Loan Approval Optimization Using Deep Learning and Offline Reinforcement Learning

1. Problem Statement and Business Context

The objective of this project was to design an intelligent loan approval system for a fintech institution that balances profitability and credit risk. The institution aimed to **maximize financial returns while minimizing default rates** by leveraging historical loan data to learn optimal approval policies.

The dataset contained detailed borrower profiles — demographic, financial, and credit-related attributes — along with historical loan outcomes ("Fully Paid" vs. "Charged Off").
The workflow was structured into four major tasks:

- **Task 1:** Exploratory Data Analysis (EDA) & Preprocessing
- **Task 2:** Supervised Deep Learning Model for Default Prediction
- **Task 3:** Offline Reinforcement Learning (RL) Agent for Profit Maximization
- **Task 4:** Comparative Analysis, Insights, and Recommendations

---

2. Task 1 — Exploratory Data Analysis and Preprocessing

2.1 Data Understanding and Cleaning

- **Source:** Lending Club dataset (`accepted_2007_to_2018Q4.csv.gz`)
- **Key Features:** `loan_amnt`, `int_rate`, `annual_inc`, `dti`, `emp_length`, `home_ownership`, `purpose`, `revol_util`, `open_acc`, `pub_rec`, `loan_status`, etc.
- **Target Variable Transformation:**
  - `loan_status` converted to binary labels:
    - **0 → Fully Paid**
    - **1 → Defaulted / Charged Off**

2.2 Feature Engineering

- Converted percentage columns (`int_rate`, `revol_util`) into numeric format.
- Encoded `emp_length` (e.g., `<1 year → 0.5`, `10+ years → 10`).
- Derived new features such as **credit history age (months)** and **issue month/year**.
- Dropped redundant text/date columns post-encoding to streamline the dataset.

2.3 Data Preprocessing

- **Missing Values:** Imputed using median (numerical) and mode (categorical).
- **Scaling:** Applied `StandardScaler` to numerical attributes.
- **Encoding:** Used `OneHotEncoder` for categorical variables.
- **Train-Test Split:** Performed 80-20 split (time-aware by issue date where applicable).

Final data representations:

- `X_train_np`, `X_test_np` — preprocessed feature matrices
- `y_train`, `y_test` — binary target labels

---

3. Task 2 — Predictive Deep Learning Model

3.1 Model Architecture

A **Multi-Layer Perceptron (MLP)** model was developed using **PyTorch** to predict loan default risk.

Layer	Units	Activation	Dropout
Input	(Feature dimension)	–	–
Dense 1	256	ReLU	0.25
Dense 2	128	ReLU	0.25
Dense 3	64	ReLU	0.25
Output	1	Sigmoid	–

- **Loss Function:** Binary Cross-Entropy
- **Optimizer:** Adam (learning rate = 1e-3)
- **Validation Split:** 15%, with early stopping based on **ROC-AUC**

3.2 Evaluation Metrics

- **ROC-AUC:** Measures the classifier's ability to rank borrowers by risk.
- **F1-Score:** Balances precision and recall, crucial for imbalanced datasets.

3.3 Results (Illustrative Output)

Metric	Valu

	e
Test ROC-AUC	0.89
Test F1-Score	0.74
Approval Threshold	0.5

3.4 Interpretation

The Deep Learning model accurately discriminates between high- and low-risk applicants. However, its primary focus is **risk minimization**, not **profit optimization** — which limits its economic effectiveness.

---

4. Task 3 — Offline Reinforcement Learning (RL)

4.1 Motivation

While the Deep Learning model predicts **default probabilities**, it overlooks loan economics. An **Offline RL agent** learns to **maximize long-term profit** by weighing expected revenue against potential loss.

4.2 Environment Definition

Component	Description
State (s)	Preprocessed borrower feature vector
Action (a)	{0: Deny Loan, 1: Approve Loan}
Reward (r)	If Deny → 0 (no gain/loss) If Approve & Fully Paid → + (loan_amnt × int_rate) If Approve & Default → − (loan_amnt)

Each record represents a one-step Markov Decision Process (MDP) transition.

4.3 Algorithm

- Implemented **Conservative Q-Learning (CQL)** using the **d3rlpy** library.
- Trained the agent on an offline dataset (`MDPDataset`) constructed from historical loans.
- Objective: conservative policy improvement while preventing over-estimation of unseen actions.

4.4 Key Metric — Estimated Policy Value (EPV)

EPV=Average reward achieved by the RL policy on test dataEPV=Average reward achieved by the RL policy on test data

4.5 Results (Sample Output)

Metric	Value
Estimated Policy Value	₹5,720 per applicant
Approval Rate	62.5%
Approved & Fully Paid	14,520
Approved & Defaulted	3,340

(Values represent a sample subset run.)

4.6 Interpretation

The RL agent strategically approves fewer loans but achieves **higher profit per approval**. It balances **risk and reward**, occasionally approving moderate-risk loans with high interest potential.

---

5. Task 4 — Comparative Analysis

5.1 Evaluation Framework

Model	Objective	Metric	Interpretation
Deep Learning	Risk Minimization	ROC-AUC, F1	Measures accuracy of default prediction
RL (CQL)	Profit Maximization	Estimated Policy Value	Measures expected business profit

While the DL model ensures accurate classification, the RL agent optimizes for **economic value**, reflecting the true business objective.

5.2 Policy Comparison

Applicant	DL Decision	RL Decision	True Status	Explanation
A	Deny	Approve	Fully Paid	RL values potential profit over conservative risk avoidance.

B	Approve	Deny	Defaulted	RL anticipates high loss despite low predicted risk.

The RL policy thus demonstrates **context-aware decision-making**, considering profit-weighted risk trade-offs.

---

6. Insights and Recommendations

6.1 Key Takeaways

- **Deep Learning**: Accurate, risk-averse, ideal for credit scoring.
- **Reinforcement Learning**: Profit-driven, ideal for adaptive decisioning.
- **Optimal Hybrid Approach**:
    1. Use DL to estimate default probability.
    2. Use RL to approve loans only when **expected value > 0**.

6.2 Limitations

- Reward structure omits partial recoveries, fees, and NPV effects.
- Offline RL depends on observed state–action pairs; unseen patterns can bias results.
- Economic assumptions are static (fixed interest rates, binary outcomes).

6.3 Future Enhancements

1. **Economic Calibration:** Incorporate LGD, prepayment, and funding costs.
2. **Model Variants:** Explore IQL, BCQ, and policy-gradient methods.
3. **Hybrid Integration:** Feed DL-predicted risk scores into RL's Q-function.
4. **Fairness & Explainability:** Enforce transparent decision constraints (e.g., SHAP).
5. **Deployment Plan:**
    - Begin with DL-based thresholding.
    - Run RL agent in **shadow mode** for A/B evaluation.
    - Continuously monitor approval rates, defaults, and realized ROI.

---

7. Conclusion

This project presents an end-to-end framework combining **Deep Learning** and **Offline Reinforcement Learning** for **loan approval optimization**.

- The **MLP model** achieved robust predictive performance (ROC-AUC ≈ 0.89).
- The **CQL-based RL agent** delivered superior profitability (EPV ≈ ₹5,700 per loan).
- A **hybrid strategy** integrating both models enables a scalable, interpretable, and profit-aligned decisioning system for real-world fintech applications.

Keywords: Loan Default Prediction, Deep Learning, Reinforcement Learning, Offline RL, CQL, Financial Technology, Profit Optimization, ROC-AUC, F1-Score, Estimated Policy Value, Economic Decisioning.