# Task 2 - Predictive Model Plan

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)

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:

Objective: The goal of this model is to predict the probability of a customer becoming delinquent (1) or remaining non-delinquent (0) in the next billing cycle. This allows the Collections team to proactively intervene with high-risk customers.

Model Type: Random Forest Classifier This approach uses an ensemble of multiple decision trees to create a robust prediction. It is chosen for its ability to handle non-linear relationships (e.g., the anomaly where high credit scores still result in default) and provide interpretable feature importance.

***GenAI-Generated Logic (Pseudo-code):***

```
# Step 1: Data Ingestion & Preprocessing

def preprocess_data(customer_data):

  # Impute missing 'Annual_Income' and 'Loan_Balance' with Median values

  customer_data['Annual_Income'].fillna(customer_data['Annual_Income'].median(), inplace=True)

  customer_data['Loan_Balance'].fillna(customer_data['Loan_Balance'].median(), inplace=True)


  # Feature Selection: Isolate key predictors identified in EDA

  features = customer_data[['Credit_Utilization', 'Late_Payment_Count',

                'Debt_to_Income_Ratio', 'Annual_Income', 'Loan_Balance']]

  target = customer_data['Delinquent']

  return features, target


# Step 2: Model Training (Random Forest)
```

```
def train_model(features, target):
    # Initialize Random Forest with 100 trees for stability
    model = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=42)

    # Split data into Training (80%) and Testing (20%) sets
    X_train, X_test, y_train, y_test = train_test_split(features, target, test_size=0.2)

    # Train the model to learn patterns (e.g., High Utilization + Low Income = Risk)
    model.fit(X_train, y_train)
    return model

# Step 3: Prediction & Risk Scoring
def predict_risk(model, new_customer):
    # Output: Probability score (0.0 to 1.0)
    risk_probability = model.predict_proba(new_customer)[:, 1]

    # Classification: Flag as "High Risk" if probability > 50%
    if risk_probability > 0.50:
        return "Action Required: High Risk"
    else:
        return "Low Risk"
```

## 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

I selected the **Random Forest Classifier** as the optimal model for Geldium because it directly addresses the specific data anomalies discovered during the Exploratory Data Analysis (EDA):

- **Handling Inverted Data Logic (Accuracy):** Our EDA revealed a "Credit Score Inversion" where delinquent customers strangely had higher average credit scores (591) than non-delinquent ones (575). A standard linear model (like Logistic Regression) assumes *Higher Score = Lower Risk* and would fail on this dataset. Random Forest creates non-linear decision boundaries, allowing it to adapt to this specific contradiction without manual interference.

- **Explainability for Intervention (Transparency):** Unlike "Black Box" Neural Networks, Random Forest provides **Feature Importance** scores. This allows the Collections team to see exactly *why* a customer was flagged (e.g., "Flagged due to 90% Credit Utilization," not just "Bad Score"). This transparency is critical for designing targeted intervention strategies and meeting regulatory requirements.

- **Robustness to Data Gaps:** With approximately 8% of **Annual Income** data missing (as found in Task 1), Random Forest is highly resilient. When combined with Median Imputation, it prevents outliers in the remaining income data from skewing the risk predictions for the entire portfolio.

## 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

To ensure the model is both accurate and fair, I will use the following evaluation framework:

**Key Metrics:**

- **Recall (Sensitivity): (Primary Metric)**
    - *Goal:* Maximize Recall.
    - *Interpretation:* In credit risk, a "False Negative" (missing a defaulter) is costly because the bank loses money. We prioritize catching as many actual delinquents as possible, even if it means slightly more false alarms.
- **Precision:**
    - *Goal:* Maintain reasonable Precision.
    - *Interpretation:* We want to minimize "False Positives" (flagging good customers) to avoid wasting the Collections team's time and annoying loyal clients.
- **AUC-ROC Score:**
    - *Goal:* Score > 0.75.
    - *Interpretation:* This measures how well the model distinguishes between "Good" and "Bad" customers across all threshold levels.

**Bias Detection & Ethical Considerations:**

- **Demographic Parity Check:** I will calculate the "Positive Prediction Rate" across different demographic groups (e.g., Age < 30 vs. Age > 50). If the model flags 40% of young people as risky but only 5% of older people, we must investigate if Age is acting as an unfair bias rather than a true risk factor.
- **Error Rate Analysis:** I will check if the model makes *more mistakes* for certain groups. For example, if it has a high False Positive rate for low-income customers, it might unfairly penalize them.
- **Mitigation:** If bias is found, I will retrain the model by rebalancing the training data or removing sensitive proxy variables (like Location) to ensure fair lending practices.