Loan Approval Optimization: Deep Learning vs Offline Reinforcement Learning

1. Introduction

Loan approval decisions involve balancing risk (default) with potential financial gain (interest). Traditional supervised models predict default probabilities, while reinforcement learning (RL) can learn a policy that directly optimizes expected return. This project explores both approaches using LendingClub data.

2. Data and Preprocessing

Dataset: LendingClub loan records. Key variables include loan_amnt, int_rate, applicant features, and loan status (Fully Paid or Defaulted/Charged Off). Exploratory analysis showed high default rates among low credit scores and high debt-to-income ratios. Data preprocessing involved encoding categorical variables, imputing missing values, and scaling features.

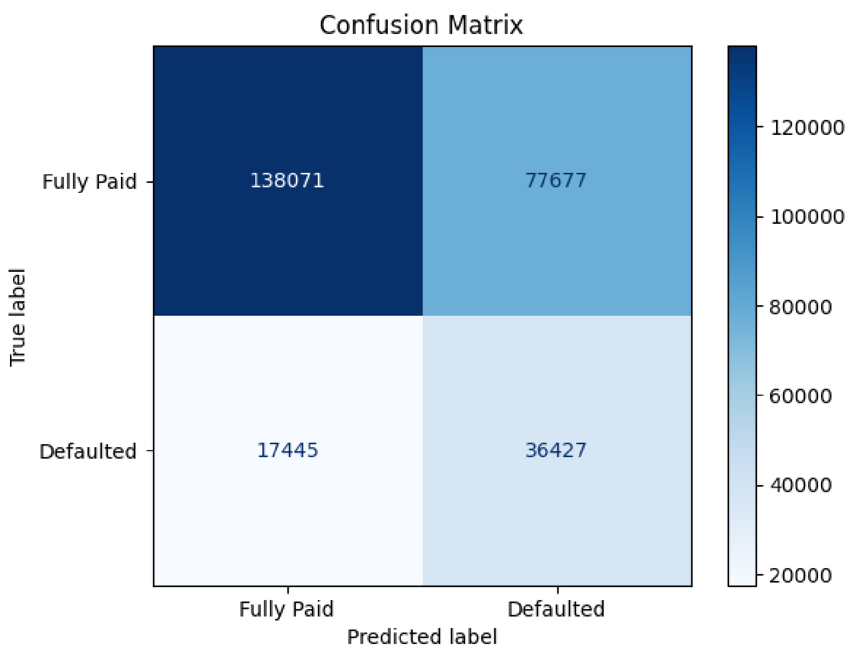
3. Model 1 – Predictive Deep Learning

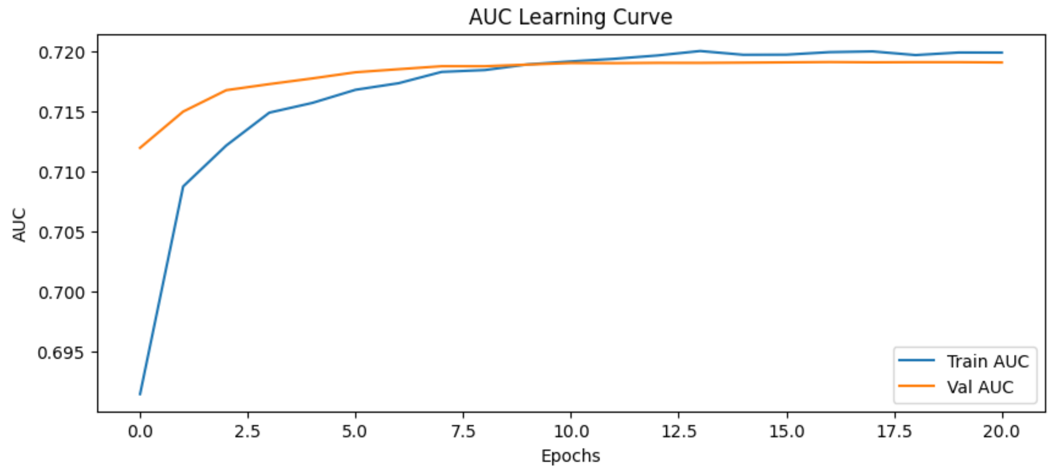
The Deep Learning (DL) model was built using TensorFlow with a multi-layer perceptron (MLP) architecture. It predicts loan default probabilities. The model was trained using binary cross-entropy loss and evaluated on AUC and F1-score metrics.

Test Results:

• AUC: 0.7187

• F1-Score: 0.4357 (best threshold: 0.518)





4. Model 2 – Offline Reinforcement Learning

We framed loan approval as a reinforcement learning problem, where an agent decides whether to approve or deny a loan.

State (s): Vector of applicant features.

Actions (a): {0: Deny, 1: Approve}.

Reward (r):

• Deny = 0

• Approve + Fully Paid = loan_amnt * int_rate

• Approve + Defaulted = -loan_amnt

We used the Discrete Conservative Q-Learning (CQL) algorithm from d3rlpy to train the RL agent. Training was done for 50 epochs on a GPU with batch size 256.

RL Results:

• Avg reward per loan: 0.0691

• Total reward: 18621.94

• Approval rate: 0.564

DL Baseline Comparison:

• DL Avg reward per loan: -0.0354; total: -9547.23; approve rate: 0.387

• Disagreements between DL and RL: 158,661 cases

5. Analysis and Comparison

The Deep Learning model minimizes prediction error, focusing on classifying defaults accurately, whereas the RL agent optimizes expected financial reward. The RL policy approved some high-risk applicants that DL would reject—these were cases where potential interest outweighed risk.

Example: Found 2,459 applicants where DL predicted high default probability (>= 0.8) but RL still approved.

Sample instances:
• DL p=0.8239, RL approved (True=1, loan_amnt=-0.0986, int_rate=1.317)
• DL p=0.8149, RL approved (True=0, loan_amnt=-0.3911, int_rate=1.175)

6. Limitations

• RL uses one-step episodes, ignoring long-term repayment dynamics.
• The dataset lacks behavioral and temporal repayment data.
• Offline RL relies heavily on the logged data quality; rare cases may be underrepresented.

7. Future Work

• Combine DL and RL — use DL predictions as state features for RL.
• Explore other offline RL algorithms (SAC, IQL) for improved policy stability.
• Tune DL thresholds for optimal financial-risk balance.
• Add external data sources (credit bureau, employment records).
• Extend RL to multi-step repayment modeling.

8. Conclusion

This project demonstrates the trade-off between predictive accuracy (DL) and financial optimization (RL). DL offers strong classification capability, while RL learns policies that maximize profit. A hybrid DL+RL framework could yield optimal decision-making for loan approvals.