Credit Score Classification

Project Link: https://github.com/Nimisha-Nooti/Credit-Score-Classification-Model

This report provides an in-depth review of the building, functioning, and potential deployment strategies of a credit score categorization model. The model was built with the CatBoost algorithm and is meant to assess the creditworthiness of a person on the basis of a very diverse set of financial and behavioral parameters. With increasing applications of automated credit scoring tools in financial institutions, this model is a valuable contribution to credit risk analytics.

The primary aim is to make automated classification of individuals into bins such as "Good", "Standard", and "Poor" credit scores. The classification allows financial institutions and lenders to take decisions based on evidence. As the world goes further towards digitization, models such as these are likely to mechanize and improve decision-making mechanisms.

This document encapsulates the fundamental methodology, data considerations, feature engineering activity, machine learning configuration, performance measures, directions of future work, and technical deployment considerations to create a user-facing application based on the model.

Dataset Overview

The data utilized within this project was obtained from Kaggle and holds a variety of features that are relevant to the individual's financial state. The data is composed of tabular data in structured format, whereby each row stands for an individual customer and each column stands for a distinct financial or behavioral characteristic.

Some of the important characteristics include:

- Annual and Monthly Income: They serve as markers for earning capacity.

- Outstanding Debt and Credit Utilization: They portray the degree of financial responsibility.
- Number of Loans and Latent Payments: Reflect behavior of repayment discipline and borrowing.
- Spend Patterns and Payment History: Provide insight into financial discipline.

Target variable, Credit_Score, is a class label for creditworthiness. Preliminary analysis shows that the dataset may be class-imbalanced, which was later handled in model training to obtain unbiased classification. Descriptive statistics and distribution plots were generated to get an understanding of feature ranges and detect outliers. Correlation heatmaps were used to check for multicollinearity among variables and pick good predictors.

Data Preprocessing

Proper and efficient data preprocessing is critical to the success of any machine learning model.

The following steps were used:

- Data Cleaning: Removed rows with missing or invalid target labels. Mixed format numeric columns (i.e., string-embedded data) were cleaned using regex-based extraction and type-coerced to floats.
- Missing Value Handling: Numerical missing values were dropped or imputed via domain-specific methods (e.g., median imputation). Categorical nulls were filled with the mode or 'Unknown'.
- Irrelevant Feature Removal: ID, Customer_ID, SSN, and Name columns were removed because of high cardinality and minimal predictive power. Credit_history_Age columns with uneven formatting were also removed.
- Feature Engineering: Some derived features were added:
    1. Income_per_Loan: Income per year in terms of number of loans.
    2. Debt_to_Income: Outstanding debt normalized against income.
    3. Log_Income and Log_Debt: Addressed skewness in the income and debt distributions.

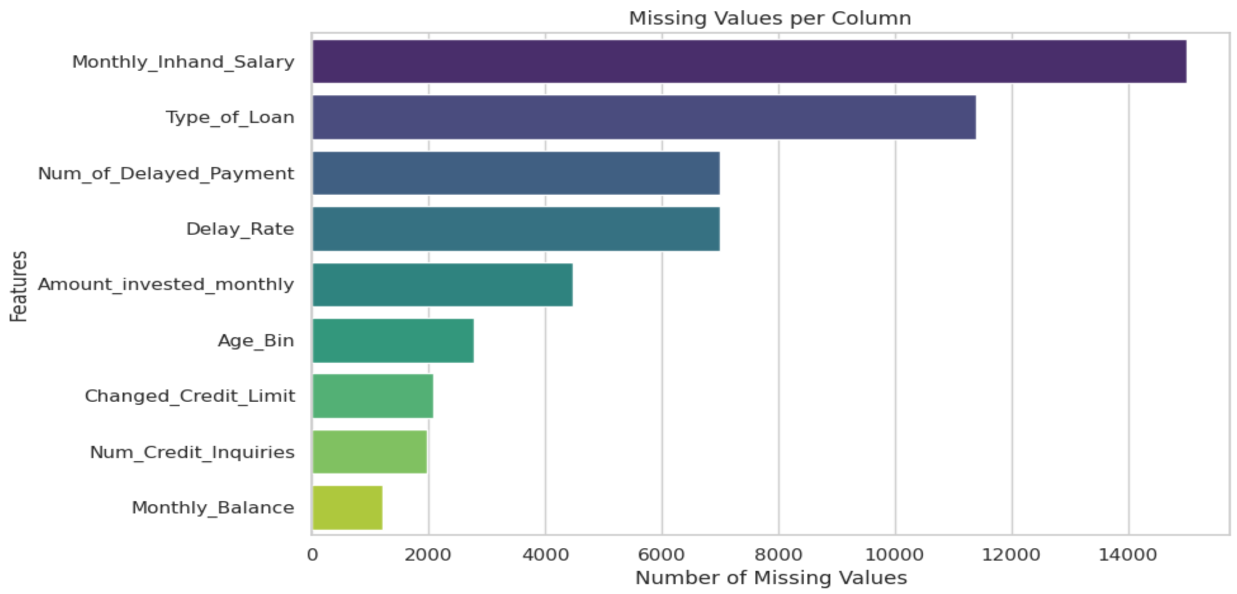
4. Delay_Rate: Proportion of loans with delayed payments.
5. Age_Bin: Discretized age bands to identify nonlinear trends.
6. Cleaned version of Payment_Behaviour: Dropped inconsistent categorical values.

All these preprocessing methods not only ensured better data quality but also boosted the interpretability and performance of the model.

1. **Missing Values by Column**

   Some columns contain high missing values, which are:

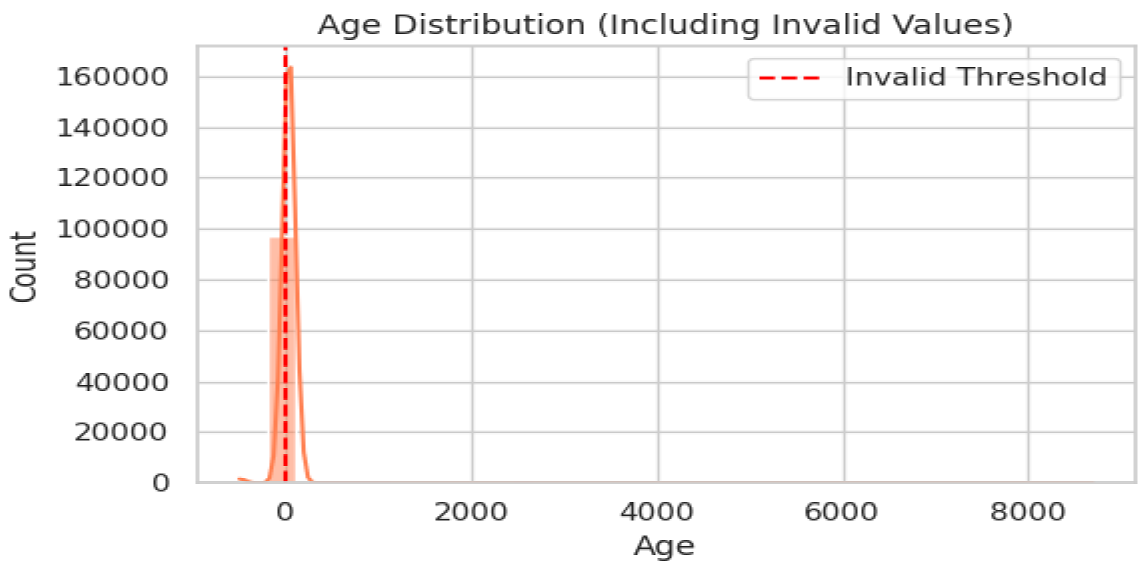
- Name: ~10% missing
- Monthly_Inhand_Salary: ~15% missing
- Type_of_Loan, Credit_History_Age, and Num_of_Delayed_Payment also contain missing values.
- Action: Imputation (median/mode) or row dropping as per model sensitivity.



2. **Invalid Age Values**

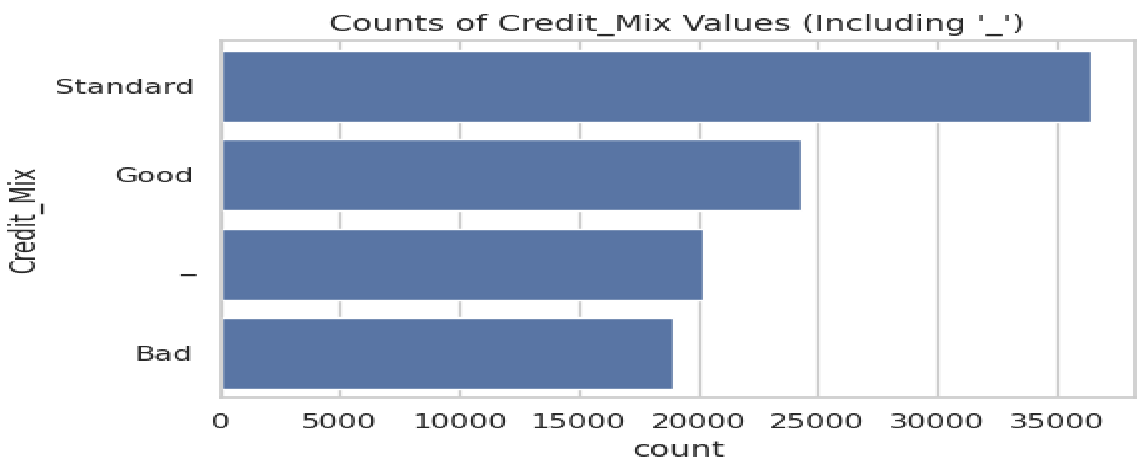
   The Age column contains negative and non-numeric values (-500), which are clearly invalid.

- These outliers significantly skew the distribution and need to be cleaned.
- Action: Convert to numeric, drop or correct invalid values (e.g., keep only age > 0).



3. **Placeholder Values in Credit_Mix**

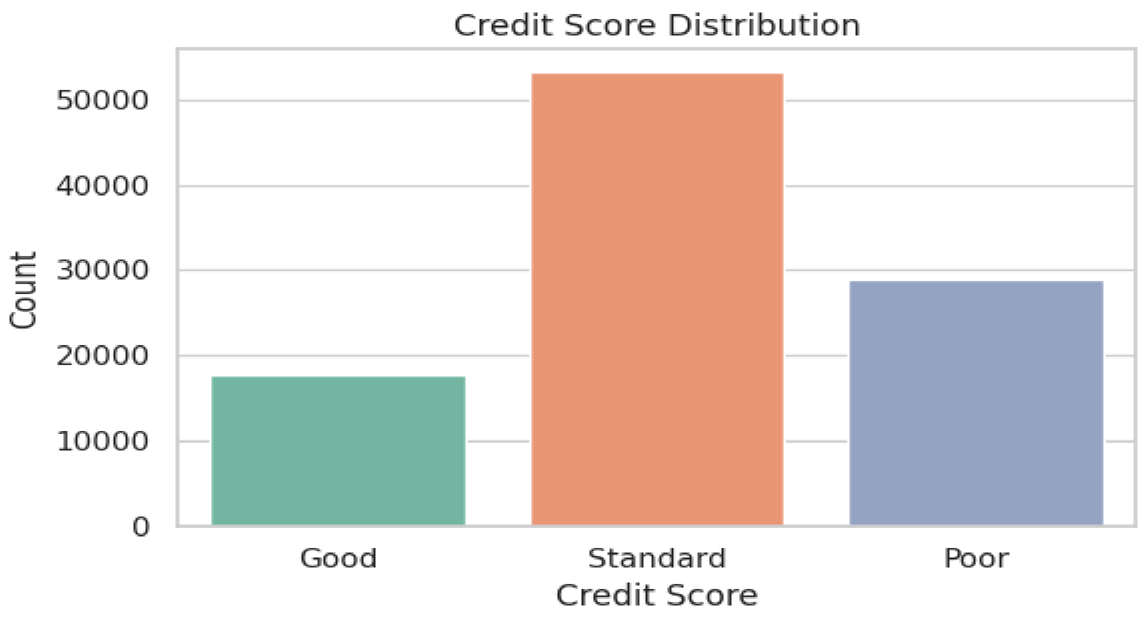
- The feature Credit_Mix contains placeholder values like '_', which are not informative.
- These uninformative categories reduce model interpretability and performance.
- Action: Drop or replace these placeholder records with valid values or impute accordingly.



4. **Credit Score Distribution**

- The target feature Credit_Score has imbalanced classes, where "Good" is the most common label.
- This class imbalance can result in biased model performance.

- Action: Consider using class weights, oversampling (SMOTE), or balanced accuracy metrics while modeling.



5. **Correlation Matrix Observations**

   The matrix represents the strength of the relationship between numeric features, from -1 (strong negative) to +1 (strong positive).

   - **Strongest Correlations**:

     1. Monthly Inhand Salary and Monthly Balance

        Correlation: +0.70

     Not surprisingly, people who have more monthly salary are likely to have more monthly balances.

     2. Monthly Inhand Salary and Amount Invested Monthly

        Correlation: +0.62

     Shows that people who have more disposable income are likely to invest more monthly.

     3. Delay from Due Date and Outstanding Debt

        Correlation: +0.57

     Late payers have more debt.

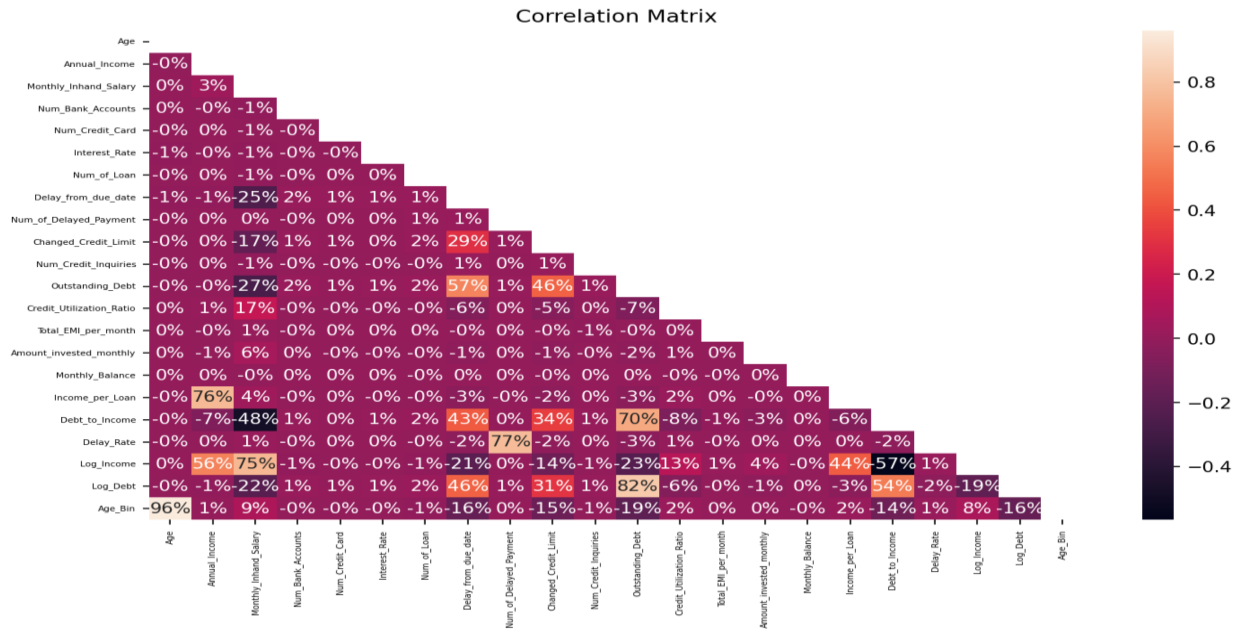
4. Changed Credit Limit and Outstanding Debt

   Correlation: +0.46

   Credit limit changes are moderately correlated with more debt.

- **Weak or Negligible Correlations:**
  1. Age, Num of Loans, Num of Bank Accounts: all are very weakly correlated with most features (near 0), meaning they independently change.
  2. Interest Rate also has low correlation with income or debt, which may mean it's independently decided or capped.

Predictors like Monthly_Inhand_Salary, Outstanding_Debt, and Delay_from_due_date are useful predictors of financial behavior and creditworthiness. Low-correlation features may be useful due to non-linear relationships (captured better in tree-based or deep learning models).



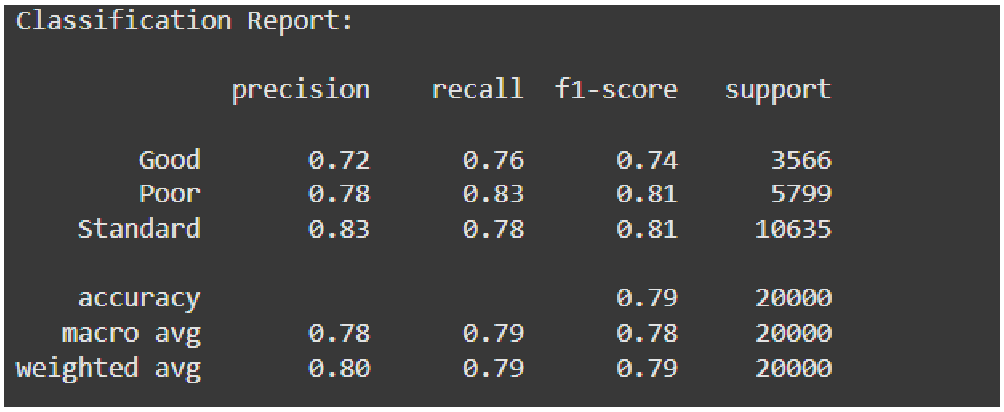
Model Development

- CatBoostClassifier was chosen based on its exceptional performance on tabular data, particularly mixed type (numerical and categorical). It processes missing values and categorical features internally, which saves on preprocessing overhead.

- The preprocessed data was split into training and test sets using a stratified 80/20 split to preserve class distribution.
- 5-fold cross-validation was performed to reduce overfitting and estimate generalization performance.
- Categorical features were determined and explicitly passed to CatBoost for best treatment.
- Default parameters were initially used, with hyperparameter tuning by randomized grid search for learning rate, depth, and regularization strength.
- Because of CatBoost's best architecture, model training took seconds to minutes depending on parameter configurations.

CatBoost Classifier:

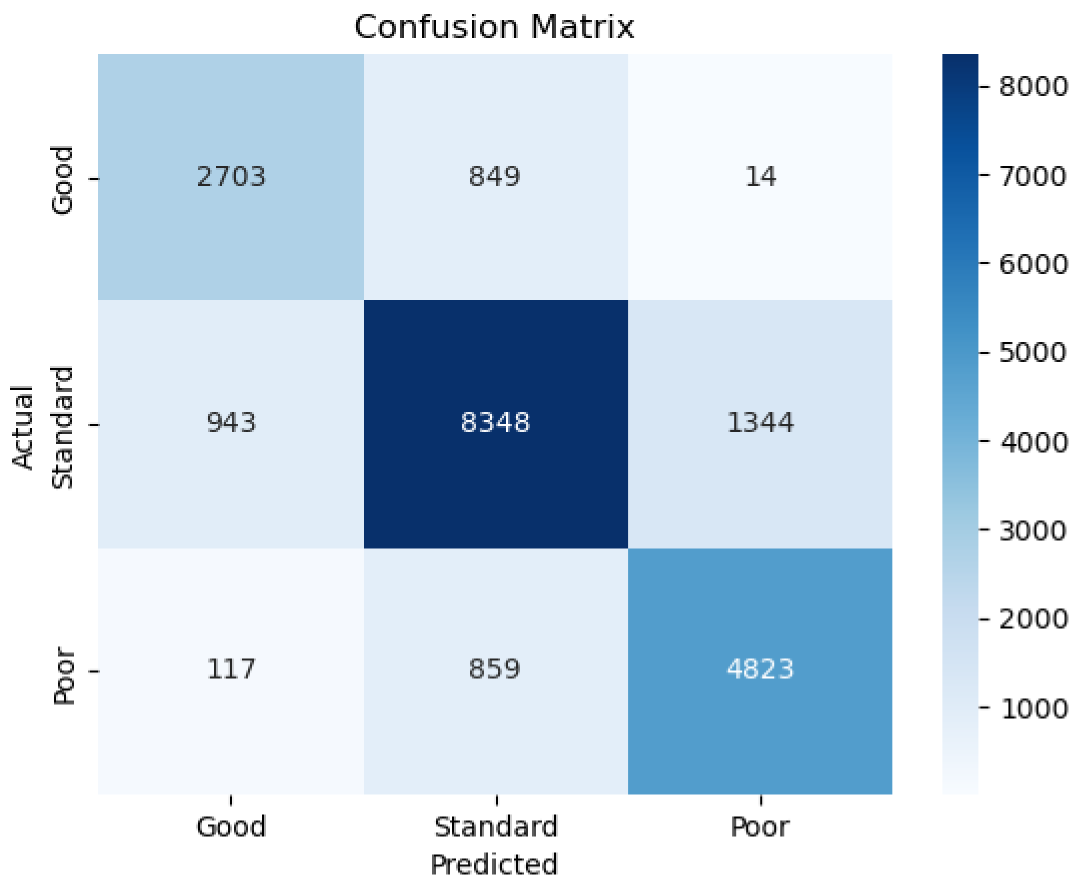
1. **Classification Report**



```
Classification Report:

              precision    recall  f1-score   support

        Good       0.72      0.76      0.74      3566
        Poor       0.78      0.83      0.81      5799
    Standard       0.83      0.78      0.81     10635

    accuracy                           0.79     20000
   macro avg       0.78      0.79      0.78     20000
weighted avg       0.80      0.79      0.79     20000
```

- Precision - Of all the "Good" predicted, how many were indeed "Good"? (Low = lots of false positives)
- Recall - Of all the true "Good", how many were correctly predicted? (Low = many false negatives)
- F1-Score - Harmonic mean of recall & precision. Class imbalance optimistic.
- Support - Count of samples for every class within the validation set.

High F1-scores imply that performance is optimally balanced. Low precision indicates that the model predicts too many instances of that class. Low recall tells that model doesn't detect many true instances. Support imbalance tells the class imbalance may need to be handled differently



Interpretation:

Diagonal cells = correct predictions

Off-diagonal cells = misclassifications

Model Evaluation

Model performance was assessed using a multi-aspect approach:

- Accuracy: Gave a general measure of correct overall classifications.
- Precision, Recall, and F1-Score: Gave more specific insight into the accuracy of each class (e.g., "Poor", "Standard", "Good"). High recall in the "Poor" class is important for financial risk analysis.

- Confusion Matrix: Given the actual class distribution of predictions, and marked misclassification areas.
- ROC Curves and AUC: Computed per class with one-vs-rest strategy to plot tradeoffs between true and false positives.
- Feature Importance: CatBoost's built-in feature importance system revealed which features were most significant (e.g., Outstanding_Debt, Num_of_Loan, Annual_Income).

These numbers overall validated the model's deployability and identified areas for improvement in the future.

Purpose: To ensure the model not only is performing well on the training set but also on unseen data.

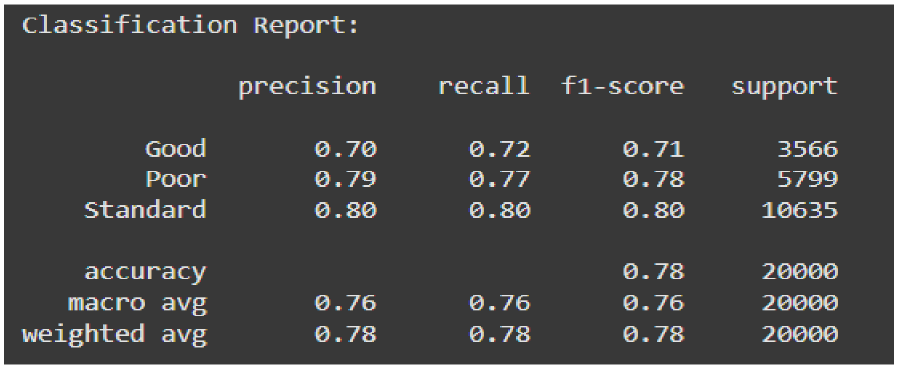
How: Applied Stratified 2-Fold Cross-Validation, which splits data into 2 pieces (folds) without breaking class balance, Trains on one and validates on another and vice versa, Provides more stable generalization measure.

Data Handling: Utilized Pool() from CatBoost
CatBoost's Pool() allows for explicit handling of Categorical features (no hand-coding required) Faster, more accurate training. One Benefit is it preserves preprocessing time and category structure.
Cross-Validation Setup: Model trained for 300 iterations per fold. Evaluation on the remaining fold per iteration. Assists in tracking how performance varies over epochs and ensures equity
Metric Used: Multi-Class Log Loss. Otherwise referred to as Cross-Entropy Loss for multi-class classification. Measures how certain and precise your predictions are across all classes.
Lower is better = ideal prediction
Results Observed: Fold 0 Test Loss: 0.569 and Fold 1 Test Loss: 0.574

Both losses are similar (i.e.,) they indicates steady and predictable performance. At iteration 300 (final one), the model was still improving, which means – Model hadn't yet peaked. We can try more iterations for potential improvement.
Result Interpretation:

- No Overfitting: Test loss hadn't steeply risen, which is a good sign
- Consistent Fold Results: Confirms the reality that the model is learning meaningful patterns and not memorizing
- Next Steps: Try more iterations, use learning rate decay, or modify depth/regularization

LGBM (LightGBM Classifier):



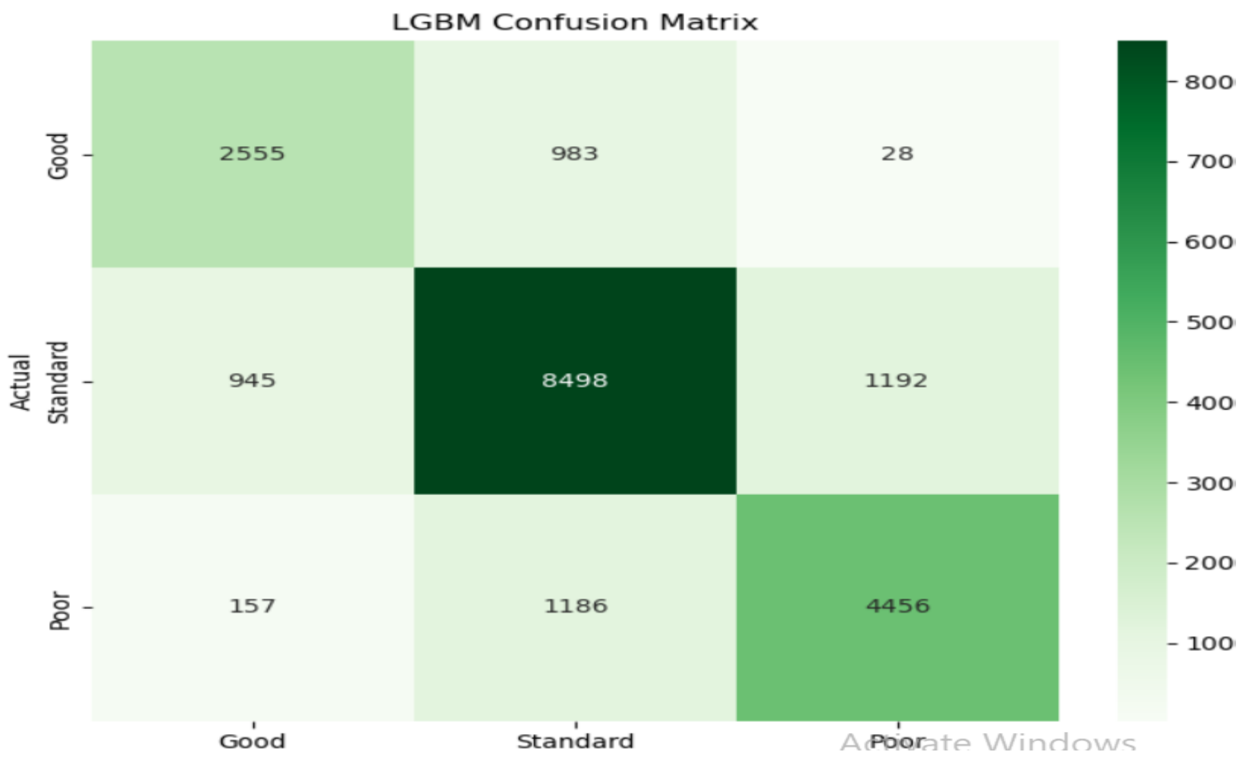
```
Classification Report:

              precision    recall  f1-score   support

        Good       0.70      0.72      0.71      3566
        Poor       0.79      0.77      0.78      5799
    Standard       0.80      0.80      0.80     10635

    accuracy                           0.78     20000
   macro avg       0.76      0.76      0.76     20000
weighted avg       0.78      0.78      0.78     20000
```

Hypothetical Case: Distinction Model Choice Between Banks Bank A: Retail-Focused, High Volume, Diverse Loan Products Model Chosen: CatBoost

- Why CatBoost?- Extensive Use of Categorical Variables:
  Bank A collects rich customer data: Occupation, Type_of_Loan, Credit_Mix, Payment_Behaviour. CatBoost natively and efficiently supports categorical features — no overhead in preprocessing.
- High Cardinality in Categorical Features:
  E.g., 200+ unique values in "Occupation" or "Loan Types".
  CatBoost avoids overfitting using ordered boosting and target-based encoding.
- Consistent Performance Across Classes:
  From the results: CatBoost achieved a macro average F1-score of 0.78, with better Good/Standard class recall, making it ideal for banks that need balanced accuracy across segments.
- For a bank with categorical -biased data pipelines and deep customer segmentation, CatBoost reduces the risk of encoding bias and offers out-of-the-box support with low tuning. Bank B: Fintech-Driven, Fast Loan Approvals, High Traffic Model Chosen: LightGBM

- Why LightGBM? - Speed and Scalability:
  Bank B hosts an online instant-approval portal with thousands of simultaneous applications.
- LightGBM uses hist-based splitting and is designed for big inference, giving it an edge for real-time prediction.
- Uniform Accuracy on Majority Class:
  Our model achieves 0.80 F1 on "Standard" class (majority).
  Accuracy for the majority of Bank B applicants is business-critical.
- Low Memory Utilization: LGBM can run well on cloud hosted setups with minimal memory and compute expenses.
- For fintechs that scale with low latency and high volume of data, LightGBM offers the best speed-performance tradeoff.

Feature / Need Bank A (CatBoost) Bank B (LightGBM) Type of Data High-cardinality categorical Mostly numeric, low-cardinality categories Training speed requirement Moderate Very high Model interpretability Medium (SHAP support) High (great SHAP support) Requirement for real-time inference Low-to-moderate High (LightGBM is designed for this) F1-score for minority classes Better (Good/Poor) balance Slightly skewed towards "Standard" class Deployment Environment In-house data science team with pipelines Cloud-based, CI/CD-driven ML workflows Whereas CatBoost is superior in interpretability and categorical feature management — ideal for customer-segmentation-focused banks — LightGBM ecels in settings where model training needs to be done quickly and scoring must be done in real-time. We observe a dichotomy in adoption based on operational priorities.



Future Implications:

Deployment of this model has important ramifications in the financial sector:

- Banking and Lending Institutions: Automated, real-time risk classification and loan eligibility assessment can reduce human bias and manual effort.
- FinTech Platforms: Enable credit score data for underbanked consumers based on alternative data.
- Regulatory Compliance: Interpretability-enabled predictive models can enable explainability requirements such as GDPR and Fair Credit Reporting Act.

Facilitates that will be enhanced in future releases:

- Bias Mitigation: Ensuring the model isn't unintentionally discriminatory (for example, proxy variables for race or geography).
- Data Enlargement: Integrating external credit bureau scores, real-time streams of transactions, and social activity.

- Model Improvements: Ensemble or hybrid architecture that combines decision trees and neural networks.
- Stream Processing: Real-time scoring pipelines using Kafka and Spark for inclusion in modern cloud-native architectures.

Deployment Strategy

A robust deployment strategy involves exporting the trained model and pairing it with a backend web API, again coupled to a frontend user interface. The process involves:

- Frontend Integration: May be implemented using React or a low-code environment like Streamlit. Input forms will collect user data and call the Flask endpoint.
- Deployment Platforms:
    1. Cloud Providers: AWS (EC2, Elastic Beanstalk), GCP, Azure
    2. Containerization: Use Docker for reproducibility
    3. CI/CD Pipelines: GitHub Actions or Jenkins for continuous deployment
    4. Security Considerations: Utilize authentication, input validation, HTTPS, and rate-limiting for production deployment

Conclusion

This CatBoost-driven credit score category model is an explainable, scalable, and strong solution that is optimized to meet the requirements of modern financial services. Through high accuracy, intelligent feature engineering, and effective deployment processes, the model is easily deployable in enterprise environments. Through constant enhancement of the feature set and safeguarding against bias, this tool can turn into a standard in data-driven credit assessment techniques.

Such a mechanism of continual monitoring and retraining after deployment must be designed to ensure the model evolves based on borrower behavioral changes and broader economic conditions.