NAMIBIA UNIVERSITY
OF SCIENCE AND TECHNOLOGY

Individual Development Plan Recommender Model Using Deep Learning

Master's Thesis

Student: Hubert Patrick Mouton

Programme: Master's in Data Science (RIT912S)

Supervisor: Prof. Muyingi, Hippolyte N'sung-Nza

Date: Saturday 10th January, 2026

Digital Signature of Supervisor: _____

Digital Signature of Programme Coordinator: _____

Digital Signature of FCI HDC Representative: _____

DECLARATION

I, Hubert Patrick Mouton, hereby declare that the work contained in this thesis, for the degree **Master's of Data Science**, entitled:

INDIVIDUAL DEVELOPMENT PLAN RECOMMENDER MODEL USING DEEP LEARNING

is my own original work and that I have not previously, in its entirety or in part, submitted it at any university or other higher education institution for the award of a degree. I further declare that I will fully acknowledge any sources of information I use for the research in accordance with the Institution's rules.

_____ _____

Signature Date

SIGNATURE OF THE SUPERVISOR

_____ _____

Supervisor's Signature Date

Preface

<preface>

Contents

List of Figures iv

List of Tables v

1 Introduction 1
- 1.1 Background to the Study . . . 1
- 1.2 Problem Statement . . . 1
- 1.3 Research Aim . . . 1
- 1.4 Research Questions . . . 2
- 1.5 Research Objectives . . . 2
- 1.6 Rationale and Motivation . . . 2
- 1.7 Research Approach Overview . . . 3
- 1.8 Choice of Methods . . . 3
- 1.9 The Research Time Horizon . . . 3
- 1.10 Expected Deliverables and Contributions . . . 3
- 1.11 Structure of the Thesis . . . 3

2 Literature Review 4
- 2.1 Introduction . . . 4
- 2.2 Individual Development Plans . . . 4
- 2.3 Recommender Systems . . . 4
- 2.4 Deep Learning . . . 4
- 2.5 Deep Learning in Recommender Systems . . . 5
- 2.6 Data Collection Methods . . . 5
- 2.7 Data Preprocessing . . . 5
- 2.8 Model Evaluation Methods . . . 5
- 2.9 Explainability of Recommender Models . . . 5
- 2.10 Future Research Directions . . . 6

3 Theoretical Framework 8

4 Research Design 9
- 4.1 Overview of the Research Design . . . 9
- 4.2 System Architecture . . . 10
- 4.3 Module Descriptions . . . 12
- 4.4 Experimental Design . . . 14
- 4.5 Technical Decisions Justification . . . 16
- 4.6 Summary . . . 18

5 Implementation 19

6 Results and Evaluation 20

7 Conclusion 21
- 7.1 Closing Remarks . . . 21
- 7.2 Research Questions . . . 21

8 Recommendations 22

References 23

A Algorithms 24

List of Figures

1.1 Current IDP Creation Process at Telecom Namibia (Interaction View). 1

4.1 System Architecture for the Deep Learning-based IDP Recommender. The models (NCF, RNN, GNN, Transformer) are independently trained and evaluated to identify the best-performing model for generating personalized IDPs. 11

List of Tables

2.1 Summary of Related Works for the IDP Recommender System Study 7

4.1 Models and Techniques Applied Across Research Phases 10

1. Introduction

1.1. Background to the Study

The study explores the development of Individual Development Plans (IDPs) using a deep learning-based recommender system. IDPs, introduced by the Federation of American Societies for Experimental Biology, offer a structured framework for self-assessment, goal setting, and action planning, particularly for doctoral trainees Vanderford, Evans, Weiss, Bira, and Beltran-Gastelum, 2018. They are crucial for aligning personal growth with organizational goals in today's complex career landscapes.

Recommender systems, widely used in e-commerce, education, and healthcare, provide personalized suggestions. However, traditional systems often struggle with dynamic user behaviors and lack interpretability Sahoo, Pradhan, Barik, and Dubey, 2019. Deep learning enhances these systems by uncovering intricate user-item patterns, enabling stronger personalization and adaptability Li et al., 2024; Mu, 2018.

Advanced deep learning techniques like attention mechanisms, knowledge graphs, and Graph Convolutional Networks (GCNs) are highly effective for modeling relationships among employee skills, career goals, and resources Chen and Zhong, 2024; Li et al., 2024. These methods form the basis for building an adaptive IDP recommender system.

1.2. Problem Statement

Telecom Namibia currently conducts competency assessments manually using Excel templates and email exchanges. This process is inefficient, prone to errors, and lacks real-time insights, thereby hindering timely decision-making. As the organization grows, scalability issues become apparent, necessitating an automated, centralized platform for more efficient and accurate IDP creation.

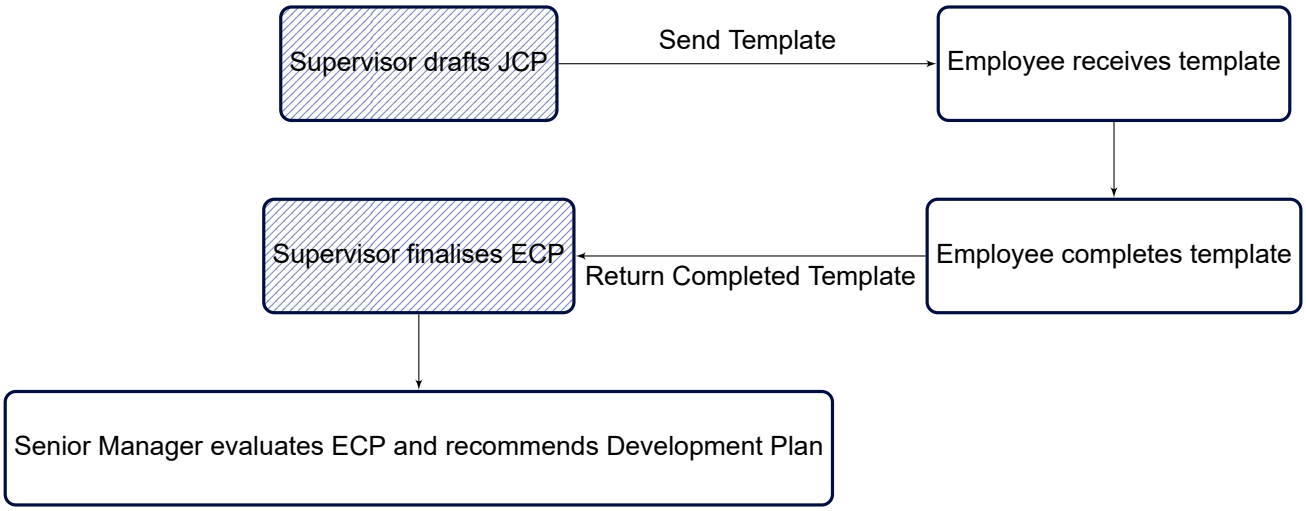

Figure 1.1: Current IDP Creation Process at Telecom Namibia (Interaction View).

1.3. Research Aim

The aim is to enable access to real-time data,through a platform where management, staff and supervisors may interact through training and skill audit assessments. This would automate workflows associated with individual development plan formulation and implementation.

1.4. Research Questions

In what ways may deep learning techniques be applied to create a personalized and efficient recommender model that appropriately evaluates employee competencies and suggests chances for customized growth for individual development plans?

Research Question 1

Which employee data types, e.g., skills, evaluations, performance reviews, etc, are most crucial to the construction of a well performing deep learning model that generates recommendations for individual development plans?

Research Question 2

How can employee characteristics and past performance data be used to refine deep learning algorithms to produce precise and customized recommendations for development plans?

Research Question 3

In what ways can the use of real-time performance data and feedback enhance the individual growth plan recommendations' adaptability and relevance?

Research Question 4

What is the difference between the accuracy, employee skill development, and organizational productivity of the deep learning-based recommender system for individual development plans and the conventional approaches?

1.5. Research Objectives

Research Objective

The primary objective of this research is to design an individual development plan recommender model using deep learning.

- **RO1:** To compile comprehensive datasets containing employee skills, evaluations, and development history.
- **RO2:** To design a deep learning model that analyzes skill gaps and generates customized growth plans.
- **RO3:** To integrate real-time feedback loops to dynamically update employee development recommendations.
- **RO4:** To evaluate the effectiveness of the proposed system compared to traditional IDP development methods.

1.6. Rationale and Motivation

The rapid evolution of employee competencies and organizational objectives makes manual development planning increasingly unsustainable. Automating IDP generation through AI enables scalability, accuracy, and responsiveness. Personally, this project reflects my passion for applying machine learning to real-world human resource development challenges, specifically optimizing organizational talent management systems.

1.7. Research Approach Overview

The research follows an interpretivist philosophical stance, utilizing an abductive mixed-methods strategy. Data will be collected from surveys, interviews, and organizational records. Analyses will include both thematic qualitative coding and quantitative statistical techniques, supporting the development of a deep learning recommender system.

1.7.1. Interpretivism as a Research Philosophy

Interpretivism emphasizes the individuality of human experiences Irshaidat, 2019; Saunders, Lewis, and Thornhill, 2012. Each employee's development needs are influenced by personal, social, and cultural factors Myers, 2008. Interpretivism opposes generalization and instead values subjective interpretations, aligning well with the need for personalized development plans.

1.7.2. The Research Approach

Following Hurley, Dietrich, and Rundle-Thiele, 2021; Mantere and Ketokivi, 2013, an abductive approach combines inductive and deductive reasoning. Thompson, 2022 provides an 8-step abductive framework, guiding data collection, feature extraction, theme development, theorizing, model implementation, and evaluation through continuous reflection.

1.7.3. Research Strategy: Action Research

Action Research (AR), particularly Canonical Action Research (CAR) as outlined by Avison, Lau, Myers, and Nielsen, 1999; Davison, Martinsons, and Malaurent, 2021, is adopted. AR fosters collaboration between researcher and participants in an iterative cycle of problem identification, intervention, observation, and reflection—making it highly suitable for information system development like the proposed IDP recommender.

1.8. Choice of Methods

The study adopts a mixed-methods approach, integrating qualitative insights and quantitative validation et al., 2021; Guetterman, 2016. Data collection will involve semi-structured interviews and structured questionnaires. Analysis will include thematic coding and statistical modeling, supporting deep learning model training and evaluation.

1.9. The Research Time Horizon

A longitudinal design will be employed to monitor participants' development plans over time. Regular assessments will capture evolving employee competencies and measure the recommender system's long-term effectiveness Jung, Kim, Lee, and An, 2023; Kelley and Rausch, 2011. Careful sampling and structured engagement strategies will maintain data integrity and participant involvement.

1.10. Expected Deliverables and Contributions

- A fully functional deep learning-based IDP recommender model.
- A prototype centralized platform for employee development planning.
- Contributions to the field of AI-driven HR systems and personalized learning pathways.

1.11. Structure of the Thesis

- **Chapter 1: Introduction** — Context, problem statement, objectives, and thesis outline.
- **Chapter 2: Literature Review** — Review of IDPs, recommender systems, and deep learning techniques.
- **Chapter 3: Methodology** — Research design, model development strategy, and data collection methods.
- **Chapter 4: Results and Analysis** — Experimental results and system evaluation.
- **Chapter 5: Discussion and Conclusion** — Findings interpretation, research limitations, and future work.

2. Literature Review

2.1. Introduction

The purpose of this literature review is to evaluate existing research on individual development plans (IDPs), recommender systems, data collection methods, data preprocessing techniques, deep learning methodologies, and model evaluation metrics. Individual Development Plans have gained increasing attention in organizational and educational contexts as they allow individuals to outline clear, personalized paths for career growth and skill enhancement. We aim to leverage this demand for individual development plans and seek to solve the challenges faced in generating them by making use of deep learning recommender models.

2.2. Individual Development Plans

Guiding Questions

- **Definition and purpose of IDPs**
- **Importance and benefits in organizational and educational contexts**
- **Challenges and limitations of current IDP methods**
- How have IDPs evolved over recent years?
- What specific benefits do organizations gain from effectively implementing IDPs?
- What are the common barriers to implementing successful IDPs?

2.3. Recommender Systems

Guiding Questions

- **Overview and types of recommender systems**
- **Algorithms and approaches traditionally used (collaborative filtering, content-based filtering, hybrid systems)**
- **Application domains beyond IDPs**
- What are the primary approaches used in recommender systems, and how do they differ?
- What types of recommender systems are most suitable for educational and professional development contexts?
- How have recommender systems impacted user experiences in various domains?

2.4. Deep Learning

Guiding Questions

- Fundamentals of deep learning (concepts, advantages, limitations)
- Key architectures: Neural Networks, CNNs, RNNs, Autoencoders, Transformers, Graph Neural Networks (GNNs)
- Deep learning vs. traditional machine learning techniques
- How does deep learning differ from traditional machine learning in terms of feature extraction and performance?
- What specific deep learning architectures are most relevant to recommender systems?

2.5. Deep Learning in Recommender Systems

Guiding Questions

- Integration of deep learning techniques into recommender systems
- Case studies and successful implementations
- Comparative analysis with traditional recommender methods
- What are some successful examples of deep learning-based recommender systems?
- How have deep learning techniques improved the performance of recommender systems in real-world applications?

2.6. Data Collection Methods

Guiding Questions

- Types of data required for recommender systems (primary vs. secondary)
- Techniques and challenges in data collection (e.g., observation, questionnaires, interviews, databases)
- Ethical considerations and privacy concerns
- What methods are typically used to collect data for recommender systems?
- How do data quality and relevance affect the performance of recommender models?
- What ethical considerations are involved in data collection for IDPs?

2.7. Data Preprocessing

Guiding Questions

- Techniques for preprocessing data (cleaning, normalization, missing data imputation, categorical encoding)
- Importance of data preprocessing in deep learning
- Automated approaches to data preprocessing
- Why is data preprocessing critical for the success of deep learning models?
- Which automated preprocessing techniques have proven most effective in recent studies?

2.8. Model Evaluation Methods

Guiding Questions

- Common evaluation metrics for recommender systems (precision, recall, F1-score, accuracy, MAE, RMSE)
- Comparative analysis of evaluation metrics
- Issues in model evaluation and validation (e.g., bias, variance)
- Which evaluation metrics are most appropriate for IDP recommender systems and why?
- How do various metrics differ in their ability to measure recommender system effectiveness?

2.9. Explainability of Recommender Models

Guiding Questions

- Importance and need for model explainability
- Introduction to SHAP and LIME frameworks
- Implementation and effectiveness of explainability methods in recommender systems
- How do explainability methods enhance user trust and model transparency?
- What are the specific benefits and limitations of LIME and SHAP methods?
- How have explainability techniques improved user engagement and acceptance in recommender systems?

2.10. Future Research Directions

Guiding Questions

- Potential improvements and research gaps
- Emerging trends and technologies (e.g., generative AI, federated learning, ethical AI)
- What emerging trends or technologies could further enhance IDP recommender systems?
- What critical research gaps currently exist in this field, and how can future research address them?

Table 2.1: Summary of Related Works for the IDP Recommender System Study

Author(s) & Year	Title	Methodology	Key Findings and Relevance
Vanderford et al. (2018)	Use of IDPs for doctoral trainees	Survey study	IDPs improve self-assessment and career planning; foundational to this study.
Sahoo et al. (2019)	Deep learning in health recommender systems	Deep collaborative filtering (RBMs)	Deep learning improves recommendation accuracy; supports choice of deep learning.
Mu (2018)	Survey of deep learning recommender systems	Literature review	Validates deep learning as superior for complex recommendation tasks.
Li et al. (2024)	Knowledge graph-enhanced recommender systems	Attention and residual networks	Knowledge graphs improve personalization; important for IDP mappings.
Chen & Zhong (2024)	GCN-based course recommendation system	Graph Convolutional Networks (GCNs)	GCNs model complex relationships; informs modelling employee competencies.
Ertürkman et al. (2019)	Personalized health management platforms	Collaborative platform development	Personalized plans outperform generic; supports personalizing IDPs.
Gulzar et al. (2018)	Personalized course recommender systems	Hybrid recommendation methods	Combining methods improves engagement; supports multi-input IDP systems.
Dabak et al. (2022)	Career development for Gen Y and Z	Developmental cycle framework	Adaptive planning needed for evolving careers; supports dynamic IDP updates.
Ghaffar et al. (2022)	Impact of personality traits on planning	Empirical study in finance	Personality traits influence decision-making; suggests incorporating traits into IDPs.
Wang et al. (2020)	Employee training course recommendations	Bayesian variational network modeling	Career goals improve recommendations; aligns with dynamic IDP needs.
Bui et al. (2016)	Text classification from PDFs	Multi-pass sieve technique	Text extraction improves preprocessing; useful for automating employee documents.
Viani et al. (2019)	Clinical event extraction with RNNs	Supervised learning	RNNs extract structured info from text; supports skill extraction automation.
Lin et al. (2018)	Sparse linear method for recommendation	L0 regularization technique	Improved recommendation precision; useful for refining IDP suggestions.

3. Theoretical Framework

4. Research Design

4.1. Overview of the Research Design

The research design for this study focuses on the development, training, and evaluation of a deep learning-based recommender system to generate personalized Individual Development Plans (IDPs) for employees.

The design follows a modular, data-driven approach, where raw employee data undergoes preprocessing, feature engineering, and is subsequently used to train and evaluate several deep learning models. These models are trained independently to predict suitable development plans based on employee skills, evaluations, learning goals, and performance histories.

Four primary deep learning models are considered: Neural Collaborative Filtering (NCF), Recurrent Neural Networks (RNN), Graph Neural Networks (GNN), and Transformer-based architectures. Each model is trained, validated, and evaluated independently using consistent data splits and evaluation metrics. The goal is not to ensemble or aggregate model outputs, but rather to identify the model that offers the best predictive performance according to established evaluation criteria such as Precision, Recall, F1-Score, and AUC-ROC.

The system architecture is organized into five major phases:

Table 4.1: Models and Techniques Applied Across Research Phases

Model/Technique	Purpose	Research Phase
Data Collection and Preprocessing	Collect, clean, normalize employee-related data (skills, feedback, evaluations)	**Data Collection and Preprocessing**
Principal Component Analysis (PCA)	Reduce feature dimensionality and highlight important employee data patterns	**Feature Engineering**
Autoencoder	Denoise and compress employee data for feature extraction	**Feature Engineering**
Clustering (e.g., K-Means)	Group employees based on similar career goals and skill gaps	**Feature Engineering**
Neural Collaborative Filtering (NCF)	Predict personalized skill development plans based on interaction data	**Model Training**
Recurrent Neural Network (RNN) / LSTM	Model sequential employee learning behaviors and engagement history	**Model Training**
Graph Neural Network (GNN)	Model relationships between skills, learning paths, and career objectives via Knowledge Graphs	**Model Training**
Transformer Model (e.g., BERT)	Understand textual employee career goals and aspirations	**Model Training**
Precision, Recall, F1-Score, AUC-ROC	Evaluate model effectiveness and recommendation quality	**Model Evaluation**
k-Fold Cross-Validation	Validate model robustness and prevent overfitting	**Model Evaluation**
Explainability Methods (SHAP, LIME)	Interpret model predictions and detect potential biases	**Model Evaluation**
Feedback Integration and Deployment Preparation	Use best-performing model to generate IDPs; integrate feedback for retraining	**Deployment Preparation**

This research design ensures both methodological rigour and flexibility, allowing for fair comparison between models and supporting the goal of building an adaptable, scalable, and accurate IDP recommender system.

4.2. System Architecture

The system architecture for the Individual Development Plan (IDP) Recommender is designed to modularly process employee data, engineer useful features, independently train multiple deep learning models, evaluate their performance, and deploy the best model to generate personalized development plans.

Figure 4.1 illustrates the overall architecture of the system. The flow begins with raw employee data and proceeds through several critical stages: preprocessing, feature engineering, model training, evaluation, and selection. A feedback loop is incorporated to ensure that the model continues to improve over time based on new data and organizational feedback.

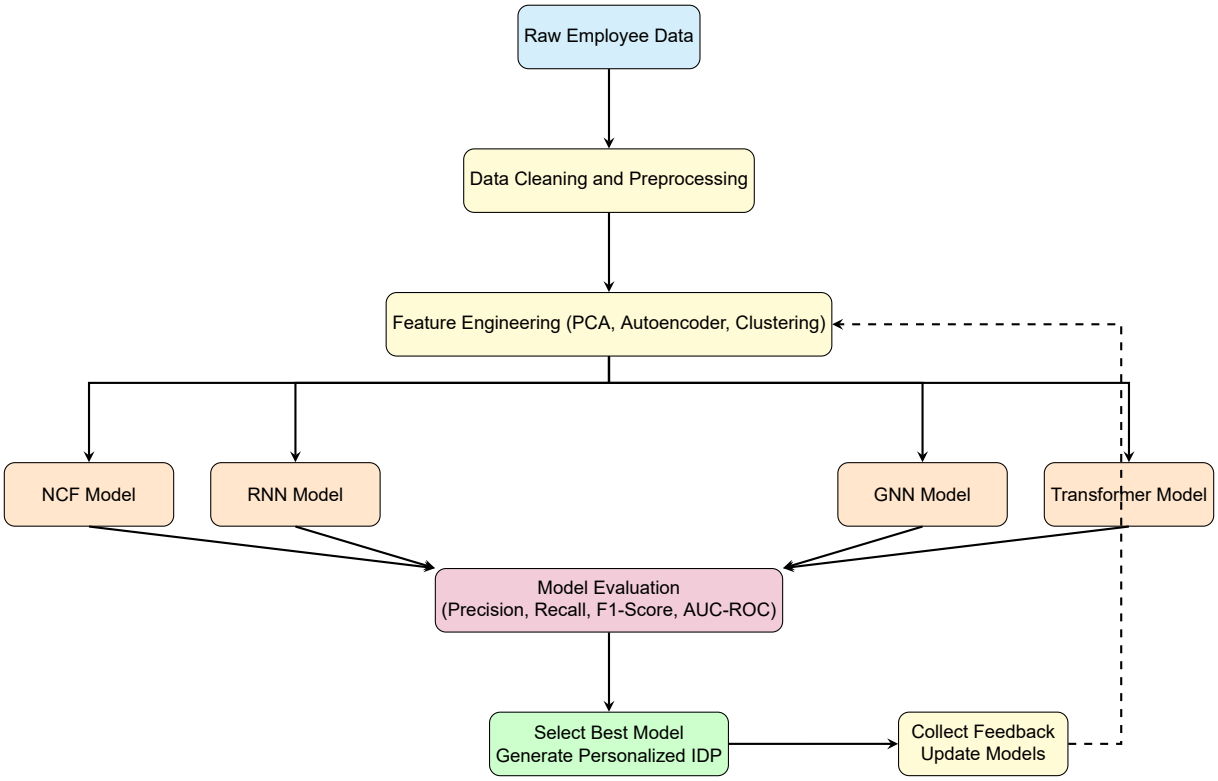

Figure 4.1: System Architecture for the Deep Learning-based IDP Recommender. The models (NCF, RNN, GNN, Transformer) are independently trained and evaluated to identify the best-performing model for generating personalized IDPs.

The system is composed of the following key modules:

4.2.1. Data Collection and Preprocessing

Employee data—including skills assessments, performance reviews, career goals, and feedback—is collected from organizational databases and manual sources. This raw data often contains inconsistencies, missing values, and unstructured formats. Therefore, a preprocessing module cleans the data by handling missing entries, standardizing formats, and preparing the inputs for feature engineering.

4.2.2. Feature Engineering

Feature engineering transforms raw data into structured formats suitable for deep learning models. Dimensionality reduction is performed using Principal Component Analysis (PCA) and Autoencoders to extract the most informative features while minimizing redundancy. Clustering techniques are also applied to group employees by similar career paths and competencies, providing additional categorical features.

4.2.3. Model Training

Four deep learning models—Neural Collaborative Filtering (NCF), Recurrent Neural Networks (RNN), Graph Neural Networks (GNN), and Transformer-based models—are independently trained using the engineered feature sets. Each model is designed to predict the most appropriate Individual Development Plan recommendations for a given employee profile.

4.2.4. Model Evaluation

The models are evaluated separately using consistent data splits (training, validation, and testing). Evaluation metrics include Precision, Recall, F1-Score, and AUC-ROC to provide a comprehensive view of each model's predictive performance. The best-performing model based on these metrics is selected for deployment.

4.2.5. Deployment and Feedback Integration

Once the best model is selected, it is used to generate personalized IDPs for employees. A feedback mechanism is implemented to collect post-recommendation evaluations from employees and supervisors. This feedback is used to update the training dataset and periodically retrain models, ensuring that the system adapts to evolving career development needs within the organization.

4.3. Module Descriptions

The system is divided into several specialized modules, each responsible for a key part of the IDP recommendation pipeline. This modular design ensures scalability, maintainability, and clear responsibilities across the system components.

4.3.1. Data Preprocessing Module

The Data Preprocessing Module is responsible for cleaning and standardizing the raw employee datasets. Key operations performed include:

- Handling missing data through imputation or removal strategies.
- Standardizing categorical variables (e.g., job titles, departments).
- Normalizing numerical features (e.g., skill ratings, evaluation scores).
- Parsing and cleaning unstructured textual fields, such as career goal descriptions.

This preprocessing ensures that the data fed into the Feature Engineering Module is consistent and machine-readable.

4.3.2. Feature Engineering Module

The Feature Engineering Module focuses on enhancing the dataset by extracting relevant features:

- **Principal Component Analysis (PCA):** Reduces the dimensionality of numerical data, capturing the most informative features.
- **Autoencoders:** Learn compact, noise-resistant representations of the original employee feature space.
- **Clustering (e.g., K-Means):** Groups employees with similar learning styles, skill gaps, or career aspirations, generating new categorical features.

The output from this module serves as the input for the model training phase.

4.3.3. Neural Collaborative Filtering (NCF) Model

The NCF Model extends traditional collaborative filtering by using a multi-layer perceptron (MLP) to learn nonlinear user-item interaction patterns. In this context, employees are treated as users and development plans as items. The model predicts the most suitable development plans based on historical training, assessment results, and career progressions.

Algorithm 1: Neural Collaborative Filtering (NCF)

Input: User-item interaction matrix $\mathbf{R}$
Output: Predicted user preferences $\hat{\mathbf{R}}$
begin
  Initialize embeddings for users and items
  Define neural network $f(\cdot)$ to model interaction between embeddings
  for *each epoch* **do**
    for *each user-item pair* (u, i) **do**
      Predict score: $\hat{r}_{ui} = f(\text{embedding}(u), \text{embedding}(i))$
      Compute loss: $L = \text{MSE}(r_{ui}, \hat{r}_{ui})$
      Update model parameters via backpropagation
    end
  end
end

4.3.4. Recurrent Neural Network (RNN) Model

The RNN Model is designed to capture sequential patterns in employee development histories. Employees' past training sessions, promotions, and evaluations form a temporal sequence, which the RNN learns to model. Long Short-Term Memory (LSTM) cells are used to mitigate the vanishing gradient problem and better capture long-term dependencies in employee progression paths.

Algorithm 2: Training and Prediction using Long Short-Term Memory (LSTM) Networks

Input: Sequential employee development data $\mathcal{D} = \{(x_t, y_t)\}_{t=1}^{T}$
Output: Trained LSTM model capable of predicting next development steps
begin
  Initialize LSTM parameters (weights, biases) randomly
  Set learning rate η
  for *each training epoch* **do**
    for *each employee sequence* $(x_1, \ldots, x_T)$ *in* $\mathcal{D}$ **do**
      Initialize hidden state $h_0 = 0$ and cell state $c_0 = 0$
      for *each time step* $t = 1$ *to* T **do**
        Compute input gate:
        $i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$
        Compute forget gate:
        $f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$
        Compute output gate:
        $o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$
        Compute candidate cell state:
        $\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$
        Update cell state:
        $c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$
        Update hidden state:
        $h_t = o_t \odot \tanh(c_t)$
        Compute output prediction:
        $\hat{y}_t = \sigma(W_{hy}h_t + b_y)$
      end
      Compute loss L between predicted outputs $\hat{y}_t$ and ground truth y_t
      Backpropagate error through time (BPTT) to compute gradients
      Update LSTM parameters using gradient descent:
      $\theta \leftarrow \theta - \eta \nabla_\theta L$
    end
  end
end

4.3.5. Graph Neural Network (GNN) Model

The GNN Model treats employee competencies, career goals, and development resources as nodes in a graph. Skills are connected based on prerequisite relationships or co-occurrence patterns. The GNN aggregates information from an employee's local skill neighborhood to predict personalized development plans, effectively leveraging relational data structures.

Algorithm 3: Graph Neural Network (GNN) for Skill Recommendation

Input: Graph $G = (V, E)$ with node features $\mathbf{H}^0$
Output: Updated node embeddings $\mathbf{H}^L$
begin
  for *layer* $l = 0$ **to** $L - 1$ **do**
    for *each node* $v \in V$ **do**
      Aggregate neighbor features:
      $\mathbf{h}_{\mathcal{N}(v)}^{(l)} = \text{AGGREGATE}^{(l)}(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\})$
      Update node representation:
      $\mathbf{h}_v^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_v^{(l)}, \mathbf{h}_{\mathcal{N}(v)}^{(l)})\right)$
    end
  end
end

4.3.6. Transformer Model

The Transformer Model is applied particularly to unstructured textual data, such as employee career goals or self-assessments. Utilizing self-attention mechanisms, the Transformer captures long-range dependencies and contextual relationships in the text, allowing the model to generate more contextually relevant development plan recommendations.

Algorithm 4: Transformer Encoder for Text Embeddings

Input: Tokenized text input $X = (x_1, \ldots, x_n)$
Output: Contextual embeddings $H = (h_1, \ldots, h_n)$
begin
  Embed tokens: $\mathbf{E} = \text{Embedding}(X)$
  Add positional encodings to $\mathbf{E}$
  for *each layer* **do**
    Compute multi-head self-attention:
    $Z = \text{MultiHead}(Q = E, K = E, V = E)$
    Apply feedforward network:
    $H = \text{FFN}(Z)$
  end
end

4.3.7. Model Evaluation Metrics

Each model's predictions are evaluated using standard classification metrics:

- **Precision:** Measures the proportion of recommended development plans that were relevant.
- **Recall:** Measures the proportion of relevant plans that were correctly recommended.
- **F1-Score:** The harmonic mean of Precision and Recall, balancing both concerns.
- **AUC-ROC:** Measures the trade-off between true positive and false positive rates across thresholds.

These metrics are used to select the best-performing model for deployment.

4.4. Experimental Design

The experimental design phase establishes the procedures for training, validating, and evaluating the deep learning models developed for the IDP recommender system. Each model—Neural Collaborative

Filtering (NCF), Recurrent Neural Network (RNN), Graph Neural Network (GNN), and Transformer—is independently trained and assessed to ensure fair performance comparisons.

4.4.1. Dataset Preparation

Employee data collected from organizational systems is preprocessed as described in Section 4.3. After cleaning and feature engineering, the dataset is split into three subsets:

- **Training Set (70%):** Used to train the models by minimizing the loss function.
- **Validation Set (15%):** Used for hyperparameter tuning and early stopping to prevent overfitting.
- **Testing Set (15%):** Used solely for final performance evaluation.

All data splits are stratified where applicable, ensuring balanced distributions of key employee attributes across the subsets.

4.4.2. Cross-Validation Strategy

In addition to a fixed train-validation-test split, 5-fold cross-validation is applied during hyperparameter tuning. This involves partitioning the training set into five folds and performing training and validation iteratively to assess the generalization capability of each model.

4.4.3. Model Training Details

Each deep learning model is trained independently using the prepared training data. The following general settings are applied:

- **Optimizer: Adam Optimizer** with an initial learning rate of 0.001.

  The Adam optimizer updates model parameters based on estimates of first (m_t) and second (v_t) moments of the gradients:

  $$\theta_{t+1} = \theta_t - \eta \times \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

  where:

  - θ_t are the model parameters at time step t,
  - η is the learning rate,
  - $\hat{m}_t$ is the bias-corrected first moment estimate (mean of gradients),
  - $\hat{v}_t$ is the bias-corrected second moment estimate (uncentered variance of gradients),
  - ϵ is a small constant for numerical stability.

- **Loss Function: Binary Cross-Entropy (BCE)** for recommendation tasks.

  The Binary Cross-Entropy loss function is defined as:

  $$L = -\frac{1}{N}\sum_{i=1}^{N} [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]$$

  where:

  - N is the number of samples,
  - y_i is the true label (0 or 1),
  - $\hat{y}_i$ is the predicted probability for the positive class.

- **Batch Size:** 64 samples per batch.

  During training, the model updates its weights based on the gradient computed from mini-batches of 64 samples, rather than using the entire training dataset at once (stochastic gradient descent).

- **Epochs:** 100, with early stopping after 10 consecutive epochs without validation loss improvement.

  An epoch is defined as one full pass through the entire training dataset. Early stopping is a regularization technique where training is halted if the model performance on the validation set does not improve for 10 consecutive epochs.

Hyperparameters such as learning rate, dropout rate, number of hidden layers, and embedding sizes are fine-tuned based on validation performance during cross-validation.

4.4.4. Evaluation Criteria

- **Precision (P):** Measures the proportion of correctly recommended development plans among all recommendations made.
$$P = \frac{TP}{TP + FP}$$
where TP = True Positives and FP = False Positives.
- **Recall (R):** Measures the proportion of relevant development plans successfully recommended.
$$R = \frac{TP}{TP + FN}$$
where FN = False Negatives.
- **F1-Score (F_1):** Harmonic mean of Precision and Recall, providing a balance between the two.
$$F_1 = 2 \times \frac{P \times R}{P + R}$$
- **AUC-ROC:** Area Under the Receiver Operating Characteristic Curve, measuring the ability of the model to distinguish between classes.

  The AUC-ROC is calculated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings:
$$\text{TPR} = \frac{TP}{TP + FN} \quad \text{and} \quad \text{FPR} = \frac{FP}{FP + TN}$$
  where TN = True Negatives.

Each model's performance is reported on the testing set, and the model achieving the best F1-Score and AUC-ROC is selected for deployment.

4.4.5. Software and Hardware Environment

The experiments are conducted using the following environments:

- Programming Language: Julia
- Deep Learning Libraries: Flux, DataFrames, Plots, Statistics
- Hardware: CPU-only training due to Hardware Constraints.

4.5. Technical Decisions Justification

This section provides the rationale behind the selection of models, methods, and techniques used in designing the deep learning-based IDP recommender system.

4.5.1. Why Deep Learning Approaches?

Traditional machine learning algorithms such as decision trees or support vector machines often struggle with high-dimensional, unstructured, and sequential data commonly found in employee development records. Deep learning models offer superior capabilities for:

- Automatically extracting complex features from large datasets.
- Capturing nonlinear relationships between employee attributes and development paths.
- Handling sequential data (e.g., training histories, performance progressions) through models like RNNs and LSTMs.
- Processing unstructured textual data such as career aspirations using architectures like Transformers.

Given these advantages, deep learning was chosen to maximize the predictive accuracy and flexibility of the IDP recommender system.

4.5.2. Why Neural Collaborative Filtering (NCF)?

NCF extends traditional collaborative filtering by replacing dot-product operations with neural networks, enabling the learning of complex user-item interaction patterns. In the context of IDPs:

- Employees are treated as users and recommended career development options as items.
- NCF can model intricate relationships between employee history and future development needs.
- It improves over matrix factorization methods by introducing non-linearity through hidden layers.

4.5.3. Why Recurrent Neural Networks (RNNs)?

Career development is inherently sequential—previous training, promotions, and evaluations influence future development plans. RNNs are selected because:

- They capture temporal dependencies in employee development trajectories.
- Using Long Short-Term Memory (LSTM) cells addresses the vanishing gradient problem, allowing modeling of long-term dependencies.
- Sequential modeling is essential for recommending development paths that logically follow an employee's career history.

4.5.4. Why Graph Neural Networks (GNNs)?

Skills, roles, and career goals form natural graph structures with prerequisite relationships and interdependencies. GNNs are appropriate because:

- They capture relational information between skills and learning objectives.
- They allow knowledge propagation across the graph, enhancing recommendations based on employee competencies and career objectives.
- They model interconnected skill frameworks more effectively than flat vector representations.

4.5.5. Why Transformer Models?

Employee career goals and self-assessments often involve natural language, which is unstructured and complex. Transformers, such as BERT, are ideal because:

- They leverage self-attention mechanisms to capture contextual relationships within text.
- They handle long-range dependencies without sequential bias, unlike RNNs.
- They have proven superior performance in many natural language processing (NLP) tasks relevant to understanding career aspirations.

4.5.6. Why Principal Component Analysis (PCA) and Autoencoders?

To reduce redundancy and enhance model training efficiency, dimensionality reduction techniques are applied:

- **PCA** projects features into a lower-dimensional space while preserving variance, useful for structured numerical data.
- **Autoencoders** learn compact, non-linear representations, beneficial when relationships among features are complex.

Both methods ensure that the input features are more manageable and that models can focus on the most informative aspects of employee data.

4.5.7. Why Standard Evaluation Metrics?

Metrics like Precision, Recall, F1-Score, and AUC-ROC are chosen because:

- They provide a comprehensive understanding of model performance.
- Precision and Recall address the correctness and completeness of recommendations.
- F1-Score balances Precision and Recall into a single figure of merit.
- AUC-ROC provides insight into model discrimination capability across decision thresholds.

4.6. Summary

This chapter presented the complete research design for developing and evaluating a deep learning-based recommender system for Individual Development Plans (IDPs).

The design is modular and data-driven, beginning with the collection and preprocessing of employee data, followed by feature engineering using dimensionality reduction and clustering techniques. Four deep learning models—Neural Collaborative Filtering (NCF), Recurrent Neural Network (RNN), Graph Neural Network (GNN), and Transformer—are independently trained and evaluated based on standard metrics such as Precision, Recall, F1-Score, and AUC-ROC.

Each model's architecture, training process, and evaluation strategy were explained in detail, including technical justifications for selecting deep learning approaches over traditional machine learning techniques. The experimental design ensures fair comparison across models through consistent data splitting, cross-validation, and hyperparameter tuning.

The best-performing model will ultimately be deployed within the IDP generation system, supported by a feedback loop for continuous system improvement.

The next chapter presents the results of the experiments and provides a detailed analysis of the models' performances.

5. Implementation

stuff

6. Results and Evaluation

7. Conclusion

7.1. Closing Remarks

7.2. Research Questions

The research questions posed in Chapter 1 are repeated below for convenience.

Research Question 1

What state-of-the-art methods are most suitable for XXX?

Reflect on research questions.

8. Recommendations

This chapter provides a brief overview of the primary recommendations for the future continuation of this research project.

Rec 1
Rec 2

References

Avison, D. E., Lau, F., Myers, M., & Nielsen, P. A. (1999). Action research. *Communications of the ACM*, *42*(1), 94–97.

Chen, L., & Zhong, J. (2024). Intelligent recommendation system for college english courses based on graph convolutional networks. *Heliyon*, *10*(8), e29052–e29052. doi:10.1016/j.heliyon.2024.e29052

Davison, R. M., Martinsons, M. G., & Malaurent, J. (2021). Research perspectives: Improving action research by integrating methods. *Journal of the Association for Information Systems*, *22*(3). doi:10.17705/1jais.00682

et al., O. (2021). Mixed methods appraisal tool: Strengthening the methodological rigor of mixed methods research studies in nursing. *Texto & Contexto - Enfermagem*. doi:10.1590/1980-265x-tce-2020-0603

Guetterman. (2016). What distinguishes a novice from an expert mixed methods researcher? *Quality & quantity*. doi:10.1007/s11135-016-0310-9

Hurley, E., Dietrich, T., & Rundle-Thiele, S. (2021). Integrating theory in co-design: An abductive approach. *Australasian Marketing Journal*, *29*(1), 66–77.

Irshaidat, R. (2019). Interpretivism vs. positivism in political marketing research. *Journal of Political Marketing*, *21*(2), 1–35. doi:10.1080/15377857.2019.1624286

Jung, H., Kim, H., Lee, I., & An, S. (2023). Designing a longitudinal database for cohort construction in medical education. *Korean Medical Education Review*, *25*(2), 84–101. doi:10.17496/kmer.23.012

Kelley, K., & Rausch, J. (2011). Sample size planning for longitudinal models: Accuracy in parameter estimation for polynomial change parameters. *Psychological Methods*, *16*(4), 391–405. doi:10.1037/a0023352

Li, W., Zhong, H., Zhou, J., Chang, C., Lin, R., & Tang, Y. (2024). An attention mechanism and residual network-based knowledge graph-enhanced recommender system. *Knowledge-Based Systems*, *299*, 112042–112042. doi:10.1016/j.knosys.2024.112042

Mantere, S., & Ketokivi, M. (2013). Reasoning in organization science. *Academy of Management Review*, *38*, 70–89.

Mu, R. (2018). A survey of recommender systems based on deep learning. *IEEE Access*, *6*, 69009–69022. doi:10.1109/access.2018.2880197

Myers, M. (2008). *Qualitative research in business & management*. Sage Publications.

Sahoo, A. K., Pradhan, C., Barik, R. K., & Dubey, H. (2019). Deepreco: Deep learning based health recommender system using collaborative filtering. *Computation*, *7*(2), 25. doi:10.3390/computation7020025

Saunders, M., Lewis, P., & Thornhill, A. (2012). *Research methods for business students* (6th). Pearson Education Limited.

Thompson, J. (2022). A guide to abductive thematic analysis. *The Qualitative Report*, *27*(5), 1410–1421. doi:10.46743/2160-3715/2022.5340

Vanderford, N., Evans, T., Weiss, L., Bira, L., & Beltran-Gastelum, J. (2018). A cross-sectional study of the use and effectiveness of the individual development plan among doctoral students. *F1000research*, *7*, 722. doi:10.12688/f1000research.15154.2

A. Algorithms

This appendix contains an example algorithm.

Algorithm 5: Principal Component Analysis (PCA)

Input: Dataset $\mathbf{X} \in \mathbb{R}^{n \times d}$ (n samples, d features)
Output: Reduced representation $\mathbf{X}_{\text{reduced}}$
begin
  Center the data: $\mathbf{X} \leftarrow \mathbf{X} - \text{mean}(\mathbf{X})$
  Compute covariance matrix: $\mathbf{C} = \frac{1}{n-1}\mathbf{X}^T\mathbf{X}$
  Compute eigenvectors and eigenvalues of $\mathbf{C}$
  Sort eigenvectors by descending eigenvalues
  Select top k eigenvectors to form matrix $\mathbf{W}$
  Project data: $\mathbf{X}_{\text{reduced}} = \mathbf{XW}$
end

Algorithm 6: Autoencoder Training for Feature Extraction

Input: Training data $\mathbf{X}$
Output: Encoded representation $\mathbf{Z}$
begin
  Define encoder network f_θ and decoder network g_ϕ
  Initialize parameters θ, ϕ
  for *each epoch* **do**
    for *each mini-batch* **do**
      Encode: $\mathbf{Z} = f_\theta(\mathbf{X})$
      Decode: $\hat{\mathbf{X}} = g_\phi(\mathbf{Z})$
      Compute reconstruction loss: $L = \|\mathbf{X} - \hat{\mathbf{X}}\|^2$
      Update θ, ϕ using gradient descent to minimize L
    end
  end
end

Algorithm 7: Neural Collaborative Filtering (NCF)

Input: User-item interaction matrix $\mathbf{R}$
Output: Predicted user preferences $\hat{\mathbf{R}}$
begin
  Initialize embeddings for users and items
  Define neural network $f(\cdot)$ to model interaction between embeddings
  for *each epoch* **do**
    for *each user-item pair* (u, i) **do**
      Predict score: $\hat{r}_{ui} = f(\text{embedding}(u), \text{embedding}(i))$
      Compute loss: $L = \text{MSE}(r_{ui}, \hat{r}_{ui})$
      Update model parameters via backpropagation
    end
  end
end

Algorithm 8: Graph Neural Network (GNN) for Skill Recommendation

Input: Graph $G = (V, E)$ with node features $\mathbf{H}^0$
Output: Updated node embeddings $\mathbf{H}^L$
begin
  for *layer* $l = 0$ **to** $L - 1$ **do**
    for *each node* $v \in V$ **do**
      Aggregate neighbor features:
      $\mathbf{h}_{\mathcal{N}(v)}^{(l)} = \text{AGGREGATE}^{(l)}(\{\mathbf{h}_u^{(l)} : u \in \mathcal{N}(v)\})$
      Update node representation:
      $\mathbf{h}_v^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \text{CONCAT}(\mathbf{h}_v^{(l)}, \mathbf{h}_{\mathcal{N}(v)}^{(l)})\right)$
    end
  end
end

Algorithm 9: Transformer Encoder for Text Embeddings

Input: Tokenized text input $X = (x_1, \ldots, x_n)$
Output: Contextual embeddings $H = (h_1, \ldots, h_n)$
begin
  Embed tokens: $\mathbf{E} = \text{Embedding}(X)$
  Add positional encodings to $\mathbf{E}$
  for *each layer* **do**
    Compute multi-head self-attention:
    $Z = \text{MultiHead}(Q = E, K = E, V = E)$
    Apply feedforward network:
    $H = \text{FFN}(Z)$
  end
end

Algorithm 10: Deep Learning-based IDP Recommender System Pipeline

Input: Raw employee data: skills assessments, job descriptions, performance reviews, career goals

Output: Personalized Individual Development Plans (IDPs)

begin

Step 1: Data Collection and Preprocessing

Collect employee datasets (skills, evaluations, feedback, goals)

Clean and normalize data (handling missing values, text cleaning)

Extract features from structured data and unstructured text using NLP

Step 2: Feature Engineering

Apply PCA for dimensionality reduction

Train Autoencoder to learn compact feature representations

Perform Clustering to identify similar employee profiles

Step 3: Model Training

Sub-step 3.1: NCF-based Interaction Modeling

Train Neural Collaborative Filtering (NCF) to predict personalized recommendations based on historical data

Sub-step 3.2: Graph Modeling

Construct Knowledge Graph of skills, learning paths, career objectives

Train Graph Neural Network (GNN) to capture relational dependencies

Sub-step 3.3: Textual Understanding

Train Transformer model (e.g., BERT) on employee goal statements

Generate embeddings for career goal descriptions

Step 4: Inference and Recommendation

For a given employee:

- Embed structured and unstructured data
- Aggregate predictions from NCF, GNN, and Transformer models
- Rank and select top development plan recommendations

Generate customized IDP output for the employee

Step 5: Feedback Integration

Collect employee and supervisor feedback post-recommendation

Update training datasets and retrain models periodically for dynamic adaptation

end