Final Report — Loan Policy Optimization (Tasks 1–4 Summary)

Author: Rajat Satonkar (22ucs160)
Date: October 26, 2025

Executive Summary
This project evaluates two approaches to the loan approval decision using the LendingClub accepted loans dataset (2007–2018): (1) a supervised deep-learning classifier that predicts loan default probability (Task 2), and (2) an offline reinforcement learning (RL) agent that directly learns an approval policy from logged historical data (Task 3). The goals are to measure predictive performance (AUC, F1) and to estimate the financial value of a learned policy (Estimated Policy Value, EPV). Below we summarize methods, key results found in the provided notebook, interpret metrics, compare policies, and propose future steps for deployment and improvement.

Methods Overview

Data & preprocessing: Performed EDA, cleaning, and feature engineering; selected top 40 uncorrelated features using SHAP. These 40 features serve as state vectors for both models.
Supervised model (Task 2): Trained a Multi-Layer Perceptron classifier to predict the binary target (0: Fully Paid, 1: Defaulted). Evaluation metrics: ROC AUC and F1-score on a held-out test set.
Offline RL (Task 3): Framed as a one-step contextual bandit (each loan is an independent decision). State = 40-feature vector, Action = {0: Deny, 1: Approve}, Reward = 0 if deny; +loan_amnt*int_rate if approve & paid; -loan_amnt if approve & default. Trained behaviour-cloning baseline and (optionally) conservative offline RL algorithms like CQL or BCQ; evaluated using Estimated Policy Value (EPV) / off-policy evaluation.

Key Results

- Supervised model AUC (ROC): 0.7300
- Supervised model F1-score: 0.4420
- Offline RL: Estimated Policy Value (EPV): Not explicitly found in the notebook outputs. See analysis below.

Why these metrics?

AUC (Area Under the ROC Curve): Measures the classifier's ability to rank positive (default) and negative (fully paid) cases across all thresholds. It is threshold-independent and useful when we care about ordering risk scores for different loan applicants. AUC = 0.73 (found in the notebook) indicates acceptable discrimination between defaulters and non-defaulters.

F1-score: Harmonic mean of precision and recall at a chosen threshold. It balances false positives (approving risky loans) and false negatives (rejecting safe loans). The notebook reports F1 ≈ 0.442, reflecting the classifier's performance at the selected decision threshold—important when classes are imbalanced and both error types matter.

Estimated Policy Value (EPV): For RL, the objective is direct business value: net profit over decisions. EPV estimates expected reward (profit minus losses) if the learned policy were deployed. It aligns directly with the company's financial objective and is therefore the primary metric for policy evaluation.

Comparing the Supervised Classifier Policy and RL Policy

Policy from the classifier (implicit): choose a threshold τ on predicted default probability p■(default|x); approve if p■ < τ. This approach optimizes prediction accuracy and class-based metrics (AUC, F1), but it does not directly optimize expected financial return.

Policy from the RL agent (explicit): learns to maximize expected reward using the reward function that encodes profit and loss. As a result, the RL policy may approve applicants with higher default risk if the expected interest revenue compensates for the chance of loss (e.g., high loan amount with high interest). Conversely, it may deny low-risk loans with tiny interest margins if approving them yields little expected profit compared to the risk.

Example of potential disagreement (illustrative): Consider Applicant A with 85% chance of default according to the classifier but with a very large loan amount and high interest rate. The classifier would likely recommend deny; an RL agent optimizing EPV might still approve if expected profit = (1 - p) * (loan_amnt*int_rate) - p * loan_amnt > 0. That is, if the interest compensates the expected loss, the RL policy will accept.

Findings & Interpretation

- The supervised model achieves reasonable discrimination (AUC ≈ 0.73) but modest F1 (≈ 0.44). This indicates the classifier ranks risk but at operational thresholds may still produce many misclassifications requiring a conservative threshold choice.
- The RL framework reframes the problem to maximize profit, which can change decisions compared to an accuracy-optimizing classifier. Without an EPV reported in the notebook, a direct numeric comparison is unavailable; however, the policy differences are interpretable via reward arithmetic.
- Behaviour Cloning is a useful baseline (it imitates historical decisions), but robust offline RL methods (e.g., CQL, BCQ) are preferred because they penalize implausible out-of-distribution actions and attempt to reliably improve expected reward under offline data constraints.

Limitations & Recommended Future Steps

Limitations:
- Logged-data bias: dataset contains only accepted loans (selection bias). The agent never observes outcomes for historically denied applicants, complicating reliable policy learning.
- Offline evaluation: EPV and OPE estimators depend on assumptions; estimates can be biased if the behavior policy is poorly modeled.
- Reward scaling: raw monetary rewards can have large variance; proper normalization/stabilization is necessary during training.

Recommended next steps:
1. Compute EPV using robust off-policy evaluation (Importance Sampling, Doubly Robust estimators) and report confidence intervals. Use tools available in d3rlpy or other OPE libraries.
2. Train conservative offline RL (CQL) and compare EPV vs behaviour cloning and a thresholded classifier policy (e.g., approve if p■ < 0.2). Save policies and Q-values for analysis.
3. Collect or simulate outcomes for historically denied loans (or run a small randomized trial/A-B test) to reduce selection bias.
4. Add constraints and risk controls (e.g., portfolio-level exposure limits, max exposure per borrower segment).
5. Build explainability diagnostics (SHAP on classifier; feature contributions to Q-values for RL) to analyze disagreements between policies.
6. If EPV advantage is convincing and OPE confidence intervals are acceptable, deploy in a staged manner (shadow mode ➜ small-scale pilot ➜ full roll-out).

Conclusion

This project shows how predictive modeling and offline reinforcement learning offer complementary approaches: one predicts risk; the other optimizes financial outcome directly. Both are valuable. Based on the notebook, the supervised model achieves reasonable discrimination (AUC~0.73). The RL approach should be developed further—compute EPV, run robust OPE, and mitigate logged-data bias—before deployment. The recommended path is to iterate on offline evaluation and then run small-scale pilots to validate EPV in production.

Appendix: Extracted Notebook Findings

- Notebook path: /mnt/data/policy_optimization_Rajat_Satonkar_22ucs160.ipynb
- Cells parsed: 51
- Found supervised metrics: AUC = 0.7300, F1 = 0.4420
- Evidence of behavioural cloning in notebook: Yes