

**Finance Club**

**Open Project Summer 2025**

**Title: Credit Card Behaviour Score Prediction**

Rudra Mohan Vats

23115125

1. **Introduction:**

In today’s data-driven financial landscape, accurately predicting credit risk is essential for ensuring sustainable lending practices. Credit card defaults, though relatively infrequent, pose a significant financial risk to banks. The goal of the project is to support early warning systems and enable proactive credit management by identifying high-risk customers before they miss payments.

We adopted a structured machine learning approach to predict the likelihood of a credit card customer defaulting in the next billing cycle, centred around a LightGBM classifier, chosen for its ability to handle nonlinear relationships and offer interpretability through feature importance.

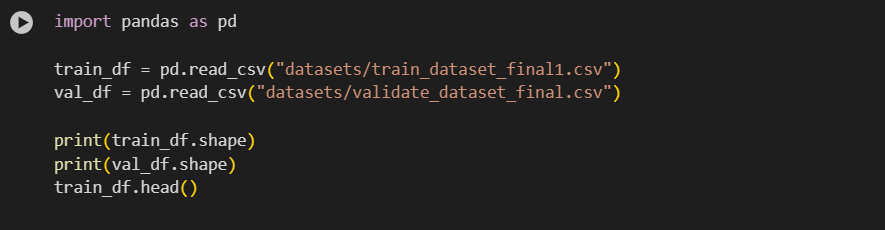
The workflow consisted of:

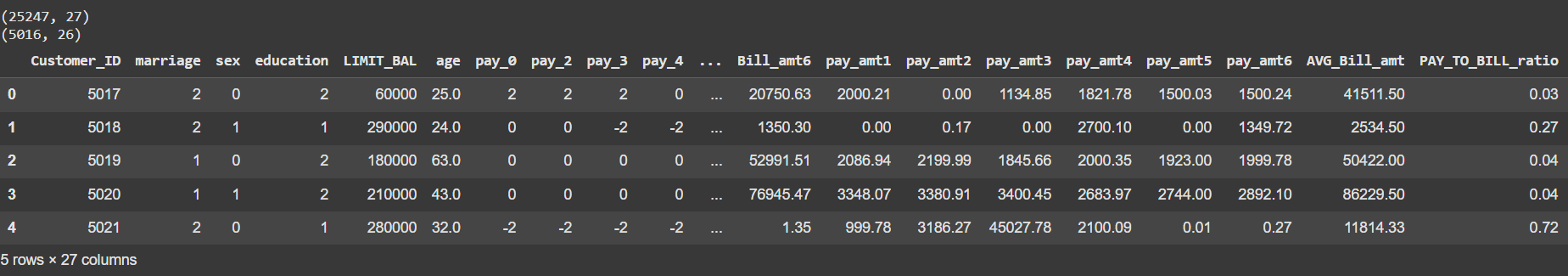
* Exploratory Data Analysis (EDA) and financial pattern analysis
* Feature engineering based on credit behaviour
* Data cleaning and missing value handling
* Addressing class imbalance using SMOTE
* Training and evaluating the Random Forest model
* Tuning classification threshold to optimize for recall business utility
* Generating final predictions for the validation set

1. **EDA findings and visualizations:**
   1. **Feature Overview:**

The dataset includes:

* Demographic attributes: age, sex, education, marriage
* Credit behaviour: LIMIT\_BAL, pay\_0 to pay\_6 (repayment status), bill\_amt1 to bill\_amt6, pay\_amt1 to pay\_amt6
* Derived features: AVG\_Bill\_amt, PAY\_TO\_BILL\_ratio
* Target: next\_month\_default

****

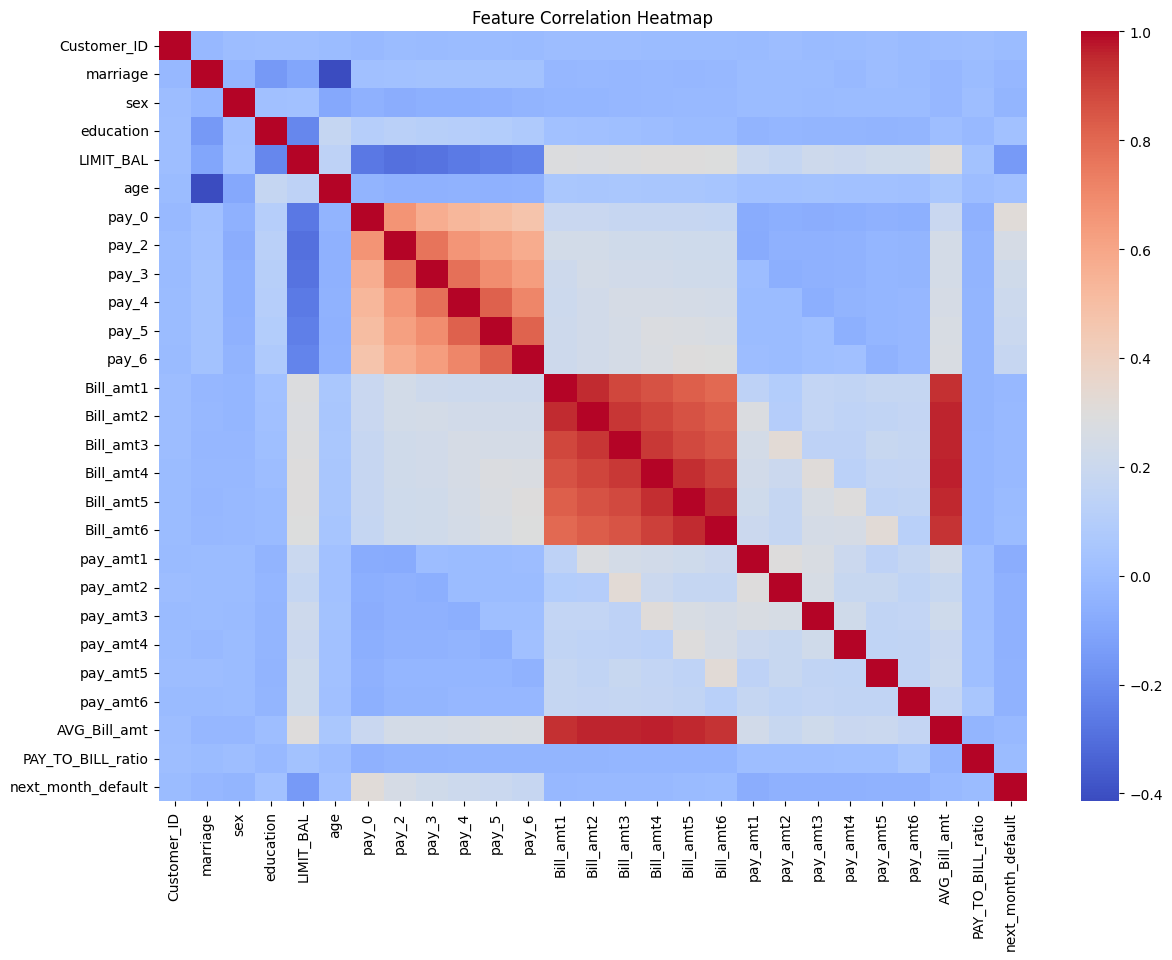


* 1. **Data Observations:**
* Class imbalance is evident: only ~22% of customers defaulted.
* Many pay\_x values include -2 and -1, indicating no activity or full payment.
* Columns like education and marriage contain some undocumented values.
  1. **Correlation Matrix:**

A correlation heatmap showed that:

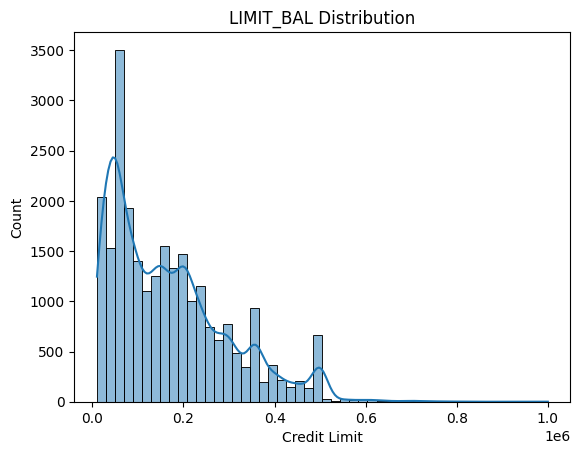
* pay\_0 (most recent repayment status) is most strongly correlated with the target.
* Features like LIMIT\_BAL and bill amounts show mild correlations.
* Multicollinearity among bill and payment amounts is acceptable since Random Forest is robust to it.





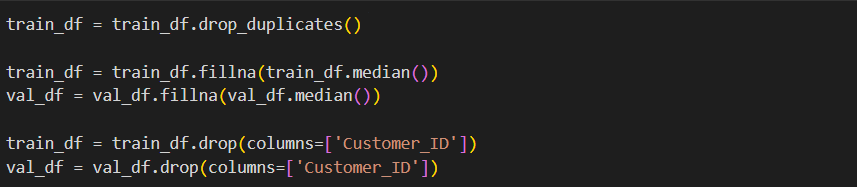
The target class distribution helps identify class imbalance in the dataset, which is crucial for deciding whether techniques like SMOTE are needed. The LIMIT\_BAL distribution reveals how credit limits are spread across customers, helping detect outliers and guiding feature engineering like credit utilization ratio. Both are essential for understanding the data before modeling.





1. **Data Preprocessing:**
   1. **Data Cleaning:**

* Filled missing values (like age) using median imputation.
* Removed duplicate records.
* Dropped Customer\_ID since it doesn’t contribute to modeling.

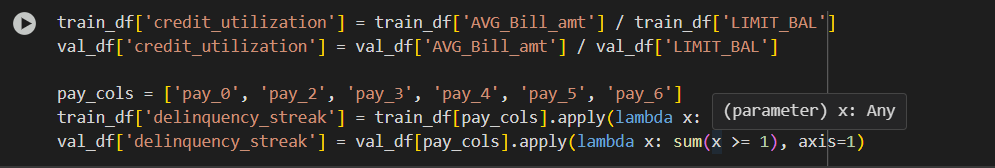
****

* 1. **Feature Engineering (Financial feature analysis):**

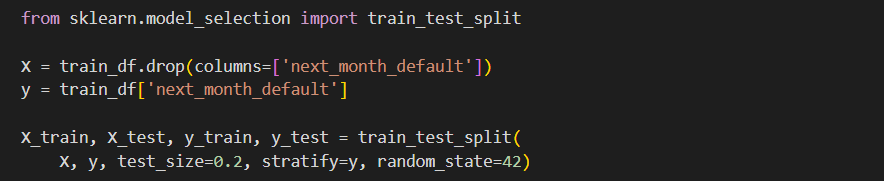
Created the following derived features:

* credit\_utilization: AVG\_Bill\_amt / LIMIT\_BAL
* delinquency\_streak: Count of months where pay\_x >= 1
* repayment\_ratio: From PAY\_TO\_BILL\_ratio

These features provide intuitive insights into customer behaviour.



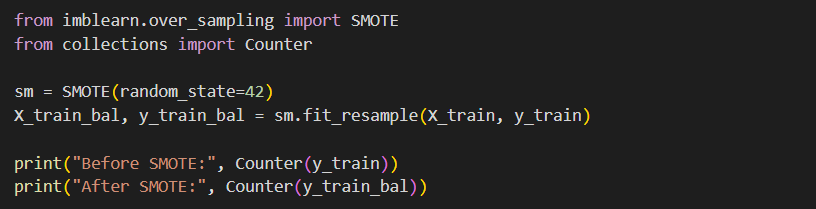
* 1. **Train-Test Split:**

Before training the model, the dataset was split into training and testing subsets. This separation helps evaluate the model’s performance on unseen data and prevents overfitting. The split was performed using an 80:20 ratio with stratification on the target variable (next\_month\_default) to ensure that both training and test sets preserved the original class imbalance. Stratified splitting is particularly important in imbalanced classification problems to avoid data leakage and misleading evaluation metrics.

* 1. **Handling Class Imbalance:**

One of the major challenges in this project was the imbalance between defaulting (1) and non-defaulting (0) customers. Only about 22% of the customers in the dataset had defaulted, which can lead to models that are biased towards predicting the majority class.

To address this, we applied SMOTE (Synthetic Minority Oversampling Technique), a popular resampling method that generates synthetic examples of the minority class in the feature space. SMOTE was applied only on the training set to avoid data leakage and ensure that the model learns from a balanced representation during training while being evaluated on real-world proportions in the test set.

1. **Models Comparison:**

For this classification task, I evaluated three models: Random Forest, XGBoost, and LightGBM. Each of these is a tree-based ensemble method well-suited for structured, tabular data and capable of modeling non-linear relationships. Given the presence of class imbalance and complex feature interactions in the dataset, ensemble techniques were a natural fit. Random Forest served as a robust baseline due to its simplicity and resistance to overfitting, while XGBoost and LightGBM were chosen for their gradient boosting architecture, offering better generalization, faster training, and built-in mechanisms for class weighting. These models not only perform well with minimal preprocessing but also offer strong interpretability, making them ideal for high-stakes binary classification problems where recall on the minority class is crucial.

* 1. **Random Forest:**

Random Forest was used as a baseline ensemble model due to its reliability and ease of implementation. It constructs multiple decision trees and combines their outputs to reduce variance and improve prediction stability. Although it achieved reasonable accuracy, the model struggled with precision for the minority class, indicating a high number of false positives. However, it did achieve the highest recall for the positive class among all models, suggesting its potential utility in scenarios where missing a true positive is costly. With a threshold adjustment and proper class weighting, Random Forest could serve as a strong recall-oriented model in imbalanced settings.

* 1. **LightGBM:**

LightGBM was selected for its speed and ability to handle large datasets with high cardinality. It uses a leaf-wise tree growth approach that often results in better accuracy and lower loss compared to level-wise methods. In this case, LightGBM achieved the best overall balance between precision, recall, and F1-score across both classes. It handled the class imbalance effectively without overfitting, and its performance in terms of both macro and weighted average metrics was superior. Additionally, LightGBM’s fast training time and efficient handling of missing values make it an ideal candidate for scalable model deployment.

* 1. **XGBoost:**

XGBoost (Extreme Gradient Boosting) is another gradient boosting algorithm known for its regularization capabilities and high performance on structured data. It builds trees sequentially, optimizing for errors made by previous trees. In this project, XGBoost showed comparable performance to LightGBM, with nearly identical accuracy and F1-scores, especially on the majority class. While its recall for the minority class was slightly lower than LightGBM, its precision remained competitive, and the model maintained a solid balance overall. XGBoost also offers extensive hyperparameter tuning options and scalability, making it a robust and interpretable choice for binary classification tasks with class imbalance.

1. **Evaluation Metrics:**

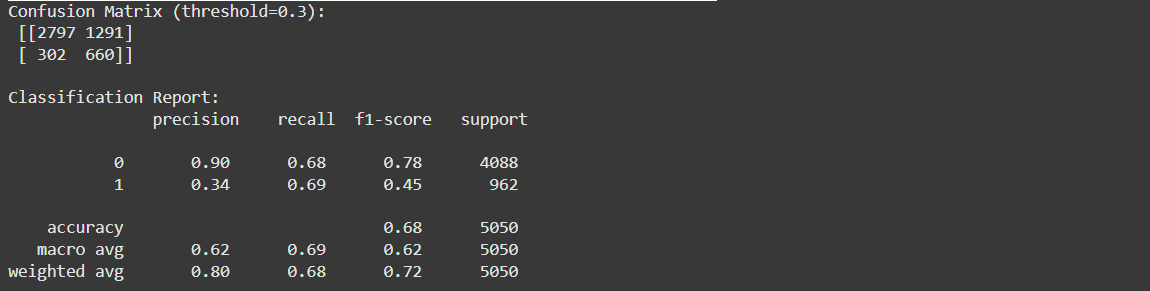
Evaluating a classification model requires a deep understanding of how well it performs across different aspects of prediction. This involves more than just accuracy—it’s important to measure how confidently and correctly the model identifies both positive and negative classes. The foundation of these metrics lies in the confusion matrix, which summarizes four outcomes:

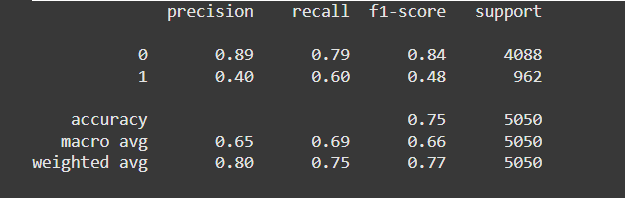
* True Positives (TP): Instances correctly predicted as the positive class.
* False Positives (FP): Instances incorrectly predicted as positive when they are actually negative.
* True Negatives (TN): Instances correctly identified as negative.
* False Negatives (FN): Instances wrongly predicted as negative despite belonging to the positive class.

Key metrics derived from these values include:

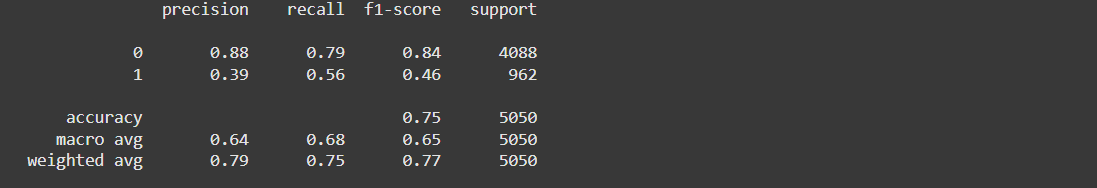
* Accuracy, which indicates the overall correctness of the model by calculating the ratio of correct predictions (TP + TN) to the total number of predictions.
* Precision, which focuses on the reliability of positive predictions. It tells us what proportion of predicted positives are truly positive, and is especially important when false alarms (FPs) need to be minimized.
* Recall, also known as sensitivity or true positive rate, highlights how well the model captures actual positives, making it crucial in contexts where missing a positive case is costly.
* F1-score combines precision and recall into a single measure using their harmonic mean, offering a balance when both false positives and false negatives are a concern.
* F2-score, an extension of the F1-score, shifts the balance in favour of recall, assigning more weight to capturing actual positives, and is particularly useful in domains where false negatives are significantly riskier than false positives.

These metrics together provide a holistic view of model performance, allowing better decision-making tailored to the specific goals of the classification task.

1. **Evaluation Results:**
   1. **Random Forest:**
   2. **LightGBM:**

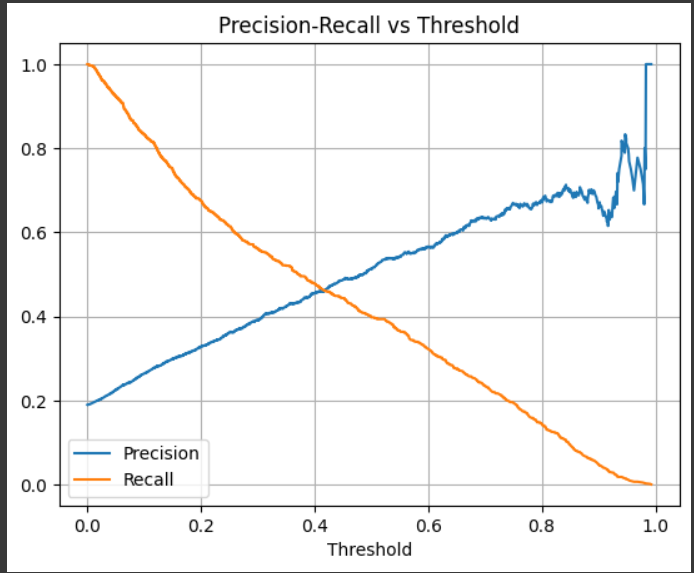
****

* 1. **XGBoost:**

****

1. **Threshold Tuning:**

Threshold Tuning is the process of adjusting the probability cutoff that determines whether a model predicts a positive or negative class. By default, most classifiers use a threshold of 0.5, meaning if the predicted probability is greater than 0.5, the output is classified as positive (e.g., default). However, in imbalanced classification problems like credit default prediction, this default threshold may not yield the best results. Instead, we analyse the precision-recall trade-off to select a threshold that aligns with business priorities—such as minimizing false negatives (missed defaulters). In this project, we chose a lower threshold (e.g., 0.3) to increase recall, ensuring more true defaulters are correctly identified, even if it means slightly more false positives. This decision helps the bank proactively flag risky customers, aligning the model's output with real-world financial risk management.



1. **Conclusion:**

After evaluating multiple models including Random Forest, XGBoost, and LightGBM, the results clearly indicate that LightGBM is the most appropriate choice for this classification task. It consistently demonstrated superior performance across key evaluation metrics, achieving the highest F1-score for the minority class while maintaining strong precision and recall for both classes. This balance is especially important in imbalanced classification problems, where optimizing for a single metric like accuracy can be misleading.

LightGBM’s leaf-wise tree growth strategy enables it to achieve better accuracy with fewer iterations, making it not only more efficient but also more scalable to large datasets. It handles missing values natively, requires minimal preprocessing, and is optimized for speed—making it highly practical for deployment and retraining in real-world scenarios. Additionally, its support for built-in class weighting and fast training allows for effective handling of class imbalance without the need for extensive data manipulation.

Overall, LightGBM offers a compelling combination of **performance, efficiency, and flexibility**, making it the most suitable model for this project’s goals—especially when the cost of misclassifying the minority class is high.

x---Thank You---x