**Predictive Model Plan**

# 1. Model Logic (Generated with GenAI)

**Code:**import pandas as pd

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

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report, roc\_auc\_score

# Load the imputed dataset

# NOTE: This file was generated in a previous step

file\_path = 'Delinquency\_prediction\_dataset\_imputed.csv'

df = pd.read\_csv(file\_path)

# Drop any rows with remaining missing values after imputation

df.dropna(inplace=True)

# Define the features (X) and the target variable (y)

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

target = 'Delinquent\_Account'

X = df[features]

y = df[target]

# Split the data into training and testing sets (80% training, 20% testing)

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(solver='liblinear', random\_state=42)

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

# Make predictions on the testing data for evaluation

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

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

# Evaluate the model's performance

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

report = classification\_report(y\_test, y\_pred)

auc\_roc = roc\_auc\_score(y\_test, y\_proba)

# Print the results

print("Logistic Regression Model Performance")

print("-------------------------------------")

print(f"Accuracy: {accuracy:.4f}\n")

print(f"AUC-ROC Score: {auc\_roc:.4f}\n")

print("Classification Report:")

print(report)

# Print the model's coefficients

print("\nModel Coefficients (Feature Importance):")

for feature, coef in zip(features, model.coef\_[0]):

print(f"{feature}: {coef:.4f}")

**Output:**

**Correlation Matrix**

|  | **Credit\_Utilization** | **Missed\_Payments** | **Credit\_Score** | **Delinquent\_Account** | **Total\_Missed\_Late\_Payments** |
| --- | --- | --- | --- | --- | --- |
| **Credit\_Utilization** | **1.000000** | **0.019666** | **-0.021550** | **0.034224** | **0.021493** |
| **Missed\_Payments** | **0.019666** | **1.000000** | **-0.014842** | **-0.026478** | **0.041001** |
| **Credit\_Score** | **-0.021550** | **-0.014842** | **1.000000** | **0.034833** | **0.026707** |
| **Delinquent\_Account** | **0.034224** | **-0.026478** | **0.034833** | **1.000000** | **-0.024332** |
| **Total\_Missed\_Late\_Payments** | **0.021493** | **0.041001** | **0.026707** | **-0.024332** | **1.000000** |

**Average Values for Key Metrics by Delinquent\_Account Status**

| **Delinquent\_Account** | **Credit\_Utilization** | **Missed\_Payments** | **Credit\_Score** | **Total\_Missed\_Late\_Payments** |
| --- | --- | --- | --- | --- |
| **0 (Non-delinquent)** | **0.488505** | **2.990476** | **575.145933** | **4.038095** |
| **1 (Delinquent)** | **0.506887** | **2.850000** | **591.150000** | **3.962500** |

**Analysis of the Code Output**

The code output shows the following:

* **Accuracy:** The model has an accuracy of **88.30%**.
* **Classification Report:**
  + **Precision:** For class 0 (non-delinquent), the precision is 0.88, meaning 88% of the customers predicted as non-delinquent were correctly identified. For class 1 (delinquent), the precision is 0.00, meaning the model failed to correctly identify any of the delinquent cases it predicted.
  + **Recall:** For class 0, the recall is 1.00, meaning the model correctly identified all actual non-delinquent customers. For class 1, the recall is 0.00, meaning the model failed to identify any of the actual delinquent customers.
* **Model Coefficients:** The coefficients for all features are very close to zero, suggesting they have a minimal impact on the model's predictions.

**Model Summary:**

The model used is **Logistic Regression**, a statistical method for binary classification. It calculates the probability of an outcome (delinquency) based on a linear combination of input features. This model is known for its interpretability, as the coefficients show the strength and direction of each feature's influence.

# 2. Justification for Model Choice

**Comparing Models for Geldium's Goals:**

| Model | Performance vs. Explainability | Speed & Scalability | Ease of Monitoring |
| --- | --- | --- | --- |
| **Logistic Regression (Best for Geldium)** | **Excellent explainability** is key for understanding the unexpected data patterns and justifying business decisions. | **Fast and efficient** to train and run, suitable for immediate use and iteration. | **Easy to monitor** because the model's logic and coefficients are simple and transparent. |
| **Decision Trees** | High explainability but can be **less stable** and prone to overfitting, which may not be robust for this dataset. | Fast, but performance can **degrade** with very deep trees. | Easy to monitor, but changes in tree structure can be hard to interpret. |
| **Neural Networks** | Offers high performance but with **low explainability**, making it a "black box" and difficult to justify to stakeholders. | **Computationally intensive** and may be overkill for the dataset's size and complexity. | Hard to monitor, as changes are not easily interpretable. |

A Logistic Regression model is the most suitable choice for Geldium's goals due to its optimal balance of performance and explainability. Given the surprising data anomalies, such as the counter-intuitive relationship between credit score and delinquency, an interpretable model allows the analytics team to understand the 'why' behind each prediction. This transparency is crucial for building trust in the model, identifying and validating true risk factors, and ultimately informing a clear, evidence-based strategy for managing credit risk. While more complex models may offer higher accuracy, their lack of explainability could hinder the ability to address the key data mysteries and justify business decisions.

# 3. Evaluation Strategy

**Evaluation Strategy**

1. **Accuracy:** Measures overall correctness.

Interpretation: A high accuracy can be misleading if the data is imbalanced, as the model might just predict the most frequent outcome.

1. **F1-Score (for Delinquent Class):** Balances precision and recall for the minority class.

Interpretation: A higher F1-score indicates better performance in correctly identifying actual delinquent customers while minimizing false positives.

1. **AUC-ROC:** Assesses the model's ability to distinguish between delinquent and non-delinquent customers.

Interpretation: An AUC closer to 1.0 indicates excellent discriminatory power, while 0.5 is random chance.

1. **Fairness Checks (e.g., Disparate Impact, Equalized Odds):** Evaluate if the model's performance or predictions vary significantly across demographic or protected groups (e.g., by Location, Age, Employment Status).

Interpretation: Significant differences indicate bias, requiring adjustments to ensure equitable outcomes for all customer segments.

Evaluation of the Logistic Regression model's performance on the credit risk prediction task:

* **Accuracy:** **88.30%**. The model correctly predicts the outcome (delinquent or not delinquent) in 88.30% of the cases.
* **Precision (Delinquent):** **0.00**. The model did not correctly predict any of the customers who were actually delinquent.
* **Recall (Delinquent):** **0.00**. The model failed to identify any of the actual delinquent customers.
* **F1-Score (Delinquent):** **0.00**. This score confirms the model's inability to correctly identify delinquent customers.
* **AUC-ROC Score:** **0.4841**. An AUC score of 0.50 indicates a model that performs no better than random chance. A score of 0.4841 is slightly below this, suggesting the model is not effective at distinguishing between delinquent and non-delinquent customers.

The high overall accuracy is misleading because the model is failing to identify the high-risk delinquent customers, likely due to the imbalanced nature of the dataset. This result highlights that while accuracy can be high, other metrics like precision, recall, and AUC-ROC are crucial for a meaningful evaluation, especially for financial risk models.

**Improvements for the Credit Risk/Delinquency Model**

**1. Address Class Imbalance:**

* **Resampling:** Balance the dataset by oversampling the delinquent customers or undersampling the non-delinquent ones.
* **Class Weights:** Assign a higher weight to the delinquent class to make the model more sensitive to these cases.

**2. Improve Model Performance:**

* **More Powerful Models:** Use a more advanced model like Gradient Boosting or Random Forest to capture complex data relationships.
* **Feature Engineering:** Create new features, such as payment history metrics or interaction terms, to enhance the model's predictive power.

**3. Mitigate Bias and Ensure Fairness:**

* **Fairness Algorithms:** Use specialized algorithms that can be trained to meet fairness constraints.
* **Stratified Sampling:** Ensure the proportion of delinquent customers is consistent across training and testing sets.
* **Continuous Monitoring:** Regularly track the model's performance for different customer groups to ensure fairness is maintained over time.

**Ethical Considerations**

* **Bias:** Ensure the model doesn't unfairly penalize certain demographic groups, as this can lead to discriminatory outcomes.
* **Transparency:** Use explainable models and document why a decision was made, building customer trust and ensuring compliance.
* **Data Privacy:** Protect sensitive customer financial data from misuse and unauthorized access.
* **Accountability:** Establish a process for human review and override of model decisions, and take responsibility for the model's impact.
* **Consequences of Errors:** Carefully weigh the impact of false positives (denying credit to a non-risky customer) and false negatives (lending to a high-risk customer) and manage the balance.