Finance Club

Open Project Summer 2025

Title: Credit Card Behaviour Score Prediction Using Classification and Risk-Based Techniques

A brief summary of the entire pipeline:

I. **Data loading & cleaning**

II. **EDA and insights**

III. **Feature engineering**

IV. **Handling imbalance**

V. **Model training & evaluation**

VI. **Threshold tuning & predictions**

VII. **Business interpretation**

Objective:

The main goal of this project was to create a reliable model that can predict whether a customer is likely to default on their credit card payment in the next month. Since this is a classification problem, I followed a structured approach:

first understanding and cleaning the data, then exploring patterns through visualizations, creating new features, building different models, and finally evaluating which one performed best.

Step 1: Data Loading, Preparation & Initial Exploration

To begin the project, I imported all the essential Python libraries required for data analysis, modeling, and visualization. This included standard packages like pandas, numpy, matplotlib, and seaborn, as well as machine learning tools from scikit-learn, xgboost, lightgbm, and imblearn for handling imbalanced data. and used warnings.filterwarnings("ignore") to keep the output clean and readable.

Step 1.1 :Column Fixes and Cleaning

One of the early cleanup steps was renaming a mislabeled column pay_0 to pay_1 in both datasets. This was likely done to maintain consistency, since the payment history features range from pay_1 to pay_6.

I also cleaned the categorical features education and marriage, which had unexpected or undefined values:

- In the education column, values like 0, 5, 6, and 9 were replaced with a general category (4) representing “others.”
- In the marriage column, values like 0 and 9 were replaced with 3, also to indicate “others.”

Step 1.2: Handling Missing Values

To make sure missing data wouldn’t negatively impact the model, I defined a function called clean dataset() that takes care of filling in missing values for numeric columns. I used median imputation because the median is more robust than the mean, especially when there are outliers which is often the case in financial data.

Here's what the function was does:

- It selects all numeric columns from the DataFrame.
- It uses SimpleImputer from sklearn to fill in missing values with the column median.
- It then replaces the original data with the cleaned version.

I applied this function to both my training and validation datasets so everything was consistent and ready for analysis.

Grouping Monthly Features

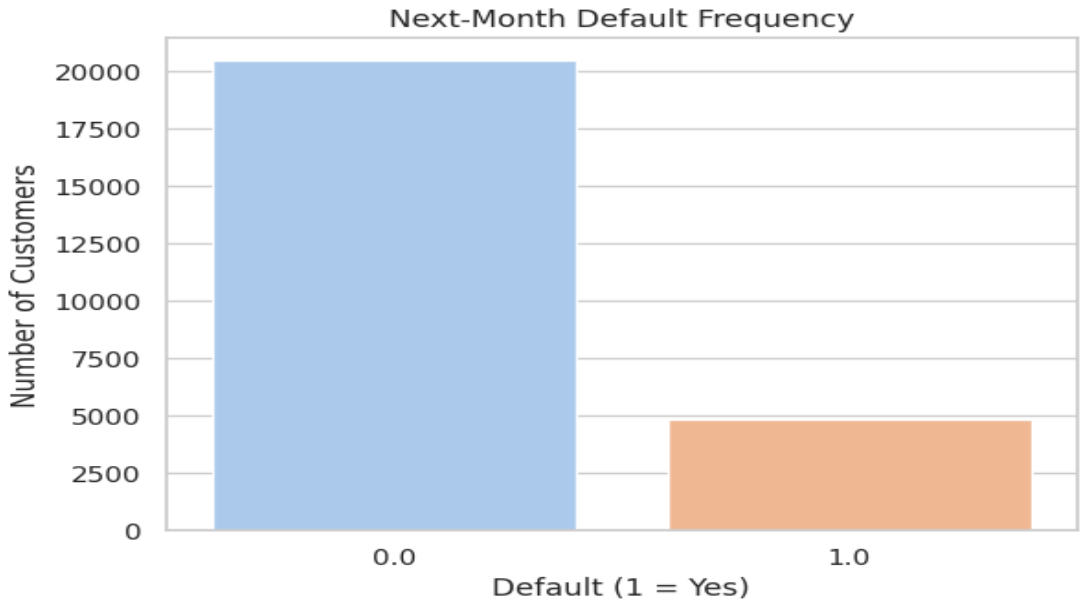
To make my analysis easier and cleaner, I grouped the monthly columns into three separate lists:

- pay_cols: These columns (pay_1 to pay_6) represent the repayment status for each of the last six months. They show how late the customer was in paying.
- pay_amt_cols: These columns (pay_amt1 to pay_amt6) show how much the customer actually paid each month.
- bill_cols: These (Bill_amt1 to Bill_amt6) represent the bill amount for each of those months.

Step 2: Exploratory Data Analysis (EDA) — Default Class Balance

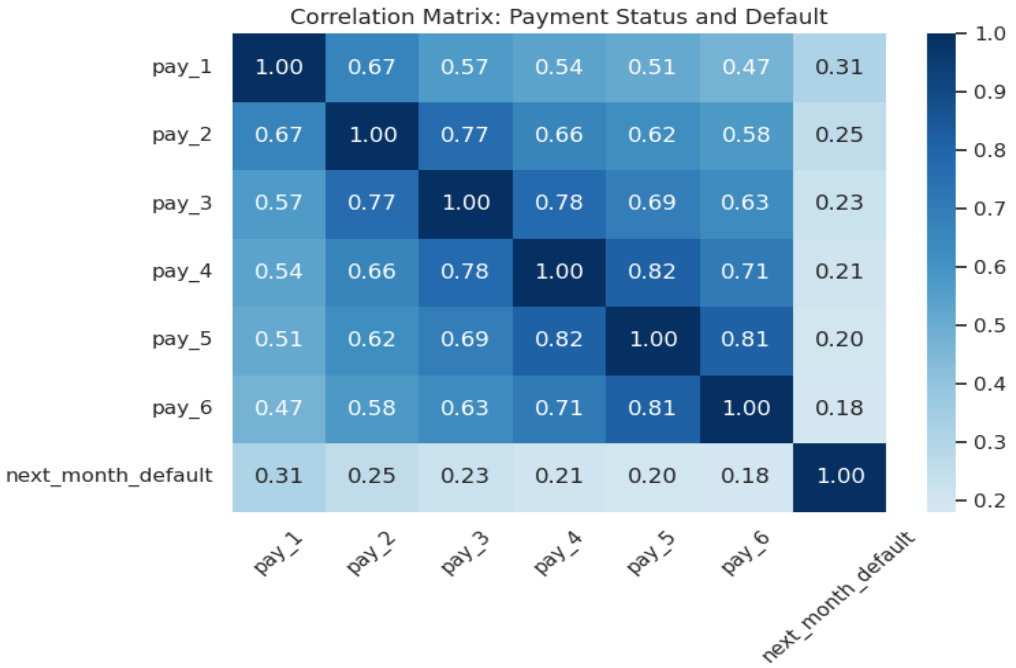
Step 2.1: Default Class Distribution

- To start, I looked at how many customers defaulted on their credit card payments compared to those who didn't. I used a bar chart to show the count of customers who did not default (0) and those who did default (1). Then, I calculated the default rate, which tells us what percentage of customers actually defaulted. It came out to about 22%.
- The number of defaulting customers is much smaller than the number of non-defaulting customers. This is called a **class imbalance**. If we don't handle this properly, the model might just predict "no default" for everyone and still seem accurate but it won't really be useful.
- To fix this, I decided to balance the data later using a method called **SMOTE**, which helps the model learn from both types of customers more fairly.



Step 2.2: Correlation Between Payment History and Default

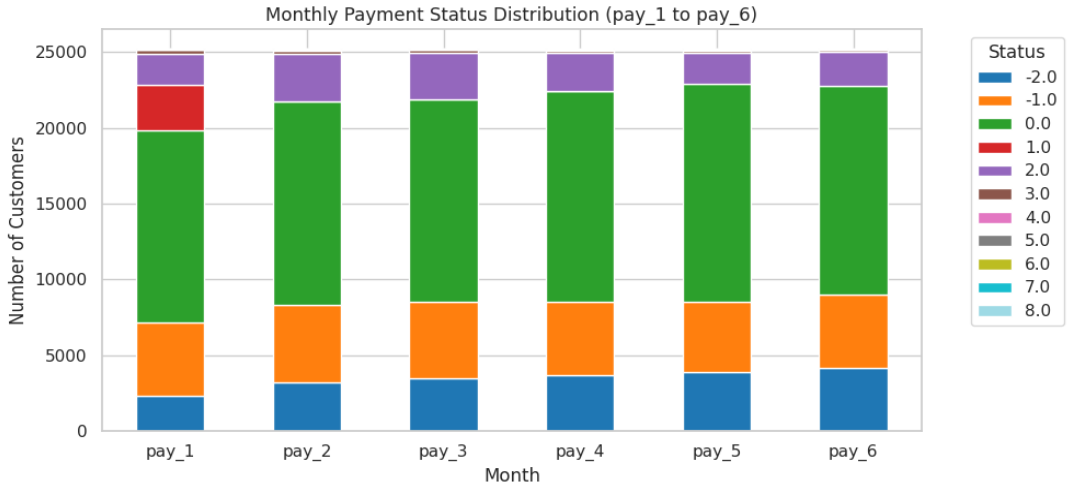
- In this step, I wanted to explore how a customer's past payment behavior is related to the chance of defaulting next month. I focused on six columns: pay_1 to pay_6. Each of these shows whether a customer delayed their payments in the past six months (with positive numbers indicating delays).
- To check the relationship, I created a correlation matrix that shows how strongly each pay_x column is related to the target variable next month default.



- The matrix shows that all six pay_x features have a positive correlation with defaulting.
- This means that as payment delays increase, the chance of defaulting next month also increases.
- pay_1 (the most recent month) usually has the strongest correlation, which makes sense — recent behavior is often the most telling.

Step 2.3: Monthly Payment Status Distribution

- This stacked bar chart shows how customers paid their dues over the past 6 months (pay_1 to pay_6). Most payments were made on time or early (0 or -1), but a noticeable number of customers had delays. The trend remains consistent across months, helping us identify repayment risk patterns.

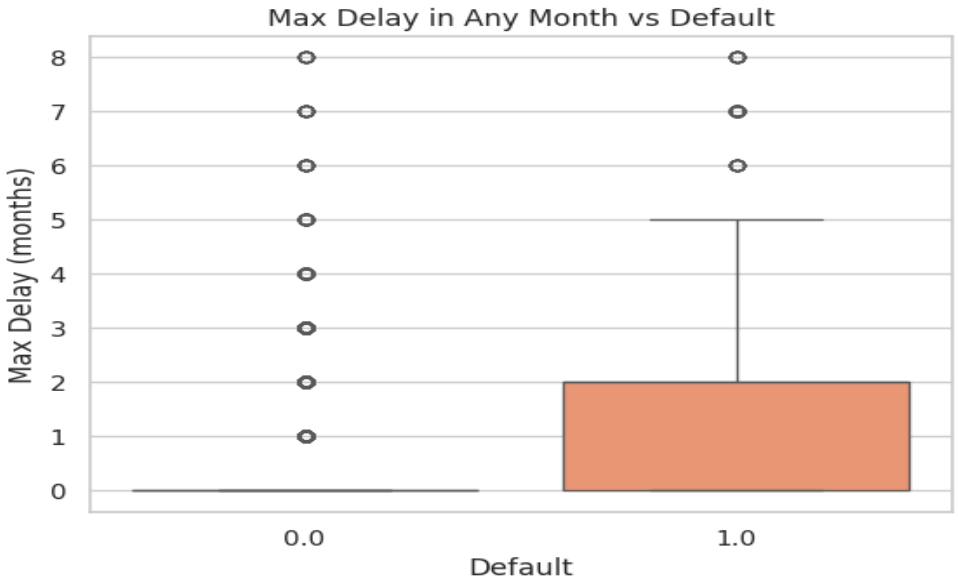


Step 2.4: Default Rate by Payment Status

- We analyzed how the default rate changes with each payment status value (pay_1 to pay_6). As expected, the default rate increases with delayed payments. Customers with higher delay values show a significantly higher chance of defaulting next month.

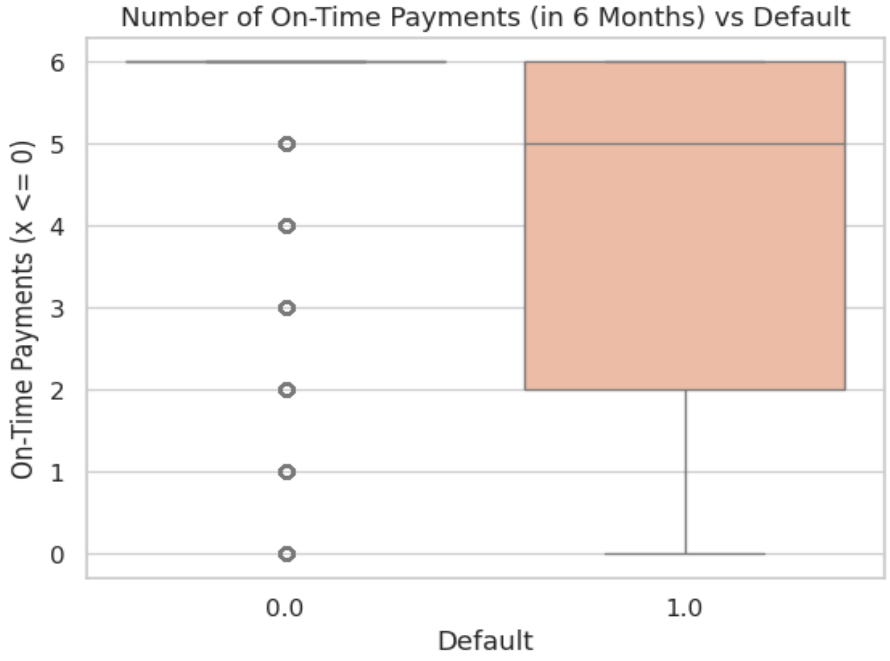
Step 2.5: Maximum Delay vs. Default

- We created a feature showing the **maximum delay** across any month (ignoring early payments). The boxplot reveals that defaulters generally have higher max delays, highlighting delayed repayment as a strong signal of risk.



Step 2.6: On-Time Payment Count vs. Default

- We counted how many times each customer paid on time or early in the last 6 months. The boxplot shows that customers who defaulted had **fewer on-time payments**, reinforcing that payment discipline is key to predicting default.

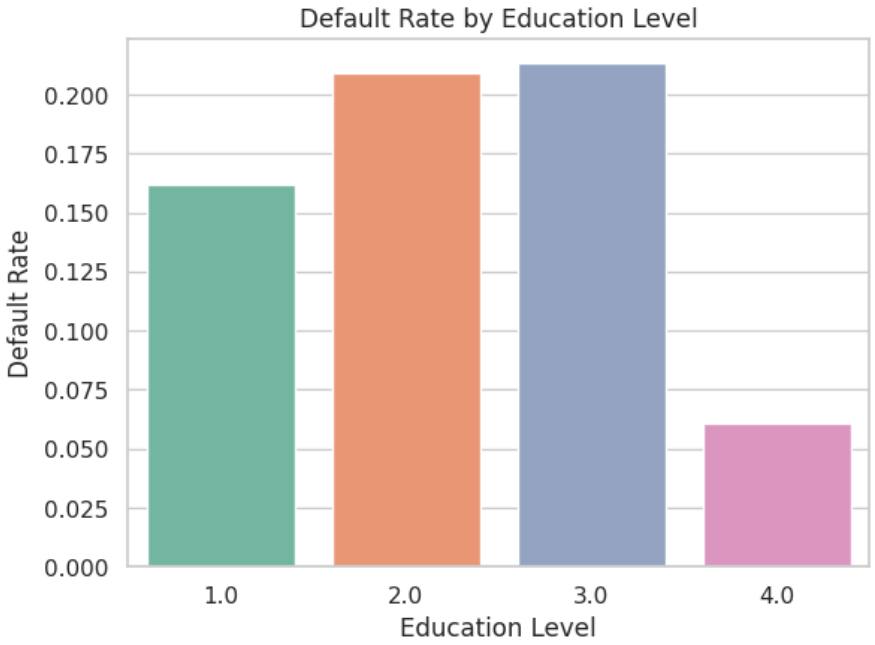


Step 2.7: Utilization Ratio vs. Default

- We calculated the credit utilization ratio using Bill_amt6 / LIMIT_BAL. Defaulters tend to have **higher utilization**, indicating that customers using a large portion of their credit limit are more likely to default.

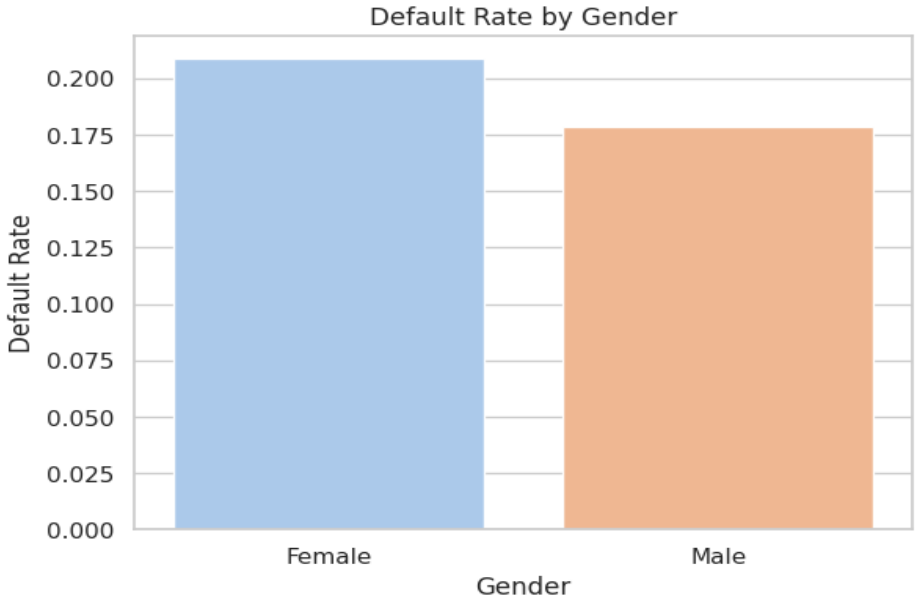
Step 2.8: Default Rate by Education Level

- Default rates vary across education levels. Some groups show a **slightly higher tendency to default**, suggesting education may have a moderate impact on credit behavior.



Step 2.9: Default Rate by Gender

- Male customers show a **slightly higher default rate** than females, indicating a potential link between gender and repayment behavior.

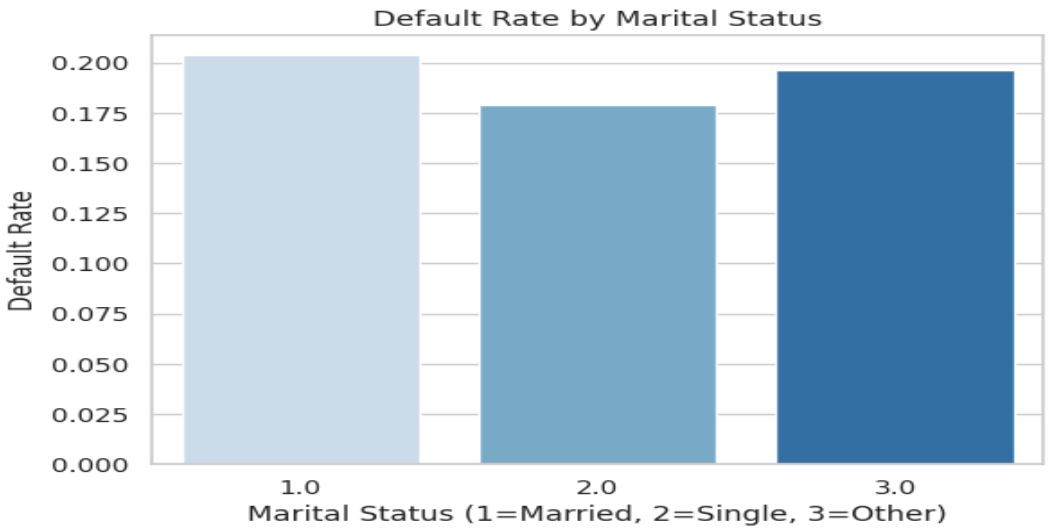


Step 2.10: Default Rate by Age Group

- Default rates vary across age groups. **Younger customers (under 30)** tend to have slightly **higher default rates**, while middle-aged groups show more stability. This insight can help in tailoring credit policies by age segment.

Step 2.11: Default Rate by Marital Status

- We compared default rates across marital statuses. **Single customers** (label 2) show a slightly **higher risk of default** compared to married individuals, suggesting personal financial stability may vary with marital status.



Step 3: Feature Engineering — Risk Indicators

We created new features to capture customer credit behavior more meaningfully:

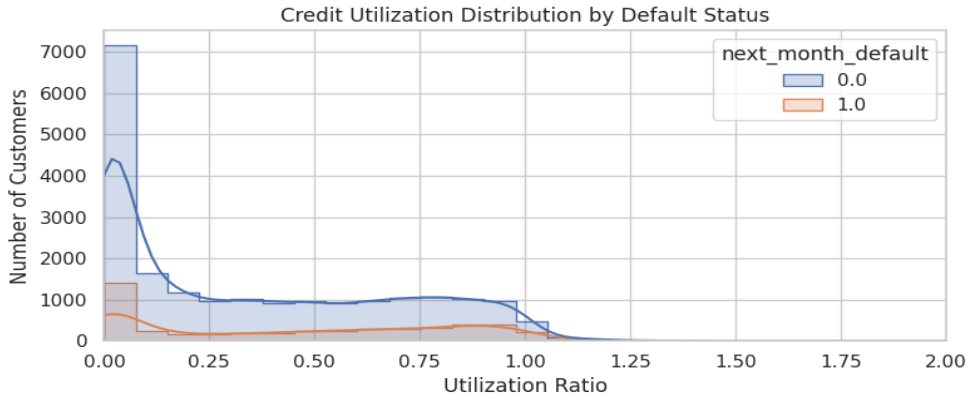
- **Credit Utilization**: Average bill amount relative to credit limit.
- **Repayment Ratio**: How much of the bill was repaid on average.
- **Max Delinquency**: The worst payment delay recorded in any month.
- **Ever Delinquent**: A binary flag indicating if the customer was ever late.

Step 3.1-Payment-to-Bill Ratio

- We created a new feature pay_to_bill_ratio to measure how much a customer pays relative to their billed amount over 6 months
- This helps capture repayment behavior:
- Lower values suggest underpayment, common among defaulters.
- Higher values indicate responsible repayment.
- To handle skewness, we also added a log-transformed feature log_pay_to_bill_ratio.
- This feature adds financial meaning and helps improve model performance.

Step 3.2-Credit Utilization Ratio

- We created the feature CREDIT_UTIL_RATIO to measure the average bill amount over the credit limit:
- A higher ratio means the customer is using more of their credit limit**,** which can indicate financial stress**.**
- A lower ratio suggests conservative credit use, usually seen in low-risk customers.
- **Defaulters** are more concentrated at higher utilization levels**.**
- **Non-defaulters** tend to have lower or moderate utilization.



Step 3.3-Delinquency Count

- The DELINQUENCY_COUNT feature counts how many months (out of 6) a customer had delayed payments.
- A higher count means more frequent past delays, indicating risky repayment behavior.
- A count of 0 suggests timely payments, typical of low-risk customers.
- **Defaulters** are more common at **higher delinquency counts.**
- **Non-defaulters** cluster around **0 or 1** late payment months.

Step 3.4-Maximum Payment Delay

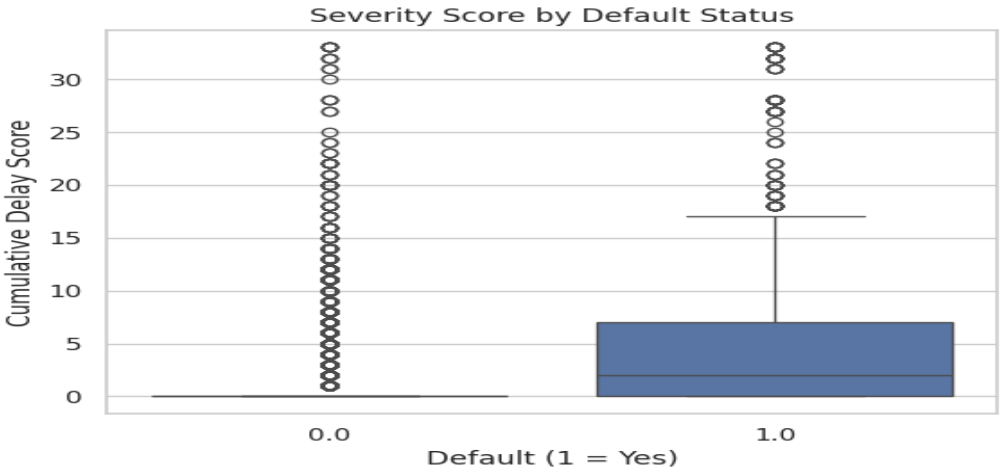
- We created MAX_DELAY to capture the worst payment delay a customer had in the past 6 months:
- Higher values indicate more severe delinquency, such as multiple months overdue.
- A value of 0 or less suggests **on-time or early payments**.
- **Defaulters** typically have **higher max delays**.
- **Non-defaulters** show lower max delay values with fewer outliers.

Step 3.5- Severity Score (Total Delay Severity)

- The severity_score feature sums up all delayed payments over the past 6 months, counting only positive delays:
- Delays are clipped at 0 to ignore early or on-time payments
- A higher score means more months and longer durations of overdue payments.

Boxplot Insight:

- **Defaulters** show significantly **higher severity scores**.
- **Non-defaulters** mostly have low or zero scores.



Step 3.6-Credit Limit Segmentation

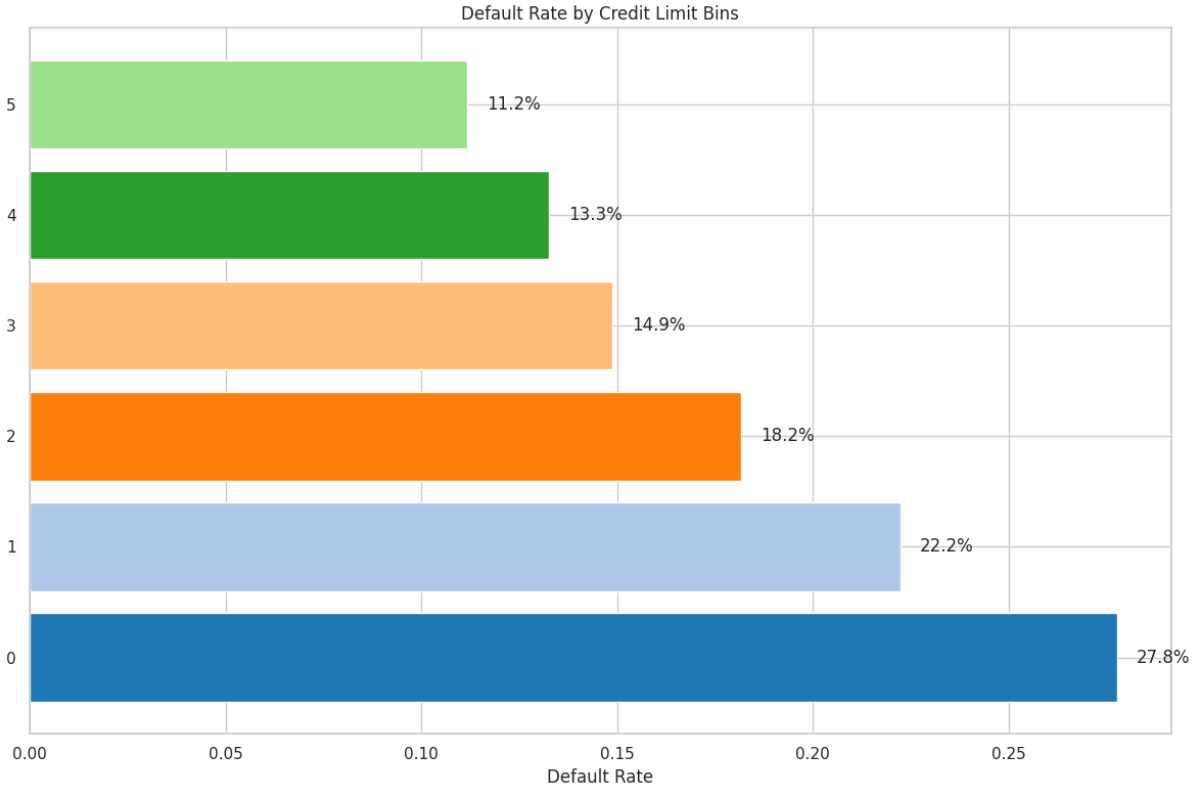
- We created a new categorical feature LIMIT_BIN_LABEL by segmenting customers based on their credit limit (LIMIT_BAL)
- Helps in understanding **default patterns across credit tiers**.
- Supports group-wise EDA and targeted **risk profiling**.
- To prepare the LIMIT_BIN_LABEL feature for modeling, we converted the categorical labels into numerical codes using
- Each credit tier (e.g. very_low, low, ….) is assigned an integer from 0 to 5.
- This encoding retains the ordinal relationship between segments.

Step 3.7-Default Rate by Credit Limit

- We analyzed how default rates vary across credit limit segments (LIMIT_BIN_LABEL) by calculating the average default rate per bin.
- A horizontal bar plot shows default rates for each credit limit tier.
- Each bar is labeled with the exact percentage.

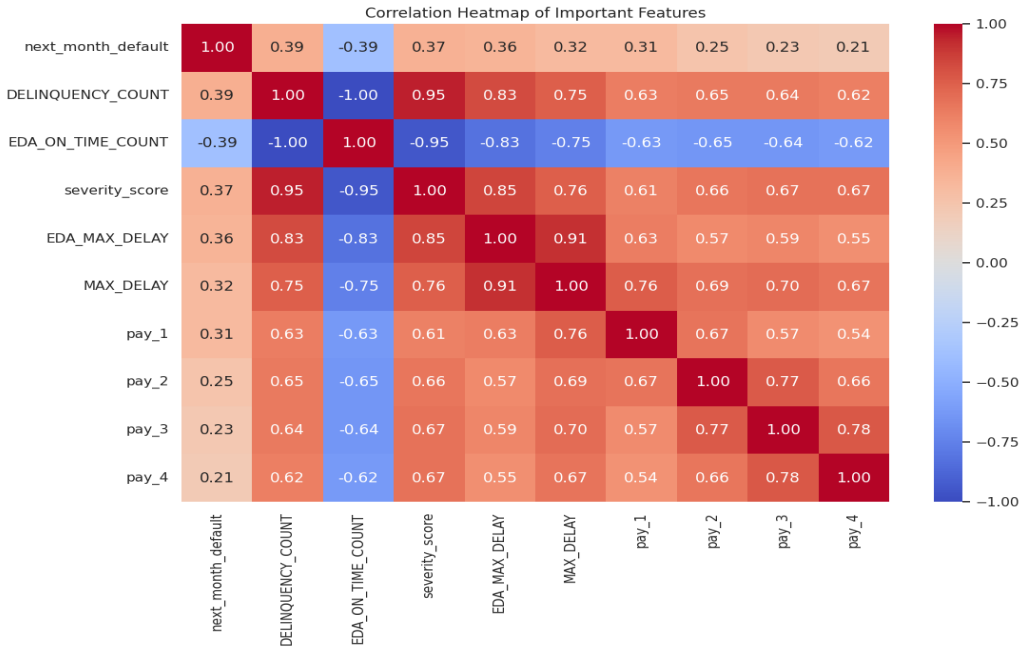
Key Observations

- Lower credit limit segments (e.g.,very_low ,low) tend to have higher default rates.
- Higher limit groups (e.g.,high, very_high) show lower risk, indicating stronger financial stability.



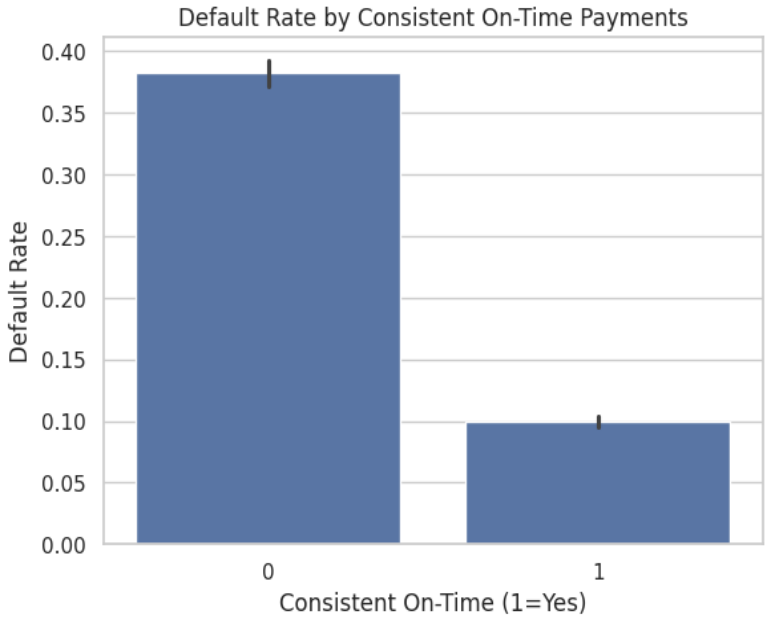
Step 3.8-Correlation Heatmap (Important Features)

- We calculated the correlation matrix for all numeric features and selected those with a strong relationship
- Displays **inter-feature correlations** among the most important variables.



Step 3.9-Consistent On-Time Payments

- We created the CONSISTENT_ON_TIME feature to identify customers who made on-time or early payments for all 6 months
- A value of 1 means **no late payments** across 6 months.
- A value of 0 indicates **at least one delayed payment**.



- Customers with consistently on-time payments have a **significantly lower default rate**.
- Those without consistent behavior show **higher risk**.

Step 3.10-Total Payment Amount (6 Months)

- We created the feature TOTAL_PAY_AMT to represent the total amount paid by a customer over the past 6 months:

Step 4.1- Splitting Features and Target

<u>Separating features and target:</u>

- X: All input features (excluding Customer_ID and next_month_default)
- y: Binary label indicating whether the customer will default next month.
- Splitting into training and validation sets using train_test_split:

- 80% training**,** 20% validation
- Used stratify=y to maintain the original class distribution
- Random state = 42 ensures reproducibility
- To maintain consistency and avoid model compatibility issues, we converted all features in the training and validation sets to float.
- Prevents **data type-related errors** during model fitting or evaluation.

Step 4- Handling Class Imbalance with SMOTE

- To address the class imbalance in the training data, we applied SMOTE (Synthetic Minority Over-sampling Technique).
- Creates **synthetic samples** for the minority class (defaulters).
- Balances the dataset without duplicating existing rows.
- Used Counter(y_train_bal) to confirm that both classes are now **evenly represented**.

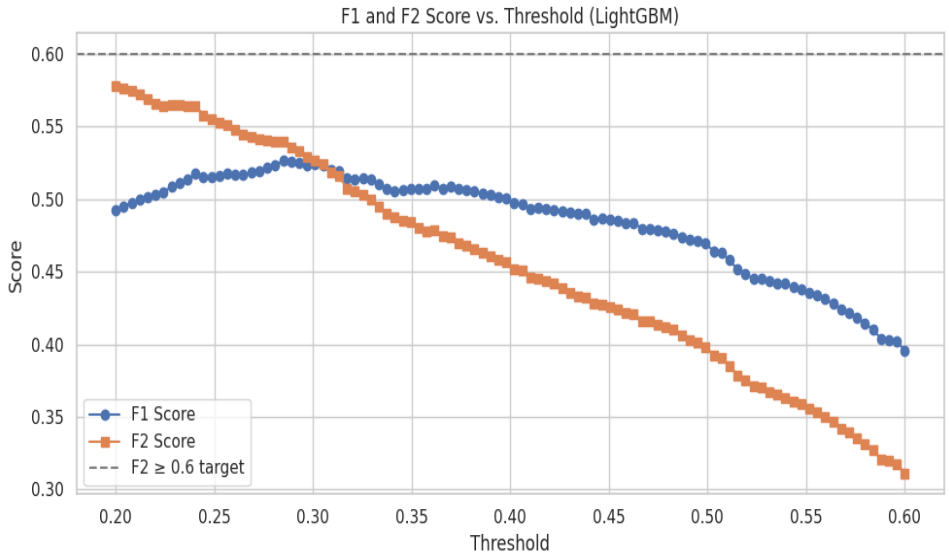
Step 5- Model Training & Performance Evaluation

- We trained and evaluated five different classification models using the balanced training set.
- Logistic Regression, Decision Tree (max depth = 5), XGBoost, LightGBM, Random Forest
- Trained each model on X_train_bal, y_train_bal.
- Predicted on the original imbalanced validation set (X_val).
- **F1 Score**: Balances precision and recall (sensitive to class imbalance).
- **AUC-ROC**: Measures model's ability to distinguish between default and non-default.
- All evaluation results were stored in a dictionary for easy comparison
- This step helps compare models on default detection performance, enabling selection of the **best model** based on F1 and AUC.

Step 6.1-Optimal Threshold Selection (LightGBM)

- To improve classification performance beyond the default threshold of 0.5, we tuned the probability threshold for LightGBM using F1 and F2 scores.

- Predicted **probabilities** (val_probs) using the LightGBM model.
- Evaluated thresholds from **0.2 to 0.6**
- **F1 Score**: Balance between precision and recall.
- **F2 Score**: Puts more weight on recall (important for detecting defaults).
- Selected threshold maximized F2 Score, giving greater weight to recall, which is critical for detecting risky borrowers.



Step 6.2: Confusion Matrix with Tuned Threshold

- After selecting the optimal threshold based on F2 score, we evaluated the LightGBM model's classification performance using a confusion matrix.
- Plotted the confusion matrix to visualize true vs. predicted labels on the validation set.
- This model enables proactive credit risk management by **detecting 65%+ of potential defaulters** before they miss payments.
- The threshold strategy sacrifices some precision for **risk containment**, aligning with the bank's need to **minimize loan loss provisions**.
- Offers actionable insights into default behavior (e.g., repayment history, credit utilization), contributing to smarter customer monitoring and policy adjustments.

Step 6.3-Final Model Deployment & Prediction Pipeline

- To operationalize the default prediction model for real-world usage, we implemented a structured and reproducible prediction pipeline using the final LightGBM model trained on a SMOTE-balanced dataset. This pipeline was applied to the bank's holdout validation dataset (unseen data) to identify customers at risk of defaulting in the upcoming billing cycle.
- The prediction pipeline is tuned not only for accuracy but also for **financial interpretability and operational utility**. By optimizing for F2 Score and targeting early identification of risky customers, the model enables the bank to:
- Trigger early warning alerts**.**
- Reprioritize collection strategies**.**
- Minimize future loan loss provisions**.**

Predicted Default Summary — Final Model Output

- After applying the final tuned LightGBM model with an optimized threshold of 0.20, we evaluated the risk exposure in the validation dataset based on the model's predictions.

Prediction Output Analysis

- The default rate of 24.84% reflects the proportion of customers identified by the model as high risk for **defaulting in the next billing cycle.**
- This rate is not a literal forecast of population-wide default but represents the model's risk flagging based on patterns learned during training.
- The high recall (and acceptable F2 score) achieved by using a low classification threshold (0.20) enables the bank to maximize early detection even if it means flagging more customers conservatively.

Step 7-Business Interpretation

The primary objective of this project was to support Bank A's credit risk strategy by building a predictive model that flags potential credit card defaulters one month in advance. This proactive approach enables the bank to

- Reduce financial losses by identifying high-risk customers before default occurs.
- Trigger early intervention measures, such as reducing credit limits, offering restructuring, or initiating customer outreach.
- Optimize risk-based decision-making, supporting smarter credit exposure management.
- By choosing a lower threshold (0.20) and optimizing for the F2 score, the model prioritizes recall over precision — intentionally flagging more customers to minimize false negatives, which aligns with the bank's risk appetite. The flagged default rate of 24.84% is conservative but valuable for activating early warning systems.
- This behavior score doesn't just predict — it informs real-world action. It equips the bank with a forward-looking tool to monitor risk dynamically and adjust policies accordingly, improving the overall health of the credit portfolio.

Conclusion

- In this project, I developed a forward-looking Behavior Score model for Bank A to predict which credit card customers may default in the next billing cycle. Using behavioral data from over 30,000 customers, I performed thorough analysis, engineered key features, and handled class imbalance with SMOTE.
- After evaluating several models, I selected LightGBM for its strong balance of accuracy and recall. To prioritize early risk detection, I fine-tuned the classification threshold to 0.20 using the F2 score, which better aligns with the bank's goal of minimizing missed defaulters.

On the validation set, the model achieved:

- F1 Score: 0.4922
- F2 Score: 0.5784

When applied to the final test data:

- **Predicted Defaulters:** 1246 out of 5016 customers
- **Default Rate Flagged: 24.84%**

    - This behavior score model helps the bank take proactive steps by identifying high-risk customers early, improving credit risk control, and supporting risk-based decision-making.