WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal Homepage: www.ijrpr.com ISSN 2582-7421

LLM-Based Personalized Financial Product Recommendation for First-Time Investors

Narava Siri Nandini [a*], Ajit Kumar Rout [a]

[a] Department of Information Technology, GMR Institute of Technology (GMRIT Deemed to be University), GMR Nagar, Rajam – 532127, Andhra Pradesh, India

A B S T R A C T

Choosing a first investment product is difficult for beginners, and most recommender systems cannot help them because they learn from past purchases, ratings or clicks that a new investor does not have. This paper presents a cold-start recommendation approach for first-time investors that replaces missing history with information the investor can state directly: an onboarding form and a free-text description of the financial need. A language model, with a rule-based fallback, converts the text into structured preferences. The investor is then placed in a broad segment using K-Means clustering, products that violate mandatory constraints are removed by explicit suitability rules, and the remaining products are ranked by cosine similarity and explained in plain language using catalogue fields. A Streamlit prototype was run for a 25-year-old beginner with moderate risk tolerance, a long-term horizon and a monthly capacity of Rs. 5,000. It assigned the Moderate segment and returned three moderate-risk, long-term, education-oriented funds with written justifications. The run used a sample catalogue and the rule-based extractor, and the three products received equal match scores, so the paper reports a working design and a proposed evaluation plan and does not claim measured accuracy gains.

Keywords: Financial product recommendation; large language models; cold-start problem; first-time investors; K-Means clustering; cosine similarity; explainable recommendation

1. Introduction

Digital platforms allow almost anyone to open an investment account, yet the choice among mutual funds, hybrid schemes and goal-based products is far from simple. A suitable option depends on the investor's goal, tolerance for loss, time horizon, monthly capacity and need for liquidity. Beginners often cannot map their situation to these terms, and an unexplained ranked list gives them little reason to trust the result.

Recommender systems normally address this by learning from behaviour. Collaborative filtering and sequence models need purchases, ratings or click streams, and approaches that update preferences over time need several rounds of feedback (Wei & Liu, 2025; Hu et al., 2024). A first-time investor has none of this at the first visit, which is the user cold-start problem. The investor does, however, know the purpose of the investment and can describe it in ordinary language, for example, saving for a child's education over seven years without taking much risk.

Large language models (LLMs) are well suited to reading such statements, but their output can be fluent and still wrong, so it should not decide on its own which financial product is shown (Zhao et al., 2024b; Oehler & Horn, 2024). This paper therefore limits the language model to one job, extracting stated preferences, and leaves eligibility and ranking to transparent rules and similarity scores. The contributions are:

a cold-start workflow that builds a reviewable investor profile from onboarding fields and free text;

a pipeline that combines K-Means segmentation, mandatory suitability rules and cosine-similarity ranking, with explanations tied to catalogue fields;

a working Streamlit prototype and a demonstration run, reported together with its limitations; and

an evaluation plan that separates what the prototype shows from what still has to be measured.

2. Related work

2.1 LLMs and financial product recommendation

Zhao et al. (2024b) survey how language models are used in recommendation, covering representation learning, fine-tuning and prompting, and Shehmir and Kashef (2025) review integration approaches and open challenges. These surveys treat recommendation generally. In finance, Wu and Li (2025) examined 65 studies and described the field by its several objectives, wide feature space, time sensitivity and parallel buying and selling behaviour. A

comparison by Oehler and Horn (2024) between a general chatbot and robo-advisors shows that language models are now part of the advice landscape, but a chatbot does not enforce suitability rules or ground its answer in a verified catalogue.

2.2 Planning, preference learning and cold start

de Zarzà et al. (2024) combine LLM suggestions with individual and household budgeting optimization so that a generated plan satisfies savings constraints, which motivates separating a language-derived preference from an enforceable rule. Wei and Liu (2025) learn a customer's utility from fund-click sequences with an epsilon-greedy strategy, and Hu et al. (2024) analyse preferences interactively with reinforcement learning; both depend on feedback that is absent at the first session. Cold-start work uses side information when history is missing, for example, residual graph meta-embeddings (Zhao et al., 2024a), few-shot and zero-shot industrial frameworks (Cao et al., 2025) and reinforcement learning for cold users (Giannikis et al., 2024). Closest to the present problem are an explainable robo-advisor that learns online for new investors without trading records (Hu et al., 2025) and a stock recommender that uses investors' behavioural traits (Takayanagi & Izumi, 2023). Beyer Díaz et al. (2025) show the value of longitudinal customer data, and a cross-domain fine-tuned LLM has been proposed for context-aware recommendation (CD-LLMCARS, 2025). Table 1 summarizes how these groups relate to a first-time investor.

Table 1 - Related approaches and their fit for a first-time investor.

Approach group	Needs history?	Gap for a first-time investor
LLM recommender surveys (Zhao et al., 2024b; Shehmir & Kashef, 2025)	No	General scope; no financial suitability rules
Financial recommendation review (Wu & Li, 2025)	No	Review only; no tested workflow
Budget optimization with LLM (de Zarzà et al., 2024)	No	Plans budgets; does not rank products
Utility learning and RL (Wei & Liu, 2025; Hu et al., 2024; Giannikis et al., 2024)	Yes	Needs feedback rounds
Cold-start frameworks (Zhao et al., 2024a; Cao et al., 2025)	Little	Not specific to investors
Robo-advisor and trait-based (Hu et al., 2025; Takayanagi & Izumi, 2023)	No	Online learning or stocks only
Longitudinal data (Beyer Díaz et al., 2025)	Yes	Depends on customer history

Across these studies, no single method combines a profile built from text and form fields, mandatory eligibility screening, attribute-based ranking and a grounded explanation for a user with no history. The next section describes such a combination.

3. Proposed approach

3.1 Overview

Fig. 1 shows the pipeline. Inputs are validated first, then a preference extractor turns the free text into structured fields. The investor is assigned to a segment, ineligible products are removed, the rest are ranked by similarity, and each of the top results receives an explanation that cites the fields used.

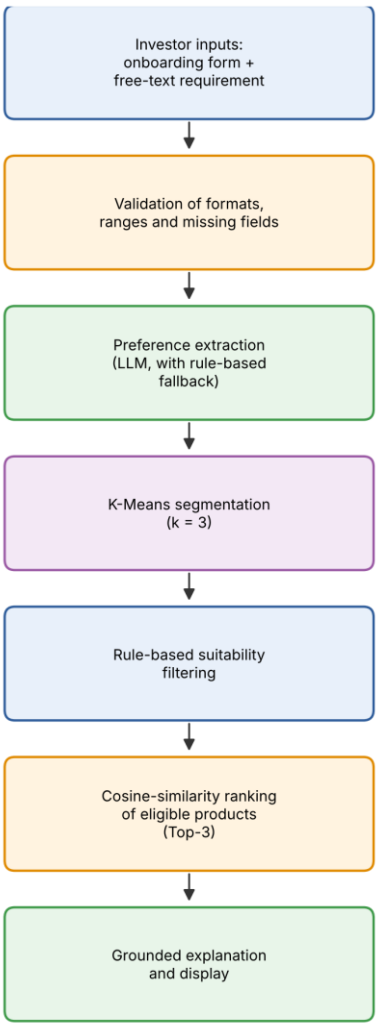


Fig. 1 - Workflow of the proposed recommendation pipeline.

3.2 Investor inputs and profile

The form collects age, annual income, monthly investment amount, goal, risk tolerance, investment horizon and financial literacy, and a text box accepts a free-text requirement. The structured fields are treated as direct evidence. The amount must be a monthly contribution, the horizon must have a clear category or unit, and malformed values are rejected. The product catalogue is a CSV file with the fields product name, type, risk level, expected return, minimum investment, investment horizon, liquidity and goal.

3.3 Preference extraction

The extractor reads the free text and returns four fields: goal, risk tolerance, horizon and liquidity need. The prompt asks the model to leave unknown values empty, and the user's text is handled as data and never as instructions that change application rules. If the language model is unavailable, a rule-based extractor applies keyword rules to the same text, so the pipeline still completes. A conflict between the form and the text should be put to the investor and not resolved silently.

3.4 Investor segmentation

A K-Means model with k = 3 groups investors using age, income, monthly capacity, risk tolerance and horizon after scaling. K-Means minimizes the within-cluster squared distance

$$J = \Sigma k=1..K \ \Sigma x \in Ck \ ||x - \mu k||^2 \quad (1)$$

where Ck is the set of profiles in cluster k and μk is its centroid. A new investor joins the nearest cluster. The segment gives a short description of the investor type, but it never overrides declared constraints.

3.5 Suitability filtering

For investor u and product p, eligibility is the conjunction of the mandatory checks

$$Eligible(u,p) = [Risk(p) \leq Risk(u)] \wedge [Min(p) \leq Amt(u)] \wedge [Hor(p) \approx Hor(u)] \wedge [Liq(p) \geq Need(u)] \quad (2)$$

where Min is the minimum investment and Amt the monthly amount. Only eligible products are scored. A product whose eligibility fields are missing is not assumed to be eligible, and if no product passes, the system reports which rules removed the candidates instead of relaxing a mandatory one.

3.6 Matching, ranking and explanation

The investor preferences and each eligible product are encoded with the same attribute order, units and scaling. The match score is the cosine similarity

$$cos(u,p) = (u \cdot p) / (||u|| \times ||p||) \quad (3)$$

The top three products are returned. Each explanation lists the matching attributes, namely risk level, goal, horizon, the monthly budget against the minimum investment and the return shown in the catalogue, so that a reader can check it against the product record. The score measures similarity under the chosen encoding; it is not a forecast of return or a calibrated probability of suitability.

4. Implementation and results

4.1 Prototype

The pipeline was implemented in Python as a Streamlit web application that runs locally. The product catalogue is read with pandas, K-Means comes from scikit-learn, and the extraction step calls a language model and falls back to the rule-based extractor. The interface has an investor-profile form, a free-text box and a button that runs the pipeline and shows the segment, the extracted parameters, three recommendation cards and a summary table.

4.2. Demonstration case

The run used one profile, listed in Table 2. The free-text requirement read: a beginner earning about Rs. 6 lakh per year who can invest Rs. 5,000 per month at moderate risk for seven years for a child's education.

Table 2 - Investor inputs in the demonstration run.

Field	Value
Age	25
Annual income	Rs. 6,00,000
Monthly amount	Rs. 5,000
Goal	Education
Risk tolerance	Moderate
Horizon	Long-term (about 7 years)
Financial literacy	Beginner

4.3 Output

The application reported successful execution and assigned the investor to the Moderate segment. The extraction panel showed goal = Education, risk = Moderate, horizon = Long-term and liquidity = Medium, and it named the rule-based extractor as the method because the call to the configured language model returned a model-not-found error. Table 3 and Fig. 2 give the three returned products, all of which are moderate-risk, long-term funds with a minimum investment of Rs. 500, well within the monthly capacity.

Table 3 - Recommendation output of the prototype (sample catalogue).

#	Product	Risk / horizon	Return shown	Match score
1	Canara Robeco Equity Hybrid Fund	Moderate / Long-term	11.5%	99.0%
2	UTI Children Career Education Fund	Moderate / Long-term	11.0%	99.0%
3	SBI Magnum Children Benefit Fund	Moderate / Long-term	10.8%	99.0%

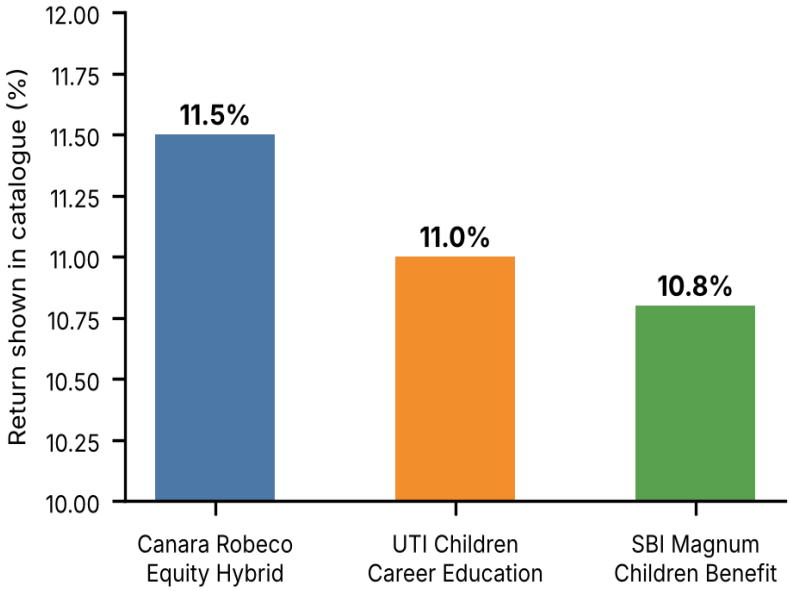


Fig. 2 - Return shown in the sample catalogue for the three recommended products.

4.4 Observations

Each card carried a written reason citing the matched risk level, goal, horizon, budget and return field, so every suggestion could be traced to the catalogue. Two of the three products are goal-based children's education funds, consistent with the stated goal. The same pipeline completed when the language model failed, which shows that the fallback keeps the system usable.

5. Discussion and limitations

The demonstration supports a modest claim: a first-time investor with no history can obtain ranked and explained suggestions from onboarding information alone. It does not show that the suggestions are better than a simpler method, and several limits should be stated.

The result comes from one profile and a sample catalogue whose product names are labelled as samples, so the returns shown are not verified market data and are not guarantees.

All three products received a match score of 99.0%, so the score did not separate them. The displayed order matches descending return, which suggests that ties are broken by return or catalogue order; this rule should be defined explicitly.

The language model was not exercised in the captured run, so the benefit of LLM extraction over the rule-based extractor is untested.

No accuracy, constraint-violation or user-comprehension figures are reported.

Table 4 lists the evaluation that would address these gaps. It needs a verified catalogue, annotated natural-language inputs, independent suitability judgements and baselines without the language model, without K-Means and with a simpler scoring rule.

Table 4 - Proposed evaluation metrics.

Metric	What it measures
Field-level precision and recall	Correct extraction of goal, risk, horizon and liquidity
Constraint-violation rate	Returned products that fail a mandatory rule
Precision@N and NDCG@N	Relevance and order of the top-N list
Empty-result rate	Profiles for which no product is eligible
Explanation consistency	Statements supported by profile and product fields
User comprehension	Understanding of reasons and score limits

Responsible use also matters because age, income and goals are personal data. The system should collect only what it needs, show the extracted profile for confirmation, keep a record of the catalogue version behind each result, avoid presenting similarity as a promise of return, and refer high-value decisions to a qualified adviser.

6. Conclusion and future work

This paper presented a cold-start recommendation approach for first-time investors that uses an LLM only to extract stated preferences and relies on K-Means segmentation, mandatory suitability rules, cosine-similarity ranking and catalogue-grounded explanations for everything else. A Streamlit prototype produced three suitable, explained suggestions for a beginner profile. Because the demonstration used a sample catalogue, one profile and a rule-based extraction fallback, the results show feasibility and not performance. Future work will build a verified catalogue with update dates, define the tie-break rule, annotate an extraction data set, run the baseline comparisons and user study in Table 4, and add consented feedback with utility learning for later sessions (Wei & Liu, 2025), together with budget optimization (de Zarzà et al., 2024).

Acknowledgements

The authors thank the Department of Information Technology, GMR Institute of Technology (GMRIT Deemed to be University), Rajam, for the facilities and support provided for this work.

References

CD-LLMCARS: Cross domain fine-tuned large language model for context-aware recommender systems. (2025). *IEEE Open Journal of the Computer Society*, *6*, 49–59.

Beyer Díaz, S., Coussement, K., & De Caigny, A. (2025). From collaborative filtering to deep learning: Advancing recommender systems with longitudinal data in the financial services industry. *European Journal of Operational Research, 323*(2), 609–625.

Cao, X., Zhang, W., Jiang, F., & Zhang, X. (2025). An industrial framework for cold-start recommendation in few-shot and zero-shot scenarios. *Information, 16*(12), 1105.

de Zarzà, I., de Curtò, J., Roig, G., & Calafate, C. T. (2024). Optimized financial planning: Integrating individual and cooperative budgeting models with LLM recommendations. *AI, 5*(1), 91–114.

Giannikis, S., Frasincar, F., & Boekestijn, D. (2024). Reinforcement learning for addressing the cold-user problem in recommender systems. *Knowledge-Based Systems, 294*, 111752.

Hu, X., Kang, S., Ren, L., & Zhu, S. (2024). Interactive preference analysis: A reinforcement learning framework. *European Journal of Operational Research, 319*(3), 983–998.

Hu, X., Kang, S., Xue, S., & Zhu, S. (2025). Explainable robo-advisor: An online learning framework for new investors without trading records. *Neurocomputing, 624*, 129463.

Oehler, A., & Horn, M. (2024). Does ChatGPT provide better advice than robo-advisors? *Finance Research Letters, 60*, 104898.

Shehmir, S., & Kashef, R. (2025). LLM4Rec: A comprehensive survey on the integration of large language models in recommender systems—Approaches, applications and challenges. *Future Internet, 17*(6), 252.

Takayanagi, T., & Izumi, K. (2023). Harnessing behavioral traits to enhance financial stock recommender systems: Tackling the user cold start problem. In *Proceedings of the 2023 IEEE International Conference on Big Data* (pp. 5694–5703).

Wei, J., & Liu, J. (2025). Personalized fund recommendation with dynamic utility learning. *Financial Innovation, 11*, 1–27.

Wu, D., & Li, X. (2025). A systematic literature review of financial product recommendation systems. *Information, 16*(3), 196.

Zhao, F., Huang, C., Xu, H., Yang, W., & Han, W. (2024a). RGMeta: Enhancing cold-start recommendations with a residual graph meta-embedding model. *Electronics, 13*(17), 3473.

Zhao, Z., Fan, W., Li, J., Liu, Y., Mei, X., Wang, Y., Wen, Z., Wang, F., Zhao, X., Tang, J., & Li, Q. (2024b). Recommender systems in the era of large language models (LLMs). *IEEE Transactions on Knowledge and Data Engineering, 36*(11), 6889–6907.