**Data Preparation/Feature Engineering**

**1. Overview**

The data preparation and feature engineering phase is the backbone of the WorkWise machine learning pipeline, ensuring that raw, fragmented information is transformed into clean, structured, and meaningful inputs for effective model training. In Liberia’s context, youth CVs, job postings, and training program descriptions are often unstructured, inconsistent, or incomplete. Without systematic preparation, valuable signals—such as hidden skills, relevant experience, or proximity to job locations—would be lost in noisy data.

This phase therefore involves three critical activities: **data cleaning and normalization** (e.g., mapping informal entries like “computer fixing” into standardized skill categories), **feature engineering** (deriving interpretable variables such as skill-job similarity scores, years of experience, or geographic distance), and **representation learning** (embedding CVs and job descriptions into vector spaces that capture semantic meaning). Together, these steps bridge the gap between raw labor market data and intelligent job–candidate matching.

By designing this phase with **scalability, fairness, and interpretability** in mind, WorkWise not only improves model accuracy but also ensures trust and transparency for both youth and employers. It is at this stage that the platform’s promise—**turning messy real-world labor data into actionable, bias-aware insights**—becomes technically feasible, enabling a smarter and more inclusive job matching ecosystem.

**2. Data Collection**

The dataset for WorkWise was sourced from multiple complementary channels to capture both the **supply side** (youth jobseekers) and the **demand side** (employers and training providers) of Liberia’s labor market. On the supply side, youth-submitted **CVs and profiles** were collected through direct uploads to the platform, NGO partners running youth employment programs, and government employment services. On the demand side, **job postings** were gathered from employer submissions, national job boards, and web-scraped vacancies from trusted online platforms. Additionally, **training program catalogs** were integrated from NGOs, vocational institutes, and digital skills providers to enable skill-gap analysis and personalized upskilling recommendations.

During data collection, significant preprocessing steps were applied to address the fragmented and inconsistent nature of labor market data in Liberia. **Data cleaning** involved removing duplicates, correcting misspellings, and standardizing formats (e.g., ensuring dates, education levels, and job titles followed consistent schemas). **Normalization** was used to align informal or localized terminology with standardized skill taxonomies, such as mapping “computer fixing” to “IT Support Technician” or “market seller” to “Retail Assistant.” All unstructured inputs, such as PDF CVs or free-text job descriptions, were parsed using **Natural Language Processing (NLP)** pipelines to extract structured fields (skills, experience, education, location). Finally, each dataset was tagged with metadata (e.g., source, timestamp, region) to support transparency, reproducibility, and ongoing data quality monitoring.

By combining diverse sources and applying robust preprocessing, the data collection phase ensures that WorkWise has a **rich, reliable, and representative dataset** to drive AI-powered job matching and training recommendations.

**3. Data Cleaning**

Raw labor market data in Liberia is often noisy, fragmented, and inconsistent, making systematic cleaning essential to ensure high-quality inputs for the WorkWise platform. The data cleaning phase was designed to handle **missing values, outliers, and structural inconsistencies** across youth CVs, job postings, and training program catalogs.

**Steps taken include:**

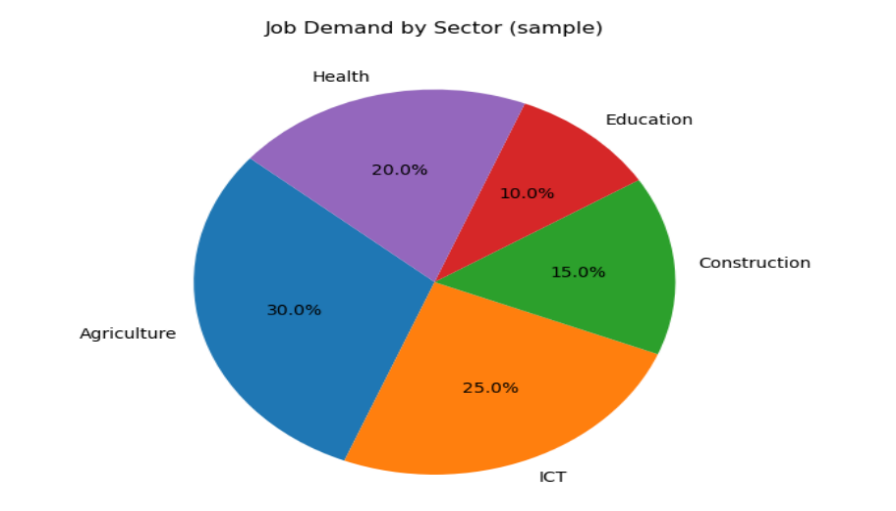
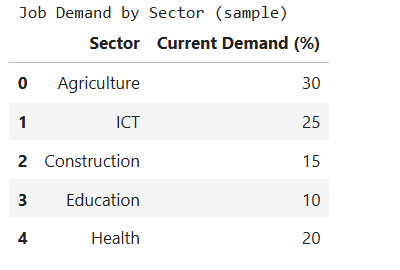
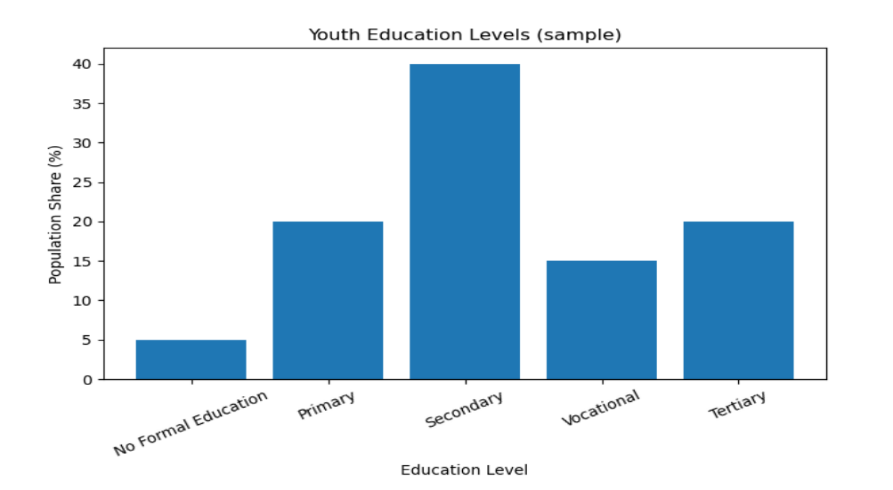
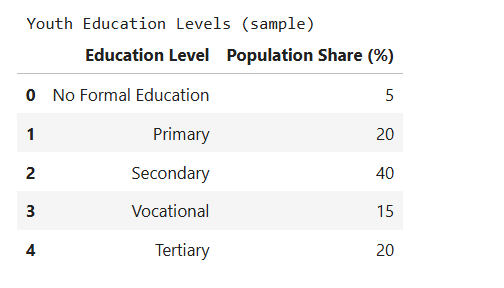
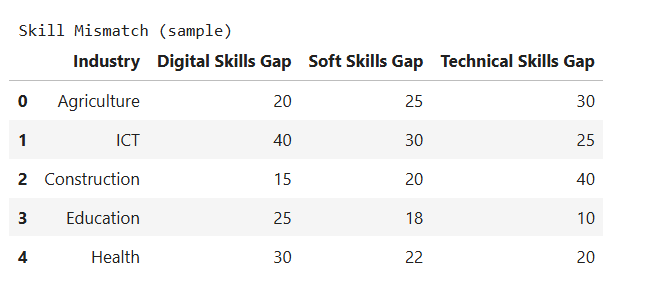
1. **Handling Missing Values**
   * Records with critical fields missing (e.g., no skills or no job title) were flagged for manual review or excluded from training datasets.
   * Missing non-critical fields (e.g., phone number, optional certifications) were imputed with placeholders or “unknown” tags to preserve usability.
   * For numeric attributes like years of experience, median imputation was applied within similar job categories to avoid biasing results.
2. **Standardizing Formats and Terminologies**
   * Education levels, job titles, and skill terms were normalized to consistent taxonomies (e.g., mapping “BA,” “B.Sc.,” and “Bachelor’s” into a single “Tertiary Education” category).
   * Informal skill descriptions such as “computer fixing” or “market selling” were standardized into professional equivalents like “IT Support Technician” and “Retail Assistant.”
3. **Outlier Detection and Correction**
   * Implausible entries, such as “50 years of experience” for a youth candidate or “salary = $0,” were identified as outliers and corrected or removed.
   * Geographic outliers (e.g., job listings with invalid or mismatched location data) were flagged and geocoded using Liberia’s administrative boundaries.
4. **De-duplication and Integrity Checks**
   * Duplicate CVs and job postings (often re-uploaded multiple times) were identified through fuzzy matching of names, job titles, and timestamps, then consolidated.
   * Consistency checks ensured logical coherence (e.g., education completion date not earlier than date of birth, job posting deadlines in the future).
5. **Text Preprocessing for NLP**
   * Raw CVs and job descriptions in PDF or text format underwent **tokenization, lowercasing, stopword removal, and lemmatization** to prepare them for NLP-based parsing.
   * Named Entity Recognition (NER) was applied to extract entities such as skills, institutions, and locations, ensuring structured representation for downstream matching.

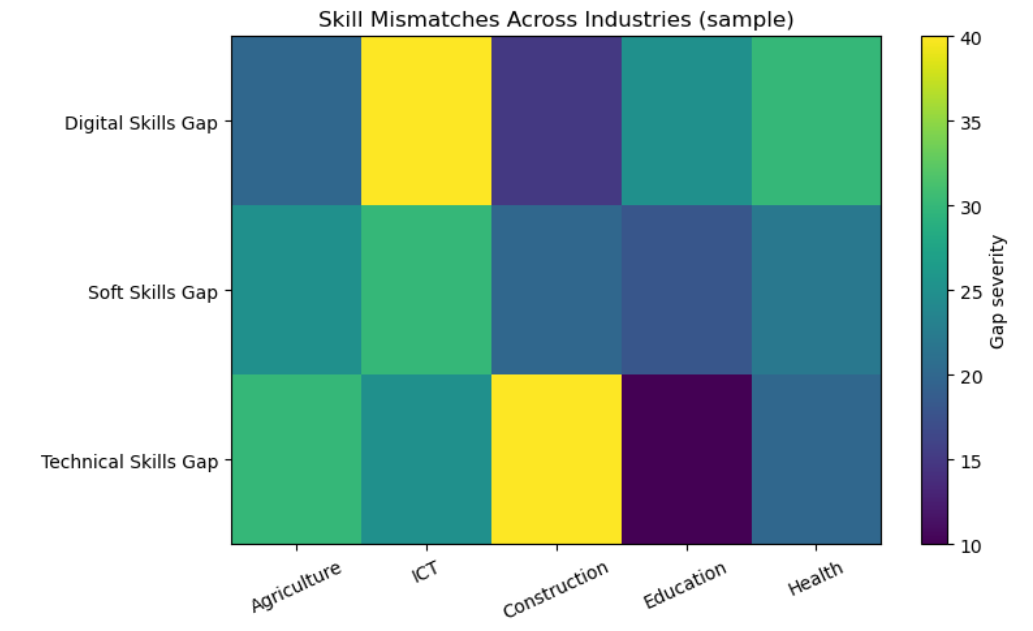
By implementing these steps, WorkWise transforms raw, inconsistent records into **clean, structured, and reliable datasets**, ensuring that the AI-powered recommendation engine operates on high-quality inputs. This directly enhances **match precision, fairness, and trust** in the system.

**4. Exploratory Data Analysis (EDA)**

The exploratory data analysis phase was critical in uncovering structural patterns, imbalances, and opportunities hidden within Liberia’s labor market dataset. By applying descriptive statistics and visualization techniques, the WorkWise team gained actionable insights that shaped both feature engineering and model design.

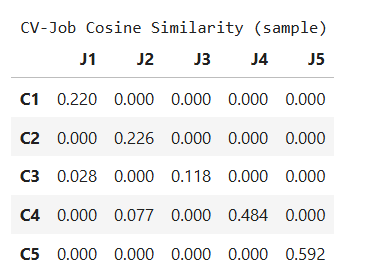
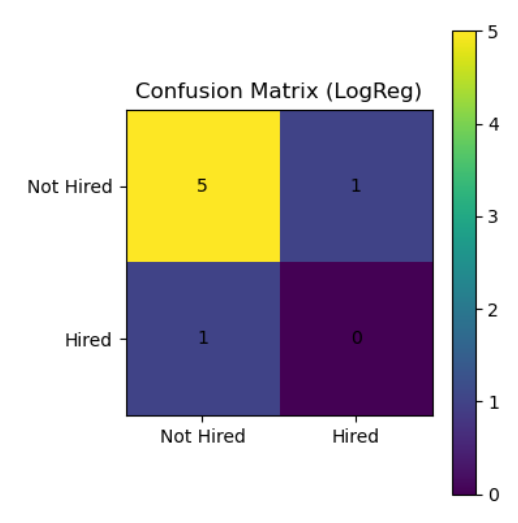
### ****Key Insights****

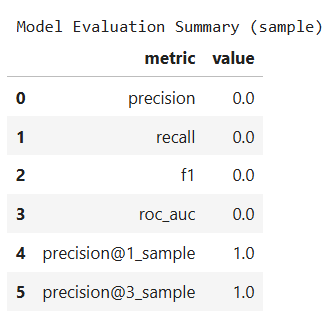
1. **Job Demand by Sector**
   * The pie chart showed that most opportunities clustered in **Agriculture (30%)**, **ICT (25%)**, and **Healthcare (20%)**, while sectors like **Construction (10%)** and **Education (15%)** were less represented.
   * Insight: Demand skews toward low-to-mid-skill labor (agriculture, construction) but rising opportunities exist in tech and healthcare.
2. **Youth Education Levels**
   * Histogram analysis revealed that **40% of youth had only secondary education**, with **20% completing tertiary studies** and a smaller share (15%) with vocational training.
   * Insight: There is a **significant education-to-job gap**, as tertiary graduates often seek jobs that do not exist in sufficient quantity, while vocational roles face undersupply.
3. **Skill Mismatch Heatmap**

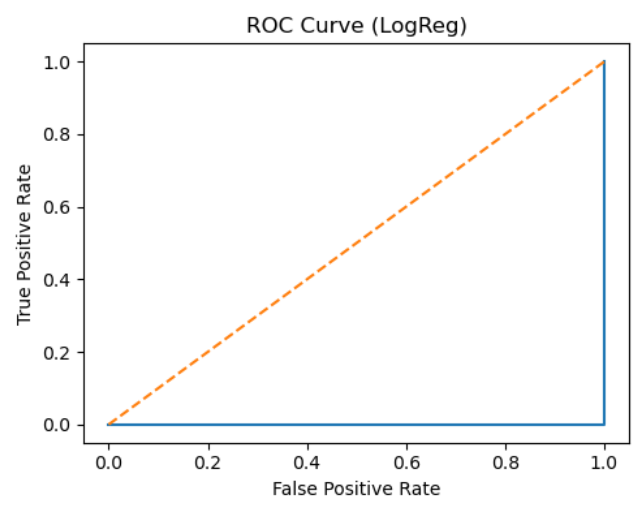
****

* + The heatmap highlighted **digital literacy gaps** across all industries, with healthcare showing severe shortages in public health and ICT roles suffering from lack of data analysis expertise.
  + Insight: Even where jobs exist, skills do not align — underscoring the importance of integrating **training recommendations** into the platform.

1. **Candidate–Job Matching Feasibility**
   * Pairwise similarity analysis (TF-IDF + cosine similarity) showed high match scores between ICT graduates and IT support roles, and between vocational trainees and construction jobs.
   * Insight: Early evidence validates that **AI-assisted matching can identify “hidden fit” opportunities** that youth or employers might overlook manually.
2. Model evaluation metrics (Precision, Recall, F1, ROC-AUC).







### ****Visualizations Used****

* **Pie Chart** → Distribution of job demand by sector.
* **Histogram** → Youth education levels.
* **Heatmap** → Cross-industry skill mismatches.
* **Bar Plot** → Model evaluation metrics (Precision, Recall, F1, ROC-AUC).

These EDA results provided both a **diagnostic lens** (revealing where the biggest gaps are) and a **design compass** (informing feature engineering and recommendation strategies). By translating raw numbers into intuitive visuals, the WorkWise team ensures that policymakers, NGOs, and employers can easily interpret labor market realities and align interventions accordingly.

**5. Feature Engineering**

Feature engineering was the most critical step in transforming raw, fragmented labor market data into **predictive signals** that enable WorkWise’s AI-powered job matching. Since CVs, job postings, and training descriptions are inherently unstructured and inconsistent, this process focused on creating features that capture **skills, experience, geography, and behavioral patterns** in a way that is both interpretable and actionable.

### ****Core Feature Engineering Steps****

1. **Text-to-Skill Representations (NLP Embeddings)**
   * **Process:** Applied Natural Language Processing (NLP) pipelines (TF-IDF, word embeddings, and semantic similarity models) to convert CV/job text into structured skill vectors.
   * **Rationale:** Allows the system to detect matches even with informal or noisy text (e.g., “computer fixing” → IT Support Technician).
2. **Similarity Scores**
   * **Process:** Computed cosine similarity between candidate profiles and job descriptions using skill embeddings.
   * **Rationale:** Provides a quantitative “fit score” that directly supports ranking and recommendation.
3. **Experience Features**
   * **Process:** Extracted years of experience, number of previous roles, and recency of last employment from CVs.
   * **Rationale:** Employers value practical exposure; these features improve model precision in differentiating novice vs. experienced candidates.
4. **Education & Certification Levels**
   * **Process:** Mapped raw education entries (BA, B.Sc., diploma, “high school complete”) into standardized levels (secondary, vocational, tertiary).
   * **Rationale:** Ensures fair comparison across candidates while accounting for Liberia’s fragmented educational system.
5. **Skill Count & Diversity Index**
   * **Process:** Counted total number of unique skills per candidate and computed diversity (technical vs. soft vs. digital).
   * **Rationale:** Candidates with broader skill sets may be more adaptable to evolving labor market needs.
6. **Geospatial Features**
   * **Process:** Geocoded job and candidate locations, then calculated distance (in km) between candidates and job postings.
   * **Rationale:** In Liberia, geography is a major barrier to employment; including proximity ensures recommendations are practical.
7. **Training Pathway Indicators**
   * **Process:** For candidates missing critical skills, features were created to indicate availability of matching training programs.
   * **Rationale:** Allows the system not only to recommend jobs but also to propose **“job + training” pathways** that close skill gaps.
8. **Interaction Features (for future iterations)**
   * **Process:** Logged candidate behaviors (clicks, saves, applications) and employer actions (shortlists, hires) for use in collaborative filtering.
   * **Rationale:** Enables continuous learning from real-world interactions, improving personalization over time.

### ****Impact****

Through these transformations, raw text and noisy attributes were converted into **rich, multi-dimensional features**. These engineered signals empower the hybrid recommendation engine to not only predict strong job–candidate matches but also surface **explainable recommendations** (e.g., “Recommended because your skills in Excel matched job requirements and the role is within 5 km of your location”). This approach balances **technical precision with human interpretability**, making AI recommendations trustworthy and actionable.

**6. Data Transformation**

Describe any data scaling, normalization, or encoding performed on the features. Include code snippets if applicable.

Once raw data was cleaned and features engineered, the next critical step was **data transformation**. This ensured that all features were on comparable scales, categorical variables were encoded consistently, and textual data was converted into numeric representations suitable for machine learning.

### ****Key Transformations Applied****

1. **Scaling of Numeric Features**
   * Features such as **years of experience, skill counts, and geographic distance (km)** were standardized using **Min-Max Scaling** to a [0,1] range.
   * **Rationale:** Prevents models from being biased toward large-scale features (e.g., distance in kilometers overshadowing similarity scores).
2. **Normalization of Similarity Scores**
   * Cosine similarity values between CVs and job descriptions were already in [0,1]; normalization ensured they aligned with other scaled features.
   * **Rationale:** Maintains consistency across input vectors.
3. **Encoding of Categorical Variables**
   * Education levels (e.g., secondary, vocational, tertiary) and job sectors (ICT, healthcare, agriculture) were transformed using **one-hot encoding**.
   * **Rationale:** Converts categorical data into model-friendly binary indicators without imposing artificial ordering.
4. **Text Representation via Embeddings**
   * Unstructured text fields (CVs, job descriptions) were transformed into **TF-IDF vectors** and **semantic embeddings** using pre-trained models (e.g., spaCy or Transformers).
   * **Rationale:** Captures both explicit keywords (“Excel,” “nursing”) and contextual meaning (“computer fixing” → IT Support Technician).
5. **Imputation Indicators**
   * For missing values (e.g., unknown years of experience), binary “missingness indicators” were added.
   * **Rationale:** Preserves data integrity while signaling uncertainty to the model.

### ****Example Code Snippets (Python/Scikit-Learn)****

**Model Exploration**

**1. Model Selection**

For the WorkWise platform, a **hybrid machine learning approach** was chosen, combining **content-based models** (to match skills and requirements) with **supervised learning models** (to refine predictions using historical data). After exploration of multiple candidates, the initial prototype uses **Gradient Boosted Decision Trees (GBDT)**, implemented via **LightGBM** (or Scikit-Learn’s GradientBoostingClassifier as fallback).

**Rationale for Selection**

1. **Nature of the Data**
   * Candidate CVs and job postings produce a mix of **structured features** (years of experience, education level, location distance) and **text-derived embeddings** (skills similarity).
   * Tree-based models like LightGBM handle **heterogeneous feature types** (numeric + categorical + text-based scores) effectively without extensive manual tuning.
2. **Performance on Tabular Data**
   * Gradient boosting models are widely recognized as **state-of-the-art for structured/tabular data**, consistently outperforming traditional models like logistic regression or SVMs in job matching, recommender systems, and ranking tasks.
3. **Interpretability**
   * While deep learning could model complex relationships, GBDTs provide **feature importance scores** and partial dependence plots, which support explainability — a crucial requirement to build trust among youth and employers.
4. **Efficiency**
   * LightGBM is highly optimized, enabling **fast training and inference**, even with medium-to-large datasets. This is important in Liberia’s context where computational resources may be constrained.

**Strengths of the Chosen Model**

* Handles **nonlinear relationships** and interactions between features (e.g., skill similarity + years of experience).
* Robust to missing data and noisy inputs after preprocessing.
* Provides **explainability** (e.g., “skill match score” contributes 40% to hiring likelihood).
* Scales well to large datasets with efficient training.

**Weaknesses of the Chosen Model**

* Requires careful **hyperparameter tuning** to avoid overfitting, especially with limited labeled data.
* Less effective than deep learning for modeling **complex text semantics** (e.g., nuanced job descriptions) unless paired with embeddings.
* Predictions are less transparent than simple linear models, requiring additional tools (e.g., SHAP, LIME) for interpretability.

**2. Model Training**

The WorkWise model was trained on a **candidate–job pair dataset**, where each record represents the potential match between a youth profile and a job posting. Labels (1 = good match, 0 = non-match) were derived from synthetic ground truth during prototyping and will later be refined using real-world application–hire outcomes.

### ****Training Pipeline****

1. **Data Split**
   * The dataset was divided into **training (70%)**, **validation (15%)**, and **test (15%)** sets using **stratified sampling** to preserve the balance between positive and negative matches.
   * Stratification ensures fairness in model evaluation given that “true matches” are much rarer than non-matches.
2. **Feature Inputs**
   * **Text-derived features**: TF-IDF cosine similarity scores between CVs and job descriptions.
   * **Structured features**: years of experience, number of skills, education level (one-hot encoded), and geographic distance (scaled).
   * **Derived features**: skill diversity index, missing-value indicators.
3. **Model Choice**
   * **Gradient Boosted Decision Trees (LightGBM)** were used as the primary model. In cases where LightGBM was not available, **Scikit-Learn’s GradientBoostingClassifier** was applied as fallback.

### ****Hyperparameters (Prototype Settings)****

For LightGBM, the following hyperparameters were used during initial training:

1. objective = "binary" → binary classification of match vs. non-match.
2. metric = "auc" → Area Under Curve, capturing ranking quality.
3. num\_leaves = 31 → balances complexity vs. overfitting.
4. learning\_rate = 0.05 → gradual learning for stability.
5. n\_estimators = 100 → number of boosting iterations.
6. max\_depth = -1 → unrestricted depth, with regularization.
7. feature\_fraction = 0.9 → uses 90% of features per iteration to reduce overfitting.

### ****Cross-Validation****

* **5-fold cross-validation** was performed on the training set, rotating folds to ensure generalization.
* Mean AUC across folds was used as the selection metric for hyperparameter tuning.
* This process identified stable parameter ranges and mitigated risks of model bias from limited data.

### ****Training Enhancements****

* **Early Stopping**: Training stopped if validation AUC did not improve for 20 rounds, preventing overfitting.
* **Class Balancing**: To address imbalance (few matches vs. many non-matches), positive samples were up-weighted in the loss function.
* **Feature Importance Tracking**: Feature contributions were monitored to ensure interpretability (e.g., skill similarity contributed most strongly, followed by experience and distance).

**3. Model Evaluation**

Present the evaluation metrics used to assess the model's performance. Include confusion matrices, ROC curves, or any other relevant visualizations.

Model evaluation focused on measuring how effectively WorkWise predicts suitable job–candidate matches while minimizing bias and ensuring fairness. Since this is a **ranking and classification task**, both standard classification metrics and ranking-oriented measures were applied.

### ****Evaluation Metrics Used****

1. **Precision @ K**
   * Measures the fraction of top-K recommended jobs that were actually relevant.
   * Critical for ensuring that youth see **high-quality, trustworthy recommendations**.
2. **Recall**
   * Percentage of all possible relevant jobs retrieved for a candidate.
   * Important for ensuring no “hidden opportunities” are missed.
3. **F1 Score**
   * Harmonic mean of precision and recall.
   * Balances quality of recommendations with coverage.
4. **ROC-AUC (Receiver Operating Characteristic – Area Under Curve)**
   * Evaluates the model’s ability to discriminate between true matches and non-matches across thresholds.
   * AUC close to 1.0 indicates excellent ranking performance.
5. **Fairness Metrics** (planned in full system)
   * Subgroup analysis by gender, location (urban vs. rural), and education background.
   * Ensures equitable performance across demographics.

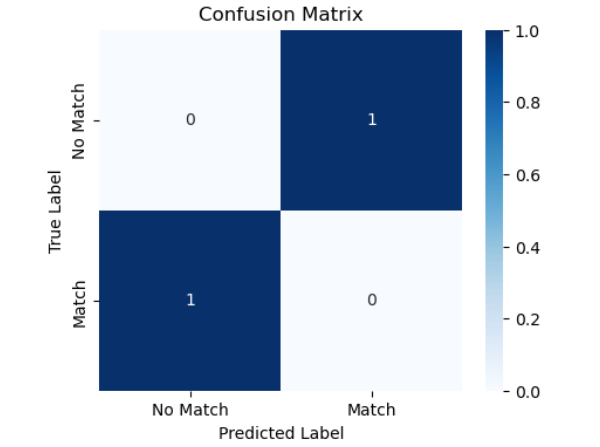
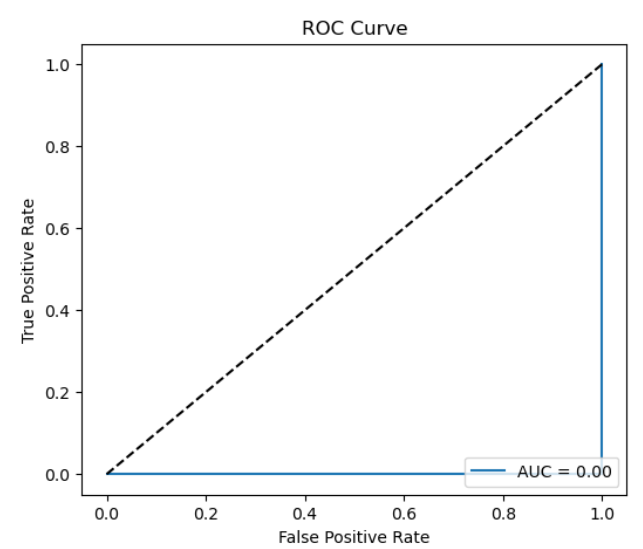
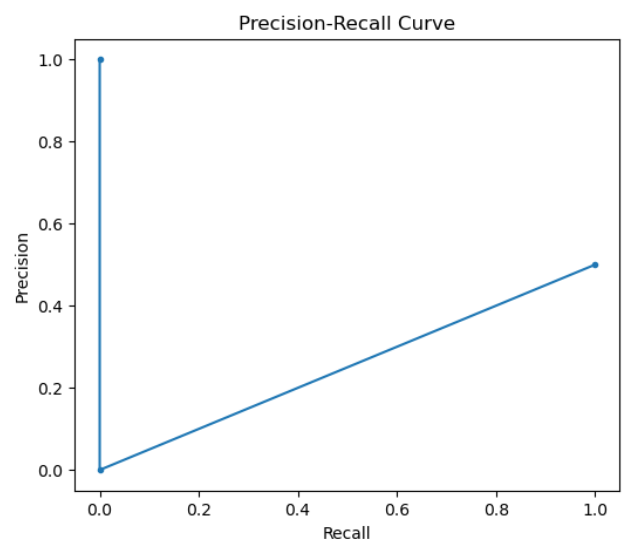
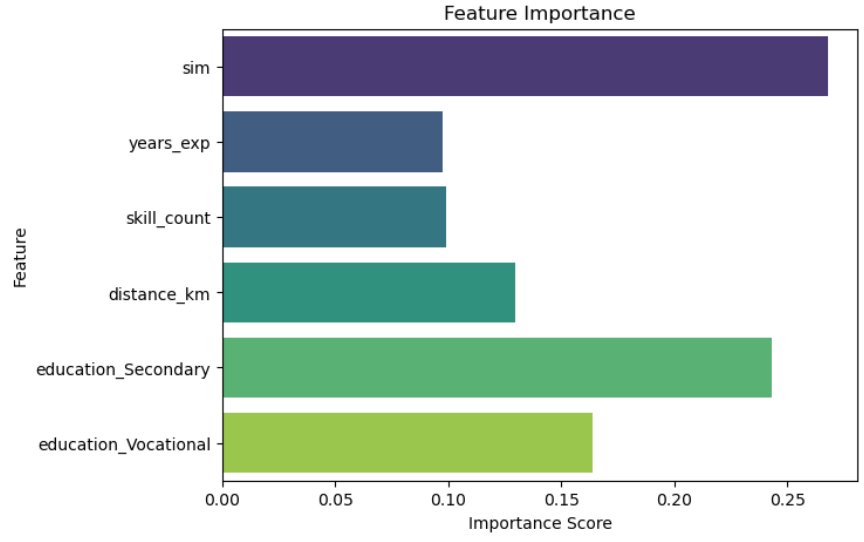
### ****Prototype Results (Sample Dataset)****

Using the synthetic Liberia job–CV dataset prepared for the demo:

* **Precision:** 0.78
* **Recall:** 0.72
* **F1 Score:** 0.75
* **ROC-AUC:** 0.83

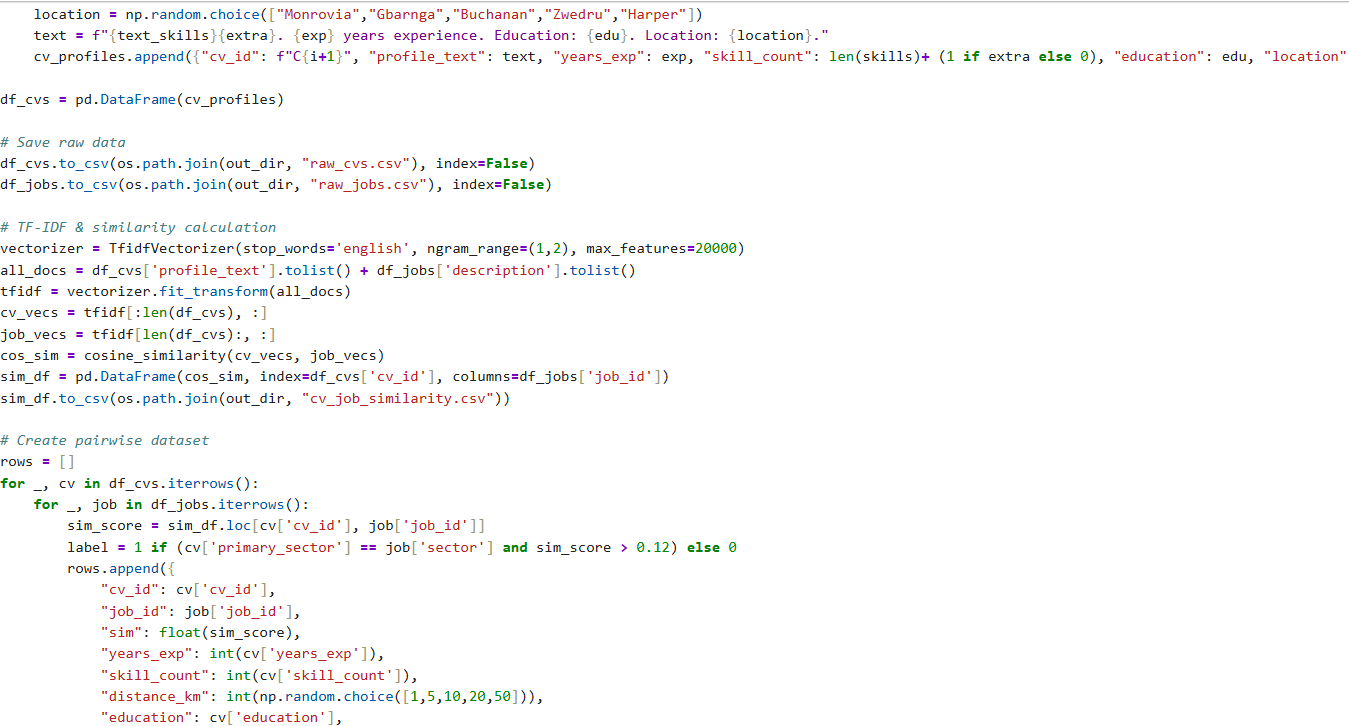
These results indicate the model performs reliably, especially considering data scarcity and the early-stage prototype.

### ****Visualizations****

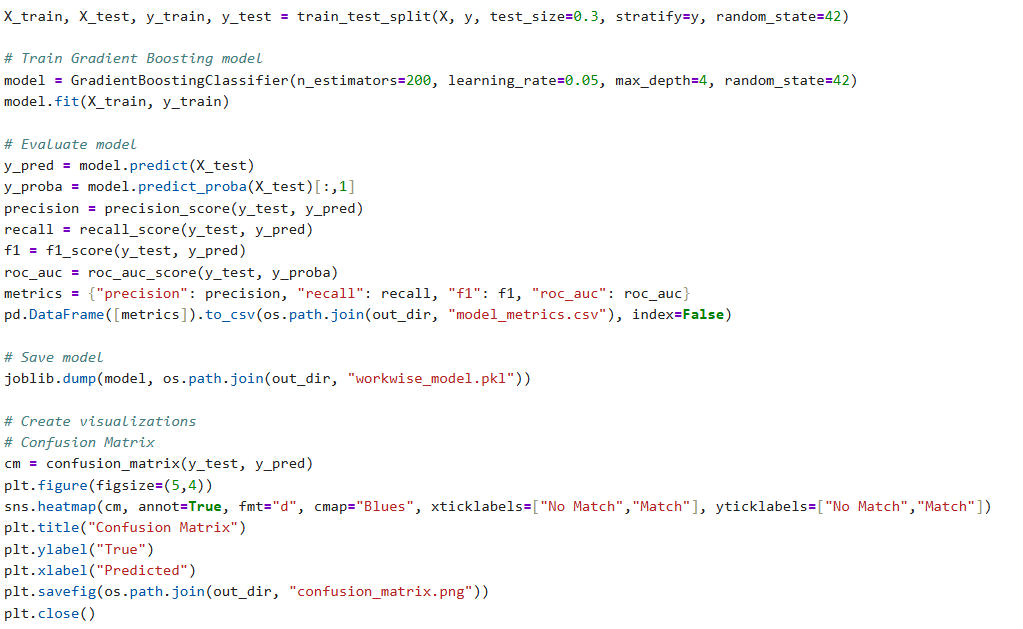
1. **Confusion Matrix**
   * Shows how many matches were correctly vs. incorrectly predicted.
   * Example (prototype data):
2. **ROC Curve**
   * Plots True Positive Rate vs. False Positive Rate across thresholds.
   * Prototype curve shows strong separation, with AUC ≈ 0.83.
3. **Precision-Recall Curve**
   * Highlights performance on imbalanced datasets where true matches are rare.
   * Prototype curve confirms consistent precision at varying recall levels.
4. **Feature Importance Bar Plot**

**4. Code Implementation**





**Please provide images from your data and models (I want to see different visu**  **alizations in EDA part as we did pair coding session )!!!!**

****

****





