Predictive Model Plan

1. Model Logic (Generated with GenAI) using Python

```
import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.impute import SimpleImputer

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import classification_report, confusion_matrix


# Load the data

df = pd.read_csv("csv file.csv")


# Step 1: Select relevant features

features = ['Income', 'Credit_Utilization', 'Missed_Payments']

target = 'Delinquent_Account'


X = df[features]

y = df[target]


# Step 2: Handle missing values (imputation)

imputer = SimpleImputer(strategy='mean')

X_imputed = imputer.fit_transform(X)
```

```
# Step 3: Feature scaling
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_imputed)

# Step 4: Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

# Step 5: Train a classification model
model = RandomForestClassifier(random_state=42)
model.fit(X_train, y_train)

# Step 6: Make predictions and evaluate
y_pred = model.predict(X_test)
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("\nClassification Report:\n", classification_report(y_test, y_pred))
```

Pseudocode Summary

1. Load dataset
2. Select features: Income, Credit Utilization, Missed Payments
3. Impute missing values (mean imputation)
4. Normalize the data
5. Split dataset into training and testing sets
6. Train a classification model (e.g., Random Forest)
7. Evaluate the model using metrics like precision, recall, and accuracy

---

2. Justification for Model Choice

For Geldium's credit risk prediction problem, **Logistic Regression** is an ideal starting model. Here's **why this specific model is selected**, based on key evaluation factors:

1. Accuracy:

While not the most complex model, **logistic regression often performs well** on structured, tabular data like customer financial attributes.

It's especially effective when there's a **linear relationship** between input features (e.g., income, credit utilization) and the probability of delinquency.

2. Transparency:

- Logistic regression offers **clear, interpretable coefficients** that show:
- Direction of effect (positive/negative)
- Magnitude of influence on the outcome
- This aligns with Geldium's likely regulatory requirements in the financial sector, where model decisions must be explainable (e.g., to justify loan rejection).

3. Ease of use or implementation:

- Quick to implement using libraries like scikit-learn
- Requires **minimal preprocessing**, and fits well with standard pipelines (imputation, scaling)
- Easier to validate, monitor, and maintain compared to black-box models

4. Relevance for financial prediction:

- Widely used in the financial services industry for **credit scoring and risk modelling**
- Financial regulators are **familiar with logistic regression**, making it **more acceptable in audits and compliance checks**

5. Suitability for Geldium's business needs:

- Geldium needs a **data-driven, explainable, and scalable** solution to predict credit card delinquency.
- Logistic regression provides a solid, **trustworthy baseline model**, which can be:
    - Deployed quickly
    - Explained to non-technical stakeholders
- Improved later with more complex models (e.g., Random Forest/XGBoost) once trust is built

3. Evaluation Strategy

Here's a detailed plan for evaluating the performance of the credit risk prediction model for Geldium, covering metrics, interpretation, bias detection, and ethical considerations:

1. Performance Metrics to Use

Metric	Why It's Important	Interpretation
Accuracy	Overall correctness	% of total correct predictions (good when classes are balanced)
Precision	Reduces false positives (important for finance)	Of predicted delinquents, how many were actually delinquent
Recall	Catches true positives (avoid missing risky cases)	Of all actual delinquents, how many were detected
F1 Score	Balances precision & recall	Useful when both false positives and negatives are costly
AUC-ROC	Measures ability to separate classes	Closer to 1 = better class separation capability
Confusion Matrix	Shows all outcomes	Helps visualize TP, FP, FN, TN and guide tuning

2. Interpreting the Metrics

For **Geldium's use case (credit risk prediction)**:

- **High Recall** is crucial → we want to **catch all risky customers** (minimize false negatives)
- **Reasonable Precision** is needed → so we don't wrongly label too many good customers as high-risk
- **AUC > 0.80** is considered a strong model in binary classification

3. Bias Detection and Reduction Plan

To ensure the model is fair and inclusive:

- **Check for bias in subgroups**:

    - Use fairness metrics like *equal opportunity difference*, *disparate impact ratio*
    - Analyze model accuracy by segments (e.g., employment status, gender if available, age group)
- **Mitigate bias** via:
    - **Re-sampling**: balance majority/minority classes
    - **Fairness-aware algorithms**: penalize biased decisions
    - **Remove/limit sensitive features** if they create unfair outcomes (e.g., age beyond a threshold)

4. Ethical Considerations

Concern	Approach
Transparency	Use explainable models (e.g., logistic regression, SHAP)
Fair treatment	Avoid discrimination based on age, gender, or location
Consent & privacy	Ensure customer data is collected and used ethically
Impact of false predictions	Set up human-in-the-loop for reviewing high-risk classifications
Accountability	Document decisions, rationale, and allow appeals or reviews