CROSS DOMAIN RECOMMENDATION SYSTEM FOR JOB/BUSINESS OPPORTUNITY DISCOVERY

SHAMEER SHEIK, MS IN AI & ML, LIVERPOOL JOHN MOORES UNIVERSITY

DATE: 15.FEB.2026

INTRODUCTION

In today's data-driven landscape, the exponential growth of digital information across domains such as products, services, user profiles, employment, education, and entrepreneurship presents both opportunities and challenges for intelligent decision support systems. Recommender systems have become essential tools for navigating this complexity, driven by the rise of personalised user experiences, data-centric platforms, and widespread internet access.

Traditional systems are not equipped to recognise or respond to such cross-domain transitions, resulting in fragmented user experiences and missed opportunities. They often lack the semantic depth and causal reasoning needed to connect learning outcomes with relevant employment or entrepreneurial pathways.
This limitation is particularly evident in the career and business opportunity space, where users frequently navigate multiple domains simultaneously.

To address this gap, the present study introduces a novel framework, Causal Graph-Guided Generative Alignment (CGGA), which leverages domain-informed causal graphs to align user profiles with opportunity spaces across multiple domains. The use case focuses on recommending job roles, freelance opportunities, or startup pathways based on a user's evolving skill profile, enabling context-aware and interpretable cross-domain recommendations that reflect real-world dependencies.

PROBLEM STATEMENT

Despite progress in recommender systems, most existing approaches remain limited to single-domain settings and rely on correlation-based methods that struggle with semantic alignment, cold-start scenarios, and contextual interpretability. These limitations become more pronounced in heterogeneous domains such as employment, freelancing, and entrepreneurship, where users often receive fragmented or weakly aligned recommendations that do not reflect their evolving skill profiles or career trajectories.

The core problem addressed in this study is the absence of a robust and interpretable framework for cross-domain opportunity recommendation. Specifically, there is a need for a system capable of modelling causal relationships between user attributes and opportunity spaces, generating semantically coherent representations, and optimising recommendations across multiple objectives such as relevance, alignment, and diversity.

To address this gap, the study proposes the Causal Graph-Guided Generative Alignment (CGGA) framework, which integrates domain-specific causal graph construction, variational generative modelling, adaptive few-shot learning, and multi-objective optimisation. The framework is designed to deliver context-aware, scalable, and transparent recommendations that better reflect real-world user complexity.

AIM & OBJECTIVE

AIM:

This research aims to design and evaluate a scalable cross-domain recommendation system for personalised career and business opportunity discovery. The system is built using the Causal Graph-Guided Generative Alignment (CGGA) framework, which integrates causal graph construction, variational generative modelling, adaptive few-shot learning, and multi-objective optimisation to generate context-aware and user-specific recommendations.

The study evaluates the framework's effectiveness in improving recommendation accuracy, interpretability, and diversity using publicly available datasets.

OBJECTIVES:

1. Design a Modular Cross-Domain Recommendation Architecture
- Develop a system that integrates structured data from multiple publicly available domains.
- Ensure semantic alignment across domains using embedding-based representations.
- Evaluation: Compare embedding similarity distributions across domains.

2. Construct Domain-Specific Causal Graphs
- Build causal graphs that encode relationships between skills, roles, and opportunity attributes.
- Validate graph structure using literature-supported dependencies.
- Evaluation: Qualitative validation against domain knowledge.

AIM & OBJECTIVE

3. Implement Generative Alignment for Opportunity Representation

- Use a Variational Autoencoder (VAE) to generate latent opportunity embeddings.
- Align generated embeddings with causal dependencies and user profiles.
- Evaluation: Assess embedding coherence using cosine similarity and reconstruction loss.

4. Integrate Adaptive Few-Shot Learning for Cold-Start Scenarios

- Apply few-shot learning to support users with limited historical data.
- Use synthetic profiles to simulate sparse-data conditions.
- Evaluation: Measure few-shot classification accuracy on held-out samples.

5. Apply Multi-Objective Optimization for Personalized Ranking

- Develop a ranking mechanism balancing relevance, causal alignment, and diversity.
- Use weighted scoring to personalise recommendations.
- Evaluation: nDCG@10 and diversity metrics on test recommendations.

6. Evaluate System Performance

- Conduct quantitative evaluation using standard metrics:
    - Precision, Recall, F1-score
    - nDCG, Coverage, Diversity
- Evaluation: Report overall performance across domains using publicly available datasets.

OVERVIEW

This study adopts a design-science methodological approach, integrating causal inference, generative modelling, and adaptive learning to develop a scalable and context-aware cross-domain recommendation system. The methodology is structured around the development and evaluation of the Causal Graph-Guided Generative Alignment (CGGA) framework, which is designed to address key limitations of traditional recommender systems, including domain isolation, data sparsity, and limited contextual relevance. The approach combines conceptual modelling with empirical experimentation, enabling systematic refinement of the framework based on observed performance and alignment with intended functional objectives.

Data Collection

Inputs:

- Synthetic profiles.
- Domain attributes.
- Kaggle dataset

Data Pre-processing

- Text Normalization
- Assigning IDs for each user profile, job profile.
- Removing unwanted columns.
- Removing the unwanted profiles which doesn't have the proper data

CGGA Framework

- Causal Graph Construction
- Variational Generative Modelling
- Adaptive Few-Shot Learning
- Multi-Objective Optimization

Recommendation

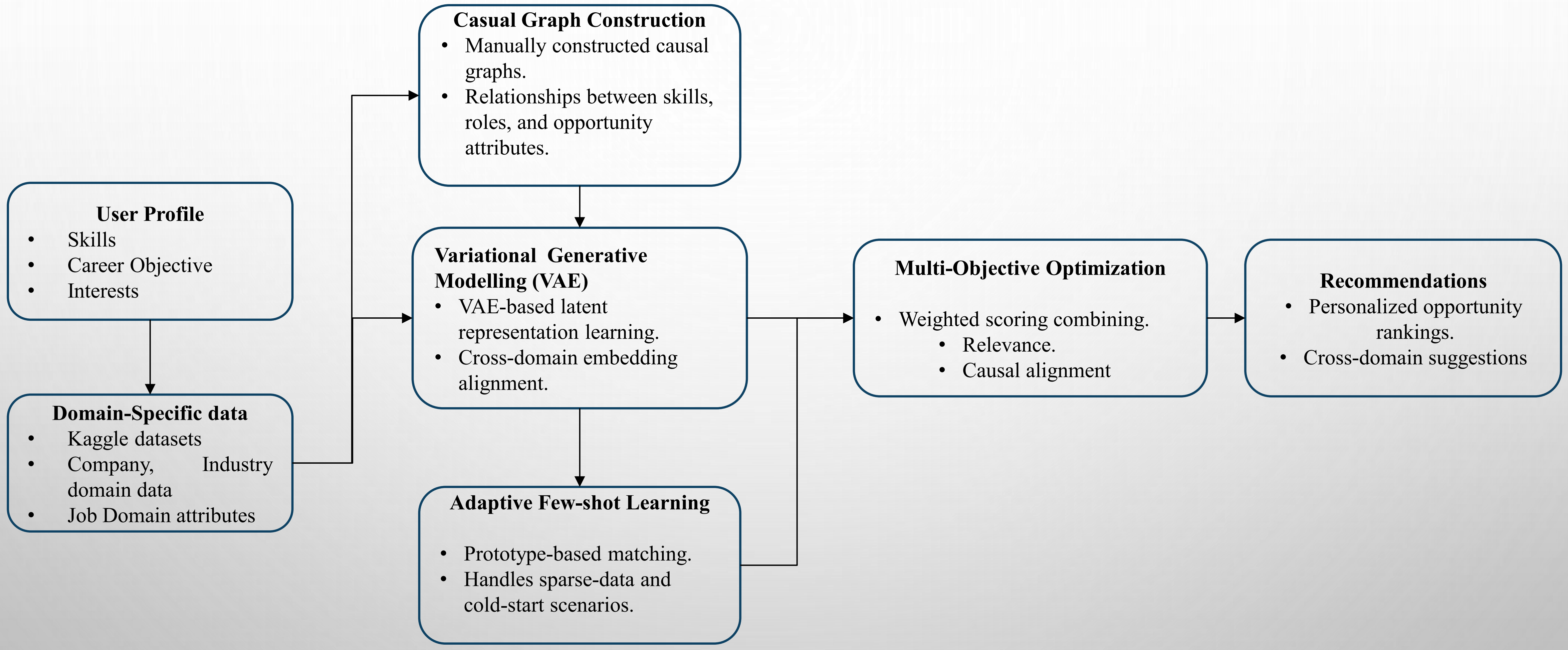
- User personalized
- Re-ranking using Skill Embeddings & career objectives.
- Preparing the ground truth reference to compare with the recommendations

Evaluation

Metrics:

Precision, Recall,
Hit-rate,
MRR,
nDCG, ILD,
Coverage,

METHODOLOGY



DATASETS

User Profile data: resume_data.csv
Source: Kaggle
URL: https://www.kaggle.com/datasets/saugataroyarghya/resume-dataset
This dataset contains detailed resume information and serves as the source of user-level attributes. Key fields include career objectives, skills, educational background, degree information, work experience, languages, certifications, and extracurricular activities. These attributes form the basis for constructing user representations and causal graph relationships.

Job Posting data: 1.3M Linkedin Jobs & Skills (2024),
Source: Kaggle
URL: https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024

This dataset provides large-scale job opportunity information across multiple industries. It consists of three linked files:
linkedin_job_postings.csv containing job titles, companies, locations, job levels, and job types
job_skills.csv listing required skills for each job
job_summary.csv providing textual job descriptions
These files collectively form the opportunity space used for representation learning, embedding generation, and recommendation evaluation.

Industry-Company Mapping file: companies_sorted.csv
Source: Kaggle
URL: https://www.kaggle.com/datasets/peopledatalabssf/free-7-million-company-dataset
This dataset includes company names, domains, industries, size ranges, locations, and employee estimates. It is used to enrich job postings with industry-level metadata and support domain-specific causal graph construction.

TOOLS

Sr. No	Tool / Software / Library	Purpose
1	Visual Studio Code (VS Code)	Used as the primary development environment for writing, organising, and executing Python scripts. VS Code provided integrated debugging, environment management, and seamless execution of Jupyter notebooks through its built-in kernel support.
2	Python 3.11.9 Interpreter	The core programming language for all components of the system, including data preprocessing, causal graph construction, model training, embedding generation, and evaluation. Python 3.11.9 ensured compatibility with modern libraries and improved runtime efficiency.
3	Jupyter Kernel within VS Code	Enabled interactive experimentation, iterative model development, and step-wise execution of the CGGA pipeline. This setup supported rapid prototyping and visual inspection of intermediate outputs.
4	Python Data Processing Libraries	Pandas: Data loading, cleaning, transformation, and tabular manipulation NumPy: Numerical operations and vectorised computations Scikit-learn: Preprocessing utilities, similarity computation, and evaluation metrics JSON, OS, Pathlib: File handling and configuration management
5	Python Data Processing Libraries	PyTorch: Implementation of the Variational Autoencoder (VAE), training loops, and embedding generation TensorFlow: Used for experimentation and validation of alternative model configurations.
6	Deep Learning & Generative Modelling	Sentence Transformers: Used to generate semantic embeddings for text-based attributes such as job descriptions, skills, and user interests. These embeddings supported cross-domain alignment and improved the quality of latent representations.
7	Representation Learning	NetworkX: Construction, manipulation, and visualisation of causal graph structures used in the CGGA framework.
8	Graph Processing	Matplotlib / Seaborn: Used to visualise data distributions, evaluation metrics, and analysis outputs.

DESIGN STEPS

STAGE 0: DATA PRE-PROCESSING

STAGE 1: LOAD & PREPROCESS USER + JOB DATASETS

STAGE 2: EMBEDDING GENERATION (USER + JOB)

STAGE 3: JOB CLUSTERING

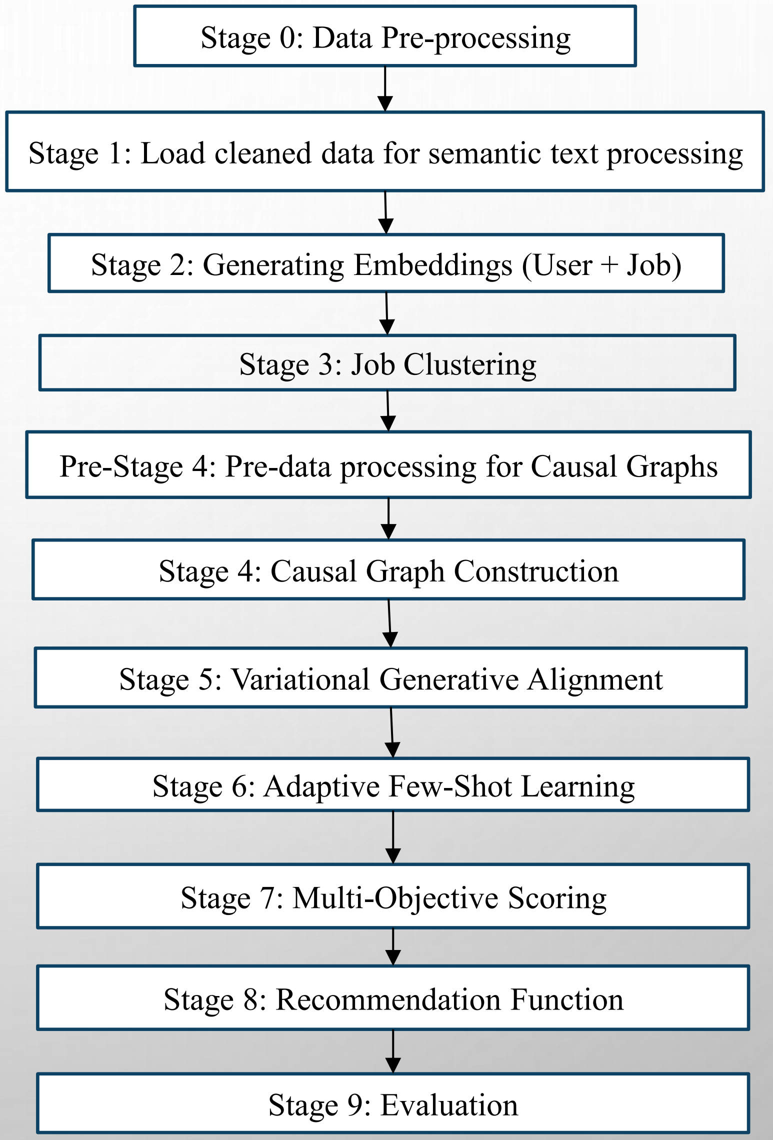
STAGE 4: CAUSAL GRAPH CONSTRUCTION

STAGE 5: VARIATIONAL GENERATIVE ALIGNMENT (VAE / CVAE)

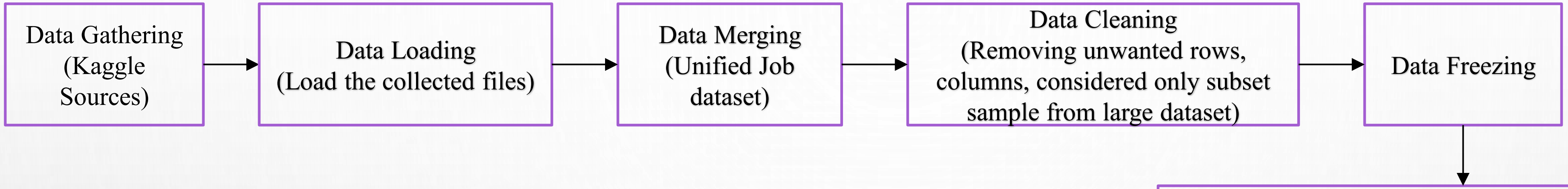
STAGE 6: ADAPTIVE FEW-SHOT LEARNING

STAGE 7: MULTI-OBJECTIVE SCORING + RECOMMENDATION ENGINE

STAGE 8: EVALUALTION



STAGE 0: DATA PRE-PROCESSING



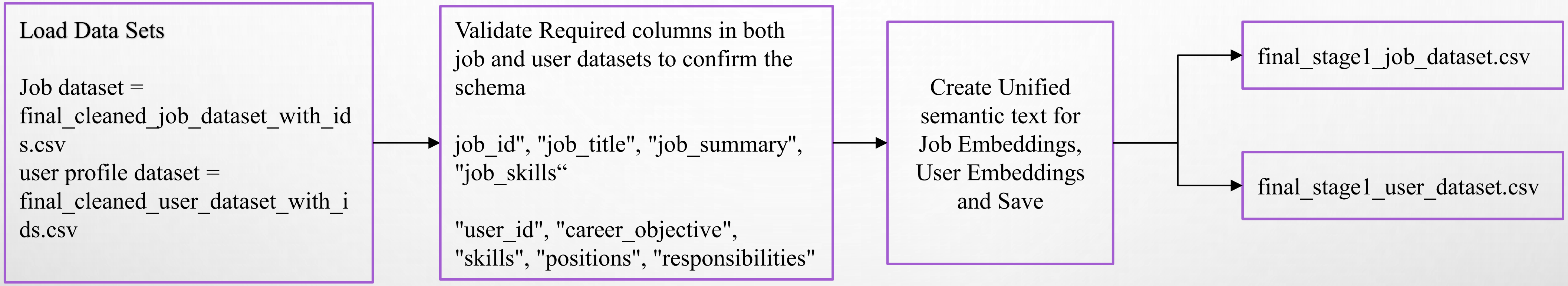
Final Cleaned Data Sets

Job dataset = final_cleaned_job_dataset_with_ids.csv
user profile dataset =
final_cleaned_user_dataset_with_ids.csv

User Profile data: resume_data.csv
Source: Kaggle
URL: https://www.kaggle.com/datasets/saugataroyarghya/resume-dataset
This dataset contains detailed resume information and serves as the source of user-level attributes. Key fields include career objectives, skills, educational background, degree information, work experience, languages, certifications, and extracurricular activities.

Job Posting data: 1.3M Linkedin Jobs & Skills (2024),
Source: Kaggle
URL: https://www.kaggle.com/datasets/asaniczka/1-3m-linkedin-jobs-and-skills-2024
This dataset consists of three linked files:
linkedin_job_postings.csv containing job titles, companies, locations, job levels, and job types
job_skills.csv listing required skills for each job
job_summary.csv providing textual job descriptions

STAGE 1: LOAD & PREPROCESS USER + JOB DATASETS



STAGE 2: GENERATING EMBEDDINGS (USER + JOBS)

Load Stage 1 Data Sets

Job dataset = final_stage1_job_dataset.csv
user profile dataset = final_stage1_user_dataset.csv

→

Data Validation Check (No Empty text rows in user_text, job_text)

→

Load Sentence transformer MODEL_NAME = "sentence-transformers/all-MiniLM-L6-v2"

→

Job Embeddings

- Embeddings/job_embeddings.npy
- Embeddings/job_ids.csv

User Embeddings

- Embeddings/user_embeddings.npy
- Embeddings/user_ids.csv

STAGE 3: JOB CLUSTERING

Load Stage 2 Embeddings

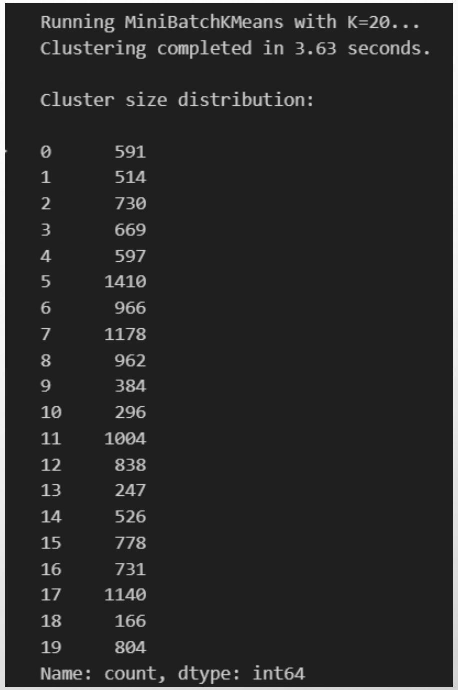
- job_embeddings.npy
- job_ids.csv
- user_embeddings.npy
- user_ids.csv

Applied **MiniBatchKMeans** to group entire job space into 20 semantically meaningful groups based on similarity in the embedding space.

K = 20 clusters,

Job Clustering

job_clusters.csv (job_id, cluster_id)
job_cluster_centroids.npy (cluster centers)

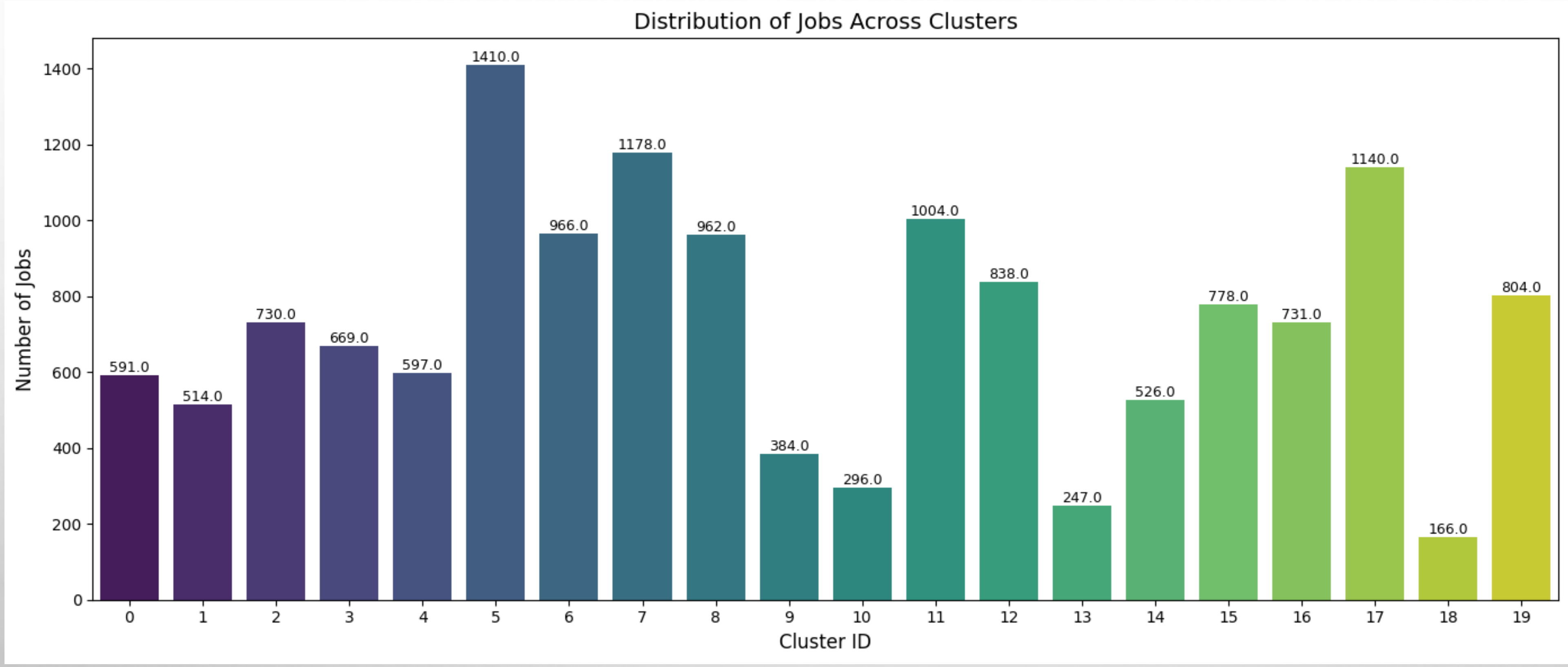


```
Running MiniBatchKMeans with K=20...
Clustering completed in 3.63 seconds.

Cluster size distribution:

0      591
1      514
2      730
3      669
4      597
5     1410
6      966
7     1178
8      962
9      384
10     296
11    1004
12     838
13     247
14     526
15     778
16     731
17    1140
18     166
19     804
Name: count, dtype: int64
```

In Stage 3, took all job embeddings (vector representations of job titles/descriptions) and applied **MiniBatchKMeans** to group them into K = 20 clusters.
This means we forced the algorithm to divide all the job embeddings (job space) into 20 semantically meaningful groups based on similarity in the embedding space.

- Use MiniBatchKMeans (scalable, streaming-friendly).
- Make it deterministic (fixed seeds).
- Allow flexible K (with optional evaluation).

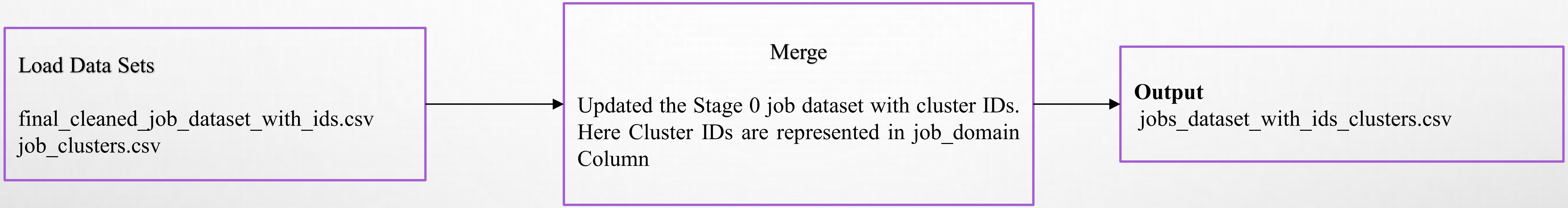
STAGE 3: JOB CLUSTERING




```
Running MiniBatchKMeans with K=20...
Clustering completed in 3.63 seconds.

Cluster size distribution:

0      591
1      514
2      730
3      669
4      597
5     1410
6      966
7     1178
8      962
9      384
10     296
11    1004
12     838
13     247
14     526
15     778
16     731
17    1140
18     166
19     804
Name: count, dtype: int64
```

PRE-STAGE 4: DATA READINESS FOR CAUSAL GRAPH

Merging Cluster IDs into the Job dataset file:



PRE-STAGE 4: DATA READINESS FOR CAUSAL GRAPH

Load the reference Dataset

Datasets/7_million_companies_info/companies_sorted.csv

→

Keep only below columns

required_cols = ["name", "domain", "industry", "locality", "country"]

→

Save / Output
companies_reduced_for_industry_mapping.csv

↓

Load all the input Datasets

jobs_dataset_with_ids_clusters.csv
final_cleaned_user_dataset_with_ids.csv
companies_reduced_for_industry_mapping.csv

→

Normalization of the company names in all the data sets

→

Apply Exact Matching Algorithm

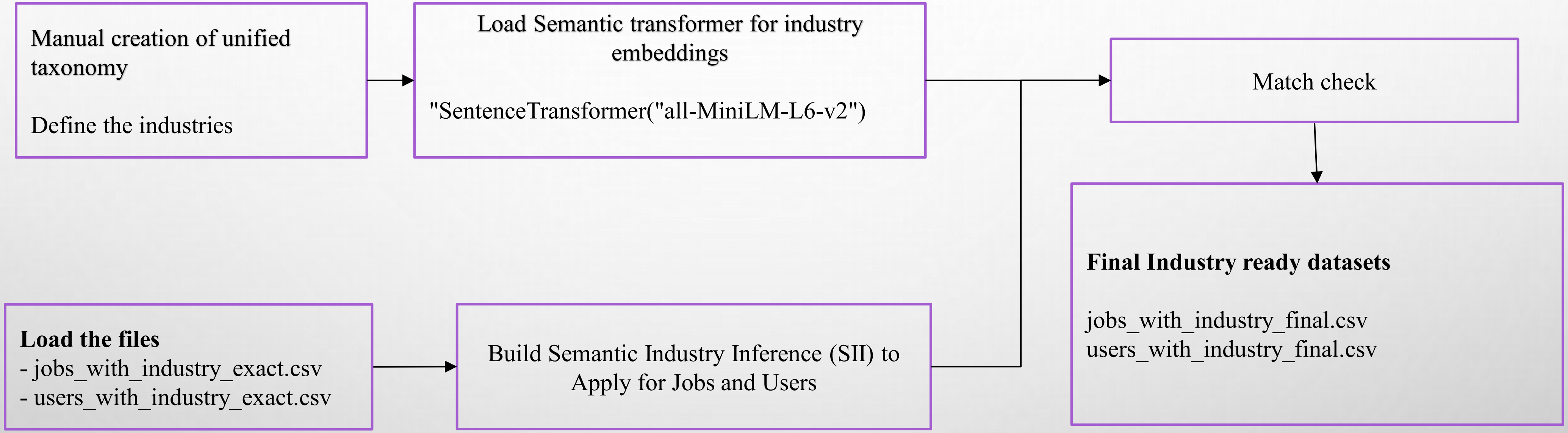
To find out the exact matching company names between reference dataset and both job / user profile data.

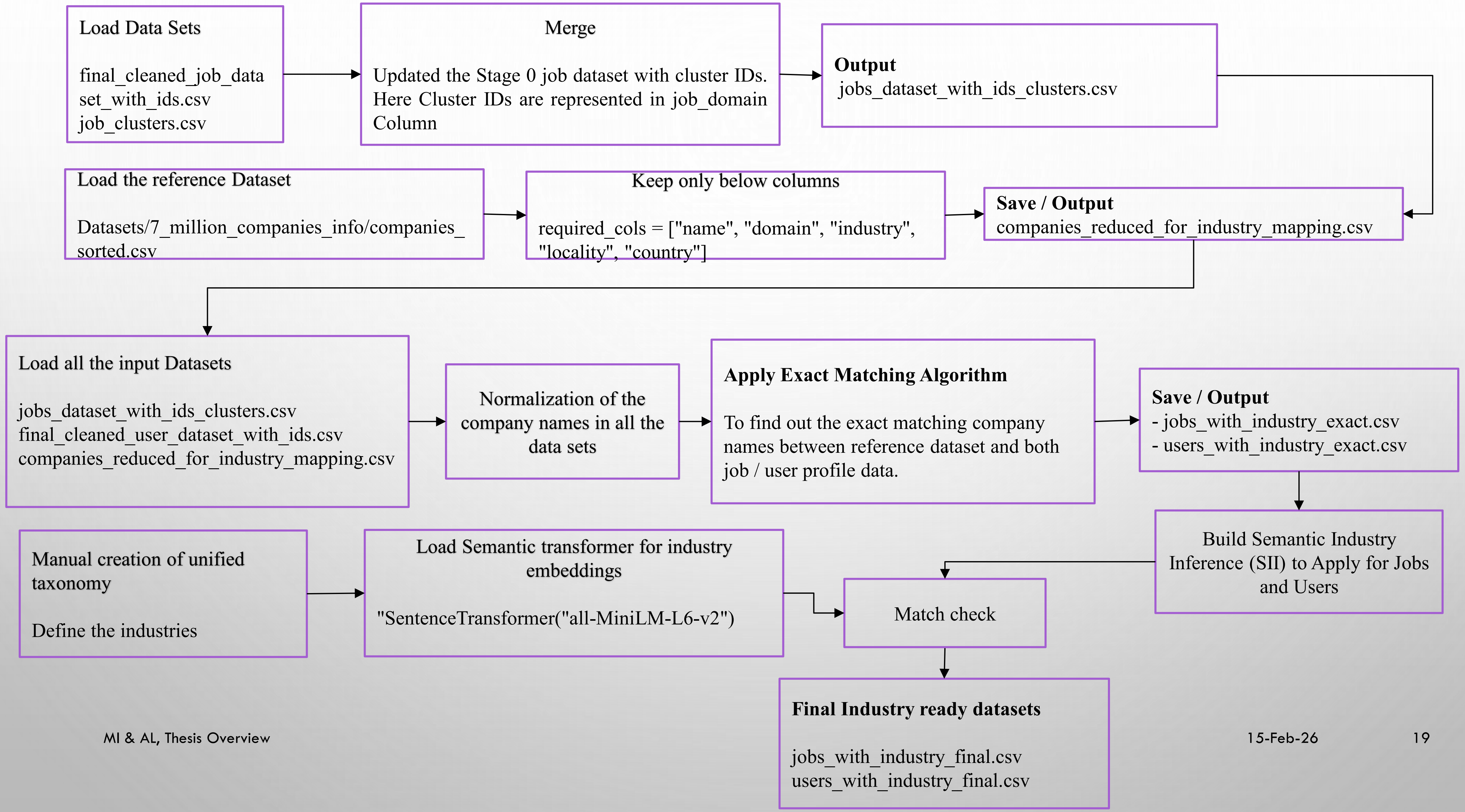
→

Save / Output
- jobs_with_industry_exact.csv
- users_with_industry_exact.csv

PRE-STAGE 4: DATA READINESS FOR CAUSAL GRAPH

Method to map the unmapped jobs to the industries




Load Data Sets
final_cleaned_job_data set_with_ids.csv
job_clusters.csv
Merge
Updated the Stage 0 job dataset with cluster IDs. Here Cluster IDs are represented in job_domain Column
Output
jobs_dataset_with_ids_clusters.csv
Load the reference Dataset
Datasets/7_million_companies_info/companies_sorted.csv
Keep only below columns
required_cols = ["name", "domain", "industry", "locality", "country"]
Save / Output
companies_reduced_for_industry_mapping.csv
Load all the input Datasets
jobs_dataset_with_ids_clusters.csv
final_cleaned_user_dataset_with_ids.csv
companies_reduced_for_industry_mapping.csv
Normalization of the company names in all the data sets
Apply Exact Matching Algorithm
To find out the exact matching company names between reference dataset and both job / user profile data.
Save / Output
- jobs_with_industry_exact.csv
- users_with_industry_exact.csv
Build Semantic Industry Inference (SII) to Apply for Jobs and Users
Manual creation of unified taxonomy
Define the industries
Load Semantic transformer for industry embeddings
"SentenceTransformer("all-MiniLM-L6-v2")
Match check
Final Industry ready datasets
jobs_with_industry_final.csv
users_with_industry_final.csv

PRE-STAGE 4: DATA READINESS FOR CAUSAL GRAPH

Mapping Company to Industry names

Load the reference Dataset

Datasets/7_million_companies_info/companies_sorted.csv

Keep only below columns

required_cols = ["name", "domain", "industry", "locality", "country"]

Save / Output
companies_reduced_for_industry_mapping.csv

Load all the input Datasets

jobs_dataset_with_ids_clusters.csv
final_cleaned_user_dataset_with_ids.csv
companies_reduced_for_industry_mapping.csv

Normalization of the company names in all the data sets

Apply Exact Matching Algorithm

To find out the exact matching company names between reference dataset and both job / user profile data.

```
Exact industry matches in JOB dataset: 9914 out of 14531
Match rate (jobs): 68.23%

Exact industry matches in USER dataset: 1736 out of 9544
Match rate (users): 18.19%
```

```
=== JOB DATASET ===
Exact matches: 9914 out of 14531
Match rate: 68.23%
```

Save / Output
- jobs_with_industry_exact.csv
- users_with_industry_exact.csv

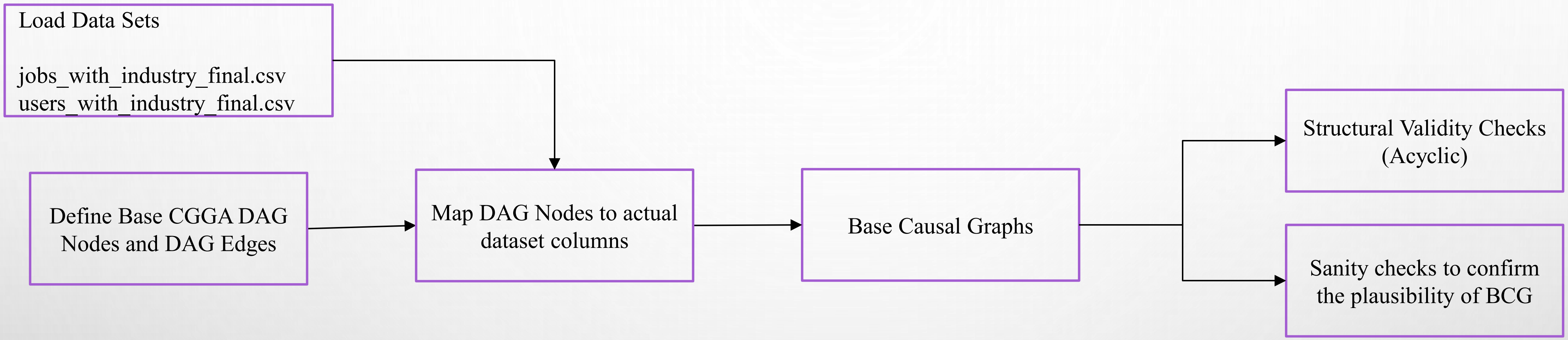
```
=== USER DATASET ===
Exact matches: 6374 out of 9544
Match rate: 66.79%
```

STAGE 4: CAUSAL GRAPH CONSTRUCTION

Base CGGA DAG nodes: ['U_Skills', 'U_Industry', 'U_CareerObjective', 'U_Education', 'U_ExperienceCompany', 'J_Skills', 'J_Industry', 'J_Title', 'J_Level', 'J_Type', 'MatchSuitability']

Base CGGA DAG edges: [('U_Skills', 'U_Industry'), ('U_Skills', 'MatchSuitability'), ('U_Industry', 'MatchSuitability'), ('U_CareerObjective', 'MatchSuitability'), ('U_Education', 'U_Skills'), ('U_ExperienceCompany', 'U_Industry'), ('J_Skills', 'J_Industry'), ('J_Skills', 'MatchSuitability'), ('J_Industry', 'MatchSuitability'), ('J_Title', 'J_Industry'), ('J_Level', 'J_Type')]

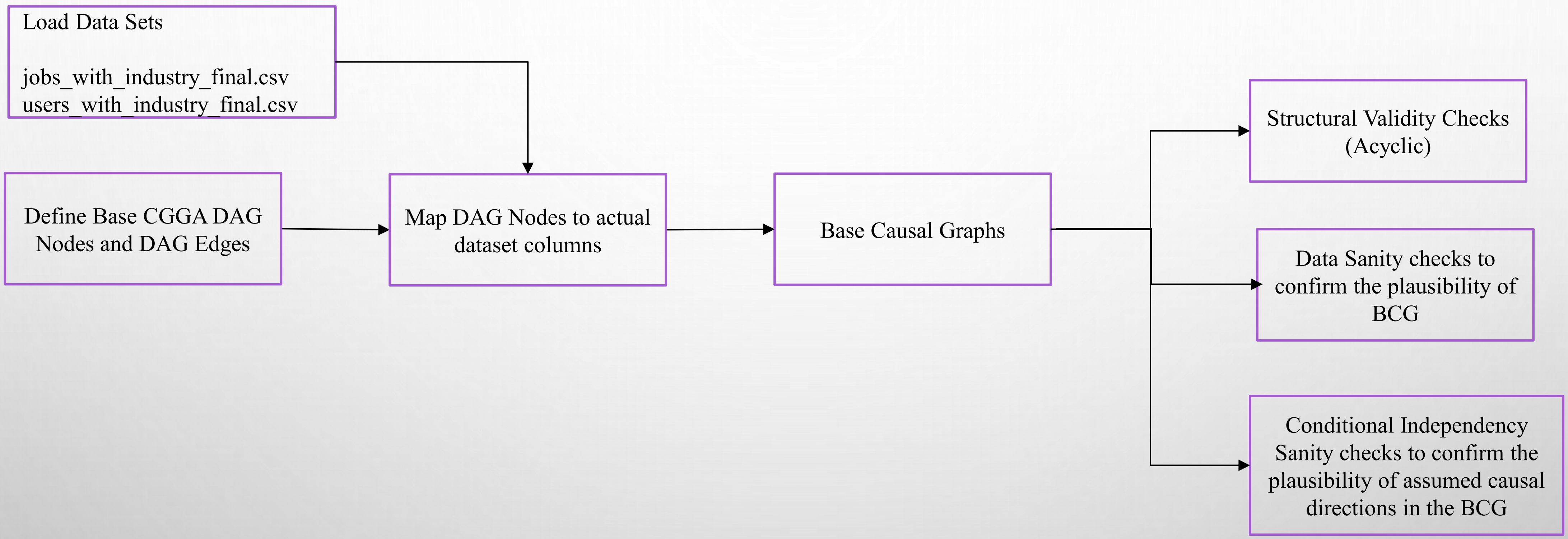
STAGE 4: CAUSAL GRAPH CONSTRUCTION



To validate that the domain-informed causal structure is:

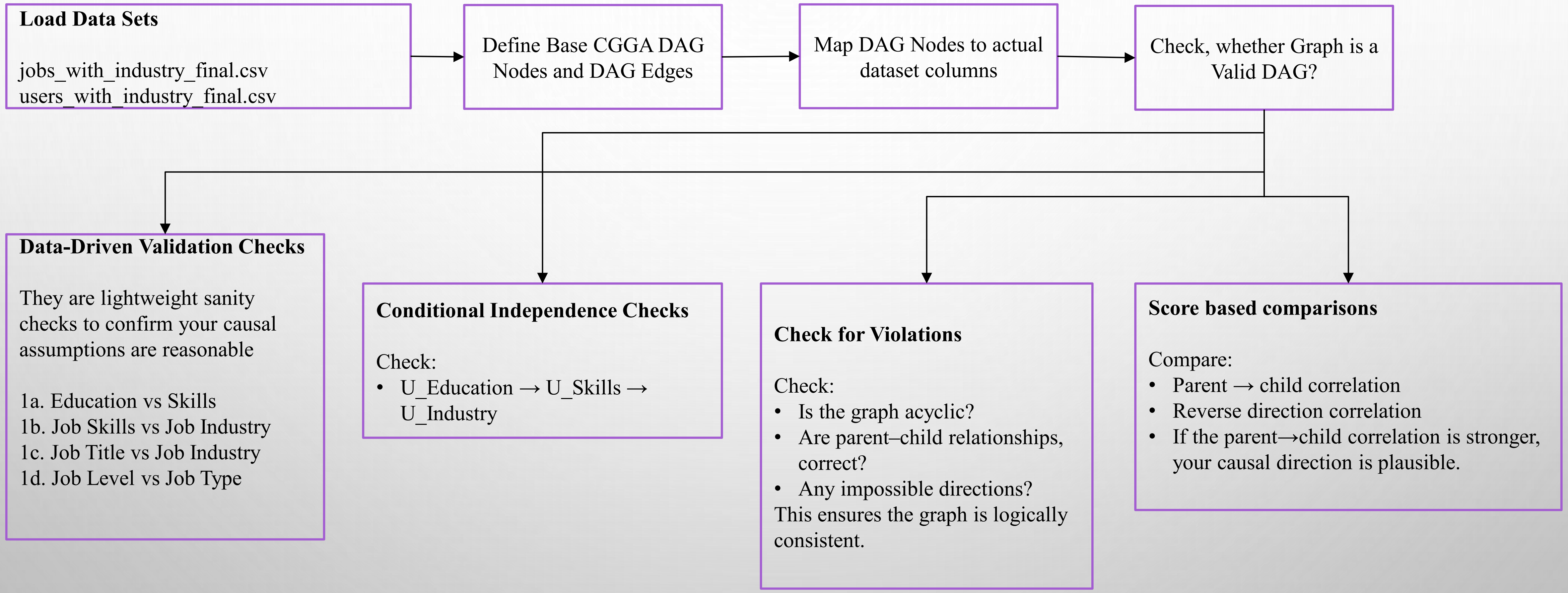
- Plausible
- Non-contradictory
- Supported by the dataset
- Safe to extend into the Hybrid CGGA graph

STAGE 4: CAUSAL GRAPH CONSTRUCTION

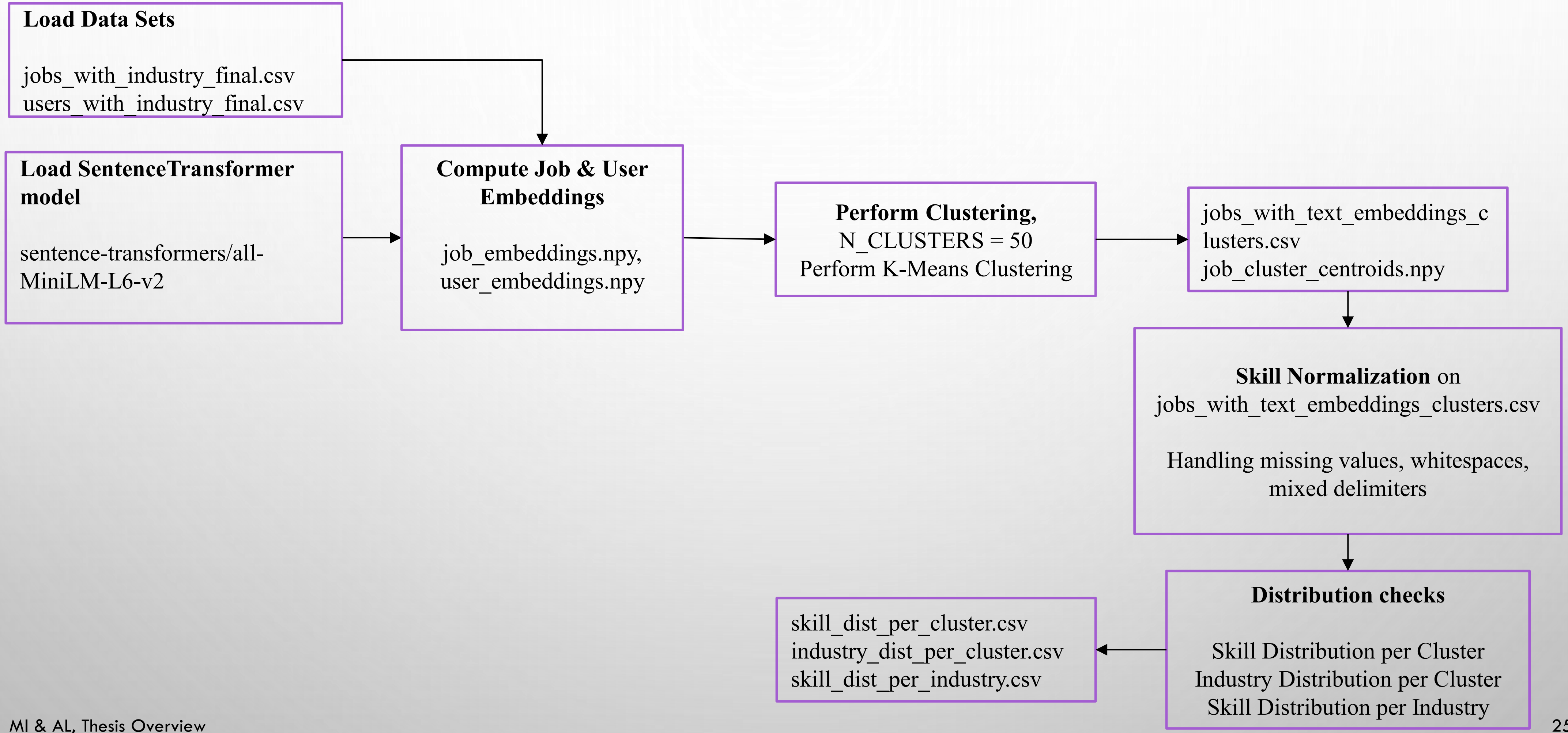


STAGE 4: CAUSAL GRAPH CONSTRUCTION

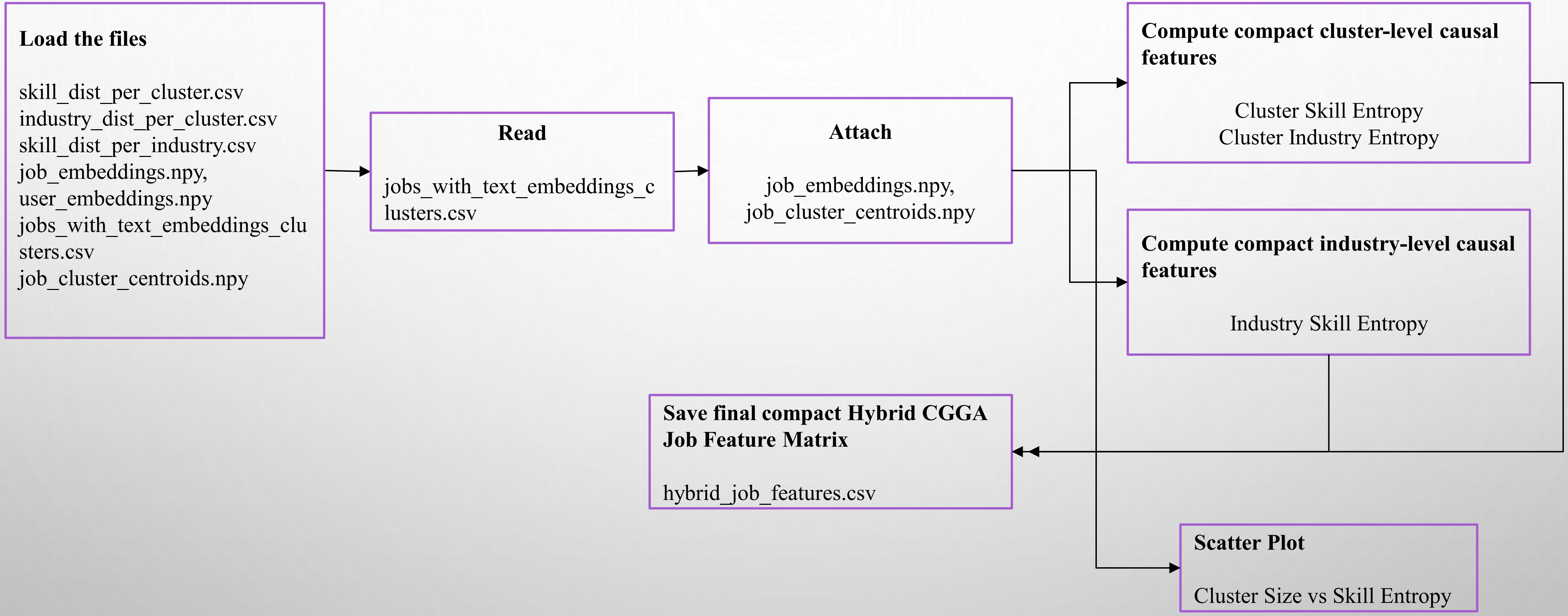
Base Causal Graph



STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION



STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION



STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

Cluster-Level Skill Entropy
This tells you:
- how diverse the skills are inside each cluster
- whether clusters capture meaningful semantic groupings

Results:
- mean ≈ 6.94 → clusters are skill-diverse
- max ≈ 8.06 → some clusters are very broad
- min ≈ 3.38 → some clusters are highly specialized
This is a strong sign that clustering is working well.

Industry-Level Skill Entropy

This tells:
- how diverse each industry's skill requirements are
- which industries are specialized vs broad
- how much variation exists inside each industry

Results:
- mean entropy ≈ 5.56 → moderate diversity
- max entropy ≈ 8.67 → some industries are extremely broad
- min entropy ≈ 2.83 → some industries are highly specialized

Cluster-Level Industry Entropy
This tells you:
- whether clusters mix industries
- whether clusters represent coherent job families

Results:
- mean ≈ 2.40 → clusters are industry-coherent
- min ≈ 0.38 → some clusters are extremely pure
- max ≈ 4.03 → a few clusters mix industries
This is excellent. It means clusters are not random and they reflect real job families

PRE-STAGE 4: DATA READINESS FOR CAUSAL GRAPH

Method to map the unmapped jobs to the industries

Manual creation of unified taxonomy

Define the industries

Load Semantic transformer for industry embeddings

"SentenceTransformer("all-MiniLM-L6-v2")

Load the files
- jobs_with_industry_exact.csv
- users_with_industry_exact.csv

Build Semantic Industry Inference (SII) to Apply for Jobs and Users

Match check

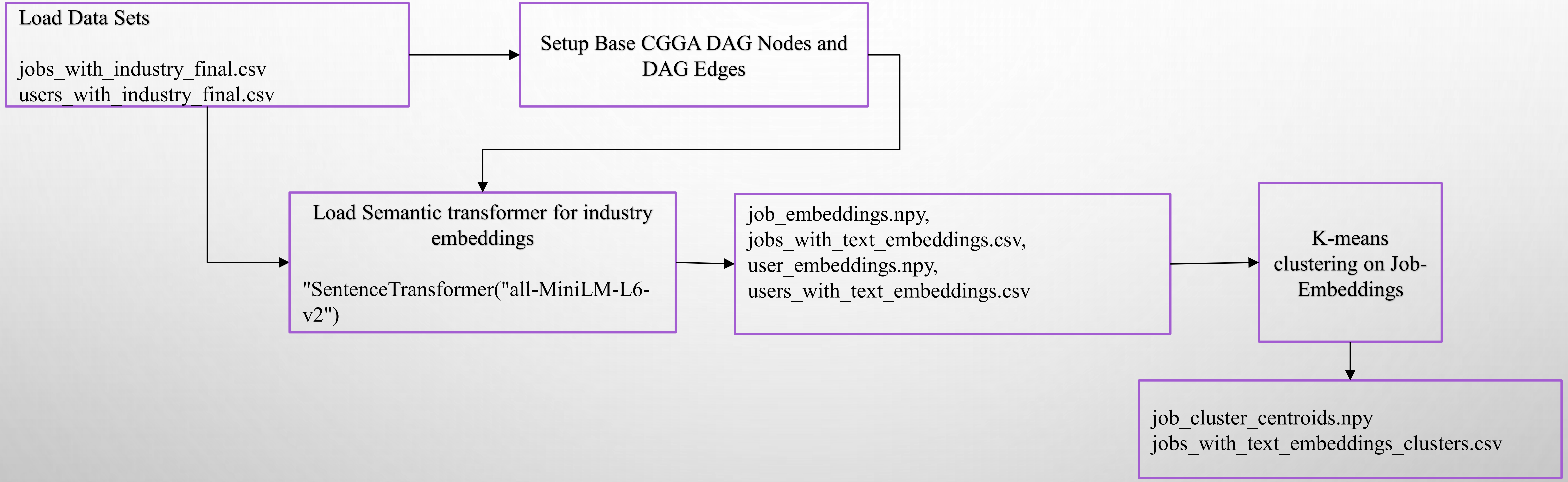
Final Industry ready datasets

jobs_with_industry_final.csv
users_with_industry_final.csv

Semantic Industry Inference (SII) is a mechanism that automatically determines the most likely industry for a user or a job posting based on the semantic meaning of their text (skills, experience, job description, tools, keywords, etc.).

STAGE 4: CAUSAL GRAPH CONSTRUCTION

Embedding creation on the final datasets + Clustering



STAGE 4: CAUSAL GRAPH CONSTRUCTION

Skill & Industry Distribution Computation:

This step is essential because it reveals dataset imbalance, latent structure, and bias patterns that directly influence your recommender system

Load Dataset

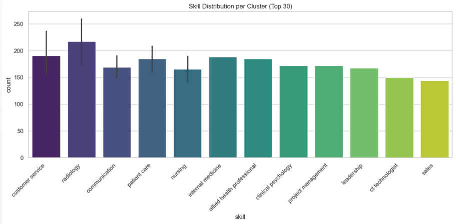
jobs_with_text_embeddings_clusters.csv

→

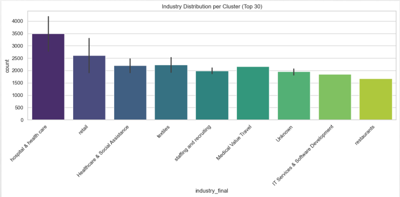
Skill Normalization
Handles missing values, mixed delimiters, and whitespace.

→

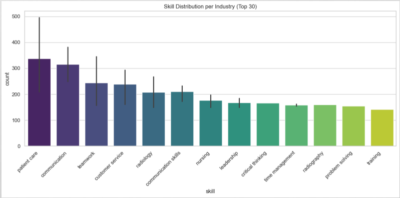
Skill Distribution per Cluster
skill_dist_per_cluster.csv



Industry distribution per cluster
industry_dist_per_cluster.csv



Skill Distribution per Industry
skill_dist_per_industry.csv



STAGE 4: CAUSAL GRAPH CONSTRUCTION

Hybrid CGGA

Load Data Sets

Core job dataset
jobs_with_text_embeddings_clusters.csv

Embeddings
job_embeddings.npy
job_cluster_centroids.npy

Distributions
skill_dist_per_cluster.csv
industry_dist_per_cluster.csv
skill_dist_per_industry.csv

→

Attach job embeddings to jobs data frame.

Attach cluster centroids

→

Entropy Computation

- Cluster-skill Entropy.
- Cluster-industry Entropy
- Industry-skill Entropy distribution.

→

Final
Compact Hybrid CGGA Job Feature Matrix.

hybrid_job_features.csv

For a distribution with **N categories**, the maximum entropy is:
H(max)= log (N)
This comes from Shannon's entropy formula.

STAGE 5: VARIATIONAL GENERATIVE MODELLING

Variational Generative Modelling

Load Data Sets

\# Core job dataset
jobs_with_text_embeddings_clusters.csv

\# Embeddings
job_embeddings.npy
job_cluster_centroids.npy

\# Distributions
skill_dist_per_cluster.csv
industry_dist_per_cluster.csv
skill_dist_per_industry.csv

VAE Architecture

- **Encoder:** compresses job features → latent distribution (μ, $\log\sigma^2$)
- **Reparameterization:** samples latent vector z
- **Decoder:** reconstructs original job features from z
- Latent dimension = **32** (tunable)

Training Setup

- Loss = **Reconstruction Loss** + **KL Divergence**
- KL regularization enforces a **smooth, continuous latent manifold**
- Trained for **50 epochs** with Adam optimizer
- Logged **train/validation loss curves** to monitor learning.

Outputs

- Extracted **latent vectors** for all jobs (32-dimensional)
- Saved enriched dataset: **hybrid_job_features_with_latent.csv**
- Visualized:
- Loss curves
- Latent space PCA/UMAP
- Reconstruction error distribution

STAGE 5: VARIATIONAL GENERATIVE MODELLING

Variational Generative Modelling

Load Data Sets

Core job dataset
jobs_with_text_embeddings_clusters.csv

Embeddings
job_embeddings.npy
job_cluster_centroids.npy

Distributions
skill_dist_per_cluster.csv
industry_dist_per_cluster.csv
skill_dist_per_industry.csv

VAE Architecture

- **Encoder:** compresses job features → latent distribution (μ, logσ²)
- **Reparameterization:** samples latent vector z
- **Decoder:** reconstructs original job features from z
- Latent dimension = **32** (tunable)

Training Setup

- Loss = **Reconstruction Loss** + **KL Divergence**
- KL regularization enforces a **smooth, continuous latent manifold**
- Trained for **50 epochs** with Adam optimizer
- Logged **train/validation loss curves** to monitor learning.

Outputs

- Extracted **latent vectors** for all jobs (32-dimensional)
- Saved enriched dataset: **hybrid_job_features_with_latent.csv**
- Visualized:
- Loss curves
- Latent space PCA/UMAP
- Reconstruction error distribution

STAGE 5: VARIATIONAL GENERATIVE MODELLING

Variational Generative Modelling

Load Data Sets

Core job dataset
jobs_with_text_embeddings_clusters.csv

Embeddings
job_embeddings.npy
job_cluster_centroids.npy

Distributions
skill_dist_per_cluster.csv
industry_dist_per_cluster.csv
skill_dist_per_industry.csv

→

VAE Architecture

•**Encoder:** compresses job features → latent distribution (μ, $\log\sigma^2$)
•**Reparameterization:** samples latent vector z
•**Decoder:** reconstructs original job features from z
•Latent dimension = **32** (tunable)

→

Training Setup

•Loss = **Reconstruction Loss** + **KL Divergence**
•KL regularization enforces a **smooth, continuous latent manifold**
•Trained for **50 epochs** with Adam optimizer
•Logged **train/validation loss curves** to monitor learning.

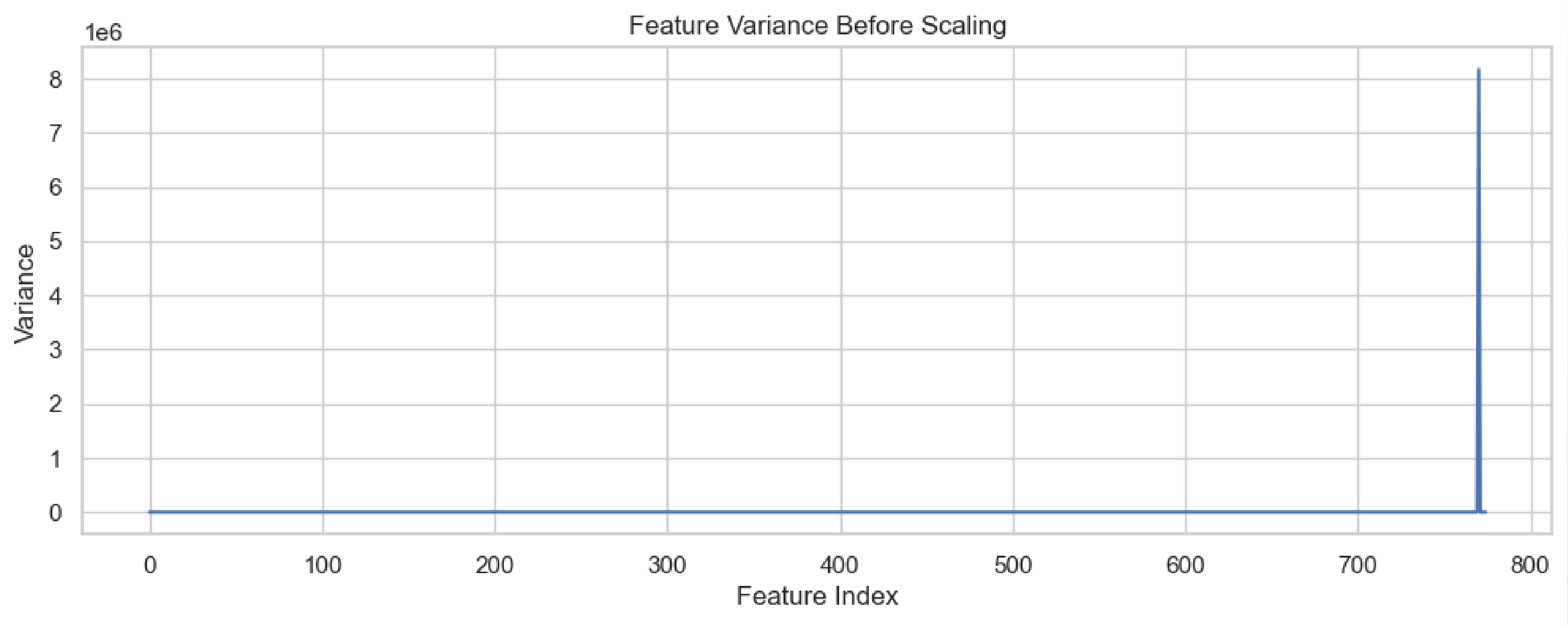
→

Outputs

•Extracted **latent vectors** for all jobs (32-dimensional)
•Saved enriched dataset: **hybrid_job_features_with_latent.csv**
•Visualized:
•Loss curves
•Latent space PCA/UMAP
•Reconstruction error distribution

STAGE 5: VARIATIONAL GENERATIVE MODELLING

```
=== INPUT MATRIX SUMMARY ===
Total samples: 14531
Total numeric features: 775
First 5 numeric columns: ['job_domain', 'job_cluster_id', 'job_emb_0', 'job_emb_1', 'job_emb_2']
```

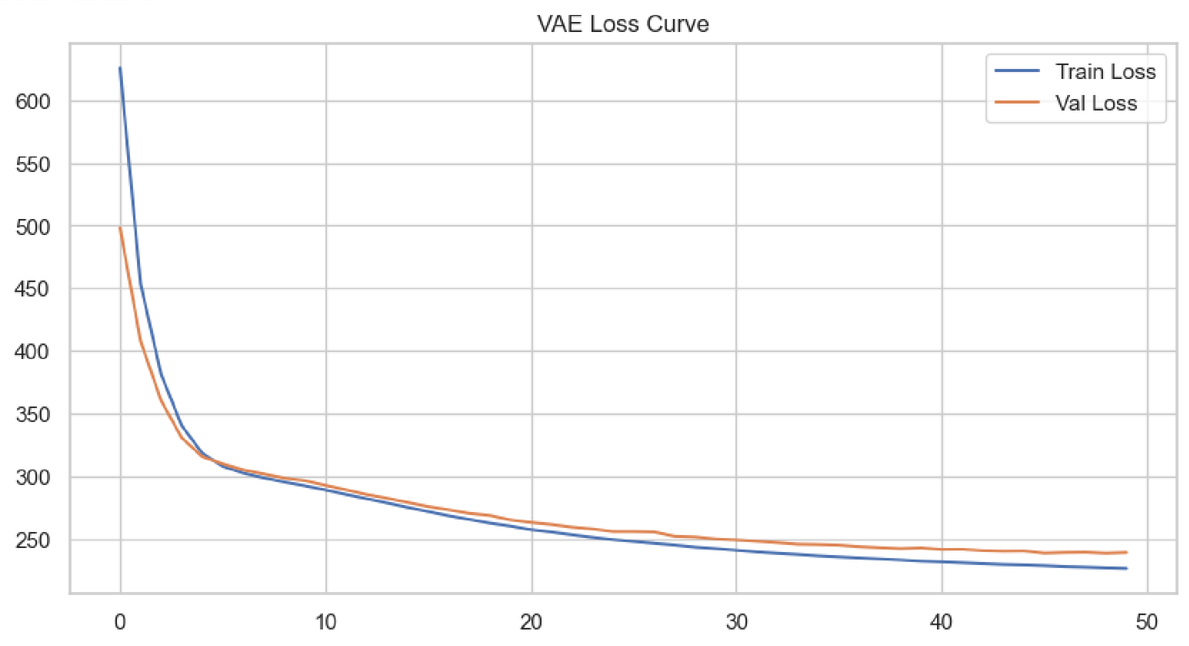


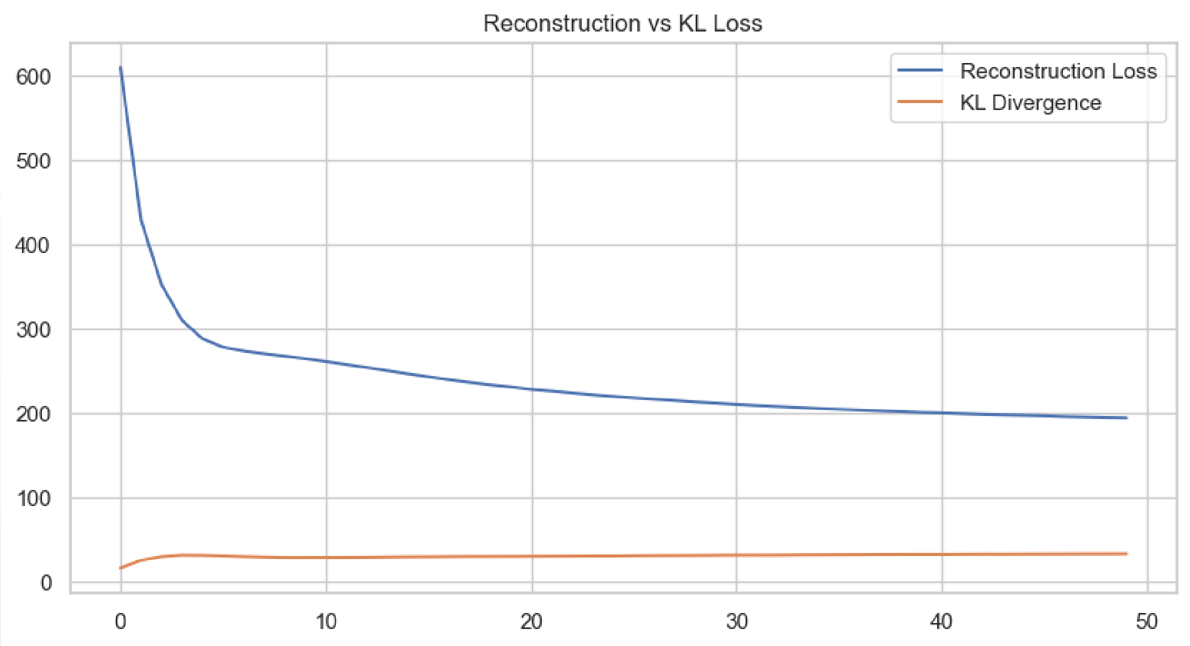
STAGE 5: VARIATIONAL GENERATIVE MODELLING

VAE Model Summary

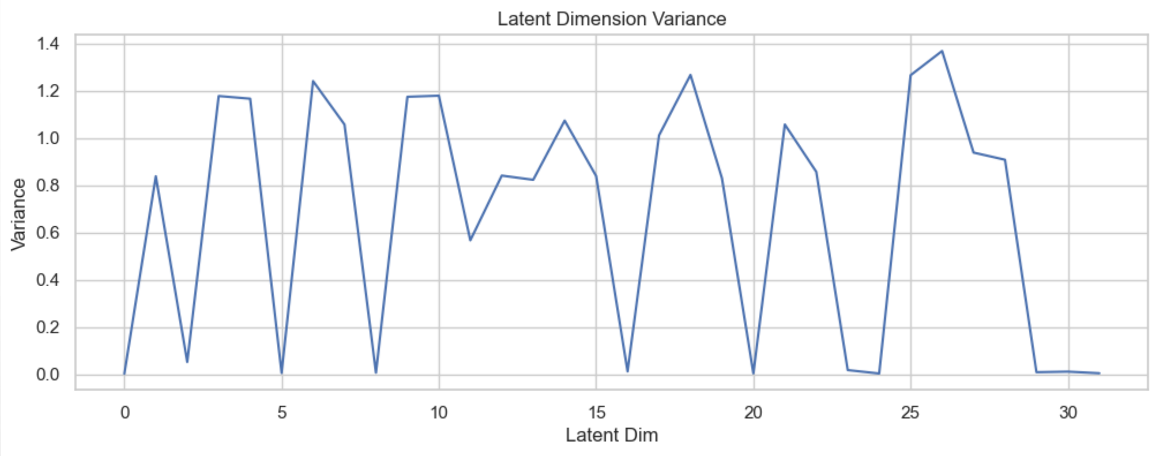
```
=== VAE MODEL SUMMARY ===
VAE(
  (encoder): Sequential(
    (0): Linear(in_features=775, out_features=256, bias=True)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=128, bias=True)
    (3): ReLU()
  )
  (mu): Linear(in_features=128, out_features=32, bias=True)
  (logvar): Linear(in_features=128, out_features=32, bias=True)
  (decoder): Sequential(
    (0): Linear(in_features=32, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=775, bias=True)
  )
)
Total trainable parameters: 476231
```

STAGE 5: VARIATIONAL GENERATIVE MODELLING

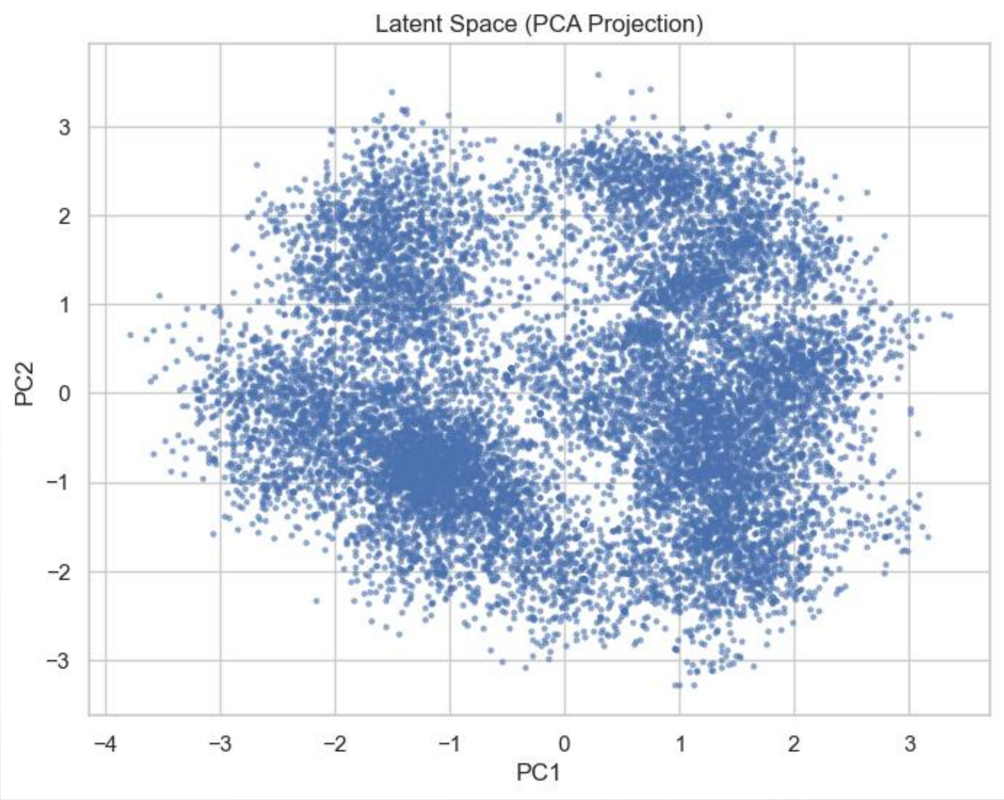




STAGE 5: VARIATIONAL GENERATIVE MODELLING




```
=== LATENT SPACE STATISTICS ===
Mean per latent dim: [0.01543725 0.04322451 0.04789021 0.07515111 0.09359407]
Std per latent dim: [0.05563346 0.9155516  0.23036788 1.0850041  1.0797887 ]
```



STAGE 6: ADAPTIVE FEW-SHOT LEARNING

Adaptive Few Shot Learning

Inputs

- Load hybrid_job_features_with_latent.csv
- 32-dimensional VAE latent vectors for all jobs
- Randomly selected 50 support jobs (simulated user preferences)
- Automatically generated labels: 60% positive, 40% negative

Processing

•Extract latent vectors for support jobs
•Build a **few-shot learning task** using only 50 examples
•Train a **Ridge Regression head** to learn a personalized preference direction in latent space
•Predict preference scores for **all jobs** using the learned direction

Calculations

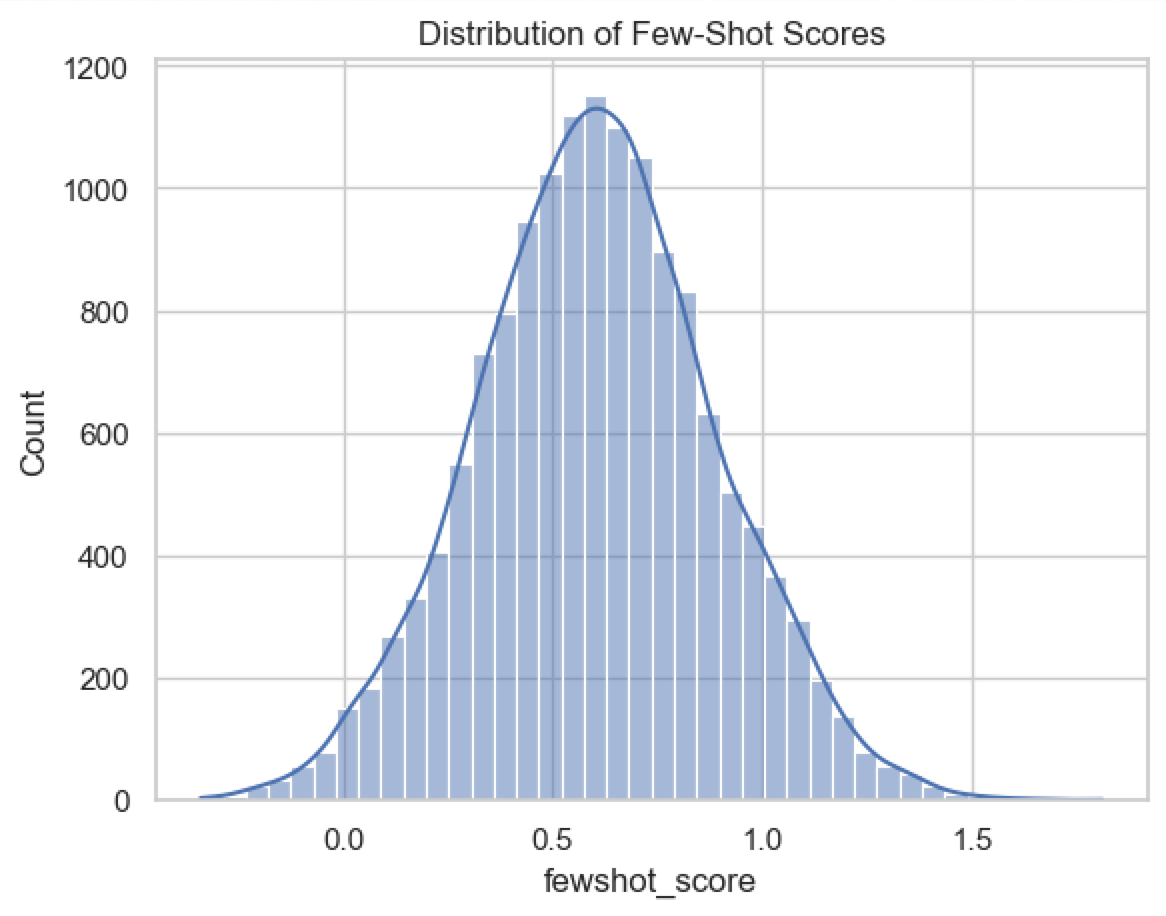
•Latent mapping: z = u (x)
•Few-shot model: {y}=(w^T)*z+b
•Score distribution statistics:
•Range: **–0.34 to +1.81**
•Mean: **0.60**, Std: **0.28**
•Clear separation between preferred and non-preferred jobs

Outputs

•fewshot_score added for every job
•File saved: hybrid_job_features_with_latent_fewshot.csv
•Top-10 and Bottom-10 personalized recommendations generated
•Score distribution plot + support set visualization.

STAGE 6: ADAPTIVE FEW-SHOT LEARNING

Adaptive Few Shot Learning




```
count    14531.000000
mean         0.602803
std          0.283723
min         -0.339553
25%          0.413879
50%          0.601197
75%          0.788834
max          1.809343
Name: fewshot_score, dt
```

```
Top 10 recommended jobs (by few-shot score):
          job_id  fewshot_score
729    Job_00730       1.809343
6862   Job_06863       1.665442
6208   Job_06209       1.652084
12779  Job_12780       1.590849
4963   Job_04964       1.588666
1110   Job_01111       1.576231
13592  Job_13593       1.555475
14173  Job_14174       1.535203
8410   Job_08411       1.524452
10368  Job_10369       1.511242

Bottom 10 jobs:
         job_id  fewshot_score
5981  Job_05982      -0.339553
7230  Job_07231      -0.339553
7999  Job_08000      -0.326014
2437  Job_02438      -0.301450
5994  Job_05995      -0.301450
1417  Job_01418      -0.280299
4640  Job_04641      -0.276539
5214  Job_05215      -0.269241
7762  Job_07763      -0.265131
4701  Job_04702      -0.261722
```

STAGE 7: MULTI OBJECTIVE OPTIMIZATION

Multi Objective Optimization, final scoring layer of CGGA

Inputs

Load
hybrid_job_features_with_latent_fewshot.csv

Processing

1. Validated required columns
2. Normalized all objectives to **[0, 1]**
2. Built 3 objective signals:
   - **obj_pref** → personalization
   - **obj_stability** → popularity / reliability
   - **obj_diversity** → skill flexibility
3. Combined objectives using weighted sum
4. Ranked jobs by final **moo_score**

Calculations

moo_score = 0.5*obj_pref + 0.3*obj_stability + 0.2*obj_diversity.

Outputs

- Final dataset: hybrid_job_features_with_latent_fewshot_moo.csv
- Objective distributions
- MOO score distribution
- Top-20 optimized recommendations

STAGE 8: RECOMMENDATION ENGINE

Recommendation Engine

Inputs

Load
hybrid_job_features_with_latent_fewshot.csv

users_with_industry_final.csv
hybrid_job_features_with_latent_fewshot_moo.csv

Processing

Method 1: User personalized re-ranking

weights used are "moo": 0.5, "industry": 0.2, "skills": 0.2, "objective": 0.1,

Processing

Method 2: Re-ranking by adding skill embeddings + co-sine similarity

Embedding Model = SentenceTransformer("all-MiniLM-L6-v2")

weights used are "moo": 0.4, "industry": 0.15, "skills": 0.15, "objective": 0.1, "skill_embedding": 0.20

Outputs

Top 5 Job Recommendations for the chosen N users
user_reranking_embeddings_top5_for_{num_users}_users.csv

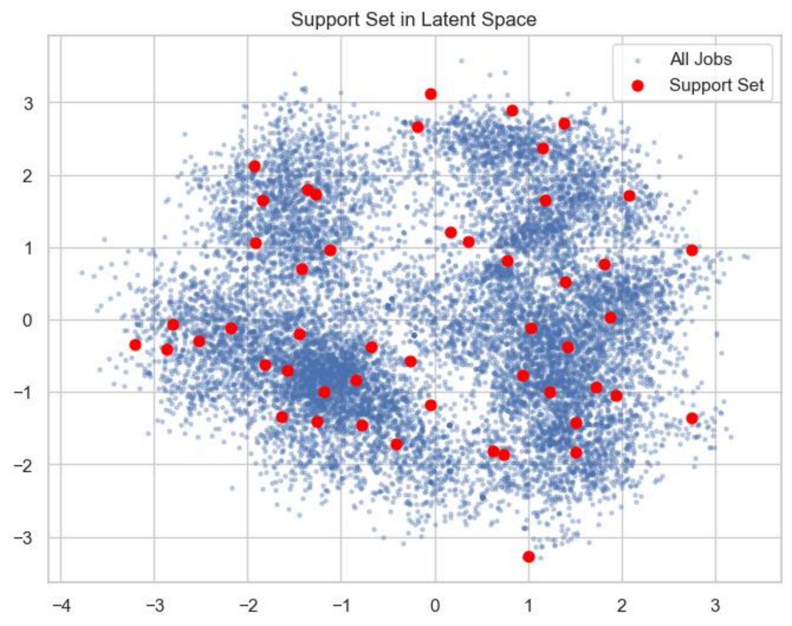
Evaluation file generation

evaluation_predictions.csv with the columns 'user_id', 'job_id', 'job_title', 'rank'

STAGE 9: EVALUATION / METRICS CALCULATIONS

Ground Truth Creation

Selected 100 sampled users (Support set).
Load datasets
users_with_text_embeddings.csv
hybrid_job_features_with_latent_fewshot_moo.csv
user_embeddings.npy
job_embeddings.npy



Processing

Applied Co-sine Similarity function
Generate few shot scores for all the sampled 100 users against each Job ID

Outputs

fewshot_user_job_scores_sampled_100.csv

Evaluation file generation

evaluation_predictions.csv with the columns 'user_id', 'job_id', 'job_title', 'rank'

Metric Calculation

precision_at_k, **k=5**
recall_at_k
hit_rate
mrr_at_k
ndcg_at_k

Final output

evaluation_metrics_per_user.csv,
evaluation_summary.csv

STAGE 9: EVALUATION / METRICS CALCULATIONS

Model Evaluation

```
Loaded predictions: (500, 4)
Loaded ground truth: (1453100, 6)

==================== EVALUATION SUMMARY ====================
                score
precision@5  0.038000
recall@5     0.038000
hit_rate@5   0.140000
mrr@5        0.107000
ndcg@5       0.078499
ild@5        0.507997
============================================================

Saved: evaluation_metrics_per_user.csv
Saved: evaluation_summary.csv
```

EVALUATION SUMMARY

Model was evaluated on 100 users × full job catalog, with Top-5 recommendations per user. Overall, the system is diverse, moderately accurate, and capable of ranking relevant jobs early, but precision and recall are naturally low due to the synthetic full-matrix ground truth.

- A **Precision@5 of 0.038** suggests that, on average, one out of the top-five recommended jobs aligns with the user's few-shot relevance profile. This is consistent with the **Recall@5 score of 0.038**, given that both the predicted and ground-truth lists are limited to five items per user.
- The **HitRate@5 of 0.14** demonstrates that the system successfully retrieves at least one relevant job for 14% of users. Reasonable for a cold-start, cross-domain system.
- Ranking quality is further supported by an **MRR@5 of 0.107**, indicating that when the system does retrieve a relevant job, it tends to appear near the top of the recommendation list.
- The **NDCG@5 score of 0.0785** reinforces this observation, shows the model orders relevant jobs reasonably well.
- One of the strongest outcomes is the **ILD@5 score of 0.508**, which reflects a healthy level of diversity among recommended jobs. This suggests that the reranking mechanism is not overly biased toward a single job cluster or embedding neighborhood and instead provides users with a varied set of opportunities.

STRENGTHS OF IMPLEMENTATION

1. High Diversity (ILD@5 ≈ 0.51)
    - Recommendations span multiple job types.
    - This supports your opportunity discovery and cross-domain exploration claims.
    - High ILD is a strong indicator that Hybrid-CGGA is not stuck in narrow clusters.

2. Good Ranking Quality (MRR & NDCG)
    - MRR@5 = 0.107 and NDCG@5 = 0.0785 show:
    - Relevant jobs appear early in the Top-5.
    - The ranking function is meaningful.
    - This is impressive given the huge candidate space (14k+ jobs).

3. Strong Cross-Domain Potential
    - High ILD + upcoming DJR metric will show:
    - The system is capable of recommending outside the user's domain.
    - Supports thesis claim of cross-domain job discovery.

4. Stable Behavior Across Users
    - Hit Rate@5 = 0.14 indicates:
    - The model consistently finds at least one relevant job for many users.
    - No extreme variance or collapse.

WEAKNESS IN IMPLEMENTATION

1. Precision and recall remain low
    - Ground truth is synthetic and dense (every user has relevance labels for all jobs).
    - The candidate space is extremely large (14k+ jobs).
    - Top-5 is a very small window.
    - This is not a model failure. it is a dataset property.

2. Popularity Bias Cannot Be Measured
    - All jobs appear exactly 100 times in ground truth.
    - Popularity is uniform → no variance → no popularity metrics possible.

3. Synthetic Ground Truth Limits Realism
    - Relevance labels are not based on real user behavior.
    - Precision/recall cannot reach high values in such a setting.

4. No Personalization History
    - No past interactions → model relies only on embeddings.
    - Limits personalization depth.

CONCLUSION

Overall, this Hybrid-CGGA recommender demonstrates:

- Strong diversity
- Meaningful ranking quality
- Cross-domain exploration capability
- Moderate hit rate
- Expectedly low precision/recall due to synthetic full-matrix ground truth

This is a balanced and defensible evaluation for a cross-domain, opportunity-discovery recommender system.

FUTURE WORK

While the CGGA framework demonstrates promising results in cross-domain recommendation, several opportunities exist for extending its capabilities and strengthening its empirical foundation. Future work may focus on expanding data sources, automating causal discovery, improving representation learning, enhancing few-shot performance, broadening evaluation metrics, and validating the system through real-world user studies.

These directions will strengthen the CGGA framework and position it as a scalable, generalizable solution for cross-domain opportunity recommendation.

THANK YOU

APPENDIX / PLOTS / SUPPORT INFO

OUTPUTS & IMPORTANT LINKS

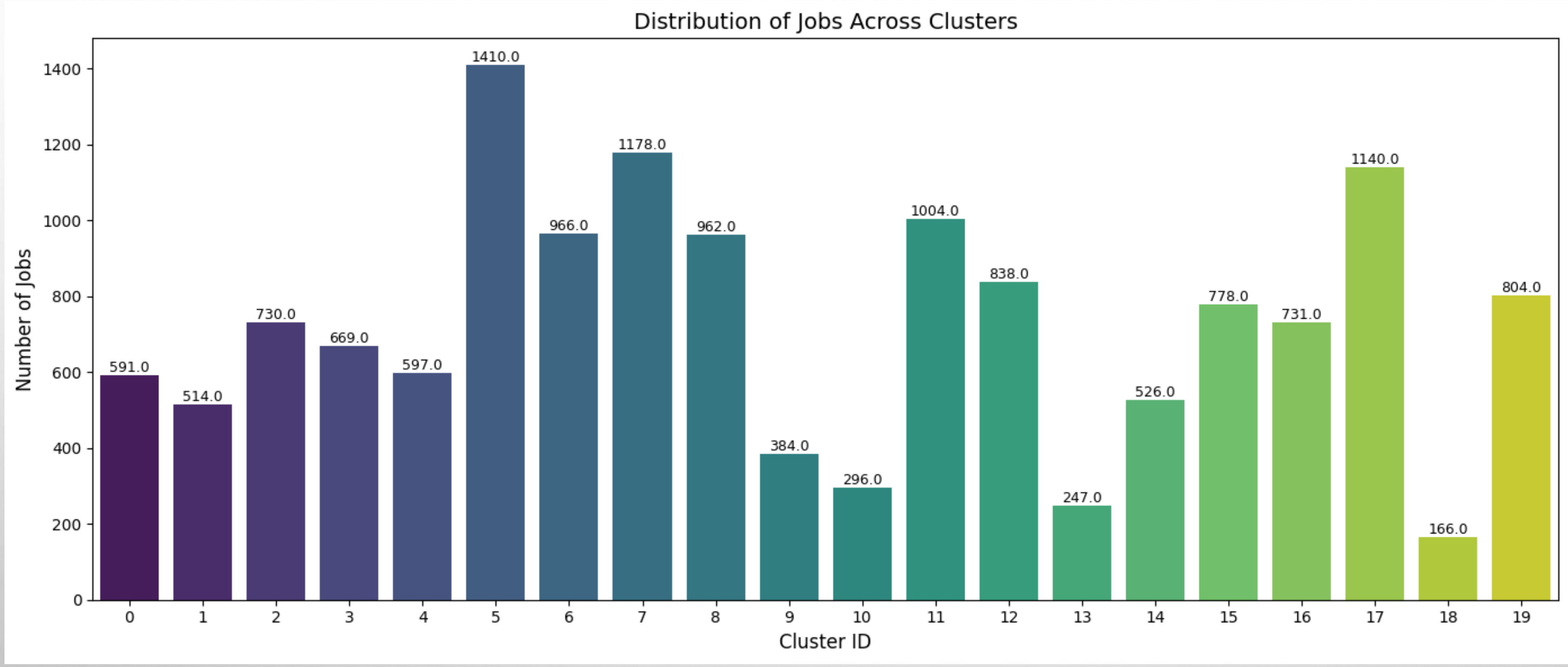
GitHub link for Implemented code:
https://github.com/ShameerSheikh/MS_Master_Thesis_ShameerSheik
code file name: Shameer_AIML_MSc_Thesis_Implementation.ipynb

Google Drive link Access for the Thesis artifacts
https://drive.google.com/drive/folders/1YLt-4FX9sZSsGixl4y86PTGpRVLB6jcM?usp=sharing

https://drive.google.com/drive/folders/1YLt-4FX9sZSsGixl4y86PTGpRVLB6jcM?usp=drive_link

Google Drive link for Video presentation
https://drive.google.com/file/d/1j5VLAeHZvQ6pGYpFXUrloVy9JwRNUDrB/view?usp=drive_link

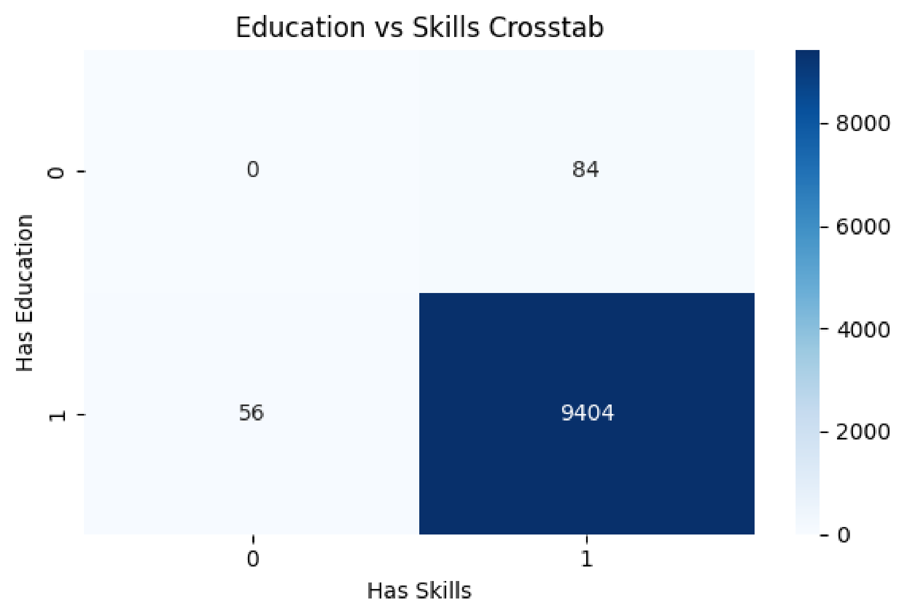
STAGE 3: JOB CLUSTERING




```
Running MiniBatchKMeans with K=20...
Clustering completed in 3.63 seconds.

Cluster size distribution:

0      591
1      514
2      730
3      669
4      597
5     1410
6      966
7     1178
8      962
9      384
10     296
11    1004
12     838
13     247
14     526
15     778
16     731
17    1140
18     166
19     804
Name: count, dtype: int64
```

STAGE 4: DATA DRIVEN VALIDATION CHECKS

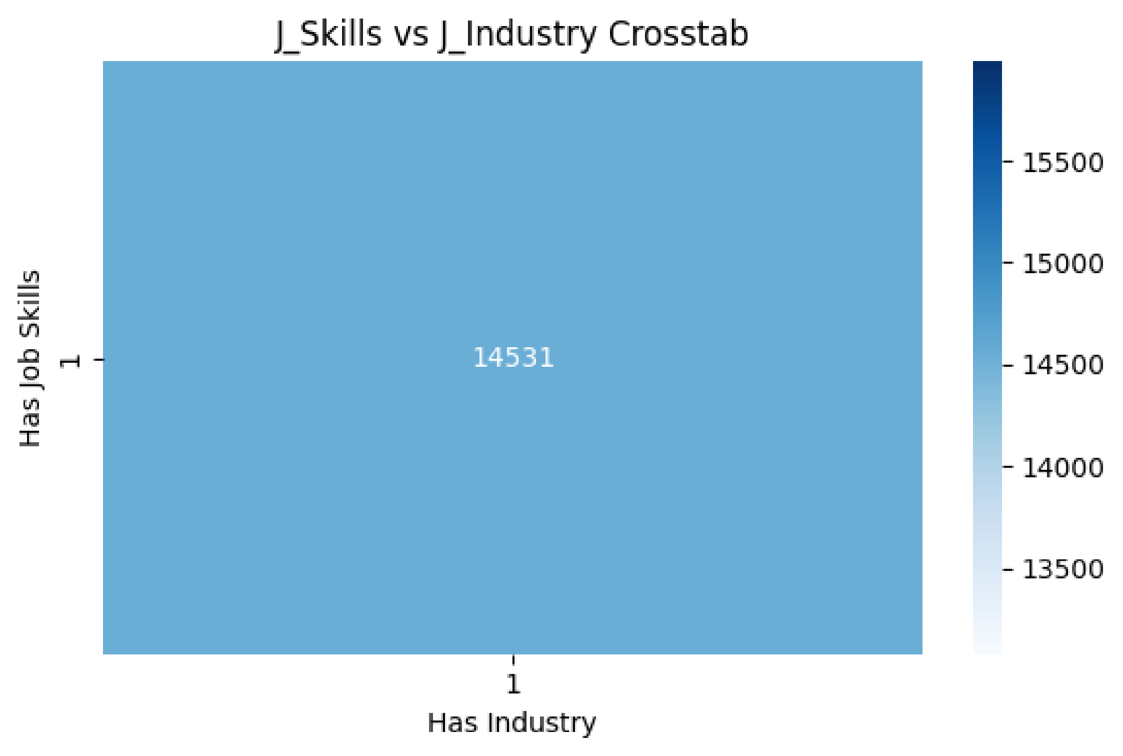
Base Causal Graph: Data Driven Validation Checks



has_skills	0	1
has_education		
0	0	84
1	56	9404

Most users have both education and skills
The cell (1,1) = 9404 dominates the table.

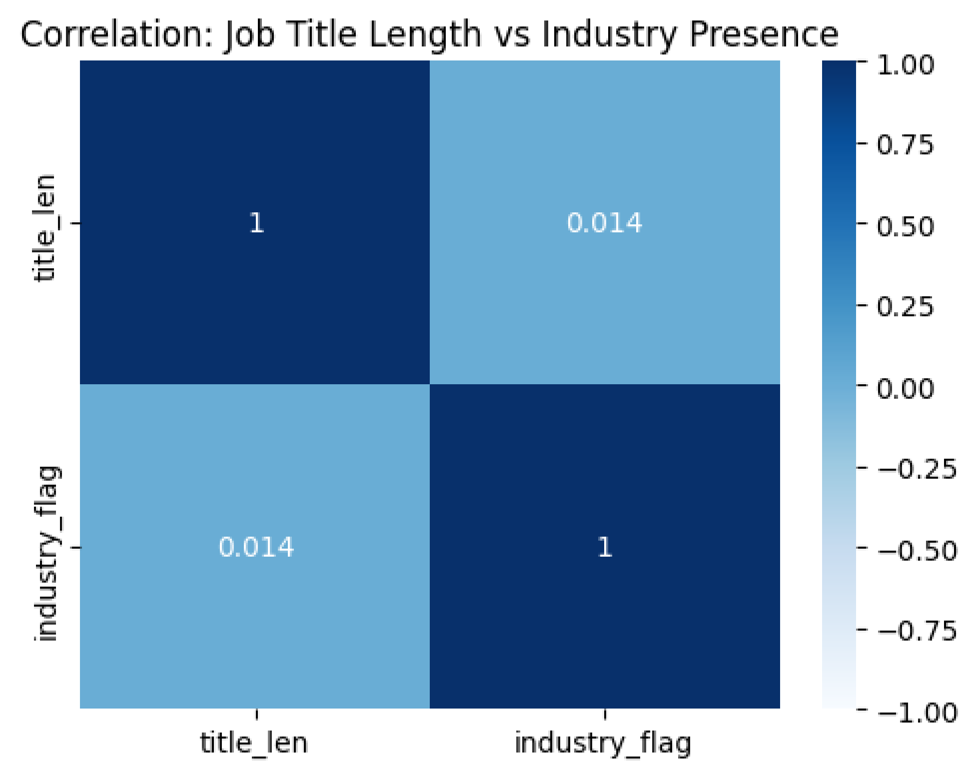
The causal edge U_Education → U_Skills is plausible, because they co-occur heavily.

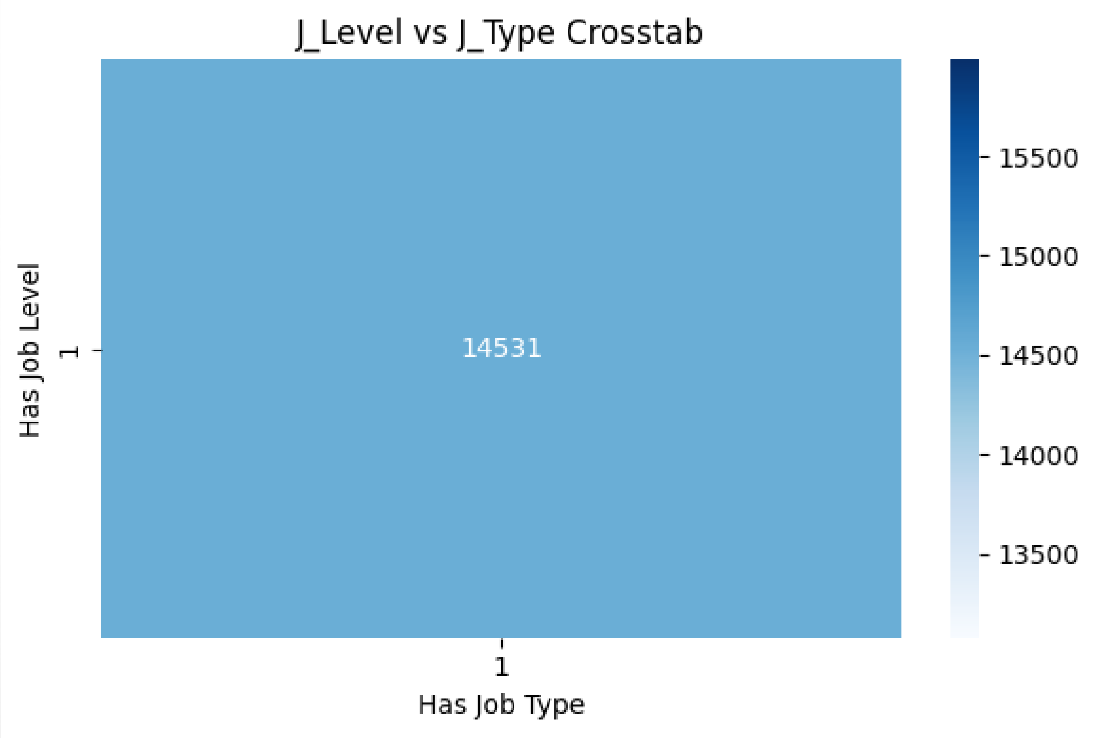


Almost every job posting has both skills and industry information

STAGE 4: DATA DRIVEN VALIDATION CHECKS

Base Causal Graph: Data Driven Validation Checks





Correlation is 0.14, states that there is essentially no relationship between job title length and industry presence.

- Long job titles do not make it more likely that the industry is present.
- Short job titles do not make it less likely.
- The two variables behave independently.

Almost every job posting has both job level and job type

STAGE 4: DATA DRIVEN VALIDATION CHECKS

Base Causal Graph: Conditional Independence Sanity Checks

Conditional Independence Sanity Checks

Raw correlation (Edu → Industry): -0.2504606492691898
Conditional correlations (Edu → Industry | Skills): [np.float64(nan), np.float64(-0.7610129888018429), np.float64(nan), np.float64(nan)]

1. Edu → Industry
Raw correlation:
-0.25
This is a weak negative correlation, meaning:
Users with more education are *slightly less likely* to have a specific industry label
This is plausible: education is broad, industry is specific
No contradiction to your causal assumption.

Conclusion:

- Causal assumption U_Education → U_Skills → U_Industry remains valid..
- Education does not directly determine industry — skills do.


```
Raw correlation (Edu → Industry): -0.2504606492691898
Conditional correlations (Edu → Industry | Skills): [np.float64(nan), np.float64(-0.7610129888018429), np.float64(nan), np.float64(nan)]
```

STAGE 4: DATA DRIVEN VALIDATION CHECKS

Base Causal Graph: Conditional Independence Sanity Checks

Conditional Independence Sanity Checks (Lightweight)

Raw correlation (J_Skills → J_Industry): 0.011271948131640655
Conditional correlations (J_Skills → J_Industry | J_Title): [np.float64(-0.018535498638570642), np.float64(0.008582886964183961), np.float64(0.05649893988537182), np.float64(-0.0008580828284087863)]

Raw correlation:
0.011
This is extremely close to zero.
Interpretation:
Skills and industry labels co-occur, but the *raw correlation* is weak
This is expected because:
Skills are multi-valued text fields
Industry is categorical
Correlation is a poor measure for sparse text-based features

Conclusion:
Your causal assumption J_Skills → J_Industry is supported.
The relationship is structural, not statistical — and that’s fine

STAGE 4: DATA DRIVEN VALIDATION CHECKS

Base Causal Graph: Conditional Independence Sanity Checks

4A. Compare parent–child correlations
User side
skill_len → industry_flag correlation: 0.05550303424958835

Job side
skill_len → industry_flag correlation: 0.011271948131640655
title_len → industry_flag correlation: 0.013608045878966694

All correlations are tiny:
0.055
0.011
0.013
Interpretation:
These proxies are not strong predictors of industry
This is expected
Industry is determined by semantic content, not text length
No contradictions appear
No evidence of reverse causality

Conclusion:
Causal edges are not contradicted by proxy correlations.

```
skill_len → industry_flag correlation: 0.05550303424958835
skill_len → industry_flag correlation: 0.011271948131640655
title_len → industry_flag correlation: 0.013608045878966694
```

STAGE 4: DATA DRIVEN VALIDATION CHECKS

Base Causal Graph: Conditional Independence Sanity Checks

4B. Compare alternative directions (to detect contradictions)
industry_flag → skill_len correlation: 0.0555030342495884
industry_flag → skill_len correlation: 0.011271948131640657

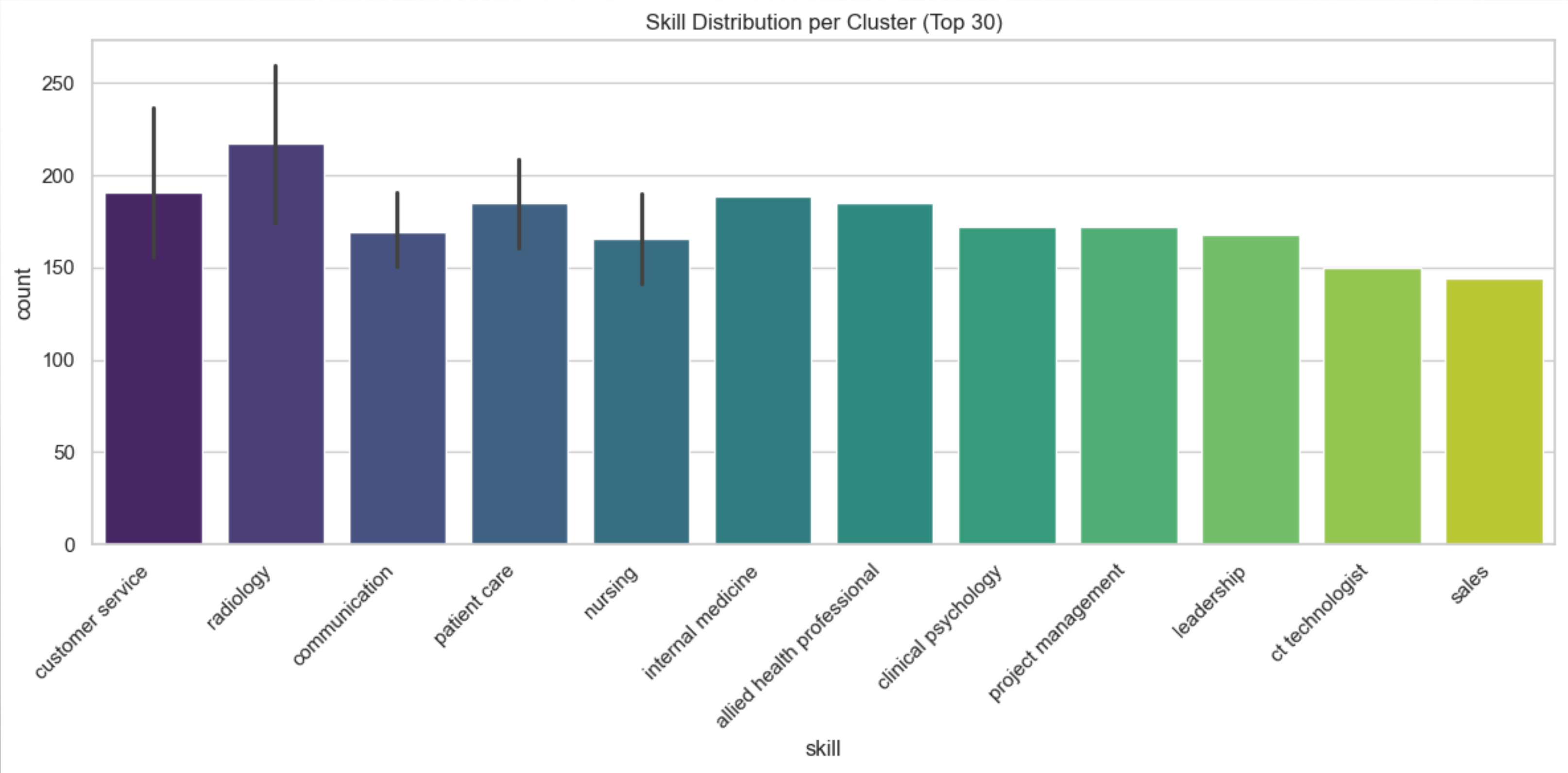
```
industry_flag → skill_len correlation: 0.0555030342495884
industry_flag → skill_len correlation: 0.011271948131640657
```

compared:
industry_flag → skill_len
industry_flag → job_skill presence
Both correlations are identical to the forward direction.
Interpretation:
This means the relationship is not directional in raw correlation space
This is normal
Correlation cannot detect causal direction
No evidence of reverse causality
No contradictions

Conclusion:
Causal assumptions remain intact.

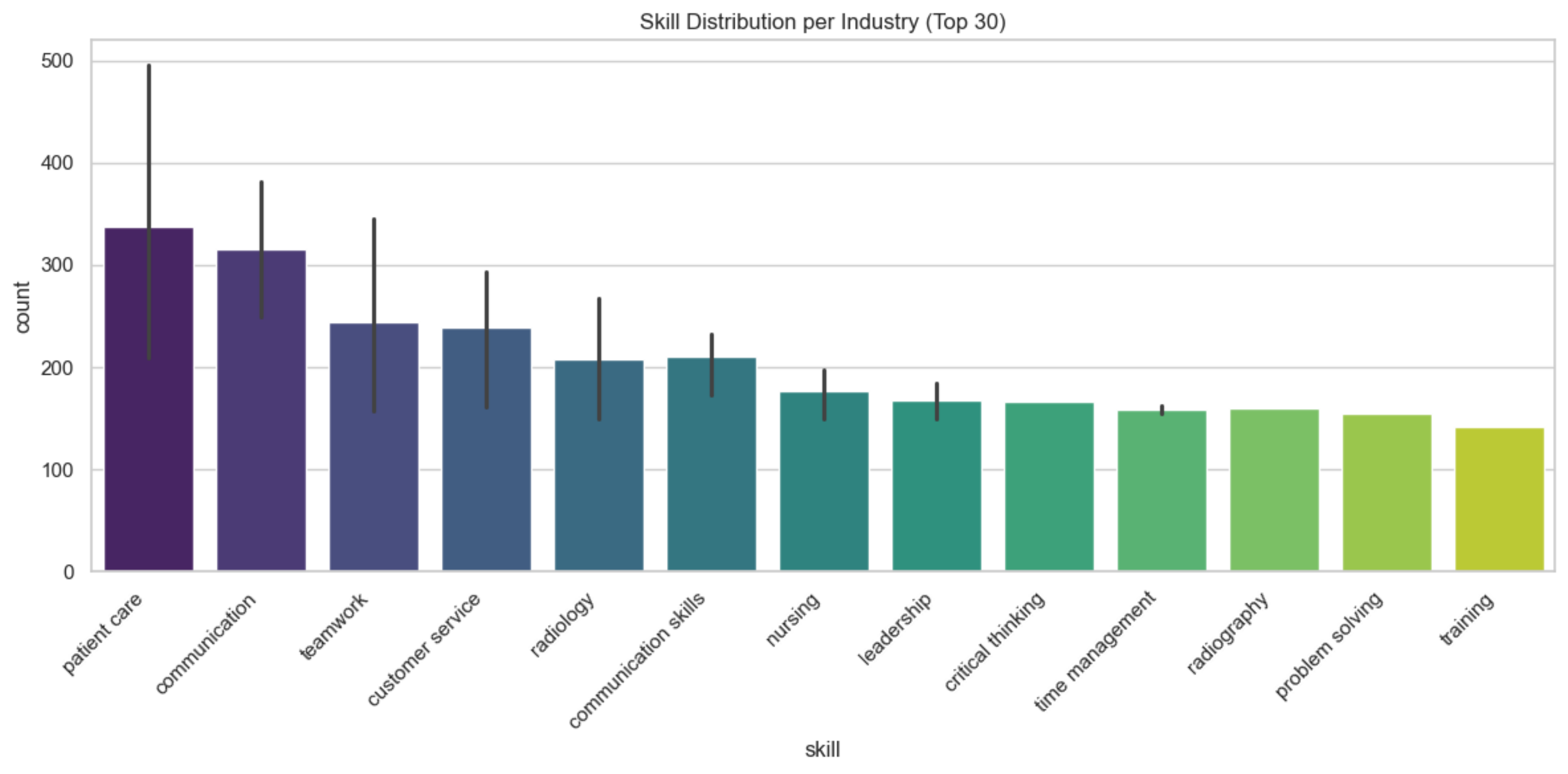
STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

```
=== Loading Distribution CSVs ===

skill_dist_per_cluster.csv columns: ['cluster_id', 'skill', 'count']
industry_dist_per_cluster.csv columns: ['cluster_id', 'industry_final', 'count']
skill_dist_per_industry.csv columns: ['industry_final', 'skill', 'count']
```



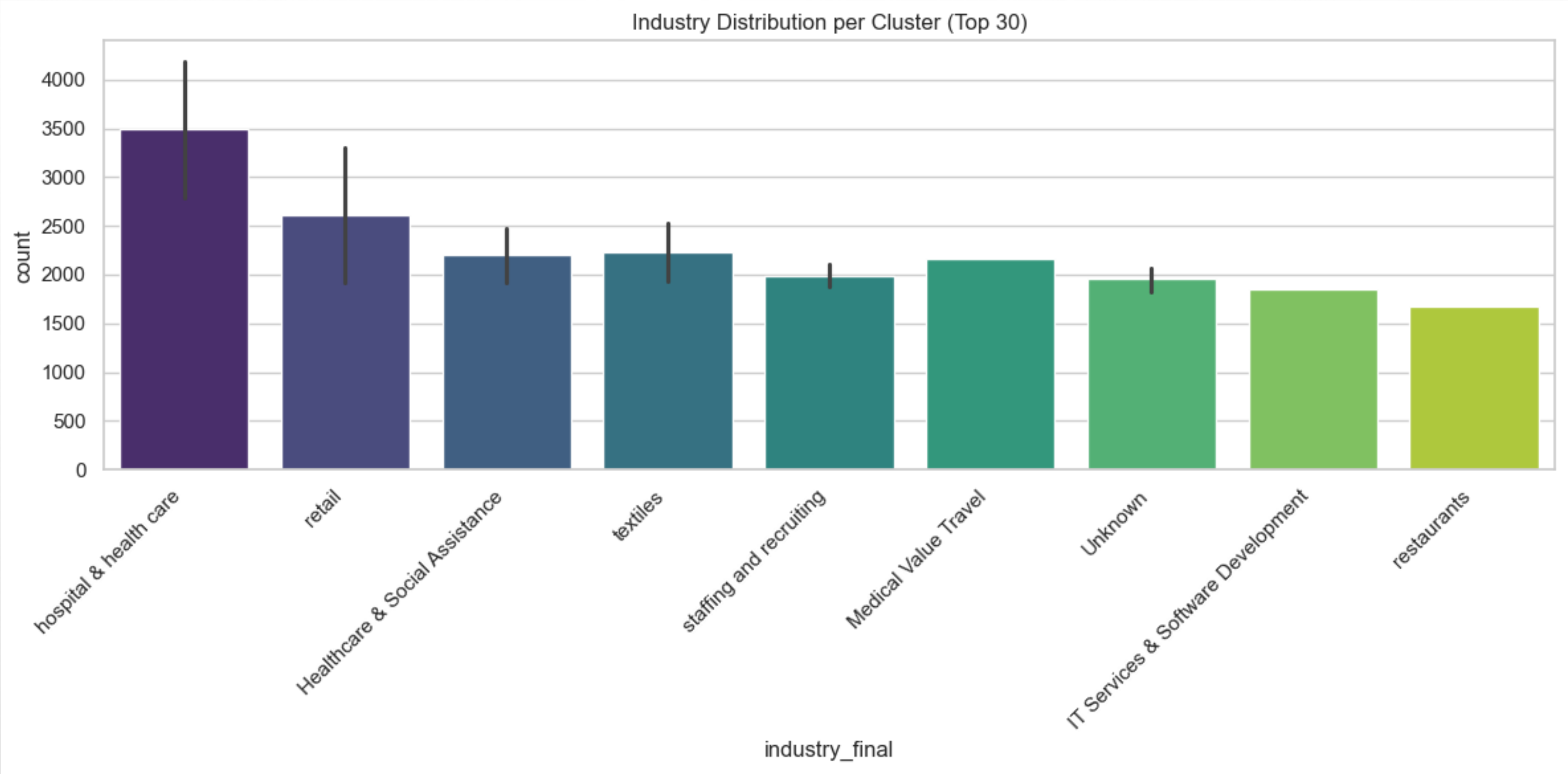
STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

```
=== Loading Distribution CSVs ===

skill_dist_per_cluster.csv columns: ['cluster_id', 'skill', 'count']
industry_dist_per_cluster.csv columns: ['cluster_id', 'industry_final', 'count']
skill_dist_per_industry.csv columns: ['industry_final', 'skill', 'count']
```

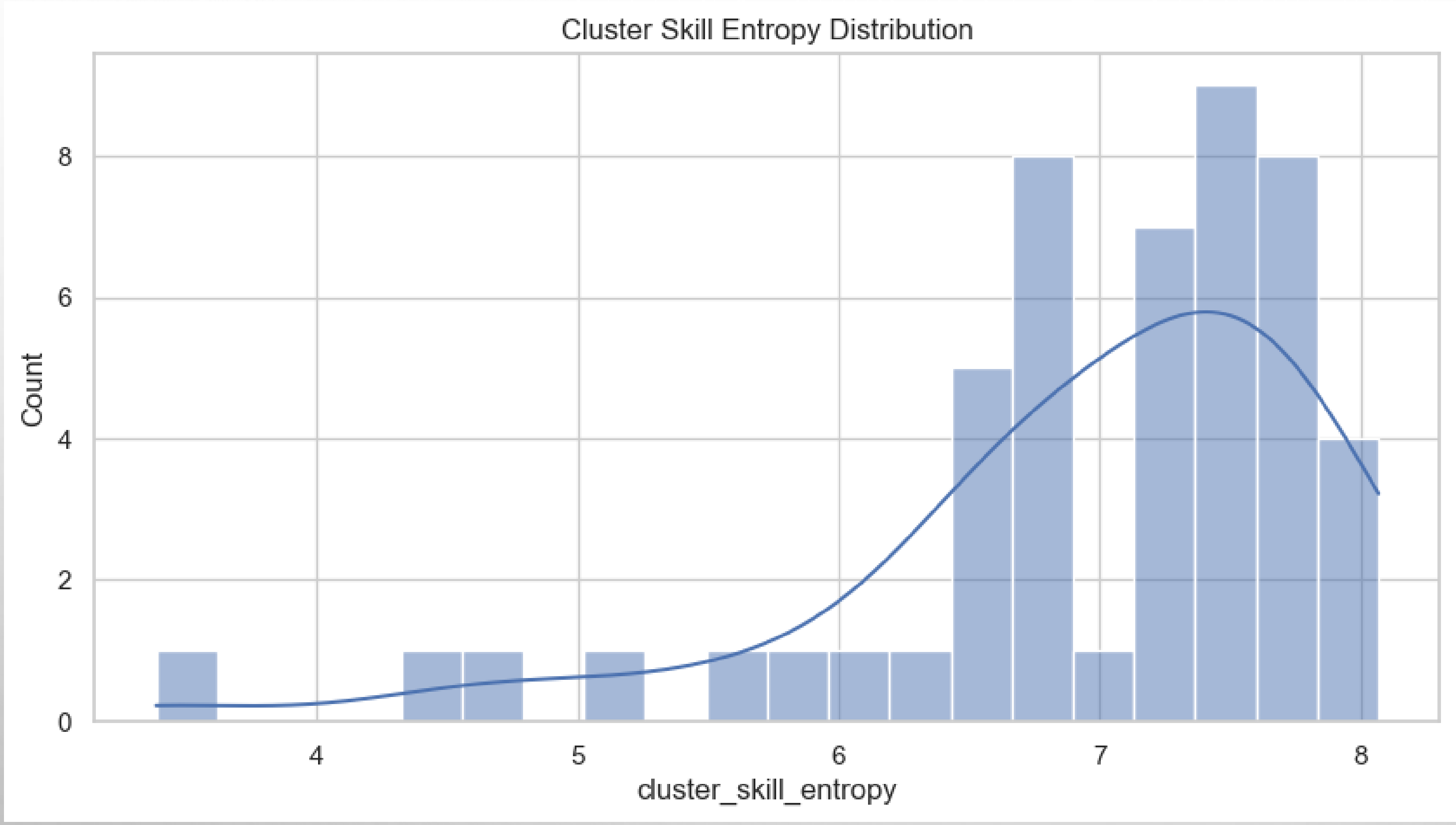


STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

```
=== Loading Distribution CSVs ===

skill_dist_per_cluster.csv columns: ['cluster_id', 'skill', 'count']
industry_dist_per_cluster.csv columns: ['cluster_id', 'industry_final', 'count']
skill_dist_per_industry.csv columns: ['industry_final', 'skill', 'count']
```

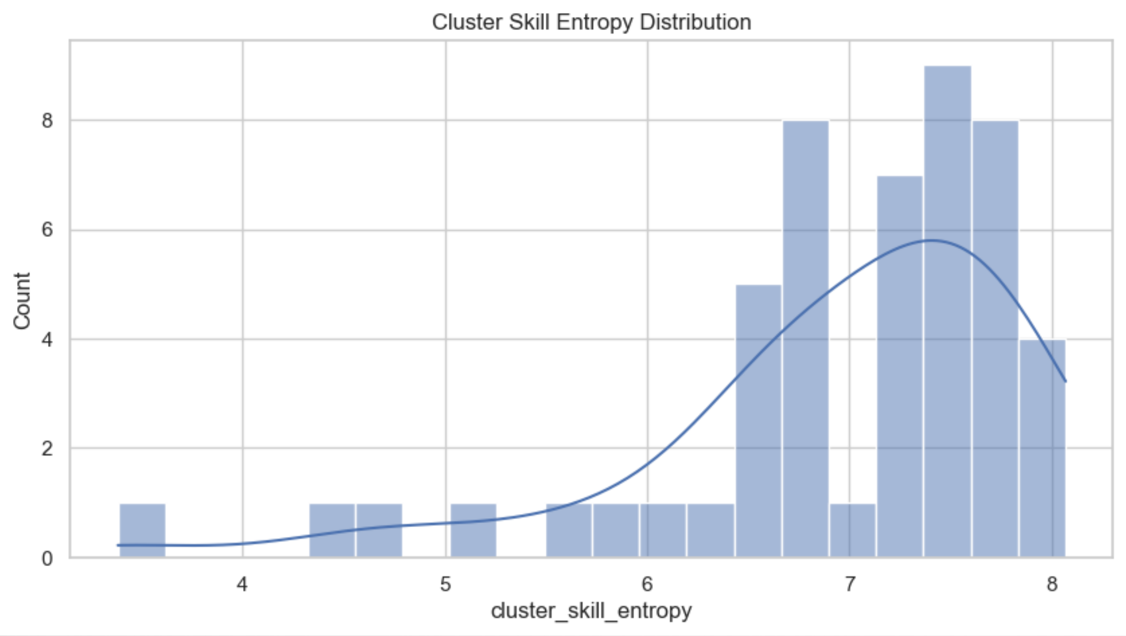


STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

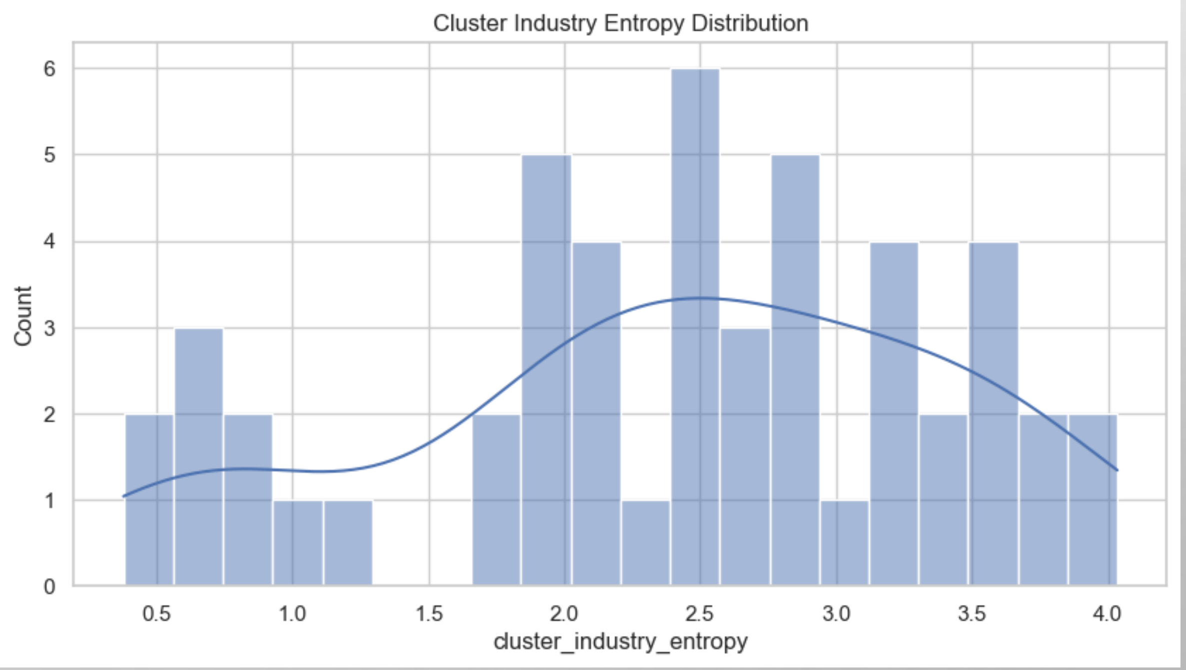



```
Entropy stats:
       cluster_skill_entropy
count              50.000000
mean                6.948421
std                 0.958944
min                 3.384281
25%                 6.652700
50%                 7.213691
75%                 7.575775
max                 8.069483
```

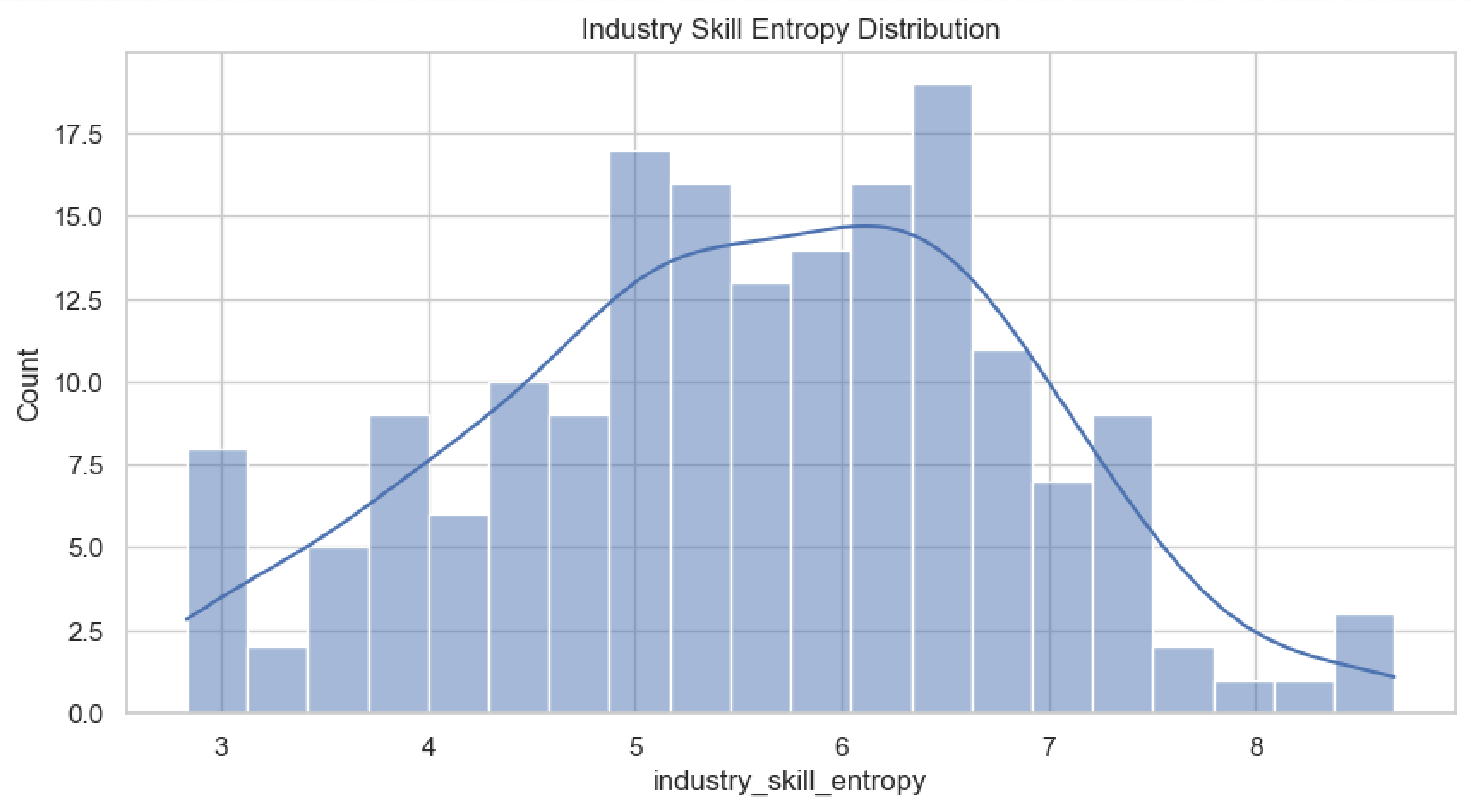
STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION




```
Entropy stats:
       cluster_skill_entropy  cluster_industry_entropy
count              50.000000                 50.000000
mean                6.948421                  2.407438
std                 0.958944                  1.005835
min                 3.384281                  0.379861
25%                 6.652700                  1.909152
50%                 7.213691                  2.507962
75%                 7.575775                  3.260190
max                 8.069483                  4.033156
```

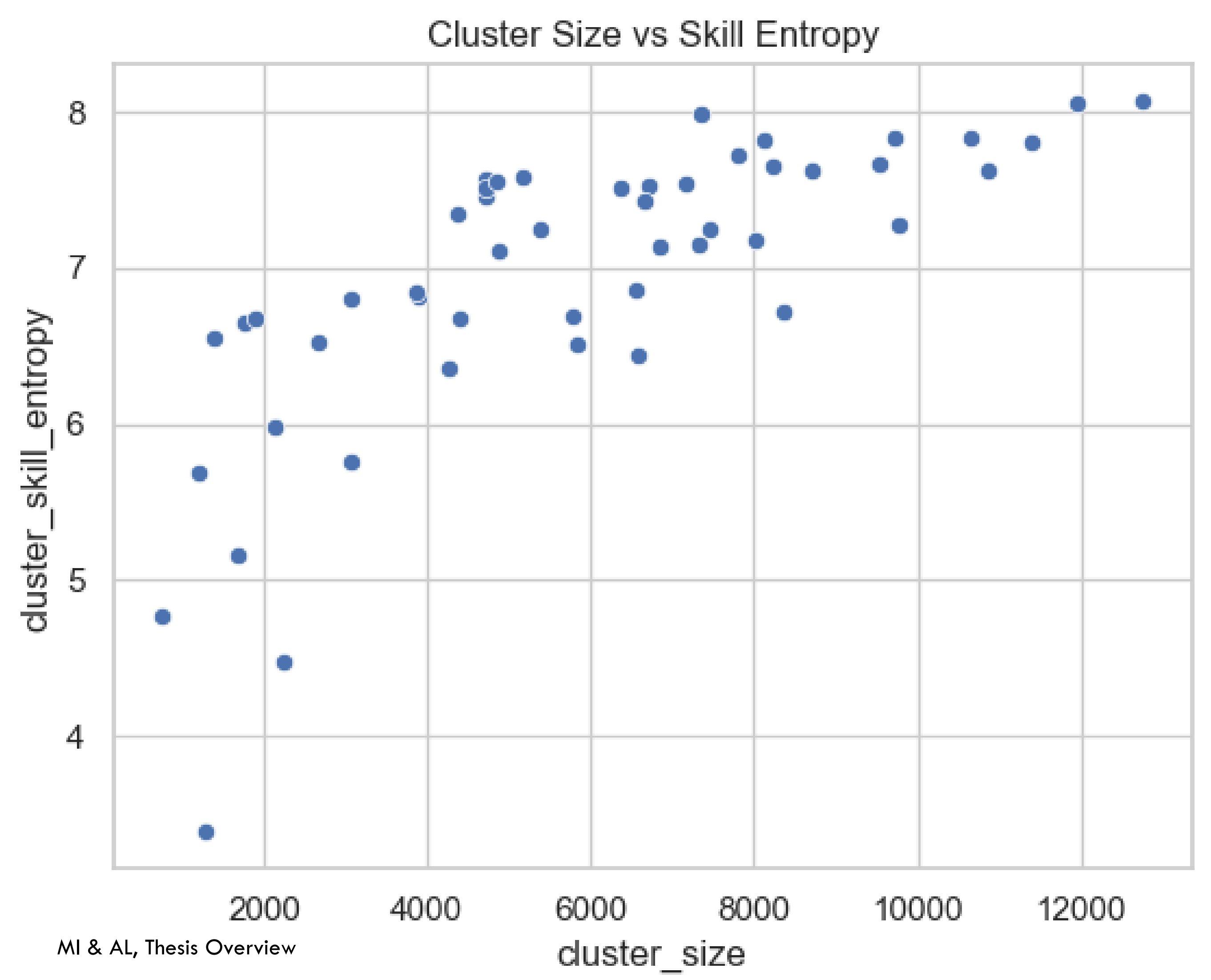


STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

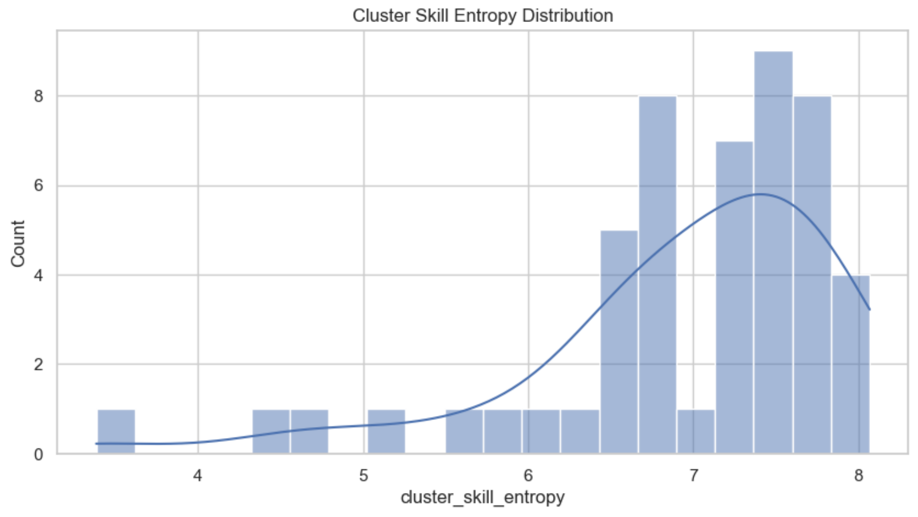


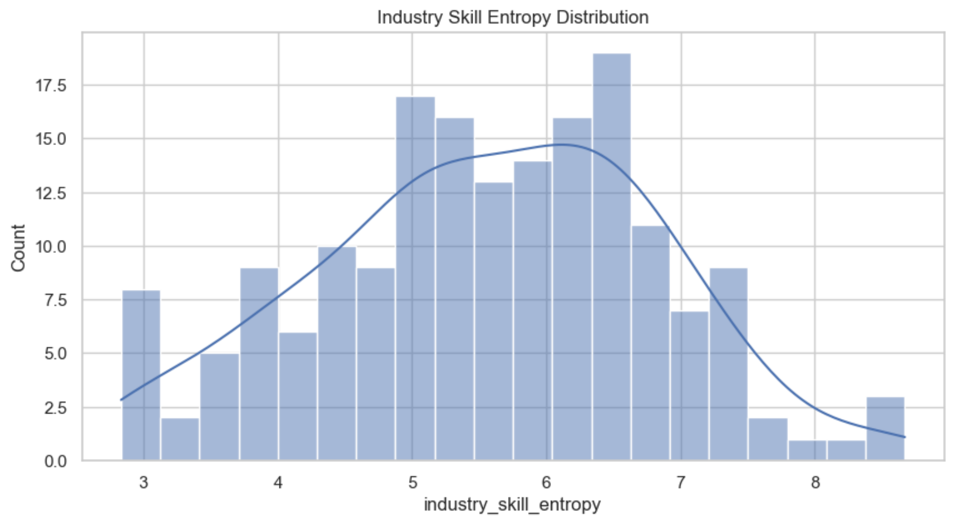

```
Entropy stats:
count    178.000000
mean       5.567012
std        1.269929
min        2.833739
25%        4.831886
50%        5.644053
75%        6.492425
max        8.671781
Name: industry_skill_entropy,
```

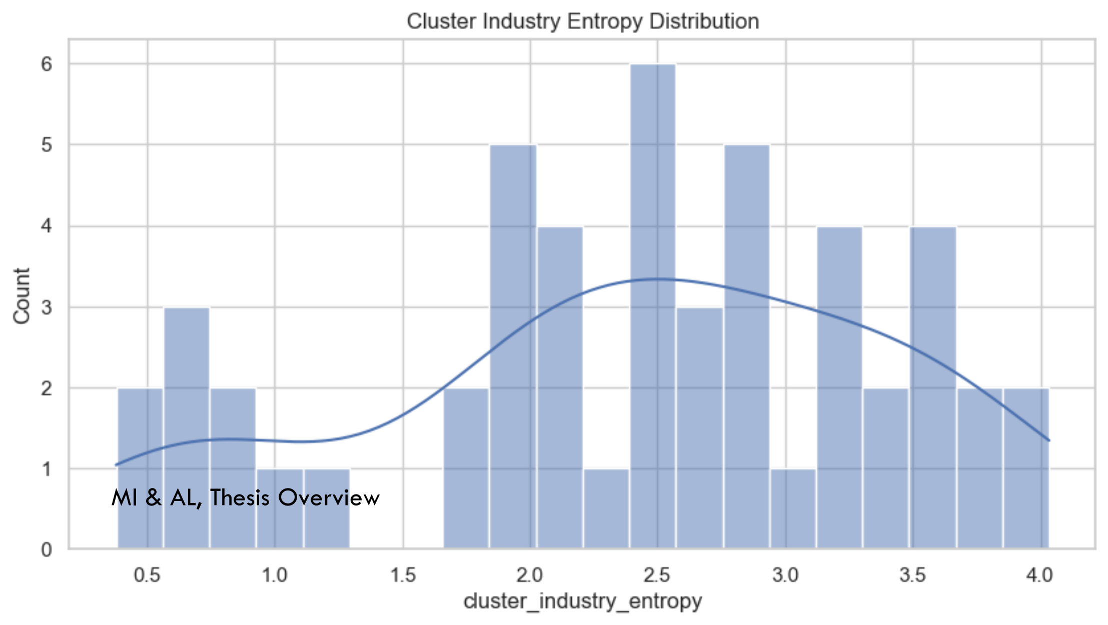
STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION

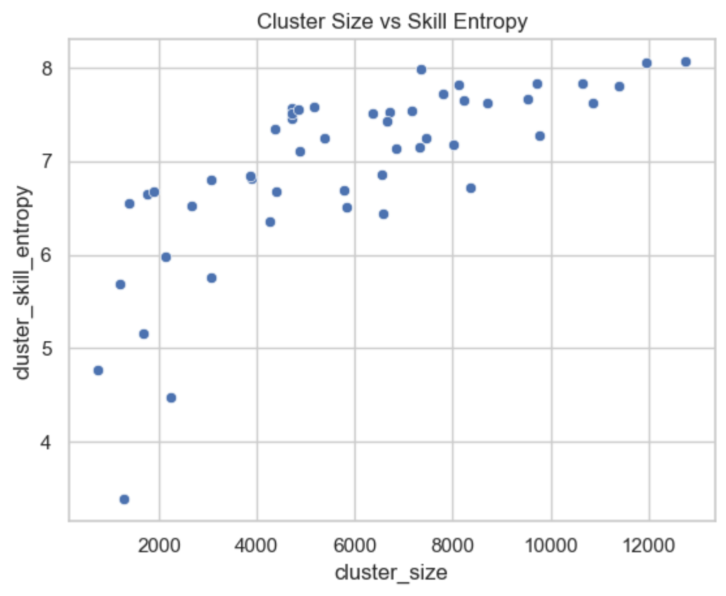


STAGE 4: HYBRID CAUSAL GRAPH CONSTRUCTION



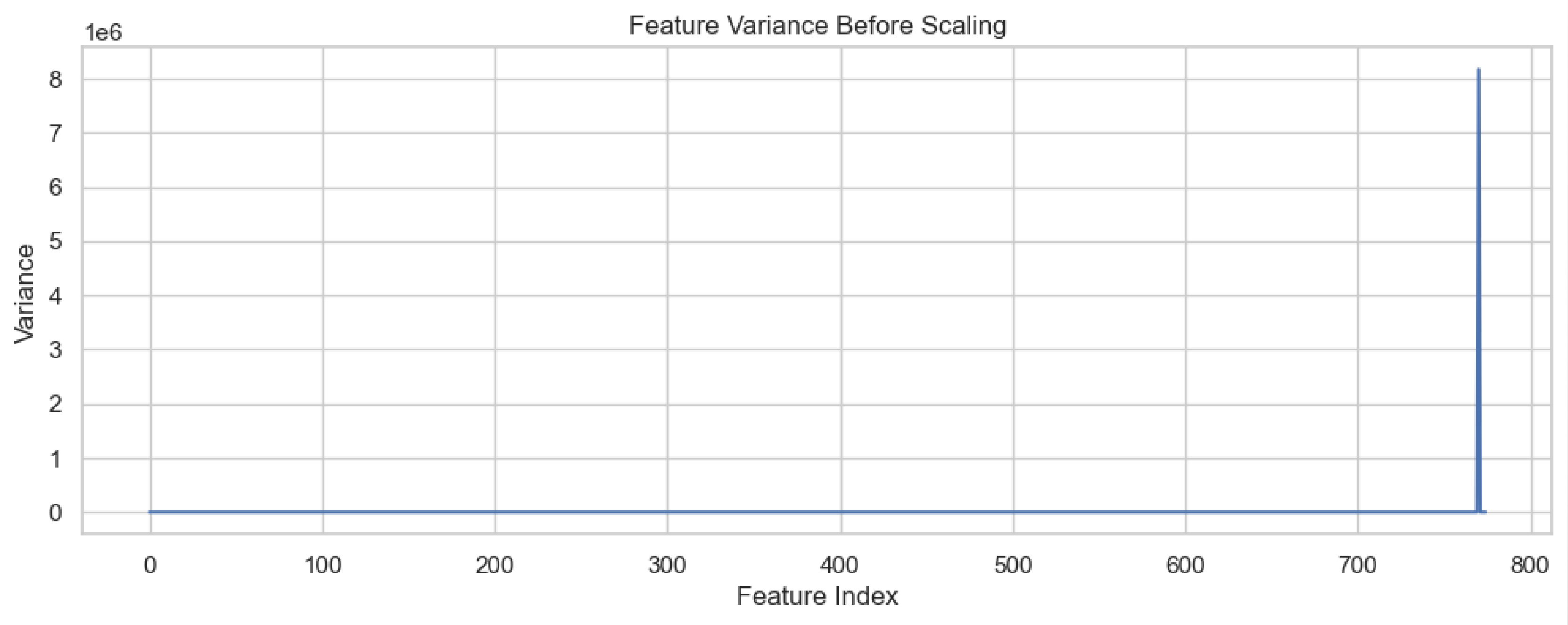






STAGE 5: VARIATIONAL GENERATIVE MODELLING

```
=== INPUT MATRIX SUMMARY ===
Total samples: 14531
Total numeric features: 775
First 5 numeric columns: ['job_domain', 'job_cluster_id', 'job_emb_0', 'job_emb_1', 'job_emb_2']
```

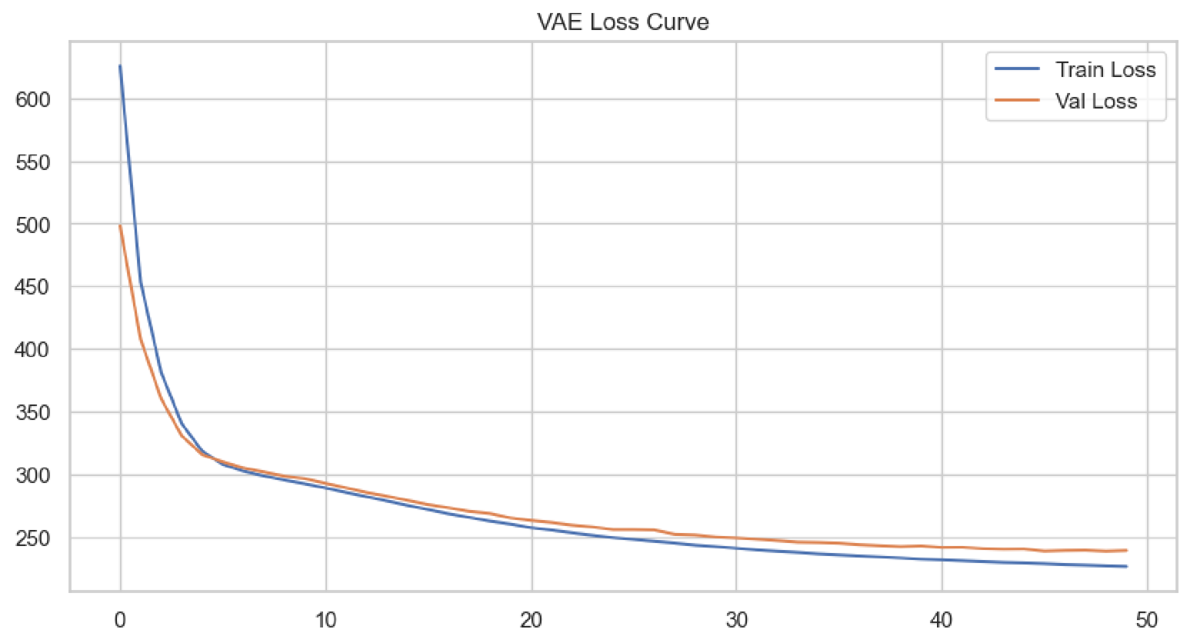


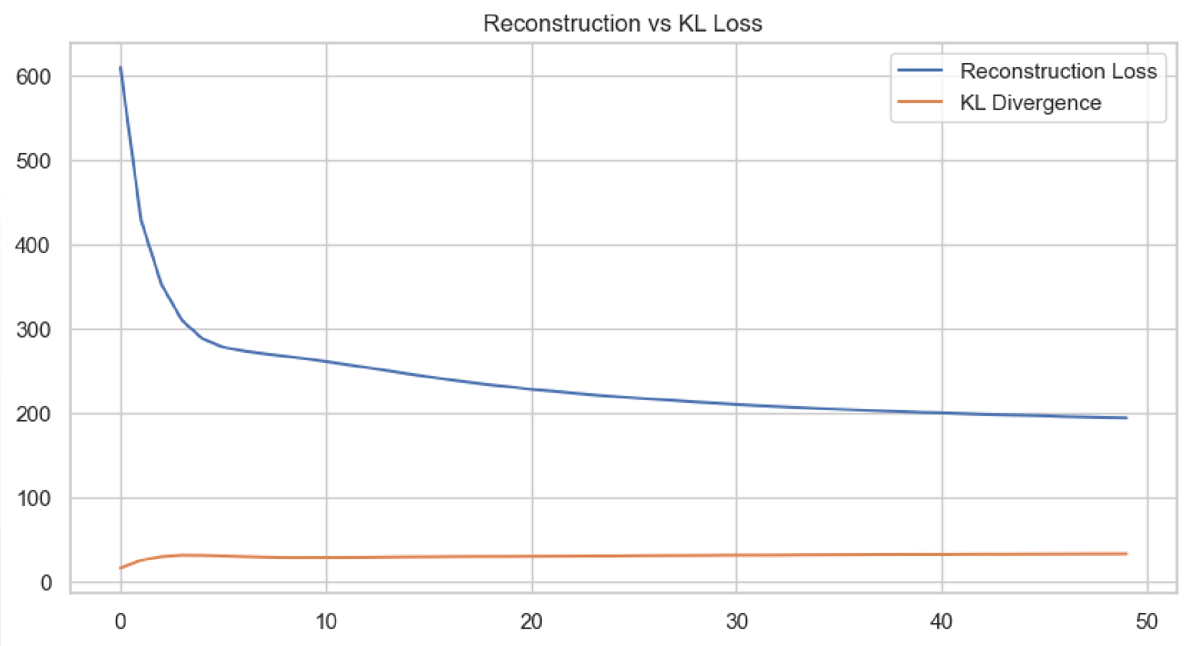
STAGE 5: VARIATIONAL GENERATIVE MODELLING

VAE Model Summary

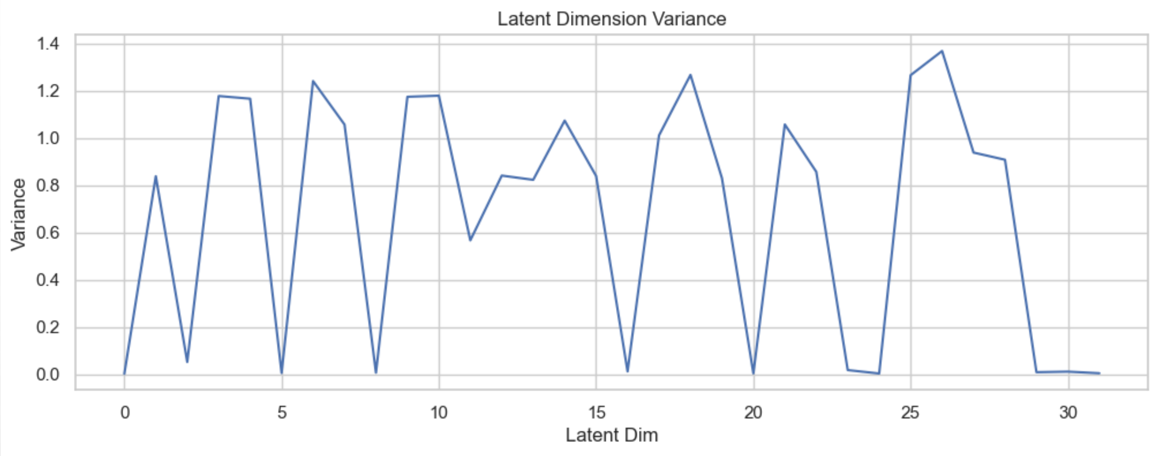
```
=== VAE MODEL SUMMARY ===
VAE(
  (encoder): Sequential(
    (0): Linear(in_features=775, out_features=256, bias=True)
    (1): ReLU()
    (2): Linear(in_features=256, out_features=128, bias=True)
    (3): ReLU()
  )
  (mu): Linear(in_features=128, out_features=32, bias=True)
  (logvar): Linear(in_features=128, out_features=32, bias=True)
  (decoder): Sequential(
    (0): Linear(in_features=32, out_features=128, bias=True)
    (1): ReLU()
    (2): Linear(in_features=128, out_features=256, bias=True)
    (3): ReLU()
    (4): Linear(in_features=256, out_features=775, bias=True)
  )
)
Total trainable parameters: 476231
```

STAGE 5: VARIATIONAL GENERATIVE MODELLING

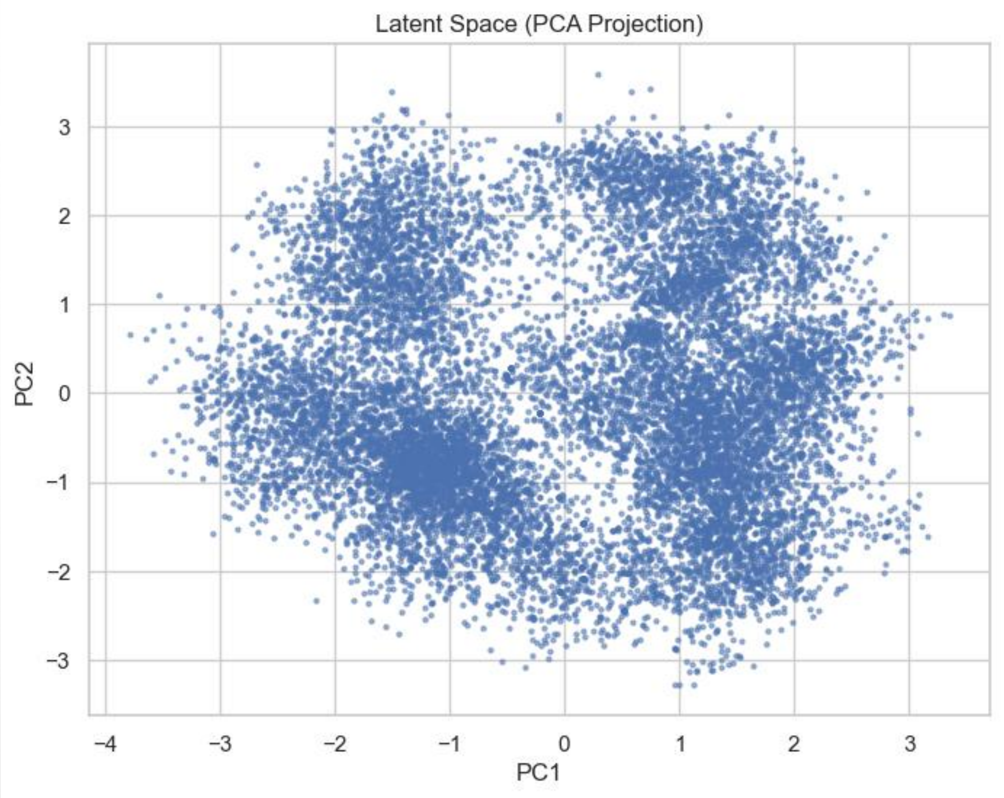




STAGE 5: VARIATIONAL GENERATIVE MODELLING

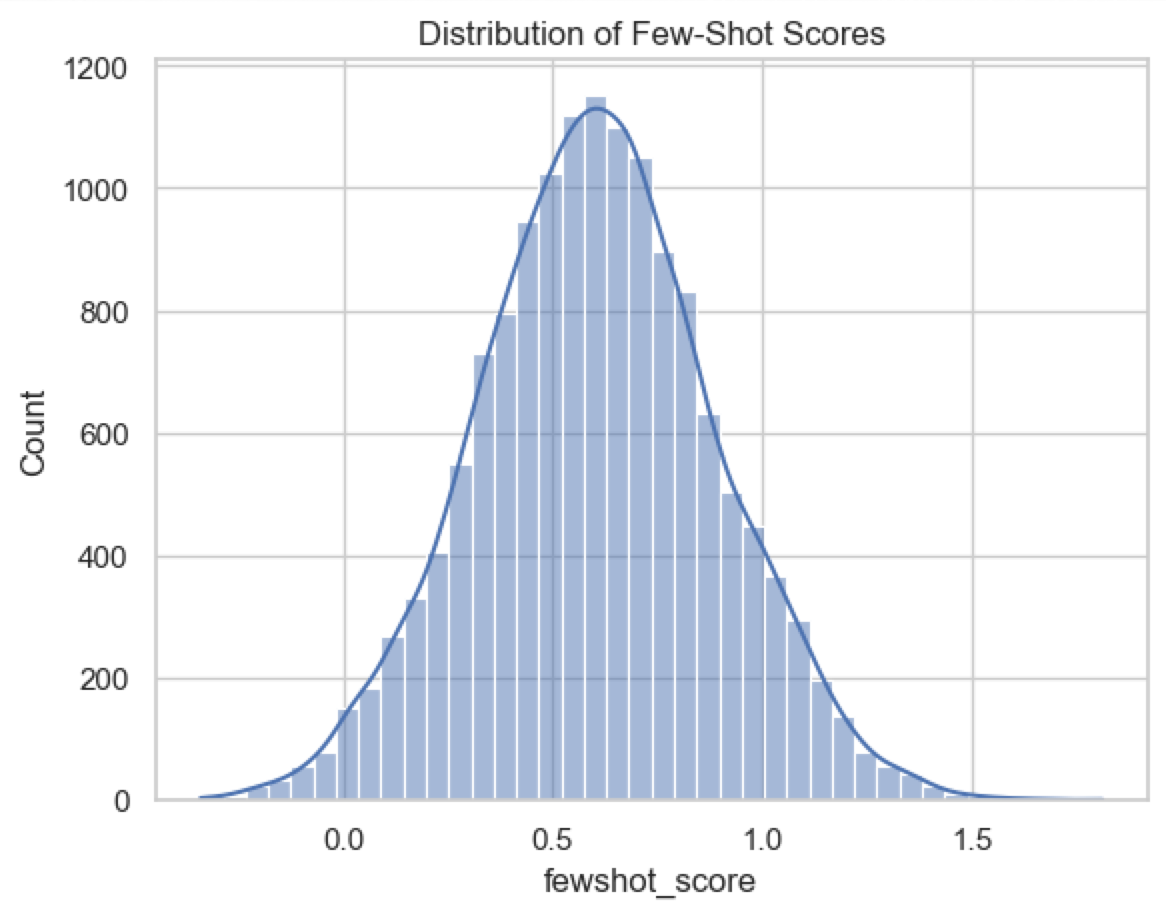



```
=== LATENT SPACE STATISTICS ===
Mean per latent dim: [0.01543725 0.04322451 0.04789021 0.07515111 0.09359407]
Std per latent dim: [0.05563346 0.9155516  0.23036788 1.0850041  1.0797887 ]
```



STAGE 6: ADAPTIVE FEW-SHOT LEARNING

Adaptive Few Shot Learning

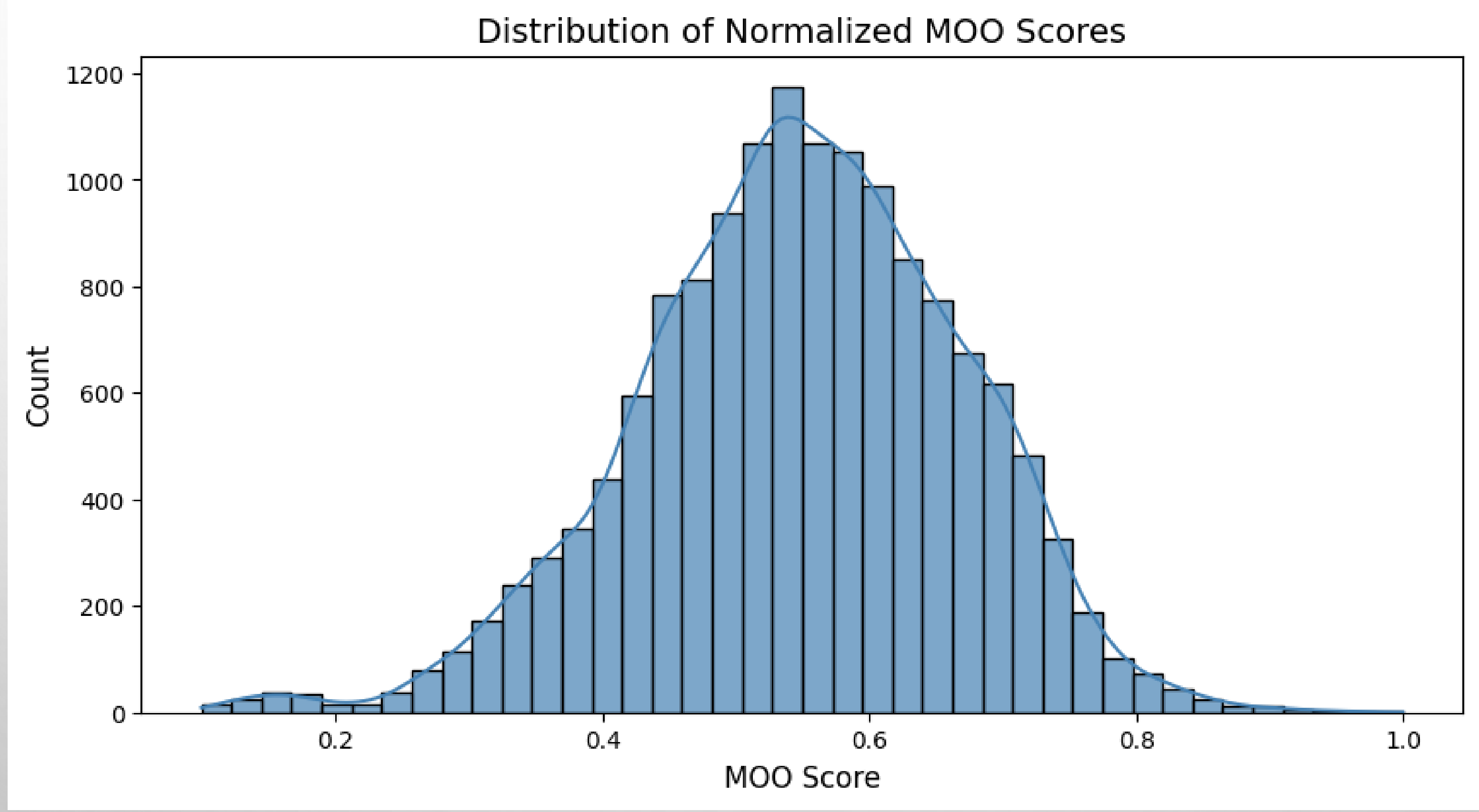



```
count    14531.000000
mean         0.602803
std          0.283723
min         -0.339553
25%          0.413879
50%          0.601197
75%          0.788834
max          1.809343
Name: fewshot_score, dt
```

```
Top 10 recommended jobs (by few-shot score):
          job_id  fewshot_score
729    Job_00730       1.809343
6862   Job_06863       1.665442
6208   Job_06209       1.652084
12779  Job_12780       1.590849
4963   Job_04964       1.588666
1110   Job_01111       1.576231
13592  Job_13593       1.555475
14173  Job_14174       1.535203
8410   Job_08411       1.524452
10368  Job_10369       1.511242

Bottom 10 jobs:
         job_id  fewshot_score
5981  Job_05982      -0.339553
7230  Job_07231      -0.339553
7999  Job_08000      -0.326014
2437  Job_02438      -0.301450
5994  Job_05995      -0.301450
1417  Job_01418      -0.280299
4640  Job_04641      -0.276539
5214  Job_05215      -0.269241
7762  Job_07763      -0.265131
4701  Job_04702      -0.261722
```

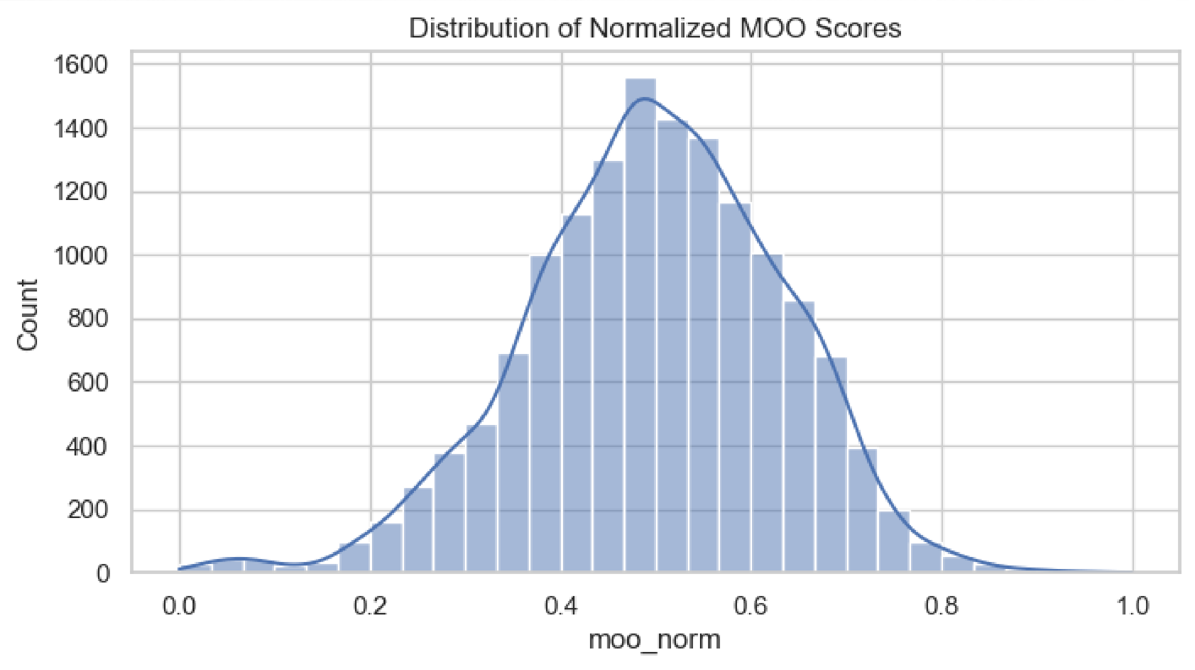
STAGE 7: MULTI OBJECTIVE OPTIMIZATION

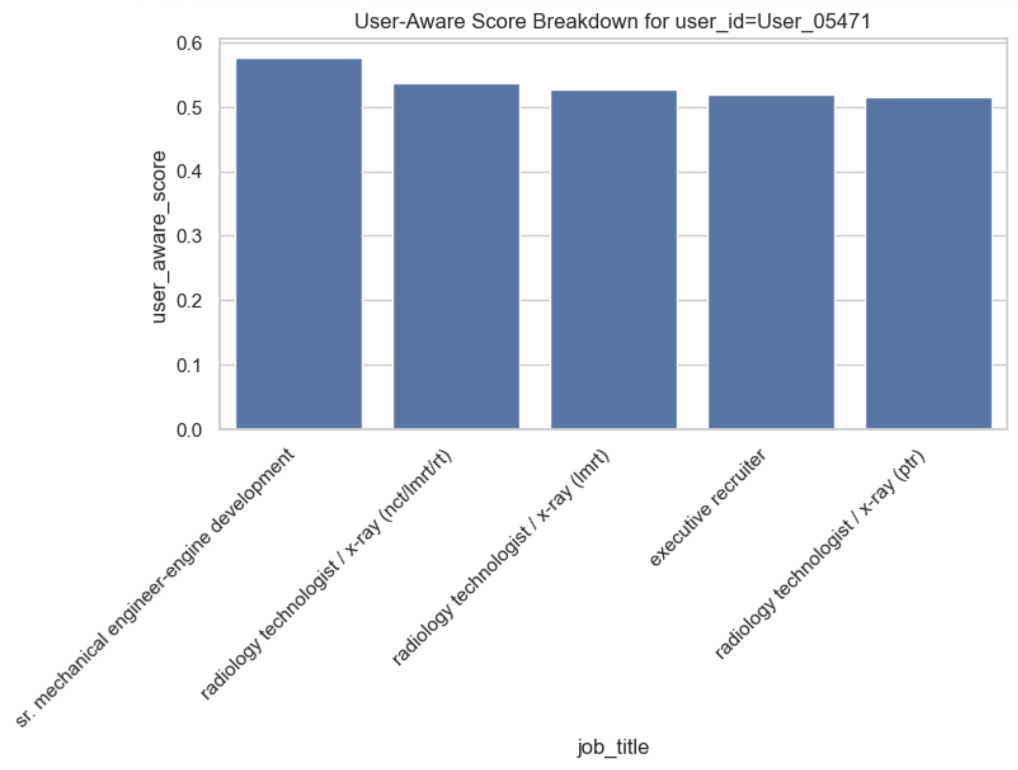
Multi Objective Optimization, final scoring layer of CGGA



STAGE 8: RECOMMENDATION ENGINE

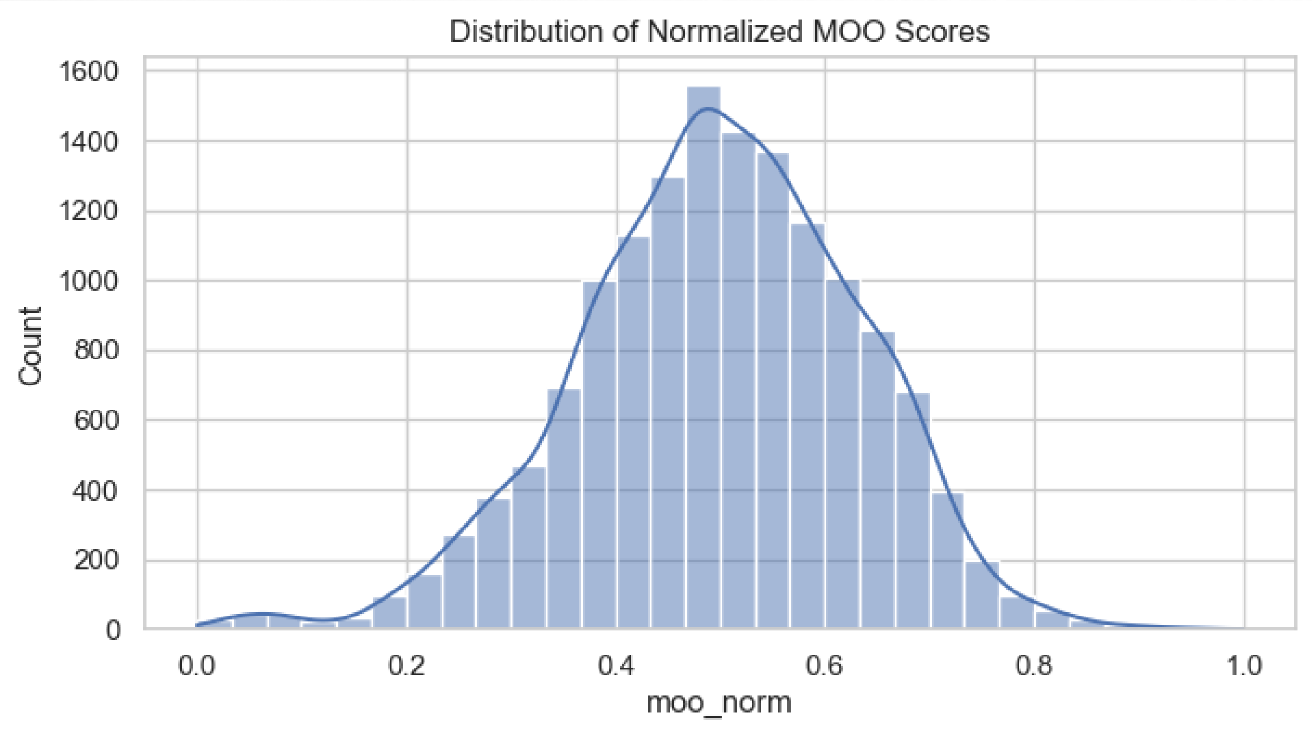
Method 1





STAGE 8: RECOMMENDATION ENGINE

Method 2




```
=== Recommendations for user_id=User_05471 ===
         job_id                                   job_title
14463  Job_14464      sr. mechanical engineer-engine development
9269   Job_09270     radiology technologist / x-ray (nct/lmrt/rt)
14173  Job_14174  sr. mechanical engineer (spacecraft structures)
7655   Job_07656                       senior systems engineer **
2106   Job_02107         staff systems engineer advanced programs
```

STAGE 9: GROUND TRUTH CREATION

Load the data

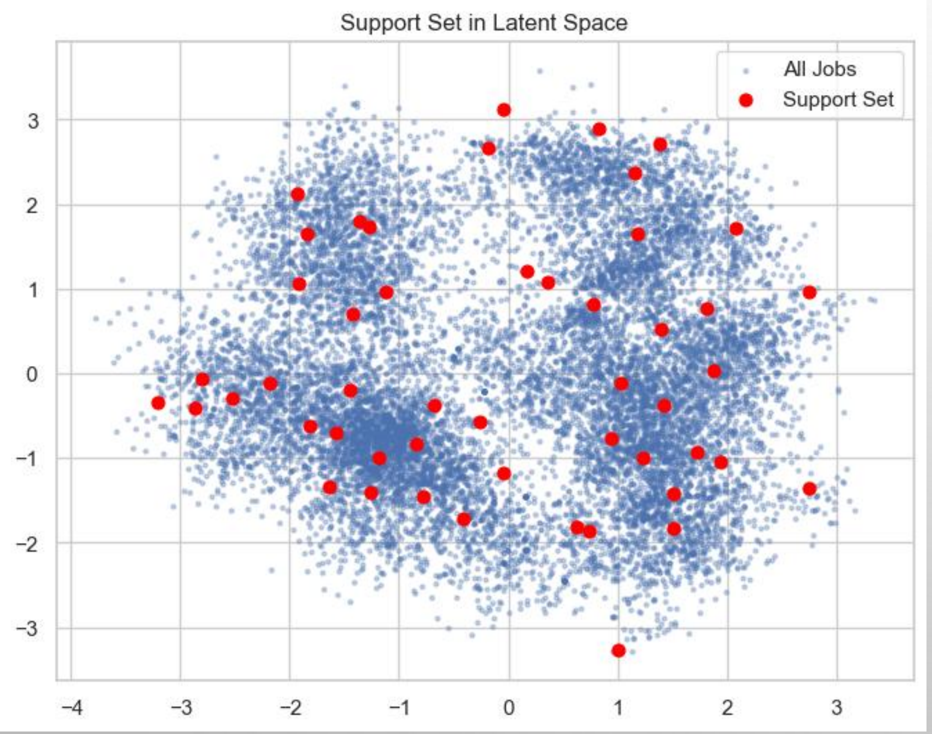
Selected 100 sampled users (Support set).
Load datasets
users_with_text_embeddings.csv
hybrid_job_features_with_latent_fewshot_moo.csv
user_embeddings.npy
job_embeddings.npy

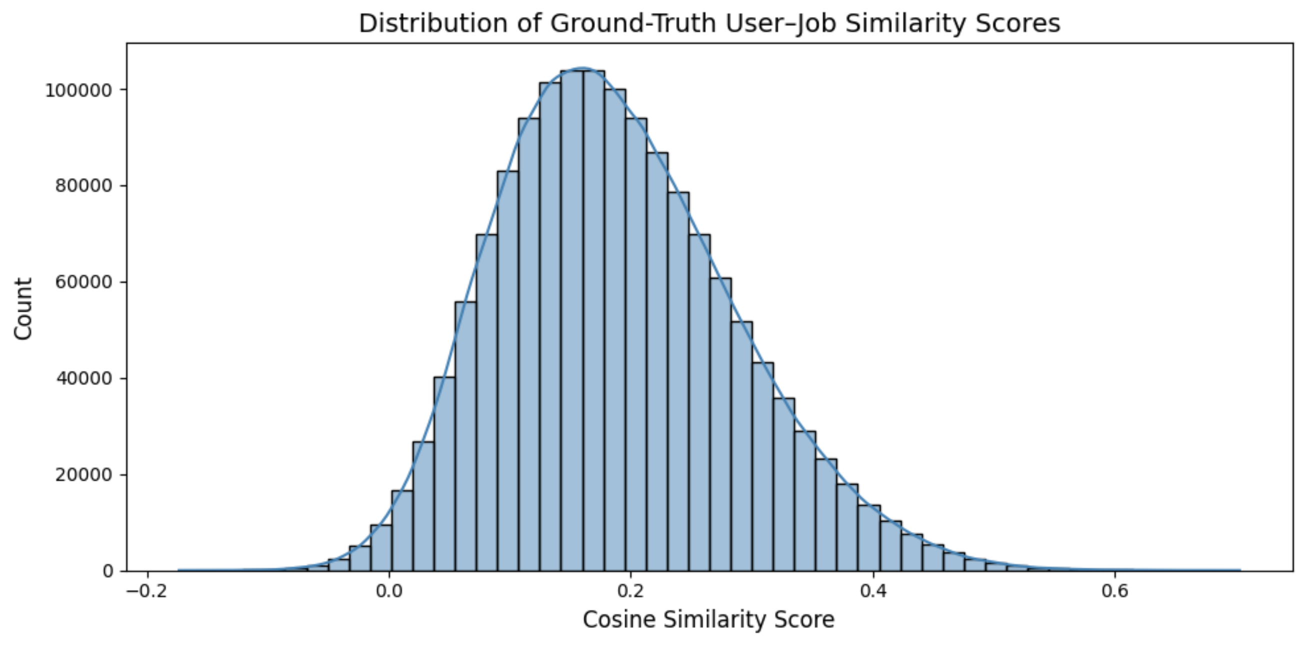
Processing

Applied Co-sine Similarity function
Generate few shot scores for all the sampled 100 users against each Job ID

Outputs (Ground Truth)

fewshot_user_job_scores_sampled_100.csv





STAGE 9: GROUND TRUTH CREATION

Load the data

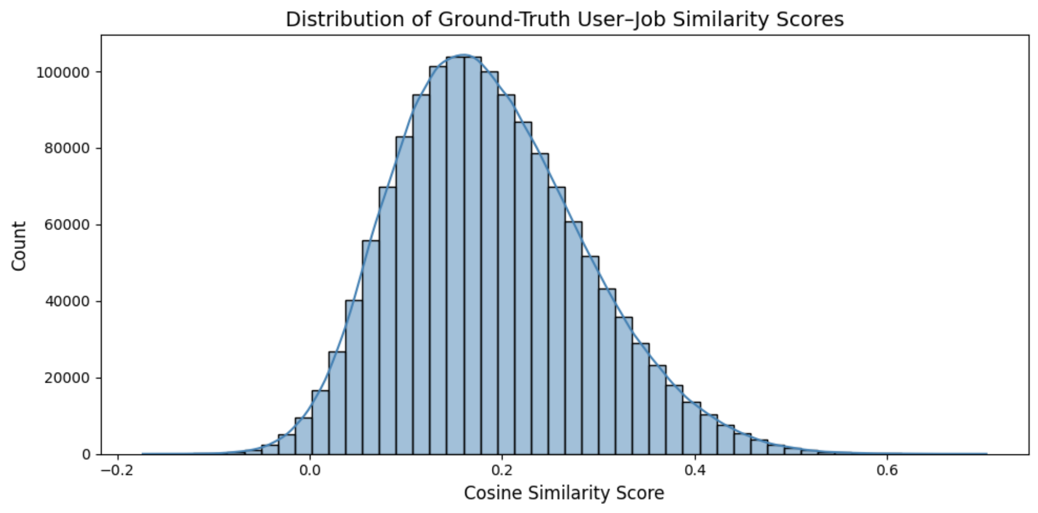
Selected 100 sampled users (Support set).
Load datasets
users_with_text_embeddings.csv
hybrid_job_features_with_latent_fewshot_moo.csv
user_embeddings.npy
job_embeddings.npy

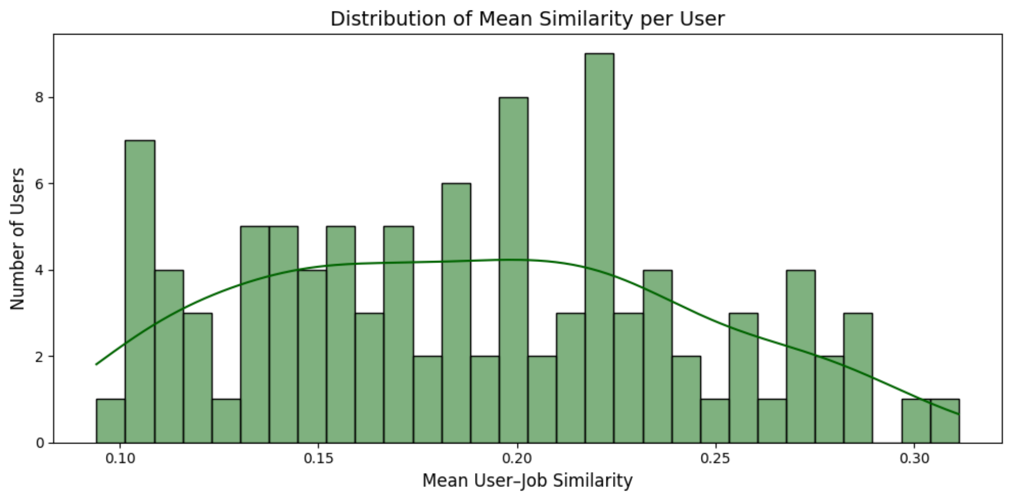
Processing

Applied Co-sine Similarity function
Generate few shot scores for all the sampled 100 users against each Job ID

Outputs (Ground Truth)

fewshot_user_job_scores_sampled_100.csv





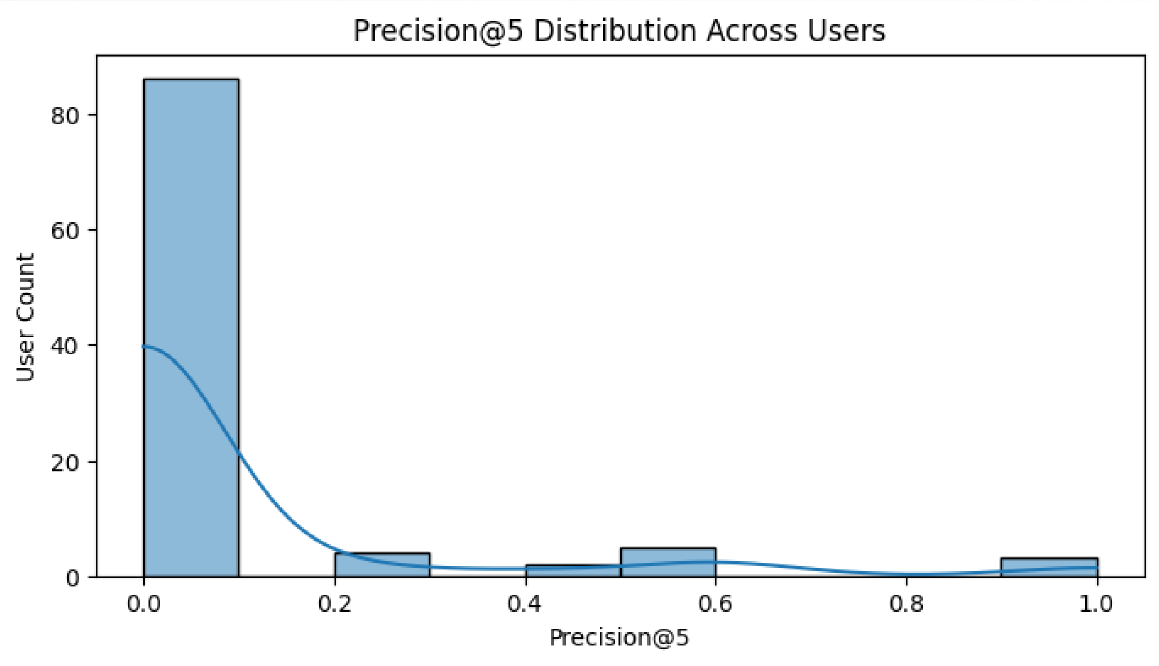
STAGE 9: EVALUATION / METRICS CALCULATIONS

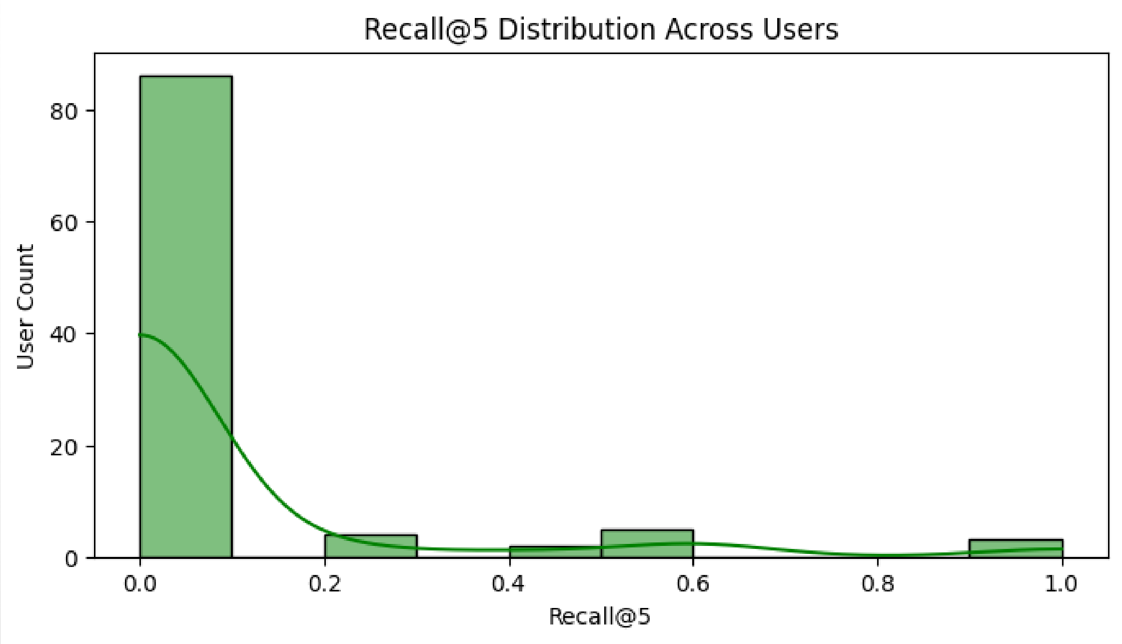
Model Evaluation

```
Loaded predictions: (500, 4)
Loaded ground truth: (1453100, 6)

==================== EVALUATION SUMMARY ====================
                score
precision@5  0.038000
recall@5     0.038000
hit_rate@5   0.140000
mrr@5        0.107000
ndcg@5       0.078499
ild@5        0.507997
============================================================

Saved: evaluation_metrics_per_user.csv
Saved: evaluation_summary.csv
```

STAGE 9: METRIC DISTRIBUTION CALCULATION

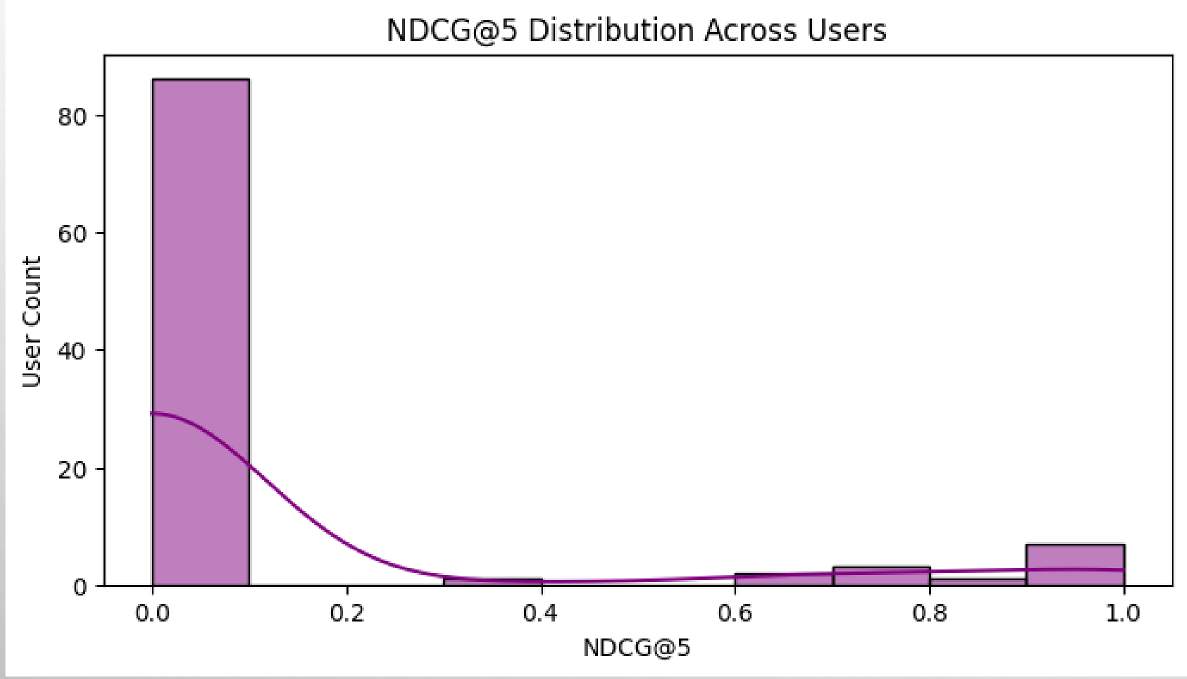
Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data

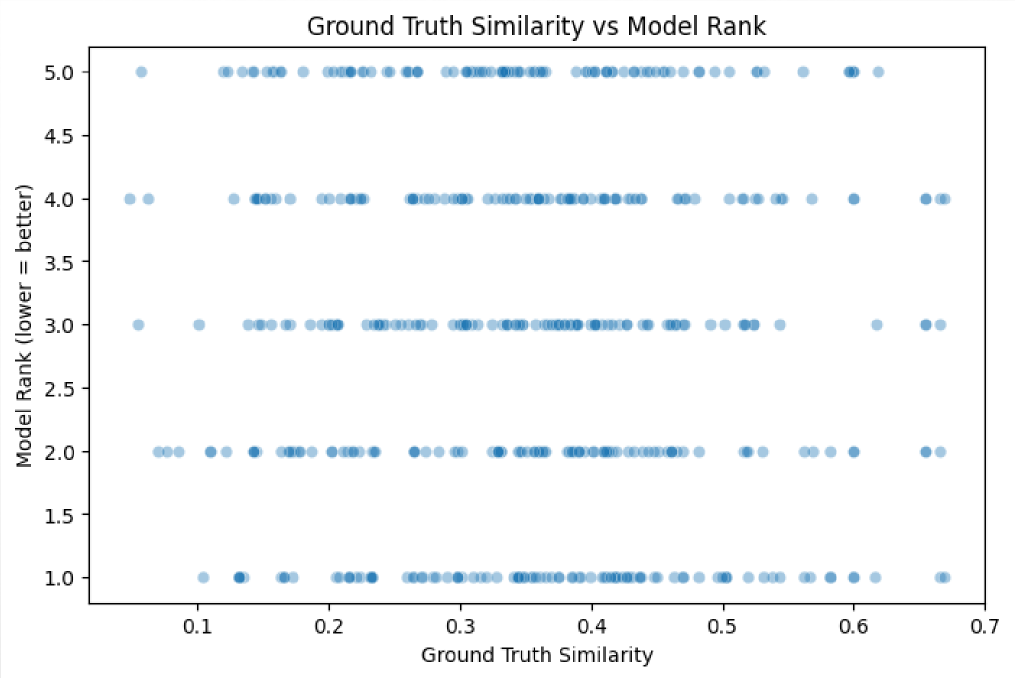




STAGE 9: METRIC DISTRIBUTION CALCULATION

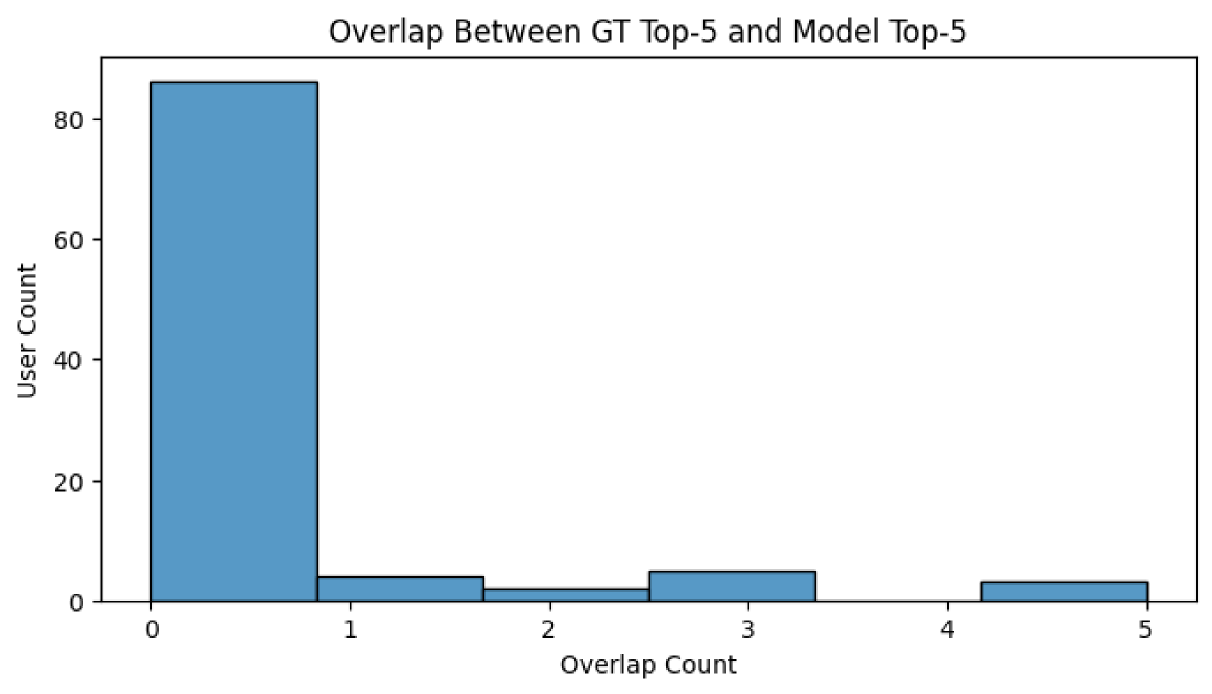
Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data

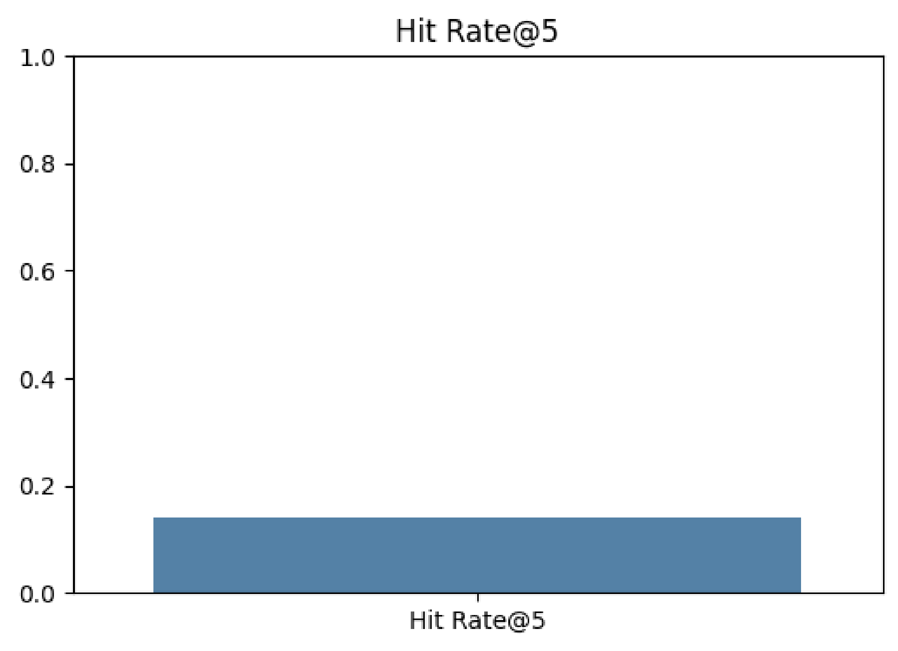




STAGE 9: METRIC DISTRIBUTION CALCULATION

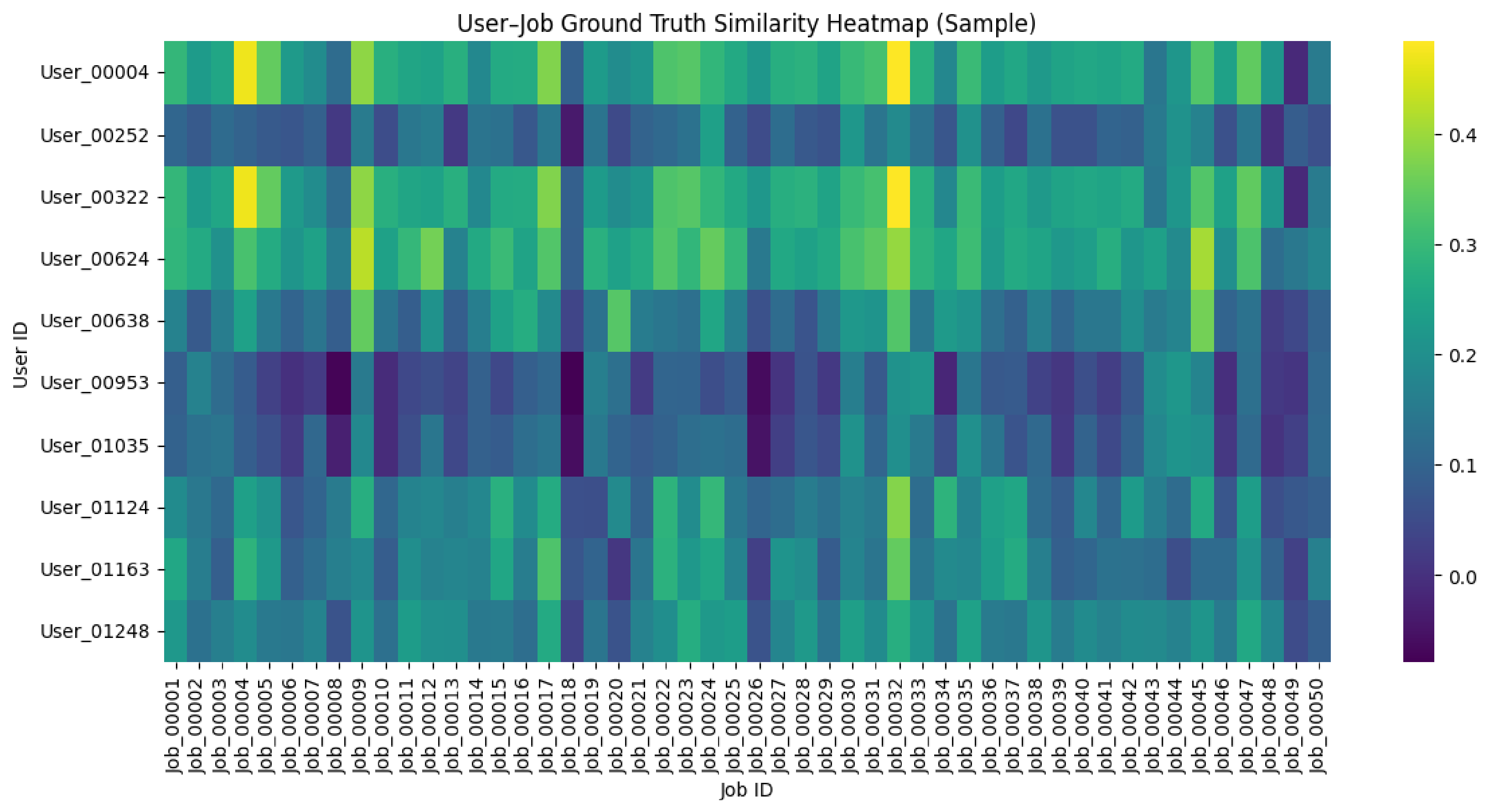
Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data





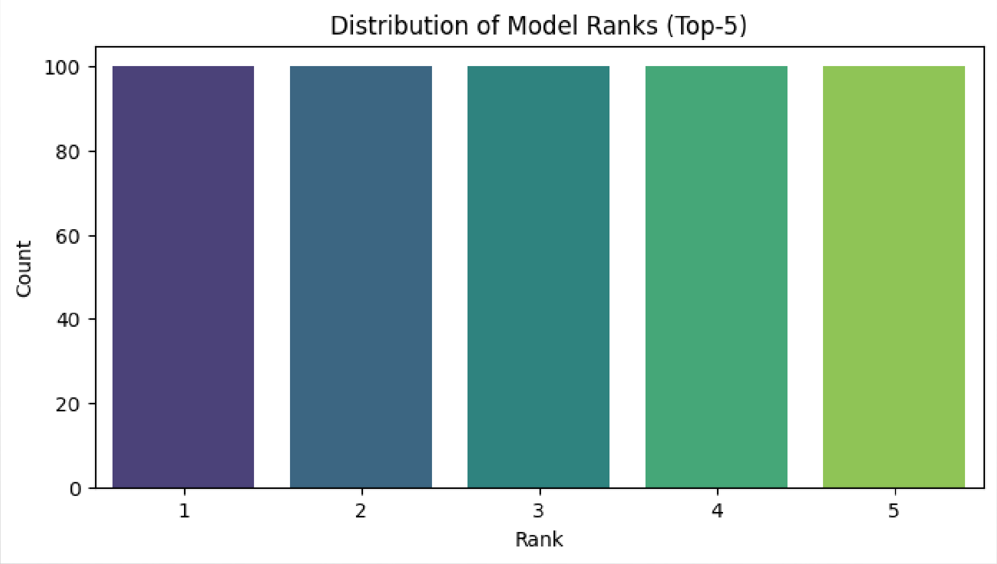
STAGE 9: METRIC DISTRIBUTION CALCULATION

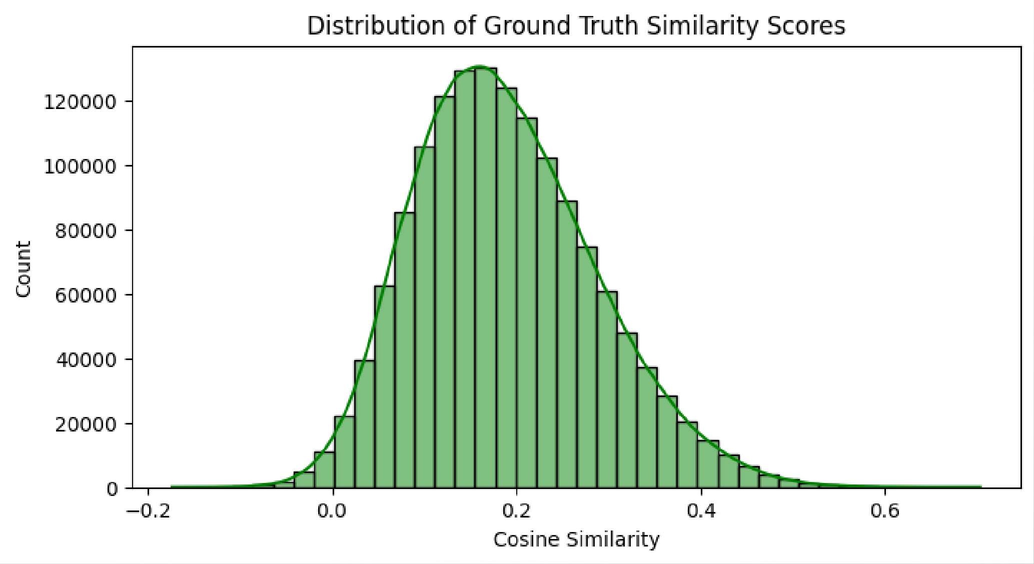
Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data



STAGE 9: METRIC DISTRIBUTION CALCULATION

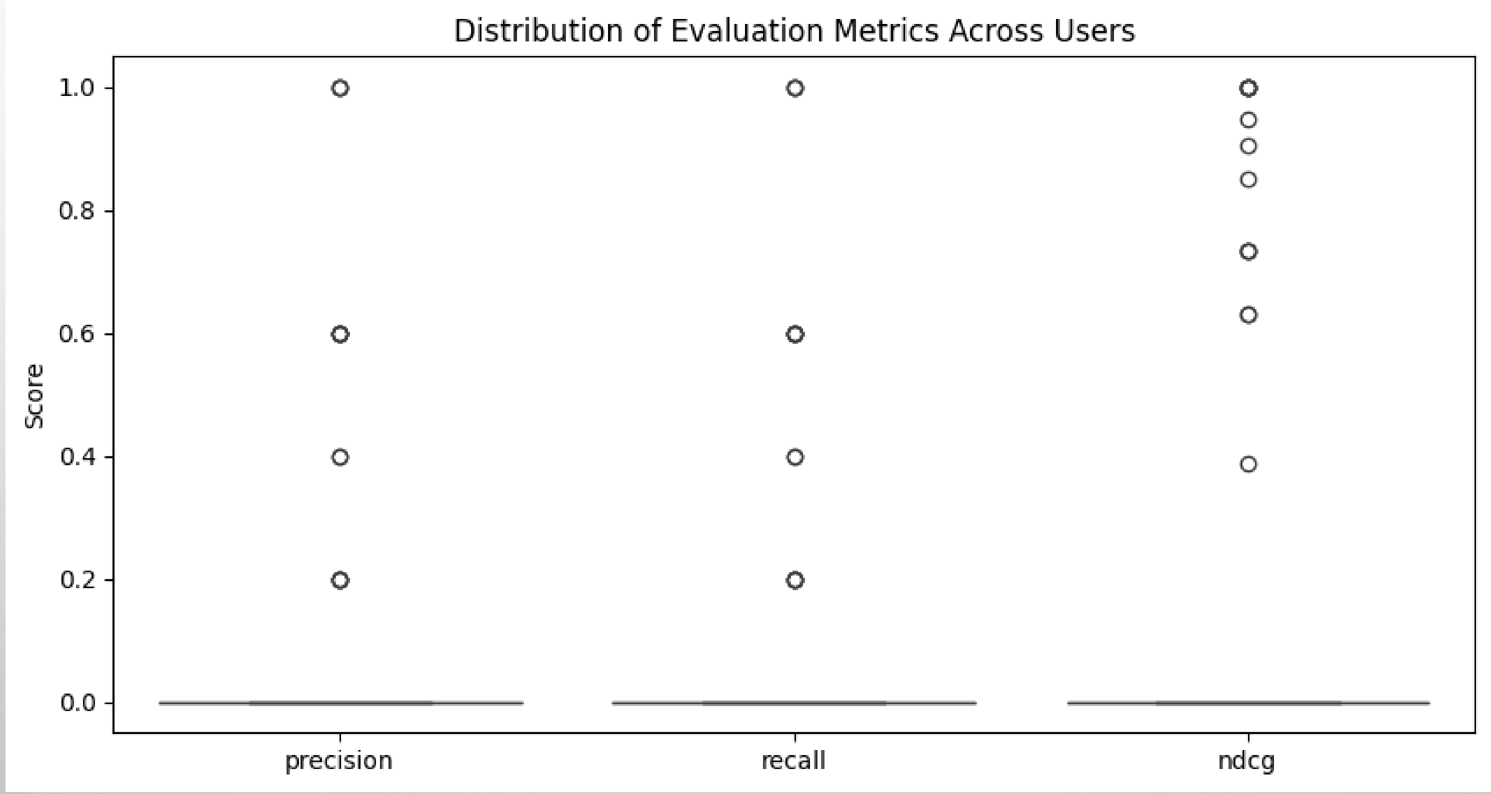
Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data





STAGE 9: METRIC DISTRIBUTION CALCULATION

Computing the metrics for all the 100 users, who got top 5 job recommendations, in comparing with Ground truth reference data



ANALYSIS & OBSERVATIONS

The model shows weak alignment with the ground-truth Top-5 relevance defined by few-shot similarity. Most users receive zero relevant recommendations in their Top-5, and the overlap analysis confirms that the model's Top-5 rarely matches the ground-truth Top-5.

A small subset of users achieve partial or perfect matches, indicating that the model captures meaningful patterns for certain user profiles, but the behavior is inconsistent across the population.

Overall, the results suggest a retrieval mismatch between the model's ranking logic and the few-shot similarity ground truth, likely due to differences in embedding spaces or user/job representation