**Predictive Model Plan**

# 1. Model Logic (Generated with GenAI)

To implement a simple logistic regression predictive model in Python for high delinquency rates, you would follow these steps: data preparation, model training, and evaluation. This structured approach, using the scikit-learn library, ensures a robust and transparent process.

### 1. Data Preparation

Load and Preprocess the Data: Start by loading the dataset into a pandas DataFrame. This is where you'll apply the data cleaning and transformation steps identified in the EDA report. Specifically, you'll need to handle missing values, standardize categorical variables, and address any anomalies. For example, you would use predictive imputation for the Income variable, median-by-group imputation for Loan\_Balance, and simple median imputation for Credit\_Score.

Define Features and Target: Separate the dataset into features (the independent variables) and the target variable. The target variable would be the delinquency status, which is likely a binary flag (e.g., Delinquent\_Account), and the features would be the predictive variables identified in the EDA, such as Missed\_Payments, Credit\_Utilization, and Employment\_Status. You may also need to perform feature engineering, creating new variables to enhance predictive power.

Split the Data: Divide the preprocessed data into training and testing sets. The training set is used to train the model, while the testing set is reserved to evaluate its performance on unseen data. A common split is 80% for training and 20% for testing.

### 2. Model Training

Import Libraries: Import the necessary libraries from scikit-learn, including LogisticRegression for the model and other relevant modules for preprocessing, such as StandardScaler to normalize numerical features and OneHotEncoder to encode categorical features.

Instantiate and Train the Model: Initialize the LogisticRegression model. Then, use the .fit() method to train the model on your training data. The model will learn the optimal weights for each feature to predict the probability of a customer being delinquent.

### 3. Model Evaluation

Make Predictions: Use the trained model to make predictions on the testing dataset. These predictions will be a set of probabilities that can be converted into binary outcomes (delinquent vs. not delinquent) based on a chosen threshold.

Evaluate Performance: Compare the model's predictions to the actual values in the testing set using evaluation metrics such as accuracy, precision, recall, and the F1 score. Precision measures the proportion of positive predictions that were actually correct, while recall measures the proportion of actual positives that were correctly identified. In a financial context, recall is often critical because it's important to identify as many delinquent accounts as possible.

Interpret Coefficients: Examine the coefficients of the logistic regression model. The magnitude and sign of each coefficient indicate the strength and direction of its relationship with the target variable. For instance, a positive coefficient for Credit\_Utilization would suggest that as a customer's credit utilization increases, their likelihood of delinquency also increases, which aligns with the EDA findings.

### 4. Pseudo code for the Model

# Import necessary libraries

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report

import pandas as pd

# Load the dataset (replace with your actual data loading)

# This is a placeholder for demonstration purposes

data = {'Missed\_Payments': [1, 4, 0, 5, 2],

'Credit\_Utilization': [0.5, 0.9, 0.3, 0.85, 0.7],

'Delinquent\_Account': [0, 1, 0, 1, 0]}

df = pd.DataFrame(data)

# Define features (X) and target (y)

X = df[['Missed\_Payments', 'Credit\_Utilization']]

y = df['Delinquent\_Account']

# Split data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Initialize and train the Logistic Regression model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Make predictions on the test set

y\_pred = model.predict(X\_test)

# Evaluate the model

print(classification\_report(y\_test, y\_pred))

# 2. Justification for Model Choice

Logistic Regression is an excellent choice for this task due to its balance of simplicity, interpretability, and effectiveness for binary classification problems like predicting credit delinquency.

* **Accuracy**: For many financial datasets, a well-tuned logistic regression model can achieve high predictive accuracy, often rivaling more complex models, especially when the relationship between features and the target variable is relatively linear.
* **Transparency**: This is a key advantage. The model's coefficients are easily interpretable, allowing analysts to understand the impact of each variable (e.g., missed payments, credit utilization) on the probability of delinquency. This transparency is crucial for financial institutions that need to explain risk decisions to customers and regulators.
* **Ease of Use or Implementation**: Logistic regression is straightforward to implement and computationally efficient, making it a good starting point for a predictive modeling project. It has few hyperparameters to tune, which simplifies the model-building process.
* **Relevance for Financial Prediction**: The model is well-suited for predicting financial events, as it directly outputs a probability score (e.g., the likelihood of a customer becoming delinquent) rather than a simple classification. This score is highly valuable for risk management, as it enables the setting of different risk thresholds.
* **Suitability for Geldium's Business Needs**: Given the need for a foundation for "accurate risk assessment and early intervention strategies" , a simple and transparent model like logistic regression is ideal. It provides reliable predictions while allowing the business to clearly understand **what** is driving the risk, which is essential for developing effective strategies. It also serves as an effective baseline against which more complex models can be compared in the future

# 3. Evaluation Strategy

To evaluate the predictive model's performance, a combination of standard classification metrics will be used, with a specific focus on identifying and mitigating potential biases and addressing ethical considerations.

### 1. Key Evaluation Metrics

For evaluating the model's performance, the following metrics will be used:

* **Accuracy:** The proportion of all predictions that were correct. While easy to understand, it can be misleading with imbalanced datasets (e.g., if only a small percentage of accounts are delinquent).
* **Precision:** Of all the accounts the model predicted as delinquent, this metric measures how many were actually delinquent. High precision is important to avoid incorrectly flagging low-risk customers, which can damage customer relationships.
* **Recall:** Of all the accounts that were actually delinquent, this measures how many the model correctly identified. High recall is critical for risk management, as failing to identify delinquent accounts could lead to financial losses.
* **F1 Score:** The harmonic mean of precision and recall. This metric provides a balanced view of the model's performance, especially when there's an uneven class distribution.

### 2. Interpreting the Metrics

The interpretation of these metrics will focus on the specific business context of credit risk:

* **High Recall is a Priority:** For a credit delinquency model, it is often more costly to miss a delinquent account than to incorrectly flag a healthy one. Therefore, the focus will be on achieving a high recall score to ensure that as many at-risk customers as possible are identified for early intervention.
* **Balancing Precision and Recall:** While high recall is important, a very low precision could lead to too many false positives. The goal is to find an optimal balance using the F1 score, which will allow for a cost-effective and efficient intervention strategy.

### 3. Bias Detection and Reduction Plan

The model will be checked for potential biases related to demographic variables, which is a critical consideration in financial modeling.

* **Bias Detection:** The model's predictions will be compared across different demographic groups (e.g., age, location, employment status) to ensure there is no disparate impact. We will look for significant differences in false positive or false negative rates between groups.
* **Bias Reduction:** If bias is detected, methods such as re-sampling the training data, adjusting the classification threshold for specific groups, or using fair machine learning algorithms will be considered to ensure the model is equitable.

### 4. Ethical Considerations

Predicting customer financial behavior requires careful ethical consideration:

* **Fairness and Transparency:** The model must be fair and transparent. It should not use protected characteristics (such as race or gender, even if not explicitly present in the dataset) to make predictions. The use of an interpretable model like logistic regression helps in understanding why a prediction was made.
* **Human Oversight:** Model predictions should not be used as the sole basis for decisions. They should serve as a tool for financial experts to guide their decisions, with human oversight and the option to override the model's recommendation.
* **Data Privacy:** All customer data will be handled with strict adherence to privacy regulations. The model will not store personally identifiable information that is not essential for its function.