**Predictive Model Plan – Student Template**

Use this template to structure your submission. You can copy and paste content from GenAI tools and build around it with your own analysis.

# 1. Model Logic (Generated with GenAI)

### Delinquency Prediction Model

The model is designed to **forecast the probability that a customer will become delinquent** on their payments. It analyzes a customer's financial history and personal information to provide a risk score, which helps a business make informed decisions about credit approvals, account management, and collection strategies.

### Sample Model Pseudocode

The following pseudocode outlines a simplified modeling pipeline using a Gradient Boosting Machine (XGBoost), which is highly effective for this type of classification problem.

Code snippet

# 1. Data Loading and Preparation

Load the dataset into a DataFrame.

# 2. Feature Engineering & Selection

Create a new feature, `Credit\_Utilization\_Risk`, by dividing Loan\_Balance by Credit\_Score.

Select the top 5 most relevant features:

FEATURES = ['Missed\_Payments', 'total\_amount\_due', 'amount\_paid\_last\_month', 'Credit\_Score', 'Loan\_Balance']

# 3. Data Splitting

Split the dataset into an 80% training set and a 20% testing set.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(DATA[FEATURES], DATA['delinquent\_flag'], test\_size=0.2, random\_state=42)

# 4. Model Training

Initialize the XGBoost model.

model = xgboost.XGBClassifier(objective='binary:logistic', eval\_metric='logloss')

Train the model on the training data.

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

# 5. Prediction and Evaluation

Make predictions on the testing set.

predictions = model.predict(X\_test)

Evaluate the model's performance.

accuracy = accuracy\_score(y\_test, predictions)

precision = precision\_score(y\_test, predictions)

recall = recall\_score(y\_test, predictions)

f1 = f1\_score(y\_test, predictions)

# 6. Output

Print evaluation metrics.

print(f"Accuracy: {accuracy:.2f}")

print(f"Precision: {precision:.2f}")

print(f"Recall: {recall:.2f}")

print(f"F1-Score: {f1:.2f}")

# Example of a final risk prediction for a new customer

new\_customer\_data = {'Missed\_Payments': 2, 'total\_amount\_due': 5000, 'amount\_paid\_last\_month': 200, 'Credit\_Score': 650, 'Loan\_Balance': 25000}

risk\_probability = model.predict\_proba(new\_customer\_data)[0, 1]

print(f"Delinquency risk probability for new customer: {risk\_probability:.2%}")

# 2. Justification for Model Choice

I selected **XGBoost** for the credit delinquency prediction model due to its high **accuracy**, **suitability for financial data**, and **practical implementation**.

* **Accuracy:** It's an **ensemble method** that consistently outperforms simpler models by correcting errors iteratively, capturing complex, non-linear patterns in financial data for a more precise forecast.
* **Relevance:** It is widely used in the financial industry for its effectiveness in handling noisy, high-dimensional, and imbalanced datasets, which are common in risk assessment and fraud detection.
* **Implementation:** The XGBoost library is easy to use and well-documented. It also simplifies the process by handling missing data and categorical variables without extensive preprocessing.
* **Suitability for Geldium's Needs:** The superior predictive power of XGBoost directly translates to a better ability for a business like Geldium to mitigate risk and make informed lending decisions. The gain in accuracy far outweighs the model's complexity.

# 3. Evaluation Strategy

### Model Evaluation Plan

Evaluating our delinquency prediction model requires a focus on key metrics that go beyond simple accuracy to provide a full picture of its performance and ethical implications.

#### Key Metrics & Interpretation 📉

* **Precision:** Of all customers the model predicted as delinquent, how many were actually delinquent? A high score here reduces false alarms, saving Geldium from unnecessarily flagging customers.
* **Recall:** Of all customers who actually became delinquent, how many did the model correctly identify? A high score here is critical for risk mitigation, ensuring we don't miss at-risk accounts.
* **F1-Score:** This metric balances precision and recall, providing a single score that is especially useful for an imbalanced dataset where delinquent cases are rare.
* **AUC (Area Under the Curve):** Measures the model's ability to distinguish between delinquent and non-delinquent customers. A higher score means better overall predictive power.

#### Bias & Ethical Considerations ⚖️

* **Bias Detection:** We will examine the model's predictions across different demographic groups to ensure fairness. We will use metrics like **demographic parity** (similar prediction rates across groups) and **equal opportunity** (similar recall rates across groups).
* **Bias Mitigation:** If bias is detected, we will use techniques like **reweighing** to adjust for imbalances in the training data to promote fairness.
* **Human Oversight:** The model's predictions will serve as a recommendation, not a final decision. Human experts will review the flagged accounts to ensure all ethical and qualitative factors are considered before any action is taken.
* **Transparency:** We will use **SHAP values** to explain the reasoning behind each prediction in a human-understandable way, ensuring customers can understand the factors that influenced their risk score.