REPORT ANALYSIS

(SHODH AI)

by: Pranav Anand

1. Summary of Results

Supervised Deep Learning Model (DL Model)

- **Objective:** Predict the probability that a borrower will default.
- **Reported Metrics:**
  - **AUC (ROC Area):** Measures how well the model separates defaulters vs. fully paid borrowers.
  - **F1-Score:** Balances precision and recall, especially important due to class imbalance in loan default datasets.

Offline Reinforcement Learning Agent (RL Agent)

- **Objective:** Learn an approval/denial **policy** that maximizes long-term financial returns using offline loan data.
- **Reported Metric:**
  - **Estimated Policy Value (EPV):** Expected financial return of the learned policy using off-policy evaluation.

2. Why These Metrics Are Relevant

Why AUC & F1-Score for the DL Model?

- **AUC** evaluates the model's ranking ability, independent of a threshold.
  - Higher AUC → model better at separating high-risk vs low-risk borrowers.
- **F1-Score** is crucial because loan datasets are **imbalanced** (defaults are rare).
  - High precision → fewer false approvals of risky loans.
  - High recall → fewer missed high-risk borrowers.

These metrics tell us **how good the model is at classification**, but not how profitable or costly the decisions are.

Why Estimated Policy Value (EPV) for RL Agent?

- RL optimizes **financial outcomes**, not accuracy.
- EPV directly measures how much profit (or loss) the policy yields under business rules:
  - Approve + fully paid → profit from interest
  - Approve + default → loss of principal
  - Deny → 0 reward

Thus, EPV directly aligns RL training with **business ROI**, not prediction accuracy.

3. Comparing the Policies

How the DL Model Makes Decisions

- DL implicitly defines a policy such as:
  “Approve if Predicted Default Probability < θ”
  where θ is a fixed threshold.
- It does **not** consider actual profit/loss — only classification correctness.

How the RL Agent Makes Decisions

The RL Agent learns a **financially optimal policy**, for example:

- Approve high-interest loans even if moderately risky (because expected return > risk).
- Deny low-interest loans even with moderate risk (reward not worth the risk).

This is because RL’s objective is **maximizing expected profit, not classification accuracy**.

Examples Where the Two Models Disagree

Example 1 — RL Approves but DL Denies

Feature	Value
Loan Amount	₹80,000
Interest Rate	27%
DL Prediction	High default probability → Deny
RL Decision	Approve

Why?

- Even if default probability is ~25%, the **expected return**
  → *0.75 × high interest profit – 0.25 × loan amount*
  may still be positive.
- RL sees long-term ROI; DL only sees risk.

Example 2 — DL Approves but RL Denies

Feature	Value
Loan Amount	₹300,000
Interest Rate	6%
DL Prediction	Low probability of default → Approve
RL Decision	Deny

Why?

- Low interest → low reward
- Even a **small chance of default** means a **high potential loss**
- RL avoids low-profit + high-risk combinations

Interpretation

- **DL is risk-sensitive** (denies many loans to avoid misclassification).
- **RL is reward-sensitive** (approves loans only when expected profit is positive).

This difference shows why **accuracy ≠ profitability**.

4. Strategic Insights & Future Steps

Which Model Is Better for Deployment?

- If **avoiding risk** is the priority → DL model is safer, more conservative.
- If **maximizing profit** is the priority → RL policy is superior because it optimizes ROI.

However, real-world deployment may require combining both.

Limitations of the Current Approach

DL Model

- Does not incorporate financial values (loan amount, interest) directly into optimization.
- Does not reflect real-world costs of misclassification (defaults are far more costly than false positives).

RL Agent

- Offline RL depends heavily on dataset quality and action distribution.
- If historical data lacks examples of optimal decisions, RL policy may be biased.
- Reward engineering is simplistic; real financial systems have complex loss structures.

What Additional Data Would Improve Both Models?

- Borrower behavioral history (salary trends, repayment patterns).
- Real-time features like credit utilization and bank statements.
- Time-dependent states for RL (e.g., credit improvement over time).
- Information on rejected loan applications (counterfactual analysis).

Future Algorithmic Improvements

For DL

- Use cost-sensitive learning or focal loss for class imbalance.
- Try gradient boosting models (XGBoost, LightGBM), known to perform extremely well on tabular financial data.

For RL

- Use advanced offline RL methods:
  - **CQL (Conservative Q-Learning)**
  - **BCQ (Batch Constrained Q-Learning)**
  - **IQL (Implicit Q-Learning)**
- Combine DL predictions with RL rewards (hybrid decision system).

Hybrid System Recommendation

A practical production system could be:

DL model screens applicants → RL model evaluates profitability for approved cases.

This reduces risk while improving returns.

Final Conclusion

- The **DL model** excels at prediction accuracy but not profit optimization.
- The **RL agent** directly optimizes for business value and can outperform DL in profitability.
- Their decisions differ because their **optimization goals differ**.
- A combined approach is likely to deliver the best real-world performance.