Finance Club

Open Project Summer 2025

Title:

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

Name: Uday Tyagi

23113155

Date: 16 June, 2025

Project Overview

This project focuses on developing a Behavior Score Model — a predictive classification system designed to estimate the likelihood of a credit card customer defaulting in the upcoming billing cycle. The model is built using anonymized historical data from Bank A, encompassing information such as credit limits, demographic details, repayment patterns, and bill/payment behavior.

The goal is not only to achieve high predictive accuracy but also to ensure financial interpretability. This allows the bank to make informed, proactive decisions such as adjusting credit limits, triggering early warning systems, or offering personalized customer interventions before a potential default occurs.

Key Challenges Addressed

- Class Imbalance: Default events are relatively rare, requiring careful handling to avoid biased predictions.
- Sequential Behavioral Patterns: Repayment history over time captures essential behavioral trends that influence default risk.
- Business-Relevant Evaluation Metrics: Prioritizing metrics like F2-score, which emphasizes recall, to minimize undetected defaulters — a critical requirement in credit risk management.

Approach Summary

- Conducted comprehensive exploratory and financial analysis to understand key drivers of customer default.
- Engineered domain-specific features such as credit utilization ratios and delinquency streak indicators to improve model interpretability and predictive power.
- Compared multiple machine learning models, including:
    - Logistic Regression
    - Decision Tree Classifier
    - XGBoost
    - LightGBM
- Applied hyperparameter tuning and threshold optimization to enhance model performance.
- Evaluated models using F2-score, giving higher importance to recall, ensuring the model captures as many potential defaulters as possible.

Final Deliverables

- A prediction CSV file containing default probabilities for the validation dataset.
- A clean, reproducible Jupyter Notebook documenting the complete workflow.
- A comprehensive report summarizing:
    - Data exploration and insights
    - Data preprocessing and feature engineering decisions
    - Model building, evaluation, and optimization results
    - Business impact analysis demonstrating the financial value of the model.

1. Data Loading and Basic Exploration

1.1 Objective

The first step involved loading the provided dataset and performing a basic exploration to understand the structure, content, and initial characteristics of the data. This is essential for identifying potential data quality issues and gaining preliminary insights.

1.2 Data Loading

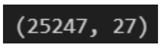
- **The dataset, provided in CSV format, was loaded using the Pandas library into two primary DataFrames:**
  - **train_df: Training dataset containing features and the target variable (next_month_default).**
  - **validate_df: Validation dataset containing the same features but without the target variable.**

1.3 Dataset Overview

Dataset	Records (Rows)	Features (Columns)
Training Set	**25,247**	**30 (including target)**
Validation Set	**5,016**	**29 (excluding target)**

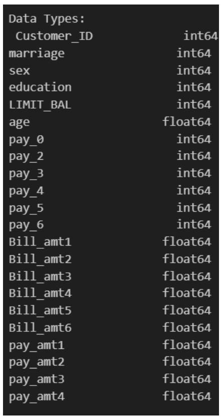
2. Exploratory Data Analysis (EDA)

2.1 Data Overview

- Shape of the DataFrame: 

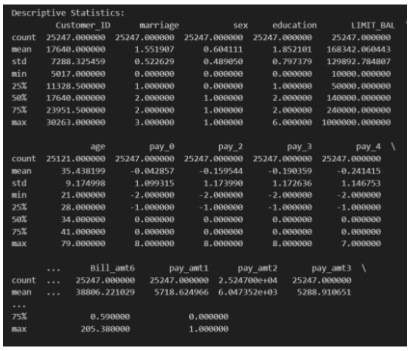
```
(25247, 27)
```

- Data Types:



```
Data Types:
 Customer_ID          int64
marriage             int64
sex                  int64
education            int64
LIMIT_BAL            int64
age                float64
pay_0                int64
pay_2                int64
pay_3                int64
pay_4                int64
pay_5                int64
pay_6                int64
Bill_amt1          float64
Bill_amt2          float64
Bill_amt3          float64
Bill_amt4          float64
Bill_amt5          float64
Bill_amt6          float64
pay_amt1           float64
pay_amt2           float64
pay_amt3           float64
pay_amt4           float64
```

- Descriptive Statistics:

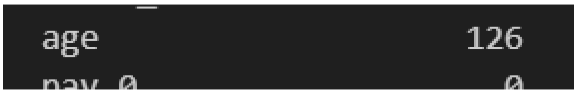


```
Descriptive Statistics:
         Customer_ID      marriage           sex     education        LIMIT_BAL
count   25247.000000  25247.000000  25247.000000  25247.000000     25247.000000
mean    17640.000000      1.551907      0.604111      1.852101    168342.060443
std      7288.325459      0.522629      0.489050      0.797379    129892.784807
min      5017.000000      0.000000      0.000000      0.000000     10000.000000
25%     11328.500000      1.000000      0.000000      1.000000     50000.000000
50%     17640.000000      2.000000      1.000000      2.000000    140000.000000
75%     23951.500000      2.000000      1.000000      2.000000    240000.000000
max     30263.000000      3.000000      1.000000      6.000000   1000000.000000

                 age         pay_0         pay_2         pay_3         pay_4  \
count   25121.000000  25247.000000  25247.000000  25247.000000  25247.000000
mean       35.438199     -0.042857     -0.159544     -0.190359     -0.241415
std         9.174998      1.099315      1.173990      1.172636      1.146753
min        21.000000     -2.000000     -2.000000     -2.000000     -2.000000
25%        28.000000     -1.000000     -1.000000     -1.000000     -1.000000
50%        34.000000      0.000000      0.000000      0.000000      0.000000
75%        41.000000      0.000000      0.000000      0.000000      0.000000
max        79.000000      8.000000      8.000000      8.000000      7.000000

       ...      Bill_amt6      pay_amt1      pay_amt2      pay_amt3  \
count  ...   25247.000000  25247.000000  2.524700e+04  25247.000000
mean   ...   38806.221029   5718.624966  6.047352e+03   5288.910651
...
75%             0.590000            0.000000
max           205.380000            1.000000
```

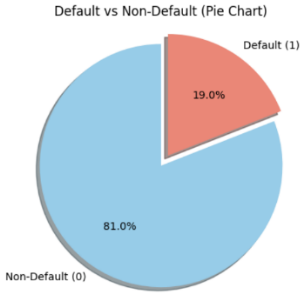
- Missing Values: rest all 0



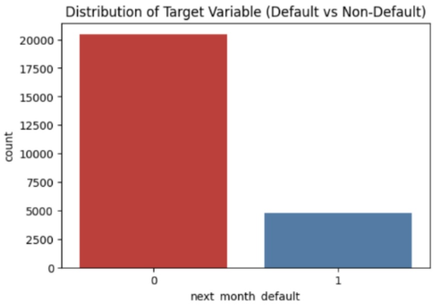
```
age                          126
pay_0                          0
```

2.2 Target Variable Distribution

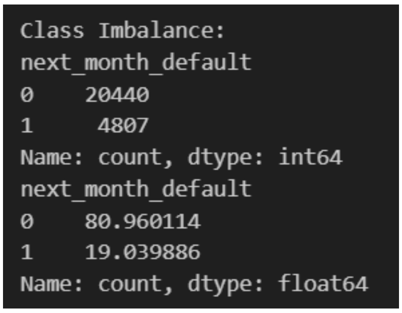
The target variable next_month_default is imbalanced, with a higher number of non-defaulters.



Default vs Non-Default (Pie Chart)



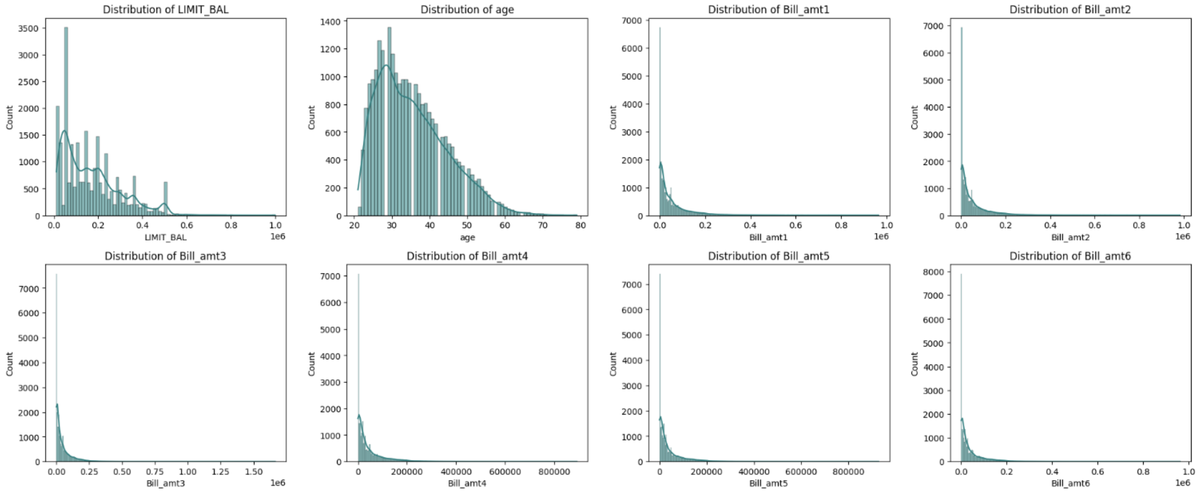
Distribution of Target Variable (Default vs Non-Default)

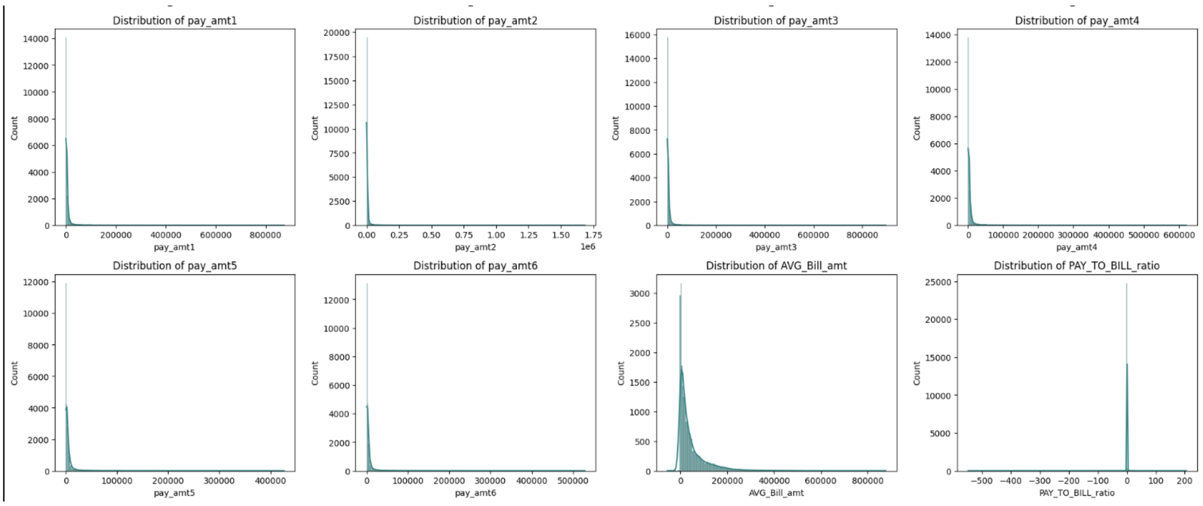


```
Class Imbalance:
next_month_default
0    20440
1     4807
Name: count, dtype: int64
next_month_default
0    80.960114
1    19.039886
Name: count, dtype: float64
```

2.3 Distribution of Numerical Features

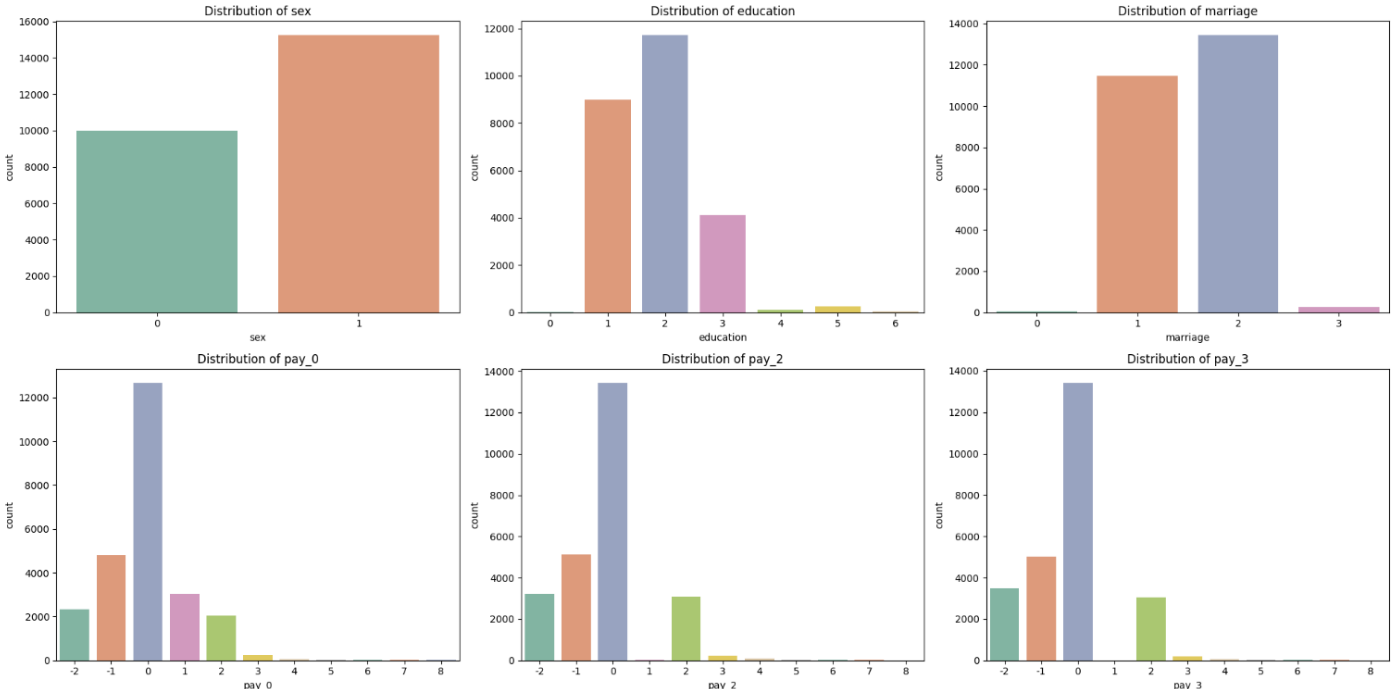
Histograms with KDE lines were plotted to check the distribution of important numerical variables like LIMIT_BAL, age, bill amounts, and payment amounts.

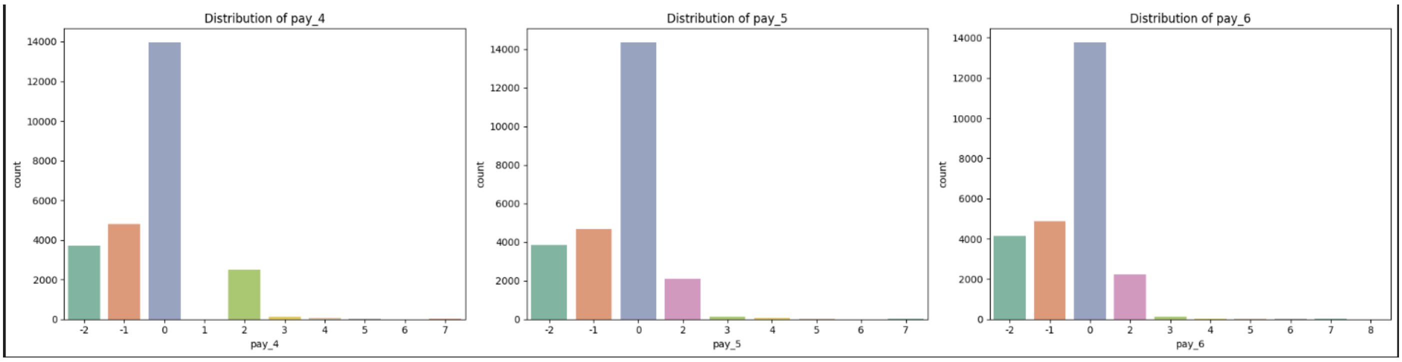




2.4 Distribution of Categorical Features

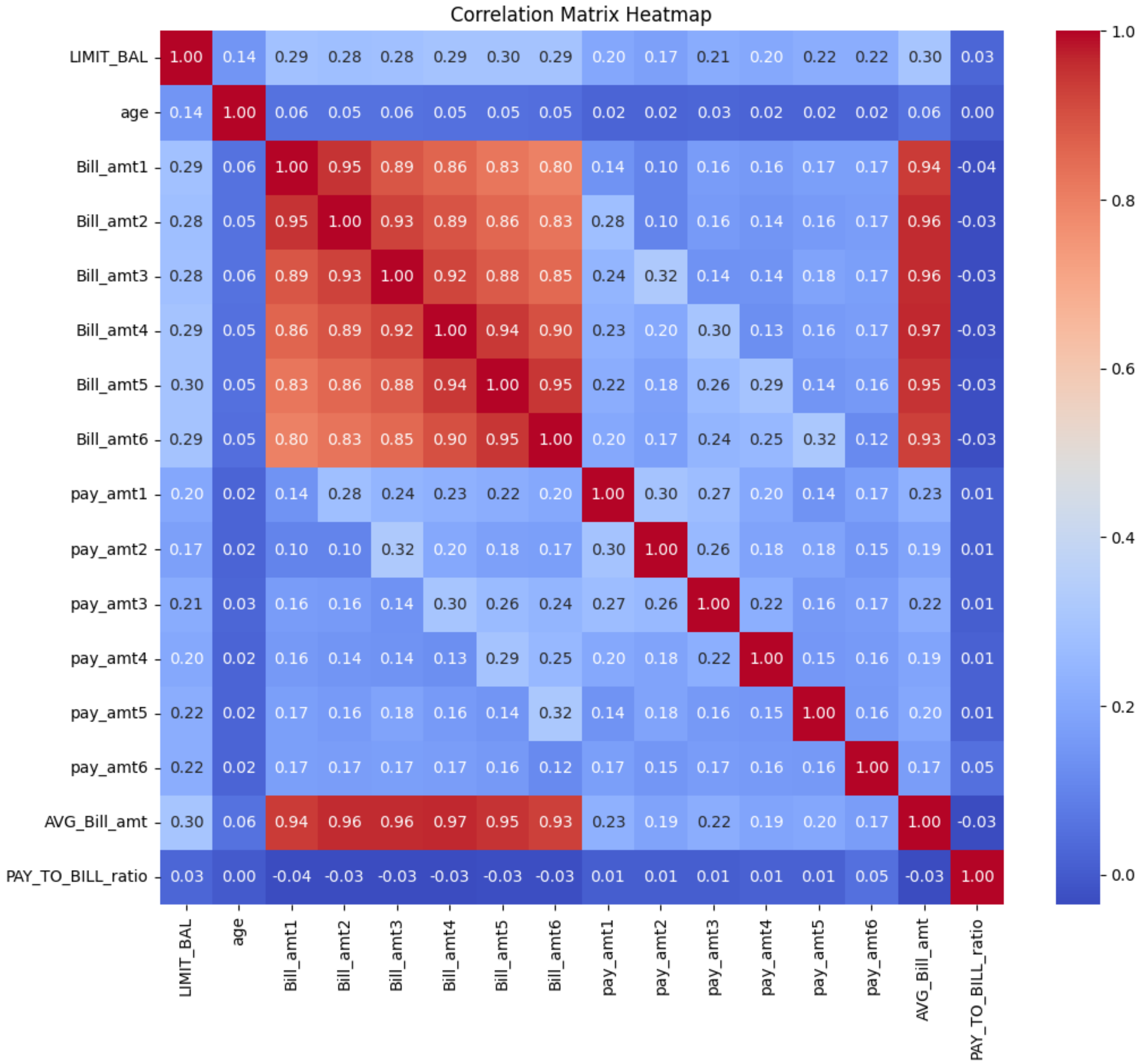
Countplots were generated for features such as sex, education, marriage, and repayment status (PAY_0 to PAY_6).





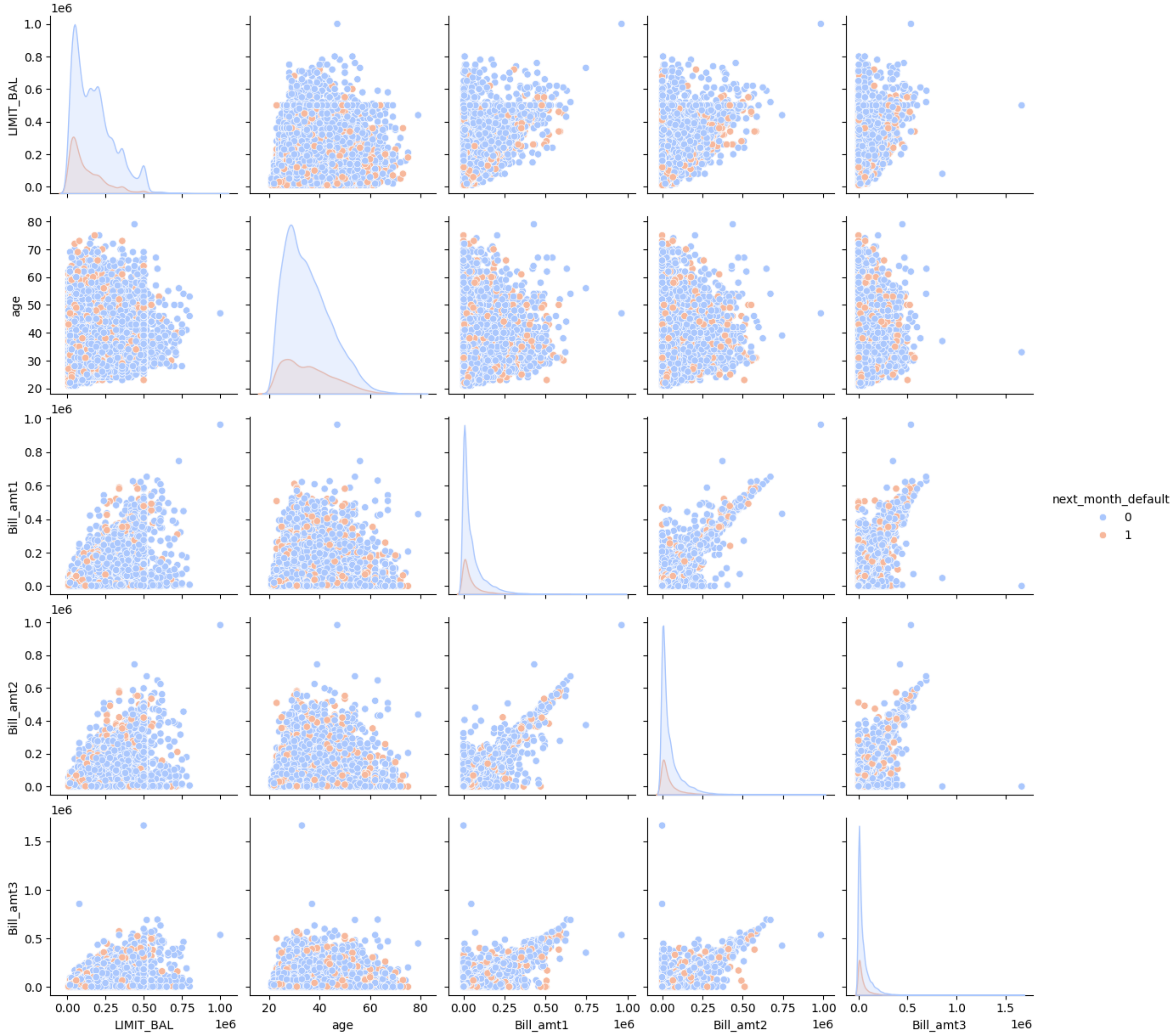
2.5 Correlation Analysis

A heatmap revealed strong correlations among bill amounts and repayment history features. PAY_0 showed the highest correlation with the target.



2.6 Pairplot Analysis

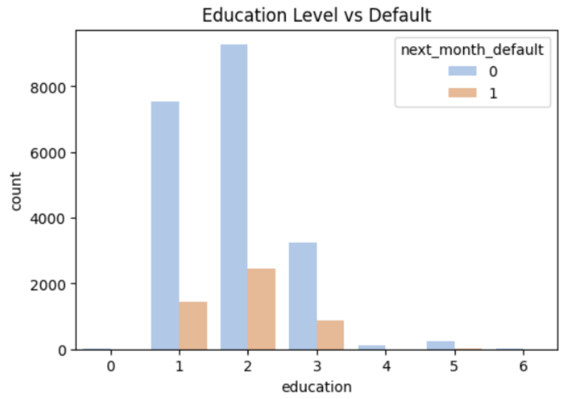
Top numerical features visualized via pairplots to detect relationships and separation patterns between defaulters and non-defaulters.



2.7 Relationship between Features and Default Rate

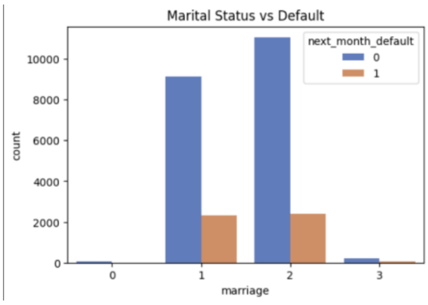
a) Education Level vs Default

Higher education levels are associated with lower default risk. (proportionally)



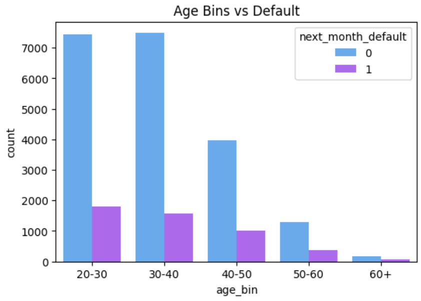
b) Marital Status vs Default

Marital status showed minor variation in default rates.



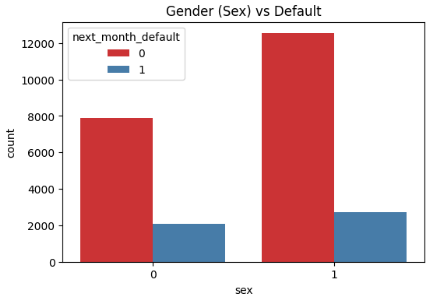
c) Age Binning vs Default

Younger customers (20–30) show higher default tendency.



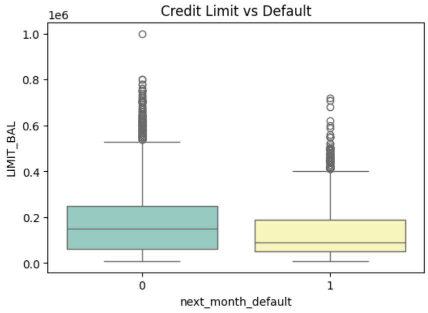
d) Gender vs Default

Male and female customers show similar default trends.



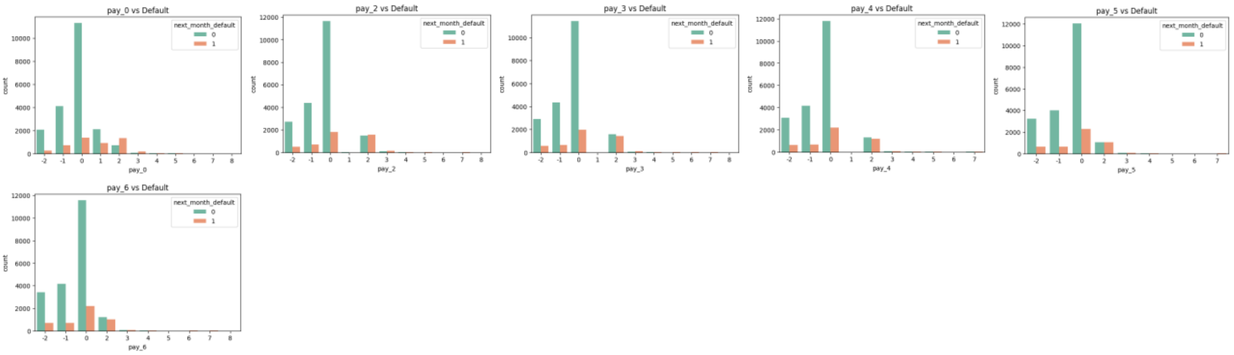
e) Credit Limit vs Default

Customers with lower credit limits tend to default more.



f) Repayment History (PAY_0 to PAY_6) vs Default

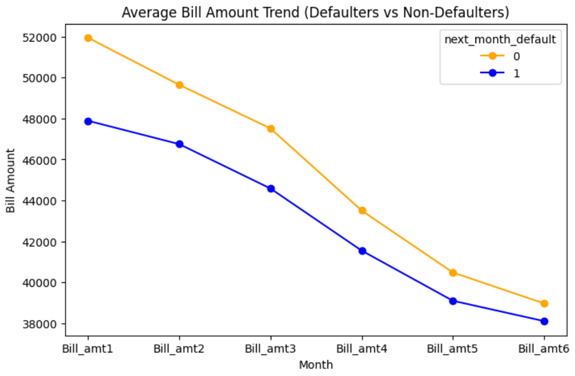
Late payments (positive PAY values) significantly increase default probability.



2.8 Bill Amount and Payment Amount Trends Over 6 Months

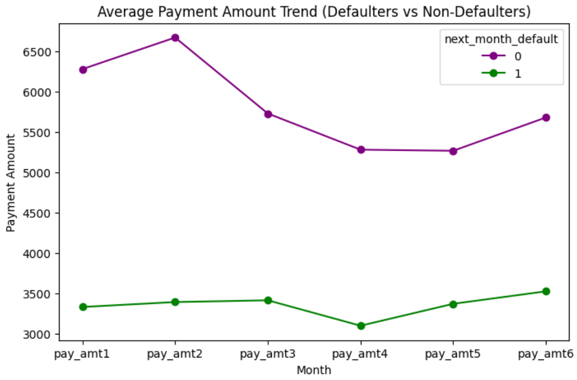
a) Average Bill Amount Trend

Non-defaulters exhibit higher and stable bill amounts over time.



b) Average Payment Amount Trend

Defaulters generally pay less compared to non-defaulters.



3. Data Cleaning and Preprocessing

After performing exploratory data analysis (EDA), the dataset underwent systematic cleaning and preprocessing to ensure consistency, accuracy, and readiness for modeling.

3.1 Handling Missing Values

- **Missing 'age' values** were identified and imputed using the **median age** of the respective dataset to avoid introducing bias.

3.2 Outlier Treatment

- Winsorization was applied to continuous numerical variables to **cap outliers at the 1st and 99th percentiles**. This was crucial to reduce the influence of extreme values on the model without removing data points.

The following numerical features were treated:

['LIMIT_BAL', 'age', 'Bill_amt1', 'Bill_amt2', 'Bill_amt3',

'Bill_amt4', 'Bill_amt5', 'Bill_amt6',

'pay_amt1', 'pay_amt2', 'pay_amt3', 'pay_amt4',

'pay_amt5', 'pay_amt6', 'AVG_Bill_amt', 'PAY_TO_BILL_ratio']

3.3 Data Encoding

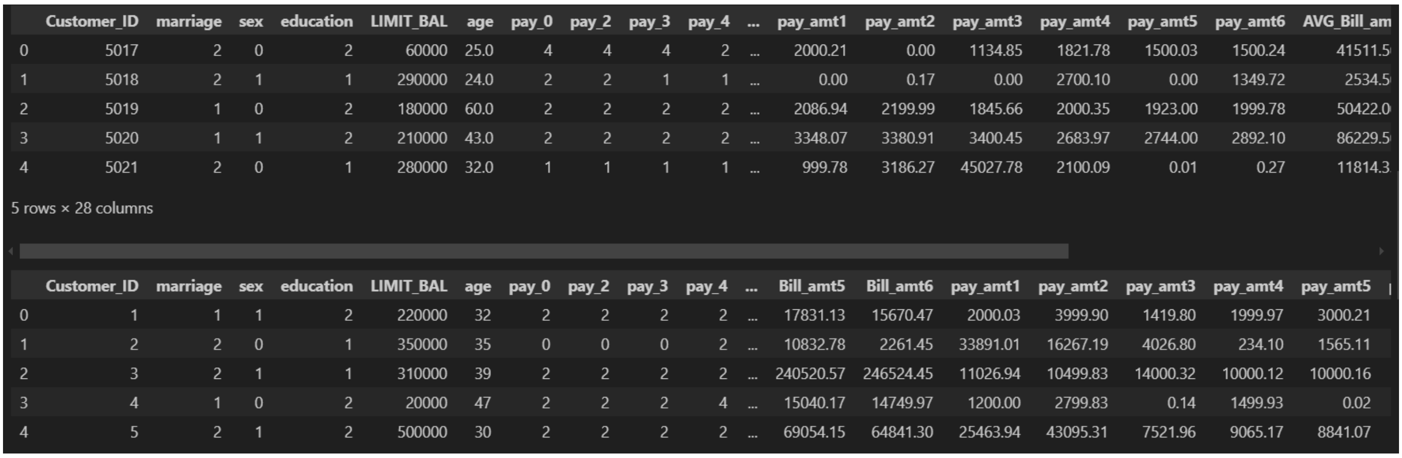
- The following **categorical variables** were converted into numeric form using **Label Encoding** to enable their use in machine learning models:
- ['sex', 'education', 'marriage', 'pay_0', 'pay_2', 'pay_3', 'pay_4', 'pay_5', 'pay_6']
- Encoding ensures that algorithms can process these features without assuming any ordinal relationship unless truly present.

3.4 Data Consistency Checks

To maintain **logical consistency and domain accuracy**, the following checks and adjustments were performed:

- **Payment Status Variables (pay_0 to pay_6)**:
  Clipped to the expected range of **-2 to 8**, where:
  - -2: No consumption
  - -1: Fully paid on time
  - 0: Minimum/partial payment made
  - 1 to 8: Payment overdue by corresponding months
- **'LIMIT_BAL' (Credit Limit)**:
  Clipped between **10,000 and 1,000,000 units** to reflect realistic credit limits based on financial product norms.
- **'age'**:
  Clipped between **20 and 80 years**, considering typical credit card holder demographics.

3.5 Post-Cleaning Data Preview



```
   Customer_ID  marriage  sex  education  LIMIT_BAL   age  pay_0  pay_2  pay_3  pay_4  ...  pay_amt1  pay_amt2  pay_amt3  pay_amt4  pay_amt5  pay_amt6  AVG_Bill_am
0         5017         2    0          2      60000  25.0      4      4      4      2  ...   2000.21      0.00   1134.85   1821.78   1500.03   1500.24      41511.5
1         5018         2    1          1     290000  24.0      2      2      1      1  ...      0.00      0.17      0.00   2700.10      0.00   1349.72       2534.5
2         5019         1    0          2     180000  60.0      2      2      2      2  ...   2086.94   2199.99   1845.66   2000.35   1923.00   1999.78      50422.0
3         5020         1    1          2     210000  43.0      2      2      2      2  ...   3348.07   3380.91   3400.45   2683.97   2744.00   2892.10      86229.5
4         5021         2    0          1     280000  32.0      1      1      1      1  ...    999.78   3186.27  45027.78   2100.09      0.01      0.27      11814.3

5 rows × 28 columns

   Customer_ID  marriage  sex  education  LIMIT_BAL  age  pay_0  pay_2  pay_3  pay_4  ...  Bill_amt5  Bill_amt6  pay_amt1  pay_amt2  pay_amt3  pay_amt4  pay_amt5
0            1         1    1          2     220000   32      2      2      2      2  ...   17831.13   15670.47   2000.03   3999.90   1419.80   1999.97   3000.21
1            2         2    0          1     350000   35      0      0      0      2  ...   10832.78    2261.45  33891.01  16267.19   4026.80    234.10   1565.11
2            3         2    1          1     310000   39      2      2      2      2  ...  240520.57  246524.45  11026.94  10499.83  14000.32  10000.12  10000.16
3            4         1    0          2      20000   47      2      2      2      4  ...   15040.17   14749.97   1200.00   2799.83      0.14   1499.93      0.02
4            5         2    1          2     500000   30      2      2      2      2  ...   69054.15   64841.30  25463.94  43095.31   7521.96   9065.17   8841.07
```

4. Feature Engineering

In addition to the original variables provided in the dataset, new **financially meaningful features** were engineered to improve model interpretability and predictive performance. These features capture customer behavior patterns related to credit utilization, payment consistency, and bill trends, which are critical indicators of credit risk.

4.1 New Features Created

The following new features were derived:

Feature Name	Description
credit_utilization	Ratio of total bill amounts to total credit limit over six months (existing in raw data)
avg_payment_delay	Average of all past payment status codes (pay_0, pay_2, ..., pay_6). Indicates general delay tendency.
max_payment_delay	Maximum delay observed in any past month. Highlights severe defaulters.
payment_delay_consistency	Standard deviation of payment delays. Higher value implies inconsistent repayment behavior.
bill_amt_trend	Difference between the latest and the oldest bill amounts (Bill_amt1 - Bill_amt6). Captures bill payment or spending trend over time.
pay_to_limit_ratio	Total payment over six months divided by the credit limit. Shows repayment ability relative to the limit.
pay_amt_std	Standard deviation of past payment amounts. Indicates fluctuation in monthly repayments.
recent_delay	Sum of the most recent two payment delays (pay_0 + pay_2). Highlights the short-term repayment risk.

4.2 Rationale Behind Feature Engineering

- **Payment Consistency Features** (avg_payment_delay, max_payment_delay, payment_delay_consistency)
  These features quantify how timely and consistently customers make payments. Irregular or high delay scores indicate potential default risk.

- **Bill and Payment Behavior Features** (bill_amt_trend, pay_to_limit_ratio, pay_amt_std)
  Capture patterns in credit usage and repayment amount behavior over time — essential for understanding financial discipline.

- **Recent Payment Behavior Feature** (recent_delay)
  Focuses on the last two months, as recent payment behavior often signals imminent default risk.

4.3 Dropped Features

To avoid multicollinearity and reduce redundancy:

- **Individual Bill Amounts (Bill_amt1 to Bill_amt6)** were dropped in favor of derived trend features (AVG_Bill_amt, bill_amt_trend).

- **Individual Payment Amounts (pay_amt1 to pay_amt6)** were replaced by aggregated indicators like PAY_TO_BILL_ratio, pay_to_limit_ratio, and pay_amt_std.

- **Customer ID** was removed from the **training set** (we needed it in validation set) as it holds no predictive power.

4.4 Post-Feature Engineering Data Shape

Dataset	Shape After Feature Engineering
Training Set	(25247 rows, 22 columns)
Validation Set	(5016 rows, 21 columns)

---

5. Data Preparation

5.1 Feature and Target Separation

The preprocessed dataset was split into:

- **Features (X)**: All independent variables excluding the target.

- **Target Variable (y)**: The dependent variable next_month_default, which indicates whether the customer defaulted in the following month (1 = Default, 0 = No Default).

5.2 Handling Missing Values

- **Target Variable (y)**:
  Any missing values in the target variable were filled with the **most frequent class (mode)** to ensure no loss of data in the modeling process.

- **Feature Variables (X)**:
  Missing values in the feature set were imputed using the **mean strategy** via SimpleImputer from Scikit-learn. This ensures that no feature has missing values which could otherwise affect model training.

5.3 Class Imbalance Treatment (SMOTE)

A significant class imbalance was observed in the target variable (next_month_default). To address this, the **Synthetic Minority Oversampling Technique (SMOTE)** was applied:

- **Purpose**:
  SMOTE synthetically generates new instances of the minority class (defaults) to balance the dataset.

- **Outcome**:
  The target classes were **perfectly balanced**, ensuring fair model training and reducing the risk of bias toward the majority class.

Class	**Count After SMOTE**
0 (No Default)	20,440
1 (Default)	20,440

5.4 Final Training Dataset Structure

After imputation and SMOTE oversampling, the **training dataset shape** is as follows:

Dataset	**Shape**
Oversampled Training Set	40,880 rows, 22 columns

---

6. Data Splitting

To evaluate the performance of the classification models and prevent overfitting, the balanced and pre-processed dataset was divided into **training** and **validation** sets.

6.1 Splitting Strategy

- The dataset after SMOTE application was split into:
  - **80% for Training** the model
  - **20% for Validation** (model performance evaluation)
- The splitting was performed using **Scikit-learn's train_test_split function**, ensuring:
  - **Stratification on the target variable (next_month_default) —** maintaining the class distribution in both sets.
  - **Random State = 42** for reproducibility of results.

6.2 Dataset Sizes After Splitting

Dataset	**Records**	**Features**
Training Set	32,704	21
Validation Set	8,176	21

6.3 Summary

Step	**Description**
Target Variable	next_month_default
Train/Validation Split	80% / 20%
Stratification Applied	Yes (to preserve class distribution)
Random State	42 (for reproducibility)

Training Set Shape	(32,704 records, 21 features)
Validation Set Shape	(8,176 records, 21 features)

7. Model Training

7.1 Models Selected

To predict the probability of a customer defaulting on their credit card payment next month, four diverse machine learning algorithms were selected for model training:

Model	**Description**
Logistic Regression	A baseline linear classifier suitable for binary classification tasks.
Decision Tree Classifier	A non-linear model capable of capturing complex relationships and interactions.
XGBoost Classifier	A powerful ensemble technique based on gradient boosting, effective for handling tabular data with high performance.
LightGBM Classifier	A gradient boosting framework known for its efficiency and fast training on large datasets.

7.2 Model Initialization and Hyperparameters

Model	**Key Parameters**
Logistic Regression	solver='liblinear', max_iter=1000 (to ensure convergence)
Decision Tree	Default settings
XGBoost	Default settings
LightGBM	Default settings

7.3 Model Training Process

- Each model was trained using the **training set (80%)** obtained after SMOTE oversampling and data splitting.
- The **training process completed successfully** for all models without errors.

8. Model Optimization (Hyperparameter Tuning)

8.1 Objective of Hyperparameter Tuning

Hyperparameter tuning was performed to enhance each model's performance by identifying the optimal set of parameters that balance bias and variance, thereby improving generalization to unseen data.

8.2 Approach Used

- **Technique**:
  RandomizedSearchCV with 5-fold cross-validation was utilized for all models.
  This approach allows a broad exploration of hyperparameter space within a feasible computation time.

- **Evaluation Metrics**:
    - **F1-Score (primary metric for refit)**: balances precision and recall, crucial for imbalanced datasets like default prediction.
    - **ROC-AUC Score**: evaluates the model's ability to distinguish between the two classes.

8.3 Hyperparameter Grids Defined

Model	Parameters Tuned
Logistic Regression	C (Regularization strength), penalty (l1, l2)
Decision Tree	max_depth, min_samples_split, min_samples_leaf
XGBoost	n_estimators, learning_rate, max_depth, subsample, colsample_bytree
LightGBM	n_estimators, learning_rate, num_leaves, subsample, min_child_samples, reg_alpha, reg_lambda

- **Search Iterations**:
    - **Logistic Regression**: 10 iterations (faster due to simpler model)
    - **Decision Tree, XGBoost, LightGBM**: 50 iterations each (due to larger parameter space)

8.4 RandomizedSearchCV Configuration

- **Cross-Validation**: 5-fold
- **Number of Parameter Settings Sampled**:
    - Logistic Regression: 10
    - Others: 50
- **Scoring Metrics**:
    - Primary: **F1-Score** (refit='f1')
    - Secondary: **ROC-AUC**
- **Parallel Processing**: n_jobs = -1 to utilize all CPU cores for faster execution.
- **Random State**: 42 (ensures reproducibility)

9. Model Evaluation and Comparison

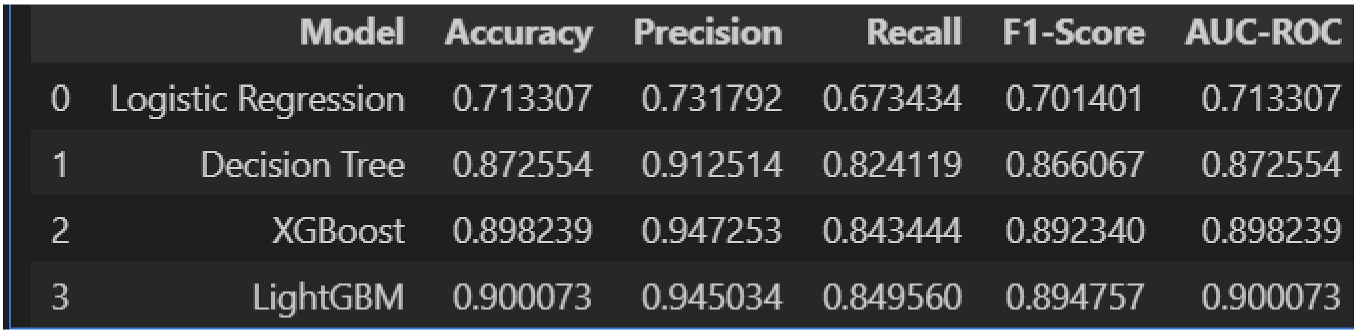
9.1 Evaluation Metrics Used

To assess the predictive performance of the trained models on the **validation set**, the following standard evaluation metrics were computed:

METRIC	DESCRIPTION
ACCURACY	Overall proportion of correct predictions (both default and no-default cases).
PRECISION	Proportion of positive (default) predictions that were actually correct.
RECALL	Proportion of actual defaults that were correctly predicted (Sensitivity).

F1-SCORE	Harmonic mean of Precision and Recall — balances False Positives and False Negatives.
AUC-ROC	Area Under the Receiver Operating Characteristic Curve — measures class separation capability.

9.2 Model Performance Summary



	Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
0	Logistic Regression	0.713307	0.731792	0.673434	0.701401	0.713307
1	Decision Tree	0.872554	0.912514	0.824119	0.866067	0.872554
2	XGBoost	0.898239	0.947253	0.843444	0.892340	0.898239
3	LightGBM	0.900073	0.945034	0.849560	0.894757	0.900073

9.3 Insights from Model Comparison

- **Logistic Regression** showed the lowest performance across all metrics, serving as a simple linear baseline.
- **Decision Tree** significantly improved over Logistic Regression but was outperformed by ensemble models.
- **XGBoost and LightGBM** performed the best, with **LightGBM slightly outperforming XGBoost** in terms of **Accuracy (90.01%)**, **F1-Score (0.8948)**, and **AUC-ROC (0.9001)**.
- **LightGBM** demonstrated the best balance of **precision, recall, and robustness**, making it the most suitable candidate for deployment in predicting credit card default risk.

9.4 Final Model Selection

Based on the evaluation metrics, **LightGBM** is selected as the **best-performing model** for the final prediction task.

10. Threshold Optimization (F2-Score)

10.1 Objective

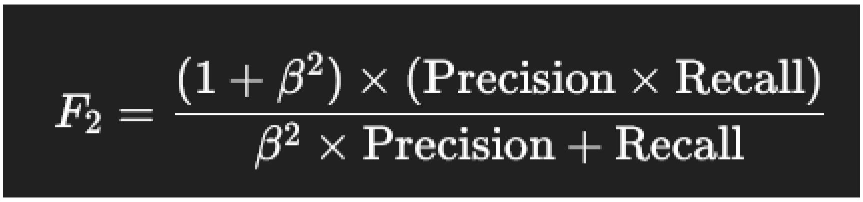
After selecting **LightGBM as the best model**, further optimization was carried out to **adjust the classification threshold**. This ensures the model's probability output is converted to a class label (0 or 1) in a way that maximizes the **F2-Score** — which gives **higher weight to Recall** compared to Precision.

10.2 Why F2-Score?

- In credit risk analysis, especially for **default prediction**, the cost of missing defaulters (False Negatives) is significantly higher than incorrectly flagging non-defaulters (False Positives).
- Therefore, **Recall (sensitivity)** was prioritized using the **F2-Score**, which penalizes False Negatives more heavily than False Positives.

10.3 Methodology

1. The **predicted probabilities** for the positive class (default = 1) from LightGBM were extracted.
2. A **Precision-Recall curve** was computed to derive multiple threshold points.
3. For each threshold, the corresponding **F2-Score** was calculated using:



$$F_2 = \frac{(1+\beta^2) \times (\text{Precision} \times \text{Recall})}{\beta^2 \times \text{Precision} + \text{Recall}}$$

where **β = 2** (giving Recall more weight).

4. The threshold maximizing the **F2-Score** was selected.

10.4 Optimal Threshold Results (LightGBM)

Metric	Value
Optimal Threshold	0.1870
Best F2-Score	0.8935
Precision at F2	0.7840
Recall at F2	0.9259

10.5 Interpretation

- The optimized threshold of **0.1870** shifts the classifier towards **capturing more defaults (high Recall = 92.59%)**, even at the cost of some decrease in Precision (78.40%).
- This trade-off is suitable for **credit risk modeling**, where **missing a defaulter is more costly than a false alarm**.

---

11. Business Impact Analysis

11.1 Objective

The purpose of the Business Impact Analysis is to **quantify the financial benefits** of deploying the developed LightGBM-based credit default prediction model compared to a scenario with no model in place.

11.2 Assumptions Used in the Analysis

Parameter	Value	Description
Average Loan Amount	₹50,000	Typical loan/credit exposure per customer.
Cost of Default	40% of Loan	Loss incurred when a customer defaults and is not identified (False Negative).

Cost of Investigation	5% of Loan	Operational cost when a non-defaulter is incorrectly flagged (False Positive).

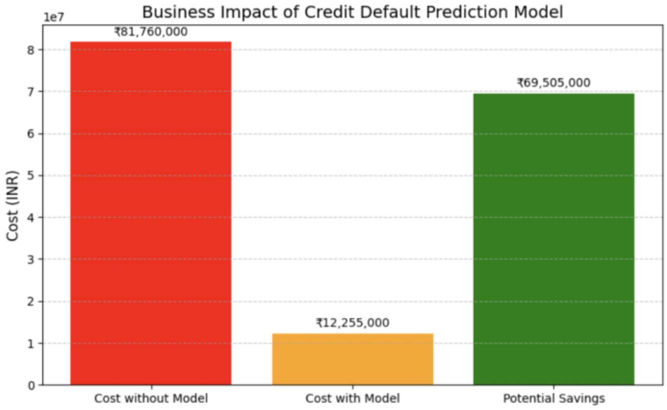
11.3 Confusion Matrix-Based Analysis

The model's performance was translated into real-world financial impact by computing costs associated with:

- **False Negatives (FN)** — Defaults that the model failed to detect.
- **False Positives (FP)** — Customers wrongly classified as high-risk, leading to unnecessary investigation costs.

11.4 Results Summary

Metric	Value
Total Cost with Model	₹12,255,000
Cost without Model (No Detection)	₹81,760,000
Potential Savings Using Model	**₹69,505,000**
Cost of False Negatives	₹11,780,000
Cost of False Positives	₹475,000



11.5 Interpretation

- Without the model, **total potential loss** due to customer defaults would have been ₹81.76 million.
- By using the LightGBM model:
  - The **total cost was reduced to ₹12.25 million** — a significant decrease.
  - This resulted in a **potential savings of ₹69.50 million**.
- Most of the remaining cost is from **False Negatives (missed defaulters)**, but the model successfully reduced this risk compared to no prediction mechanism.
- The relatively low **cost of False Positives (₹0.475 million)** indicates that operational investigations remain affordable and manageable.

11.6 Business Conclusion

- **Deploying the model is financially beneficial**, reducing credit risk exposure substantially.

- The model enables **early detection of potential defaulters**, offering banks an opportunity to take **preventive risk-mitigation actions**, such as credit limit adjustment or customer outreach.
