

# Policy Optimization for Financial Decision-Making

## Final Report

**Author:** Siddhant Garg

**Project:** Loan Approval Optimization using Deep Learning and Reinforcement Learning

---

## 1. Introduction and Problem Overview

### Business Context

This project addresses the challenge of optimizing loan approval decisions for a fintech company. The goal is to develop an intelligent system that maximizes financial return while managing default risk. Using the LendingClub dataset containing 2.2M historical loan records, I implemented and compared two distinct approaches: supervised deep learning and offline reinforcement learning.

### Dataset

- **Source:** LendingClub Loan Data (2007-2018)
  - **Size:** 1,348,059 samples after filtering completed loans
  - **Features:** 31 selected features including borrower financials, loan characteristics, and credit history
  - **Target:** Binary outcome (Fully Paid vs Default)
  - **Class Distribution:** 79.9% Fully Paid, 20.1% Default (imbalance ratio: 4.0:1)
- 

## 2. Methodology

### Task 1: Data Preprocessing

I performed comprehensive exploratory data analysis and preprocessing:

- **Missing Value Handling:** Filled numerical features with median, categorical with mode

- **Feature Selection:** Selected nine features, avoiding data leakage (excluded payment history, recoveries)
- **Feature Engineering:** Created average FICO score, converted term to numerical, processed percentages
- **Encoding:** Applied label encoding for high-cardinality features, one-hot encoding for low-cardinality
- **Scaling:** StandardScaler for all numerical features
- **Split:** 70% train, 15% validation, 15% test (stratified)

Key insights from EDA:

- **Loan Amount & Installment are almost perfectly correlated ( $\approx 0.95$ )**  
This means higher loan amounts directly lead to higher installments. One of these features may be redundant for your ML model.
- **Open Accounts & Total Accounts show a strong positive correlation ( $\approx 0.72$ )**  
This is expected since total accounts usually include open ones. Again, potential feature redundancy.
- **Revolving Balance has moderate correlation with Loan Amount & Installment ( $\approx 0.30-0.32$ )**  
Customers taking larger loans tend to carry higher revolving balances, indicating higher credit usage behavior.
- **Interest Rate is weakly correlated with most variables (mostly between  $-0.05$  to  $0.12$ )**  
This suggests interest rate is likely influenced by **external factors (like credit score or policy)** rather than these financial variables.

## Task 2: Deep Learning Classifier

**Architecture:**

- Multi-Layer Perceptron: Input(31)  $\rightarrow$  128  $\rightarrow$  64  $\rightarrow$  32  $\rightarrow$  Output(1)
- Batch Normalization and Dropout (0.3) for regularization
- Sigmoid activation for binary classification

**Training:**

- Loss: BCEWithLogitsLoss with class weights (ratio: [YOUR\_RATIO])
- Optimizer: Adam (lr=0.001, weight\_decay=1e-5)
- Batch size: 512, Early stopping: patience=7
- Training completed in 30 epochs

**Results:**

Metric	Score
AUC-ROC	0.7154
F1-Score	0.4167
Accuracy	0.7438
Precision	0.3823
Recall	0.4538
Optimal Threshold	0.60

### Task 3: Offline Reinforcement Learning Agent

#### MDP Formulation:

- **State (s):** Preprocessed feature vector for each applicant
- **Action (a):** {0: Deny, 1: Approve}
- **Reward (r):**
  - Deny: 0 (no risk, no gain)
  - Approve + Fully Paid:  $\text{loan\_amnt} \times \text{int\_rate}$  (interest profit)
  - Approve + Default:  $-\text{loan\_amnt}$  (principal loss)

#### Algorithm: Conservative Q-Learning (CQL)

- Specifically designed for offline RL
- Prevents overestimation of Q-values for unseen actions
- Learning rate:  $3e-4$ , Batch size: 256
- Conservative weight ( $\alpha$ ): 1.0
- Training steps: 5

#### Results:

Metric	Value
Policy Value	\$-1, 648.87 per loan
Total Return	\$-333,427,030.59
Approval Rate	100%
Default Rate (approved)	19.98%
Improvement over Baseline	0%

---

## 3. Analysis and Comparison

### 3.1 Why Are the Metrics Different?

**Deep Learning Model** uses classification metrics (AUC, F1-Score):

- These measure **prediction accuracy**
- AUC-ROC: How well the model distinguishes between classes
- F1-Score: Balance between precision and recall
- Goal: Correctly classify loans as "will default" or "will be paid"
- **Limitation:** Does not consider business impact (profit/loss)

**RL Agent** uses policy value (average reward):

- This measures **financial performance**
- Policy Value: Expected profit/loss per loan decision
- Total Return: Cumulative profit across all decisions
- Goal: Maximize financial return, not prediction accuracy
- **Advantage:** Directly optimizes business objective

**Key Insight:** A model with lower accuracy might have higher profit if it approves high-interest loans with acceptable risk.

### 3.2 Policy Comparison

Converting the DL model to a policy (threshold=0.60):

- **DL Policy Value:** \$-481.55 per loan
- **RL Policy Value:** \$-1,648.87 per loan
- **Difference:** \$-1,167.33 per loan

**Approval Rates:**

- Baseline (Approve All): 100%
- DL Model: 76.06%
- RL Agent: 100%

### 3.3 Disagreement Analysis

The models disagreed on 48,415 cases (23.94%).

**Example 1: High-Risk Loans RL Approves** Found 22500 high-risk loans (>70% default probability) that RL approves but DL denies.

Example case:

- Loan Amount: \$22,375
- Interest Rate: 18.25%
- DL Predicted Default: 72.84%
- True Outcome: DEFAULT
- **Why RL Approves:**

$$\text{Expected Value} = (1 - 0.7284) \times 22,375 \times 18.25\% - 0.7284 \times 22,375$$

$$\text{Expected Value} = \$-15,189.40$$

The high interest rate compensates for the risk, making the expected value positive.

---

## 4. Limitations and Future Work

### Current Limitations

1. **Offline RL Constraints:** Learned only from approved loans; no counterfactual data on denied loans
2. **Simplified Reward:** Does not account for recovery costs, opportunity costs, or customer lifetime value
3. **Single-Step Decision:** Real lending involves multiple decisions (amount, rate, term adjustments)
4. **Potential Bias:** Historical data may contain demographic biases
5. **Static Environment:** Does not adapt to changing economic conditions

### Proposed Future Steps

#### Immediate Next Steps:

1. **Pilot Deployment:** Test RL agent on small portfolio (5-10% of applications)
2. **A/B Testing:** Compare against current approval process
3. **Fairness Audit:** Check for demographic bias in approval decisions
4. **Human Oversight:** Require manual review for loans above \$0.60

#### Technical Improvements:

1. **Enhanced Rewards:** Include recovery amounts, opportunity costs, customer lifetime value
2. **Online Learning:** Implement contextual bandits for continuous learning from new data
3. **Multi-Objective:** Optimize for profit AND fairness constraints
4. **Explainability:** Add SHAP values for interpretable decisions

5. **Advanced Algorithms:** Experiment with IQL, BCQ, or ensemble methods

#### Data Collection:

1. Employment verification details and bank transaction history
  2. Payment history from other lenders
  3. Post-approval monitoring data for continuous learning
- 

## 5. Conclusion

This project successfully demonstrated two approaches to loan approval optimization:

- **Deep Learning:** Achieves **0.7154 AUC-ROC**, effectively predicting default risk.
- **Reinforcement Learning:** Achieves **\$-1,648.87 policy value per loan**, directly optimizing financial return.

#### Key Takeaway:

In this study, the **Deep Learning model clearly outperforms the Reinforcement Learning agent in financial outcomes when converted into a decision policy**. While the RL agent is designed to maximize long-term reward, it failed to outperform even the baseline strategy due to limitations of offline training and historical data bias.

- The **DL-based policy achieved a better policy value of \$-481.55 per loan**, compared to the RL agent's **\$-1,648.87 per loan**.
- The DL model also reduced the **default rate on approved loans to 14.24%**, compared to **19.98%** for the baseline and RL agent.

This confirms that the **DL model provides more reliable risk assessment and stronger financial control under the current dataset and reward setup**.

---

## Recommendation

**Deploy the Deep Learning model** as the primary loan approval system with the following conditions:

- Use the **optimized threshold of 0.60** for approval decisions
- **Implement continuous monitoring and periodic retraining** to adapt to market changes
- **Proceed with a pilot deployment** on a limited portfolio before full-scale rollout
- Integrate **human oversight for high-value loan applications**

At the same time, the **Reinforcement Learning agent should be retained for future research and enhancement**, especially after improving:

- Reward function design
- Offline RL limitations
- Data coverage and fairness auditing

---

## Final Insight

This project demonstrates that the choice between **prediction accuracy (Deep Learning)** and **direct policy optimization (Reinforcement Learning)** depends heavily on the **quality of data, reward formulation, and business constraints**. While DL proved superior under current conditions, **future hybrid models combining DL-based risk scoring with RL-based decision optimization represent the most promising direction forward**.

---

## References

1. LendingClub Loan Data: <https://www.kaggle.com/datasets/wordsforthewise/lending-club>
2. Kumar, A. et al. (2020). "Conservative Q-Learning for Offline Reinforcement Learning"
3. d3rlpy Documentation: <https://d3rlpy.readthedocs.io/>
4. PyTorch Documentation: <https://pytorch.org/docs/>