CREDIT CARD DEFAULT PREDICTION

Project Report

Author

Tanuj Kumawat

(Enrollment no. 23112106)

Date

16 June 2025

1. Problem Statement / Objective

This project aims to predict whether a credit card customer will default in the upcoming month based on historical behavioral and demographic data. Accurately identifying potential defaulters is critical for risk management and financial planning in lending institutions like banks.

2. Dataset Description

- Source: Dataset loaded from Google Drive via Google Colab (provided in the problem statement by Finance Club, IITR)
- Rows: 25,247 records.
- Columns: 27 original features including demographics (age, sex, education, marriage), payment history, billing amounts, and the target variable (`next_month_default`).

3. Data Cleaning & Preprocessing

- There are 0 duplicated values and 126 null values in age column, otherwise no null values. Nulls in age column were replaced with median age.
- Checked the categorical variables, if there is any encoding other than expected. Marriage should have:
  1. 1 for married
  2. 2 for single
  3. And 3 for others.

  Similarly, education should have 1, 2, 3 and 4. But marriage contained 0's and education contained 0, 5 and 6.
- So, for marriage, mapped 0 to 3 (others), which is common in these cases. And, education values 0, 5, 6 are mapped to 4 (others).
- The columns AVG_bill_amt and PAY_TO_BILL_ratio have errors. They are not calculated properly. No value in pay_amt 's is negative but avg bill amounts are negative. Also this column has values which do not match with the average of bill amounts. That's why.new AVG_bill_amt is calculated using the Bill_amt columns .
- Similarly, new pay to bill ratio was made according to new averages.

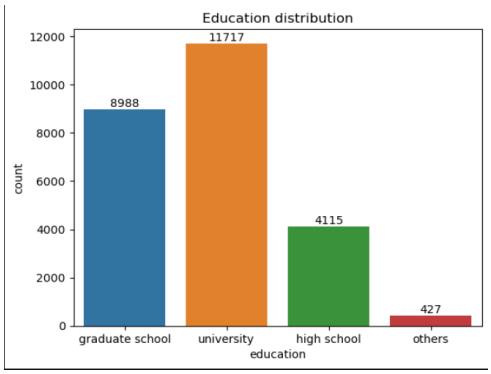
- For numerical features, the majority of distributions are right-skewed. The distribution of all the bill amounts and pay amounts is highly skewed to the right. It demonstrates that these columns have many outliers.

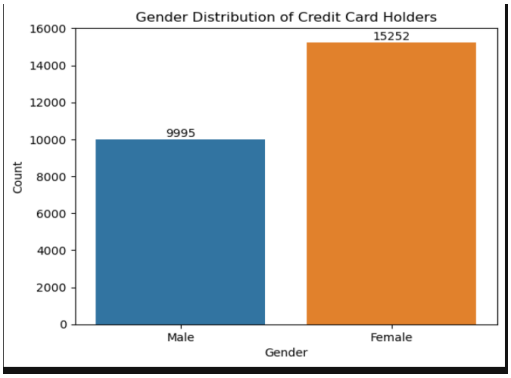
- Rename pay_0 to pay_1 for consistency across the analysis.
- Most of the variables can get a normal distribution when outliers are handled by **Clipping Method**.

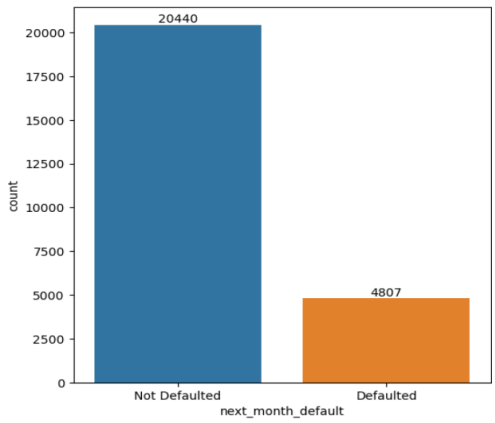
4. Exploratory Data Analysis (EDA):

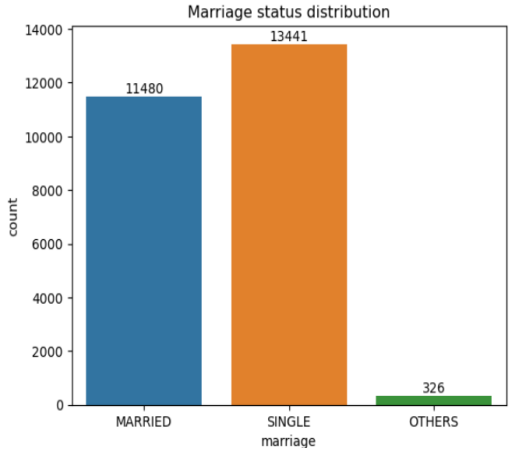
- Univariate and bivariate analyses were performed.
- Histograms and count plots for distributions.
- Heatmap revealed moderate correlation among payment history variables.

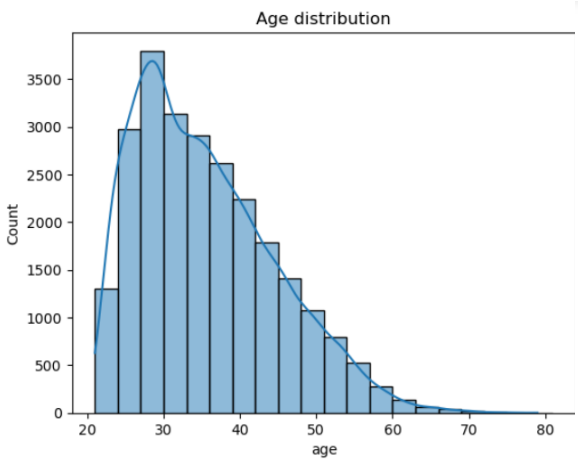
Univariate Analyses graphs and Result :

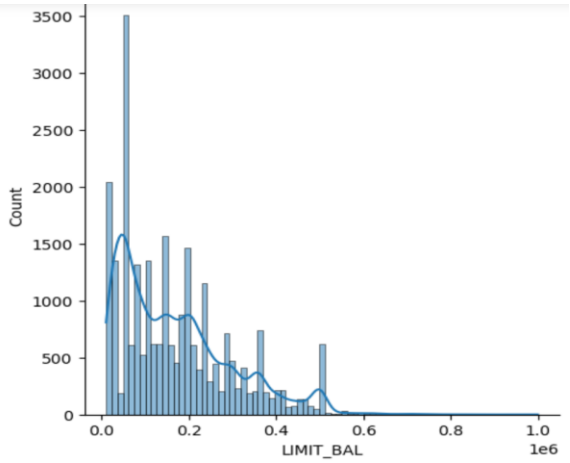






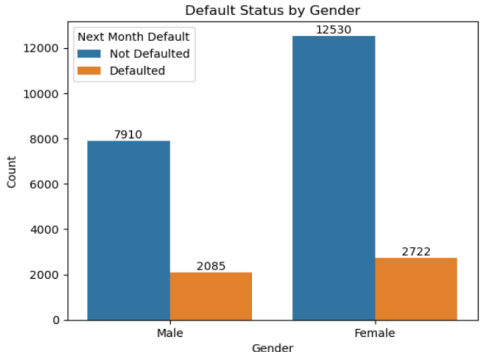


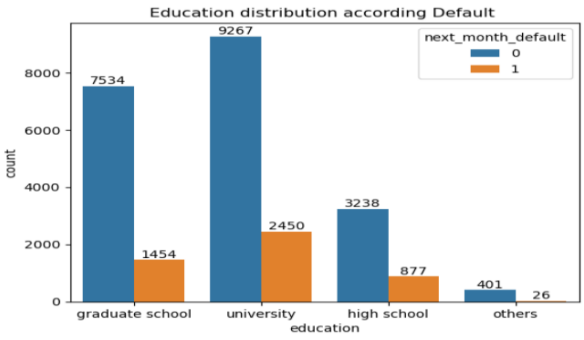


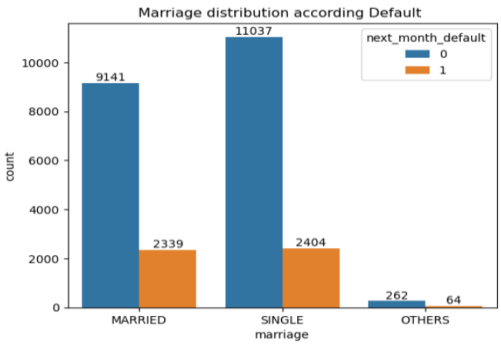


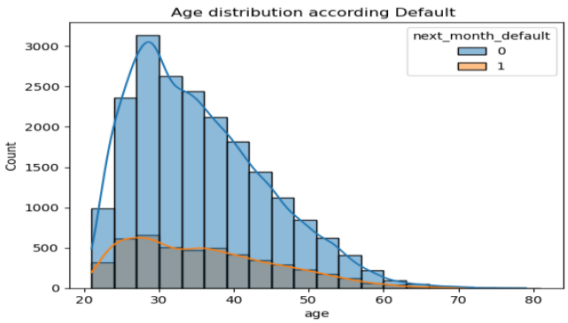
- It is clear from the graph that most of the people in the dataset are male, university level educated, single and have a mode age of 30.

Bivariate Analyses graphs and Results:









- Married people are the group which has highest fraud cases(20.3%), High School students are the group which has highest fraud cases (21.3%) , follow by University student(20.9%),20.8% of male clients fraud credit card while the ratio for female is around 17.8%.

5.Feature Engineering :

Created features related to:

- **credit_utilization**:

*M*easures the proportion of the credit limit that a customer has utilized, indicating their dependency on credit.

- **delinquency_streak**:

Counts the number of months a customer has delayed payments, reflecting payment irregularity.

- **on_time_rate**:

Represents the percentage of months where the customer paid dues on time or made the minimum payment.

- **late_payment_rate**:

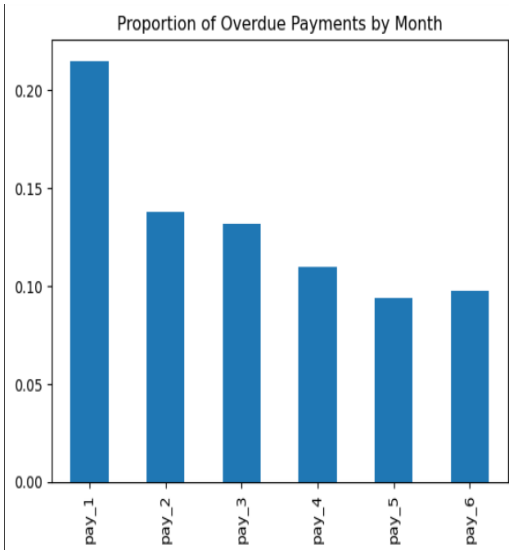
Denotes the proportion of months the customer delayed payments beyond the due date.

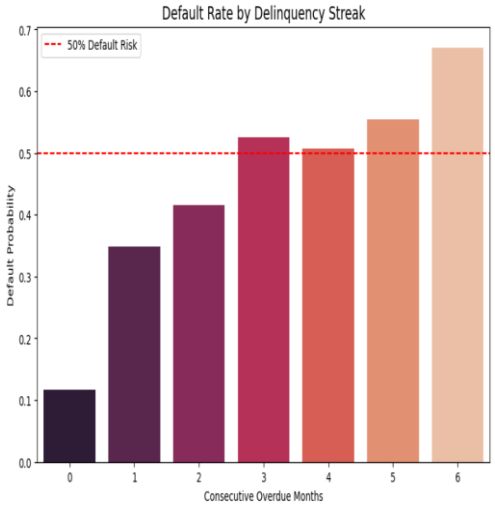
- **min_payment_rate**:

Indicates the frequency of months where only the minimum due was paid, showing potential financial stress.

- **max_late_streak**:

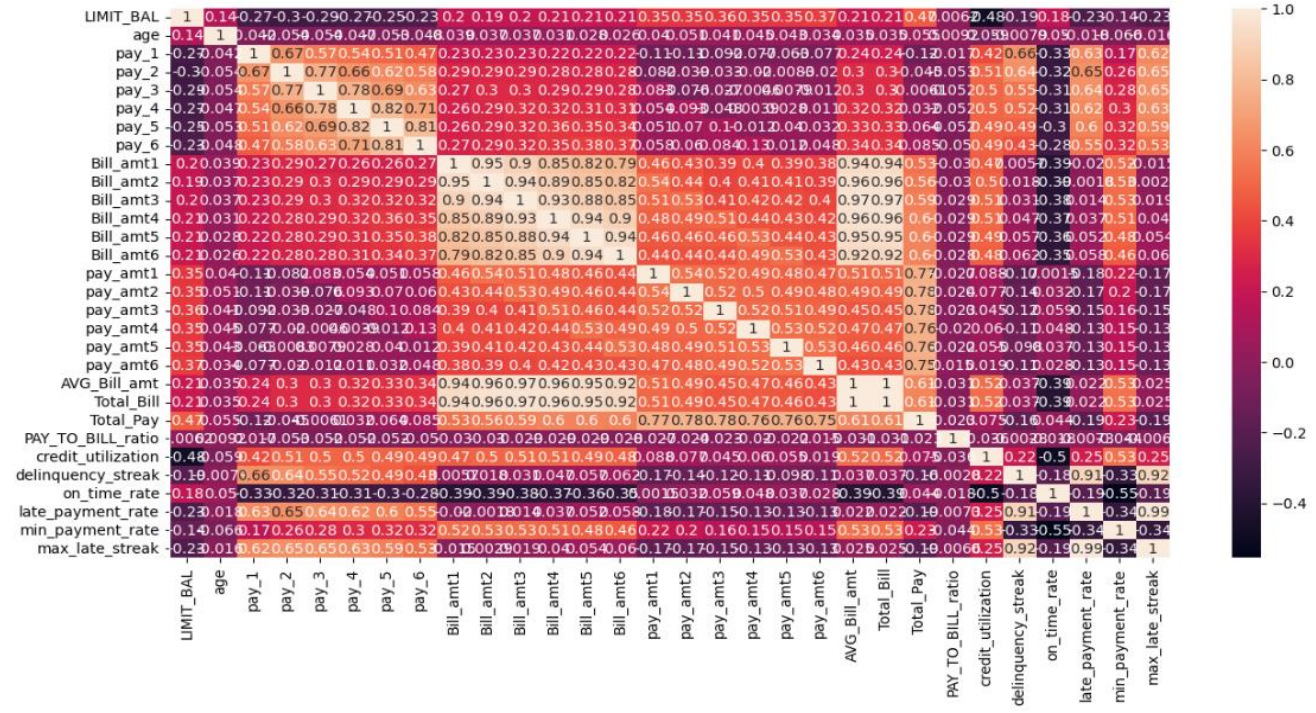
Captures the longest continuous sequence of late payments, highlighting persistent delinquency behavior.

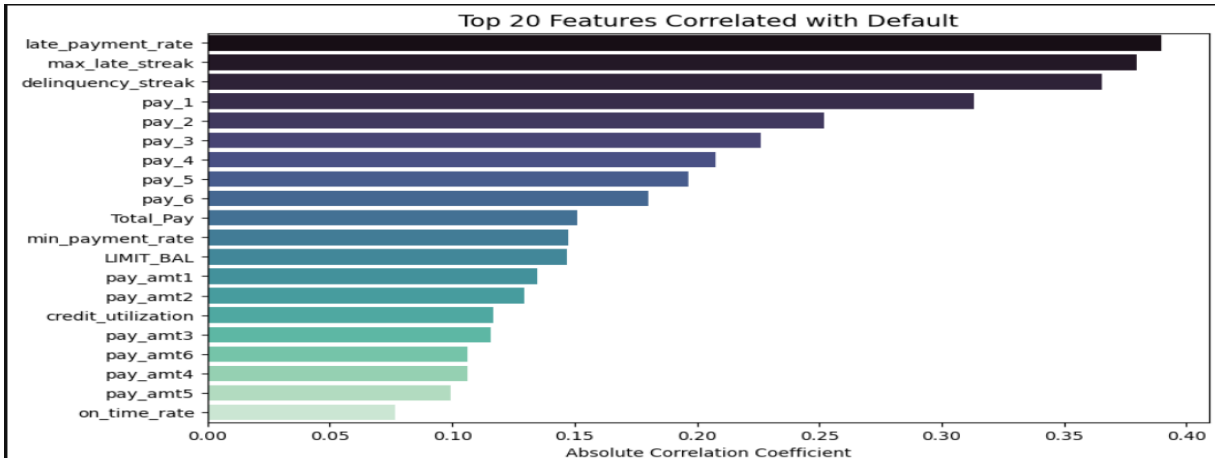




Correlation Coefficient and Heatmap:

- The correlation coefficient is an important tool in data analysis and machine learning, as it can help to identify relationships between variables and can be used in feature selection techniques to remove highly correlated features, which can reduce overfitting and improve the performance of the model.





6. Feature Encoding & Standardisation

Applied **one-hot encoding to categorical variables** and **Standard Scaler Method** to all Numerical Variables.

7. Handling Imbalance

The dataset is imbalanced:

Next Month Default	Count
0	20440
1	4807

Next month default true values are very less (19%) compared to false values. This can be handled using SMOTE (Synthetic Minority Oversampling Technique). SMOTE was applied on training data (X_train, y_train). SMOTE made the distribution of 0 and 1 equal in the training split.

Before SMOTE:

0	**16352**
1	3845

After SMOTE:

0	**16352**
1	16352

8. Model Building

Models used:

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

Train-Test split of 80-20 was used. Then, Hyperparameter tuning was done for all the above mentioned models using GridSearchCV. The grid is in the project notebook.

- Scoring for tuning is done according to F2 score. That means we selected those hyperparameters which gave the most F2 Score.

- $F2\ Score = (1 + 2^2)\frac{Precision.Recall}{4Precision+Recall}$
- F2 Score gives more importance to Recall. Recall is sensitive to False negative predictions. It is important in this project as a bank or any lending institution does not want to misclassify any Default customer. This comes in the risk management of the institutions. Recall is given importance due to the risk of leaving a default customer.

Following tabulates the best parameters:

Logistic Regression	{'C': 10, 'penalty': 'l2'}
Decision Tree	{'criterion': 'gini', 'max_depth':5 None, 'min_samples_leaf': 1, 'min_samples_split': 2}
Random Forest	{'max_depth': 70, 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 75}
XGBoost	{ 'learning_rate': 0.1, 'max_depth': 6, 'n_estimators': 75, 'min_samples_split': 5, 'min_samples_leaf': 6}
Light GBM	{'max_depth': 7, 'min_samples_leaf': 6, 'min_samples_split': 4, 'n_estimators': 100}

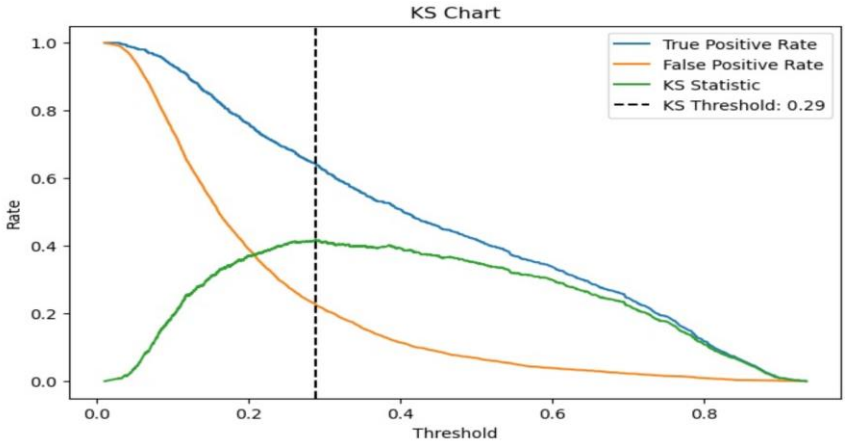
9. Model Evaluation

All models were evaluated on the basis of ScikitLearn's Classification report, train and test accuracy,Recall,F1score, ROC-AUC score and F2 score. Results are tabulated below.

Models	Train accuracy	Test Accuracy	Recall	F1 Score	F2 Score	Auc_roc
LogisticRegression	0.698722	0.760990	0.628898	0.500621	0.570432	0.765312
DecisionTree	0.758623	0.777228	0.592516	0.503311	0.553291	0.754765
RandomForest	0.960433	0.724158	0.660083	0.476906	0.572175	0.769216
XGBoost	0.855431	0.765149	0.633056	0.506656	0.575614	0.772255
LightGBM	0.865001	0.749505	0.643451	0.494606	0.574318	0.770789

- LightGBM selected as the final model based on superior Test F2-score (0.574) and high Recall (64.34%), aligning with the business objective of minimizing False Negatives (missed detections).
- Random Forest discarded due to overfitting (Train Accuracy 96% vs Test 72%).
- Thresholding based on KS-statistic optimized Recall-F2 trade-off.
- Threshold optimized for KS statistic, ensuring best separation between classes.

- Improved Recall & F2 Scores at the cost of slight drop in Precision — acceptable in high Recall-focused problems.



10. Model Comparison

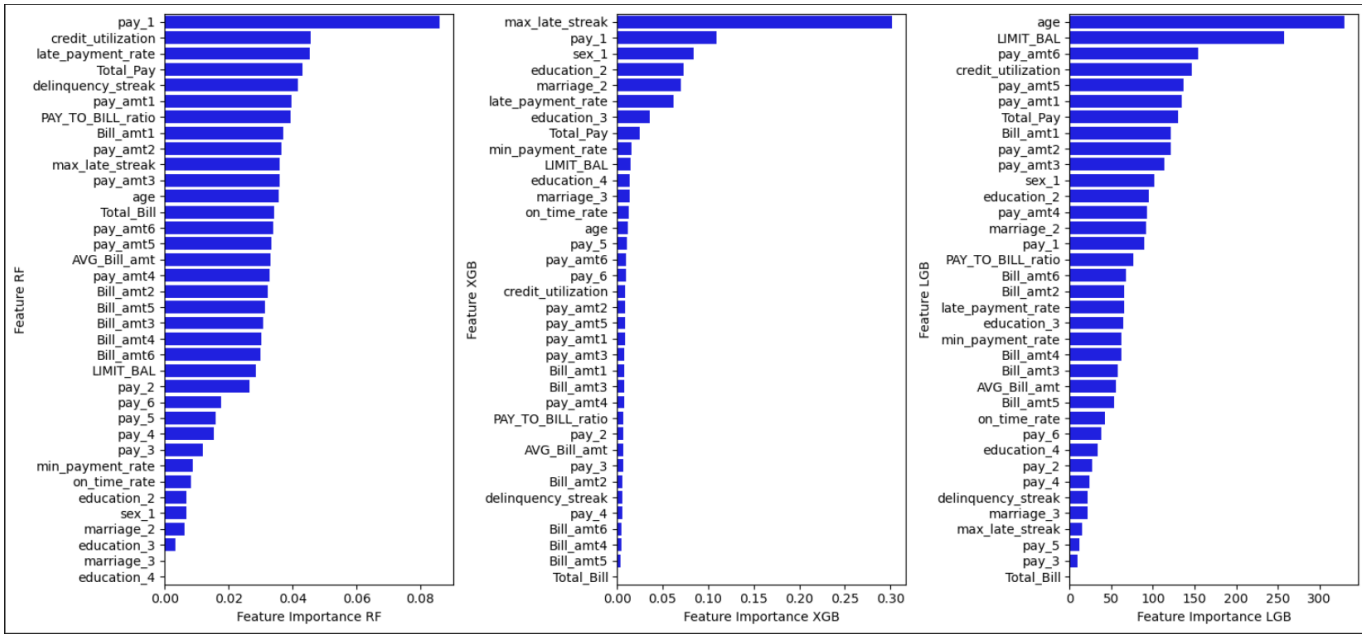
- Random Forest
  Best F2-Score: 0.572
  Accuracy at best threshold: 0.724
  Threshold: 0.317
- XGBoost
  Best F2-Score: 0.576
  Accuracy at best threshold: 0.765
  Threshold: 0.347

- LightGBM
  Best F2-Score: 0.574
  Accuracy at best threshold: 0.75
  Threshold: 0.29

We will use LightGBM as final model for future predictions.

11.Model Explanability (Feature Importance order in terms of Bar Chart)

- **Random Forest (RF):**
  - Top: `pay_1`, `credit_utilization`, `late_payment_rate`.
  - Focus on **payment behavior & utilization**.
- **XGBoost (XGB):**
  - Top: `max_late_streak`, `pay_1`, `sex_1`.
  - Stresses on **late streak & demographic factors**.
- **LightGBM (LGB):**
  - Top: `age`, `LIMIT_BAL`, `pay_amt6`.
  - Strongly influenced by **age & credit limit**.



11. Application of Validation dataset

We used the trained random forest on provided validation dataset to do predictions. The predictions csv is also attached. The percentage of predicted defaults in validation csv were 12.89%.

12. Financial Analysis, Business Insights & Recommendations

Business Insights

1. Risk Assessment

- Payment delays (especially pay_0) are strong predictors of default risk
- Customers with lower credit limits may be higher risk
- Younger customers (age 24-44) appear more frequently in the data

2. Customer Segmentation

- Could segment customers by:
  - Credit limit ranges
  - Payment behavior patterns
  - Demographic groups
  - Default risk levels

3. Financial Health Indicators

- Pay to bill ratio shows what portion of bill is being paid
- avg_bill_amt gives average spending patterns

Recommendations

1. Risk Management

1. **Early Warning System**: Monitor customers showing payment delays (especially pay_0 ≥ 1)
2. **Credit Limit Adjustments**: Review limits for customers consistently showing high utilization
3. **Targeted Interventions**: Focus on higher-risk segments (younger, lower credit limits, payment delays)

2. Customer Engagement

1. **Payment Reminders**: For customers showing first signs of delinquency
2. **Financial Education**: For younger customers and those with irregular payment patterns
3. **Customized Offers**: For reliable payers (increased limits, rewards)

3. Data-Driven Improvements

1. **Feature Engineering**: Create more predictive features from existing data
    - Payment delay trends over time
    - Utilization ratios
    - Spending volatility measures
2. **Predictive Modelling**: Build default prediction models using this rich dataset
3. **A/B Testing**: Test different intervention strategies on customer segments

4. Portfolio Management

1. **Risk-Based Pricing**: Adjust interest rates based on customer risk profiles
2. **Portfolio Diversification**: Balance high-risk and low-risk customers
3. **Reserve Planning**: Better estimate potential defaults for financial planning

13. Conclusion

Data was cleaned, features were engineered, models were tested. Results are promising but need further validation.

14. Appendices

Code in Jupyter Notebook.
Libraries: pandas, numpy, matplotlib, seaborn, sklearn, xgboost, lightgbm.