Loan Default Prediction System

A Machine Learning Web Application Report

Project Overview

The loan default prediction system represents a practical implementation of machine learning in financial risk assessment. This web-based application leverages a Decision Tree Classifier to evaluate loan default probability based on customer financial profiles. The system was developed to bridge the gap between complex ML models and user-friendly interfaces, making predictive analytics accessible to financial professionals without extensive technical backgrounds.

Technical Architecture & Implementation

The application follows a three-tier architecture comprising a Flask backend, HTML/CSS frontend, and scikit-learn ML pipeline. The backend handles data preprocessing, model inference, and file operations, while the frontend provides an intuitive interface for both single predictions and batch processing via CSV uploads.

The core prediction engine utilizes four key financial indicators: total income (AMT_INCOME_TOTAL), credit amount (AMT_CREDIT), annuity amount (AMT_ANNUITY), and family member count (CNT_FAM_MEMBERS). These features were selected based on their predictive power and availability in typical loan applications. The Decision Tree algorithm was chosen for its interpretability—crucial in financial applications where decision transparency is often legally required.

Data preprocessing includes median imputation for missing values, with precomputed medians stored in JSON format for consistency across predictions. An optimal classification threshold, determined during model training, is applied to convert probability scores into binary default/no-default decisions.

Development Challenges & Solutions

Several technical hurdles emerged during development. File handling proved particularly challenging, requiring robust error checking for CSV uploads and ensuring proper data validation. The system needed to gracefully handle malformed files, missing columns, and data type inconsistencies. We implemented comprehensive input validation that provides clear error messages rather than cryptic system failures.

Model deployment presented another challenge. Serializing the trained model, feature columns, and preprocessing parameters required careful coordination to ensure consistency between training and inference environments. The solution involved storing these components as separate files (PKL, JSON, NPY formats) and loading them synchronously during application startup.

Memory management became critical when processing large CSV files. The initial implementation loaded entire datasets into memory, causing performance issues with files containing thousands of records. We addressed this by implementing streaming processing for batch predictions, processing data in chunks while maintaining prediction accuracy.

Real-World Application & Results

Testing revealed interesting patterns in model behavior. High-income customers with reasonable credit amounts typically receive low default probabilities, while customers with high credit-to-income ratios trigger default warnings. The model correctly identifies risk factors such as large family sizes combined with high financial obligations.

The batch processing feature proved valuable for financial institutions evaluating multiple applications simultaneously. Processing times averaged 2-3 seconds for 100 records, making it practical for daily operations. The downloadable results format allows seamless integration with existing financial workflows.

Technical Insights & Lessons Learned

The project highlighted the importance of feature engineering in financial modeling. Initial attempts using raw financial amounts showed poor performance until we incorporated derived ratios and normalized values. The Decision Tree's ability to handle non-linear relationships proved advantageous for capturing complex financial behaviors.

User interface design required careful consideration of financial domain expertise. Initial versions overwhelmed users with technical details, while simplified versions lacked necessary context. The final interface strikes a balance, providing essential information without technical jargon while maintaining prediction transparency.

Limitations & Future Enhancements

The current system relies on a limited feature set, potentially missing important risk indicators like employment history, credit scores, or regional economic factors. The Decision Tree, while interpretable, may not capture complex feature interactions as effectively as ensemble methods.

Future iterations could incorporate Random Forest or Gradient Boosting algorithms for improved accuracy. Feature importance visualization would help users understand prediction drivers. Additionally, implementing confidence intervals would provide risk ranges rather than binary classifications, offering more nuanced decision support.

Conclusion

This loan default prediction system demonstrates the practical deployment of machine learning in financial services. By focusing on usability and interpretability, the application successfully translates complex predictive models into actionable business tools. The project serves as a

foundation for more sophisticated financial risk assessment systems while maintaining the simplicity necessary for widespread adoption.

The experience reinforced the importance of balancing technical sophistication with practical usability. Successful ML applications require not just accurate models, but thoughtful engineering that considers real-world deployment constraints and user needs. This project achieves that balance, providing a robust platform for loan default assessment that can be easily extended and maintained.

This report summarizes the development and implementation of a web-based loan default prediction system, highlighting technical approaches, challenges encountered, and practical applications in financial risk assessment.