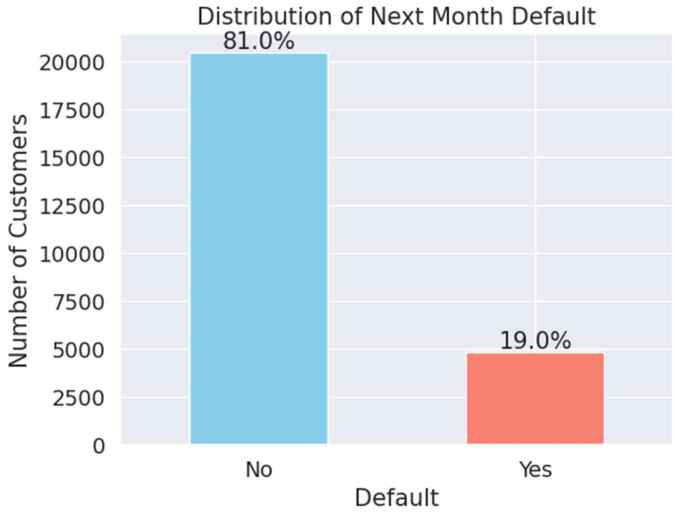
Credit Card Default Prediction Model

1. Overview of Approach and Modelling Strategy

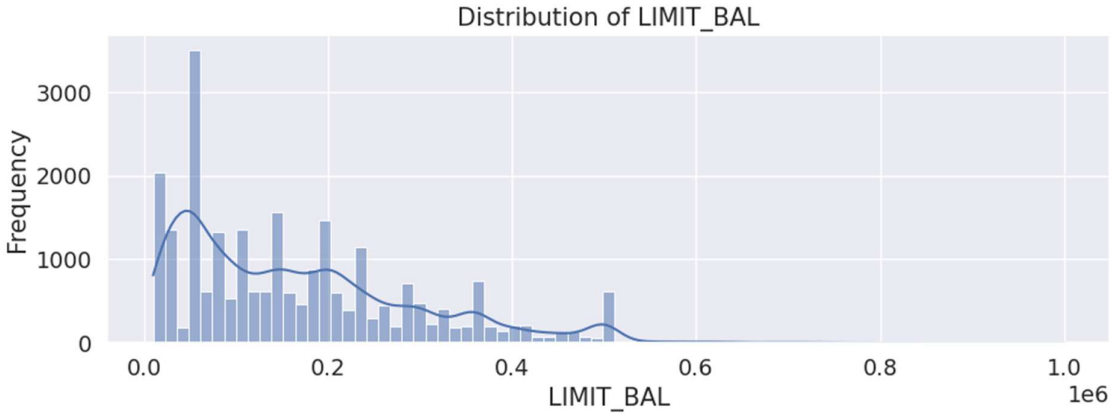
The task was to build a binary classification model predicting whether a customer will default on their next credit card payment. The dataset is imbalanced: only about 19% of customers default (target = 1) while 81% do not. In credit scoring, this minority/majority class imbalance is typical. I handled it by exploring resampling/class-weighting and by tuning the decision threshold to favour the minority class. In practice I tested several algorithms – for example, logistic regression, decision trees and ensemble methods like XGBoost – and evaluated them with cross-validation. LightGBM was ultimately chosen as the final model because of its superior performance on recall-focused metrics. Gradient-boosting models (such as LightGBM) can capture complex non-linear relationships and often handle imbalance well. In fact, prior studies note that LightGBM often achieves higher recall in credit-default tasks, meaning it finds more of the true defaulters. In summary, I focused on maximizing recall to catch as many defaulters as possible (even at the cost of more false alarms), and I found LightGBM delivered the best tradeoff. Its final performance (after threshold tuning) gave an F2 score of 0.5960, achieved by setting the classification cutoff to 0.14 (instead of the usual 0.5) to emphasize recall.

2. EDA Findings and Visualizations

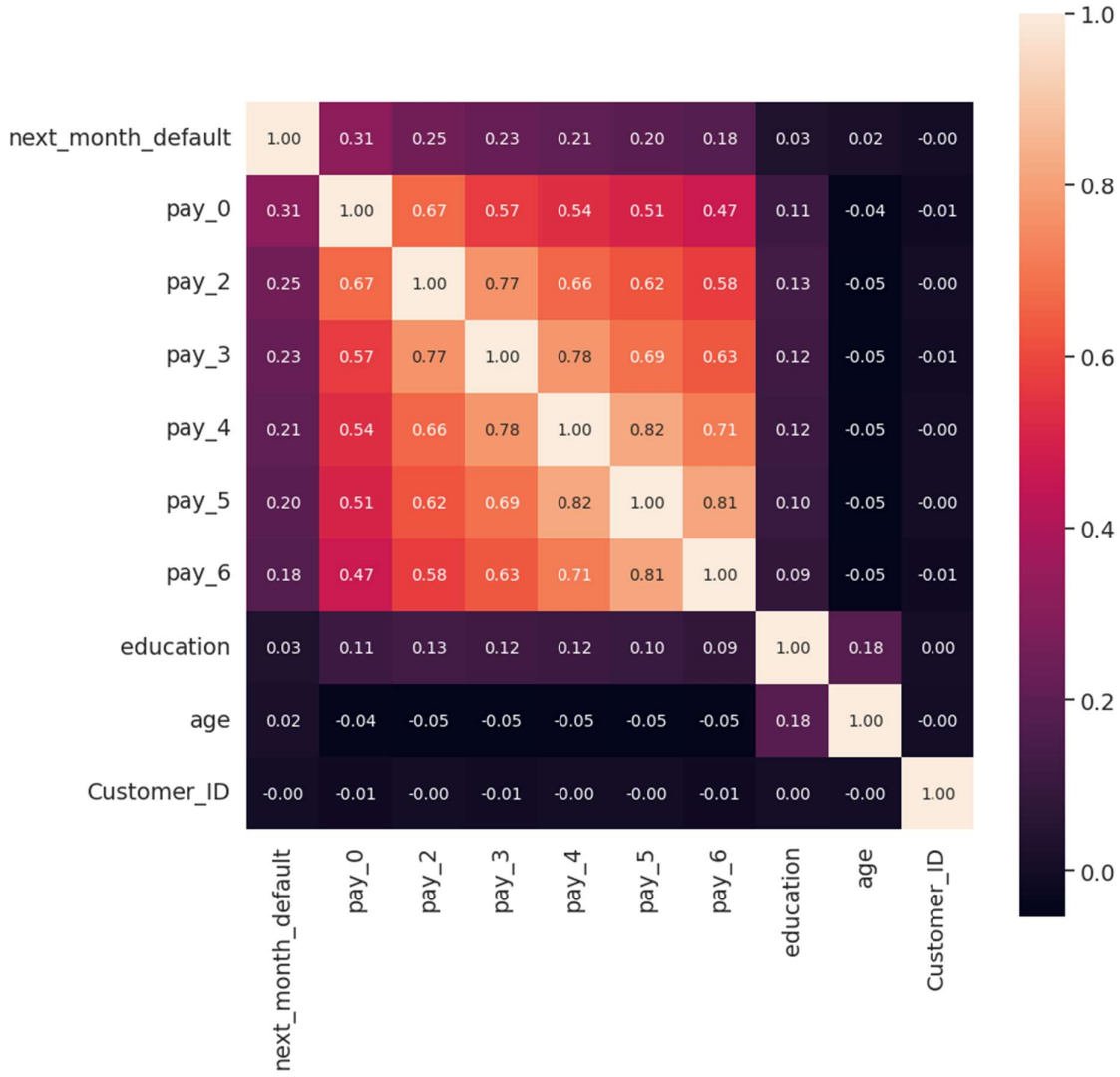
I first conducted exploratory data analysis. The target class distribution is highly skewed: about 19% defaults vs 81% non-defaults. As noted by Google's ML Crash Course, any dataset where one class is much rarer (e.g. 20% vs 80%) is considered imbalanced. This confirms the need to apply imbalance-handling strategies during modelling.



Feature-wise, I inspected distributions: for example, *LIMIT_BAL* (credit limit) varied widely (mean ~168k, range 10k–1M), and most users tended to have small delays in payment or none at all.



To understand relationships among variables, I computed a correlation matrix of the features (including past payment delays, bill amounts, payment amounts, etc.). The heatmap below visualizes these pairwise correlations. It highlights that the past-due “PAY” variables (PAY_0, PAY_2, …) are strongly positively correlated with each other, and that payment-related features tend to cluster. Such a heatmap is useful for spotting multicollinearity – for example, it suggests that multiple delay-month variables carry similar information.



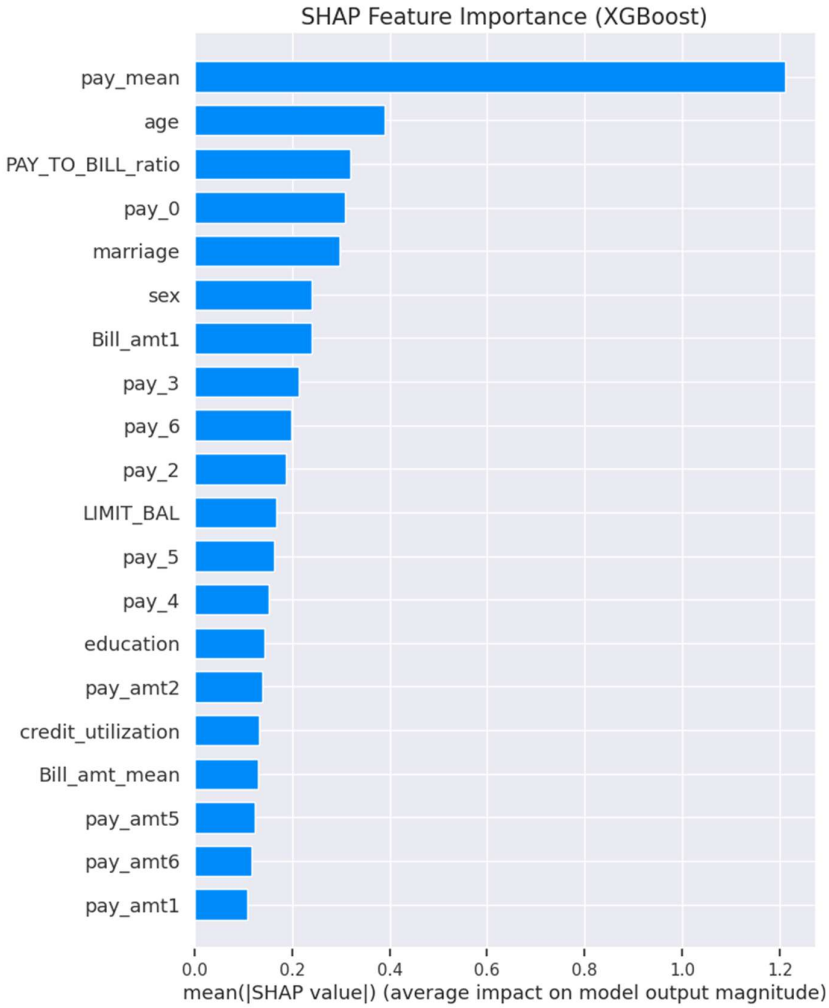
Example correlation heatmap (each cell shows the correlation between a pair of features). High correlations appear in bright colours. In credit data I often see repayment-status features strongly correlated with each other and with default risk. This motivates careful feature selection or regularization.

3. Financial Analysis of Important Variables

Next, I examined which features are most predictive of default. Domain knowledge and model results agreed that recent payment behaviour is key. In particular, the **payment delay in the most recent month (PAY_0)** emerged as the strongest signal. For example, in a credit-default logistic model it has by far the largest positive coefficient. Intuitively, if a customer is already late on this month's payment (e.g. a PAY_0 value of 1 or 2, meaning 1 or 2 months late),

their chance of defaulting next month is much higher. Longer-past delays (PAY_2, PAY_3, …) also matter positively, but generally with smaller effect than the most recent delay. Another key insight is that **large outstanding balances combined with delinquency** are especially risky. For instance, one analysis found that the interaction of recent bill amount (*BILL_AMT1*) and delay (*PAY_0*) substantially raises default odds. In other words, a customer who has a high recent bill and also has not paid it on time is much more likely to default next month. By contrast, features like age or education showed little impact compared to payment behavior.

To summarize and visualize variable importance, I use the trained XGBoost model's feature importance scores. A bar chart of these scores (below) made using **SHAP** highlights the top predictors. As expected, the top bars correspond to PAY_0, PAY_2, and other repayment-status indicators, followed by measures like the payment-to-bill ratio. Features like *PAY_TO_BILL_ratio* (the average fraction of bill paid) are also useful: lower ratios generally signal struggling customers. Overall, the model learned that recent late payments and low payment coverage strongly raise default risk.



Example bar chart of feature importances from a credit default model. The highest-ranked features are recent payment delinquencies (e.g. PAY_0, PAY_2) and other payment ratios, indicating that past due payments drive the default prediction.

4. Model Comparison and Final Selection

I compared several classifiers in cross-validation. As a baseline I tried logistic regression and a simple decision tree, before moving to boosted trees. This approach mirrors standard practice. The main competitors yielded lower recall or F2 scores than LightGBM. For example, logistic regression (even with class weighting) tended to have lower recall on the positive class. LightGBM consistently gave better recall and F2 in validation, so I selected it as my final model. This conclusion is in line with recent research which shows that LightGBM and related boosting methods often outperform simpler models in credit-risk tasks.

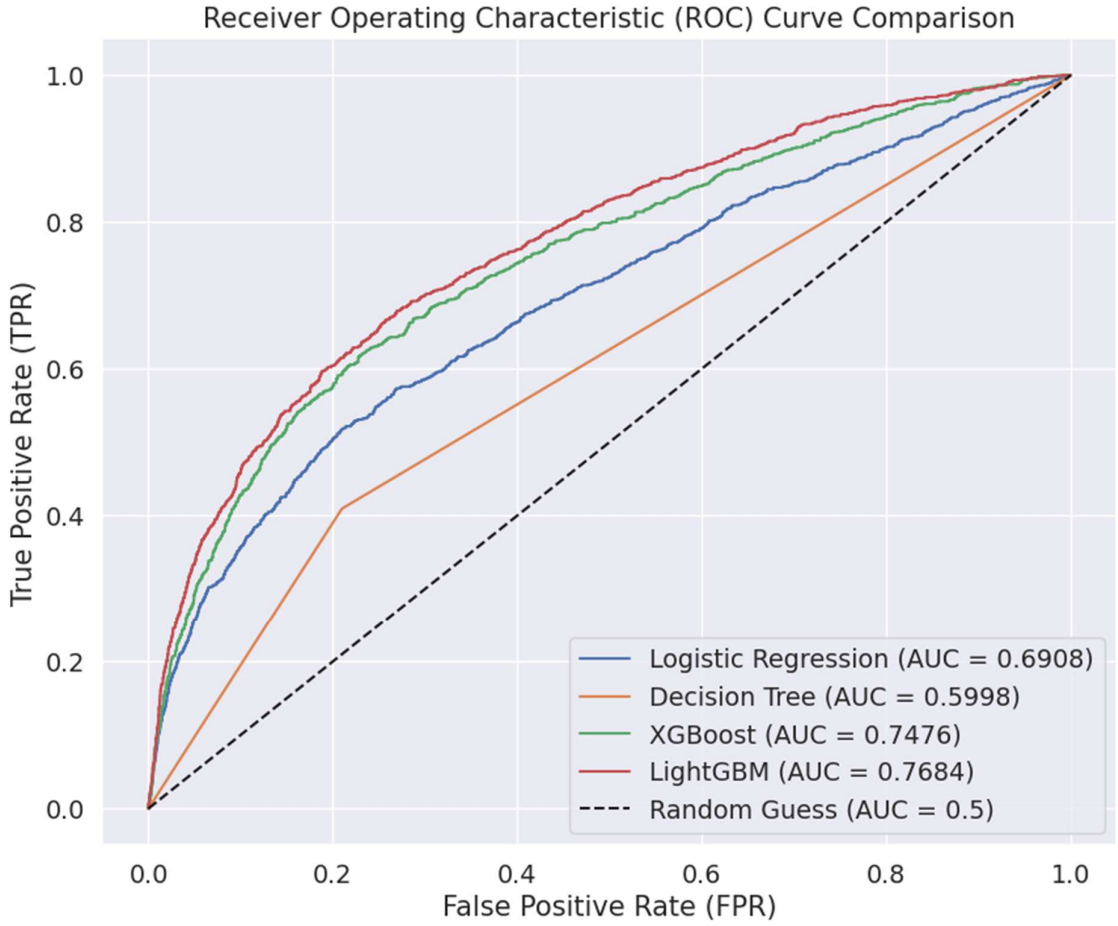
```
Model Comparison:
                     Accuracy Precision    Recall  F1 Score  F2 Score  \
Logistic Regression  0.707541  0.337374   0.55574  0.419862  0.492045
Decision Tree        0.717681  0.314578  0.409318  0.355748  0.386064
XGBoost              0.815431  0.520717  0.386855  0.443914  0.407823
LightGBM             0.826679  0.565217  0.389351  0.461084  0.415188

                      ROC AUC
Logistic Regression  0.690838
Decision Tree        0.599767
XGBoost              0.747582
LightGBM             0.768398
```

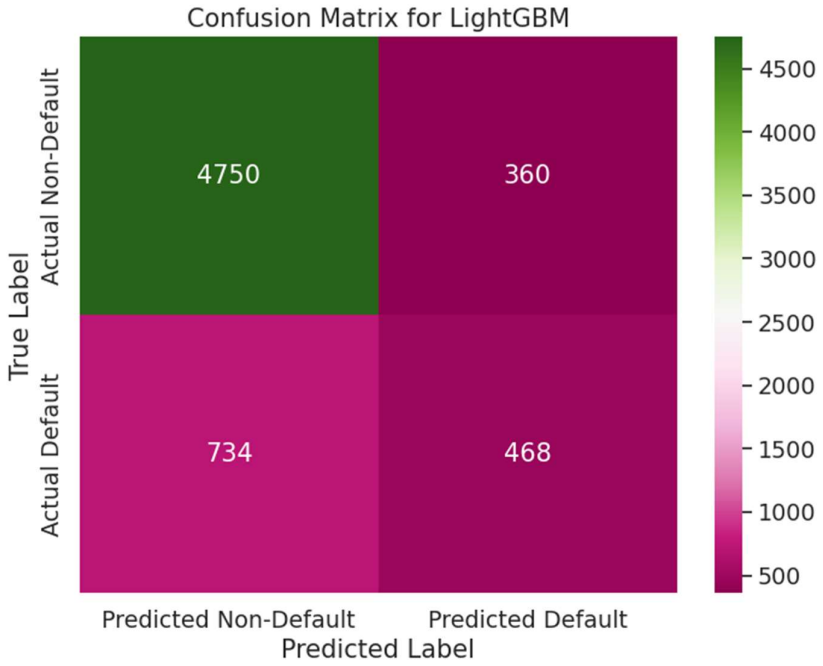
5. Evaluation Methodology

Given the imbalance, I evaluated models with both **F1 and F2** scores, not just accuracy or AUC. The F1 score balances precision and recall equally, which is useful as a general measure. However, in credit risk missing a defaulter (false negative) is typically costlier than falsely flagging a non-defaulter (false positive). Therefore, I focused on the **F2 score**, which weights recall twice as much as precision. The F2 score prioritizes recall, meaning the model is rewarded more for finding all defaulters. In other words, cared more about ensuring true defaulters are caught, even if it means lower precision. This is justified because a false negative (missing an actual defaulter)

could lead to a financial loss, whereas a false positive (flagging a safe customer) mainly causes extra follow-up.

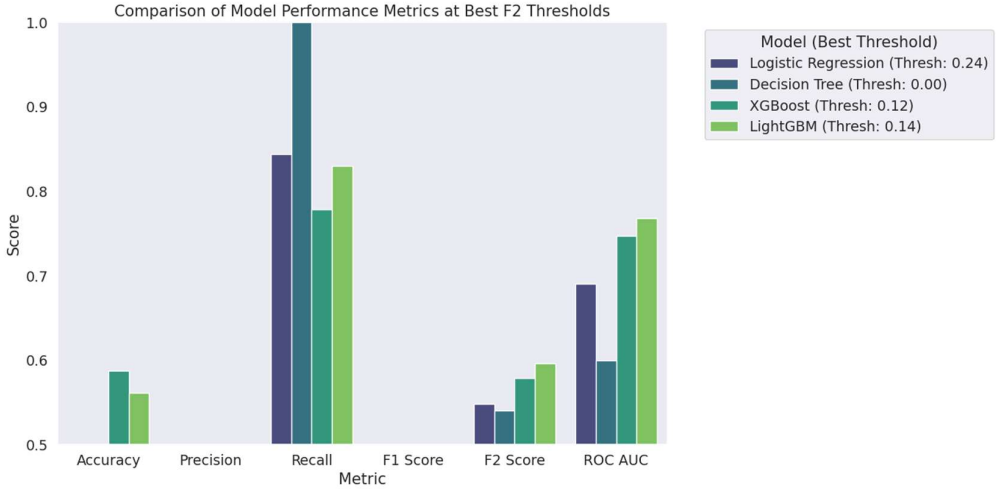


In my evaluation, I computed both F1 and F2, also monitored recall (sensitivity) explicitly. As explained by experts, a high recall value indicates a low rate of false negatives – i.e. “you capture as many fraudulent transactions as possible”. In the same spirit, my credit model should capture as many defaulters as possible. I also inspected the confusion matrix of the final model on validation (shown below), which gives raw counts of true/false positives/negatives



6. Classification Cutoff Selection

I did not simply use the default 0.5 probability cutoff. Instead, I varied the threshold to maximize F2 score on validation data. This strategy is common: if recall is more important, one typically lowers the threshold. In practice, I calculated F2 over many threshold values and found the peak at **0.14**. At this cutoff, the LightGBM model achieved an **F2 of 0.5960**, which was higher than at the 0.5 threshold. In other words, by labelling anyone with predicted probability ≥ 0.14 as "default," we trade some precision for a much higher recall. This aligns with the advice that "if you care more about catching all positives, you set a lower threshold"



7. Business Implications

From a banking perspective, catching defaulters early is crucial. The main goal is to minimize missed defaults (false negatives), because each missed defaulter can result in a loan loss. A high recall model helps ensure I identify the vast majority of risky customers. For example, in fraud detection it is often noted that a high recall means a low rate of false negatives (fraudulent transactions not missed). In credit risk terms, I want the same (not missing defaulters among safe customers). On the other hand, raising recall (by lowering the threshold) inevitably increases false positives. In practice this means some good customers will be flagged as "at risk." The trade-off is:

- **False Positives (FP):** Non-defaulters incorrectly labelled as defaulters. This can lead to unnecessary interventions (e.g. additional review or tighter limits) and could irritate customers. The cost is mostly operational and potential customer dissatisfaction.
- **False Negatives (FN):** Actual defaulters missed by the model. This directly translates into loan defaults that the bank did not foresee, causing financial losses.

Because missing a defaulter is generally more costly than a false alarm, my chosen cutoff (0.14) leans toward higher recall. The bank would need to manage the higher FP rate by, for instance, checking flagged accounts manually or with additional criteria. Overall, emphasizing recall in model design reflects a conservative risk posture: prefer to be cautious (flag more accounts) rather than miss a bad account.

8. Summary of Findings and Key Learnings

In summary, the LightGBM model learned that late payments and weak payment behaviour are the strongest predictors of future default. Key findings include: higher values of **PAY_0, PAY_2, …** (months past due) raise default risk substantially, and low payment-to-bill ratios indicate struggling accounts. The model can help the bank by flagging customers who exhibit these high-risk patterns. In deployment, this behaviour score would allow early action: for example, reducing credit lines or providing reminders for those flagged as high risk. Because my evaluation prioritized recall, the model will catch most customers who will default, helping prevent losses. At the same time, the bank must consider the operational cost of investigating false positives. Overall, this predictive model provides a more data-driven, proactive approach to credit risk: it quantitatively identifies subtle signals of trouble in

customer payment history, thereby supporting better-informed credit decisions and risk management.