**Pathfinder**

Personalized Career Path Recommender System for University Students

Members:

Leshan Chathuranga Kuruppuarachchi – 300363077 (Team Lead)

Hyunhee Kim - 300372531

Course: CSIS-4495 - Applied Research Project

Section: 050

**Table of Contents**

[**1.** **Introduction** 3](#_Toc212132850)

[**2.** **Proposed Research Project** 5](#_Toc212132851)

[**2.1** **Methodology** 5](#_Toc212132852)

[**2.2** **Technologies to be used** 5](#_Toc212132853)

[**2.3** **Expected Results** 6](#_Toc212132854)

[**3.** **Project Planning and Timeline** 7](#_Toc212132855)

[**2.4** **Responsibilities** 8](#_Toc212132856)

[**2.5** **Project Timeline Gantt Chart** 8](#_Toc212132857)

[**4.** **Implemented Feature: Web scraping job data and store in database** 9](#_Toc212132858)

[**5.** **Implemented Feature: Data Modeling and Embedding-Based Recommendation work flow** 11](#_Toc212132859)

[**6.** **References** 16](#_Toc212132860)

[**Appendix A: Data normalization and key generation** 17](#_Toc212132861)

[**Appendix B: Embedding Training (TF-IDF, Optional LSA/SVD)** 18](#_Toc212132862)

[**Appendix C: Pathfinder Career Recommender Interface** 19](#_Toc212132863)

# **Introduction**

In today’s competitive job market, university students face increasing challenges when choosing their future career path. The traditional methods of career finding, such as personality tests, career questionnaires, or career advisers often fail to understand individual skills and interests, because they are too broad and fail to understand individual differences in skills, academic background and personality.

These limitations have led to interest in the application of intelligent systems in the field of career counseling and education technology.

To overcome these limitations, personalized career recommendation system has emerged as a solution. These systems collect user-provided information such as gender, academic background, skills, interests, and analyze the data to generate personalized recommendations. Based on this context, this research addresses two key questions:

1. How can machine learning models improve the accuracy and related to these recommendations compared to traditional methods?
2. How can student-provided information be used effectively to generate personalized career recommendations?

These questions are important because wrong career choices can lead to poor experience for students and labor market needs. By developing a data-driven and personalized recommendation system, this research aims to help students make confident decisions about their future careers.

Recent career-recommendation studies range from classical ML pipelines to modern hybrid ensembles. Classical content-based pipelines [1] preprocess user-provided descriptions, vectorize with TF-IDF, segment users via K-Means, reduce dimensions with PCA, and rank careers using cosine similarity; a full-stack reference implementation integrates these components into a web app (e.g., Django). Hybrid stacking ensembles [2], recent work for competency-based education creates a structured student dataset (5 features, 5,000 records) and combines Deep Neural Networks and Random Forest via stacking to recommend STEM tracks, reporting ~90.06% accuracy and 92.07% precision with 5-fold CV. Mobile applications such as CareerX [3] have demonstrated the integration of NLP and machine learning into mobile applications, with user satisfaction rates of around 90%. Other studies have highlighted Hybrid filtering approaches [4], combining collaborative and content-based methods to generate personalized educational roadmaps. Hierarchical multi-tiered systems [5] have also been developed to address sparsity and cold-start problems, achieving over 99% accuracy across diverse career datasets.

Despite these advances, several gaps remain. Classical pipelines often rely on limited or generic features, while hybrid models tend to be tailored to narrow contexts such as STEM education. Feedback mechanisms are rarely integrated in near real-time, leaving evaluations dependent on static, offline metrics. Furthermore, most systems lack generalizability, as results validated in one population may not transfer well to others without adaptation.

Based on these gaps, this research assumes that a machine learning based recommendation system using effectively diverse student-provided information will significantly enhance personalization and accuracy. The expected benefit is to provide students with reliable, data-driven guidance for career choices, while contributing to the development of intelligent systems in education technology.

# **Proposed Research Project**

The proposed Pathfinder project is a data-driven experimental design. The main objective is to build a personalized career recommendation system that uses user provided information, such as educational background, gender, skills and interests, to recommend personalized career pathways.

## **Methodology**

Pathfinder follows a four-step methodology, data collection and preprocessing, model training and recommendation engine design, skill gap analysis and job market trend visualization.

First, training data will be collected from public sources such as Kaggle and through web scraping platforms like LinkedIn and Indeed. Operational data will consist of information provided directly by users via Pathfinder website, including major, GPA, gender, interest, skill, and personality traits. Next, the collected data will be used to train models capable of recommending career paths and personality traits to the suggested career path based on similarity measures such as cosine similarity, KNN, or other machine learning classifiers.

Following recommendation generation, a skill gap analysis will be conducted by comparing the requirements of each recommended career with the user’s current skills. To complement this, job market datasets will be analyzed to highlight trends and demand for recommended careers over the last several years. Finally, the system will be implemented as a functional platform that integrates all these components.

## **Technologies to be used**

This project will integrate several technologies

* Operating System: The system will run on both Windows and mac environments.
* Programming Language/Frameworks: Python will be the primary language, using machine learning libraries such as scikit-learn, TensorFlow, and PyTorch.
* Database: MySQL or H2 database will be used to store cleaned data for model and visualization data since those databases are open source and easy to config.
* Frontend: React or Vue.js will be used to build a responsive and interactive user interface that allows users to input their profiles and view career recommendations, skill gap analysis, and job market trends.
* Backend: FastAPI or Flask will serve as the backend to manage model inference, skill gap analysis, and job trend visualization.
* Visualization Tools: Libraries such as Chart.js, D3.js, or Recharts will be used to display skill gaps and job market trends on the dashboard.
* Collaborative Development: GitHub will be used for version control, source code management, and team collaboration. Team members will work together using branches, pull requests and reviews during development.

## **Expected Results**

The expected outcomes of this research are multifaceted. First, the system is anticipated to provide improved accuracy and personalization compared to traditional career counseling methods. Each user will receive the top three to five most suitable career recommendations, offering clear and actionable pathways. Alongside these recommendations, users will also receive a breakdown of the skills they lack, the importance of these skills, and the estimated effort required to bridge the gap.

Additionally, users will gain insights into job market trends, with visualizations of labor market data over the last five years to highlight demand in various careers. The platform will be designed as a user-friendly and interactive tool that encourages students and job seekers to explore potential career paths and refine their skills continuously

Ultimately, this project will contribute to the advancement of educational technology by developing an intelligent system that supports students in making better-informed career decisions while also aligning educational outcomes with labor market needs.

# **Project Planning and Timeline**

This section presents the planned schedule for completing the personalized Career Path Recommender System. The project is divided into clear phases, each with specific milestones and deliverables, to ensure steady progress and timely completion. The timeline follows the official course deadlines for the proposal, progress reports, midterm and final submissions.

Table 1: Proposed project schedule

|  |  |  |  |
| --- | --- | --- | --- |
| Phase | Duration/  Deadline | Milestones | Deliverables |
| Phase 1:  Literature Review & Project Initialization | 2025/09/04 – 2025/09/11 | Collect references on career recommendation systems | Project Proposal |
| Phase 2:  Data Collection & Preprocessing | 2025/09/08 – 2025/09/25 | * Identify data sources * Implement scraping & preprocessing | Cleaned dataset |
| Phase 3:  System Design | 2025/09/12 – 2025/09/25 | * Define system architecture (frontend, backend) * Design recommendation logic | * System architecture diagram * Progress Report 1 |
| Phase 4:  Recommendation Model Development | 2025/09/17 – 2025/10/16 | * Implement recommendation model * Train/test with sample student data | * Working recommendation engine * Progress Report 2 |
| Phase 5:  Frontend & User Interaction | 2025/10/08 – 2025/11/05 | * Build React UI for student input * Connect frontend to backend API | * Functional prototype * Midterm Report * Progress Report 3 |
| Phase 6:  Testing & Refinement | 2025/11/03 – 2025/11/18 | * Test system with multiple profiles * Debug & bug fixing | * Progress Report 4 * Midterm Video Report |
| Phase 7:  Documentation & Final Report | 2025/11/12 – 2025/11/27 | * Prepare final report and presentation | * Progress Report 5 * Final Report & Implementation * Project Defense |

## **Responsibilities**

Leshan Chathuranga Kuruppuarachchi

* Overall project management and coordination
* Data scraping design and implementation
* Model design and implementation
* Frontend development
* Testing and bug fixing

Hyunhee Kim

* Data collection and preprocessing
* Model design and implementation
* Backend API development
* Data visualization implementation
* Testing and bug fixing
* Documentation

## **Project Timeline Gantt Chart**

Figure 1: Proposed project Gantt chart

# **Implemented Feature: Web scraping job data and store in database**

As part of the Pathfinder project, the first major feature implemented was a web scraping component designed to extract job information from the job search websites. Apart from the Kaggle datasets this feature is an important part of the data collection process for the Pathfinder career recommendation system.

In the original proposal, the database was planned to be MySQL or H2. However, due to library dependency issues and setup difficulties, We decided to use SQLite instead. SQLite was easier to configure, lightweight, and worked well with Python. This change allowed faster development and testing without affecting the functionality of the project.

A screenshot of a computer code

AI-generated content may be incorrect.

Figure 2: code snap of web scraper

The main objective of this feature is collecting real job posting data such as job title, company name, location, and description and store the collected data in a structured database to prepare data that can later be used for recommendation and skill analysis.

We used selenium, webdriver\_manager libraries and implemented the scraper to handles cookie consent pop-ups, random delays to mimic human browsing, and automatically save the extracted data into the database.

We assume the indeed website maintains a consistent structure for job postings and use ethical scraping principles respecting delays and not overloading the site. We tried different scraping methods using Selenium and BeautifulSoup and selected Selenium for its ability to handle dynamic JavaScript-rendered content.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 3: snapshot of console output

# **Implemented Feature: Data Modeling and Embedding-Based Recommendation work flow**

This section presents the implemented data modeling and recommendation workflow for the Career Path Recommendation system. The System goal is suggestion suitable job positions and relevant online courses based on the user’s background, including major, GPA, skills, and interests. The figure below illustrates the modeling architecture and workflow, which includes six major components:

* 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

  AI 생성 콘텐츠는 정확하지 않을 수 있습니다.Data Preprocessing: cleaning and normalizing job/course data.

Figure 4: data modeling diagram

* Feature Engineering: constructing model, ready text fields.
* TF-IDF Vectorization: converting text to numerical features.
* Skill Encoding (Jaccard Similarity): comparing explicit skill sets
* LSA (Dimensionality Reduction): improving efficiency and semantics
* Content-Based Recommendation and Scoring: combining ranking jobs and courses based on similarity and GPA

This workflow establishes the foundation of the model’s ability to represent, compare, and rank career-related information effectively.

**Data Modeling Architecture Explanation**

1. Data Preprocessing

In the first stage, raw job and course datasets were cleaned and standardized.

This step included that text normalization (lowercasing, symbol removal, white space trimming), tokenization and cleaning of skills and organization names, conversion of ratings and duration into numerical values. The main objective of this step was to ensure data consistency and prevent noise from influencing similarity computations.

1. Feature Engineering

Feature engineering integrates and prepares text-based and numerical features for model training. New derived fields were generated, such as:

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Figure 5: screenshot of Feature names and description

This step ensures that the system has structured and comparable representations of both job and course descriptions.

1. TF-IDF Vectorizer

The TF-IDF (Term Frequency-Inverse Document Frequency) method transforms textual descriptions into numerical vectors. This allows the system to measure how important each term is within the collection of text. Each job\_text and course\_text entry is vectorized to quantify their contextual similarity. TF-IDF is used as the core text representation technique due to its simplicity, interpretability, and efficiency.

1. Skill Encoding (Jaccard Similarity)

Beyond text features, the system also compares explicit skill set between the user and jobs. The Jaccard Similarity metric measures the overlap between two sets (user skill vs job-required skills). This additional layer allows the model to reflect actual technical alignment, rather than relying solely on textual similarity.

1. LSA (Dimensionality Reduction)

To enhance computational efficiency and semantic accuracy, Latent Semantic Analysis (LSA) is applied on top of TF-IDF vectors using Truncated SVD. This technique reduces high-dimensional sparse vectors into a lower-dimensional semantic space, helping to capture hidden relationships among keywords such as ‘data analysis’, ‘machine learning’ and ‘AI’.

1. Content-Based Recommendation & Scoring

The final recommendation score combines multiple metrics:

* Cosine Similarity (between user profile and job vector)
* Jaccard Similarity (between user and job skills)
* GPA weighting (slight adjustment based on academic performance)

Each job receives a match\_score (0-1 range). For each job, the model also identifies missing\_skills and recommends top matched courses to fill those gaps, leveraging cosine similarity between skill queries and course embeddings.

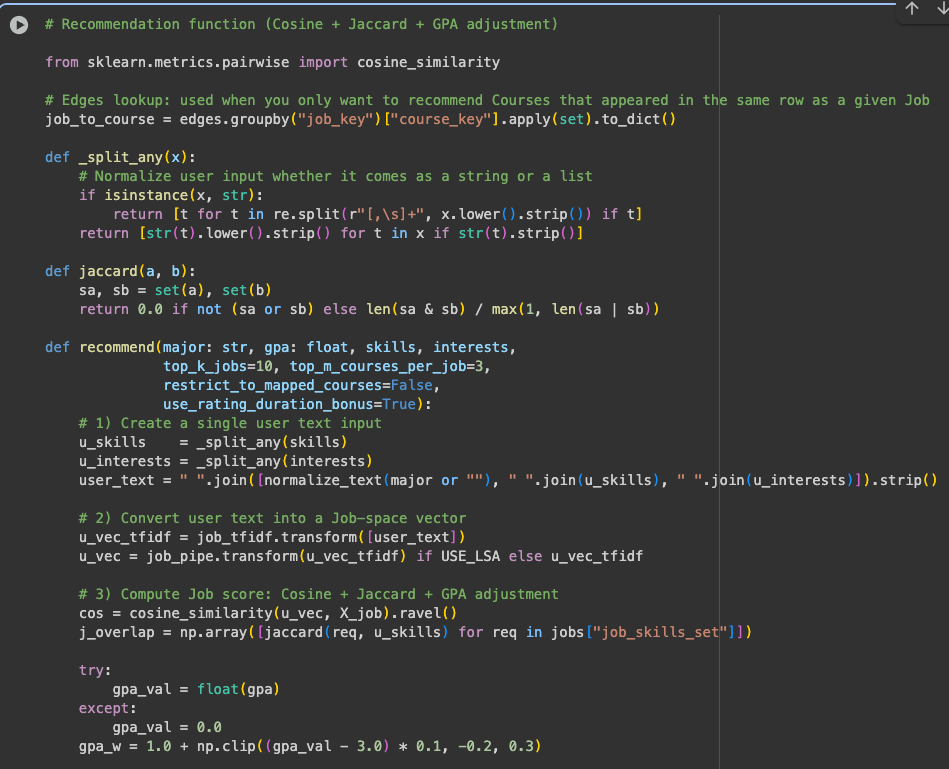
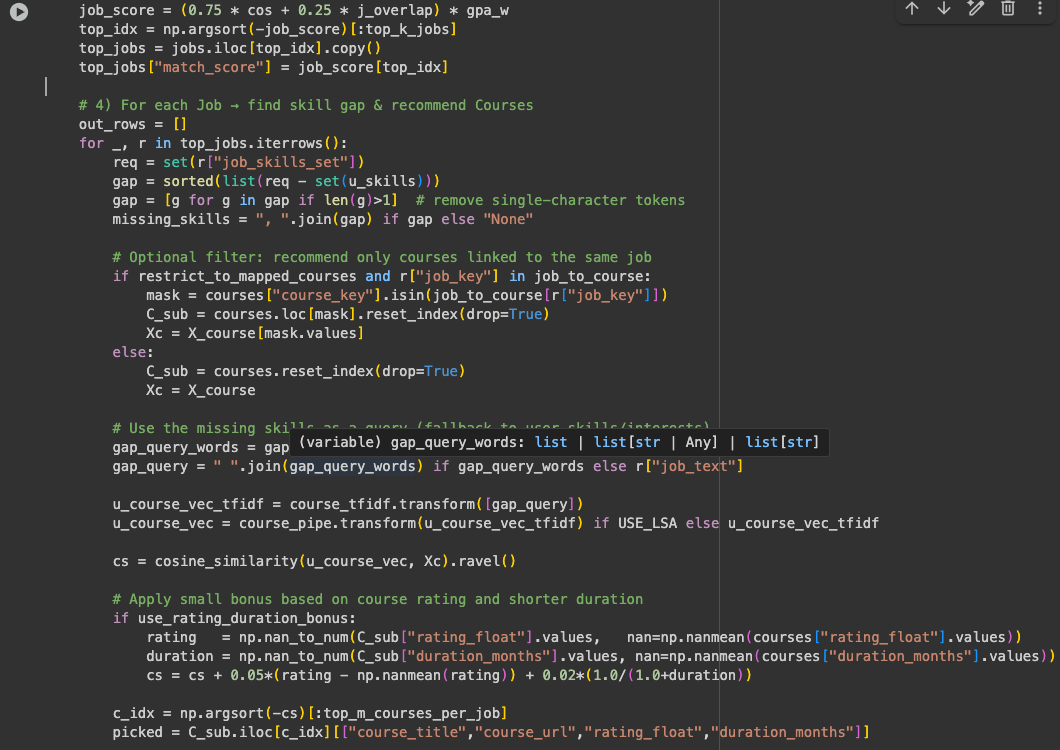


Figure 6: code screenshot – Recommendation Function

The main logic integrates cosine similarity, Jaccard similarity, and GPA adjustment into a unified recommendation model. The function computes how closely a user’s academic profile aligns with job descriptions then recommends courses the bridge skill gaps

The scoring process integrates the above similarity measures to generate ranked outputs. Each recommendation includes:

* Job title, company name, and link
* Match Score = (0.75 \* Cosine Similarity) + (0.25\* Jaccard Similarity) \* GPA weight
* Missing skills (gap between required and user-provided skills)
* Top 3 course recommendations, including course title, rating, and estimated duration

The hybrid formula balances text-based similarity, skill overlap, and academic performance. It ensures that the model prioritizes jobs that are both semantically relevant and technically achievable for the user.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Figure 7: test input

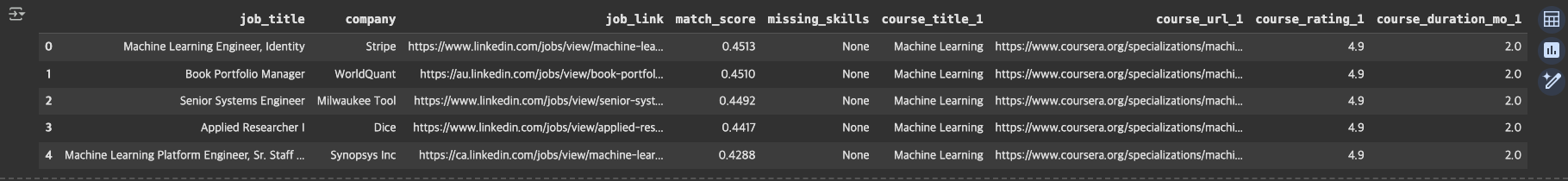


Figure 8: output preview

Each recommended job is ranked by similarity, and the missing skills column identifies what the user lacks. The corresponding courses are chosen to help bridge those skill gaps.

The system performs effectively in generating meaningful recommendations even with limited data. Match scores range between 0.42-0.50, which indicates moderate to high alignment between user profiles and job roles.

While cosine similarity remains the dominant metric, Jaccard and GPA adjustments make the results more contest aware. Future improvements will include integrating FastAPI for real-time user interaction, expanding the dataset with user personality.

# **References**

|  |  |
| --- | --- |
| [1] | N Deepak, M Gowtham, J Mithilesh and A Naresh Kumar, "Personalized Career Recommendation System Based on Customer Segmentation," *Journal of Emerging Technologies and Innovative Research,* 2024. |
| [2] | F. Kainyu, M. Mwadulo and S. Munialo , "A Hybrid Recommender Model for Career Pathway Selection in Competency-based Education," *International Journal of Computer Applications,* 2025. |
| [3] | Y. Gupt, "CareerX: AI-Powered Career Path Recommender System for College Students," *ResearchGate,* 2025. |
| [4] | H. Kumar, U. P. Pandey and P. Kumar, "AI-Based Career Path Recommendation System," [Online]. Available: https://amity.edu/UserFiles/aijem/293Harsh1.0%20(AJCS).pdf. |
| [5] | M. QAMHIEH, H. SAMMANEH and M. N. DEMAIDI, "PCRS: Personalized Career-Path Recommender," *IEEEAccess,* 2020. |

# **Appendix A: Data normalization and key generation**

This section shows the process of structuring the merged dataset into three normalized relational views. The goal is to ensure a unique and consistent reference structure before vectorization and modeling. Each job and course entry is assigned normalized key using text standardization (keyify), which removes spaces, coverts text to lowercase and concatenates multiple fields to avoid duplication bias.

After generating these keys, Jobs view contains one record per job posting, Courses view contains one record per course and Edges view captures job-course co-occurrences, rows where both appear together.

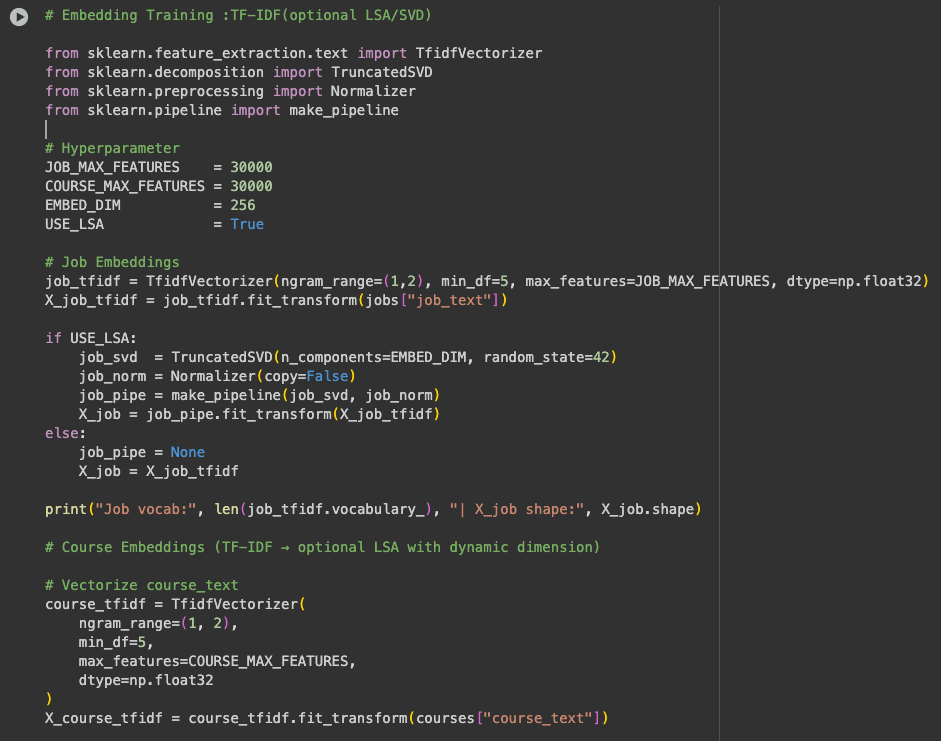
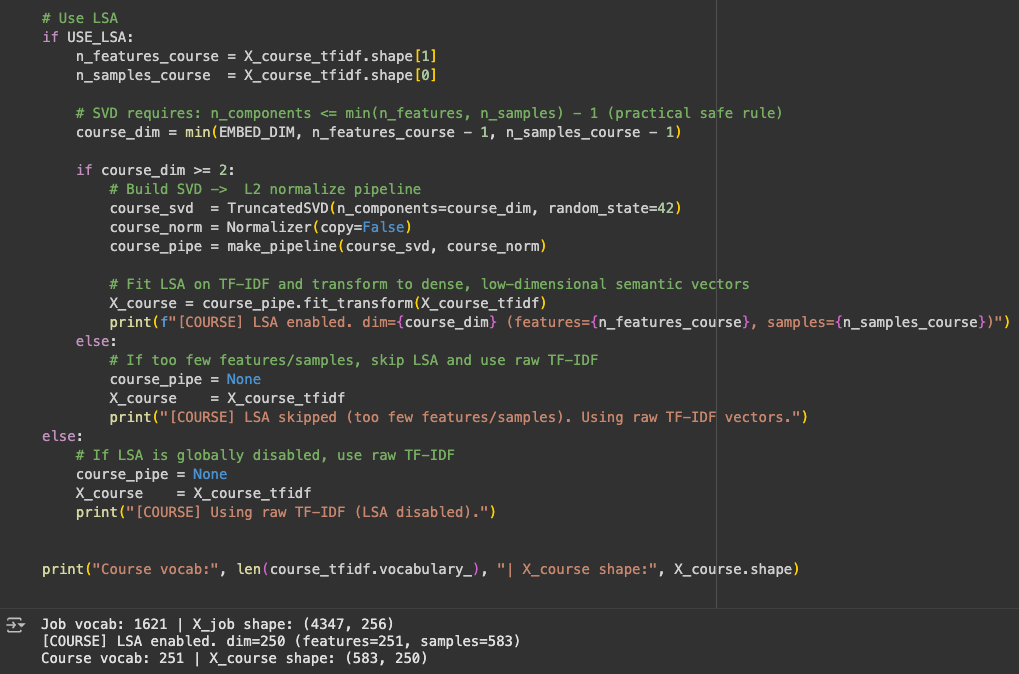
텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

Figure 9: Code snippet for creating normalized views/courses/edges.

# **Appendix B: Embedding Training (TF-IDF, Optional LSA/SVD)**

This section details the vectorization and embedding process that converts textual features (job\_text) into numerical vectors suitable for similarity computation. Two separate embedding pipelines are trained: first, Job embeddings are based on job titles, companies, and skills. Second, Course embeddings are based on course titles, organizations, and topics.



# **Appendix C: Pathfinder Career Recommender Interface**

The frontend interface for the Pathfinder Career Recommendation System was developed using React.js with Tailwind CSS for modern, responsive styling. The goal of this feature is to provide an interactive and user-friendly interface for students to input their academic background, interests, skills, and personality preferences, and receive personalized career suggestions.

A screenshot of a computer

AI-generated content may be incorrect.

Figure 10: snapshot of pathfinder frontend