Dhanmitra: A ML Powered Financial Advisor

Ms. Anitta Antony[1], Abhinav Prateek[2] Doi Vasif[3] Arpita Sexena[4] Anwesha Patra[5]
[1]*Assistant Professor, Dept. of Computer Science and Engineering, Acharya Institute of Technology Bengaluru, India*
[2,3,4,5]*UG Student, Dept. of Computer Science and Engineering, Acharya Institute of Technology Bengaluru, India*

***Abstract*—Managing personal finances is by no means a trivial task for many individuals, either because access to professional financial planners is restricted or because financial literacy is low. The use of robo-advisors has made investment advice more widely accessible, but many of the currently available solutions rely on short questionnaires and coarse risk categories, usually resulting in generic and weakly personalized portfolios. This paper presents *DhanMitra*, a machine-learning–driven robo-advisory platform that elicits rich user information on income, expenditure patterns, existing assets and liabilities, and explicit financial goals and converts it into tailored savings and investment strategies. The system estimates user-specific risk preferences, designs optimized portfolios using techniques such as mean–variance optimization, and generates budget and goal-tracking recommendations that can be delivered through natural language interfaces.**

We describe the overall architecture, including the data processing pipeline, risk profiling models, optimization engine, and user interface layer. Using illustrative case studies representing students, mid-career professionals, and retirees, we demonstrate how DhanMitra can adapt allocations and contribution paths to different stages of the financial life cycle. A simulation-based evaluation compares DhanMitra against simple rule-based baselines and shows that the proposed system can improve projected goal achievement and risk-adjusted returns while preserving low operational cost. The modular design allows integration of advanced language models and additional financial products, enabling DhanMitra to evolve into a comprehensive digital financial companion.

***Index Terms*—Financial advisory, robo-advisor, machine learning, portfolio optimization, personalization, financial literacy, AI-driven planning.**

I. INTRODUCTION

Households around the world increasingly interact with financial markets through digital channels. Mobile applications make it easy to open brokerage accounts, purchase mutual funds, or invest in low-cost index products. However, the decision of *how much* to invest, *where* to invest, and *how* to balance risk and return over time remains challenging for non-expert users. Many individuals rely on ad-hoc heuristics, informal advice, or product-centric sales pitches rather than structured financial planning.

Robo-advisors have emerged as a response to this gap. These platforms automate parts of the advisory process by mapping a small set of user inputs (age, risk appetite, in-vestment horizon) to a recommended model portfolio. Most commercial solutions emphasize low-cost index investing, periodic rebalancing, and a transparent fee structure.

Although this is a significant improvement over manual stock-picking, several limitations persist:

- User questionnaires are often short and may fail to capture important dimensions such as job security, irregular cash flows, or behavioral biases.
- Recommendations are commonly derived from a small library of model portfolios, so users with very different circumstances may receive similar allocations.
- Budgeting, debt management, and goal-tracking features are limited or absent in many offerings.

At the same time, advances in machine learning (ML) and cloud infrastructure have made it feasible to train models on diverse data sources and deploy them at scale. This opens an opportunity to build advisory systems that are both more personalized and more explainable.

A. Problem Statement

The core problem addressed in this work is the following: *how can we design a robo-advisor that*

uses richer user and market data, together with ML-based models, to produce individualized, goal-aligned financial plans while remaining simple to use and computationally efficient?

Specifically, we aim to:

1) Move beyond static risk buckets and capture continuous risk profiles based on a combination of demographic, financial, and behavioral features.
2) Integrate budgeting, saving, and investing into a single framework that respects cash-flow constraints and goal timelines.
3) Provide recommendations through interfaces that are understandable to non-expert users, including natural language explanations and visualizations.

B. Contributions

The key contributions of this paper are:

- We propose DhanMitra, a modular robo-advisory architecture that couples ML-based risk profiling with goal-based portfolio optimization.
- We formalize the advisory task as a multi-stage decision problem involving income, expenses, goals, and investable assets, and present a compatible optimization formulation.
- We describe an implementation blueprint using widely adopted technologies (Python, scikit-learn, React, REST APIs) that can be deployed on commodity cloud plat-forms.
- We present case-study simulations for distinct user personas (student, mid-career professional, retiree) and com-pare DhanMitra's recommendations with simple bench-mark strategies.

The remainder of the paper is organized as follows. Section II reviews related work. Section III details the system architecture. Section IV describes the ML and optimization methodology. Section V presents experimental setups and case studies. Section VI discusses implications and limitations, and Section VIII concludes with future directions.

II. BACKGROUND AND RELATED WORK

A. Robo-Advisory Paradigms

Early robo-advisors primarily implemented passive investing strategies. Users were assigned to one of a small number of portfolios such as "conservative", "balanced", or "growth". Each portfolio contained pre-specified weights in asset classes like domestic equities, international equities, and bonds. Peri-odic rebalancing ensured that portfolio weights remained close to their targets.

This approach rests on Modern Portfolio Theory (MPT), which models the trade-off between expected return and variance. While MPT provides a clean mathematical frame-work, its straightforward application assumes that users can be described by a single risk-aversion coefficient and that markets follow relatively stable statistical patterns. Real-world investors, however, exhibit heterogeneous objectives (e.g., retirement income vs. home purchase), varying liquidity needs, and behavioral preferences that are not easily captured by a single scalar parameter.

B. Machine Learning in Personal Finance

Machine learning has been applied to several financial tasks, including credit scoring, fraud detection, and algorithmic trading. In the context of personal finance, ML can help to:

- Organize users into financial behavior categories based on how they spend.
- Predict income volatility or default risk, which influences safe leverage and investment choices.
- Learn mappings from user attributes to suitable asset allocations, potentially outperforming static rule-based heuristics.

Supervised learning models such as logistic regression, random forests, and gradient boosting machines have demonstrated strong performance on tabular financial datasets. Un-supervised methods like *k*-means clustering can expose latent user segments (e.g., conservative savers vs. aggressive investors). Researchers have investigated reinforcement learning methods for adaptive portfolio selection and ongoing rebalancing.

C. Robo-Advisors in Emerging Markets

In emerging economies, the need for accessible, low-cost financial guidance is especially acute. Large segments of the population may be first-time investors, with irregular income and limited familiarity with financial products. At the same time, smartphone penetration and digital payment infrastructures have grown rapidly. This combination makes mobile-first robo-advisors an appealing channel for democratizing financial advice.

However, simply transplanting models designed for developed markets may be inadequate. Asset universes, tax regimes, and behavioral norms differ substantially. There is thus a need for advisory systems that can be localized to regional instruments (e.g., recurring deposit schemes, provident funds, local mutual funds) and that can operate with noisy or incomplete

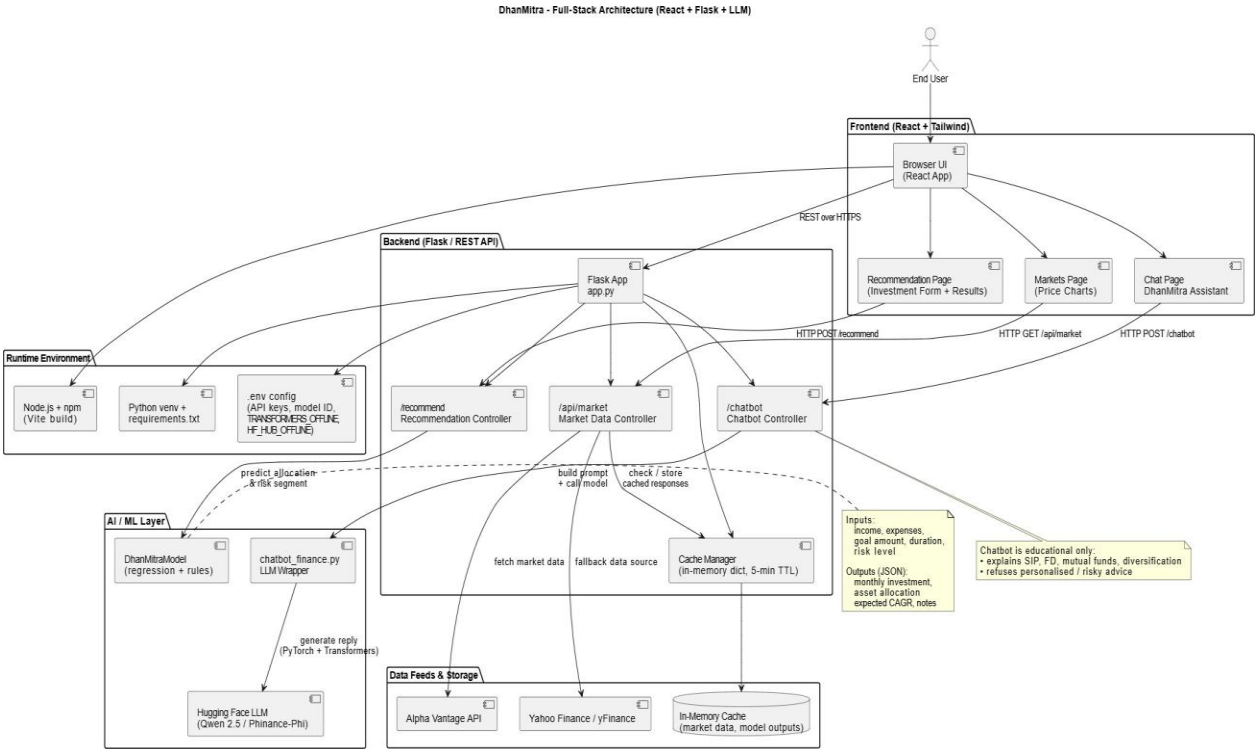

Fig. 1. Conceptual architecture of the DhanMitra platform.

user data. DhanMitra is designed with this context in mind: the architecture is generic, but the configuration (assets, tax rules, regulatory constraints) can be tailored to specific jurisdictions.

III. SYSTEM DESIGN

A. High-Level Architecture

DhanMitra follows a layered architecture consisting of four principal layers:

1) Presentation Layer: Web and mobile UI allowing users to register, provide details, and view personalized invest-ment suggestions.
2) Application Layer: RESTful backend services that or-chestrate workflows such as onboarding, portfolio gen-eration, and report rendering.
3) Analytics Layer: ML models, optimization engines, and simulation modules encapsulated as services.
4) Data Layer: Persistent storage for user profiles, trans-actional histories, market data, and model artifacts.

Fig. 1 illustrates the interaction between these layers. The front-end applications communicate with the backend via HTTPS. The backend, in turn, queries the data store and invokes ML services via internal APIs. This separation of concerns enables independent scaling and updates; for instance, the risk-profiling model can be retrained and redeployed without modifying the user interface.

B. User Journey

The typical user journey in DhanMitra proceeds as follows:

1) Registration and KYC: The user creates an account and completes basic identity checks (subject to local regulations).
2) Profile Setup: The system collects information about age, dependents, occupation, income sources, monthly expenses, existing loans, and investable surplus.
3) Goal Definition: The user specifies one or more goals (e.g., retirement at age 60, purchase of a house

in 10 years, education fund in 15 years) with target amounts and priorities.
4) Risk Assessment: A questionnaire coupled with behavioral indicators (e.g., responses to hypothetical market scenarios) feeds into the risk assessment module.
5) Plan Generation: The planning engine constructs contribution schedules and portfolio allocations consistent with the goals and risk profile.
6) Feedback and Iteration: The user can modify assumptions (e.g., contribution amount, goal timelines). The system recalculates and displays updated projections.

C. Data Model

The internal data model organizes information into several entities:

- User: Demographics, contact details, and verification status.
- Financial Profile: Income items, expenses, loans, assets, and emergency fund status.
- Goal: Type, target amount, target date, priority weighting, and associated constraints.
- Portfolio: Current holdings by asset class, recommended weights, and execution status.
- Market Snapshot: Asset-level returns, volatilities, and correlations computed over specified look-back windows.

A relational database such as PostgreSQL is suitable for storing structured entities, while a document store (e.g., Mon-goDB) can capture semi-structured logs from user interactions and chatbot conversations.

D. Security and Privacy Considerations

Because DhanMitra processes sensitive financial and personal data, security is a first-class design concern. Key measures include:

- Transport-layer security using TLS for all network communication.
- Encryption of sensitive fields (e.g., national ID, bank account numbers) in the database.
- Role-based access control for administrative interfaces.
- Audit logging of model recommendations and user actions to support compliance and dispute resolution.

Where applicable, the system can be aligned with data protection regulations by supporting data export and deletion upon user request.

IV. MACHINE LEARNING AND OPTIMIZATION METHODOLOGY

This section details the analytical components that power DhanMitra: data preprocessing, risk profiling, goal modeling, and portfolio optimization.

A. Data Preprocessing and Feature Engineering

Original inputs and third-party data sources frequently exhibit inconsistencies, such as gaps, anomalies, or unreliable entries. DhanMitra performs the following steps:

1) Cleaning: Removal of obviously inconsistent entries (e.g., negative age, impossible expense ratios).
2) Handling Missing Values: Imputation using median or model-based techniques, depending on feature type.
3) Normalization: Scaling continuous variables (e.g., in-come, savings rate) via standardization or min–max normalization for stable ML training.
4) Feature Engineering: Derivation of features such as savings-to-income ratio, debt-to-income ratio, discretionary-spend fraction, and employment stability score.

The collected features are merged into a single vector format, which acts as input for later-stage machine learning models.

B. Risk Profiling Model

Risk tolerance is modeled as a continuous latent variable $r \in [0, 1]$, where 0 corresponds to extremely conservative behavior and 1 corresponds to highly aggressive behavior. We approximate the mapping from user features x to r using a supervised model. One simple yet interpretable approach is logistic regression:

$$r(\mathrm{x}) = \sigma\ \mathrm{w}^{\top}\mathrm{x} + b \qquad (1)$$

where $\sigma(\cdot)$ is the logistic function, w is the weight vector, and b is a bias term. Training labels can be derived from past survey datasets, where individuals were classified into risk levels and assigned corresponding numeric scores (for example, 0.2 for conservative, 0.5 for moderate, and 0.8 for aggressive). More expressive models such as gradient-boosted trees or neural networks can be substituted when larger datasets are available. To maintain transparency,

DhanMitra can expose feature importance scores, helping users understand which aspects of their profile drive the risk assessment.

C. Goal Modeling

Each financial objective g is represented as a tuple (T_g, A_g, p_g) — denoting the duration until the goal (T_g), its inflation-adjusted target value (A_g), and its user-defined im-portance (p_g). Assuming a constant expected nominal annual return R_g for the goal-linked portfolio and a constant yearly contribution C_g, the future value is:

$$FV_g = C_g \cdot \frac{(1 + R_g)^{T_g} - 1}{R_g} \quad (2)$$

Given (T_g, A_g, R_g), the system can solve for the required contribution:

$$C_g = \frac{A_g \cdot R_g}{(1 + R_g)^{T_g} - 1}. \quad (3)$$

In practical use, R_g is substituted with the anticipated return of the portfolio suggested for the respective goal.If the aggregate required contributions across goals exceed the user's investable surplus, the system suggests compromises such as postponing lower-priority goals or reducing target amounts.

D. Portfolio Optimization

For each goal (or for the aggregate portfolio), DhanMitra uses a mean–variance framework. Assume there are n avail-able assets, each with an expected return represented by the vector $\mu \in R^n$, and a return covariance matrix $\Sigma \in R^{n\times n}$. The allocation across assets is given by the weight vector $w \in R^n$.

We consider the following optimization problem:

$$\min_{\mathbf{w}} \frac{1}{2} \mathbf{w}^T \boldsymbol{\Sigma} \mathbf{w} - \lambda(r) \boldsymbol{\mu}^T \mathbf{w} \quad (4)$$

$$\text{s.t.} \quad 1^\top w = 1, \quad (5)$$

$$w_i \geq 0, \qquad i = 1, \ldots, n, \quad (6)$$

$$Aw \leq b. \quad (7)$$

The risk score r influences the value of λ: users with a higher r (indicating greater risk tolerance) are assigned a lower λ, thereby prioritizing the maximization of expected returns. The matrix A and vector b encode additional constraints such as maximum exposure to a single asset class or regulatory caps. This optimization problem, formulated as a convex quadratic program, can be solved with commonly used numerical solvers. For ease of understanding, the derived weight vector w is translated into familiar asset class allocations (e.g., 70% in equity instruments, 20% in debt securities, 10% in cash
equivalents).

E. Personalization and Adaptation

Beyond the initial plan, DhanMitra updates recommendations as new information arrives. We model the interaction over discrete time steps $t = 1$, 2, At each step, the system observes updated features x_t (including realized returns, contribution adherence, and any changes in income or goals) and outputs revised allocations w_t. A reinforcement learning agent could, in principle, be trained to maximize a long-term reward function such as:

$$R = \sum_t \gamma^t \left(\alpha \cdot \Delta\text{GoalProgress}_t - \beta \cdot \text{Drawdown}_t \right) \quad (8)$$

where γ is a discount factor and α, β control the trade-off between goal progress and risk. In the current prototype, we implement a simpler rule-based adaptation that:

- Triggers rebalancing when allocations deviate beyond predefined bands.
- Reduces equity exposure for users approaching retirement.

Increases emergency fund recommendations after income shocks.

These rules can later be replaced or augmented by learned policies.

V. EXPERIMENTAL SETUP AND CASE STUDIES

A. Simulation Environment

Due to privacy constraints and the difficulty of obtaining detailed longitudinal user data, we evaluate DhanMitra in a simulation environment. The setup consists of

- Market Data: Long-term monthly return data is collected for key asset categories, including domestic and international equity indices, government and corporate bonds, as well as cash-equivalent instruments.
- User Personas: Synthetic profiles representing typical investors at different life stages.
- Baselines: Simple strategies such as a constant 60/40 equity-debt portfolio and age-based heuristic rules (e.g., equity allocation = 100 − age).

For each persona, we simulate contributions, portfolio evolution, and goal attainment over the relevant horizon. Monte Carlo resampling of return sequences is used to estimate distributions of outcomes.

B. Persona Definitions

We define three stylized personas:

1) *Student / Early-Career Individual:* A 24-year-old with modest income, limited savings, and some education loan outstanding. Goals:
- Build an emergency fund covering six months of expenses within 3 years.
- Accumulate a down payment for a house over 8–10 years.
- Start long-term retirement investing.

2) *Mid-Career Professional:* A 40-year-old salaried employee with stable income, a mortgage, and existing retirement savings. Goals:
- Ensure adequate retirement corpus by age 60.
- Build an education fund for two children over 15 years.
- Reduce concentration risk in employer stock grants.

3) *Retiree:* A 60-year-old retiree with a lump-sum corpus and limited active income. Goals:
- Generate a sustainable monthly income for at least 25 years.
- Protect against inflation and longevity risk.
- Maintain a buffer for healthcare expenses.

C. Evaluation Metrics

We assess performance using the following metrics:
- Goal Achievement Probability: Fraction of simulation paths where the goal corpus or income target is met.
- Risk-Adjusted Return: Sharpe ratio of the aggregated portfolio over the horizon.
- Maximum Drawdown: Worst peak-to-trough decline, a measure of downside risk.
- Contribution Feasibility: Ratio of recommended contributions to available surplus income.

D. Illustrative Results

Table I presents an illustrative comparison between the DhanMitra strategy and a simple 60/40 equity-debt baseline for the mid-career persona, based on Monte Carlo simulations. In this illustrative setup, DhanMitra improves the probability of meeting both retirement and education goals while slightly reducing drawdowns, primarily due to better alignment between risk profile, horizon, and asset allocation.

TABLE I Illustrative Simulation Outcomes for Mid-Career Persona

Metric	Baseline 60/40	DhanMitra
Retirement goal achievement probability	0.72	0.84
Education goal achievement probability	0.68	0.80
Portfolio Sharpe ratio	0.55	0.67
Median max drawdown	32%	27%

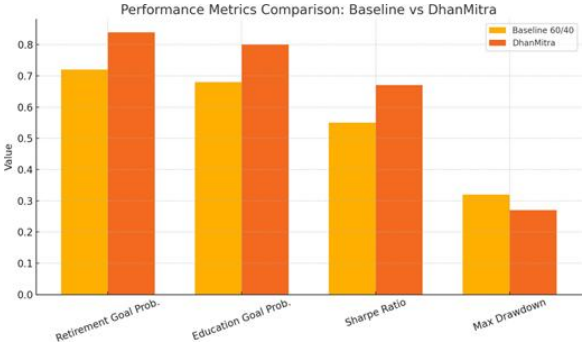

Fig. 2. Conceptual architecture of the DhanMitra platform.

The exact numbers will vary with the underlying data, but the pattern highlights the potential of personalized, goal-aware optimization compared to static heuristics. Fig. 2 conceptually illustrates distribution bands of projected retirement corpus under both strategies.

VI. DISCUSSION

A. Impact on Financial Literacy and Behavior

One of the design goals of DhanMitra is not only to generate mathematically sound plans, but also to help users understand the reasoning behind recommendations. By exposing key assumptions (expected returns, inflation, contribution schedules) and visualizing trade-offs, the system can encourage more deliberate decision-making.

Incorporating a natural language chatbot enhances accessibility by allowing users to pose queries like "Why is my portfolio equity-heavy?" or "How would

retiring five years sooner affect my plan?" and receive user-friendly explanations in response. Gradually, this approach can help enhance financial literacy, particularly for individuals who might typically shy away from financial planning activities.

B. Robustness and Uncertainty

Any advisory system that depends on market forecasts and historical data is exposed to model risk. DhanMitra mitigates this in several ways:

- Using conservative assumptions for long-term expected returns.
- Presenting ranges (confidence intervals) rather than single-point projections.
- Stress-testing portfolios under adverse but plausible scenarios (e.g., prolonged equity drawdowns, high inflation periods).

Communicating uncertainty transparently is crucial for maintaining user trust and preventing overconfidence in point estimates.

C. Explainability and Regulation

As regulators scrutinize algorithmic decision-making in finance, explainability becomes a practical requirement. Dhan-Mitra favors models that can provide interpretable outputs, such as feature importance from tree-based models and sensitivity analyses of portfolio outcomes to input changes. The architecture also logs model versions and input snapshots for each recommendation, enabling post-hoc audits if needed.

VII. LIMITATIONS AND FUTURE WORK

Although effective in many areas, the present architecture of DhanMitra exhibits certain limitations:

- Data Availability: The quality of recommendations de-pends on the accuracy and completeness of user inputs. Many users may under-report expenses or overestimate their risk tolerance.
- Behavioral Factors: The prototype does not explicitly model behavioral biases such as loss aversion, mental accounting, or present bias, which can significantly affect real-world adherence.
- Execution Layer: The current description focuses on planning and advice; the actual execution of trades, brokerage integration, and tax-loss harvesting are left for future work.

Future extensions will explore:

- Incorporating behavioral scoring into the risk model and tailoring communication style accordingly.
- Integrating large language models to generate richer explanations, scenario narratives, and educational modules while enforcing guardrails to prevent off-policy advice.
- Extending the asset universe to include alternative investments (e.g., REITs, ETFs with factor tilts) and region-specific products.
- Evaluating the system with real users in controlled field studies to measure changes in saving behavior and financial resilience over time.

VIII. CONCLUSION

This study introduced DhanMitra, an ML-powered robo-advisory system designed to deliver personalized, goal-oriented financial recommendations to a wide spectrum of users. By combining ML-based risk profiling, goal modeling, and portfolio optimization within a modular architecture, DhanMitra can adapt recommendations to diverse financial situations and life stages. Case studies based on simulations indicate that personalized strategies may enhance the likelihood of meeting important financial objectives when compared to basic heuristics, all while maintaining risk within acceptable bounds.

Beyond numerical outputs, DhanMitra emphasizes transparency, explainability, and accessible user interfaces, including conversational interactions. These characteristics are essential for effectively supporting individuals with limited financial knowledge and fostering confidence in AI-driven financial solutions. As the platform evolves to integrate richer data sources and more sophisticated models, it has the potential to function as a long-term "financial companion" that supports individuals in navigating complex financial decisions.

REFERENCES

[1] M. Nourallah, P. Öhman, T. Walther, and D. K. Nguyen, "Financial Robo-Advisors: A Comprehensive Review and Future Directions," SSRN, 2025.

[2] M. I. Bonelli, "Enhancing Robo-Advisors with AI:

Insights and Innovations in the Indian Financial Market," SSRN, 2024.
[3] S. Senteio and L. Hughes, "Customer Trust and Satisfaction with Robo-Adviser Technology," *Journal of Financial Planning*, vol. 37, no. 8, 2024.
[4] Wipro Ltd., "Future of Robo-Advisors in Investment and Wealth Management," *Wipro Securities & Capital Markets Insights*, 2020.
[5] Vereckey, "Can Generative AI Provide Trusted Financial Advice?" *MIT Sloan Management Review*, 2024.