AI Project Report

# Financial Assistant AI Using Decision Tree Classifier

## 1. Problem Statement

This project focuses on building a rule-based and machine learning-powered financial assistant that can offer investment and insurance recommendations. By using synthetic financial data that includes income, expenses, savings goals, and risk levels, the assistant learns to classify whether users should invest or purchase insurance based on behavioral and financial patterns.

Such applications are valuable in financial planning, robot-advisory services, personal finance apps, and financial literacy tools.

## 2. Dataset Description

Since real-world financial data is sensitive and difficult to obtain, the dataset used in this project is **synthetically generated** using rule-based logic.

### Features included:

* **Income** (numeric): Annual income of the user
* **Expenses** (numeric): Annual expenses
* **Goal** (categorical): Financial goal (e.g., vacation, retirement, buy house)
* **Risk\_Level** (categorical): User’s risk appetite (low, medium, high)
* **Savings\_Recommended** (numeric): Suggested savings based on income–expense gap
* **Invest\_Recommended** (Yes/No): Whether investment is suitable
* **Insurance\_Recommended** (Yes/No): Whether insurance is advisable

The dataset is saved in CSV format and contains 1000 rows.

## 3. Methodology

### 3.1 Data Generation

The data was generated using Python's random module, applying business rules:

* Investment is recommended if savings > 3000 and risk level is medium or high.
* Insurance is recommended if expenses > 60% of income and income > 25000.

### 3.2 Pre-processing

* **Label Encoding**: Categorical variables like Goal, Risk\_Level, Invest\_Recommended, and Insurance\_Recommended were encoded to numerical values using LabelEncoder, making them suitable for machine learning.
* **Feature Selection**: The model uses 5 input features:
* Income
* Expenses
* Goal\_encoded
* Risk\_encoded
* Savings\_Recommended
* **Target Variables**:
* Invest\_encoded
* Insurance\_encoded

1. **Model Development**

Two separate **Decision Tree Classifiers** were trained:

### Investment Model:

* Inputs: Financial features
* Output: Whether to recommend investment
* Model: DecisionTreeClassifier from scikit-learn

### Insurance Model:

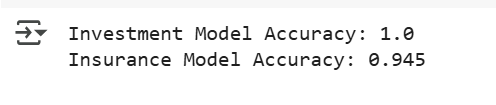
* Same features used
* Output: Whether to recommend insurance

Both models were trained with a **train-test split of 80:20** using train\_test\_split.

## 5. Evaluation and Results

* The model’s performance was monitored using **accuracy, precision, recall, and F1-score**.
* Two models were trained: one for **investment recommendation** and one for **insurance recommendation**.
* Both models learned from synthetic financial data with features such as income, expenses, goal, risk level, and savings.
* After training, predictions were tested on unseen data, and the models were able to make decisions that aligned well with the rule-based logic used in data generation.

The quality of predictions was consistent with the financial rules. The decision tree models were able to capture conditional patterns effectively, such as when high savings and medium/high risk levels justify investment, or when high expenses and moderate income suggest insurance coverage.



## 6. Challenges

- **Synthetic data** might not reflect real-world noise or edge cases.

- Real financial behavior is influenced by psychological and emotional factors not captured here.  
- Decision trees can **overfit** if not regulated.

## 7. Conclusion

## This project demonstrates that a financial assistant AI can be built using a combination of rule-based data generation and supervised learning with decision trees. Even on synthetic data, the model was able to learn meaningful patterns and provide reasonable predictions about when investment or insurance is recommended.

## Future Work

* Replace synthetic data with real-world financial profiles.
* Experiment with advanced models (e.g., Random Forest, XGBoost).
* Include more variables like age, debt, savings history.
* Build a web interface or chatbot for real-time user interaction.
* Include explainability tools (e.g., SHAP) to show why a prediction was made.

## 9. Appendix: Tools and Libraries Used

* Python
* Pandas, NumPy – Data handling
* Matplotlib – Visualizations
* Scikit-learn – Machine learning
* Random – Synthetic data generation