SkillMatch

Two Stage Skill-Aware Matching System

Match candidates to jobs they're actually qualified for.

SKILL - MATCH PIPELINE

STAGE 1: RETRIEVAL	STAGE 2 : RE-RANKING	STAGE 3 :OUTPUT
SBERT	LogReg + 8 features	Skill Gap Diagnosis

Problem Statement

The Job Seeker Problem

- Candidates apply to a large number of roles with limited visibility into their actual suitability
- There is no clear understanding of:
  - alignment with job requirements
  - reasons for rejection
  - specific skill gaps that need to be addressed
- Existing hiring processes provide minimal or no structured feedback

System Failure: logical explanation

47.5% of top job matches are actually relevant according to Gold standard

↓

the rest are misleading false positives

Similarity ≠ qualification

Missing skills are ignored

We don't just produce a score, we tell candidates which skills to learn next.

Literature Review — Paper Deep Dive

AI Skill Gap Analyzer (Dash et al., 2025)

International Journal of Engineering Development and Research (IJEDR), 2025

What They Did

- Normalises skills across industries using ontology
- Full-stack platform: BERT/RoBERTa benchmarked head-to-head for NER

Key Findings

- RoBERTa consistently outperforms BERT across all metrics
- Semantic embeddings outperform keyword matching on unstructured text
- Microservice architecture scales to 500 concurrent users, <2% error

Limitations

- Extremely semantic-centric — no skill-aware reranking or gap scoring
- User satisfaction ≠ career outcomes; no longitudinal validation
- No modeling of missing critical skills or capability asymmetry.

90%

Extraction accuracy

92%

User satisfaction

500

Concurrent users

Literature Review — Paper Deep Dive

ResumeInsight — Daberao et al.

IEEE Global Conference on IT & Communication Networks (GITCON), 2025

What They Did

- 7-feature vector: Levenshtein skill similarity + CGPA/board grades
- Targeted Indian campus fresher recruitment context
- Forest vs XGBoost vs ANN evaluated with confusion matrices

Key Findings

- String distance sufficient when resumes contain clean, explicit labels
- Academic metrics (CGPA, board %) improve fit prediction in campus hiring
- No transformer infrastructure required for structured resume datasets

Limitations

- Depended heavily on lexical/string-based matching with weak semantic understanding.
- Used clustering-generated labels instead of reliable ground truth.
- Produced fit scores only, without explainable skill-gap analysis.

72%

Fit-prediction F1

89.3%

Skill extraction F1

1,500

Resume dataset size

Literature Review — P3 Deep Dive & Comparative Analysis

Scopira AI — Sribharathi et al.

IEEE International Conference on Smart Structures & Systems (ICSSS), 2025

91.8%	89%	10K
Matching accuracy	User satisfaction	Concurrent users

What They Did

- TF-IDF + BERT cosine similarity for career-skill matching
- Career path generator via decision trees; portfolio branding advice
- One-month pilot across 5 institutions, 200+ users; Likert scale eval

Key Findings

- BERT + cosine outperforms TF-IDF baseline by ~13%
- Outperforms LinkedIn Skills Match & IBM Watson Career Explorer
- Modular microservice arch handles 10,000 concurrent users at 1.8s avg

Limitations

- Focused mainly on document-level similarity rather than capability-gap reasoning.
- Lacked explicit skill-aware and structural reranking features.
- Did not analyze supervision quality, label reliability, or hard-negative robustness.

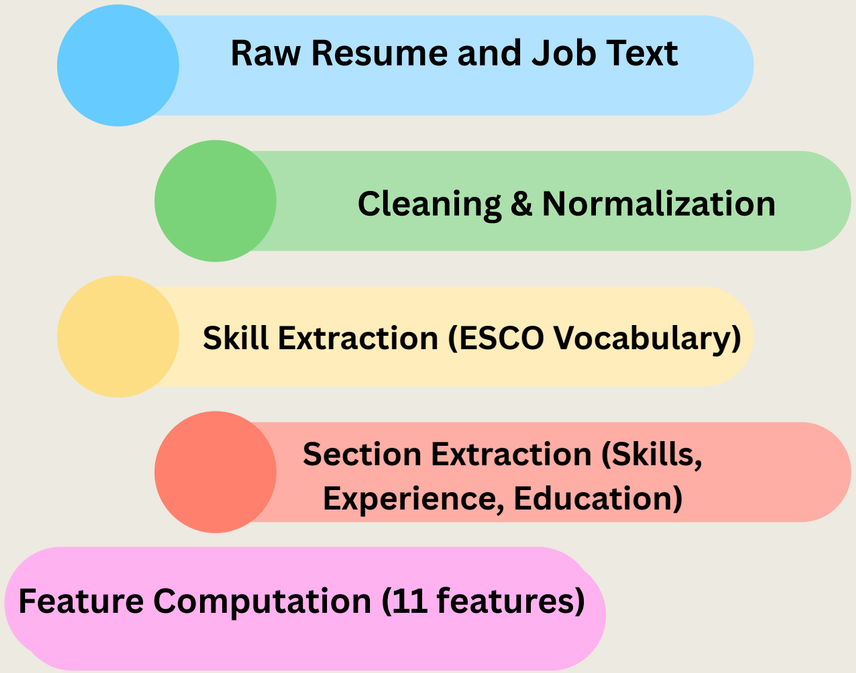
Comparative Analysis

Aspect	AI Skill-Gap Analyser	ResumeInsight	Scopira AI	Ours
Matching	SBERT only	Features+ML	TF-IDF+BERT	**SBERT+Rerank**
Labels	Unclear	Clustering	Unclear	**3-LLM consensus**
Skill Gap	None	None	Limited	**Explicit**
Rigor	Low	Very Low	Moderate	**High**
Evaluation	Limited	Weak	Limited	**Strong**
Interpretability	Partial	Low	Limited	**Explainable**

Key Takeaway: All prior work shares the same pipeline (extract → match → score) but none addresses label quality, skill-aware reranking, or explicit gap diagnosis simultaneously. SkillMatch closes all three gaps.

Dataset & Preprocessing

Preprocessing Pipeline



KEY NUMBERS

2,484
Resumes
(Kaggle, anonymized)

853
Job descriptions
(public corpus)

50,650
Candidate pairs
(diversified pool)

~20,000
ESCO
Skills with hierarchy

Key Processing Steps

- Text Cleaning
- Skill Extraction
- Lemmatization

ESCO vocabulary expansion lifted skill_overlap > 0 from 36% to 68% of pairs.

Dataset & Preprocessing

Fixing Hidden Data Quality Failures

NO Variation in skill features

Problem:
Sparse skill coverage
Fix:
ESCO vocabulary expansion
Impact:
Greater diversity and richer features

Ambiguous Tech Tokens

Problem:
Languages like **R, Go, and Swift** caused **false matches.**
Fix:
Added capitalization + context-aware filtering.
Impact:
Removed **654 false positives.**

Resume Title Extraction

Problem:
Broad category labels reduced title specificity.
Fix:
Extracted titles directly from resume first lines.
Impact:
Expanded unique title matches from **7K → 31K.**

Methodology : Two-Stage Architecture

Stage1 : Retrieval

SBERT semantic similarity

- • SBERT semantic retrieval
- 2.1M possible pairs
- Top-50 candidates retained

Stage2 : Re- Ranking

Logistic Regression on 8 features

Trained on 500 LLM-consensus labels

- Logistic Regression
- 8 engineered features
- Outputs:
  - match probability
  - skill-gap analysis

The 8 features (4 Channels)

SEMANTIC

- `embedding_similarity`
- `title_similarity`

LEXICAL

- `tfidf_similarity`

SKILL

- `skill_overlap`
- `weighted_skill_score`
- `num_missing_skills`
- `avg_missing_skill_importance`

STRUCTURAL

- `years_of_experience`

Methodology Journey: 5 Strategies tested

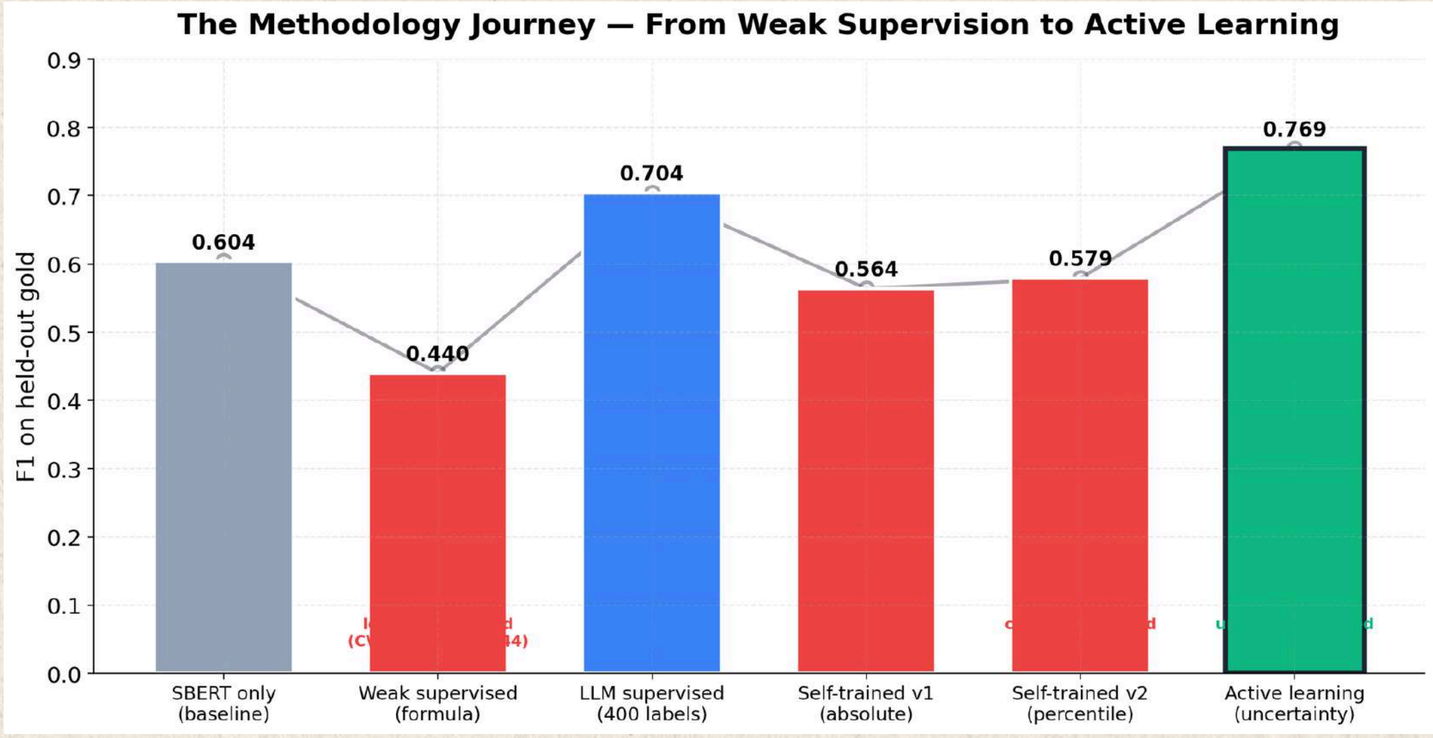
1) Weak Supervision
Formula labels
F1=0.44 · CV-test gap 0.55 (leakage)

2)LLM Supervision
400 LLM labels
F**1=0.704 · +10 over SBERT baseline**

3) SELF-TRAIN v1
Absolute threshold
F1=0.564 · class balance flipped

4) SELF-TRAIN v2
Percentile + diversified
F1=0.579 · hard cases excluded

5) ACTIVE LEARNING
Uncertainty Band Sampling
F1=0.769 · +6.5 over supervised



Confidence-based selection (self-training) fails. Uncertainty-based selection (active learning) wins. Same label budget but opposite mechanisms.

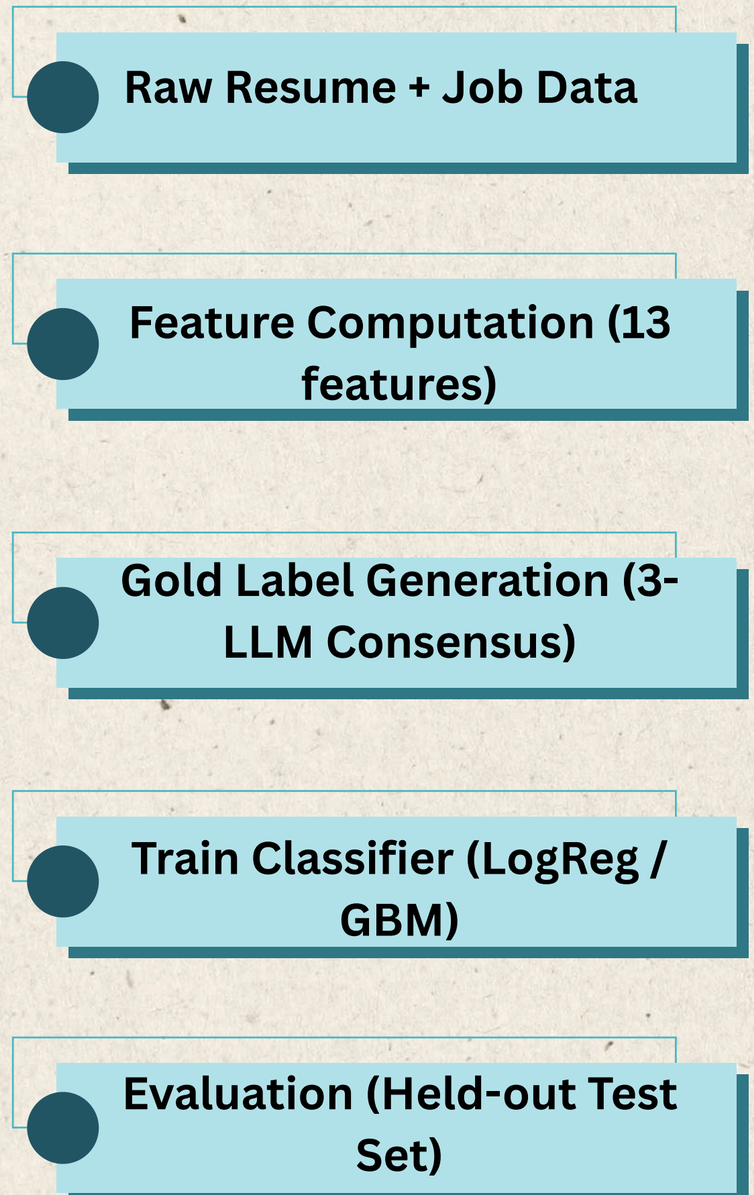
Methodology 1: Weak Supervision

- **Raw Resume + Job Data**
- **Feature Computation (13 features)**
- **Weak Label Generation (Formula-based)**
- **Label Filtering (Top 30% / Bottom 30%)**
- **Train Classifier (LogReg / GBM)**
- **Evaluation CV vs External Test Set**

1. **Raw Data**: Resumes **(2,484)**, Job Descriptions **(853)**
2. **Feature Computation**: Feature Extraction - Embedding similarity (SBERT), TF-IDF similarity, Skill overlap, Missing skills, Experience features
3. **Weak Label Generation**: Heuristic Labeling- Score = **0.7 × weighted_skill_score + 0.3 × title_similarity**
4. **Label Filtering**: Quantile Thresholding - **Top 30% → Positive** (1), **Bottom 30% → Negative** (0), **Middle 40% → Discard**
5. **Model Training**: Classifier- Logistic Regression, Gradient Boosting (trained on weak labels)
6. **Evaluation Performance:** Check Cross-validation - **F1 External test, F1 (LLM-labeled)**

F1 score = 0.44

Methodology 2: LLM Supervision



1. **Raw Data**: Resumes **(2,484)**, Job Descriptions **(853)**
2. **Feature Computation**: Embedding similarity (SBERT), TF-IDF similarity, Skill overlap, Missing skills, Experience features
3. **Gold Label Generation**: **LLM-based labeling** using Claude + GPT-4 + Gemini → Majority vote
4. **Model Training: Classifier** – Logistic Regression, Gradient Boosting (**trained on LLM-labeled data**)
5. **Evaluation Performance: Stratified train-test split** → F1 on held-out gold test set

F1 score = 0.704

Methodology 3: Self-Trainingv1 (Confidence-Based)

- LLM-Supervised Model
- Predict on Unlabeled Data
- High-Confidence Selection (≥0.8 / ≤0.2)
- Pseudo-Label Generation
- Retrain Classifier Evaluation(Test Set)

1. **Base Model:** Trained on **400 LLM**-labeled pairs
2. **Prediction Step**: Model predicts probabilities on **unlabeled dataset**
3. **Pseudo-Label Selection**:
4. ≥ 0.8 → Positive (1)
5. ≤ 0.2 → Negative (0)
6. **Others discarded**
7. **Pseudo-Label Generation**: High-confidence predictions treated as new labels
8. **Model Retraining**: Combine **gold + pseudo-labeled data** → retrain classifier
9. **Evaluation Performance**: F1 on held-out test set

F1 score = 0.564

Methodology 4: Self-Training v2 (Percentile + Diversified Pool)

- **LLM-Supervised Model**
- **Predict on Diversified Pool**
- **Balanced Pseudo-Labels**
- **Retrain Classifier**
- **Evaluation (Test Set)**

1. **Base Model:** LLM-supervised classifier
2. **Diversified Pool**: Includes **topK, mid-similarity, cross-category, random pairs**
3. **Pseudo-Label Selection**:
4. Top 5% → Positive
5. Bottom 5% → Negative
6. **Balanced Labels**: Ensures **~50/50 class distribution**
7. **Model Retraining**: Combine **gold + pseudo-labels**
8. **Evaluation Performance**: F1 on held-out test set

F1 score = 0..579

Methodology 5: Active Learning (Final Model)

LLM-Supervised Model

Uncertainty Sampling (0.4 – 0.6)

LLM Labeling (Hard Cases)

Expanded Training Set

Retrain Classifier

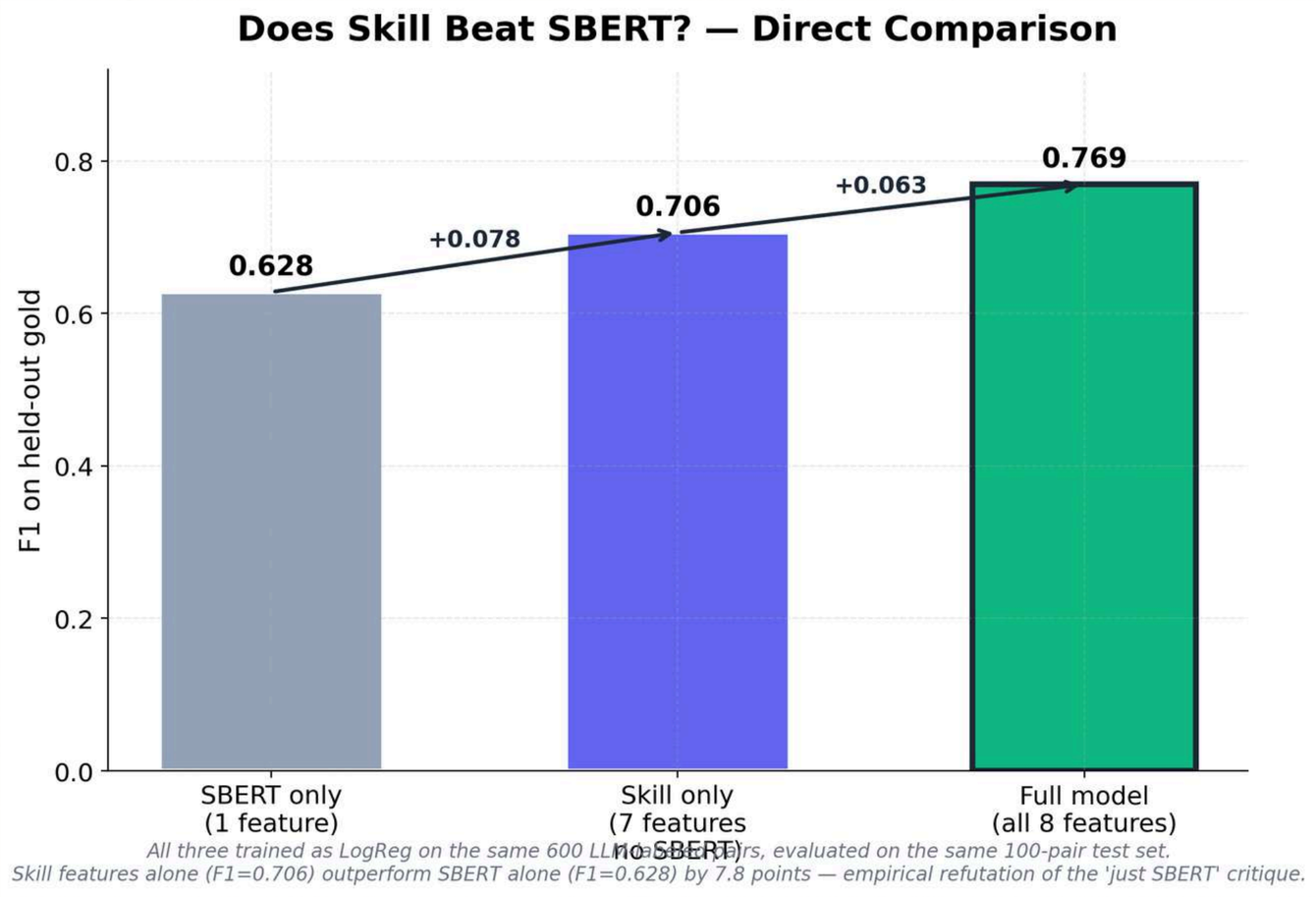
1. **Base Model**: LLM-supervised classifier **(400 labeled pairs)**
2. **Uncertainty Sampling**: Select pairs with prediction **probability in [0.4, 0.6]**
3. **Hard Case Selection**: **200 most uncertain samples** from diversified pool
4. **LLM Labeling**: **Claude + GPT + Gemini** → majority vote
5. **Model Retraining**: 400 + 200 = **600 labeled samples**
6. **Evaluation Performance**: **Final F1 on held-out test set** (best performance)

F1 score = 0.769

Results - Performance Matrix

Approach	Accuracy	Precision	Recall	F1
SBERT only	0.79	0.516	0.727	0.604
Weak- Supervised	0.72	0.393	0.500	0.440
Self- trained V1	0.83	0.647	0.500	0.564
Self- trained V2	0.84	0.688	0.500	0.579
LLM-Supervised	0.84	0.594	0.864	0.704
Active learning	0.88	0.667	0.909	0.769

Why SBERT Alone is Not Enough



SBERT is a strong retriever, but requires skill-aware features and high-quality labels to make reliable match decisions.

Why LLM Labeling + Reranking

- SBERT output is noisy → needs filtering
- Weak labels fail (leakage)
- Self-training fails (confirmation bias)

Solution:

- LLM consensus labels
- Reranker (LogReg)

SBERT Limitation (Retrieval ≠ Matching)

- Only 47.5% of SBERT top-50 are real matches
- High similarity → many false positives

SBERT retrieves plausible candidates, not correct ones

Deployability — Plaksha Use Cases

The dashboard provides an end-to-end career guidance system by predicting resume–job match probability, identifying missing skills ranked by ESCO importance, and generating a personalized 2-year learning roadmap. It also includes a forward simulation module that re-evaluates candidate fit after acquiring new skills.

PLACEMENT PORTAL

1. Internship matching tool

When a student logs in, surface top-K live internship matches with per-job skill-gap reports.

COURSE PLANNING

2. Course-to-career mapping

Given transcript + target role (e.g. ML engineer), recommend electives next semester.

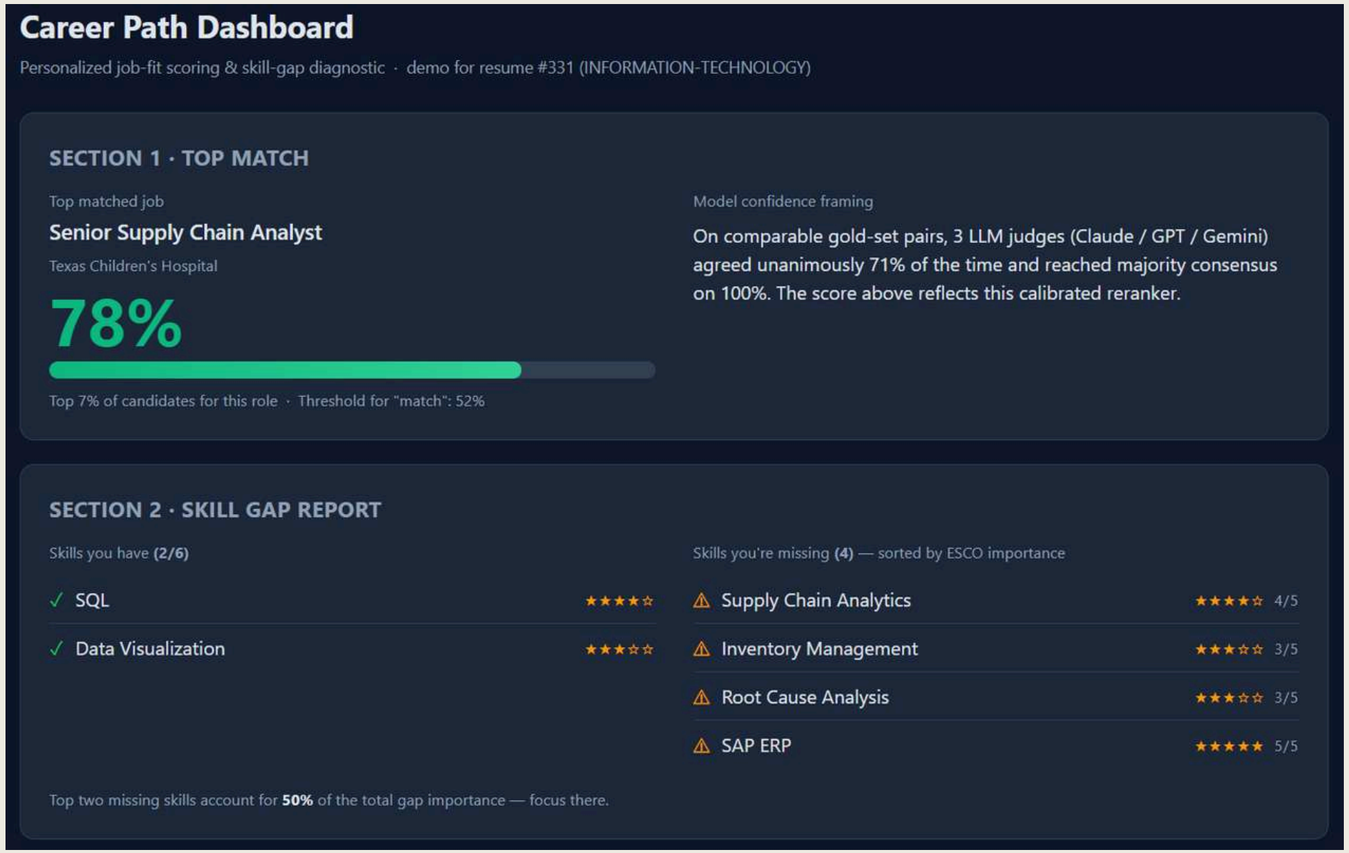
CAREER ADVISORY

3. Skill-development advisor

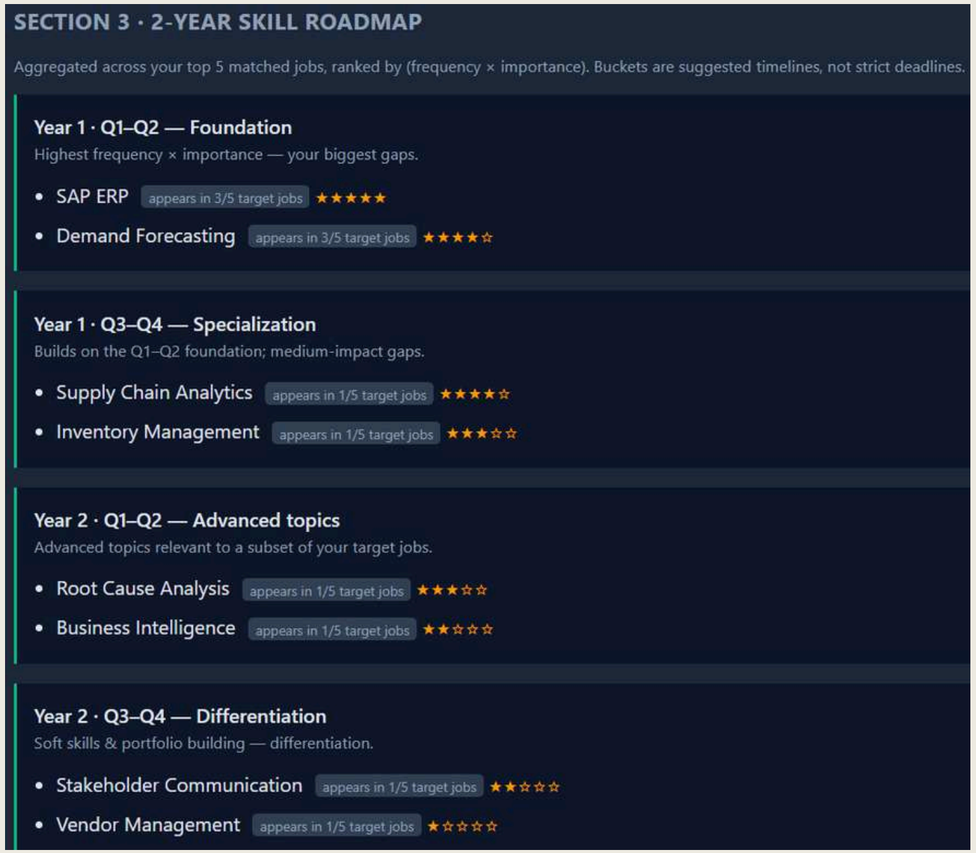
Aggregate gaps across student's top-10 dream jobs → 5 highest-impact skills to learn.

Deployability - Dashboard

User Interface



Two - Year Roadmap



System Limitations & Future Direction

Strong performance with known limitations and clear improvement path

Strengths

- Strong performance: F1 = 0.769 (+16.5 over SBERT)
- Skill-aware features improve decision quality
- Active learning improves label efficiency
- Interpretable outputs: skill gaps + roadmap

Known Limitations

- Small test set (N=100)
- LLM labels ≠ true ground truth (correlated bias)
- Residual noise in skill vocabulary
- Domain mismatch possible (IT-heavy dataset)
- No real-world deployment validation

Future Work Roadmap

Current Model (Research Prototype)

↓

Improve Data Quality

↓

Expand Generalization

↓

Human labeled ground truth

↓

Real world Deployment

Thank You

Aditya

Kuhuk

Reya