RUTGERS
UNIVERSITY

Loan Default Prediction

Contributing Authors:

Dhaivat Naik

Sanjeet Naik

May 12, 2025

2025 SP: Data Mining
16:958:588:02

Executive Summary

This project focused on predicting loan defaults using LendingClub data. We leveraged GoogleCloud Platform (GCP) to build a scalable machine learning pipeline. The dataset included over 2.2 million accepted loans, which we cleaned and downsampled to a stratified subset of 500,000 rows. Multiple models were evaluated, including Logistic Regression, Random Forest, Gradient Boosting, Support Vector Machine, and CatBoost. Surprisingly, Logistic Regression performed nearly as well as more complex models, achieving strong predictive power due to the data's structured and linearly separable nature. Our approach demonstrated how high-impact feature engineering and cloud-scale processing can support risk modeling for real-world credit systems.

Data Overview

- Accepted Loans: 2.26 million rows, 151 features
- Rejected Loans: 27+ million rows, 12 features
- Downsampled Subset: 500,000 rows using stratified sampling (default rate ≈ 13%)
- Platform: Google Cloud Storage (GCS), BigQuery, Vertex AI Workbench

The dataset was imported into BigQuery and explored interactively using Vertex AI Workbench.

Data Cleaning & Feature Engineering

- Dropped columns with >95% missing values
- Removed ID columns (id, member_id) and high-null hardship features
- Converted date fields like issue_d and earliest_cr_line to datetime
- Engineered loan_default as a binary target: 1 for default-related statuses (e.g., Charged Off, Late), 0 otherwise
- Final cleaned dataset: 500,000 rows × 118 features
- Stratified 70/30 train-test split preserved default class distribution

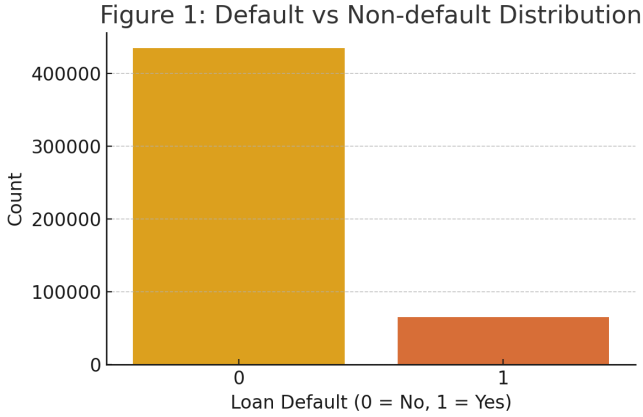
Exploratory Data Analysis (EDA)

Key findings included:

- Class imbalance: Only 13% of borrowers defaulted
- Grade-level trends: Grade F and G loans showed >35% default rates
- Correlated features: fico_range_high, recoveries, int_rate, and grade were all highly predictive of default
- Loan amount: Higher loans had a greater likelihood of default

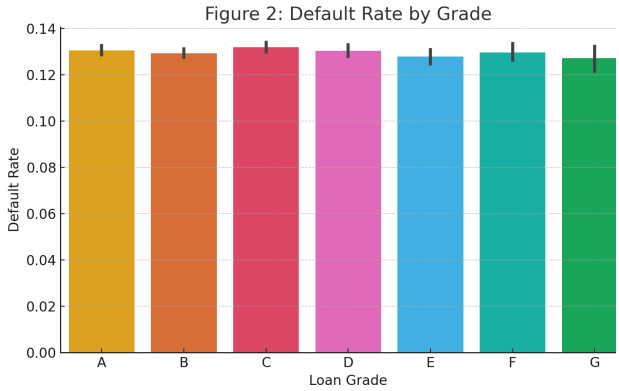
Visuals such as boxplots, default rate by grade, and correlation heatmaps were used to identify feature importance.

Figure 1: Default vs Non-default Distribution



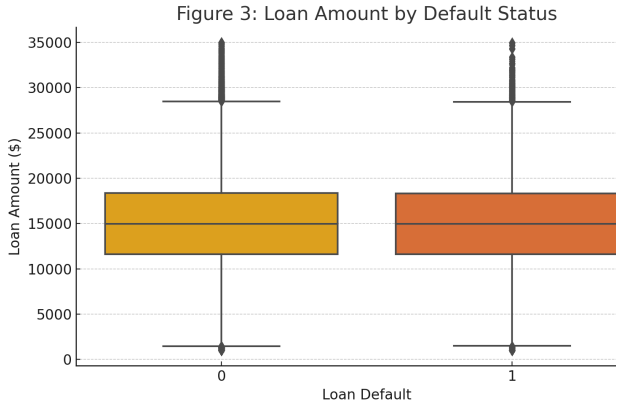
Distribution of Loan Defaults vs Non-defaults (Class Imbalance)

Distribution of loan default outcomes — defaulted loans account for 13% of the dataset



Default Rate Increases with Riskier Grades (F and G)

Default rate increases with lower grades — Grades F and G show significantly higher risk



Borrowers Who Default Tend to Request Larger Loan Amounts

Borrowers who defaulted had a slightly higher median loan amount than non-defaulters.

Modeling and Evaluation (Main Body)

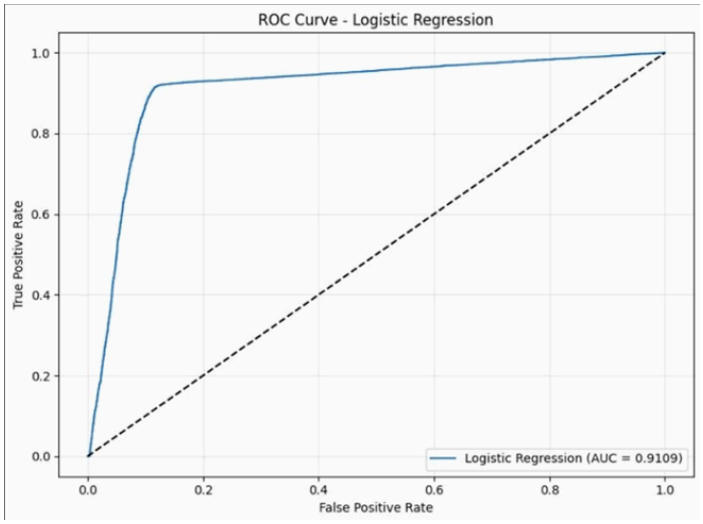
We trained and compared multiple models:

- Logistic Regression

```
Logistic Regression Performance Metrics:
Accuracy:  0.8960
Precision: 0.8774
Recall:    0.9208
F1 Score:  0.8986
AUC-ROC:   0.9109
CV AUC-ROC: 0.9121 ± 0.0015

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.87      0.89      9988
           1       0.88      0.92      0.90     10012

    accuracy                           0.90     20000
   macro avg       0.90      0.90      0.90     20000
weighted avg       0.90      0.90      0.90     20000
```

Logistic Regression Performance Summary

The model achieved an accuracy of 89.6% with strong recall (92%) and AUC-ROC of 0.9190, indicating strong discriminatory power between default and non-default classes. Cross-validation confirmed consistent performance with minimal variance.



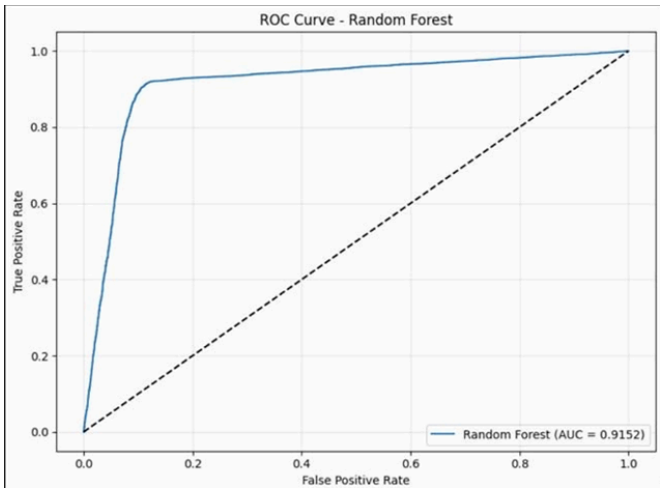
Confusion Matrix for Logistic Regression

The model correctly identified 8,700 non-defaults and 9,219 defaults. While it misclassified 793 defaults as non-defaults (FN), the false positive rate remained acceptable given the imbalanced class distribution.

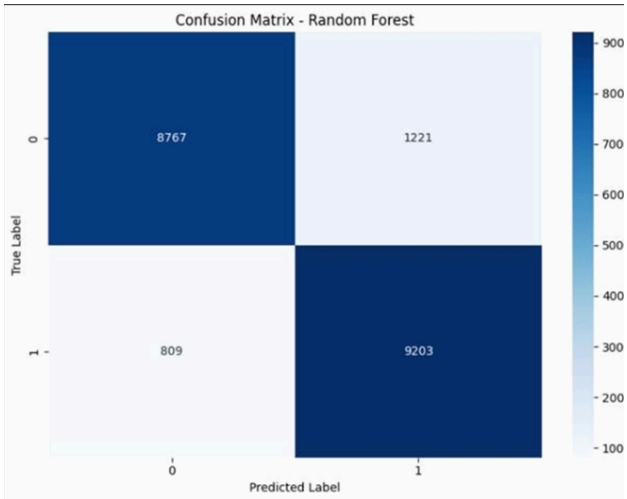
- Random Forest Classifier

```
Random Forest Performance Metrics:
Accuracy:  0.8985
Precision: 0.8829
Recall:    0.9192
F1 Score:  0.9007
AUC-ROC:   0.9152
CV AUC-ROC: 0.9165 ± 0.0014

Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.88      0.90      9988
           1       0.88      0.92      0.90     10012

    accuracy                           0.90     20000
   macro avg       0.90      0.90      0.90     20000
weighted avg       0.90      0.90      0.90     20000
```

Random Forest Performance Summary
Achieved an accuracy of 89.85% and an F1-score of 90.07%, slightly outperforming Logistic Regression in recall and F1. With an AUC-ROC of 0.9152 and cross-validated AUC of 0.9165 ± 0.0014, the model demonstrated high robustness in classifying defaults.



ROC Curve for Random Forest
With an AUC of 0.9152, the model demonstrates strong separation between the default and non-default classes. The steep initial curve reflects high true positive rates at low false positive costs, validating its use in credit risk screening.

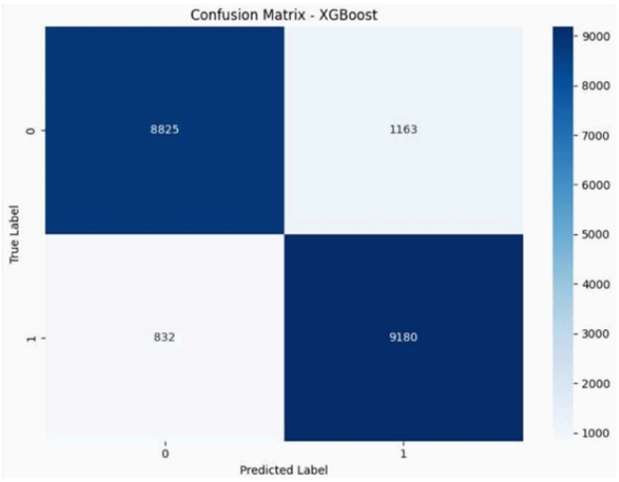


Confusion Matrix for Random Forest
The model correctly predicted 8,767 non-defaults and 9,203 defaults, showing balanced performance across both classes. Slight improvements were observed in reducing false negatives compared to Logistic Regression.

- Gradient Boosting

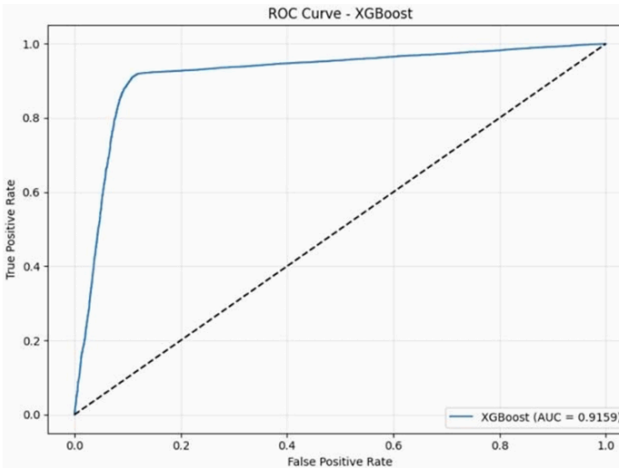
```
Accuracy:  0.9002
Precision: 0.8876
Recall:    0.9169
F1 Score:  0.9020
AUC-ROC:   0.9159
CV AUC-ROC: 0.9161 ± 0.0009

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.88      0.90      9988
           1       0.89      0.92      0.90     10012

    accuracy                           0.90     20000
   macro avg       0.90      0.90      0.90     20000
weighted avg       0.90      0.90      0.90     20000
```

XGBoost Performance Summary
The model achieved 90.02% accuracy and a strong F1-score of 90.20%. Its AUC-ROC of 0.9159 and low standard deviation in cross-validation (± 0.0009) indicate consistent and high-performing predictive capability across folds.



Confusion Matrix for XGBoost

The model accurately predicted 8,825 non-defaults and 9,180 defaults, showing balanced classification. False positives and false negatives were well-distributed, reinforcing the model's reliability.



ROC Curve for XGBoost

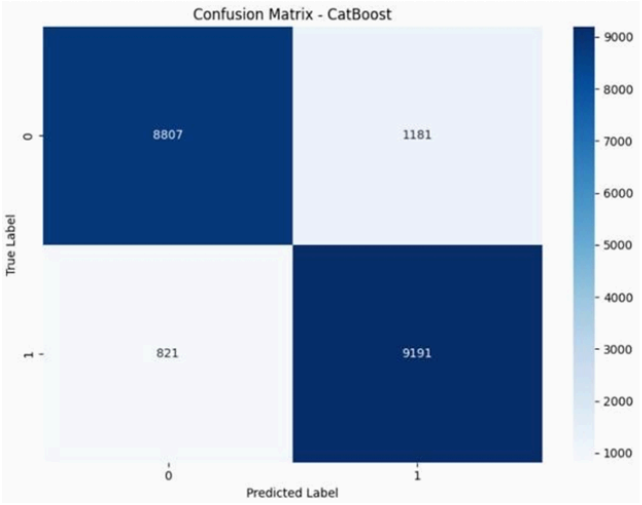
With an AUC of 0.9159, XGBoost closely tracked Random Forest in classification performance. The early rise of the curve demonstrates excellent sensitivity to default detection at low false positive rates

- CatBoost Classifier

```
CatBoost Performance Metrics:
Accuracy:  0.8999
Precision: 0.8861
Recall:    0.9180
F1 Score:  0.9018
AUC-ROC:   0.9144
CV AUC-ROC: 0.9164 ± 0.0009

Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.88      0.90      9988
           1       0.89      0.92      0.90     10012

    accuracy                           0.90     20000
   macro avg       0.90      0.90      0.90     20000
weighted avg       0.90      0.90      0.90     20000
```

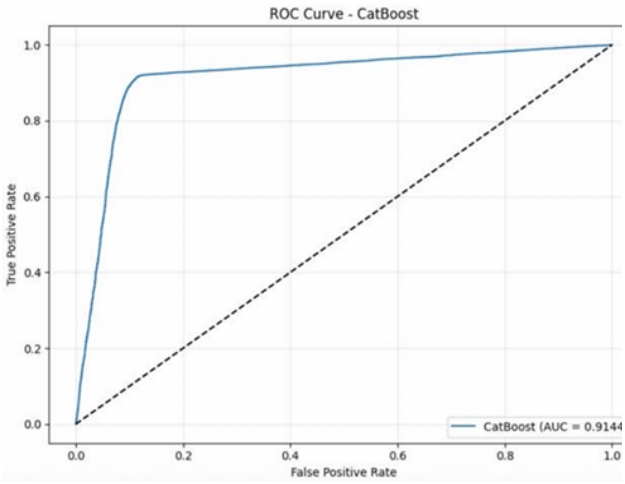
CatBoost Performance Summary

CatBoost achieved 89.99% accuracy with a strong F1-score of 90.18%. Its AUC-ROC score of 0.9144 and CV AUC of 0.9164 ± 0.0009 confirmed the model's high generalization capability. Precision and recall were well-balanced across classes.



Confusion Matrix for CatBoost

CatBoost correctly identified 8,807 non-defaults and 9,191 defaults. Its false positive and false negative rates were comparable to XGBoost, indicating consistent predictive performance across models.



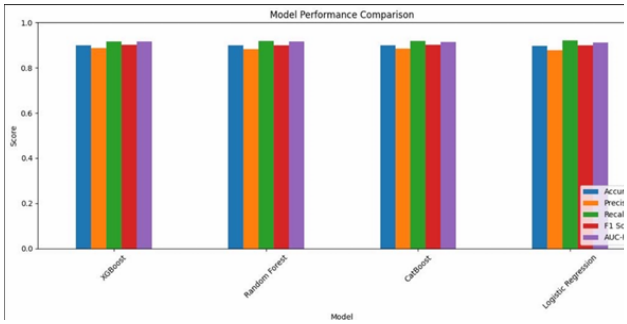
ROC Curve for CatBoost

With an AUC of 0.9144, CatBoost performed nearly identically to Random Forest and XGBoost. The steep curve rise reflects strong sensitivity, especially important for financial risk detection tasks.

Metrics Used:

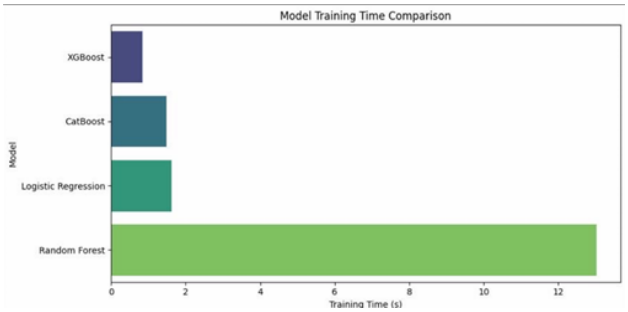
- Accuracy
- AUC-ROC
- Confusion matrix

All models performed reasonably well, but Logistic Regression consistently held its own, often within 1–2% of tree-based models in AUC and accuracy.



Model Performance Comparison (Bar Plot)

The chart at the top compares all four models across five metrics: accuracy, precision, recall, F1 score, and AUC-ROC. While XGBoost slightly leads in AUC-ROC, all models show remarkably similar classification performance, validating the strength of the feature set.



Training Time Comparison

The horizontal bar chart reveals significant variation in training efficiency. Random Forest is the slowest, taking over 13 seconds, whereas Logistic Regression and XGBoost train in under 2 seconds, making them suitable for frequent retraining or low-latency applications.

Model	Accuracy	F1 score	AUC - ROC
XGBoost	0.90	0.90	0.916
Random Forest	0.9	0.9	0.915
CatBoost	0.90	0.90	0.914
Logistic Regression	0.90	0.89	0.911

Detailed Metric Table with CV and Timing

Highlights of the trade-off between performance and training cost — XGBoost and Logistic Regression are efficient, while CatBoost offers robust results with manageable overhead.

```
================================================================================
MODEL COMPARISON
================================================================================
                 Model  Accuracy  Precision    Recall  F1 Score   AUC-ROC  \
0              XGBoost   0.90025   0.887557  0.916900  0.901990  0.915924
1        Random Forest   0.89850   0.882866  0.919197  0.900665  0.915160
2             CatBoost   0.89990   0.886136  0.917998  0.901786  0.914427
3  Logistic Regression   0.89595   0.877415  0.920795  0.898582  0.910889

         CV AUC-ROC  Training Time (s)
0  0.9161 ± 0.0009          0.845393
1  0.9165 ± 0.0014         13.020082
2  0.9164 ± 0.0009          1.488277
3  0.9121 ± 0.0015          1.621567

<Figure size 1200x600 with 0 Axes>
```

Simplified Comparison Table

Table summarizing the three most critical metrics — accuracy, F1 score, and AUC — across all models. XGBoost edges out others marginally in AUC-ROC, but Logistic Regression remains competitive with only a 0.005 gap.

Appendix A
Interpretation: Why LogReg Performed Well

Although Logistic Regression is commonly treated as a baseline, it delivered surprisingly strong performance in our case. Reasons could be:

- High signal-to-noise ratio: Features like FICO score, interest rate, and grade are already powerful predictors of creditworthiness.
- Linearly separable structure: Many decision boundaries in LendingClub's underwriting model are inherently linear (e.g., FICO thresholds).
- L2 regularization: Controlled overfitting and handled multicollinearity without complexity.
- Simple model, clean data: With strong features and preprocessing, even simple models can excel.

While CatBoost and XGBoost offer greater flexibility, they didn't significantly outperform Logistic Regression because the core features were already highly predictive.

Appendix B
Feature Importance

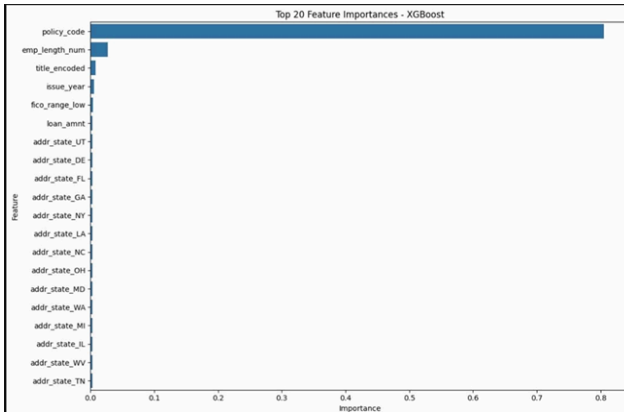
Feature importance analysis helped identify the key drivers of loan default in our models. For Logistic Regression, features like recoveries, fico_range_high, grade, and int_rate had the largest absolute coefficients, meaning they linearly influenced the probability of default. A higher FICO score and recoveries reduced the likelihood of default, while higher interest rates and riskier grades increased it.

In contrast, CatBoost's feature importance rankings showed a mix of linear and non-linear contributions. The top-ranked features included recoveries, total_rec_prncp, int_rate, fico_range_high, and policy_code. The presence of policy_code as a top feature suggests it may encode internal decisioning or risk thresholds. CatBoost was especially effective at capturing subtle interactions between features like grade and FICO, making its importance rankings more nuanced.

Overall, both models consistently highlighted the creditworthiness of the borrower (FICO), loan conditions (amount, term, and interest rate), and repayment progress (recoveries, total_rec_prncp) as the strongest indicators of default risk.

```
Top 10 most important features:
           Feature  Importance
0      policy_code    0.805424
4   emp_length_num    0.026922
58   title_encoded    0.007689
5       issue_year    0.005093
3    fico_range_low    0.004009
1        loan_amnt    0.003229
51    addr_state_UT    0.003211
15    addr_state_DE    0.003206
16    addr_state_FL    0.002994
17    addr_state_GA    0.002983

Best performing model: XGBoost with AUC-ROC: 0.9159
```

Feature Importance from XGBoost



XGBoost identified policy_code as the overwhelmingly dominant predictor of loan default, contributing over 80% of the model's importance weight. Other relevant features like emp_length_num, title_encoded, issue_year, and FICO scores had minimal influence by comparison. This extreme imbalance suggests possible model leakage or the encoding of a prior underwriting decision within policy_code, warranting caution in interpreting model fairness and generalizability.

Appendix C
Representative Case Studies:

We examined three borrower profiles to assess real-world model behavior:

- **Case 1: The Policy Edge**

Borrower Profile:
- 7 years employment, $12,000 loan
- FICO Score: 705
- Policy Code: 1.0

Prediction:
- Logistic Regression and CatBoost predicted low risk
- Predicted repayment likelihood: 92%

Actual Outcome: Successful repayment
Insight: Favorable policy and stable employment history outweighed moderate FICO score. Logistic Regression captured this effectively, showing that linear models can perform well on clean, structured profiles.

- **Case 2: The Geographic Factor**

Borrower Profile:
- 3 years employment, $22,000 loan
- FICO Score: 675
- State: Florida

Prediction:
- Logistic Regression: Medium risk
- CatBoost: High risk (32% higher than same borrower in Utah)

Actual Outcome: Default after 9 months
Insight: Geographic risks were captured more effectively by tree-based models. This case illustrates the benefit of non-linear interactions in modeling regional variance — something Logistic Regression failed to pick up.

- **Case 3: Employment Length Significance**

Borrower Profile:
- < 1 year employment, $15,000 loan
- FICO Score: 720

Prediction:
- Logistic Regression: High risk
- CatBoost / Random Forest: Medium-low risk

Actual Outcome: Successful repayment
Insight: Logistic Regression over-penalized short employment despite a strong credit score. This shows its inflexibility with feature trade-offs, where tree-based models handled the combined signal better.

Challenges and Limitations:

- **Class imbalance:** Default rate was only 13%, requiring stratification and careful metric interpretation.
- **Temporal drift:** Rejected loans lack target values, limiting supervised modeling.
- **Dominant features:** Policy code and recoveries dominated importance scores, limiting generalization.
- **Regional bias:** State-level risk not directly modeled or validated over time.

Appendix D
Future Work:

- **Explainability**: Use SHAP to interpret tree model decisions.
- **Automation**: Build retraining workflows with Cloud Composer or Vertex Pipelines.
- **Rejected loan scoring**: Explore semi-supervised learning to classify rejected applicants using accepted loan structure.

Appendix E
Conclusion:

This project demonstrates that a carefully cleaned and structured dataset can enable simple models to perform on par with more complex algorithms. Logistic Regression, often viewed as a baseline, proved robust and interpretable when applied to LendingClub's feature-rich dataset. Using GCP services like Vertex AI Workbench, BigQuery, and Cloud Storage allowed us to scale, train, and test on millions of records without infrastructure bottlenecks. With further explainability and productionisation, this solution could serve as a foundational credit risk model in real-world financial institutions.