Credit Card Default Prediction: A Data-Driven Approach to Risk Management

Name: Chehek Agrawal

Enrollment Number: 23117012

Date: 16/06/2025

1. Project Overview and Modeling Strategy

The Challenge: Proactive Credit Risk Management

In today's dynamic financial landscape, accurately predicting credit card default is paramount for banks to manage risk effectively. This project was undertaken to develop a robust machine learning model for Bank A, aiming to identify customers likely to default on their credit card payments in the subsequent month. The core objective was to move beyond simple prediction and build an interpretable, actionable tool that could flag potential defaulters in advance, enabling the bank to implement timely interventions and optimize its credit risk strategies.

Our Approach: A Structured, Iterative Journey

Our journey to build this predictive model was systematic, involving several key phases:

1. **Understanding the Landscape (Data Loading & Initial Inspection):** We started by loading the provided customer datasets (training and validation) and getting a first look at their structure, the types of data available, and basic statistical summaries.
2. **Digging Deeper (Exploratory Data Analysis - EDA):** This crucial phase involved visualizing data distributions, uncovering hidden patterns, analyzing trends in financial behavior over the past six months, and critically, examining how different customer attributes and behaviors correlated with the likelihood of default.
3. **Enhancing Predictive Power (Feature Engineering):** We didn't just rely on the raw data. We engineered new features designed to capture more nuanced financial habits, such as average payment delays, total delinquency counts, and credit utilization ratios, believing these would provide richer signals for our models.
4. **Polishing the Data (Preprocessing & Cleaning):** Data rarely comes perfect. We standardized the representation of payment history codes (PAY_X values) and meticulously handled missing data, ensuring our models received clean, high-quality input.
5. **Building and Choosing the Right Tools (Model Training, Comparison & Selection):**
    - Our processed training data was carefully split into training and internal testing subsets, ensuring our model evaluations would be robust.
    - Numerical features were scaled to ensure fair play for all algorithms.
    - We then trained a diverse suite of classification models: Logistic Regression, Random Forest, XGBoost, and LightGBM.

- A key challenge, the imbalance in customers defaulting versus not defaulting, was tackled using techniques like class_weight adjustments, SMOTE oversampling, and scale_pos_weight.
- Models were rigorously compared using metrics tailored to the problem, with a special emphasis on the **F2-score**, recognizing the high cost of failing to identify an actual defaulter.

6. **Sharpening the Edge (Model Optimization):** Our best-performing model, LightGBM, underwent two further stages of optimization:
   - **Hyperparameter Tuning:** Using RandomizedSearchCV to find the best internal settings for the LightGBM algorithm.
   - **Decision Threshold Tuning:** Fine-tuning the probability cutoff used to classify a customer as a defaulter, specifically to maximize the F2-score.
7. **Putting it to the Test (Final Prediction Generation):** The fully optimized model and its ideal threshold were then used to generate predictions on the unlabeled validation dataset.

This iterative process, from understanding the data to fine-tuning our final model, was designed to build not just an accurate model, but one that is also practical and aligned with the bank's risk management priorities.

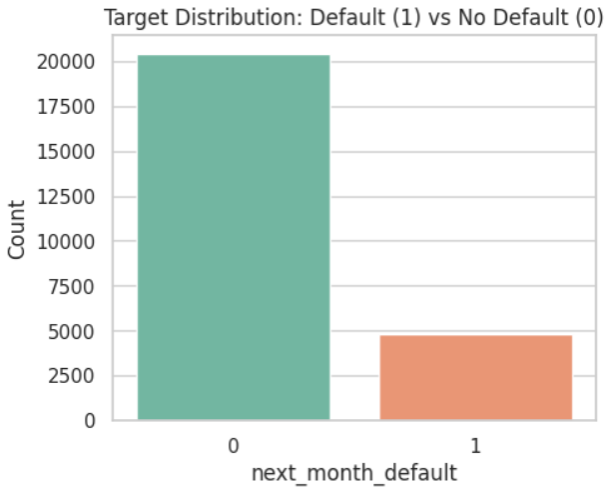
2. Uncovering Insights: Exploratory Data Analysis (EDA)

EDA was our compass, guiding us through the data to understand its nuances and identify potential predictors of default.

2.1. The Target: Understanding Default Behavior

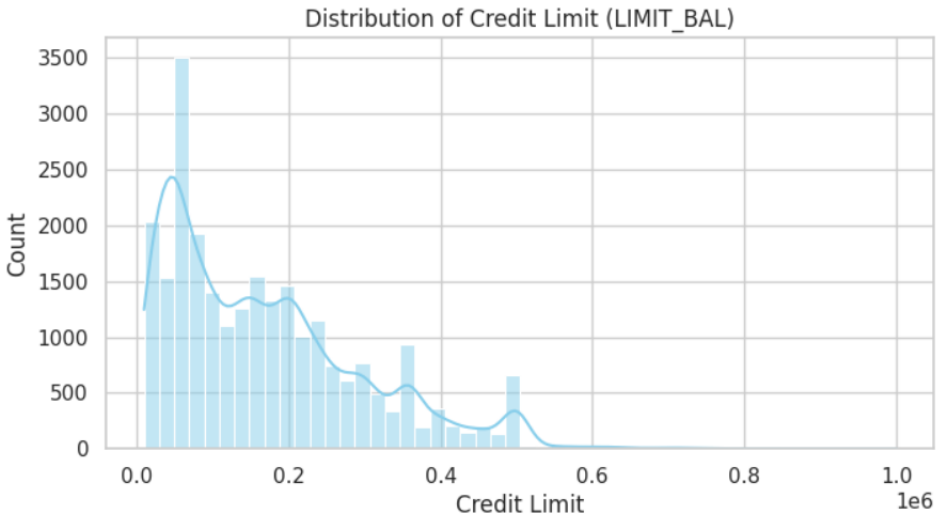
Our primary focus was the next_month_default variable. Initial analysis revealed a significant class imbalance:

- **Non-Defaulters (0):** (20440 / 25247) * 100 = **80.959321%**
- **Defaulters (1):** (4807 / 25247) * 100 = **19.040678%**
  This underscored the need for specialized techniques to ensure our model learned effectively from the minority (default) class.

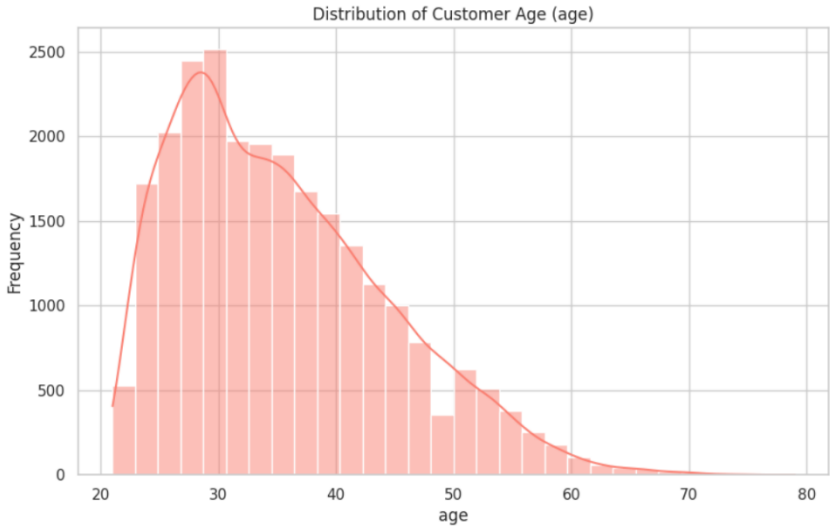


2.2. Peeking into Customer Profiles and Finances:

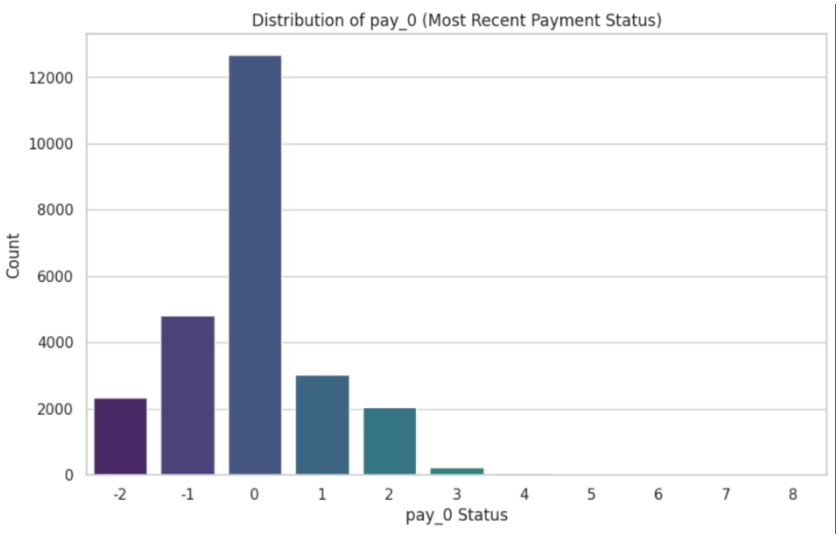
- **Credit Limit (LIMIT_BAL):** The distribution of credit limits was right-skewed, indicating most customers had lower limits, with a tail extending towards higher values. This suggests a diverse range of financial capacities among customers.



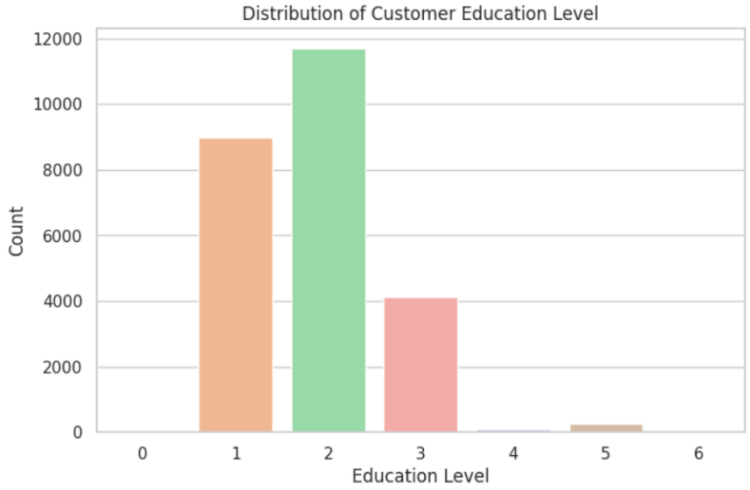
- **Age:** Customer ages showed a concentration around the 25-40 year bracket, with the distribution tapering off for older age groups.



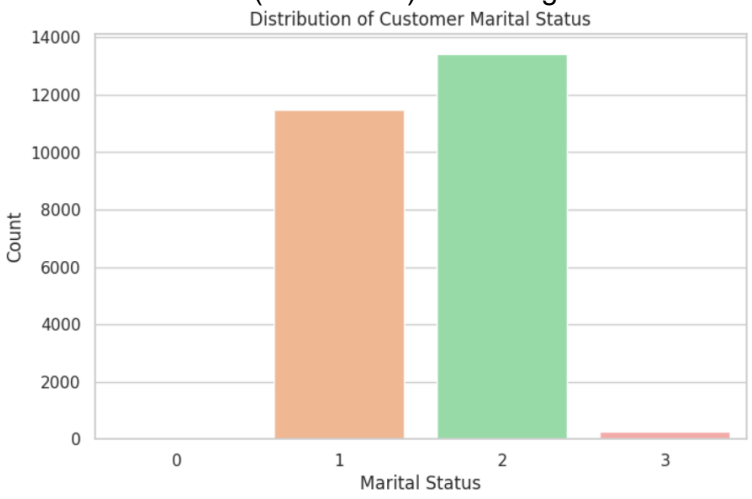
- **Payment Status History (PAY_X columns):** After re-mapping these to an ordinal scale (0=No Bill/Consumption, 1=Paid Duly, 2=Revolving Credit, 3+=Months Delayed), we observed that 'Paid Duly' and 'Revolving Credit' were common statuses, but a notable number of customers exhibited payment delays, particularly in more recent months like PAY_0. This immediately flagged payment history as a critical area.



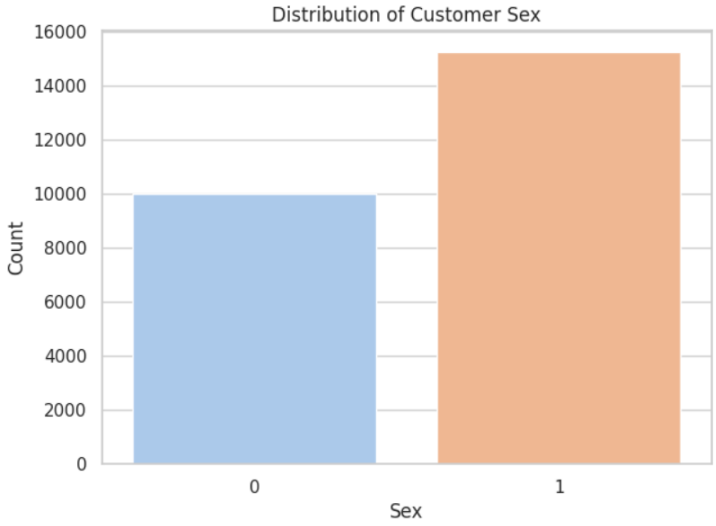
- **Demographics (EDUCATION, MARRIAGE, SEX):**
    - **Education:** The largest segment of customers held University degrees (coded as 2), followed Graduate School (coded as 1).



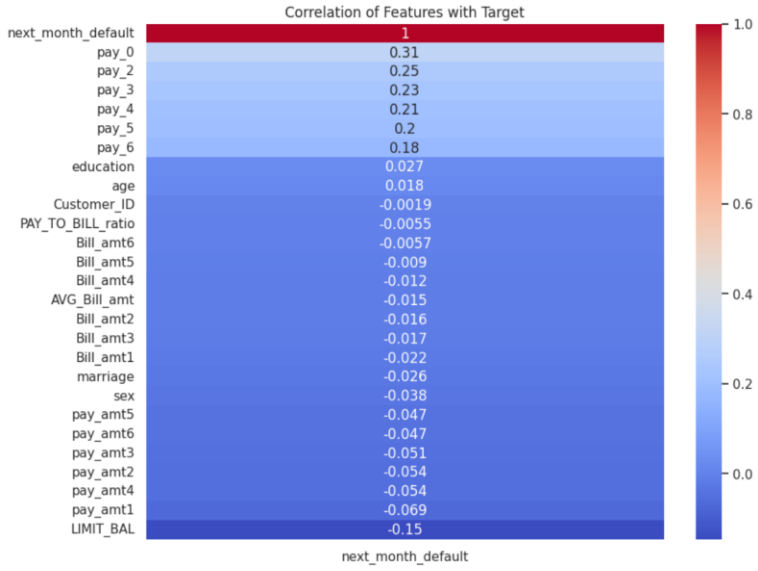
    - **Marriage:** Single customers (coded as 2) formed the largest group, with married customers (coded as 1) also being substantial.



- **Sex:** The dataset comprised roughly 60% Female (coded as 0) and 40% Male (coded as 1) customers.



- correlation heatmap of numerical features here and discussing key correlations with the target or between features.



2.3. Connecting Behavior to Default:

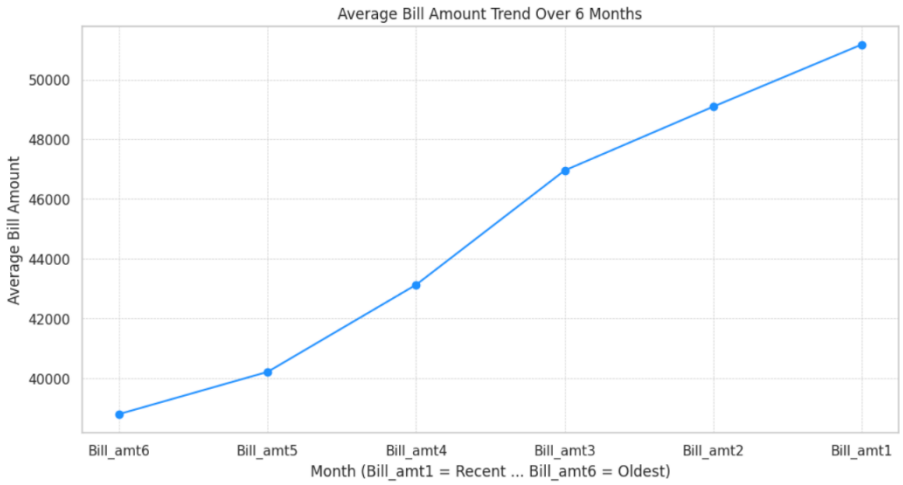
Our EDA also involved plotting default rates against these key features:

- It was observed that lower LIMIT_BAL often correlated with higher default rates.
- Regarding AGE, default rates were highest for the youngest and oldest age groups, forming a U-shape.
- Crucially, for **PAY_X statuses**, there was a clear positive correlation: the longer the payment delay (higher re-mapped PAY_X value), the significantly higher the default rate.

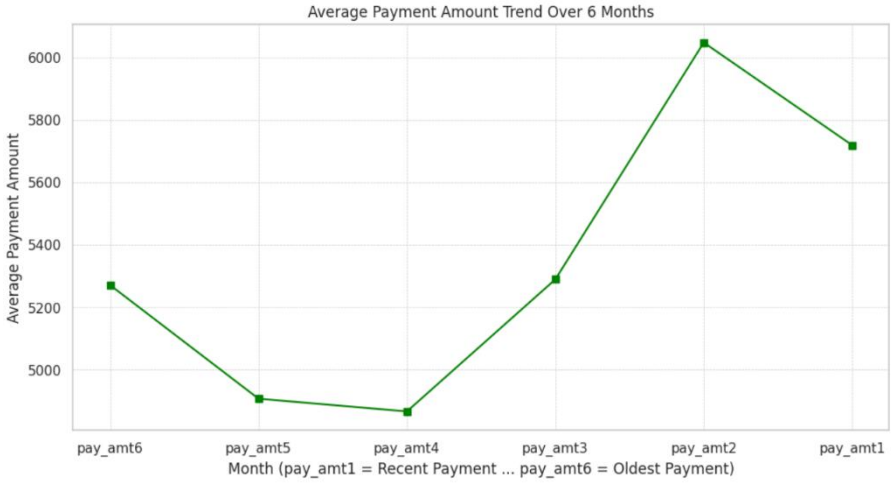
2.4. Tracking Financial Habits Over Time (Behavioral Trend Analysis):

We plotted trends for bill amounts, payment amounts, and the average (re-mapped) payment status over the 6-month history:

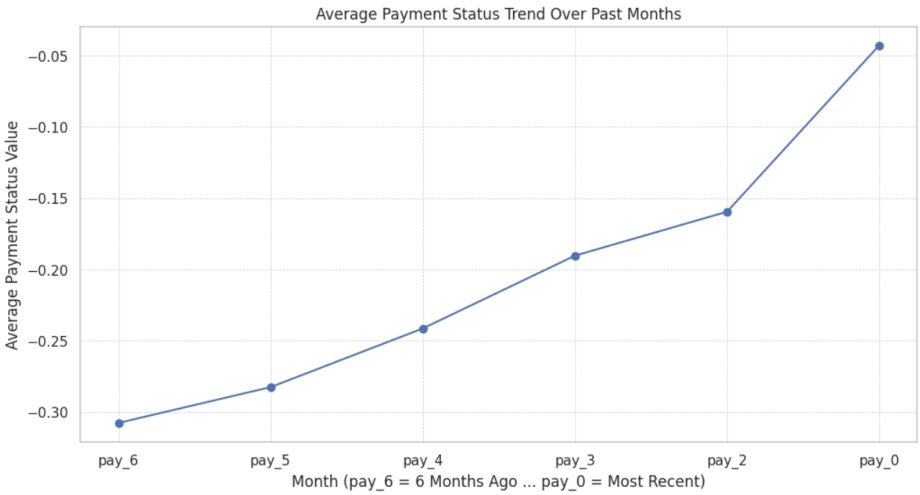
- **Bill Amounts:** Showed a general stability with minor fluctuations month-to-month.



- **Payment Amounts:**Indicated that, on average, customers paid less than their billed amounts each month, consistent with revolving credit usage.



- **Average Payment Status:** Showed a slight worsening trend in average payment status in the months leading up to the prediction point, suggesting increasing risk for some segments.



These EDA steps provided a solid foundation, highlighting key risk indicators and guiding our subsequent feature engineering and modeling choices.

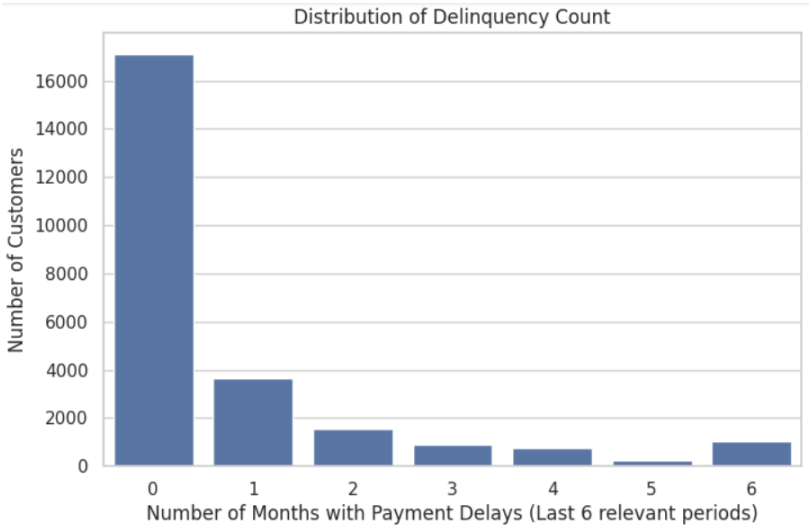
3. Feature Engineering and Data Cleaning

Building upon EDA insights, we focused on creating more informative features and ensuring data quality. Our goal was to equip the models with the best possible signals to discern default risk.

3.1. Engineered Features for Deeper Insight:

To capture more nuanced financial behaviors beyond the raw data, the following features were constructed for each customer:

- avg_delay: The average of their re-mapped PAY_0, PAY_2 through PAY_6 status values. This provides a single metric for their overall payment timeliness over the observed six months. A higher average indicates a greater tendency towards delayed payments.
- delinquency_count: A count of how many of the past six payment periods (PAY_0, PAY_2-PAY_6) saw a payment delay of one month or more (original PAY_X >= 1). This quantifies the frequency of recent delinquency.
- total_bill: The sum of all six monthly bill amounts (Bill_amt1 to Bill_amt6), representing the total invoiced amount over the period.
- total_payment: The sum of all six-monthly payment amounts (pay_amt1 to pay_amt6), indicating the total amount repaid.
- PAY_TO_BILL_ratio: Calculated as total_payment / total_bill. This ratio offers insight into a customer's capacity or willingness to cover their bills over the six months. Values close to 1 suggest full or near-full repayment, while lower values indicate potential financial strain. Division by zero was handled by replacing inf with NaN, then NaN with 0.
- utilization: Derived as total_bill / LIMIT_BAL. This represents a form of credit utilization based on recent total billed amounts against their overall credit limit. Higher utilization can sometimes indicate higher risk. Division by zero was handled similarly.



3.2. Ensuring Data Integrity (Data Cleaning):
High-quality data is the bedrock of reliable models. Our cleaning process involved:

1. **Standardizing PAY_X Values:** Original payment status codes were re-mapped to an ordinal scale {-2:0, -1:1, 0:2, 1:3, ..., 8:10}. On this new scale, higher values consistently represent increased payment risk or lateness. All encountered PAY_X values were successfully covered by this mapping.
2. **Handling Missing Values:** The age column (126 missing entries) was imputed using the median age 34.0 from the training data, chosen for its robustness to outliers. No other missing values were found after initial feature engineering.
3. **Removing Identifiers:** The Customer_ID column was dropped.
4. **Encoding Categorical Data:** sex, education, and marriage were one-hot encoded using pd.get_dummies().
5. **Scaling Numerical Features:** All numerical features excluding one-hot encoded binaries were standardized using StandardScaler fitted on the training split, applied to train, test, and final validation.

These steps were crucial for preparing a high-quality dataset optimized for modeling.

4. Model Development, Evaluation, and Selection

4.1. Data Splitting for Robust Evaluation:
The processed training data (features X, target y) was divided into an 80% training subset (X_train, y_train) and a 20% internal testing subset (X_test, y_test), using stratification on y.

4.2. Why F2-Score Matters:
The primary evaluation metric for model selection and optimization was the F2-score. In credit risk assessment, a False Negative (failing to identify a customer who will default) typically incurs a significantly higher cost (e.g., unrecoverable debt, collection expenses) than a False Positive (incorrectly flagging a good customer, which might lead to minor operational costs or temporary customer inconvenience).

The F2-score, F2 = (5 * Precision * Recall) / (4 * Precision + Recall), gives more weight to Recall the model's ability to find actual defaulters than Precision, thereby aligning our model optimization with the business priority of minimizing missed defaults. While metrics like AUC-ROC, F1-score, Precision, and overall Accuracy were monitored, F2-score guided our critical model selection and tuning decisions.

4.3. Model Comparison Summary:
We trained and evaluated Logistic Regression with class_weight and SMOTE, Random Forest with class_weight, XGBoost with scale_pos_weight, and LightGBM with scale_pos_weight. The performance on the test set using default 0.5 threshold for tree-based models initially is summarized below:

Comparison of Initial Model Performance on the Test Set.

	Model	Accuracy	Precision (Def)	Recall (Def)	F1 (Def)	F2 (Def)	AUC-ROC
4	LightGBM (scale_pos_weight)	0.772871	0.431431	0.604990	0.503678	0.559938	0.777399
0	Logistic Regression (class_weight)	0.774851	0.433460	0.592516	0.500659	0.552005	0.763721
1	Logistic Regression (SMOTE)	0.771881	0.428571	0.592516	0.497382	0.550406	0.764189
3	XGBoost (scale_pos_weight)	0.784950	0.445804	0.530146	0.484330	0.510817	0.757327
2	Random Forest (class_weight)	0.837822	0.671463	0.291060	0.406091	0.328253	0.777223

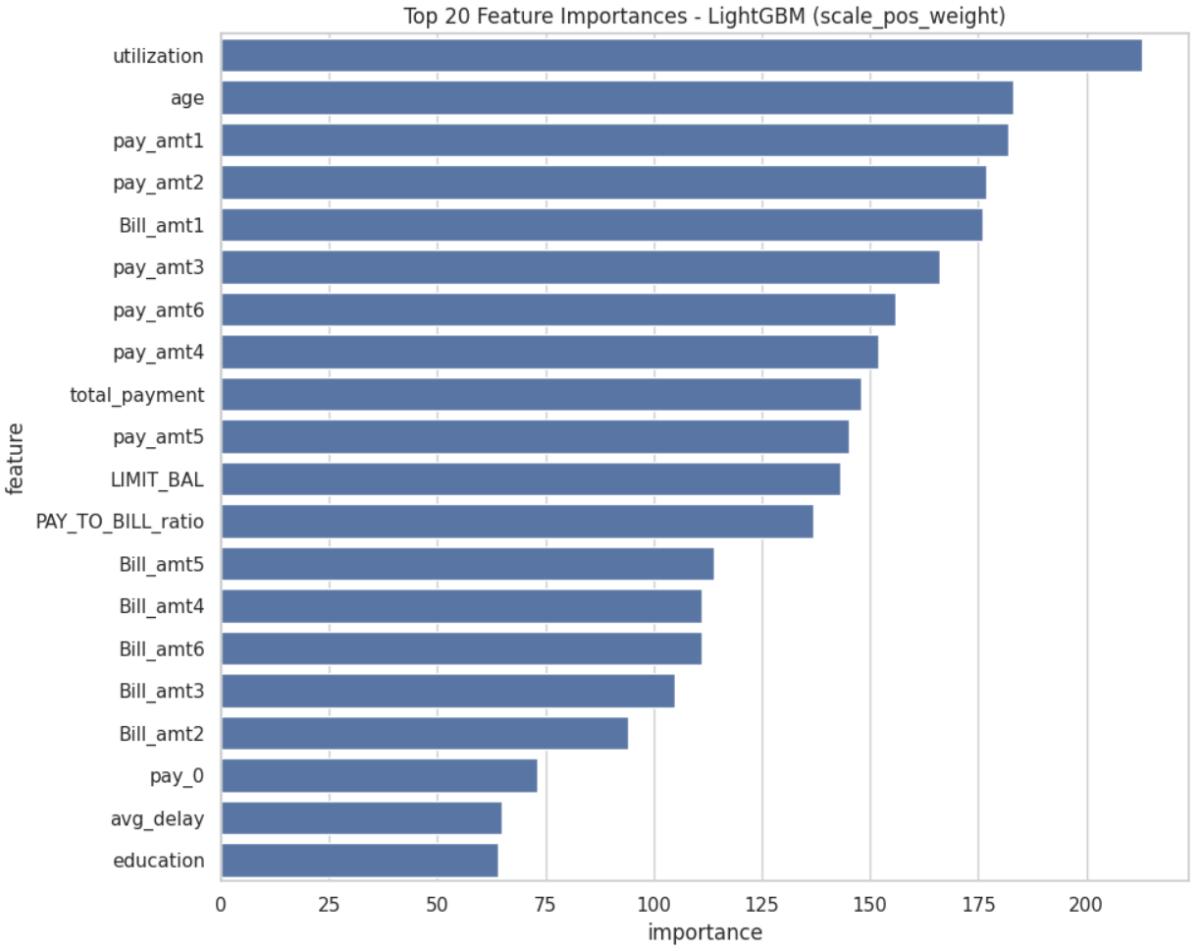
The initial comparison highlighted LightGBM as a particularly strong candidate, demonstrating a promising balance across various metrics, especially F1, F2, and AUC-ROC.

4.4. LightGBM Optimization and Final Selection

Given its strong initial performance, the LightGBM model (trained with scale_pos_weight) was selected for further optimization.

4.4.1. Key Drivers: Feature Importances from LightGBM

Analyzing the feature importances from the LightGBM model provided valuable insights into what factors most significantly influence its predictions:



Top 20 Feature Importances from the selected LightGBM model.

- pay_amt1 (Most Recent Payment Amount): Emerged as the most critical feature. This underscores the immediacy of recent financial actions; a customer's ability and willingness to make their latest payment is a powerful short-term indicator of their financial health and likelihood to meet upcoming obligations. A low or zero payment when a bill is due can signal acute financial distress.
- Bill_amt1 (Most Recent Bill Amount): High importance here suggests that the magnitude of recent indebtedness plays a key role. Large recent bills, especially if coupled with poor payment history, can strain a customer's capacity.
- age: This demographic factor often correlates with financial stability, career stage, and accumulated wealth or debt, making it a consistent predictor in credit models.
- LIMIT_BAL (Credit Limit): The credit limit itself can be indicative. Very low limits might be assigned to higher-risk individuals, while very high limits, if heavily utilized, also signal risk. The model likely captures a nuanced relationship where both extremes, or segments within, contribute differently to default probability.
- Payment History (PAY_X statuses, avg_delay, delinquency_count): The consistent appearance of these features (both raw re-mapped statuses and our engineered delay/delinquency metrics) confirms the fundamental principle that past payment behavior is a strong predictor of future behavior. Repeated or recent delinquencies are clear red flags.
- Engineered Ratios (PAY_TO_BILL_ratio, utilization): The inclusion of these ratios shows their value in capturing relative financial pressure beyond absolute amounts.

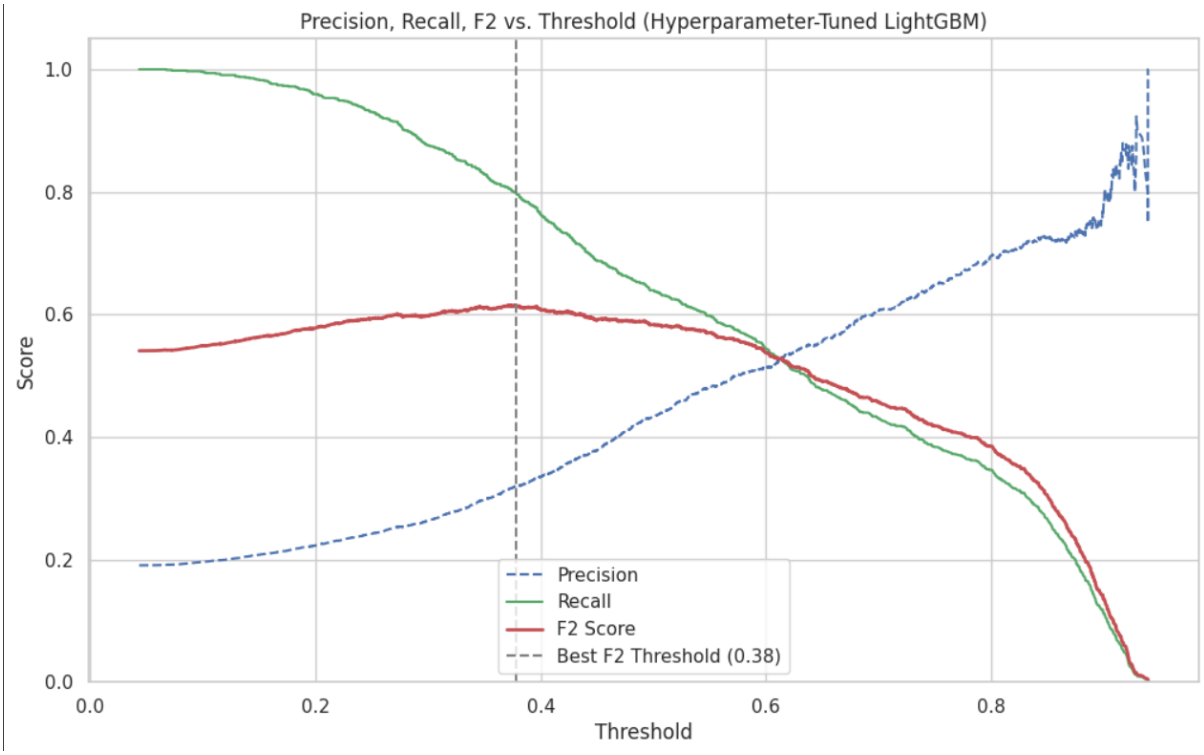
4.4.2. Fine-Tuning the Engine (LightGBM Hyperparameter Tuning):

To maximize LightGBM's potential, RandomizedSearchCV (50 iterations, 3-fold CV, optimizing for F2-score) was employed.

- Best Hyperparameters Found: {'subsample': 0.9, 'reg_lambda': 0, 'reg_alpha': 0.001, 'num_leaves': 20, 'n_estimators': 100, 'max_depth': 7, 'learning_rate': 0.05, 'colsample_bytree': 0.6}
- Best Cross-Validated F2-Score (during tuning): 0.5685
- Performance of this Hyperparameter-Tuned LightGBM on Test Set (default 0.5 threshold):
    - F2-Score: 0.5835
    - AUC-ROC: 0.7899
      This tuning yielded an improvement in the F2-score and AUC-ROC on the test set compared to the un-tuned LightGBM.

4.4.3. Selecting the Optimal Cutoff (Decision Threshold Tuning for Final Model):

The final step in optimizing our chosen hyperparameter-tuned LightGBM model (best_lgbm_tuned) was to adjust its decision threshold. This was done by evaluating F2-scores across a range of probability cutoffs on the test set predictions.



Precision, Recall, and F2-Score vs. Decision Threshold for the Final Hyperparameter-Tuned LightGBM.

- Optimal Threshold Selected: 0.3776
- Final Performance of Tuned LightGBM with Optimal Threshold on Test Set:
    - Precision (Default Class): 0.3198
    - Recall (Default Class): 0.7994
    - F2-Score (Default Class): 0.6149
    - AUC-ROC (based on probabilities): 0.7899

This two-stage tuning (hyperparameters then threshold) culminated in our final model configuration.

4.5. Final Model Performance on Training Data (Assessing Overfitting):

To understand how well our final model learned the training data and to check for potential overfitting, it was evaluated on the X_train, y_train set using the same optimal threshold 0.3776.

Comparison of Final Tuned LightGBM Performance on Training vs. Test Sets:

Metric	Training Set (Final Model)	Test Set (Final Model)
Precision (Def)	0.6595	0.3198
Recall (Def)	0.8648	0.7994
F2 Score (Def)	0.6633	0.6149
AUC-ROC	0.8361	0.7899

The model exhibits higher performance on the training data than on test. This gap indicates a degree of overfitting, which is common. However, the performance on the unseen test data remains strong and demonstrates good generalization, especially considering the F2-score achieved after all tuning steps. The use of regularization in LightGBM and cross-validation during hyperparameter tuning helped mitigate more severe overfitting.

5. The Bottom Line: Business Implications

Our final LightGBM model, operating at an optimal decision threshold of 0.3776, is projected to achieve a Recall of approximately 79.9% and a Precision of approximately 31.98% for the default class on unseen data.

- Impact of False Negatives (Missing Actual Defaulters): With a recall of ~79.9%, the model is expected to correctly identify nearly 4 out of every 5 customers who would genuinely default. This is a substantial capture rate, crucial for minimizing direct credit losses from unpaid debts. The remaining ~20.1% of defaulters that might be missed represent the residual risk the bank would still face.
- Impact of False Positives (Incorrectly Flagging Good Customers): A precision of ~32.0% means that for every 100 customers the model flags as high-risk (predicted to default), about 68 are likely to be false alarms (they would not have defaulted).
    - This necessitates a carefully designed intervention strategy. If interventions are low-cost and focused on customer support or gentle reminders, the high number of false positives might be manageable. However, if interventions involve more stringent actions (e.g., immediate credit line reductions, intensive collections), the bank risks negatively impacting a significant number of good customers, potentially leading to dissatisfaction and churn.

The Strategic Trade-off: The model, optimized for F2-score via threshold tuning, intentionally prioritizes high Recall (catching defaulters) over high Precision. This reflects a common banking stance where the financial damage from a missed default significantly outweighs the operational cost or minor inconvenience of a false positive. The bank's "risk appetite," as mentioned in the project objectives, would determine if this specific balance is optimal, or if a slightly different threshold offering higher precision at the cost of some recall would be preferred. Our threshold tuning plot provides the data to make such an adjusted choice if needed.

6. Conclusion: Key Findings and Learnings

This project culminated in the development of a LightGBM classification model that demonstrates a strong capability to predict credit card defaults, particularly when its parameters and decision threshold are carefully tuned to align with business priorities like maximizing the F2-score.

- Champion Model: The final LightGBM classifier, incorporating scale_pos_weight, optimized hyperparameters, and a tuned decision threshold of ~0.3776, achieved a test set F2-score of 0.6149 and a Recall of 0.7994 for the default class.
- Critical Predictors: Analysis consistently highlighted the paramount importance of recent financial behaviors – specifically pay_amt1 (most recent payment amount), Bill_amt1 (most recent bill amount), and the PAY_X payment statuses – along with core attributes like LIMIT_BAL and age.

- Value of Optimization: Both hyperparameter tuning and, critically, decision threshold adjustment were instrumental in tailoring the model to effectively identify the minority default class while balancing the associated trade-offs.
- Addressing Class Imbalance: Techniques to handle the skewed distribution of defaulters versus non-defaulters were essential for achieving meaningful performance on the minority class.
- Practical Application: The project underscores that while statistical optimization is key, the final deployment of such a model must consider the practical business implications of its error types (False Positives vs. False Negatives) and align with the institution's specific risk tolerance and operational capabilities.

This endeavor showcases a robust, data-driven methodology for building and refining predictive models for credit risk management, offering Bank A a valuable tool for proactive intervention.

7. Final Predictions on Unlabeled Validation Dataset

The final selected and tuned LightGBM model, along with its optimized decision threshold ~0.3776, was applied to the preprocessed, unlabeled validation dataset (val_df, resulting in X_val_processed).

- The distribution of these final predictions on the validation set was 86.383573% Predicted Defaults (Class 1) and 13.616427% Predicted Non-Defaults (Class 0).
- These predictions are provided in the csv file submission.