IIT ROORKEE

CREDIT CARD DEFAULT PREDICTION

Using Machine Learning

PRESENTED TO

Finance Club IIT R

PRESENTED BY

Suraj Singh Chahar
23410034
GT

TABLE OF CONTENTS

INTRODUCTION

Credit card default prediction is a critical challenge faced by financial institutions worldwide. The ability to accurately identify customers who are likely to default on their credit card payments in the upcoming billing cycle enables banks to proactively manage credit risk, reduce potential financial losses, and optimize their lending strategies. Given the highly competitive nature of the credit market and the significant financial exposure associated with unsecured lending, robust risk assessment tools are essential for sustainable banking operations.

The objective of this project is to build a financially interpretable classification model that not only flags potential defaulters in advance but also helps the bank understand default patterns and take timely risk-based actions. The dataset provided consists of anonymized historical records from over 30,000 credit card customers, capturing a wide range of demographic and behavioral variables. These include customer age, gender, education, marital status, credit limit, monthly bill amounts, repayment amounts, and detailed payment status indicators over the past six months. The target variable, next_month_default, indicates whether the customer defaulted on their payment in the following billing cycle.

What sets this project apart is its emphasis on financial interpretability in addition to predictive accuracy. The goal is not only to flag potential defaulters in advance but also to uncover underlying behavioral patterns and financial signals that drive default risk. This dual focus empowers the bank to take early, data-driven actions such as adjusting credit exposure, triggering early warning systems, and prioritizing risk-based interventions.

DATA DESCRIPTION

The dataset used for this project contains anonymized historical behavioral and demographic information for over 30,000 credit card customers provided by the Finance Club IIT R. Its primary purpose is to enable the prediction of whether a customer will default on their credit card payment in the next billing cycle. The data is structured to provide a comprehensive view of each customer's credit profile, spending, and repayment behavior over the past six months.

Key Features:

- Customer_ID: Unique identifier for each customer.
- Demographic Variables:
  - marriage: Marital status (1 = Married, 2 = Single, 3 = Others).
  - sex: Gender (1 = Male, 0 = Female).
  - education: Education level (1 = Graduate School, 2 = University, 3 = High School, 4 = Others).
  - age: Age in years.
- Credit Profile:
  - LIMIT_BAL: Credit limit assigned to the customer (in currency units).
- Behavioral Variables (Tracked Monthly):
  - pay_0 to pay_6: Payment status for the current and previous six months. Values indicate:
    - -2: No credit consumption (no bill) in that month
    - -1: Bill generated and fully paid in the same month
    - 0: Partial or minimum payment made (revolving credit)
    - ≥1: Payment overdue by that many months (e.g., 1 = 1 month overdue)
  - Bill_amt1 to Bill_amt6: Total bill amount at the end of each month. Positive values indicate outstanding amounts, zero indicates no spending, and negative values mean the customer overpaid.
  - Pay_amt1 to Pay_amt6: Payment amount made in each month towards the previous month’s bill.

DATA DESCRIPTION

- Engineered Features:
  - AVG_Bill_amt: Average bill amount across the six months.
  - PAY_TO_BILL_ratio: Ratio of total payments to total bill amounts over six months, indicating overall repayment behavior21.
- Target Variable:
  - next_month_default: Binary indicator of default in the next billing cycle (1 = Default, 0 = No Default).

Dataset Structure:

- Training Set: Approximately 25,000 records, including all features and the target variable.
- Validation Set (depicted as test): Approximately 5,000 records, containing the same features but without the target variable. The model is required to predict default status for these customers.

This rich dataset enables both predictive modeling and financial interpretation, supporting the goal of not only identifying likely defaulters but also understanding the behavioral drivers of credit risk.

METHODOLOGY AND WORKFLOW

The objective of this project was to build a predictive classification model that identifies credit card customers likely to default in the next billing cycle. The modeling pipeline was designed to be both data-driven and financially interpretable. The methodology followed is outlined below:

Key Steps in Workflow:

1. Data Loading and Understanding
   - Imported and explored the training dataset (train_dataset_final1.csv) in the beginning and looked into the basic details which were given by .info() and .describe().
   - Reviewed variable descriptions and clarified the temporal meaning of variables like pay_m, bill_amt_m, and pay_amt_m.

2. Exploratory Data Analysis (EDA)
   - Univariate Analysis: Investigated distribution of key variables such as LIMIT_BAL, AGE, and payment status variables using histograms and boxplots.
   - Bivariate Analysis: Explored relationships between default status and categorical variables (e.g., sex, education, marriage) using count plots and bar charts and also combined them for gaining meaningful insights.
   - Behavioral Trend Analysis:
     - Analyzed repayment consistency, payment delays, and credit utilization behavior over 6 months.
     - Analyzed derived financial indicators like average bill amount and total payments and their relationship with each other.
   - Correlation with Target variable and Multicollinearity: Explored the collinearlity between columns and target variable and aslo checked for multicollinearity with VIF numbers.

METHODOLOGY AND WORKFLOW

3. Feature Engineering:

- Created features such as:
  - Total Payment Delays.
  - Age_groups and bins.
  - Delinquency streak indicators.
  - Credit utilization measures.
- Applied label encoding to categorical variables(Only age_groups was required to be encoded).

4. Handling Class Imbalance:

- Since the dataset showed imbalance (~20% defaulters), applied SMOTE (Synthetic Minority Oversampling Technique) on the training data inside the RandomSearchCV to balance classes.
- Also experimented with class_weight='balanced' for some models as a comparative approach.

5. Model Building and Evaluation:

- Trained and evaluated multiple classification models:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - XGBoost (final model chosen)
  - LightGBM
- Used StratifiedKFold Cross-Validation for reliable evaluation.
- Recall, precision, F1, accuracy, confusion matrix, AUC-ROC and Calibration Curve were computed for comparison.
- Primary evaluation metric: F2-score, to prioritize minimizing false negatives (missed defaulters).

METHODOLOGY AND WORKFLOW

6. Feature Importance Analysis:

- Identified key drivers of default (e.g., payment delays, credit utilization) using feature_importances_ .

7. Threshold Optimization

- The default classification threshold of 0.5 was not optimal for imbalanced data.
- Threshold was tuned based on F2-score optimization from precision-recall outputs. The final chosen threshold maximized F2, aligning with the business need to catch as many defaulters as possible as catching a defaulter is more important in the world of crdit default than flagging a non defafulter.

8. Final Prediction and Output:

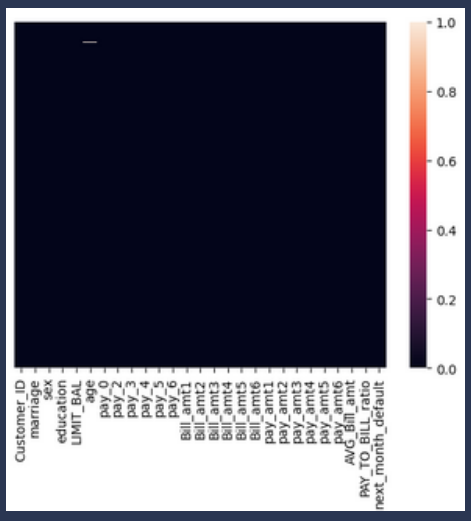
- Best model was used to predict next_month_default on the unlabeled test dataset.
- Predictions were generated using the optimized threshold and exported as a CSV file with Customer_ID and predicted next_month_default.

UNDERSTANDING BASIC STRUCTURE AND DATA CLEANING

First loaded the dataset (train_dataset_final1.csv) and went for .describe() and .info() which provide these basic results:

```
RangeIndex: 25247 entries, 0 to 25246
Data columns (total 27 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Customer_ID         25247 non-null  int64
 1   marriage            25247 non-null  int64
 2   sex                 25247 non-null  int64
 3   education           25247 non-null  int64
 4   LIMIT_BAL           25247 non-null  int64
 5   age                 25121 non-null  float64
 6   pay_0               25247 non-null  int64
 7   pay_2               25247 non-null  int64
 8   pay_3               25247 non-null  int64
 9   pay_4               25247 non-null  int64
 10  pay_5               25247 non-null  int64
 11  pay_6               25247 non-null  int64
 12  Bill_amt1           25247 non-null  float64
 13  Bill_amt2           25247 non-null  float64
 14  Bill_amt3           25247 non-null  float64
 15  Bill_amt4           25247 non-null  float64
 16  Bill_amt5           25247 non-null  float64
 17  Bill_amt6           25247 non-null  float64
 18  pay_amt1            25247 non-null  float64
 19  pay_amt2            25247 non-null  float64
 20  pay_amt3            25247 non-null  float64
 21  pay_amt4            25247 non-null  float64
 22  pay_amt5            25247 non-null  float64
 23  pay_amt6            25247 non-null  float64
 24  AVG_Bill_amt        25247 non-null  float64
 25  PAY_TO_BILL_ratio   25247 non-null  float64
 26  next_month_default  25247 non-null  int64
dtypes: float64(15), int64(12)
```

- We see that only two datatypes are present both numerical ,this means that categorical variables like marriage, sex have been encoded already, also we see that some values are missing in the age column.
- To check for other missing values I made a heatmap of missing values for whole dataset .



- After checking the data, I found that the only missing values were in the AGE column. There were 126 missing entries out of a total of 25,247, which is just about 0.5% of the data.
- Since this is a very small amount, removing these rows won't affect the overall analysis. Also, because age is an important personal detail that's hard to guess correctly, it's better to remove the rows rather than risk adding inaccurate values.
- This approach ensures data integrity while having a negligible impact on the overall dataset size and statistical representativeness.

UNDERSTANDING BASIC STRUCTURE AND DATA CLEANING

The .describe() function revealed that there were some anamolies in the marriage and education columns.

	marriage	education
count	25121.000000	25121.000000
mean	1.551769	1.851996
std	0.522538	0.797132
min	0.000000	0.000000
25%	1.000000	1.000000
50%	2.000000	2.000000
75%	2.000000	2.000000
max	3.000000	6.000000

- The min value of marriage came out to be 0 while the data description mentioned presence of only 1, 2, 3 values.
- Since these entries likely represent unspecified or incorrect data, and because their count was small (only 53), I treated them as 'Others' by mapping 0 to 3. This approach preserves useful records without introducing bias or affecting model performance.

```
df.loc[(df['marriage'] == 0), "marriage"] = 3
```

- While reviewing the education column, I found a few invalid values (e.g., 0, 5, 6) that were not part of the defined categories (1 = Graduate School, 2 = University, 3 = High School, 4 = Others).
- Since these entries likely represent misclassified or missing data, and their count was low (about 1.1% of the dataset), I grouped them under 'Others' by mapping all values outside 1–4 to category 4.
- This ensured that the data remained clean without needing to remove any useful entries.

```
df.loc[(df['education'] > 4) | (df['education'] == 0), "education"] = 4
```

- No duplicates were present in the dataset as seen by :

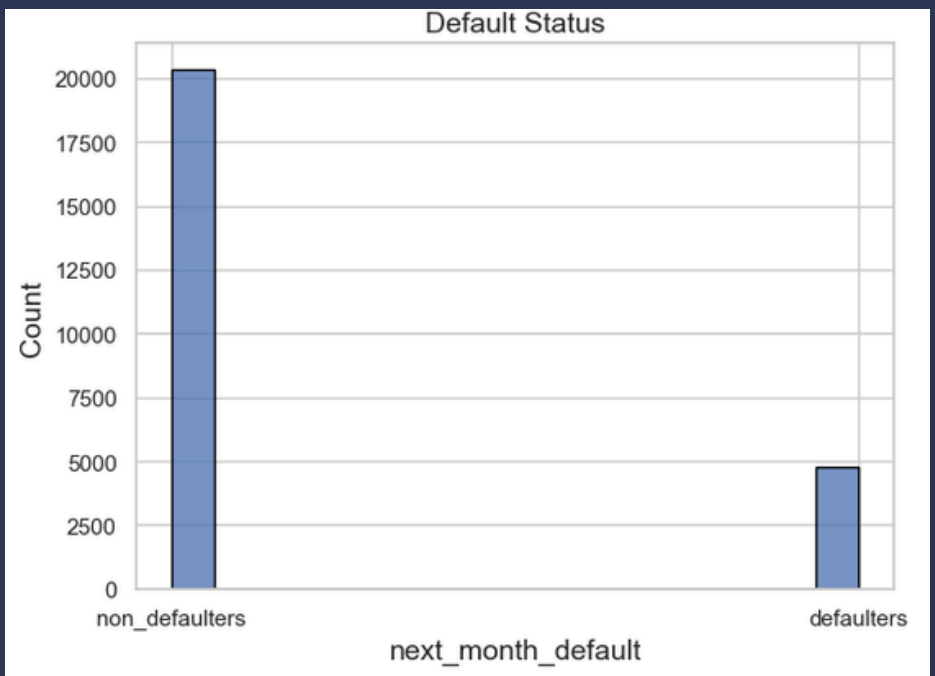
```
df.duplicated().sum()
```

```
0
```

EXPLORATORY DATA ANALYSIS (EDA)

The EDA phase was designed not only to understand the dataset but also to extract financial and behavioral insights that could improve model performance and interpretability. The following structured approach was followed:

1. Understanding the Target variable (next_month_default):
   A simple histplot revealed the following results




```
print(((len(df[df.next_month_default == 1])))/(len(df)))
```

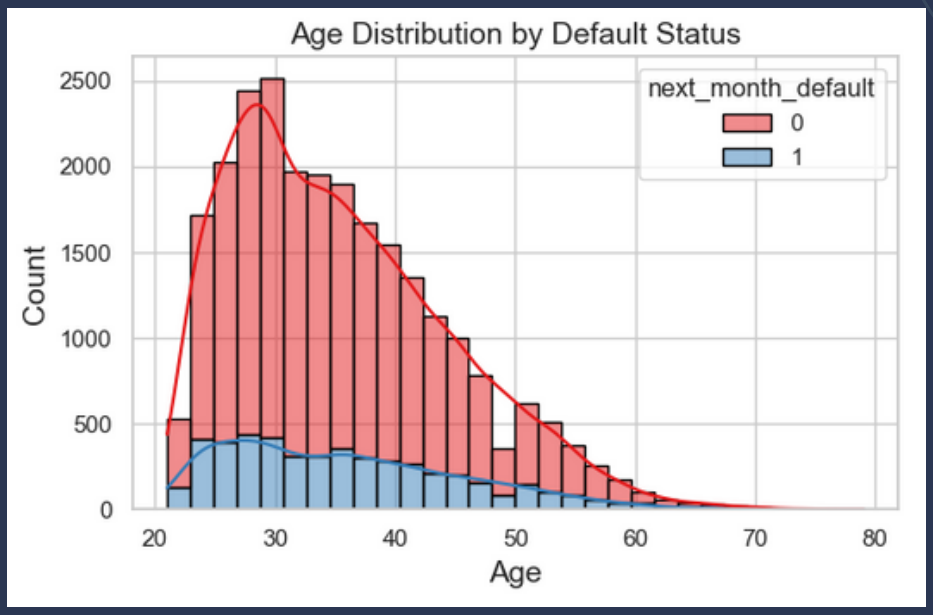
```
0.19043827873094224
```

The target variable next_month_default is highly imbalanced, with only ~19% of customers defaulting. This imbalance can mislead standard classifiers, which may focus on predicting the majority class ('no default') to achieve high accuracy. To address this, I used SMOTE during training and evaluated models using the F2-score — a metric that prioritizes recall which will be explained further down in report. This ensured the model focused on identifying defaulters, which is critical for a risk-sensitive financial application. I also optimized the classification threshold based on F2-score to further align predictions with business needs.

EXPLORATORY DATA ANALYSIS (EDA)

2. Understanding the demographic variables (age, marriage, sex, education):

- Age : Plots like histplot give us visual understanding of the variable.



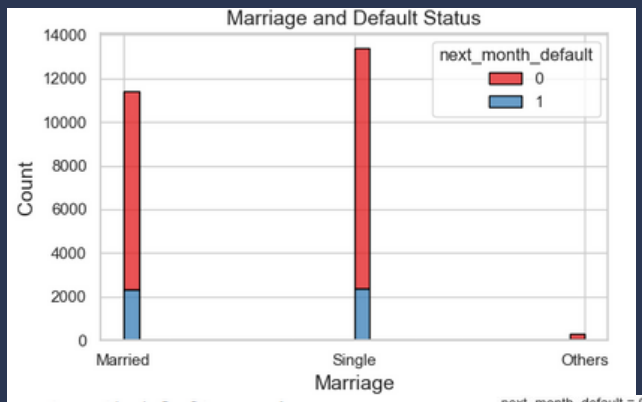
- The majority of both defaulters (next_month_default = 1, blue) and non-defaulters (next_month_default = 0, red) are concentrated in the younger age brackets, particularly between ages 25 and 40. However, the proportion of defaulters appears relatively higher among younger individuals compared to older age groups.
- As age increases, the number of both defaulters and non-defaulters decreases, but the decline is sharper for defaulters. Very few defaults are observed beyond age 50, indicating that older individuals are less likely to default.
- Age appears to be a significant feature for predicting default risk. Younger customers may pose a higher default risk, while older customers are more likely to be reliable in repayment.

To better capture the relationship between age and default risk, the continuous age variable was transformed into categorical age groups.
The highest concentration of defaulters is among those under 30 years old, suggesting elevated risk in this group.The 30-40 age group shows a moderate default rate.Individuals older than 40 have comparatively lower default rates.
By creating these bins, the model can more effectively capture non-linear relationships and risk patterns associated with age, which might be diluted or overlooked if age were treated as a continuous variable. Additionally, categorical age groups simplify interpretation and enable targeted business strategies tailored to each demographic segment.

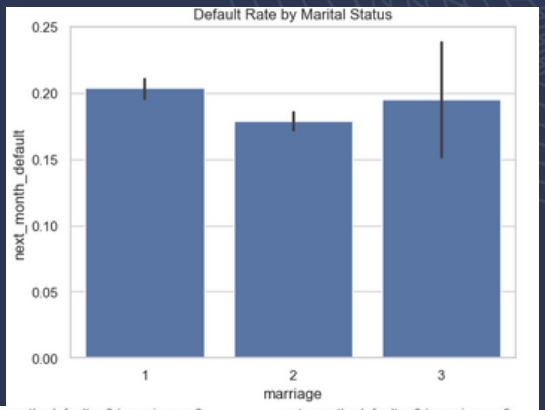
EXPLORATORY DATA ANALYSIS (EDA)

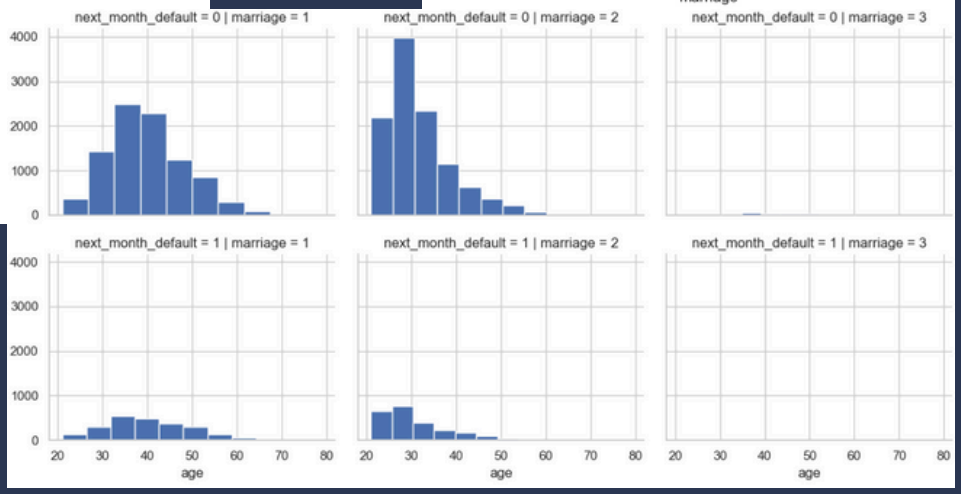
2. Understanding the demographic variables (age, marriage, sex, education):

- Marriage:




```
next_month_default  marriage
0                   2           10982
                    1            9095
                    3             260
1                   2            2392
                    1            2329
                    3              63
Name: count, dtype: int64
```





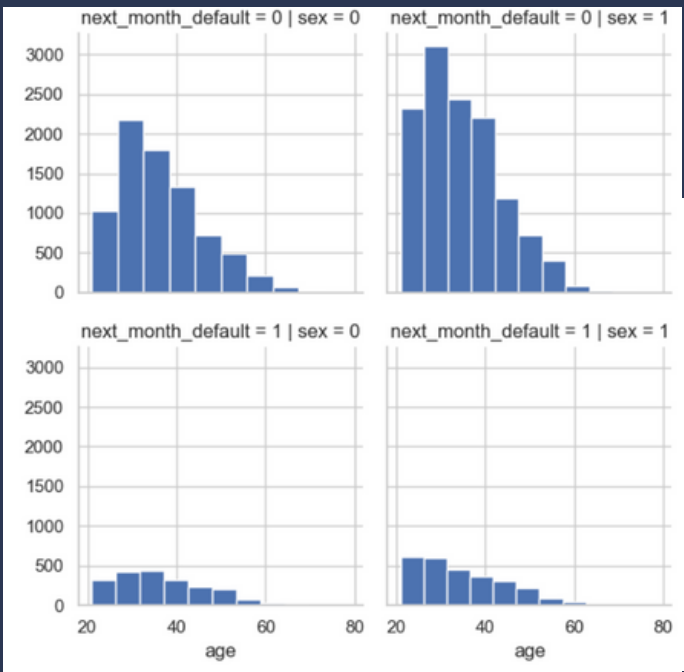
- Singles exhibit the highest default rate at just over 20%, followed by the "Others" group (marriage = 3) at around 19%, and married individuals at approximately 18%. This indicates that singles are more likely to default on payments compared to married customers.
- Across all marital status groups, most defaults occur among younger individuals (ages 25–40), and the number of defaulters sharply declines with increasing age, indicating that younger customers—regardless of marital status—are at higher risk of default.
- The majority of the dataset consists of singles and married individuals, with "Others" representing a very small fraction of the total records. Specifically, there are only 323 in the "Others" category.
- The observed differences in default rates suggest that marital status is a relevant predictor for default risk. Including this feature in the model can help capture behavioral and financial differences associated with different marital statuses.

In summary, marital status shows a clear association with default behavior, making it a valuable feature for both predictive modeling and business decision-making.

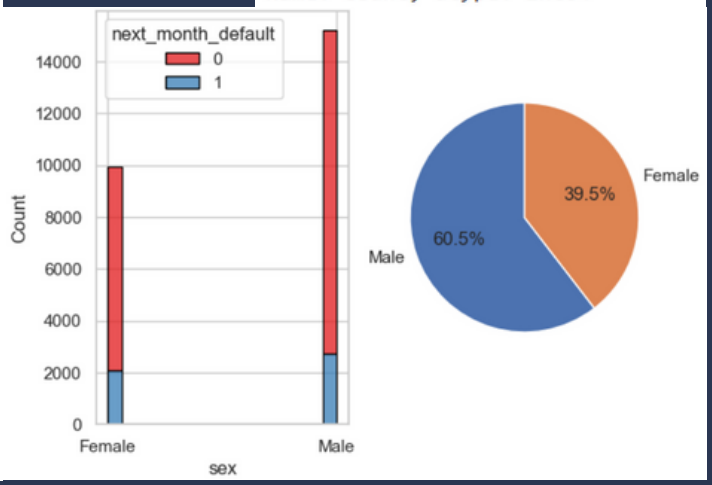
EXPLORATORY DATA ANALYSIS (EDA)

2. Understanding the demographic variables (age, marriage, sex, education):

- Sex:




```
next_month_default  sex
0                   1      12479
                    0       7858
1                   1       2712
                    0       2072
Name: count, dtype: int64
```



- Males make up the majority of the dataset (60.5%), while females account for 39.5%.

- Default Counts and Rates:
- Among males, 2,712 out of 15,191 (17.9%) defaulted.
- Among females, 2,072 out of 9,930 (20.9%) defaulted.
- This indicates that, proportionally, females have a slightly higher default rate than males, even though the absolute number of male defaulters is higher due to their greater representation.

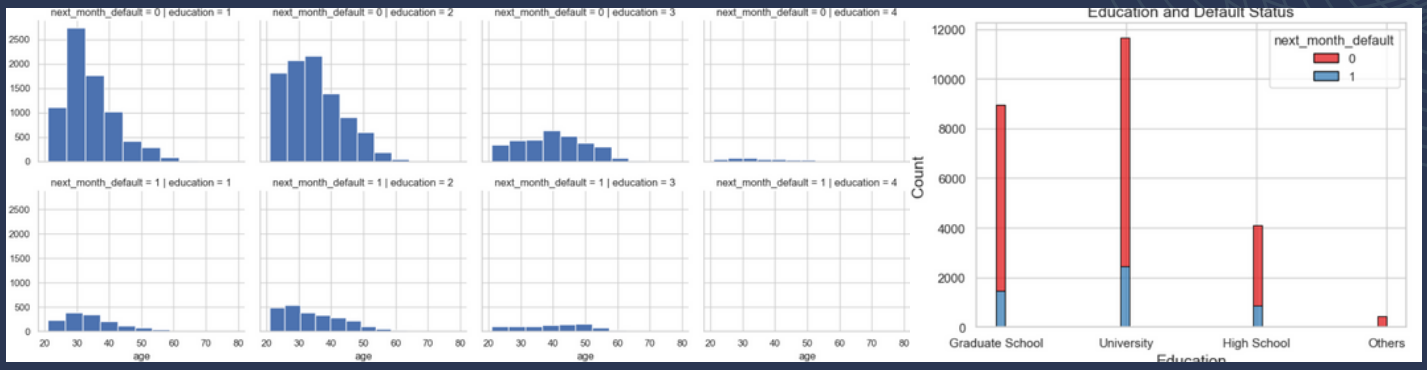
- For both males and females, defaults are most common among younger age groups (especially ages 20–40), and the number of defaulters decreases with increasing age.
- Most of the males defafulting belong to the age bracket of 20-35 while the peak for females comes near 30-35.

- The majority of both males and females are non-defaulters, but the difference in default rates between sexes suggests that gender, in combination with age, may be a relevant predictor for default risk.

EXPLORATORY DATA ANALYSIS (EDA)

2. Understanding the demographic variables (age, marriage, sex, education):

- Education:



- University graduates represent the largest segment in the dataset (11,657 total), followed by graduate school (8,944), high school (4,096), and others (424). This indicates the dataset primarily consists of well-educated individuals.

- Default Rates by Education Level:
  Graduate School: 16.2% default rate (1,449 out of 8,944)
  University: 20.9% default rate (2,438 out of 11,657)
  High School: 21.3% default rate (872 out of 4,096)
  Others: 5.9% default rate (25 out of 424)

- Contrary to expectations, university graduates have a higher default rate than graduate school alumni. This suggests that having a university degree alone doesn't guarantee lower default risk compared to advanced degrees.
- Graduate school graduates demonstrate better payment behavior with the lowest default rate among major education categories, suggesting that advanced education correlates with improved financial stability.
- The age distribution histograms reveal that for all education levels, defaults are concentrated among younger individuals (ages 25-40).
- Education level is a significant predictor of default risk, the data suggests implementing differentiated credit policies based on education level, with particular attention to university graduates who show unexpectedly high default rates despite their education level.

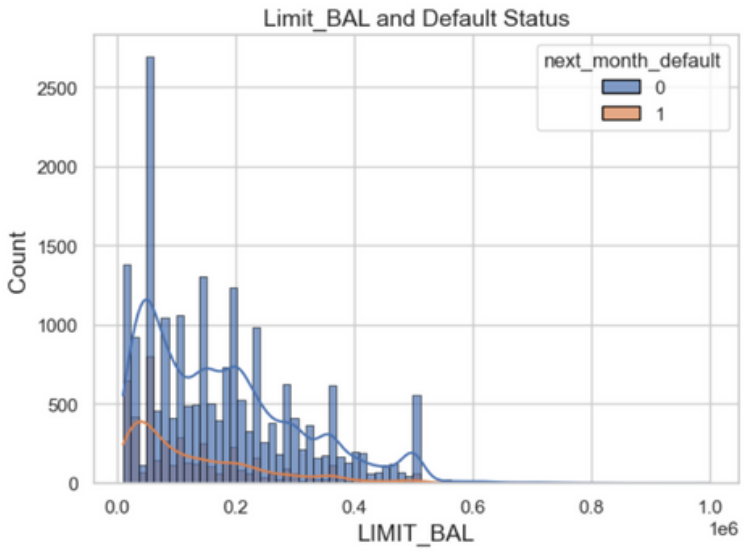
EXPLORATORY DATA ANALYSIS(EDA)

3. Now we try to understand more numerically continous variables like bill_amt, limit_bal, average_bill_amt, pay_to_bill_ratio.

LIMIT_BAL

Key Inferences:

```
Skewness on limit bal is :  0.9948938068300075
```



- The distribution is highly right-skewed, meaning most customers have lower credit limits.
- Very few customers have limits approaching 600,000+.
- Peak defaulter counts are seen around 50,000 to 200,000 credit limit range.As the credit limit increases, the number of defaulters sharply declines.
- Non-defaulters are more evenly spread across the higher limit brackets.This suggests that creditworthy customers tend to be assigned higher limits, or they manage credit better.
- Possible reasons why this is seen:
- Customers with lower credit limits may be more financially constrained, increasing their risk of default.
- Banks may already assign lower limits to riskier customers, which makes LIMIT_BAL both a predictive and policy-driven variable.

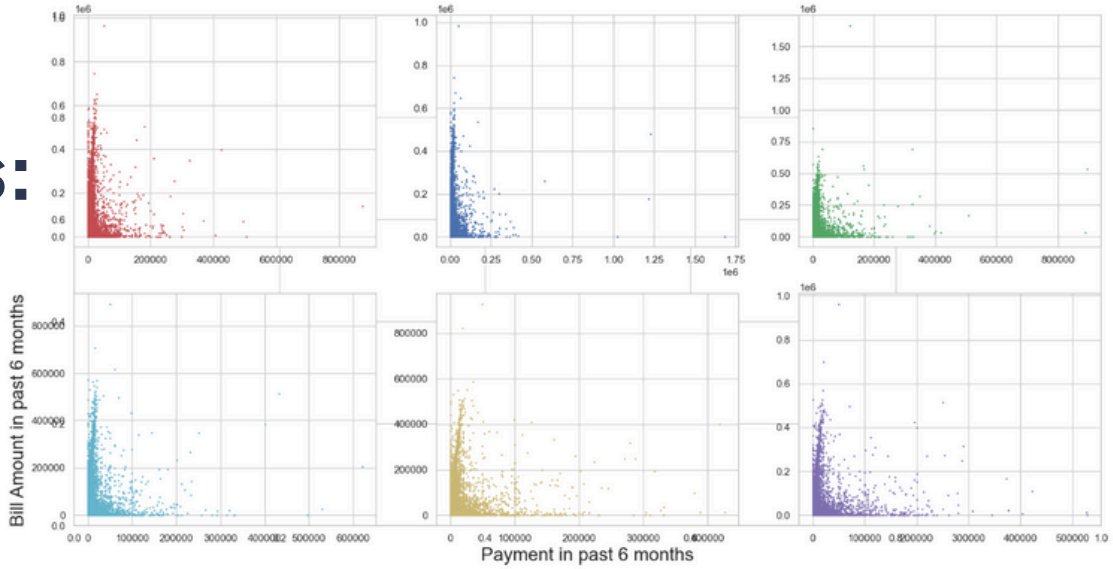
LIMIT_BAL is both a financially meaningful and statistically useful variable. It may reflect a customer's creditworthiness as assessed by the bank, and it plays a key role in predicting default risk.

EXPLORATORY DATA ANALYSIS(EDA)

3. Now we try to understand more numerically continuos variables like bill_amt and pay_amt, limit_bal, average_bill_amt, pay_to_bill_ratio.

pay_m vs Bill_amtm

Key Inferences:



- The scatter plots show a dense cluster of points near the origin, indicating that for the majority of customers, both bill amounts and payments in the past six months are relatively low. There is a clear positive association: higher bill amounts generally correspond to higher payments, suggesting that most customers pay in proportion to their billed amounts
- There is no strong linear correlation — meaning payments don't scale proportionally with bill amounts for many customers.
- Many points lie horizontally far left of the diagonal, this suggests a substantial number of underpayments or minimum payments.
- A few dots extend far right, indicating very high payments, possibly prepayments or repayments of past overdues.
- These may be wealthy or highly disciplined payers — or could represent data reporting anomalies.
- To capture this behavior more effectively in the model, I created features such as the repayment consistency and used the already present PAY_TO_BILL_ratio.

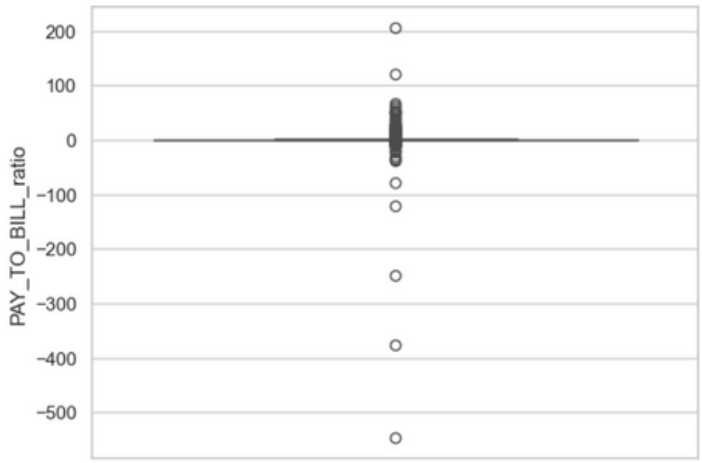
EXPLORATORY DATA ANALYSIS(EDA)

3. Now we try to understand more numerically continuos variables like bill_amt, limit_bal, average_bill_amt, pay_to_bill_ratio.

PAY_TO_BILL_ratio

Key Inferences:

```
count    25121.000000
mean         0.362015
std          5.058260
min       -546.930000
25%          0.040000
50%          0.090000
75%          0.590000
max        205.380000
Name: PAY_TO_BILL_ratio, dtype: float64
```



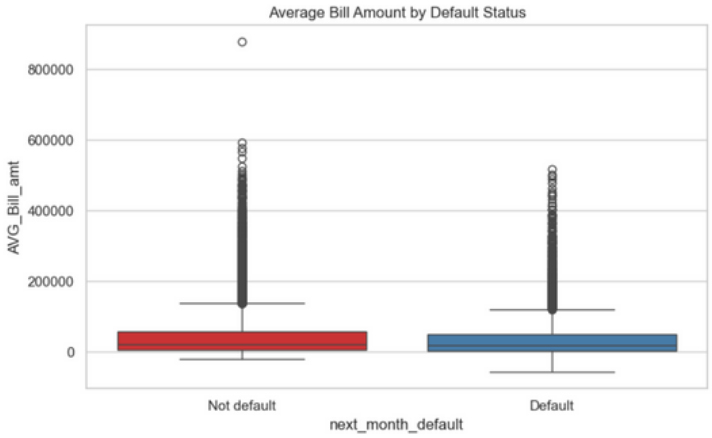
- The PAY_TO_BILL ratio shows extreme variability with a mean of 0.36 and standard deviation of 5.062. The median value (0.09) is much lower than the mean, indicating a heavily right-skewed distribution with significant outliers.
- 50% of customers have a ratio below 0.09, meaning they pay less than 9% of their bill amount, 75% of customers pay less than 59% of their bill amount. This suggests many customers carry forward balances, which is typical credit card behavior.
- Maximum ratio of 205.38 indicates some customers pay more than twice their bill amount. Minimum ratio of -546.93 suggests unusual scenarios like refunds, credits, or data errors.
- Customers with ratios near 1.0 are paying their full bill amount. Ratios above 1.0 could indicate overpayments, advance payments, or account credits. Negative ratios may represent refunds, disputed charges, or data quality issues.
- XGBoost and LightGBM are more robust to outliers than linear models, the extreme nature of PAY_TO_BILL ratios suggests that some preprocessing or at least performance monitoring would be beneficial to ensure optimal model performance(I did create a seperate feature called binary bins to classify them ratios just like age_bins but turns out the calibration curve and roc showed a reduced performance so finally removed it and kept the feature space simple.)

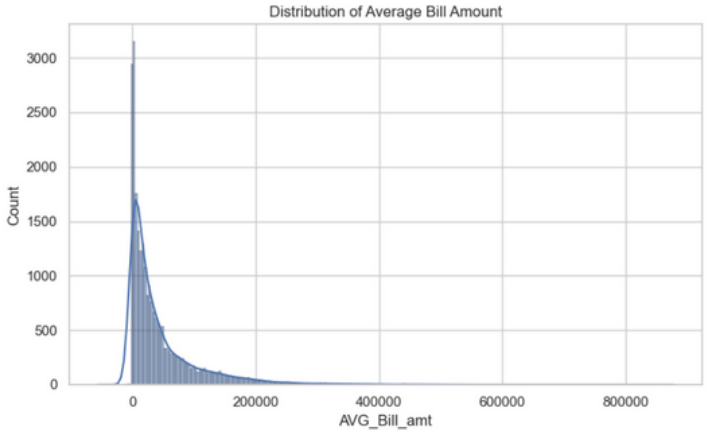
EXPLORATORY DATA ANALYSIS(EDA)

3. Now we try to understand more numerically continuos variables like bill_amt, limit_bal, average_bill_amt, pay_to_bill_ratio.

AVG_Bill_amt

Key Inferences:

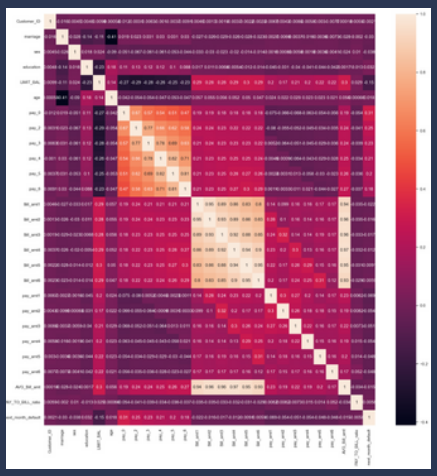




- Histogram reveals a heavily right-skewed distribution of average bill amounts, with the vast majority of customers concentrated at lower bill amounts (near zero) and a long tail extending toward higher values. This indicates that most customers have relatively modest credit usage, while a small subset maintains significantly higher balances.
- Both defaulters and non-defaulters exhibit similar patterns of extreme outliers
- Both groups show comparable median values and interquartile ranges. The boxes (representing 25th to 75th percentiles) appear nearly identical between defaulters and non-defaulters.
- This suggests that average bill amount alone may not be a strong discriminator for default risk, it appears to have limited discriminative power between defaulters and non-defaulters, suggesting that payment behavior and other factors are more critical for default prediction.

EXPLORATORY DATA ANALYSIS (EDA)

4. Understanding the Correlation and multicollinearilty through corr matrix and VIF :



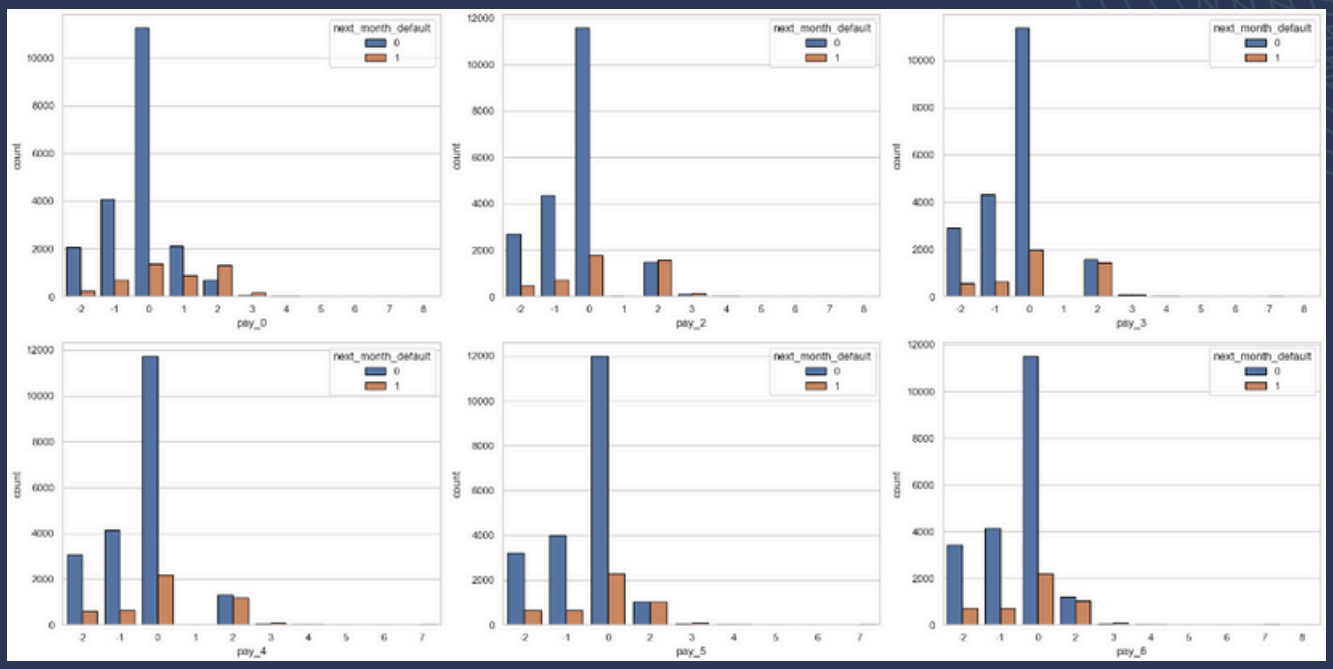
(I tried to put the correaltion mactrix but it is very big ,view it in the code)

	feature	VIF
0	const	58.155685
1	Customer_ID	1.001379
2	marriage	1.224117
3	sex	1.023707
4	education	1.137584
5	LIMIT_BAL	1.560304
6	age	1.277154
7	pay_0	1.898579
8	pay_2	3.170345
9	pay_3	3.635328
10	pay_4	4.238668
11	pay_5	4.670172
12	pay_6	3.194182
13	Bill_amt1	325.522549
14	Bill_amt2	320.437934
15	Bill_amt3	292.992461
16	Bill_amt4	258.622842
17	Bill_amt5	241.804284
18	Bill_amt6	215.872203
19	pay_amt1	1.719684
20	pay_amt2	2.333668
21	pay_amt3	1.767541
22	pay_amt4	1.684441
23	pay_amt5	1.702687
24	pay_amt6	1.172175
25	AVG_Bill_amt	8267.042946
26	PAY_TO_BILL_ratio	1.007332

- AVG_Bill_amt: VIF = 8,267 (extremely high) - this feature caused massive multicollinearity with the individual bill amount features, it is justified as it is derived from the total bill_amtm columns.
- VIF values ranging from 215-325 (all extremely high) - Bill_amt1 through Bill_amt6 are highly correlated with each other, which is expected since they represent consecutive months of billing.
- Payment Status Variables: pay_2 through pay_5 show VIF values between 3-5, indicating moderate correlation among payment delay statuses across months.
- Customer_ID (1.00), marriage (1.22), sex (1.02), education (1.14), and age (1.28) all show low VIF values, indicating minimal multicollinearity.
- I decided to remove the AVG_bill_amt from the feature space due to its low importance and multicollinear factor in prediction.

ADVANCED EXPLORATORY DATA ANALYSIS (EDA)

Understanding the pay_m columns and total_payment_delays :

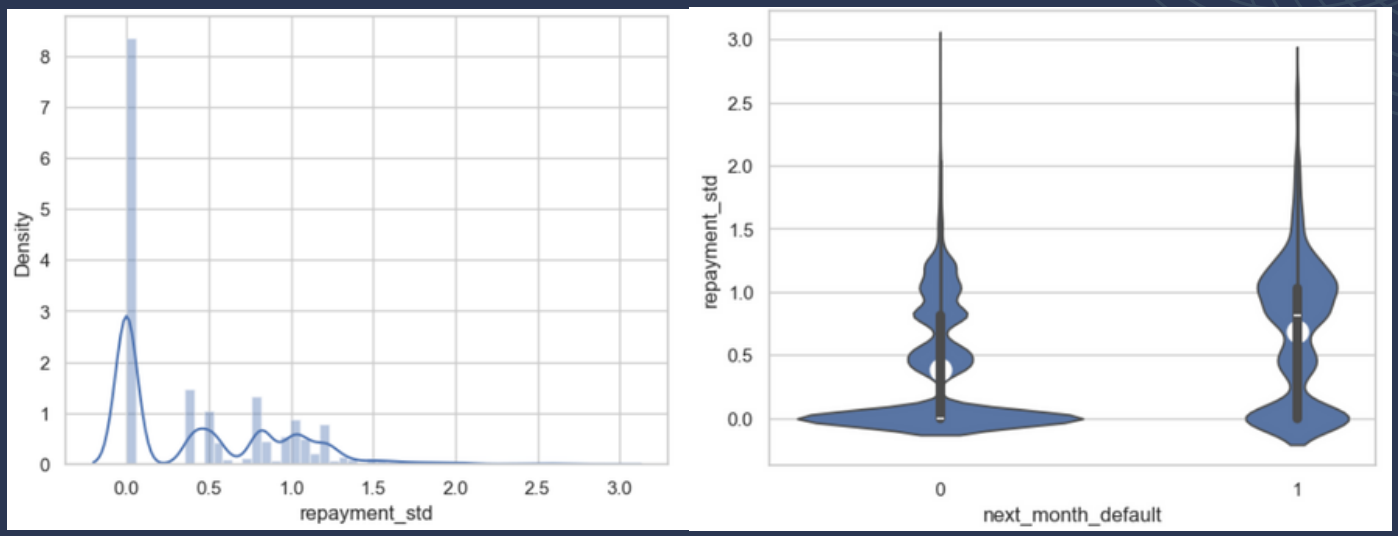



```
df['total_payment_delays'] = df[pay_cols].apply(lambda row: np.sum(row >= 1), axis=1)
```

- Customers who defaulted (next_month_default = 1) are more frequently associated with values ≥1 (delayed payments). In all 6 plots, defaulters have visibly higher relative counts in delay buckets like 1, 2, 3+ months overdue.
- General pattern of peak at 0 and and the count of defaulters having values 0 roughly remains same in every month and so is the value 2 in initial 3 months indicating a delayed payment, the effect significantly dies at value 3.
- The number of people who defaulted is much smaller than those who didn't — evident from the much shorter orange bars. This suggests that we will have to go for SMOTE during the hyperparameter tuning. This will be done further down in the notebook.
- To capture this behavior in a single, interpretable feature, I engineered total_payment_delays, which counts how many of the past six months a customer was late. This feature strongly correlates with default risk and improves both model performance and explainability.
- Not only this but two more features were developed from pay_m columns to explain whether its the same people who are paying consistenly (repayment_std) and how each individuals delays are distributed (delinquency_streak), they will be explained further.

ADVANCED EXPLORATORY DATA ANALYSIS (EDA)

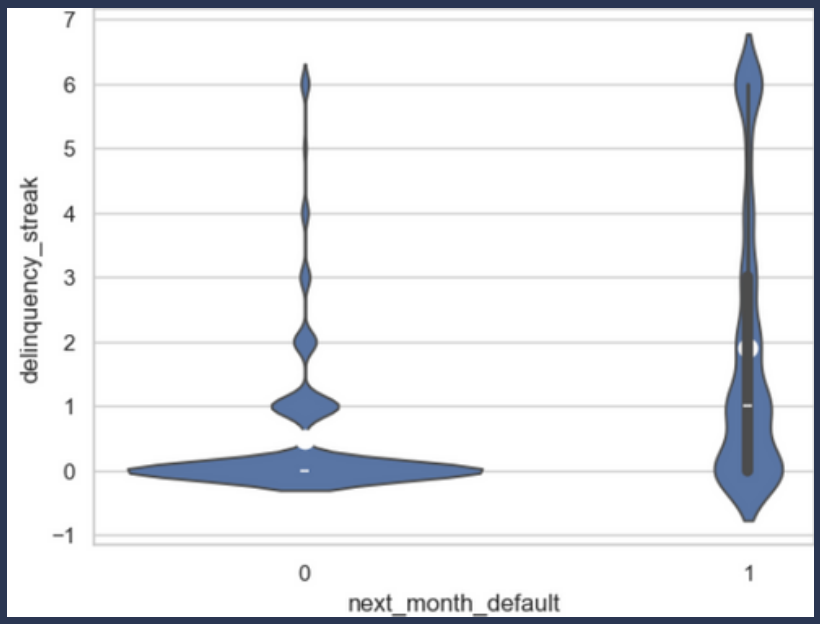
Understanding the repayment_std :



- The majority of customers have a repayment_std near 0, indicating highly consistent repayment behavior over the 6-month period.
- A long tail extends rightward, showing a smaller population with high variability in payment status — customers who alternate between on-time and delayed payments.
- The distribution is right-skewed, which is expected in real-world financial data where erratic behavior is less common but critical.
- Defaulters (next_month_default = 1) tend to have higher repayment_std on average than non-defaulters. Non-defaulters are tightly concentrated near 0 — indicating stable and disciplined payment patterns.
- In contrast, defaulters have a wider spread, suggesting irregular or fluctuating repayment behavior.
- This shows that repayment_std is an effective predictor of financial risk, capturing customers with inconsistent habits that may precede default.

ADVANCED EXPLORATORY DATA ANALYSIS (EDA)

Understanding the delinquency_streak:

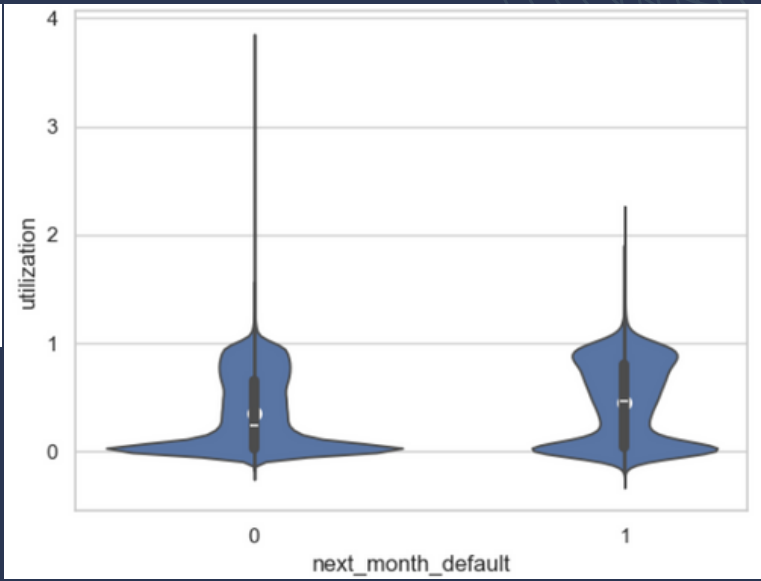


- Non-defaulters (next_month_default = 0) are heavily concentrated at 0, meaning most never had a consecutive overdue period.
- The distribution flattens quickly beyond 1 or 2 months — indicating that prolonged delinquency is rare among reliable customers.
- Defaulters (next_month_default = 1), however, show a broad spread across values 1 through 6, Many have streaks of 2–4 consecutive late months. Some go all the way up to 6, indicating sustained repayment failure.
- There is clear visual separation between the default and non-default groups. This means delinquency_streak is a powerful predictor of future default, as longer streaks are strongly associated with financial distress.

ADVANCED EXPLORATORY DATA ANALYSIS (EDA)

Understanding the utilization:

```
count    25121.000000
mean         0.370417
std          0.349391
min         -0.200154
25%          0.029783
50%          0.280946
75%          0.681765
max          3.759125
Name: utilization, dtype: float64
```



- I engineered the utilization feature to measure how much of a customer's credit limit is used on average (AVG_Bill_amt / LIMIT_BAL). The distribution shows that while most customers maintain moderate usage, defaulters tend to have slightly higher utilization ratios.
- Both defaulters and non-defaulters are centered around 0.2-0.4 utilization, however, defaulters tend to have slightly higher utilization on average — with their distribution skewed toward the right.
- Defaulters have more density around 0.5-0.7, indicating higher credit use.
- Some extreme utilizers (above 1.0) are present in both groups but may indicate risk escalation when combined with other behaviors.
- High utilization often reflects financial strain or credit dependence, especially when consistent over months.
- Credit scoring systems (e.g., FICO) consider utilization above 30% a risk signal — the data reflects a similar pattern. The ratio is simple, interpretable, and highly relevant to real-world credit policy.

SELECTION OF FEATURES/ SCALING AND FORMATION OF TRAIN-TEST SPLITS

Before training the model, I prepared the dataset by selecting appropriate features and standardizing them:

Feature Selection:

I included all relevant features except:

- 'Customer_ID' — a unique identifier with no predictive value.
- 'next_month_default' — the target variable.
- 'AVG_Bill_amt' — excluded due to high correlation with other billing features and no added predictive strength in initial tests.

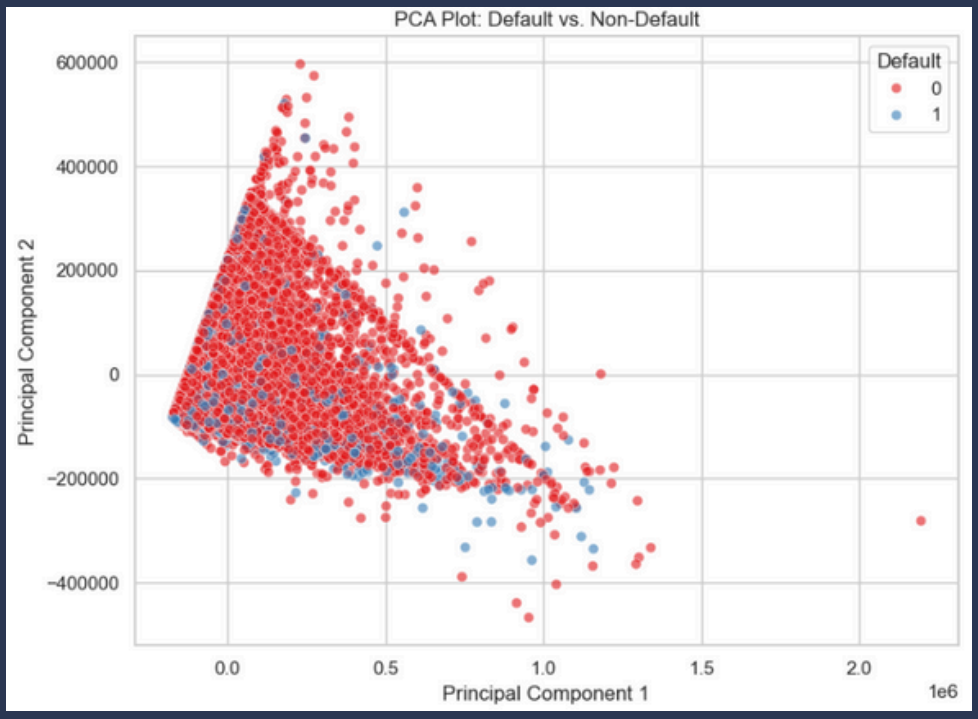
- Action on Outliers :
- While outliers were visualized during exploratory analysis—particularly in features like PAY_TO_BILL_ratio, LIMIT_BAL, and average_bill_amount—they were not removed. This decision was intentional, as the primary models used (XGBoost, LightGBM, Random Forest) are tree-based and inherently robust to outliers.
- Moreover, several engineered features helped normalize behavioral extremes, reducing the potential negative impact of extreme values. Removing outliers in a credit risk context could also eliminate important minority behaviors indicative of default risk, which the model is designed to detect. Therefore, preserving these outliers was deemed more beneficial for model learning and financial interpretability.

- Reason for Ordinal Encoding:
- The age_group feature was encoded using ordinal mapping, with <30 as 0, 30-40 as 1, and >40 as 2. This method was chosen because the age brackets represent a natural order with increasing maturity and generally decreasing default risk. Unlike one-hot encoding, which treats categories as independent, ordinal encoding preserves this inherent hierarchy.
- Tree-based models like XGBoost and Random Forest can leverage such ordinal relationships to create more meaningful splits, improving interpretability and model efficiency without inflating feature space.

SELECTION OF FEATURES/ SCALING AND FORMATION OF TRAIN-TEST SPLITS

Scaling with StandardScaler:
Since the dataset contained variables on vastly different scales (e.g., credit limits in hundreds of thousands vs. binary flags), I applied standardization using StandardScaler. This centers each feature to have zero mean and unit variance, ensuring fair weight distribution across features during model training.

PCA Analysis for a Rough Understanding of classification difficulty:
To visualize how well the features differentiate between defaulters and non-defaulters, I used Principal Component Analysis (PCA) to project the 29-dimensional feature space into two principal components. The resulting scatterplot shows overlap between the two classes, suggesting that default behavior is not easily linearly separable in low-dimensional space. This supports the use of more complex models capable of capturing non-linear relationships in the data.



Train-test-split: the purpose of test dataset created here is solely for hyperparameter tuning.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=42)
# the test dataset will work like a validation dataset here for hypertuning parameters

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

To gain an initial understanding of model performance and identify promising candidates, I trained and evaluated a set of commonly used classification algorithms using their default or basic hyperparameters:

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost
- LightGBM

Each model was trained on the scaled training data (X_train) and evaluated on the test set (X_test) using the following metrics:

1. Accuracy:
- Accuracy was included as a basic benchmark. While it gives a general sense of overall correctness, it can be misleading in imbalanced datasets—such as this one, where non-defaulters are the majority.

2. Confusion Matrix:
- The Confusion Matrix provides a granular view of true positives, true negatives, false positives, and false negatives. It helps to evaluate the cost of errors, especially false negatives, which are critical in credit risk (i.e., failing to flag a defaulter).

3. Classification Report (Precision, Recall, F1-score):
- Precision and Recall help assess the model's ability to correctly identify defaulters.
- F1-score balances precision and recall, offering a more holistic view of model performance than accuracy alone—especially useful when we have to judge both false positives and false negatives.

4. F2-score (to emphasize recall for defaulters):
- F2-score was used to emphasize recall by giving it higher weight. This is aligned with the business goal of minimizing false negatives, i.e., missing actual defaulters, which can be costlier to the bank than flagging a few non-defaulters incorrectly.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

AUC-ROC and calibration curves were also used to evaluate the model's ability to distinguish between classes and assess the reliability of predicted probabilities, respectively.

Using this suite of metrics ensured that models were evaluated not only for correctness, but also for risk sensitivity and business relevance before tuning and threshold adjustment.

This evaluation phase served as a critical benchmarking step, offering quick insights into how each model performs out of the box. It allowed for early identification of strong candidates for further tuning and enabled fair comparisons using balanced, business-aligned metrics.

I prioritized F2-score as the primary evaluation metric to reflect the real-world objective of this project: minimizing false negatives (missed defaulters), which carry higher financial risk than false positives in credit lending scenarios. Models that favored recall without severely compromising precision were considered more viable.

Below are the results of initial training of various models mentioned above and evaluation metrics:

1. Logistic Regression :

```
Logistic Regression Results:
Accuracy: 0.8314427860696517
Confusion Matrix:
 [[3906  162]
 [ 685  272]]
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.96      0.90      4068
           1       0.63      0.28      0.39       957

    accuracy                           0.83      5025
   macro avg       0.74      0.62      0.65      5025
weighted avg       0.81      0.83      0.80      5025

F2 Score:
 0.31909901454716094
```

While Logistic Regression performs well in separating non-defaulters, it fails to adequately capture high-risk (defaulting) customers, which are the primary concern. Due to its linear nature and limited flexibility, it's not the best fit for this imbalanced, non-linearly separable problem without further tuning or balancing techniques.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

2. Decission Tree :

The Decision Tree classifier achieved an overall accuracy of 74.6%, which is lower than Logistic Regression, but its ability to detect defaulters improved modestly.

```
Decision Tree Results:
Accuracy: 0.7440796019900497
Confusion Matrix:
 [[3375  693]
 [ 593  364]]
Classification Report:
              precision    recall  f1-score   support

           0       0.85      0.83      0.84      4068
           1       0.34      0.38      0.36       957

    accuracy                           0.74      5025
   macro avg       0.60      0.61      0.60      5025
weighted avg       0.75      0.74      0.75      5025

F2 Score:
 0.3725690890481064
```

While the Decision Tree model shows slightly improved recall and F2-score compared to Logistic Regression, it still struggles to accurately identify a large portion of defaulters. Its performance is also sensitive to overfitting, especially without pruning or hyperparameter tuning. This model serves as a baseline, but would benefit significantly from tuning or ensemble techniques like Random Forest.

3. Random Forest :

The Random Forest classifier achieved an accuracy of 83.4%, with improved detection of defaulters, highlighting the benefit of using ensemble methods for credit risk prediction.

```
Random Forest Results:
Accuracy: 0.8338308457711443
Confusion Matrix:
 [[3888  180]
 [ 655  302]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.96      0.90      4068
           1       0.63      0.32      0.42       957

    accuracy                           0.83      5025
   macro avg       0.74      0.64      0.66      5025
weighted avg       0.81      0.83      0.81      5025

F2 Score:
 0.3503480278422274
```

Random Forest shows balanced performance, especially in terms of precision and generalization. However, its recall for defaulters remains low (32%), which limits its standalone usefulness for early warning in credit risk. It performs better than Logistic Regression and Decision Tree in precision, and can be a strong candidate for further hyperparameter tuning or ensemble stacking.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

4. XGBoost

The XGBoost model achieved an accuracy of 83.0%, closely aligning with previous models, but it delivered a notable improvement in defaulter detection, making it one of the better performers in the baseline comparison.

```
XGBoost Results:
Accuracy: 0.8330348258706468
Confusion Matrix:
 [[3856  212]
 [ 627  330]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.95      0.90      4068
           1       0.61      0.34      0.44       957

    accuracy                           0.83      5025
   macro avg       0.73      0.65      0.67      5025
weighted avg       0.81      0.83      0.81      5025

F2 Score:
 0.37757437070938216
```

XGBoost offers a strong combination of recall and precision compared to other baseline models, making it more suitable for credit risk tasks where catching potential defaulters is critical. It benefits from regularization and gradient boosting, helping it generalize better to subtle patterns missed by simpler models. With threshold tuning and SMOTE, this model is a strong candidate for further optimization.

5. LightGBM :

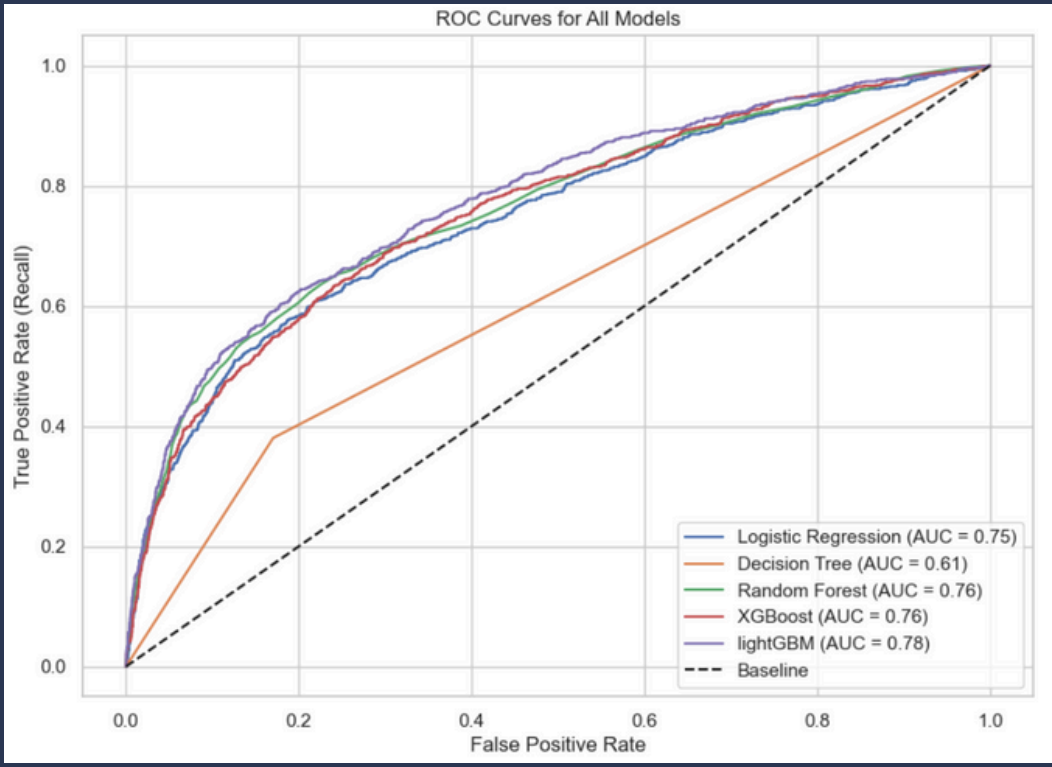
The LightGBM model achieved highest accuracy (83.9%) among all models and demonstrated solid performance in detecting defaulters, making it one of the top candidates for further tuning and deployment.

```
lightGBM Results:
Accuracy: 0.8390049751243781
Confusion Matrix:
 [[3890  178]
 [ 631  326]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.96      0.91      4068
           1       0.65      0.34      0.45       957

    accuracy                           0.84      5025
   macro avg       0.75      0.65      0.68      5025
weighted avg       0.82      0.84      0.82      5025

F2 Score:
 0.376269621421976
```

LightGBM provided the best overall performance among baseline models, with the highest accuracy and competitive F2-score and recall. Its gradient boosting approach, fast training speed, and ability to handle feature interactions make it an excellent candidate for further improvement using threshold tuning, SMOTE, and hyperparameter optimization.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

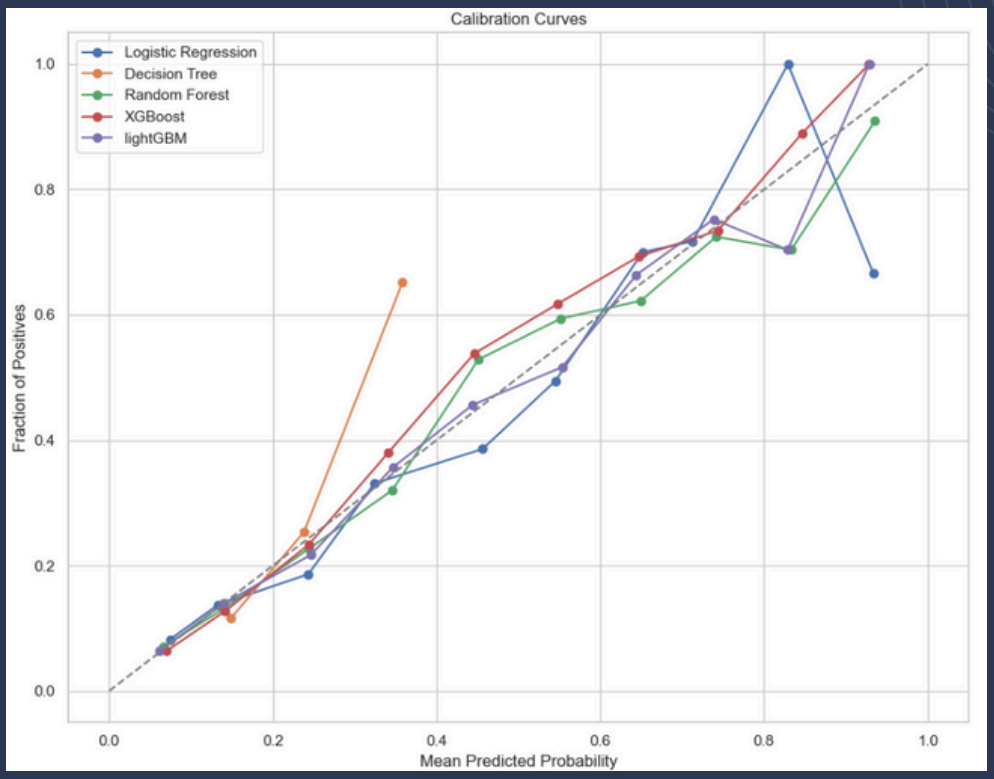
I also used AUC-ROC curve for chossing the models which will be furhter tuned.The ROC (Receiver Operating Characteristic) curve illustrates the trade-off between True Positive Rate (Recall) and False Positive Rate for each model across different classification thresholds. The AUC (Area Under Curve) score summarizes the model's ability to distinguish between defaulters and non-defaulters.



- LightGBM has the highest AUC (0.78), indicating it is the most effective among the models at distinguishing between the two classes.
- XGBoost and Random Forest follow closely at 0.76, reinforcing their strength as ensemble models in imbalanced classification tasks.
- Logistic Regression, though linear, still performs reasonably well with an AUC of 0.75, showing its ability to rank predictions effectively despite lower recall.
- Decision Tree performs the worst (AUC = 0.61), reflecting its tendency to overfit and its limited generalization without tuning.
- All models (except the Decision Tree) perform significantly better than the baseline (dashed diagonal line), which represents random guessing (AUC = 0.5).

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

The use of Calibration Curve:



- XGBoost and LightGBM lie closest to the diagonal across most probability ranges, indicating well-calibrated predictions.
- Random Forest is slightly underconfident at higher predicted probabilities but overall tracks the true probabilities reasonably well.
- Logistic Regression shows overconfidence at high predicted probabilities, predicting extreme default probabilities not reflected in actual outcomes.
- Decision Tree is the most erratic and poorly calibrated, especially beyond the 0.4 range.

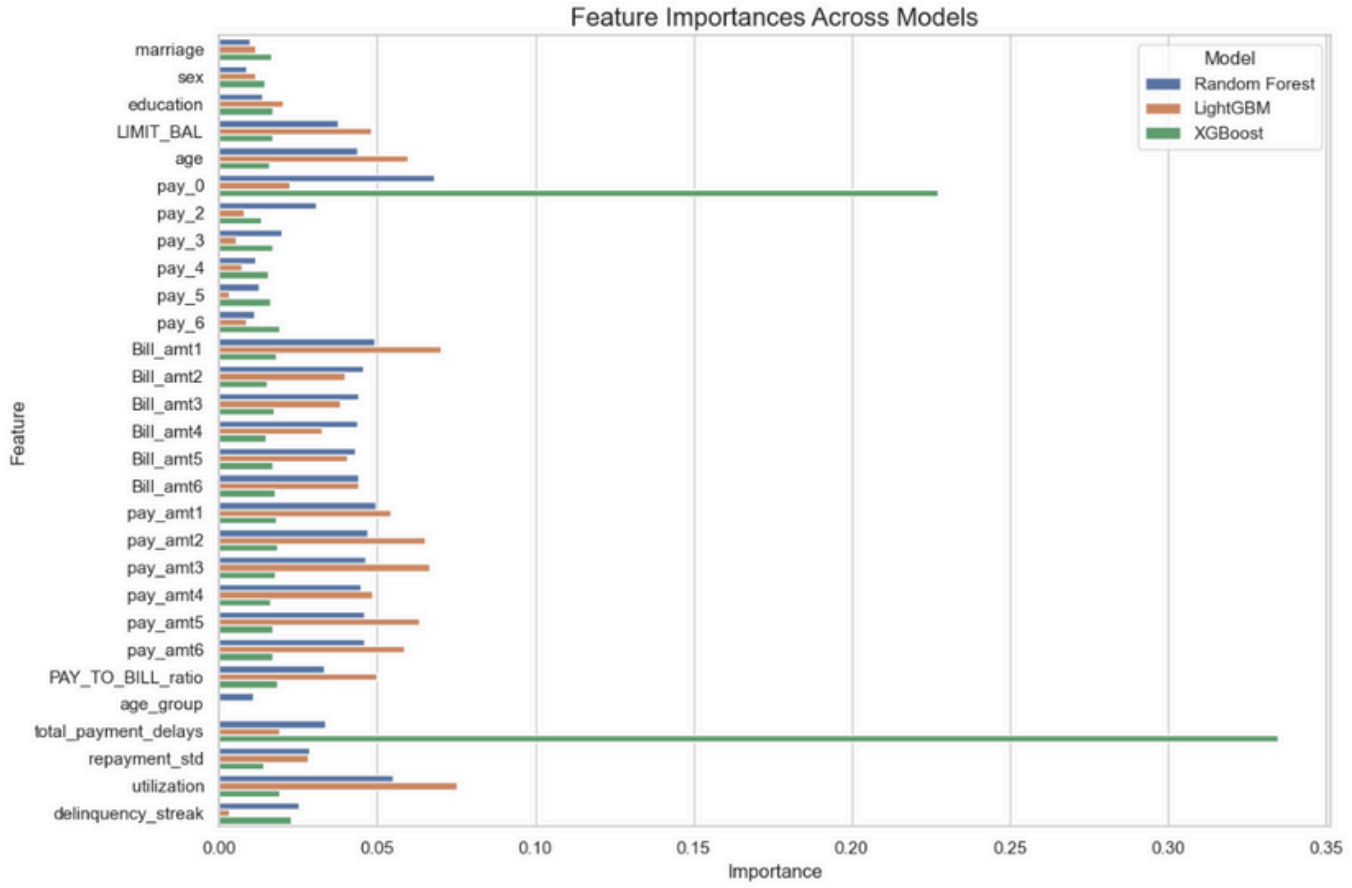
Based on the combined evaluation using F2-score, Recall, AUC-ROC, and Calibration, I selected the following models for further improvement via hyperparameter tuning:

- Random Forest — strong baseline with robust performance and interpretability.
- XGBoost — well-calibrated, high-performing, and proven in imbalanced settings.
- LightGBM — top performer in AUC and calibration, and computationally efficient.

These models demonstrated the best trade-offs between default detection power and reliability of predicted probabilities, aligning well with the project's goal of minimizing missed defaulters while maintaining prediction quality.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

Before applying hyperparameter tuning, I analyzed feature importances from the three selected models. Initially I had gone for using SHAP but it was not able to plot all features as expected and also it was computationally heavy and took very long to form and respond so I decided to stick to feature importances, but for the last selected model SHAP has been used.



All models consistently ranked pay_0, total_payment_delays, and PAY_TO_BILL_ratio among the most important predictors. This confirms that repayment history and financial behavior are the primary drivers of default risk. In contrast, demographic variables like marriage, sex, and education had minimal influence. The analysis also validated the effectiveness of several engineered features, such as repayment_std and delinquency_streak, in enhancing model interpretability and performance.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

To optimize model performance, I used RandomizedSearchCV instead of GridSearchCV for the following reasons:

- Efficiency: Grid search exhaustively tests every parameter combination, which becomes computationally expensive with large grids.
- Diminishing Returns: Beyond a point, many hyperparameter combinations have minimal incremental gain. Random search efficiently explores a wider range of values, which is particularly helpful for tree-based models.
- Suitable for First Pass Tuning: Randomized search is ideal when initially narrowing down good parameter regions.

```
RandomForest_grid = {
    'clf__n_estimators': [100, 200, 300, 500],
    'clf__max_depth': [5, 10, 20, None],
    'clf__min_samples_split': [2, 5, 10],
    'clf__min_samples_leaf': [1, 2, 4],
    'clf__max_features': ['sqrt', 'log2', None],
    'clf__bootstrap': [True, False],
    'clf__class_weight': ['balanced', 'balanced_subsample']
}

lightgbm_grid = {
    'clf__n_estimators': [100, 200, 300],
    'clf__max_depth': [5, 10, -1],
    'clf__learning_rate': [0.01, 0.05, 0.1],
    'clf__num_leaves': [15, 31, 63],
    'clf__subsample': [0.6, 0.8, 1.0],
    'clf__colsample_bytree': [0.6, 0.8, 1.0]
}

XGBoost_grid = {
    'model__n_estimators': [100, 200, 300],
    'model__learning_rate': [0.01, 0.05, 0.1],
    'model__max_depth': [3, 5, 7],
    'model__subsample': [0.7, 0.8],
    'model__colsample_bytree': [0.6, 0.8],
    'model__scale_pos_weight': [1, 3, 5, 7]
}
```

The purpose of important hyperparameters chosen is present in the Jupyter notebook submitted along.

Credit default is a high-risk, low-tolerance problem, so recall-focused metrics (like F2-score) guided parameter ranges.

Parameters like scale_pos_weight and class_weight were chosen to ensure the model doesn't under-predict defaulters, even at the cost of false positives.

min_samples_leaf and num_leaves were tuned to encourage generalizable trees that avoid overfitting rare customer profiles.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

To enhance model performance and better handle class imbalance, I used a pipeline that integrates SMOTE with RandomizedSearchCV for hyperparameter tuning.
The dataset is highly imbalanced, with only ~19% defaulters. To address this:

- I applied SMOTE during cross-validation to synthetically generate samples of the minority class (defaulters) within the training folds.
- This ensures that each training split is balanced, reducing bias toward the majority class and improving the model's ability to learn patterns associated with defaults.
- Integrating SMOTE inside the pipeline (via ImbPipeline) ensures it is only applied to the training folds, not to the entire dataset — preventing data leakage.
- The tuning was done using Stratified K-Fold cross-validation to ensure each fold had a proportional representation of defaulters and non-defaulters.

The evaluation metric was the F2-score, which gives more weight to recall — reflecting the business need to catch as many defaulters as possible.
Now we see that final results of hyperparameter tuning of all models.

1. Random Forest Final Results:

```
Best F2 Score:
 0.5633034450226688
Confusion Matrix:
 [[3186  882]
 [ 364  593]]

Classification Report on Test Set:
              precision    recall  f1-score   support

           0     0.8975    0.7832    0.8364      4068
           1     0.4020    0.6196    0.4877       957

    accuracy                         0.7520      5025
   macro avg     0.6497    0.7014    0.6621      5025
weighted avg     0.8031    0.7520    0.7700      5025

ROC AUC Score: 0.7655
```

- The model correctly identified 593 out of 957 defaulters, achieving a recall of 61.9% — a significant improvement over the initial baseline.
- While precision (40.2%) for class 1 remains moderate, this tradeoff is expected due to the model's tuning with the F2-score, which prioritizes recall.
- The F2-score improved to 0.563, up from a baseline of approximately 0.35, highlighting the positive impact of SMOTE and hyperparameter optimization.
- Performance on the majority class (non-defaulters) remains robust, with a precision of 89.8% and recall of 78.3%.

Overall, the model achieved an accuracy of 75.2% and a ROC AUC Score of 0.766, indicating strong discriminatory ability despite class imbalance.

MODEL BUILDING / EVALUATION / HYPERPARAMETER TUNING

2. LightGBM Final Results :

```
Best F2 Score:
 0.5390962687770262
Confusion Matrix:
 [[3347  721]
 [ 397  560]]

Classification Report on Test Set:
              precision    recall  f1-score   support

           0     0.8940    0.8228    0.8569      4068
           1     0.4372    0.5852    0.5004       957

    accuracy                         0.7775      5025
   macro avg     0.6656    0.7040    0.6787      5025
weighted avg     0.8070    0.7775    0.7890      5025

ROC AUC Score: 0.7619
```

- The model correctly identified 560 out of 957 defaulters, achieving a recall of 58.5% — a solid result given the class imbalance.
- With a precision of 43.7% for defaulters, the model demonstrates reasonable effectiveness in minimizing false positives while still prioritizing recall.
- The F2-score of 0.539 indicates a well-balanced trade-off favoring recall, aligning with the business goal of capturing as many defaulters as possible.
- The model also performs strongly on the majority class, achieving 89.4% precision and 82.3% recall for non-defaulters.
- Overall, it maintains a strong accuracy of 77.8%, a macro average recall of 70.4%, and a ROC AUC of 0.762, reflecting dependable overall discrimination.

3. XGBoost Final Results :

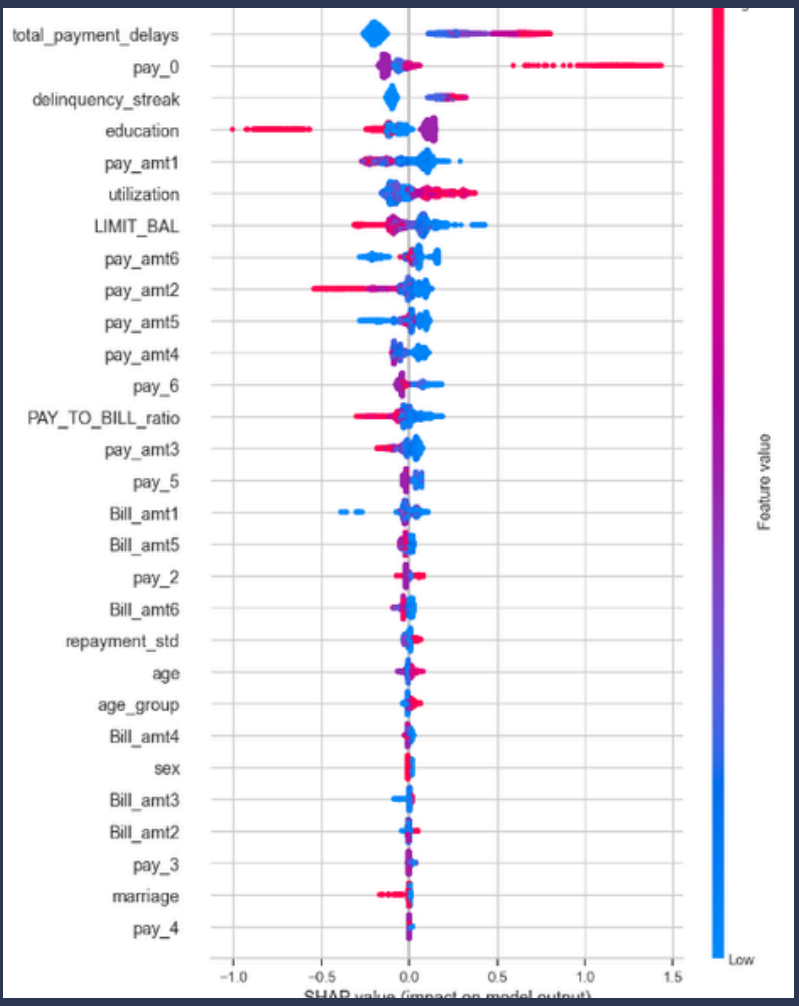
```
Best F2 Score:
 0.5924466356267326
Confusion Matrix:
 [[1469 2599]
 [ 104  853]]

Classification Report on Test Set:
              precision    recall  f1-score   support

           0     0.9339    0.3611    0.5208      4068
           1     0.2471    0.8913    0.3869       957

    accuracy                         0.4621      5025
   macro avg     0.5905    0.6262    0.4539      5025
weighted avg     0.8031    0.4621    0.4953      5025

ROC AUC Score: 0.7671
```

MODEL SELECTION/ THRESHOLD TUNING

- XGBoost delivered the highest recall for defaulters, correctly identifying 853 out of 957 defaulters, yielding an impressive recall of 89.1%.
- This performance came with a trade-off: precision dropped to 24.7%, and overall accuracy fell to 46.2%, largely due to a sharp increase in false positives.
- The confusion matrix reveals that 2,599 out of 4,068 non-defaulters were misclassified — meaning over 63% of non-defaulters were wrongly flagged.
- Despite these limitations, XGBoost achieved the highest F2-score of 0.592, confirming its effectiveness in prioritizing recall and minimizing false negatives — which aligns with business goals.

Given this strong recall performance and highest F2-score among all models evaluated, XGBoost has been selected as the final model, with plans to further fine-tune its classification threshold to strike a better balance between precision and recall.

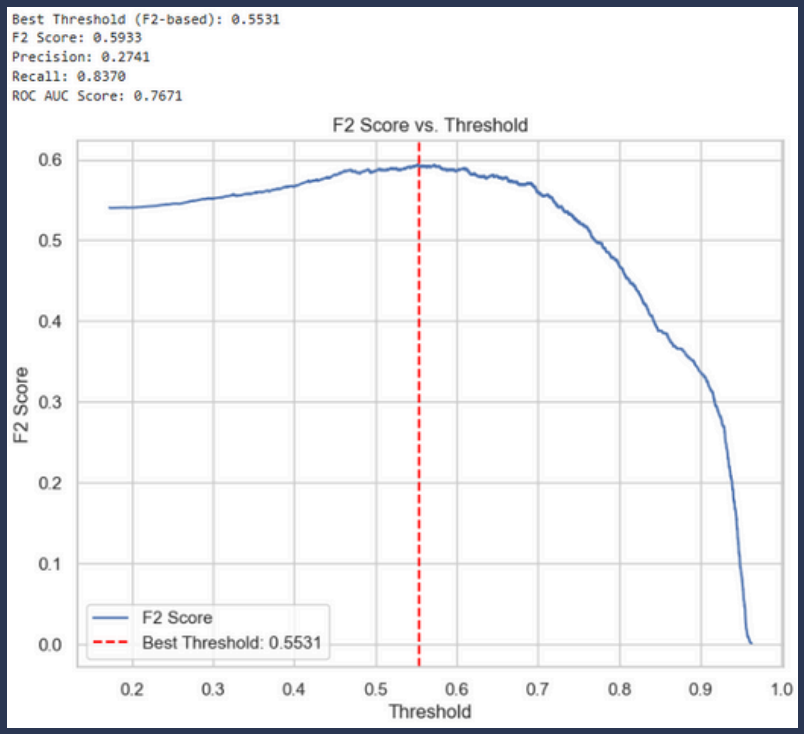
SHAP inferences from Beeswarm plot :



MODEL SELECTION/ THRESHOLD TUNING

- The model heavily relies on behavioral signals: past delays, payment patterns, and utilization ratios.
- Demographic features like age, sex, and marriage have much less influence compared to credit behavior.
- This aligns with domain expectations, as recent repayment behavior is often the most predictive for credit default.

XGBoost Threhold tuning :



To better align the model with business objectives, the classification threshold was optimized based on the F2-score, which places greater emphasis on recall — crucial in credit risk prediction, where missing defaulters is more costly than incorrectly flagging non-defaulters. The analysis revealed that a threshold of 0.5531 yielded the highest F2-score of 0.5933, with a recall of 83.7% and a precision of 27.4%.This represents a notable improvement over the default 0.5 threshold, striking a more effective balance between catching defaulters and limiting false positives.The ROC AUC score further confirms the model's ability to distinguish between defaulters and non-defaulters.This optimized threshold will be adopted for final predictions.

PREPROCESSING OF REAL TEST SET AND SUBMISSION FILE

The same steps which were followed for processing the train dataset were used for test set . This iincluded removing the anamolies and and forming the features like total_payment_delay etc, I also scaled it and remove the AVG_Bill_amt column and used the threshold given previously. Then predictions were made on it and some results are mentioned below in the images.

```
y_probs = XGBsearch.predict_proba(test_scaled)[:, 1]
y_pred_final = (y_probs >= 0.5531).astype(int)
```

```
next_month_default( 0 or 1 )
1    2980
0    2036
Name: count, dtype: int64
```

	Customer_ID	next_month_default(0 or 1)
count	5016.000000	5016.000000
mean	2508.500000	0.594099
std	1448.138806	0.491115
min	1.000000	0.000000
25%	1254.750000	0.000000
50%	2508.500000	1.000000
75%	3762.250000	1.000000
max	5016.000000	1.000000

BUSINESS IMPLICATIONS AND SUMMARY

The analysis and modeling performed in this project provide critical insights for financial institutions, particularly in credit risk assessment. The goal was to predict whether a customer will default on their credit payment next month. Key implications include:

1. Improved Risk Management: By identifying high-risk individuals using predictive modeling, banks can take proactive measures such as adjusting credit limits, requesting additional guarantees, or offering financial counselling.
2. Informed Credit Policy Decisions: The model helps shape data-driven policies for issuing credit. For example, stricter approval thresholds can be applied to profiles similar to historically high-risk customers.
3. Optimized Customer Retention: Knowing which customers are likely to default allows institutions to balance risk and reward, possibly retaining valuable customers with temporary financial issues through tailored plans.
4. Cost Savings and Profit Maximization: Reducing loan defaults directly decreases financial loss. Moreover, efficient allocation of credit improves capital utilization and return on assets.

The project focused on building a predictive model to identify potential credit defaulters, beginning with comprehensive data cleaning and anomaly correction in key categorical variables like marriage and education. Exploratory Data Analysis revealed important patterns and a noticeable class imbalance in the target variable. To address this, SMOTE was employed during training to oversample the minority class, improving the model's ability to detect defaults. Feature engineering and advanced model tuning through RandomizedSearchCV further enhanced performance. Evaluation based on precision, recall, and F1-score—rather than just accuracy—highlighted the trade-offs between false positives and false negatives. Overall, the project demonstrated how machine learning can support smarter credit decisions by combining data-driven insights with thoughtful preprocessing and model selection.

References

Along with this report there will be two other files, the code and the submission_23410034.csv
All the images, graphs were taken from the code file named Credit_default_23410034.

Rest I took help of following online sourcesv for clarifying things :

- https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234
- https://towardsdatascience.com/precision-recall-f1-score-what-do-they-mean-and-how-to-compare-them-61b0b45ad2f4
- https://imbalanced-learn.org/stable/references/generated/imblearn.over_sampling.SMOTE.html
- https://scikit-learn.org/stable/modules/calibration.html
- https://www.analyticsvidhya.com/blog/2021/06/why-are-tree-based-algorithms-better-at-handling-outliers/
- https://mitpress.mit.edu/9780262026399/credit-scoring-and-its-applications/