Final Report

Project: Policy Optimization for Financial Decision-Making
Name: Lekhan Thati
Date: 12-09-2025

Introduction

- **Objective:**
    - This project is designed to assess the ability to handle a real-world machine learning problem from start to finish. We will take a public dataset, perform analysis, frame it for both supervised learning and reinforcement learning, build and train models, and—most importantly—critically analyze and compare their behaviors and outcomes.

- **Business Context:**
    - Imagine you are a Research Scientist at a fintech company. The company wants to improve its loan approval process. You have access to a historical dataset of all loans granted in the past, including applicant details and whether they ultimately defaulted or paid off the loan. The goal is to develop an intelligent system that can decide whether to approve or deny a new loan application to maximize the company's financial return.

- **Dataset:** The LendingClub dataset (link)

Task 1: Exploratory Data Analysis (EDA) and Preprocessing

- **Analyzed the Data:** Performed an initial EDA to understand the features, identified missing values, and examined the distribution of key variables (e.g., loan status, interest rate, applicant income).

- **Engineered & Selected Features:** From the many available columns, selected a meaningful subset of features that were likely to be predictive of loan default and justified the choices made.

- **Cleaned the Data:** Preprocessed the selected features by handling missing values, encoding categorical variables, and applying feature scaling. All steps were clearly documented.

Task 2: Model 1 – Predictive Deep Learning Model

- **Defined the Target**
- **Built and Trained the Model:** Implemented a deep learning model (a Multi-Layer Perceptron using TensorFlow) to predict the probability of default based on applicant features.
- **Evaluated the Model:** Trained the model and evaluated its performance using standard classification metrics, reporting both the AUC (Area Under the ROC Curve) and the F1-Score on a held-out test set.

Task 3: Model 2 – Offline Reinforcement Learning Agent

Defined the RL Environment (from the static dataset):

- **State (s):** Represented as the vector of preprocessed features for each loan applicant.
- **Action (a):** Defined as a discrete action space {0: Deny Loan, 1: Approve Loan}.
- **Reward (r):** Engineered a reward structure where:
  - If action = Deny → reward = 0 (no risk, no gain).
  - If action = Approve and the loan was Fully Paid → reward = + (loan_amnt × int_rate).
  - If action = Approve and the loan was Defaulted → reward = − loan_amnt (loss of the principal).

Trained an Offline RL Agent:

- Used a modern offline RL framework d3rply to train the agent.
- Selected DQN algorithm a suitable offline RL algorithm.
- Trained the agent on the prepared dataset.

Task 4: Analysis, Comparison, and Future Steps

5.1 Results Summary

- **DL Model :**
    - AUC : 0.999, F1-Score : 0.998

- **RL Agent :**
    - Estimated Policy Value : 1446.71

5.2 Why These Metrics?

- **AUC (Area Under the ROC Curve):**
    - Most suitable metric for Binary class classification and can also be used as a metric to compare performance across different models
    - In this business context AUC can help us understand how well the model ranks the good buyers as well as bad buyers at all thresholds.
    - This tells the bank how reliably the model separates *creditworthy* from *risky* applicants
- **F1-Score (Harmonic mean of Precision & Recall):**
    - Precision = out of predicted positives, how many were actually positive.
    - Recall = out of actual positives, how many did we correctly predict.
    - F1 shows how well the model performs when put into action with a chosen threshold.
- **Estimated Policy Value (EPV):**
    - Unlike for DL Models AUC and F1 score may not be the best metric for Reinforcement learning since training is done in the form of episodes or sequential decision making
    - EPV estimates the expected return (cumulative reward) if we deploy a learned policy
    - EPV directly reflects the business impact of the policy.

5.3 Policy Comparison

- **Examples of applicants where the two models would make different decisions:**
    - **Very High Reward :**
        - Applicants having a high loan amount and high interest implies high reward. The RL model can approve the request (since high reward) and the DL model can reject the request since a potential bad borrower.

    - **Very Low Reward :**
        - Applicants having low loan amount and low interest implies lower reward. The RL model can reject the request (Since low return) and the DL model can accept the request since a potential good borrower.
    - **Overall accumulated reward :**
        - The RL model may have learned a certain pattern where the reward at a certain step may be negative but the overall accumulated reward of that episode is positive.And if an applicant with similar state is passed through the RL model then it approves the request and DL model may reject it.
    - **Edge cases :**
        - During the training of RL model during the exploration phase, the model tries to make a random choice rather than the policy. This random choice may not guarantee to cover a certain edge case where the DL model is trained on and these models may choose a different decision.

5.4 Future Directions

- The DL model performance seems promising and can be deployed.

- The RL model is not suitable for this use case due to architectural drawbacks.

- The neutral network architecture is sufficient for this business use case, but if there are any computational constraints Logistics Regression can also be used.

---

6. Conclusion

- Recommendation for business use case : Deep Learning model.
