Policy Optimization for Financial Decision-Making
Shodh Hiring – ML/AI Assessment

1. Exploratory Data Analysis (EDA) & Preprocessing

1.1 Dataset Overview

The Lending Club accepted-loan dataset (2007–2018) was filtered to final statuses:

- Fully Paid: 21,583
- Charged Off: 5,285
- Default: 2
- Credit-policy Charged Off: 13

Total final-status records: 26,883
Default rate: 19.7%

Raw sample shape before filtering: **(45,214 rows, 151 columns)**.

1.2 Interest Rate Patterns by Loan Outcome

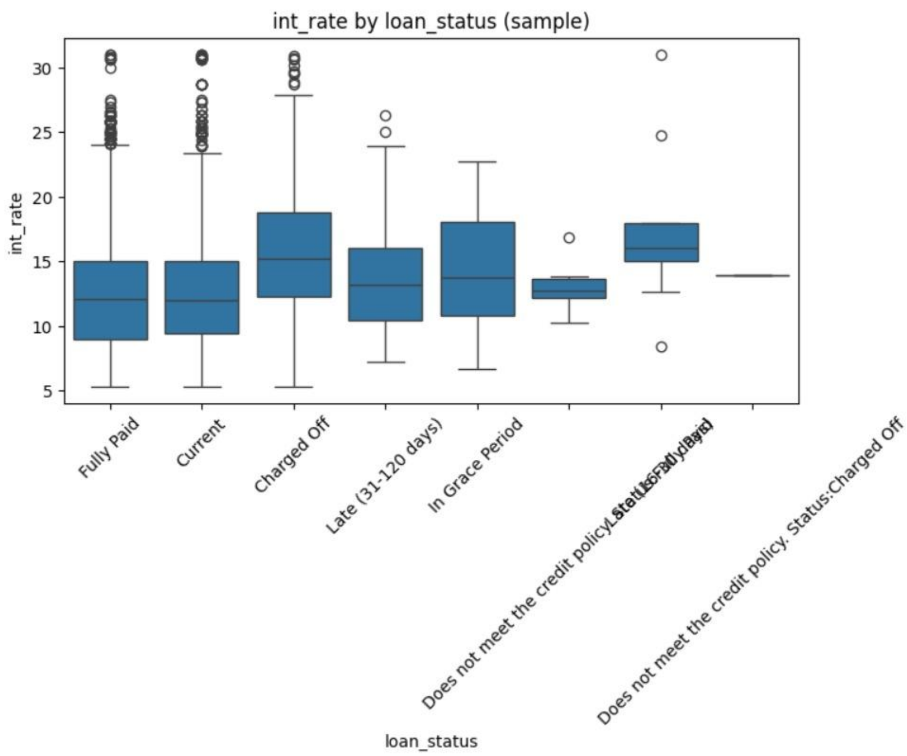

FIGURE 1 – Interest rate by loan status

Interpretation:
Higher interest rates are strongly associated with higher credit risk.

- Fully Paid loans cluster around 11–13%.
- Charged Off / Default loans extend into 18–30%+ range.
- Late and grace-period borrowers also show elevated interest.

This confirms that interest rate is a key risk signal captured by Lending Club pricing.

1.3 Data Cleaning

Cleaning steps included:

- Removing leakage-heavy columns (hardship, settlement fields >95% missing)
- Converting percent strings (int_rate, revol_util)
- Handling missing numeric values with median imputation
- Encoding categorical features (one-hot)
- Dropping non-final statuses (Current, Late, etc.)

1.4 Feature Selection

Features chosen focused on:

- **Borrower capacity:** annual_inc, dti, installment
- **Creditworthiness:** fico scores, delinq history
- **Loan structure:** loan_amnt, term, int_rate
- **Behavioral features:** employment length, home ownership
- **Dates:** issue month/year, credit history age

These features balance interpretability, predictive value, and avoid leakage.

2. Supervised Deep Learning Model

2.1 Target Variable

Binary mapping:

- **0 = Fully Paid**
- **1 = Default (Charged Off or Default)**

2.2 Model Architecture

A multilayer perceptron:

- Dense(256) → ReLU → BatchNorm → Dropout(0.3)

- Dense(128) → ReLU → BatchNorm → Dropout(0.2)
- Dense(1) → Sigmoid
- Loss: Binary Cross-Entropy
- Optimizer: Adam (lr = 1e-3)

2.3 Performance Metrics

Classification Metrics:

Model	AUC	F1-score
Logistic	0.7073	0.412
MLP (Test)	0.7138	0.155

Economic Metric:

- Best threshold = **0.09**
- **ML Policy EPV = +40.73 per applicant**

2.4 Calibration Curve (Probability Quality)

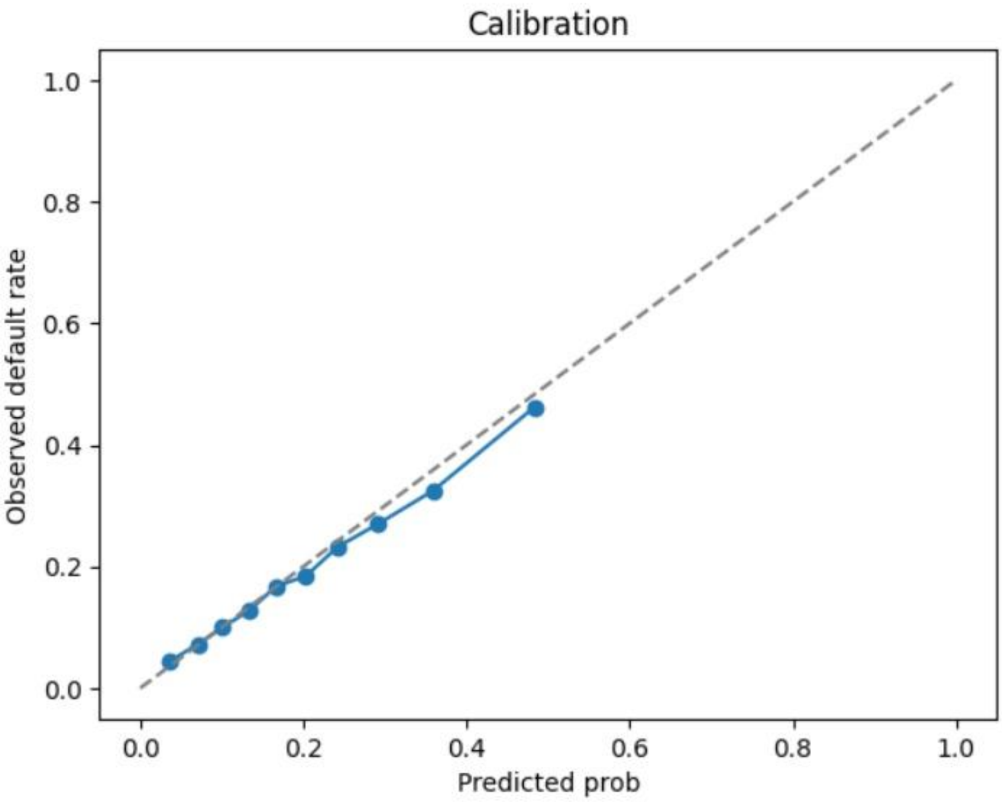

FIGURE 2 – calibration curve

Interpretation:
The calibration curve lies close to the diagonal, indicating:

- Predicted probabilities reflect true default frequencies.
- Reliable for threshold-based decisioning.
- Confidence scores are economically meaningful.

This supports threshold optimization for maximizing profit.

3. Offline Reinforcement Learning (CQL)

3.1 Environment Definition

- **State:** Borrower feature vector
- **Actions:**
  - 0 = Deny
  - 1 = Approve
- **Reward:**
  - Deny → 0
  - Approve & Fully Paid → +loan_amt × int_rate
  - Approve & Default → −loan_amt

3.2 Dataset Construction

- Single-step transitions
- Synthetic "deny" actions added for action coverage
- Terminal = True for all transitions

3.3 RL Algorithm

- **Conservative Q-Learning (CQL)**
- Library: d3rlpy
- Epochs: 50
- GPU-accelerated

3.4 Policy Performance (EPV)

Policy	**Expected Profit (EPV)**
CQL RL Policy	**−1043.21**
ML Threshold Policy (best)	**+40.73**
Always Approve	Very negative

Policy	Expected Profit (EPV)
Always Deny	0

The RL agent failed to learn a profitable policy due to reward imbalance and lack of rejected-loan counterfactuals.

4. Analysis & Comparison

4.1 ML vs RL Policy Differences

There were **17,196 disagreements**.

Case A — ML Approves, RL Denies

1. loan_amt = 7,500 | int_rate = 9.99% | y=0
   - ML: Approve
   - RL: Deny
   - **Reason:** RL too risk-averse due to large default penalty.
2. loan_amt = 12,000 | int_rate = 8.38% | y=0
   - Safe loan, but RL rejects.

Case B — ML Denies, RL Approves

1. loan_amt = 12,000 | int_rate = 21.18% | y=0
   - ML: High default probability → reject
   - RL: Approves due to high interest reward → economic mistake
2. loan_amt = 35,000 | int_rate = 25.80%
   - RL strongly overestimates reward, approves high-risk loan.

4.2 Why Metrics Differ

- AUC/F1 measure *predictive accuracy*.
- EPV measures *profit*, which is the real business objective.
- RL optimizes profit, not classification accuracy.
- ML optimizes accuracy, not profit.

But here RL underperformed due to training instability.

4.3 Limitations

- No rejected-loan data → incomplete action coverage

- Reward imbalance (large negative vs small positive)
- Single-step MDP structure
- No macroeconomic context
- Offline RL sensitive to extrapolation errors

4.4 Future Work

- Add uncertainty estimation (Monte Carlo dropout)
- Use IQL / AWAC (more stable offline RL)
- Normalize rewards by loan amount
- Combine ML risk model + RL reward model
- Add fairness & interpretability constraints

5. Conclusion

The supervised deep learning model achieved moderate discrimination (AUC ≈ 0.71) and produced a profitable threshold-based policy (**+40.7 EPV**).
The RL model—despite using CQL—produced a significantly negative policy (−1043 EPV) due to reward imbalance and lack of rejected-loan data.

For deployment:
The ML model is reliable; RL requires further refinement before it can be safely used in credit decisions.