MACHINE LEARNING IN FINANCIAL ANALYTICS

FinTrack: An Intelligent Expense Tracking and Behavioural Analysis System

S D. Tharun Kumaar, S. Lakchan

Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai, India

Published November 2026

Abstract

Personal expense tracking applications commonly record transactions but provide limited support for interpreting spending behaviour over time. This paper presents FinTrack, a locally deployable web-based system that combines expense recording, machine-learning-based behavioural classification, trend visualisation, rule-based recommendations, and month-to-month drift alerts. The application is implemented using Python and Streamlit, with SQLite for persistence, Pandas and NumPy for data processing, Scikit-learn for classification, and Matplotlib for visualisation. A Decision Tree classifier uses three features—expense-to-income ratio, savings rate, and number of active spending categories—to assign one of three labels: Saver, Balanced, or Overspender. The model is trained on 360 synthetically generated labelled profiles and evaluated using a stratified 80:20 holdout split. The project report records an accuracy of 97.22% (70 correct predictions from 72 test samples). The system also applies threshold-based alerts for increases in expenditure, declines in savings, and category-level spikes, and provides a dashboard for examining monthly and category-wise patterns. These results demonstrate the technical feasibility of integrating lightweight, interpretable analytics into a personal finance application. However, because the classifier is evaluated on synthetic data generated from rule-based class boundaries, the reported accuracy should not be interpreted as evidence of generalisation to real-world users. Further evaluation on representative real transaction histories is required.

Keywords: personal finance management; expense tracking; behavioural classification; decision tree; synthetic data; spending analysis; drift detection

1 Introduction

The growing use of digital payment methods has increased the frequency of everyday financial transactions, making it difficult for individuals to maintain a clear and consistent view of their expenditure. Conventional expense trackers primarily provide transaction records and category totals. Although these records are useful, users may still struggle to identify changing spending patterns, determine whether expenditure is proportionate to income, or translate historical data into practical actions.

FinTrack addresses this problem by combining expense logging with behavioural classification, monthly comparisons, visual analytics, and personalised recommendations. Its design emphasises local operation and interpretability: records are stored in a local SQLite database, and a Decision Tree classifier produces one of three behavioural labels—Saver, Balanced, or Overspender. The system is implemented as a Streamlit application and does not require a banking API or cloud service for its core functionality.

1.1 Problem statement and objectives

The system targets three practical limitations in basic expense management: (i) a lack of concise behavioural interpretation beyond totals, (ii) limited visibility into month-over-month changes, and (iii) generic advice that is not connected to an individual's recorded expense distribution. The objectives are to implement a multi-user expense tracker, classify monthly profiles using a lightweight supervised model, visualise financial patterns, generate rule-based recommendations, and detect significant changes between consecutive months.

1.2 Contributions

- A modular, local-first application integrating authentication, transaction storage, analytics, machine learning, and visualisation.
- A three-class Decision Tree pipeline using three derived financial features.
- A threshold-based behavioural alert module for total spending, savings, expense ratio, and category-level changes.
- A functional evaluation based on a synthetic holdout dataset and constructed end-to-end scenarios, with limitations explicitly identified.

2 Related work

Personal finance software has traditionally focused on transaction logging, categorisation, and summary reporting. More recent systems use supervised learning to classify transactions or segment users according to financial behaviour. Research in credit scoring and fraud detection has also explored a wide range of classification and anomaly-detection techniques. These domains are relevant methodologically, but their labels and objectives differ from the behavioural labels used in FinTrack.

Interpretability is an important consideration when analytical outputs are intended for non-specialist users. Decision Trees provide explicit decision paths and are computationally lightweight, making them suitable for a small local application. In contrast, more complex ensemble models may improve performance on some datasets but can be harder to explain. Visualisation complements classification by showing proportional expenditure, comparisons across months, and category-wise trends. The source project report cites prior work on transaction categorisation, behavioural segmentation, credit scoring, spending alerts, and visual perception [1–10]. The bibliographic details supplied in the original report should be independently verified before publication.

3 System architecture

FinTrack follows a layered, single-process architecture. The Streamlit interface accepts user input and provides five functional pages: Add Expense, Dashboard, Insights and Analysis, Month Comparison, and Manage Data. The database module manages user and transaction records in SQLite. The analytics module converts records into Pandas DataFrames and calculates summaries and month-to-month differences. The machine-learning module trains or loads a Scikit-learn Decision Tree and returns a behavioural label. The visualisation module uses Matplotlib to generate charts rendered in the Streamlit interface.

3.1 Data storage and access control

The database contains user and transaction tables. Monthly transaction records include income, five expense categories (Food, Transport, Shopping, Bills, and Entertainment), total expenditure, savings, expense ratio, and predicted behavioural category. Database queries are scoped by user identifier to separate records between accounts. The implementation described in the report hashes passwords with SHA-256. For production use, this should be replaced with a password-specific key-derivation function such as Argon2id, scrypt, or PBKDF2, with appropriate salts and rate limiting.

3.2 Analytics and visualisation

The analytics pipeline parses month identifiers, converts numerical fields, handles missing numeric values, sorts records chronologically, and computes total income, expenditure, savings, expense ratio, category totals, and the largest spending category. Matplotlib produces a donut chart for expense distribution, a grouped bar chart for monthly income, expenditure, and savings, a multi-line chart for category trends, a horizontal category bar chart, and a semi-circular expense-ratio gauge. These views support both within-month interpretation and comparison across time.

4 Methodology

4.1 Dataset construction

The classifier is trained on 360 synthetic monthly financial profiles generated with a fixed random seed of 42. The dataset is balanced across three classes: 120 Saver, 120 Balanced, and 120 Overspender samples. Each sample contains expense_ratio, savings_rate, and categories_used. The report defines the classes using expense-to-income ratio thresholds: below 0.50 for Saver, 0.50–0.79 for Balanced, and 0.80 or above for Overspender. Gaussian noise with a standard deviation of 0.02 is added to the savings-rate feature.

An 80:20 stratified split allocates 288 samples to training and 72 to testing, with 96 training and 24 test samples per class. Because labels are constructed from thresholds closely related to the input features, this dataset measures the model's ability to reproduce the synthetic labelling rule rather than its ability to infer validated financial well-being from real-world behaviour.

4.2 Feature engineering and model training

The expense-to-income ratio is computed as total monthly expenditure divided by monthly income. The savings rate is defined in the report as one minus this ratio, and categories_used counts active spending categories. The classifier is a Scikit-learn DecisionTreeClassifier with max_depth=5, min_samples_split=5, and random_state=42. The implementation also applies StandardScaler. Scaling is not required for Decision Trees, but retaining it may simplify later changes to a scale-sensitive model.

4.3 Behavioural drift detection

The compare_months() procedure compares a current monthly record with a previous record. Alerts are generated when total expenditure increases by more than 10%, savings decrease by more than 10%, the expense ratio exceeds 0.80, or any individual category increases by more than 20%. These are deterministic business rules rather than a statistical drift detector; their usefulness therefore depends on the selected thresholds and the quality and sufficiency of the user's history.

4.4 Evaluation protocol

The source report describes two evaluation stages. The classifier is evaluated on the 72-sample holdout set using accuracy and per-class precision and recall. The complete application is also checked using ten constructed monthly profiles and structured tests for authentication, data isolation, boundary predictions, data processing, alerts, chart generation, and the end-to-end pipeline. The drift module is additionally assessed on six constructed two-month scenarios.

5 Results

5.1 Classification performance

The reported holdout accuracy is 97.22%, corresponding to 70 correct classifications out of 72 test samples. All 24 Saver samples were classified correctly. For the Balanced and Overspender classes, the report records 23 correct predictions out of 24 for each class, with precision and recall of 95.8%. Table 1 gives the per-class results.

Table 1. Per-class classification results on the 72-sample synthetic holdout set.

Class	Test samples	Correct	Precision	Recall
Saver	24	24	100.0%	100.0%
Balanced	24	23	95.8%	95.8%
Overspender	24	23	95.8%	95.8%
Overall	**72**	**70**	**97.2%**	**97.2%**

Both reported errors occur near the Balanced–Overspender threshold, at expense ratios of 0.791 and 0.804. This is consistent with the threshold-based construction of the labels. The result should be interpreted cautiously: a small synthetic dataset and a holdout split from the same generating process do not establish performance on independent users or naturally occurring financial records.

5.2 Functional and drift evaluation

The report states that all five visualisation types rendered successfully in the tested cases and that the authentication, data-processing, alert, and end-to-end tests passed. In the six constructed drift scenarios, alerts were triggered for spending increases, a large spending spike, savings collapse, and a category spike; the stable control produced no alert. A 9.8% savings improvement remained below the 10% threshold. This scenario demonstrates the sensitivity of fixed thresholds to small changes around their decision boundary.

5.3 Reported operational metrics

The original report also records inference time below 1 ms per record, local SQLite write latency below 5 ms, and next-month prediction error within 8% in a six-month simulation. These values are reported by the project document; the underlying timing setup, forecast metric definition, and detailed raw measurements were not provided in sufficient detail here to independently reproduce them.

6 Discussion

The implementation illustrates how interpretable machine learning can be combined with simple temporal comparisons and visual summaries in a lightweight local application. Its main engineering strength is the separation of storage, analytics, classification, and presentation responsibilities. The resulting architecture is straightforward to test and can operate without a cloud service or bank integration.

Several limitations affect the strength of the evidence. First, the training labels are synthetic and threshold-derived, so high accuracy primarily reflects agreement with the constructed rules. Second, the dataset is small and does not represent the diversity of income patterns, household obligations, debt, or regional spending behaviours. Third, the reported functional scenarios are constructed rather than a user study. Fourth, savings-rate calculations and percentage changes require careful handling when income or previous-month values are zero or negative. Finally, SHA-256 alone is not an appropriate password-storage scheme for a production service. These limitations should be addressed before describing FinTrack as validated for real-world financial guidance.

7 Conclusion and future work

This paper presented FinTrack, a local-first expense tracking and behavioural analysis application that integrates SQLite storage, Pandas-based analytics, a Decision Tree classifier, Matplotlib visualisations, rule-based recommendations, and threshold-based alerts. The source report records 97.22% classification accuracy on a 72-sample synthetic holdout set and successful results for the specified functional tests. The work demonstrates a feasible software architecture for providing interpretable summaries of personal expenditure, while the synthetic evaluation limits conclusions about real-world predictive performance.

Future work should evaluate the system on ethically sourced, representative, and appropriately anonymised real financial data; compare the Decision Tree against simple rule-based baselines and alternative models; use cross-validation and report confusion matrices and confidence intervals; test recommendations with users; and define forecasting metrics and baselines transparently. Additional improvements include more robust password hashing, explicit handling of zero income and negative savings, data-sufficiency warnings, an uncertainty or borderline class near thresholds, and optional secure transaction import.

Data availability

The project report describes a synthetic dataset generated by the implementation, but the dataset file and source code were not included in the supplied PDF. For publication, provide a repository or supplementary file containing the generation script, model configuration, evaluation code, and synthetic data, if permitted.

Ethics statement

No human-subject experiment or real bank dataset is described in the supplied report. The evaluation presented here is based on synthetic profiles and constructed functional scenarios. Any future use of real financial records should include appropriate consent, minimisation, anonymisation, access control, and institutional review where applicable.

References

[1] Agrawal R, Gehrke J, Gunopulos D, Raghavan P. Automatic transaction categorization using machine learning. In: *Proceedings of the International Conference on Knowledge Discovery and Data Mining*; 2015. p. 112–120.

[2] Wang Y, Li X, Zhang J. Automated personal finance categorization using Naive Bayes classification on transaction narratives. *Journal of Financial Technology Research*. 2018;4(2):45–61.

[3] Sharma A, Kapoor N. Usability study of personal finance applications in the Indian market: Barriers to sustained engagement. *International Journal of Human-Computer Studies*. 2019;12(1):88–97.

[4] Lessmann S, Baesens B, Seow HV, Thomas LC. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research*. 2015;247(1):124–136.

[5] Koudela P, Novak M, Bartosik L. Behavioral segmentation of personal finance users using supervised machine learning. In: *Proceedings of the 14th International Conference on Business Intelligence and Data Mining*; 2020. p. 203–212.

[6] Zhang L, Chen H, Wang F. Comparing supervised classifiers for spending behavior prediction: Interpretability versus accuracy. *IEEE Access*. 2021;9:78342–78356.

[7] Bahnsen AC, Aouada D, Stojanovic A, Ottersten B. Feature engineering strategies for credit card fraud detection. *Expert Systems with Applications*. 2016;51:134–142.

[8] Jain S, Patel R. The effect of automated spending alerts on financial behavior: A controlled experiment. *Behavioral Finance Quarterly*. 2019;7(3):34–48.

[9] Few S. *Show me the numbers: Designing tables and graphs to enlighten*. 2nd ed. Burlingame (CA): Analytics Press; 2012.

[10] Heer J, Bostock M. Crowdsourcing graphical perception: Using Mechanical Turk to assess visualization design. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*; 2010. p. 203–212.