Credit Risk Modeling using Custom Gradient Boosting

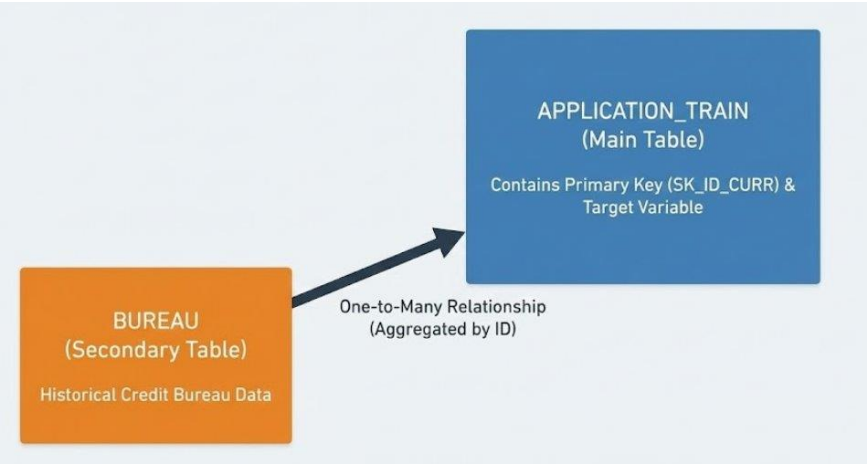
Erdem Mesut

1.Introduction
Banks lose billions when borrowers fail to repay. Predicting who will fail is difficult because:
Rarity: People who fail to repay are rare (only ~8% of cases).
Complexity: Risk depends on past loans, current income and demographics.

Project: A model to predict the probability of not repaying.

2.Dataset
The project utilizes the Home Credit Default Risk dataset. The primary objective is to predict whether an applicant will have difficulty repaying a loan. The target variable is binary: 0 (Repaid) and 1 (Default). The dataset is characterized by a significant class imbalance, with defaults constituting only ~8% of the samples.
Unlike standard flat-file datasets, this data is relational and distributed across two primary tables:
application_train.csv: The main table containing static information (income, age, region) for each current loan application of about 307.000 clients. It is indexed by the primary key SK_ID_CURR.

bureau.csv: A secondary table containing the applicant's credit history with other financial institutions. This table has a One-to-Many relationship with the main table (one applicant may have multiple past loan records).



3. Feature Engineering

To convert this relational structure into a ready format, the following preprocessing pipeline was implemented:
Relational Aggregation: Since the model requires a single row per applicant, the historical data in bureau.csv was aggregated by grouping SK_ID_CURR. Statistical summaries including Mean, Sum, and Max were calculated for numerical columns to compress the historical credit behavior into a fixed length feature vector.

Missing Value Imputation: Financial data often have outliers because of rich people. Therefore, median imputation was selected over mean imputation to fill missing values, ensuring robust handling of skewed distributions.

Categorical Encoding: Categorical variables were transformed using label encoding. This method maps each unique category to an integer, preserving the ordinal nature of variables like education levels for the decision tree splits.

4.Algorithm
This project implemented a gradient boosting classifier purely in python. The core part utilizes an additive model where each weak learner is a Regression Tree trained to predict the pseudo residuals of the previous iteration, rather than the class labels directly.

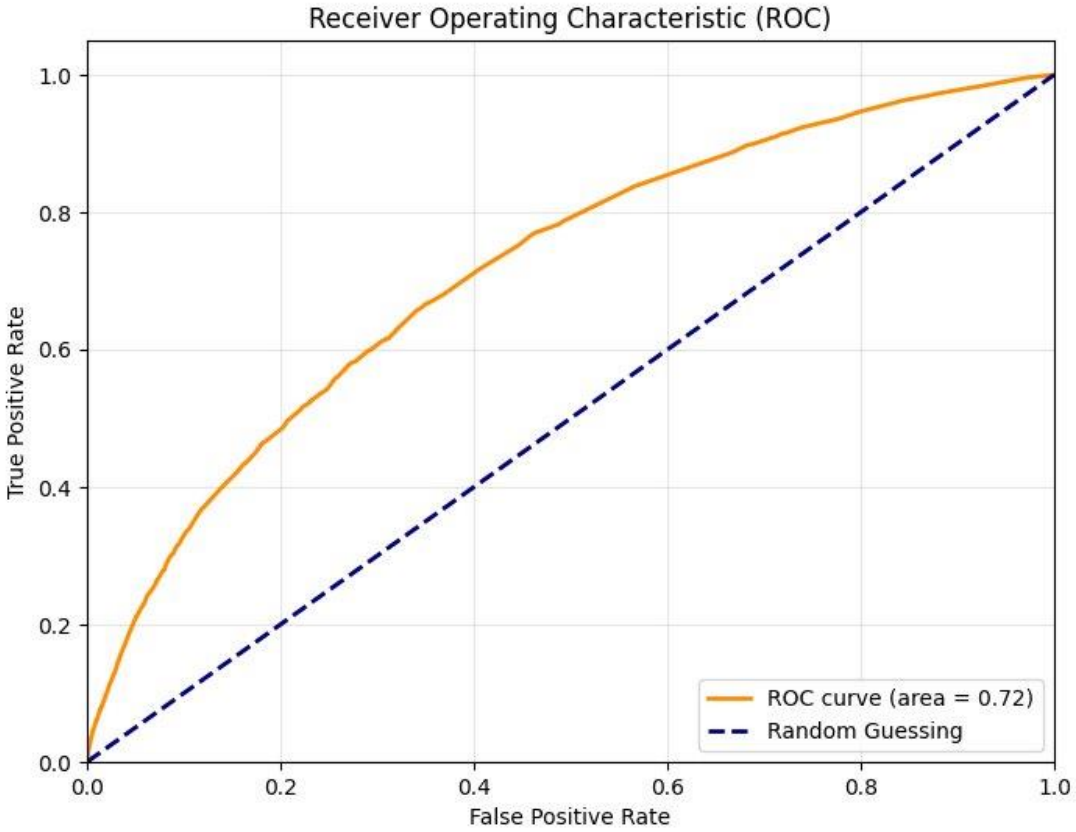
A challenge was the computational cost of finding the optimal split on a dataset with over 100 features and thousands of rows. To address this, the get_best_split function was implemented with two distinct strategies based on data type:
Exact Greedy Split: For features with fewer than 20 unique values (e.g. CODE_GENDER, NAME_EDUCATION_TYPE), the algorithm iterates through every unique value to find the precise optimal cut.

Approximate Split: For continuous variables like AMT_INCOME_TOTAL, testing every unique value would be computationally prohibitive (O(N^2)). Instead, I implemented a percentile based optimization. The algorithm tests potential thresholds at fixed percentiles (10th, 20th, ..., 90th). This reduces the search space from thousands of candidates to just 9 per feature, lowering the training time.

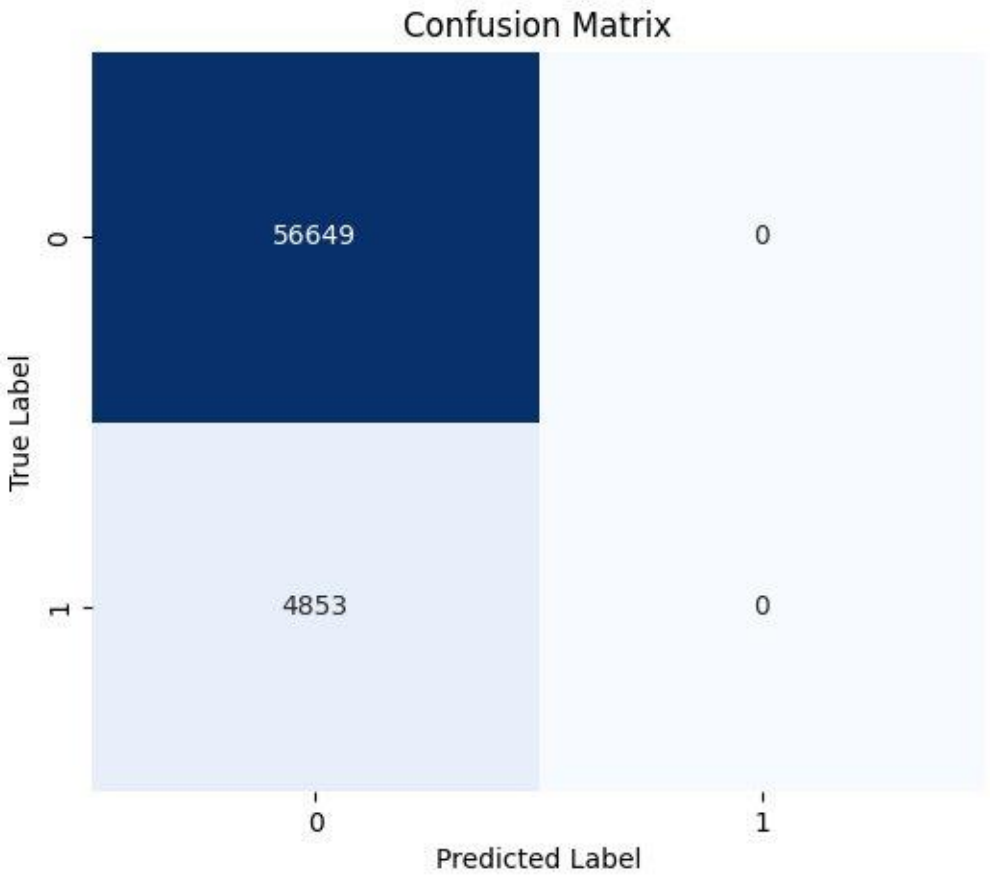
To prevent overfitting the model implements Stochastic Gradient Boosting. Instead of training on the full dataset for every iteration, each tree is trained on a random 70% subsample of the data.
The final prediction is an ensemble sum of the baseline (initial mean) and the weighted contributions of all trees (scaled by a learning rate of 0.1). Since the output is a continuous log-odds value, it is passed through a Sigmoid Function to map the final result to a probability between 0 and 1.

5.Evaluation
To verify the correctness of the implementation, the model was compared against sklearn.ensemble.GradientBoostingClassifier. Both models were trained on identical 80/20 train-test splits with strictly controlled hyperparameters:
Estimators: 10
Max Depth: 3
Learning Rate: 0.1

The primary metric for evaluation was the Area Under the ROC Curve (AUC):
Scikit-Learn: 0.7166
Custom Implementation: 0.7155

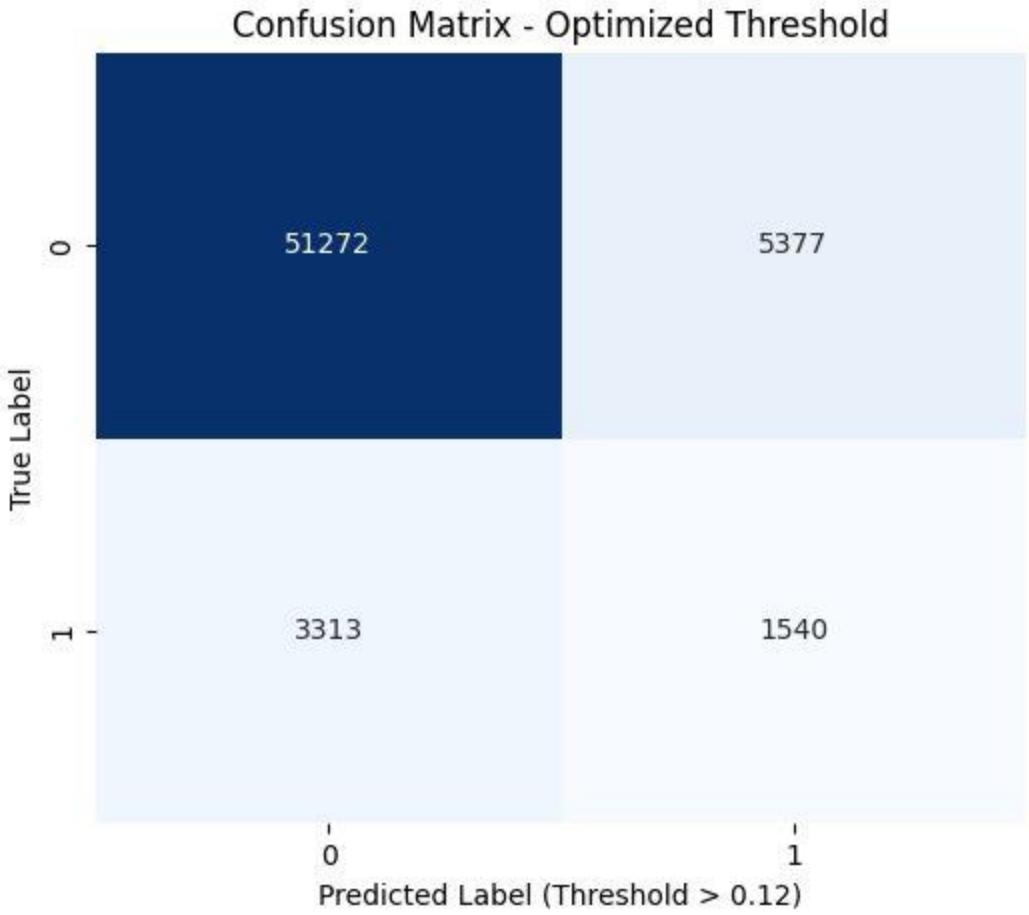


The fact that the custom implementation achieved a score nearly identical to the library validates the correctness of the implementation.
Baseline (Threshold 0.5): The model achieved 92.1% Accuracy but failed to identify a single defaulter (Recall = 0%). This confirmed that the standard decision boundary was operationally useless, biasing entirely toward the majority class.



An iterative search was performed to maximize the F1 score. The optimal decision boundary was found at 0.12.

By shifting the threshold to 0.12, the model sacrificed raw accuracy to achieve a recall of ~32%. This adjustment successfully identified 1,540 high-risk applicants that the baseline model completely ignored, representing a significant reduction in potential financial risk.



The analysis confirmed that External Sources (EXT_SOURCE_1, 2, 3)—normalized scores from external credit bureaus—were the dominant predictors, appearing as the top split nodes. This suggests that an applicant's historical reputation is a far stronger predictor of default than their current self-reported demographic data.

```
1. EXT_SOURCE_3: Used to split 28 times
2. EXT_SOURCE_2: Used to split 24 times
3. EXT_SOURCE_1: Used to split 8 times
4. CODE_GENDER: Used to split 4 times
5. DAYS_BIRTH: Used to split 3 times
6. AMT_CREDIT: Used to split 1 times
7. DAYS_CREDIT_mean: Used to split 1 times
8. NAME_EDUCATION_TYPE: Used to split 1 times
```

5.Conclusion

This project successfully demonstrated the end-to-end engineering of a Credit Risk Prediction System, moving from raw relational data to a fully functional custom machine learning engine.

The Gradient Boosting implementation achieved an AUC of 0.7155, effectively matching the performance of the scikit-learn library (0.7166). This result validates the correctness of the manually implemented residual fitting and variance reduction logic.

The analysis exposed the dangers of relying on standard accuracy metrics for imbalanced financial data. By computationally optimizing the decision threshold to 0.12, the model transitioned from catching 0 defaulters to successfully identifying 1,540 high-risk applicants.

Feature importance analysis revealed that External Sources (normalized credit scores from other institutions) were the most critical predictors, significantly outperforming demographic factors like income or age. This suggests that credit risk is fundamentally a reputation based metric rather than purely a capacity based one.