

# From Code to Ratings: Converting Programming Data to Enhance Collaborative Filtering in Educational Online Judge Systems

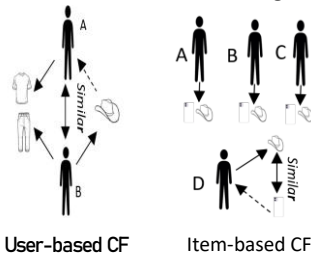
Muepu Mukendi Daniel and Yutaka Watanobe

List of Variables: D= Problem Difficulty, PU= User performance, P=Problem, W= Weight, AVG\_SR=Average of SPS, AVG\_DT= Average of DT, SPS= Solving Performance Score, NA= Number of Attempts, DT= Debugging Time, SR= Success Rate, S= Satisfaction Score, SU= User Skill, Level, I= Ideal Easy Problem, ND= Negative Difference, TD= Total Differences

## Abstract

This study explores and compares three approaches for recommending programming problems in an Online Judge (OJ) system, focusing on deriving implicit ratings from user interactions, as explicit ratings are often unavailable. The study analyzes OJ submissions to develop implicit ratings based on user engagement and problem-solving performance. Each approach assigns satisfaction scores (1-5) and integrates them into a Neural CF model to enhance recommendation accuracy. These approaches were compared against a baseline binary method, with results showing the proposed methods provided more accurate and diverse recommendations, factoring in problem difficulty and user skill for better personalization.

## Collaborative Filtering (CF)



CF is based on the assumption that users who agreed in the past will agree in the future

## Data suitable for CF

| user | Movie | Rating |
|------|-------|--------|
| 1    |       | X 5    |
| 2    |       | X 3    |
| 3    |       | Y 2    |
| 4    |       | Y 1    |
| 1    |       | Y 1    |
| 2    |       | Y 6    |

Traditional CF methodologies primarily rely on explicit user ratings, which are not typically available in educational contexts.

## User problem Data in OJ

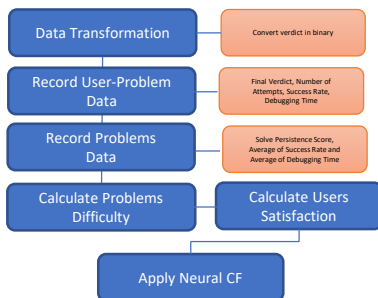
| user | problem  | verdict       | date          |
|------|----------|---------------|---------------|
| 1    | ITP1_1_A | Wrong Answer  | 1403864355640 |
| 1    | ITP1_1_A | Compile Error | 1403864348261 |
| 1    | ITP1_1_A | Accepted      | 1403864349952 |
| 2    | ITP1_1_A | Accepted      | 1403864369443 |

| user | problem  | verdict |
|------|----------|---------|
| 1    | ITP1_1_A | 1       |
| 2    | ITP1_1_A | 1       |

A common method to handle implicit feedback is to use a binary system, which in the context of programming problems, simply records whether a user solved a problem (1) or did not (0).

This Approach lacks scalability, fails to capture satisfaction levels, and overlooks problem difficulty and user skill.

## Proposed Architecture



The core concept is to analyze the data submitted by users to derive a meaningful satisfaction score.

## Approaches

### 1. Direct Weighted Sum

$$D = W_1 \cdot \text{AVG\_SR} + W_2 \cdot \text{AVG\_DT} + W_3 \cdot \text{SPS}$$
$$PU = \frac{1}{NA} \times \frac{DT}{DT}$$

### 2. Formula-Based

$$D = W_1 \cdot \log(1 + \frac{1}{\text{AVG\_SR}}) + W_2 \cdot \text{AVG\_DT} + W_3 \cdot \frac{1}{\log(1 + \text{SPS})}$$
$$PU = \frac{1}{\log(NA+1)} \times e^{-\frac{DT}{DT}}$$

### 3. Distance-Based

$$\text{AVG\_SR}(i) = \max(\text{AVG\_SR}(P))$$
$$\text{SP}(i) = \max(\text{SP}(P))$$
$$\text{AVG\_DT}(i) = \min(\text{AVG\_DT}(P))$$

$$\text{Diff}_H(X, Y) = \frac{M(X) - M(Y)}{M(Y)}$$
$$D = W_H \sqrt{\sum_H (\text{Diff}_H(P_i))^2}$$

$$\text{Dist}(U_i) = \sqrt{\text{Diff}_{DT}^2 + \text{Diff}_{NA}^2}$$

$$PU = \text{Dist}(U) \times (1 - \frac{ND}{TD})$$

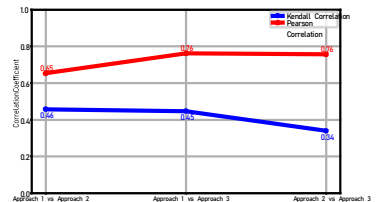
$$\text{Satisfaction} = \frac{S^D}{2} \times v + 2.5 \times \min(PU \times \frac{D}{SU}, 1)$$

$$SU = \frac{1}{k} \sum_{i=1}^k (D_i)$$

We developed three distinct approaches, each of which calculates problem difficulty and user performance. These values are then integrated into a common formula to determine user satisfaction

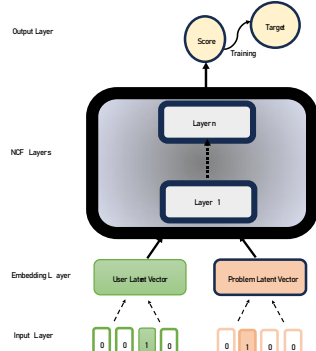
## Generated Satisfaction

| user | Problem  | Rating |
|------|----------|--------|
| 1    | ITP1_1_A | 2.5    |
| 2    | ITP1_1_A | 4      |



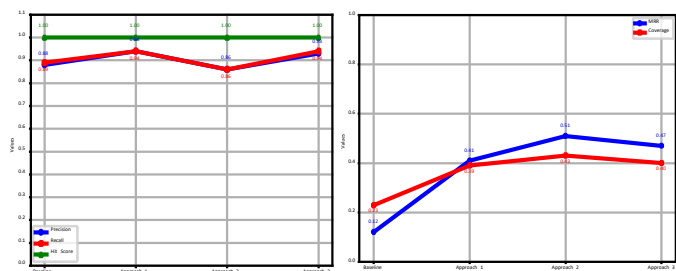
Each approach generates a User-Item Satisfaction matrix, resulting in a total of three different matrices that share some similarities.

## Neural CF



Neural CF is applied to each matrix, and we also included a baseline matrix using a binary system."

## Result



All the approaches performed well in terms of accuracy. However, the proposed approaches significantly outperformed the baseline approach in recommendation quality by presenting relevant items to users earlier and offering a broader range of items.