Final Report: Policy Optimization for Financial Decision-Making

This project explores how data-driven methods can optimize LendingClub's loan approval process to balance profit and risk. Two approaches were compared:

1. A Deep Learning (MLP) model for predicting default probabilities.
2. An Offline Reinforcement Learning (CQL) agent for directly learning profit-maximizing approval policies.

After extensive analysis using time-based data splits (train: pre-2017, test: 2017–2018), the results reveal that while the MLP model is effective for ranking loan risk (AUC = 0.7165), the RL agent was unable to improve upon the existing policy due to selection bias ,it only had data from previously approved loans.

In short:

- The historical policy was unprofitable, losing about $191.86 per loan during the test period.
- The RL agent simply mirrored the historical "approve-all" policy, because it had never seen the outcomes of denied loans.
- The MLP model, though imperfect for binary decisions, can still be valuable as a risk-ranking engine to guide approval thresholds.

The most practical next step is to deploy the MLP as a probability-based approval system while collecting new data that includes both approved and denied loans. This would later allow a properly trained RL agent to optimize the lending strategy more effectively.

Methodology

Data Preparation

- **Dataset:** accepted_2007_to_2018Q4.csv (LendingClub).
- **Target:** is_default = 1 for 'Charged Off' or 'Default', and 0 for 'Fully Paid'.
- **Non-terminal statuses** (e.g., 'Current', 'Late') were excluded.
- **Train/Test Split:** Time-based — loans before 2015 for training, 2015–2018 for testing.
- **Feature Filtering:** Only pre-loan attributes were used (e.g., fico_score, dti, annual_inc) to prevent data leakage.

Deep Learning (MLP)

Goal: Predict default probability.
Results:

Metric	**Score**
AUC	0.7165
Precision	0.34
Recall	0.66

The MLP ranks risk effectively but misclassifies many good borrowers. Best used as a **risk-ranking model**, not for direct automated decisions.

Results

Policy	**Avg. Profit/Loss per Loan**
Always Deny	$0.00
Historical (Approve All)	–$191.86
RL Agent (CQL)	–$191.86

Difference in Metrics

- **AUC/F1 (for MLP):** These measure prediction quality.
  - AUC (0.7165): Proves your model is good at *ranking risk*. It can tell a risky applicant from a safe one.
  - F1 (0.4512): Proves your model is a bad *decision-maker* at the 0.5 threshold because its Precision (0.34) is too low (it flags too many good loans as bad).
- **Estimated Policy Value (for RL):** This measures financial outcome.
  - What it is: It calculates the average profit or loss (in dollars) per loan, directly answering the business goal. Your EPV of $-191.86 means the policy loses $191.86 per loan.

Policy Comparison

- **DL vs. RL Decisions:** A DL model denies based on risk probability. An RL agent denies based on expected value. An RL agent could approve a high-risk loan if the interest rate (reward) is high enough to make it a profitable gamble.
- **Your Finding:** Your RL agent learned to "Always Approve" because your dataset had selection bias. It only saw examples of "Approve" (action 1) and had no data for "Deny" (action 0). It copied the only action it knew.

Future Steps

- **Immediate Next Step**: Use your MLP (Model 1). Instead, find a new, profit-maximizing threshold by testing its predicted probabilities against the actual financial rewards on the test set.
- **Limitations:** The selection bias of the "accepted-only" dataset is the biggest limitation.
- **Data to Collect:** You must collect data for the "Deny" action. The best way is an A/B test (e.g., randomly deny 5% of loans you would normally approve) to build an unbiased dataset.
- **Other Algorithms:** Use XGBoost or LightGBM (which are better for tabular data) to improve your risk prediction (AUC).