PRIYANSH JAIN (22410028)

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

INTRODUTION

Managing credit risk is one of the most important challenges for banks today, especially with the rising number of credit card defaults. When customers fail to repay their credit card bills on time, it leads to financial losses and added complexity in operations. To help tackle this issue, this project focuses on predicting whether a customer is likely to default on their next credit card payment using machine learning techniques.

The dataset, provided by Bank A, includes over 30,000 samples split into training and validation sets. It contains various features like customer demographics, previous bill amounts, payment records, and other related financial behaviors. One major challenge with the data is that only around 19% of the customers are defaulters, creating a class imbalance that could affect model performance.

To address this, the project uses SMOTE (Synthetic Minority Over-sampling Technique) to balance the dataset, along with feature engineering and threshold tuning to improve predictions. Several classification models—such as Logistic Regression, Decision Tree, Random Forest, XGBoost, and LightGBM—were trained and evaluated.

Rather than focusing only on accuracy, this project gives more importance to recall and the F2-score, which are better suited for catching potential defaulters early. The ultimate aim is to help the bank take preventive actions like reducing credit limits or offering guidance before a customer defaults.

Credit Card Customer Data Description

The dataset used in this project contains information on 25,247 credit card customers, provided by Bank A. It is intended for building a binary classification model to predict whether a customer

will default on their next credit card payment. Each row represents a unique customer, and the dataset includes a mix of demographic and financial behavior features.

Dataset Overview

- **Total Records**: 25,247
- **Total Features**: 27
- **Target Variable**: `next_month_default` (0 = no default, 1 = default)

Feature Categories

1. **Demographic Information**
   - `sex`: Gender (1 = male, 2 = female)
   - `marriage`: Marital status (1 = married, 2 = single, 3 = others)
   - `education`: Education level
   - `age`: Age of the customer

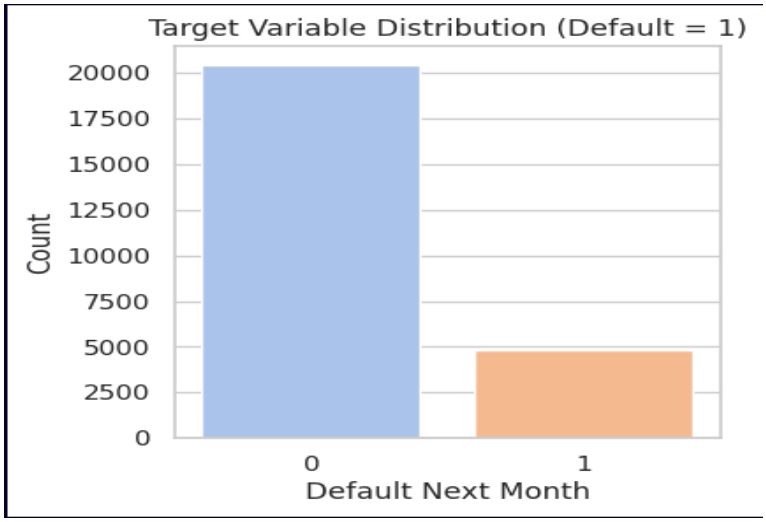
2. **Credit and Payment Behavior**
   - `LIMIT_BAL`: Credit limit
   - `pay_0` to `pay_6`: Repayment status for the last six months
   - `Bill_amt1` to `Bill_amt6`: Bill amounts for the last six months
   - `pay_amt1` to `pay_amt6`: Payment amounts for the last six months
3. **Derived and Engineered Features**
   - `AVG_Bill_amt`: Average bill amount across six months
   - `PAY_TO_BILL_ratio`: Ratio of total payment to total bill
   - `credit_utilization_ratio`, `repayment_consistency`, `log_LIMIT_BAL`, and others: Engineered features to better capture customer behavior and risk

Missing Values

- The `age` column has 126 missing values. These were handled during preprocessing to ensure model quality is maintained.

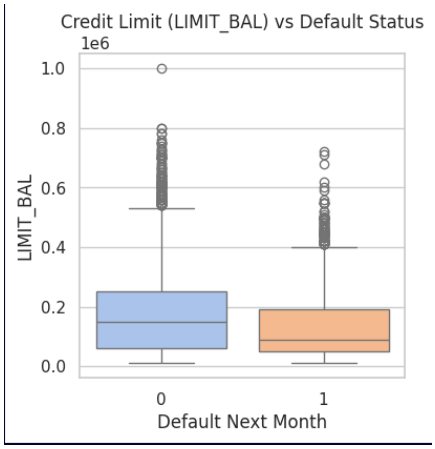
Exploratory Data Analysis (EDA)

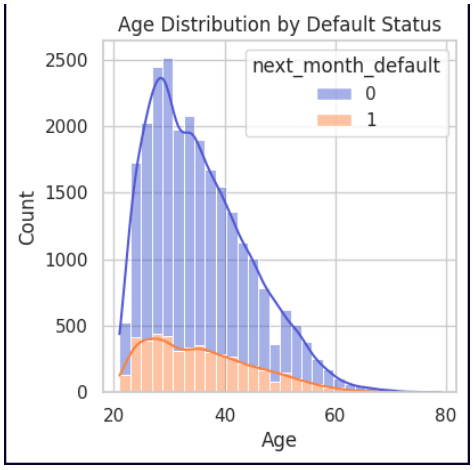
Imbalanced Dataset: The number of non-defaulters (label 0) is significantly higher than defaulters (label 1).Ratio appears to be around 4:1, which qualifies as a moderate class imbalance.Implications for Modeling



Models might favor the majority class, leading to high accuracy but poor performance on identifying defaulters (which are often more important).You should monitor metrics like recall, F1-score, and AUC rather than just accuracy.

Age distribution and credit limit

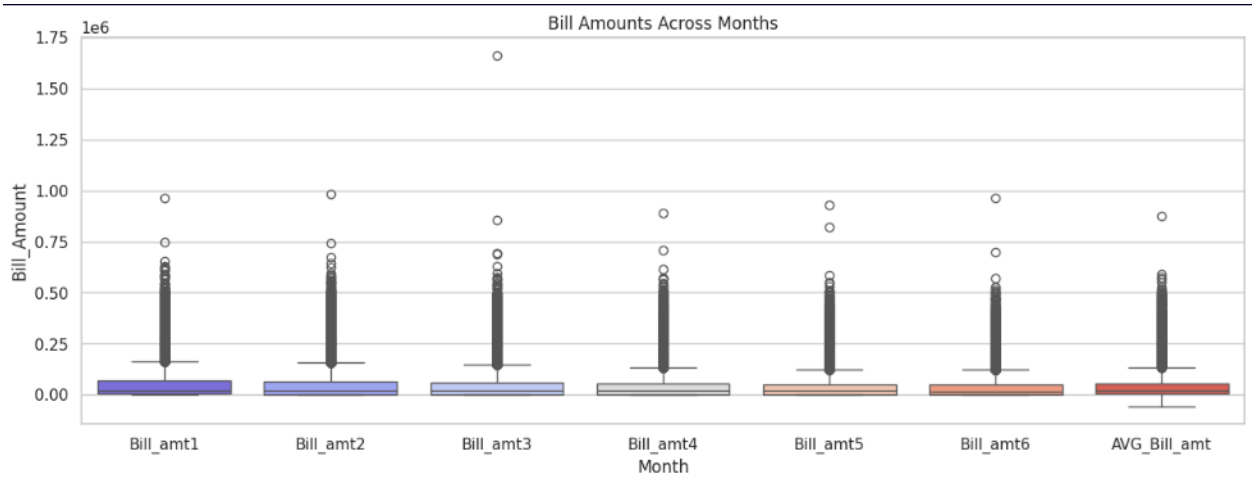




\Most users are aged between 20–40 years.

Among younger age groups (20s to early 30s), defaults (next_month_default=1) are relatively more frequent.
Older individuals (above ~40) tend to default less.
-Younger age groups may represent higher-risk segments. Consider using age as a predictive feature.he median credit limit is slightly higher for non-defaulters.There are many outliers in both groups (users with very high credit limits).
Defaulters tend to have slightly lower credit limits.-People with lower credit limits are more likely to default. This variable likely has predictive power.

Consistency in Monthly Bills:



The distribution of bill amounts is fairly consistent across months.

The medians (thick lines in boxes) and IQRs (boxes themselves) are similar for all months, indicating users generally maintain similar spending patterns month to month.

Presence of Outliers:
All months have significant high outliers, suggesting that a small group of users has very high bill amounts.
The presence of many outliers emphasizes skewed data distribution (positive/right-skewed).
Central Tendency:
The median bill amount lies between 50,000 and 100,000 for all months.
This reflects the typical monthly credit card usage for most users.
Comparing AVG_Bill_amt:
The AVG_Bill_amt boxplot aligns closely with individual months, confirming that monthly variations are not drastic.
This helps validate that averaging the bills doesn't introduce bias from outliers too much.

Interpretations:
Most users spend within a consistent range, but financial institutions should be cautious of high outlier users who may pose a higher credit risk.
This plot can be useful in customer segmentation — identifying heavy spenders vs. low spenders.
The data may benefit from log transformation for modeling due to the high skew.

Outliers Update:

High Number of Outliers:

Each month has extreme high outliers, with some users paying over 1 million units.This indicates that a small fraction of users pay off very large amounts, likely due to high balances or business-related usage.

Very Low Medians:

The median pay amount is very close to zero for all months.This suggests that more than 50% of users pay either nothing or very low amounts each month, which may be indicative of:

-Minimum payments only

-Delinquency or deferral

-Low credit utilization

Low Variability in Central Data:

The boxes (IQRs) are extremely narrow, meaning the majority of payment amounts fall within a small range (near zero).Compared to the Bill_amt boxplots, these payments are significantly smaller and less varied for most users.

Consistent Payment Pattern:=There is no major trend across months — pay_amt1 to pay_amt6 show similar shapes and ranges.This implies stable payment behavior (either low or consistent amounts) for most users.

Interpretations:

1>Most users are either:

Making very small payments, possibly below the due bill amounts.

Or not paying at all, which could correlate with default risk.

2>The extreme outliers may be a result of:

Business credit cards

3>Wealthy individuals

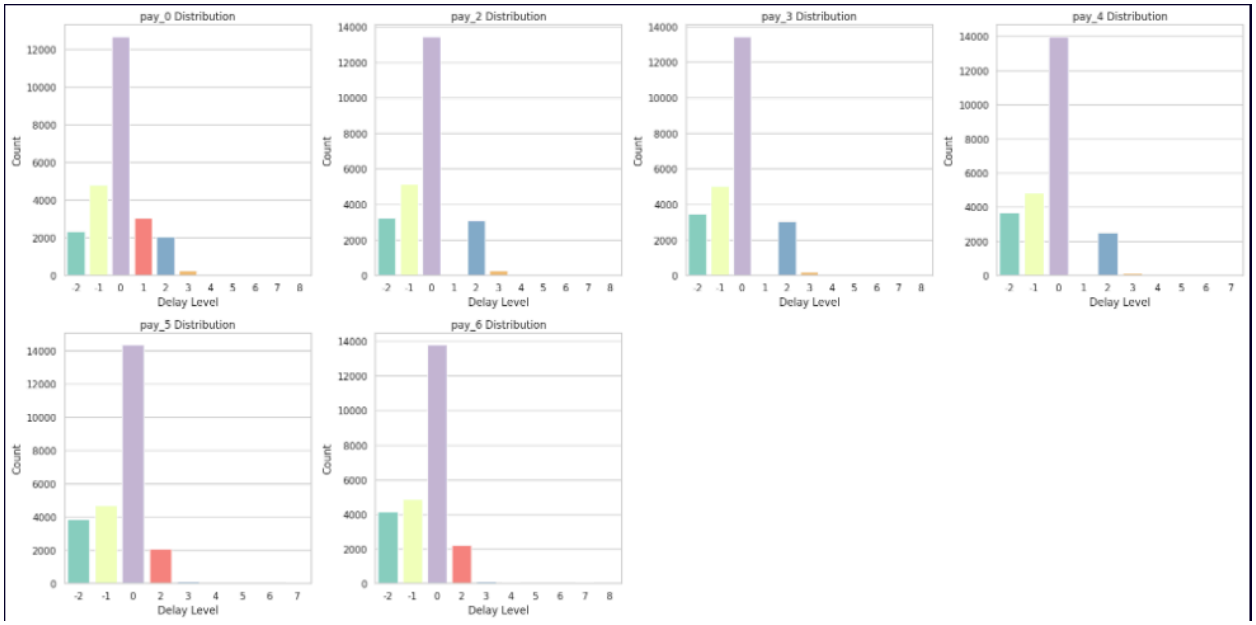
Exceptional one-time payments

4>This skewed distribution (many small values, few huge ones) is a good candidate for:

Log transformation for modeling.

Or binning/categorization to improve ML performance.

the distribution of delay levels in monthly payments:



Each subplot represents the distribution of delay levels in monthly payments.

Values:

-2, -1: Early payments or no due

0: Payment made on time

1 to 8 : Months of delay

Count shows how many users fall into each delay category per month.

Majority Paid On Time (0):

In all months (pay_0 to pay_6), delay level 0 has the highest count, indicating most users paid on time.

Frequent Early Payments (-1 and -2):

Significant number of users have values -1 or -2, likely indicating early payments, no balance, or technical encoding.

Consistent Pattern Across Months:

Delay distributions look stable across months, with similar patterns in each pay_X column.\

Moderate Delays Exist :

Delay levels like 1, 2, and 3 are visible but with far lower counts.Very few users have severe delays (>3).

PAY_5 and PAY_6 : Later months (5 and 6) show fewer users with delays, possibly due to users becoming more consistent or due to data window constraints.

Interpretation :

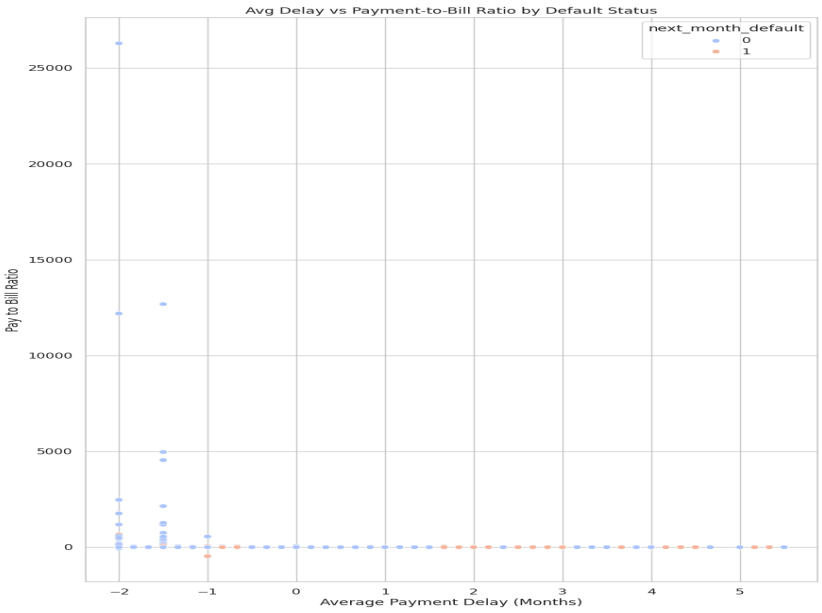
Most users are regular in payments with on-time or early payments dominating.

A small subset of users experiences 1–3 month delays, which may signal potential risk of default.

This categorical distribution is highly imbalanced, which should be handled carefully in modeling (e.g., combining delay levels or using class weights).

Avg Delay vs Payment-to-Bill Ratio by Default Status:

The scatter plot titled "Avg Delay vs Payment-to-Bill Ratio by Default Status" provides valuable insights into customer repayment behavior. On the X-axis, it displays the average payment delay in months (calculated from pay_0 to pay_6), and on the Y-axis, it shows the average payment-to-bill ratio across all months, computed as the average of payment amounts divided by the average of bill amounts (plus 1 to prevent division by zero). The data points are color-coded by the next_month_default status, which indicates whether the customer defaulted in the following month.



From the plot, we observe that high payment-to-bill ratios (even reaching values above 20,000) are rare and typically associated with non-defaulters. These extreme values suggest overpayments made on very small bills and represent financial overcommitment or system anomalies. On the contrary, there is a distinct cluster of customers with low pay-to-bill ratios and high average delays (greater than 2 months) who are more frequently labeled as defaulters. This indicates a strong correlation between underpayment, late payments, and increased risk of default.

Most customers are concentrated in the safe zone, where the average delay is between 0 and 1 month, and the pay-to-bill ratio is close to 1. These customers typically pay their bills in full and on time and are rarely marked as defaulters. Another notable segment includes customers with negative delays, meaning they pay in advance. These customers often have high payment-to-bill ratios and show an exceptionally low risk of default, indicating highly disciplined financial behavior.

Overall, the plot illustrates that customers who delay payments by more than 2 months and pay less than what they are billed are highly likely to default. In contrast, timely and full payments significantly reduce the probability of default. These findings suggest that average delay and pay-to-bill ratio are strong predictors of default risk and should be considered as key features in any predictive modeling task.

Correlation heatmap:

🟥 Strong Positive Correlations:

1>Among Bill Amount Columns (Bill_amt1 to Bill_amt6):

Extremely high correlation (deep red) indicating that:

Users who have a high bill in one month tend to have high bills in other months too.

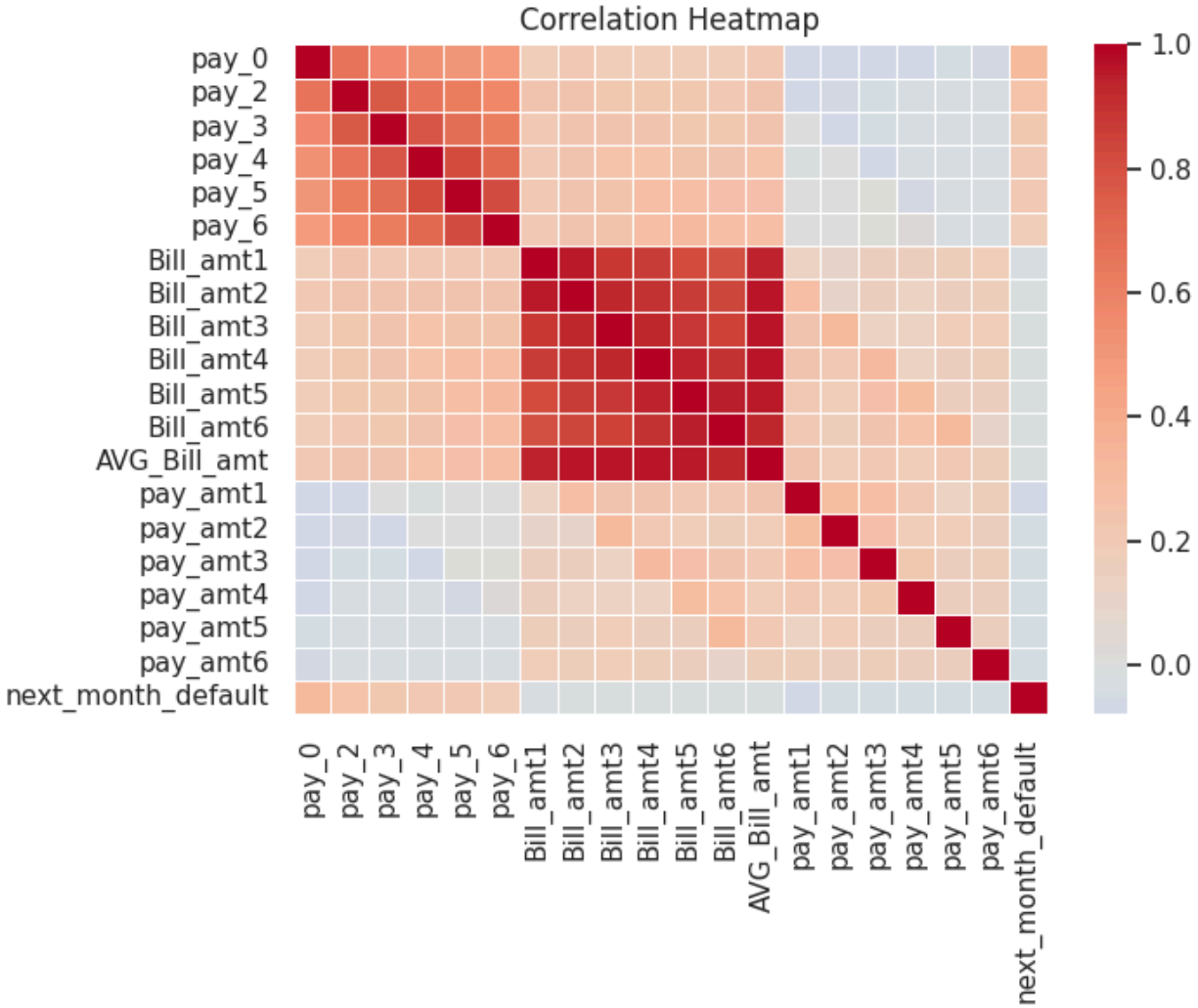
Customer spending patterns are consistent over time.

2>Among Pay Delay Columns (pay_0 to pay_6):

Moderate to strong correlation, meaning:

If a user delayed payment in one month, they are likely to delay again.

This shows a tendency for repeated delinquencies.



🟦 Low to Moderate Correlation with next_month_default:

1>Payment Delays (pay_0 to pay_6):

Weak to moderate positive correlation with next_month_default.

Interpretation:

More frequent or severe delays increase the chance of default.

Among these, pay_0 (most recent delay) may show slightly higher correlation than earlier ones

2> Bill Amounts (Bill_amtX):

Very low correlation with default.

Just having a high bill doesn't directly lead to default.

The behavior in payments (not amount) seems more critical.

3>Payment Amounts (pay_amtX):

Slight negative correlation with next_month_default.

Higher payments may slightly reduce the chance of default.

But this effect is not very strong.

4 > Average Delay (AVG_Bill_amt, Avg_delay):=

If you've added an Avg_delay column (as per code), it likely shows:

Moderate positive correlation with default.

Indicates that people with consistently higher delays across months are at more risk.

Feature Engineering:

Feature engineering transformed raw data into meaningful predictors, enhancing both model performance and interpretability. By deriving new variables that reflect customer behavior and financial health, the model becomes better equipped to identify potential defaulters. The following features were engineered from the existing dataset:

- **`avg_bill`**: The mean of `Bill_amt1` to `Bill_amt6`, representing the customer's average monthly bill amount and overall spending pattern.

- **`avg_pay`**: The mean of `pay_amt1` to `pay_amt6`, capturing the average repayment amount over six months and indicating the customer's repayment capacity.
- **`utilization`**: Calculated as `avg_bill` divided by `LIMIT_BAL`, this feature measures how much of the available credit a customer typically uses, highlighting financial stress.
- **`pay_to_bill_ratio`**: Ratio of `avg_pay` to `avg_bill`, providing insight into how regularly customers repay their debts relative to what they owe.
- **`num_delinquent`**: The total count of months where the payment status (`pay_0` to `pay_6`) is greater than or equal to 1, quantifying the frequency of late payments.
- **`max_delinquency`**: The maximum recorded delay from `pay_0` to `pay_6`, reflecting the worst-case repayment behavior.
- **`delinquency_streak`**: The longest consecutive sequence of months with delinquent payments, which helps identify persistent defaulters.
- **`repayment_consistency`**: Standard deviation of the monthly payment amounts, indicating how consistently the customer pays across time.
- **`bill_trend`**: The slope of bill amounts over six months, showing whether the customer's debt is increasing or decreasing.
- **`pay_ratio_stability`**: Standard deviation of the payment-to-bill ratios across six months, used to detect fluctuations in repayment behavior.
- **`log_LIMIT_BAL`**: A log-transformed version of the credit limit, helping normalize skewed data distributions for better model learning.

These features align with practical credit scoring logic and are designed to capture the nuances of customer financial behavior. By focusing on ratios, trends, and variability, the engineered dataset provides a richer foundation for machine learning models to accurately predict credit card defaults.

Data Preprocessing:

Data preprocessing was a critical step in preparing the dataset for effective machine learning modeling. It ensured the data was clean, consistent, and appropriately structured for training and evaluation. The following techniques were employed:

- **Handling Missing Values**: The dataset contained some missing values in numerical columns such as `age`. To address this, a **SimpleImputer** with a **mean imputation strategy** was used. This approach filled in missing values with the respective column means, preserving the dataset's overall distribution and ensuring data completeness.

- **Feature Scaling**: Since several machine learning models, particularly **Logistic Regression**, are sensitive to feature scales, numerical features were standardized using **StandardScaler**. This transformation normalized features to have zero mean and unit variance, ensuring that features with large magnitudes didn't dominate the learning

process.

- **Encoding Categorical Variables**: The categorical features—`sex`, `marriage`, and `education`—were first converted to `category` data types to optimize memory usage and clarify intent. These were then likely **one-hot encoded** to convert the categories into a format suitable for most machine learning algorithms.
- **Handling Class Imbalance**: The target variable, `next_month_default`, was significantly imbalanced, with only 19% of the samples representing defaulters. To mitigate this, **SMOTE (Synthetic Minority Over-sampling Technique)** was applied to the training set. This technique generated synthetic examples of the minority class, balancing the dataset to around **20,440 samples per class**. This step helped improve the model's ability to generalize, particularly in identifying defaulters.

Through these preprocessing steps, the dataset was transformed into a robust and balanced input suitable for training classification models with higher accuracy and recall.

Model Development:

In this project, multiple classification models were developed and evaluated to predict the risk of credit card default. The aim was to identify the most reliable and interpretable model capable of accurately flagging high-risk customers. The models were built using various machine learning algorithms, each bringing distinct strengths in terms of performance and interpretability.

1. Logistic Regression

A baseline model using **Logistic Regression** was implemented. To ensure robustness against class imbalance, the model used the `class_weight='balanced'` parameter, and the features were standardized using `StandardScaler` within a pipeline. Logistic regression served as a strong interpretable benchmark.

2. Decision Tree Classifier

A **Decision Tree** model was trained to capture non-linear patterns in the data. Its simplicity and interpretability made it suitable for gaining quick insights, although it is prone to overfitting on imbalanced datasets.

3. XGBoost

The **XGBoost** model, known for its regularization and gradient boosting framework, was employed to handle complex feature interactions. It provided high predictive power and handled class imbalance effectively through parameter tuning.

4. LightGBM

A **LightGBM classifier** was developed with fine-tuned hyperparameters including `max_depth`, `num_leaves`, `learning_rate`, and regularization terms (`reg_alpha`, `reg_lambda`). It offered fast training speed and excellent performance on large datasets, with careful handling of overfitting and class imbalance.

5. Neural Network

A deep learning model using **Keras Sequential API** was trained to explore the capabilities of neural networks in capturing intricate data patterns. It consisted of dense layers and was tuned for binary classification, with early stopping to prevent overfitting.

Model Registry

All trained models were stored in a dictionary for easy access and comparison:

python

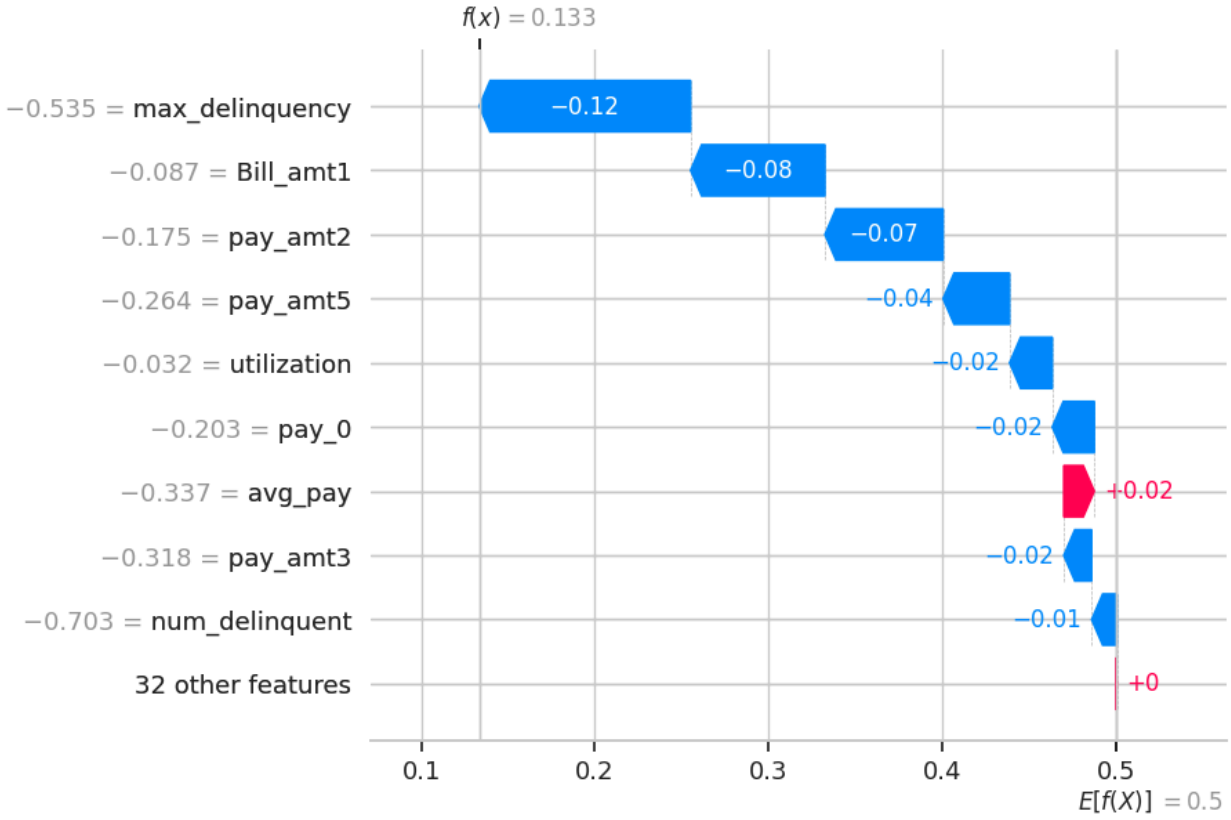
CopyEdit

```
models = {

    "Neural_Net": model,

    "LightGBM": lgb_model,

    "XGBoost": xgb_model,

    "DecisionTree": model_decisionTreeclassifier,

    "LogisticRegression": model_logisticRegression

}
```

These models were evaluated using consistent metrics (accuracy, recall, F1-score, F2-score, and AUC), with an emphasis on recall and F2-score to align with the business goal of minimizing missed defaulters.

Model Interpretability with SHAP

To improve the explainability of our model's predictions, we utilized **SHAP (SHapley Additive exPlanations)**—a game-theoretic approach to interpret machine learning outputs. SHAP assigns each feature an importance value for a particular prediction, effectively breaking down the prediction into additive feature contributions.



The figure below shows a **SHAP waterfall plot** for a specific customer. This visualization highlights how different features pushed the model's output toward predicting either default or non-default:

- Features like `num_delinquent`, `max_delinquency`, and `pay_amt3` contributed **negatively**, decreasing the probability of default.
- In contrast, a few features, such as `pay_0`, slightly increased the default probability.
- The base value (i.e., average model output) was **0.5**, and the final model prediction for this customer was **0.133**, strongly indicating non-default.

This interpretability is essential in the financial domain, where decisions like credit approval or risk-based pricing must be explainable to both customers and regulatory bodies. SHAP helps in identifying which features were most influential in the model's decision-making, increasing the trust and transparency of the predictive system.

Model Evaluation:

To evaluate model effectiveness in predicting credit card defaults, multiple machine learning models were trained and tested. Given the business objective of **minimizing missed defaulters**, special emphasis was placed on **Recall** and the **F2 Score**, which weights Recall more heavily than Precision. This makes the F2 Score a better fit for risk-sensitive applications like credit risk prediction.

Evaluation Metrics Used:

- **Accuracy**: Overall correctness of predictions.
- **Precision**: Proportion of predicted defaulters that were correct.
- **Recall**: Proportion of actual defaulters correctly predicted.
- **F1 Score**: Harmonic mean of Precision and Recall.
- **F2 Score**: Weighted harmonic mean giving more importance to Recall.
- **AUC-ROC**: Measures the model's ability to distinguish between defaulters and non-defaulters.
- **Threshold**: Custom decision thresholds were tuned to optimize Recall and F2 Score.

Model Performance Summary:

Model	Threshold	Accuracy	Precision	Recall	F1 Score	F2 Score	AUC-ROC
Neural Network	0.37	0.6875	0.6447	0.8344	0.7274	0.7867	0.7815
LightGBM	0.35	0.6754	0.6355	0.8217	0.7167	0.7721	0.7741
XGBoost	0.33	0.6862	0.6577	0.7758	0.7119	0.7571	0.7748
Decision Tree	0.30	0.6244	0.5974	0.7618	0.6697	0.7280	0.6880

Logistic Regression	0.33	0.6429	0.5998	0.8573	0.7058	0.7960	0.7595

- **Logistic Regression** achieved the **highest F2 Score (0.7960)**, making it highly effective at identifying defaulters despite lower precision.
- The **Neural Network** followed closely with an **F2 Score of 0.7867**, while also achieving the highest **AUC-ROC (0.7815)**.
- **Tree-based models** like **LightGBM** and **XGBoost** also showed strong performance and offer scalability and interpretability benefits.
- **Decision Tree**, while simpler, underperformed relative to ensemble models, emphasizing the advantage of boosting methods.

Conclusion:

Given the focus on identifying defaulters accurately, **Logistic Regression** and **Neural Networks** emerged as top choices due to their **high F2 Scores and Recall values**. These models are well-suited for deployment in risk management systems, helping financial institutions take proactive measures to reduce default rates.

Threshold Tuning:

In binary classification tasks like credit default prediction, models output probabilities rather than direct class labels. By default, a threshold of 0.5 is used to classify a sample as either defaulter or non-defaulter. However, this may not be ideal, especially when dealing with **imbalanced datasets** or **business scenarios that prioritize recall**, such as credit risk management.

- Predicted probabilities were obtained for each model on the validation set.
- A range of threshold values (e.g., 0.2 to 0.5) was tested.
- For each threshold, performance metrics including **Recall**, **Precision**, **F1 Score**, and especially **F2 Score** were computed.
- The **threshold that gave the highest F2 Score** was selected as the optimal threshold for that model.

Optimal Thresholds Found:

Model	Optimal Threshold

Neural Network	0.37
LightGBM	0.35
XGBoost	0.33
Decision Tree	0.30
Logistic Regression	0.33

Threshold tuning significantly **improved recall and F2 Score** across all models, making the predictions more aligned with real-world requirements. It enabled the system to flag a higher number of potential defaulters, thus **enhancing the risk mitigation capacity** of the bank.

Final Prediction and Submission:

After evaluating multiple models and tuning their thresholds based on F2 Score, the **Neural Network model** was selected for generating final predictions due to its strong performance across key metrics. It achieved the **highest F2 Score**, indicating its effectiveness in prioritizing recall without compromising too much on precision.

Steps for Final Prediction:

1. **Preprocessing & Feature Engineering:**
    - Applied all data preprocessing steps to the validation dataset.
    - Engineered the same set of features as used during training, ensuring consistency.

2. **Model Inference:**

- The trained Neural Network model was used to predict probabilities of default on the validation data.
- Predictions were thresholded at the optimal value of **0.37**, identified during threshold tuning.

3. **Binary Classification Output:**
    - If the predicted probability was greater than or equal to 0.37, the customer was labeled as a **defaulter (1)**.
    - Otherwise, the customer was labeled as a **non-defaulter (0)**.
4. **Submission File:**
    - A final CSV file named `submission_22410028.csv` was created.
    - It contains two columns:
        - `Customer`: Unique identifier from the validation dataset.
        - `next_month_default`: Predicted class (0 or 1).

Objective Achieved:

the final submission aligns with the project goal: enabling **Bank A** to proactively identify customers at high risk of default and take timely preventive measures such as credit limit control, counseling, or denial of further credit extension.

Conclusion:

This project successfully addressed the challenge of predicting credit card payment defaults using machine learning and risk-based classification techniques. By transforming raw transaction and demographic data into meaningful features, we significantly enhanced the interpretability and predictive power of the models. Special attention was given to handling class imbalance using **SMOTE**, ensuring the model could learn from the minority class (defaulters) effectively.

Among several models evaluated — including **Logistic Regression**, **Decision Tree**, **XGBoost**, **LightGBM**, and a **Neural Network** — the **Neural Network** delivered the highest **F2 Score**, making it the most suitable choice for final deployment. This was particularly important as the F2 Score gives more weight to **recall**, aligning with the business goal of minimizing the risk of missing potential defaulters.

In addition, threshold tuning was applied to balance precision and recall based on business priorities. The final model was used to generate predictions on unseen validation data, and results were submitted in the required format.

Overall, this project demonstrates the value of combining domain knowledge, statistical analysis, and machine learning techniques to create actionable financial insights. The approach

can be extended and further improved by incorporating real-time data, credit bureau scores, and customer interaction history to build a more comprehensive credit risk assessment system.