**Predictive Model Plan**

# 1. Model Logic (Generated with GenAI)

Use a GenAI tool (e.g., ChatGPT, Gemini) to generate the logic or structure of your predictive model.  
- You may include pseudo-code, a step-by-step process, or a simplified code snippet.  
- Briefly explain what the model is designed to do.

Paste your GenAI-generated output below or describe the logic in your own words:

I will be using decision tree, logistics regression and neural network for credit risk assessment. I am using ChatGPT and Gemini to generate code while I give prompt to it.

Decision tree model:

* Using ChatGPT and Google Gemini for code generation, a SMOTE-balanced Decision Tree Classifier was built to predict delinquent accounts, tuned for recall to better identify high-risk customers.

BEGIN

LOAD dataset "data\_iterative\_imputed"

SPLIT dataset into features X and target y (Delinquent\_Account)

SPLIT X, y into training and testing sets (70% train, 30% test) using stratified sampling

APPLY SMOTE to the training set only to handle class imbalance

→ Generate synthetic samples for minority class

INITIALIZE Decision Tree Classifier with:

- class\_weight = 'balanced' (give equal importance to both classes)

- max\_depth = 5 (limit tree growth to avoid overfitting)

- min\_samples\_split = 10 (minimum samples to split a node)

- min\_samples\_leaf = 5 (minimum samples per leaf node)

- criterion = 'entropy' (split quality measure)

TRAIN the Decision Tree model on resampled training data

PREDICT delinquency status for the original test set

EVALUATE model using:

- Confusion Matrix

- Precision, Recall, F1-score

OUTPUT results for performance review

END

Logistics Regression model:

BEGIN

LOAD dataset and split into X (features) and y (target)

SPLIT into train/test sets

APPLY SMOTE to training data to balance classes

INITIALIZE Logistic Regression with tuned parameters

TRAIN model on balanced data

PREDICT probabilities on test set

APPLY threshold = 0.3 for classification

EVALUATE using confusion matrix and classification report

END

* Using ChatGPT and Google Gemini for code generation, a SMOTE-balanced Logistic Regression model with tuned hyperparameters was trained to predict delinquency. Probabilities were converted to labels using a 0.3 threshold to improve recall for high-risk customers. Model performance was assessed via precision, recall, and F1-score to ensure balanced risk detection

Neural Network Model:

* This SMOTE-balanced neural network predicts delinquent accounts by learning complex patterns from customer features. Class weights heavily prioritize recall to catch more risky customers, while a lowered decision threshold (0.2) further boosts sensitivity. Early stopping prevents overfitting, and scaling ensures stable gradient updates*.*

Load dataset and drop target + ID columns from features

Split data into training and testing sets (stratified split)

Apply SMOTE to balance minority (delinquent) class in training data

Scale features using StandardScaler

Define neural network:

- Dense(64, relu) + Dropout(0.4)

- Dense(32, relu) + Dropout(0.3)

- Dense(1, sigmoid) for binary classification

Compile model with:

- Adam optimizer (lr=0.0005)

- Binary crossentropy loss

- Accuracy, Precision, Recall as metrics

- Class weights to penalize false negatives (1:8 ratio)

Train model with early stopping, 200 epochs, batch size 8

Predict probabilities on test set

Convert probabilities to binary predictions using threshold = 0.2

Output confusion matrix and classification report

# 2. Justification for Model Choice

Explain why you selected this specific model type (e.g., logistic regression, decision tree, neural network). Consider:  
- Accuracy  
- Transparency  
- Ease of use or implementation  
- Relevance for financial prediction  
- Suitability for Geldium’s business needs

For the delinquent dataset, where a delinquent account is represented by 1 and a non-delinquent account by 0, I evaluated multiple model types for classification, including **decision trees, logistic regression, and a simple neural network**. The choice of logistic regression as the preferred model was based on several factors:

1. **Accuracy:** Among the models tested, logistic regression provided the best balance of accuracy and recall, making it effective in correctly identifying delinquent accounts while minimizing false negatives. This is crucial in financial settings where missing a delinquent account can have significant implications.
2. **Transparency:** Logistic regression is highly interpretable. Its coefficients clearly indicate the influence of each feature on the probability of delinquency, which is valuable for explaining predictions to stakeholders and complying with regulatory requirements. Decision trees are also interpretable but tend to overfit, and neural networks are largely black-box models.
3. **Ease of Use / Implementation:** Logistic regression is straightforward to implement and computationally efficient, which allows for quick model training and easier integration into Geldium’s existing systems. Neural networks require more tuning and computational resources, while decision trees require careful pruning to avoid overfitting.
4. **Relevance for Financial Prediction:** Logistic regression is widely used in credit scoring and financial risk modeling due to its probabilistic output and ability to handle binary outcomes effectively. It allows Geldium to predict the likelihood of delinquency for each account.
5. **Suitability for Geldium’s Business Needs:** Given Geldium’s need for actionable insights with clear explanations, logistic regression strikes the best balance between **predictive performance** and **interpretability**, ensuring decisions based on model outputs are reliable and justifiable.

# 3. Evaluation Strategy

Outline how you would evaluate your model’s performance. Include:  
- Which metrics you would use (e.g., accuracy, precision, recall, F1 score, AUC)  
- How you would interpret those metrics  
- Any plans to detect or reduce bias in your model  
- Ethical considerations in making predictions about customer financial behavior

**1. Metrics Used:**  
To evaluate the model, I would use the following metrics:

* **Accuracy:** Measures the overall proportion of correct predictions. In this case, accuracy is 0.52, which is low, indicating that simple accuracy alone may not reflect the model’s effectiveness due to class imbalance.
* **Precision:** Measures how many of the predicted positive cases (delinquent accounts) are actually positive. For class 1 (delinquent), precision is 0.20, indicating many false positives.
* **Recall (Sensitivity):** Measures how many actual positive cases are correctly predicted. For class 1, recall is 0.67, showing that the model detects most delinquent accounts, which is important in financial risk management.
* **F1 Score:** Harmonic mean of precision and recall; balances the trade-off. For class 1, F1 is 0.31, highlighting that while recall is decent, low precision reduces overall effectiveness.

**Interpretation:**

* The low precision but high recall for delinquent accounts suggests that the model tends to flag many accounts as delinquent, capturing most actual delinquents but also producing false alarms.
* Adjusting the **decision threshold to 0.3** improves recall, which aligns with a business priority of minimizing missed delinquent accounts, even at the cost of more false positives.

**2. Detecting and Reducing Bias:**

* **Class Imbalance:** Since the dataset has fewer delinquent accounts (minority class), I plan to monitor class-wise metrics rather than overall accuracy.
* **Threshold Adjustment:** Using a lower threshold (0.3 instead of 0.5) to improve recall for the minority class.
* **Regularization and Class Weights:** Using L1 penalty and potentially experimenting with class\_weight='balanced' to reduce bias toward the majority class.
* **Cross-Validation:** Implement k-fold cross-validation to ensure model performance is consistent across different subsets of data.

**3. Ethical Considerations:**

* Predictions about financial behavior can significantly impact customers (e.g., loan approvals, interest rates).
* Ensuring fairness: Avoid discrimination based on sensitive attributes like gender, caste, or location.
* Transparency: Logistic regression coefficients provide interpretable insights into why an account is flagged as delinquent.
* Continuous Monitoring: Regularly evaluate the model to ensure it does not unfairly penalize certain groups or overfit to past biases.