Multi-objective Optimization on the German Credit Dataset

Sarvesh Kumar *IMT2022521*
IIIT Bangalore
Bangalore, India
SarveshKumar.A@iiitb.ac.in

Ayush Gupta *IMT2022546*
IIIT Bangalore
Bangalore, India
Ayush.Gupta546@iiitb.ac.in

1 Introduction and Motivation

The German Credit dataset is a widely studied benchmark in the fairness and classification communities. The core objective is to build a classifier that not only performs well in terms of predictive accuracy but also upholds fairness across sensitive attributes such as gender or age.

In many real-world applications like loan approval or hiring decisions, there is often a trade-off between model accuracy and fairness. Multi-objective optimization (MOO) provides a principled approach to handle such conflicting objectives by generating a Pareto frontier that represents optimal trade-offs.

This project focuses on the dual objectives of minimizing classification error and improving fairness metrics. We aim to discover a diverse set of non-dominated solutions that enable stakeholders to make informed trade-offs.

2 Choosing a Baseline Model

We evaluated several models based on their accuracy on this dataset and found that XGBoost consistently outperformed the others. Consequently, we selected XGBoost for further hyperparameter optimization.

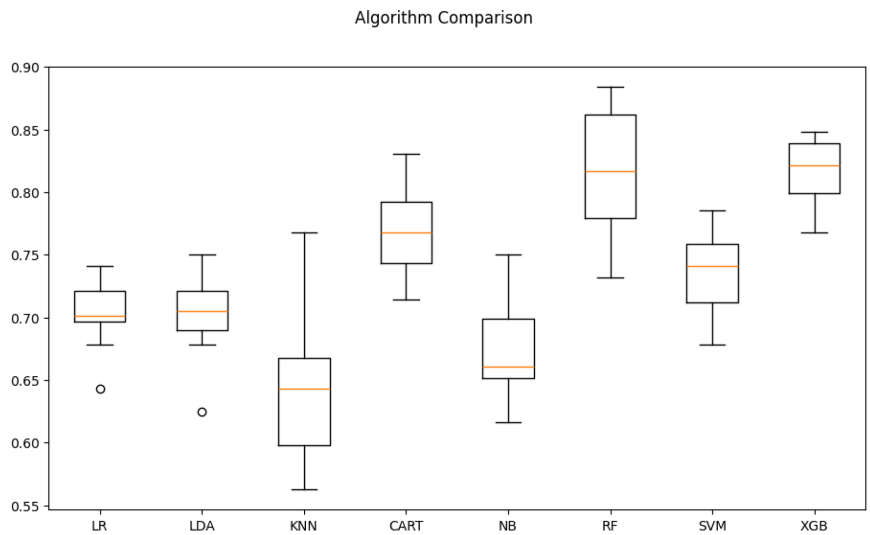

Figure 1: Different models analyzed and their performance.

3 Algorithm Description and Justification

We employ a customized NSGA-II (Non-dominated Sorting Genetic Algorithm II) based strategy for multi-objective optimization. This is done using the Optuna library. Our method works as follows:

1. For a given value of α, we perform NSGA-II optimization over 100 iterations to find a set of non-dominated solutions.
2. The objective function is a weighted combination of classification loss and a fairness metric, parameterized by α. The XGBoost model is trained with a custom loss that is a weighted objective of accuracy and dpd score. It is mathematically represented as

$$(1 - \alpha) \cdot (1 - \mathrm{DPD}) + \alpha \cdot \mathrm{Accuracy}$$

3. From the resulting set of non-dominated points, we select the one that yields the best test performance for the given alpha. The same objective function is used here.
4. Initially, we run the optimizer for $\alpha = 0$ and $\alpha = 1$ to understand the boundaries of the Pareto front.
5. Subsequently, we adaptively generate new α values to fill in the gaps and obtain a well-distributed Pareto frontier. This is repeated until we cover 30 different α values.

We chose this strategy due to its effectiveness in approximating Pareto-optimal solutions in high-dimensional search spaces and its ability to handle non-convex trade-off surfaces. NSGA-II is particularly well-suited for our case since it maintains diversity in the population and promotes uniform coverage of the frontier.

4 NSGA-II: A Brief Overview

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) is a widely used evolutionary algorithm for solving multi-objective optimization problems. It maintains a population of candidate solutions and evolves them over multiple generations using genetic operations. Key steps in NSGA-II:

1. **Initialization:** Generate an initial population of random solutions.
2. **Non-dominated Sorting:** Classify the population into Pareto fronts based on dominance. Solutions that are not dominated by any others form the first front, and so on.
3. **Crowding Distance:** For diversity, compute a crowding distance metric within each front to favor solutions in less crowded areas.
4. **Selection:** Use a binary tournament based on rank (front number) and crowding distance to select parents.
5. **Crossover and Mutation:** Generate offspring using crossover and mutation operators.
6. **Survivor Selection:** Combine parent and offspring populations, sort them again, and select the top solutions for the next generation.

5 Numerical Results

We present below the Pareto frontier obtained by our method using 30 values of α and 100 iterations for each. Each point represents a non-dominated trade-off between accuracy and fairness. Some values of α returned points that were dominated and hence are not included in the pareto front.

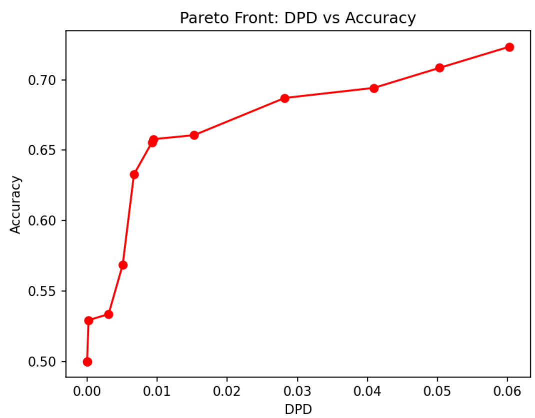

Figure 2: Pareto frontier showing trade-offs between accuracy and fairness.

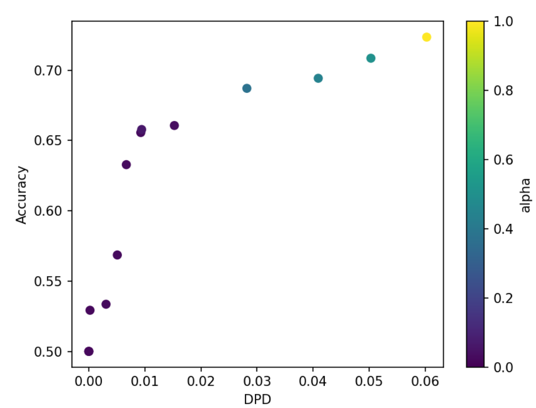

Figure 3: Non-dominated solutions and corresponding alpha values.

The results indicate a smooth and diverse frontier, demonstrating that our adaptive strategy succeeds in achieving uniform coverage of the trade-off space. Figure 3 shows how different alphas prioritize Accuracy and Fairness. We see with $\alpha = 0$, DPD is 0. This indicates perfect fairness, however, accuracy is only 50%. With $\alpha = 1$, we see a relatively high DPD score of 0.06, however we get a better accuracy with 72.36%.

6 Code and Execution Instructions

The codebase has been organized and shared in the GitHub repository. All relevant notebooks and scripts are included in a zip file. The core optimization logic is implemented in the notebook *optuna-solver.ipynb*. Run the notebooks in the following order:

1. *pre-process.ipynb* - Prepare and preprocess the data.
2. *baseline.ipynb* - Establish baseline model performance.
3. *optuna-solver.ipynb* - Execute the optimization and generate results.