Predictive Model Plan

1. Model Logic

Here is the step-by-step process for building a credit risk prediction model using GenAI:

1) Problem Definition and Data Understanding

- Define the Target Variable: Clearly define what "credit risk" means. In the context of your dataset, the target variable is Delinquent_Account (a binary outcome: 1 for delinquent, 0 for not delinquent).
- Select the Time Window: Determine the period over which you are predicting delinquency (e.g., predicting the risk of default within the next 12 months).
- Identify Key Feature Categories:
  - Credit Profile: Credit_Score, Credit_Utilization, Debt_to_Income_Ratio, Loan_Balance.
  - Payment History: Missed_Payments, Month_1 to Month_6.
  - Customer Stability: Age, Income, Employment_Status, Account_Tenure.
  -

2) Data Preprocessing and Feature Engineering

- Handle Missing Values:
  - Numerical: Impute with the median (as we did for Income, Credit_Score, and Loan_Balance) to maintain the shape of the distribution and minimize the impact of outliers.
  - Categorical: Impute with the mode or create a new 'Missing' category.
- Clean Categorical Data: Standardize text entries (e.g., ensure variations like 'employed' and 'EMP' are combined into one consistent category).
- Feature Engineering (Creating Predictive Power):
  - Payment Aggregation: Create a feature like Total_Late_Missed by counting the number of 'Late' or 'Missed' entries across Month_1 to Month_6. This summarizes short-term behavior.
  - Debt Ratios: Ensure key ratios like Debt_to_Income_Ratio are properly calculated.
- Handle Categorical Variables: Convert all categorical features (e.g., Employment_Status, Credit_Card_Type) into a numerical format suitable for modeling using One-Hot Encoding (creating binary columns for each category).

3) Model Selection and Training

- Split the Data: Divide the cleaned data into three sets:
  - Training Set (e.g., 60%): Used to train the model.
  - Validation Set (e.g., 20%): Used for hyperparameter tuning.
  - Test Set (e.g., 20%): Used for final, unbiased performance evaluation.
- Choose Model Architecture: Popular models for credit risk prediction include:
  - Logistic Regression: Simple, highly interpretable, and provides probability scores.
  - Gradient Boosting Machines (GBMs) or XGBoost: Powerful ensemble methods for higher accuracy.
  - Random Forests: Effective at handling non-linear relationships.
- Train the Model: Use the Training Set to fit the chosen model, ensuring that the model learns the relationship between the features and the Delinquent_Account target.

4) Model Evaluation and Validation

- Evaluate on the Test Set: Use the Test Set (which the model has never seen) to evaluate performance.
- Key Evaluation Metrics:
  - Area Under the Receiver Operating Characteristic Curve (AUROC or AUC): Measures the model's ability to distinguish between delinquent and non-delinquent accounts (the higher, the better).
  - Precision, Recall, and F1-Score: Important for handling imbalanced data (since delinquency is usually rare).
  - Stability (Population Stability Index - PSI): Ensures that the feature distributions used for training are stable when compared to new, incoming data.

This model is designed to predict the credit risk of a customer by determining the likelihood that the customer will become delinquent on their payments or default on a loan within a specified future time period.

2. Justification for Model Choice

Justification for Model Type Selection (Logistic Regression)

1. Transparency (Interpretability)

- High Transparency: Logistic Regression is a linear model, meaning the relationship between the features and the prediction is easy to see and explain. It provides coefficients for each feature, which directly indicate the direction and strength of its impact on the risk.
- Regulatory Requirement: For financial predictions, regulators often require models that are highly interpretable (explainable) to ensure fair lending

practices and prevent discrimination. Logistic Regression is the industry standard for generating regulatory-compliant scorecards.

2. Relevance for Financial Prediction (Risk Quantification)

- Probability Output: The model's output is a probability(P), specifically the probability of default(P(Delinquent_Account)= 1)). This is exactly what credit institutions need to set pricing, reserve capital, and make informed lending decisions.
- Scorecard Conversion: The linear form of the model is mathematically straightforward to convert into a simple, point-based credit scorecard, which is the primary tool used by loan officers.

3. Ease of Use or Implementation (Simplicity)

- Fast and Stable: Logistic Regression is computationally efficient to train and deploys quickly. It is less prone to overfitting than complex models like Neural Networks, and its predictions are very stable.
- Low Maintenance: It requires less hyperparameter tuning than non-linear models, making it easier for Geldium's business team to implement and maintain.

4. Accuracy (As a Baseline)

- Good Starting Point: While not always the most accurate model (complex non-linear models often beat it), Logistic Regression provides a strong benchmark. If a sophisticated model doesn't significantly outperform the Logistic Regression baseline, the added complexity is usually not justified.
- Feature Focus: In credit risk, risk is often captured by a few key linear relationships (e.g., higher Credit_Score = lower risk), which Logistic Regression handles well.

5. Suitability for Geldium's Business Needs (Prioritization)

- Risk vs. Opportunity: For a financial institution, Transparency and Trust in the model often outweigh a marginal increase in Accuracy. A simple, explainable model minimizes compliance risk and builds confidence among stakeholders and regulators.
- Default Choice: Unless higher accuracy is strictly necessary to differentiate between highly competitive risk buckets, Logistic Regression is the default, most suitable choice for initial deployment in credit risk.

3. Evaluation Strategy

For a model like credit risk prediction, where the target class (delinquency) is usually rare, we must use metrics that account for the relative costs of different types of errors.

The key metrics and Interpretation:

- Area Under the ROC Curve (AUC) - Primary Metric. A high AUC (close to 1.0) means the model consistently ranks risky customers higher than safe ones. This is the best overall measure of model quality.
- Recall (Sensitivity) - Critical Metric. A high recall minimizes False Negatives (FN). A False Negative is a "bad" loan approved, which results in direct financial losses (principal and interest).
- Precision - Business Metric. A high precision minimizes False Positives (FP). A False Positive is a "good" customer rejected, which results in a lost business opportunity.
- F1 Score - Used to find the optimal balance when we want to minimize both False Positives and False Negatives, particularly useful when the classes are imbalanced.
- Accuracy - Least Useful. Can be misleading when the target is imbalanced (e.g., if 95% are non-delinquent, a model predicting '0' for everyone would still have 95% accuracy).

Bias Detection and Reduction

Detection Plan

1) Identify Protected Groups: Define sensitive attributes (where available and permissible to use for testing), such as Age Bracket, Location, and Employment Status (as a proxy for socio-economic status).
2) Use Fairness Metrics:
    - Disparate Impact (DI): Measures the ratio of the approval rate for the unprivileged group to the approval rate for the privileged group. A ratio significantly less than 0.8 often indicates bias.
    - Equal Opportunity Difference: Checks if the Recall (True Positive Rate) is approximately equal across groups. A significant difference means the model is better at identifying good loan candidates in one group than another.

Reduction Plan

1) Data Preprocessing: Re-sample the training data to balance the representation of protected groups, ensuring the model sees a balanced picture.
2) Threshold Adjustment: For groups exhibiting bias, adjust the classification probability threshold locally to achieve parity in a chosen fairness metric (e.g., setting a lower rejection threshold for the under-represented group).

3) Use Fair Models: Employ adversarial de-biasing techniques or constrained optimization models that explicitly include fairness constraints during the training phase, forcing the model to minimize predictive differences across groups.