

## **Problem Selected:** Building a Loan Approval Classification Model

### **Prompt:**

Using the provided dataset containing customer demographic and financial information for loan applications, your task is to design and train a classification model using Python to predict whether a loan application will be accepted or rejected.

Consider the following points:

**Data Exploration and Preprocessing:** Begin by exploring the dataset to understand the features and their distributions. Perform necessary preprocessing steps such as handling missing values, encoding categorical variables, and scaling numerical data.

**Feature Selection and Engineering:** Identify relevant features that contribute to the prediction task. Engineer new features if necessary to improve model performance.

**Model Selection and Training:** Experiment with various classification algorithms such as logistic regression, decision trees, random forest etc. to find the most suitable model. Evaluate their performance using appropriate metrics like accuracy, precision, recall, and F1-score.

**Hyperparameter Tuning and Validation:** Optimize the chosen model by fine-tuning hyperparameters using techniques like grid search or randomized search. Validate the model using cross-validation techniques to ensure robustness.

**Interpretation and Analysis:** Interpret the model's predictions and feature importance. Discuss the key factors influencing loan approval or rejection based on your model's insights.

Provide a clear breakdown of your approach, including code snippets, visualizations, and explanations at each stage. Additionally, propose possible avenues for further improving the model's performance.

### **Rationale:**

This prompt engages in the entire process of building a classification model for loan approval prediction. It activates prior knowledge by emphasizing data exploration, preprocessing, feature selection, and model evaluation. The task is goal-oriented and specific, guiding through each step of the machine learning pipeline. It encourages critical thinking by asking for insights into model interpretations and suggestions for further enhancements. The language used is clear and avoids ambiguity, providing a structured framework for problem-solving in Python-based classification tasks.