Policy Optimization for Loan Approval: DL vs Offline RL

Jai Awasthi

October 26, 2025

1 Introduction

We analyze two approaches for loan approval: a tabular deep learning (DL) classifier for default prediction and a modern offline reinforcement learning (RL) agent (DQN) trained on historical loan data. This report presents key results, metric rationale, policy comparison, and future directions.

2 Presenting Results

2.1 Deep Learning Model

Key metrics:

- **ROC AUC**: 0.71
- **F1-Score (defaults)**: 0.40
- **Precision (defaults)**: 0.28
- **Recall (defaults)**: 0.69

The DL model achieves moderate accuracy, with strong default recall and fair class separation.

2.2 Offline RL Agent (DQN)

Key metrics:

- **Estimated Policy Value (mean)**: +0.0214
- **Total Reward**: 437.54
- **Approval Fraction**: 0.52

The policy delivers a positive expected return (**+2.14%** per applicant), outperforming baseline strategies (always approve/deny) and showing stable learning.

Policy	Approval %	Avg Reward	Trend
Always Approve	100%	-0.0057	Loss-making
Always Deny	0%	0.0	Neutral
Previous RL (CQL)	60%	+0.0399	Good profit
DQN (current)	52%	+0.0214	Stable, conservative

Table 1: Comparative Policy Performance

3 Metric Choices and Interpretation

3.1 Why AUC/F1 for DL?

AUC measures the model's ability to separate defaulters from non-defaulters over all thresholds, vital for screening loan risk.
F1-score balances precision and recall, critical in class-imbalanced settings: high recall ensures catching risky loans, while F1 prevents over-alerting. DL models implicitly define a threshold-based policy.

3.2 Why Estimated Policy Value for RL?

The RL agent's goal is to maximize expected reward—directly corresponding to business profit/loss for approved loans. Thus, **Estimated Policy Value** is the proper "profit" metric for decisions. Approval fraction and policy stability are also tracked.

4 Policy Comparison

4.1 Policy Mechanism

- **DL Policy**: Approve if predicted default probability < threshold (risk-screening).
- **RL Policy**: Approve/deny by direct reward maximization, learning nuanced tradeoffs.

4.2 Divergent Decisions

Case example: High-risk applicant flagged by DL model (high default score), but RL agent approves.
RL rationale: RL occasionally approves risky applicants when the expected reward (interest vs. loss) is favorable, exploiting business asymmetries not captured by pure risk thresholds.

5 Future Steps and Limitations

5.1 Next Actions

- Test more RL algorithms (CQL, IQL); RL showed lower approval and less profit than CQL.
- Refine reward design, gamma, and risk aversion.
- Conduct robustness checks on policy—simulate new data, validate on out-of-time samples.
- Monitor fairness, adaptivity, and stability before real-world deployment.

5.2 Limitations

- Both models depend on historical data distributions, which may shift.
- DL model struggles with limited feature interactions.
- RL agent is sensitive to reward engineering and discount factors.

5.3 Data and Algorithm Extensions

- Collect more granular recovery, payment history, and external credit info.
- Explore attention-based deep architectures, robust RL variants.
- Evaluate hybrid policy ensembling DL+RL.

6 Summary

Our experiments show that the RL agent learns a conservative, profit-positive policy, while the DL classifier provides a risk-sensitive screening tool. Joint refinement and further testing are needed for production deployment.