300372531 Hyunhee Kim

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

1. **Overview**

**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:**

***Figure 1. Data Modeling diagram***

* 텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

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

1. System Architecture Explanation
2. 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:

*Figure 2. screenshot of Feature names and description*

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

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 3. code screenshot – Recommendation Function*

텍스트, 스크린샷, 소프트웨어, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

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

1. Scoring and Output

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

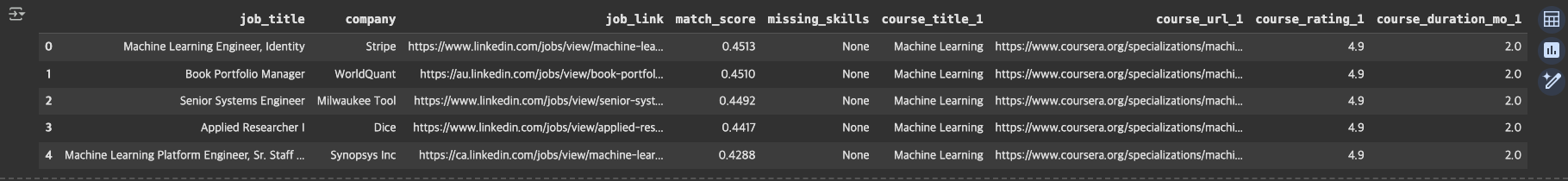
The hybrid formula balances text-based similarity, skill overlap, and academic performance. It ensures that the model prioritizes jobs that are both semantically relevant and technically achievable for the user.

*Figure 4. test input*

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

*Figure 5. 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 recommendation 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 that integrating FastAPI for real-time user interaction, expanding the dataset with user personality.

**Appendix A – Data normalization and key generation**

Code snippet for creating normalized views/courses/edges.

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.

**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.

텍스트, 스크린샷이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

These two screenshot should be stick together -> please remove this line when you’re editing