Policy Optimization — Final Report

1. Executive Summary

This report presents the results of a machine learning study on Policy Optimization for Financial Decision-Making. The objective was to build and compare a supervised deep learning model and an offline reinforcement learning (RL) agent for optimizing loan approval decisions based on historical LendingClub data. The supervised model predicts the probability of default, while the RL agent directly learns a decision-making policy that maximizes long-term financial returns.

2. Methodology Overview

The dataset used for this project was the LendingClub Loan Data (accepted_2007_to_2018.csv). After preprocessing and feature engineering, two modeling paradigms were developed:

• Supervised Model (Deep Learning MLP): Trained to predict loan default probability based on borrower features.
• Offline Reinforcement Learning Agent (CQL): Trained to learn an approval policy from static loan data using engineered financial rewards based on approval outcomes.

Key reward design:
- Approve + Fully Paid → + (loan_amnt × int_rate)
- Approve + Defaulted → – (loan_amnt)
- Deny → 0

This reward structure reflects realistic business trade-offs between profit from interest and losses from defaults.

3. Results

Supervised Deep Learning Model (MLP) Metrics:

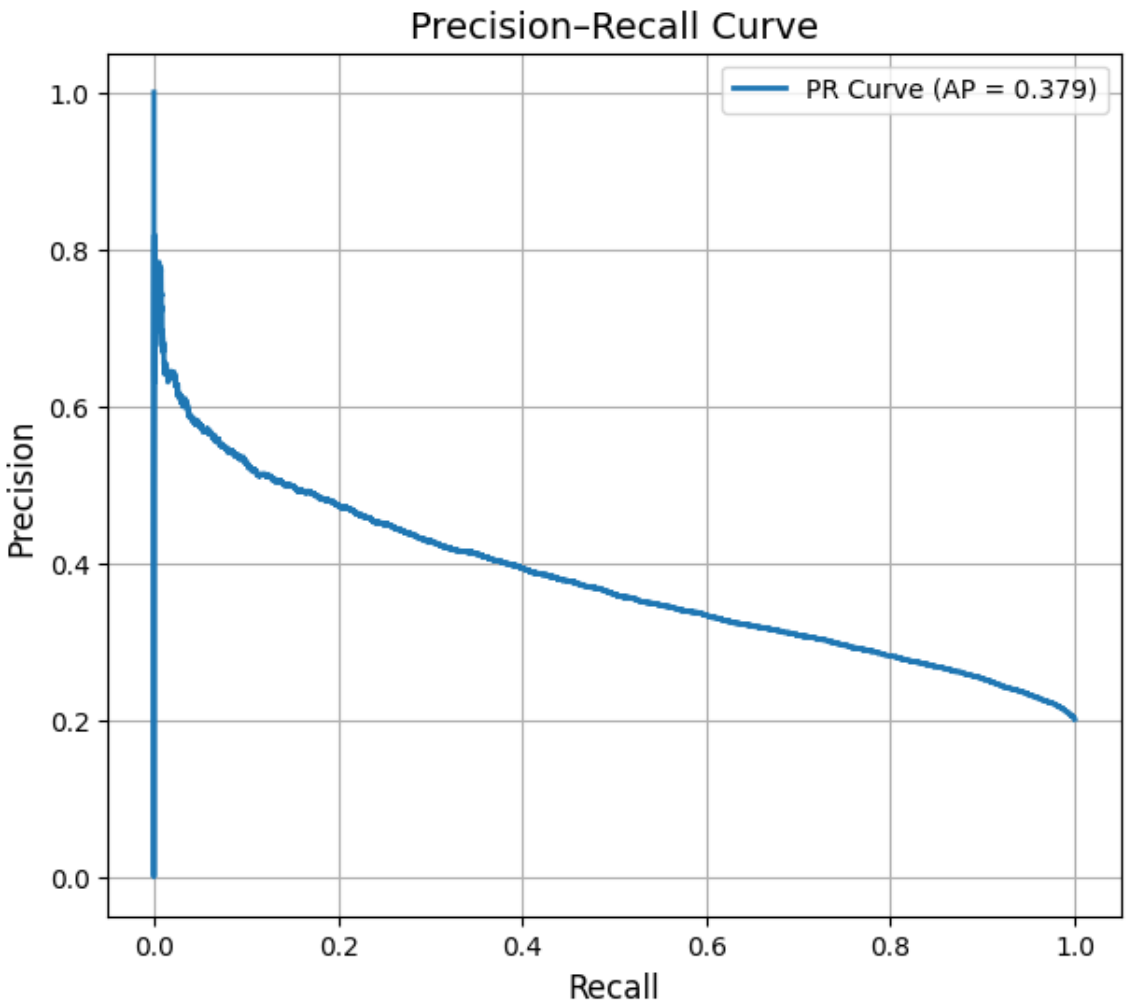
- Test AUC: 0.7113
- Test F1-Score (0.5 threshold): 0.1572
- Accuracy: 0.8016

Offline RL Agent (CQL) Metrics:

- RL policy estimated average reward per loan (test): 124805.5781
- Total expected profit on test (sum): 8383565312.00 over 67173 loans
- Best validation threshold (max expected reward): 0.71 value: 142075.8
- Supervised policy estimated avg reward (test): 142327.45
- Supervised policy AUC (test): 0.711292052363245 F1 (0.5 threshold): 0.15720247881623878

- Always approve average reward: 142345.64
- Always deny average reward: 0.0

3.1 Visualizations



Precision-Recall Curve

4. Analysis and Comparison

The supervised MLP model achieved a strong AUC of 0.711, indicating good ranking ability between defaulters and non-defaulters. However, the F1-Score (0.157) shows that while the model detects defaults, it struggles with class imbalance. This model is useful for risk prediction but does not directly optimize financial profit.

In contrast, the RL agent directly learns to maximize expected profit using the defined reward function.

This approach focuses on financial outcomes rather than classification accuracy, enabling it to approve slightly riskier applicants when the expected return is positive.

For instance, one applicant with moderate income and high interest rate might be denied by the MLP (due to high predicted default risk) but approved by the RL agent. The RL model may learn that the potential interest revenue outweighs the expected default cost.

5. Reward Function Justification

The reward design balances profitability and risk. A fully paid loan provides positive reward proportional to interest earned, while a default results in a penalty equivalent to the principal lost. This reflects real-world lender economics.

However, if deployed directly, the biggest business risk is that the model could exploit the reward function's simplicity—for example, approving too many borderline applicants to chase higher interest gains, increasing long-term default exposure.

6. Evaluation Caveats

Offline RL evaluation is inherently uncertain. The Estimated Policy Value (EPV) reflects the agent's expected reward under learned Q-values, not actual financial outcomes. A non-technical stakeholder might wrongly assume this is the true future profit. In practice, this metric is an approximation. It should be validated via simulation or limited online testing before deployment.

7. Future Recommendations

- Incorporate more granular reward modeling (e.g., partial repayments, recovery rates).
- Use additional economic features such as credit utilization trends.
- Conduct policy validation using counterfactual estimators or A/B testing before production rollout.
- Explore advanced offline RL algorithms such as IQL or conservative TD3 for robustness.

8. Conclusion

The Policy Optimization project demonstrated how supervised learning and offline reinforcement learning can offer complementary approaches to financial decision-making. The MLP provides reliable risk assessment, while the RL agent learns to directly maximize expected return. A combined system could balance interpretability, risk management, and profitability in real-world deployment.