**Predictive Model Plan**

# 1. Model Logic (Generated with GenAI)

* Data Ingestion & Splitting: The dataset is loaded and split into a training set (80%) and a test set (20%). A stratify flag is used to ensure that the small percentage of delinquent customers is equally represented in both sets.
* Preprocessing (ColumnTransformer): A single preprocessor is built to handle the data's mixed types:

1. Numerical Features: A SimpleImputer fills missing values (e.g., Income, Credit\_Score) using the "median" strategy, which is robust to outliers. Then, a StandardScaler scales all numerical data to have a mean of 0 and a standard deviation of 1, which is necessary for many models.
2. Categorical Features: A OneHotEncoder converts categorical text data (e.g., Employment\_Status, Location) into numerical columns. handle\_unknown='ignore' is used so the model doesn't crash if it sees new data.

* Imbalance Handling (SMOTE): The model uses an imblearn.Pipeline. After preprocessing, the SMOTE (Synthetic Minority Over-sampling Technique) step is applied only to the training data. It synthetically creates new "delinquent" samples to balance the dataset, giving the model a fair chance to learn their patterns.
* Classification (DecisionTreeClassifier): A DecisionTreeClassifier is used as the core predictive algorithm. This model was chosen for its transparency and its proven ability (in our tests) to handle the noisy, synthetic data from SMOTE better than other models.
* Hyperparameter Tuning (GridSearchCV): The entire pipeline is wrapped in a GridSearchCV. This tool automatically tests 30 different combinations of the Decision Tree's parameters (like max\_depth and min\_samples\_leaf) to find the single "best" version, scoring each one on its roc\_auc score to find the most robust and generalizable model.

# 2. Justification for Model Choice

The tuned Decision Tree was selected after testing against other models (Logistic Regression, Random Forest) because it was the only model that could successfully identify any delinquent customers (i.e., achieve a recall score > 0.00) on the unseen test data.

* Transparency & Suitability: In finance, model transparency is critical for business trust and regulatory compliance. The Decision Tree is a "white-box" model, meaning its logic can be visualized and explained. We can easily see the rules it learned (e.g., "IF Missed\_Payments > 2 AND Credit\_Utilization > 0.8..."). This is far more suitable than a "black-box" model (like a neural network) where the logic is hidden.
* Performance on Imbalanced Data: The primary challenge of this project was the small, highly imbalanced dataset. More complex models like RandomForestClassifier completely failed; they learned to just guess the majority class ("non-delinquent") and achieved 0.00 recall.
* Robustness to Noise: The simple DecisionTreeClassifier, when constrained by GridSearchCV (e.g., max\_depth=5), was more robust to the noisy, synthetic data created by SMOTE. It was forced to learn simple, general rules rather than overfitting to the noise, which is what the Random Forest model did.

While the final model's accuracy is low (due to data limitations), it was the only model framework that proved viable for this specific business problem.

# 3. Evaluation Strategy

Metrics and Interpretation

* Accuracy: This metric will be de-prioritized. In an imbalanced dataset, a model that predicts "0" for everyone can have high accuracy (e.g., 84%) but be completely useless.
* AUC-ROC: This was our primary metric for model tuning (GridSearchCV). It measures the model's overall ability to distinguish between delinquent and non-delinquent customers. A score of 0.5 is a random guess; our goal was to maximize this.
* Recall (Class '1' - Delinquent): This is the most important business metric. It answers: "Of all the customers who actually became delinquent, what percentage did we successfully flag?" A low recall means we are failing to find at-risk customers, defeating the model's purpose.
* Precision (Class '1' - Delinquent): This is a secondary business metric. It answers: "Of all the customers we flagged as delinquent, what percentage actually were?" A low precision means we have many "false positives," which could waste company resources on customers who are not at risk.
* Confusion Matrix: This will be used as the main visual tool to show the trade-off between False Positives (wasted resources) and False Negatives (missed risks).

Bias and Ethical Considerations

* Bias Detection: The model could inadvertently learn biases from the data. For example, if the Location feature was used, the model might unfairly penalize all "Urban" customers. To detect this, we would perform a disparate impact analysis by checking the model's error rates (e.g., False Positive Rate) across different demographic groups.
* Ethical Use: The primary ethical consideration is how the model's predictions are used.
* Proactive Support (Ethical): The model should be used to help customers. A prediction of "high risk" should trigger a proactive, supportive outreach, like offering financial counseling or a modified payment plan.
* Punitive Action (Unethical): The model should not be used to automatically punish customers (e.g., by increasing their interest rates or denying services), as this can create a "death spiral" for those already struggling.
* Transparency (Ethical): Because we chose a Decision Tree, we can (and should) be transparent about the key risk factors. This aligns with regulations that require "adverse action notices" explaining why a customer was denied credit.