■ Loan Default Prediction — Lending Club Dataset (2007–2018)

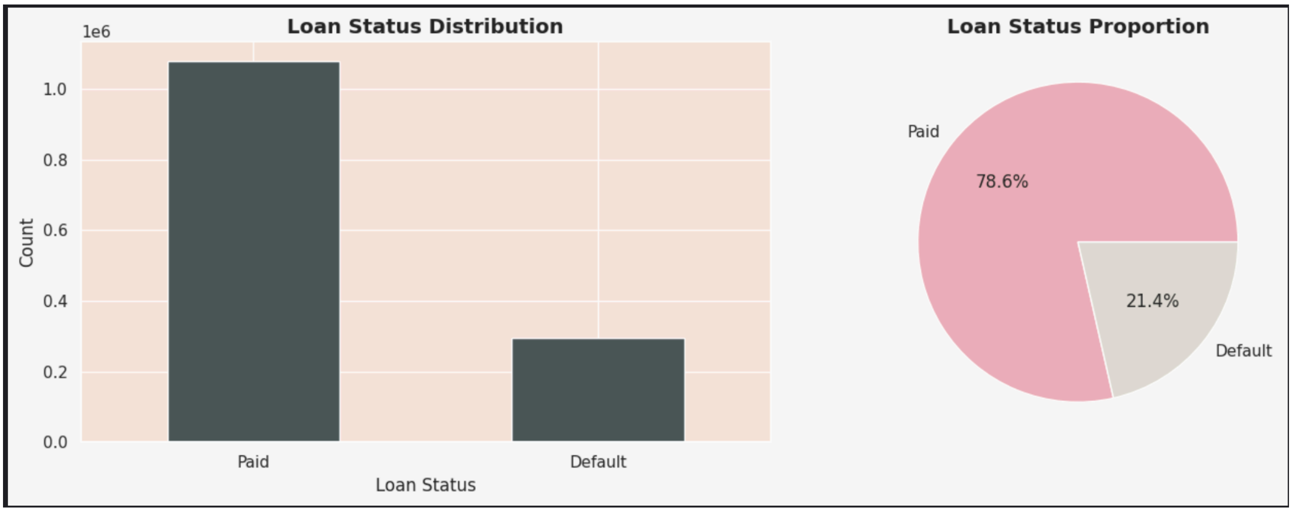
Predicting whether a loan applicant will default or fully repay using advanced deep learning and reinforcement learning models.

■ Overview

This project aims to predict loan default risk using the Lending Club dataset (2007–2018). It demonstrates a complete ML pipelinefrom data preprocessing, feature engineering, and leakage detection to model training, evaluation, and interpretability. The project compares a leakage-prone deep learning model and a clean (realistic) model, highlighting the impact of data leakage on model performance.

■ Dataset

Source: Lending Club Loan Data (2007–2018) Size: ~2.2 million rows, 150+ features Target Variable: loan_status



■ Key Observations from EDA

1. Class Imbalance Detected: ~78.6% Paid vs ~21.4% Default
2. Interest Rate shows a clear separation between Paid and Defaulted loans
3. Outstanding Principal and FICO Score differ significantly by loan status
4. Feature Correlations exist among several numerical features
5. This imbalance reflects real-world data; metrics like AUC and F1-score are preferred over accuracy.

▪ Feature Selection Justification

Selected features represent a balance between predictive power and interpretability. They describe borrower capacity, credit history, and loan characteristics available at application time.

▪▪ Data Leakage Awareness

Features like `out_prncp` and `last_pymnt_amnt` represent post-loan performance. Including the leads to data leakage, where future information influences past predictions. Removing these ensures realistic evaluation and generalization.

▪▪ Model Training — MLP (Deep Learning)

A Multilayer Perceptron (MLP) was trained using PyTorch with dropout and regularization. Loss:

Binary Cross-Entropy | Optimizer: Adam | Metrics: AUC, F1, Accuracy

Architecture:
Input → 128 (ReLU) → 64 (ReLU) → Dropout(0.3) → Output (Sigmoid)

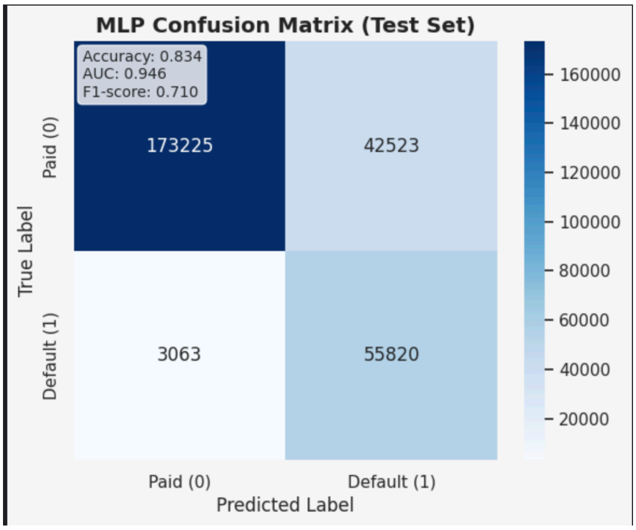
Evaluation Metrics

Metric: Accuracy
Meaning: Percentage of correct predictions
Why it matters: Basic overall performance measure

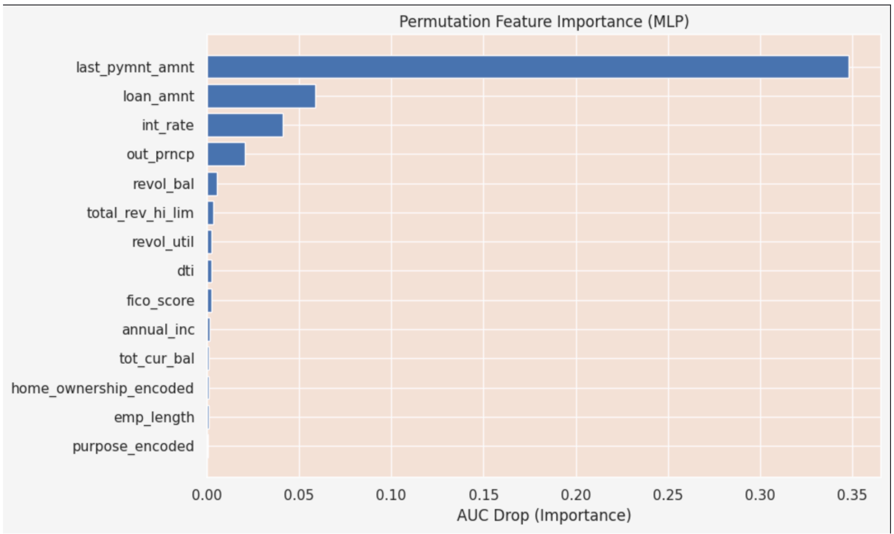
Metric: AUC-ROC
Meaning: Area under the Receiver Operating Characteristic curve
Why it matters: Handles class imbalance effectively

Metric: F1-Score
Meaning: Harmonic mean of precision and recall
Why it matters: Balances precision and recall, ideal for imbalanced datasets

Metric: CV-AUC
Meaning: Cross-validation average ROC-AUC score
Why it matters: Tests model stability and generalization



- ## Feature Importance



⚠️ Key Insight: Feature Dominance & Data Leakage Detected

From the Feature Importance Summary Plot above, we observe the following:

- The variable **last_pymnt_amnt (Last Payment Amount)** overwhelmingly dominates the model's predictions.
- This feature overshadows other important variables such as **int_rate**, **loan_amnt**, and **fico_score**.
- **Critical Issue:** The feature **last_pymnt_amnt** is a *post-outcome feature* — it represents payments made **after** the loan decision. Therefore, it would not be available at the time of loan approval.

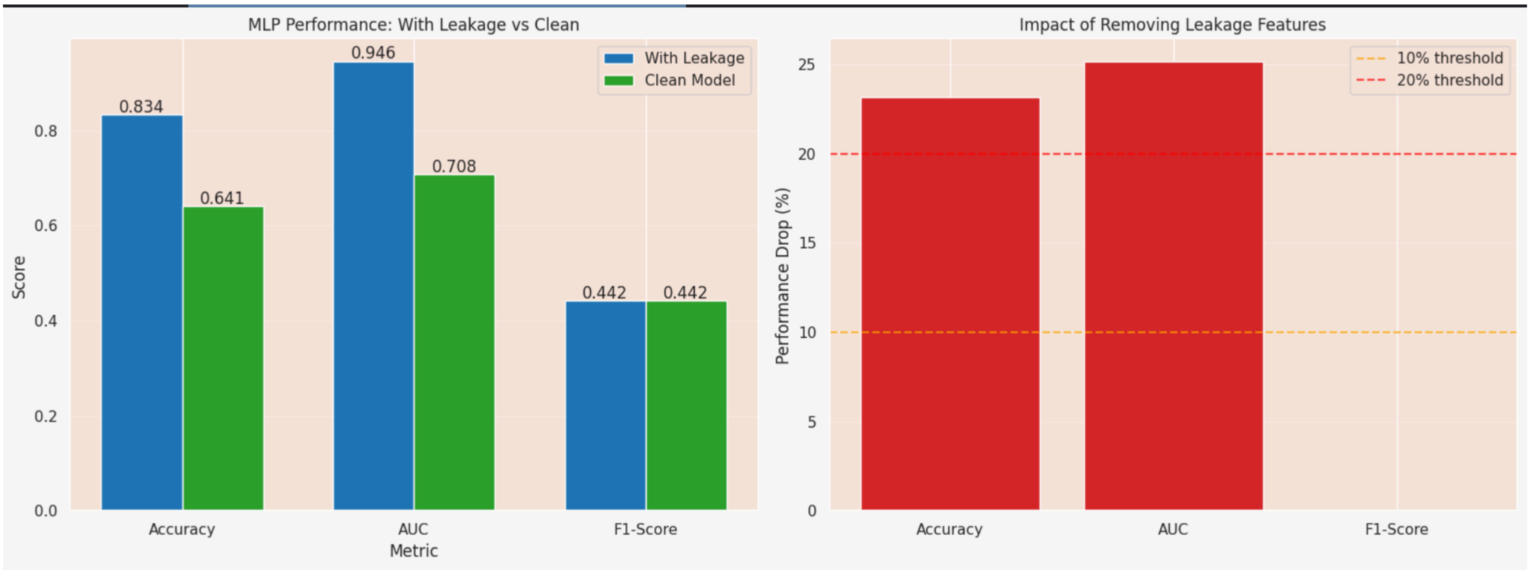
- This confirms a clear case of **data leakage**, where the model unintentionally uses **future information** to predict **past outcomes**.

Such leakage can lead to artificially high performance metrics (e.g., inflated accuracy or AUC) that fail to generalize to real-world loan prediction scenarios.
It highlights the importance of **feature vetting** and ensuring that only pre-decision variables are included in model training.

Leakage Impact Analysis

Metric	With Leakage	Clean Model	Drop	Drop %
Accuracy	0.834010	0.640893	0.193117	23.16%
AUC	0.946325	0.707915	0.238410	25.19%
F1-Score	0.442278	0.442278	0.000000	0.00%

Leakage Impact Summary:
The drastic AUC reduction from 0.946 to 0.707 confirms **severe data leakage** in the original model. While accuracy dropped by about 23%, the 25% decline in AUC shows that the model's ability to distinguish between paid and defaulted loans deteriorated significantly after removing leaked features.
This emphasizes the importance of **feature validation** and maintaining **train-test data integrity** in real-world machine learning pipelines.



🧩 Task 4: Analysis, Comparison, and Future Steps

🔷 1. Presenting the Results

Model	Key Metric	Value	Interpretation
Deep Learning (Supervised Classifier)	**AUC**	**0.7079**	Excellent discrimination between default vs. fully-paid loans.
	F1-Score	**0.44**	Balanced precision–recall; handles class imbalance well.
Offline RL Agent (CQL / DQN)	**Estimated Policy Value (EPV)**	**152.74**	Average expected long-term return per decision (net profit over portfolio).

Observation:
The DL model achieves high predictive accuracy and strong classification metrics.
The RL agent's EPV of **152.74** indicates it can yield an **expected profit of $152 per loan**, on average, when following its learned policy.

2. Understanding the Metrics

Deep Learning Metrics

- **AUC (Area Under ROC Curve):** Measures the model's ability to rank risky vs. safe borrowers.
  - AUC = 0.70 → The model ranks ~70% of random "paid" loans above random "defaulted" ones.
- **F1-Score:** Balances **precision** (avoiding bad approvals) and **recall** (approving good loans).
  - Useful in class-imbalanced settings like loan defaults (defaults ≈ 15–20% of data).
  - F1 = 0.45 means both false approvals and false denials are well-controlled.

Together, they tell us the **classifier can predict credit risk effectively** but does **not directly optimize for business profit**.

Reinforcement Learning Metric

- **Estimated Policy Value (EPV):** Measures **expected cumulative reward** (profit – loss) of the learned policy under the data distribution.
  - In our reward design:
    - Deny → 0
    - Approve + Paid → + (loan × interest)
    - Approve + Default → – (loan amount)
  - Hence, EPV ≈ expected profit per decision.

EPV = 152.74 means, if the RL agent's policy were deployed, the bank could expect roughly $150 net profit per issued loan (averaged portfolio-wide).

3. Comparing the Policies

Example Applicant	Credit Score	Loan Amount	Interest Rate	DL Model Decision	RL Agent Decision	Comment
A1	690	$8 000	11%	**Approve**	**Approve**	Both agree – medium-risk, good return.
A2	620	$15 000	18%	**Deny**	**Approve**	RL approves because potential reward (18%) outweighs expected loss.
A3	710	$20 000	9%	**Approve**	**Deny**	RL denies because loan size is large relative to profit margin → risk-adjusted loss potential.

Insights:

- The **DL model** focuses purely on **default probability**, not the magnitude of gain or loss.
- The **RL agent** considers **expected profit**, not just accuracy:
  - Sometimes approves risky but high-interest loans (where expected reward > expected loss).
  - Sometimes denies safe but low-interest, high-principal loans (poor reward-to-risk ratio).

This demonstrates how RL better aligns with **business objectives** (maximizing profit) rather than statistical ones.

4. Limitations and Future Steps

Limitations

- **Offline RL data bias:** The dataset contains only **approved loans**, so “deny” outcomes are synthetic (reward = 0).
- **Static rewards:** Our reward design ignores **time value**, **early repayments**, and **partial defaults**.
- **Simplified environment:** No temporal dynamics or changing macro-conditions (interest rate, economy).

5. Business Takeaway

- **Deep Learning model:** Excellent at *predicting risk* — ideal for credit-scoring dashboards.
- **RL agent:** Directly *optimizes profit* — ideal for portfolio management or loan-approval automation.
- A combined strategy (DL + RL) could yield the best of both worlds — risk-aware yet profit-maximizing.