# Documentation for
# Employee and Coach Matchmaking Algorithm

## Introduction

This script implements a matchmaking algorithm to pair employees with suitable coaches based on their attributes, preferences, and needs. It uses a deep learning model for scoring compatibility between employees and coaches, facilitating an intelligent and efficient matchmaking process.

---

## Key Components of the Script

### 1. Data Generation

#### Libraries Imported

- **Random**: For generating random employee and coach attributes.
- **Pandas**: For data manipulation and analysis.
- **NumPy**: For numerical operations.
- **scikit-learn**: For preprocessing and splitting datasets.
- **TensorFlow/Keras**: For building and training the deep learning model.
- **Seaborn/Matplotlib**: For data visualization.

#### Employee Attributes

Generated for 100 employees with attributes:

- `EmployeeID`
- `Department` (IT, HR, Finance, Marketing, Sales)
- `CareerGoals` (Leadership, Management, Technical Expertise, Soft Skills Improvement)
- `EmotionalSupportNeeds` (High, Medium, Low)
- `SkillDevelopmentAreas` (e.g., Python, Data Analysis)
- `CoachingStyle` (Structured, Flexible, Mentorship)
- `Availability` (3 random hours from 9 AM to 5 PM)

#### Coach Attributes

Generated for 10 coaches with attributes:

- `CoachID`
- `Certifications` (PCC, ICF, etc.)
- `Specializations` (Career, Emotional, Skill Up)
- `CoachingStyle`
- `ExperienceLevel` (Years of experience)
- `Availability` (3 random hours from 9 AM to 5 PM)

---

## 2. Data Visualization

Visualization techniques were used to analyze relationships among:

- Employee and coach preferences.
- Employee distribution across `CoachingStyle`, `EmotionalSupportNeeds`, and `CareerGoals`.

### Visualizations Used

- Heatmaps: Show correlations between categorical variables.
- Bar Plots: Highlight the distribution of preferences.
- Histograms and Line Plots: Analyze continuous variables like `ExperienceLevel` and `CompatibilityScore`.

---

## 3. Compatibility Scoring

### Compatibility Logic

A function calculates a compatibility score between an employee and a coach based on:

- Department relevance.
- Alignment of `CareerGoals` and `Specializations`.
- `EmotionalSupportNeeds` and specialization match.
- Overlapping `SkillDevelopmentAreas`.
- Matching `CoachingStyle`.
- Overlapping `Availability`.

### Scoring Output

Scores are stored in a dataset for model training.

---

## 4. Data Preparation

### Encoding

Categorical data (e.g., `Department`, `CareerGoals`) was label-encoded for numerical processing.

### Splitting Data

- **Features (`X`)**: Employee and coach attributes.
- **Target (`y`)**: Compatibility score.
- Split into training and testing datasets (80% train, 20% test).

---

## 5. Deep Learning Model

### Model Architecture

- **Input Layers**: One for each categorical and numerical feature.
- **Embedding Layers**: For categorical features (Department, CareerGoals, etc.).
- **Dense Layers**: Fully connected layers with `ReLU` activation.
- **Dropout Layers**: Prevent overfitting.
- **Output Layer**: Single neuron predicting compatibility score (regression).

### Model Compilation

- **Optimizer**: Adam.
- **Loss Function**: Mean Squared Error.
- **Metric**: Mean Absolute Error (MAE).

### Training

- Trained for 50 epochs with a batch size of 10.
- Validation split of 20%.

---

## 6. Model Evaluation

### Performance Metrics

- `Mean Absolute Error (MAE)`: Quantifies prediction error on test data.

---

## 7. Prediction for New Employee

### Steps

1. Define new employee attributes.
2. Encode attributes using pre-trained label encoders.
3. Predict compatibility scores with all coaches.
4. Rank coaches by predicted scores.

### Output

Top matches for the new employee are displayed in a sorted DataFrame.

---

## 8. Visualization of Compatibility Scores

Compatibility scores are visualized using line plots and histograms for deeper insights.

---

# Conclusion

This script demonstrates an end-to-end pipeline for creating and training a matchmaking model, leveraging data science and machine learning techniques. The solution is scalable and can be integrated into a real-world system for automated employee-coach pairing.