Policy Optimization for Decision Making

Final Project Report

Harsh Vardhan

December 12, 2025

Abstract

This report documents the design, implementation, and evaluation of data-driven methods for making optimal approval decisions in a loan underwriting context. Two approaches are compared: a supervised Deep Learning classifier (MLP) trained to predict repayment, and an Offline Reinforcement Learning (CQL-based) policy trained to maximize economic reward. Both methods are evaluated on held-out test data using standard machine learning metrics as well as financial reward-based policy evaluation. The study demonstrates that the Offline RL policy—despite having weaker classification metrics—achieves higher expected business reward, highlighting the importance of optimizing for utility rather than accuracy in decision-making tasks.

Contents

1 Introduction

1.1 Problem Overview

Financial institutions commonly rely on predictive models to guide loan approval decisions. However, traditional supervised models optimize classification metrics (such as AUC and F1), not business outcomes.

This project explores **policy optimization** using:

- A supervised Deep Learning classifier (MLP),
- An Offline Reinforcement Learning agent (CQL),

and evaluates how each approach impacts overall decision reward.

1.2 Project Goals

- Build a reproducible ML and RL pipeline.
- Train a supervised Deep Learning model for loan approval prediction.
- Train an Offline RL policy using logged historical transitions.
- Compare ML and RL models on both classification and reward metrics.
- Produce a final analytical report and reproducible artifacts.

2 Data Overview

2.1 Dataset

The dataset is derived from LendingClub's publicly available loan histories. A uniform sampling procedure produced a representative dataset suitable for Offline RL.

2.2 Train/Val/Test Split

Time-based partitioning ensures realistic future-facing evaluation:

- Older rows → training set
- Middle rows → validation set
- Most recent rows → test set

2.3 Features and Target

The target variable:

$$y = \{ 1 \text{ loan repaid0loan defaulted}$$

Features include borrower credit history, financial ratios, loan terms, income data, employment stability, etc.

3 Preprocessing Pipeline

Preprocessing was implemented as a scikit-learn pipeline and saved to disk as:

```
models/preprocessor_dl.joblib
```

Key steps:

- Numeric conversion and cleaning,
- Missing value imputation,
- Standardization of numerical features,
- Persisted feature ordering for downstream ML/RL steps.

4 Supervised Deep Learning Model

4.1 Architecture

A multi-layer perceptron (MLP) with two hidden layers and ReLU activations.

4.2 Training

- Loss: Binary Cross-Entropy
- Optimizer: Adam
- Regularization: Dropout + class weighting
- Early stopping on validation performance

Model saved as:

```
models/dl/dl_model.pt
```

4.3 Threshold Selection

The classifier outputs probabilities. On validation, sweeping thresholds yielded:

$$t^* = 0.22$$

which maximized F1.

5 Offline Reinforcement Learning (CQL)

5.1 Reward Function

The reward reflects financial outcomes of approval decisions:

$$R = \{ \ loan\_amnt \cdot \frac{\text{int\_rate}}{100} \text{if approved and repaid} - \text{loan\_amntif approved and defaulted0if rejected}$$

5.2 Dataset Format

Each transition contains:

$$(s, a, r, s', \text{terminal})$$

5.3 CQL Training

The script handles multiple d3rlpy versions and falls back to a Logistic Regression policy if CQL fails.

Final RL policy saved as:

```
models/rl/rl_policy.pkl
```

6 Evaluation

6.1 Classification Metrics

Test-set classification metrics:

Table 1: Classification Metrics on Test Set

Model	AUC	F1	Precision	Recall
DL (MLP)	0.578	0.25	0.224	0.4375
RL Policy	–	0.175	0.211	0.593

6.2 Reward Comparison

Decision policies were evaluated by computing the mean reward over the test set.

Table 2: Expected Reward Comparison

Model	Avg Reward	Interpretation
DL (MLP)	-0.0431	Negative expected return per applicant
RL Policy	+0.0363	Positive expected return per applicant

Key finding: The RL policy achieves positive financial return despite inferior classification metrics, illustrating that reward optimization is distinct from prediction accuracy.

7 Discussion

7.1 Why RL Outperforms on Reward

The supervised MLP predicts repayment probability, but this does not align directly with financial value. RL optimizes the policy $\pi(a|s)$ to maximize:

$$E[R]$$

Thus, RL focuses on **utility**, not accuracy.

7.2 Limitations

- Logged bandit-style dataset contains inherent selection bias.
- Reward formulation simplifies the economics of lending.
- d3rlpy signature instability required fallback mechanisms.

8 Artifacts and Reproducibility

Artifacts produced:

- Trained DL model
- Trained RL policy (CQL or fallback)
- Preprocessing pipeline
- Evaluation JSON files
- Final report script

8.1 One-Line Command to Generate Final Report

```
python scripts/final_report.py --test data/processed/test_processed.csv.gz \
--val data/processed/val_processed.csv.gz \
--dl_model models/dl/dl_model.pt --dl_preproc models/preprocessor_dl.joblib \
--rl_policy models/rl/rl_policy.pkl \
--comparison reports/policy_comparison.json --out reports
```

9 Conclusion

This project demonstrates that decision-making algorithms should be evaluated on expected reward rather than predictive accuracy alone. Offline RL—specifically CQL—provides a principled framework to optimize approval policies using logged data, and outperforms supervised learning on business value.

Future work:

- Address selection bias with IPS / doubly robust estimators,
- Develop causal models for counterfactual evaluation,
- Deploy real-time policy serving with monitoring.