**Predictive Model Plan-Credit Delinquency Forecasting**

# 1. Model Logic (Generated with GenAI)

**Goal:** Predict which customers are at risk of credit card delinquency using their demographics, credit behaviour, and payment history.

**Type of Model (Simple & Complex Options)**

* **Option 1 – Logistic Regression (Simple):** Easy to implement, interpretable, widely used in finance.
* **Option 2 – XGBoost (Complex):** Handles non-linear patterns, higher accuracy, provides feature importance.

**Recommended:** Start with **Logistic Regression** as a transparent baseline and then test **XGBoost** for better accuracy.

**Key Input Features (Top 5)**

✅ **Income** – Customers with lower income are at higher delinquency risk.  
✅ **Credit\_Score** – Lower credit scores indicate higher default probability.  
✅ **Credit\_Utilization** – Higher utilization increases risk.  
✅ **Missed\_Payments** – Strong indicator of future delinquency.  
✅ **Debt\_to\_Income\_Ratio** – Shows repayment capacity relative to debt.

**General Workflow (How the Model Works)**

1️⃣ **Data Ingestion:** Load customer dataset with demographics, credit history, and payment behaviour.  
2️⃣ **Preprocessing:** Handle missing values, encode categorical features, and scale numeric variables.  
3️⃣ **Feature Engineering:** Create new variables like % of on-time payments, total missed payments, and utilization buckets.  
4️⃣ **Model Training:** Train Logistic Regression as a baseline, then test XGBoost for improved accuracy.  
5️⃣ **Prediction Output:** Model predicts the probability of delinquency → classify customers into **Low / Medium / High risk**.

**Sample Pseudocode**

import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.preprocessing import StandardScaler

# Load dataset

df = pd.read\_csv("customer\_data.csv")

# Select features and target

X = df[['Income', 'Credit\_Score', 'Credit\_Utilization', 'Missed\_Payments', 'Debt\_to\_Income\_Ratio']]

y = df['Delinquent\_Account']

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, stratify=y)

# Scale features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Train Logistic Regression

model = LogisticRegression()

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

# Predict delinquency probabilities

y\_pred\_prob = model.predict\_proba(X\_test\_scaled)[:, 1]

**Summary (2–3 Sentences)**

Logistic Regression is chosen as the baseline model because it is **transparent, interpretable, and widely used for financial risk prediction**. XGBoost will also be tested as a more **accurate, non-linear alternative**. The model uses key features such as **Income, Credit\_Score, Credit\_Utilization, Missed\_Payments, and Debt\_to\_Income\_Ratio** to predict the probability of a customer becoming delinquent.

# 2. Justification for Model Choice

Logistic Regression is selected as the baseline model because it is **highly transparent**, **interpretable**, and widely used in financial risk prediction. It provides clear coefficients that show how each feature (e.g**., income, credit score, missed payments**) influences delinquency risk, which is crucial for regulatory compliance and business understanding. However, Logistic Regression may not fully capture complex, non-linear relationships in the data.

To improve predictive performance, we also recommend testing **XGBoost**, a gradient boosting model that handles complex feature interactions and typically achieves **higher accuracy** and **AUC scores**. Although XGBoost is less interpretable than Logistic Regression, explainability tools like **SHAP values** can help provide insights into feature importance.

**Balancing Performance and Explainability:**

Logistic Regression offers strong interpretability but moderate performance, while XGBoost provides higher accuracy at the cost of reduced transparency.

By starting with Logistic Regression and then testing XGBoost, Geldium Finance can achieve both regulatory-friendly explainability and improved predictive power.

**Justification Connected to Geldium’s Goals:**

Using Logistic Regression ensures compliance and business transparency, while incorporating XGBoost provides advanced accuracy for better risk segmentation. This combination aligns with Geldium’s objective of adopting a structured, data-driven approach to identify at-risk customers ethically and effectively.

# 3. Evaluation Strategy

**✅ Metrics to Use:**

* **Recall** – Ensures most at-risk customers are correctly identified (reduces false negatives).
* **Precision** – Prevents too many good customers from being wrongly flagged (reduces false positives).
* **F1-score** – Provides a balanced measure of precision and recall, useful for imbalanced data.
* **ROC-AUC** – Measures how well the model separates delinquent and non-delinquent customers.

**✅ Interpretation of Metrics:**

* High **recall** → The model successfully identifies most delinquent customers.
* High **precision** → The customers flagged as delinquent are actually risky.
* **F1-score** → Helps choose the best balance between precision and recall.
* **ROC-AUC** → Closer to 1 indicates strong predictive power.

**✅ Bias Detection and Reduction:**

* Evaluate model performance for different **income groups, locations, and employment statuses** to ensure fairness.
* Use only **financial behaviour features** to avoid discrimination.
* Apply **SHAP values** to confirm that predictions are based on relevant factors.

**✅ Ethical Considerations:**

* Predictions will be used to **offer support** (e.g., reminders, flexible payment options), not to unfairly deny services.
* Model decisions will be **explainable to regulators and customers**.
* Data privacy will be maintained to protect customer information.