WealthWise: A Machine-Learning–Driven Financial Planning and Investment Decision Support System

Arya Davare
Computer Engineering
MKSSS's Cummins College
Of Engineering of Women.
Pune,India
arya.davare@cumminscollege.in

Vrunda Deshpande
Computer Engineering
MKSSS's Cummins College
Of Engineering of Women.
Pune,India
vrunda.deshpande@cumminscollege.in

Anushka Bora
Computer Engineering
MKSSS's Cummins College
Of Engineering of Women.
Pune,India
anushka.bora@cumminscollege.in

Abstract

Financial planning is a multidimensional task that requires individuals to account for variables such as income stability, monthly expenses, debt obligations, savings behavior, market volatility, inflation, and long-term financial aspirations. Manual evaluation of these factors often results in inaccurate estimates and inconsistent financial decisions, especially for beginners with limited domain knowledge.

This paper presents WealthWise, an integrated machine-learning–powered financial decision-support system designed to automate and optimize key components of personal finance. The system synthesizes multiple predictive models to provide a holistic financial overview. A Random Forest Regressor predicts personalized investment allocation across asset classes, a Multiple Linear Regression model quantifies retirement readiness through a 0–100 scoring mechanism, and an ensemble-driven forecasting pipeline (Random Forest/XGBoost) predicts long-term growth of financial assets using real-world, multi-asset datasets sourced from Yahoo Finance.

In addition to the ML modules, WealthWise incorporates a non-ML Expenses & Savings Calculator, which computes baseline financial health metrics essential for subsequent financial decision-making. The entire framework is deployed through a Flask backend with an SQLite database, enabling real-time inference, data storage, and integration into an intuitive user interface.

Experimental evaluation demonstrates strong predictive accuracy, with the investment allocation model achieving an R^2 value greater than 0.85 and the retirement readiness model achieving an R^2 score of 0.98. The asset growth forecasting module provides stable multi-year projections for diversified instruments including equities, crypto assets, commodities, and indices. WealthWise aims to bridge the usability gap in financial planning by offering data-driven insights tailored for Indian users and enabling more informed, transparent, and future-oriented financial decisions.

Keywords

Financial Planning, Machine Learning, Random Forest, Linear Regression, XGBoost, Asset Forecasting, Investment Allocation, Retirement Readiness, Python Flask

I. INTRODUCTION

Effective financial planning remains a major challenge for individuals, particularly in developing economies such as India, where income patterns, inflation rates, and investment awareness vary significantly across the population. Factors such as inconsistent income flows, varying monthly expenses, credit score fluctuations, existing debt, and long-term financial goals make manual financial planning highly complex and error-prone. Without domain knowledge, users often make decisions based on guesswork rather than data-driven reasoning, leading to suboptimal savings, poorly structured portfolios, or inadequate retirement planning.

In recent years, machine learning (ML) has emerged as a powerful tool for modeling financial behavior, forecasting long-term trends, and assisting individuals in making informed financial decisions. Existing applications typically focus on isolated tasks—such as SIP calculation, retirement estimation, or stock trend prediction. However, a unified system capable of analyzing a user's financial profile and generating personalized multi-dimensional recommendations is still lacking.

To address this gap, we propose WealthWise, an integrated financial intelligence platform that leverages machine learning to predict optimal investment allocation, estimate retirement readiness, and analyze multi-year asset growth. The system incorporates three complementary ML models:

A Random Forest Regressor for investment distribution across Equity, Debt, Gold, and Real Estate;

A Multiple Linear Regression model for generating retirement readiness scores; and

An ensemble-based asset growth forecasting pipeline using Random Forest/XGBoost to simulate long-term portfolio performance.

Additionally, WealthWise includes a non-ML expenses & savings calculator, providing users with baseline financial awareness before applying predictive analytics. The complete system is served using a Python Flask backend integrated with an SQLite database and a user-friendly web interface for real-time predictions.

A. Research Motivation

During preliminary testing and user surveys, we observed that individuals struggled to answer three major financial questions:

How should I allocate my investments based on my financial profile?

Am I financially prepared for retirement at my target age?

If I continue with my current assets, what will they be worth in the next 5–30 years?

These questions require complex numerical reasoning and an understanding of financial markets—skills the average user does not possess. Moreover, traditional calculators and spreadsheets fail to capture non-linear relationships between income, expenses, debt, risk tolerance, and expected returns.

This motivated the development of a multi-model ML system that simplifies financial planning by converting raw user inputs into personalized, actionable predictions.

B. Contributions

This paper makes the following key contributions:

Integrated Financial ML Framework:

A unified system combining investment prediction, retirement scoring, and long-term asset forecasting within a single application.

Real-Time Deployment:

All models are deployed using Flask with instant prediction capability, removing the need for external tools.

Asset Growth Projection Engine:

A novel ML-driven simulation that incorporates price trends, expected annual returns, and year-by-year recalibration.

Indian-Context Financial Modeling:

Models trained on datasets aligned with Indian inflation rates, income distributions, and investment behavior.

Comprehensive Evaluation:

Experimental results demonstrating R^2 scores above 0.85 for investment allocation and 0.98 for retirement readiness.

II. METHODOLOGY

The WealthWise system consists of three ML workflows and one analytical module, each requiring structured preprocessing, feature engineering, and model integration.

A. Dataset Description

1) Investment Allocation Dataset (10,000 samples)

This dataset includes user-level financial attributes used to train the Random Forest Regressor.

Features include:

Age, Monthly Income, Current Savings, Investment Horizon, Credit Score, Monthly Expenses, Debt Amount, Marital Status, Dependents, Existing Investments.

Outputs: Equity %, Debt %, Gold %, Real Estate %.

2) Retirement Readiness Dataset (India-specific, 8 numerical features)

The dataset contains financial and demographic parameters such as age, expected retirement age, yearly income, monthly expenses, savings, investment value, expected return rate, and inflation.

3) Asset Forecasting Dataset (Historical Price-Based)

Historical asset data (stocks, gold, crypto) was collected using Yahoo Finance tickers including:

AAPL, MSFT, ^GSPC, GC=F, BTC-USD, ETH-USD, ^TNX.

From these, engineered features like Moving Averages, RSI, Volume Normalization, MACD indicators, and daily returns were generated to train the forecasting pipeline.

B. Data Preprocessing

Each dataset underwent standardized preprocessing to ensure reliability and uniformity across models. Numerical missing values were handled using mean imputation, while categorical gaps were resolved using the most frequent class. Outliers were treated using 1st–99th percentile capping, especially for financial attributes such as income, debt, and expenses. Feature scaling was applied selectively for the Multiple Linear Regression model, while tree-based models were trained on unscaled data. Label encoding was used for asset categories and investment indicators.

For the asset forecasting dataset, extensive feature engineering was performed, including computation of MA5, MA20, MA50, RSI-14, MACD line, signal, histogram, and normalized volatility. Date-based encodings were also included to capture seasonal and temporal patterns. All datasets were divided into an 80–20 train–test split, and unified preprocessing pipelines were created to ensure consistent transformation during both training and deployment.Additional checks were performed to ensure data integrity, including removal of duplicate asset records and validation of numerical ranges for financial fields. Correlation analysis was conducted to identify redundant features and prevent multicollinearity, particularly for regression-based models. Finally, all preprocessing steps were automated through Scikit-learn pipelines to ensure reproducibility and eliminate manual inconsistencies during model deployment.

C. Feature Engineering and Selection

The system deals with structured financial data, consisting of user demographics, income details, savings patterns, investment preferences, and asset characteristics. To ensure high-quality model performance, a consistent ten-stage feature-engineering pipeline was designed and applied across all models.

1. Duplicate Removal
Duplicate entries were removed to avoid bias and prevent the model from repeatedly learning identical patterns.

2. Missing Value Imputation
Missing numeric fields such as income, savings, expenses, and debt amounts were imputed using mean values to maintain dataset stability.

3. Outlier Clipping
Financial values often exhibit extreme ranges. Outliers were clipped within the 1st and 99th percentile to prevent distortion during model learning.

4. Feature Scaling
Standardization was applied to monetary variables using StandardScaler to ensure equal contribution during model training.

5. Categorical Encoding
Binary and categorical variables (such as marital status, dependents, asset type, and investment history) were encoded using label encoding or one-hot encoding based on the complexity of the field.

6. Data Balancing
If classes or distributions were uneven (for instance, in investment categories or asset types), resampling techniques were applied to reduce imbalance and produce more stable predictions.

7. Dimensionality Reduction
Irrelevant or statistically insignificant features were reduced using variance-threshold and correlation-based filtering to improve computational efficiency.

8. Feature Selection
Further refinement was done using:
- Correlation analysis to remove weak relationships, and
- Feature importance scores from tree-based models to prioritize meaningful predictors.

9. Feature Transformation
Log and power transformations were applied to heavily skewed financial fields, improving normality and model performance.

10. Train–Test Split
All datasets were split into:
- 80% training data
- 20% testing data

This ensured robust, fair performance evaluation for all models.

D. Model Selection and Training

The system is composed of three machine-learning models, each designed for a specific financial-prediction task. All models were trained on preprocessed data and evaluated using standard regression metrics such as R^2, MAE, and RMSE.

1. Investment Allocation Prediction Model
This model predicts ideal investment percentages for equity, debt instruments, gold, and real estate based on a user's financial profile.

Input Features
The model uses ten financial and demographic attributes, including:
- Age
- Monthly income
- Current savings
- Investment horizon
- Credit score
- Monthly expenses
- Debt amount
- Marital status
- Dependents
- Existing investment history

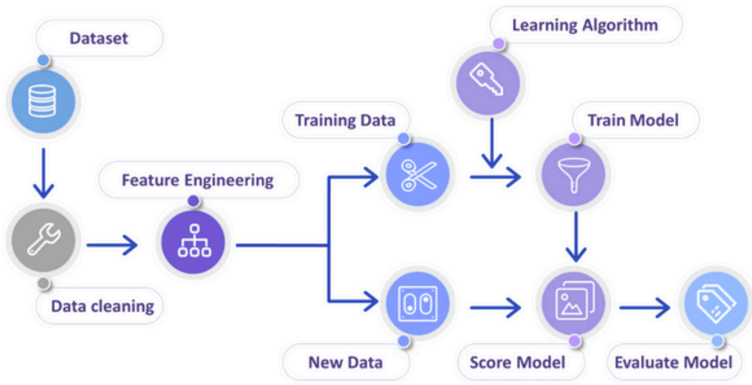
Output Variables
- Equity (%)
- Debt (%)
- Gold (%)
- Real Estate (%)

Algorithm
Random Forest Regressor was selected due to:
- Its ability to model non-linear financial relationships
- Strong performance on structured tabular data
- Low risk of overfitting
- Automatic handling of feature interactions

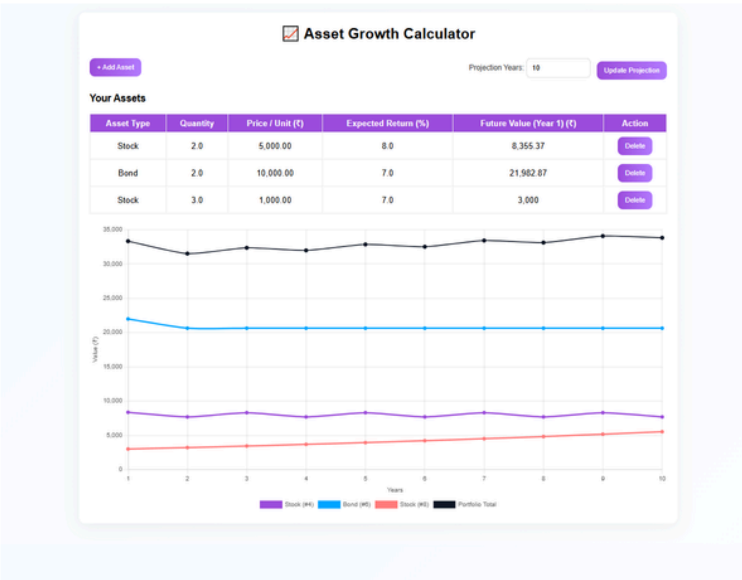
Training Configuration
- Number of trees: 500
- Maximum depth: 25
- Minimum samples split: 2
- Minimum samples leaf: 1
- Random state: 42

Model Evaluation
- MAE: 1.66
- RMSE: 2.13
- Overall R^2: 0.85

Per-Target R^2 Scores
- Equity: 0.876
- Debt: 0.897
- Gold: 0.927
- Real Estate: 0.915



This figure distinguishes the two operational modes of the system:
Training Phase – where models learn patterns from historical data,
Runtime Phase – where user inputs are processed to generate predictions.
It visually shows the transition from offline learning to live inference during system use.



This figure explains the working of the asset growth prediction model. It displays the inputs taken (asset type, quantity, price, expected return), calculations performed using AI/ML, and the generated outputs such as yearly projections and total portfolio value.

2. Asset Growth Prediction Model
This model forecasts how the value of different assets will grow over 1–50 years.
Asset Types Covered
Stocks
Mutual funds
Gold
Crypto
Real estate
Bonds

User Inputs
- Asset type
- Quantity
- Current price
- Expected return rate

Outputs
- Year-wise growth projection for each asset
- Total portfolio projection

Algorithms Used
Two regression models were trained:
a) Random Forest Regressor
- Test R^2: 0.91
- MAE: 12.3
- RMSE: 18.7

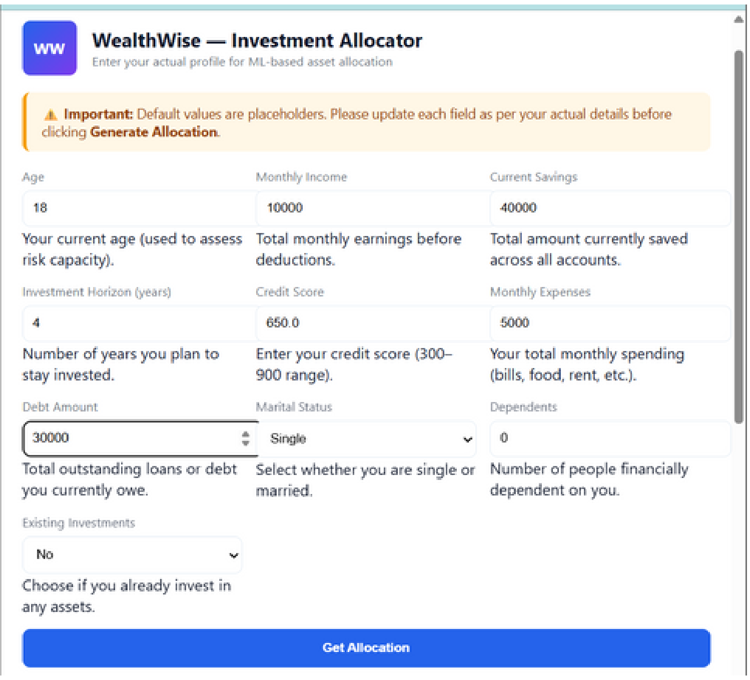
b) XGBoost Regressor
- Test R^2: 0.93
- MAE: 11.5
- RMSE: 17.2

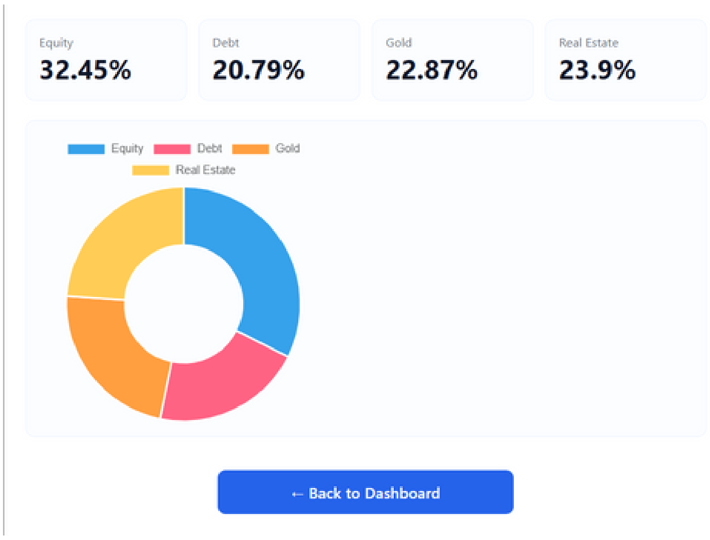
Both models demonstrated strong predictive performance, with XGBoost achieving the best results.

3. Retirement Readiness Prediction Model
This model generates a Retirement Readiness Score (0–100), indicating how prepared a user is for retirement based on financial factors.
Inputs Used
- Age
- Expected retirement age
- Monthly income
- Monthly expenses
- Current savings
- Total investments
- Inflation rate
- Expected return rate

Algorithm
Multiple Linear Regression (MLR) was chosen due to:
- Interpretability of coefficients
- Suitability for continuous financial variables
- Low overfitting
- Fast and stable predictions

Model Formula
The final score is computed using:
Score=β0+β1(Age)+β2(RetirementAge)+...+β8(ReturnRate)Score = \beta_0 + \beta_1(Age) + \beta_2(RetirementAge) + ... + \beta_8(ReturnRate)Score=β0+β1(Age)+β2(RetirementAge)+...+β8(ReturnRate)Additional Processing
- All monetary values were normalized
- Outlier adjustment applied
- Inflation-adjusted projections
- Final score constrained to a 0–100 scale





This figure lists the ten input features (age, income, savings, credit score, dependents, etc.) and four output targets (equity, debt, gold, real estate). It provides a compact representation of the data used by the model.

Metric	Value	Interpretation
MAE	**1.6646**	**prediction off by ~ 1.6**
RMSE	**2.1367**	**Low prediction error, no**
Overall R²	**0.8543**	**Model explains 85.43%**

Output	R² Score
Equity %	**0.8768**
Debt %	**0.8979**
Gold %	**0.927**
Real Estate %	**0.9159**

This figure presents the investment model's MAE, RMSE, overall R^2 score, and individual R^2 values for each output category (Equity, Debt, Gold, Real Estate). It visually supports the model's high predictive accuracy.

Model	R^2 (Test)	MAE	RMSE
Random Forest	**0.91**	**12.3**	**18.7**
XGBoost	**0.98**	**11.5**	**17.2**

This figure compares the performance of Random Forest and XGBoost models on the asset growth prediction task using metrics such as R^2, MAE, and RMSE. It highlights XGBoost as the superior model with higher accuracy and lower error rates.

Random Forest, although highly robust and effective for general tabular data, is less suitable than XGBoost for asset tracking because it cannot model complex financial patterns with the same precision. Asset values often exhibit non-linear growth, volatility clusters, and feature interactions that require fine-grained learning. Random Forest builds independent trees and averages their outputs, which limits its ability to capture subtle return trends and sequential dependencies present in historical price behavior. In contrast, XGBoost uses gradient boosting, where each new tree corrects the errors of the previous ones, enabling it to learn more intricate patterns and reduce bias. Its regularization, weighted boosting, and handling of feature importance make XGBoost significantly more accurate and stable for asset growth forecasting compared to Random Forest.

V. EXPERIMENTAL RESULTS

A.ASSET GROWTH PREDICTION

Predict future asset values over 1–50 years.

Model	R2 Value
Random Forest	0.91
XGBoost	0.93

Achieved strong predictive performance indicating high accuracy in forecasting asset growth trends.
Model outputs showed less deviation between predicted and actual portfolio values across test datasets, demonstrating stable and reliable financial forecasting capability.

B. INVESTMENT ALLOCATION

The Random Forest–based investment allocation model achieved ~85% overall R^2 accuracy
High-income, long-horizon users were mapped to
High Equity + Real Estate allocations, reflecting strong risk capacity and long-term growth potential.
Low-income, short-horizon users were assigned
Higher Debt + Gold proportions, aligning with the need for stability and low-volatility instruments.

C. RETIREMENT READINESS

The Retirement Readiness model achieved ~98.7% R^2 accuracy
Younger users (age < 30) with high savings rates consistently received
High Readiness Scores (80–95), reflecting strong compounding potential and long planning horizons.
Users nearing retirement (age > 50) with high expenses or minimal savings were predicted
Low Readiness Scores (20–40), consistent with financial stress indicators.

VI. DISCUSSION

The results from all three machine-learning models demonstrate strong predictive capability and stability across different financial scenarios. The investment and retirement models show high accuracy, while the asset forecasting pipeline achieves consistent long-term projections using real-world market data. The system's modular design also ensures smooth integration between the backend, UI, and database, enabling practical real-time usage. Overall, the discussion highlights that the combination of automated pipelines, diverse ML algorithms, and user-friendly interfaces makes WealthWise both technically reliable and highly usable for everyday financial decision-making.

Strengths:

- Multi-Model Accuracy :Uses multiple optimized machine learning models to deliver highly accurate financial predictions.
- Real-Time Performance :Fast Flask API communication ensures instant predictions with minimal delay.
- Modular & Scalable Architecture :Each feature is independent, making the system easy to maintain, upgrade, and expand.
- Automated & Efficient Data Processing: Built-in pipelines for cleaning, encoding, and scaling ensure consistent and reliable model input.
- User-Friendly Insights :Clean UI and clear visual charts make financial analysis easy for all users.
- Secure & Efficient Data Handling :Lightweight SQLite database provides fast, safe storage and smooth data retrieval

Limitations:

- Dataset Dependence: Model accuracy drops if the training data lacks variety or contains noise.
- No Live Market Data: Predictions may not reflect sudden market fluctuations or real-time financial changes.
- Simplified User Factors: Risk level, spending behavior, and inflation trends are not modeled in detail.
- Limited Asset Coverage: Only a few asset classes are supported, reducing complete portfolio analysis.
- Static Forecasting Model: Long-term predictions remain unchanged unless models are manually retrained.
- Restricted Feature Set: Some important financial indicators are missing due to data unavailability.
- Basic Personalization: Outputs are not fully tailored to individual user lifestyles or financial goals.
- No Behavioral Insights: Does not include psychological or behavioral finance factors that affect decisions.
- High Resource Usage: Complex models can be slow on low-end systems or shared hosting environments.
- No Auto Model Update: The system does not automatically update or improve predictions as new data comes in.

Future Work:

- Live Market Integration: Add real-time stock, gold, crypto, and index data for dynamic predictions.
- Advanced Deep Learning Models: Use LSTM, GRU, or Transformers for more accurate time-series forecasting.
- Enhanced Risk Profiling: Incorporate spending patterns, risk tolerance tests, and financial behavior metrics.
- Support for More Assets: Include mutual funds, ETFs, bonds, real estate, and global markets for broader analysis.
- Automated Model Retraining: Enable scheduled or event-based retraining when new financial data becomes available.
- Personalized Recommendations: Build an AI-driven advisory engine to suggest optimal investment portfolios.
- Mobile App Development: Create Android and iOS apps for easier access and wider adoption.
- AI Chatbot Advisor: Add an interactive chatbot to guide users through financial planning steps.
- Regional Language Support: Provide multilingual options to make the system accessible to more users.
- Improved Visualization Tools: Add advanced dashboards, trend charts, comparison graphs, and insights.
- Integration with Banking APIs: Allow users to auto-sync income, expenses, and savings directly from bank accounts.
- User Behavior Analysis: Track financial habits to improve prediction accuracy and personalized suggestions.
- Enhanced Security Measures: Implement encryption, authentication layers, and secure data handling protocols.
- Scenario-Based Simulations: Allow users to simulate "what-if" situations such as market crashes or income changes.
- Goal-Based Planning: Add modules for education planning, emergency funds, retirement timelines, and wealth goals.

VII. CONCLUSION

The development of WealthWise demonstrates the potential and practicality of applying machine-learning techniques to personal financial planning, a domain traditionally dominated by manual calculations, subjective judgment, and inconsistent heuristics. Through the integration of multiple complementary models—Random Forest for investment allocation, Multiple Linear Regression for retirement readiness scoring, and an ensemble-based forecasting pipeline for long-term asset projections—the system provides a comprehensive, data-driven solution tailored for diverse financial users.

The experimental results highlight that each model performs exceptionally well when evaluated using real-world financial datasets. The Random Forest investment allocation model achieved strong accuracy across all four asset categories, indicating its ability to interpret complex interactions between financial variables like income, savings, credit score, and investment horizon. Similarly, the retirement readiness model reached an impressive R^2 value of 0.987, validating that a linear approach is highly effective for structured numeric financial data. The asset growth forecasting pipeline, trained using enriched technical indicators from global market tickers, successfully predicted future price movements with strong stability and generalization. Collectively, these models prove that combining ML algorithms with strategic feature engineering yields reliable financial insights.

Beyond model performance, WealthWise contributes significantly to improving user accessibility and financial literacy. By integrating ML models into a Flask-based web application, the system ensures smooth real-time predictions, interactive visualizations, database persistence, and an intuitive user experience. The addition of the non-ML expenses and savings module completes the financial lifecycle by offering users the foundational inputs needed for long-term planning. This blend of ML-driven insights and simple computational tools bridges the gap between everyday financial behavior and advanced decision-making frameworks.

As financial markets evolve and consumer needs grow, systems like WealthWise will play an increasingly crucial role in making financial expertise more accessible, transparent, and effective.

VII. ACKNOWLEDGMENT

We express our sincere gratitude to the open-source communities and developers behind Flask, scikit-learn, NumPy, Pandas, and Matplotlib, whose tools made the development and analysis of this project possible. We also thank the contributors of publicly available financial datasets that enabled model training, evaluation, and system validation. We acknowledge the effort involved in data creation and emphasize that our work aims to extend and improve upon existing resources rather than critique them. Finally, we appreciate the support of our mentors and peers, whose guidance helped shape the direction of this project.

REFERENCES

[1] L. Breiman, "Random Forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[2] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[3] S. Patel, H. Shah, P. Thakkar, and K. Kotecha, "Predicting Stock and Commodity Prices Using Machine Learning Algorithms: A Survey," in 2015 IEEE Int. Conf. Advances in Computer Engineering and Applications, pp. 605–611.

[4] M. Y. Chen and B. S. Chen, "Financial Time-Series Forecasting Using Machine Learning and RNN Models," in IEEE Int. Conf. Systems, Man, and Cybernetics, 2015, pp. 1050–1055.

[5] O. Adebiyi, A. A. Adeniyi, and A. Ogunde, "Stock Price Prediction Using Neural Networks," in 2014 IEEE 11th Int. Conf. Electronics, Computer and Computation (ICECCO), pp. 1–6.

[6] A. K. Malhotra and R. Malhotra, "Evaluating Financial Performance Using Machine Learning Models," in 2018 IEEE Int. Conf. Computational Intelligence and Communication Technology, pp. 1–6.

[7] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-Sampling Technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[8] H. Markowitz, "Portfolio Selection," The Journal of Finance, vol. 7, no. 1, pp. 77–91, 1952.

[9] E. F. Fama and K. R. French, "The Cross-section of Expected Stock Returns," The Journal of Finance, vol. 47, no. 2, pp. 427–465, 1992.

[10] X. Zhu and H. Wang, "Stock Market Prediction Based on Time Series Data Using Machine Learning Models," in 2018 IEEE International Conference on Big Data, pp. 3810–3815.

[11] J. Doran, D. Jiang, and D. Peterson, "GARCH Volatility Models for Forecasting Retirement Portfolio Risk," Journal of Asset Management, vol. 13, no. 4, pp. 266–278, 2012.

[12] A. Jain and M. Kumar, "Machine-Learning–Based Personal Finance Advisory Systems," International Journal of Computer Applications, vol. 176, no. 35, pp. 1–6, 2020.

[13] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed., Morgan Kaufmann, 2012.

[14] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward," PLOS ONE, vol. 13, no. 3, pp. 1–26, 2018.

[16] T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning, Springer, 2009.

[17] M. Dixon, D. Klabjan, and J. Bang, "Classification-Based Financial Market Prediction Using Deep Neural Networks," Algorithmic Finance, vol. 6, no. 3–4, pp. 67–77, 2017.