Loan Approval Optimization: A Comparative Analysis of Deep Learning and Offline Reinforcement Learning

1. Problem Statement and Business Context

The objective of this project was to design an intelligent loan approval system for a fintech institution that balances profitability and credit risk. The institution aimed to maximize financial returns while minimizing default rates by leveraging historical loan data to learn optimal approval policies. The dataset contained detailed borrower profiles and historical loan outcomes ("Fully Paid" vs. "Defaulted"). The workflow was structured into four major tasks: Task 1 (EDA & Preprocessing), Task 2 (Deep Learning), Task 3 (Offline Reinforcement Learning), and Task 4 (Comparative Evaluation).

2. Task 1: Data Preprocessing

Source: A pre-cleaned dataset (loan_clean_subset.csv) was used. Target Variable: Binary 'default' column (0 = Fully Paid, 1 = Defaulted). Data Split: 80% training (141,612 samples) and 20% testing (35,404 samples). Pipeline: ColumnTransformer with SimpleImputer + StandardScaler (numerical) and SimpleImputer + OneHotEncoder (categorical). Final Features: 88 engineered attributes.

3. Task 2: Predictive Deep Learning Model

A Multi-Layer Perceptron (MLP) was built in PyTorch with 3 hidden layers (256, 128, 64 neurons, ReLU activation, dropout 0.25). Loss: BCEWithLogitsLoss, Optimizer: Adam (lr=1e-3), Early Stopping on validation AUC. Results: Test ROC-AUC = 0.7453, F1-Score = 0.2960. Interpretation: The model ranks risk well but isn't a profitable decision policy.

4. Task 3: Offline Reinforcement Learning (RL) Agent

An RL agent (Discrete Conservative Q-Learning via d3rlpy) was trained to maximize loan profitability. State: 88 borrower features; Actions: Approve or Deny; Reward: +interest if paid, -loan if defaulted. Results: Estimated Policy Value (EPV) = -■1,248.34 per applicant, Approval Rate = 92.79%. Interpretation: The RL agent failed to converge and produced an unprofitable aggressive policy.

5. Task 4: Comparative Analysis

DL and RL policies disagreed on 697 cases. In some, RL captured high-reward loans missed by DL, but also approved many defaults. Conclusion: RL attempted risk-taking but was catastrophically wrong in most cases, confirming model instability.

6. Insights and Recommendations

Key Findings: 1. Deep Learning: Reliable for risk ranking (AUC 0.745). 2. RL: Failed to learn safe/profitable policy. Recommendation: Reject standalone RL policy; use hybrid Economic Thresholding. Economic Thresholding Formula: EV = [(1 - P(Default)) * Profit] + [P(Default) *

(-Loss)] Approve only if EV > 0.

7. Future Enhancements

1. Refine reward functions (include recovery rate, NPV, fees). 2. Use tree-based models (XGBoost, LightGBM) for better tabular performance. 3. Revisit RL via IQL/BCQ once stable hybrid baseline is achieved.

8. Conclusion

This project demonstrated a full-cycle loan decision framework combining deep learning and offline reinforcement learning. Deep Learning achieved AUC = 0.745; RL failed (EPV = -■1,248). Final recommendation: Adopt hybrid DL + Economic Thresholding approach for profitability and stability.