Problem 7: Course Recommendations (25 points) [File Upload]

By Devan Shah

Problem Statement

Given a COSCON University student's schedule from Fall 2024 and Spring 2024, can you determine what their (prospective) schedule is for Spring 2025 — and how accurate can you be?

	Spring 2025	Fall 2024	Spring 2024
0	APC3471, AST2949, ORF3140, COS3230, ...	COS2318, ...	COS2391, ...
1	MAE2151, ECO3810, CLA2511, COS2042, ...	HIS2413, ...	WRI1641, ...
2	SPI3941, ITA3241, GER3205, POL2400, ...	SML2257, ...	SAS3609, ...
3	MOL3424, MOL4459, POL3355, ECE3565, ...	ECO3245, ...	CBE4363, ...
4	CHM3730, ORF2179, EGR2082, MAT4949, ...	NES2968, ...	MOL3084, ...

Files and the example strategy are available here:
https://drive.google.com/drive/folders/1fS-B2txL-jP5mwJx7tObWPaalCsH_CRl

Background: Recommendation Systems

Recommendation systems are a subfield of machine learning focused on matching problems (i.e. matching users with videos, consumers to products, and students to courses). In industry use cases, the amount of content is massive — Amazon has on the order of $O(10^9)$ products, and so to handle the volume, recommendation systems are typically decomposed into two parts. When a user searches Amazon, Amazon quickly constructs a candidate set of around $O(10^5)$ products, leveraging vector search and the results of similar queries to avoid searching over all $O(10^9)$ products. In the next stage, Amazon provides a score for each candidate product and its relation to the user and their query. This score is used for selecting the final set of products that appear and ordering them, typically involving a large deep learning model.

In this challenge, we focus on the second part — selecting from the candidate set. Given a candidate set of 10 schedules, which one is the user most likely to enjoy (and so which one should we recommend). The synthetic dataset for this problem is based on real trends, but the courses and course numbers are fake.

The Problem Setup

For this problem, there are four types of data files. First we have **feature** table (i.e., `Demo_Data.csv, Eval_Data.csv`), where each row is a unique student's past schedule and the indices of 10 potential Spring 2025 schedules. One of these indices corresponds to the students actual plan for Spring 2025, with the rest being plans from other students.

	Fall 2024	Spring 2024	Index 1	Index 2	Index 3	Index 4	Index 5	Index 6	...
0	PHY1623, ...	MAT2841, ...	164	120	283	127	164	369	...
1	MPP2914, ...	ATL4152, ...	49	392	83	164	68	410	...
2	HIS4728, ...	CLA2677, ...	76	98	331	75	90	81	...

We next have the **courses table** (i.e., `Demo_Courses.csv, Eval_Courses.csv`), which associates indices in the **feature** table to schedules, and we have the **associations** table (i.e., `Demo_Ans.csv`), which gives the correct index per row. Lastly, we will give you an **example** table (i.e. `Training_Data.csv`), which consists of some examples of complete schedule data.

You will be given the **feature** and **course** table for the demo and evaluation code, and you will only be given the **associations** table for the demo data. Your goal is to give us a predicted **associations** table for the evaluation data. *If this sounds like a lot, it's much simpler when you check out the data!*

Starting Code and Scoring

To help you get started, we've implemented a strategy for random guessing that showcases useful template code. This problem is evaluated based on the accuracy of your predicted **associations** table for the evaluation data `eval_data.csv`. With accuracy `p`, you will receive points according to the scoring function: $f(p) := \lceil \min(45\mathrm{p}^2, 25) \rceil$.

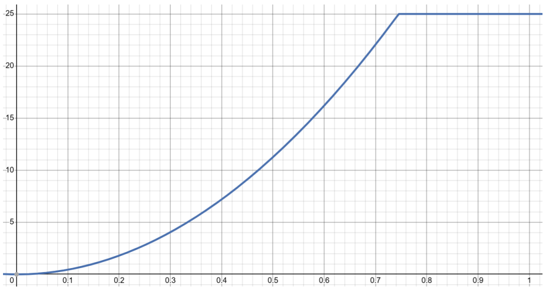

Figure 1: Graph of $\min(45\mathrm{p}^2, 25)$

How to Submit:

Your submission should be a `csv file` with only one column titled `schedule.csv` and 281 rows, one for each row in the evaluation data. We will also ask that you submit your code. We will then evaluate your accuracy based on the submitted csv and determine your points based on the formula described earlier:

Index	Schedule
0	369
1	392
2	345
3	194
4	197
5	296
⋮	⋮

Tips: Check out the example code!

P.S. if you enjoy this problem, this is quite similar to what students at Hoagie.io are working on.