**Predictive Model Plan – Student Template**

# 1. Model Logic (Generated with GenAI)

**Use a GenAI tool (e.g., ChatGPT, Gemini) to generate the logic or structure of your predictive model.**  
**Paste your GenAI-generated output below or describe the logic in your own words:**

**Predictive Modeling Approach for Credit Delinquency Detection**

This model is developed to predict whether a customer will default on credit (delinquent) or not. It’s a **binary classification task**, where the target label is Delinquent\_Account:

* **1** → Delinquent
* **0** → Not Delinquent

The structure of the model was generated using GenAI and follows these logical steps:

**Step-by-Step Framework:**

**Step 1: Importing Customer Data**  
The dataset includes multiple customer-level financial and behavioral features:

* Demographics: Age, Employment\_Status, Location
* Financials: Income, Loan\_Balance, Debt\_to\_Income\_Ratio, Credit\_Score
* Credit Behavior: Credit\_Utilization, Missed\_Payments, Account\_Tenure, Credit\_Card\_Type
* Payment history over 6 months: Month\_1 to Month\_6

**Step 2: Data Cleaning and Encoding**

* Handle missing entries using the **mean** or **median** for numeric fields.
* For categorical features like Employment\_Status, Credit\_Card\_Type, and Location, apply **one-hot encoding**.
* Payment status values for Month\_1 to Month\_6 are encoded as:
  + On-time → 0
  + Late → 1
  + Missed → 2

**Step 3: Create New Features (Feature Engineering)**

* **Total Missed Payments** = Sum of values in Month\_1 to Month\_6
* **Late Payment Ratio** = (Number of late payments) / 6

These features help capture recent payment trends and strengthen model accuracy.

**Step 4: Normalize the Data**

* Standardize numerical columns using **StandardScaler**, which improves model performance and stability.

**Step 5: Prepare for Modeling**

* Split the dataset into **training (80%)** and **testing (20%)** using **stratified sampling** to maintain class balance.

**Step 6: Select and Train the Model**

* Choose **Logistic Regression** for its simplicity, interpretability, and wide use in credit scoring.
* Fit the model to the training data so it learns the relationships between customer features and credit behavior.

**Step 7: Make Predictions**

* Generate probability scores for each customer.
* Predict a customer as **delinquent** if the score > 0.5, else **not delinquent**.

**Step 8: Evaluate Performance**  
Evaluate the model using multiple classification metrics:

* **Precision** – Measures accuracy of positive predictions
* **Recall** – Measures how many actual delinquents were found
* **F1 Score** – Balances precision and recall
* **AUC-ROC** – Measures the model’s ability to distinguish between classes

**Step 9: Fairness and Ethical Checks**

* Inspect for **bias** across groups (age, gender, location, etc.)
* Perform fairness tests like:
  + **Disparate Impact**
  + **Equal Opportunity**

If bias is found, apply mitigation strategies:

* **Pre-processing**: Rebalance data
* **In-training**: Add fairness constraints
* **Post-processing**: Adjust decision thresholds per group

**Final Goal of the Model:**  
The main purpose is to **flag high-risk customers** in advance so that Geldium can:

* Offer early support or interventions
* Reduce overall credit losses
* Maintain ethical, transparent, and fair AI systems

# 2. Justification for Model Choice

Logistic Regression was selected as the preferred algorithm for predicting credit delinquency based on the following considerations:

**Reliable Performance in Binary Classification**

Logistic Regression is highly effective for problems where the outcome involves two categories — such as identifying if a customer is delinquent (1) or not (0). It performs well when patterns in the data follow a linear relationship or can be transformed into one.

**Clear and Interpretable Results**

One of the key strengths of this model is its **transparency**. It provides easy-to-understand coefficients that highlight which features contribute most to the prediction. This is essential for:

* Meeting **regulatory and compliance** standards,
* Building **confidence among business stakeholders**, and
* Offering **clear, actionable decisions**.

**User-Friendly and Computationally Light**

Logistic Regression is straightforward to implement, doesn’t need complex settings or tuning, and runs efficiently even on basic systems. Its simplicity makes it ideal for fast deployment without compromising on reliability.

**Financial Industry Alignment**

In the credit and finance sector, **interpretability matters a lot**. Logistic Regression has been widely adopted in **credit risk modeling** because it outputs risk probabilities and allows analysts to understand **why a customer was flagged as risky**.

**Best Match for Geldium’s Needs**

For Geldium, it’s important that the model is not just accurate, but also **fair, ethical, and explainable**. Logistic Regression achieves this balance. While other models like decision trees can become overly complex and overfit the data, and neural networks act as "black boxes," Logistic Regression ensures both performance and clarity — making it the most suitable choice for this use case.

# 3. Evaluation Strategy

To ensure the model performs accurately, fairly, and responsibly, the following evaluation strategy will be followed:

**Key Performance Metrics:**

* **Precision**: This measures how many of the customers predicted as delinquent are actually delinquent. A high precision score helps avoid wrongly flagging good customers, which can lead to unnecessary actions.
* **Recall (Sensitivity)**: This checks how many actual delinquent cases the model correctly detects. A high recall is important to ensure high-risk individuals are not overlooked.
* **F1 Score**: The F1 score balances both precision and recall. It is especially useful when dealing with imbalanced datasets where both false positives and false negatives are costly.
* **AUC-ROC Curve**: This evaluates the model’s ability to distinguish between delinquent and non-delinquent customers across various probability thresholds.

**Bias Detection and Fairness Checks:**

* **Review for Data Imbalance**: Analyze the dataset for unequal representation across groups like age, job status, or geographic location.
* **Disparate Impact Analysis**: Investigate whether predictions unfairly vary between different subgroups — such as some groups receiving more false positives or negatives than others.
* **Equal Opportunity Testing**: Check if the model maintains similar true positive rates for all demographic groups, ensuring that one group is not unfairly disadvantaged.

**Bias Mitigation Approaches (If Needed):**

* **Pre-processing**: Use data balancing techniques like oversampling, undersampling, or re-weighting to make sure all groups are properly represented before training.
* **In-training Solutions**: Incorporate fairness-aware learning objectives or constraints during model training to reduce bias.
* **Post-processing Adjustments**: Modify decision thresholds across groups to improve fairness in model outputs.

**Ethical Considerations:**

* **Transparency**: Provide clear and understandable reasons behind every prediction, especially in cases where customers are marked high risk.
* **Fairness**: Prevent indirect or proxy discrimination by carefully selecting features and performing regular fairness checks.
* **Data Privacy**: Ensure all model development and deployment comply with local data protection laws, such as GDPR.
* **Human Review**: Always include a human analyst in the decision-making loop. AI predictions should support, not replace, human judgment.
* **Customer Impact Monitoring**: Actively track and reduce the potential harm caused by false predictions. Set up channels to receive and respond to customer feedback.
* **Ongoing Monitoring**: Continuously evaluate the model over time to detect data drift, reduced accuracy, or new bias patterns. Retrain or update the model as needed.