Finance Club Open Project Summer 2025

Name **– Kris Keshav**

Enrollment Number **– 23115068**

Branch and Year **– Electrical 3Y**

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

Overview

In this project I have built an effective model to predict whether a credit card holder will default on their next payment. My goal was to optimize recall and F2-score for defaulters (minority class), as they are the most financially impactful. I tried various models including Logistic Regression, XGBoost, LightGBM, Ensemble models and Neural Networks in the last. I also evaluated weighted ensemble and a fine-tuned LightGBM model, with the fine-tuned LightGBM selected for final predictions.

Problem Statement

Financial institutions face losses due to credit card defaulters. Predicting defaulters allows for proactive risk mitigation. This binary classification task labels customers as 1 (default) or 0 (no default) for the next month.

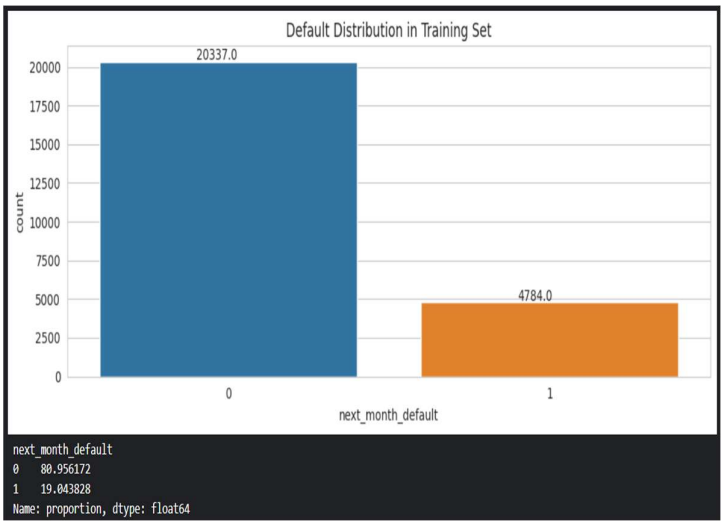
Dataset Overview

```
Train shape: (25247, 27)
Validation shape: (5016, 26)
```

	Customer_ID	marriage	sex	education	LIMIT_BAL	age	pay_0	pay_2	pay_3	pay_4	...	Bill_amt6	pay_amt1	pay_amt2	pay_amt3	pay_amt4	pay_amt5	pay_amt6	AVG_Bill_amt	PAY_TO_BILL_ra
0	5017	2	0	2	60000	25.0	2	2	2	0	...	20750.63	2000.21	0.00	1134.85	1821.78	1500.03	1500.24	41511.50	0.
1	5018	2	1	1	290000	24.0	0	0	-2	-2	...	1350.30	0.00	0.17	0.00	2700.10	0.00	1349.72	2534.50	0.
2	5019	1	0	2	180000	63.0	0	0	0	0	...	52991.51	2086.94	2199.99	1845.66	2000.35	1923.00	1999.78	50422.00	0.
3	5020	1	1	2	210000	43.0	0	0	0	0	...	76945.47	3348.07	3380.91	3400.45	2683.97	2744.00	2892.10	86229.50	0.
4	5021	2	0	1	280000	32.0	-2	-2	-2	-2	...	1.35	999.78	3186.27	45027.78	2100.09	0.01	0.27	11814.33	0.

5 rows × 27 columns

- **Train set:** 25,247 rows, 27 columns
- **Validation set:** 5,016 rows, 26 columns
- **Target variable:** next_month_default
- **Class imbalance:** 80.96% no-default, 19.04% default

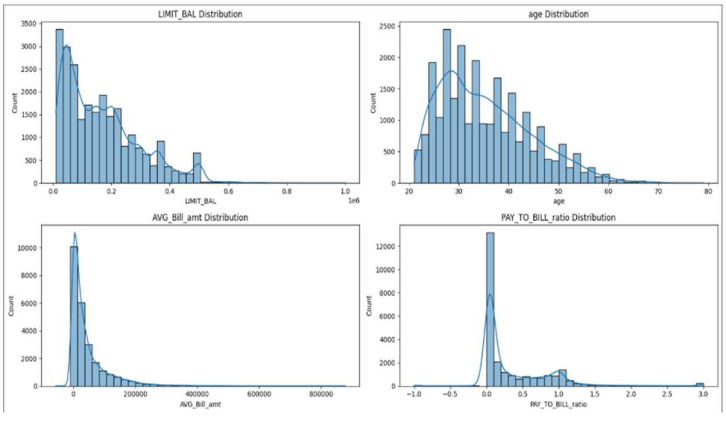


Data Preprocessing

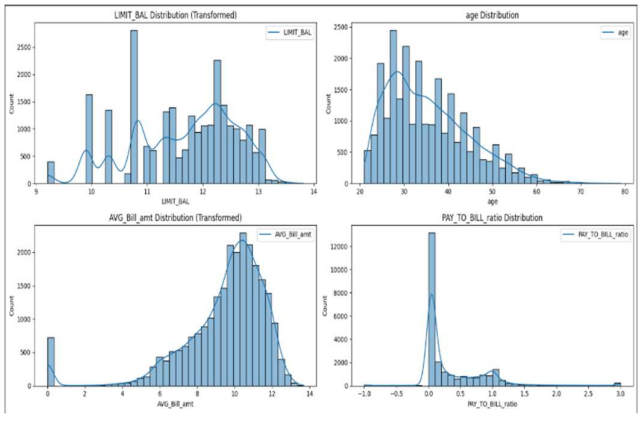
- I found out that there were no missing values in the dataset
- Then I did feature scaling using StandardScaler
- Then I removed identifier columns

Exploratory Data Analysis (EDA)

- I plotted distributions for features like LIMIT_BAL, AVG_Bill_amt, and PAY_TO_BILL_ratio and applied **logp** transformation for gaining better insights.

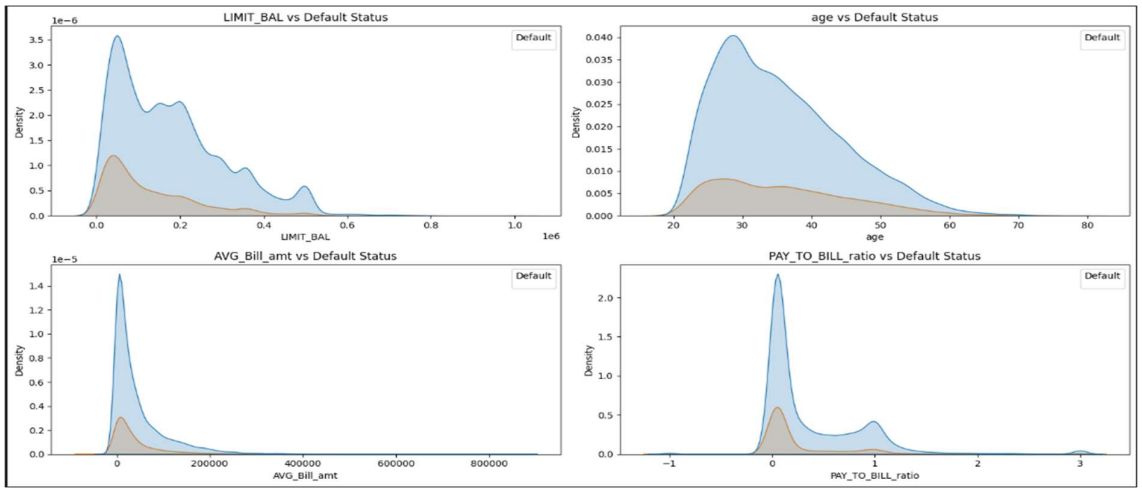


Before Transformation

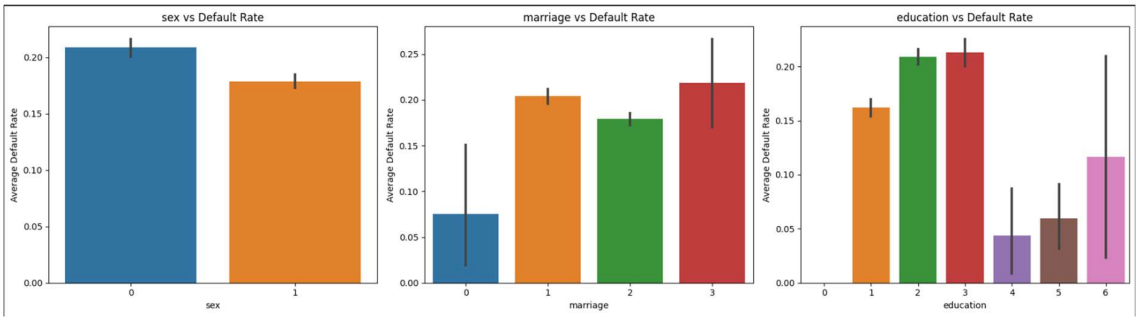


After Transformation

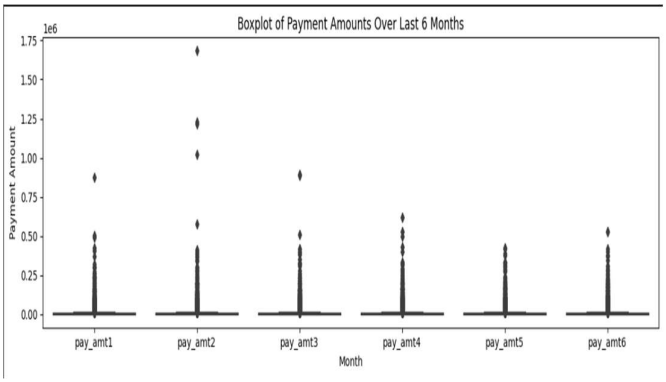
- KDE plots showed feature separability across default status

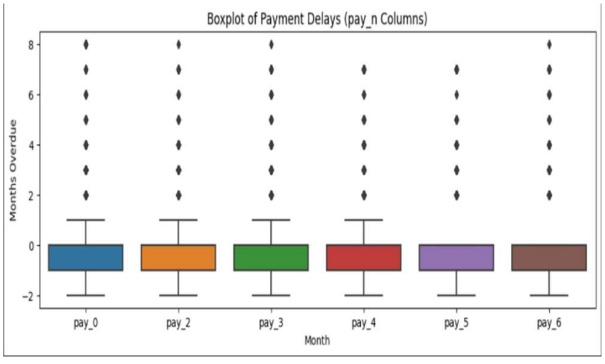


- I also analysed categorical features (sex, education, marriage) against default rate



- Then I identified and visualized payment behaviour using payment amounts and payment delays (showed more dependence).

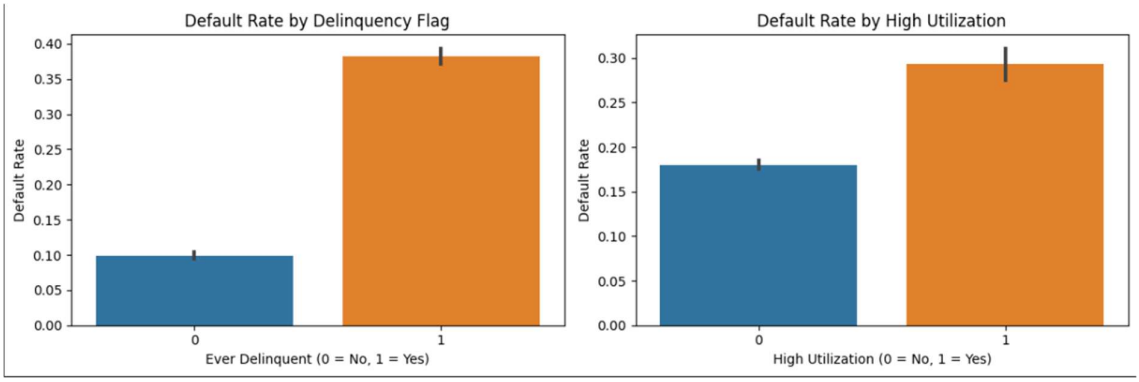




Feature Engineering

I created the following features for extracting more information and used them for making better predictions--

- delinquency_count: number of months delayed
- high_utilization: flag for >90% credit usage
- ever_delinquent, max_delay, delay_std



These engineered features reflect key financial risk indicators used in credit scoring:

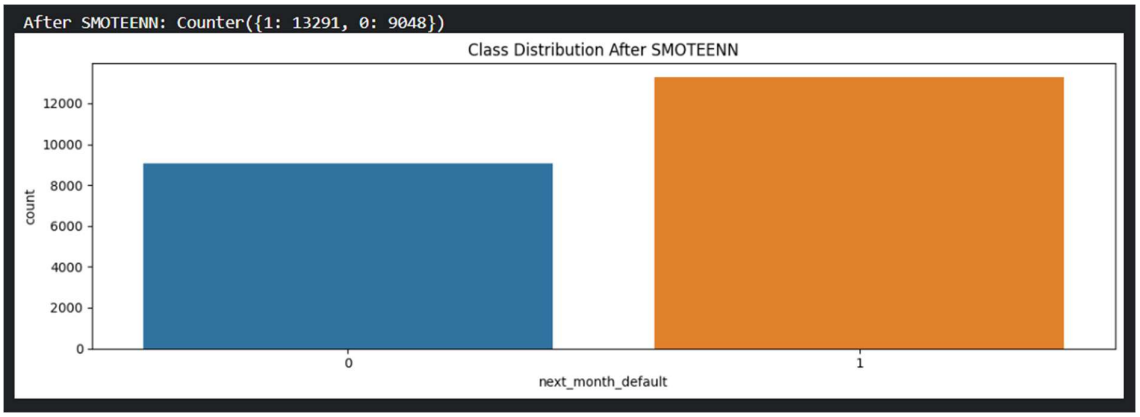
- **delinquency_count** captures how often a customer delayed payments, indicating habitual risk.
- **high_utilization** flags customers using over 90% of their credit limit, suggesting financial stress or over-reliance on credit.
- **ever_delinquent** marks if the customer was ever late, as any past delinquency raises risk flags for lenders.
- **max_delay** shows the worst-case delay, helping identify customers who faced serious financial distress.
- **delay_std** measures inconsistency in payment behavior, which often correlates with income instability.

Together, these features model a customer's **credit discipline, risk exposure, and financial reliability** — all critical for predicting default.

Next I did Class Imbalance Handling (as the dataset was highly imbalanced**) + Training**

- *Step 1: Handling Class Imbalance with SMOTE*
- *Step 2: Training Logistic Regression and XGBoost*
- *Step 3: Evaluation & Metric Justification*
  - Used SMOTE to oversample the minority class
  - Trained models on the balanced data
  - Used F2-score, recall, and ROC AUC for evaluation (not accuracy) as accuracy is not an appropriate metric for such classification tasks
  - Tuned classification threshold for different models
  - Tested on the original imbalanced test set (X_test) to avoid biased validation

Here it was needed to first install imbalanced-learn library and then the session was restarted. (later I shifted this part to start to avoid starting again and again)



```
===== Logistic Regression (SMOTEENN + F2 Optimized) =====
Best Threshold: 0.3468
Best F2 Score: 0.5502
ROC AUC: 0.7182
Classification Report:
              precision    recall  f1-score   support

           0     0.9033    0.2847    0.4329      4068
           1     0.2225    0.8704    0.3545       957

    accuracy                         0.3962      5025
   macro avg     0.5629    0.5775    0.3937      5025
weighted avg     0.7736    0.3962    0.4180      5025
```

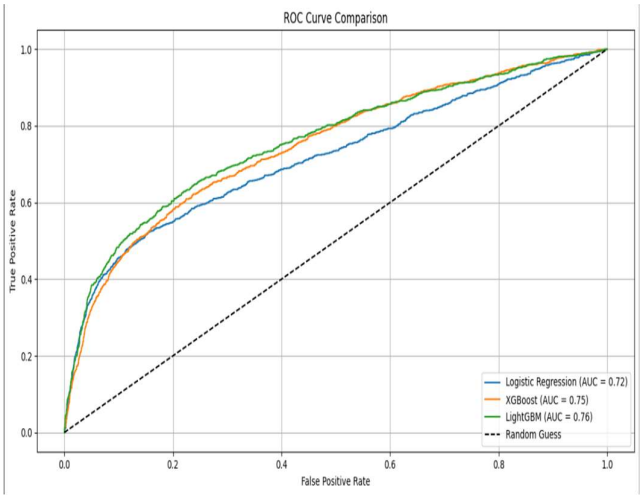
```
===== XGBoost (F2 Optimized) =====
Best Threshold: 0.1333
Best F2 Score: 0.5904

Classification Report:
              precision    recall  f1-score   support

           0     0.9224    0.4941    0.6435      4068
           1     0.2769    0.8234    0.4144       957

    accuracy                         0.5568      5025
   macro avg     0.5997    0.6588    0.5290      5025
weighted avg     0.7995    0.5568    0.5999      5025
```

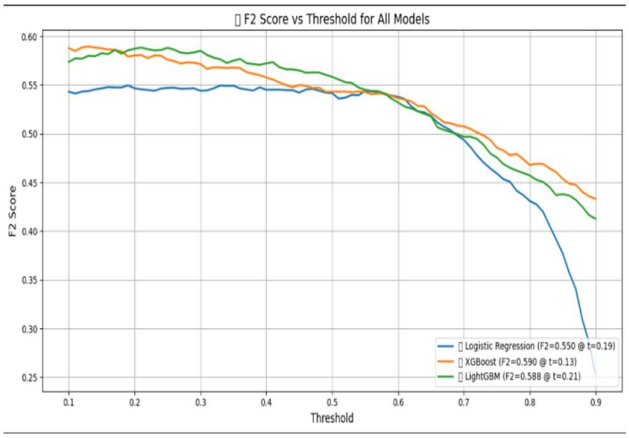
```
===== LightGBM (F2 Optimized) =====
Best Threshold: 0.2529
Best F2 Score: 0.5896
ROC AUC: 0.7652

Classification Report:
              precision    recall  f1-score   support

           0     0.9128    0.6332    0.7478      4068
           1     0.3227    0.7429    0.4500       957

    accuracy                         0.6541      5025
   macro avg     0.6178    0.6881    0.5989      5025
weighted avg     0.8004    0.6541    0.6910      5025
```

```
✅ Best Threshold (Ensemble): 0.32
⭐ Best F2 Score (Ensemble): 0.5877

📋 Classification Report (Ensemble @ Best Threshold):
              precision    recall  f1-score   support

           0     0.9132    0.6104    0.7317      4068
           1     0.3127    0.7534    0.4419       957

    accuracy                         0.6376      5025
   macro avg     0.6129    0.6819    0.5868      5025
weighted avg     0.7988    0.6376    0.6765      5025
```

Models Tried

- Logistic Regression
- XGBoost
- LightGBM (default and tuned via Optuna)
- Simple and Weighted Soft Voting Ensemble (0.2 LR + 0.3 XGB + 0.5 LGBM) but that weighted ensemble didn't performed well so I **shifted to**





simple ensemble which yielded better results

- **Neural Network** (Keras MLP with dropout and batch normalization) (**performed almost similar but more consistent** than finetuned LightGBM)

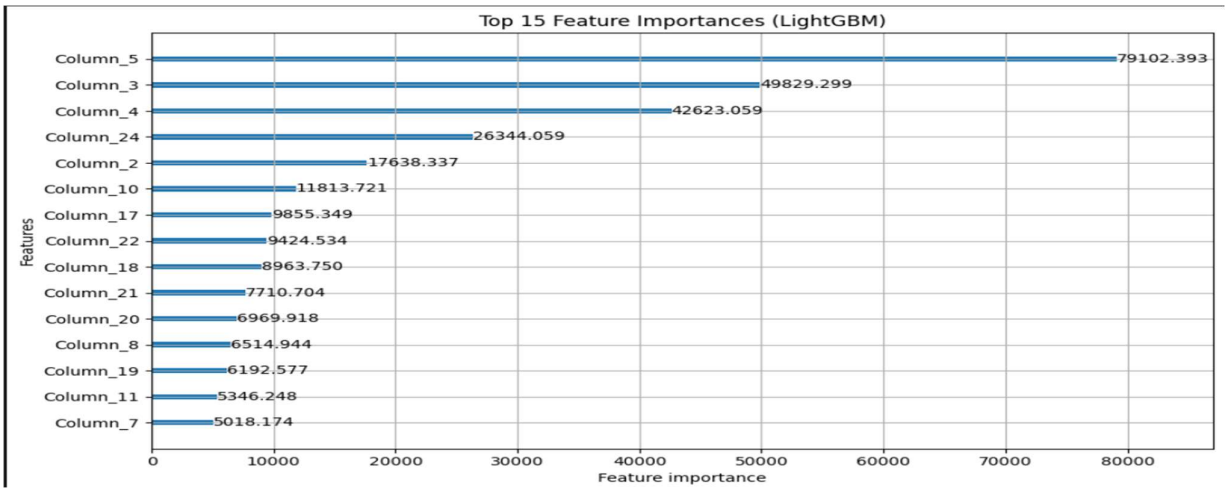
Threshold Optimization

- Tested thresholds from 0.1 to 0.9 tested to maximize F1
- Final tuned thresholds:
  - LightGBM: 0.246
  - Ensemble: 0.32
  - Neural Network: 0.303

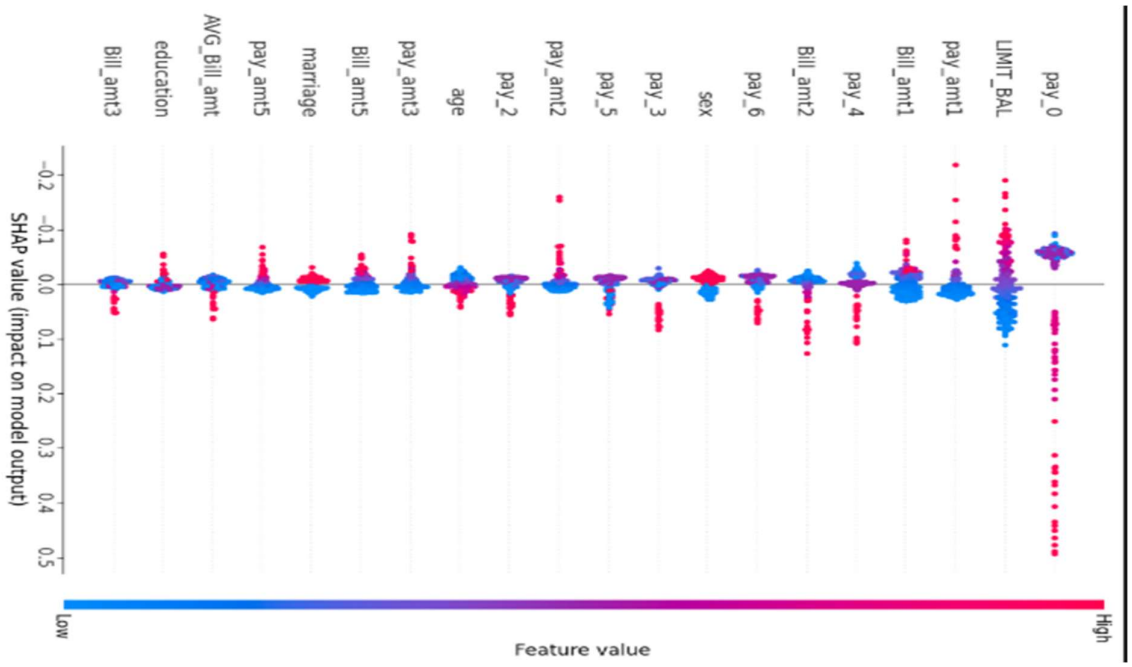
Performance Summary

Model	F2 (Class 1)	ROC AUC	Best Threshold
Logistic Regression	0.550	0.718	0.347
XGBoost	0.591	0.759	0.13
LightGBM (default)	0.590	0.765	0.253
LightGBM (tuned)	**0.600**	**0.771**	0.246
Neural Network	**0.595**	**0.768**	0.303
Ensemble	0.588	0.759	0.32

Explainability



- LightGBM: plotted gain-based feature importance
- Neural Net: SHAP values via KernelExplainer



Final Model Selection

- **Model selected**: LightGBM and Neural Network (outperformed LR, Ensemble and XGBoost)
- **Threshold used**: 0.246 (for Ensemble) and 0.303 (for Neural Network)
- **Saved artifacts**:

- lgbm_tuned_final_predictions.csv
- nn_predictions.csv
- **Output CSV**: lgbm_tuned_final_predictions.csv (Customer_ID, next_month_default)

Business Implications

- **Early Risk Detection**: This model enables timely identification of potential defaulters, allowing proactive actions like limit reductions or warnings.
- **Smarter Credit Decisions**: Features like high utilization and payment delays allow dynamic credit line management — increasing limits for low-risk users and flagging risky ones.
- **Personalized Pricing**: Segmented risk profiles allow interest rate adjustments and credit offers tailored to customer behavior.
- **Operational Efficiency**: Automating risk detection reduces manual checks, helping focus resources on high-risk cases.
- **Regulatory Readiness**: Explainable models (e.g., LightGBM + SHAP) support transparency in credit decisions — critical for compliance.

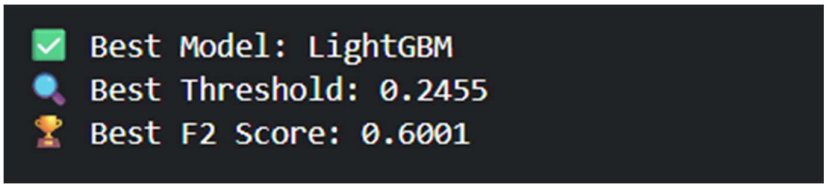
Summary of Findings and Key Learnings

- **Behavioural Features Matter**: Variables like delinquency_count and PAY_TO_BILL_ratio are strong predictors of default.
- **Model Performance**: LightGBM and Neural Networks gave the best F2-scores for defaulters. Threshold tuning helped balance precision and recall (high recall was required in this case as we cannot afford to miss defaulters in practical life).

- **SMOTE Improved Results**: Handling class imbalance was key to improving minority class (defaulter).
- **Explainability Builds Trust**: SHAP and feature importance made model decisions interpretable for business stakeholders.
- **Actionable Insights**: EDA findings translated directly into business strategies like customer segmentation and early interventions.

Conclusion

- SMOTE(nn) helped mitigate imbalance effectively
- LightGBM and Neural Networks yielded strong F2-scores with better recall
- LightGBM and Neural Network model chosen for its explainability and stable performance
- Final best result came from LightGBM with a rigorous hyperparameter optimization but neural networks were more consistently giving good results.



Future Improvements

- We can add more behavioral and income features but that would require a deeper knowledge of finance
- Using cost-sensitive learning
- Exploring AutoML and deep ensembles (not much feasible to implement on small scale due to limited resources)