Policy Optimization for Financial Decision-Making

1) Executive Summary

We built two complementary approaches to optimize lending decisions:

- **Supervised Deep Learning (DL)**: an MLP that predicts **default probability** from borrower/application features, converted to a policy via a probability threshold ("approve if p(default) < τ").
- **Offline Reinforcement Learning (RL)**: a **Conservative Q-Learning (CQL)** policy trained on logged approvals/denials to **directly** optimize expected profit under a handcrafted reward.

Headlines (your results):

- **DL model:** AUC **0.717**, F1@0.5 **0.234**; best operating threshold **0.10** yields **+45.41** expected profit per application.
- **RL policy (CQL):** Approval rate **66.7%**; Estimated Policy Value (EPV) ≈ **−1,253** across On-support, IPS, DM, DR estimators.

Interpretation:
The DL model ranks risk reliably and, when tuned at τ=0.10, achieves a small **positive** profit per application. The RL policy's **negative** EPV originates from the **reward specification** (gain on payoffs is interest × principal, but loss on default is full principal) rather than an algorithmic failure; it overweights the downside of defaults relative to the upside, pushing absolute EPV negative even for sensible policies. This is expected under such reward scaling and can be corrected with more realistic business assumptions (term-scaled interest and default recoveries).

2) Data, Preprocessing & Feature Engineering (Task 1)

Data: accepted_2007_to_2018Q4.csv (and rejected_2007_to_2018Q4.csv for deny coverage in RL).
Target: Binary loan_status → 0 = Fully Paid, 1 = Default/Charged-off.
Selected predictors (22): loan amount, interest rate, installment, annual income, DTI, **FICO midpoint (engineered)**, delinquency/inquiry counts, open accounts, public records, revolving balance/utilization, total accounts; plus categorical: term, grade, employment length, home ownership, verification status, purpose, application type, state, **issue year (engineered)**.

Preprocessing Pipeline (reusable):

- Numeric: median imputation + standard scaling.
- Categorical: most-frequent imputation + one-hot encoding (ignore unknown).
- Saved as **preprocessor.joblib** to guarantee consistent transforms across DL and RL.

Key EDA observations:

- Defaults correlate with **higher interest rates**, **worse grades (E–G)**, **longer terms (60m)**, **lower FICO** and **higher DTI**.
- Class imbalance is meaningful (defaults ≈ 20%); metrics like AUC and PR/F1 are appropriate for evaluation.

Model 1 — Supervised Deep Learning (Task 2)

Architecture: MLP (256→128→64 with BN & Dropout), sigmoid output for default probability.
Training: Stratified split; BCE loss; Adam; P100 GPU on Kaggle.
Evaluation metrics:

- **AUC:** 0.717 (good ranking performance for credit risk)
- **F1 @ 0.5:** 0.234 (reasonable given imbalance)

Turning scores into a policy:
We compute expected profit per application as:

- Approve & **Fully Paid**: + loan_amount × interest_rate
- Approve & **Default**: − loan_amount
- **Deny**: 0

Sweeping thresholds, we select τ=**0.10** (approve if p(default) < 0.10), which maximizes the profit curve at **+45.41** per application. This is a conservative but profitable operating point.

Takeaway: The DL model is a strong **risk ranker** and, when thresholded for business goals, yields **positive expected profit**.

4) Model 2 — Offline RL Policy (Task 3)

Setup: One-step static bandit framing.

- **State (s):** preprocessed feature vector.
- **Action (a):** {0 deny, 1 approve}.
- **Reward (r):**
    - If **deny**: 0
    - If **approve & fully paid**: + loan_amnt × int_rate
    - If **approve & default**: − loan_amnt

Dataset for RL:

- Accepted loans as action=1 with realized rewards per target.
- Rejected loans as action=0 with reward=0 (used to ensure deny coverage; zero-state placeholders for features).
- Sampled **100k accepted + 50k rejected** to fit memory/time budgets.

Algorithm: CQL (Discrete) via d3rlpy; trained 50k steps.
Policy statistics: Approval rate **66.7%**.
Policy evaluation (EPV):

- On-support: **−1,252.83**
- IPS (clipped): **−1,252.83**
- Direct Method: **−1,253.33**
- Doubly Robust: **−1,253.45**

Why EPV is negative:
Under the reward design, a default inflicts a very large loss (**−principal**) while a fully paid loan yields a comparatively smaller gain (**+principal × annual rate**). Even with many good approvals, occasional defaults drive the **absolute** EPV negative. This is a **reward scale** choice, not a training bug.

Business-realistic reward should include:

- **Term scaling**: interest × (term_months / 12)
- **Recovery on default**: e.g., recover 30–50% of principal
- (Optional) servicing costs, discounting, late fees

With that, the same CQL policy is expected to move EPV closer to or above zero, making RL numbers directly comparable to the DL profit curve.