

# FoodIDLike: Reflections and Learnings

## Overview

The Programming Project #3 in my comparative languages class involved the implementation of a backend for a mobile application called "FoodIdLike" (FL). The aim of the project was to suggest the best restaurant to a user based on their prior dining experiences and restaurant ratings. This task offered an array of learnings regarding various technologies and methodologies.

## Data Handling

Working with comma-separated values (CSV) files was an enlightening experience. The project demanded skillful manipulation of these data sets to extract necessary user information, restaurant details, and ratings. This exercise provided hands-on experience in data management, particularly in processing and handling large CSV files.

## Feature Vector Creation

Creating a feature vector for each restaurant, where each entry represented a user's rating for the restaurant, was a key learning point. It provided insights into the importance of structuring data in a way that allows easy and efficient extraction of meaningful information.

## Collaborative Filtering

This project marked my first foray into the use of collaborative filtering (CF). Implementing CF as a heuristic for the backend was a valuable lesson in using user behavior and preferences to provide personalized recommendations. The mathematical computations involved in determining the ratings of all restaurants in the vicinity was a great practical experience in the utilization of CF algorithms.

## Usage of External Libraries

Learning to incorporate pre-existing functions from external libraries, like the `computeCoefficientsAndScores` function from `computeAll.jar`, helped me appreciate the value of reusability in software development. Understanding how to transform the input data to match the function's parameters was a good exercise in data transformation and adaptation.

## Scala Programming

Implementing the solution in Scala, with a focus on strict functional programming, significantly enhanced my Scala skills. The avoidance of mutable state, loops, and extensive use of if statements brought out the beauty of functional programming and its efficacy in solving complex problems.

## Challenges

Working on this project wasn't without challenges.

**Data Transformation:** Modifying input data to fit the parameters of `computeCoefficientsAndScores` function was initially challenging due to the difference in data structures. It required careful consideration and understanding of the data at hand.

**Collaborative Filtering Implementation:** Implementing the collaborative filtering algorithm presented some difficulties, particularly in understanding the mathematical computations involved in predicting restaurant ratings.

**Scalability:** Processing large data sets for multiple users and restaurants was a challenge in terms of efficiency and execution time. It required optimization strategies to handle the data effectively.

**Functional Programming Paradigm:** Given that Scala is a multi-paradigm language, refraining from using any mutable state or loops required a significant paradigm shift in the programming approach.

## Conclusion

Despite the challenges, the project was a great learning experience. It fostered an understanding of data handling, feature vector creation, collaborative filtering, and functional programming in Scala. The project also served as a useful exploration of software development for personalized user recommendations, which has wide applications in the current digital era. The hands-on experience gained from this project will be beneficial for future development roles and projects.