performed at this stage.

LLM Invocation: The prompt is submitted to a locally hosted language model, with inference restricted to models that can be executed on a GPU with 8GB of VRAM. This design constraint promotes offline operability.

Post-processing and Attribution: The LLM output is parsed and linked to its underlying features. Numerical drivers are tagged and cross-referenced with the original strategy inputs to enable traceability. Where applicable, confidence indicators and relevant feature values are surfaced alongside the textual response.

Model Selection Rationale: For the purpose of this project, fine-tuning large language models is deliberately excluded due to its high resource demands and limited added value in the context of structured explanation generation. Instead, the system leverages pre-trained models served locally via the Ollama framework, which supports quantised variants of open-source architectures such as Mistral, Phi-2, and LLaMA2. These models strike a practical balance between performance and hardware feasibility, running comfortably on an 8GB VRAM GPU. The focus is placed on prompt design and response evaluation rather than model adaptation. This strategy aligns with the project's emphasis on transparency and reproducibility, while avoiding the complexity and risks associated with custom model training.

By isolating the explanation function from decision-making logic and constraining it to local, auditable models, this design ensures the advisory system remains intelligible, trustworthy, and compliant with practical deployment constraints.

3.5 User Interface

The user interface (UI) serves as the primary point of interaction between the end-user and the advisory system. It is designed to surface system outputs in a clear and context-aware format while abstracting away the technical complexity of the underlying architecture.

Interface Design Principles: The UI is guided by the principles of usability, transparency, and minimalism. It aims to provide information density without cognitive overload, and supports both exploratory queries and structured evaluations.

Interaction Model: The primary mode of interaction is a ticker selector and a time window. Queries are routed through the backend pipeline and responses are returned in astructured, explanation-rich format.

Presentation Layer: Output is displayed in modular response cards, with each card corresponding to a company. Each card includes:

- The recommended action (BUY, HOLD, SELL)

- A summarised rationale generated by the explanation layer
- Key features (e.g. P/E ratio, RSI, beta) highlighted inline
- Option to expand for full explanation and numerical context

Cards are sorted based on confidence or relevance, and users can collapse or pin cards for comparison.

Technical Implementation: The frontend is built using a lightweight web framework such as Streamlit. This ensures rapid prototyping, cross-platform compatibility, and support for secure API calls to backend services. The UI is deployed as a local web application accessible via browser.

Responsiveness and Accessibility: The interface is optimised for both desktop and mobile devices. Layouts use responsive containers and typography is chosen for readability. Accessibility guidelines such as contrast ratios and semantic labelling are followed to ensure inclusivity.

This interface completes the user-facing layer of the system, enabling seamless communication of insights derived from structured analysis and local language model generation.

3.6 Backtesting & Evaluation

This layer is responsible for assessing the performance, consistency, and interpretability of the GenAI Advisor system. Evaluation is conducted along two axes: (i) financial effectiveness of the recommendations and (ii) fidelity and stability of the generated explanations.

Backtesting of Strategy Output: The system supports retrospective evaluation of strategy outputs using historical financial data stored in MariaDB. Given a time window and a set of tickers, the engine replays strategy decisions based on data available at the time. The simulated trades are benchmarked using standard performance metrics such as:

- Cumulative return
- Sharpe ratio
- Maximum drawdown
- Hit rate (proportion of correct directional calls)

These simulations validate whether the rule-based and ML-driven components produce economically viable signals under realistic conditions.

Evaluation of Explanation Consistency: Because the system includes a generative component, it is essential to assess the consistency and fidelity of explanations. For a given strategy output, repeated LLM invocations are evaluated for semantic agreement and factual stability. Techniques include:

- *Stability under identical prompts:* The same input is submitted multiple times to check for drift.
- *Paraphrase testing:* Slightly altered prompts are compared for explanation robustness.
- *Key fact alignment:* Extracted metrics in the explanation are compared against known strategy outputs.

These tests ensure that the narrative remains faithful to the underlying numerical reasoning.

User Feedback Loop: The UI logs user reactions to recommendations through binary feedback (e.g. thumbs up/down). Feedback is stored with associated prompt and strategy metadata, enabling post-hoc review and potential retraining of classification thresholds or strategy weights.

Qualitative Auditing: In addition to automated tests, a sample of responses is manually audited to assess:

- Clarity of the natural language explanation
- Justification coverage (i.e., do the reasons align with the action?)
- Absence of hallucinated content or misleading terminology

This step supports ethical AI principles by ensuring outputs remain comprehensible and appropriate for retail consumption.

This evaluation layer closes the loop between system logic, narrative explanation, and end-user experience. It provides the foundations for continuous improvement while maintaining interpretability and trust.

4 Feature Prototype and Evaluation

This section presents a focused implementation of the explanation generator and a comparative evaluation of several local large language models (LLMs). The objective is to determine which model best aligns with the system's requirements for explainability and accessibility. The evaluation is scoped to models runnable within an 8 GB VRAM constraint to support reproducibility and local deployment.

4.1 Prototype Objective and Scope

The prototype isolates the final layer of the GenAI Advisor architecture, the explanation generator. It takes as input a structured recommendation, typically generated by the strategy engine, and produces a textual rationale. This explanation layer is central to the system's commitment to transparency and user trust. For the purpose of this prototype, all upstream processes (data collection, feature engineering, and decision logic) are simulated using static inputs. The design focuses on testing whether modern, instruction-tuned LLMs can interpret structured financial metrics and generate plausible justifications for retail-facing outputs.

4.2 Model Selection Rationale

The four models selected for evaluation were chosen based on their availability through the Ollama runtime, compatibility with 8GB VRAM constraints, and representativeness of recent LLM design trends. Each model was expected to demonstrate distinctive strengths aligned with the explanation generator's requirements: factual precision, clarity, and educative capacity.

- `mistral:7b` was selected due to its strong performance on standard instruction-following benchmarks. As an open-weight model built with modern decoder architecture and dense pretraining, it has demonstrated balanced performance across summarisation, QA, and reasoning tasks. Its outputs are typically well-structured, concise, and fluid.
- `deepseek-r1:8b` is a reasoning-optimised model developed to excel at complex multi-step tasks. It remains operational on standard consumer-grade GPUs when quantised. It was chosen for its potential to deliver highly grounded and stepwise rationale construction, a valuable trait for explanatory outputs in finance.
- `llama3.1:8b` was included as Meta's latest general-purpose model. It represents the frontier in pretraining scale and refinement, with improvements in response quality and instruction tuning. It was expected to deliver polished and high-coverage answers, potentially balancing fluency and informativeness.
- `gemma:7b` from Google was selected for its lightweight design and efficient inference profile. While it lacks the same instruction-following depth as the others, it serves as a practical baseline for evaluating trade-offs between model size, efficiency, and output utility.

These models span a useful spectrum from high-performance instruction tuning (`mistral`) and reasoning depth (`deepseek`), to baseline fluency (`llama3.1`) and efficient deployment (`gemma`). Their inclusion enables a comparative understanding of how different model classes handle structured explanation generation within the constraints and goals of this project.

4.3 Implementation Details

The implementation is a standalone Python script that integrates with the `Ollama` runtime to serve quantised LLMs locally. This approach avoids reliance on cloud APIs or external dependencies and ensures compatibility with standard workstation-grade hardware.

Each model received the same structured input and was prompted as follows:

```
You are a financial assistant. Based on the following metrics:

Ticker: {ticker}
Action: {action}
P/E Ratio: {pe}
RSI: {rsi}
Beta: {beta}

Generate a short, clear explanation of why this action was recommended,
using only the signals provided.

Your explanation will be evaluated based on:
1. Factual alignment { Does it correctly reference and reflect the input signals?
2. Clarity { Is it readable and jargon-free for a non-expert?
3. Educational value { Does it help the user understand financial reasoning?

Write in a tone that is simple, supportive, and instructive.
```

Using this prompt, the four language models were evaluated for their ability to generate explanatory justifications from structured financial signals. Each model was tasked with producing a natural language explanation based on three key indicators: P/E ratio, RSI, and beta, along with a pre-assigned action recommendation.

Mistral 7B typically produced structured, metric-by-metric justifications. Its responses tended to follow a clear, segmented format, interpreting each signal independently before summarising the overall recommendation. The model showed a tendency to anchor its language in established financial reasoning, often framing explanations within a cautiously optimistic narrative.

DeepSeek-R1 8B adopted a more process-oriented style. It often simulated internal reasoning steps, providing commentary on its own interpretive process before delivering a final summary. This approach tended to produce verbose but richly annotated responses, balancing technical correctness with pedagogical tone.

LLaMA 3.1 8B favoured a polished and highly readable output structure. Its responses were often framed as formal recommendations, with summarised takeaways. The tone was typically confident and informative, with a focus on simplifying terminology while preserving the logic behind financial metrics.

Gemma 7B generally produced shorter, more direct explanations. Its responses prioritised brevity and tended to cover each input signal in a minimal fashion. While the output was usually factually consistent with the input, it often lacked contextual depth or elaboration.

Across all models, the task of converting quantitative financial indicators into qualitative, user-facing explanations highlighted different language generation strategies. Some models leaned toward verbosity and teaching, while others prioritised conciseness and structure. These differences reflect varying underlying instruction tuning and decoding preferences, underscoring the importance of model selection depending on the target user experience.

4.4 Evaluation Methodology

A peer-evaluation strategy was employed where each model was tasked with anonymously evaluating the outputs of the others. For each of the three input scenarios, four anonymised responses were scored against the following dimensions:

1. **Factual alignment**: Did the response correctly use the input metrics?
2. **Clarity**: Was the response understandable and free of jargon?

3. **Educational value**: Did the response help the user learn financial reasoning?

Each evaluation was repeated three times per model to control for decoding variance. All scores were normalised to a 1–5 scale. Fluency (clarity) was given a slightly higher weight (0.4) than factual and educational value (0.3 each) during ranking.

4.5 Results and Comparison

The final weighted scores are summarised in Table 1.

Model	Factual	Clarity	Learning
`deepseek-r1:8b`	4.66	4.20	4.66
`mistral:7b`	4.42	4.06	4.44
`llama3.1:8b`	4.33	4.00	4.22
`gemma:7b`	4.14	3.80	4.00

Table 1: Peer-evaluated model scores (1–5 scale).

4.6 Discussion of Results

Before conducting the evaluation, it was anticipated that `mistral:7b` would lead due to its strong benchmark performance and widespread use in instruction-following tasks. Similarly, `deepseek-r1:8b` was included for its reputed strength in reasoning tasks, while `llama3.1:8b` was expected to provide fluent yet generalist output. `gemma:7b`, although efficient, was included as a performance baseline.

Interestingly, `deepseek-r1:8b` achieved the highest aggregate score, particularly due to its consistently accurate references to input signals and educational structure. It often contextualised the rationale using analogies and progressive explanation, aligning with the system's goal of financial literacy enhancement. `mistral:7b` remained competitive, delivering slightly more fluent prose but occasionally at the cost of explicit metric referencing.

`llama3.1:8b` and `gemma:7b` underperformed in clarity and interpretability, with some responses lacking actionable justification or oversimplifying the rationale. These findings reinforce the need for balance between readability and groundedness in explanation generation, a trade-off that `deepseek-r1:8b` currently handles most effectively.

4.7 Conclusion

Based on peer evaluations across multiple structured prompts and consistent scoring, `deepseek-r1:8b` achieved the highest aggregate marks, particularly for its detailed and pedagogically aligned explanations. However, its verbosity introduces a need for output post-processing, which adds complexity to the current system. Given the project's focus on simplicity and low-overhead deployment, `mistral:7b` is selected as the explanation engine for the GenAI Advisor prototype. It strikes a more practical balance between fluency, interpretability, and implementation feasibility, making it better suited for the intended use case and resource constraints.

5 Minimum Viable Product Approach

In the development of the GenAI Advisor, a structured Minimum Viable Product (MVP) approach was followed to ensure systematic progress while respecting the constraints of time, computational resources, and project evaluation needs. The MVP methodology allowed for the early identification of potential technical limitations while enabling the iterative refinement of core functionalities essential to the project.

The MVP was designed to operate fully offline, respecting user data privacy and ensuring explainability. It integrates four layers: data ingestion, strategy engine, explanation generation, and a user-facing Streamlit interface. The initial focus was on constructing a thin, end-to-end pipeline capable of performing equity data ingestion using Yahoo Finance, caching the data locally in CSV format to avoid repeated API calls while supporting reproducibility.

The strategy engine implemented a simple rule-based mechanism using a moving average crossover technique, providing structured recommendations (BUY or HOLD) alongside clear reasons based on the underlying signal. This approach balanced simplicity with interpretability, aligning with the goal of transparent, explainable AI systems suitable for retail investors.

Explanation generation was initially prototyped with placeholder explanations before being integrated with a locally hosted Mistral 8B model via Ollama, enabling offline large language model inference. This allowed for the generation of user-friendly, clear explanations for investment signals without requiring online API dependencies.

A Streamlit frontend was developed to facilitate user interaction, enabling the entry of tickers, the execution of the pipeline, and the clear display of structured recommendations, explanations, and recent price charts. This immediate feedback loop supported the testing and evaluation processes, aligning with agile development practices.

Pytest was configured to establish a lightweight test-driven development environment, ensuring that data ingestion, strategy engine outputs, and explanation generation remained reliable as further functionalities were layered onto the system.

The MVP was then extended with a backtesting utility, enabling the analysis of recommendations against historical price movements over a defined lookahead period. A batch backtesting pipeline was constructed to systematically test multiple tickers across various dates, generating CSV outputs to support quantitative evaluation. An evaluation notebook was prepared to visualise the distribution of price movements following recommendations, allowing for the analysis of the effectiveness of the generated signals.

This MVP approach ensured that a functioning, test-backed advisor system was available early in the project timeline, providing a robust foundation for evaluation and report writing, and allowing for targeted refinement and iteration in subsequent development sprints.

6 Implementation Challenges in the MVP

During the MVP development, several non-trivial challenges arose that required careful technical decisions to ensure system robustness and correctness. Firstly, handling timezone consistency was critical when implementing the backtesting pipeline. The `yfinance` library returns time series data indexed with a timezone, whereas initial cutoff dates for historical evaluations were timezone-naive, resulting in type errors during comparisons. This was resolved by explicitly localising the cutoff dates using `pd.to_datetime().tz_localize('America/New_York')`, ensuring consistent and error-free slicing of historical data.

Another challenge involved achieving reproducible and API-resilient data ingestion through CSV caching. Early iterations faced hidden header misalignments and unnamed indexes when saving and reloading data, which caused failures in downstream modules. This was addressed by enforcing consistent use of the `index_label` parameter during saving and explicitly specifying `index_col` during loading, ensuring data could be reliably reused without corruption.

Integrating a local large language model explanation generator with `Ollama` also required precise engineering. Instead of using an external API, a local Mistral 8B model was leveraged to maintain offline functionality and data privacy. This necessitated the design of a robust subprocess management approach, correctly piping structured prompts and capturing outputs while handling encoding and potential model unavailability gracefully.

Additionally, implementing consistent test discovery using `Pytest` required an understanding of Python's module resolution, as the project initially failed to locate modules due to the package structure. This was resolved by enforcing the use of `PYTHONPATH=.` during test execution, ensuring reliable test discovery across environments.

Finally, careful attention was required in the design of the backtesting pipeline to prevent future data leakage while computing lookahead returns. Ensuring that data used to generate recommendations strictly respected the cutoff date, and accurately measuring post-recommendation price movements, was critical for the validity of the evaluation metrics used in the later stages of the project.

Collectively, addressing these challenges not only ensured the stability and correctness of the GenAI Advisor MVP but also demonstrated disciplined software engineering practices that align with industry and academic expectations for reliable, reproducible systems.

7 Project workplan

The project is progressing well as per workplan below.

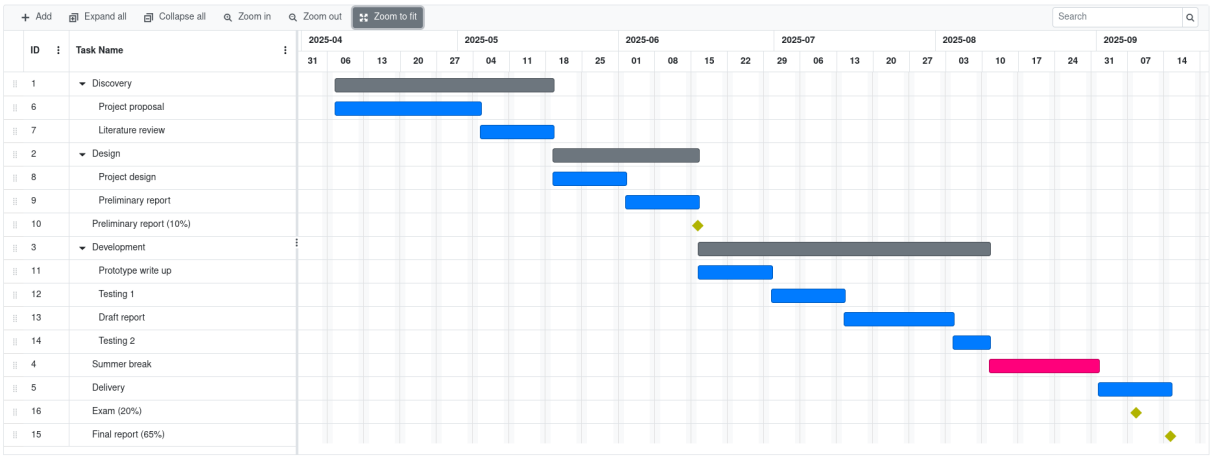

Figure 3: Workplan for Final Project.

References

[1] David H. Bailey, Jonathan M. Borwein, Marcos Lopez de Prado, and Qiji Jim Zhu. 2014. The Probability of Backtest Overfitting. *Journal of Computational Finance* 20, 4 (2014), 39–69.

[2] Daphna Ben David, Yedidya S. Resheff, and Tal Tron. 2021. Explainable AI and Adoption of Financial Algorithmic Advisors: an Experimental Study. *arXiv preprint arXiv:2101.02555* (2021). https://arxiv.org/abs/2101.02555

[3] Chiranjit Chakraborty and Alex Joseph. 2017. *Machine Learning at Central Banks.* Staff Working Paper 674. Bank of England. https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2017/machine-learning-at-central-banks.pdf

[4] Fa Chen. 2021. Variable interest entity structures in China: are legal uncertainties and risks to foreign investors part of China's regulatory policy? *Asia Pacific Law Review* 29, 1 (2021), 1–24. https://doi.org/10.1080/10192557.2021.1995229

[5] Thomas Fischer and Christopher Krauss. 2018. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research* 270, 2 (2018), 654–669. https://doi.org/10.1016/j.ejor.2017.11.054

[6] David Gunning and David Aha. 2019. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine* 40, 2 (2019), 44–58. https://doi.org/10.1609/aimag.v40i2.2850

[7] I. Kaastra and M. Boyd. 1996. Designing a Neural Network for Forecasting Financial and Economic Time Series. *Neurocomputing* 10, 3 (1996), 215–236. https://doi.org/10.1016/0925-2312(95)00039-9

[8] Andrew W. Lo. 2002. The Statistics of Sharpe Ratios. *Financial Analysts Journal* 58, 4 (2002), 36–52. https://doi.org/10.2469/faj.v58.n4.2453

[9] D. Mai. 2024. StockGPT: A GenAI Model for Stock Prediction and Trading. *SSRN* (2024). https://ssrn.com/abstract=4787199

[10] N. Marey, A. A. Abu-Musa, and M. Ganna. 2024. Integrating Deep Learning and Explainable Artificial Intelligence Techniques for Stock Price Predictions. *International Journal of Accounting and Management Sciences* 3, 4 (2024), 479–504.

[11] John J. Murphy. 1999. *Technical Analysis of the Financial Markets: A Comprehensive Guide to Trading Methods and Applications.* New York Institute of Finance.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. Why Should I Trust You? Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* ACM, 1135–1144.

[13] Lukas Ryll and Sebastian Seidens. 2019. Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey. *arXiv preprint arXiv:1906.07786* (2019). https://arxiv.org/abs/1906.07786

[14] S&P Global Market Intelligence. 2025. *GenAI funding hits record in 2024, boosted by infrastructure interest.* https://www.spglobal.com/market-intelligence/en/news-insights/articles/2025/1/genai-funding-hits-record-in-2024-boosted-by-infrastructure-interest-87132257 Accessed May 2025.

[15] Tomonori Takahashi and Takayuki Mizuno. 2024. Generation of synthetic financial time series by diffusion models. *arXiv preprint arXiv:2410.18897* (2024).

[16] U.S. Department of Commerce. 2022. *Commerce Implements New Export Controls on Advanced Computing and Semiconductor Manufacturing Items to the People's Republic of China (PRC).* Technical Report 2022-21658. Bureau of Industry and Security. `https://www.govinfo.gov/content/pkg/FR-2022-10-13/pdf/2022-21658.pdf` Federal Register, Vol. 87, No. 197, pp. 62186–62215.

[17] Alex Zarifis and Xusen Cheng. 2024. How to build trust in answers given by Generative AI for specific, and vague, financial questions. *arXiv preprint arXiv:2408.14593* (2024). `https://arxiv.org/abs/2408.14593`