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

FinClub IIT Roorkee - Summer Project

June 2025

1 Overview of Modeling Strategy

This project aimed to predict credit default risk by modeling credit card behavior using both domain-specific feature engineering and machine learning. The pipeline includes:

- Data preprocessing and extensive feature engineering (e.g., delay behavior, credit utilization, volatility)
- Cluster-based segmentation to extract customer behavior profiles
- Model training with hyperparameter tuning via GridSearchCV and Optuna
- Threshold selection focused on F2-score maximization
- Interpretation using SHAP and surrogate decision tree for transparency

2 Data Preprocessing

Before feature engineering, categorical and numerical variables were cleaned, transformed, and consolidated to improve interpretability and model robustness. This preprocessing step was essential to ensure consistency across the dataset and prepare it for feature derivation.

Categorical Variable Mapping

Education: To reduce noise from infrequent categories, the `education` variable was remapped as follows:

- 1: Graduate School (Masters or PhD)
- 2: University (Bachelor's)
- 3: High School
- 4: Others (Vocational/Trade, Less than High School, Dropouts, Not Disclosed)

Justification: Categories 0 (Not Disclosed), 5 (Less than High School), and 6 (Dropout) together represented only 1.23% of the dataset and were therefore grouped under "Others" to minimize sparsity and enhance model generalization.

Marital Status: The `marriage` variable was simplified as:

- 1: Single

- 2: Married
- 0: Others (Divorced and Not Disclosed)

Justification: Categories 0 (Not Disclosed) and 3 (Divorced) accounted for just 1.3% of the dataset and were thus combined into a single "Others" category for modeling stability.

Numerical Imputation

Age: Missing values in the `age` column were imputed using the median of the available values.

- `df['age'].fillna(df['age'].median(), inplace=True)`

Justification: Median imputation was chosen to handle missing values in `age` as it is robust to outliers and preserves the central tendency of the data, making it suitable for skewed distributions.

3 Exploratory Data Analysis

3.1 Credit Limit vs Default Status

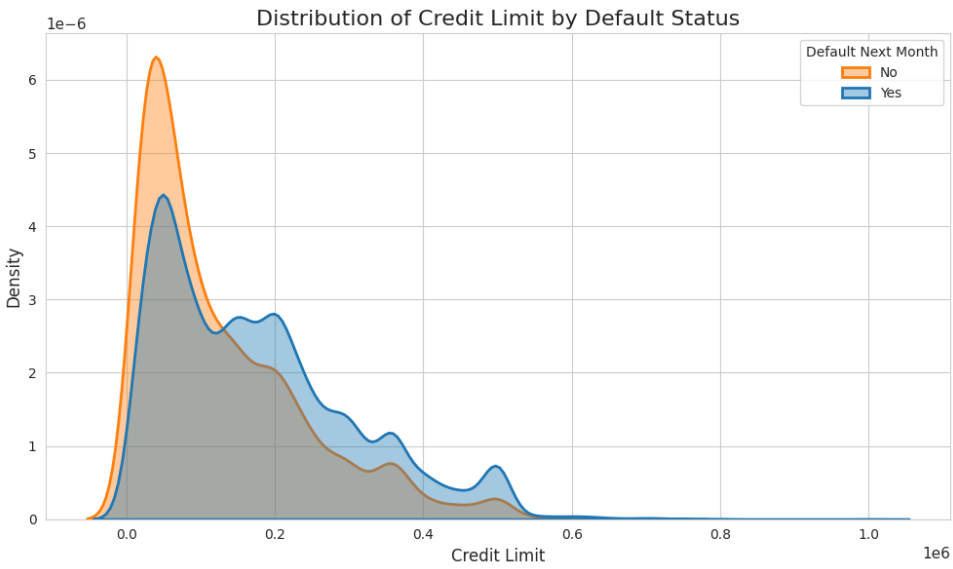

Figure 1: Distribution of Credit Limit by Default Status

Insights:

- **Defaulters (Yes)** show a **left-shifted peak**, indicating that most defaults occur at lower credit limits.
- **Non-Defaulters (No)** have a broader spread of higher credit limits, suggesting more financial flexibility.
- Low-limit borrowers are more vulnerable to default, likely due to higher credit utilization and financial stress.
- **Feature Engineering:** A new feature `utilization = bill_amt / LIMIT_BAL` was created to capture spending pressure.

3.2 Default Rate by Age Group and Education Level

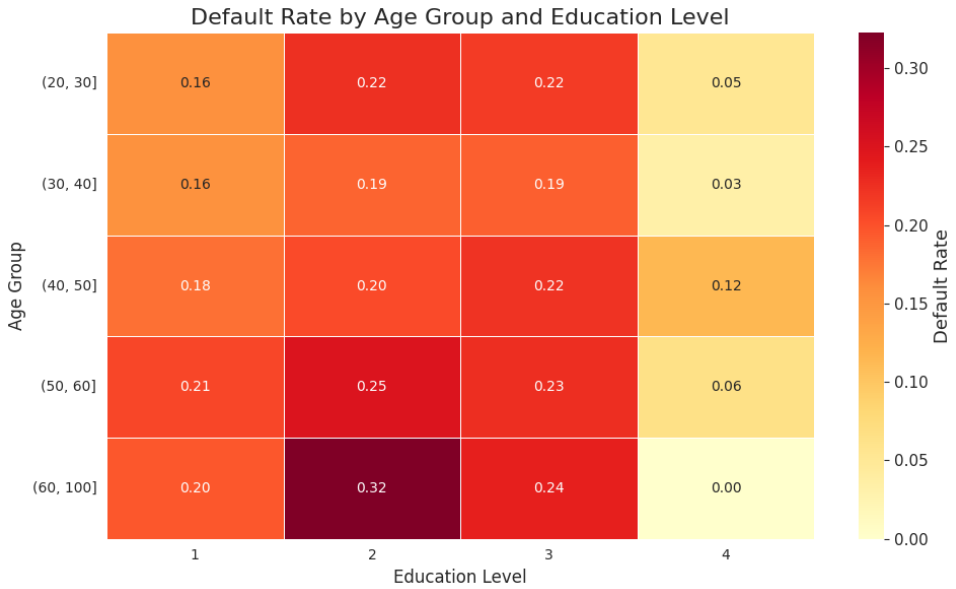

Figure 2: Default Rate by Age Group and Education Level

Insights:

- **Least Educated (Level 4)** group shows the **lowest default rates** (0.00–0.12), possibly due to conservative credit behavior or limited credit exposure.
- **More Educated (Levels 1–3)** exhibit **higher defaults**, peaking at **0.32** in the 60+ age group with Education Level 2.
- Potential drivers include overleveraging, educational loan burdens, or post-retirement income drops.
- *Note:* Education Level 4 has a small sample size, so findings may be skewed.

3.3 Default Rate by Credit Limit Bin

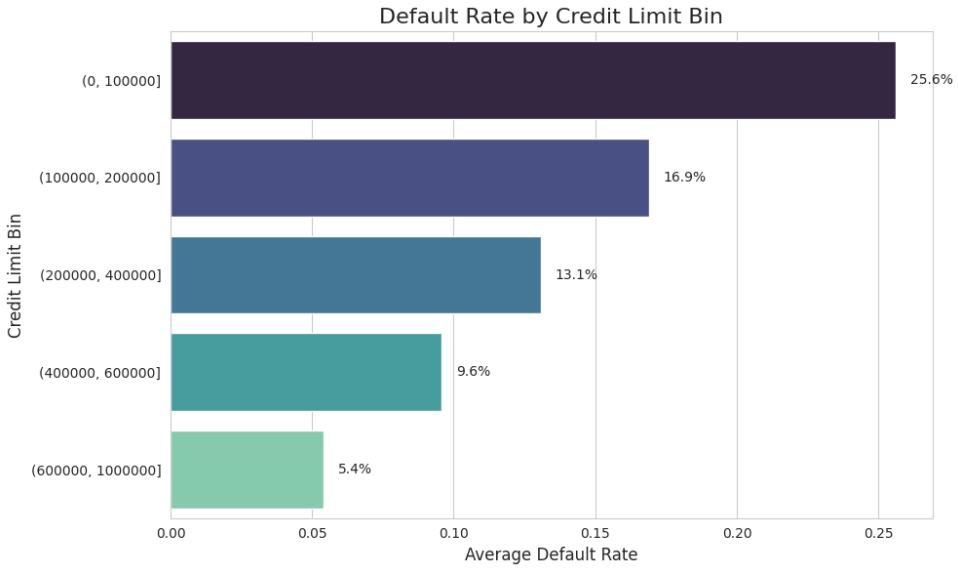

Figure 3: Default Rate by Credit Limit Bin

Insights:

- **Higher limits → Lower default risk**.
- Default rate drops from **25.6%** in the 0–100K bin to just **5.4%** in the 600K–1M bin.
- **Risk thresholds:**
  - **Critical zone:** Credit limits below 200K have 17–26% default rates.
  - **Safe zone:** Credit limits above 400K see default rates below 10%.
- Low-limit borrowers tend to overutilize credit and may fall into higher-risk segments.
- **Feature Engineering:** Created a categorical variable `limit_bin` for modeling.

3.4 Customer Segmentation via Clustering

To uncover hidden customer profiles, we employed unsupervised clustering techniques using KMeans. The optimal number of clusters for each analysis was determined using the elbow method. Clusters were then analyzed based on default rates to extract risk-oriented groupings. Each cluster's label was later used as a new categorical feature for modeling.

3.4.1 Repayment Dynamics Clusters

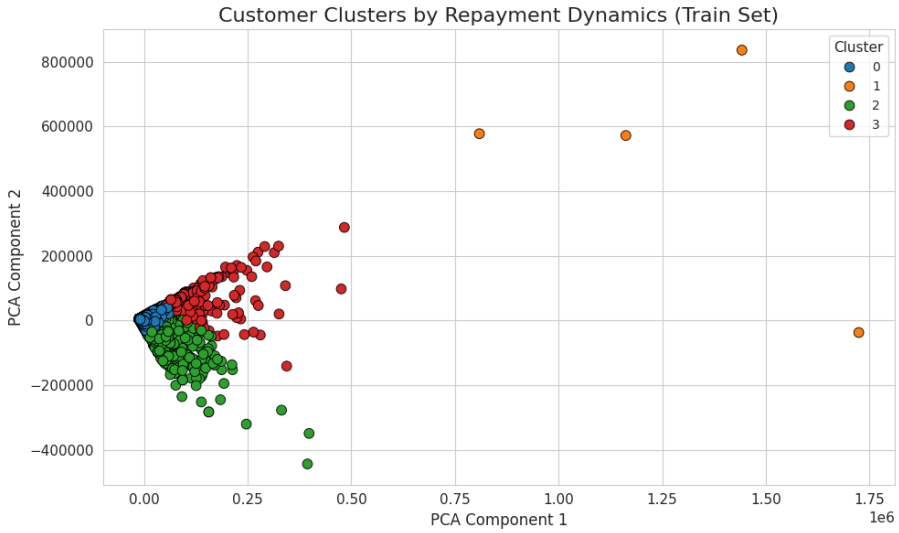

Figure 4: Customer Clusters by Repayment Dynamics

Based on repayment-related features (`pay_amt1` to `pay_amt6`), KMeans identified clear behavioral groups:

- **Cluster 0 (Highest Default)**: Customers with irregular and low repayments.
- **Cluster 1 (Lowest Default)**: Consistent repayment behavior and healthier patterns.

Feature Engineering: Cluster labels were added as a new categorical feature named `repay_cluster`.

3.4.2 Credit Usage Clusters

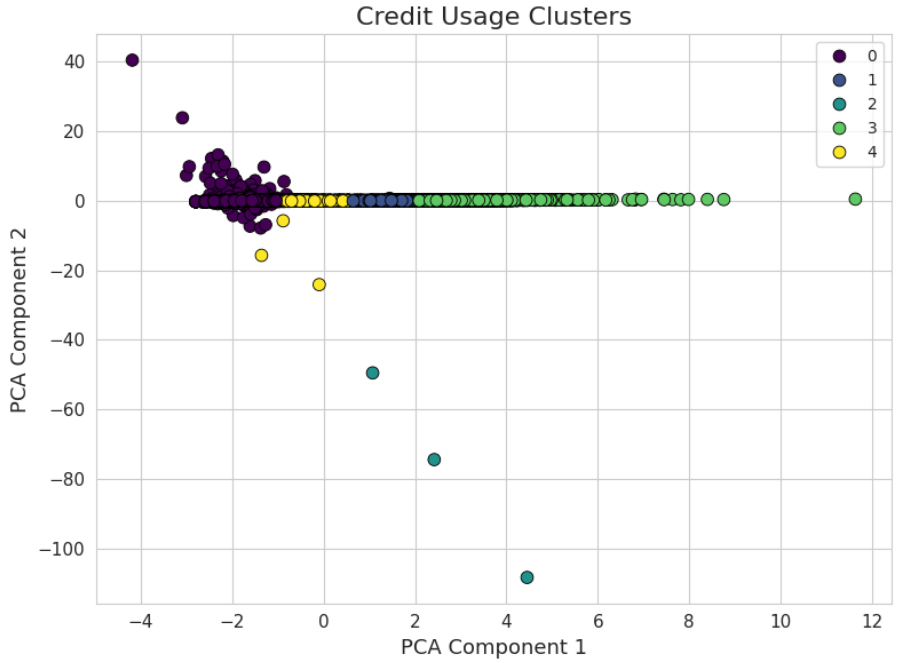

Figure 5: Credit Usage-Based Clustering of Customers

This segmentation used financial health indicators: `utilization`, `bill_limit_ratio`, `AVG_Bill_amt`, `PAY_TO_BILL_ratio`, and `missed_payments`.

- **Cluster 3 (Highest Default)**: High utilization and frequent missed payments.
- **Cluster 2 (Lowest Default)**: Responsible credit usage and timely payments.

Feature Engineering: Added `usage_cluster` as a categorical input feature.

3.4.3 Demographic and Behavior Clusters

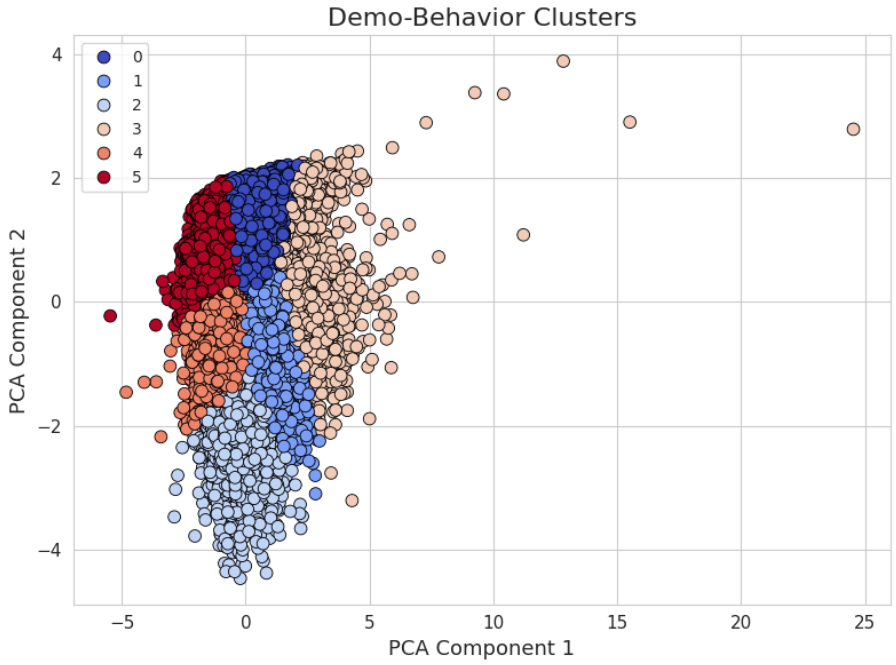

Figure 6: Customer Clusters by Demographics and Financial Behavior

This clustering considered both static and dynamic variables: `age`, `education`, `marriage`, `LIMIT_BAL`, `AVG_PAY_amt`, and `utilization`.

- **Cluster 4 (Highest Default)**: Younger, riskier profiles with fluctuating repayment.
- **Cluster 3 (Lowest Default)**: Mature profiles with balanced financial behavior.

Feature Engineering: Cluster identities were encoded as `demo_cluster`.

4 Feature Engineering

Feature engineering was performed in multiple stages to progressively refine raw data into insightful and predictive variables. Notably, the distinction between *initial* and *final* features arises from the timing and purpose of their creation: some features were derived prior to EDA to aid visualization and analysis, while others were created as a result of patterns discovered during EDA. The final features represent a financially meaningful and model-optimized subset built upon these insights.

4.1 Initial Features

These features were engineered before exploratory data analysis (EDA) to assist in understanding payment and billing patterns visually and statistically:

- **AVG_PAY_amt:** Mean of the last 6 monthly payments.
- **bill_limit_ratio:** Ratio of average bill amount to the credit limit.
- **missed_payments:** Count of months with delayed or zero payments.
- **bill_trend:** Slope of a linear regression line fitted to monthly bill amounts.

4.2 EDA-Inspired Features

Several features were derived based on patterns observed during exploratory data analysis (EDA). These transformations were guided by distributional insights and were designed to enhance the interpretability and predictive power of existing variables.

- **utilization:** Computed as the ratio of average bill amount to credit limit:

$$\texttt{utilization} = \frac{\texttt{AVG\_Bill\_amt}}{\texttt{LIMIT\_BAL}}$$

  This ratio captures how much of the assigned credit a user typically utilizes. High utilization is often associated with elevated credit risk.

- **LIMIT_BIN:** Credit limit was binned into discrete categories using the following rule:

$$\texttt{bins} = [0, 100{,}000,\ 200{,}000,\ 400{,}000,\ 600{,}000,\ 1{,}000{,}000]$$

  This binning was motivated by histogram plots that revealed natural groupings in the credit limit distribution, and enables categorical analysis based on financial capacity tiers.

4.3 Final Features

The final feature set was carefully selected to capture both behavioral risk indicators and financial usage patterns. Each feature was derived using targeted transformations based on domain knowledge and exploratory insights:

- **payment_std:** Standard deviation of the last 6 payment amounts, measuring inconsistency in repayment behavior.
- **bill_std:** Standard deviation of the last 6 billed amounts, indicating volatility in monthly expenses.
- **max_delay:** Maximum recorded delay in repayment across months 0, 2, 3, 4, 5, and 6 — a strong proxy for worst-case credit behavior.
- **avg_delay:** Average delay across the same months, summarizing overall repayment punctuality.
- **count_delay:** Total number of months (out of 6) where a payment delay was observed (delay ¿ 0), quantifying frequency of delinquency.
- **pay.delay_x_util:** Interaction between maximum delay and credit utilization; highlights users who are both overleveraged and delinquent.

- **bill_cv:** Coefficient of variation of bill amounts, calculated as standard deviation over mean bill — normalizes billing irregularity.
- **payment_cv:** Coefficient of variation of payments, adjusting payment inconsistency for average amount.
- **bill_trend_x_util:** Interaction between the slope of billing trend and utilization, used to detect rising spending combined with high leverage.
- **bill_trend_dir:** Binary indicator of whether the billing trend is increasing (1) or not (0), simplifying the slope into directional insight.
- **recent_bill_spike:** Ratio of the latest bill amount (month 6) to average billing, capturing abnormal short-term spikes.
- **limit_util_peak:** Peak utilization ratio over the last 3 months, computed as the maximum of bill-to-limit ratio for months 4–6.
- **avg.delay_x_util:** Product of average delay and credit utilization, emphasizing risk from consistently delayed payments in high-utilization accounts.
- **no_payment_flag:** Binary flag marking complete payment default in the most recent month (payment in month 6 = 0).
- **new_activity_flag:** Indicates sudden account reactivation — defined by a payment in month 6 following zero payment in month 5.

5 Feature Selection Strategy

5.1 Multicollinearity Management

Highly correlated features were retained selectively, leveraging the inherent robustness of tree-based models to multicollinearity—provided the features contributed meaningfully to model performance.

Significant correlation patterns were observed within and across the following groups:

- **Original payment-related features:** `PAY_0`, `PAY_AMT0`, `BILL_AMT0` etc, and their subsequent month-wise counterparts.
- **Engineered temporal delay features:** `count_delay`, `avg_delay` etc, and interaction terms such as `avg_delay_x_util` etc.

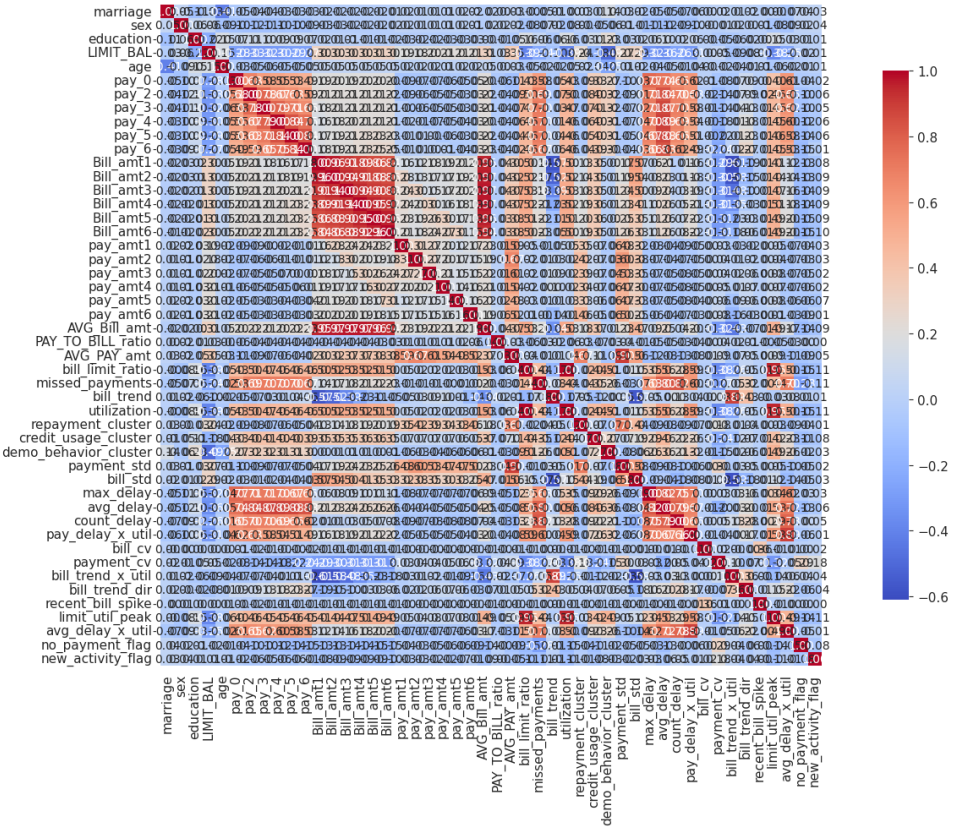

Figure 7: Heatmap of Feature Correlations for Multicollinearity Analysis

5.2 SHAP-Driven Feature Dropping

- Conducted 15+ SHAP-based iterations, continuously monitoring F_2-score and accuracy to ensure performance-driven pruning.
- Empirically removed features with persistently low SHAP values. Significant removals included:
  - Clustering and engineered signals: `repayment_cluster`, `credit_usage_cluster`, `bill_trend`
  - Redundant payment features: `pay_5`, `pay_6`, `AVG_PAY_amt`
  - Low-impact bill stats: `Bill_amt2`–`Bill_amt6`, `bill_std`, `payment_std`
  - Surprisingly, even `utilization` was removed in favor of stronger derived signals like `avg_delay_x_util`
- Retained correlated features if SHAP consistently indicated value — prioritizing impact over strict multicollinearity elimination.
- **Result**: Achieved consistent gains in both classification accuracy and F_2-score post-pruning.

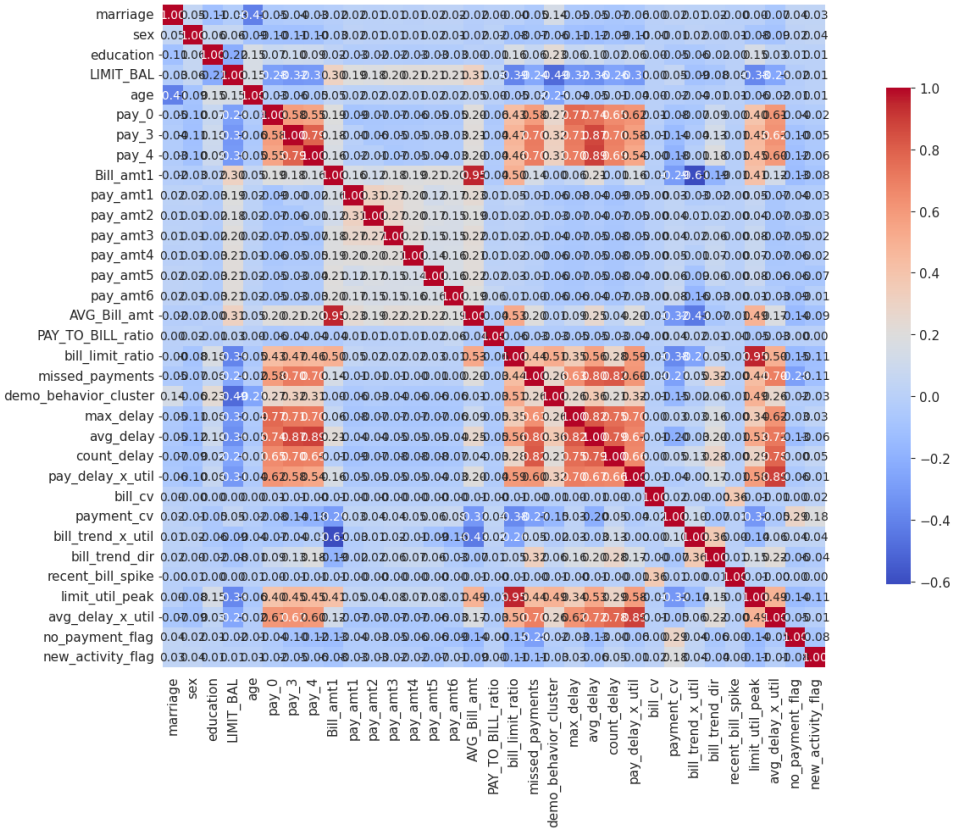

Figure 8: Post-SHAP Heatmap Showing Reduced Feature Correlation

6 Modeling Pipeline

6.1 Model Comparison

Initial benchmarking was conducted using `GridSearchCV` across multiple classifiers, including Logistic Regression, Support Vector Machine (SVM), Random Forest, and LightGBM. Ensemble approaches such as `XGBoost + LightGBM + Logistic Regression` were also tested.

However, **LightGBM consistently outperformed all other models**, striking the best balance between **F2-score** and **overall accuracy**, making it the preferred choice for subsequent development.

6.2 Optuna Tuning

Optuna was used to fine-tune LightGBM's parameters using wide hyperparameter ranges.

7 Threshold Optimization

In classification problems where class imbalance is a concern — such as credit default prediction — using the default threshold of 0.5 for probability classification can be suboptimal. Instead, optimizing the classification threshold using a metric that favors recall is crucial. For this purpose, we optimize the F_2-score, which prioritizes recall more heavily than precision.

Objective

Our goal is to find a threshold that maximizes the F_2-score:

$$F_\beta = \frac{(1+\beta^2) \cdot \text{precision} \cdot \text{recall}}{(\beta^2 \cdot \text{precision}) + \text{recall} + \varepsilon}$$

Here, we use $\beta = 2$ to weigh recall higher than precision, and ε is a small constant added to prevent division by zero.

Step 1: Generating Out-of-Fold Predictions

To ensure unbiased threshold tuning, we used 5-fold `StratifiedKFold` cross-validation. In each fold:

- The LightGBM model (with tuned hyperparameters) was trained on the training split.
- Probabilities were predicted for the validation fold.

This approach yields out-of-fold predictions for every sample in the training set without any data leakage, stored in `oof_preds`.

Step 2: Threshold Sweep using Precision-Recall Curve

Using `precision_recall_curve`, we computed precision and recall across a range of threshold values. Then, for each threshold, the corresponding F_2-score was calculated.

Implementation Snippet:

```
def f2_score(precision, recall, beta=2):
    return ((1 + beta**2) * (precision * recall)) /
           ((beta**2 * precision) + recall + 1e-8)
```

The optimal threshold is chosen as the one that maximizes the F_2-score.

Step 3: Final Threshold Selection

After computing the F_2-score for each threshold, we selected the threshold yielding the maximum F_2. This threshold was then used for final classification, ensuring optimal balance between recall and precision for the imbalanced setting.

Business Justification

In financial settings, especially credit risk modeling, the cost of a false negative (predicting non-default when it's a default) is typically higher than a false positive. Hence, optimizing recall-centric metrics like F_2-score ensures better real-world performance and aligns with business priorities.

8 Evaluation Metrics

- Accuracy: (0.69)
- F1 Score: (0.72)
- Recall: (0.69)
- F2 Score: (0.594)

9 Model Interpretability

9.1 SHAP Summary

SHAP summary plots offer a global perspective on feature importance and how individual feature values impact model predictions.

- `pay_delay_x_util`, `avg_delay`, and `marriage` emerged as the most influential predictors across the dataset.
- Features like `sex` and `missed_payments` also played a moderate role in shaping outcomes.
- The color gradient reflects feature value (red = high, blue = low), and horizontal dispersion shows feature impact on the model output.

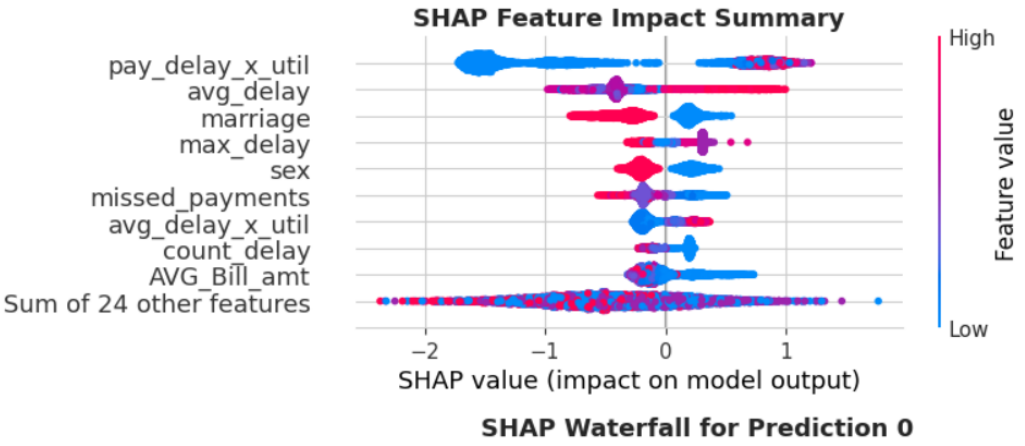

Figure 9: SHAP Summary Beeswarm Plot: Global Importance and Value Impact

9.2 SHAP Waterfall Plot

The SHAP waterfall plot explains the contribution of each feature for a specific prediction instance, showing how the base value (expected model output) is pushed up or down by feature effects to arrive at the final prediction.

- For example, `pay_delay_x_util` had a strong negative impact (-1.5), pulling the prediction toward a lower risk score.
- Features like `avg_delay`, `marriage`, and `max_delay` also contributed moderately.
- Positive SHAP values (in red) increase the prediction, while negative values (in blue) decrease it.

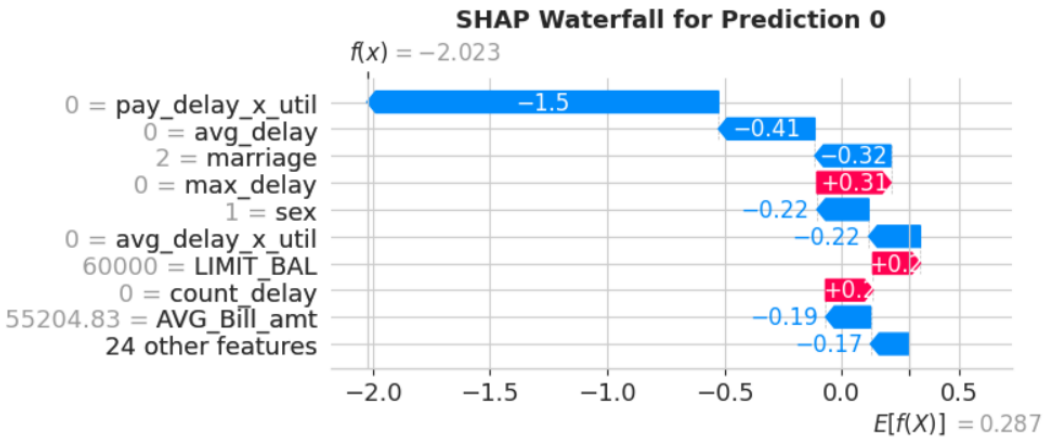

Figure 10: SHAP Waterfall Plot: Feature Contributions for One Prediction

9.3 SHAP Dependence Plots

SHAP dependence plots illustrate the marginal impact of each feature across their value ranges. They also help verify monotonic trends and feature influence. Notably, features with lower SHAP values were retained, as dropping them led to reduced F2-score and accuracy — indicating their utility in classifying borderline or ambiguous cases.

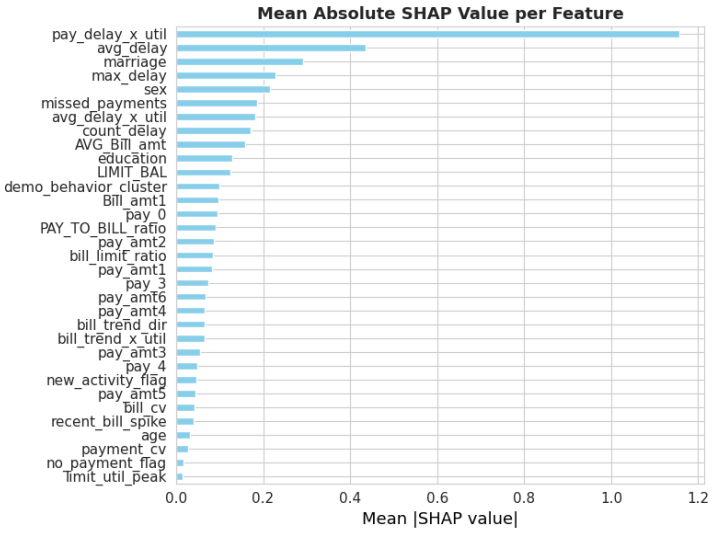

Figure 11: SHAP Value per Feature Plot

10 Surrogate Decision Tree Interpretation – Credit Default Model

10.1 Purpose

The original model for credit default prediction is likely a complex ensemble or non-linear model, which limits its interpretability. A **surrogate decision tree**, trained to approximate the black-box model's predictions (not the ground truth), serves as a transparent approximation. It reveals which features drive the model's decisions and how those decisions are structured hierarchically.

10.2 Tree Structure

- The root node splits on `pay_delay_x_util`, a custom feature combining payment delays with credit utilization — suggesting this is the most predictive signal.
- Subsequent splits are based on:
  - Average and recent bill amounts (`AVG_Bill_amt`, `Bill_amt1`)
  - Past maximum delays (`max_delay`)
  - Marital status (`marriage`)
- Node coloring: **Orange** for "No Default", **Blue** for "Default".
- Each node displays:
  - Percentage of total samples that reach it
  - Class counts: [No Default, Default]
  - Predicted class at that node

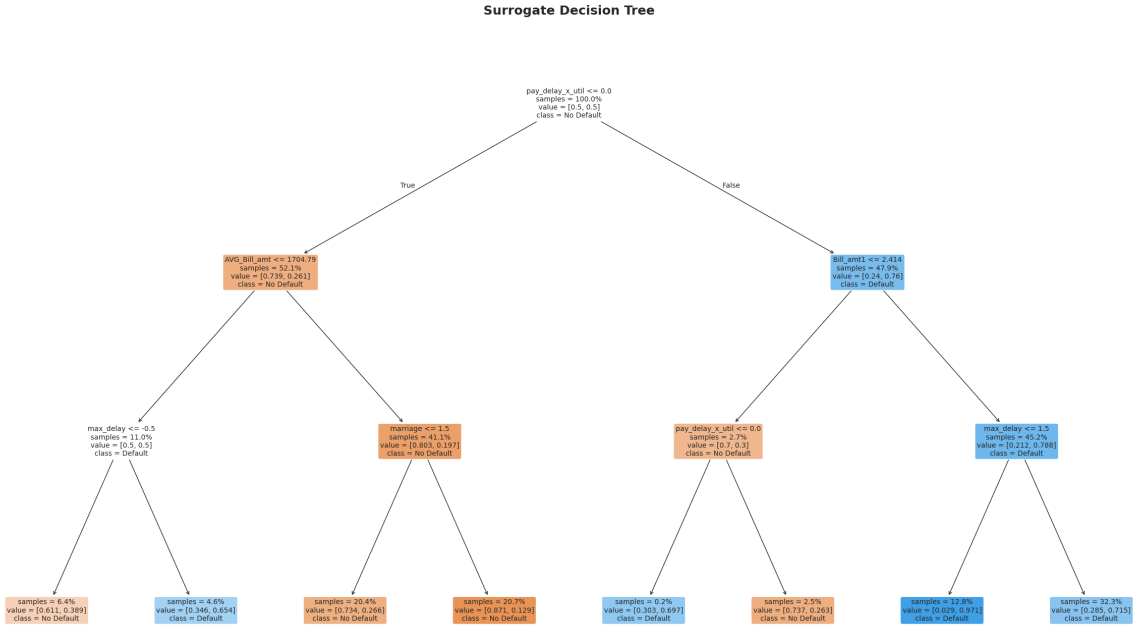

Figure 12: Surrogate Tree for Default Risk Prediction

10.3 Interpretation of Branches

Root Node: `pay_delay_x_util` ≤ 0 → interpreted as low payment delay or low credit utilization. *Interpretation:* These customers are typically lower-risk.

Left Branch (52% of samples): → Features like low `AVG_Bill_amt`, no history of delays, and single marital status often lead to a "No Default" prediction. *Interpretation:* Represents low exposure and good financial behavior.

Right Branch (48% of samples): → Includes high values of `pay_delay_x_util`. → Among these, severe past delays (`max_delay` ¿ 1.5) dominate prediction and push it toward "Default". *Interpretation:* Delay patterns outweigh moderate bills or marital status in predicting default.

10.4 Key Features Driving Splits

Feature	Interpretation
`pay_delay_x_util`	Captures both credit usage and delays. High values correlate strongly with default.
`AVG_Bill_amt`	Lower average billing correlates with lower risk exposure.
`max_delay`	The strongest red flag. Severe delays sharply increase default probability.
`Bill_amt1`	Most recent bill amount. Less influential unless combined with high delays.
`marriage`	May indirectly reflect financial responsibility. Singles in some branches showed lower default.

Table 1: Summary of Influential Features

10.5 Final Remarks

The surrogate decision tree offers a simplified yet interpretable view of how the original model makes predictions. It emphasizes that delay-related features—especially when combined with utilization—are central to risk assessment. While the tree abstracts away interaction terms and non-linearities, it helps uncover the model's decision structure and is particularly useful for communicating insights to stakeholders or regulators.

11 Evaluation Curves

11.1 Precision-Recall Curve

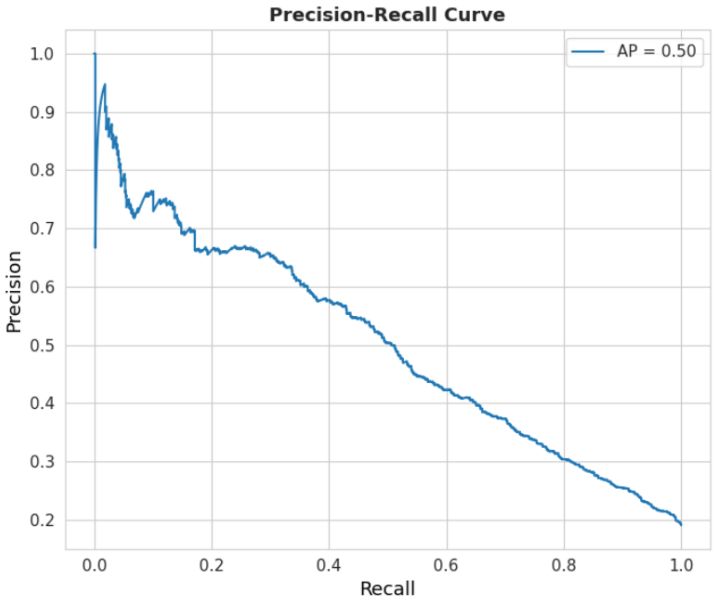

Figure 13: Precision-Recall Curve (AP = 0.50)

The Precision-Recall (PR) curve provides insight into the model's performance under class imbalance. In this context:

- **Average Precision (AP):** 0.50, indicating moderate precision across different recall thresholds.
- The curve starts with high precision at low recall, suggesting the model is effective at identifying high-confidence defaulters.
- However, precision drops steadily as recall increases, reflecting trade-offs when trying to capture more defaulters.
- This makes the model suitable for use cases where minimizing false positives is more important than capturing all defaulters.

Overall, the model demonstrates good early precision but lacks general robustness across the full recall range, highlighting the need for potential improvements or threshold tuning.

11.2 Calibration Curve

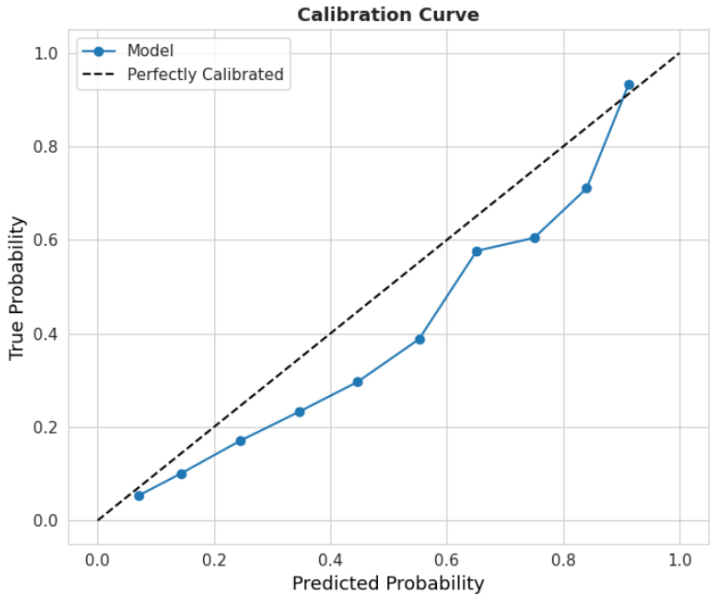

Figure 14: Calibration Curve

The calibration curve compares predicted probabilities with actual outcomes, evaluating how well the model's confidence aligns with observed behavior.

- The model shows **slight overconfidence** in the 0.2–0.6 predicted probability range, where predicted risks are higher than actual outcomes.
- Calibration improves significantly for predictions near 1.0, indicating better reliability for high-confidence cases.
- Deviation from the ideal diagonal line suggests that post-processing with calibration techniques such as *Platt scaling* or *isotonic regression* may improve probabilistic interpretation.

Accurate calibration is particularly important when predicted probabilities are used for downstream decisions, such as setting thresholds for risk interventions.

11.3 ROC Curve

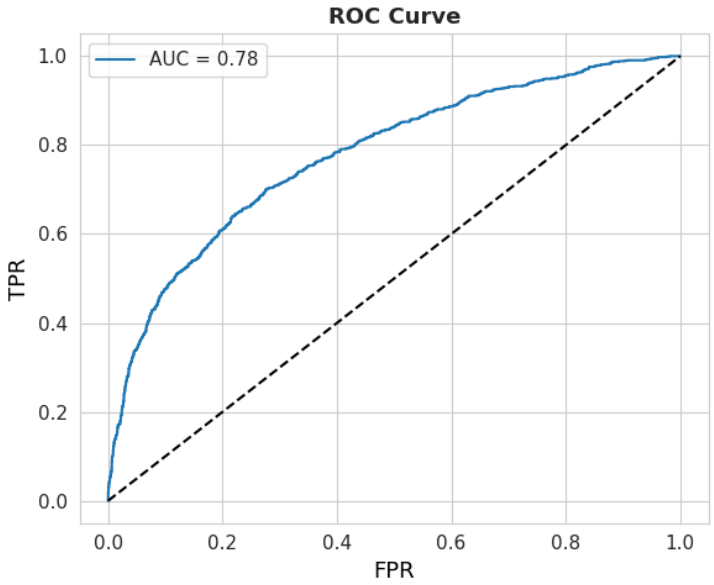

Figure 15: ROC Curve

The ROC curve illustrates the trade-off between the true positive rate (TPR) and false positive rate (FPR) across various classification thresholds.

- **Area Under Curve (AUC)**: 0.78 — a strong indication of decent discriminative power.
- The curve consistently stays above the diagonal, suggesting better-than-random classification.
- While performance is generally good, there remains headroom for improving sensitivity-specificity balance.

This performance is acceptable in practical settings, particularly where distinguishing defaulters is more critical than perfect classification.

12 Business Implications

- Customers showing **high credit utilization along with repayment delays** are consistently flagged as high-risk profiles. These signals can be used to trigger timely interventions, such as reminders or temporary freezes.
- The model enables **risk-based adjustments in credit limits or interest rates**, guided by stable predictors like utilization spikes, delay patterns, and recent repayment behavior.
- Techniques like **SHAP values and surrogate tree visualizations** increase model transparency, making it easier to justify predictions in high-stakes environments and build trust with compliance teams.

13 Conclusion and Learnings

- **SHAP-driven feature reduction** helped simplify the model while improving F_2-score — showing that focusing on interpretable, high-impact features leads to better generalization.
- **Delay-related variables**, especially when combined with utilization, emerged as the most reliable signals across all folds, validating their use in credit risk modeling.
- Several engineered features added value, but some complex transformations like some of the clusters had minimal impact — highlighting the importance of **validating features through metrics and interpretability tools**.
- **F_2-based threshold tuning** ensured that the model prioritized recall, which is critical in minimizing missed defaulters in financial applications.

End of Report