Ministry of Science and Higher Education of the Republic of Kazakhstan

SDU University

Assemay Imankulova

Comprehensive Analysis of a Recommender System for Career Guidance

THESIS

Presented in Partial Fulfilment for the

Degree of Master in Computer Science

(degree code: 7M06102)

Department of Computer Science

Faculty of Engineering and Natural Sciences

Supervisor: **Meraryslan Meraliyev**

Kaskelen, June 2024

SDU University
Faculty of Engineering and Natural Sciences
Department of Computer Science

Dean of Faculty of Engineering and Natural Sciences

Assistant Professor, PhD Akhmedov Ramis

____

« 04 » 06 2024

Topic of the thesis:

Comprehensive Analysis of a Recommender System for Career Guidance

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science", SDU University

Head of Department Zhanar Mukash

Academic Supervisor Meraryslan Meraliyev

Master student Assemay Imankulova

Kaskelen, 2024

Declaration

I confirm that this is my own work and the use of all material from other sources has been properly and fully acknowledged.

Assemay Imankulova

June 2024

Acknowledgements

My supervisor, Merarislan Meraliev, has provided me with essential advice, encouragement, and support during the completion of my master's thesis, for which I am truly grateful. His expertise, tolerance, and perceptive criticism were invaluable in helping to mold and elevate this work. I consider myself really lucky to have had a mentor who was so committed to helping me negotiate the challenges of academic research and who gave me the skills and information I needed.

Dedication

This thesis is dedicated to:
My dear family, who have always been a constant source of inspiration and strength because of their unwavering belief in me and unwavering support. Your support and affection have meant the world to me.
Additionally, I want to sincerely thank my coworkers for their patience and encouragement during this trip.

Abstract

Choosing a suitable specialty is crucial for students, influenced by personal interests, academic performance, and career prospects. However, many struggle due to a lack of clear guidance and information overload. This research proposes a recommendation system to help students choose appropriate specialties and elective courses based on their academic performance and grades.

The study focuses on IT students, as this field offers a wide range of specialties and career opportunities. It utilizes machine learning techniques, including reinforcement learning algorithms, to analyze academic data and provide personalized recommendations. The study compares traditional machine learning algorithms like Decision Tree, Support Vector Machine, and Random Forest with reinforcement learning algorithms such as Q-learning and Deep Q-Network.

The methodology involves data collection, preparation, and feature engineering, followed by implementing various classifiers to build the recommendation system. Results indicate that Q-learning achieves the highest accuracy in recommending specialties, outperforming other algorithms. However, traditional machine learning algorithms also show competitive performance, suggesting both approaches can be effective.

This research contributes to educational technology by offering a practical solution to help students make informed academic and career decisions. Future work includes enhancing the recommendation system with real-time data and user feedback mechanisms to improve its effectiveness and usability.

Аңдатпа

Мамандықты дұрыс таңдау студенттер үшін өте маңызды және оған жеке қызығушылықтар, оқу үлгерімі және мансаптық перспективалар әсер етеді. Дегенмен, көптеген жастар нақты нұсқаулар мен ақпараттың шамадан тыс жүктелуіне байланысты қиындықтарға тап болады. Зерттеу студенттерге олардың академиялық үлгерімі мен бағалары негізінде сәйкес мамандықтар мен таңдау пәндерін таңдауға көмектесетін ұсынымдар жүйесін ұсынады.
Бұл жұмыс ақпараттық технологиялар мамандығын таңдаған студенттеріне бағытталған, өйткені бұл сала көптеген мамандықтар мен мансаптық мүмкіндіктерді ұсынады. Ол академиялық деректерді талдау және жекелендірілген ұсыныстар беру үшін машиналық оқыту әдістерін, соның ішінде күшейтетін оқыту алгоритмдерін пайдаланады. Зерттеу шешім ағашы, қолдау векторлық машинасы және кездейсоқ орман сияқты дәстүрлі машиналық оқыту алгоритмдерін Q-learning және Deep Q-Network сияқты күшейтетін оқыту алгоритмдерімен салыстырады.
Әдістеме ұсынымдар жүйесін құру деректерді жинаудан, дайындаудан және әзірлеуден тұрады. Дайын деректерді әртүрлі классификаторларға енгізуді қамтиды. Нәтижелер Q-learning басқа алгоритмдерден озып, мамандық бойынша ұсыныстарда ең жоғары дәлдікті қамтамасыз етті. Дегенмен, машинамен оқытудың дәстүрлі алгоритмдері де бәсекеге қабілетті өнімділікті көрсетті, бұдан екі тәсілдің де тиімді болуы мүмкін екенін көре аламыз.
Бұл зерттеу студенттерге негізделген академиялық және мансаптық шешімдер қабылдауға көмектесетін практикалық шешімді ұсына отырып, білім беру технологиясына ықпал етеді. Болашақ жұмыс ұсынымдар жүйесінің тиімділігі мен ыңғайлылығын арттыру үшін нақты уақыттағы деректермен және пайдаланушының кері байланыс механизмдерімен ұсыныстар жүйесін кеңейтуді қамтиды.

Аннотация

Выбор подходящей специальности имеет решающее значение для студентов, на него влияют личные интересы, успеваемость и перспективы карьерного роста. Однако многие испытывают трудности из-за отсутствия четких указаний и информационной перегрузки. В исследовании предлагается система рекомендаций, которая поможет студентам выбрать подходящие специальности и факультативные курсы на основе их успеваемости и оценок.
Исследование ориентировано для студентов специальности информационных технологии, поскольку эта область предлагает широкий спектр направлении и возможностей карьерного роста. Он использует методы машинного обучения, в том числе алгоритмы обучения с подкреплением, для анализа академических данных и предоставления персонализированных рекомендаций. В исследовании сравниваются традиционные алгоритмы машинного обучения, такие как дерево решений, машина опорных векторов и случайный лес, с алгоритмами обучения с подкреплением, такими как Q-learning и Deep Q-Network.
Методология включает сбор, подготовку и разработку функций с последующей реализацией различных классификаторов для построения системы рекомендаций. Результаты показывают, что Q-learning обеспечивает высочайшую точность рекомендации специальностей, превосходя другие алгоритмы. Однако традиционные алгоритмы машинного обучения также демонстрируют конкурентоспособную производительность, что позволяет предположить, что оба подхода могут быть эффективными.
Это исследование вносит вклад в образовательные технологии, предлагая практическое решение, которое поможет студентам принимать обоснованные академические и карьерные решения. Будущая работа включает в себя расширение системы рекомендаций за счет данных в реальном времени и механизмов обратной связи с пользователями для повышения ее эффективности и удобства использования.

Abbreviations

ANN – Artificial Neural Network

AP – Average Precision

API – Application Programming Interface

AUC – Area Under the Curve

Bi-LSTM – Bi-Directional Long Short-Term Memory

BPA – Business Process Analyst

CAR – Classification based on Association Rules

CRS – Course Recommendation Systems

CSV – Comma-Separated Values

CFS – Correlation-based Feature Selection

CV – Curriculum Vitae

DBSCAN – Density-Based Spatial Clustering of Applications with Noise

DT – Decision Trees

DQN – Deep Q Network

ELU – Exponential Linear Units

FIFO – First In, First Out

GBC – Gradient Boosted Classifier

GBT – Gradient Boosted Trees

GBDT – Gradient Boosting Decision Trees

GNNs – Graph Neural Networks

GraphSAGE – Graph Sample and Aggregation

HCN – Hybrid Convolutional Neural Network

IT – Information Technology

IOT – Internet of Things

JSON – JavaScript Object Notation

KNN – K-Nearest Neighbors

LSTM – Long Short-Term Memory

MBTI – Myers-Briggs Type Indicator

MAE – Mean Absolute Error

MEDLINE – Medical Literature Analysis and Retrieval System Online

ML – Machine Learning

MLE – Machine Learning Engineer

Naive Bayes – Naive Bayes Classifier

NLTK – Natural Language Toolkit

NLP – Natural Language Processing

OASiS – Optimized Asynchronous Scheduling System

Pandas – Python Data Analysis Library

PDF – Portable Document Format

PSR – Professional Social Recommender

QL – Q-learning

RF – Random Forest

ROC – Receiver Operating Characteristic

RL – Reinforcement Learning

RMSE – Root Mean Squared Error

ROI – Return on Investment

RRH – Resource Ratio Based Heterogeneous

Saas – Software as a Service

SMOTE – Synthetic Minority Over-sampling Technique

SVM – Support Vector Machines

SVD – Singular Value Decomposition

TGN – Temporal Graph Networks

TF-IDF – Term Frequency-Inverse Document Frequency

URL – Uniform Resource Locator

JSON – JavaScript Object Notation

Table of Contents

Chapter 1

Background and motivations

1.1 Introduction

The 21st century has become the century of technology and opportunity, with rapid development in various fields such as medicine, science, education, business, and other areas of human activity. The educational system should also utilize the development of technology in the field of career guidance.

When choosing a university or a future profession, the applicant may feel responsible for their life, and various doubtful thoughts may haunt them. According to [1], these thoughts can lead to self-doubt and may reflect in the academic success of young people. These doubts can also haunt students after graduation if they made the wrong choice, leading to regret. Regret is an emotion that comes from our thoughts, as we think that it would be better if we chose another option [2]. Unfortunately, no one can go back in time, and students may not dare to leave university and start over due to various problems, the most common being financial.

Also in the rapidly developing age of technology, there are many different directions, especially in IT. Such a large number of directions also confuse young guys and even put pressure on them. Since many of them may not know their strengths and weaknesses, and not see their potential. This can lead to more regrets or even anger and stress towards oneself, which in the future will increase dissatisfaction and lower self-esteem.

Therefore, based on this, we have decided to create a recommendation system to help students choose the most suitable specialty based on their academic performance and grades at school. After entering university, the system will help them choose elective courses that will become the foundation for them and help them find a job with the necessary knowledge base.

As the right choice of specialty and finding a job after graduation are the goals of young people, this tool will be essential. Nowadays, it is challenging to find a job without work experience or the required knowledge. Employers usually prefer hiring candidates with the necessary knowledge and experience. Additionally, many young people who have completed their studies decide not to work in their field of study, either because they chose the wrong field or leave the university without graduating. According to statistics from 2022, 112 thousand students left university without graduating [3], and 60% of students do not work in their specialty after graduation [4]. Therefore, young people need such a tool as a recommendation system that can guide them towards the right choice of specialty.

1.2 Problem statement

Unfortunately, many individuals are unsure about their desired career path and often choose a profession based on the opinions and recommendations of family members, such as parents, rather than their own interests and talents. This lack of clarity can lead to individuals pursuing careers that do not align with their true passions and abilities. Even if they have a natural aptitude for art or design, someone born into a family of doctors, for example, could feel pressured to pursue a career in medicine like their parents did. Feelings of discontent and career unfulfillment might arise from the pressure to live up to family expectations.

Young people's access to career advising is sometimes restricted to university or college marketing, which could not give them a thorough grasp of the career path they want to take. Students may make poor decisions about their future jobs as a result of their restricted exposure. The lack of ongoing support provided by traditional career counseling techniques might also deprive people of the information they need to make wise career decisions. The lack of ongoing support and information can result in individuals feeling lost and unsure about their career paths.

Choosing the right profession can have a significant impact on various aspects of one's life, including job satisfaction, financial stability, and personal fulfillment. It is a decision that will affect their entire career unless they choose to change it. Opting for a suitable profession can make it easier for an individual to achieve success in their career, as they are more likely to be motivated and engaged in their work. Similarly, selecting appropriate university courses can save time and effort, and help students become well-prepared professionals in the job market, armed with the necessary knowledge and skills. While work experience is valuable, the foundation of one's profession is established by initial education.

Therefore, it is hoped that the proposed recommendation system will act as a bridge between the university and the school, providing a clear understanding of the chosen profession and guiding students towards the right path. By offering comprehensive and continuous support, the recommendation system can help individuals make informed decisions about their careers, leading to greater job satisfaction and personal fulfillment..

1.3 Aim and Objectives of the Research

Aim of the Research:

Develop a RL based recommendation system for IT professions students by May 2024 and compare it with other methods.

Objectives of the Research:

- Find and prepare a dataset.
- Implement machine learning algorithms such as Decision Trees, Support Vector Machines, and Random Forest algorithms to develop the initial recommendation system.
- Develop and integrate a reinforcement learning-based component such as Q-learning and Deep Q-Network into the recommendation system framework.
- Analyze the results obtained from the comparison to identify the most effective and efficient algorithmic approach for the recommendation system.

1.4 Research Question

How can new technologies help young people choose the most suitable profession?

Chapter 2

Literature review

2.1 Machine learning

In the realm of recommendation systems based on machine learning, several studies have contributed significant advancements. For instance, in [5] the authors made a recommendation system based on machine learning. This is a web application that processes and analyzes data using a specific hybrid model. In their project, the user answers questions and the program recommends based on the provided skills and academic performance. The authors created the database on their own using courses after the 10th, 12th grade and after graduation. After the collection is completed, this data is pre-processed in the machine learning format.The authors' methodology is to use a hybrid format of collaborative filtering and content-based filtering (Figure 2.1).

Collaborative filtering recommendation is a method that involves filtering information from collections of data from related users to provide recommendations. This technique is widely used in recommendation systems and can be categorized into different types such as user-based, item-based, and model-based collaborative filtering methods [6]. Collaborative filtering algorithms are essential in providing recommendations based on user preferences and behaviors, and they can be enhanced by incorporating various approaches.

A content-based filtering recommendation is a method where the system filters information for you based only on your characteristics and preferences [7]. Content-based recommendation algorithms establish interest descriptions based on user behavior to recommend similar items [8].

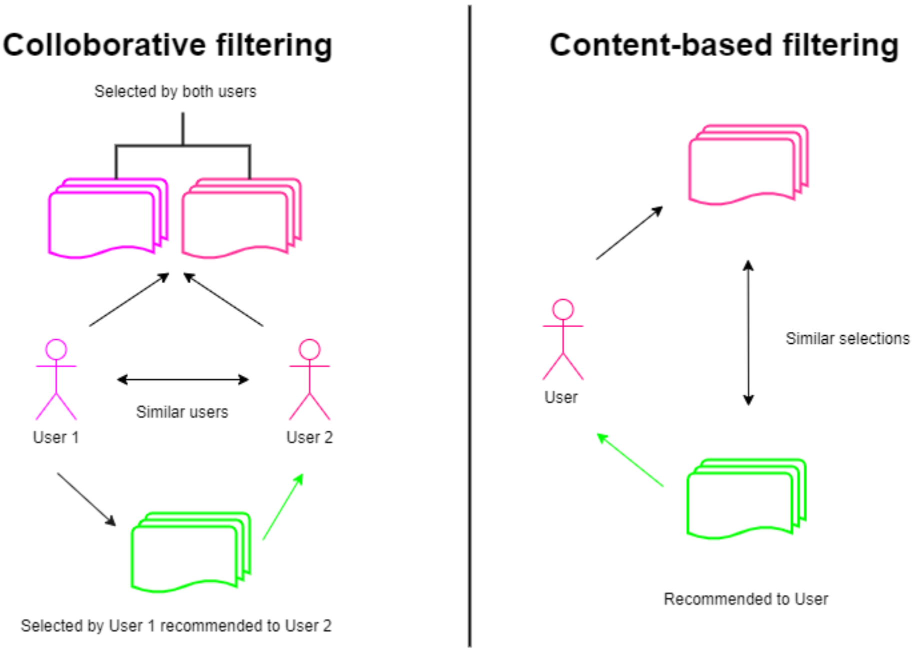

Figure 2.1 – Collaborative and Content-based filtering [9]

Building on this, [10] implemented a recommendation system in Saudi Arabia for IT students. You can view the methodology flowchart in Figure 2.2, which illustrates the use of a questionnaire for data collection. Thanks to Linkedin, they were able to collect 2255 responses and use 2167 entries.

In the pre-processing of the data, the errors of the participants were corrected, divided into categories. After the study, 49 programming languages were counted in the questionnaire, and the authors made it so that if the participant knew the language they put "1", if they did not know, they put "0".

In the section feature engineering, authors removed some data like gender, nationality, job location, etc. After that, they added encodings for soft skill, hard skill, degree and specialty. To help choose the most suitable profession, the authors used 5 different python algorithms., like: K-Nearest Neighbors, Decision Tree, Baggling meta-estimator, Gradient Boosting, and XGBoost. After testing, XGBoost was found to be better than others with 70.47% accuracy on a balanced dataset.

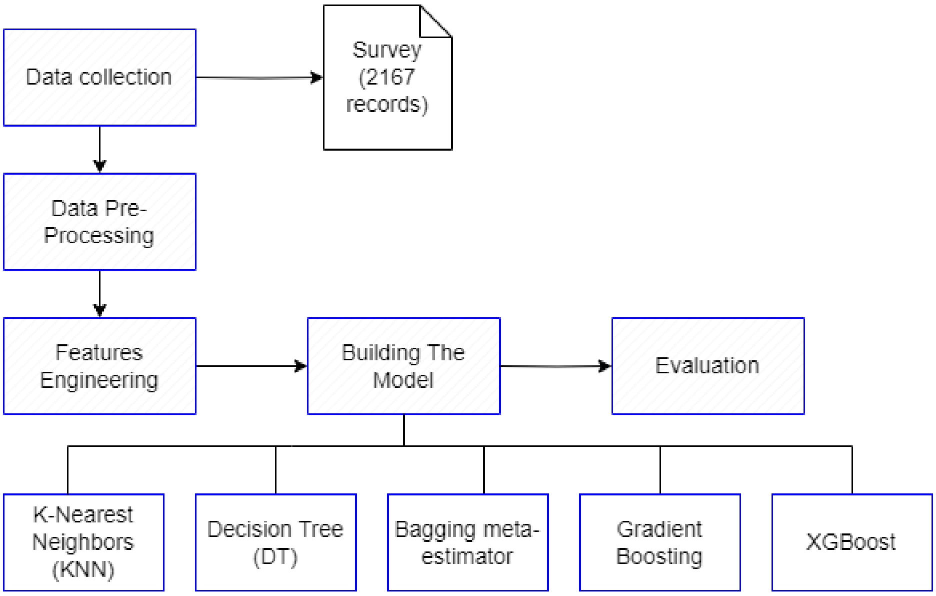

Figure 2.2 – Flowchart of the methodology in research

In another study [11], Saudi Arabian authors have developed a learning model that recommends job applicants based on resume content. For the data set, the authors used a combination of BeautifulSoup and Selenium. All data is saved as a JSON file by scraping the website.
Four methods were used in data preprocessing:

1. Clear tags
2. Tokenization
3. Lemmatization
4. Stop words

By leveraging Natural Language Processing, the system is capable of matching the skills of applicants with the job requirements listed in the resumes and job descriptions. But the most important part is to determine the similarities of the text. The authors chose the word2vec method, a vector space model. This approach involves feeding a vast collection of words into the model, then creates a vector space where you can determine how similar words are, using similarity measures: Jaccard coefficient, Cosine similarity. But as it turned out, Cosine similarity is more accurate than the Jaccard Coefficient.
As a result, the authors have a system where the user can upload a resume, and get the 5 most relevant jobs with a description.

Similarly in [12], authors created a career guidance system as a web site, where users answer questions about their psychological characteristics and skills. The system consists of three modules as you can see in Figure 2.3 .

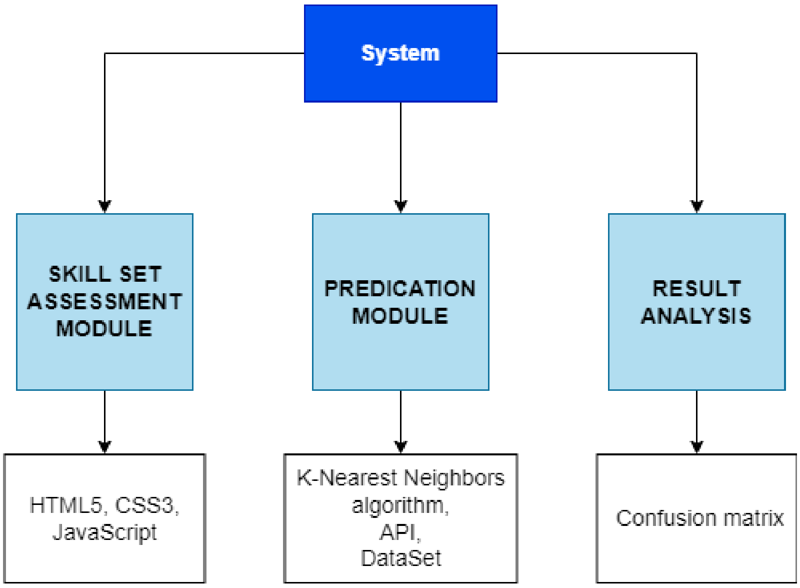

Figure 2.3 – System modules

The main module among three modules is the second module, where you can see that the framework uses Python to build the prediction module, which includes machine learning algorithms like K-Nearest Neighbors for classification and K-Means Clustering for clustering.

The Flask API facilitates the interaction or data exchange between the front-end and back-end elements. The dataset used for the model was manually created with numerical values and includes seven different skills, with some being considered core for certain departments. The dataset has five different target labels for specific departments, with 80% used for training and 20% for validation and testing. To analyze the results and determine performance, a confusion matrix is employed, which is a tabular representation of: namely true positive, true negative, false positive, and false negative.

In [13], authors collect data using google form. In data preprocessing they removed unnecessary data such as: nationality, gender,email. Next, the data was properly formatted and utilized for the modeling process. The authors utilized four ML algorithms, including Decision tree, Naive Bayesian, K-nearest neighbor, and Support Vector Machines, as depicted in Figure 4, to construct their model.

Authors divide the dataset 2 parts: 90% for training and 10% testing . The testing dataset used in evaluation. In order to check performance authors as in [12], use a confusion matrix. If the model performs well, it can be deployed. To improve the performance of classification algorithms used ensemble techniques, such as bagging classifiers. Bagging classifiers involves splitting the dataset into multiple training subsets, training a classifier on each subset, and combining the results to produce better overall performance than a single classifier (Figure 2.4).

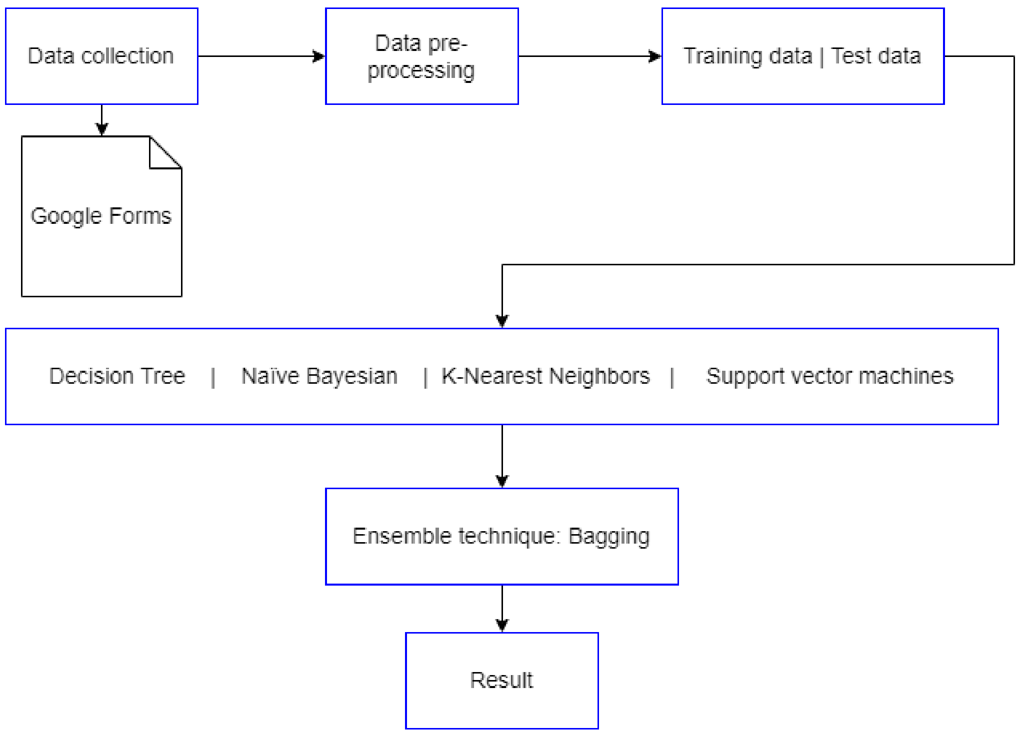

Figure 2.4 – Flowchart of the methodology in research

Continuing this trend, [14] authors used a portal where you can see student grades to collect data. Also, through a questionnaire, they collected students' interests, such as hobbies, sports, etc. The authors saved all the information in a database and utilized the Artificial Neural Network Algorithm (ANN) to develop a predictive model.This algorithm is similar to the human brain, where communication occurs through signals (see Figure 2.5).

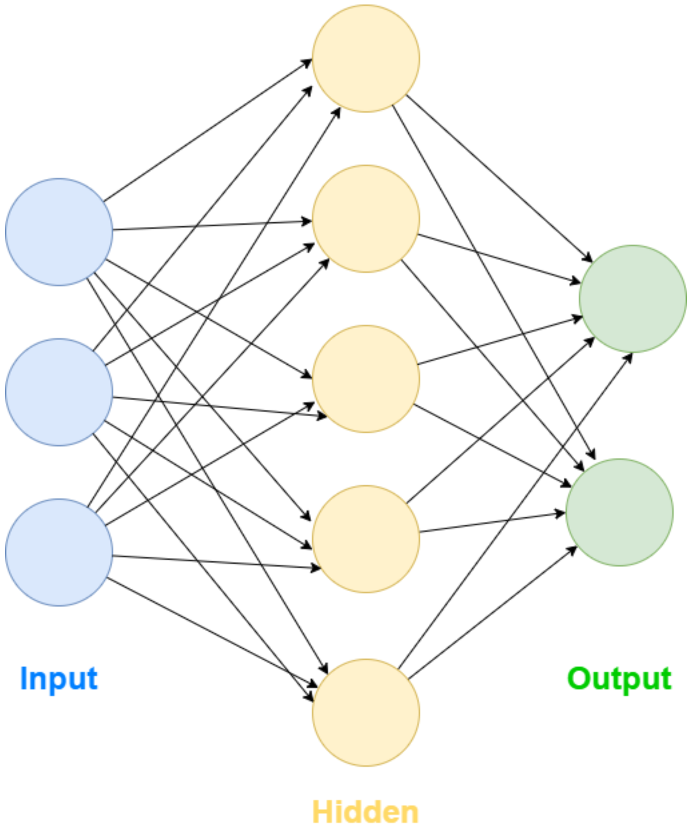

Figure 2.5 – Architecture of ANN

In this research work[15], the authors employed a questionnaire-based approach to collect data from 977 IT students in Lagos state, Nigeria. The data collection encompassed demographic information, company details, and IT experience, providing a comprehensive dataset for analysis. Five key attributes were selected for recommendation modeling: company location, field, allowance, acceptance status, and retaining status. The decision tree algorithm C4.5 was utilized to construct a predictive model based on these attributes. The accuracy of the model was evaluated using training data, achieving a classification accuracy of 78.84%. The model's effectiveness in predicting IT firm recommendations for students was further validated by additional evaluation metrics, including the Kappa statistic, mean absolute error, and root mean squared error.

The outcomes show that the constructed model has a significant predictive capacity, with a high degree of agreement between the predicted and actual class labels. The precision, recall, F-measure, MCC, ROC area, and PRC area are among the evaluation measures that consistently demonstrate the accuracy and dependability of the model. The technique provides a clear picture of the implementation process through its activity diagrams and system architecture, which improve the reproducibility and comprehension of the offered recommender system.

In addition, [16] provided a comprehensive methodology that places an emphasis on pre-processing, system architecture, and data collection in order to develop a system that suggests careers to high school students. The dataset had a variety of characteristics, including gender, age, parent income, and final grade, among others.It was collected utilizing a standardized questionnaire from senior high school students in the province of Isabela. The data were processed and divided into test, validation and training sets with a view to facilitating the design and evaluation of models. The system architecture has been enhanced through the integration of a fuzzy inference mechanism and five filtering techniques to offer students career options that align with their distinct characteristics. The study outcomes support that the fuzzy model developed had a high level of effectiveness in predicting student career preferences.

The evaluation of the efficiency of the fuzzy model in career counseling yielded a positive outcome as indicated by its mean absolute error (MAE) and root mean square error (RMSE). This suggests that it has dependable prediction abilities. Additionally, reducing RMSE between training and test sets showed stability within the model to improve accuracy predictions. In summary, this study presents essential knowledge on utilizing fuzzy logic to direct high school students towards appropriate professional paths considering their individual attributes and preferences.

The authors of [17] developed a recommendation system that employs tab-separated files to evaluate input datasets. Their approach involves four modules: data collection, transformation, computation and suggestion. The system utilizes various methods such as cosine similarity along with item-item collaborative filtering and user-user collaborative filtering to provide recommendations based on either matching or content-based filters. Users were given employment recommendations based on two criteria (state and city), and the method produced results based on popularity.

Collaborative filtering was found to be more appropriate in delivering recommendations as the size of the data set increased. The Root Mean Square Error (RMSE) decreased with dataset size, according to an investigation of the relationship between the two. This suggests that the recommendation system was growing more accurate in its predictions.

Moreover, [18] provided a thorough technique with an emphasis on data segmentation, tagging, skill extraction, and graph generation for creating skill graphs from user profiles. Modules for skill extraction, skill graph construction, and data segmentation and labeling are included in the system architecture. Key data including candidate details, education, and experience are extracted by the data segmentation module from user profiles processed in a variety of forms, including Word, PDF, and social networking URL text. The skill extraction module uses a probabilistic method to extract skills from candidate profiles and makes use of a skill ontology constructed from many data sources. The system's performance was evaluated using precision, recall, and accuracy metrics. The findings demonstrated that the system could extract abilities with an accuracy of 0.887% from user profiles.

The skill graph generator module uses the retrieved and augmented skill data to build a sequence graph of the candidate's talents, allowing many skill routes to be examined. The skill graph makes it possible to identify essential skill transitions for professional advancement and shows the chronological order in which abilities were learned. In order to improve the precision and applicability of skill recommendations, the study additionally addresses the enrichment of extracted skills with variables related to competence, parent-child ties, and similarity. All things considered, the suggested methodology offers a strong framework for creating skill graphs from user profiles, providing insightful information for skill improvement and career growth.

The proposed framework [19], introduces an innovative approach to coordinating and collecting relevant data from the vast amount of information available on the Internet, particularly focusing on educational content such as course providers, course contents, and vocational information. The framework addresses the limitations of traditional web crawlers, which often return numerous irrelevant pages along with relevant ones due to topic-specific elements. By leveraging an ontology-based web crawler, the system aims to improve the accuracy and effectiveness of information retrieval, especially for students seeking educational resources. Preprocessing user queries, giving words semantics based on ontological reasoning, building a hierarchy of semantic correlations, and making decisions with fuzzy logic are all part of the framework's methodology. Through the use of a hybrid recommendation technique that combines content-based suggestion with collaborative filtering, the system is intended to deliver highly relevant material to users with increased accuracy.

The preprocessing and M-Tree generation algorithms, among other algorithmic parts of the framework, are intended to improve the efficacy and efficiency of the recommendation system. By removing stopwords, identifying stemming substrings, and joining words to create a coherent string, the preprocessing technique aims to improve user queries. However, in order to create a tree structure that makes it easier to rank crawled data based on cosine similarity, the M-Tree building technique is essential. The M-Tree approach has a temporal complexity of O(log(n)), where n is the number of occupations in the IT industry.This implies that the approach is scalable and successful. The system's performance is evaluated using metrics such as Mean Absolute Error (MAE) and Normalized Mean Absolute Error (NMAE). The findings demonstrate the system's high level of accuracy and effectiveness in suggesting career possibilities. All things considered, the proposed framework provides a comprehensive and innovative method for developing a career development system. Fuzzy logic, ontologies, and sophisticated algorithms are utilized to give pupils well-informed counsel and advice while making decisions about their futures.

An autoencoder algorithm was used in the study [20] to analyze in detail a personalized course recommendation system in an online learning environment (see autoencoder model in Figure 2.6). User ratings were predicted using autoencoder. Based on historical ranking data, a prediction was made by recommending courses based on student evaluations. The study assessed the autoencoder model's performance in comparison to conventional techniques. Metrics like MAE and RMSE were used by the authors to compare the KNN, SVD, SVD++, and NMF algorithms.

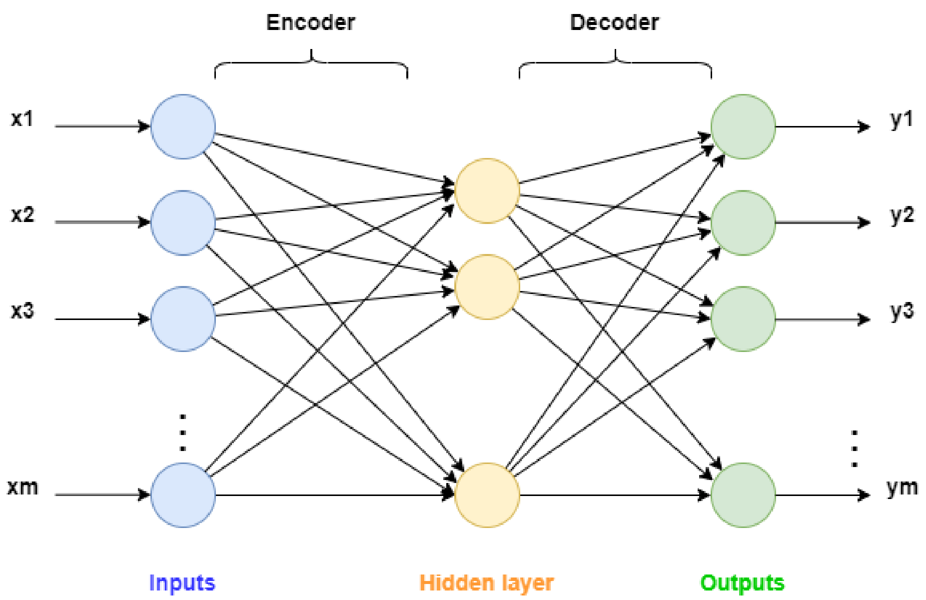

Figure 2.6 – Autoencoder model

The lowest MAE and RMSE values show that the autoencoder model outperformed conventional techniques. The autoencoder model's enhanced scalability, its capacity to manage sparse data, and its capacity to recognize intricate patterns in user interactions with products to produce more precise predictions all contributed to the improvement of course evaluations. The study demonstrated how effective recommendation systems for online learning environments may be made using the autoencoder algorithm. By choosing the appropriate courses, these technologies can enhance students' entire educational experience.

Researchers in the paper [21] used information from an online survey of variables that influence students' decisions about their university major. The accuracy of their university program recommendation system improved because the researchers used feature engineering and metaheuristic selection to refine important qualities. The Gradient Boosted Trees (GBT) model achieved a classification accuracy of 82.88%. This score is comparatively better when compared to Decision Trees (DT) and Random Forests (RF). However, the processing time of the GBT model was relatively high at 1 minute 41 seconds. This indicates the potential for improvement through hybridization with metaheuristics.

The study assessed the effectiveness of several feature selection techniques and discovered that some, like the Ant and Bat algorithms, did not considerably increase accuracy. Other techniques, like the Firefly and Cuckoo algorithms, produced encouraging outcomes. Firefly algorithm, in particular, showed a notable increase in accuracy to 83.97%, indicating its potential in enhancing the recommender system's performance. Overall, the study highlighted the importance of feature selection in improving the accuracy and efficiency of university program recommender systems, with implications for enhancing the decision-making process for students in selecting their majors.

Authors in [22], aimed to develop a model for MBTI personality classification using machine learning techniques. The dataset used in the research was obtained from the Personality Cafe forum, consisting of 8675 rows with MBTI types and individual posts.

The data preparation involved dividing the MBTI type data into four different classes (Introvert-Extrovert, Intuition-Sensing, Thinking-Feeling, Judgment-Perception) and cleaning the data by converting letters to lowercase, removing links, punctuation, and stopwords. Tokenization was performed using the NLTK library, and Word2Vec was used for word embedding. The data were split into training and testing sets, with the training set used to train the classification model.

Several machine learning models were used for classification, including Logistic Regression, Linear Support Vector Classification, Stochastic Gradient Descent, Random Forest, Extreme Gradient Boosting, and CatBoost. The results showed that the Logistic Regression model performed the best, with an average F1 score of 0.8337 after using SMOTE to handle the imbalanced dataset. You can see the process in Figure 2.7.

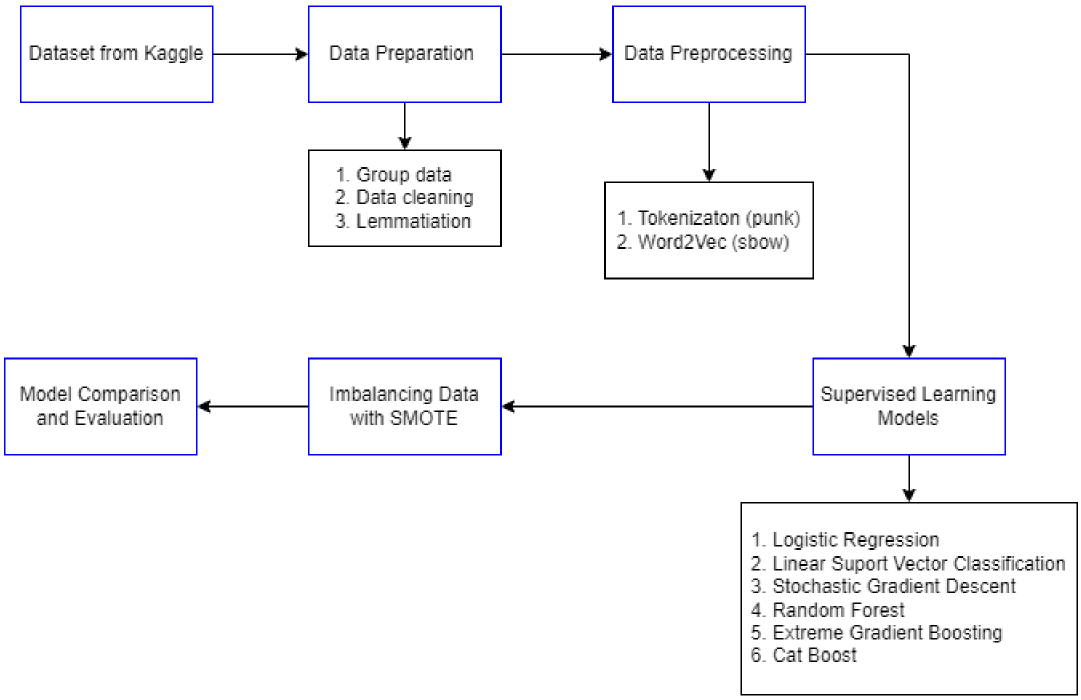

Figure 2.7 – Flowchart of Classification Process

The proposed system [23], is a comprehensive framework designed to provide personalized job recommendations to users. It comprises four key modules: Data acquisition, Transformation, Computation, and Recommendation. In the Data acquisition module, user data including profiles, skills, and browsing history are collected and stored. The Transformation module processes this data, extracting relevant information and transforming it into a format suitable for analysis. The Computation module performs the actual calculations, including data filtering and result set generation, to identify suitable job matches. Finally, the Recommendation module utilizes the filtered results to generate personalized job recommendations for users.

One of the system's strengths lies in its use of multiple recommendation approaches, including matching-based, content-based, and collaborative filtering. While matching-based recommendations are fast, they may lack personalization. In contrast, collaborative and content-based recommendations offer personalized suggestions, but with potential drawbacks such as the cold start problem for collaborative filtering and over-specialization for content-based recommendations. By combining these approaches, the system aims to provide accurate and relevant job recommendations tailored to each user's preferences and needs.

The research [24] proposes an innovative automated system for recommending the best educational program to individual students based on their class 10th educational performance. It involves splitting the dataset into six sub-datasets, one for each educational program, and then further splitting them into training and test datasets. Features are extracted, and resampling techniques are applied to address imbalanced data. Various machine learning algorithms are used for prediction, and the best model is selected based on performance evaluation metrics. The system achieved high performance, with F-measure, Cohen's Kappa, and ROC-AUC averaging around 97% for all educational programs. The best model varied for each program, with LightGBM performing best for the science-based program, CatBoost for commerce-based, and Random Forest for ITI-based, among others.

The experiment results indicate that the proposed system effectively predicts the best educational program for students. Each program's best model achieved high performance, with some programs reaching 100% in F-measure, Cohen's Kappa, and ROC-AUC. The system's ability to identify relevant features for each program further strengthens its effectiveness. For example, in the science-based program and Life Science were crucial, while in the arts/humanities-based program played a significant role.

The authors in paper [25] proposes a hybrid convolutional neural network model for recommending employment to college students. The model utilizes transformed features as input, which are then processed through a multichannel hybrid convolution submodel and a convolution and local connection hybrid submodel. The model structure incorporates techniques such as data combination, splicing operations, and the cross-entropy loss function for training. Several optimizations are discussed, including activation function optimization (using ELU), pooling strategy optimization (using a hybrid pooling strategy), and loss function improvement (using penalty weights and a threshold for difficult-to-differentiate samples). Experimental results show improvements in recall rate and F1-Score compared to baseline models, demonstrating the effectiveness of the proposed approach.

The suggested approach is also compared to alternative recommendation algorithms in the study, such as a four-layer deep neural network, content-based recommendation, probability-based matrix factorization, and user-based collaborative filtering. The findings show that the hybrid convolutional neural network model performs better in terms of F1-Score and recall rate. In comparison to previous algorithms, the model also demonstrates promise for lowering forecast time and space consumption, which makes it a viable strategy for raising the caliber of human resources recommendations.

The investigation [26] seeks to enhance course recommendation by removing misleading recommendations. Based on the most prevalent characteristics that the products have in common, it recommends a route using the Cosine Similarity and Stemming Method. Cosine Similarity treats data objects as vectors in a multidimensional space, and is used to measure how similar they are to one another. Word inflections are reduced to their base tokens by the Stemming Algorithm, which disassembles words into their most fundamental components.

In order to translate textual data into a numerical format that NLP models can understand, the system uses Count Vectorization. The study's dataset, which comes from Kaggle, includes a range of courses along with information about universities, ratings, and difficulty levels. Students can choose courses based on their preferences with the aid of the dataset parameters. The dataset is dumped onto Google Colab, the data is visualized, preprocessed and cleaned, stemming is used, and lastly, Cosine Similarity is used to select courses. Visual Studio Code is used to construct an interface that asks users to select their interest domain and returns course recommendations. The system's overall goal is to deliver users accurate and pertinent course recommendations.

The study [27] describes a thorough method for applying machine learning and fuzzy logic to create a recommendation system for educational programs. Using the CFS attribute selection method, the study identified crucial features from a dataset comprising 1000 cases and 21 characteristics. This technique boosts the system's capacity to predict apt academic programs for students by identifying fifteen strong indicators that align with class and not other variables. Thus, various machine learning algorithms like Random Forest, Fuzzy SVM and C4.5 are used to analyze the collected data to create a recommendation model. Authors leveraged the individual strengths of each algorithm, such as fast Random Forest predictions, accurate classification using non-linear projections to fuzzy SVMs, and iterative categorization to achieve maximum C4 accuracy.

The study highlights that evaluating the effectiveness of recommender systems mainly depends on several criteria such as specificity, sensitivity and accuracy. So the fuzzy SVM method showed a higher level of accuracy and performance compared to C4.5 and Random Forest. This means that machine learning algorithms and fuzzy logic need to be used to help students effectively find suitable academic programs.

The article [28] presents a comprehensive approach for designing a job recommendation system that caters to computer engineering students. The method involved careful selection of 18,000 pertinent records from a vast dataset that encompassed employment positions, extracurricular activities, hobbies as well as grades. To preprocess the data and eliminate redundant features; it was cleaned up and standardized before encoding categorical information with OneHot technique. Furthermore , ReLu activation functions were incorporated alongside categorical cross-entropy loss functions in conjunction with backpropagation techniques during modeling stage which led to improved weights and biases utilized by feedforward computation methodologies resulting in efficient training highlighted by achieving an effectiveness rate of 99% while testing accuracy exceeded at least 94.%.

The artificial neural network model gave higher accuracy in comparison to traditional machine learning techniques like SVM, XGBoost and Decision Trees. Both recruiters and job seekers in the computer engineering industry can benefit from this approach's precise prediction of desired professions. With a remarkable 94.9% success rate, it surpasses previous research-based measures; thereby representing significant promise as an essential tool for IT recruitment strategies.

In the paper [29], machine learning techniques like Random Forest (RF) and Decision Tree (DT) are explored for predicting student careers. To improve accuracy of predictions, traditional questionnaire methodologies are critiqued in favor of utilizing advanced computer concepts such as machine learning. Pretreatment methods required to prepare data and factors including age, parents' level of education, health and family income used in prediction process are discussed extensively within the research study. The main objective is to aid hiring managers with selection by using these traits to provide insight into students' career choices.

The comprehensive explanation of how the DT and RF algorithms are implemented emphasizes their importance for prediction and decision analysis. By utilizing decision trees, data can be sorted based on entropy and information gain. To increase prediction accuracy, Random Forest creates many decision trees, as seen in Figure 2.8. The study's findings show that RF is more accurate than DT at projecting student routes, and they also illustrate the advantages of using machine learning approaches in hiring and training.

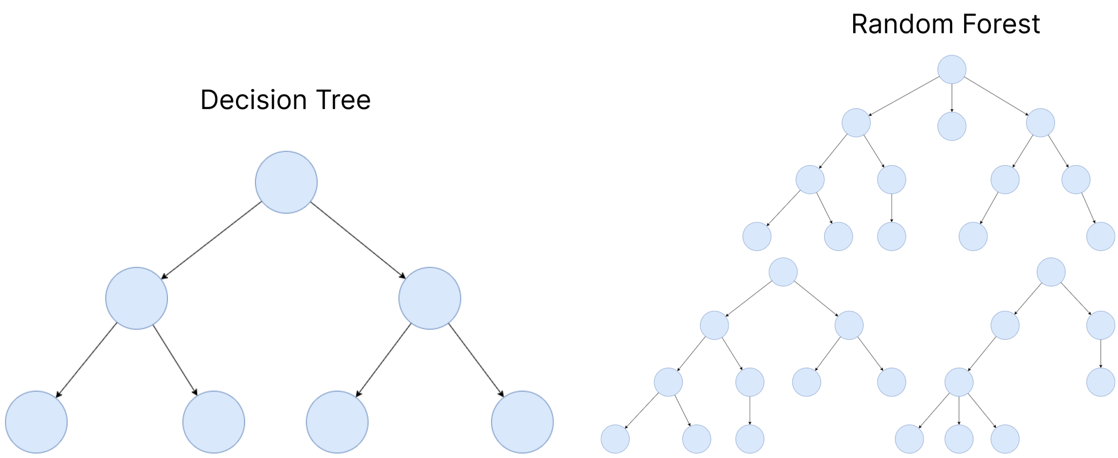

Figure 2.8 – Decision Tree vs Random Forest [9]

The paper [30] presents a research approach that tackles the difficulties associated with job recommendation systems. It specifically addresses the cold start issue and emphasizes the necessity of effective methods to raise suggestion performance. Using The proposed approach utilizes Python programming and NLTK for text preprocessing, as well as scikit-learn to produce machine learning models (e.g. SVM, KNN, Naive Bayes), aiming at expediting the job hunting process for both recruiters and applicants. To enhance suggestion precision levels, the system employs content-based filtering strategies like hybrid methods and collaborative approaches. Confusion matrix measurements are used in the model evaluation approach to determine the optimal algorithms for job and applicant recommendations.

The methodology of the study is broken down into nine steps: data collection, literature review, problem identification, and preprocessing. Numerous techniques are used to preprocess the data, including as vectorization, tokenization, case folding, stemming, stopword removal, and punctuation removal. Subsequently, the collaborative, content-based, and hybrid filtering techniques are put into practice, and machine learning algorithms are employed to create classification models. Confusion matrix measures are used to assess the models' performance after they have been trained and tested on the gathered dataset.

Employers and job seekers can easily navigate the [31] application, which is a planned web-based job portal. Employers can post job vacancies and screen applications, while employees can browse and apply for positions using the system, which uses Flask and Python for backend development. The system recommends relevant jobs with a commendable 94.31% accuracy using the Support Vector Machine (SVM) technique. Both companies and job seekers must register on the portal, which allows job seekers to tailor search results according to their preferences. Through validation tests at each level, the system guarantees data validity, improving both data integrity and user experience overall.

First, labor and contractor data are entered into the system architecture. Next, the SVM classifier is used to anticipate and match appropriate jobs. The final job recommendations are presented on maps based on the labor's location by the solution, which also integrates job recommendations and maps job locations. By providing individualized job recommendations, this strategy tackles the issue of unemployment among daily wage workers.

Two experiments are part of the process for creating a scalable and accurate job recommender system [32]. The accuracy of job recommendations is evaluated in the first experiment using an algorithm based on similarity measurements. Using test data collected from interviews with respondents in the IT industry, the algorithm's performance is evaluated. In order to assess the scalability of the recommender system, a substantial dataset comprising job posts and candidates is handled in the second experiment. The results indicate that the computation speed of the recommender system scales practically linearly with the amount of data up to a certain point. Scalability concerns for a centralized system are shown by the system's constraints in terms of memory usage and hard drive capacity as the volume of data increases.

On the other hand, the Hadoop MapReduce-built distributed cluster job recommender system has superior scalability. The 6-node cluster outperforms the 4-node cluster in terms of performance, although both clusters exhibit linear scalability to the volume of data. The system's memory and storage capacity can be expanded based on demand by adding more worker nodes. This feature improves the distributed cluster job recommender's scalability and presents a viable way to manage big datasets and guarantee effective task recommendations.

Using a combination of datasets from Google searches, Kaggle, and personal contacts, the job recommendation system described in the paper [33] creates a comprehensive Job Dataset with 13,001 job descriptions in CSV format. Furthermore, a skill dataset of thirty-two particular occupational skills is assembled. Using stop words and Porter's stemmer, the methodology preprocesses resumes and job descriptions to produce a meaningful bag of words, which are then vectorized using TF-IDF to construct matrices. The algorithm compares resumes and job descriptions using cosine similarity, then suggests positions based on the similarity results. On the basis of the job recommendations, it also makes improvement suggestions for skills.

The main features of the system are that it may be utilized by individuals via a subscription model and by big MNC organizations for hiring recommendations. The system seeks to lower unemployment and assist people in finding acceptable employment by making recommendations about jobs and skills based on user input. The architecture of the system includes processing job descriptions and resumes, determining similarity scores, and making recommendations for the best positions and abilities. According to experimental data, cosine similarity is the favored similarity metric for job recommendations, and pie charts are used by the system to display recommendations.

A comprehensive approach to job suggestion systems is presented in the paper [34], which aims to address the cold-start issue for jobs as well as candidates and to increase the serendipity of job recommendations. To anticipate candidate-job interactions, the authors used deep neural networks such as Artificial Neural Networks (ANN) and Bi-Directional LSTM with Attention, as well as machine learning models such as Random Forests and XGBoost. The Bi-LSTM with attention model outperformed the rest, using previous interactions to forecast future results.

The authors proposed a hybrid technique to address monotonicity and improve the diversity of recommendations. This strategy combines machine learning predictions with non-machine learning techniques, such suggesting jobs that candidates have applied to and jobs that the candidate has applied to. On the job web portal, the mixed strategy resulted in a 63% relative rise in click-through rates (CTR), suggesting more user engagement and maybe more appropriate job recommendations.

Authors presents a comprehensive approach to job recommendation systems [35], particularly focusing on enhancing machine learning models' results with real-life scenario inspirations. By training a Bi-LSTM model with attention, the system predicts job interactions more accurately, considering candidate preferences and job relevance. However, it acknowledges the monotony of solely ML-based recommendations, where similar jobs are overly suggested. To tackle this, the system blends ML predictions with additional recommendations: jobs applied to by similar candidates and similar jobs applied to by the candidate. This blending strategy aims to make recommendations more diverse and serendipitous, mimicking real-life scenarios where candidates seek advice from peers or express specific interests in jobs.

The implementation also addresses the cold-start problem for both jobs and candidates, ensuring new entities are not overlooked. For new candidates, similar candidates' job applications are suggested, while new jobs are compared with similar existing jobs applied to by candidates. This strategy, combined with the blended recommendation approach, significantly increased the click-through rate (CTR) by 63%, indicating a more engaging and effective job recommendation system.

The inquiry [36] explores various recommendation systems, focusing on key algorithms and equations like MAE, RMSE, KNN, SVD, and ARHR. The implementation section delves into essential algorithms such as Top-N recommender, KNN recommender, SVD Matrix recommender, and SVD++ Hybrid recommender, discussing their significance and application. Top-N recommender systems, like those seen on Amazon, provide users with ranked lists of items, while KNN recommender systems use machine learning to cluster similar users based on ratings. SVD Matrix recommender applies matrix factorization to reduce dataset dimensions, crucial in collaborative filtering for recommendation systems. By merging the previously stated methodologies, the SVD++ Hybrid recommender improves choices to give customers more precise and tailored options.

The approach described in the study improves recommendation systems by combining machine learning techniques and mathematical equations. Through the integration of many recommendation systems and algorithms, the system endeavors to enhance its precision and surmount its constraints by recommending pertinent information to consumers. The paper's focus on using a variety of strategies to improve suggestion accuracy and user experience is highlighted by the discussion of hybrid recommender systems, which combine several recommendation methodologies.

In this research [37], a hybrid convolutional neural network (HCN) model is constructed by combining machine learning and deep learning approaches, offering a new way to recommend careers to college students. By utilizing convolutional processing's high-level feature learning capability, the model seeks to improve the caliber of recommendations. A convolution and local connection hybrid submodel as well as a multichannel hybrid convolution submodel make up the HCN model. The proposed solutions are obtained by feeding the updated features into the HCN model using model stacking integration.

The primary focus of the work is to optimize the model's activation function, loss function, and pooling technique, among other aspects. The activation function is optimized using Exponential Linear Units (ELU) to alleviate the issue of neuron death. Pooling strategies are compared, with the hybrid pooling strategy showing superior performance. Additionally, the loss function is improved to reduce the dominance of easy-to-classify samples and pay more attention to difficult-to-differentiate samples, enhancing the model's ability to classify sample categories. Experimental results demonstrate that the proposed algorithm outperforms traditional recommendation algorithms and ordinary convolutional neural networks, highlighting its effectiveness in improving the quality of human resources recommendation for college students.

The proposed a mechanism for building an accurate execution plan recommendation system [38] for database queries introduces a textual representation model and a similarity evaluation module to identify similarities between queries based on their SQL statements and execution plans. The Features Extractor module captures features from query statements, while the Similarity Evaluation module predicts similarity using machine learning techniques. The paper compares three machine learning algorithms and discusses their accuracy in this context. It also presents the Fragments Model, which splits SQL queries into fragments to reduce the feature space dimensions and improve entity recognition in queries.

In terms of experimentation and results, the paper evaluates the performance of the proposed approach on two datasets: one with simple queries and another with queries containing subqueries. It uses k-fold cross-validation for model validation and three metrics for evaluation: prediction rate, RMSE, and Kappa statistic. The results show that the proposed solution using Classification based on Association Rules (CAR) achieves a high prediction rate of 91%, indicating its effectiveness in query recommendation. The paper concludes that while the straightforward solution based on textual models showed some limitations, the improved solution enabled by machine learning techniques significantly improved recommendation accuracy, particularly with the use of CAR.

The investigation [39] utilized a two-phase methodology to enhance candidate-to-job matching. Initially, users annotated data by indicating the matching quality of candidate-vacancy pairs using green thumbs up and red thumbs down widgets. Despite providing guidelines, some users did not adhere to them, necessitating additional instructional sessions. The second phase involved preprocessing the annotated data, filtering out queries with incomplete information needs, and augmenting the dataset with implicit signals from unassessed documents. When LambdaMART and linear regression, two learning-to-rank (LTR) algorithms, were used on the dataset, LambdaMART performed noticeably better than the linear model. The algorithm's ability to adapt to user preferences, including location preference in certain situations, was shown by the reranked results.

The methodology of the study emphasizes the difficulties associated with user annotation and data preprocessing in LTR activities. Although users were educated, heuristic procedures had to be implemented to enhance the dataset because of incomplete annotations. The results demonstrate that LambdaMART outperforms linear regression and applying advanced algorithms to LTR jobs can bring advantages. Nevertheless, certain reranked outcomes showed peculiar behavior which highlighted the algorithm's weaknesses in learning patterns. This implies further investigation is needed into the decision-making process of the algorithm and unexpected data trends may impair its effectiveness.

OASiS, a new approach for scheduling jobs and allocating resources in distributed machine learning (ML) systems by Examinee [40], considers resource costs while managing asynchronous training tasks. It adapts the number of workers and parameter servers dynamically to meet changing requirements as well as incoming tasks. The algorithm's polynomial running time and competitive ratio are two examples of how useful it could be in practical situations, as demonstrated by the theoretical analysis.

Using testbed trials and extensive modeling simulations, OASiS is contrasted with a number of current scheduling techniques, such as RRH, DRF, FIFO, and Dorm. The results continuously show the superiority of OASiS in terms of overall work value and adherence to target completion timeframes, particularly in circumstances with limited resources and different project priorities. All things considered, the results demonstrate that OASiS represents a noteworthy breakthrough in distributed machine learning system management, providing more effective resource utilization and job scheduling for enhanced system performance.

The writers in [41] use the Cosine Similarity and Stemming Method when recommending courses in order to remove suggestions for misleading courses. The system accurately determines how similar courses are to one another by finding the cosine of the angle between vectors in multidimensional space and evaluating the most common attributes that products possess. The primary objective of the Stemming Algorithm is simplifying words by highlighting vital morphological characteristics. This facilitates a more enhanced search experience and promotes careers in Natural Language Processing (NLP). Count Vectorization, which converts textual data into numerical values, greatly enhances system efficiency that enables NLP models to handle it with ease.

Using data mining techniques such as counting vectorization and stemming, the course recommendation process was improved through the implementation of the system. The dataset included university names, difficulty ratings and descriptions to help students choose courses that suit their preferences. The recommendations' accuracy has increased due to user feedback. Also, students are able to make well-informed decisions regarding their overall academic goals. We observe improved accuracy and efficiency as a result. In the end, this produces tailored advice that is advantageous to all parties.

The [42] describes a mechanism that uses machine learning to suggest educational resources. The authors paid special attention to initial data processing methods, such as handling zeros and normalization. Logistic regression was chosen as the optimal algorithm for real-world applications. After evaluation, it performed well on the criteria of speed, accuracy and ease of use. The article discusses improving the prediction and training phases throughout model development. To avoid overfitting, the data is divided into test and training sets to simplify the construction processes and thereby help in decision making. The critical components that are involved in the development of machine learning models are also discussed, along with detailed recommendations aimed at strengthening reliable forecasting systems. Ultimately, this paper's approach can help improve academic achievement by integrating practical applications of ML into a scientific context. Because it simultaneously promotes practical recommendations throughout the discussion.

The researchers conducted a study [43] that aimed to predict elective courses for graduate students in four specific areas. For the data set, the researchers collected records from 250 people. In order to carefully analyze the results and classify each course, the authors used various data variables. To train the system effectively, the authors wanted to compare different models. The models used were Decision Trees, K-Nearest Neighbors (KNN), Logistic Regression, and Linear Support Vector Machine with Naive Bayes. Based on their results, it was found that both the decision tree model and the Naive Bayes model accurately predicted alternative routes with high F1 scores in this particular scenario. Therefore, researchers strongly recommend using decision trees and Naive Bayes in the future.In addition to being effective alternatives for precise categorization of students' predicted electives compared to other machine learning algorithms tested here such as Linear support vector machines or logistic regression which have averaged performance when employed alone without any ensemble techniques. The study demonstrates how supervised machine learning is a useful tool for academic advising because it can accurately predict student performance and course choice. The importance of selecting the appropriate algorithm for a specific task to provide accurate and dependable predictions is demonstrated by the comparison of classification algorithms and the selection of the optimal model based on performance indicators.

An agent-based recommendation system for e-learning environments is presented in the paper [44], with a particular emphasis on customized document recommendations. The three stages of its methodology are information gathering, proposal preparation, and pre-processing.The system employs text mining algorithms to gather information from learner and user profiles, enabling the provision of suitable learning resources. It also helps pinpoint relevant courses that can improve skills based on previous search activities and individual student preferences. The precision of recommendations is maximized through GridSearch integration featuring a suggestion function within combined modules with custom-tailored parameters unique to each student's needs for refined accuracy during interactive sessions in the framework produced by said modules.

Due to the fact that the authors used advanced clustering technologies such as semantic analysis and topic modeling, the system was able to surpass traditional methods in accuracy and user satisfaction. Thus, the recommendation system provides more accurate sets of educational content options to e-learners. This will help them expand their learning journey. Moreover, by adding natural language processing to machine learning on the online learning platform, it helps to realize further customization. Namely, based on individual learning requirements. The results showed that the integration of advanced technological innovations is helping to change preferences among users who are looking for an enhanced e-learning experience.

The article [45] presents an automated system for recommending career paths based on machine learning. The system uses three techniques, namely Logistic Regression, Support Vector Machine, and Decision Tree Classifier to obtain accurate assessments of individuals' skills. To prepare student data for processing and skill prediction with respect to relevant professional certifications, Google Forms are integrated into the program. Various methodologies were employed in assessing the accuracy of this tool. Notably coupling Linear Discriminant Analysis with Logistic Regression resulted in a superior performance outcome.

Prior to and after incorporating semester subject grades, the precision of different algorithms is demonstrated by the results. For example, an entropy-based decision tree classifier achieved an accuracy of 74.0% after adding semester marks, compared to 80.0% before. In a similar vein, the accuracy of logistic regression was 82.0% after correction and 94.0% prior. These outcomes show how well the algorithm predicts student abilities and suggests career qualifications.

Similarly, the study [46] focuses on students without programming experience and offers a novel hybrid recommendation method for career recommendations.

The system's goal is to recommend useful and inspiring resources according to various circumstances, tastes, and expertise. The system groups learners and mines sequential patterns by merging many learning models and behaviors with clustering algorithms. The suggested model improves the recommendation accuracy by evaluating using a modified TSVM. The efficacy of the suggested method is demonstrated by experimental findings on publicly available datasets, where the accuracy ranges from 82% to 98% and the Mean Absolute Error (MAE) varies from 5% to 19.2%. As the number of learners increases, the model outperforms current approaches and achieves improved accuracy.

Additionally, the paper presents novel approaches to recommendation list generation, clustering, and data preparation. While the enhanced DBSCAN clustering technique is used to group learners that are similar, the MOSD approach is used to reduce noise in learner data. Learner actions like as clicks, selections, and activities are taken into consideration by the recommendation process in order to tailor recommendations. The experimental study shows that, in comparison with existing methods, the suggested model improves computing time, accuracy, precision, recall, and ranking score. In terms of improving student learning outcomes, the study shows promise and provides a solid basis for customized recommendations in online learning environments.

To circumvent the limitations of traditional collaborative filtering recommendation systems, the Fusing Knowledge Graph and Collaborative Filtering (FKGCF) algorithm in [47] includes semantic similarity into the similarity calculation procedure. Unlike existing methods that only use users' historical rating data, FKGCF considers the semantic linkages between recommended items, improving the accuracy and relevancy of the suggestions. Utilizing additional data sources to assess item similarity and produce more precise course recommendations, FKGCF constructs a knowledge graph that connects interactive courses with non-interacting ones.

By blending collaborative filtering and knowledge graph representation learning, the approach foretells user-item ratings and crafts recommendation lists. FKGCF calculates a weighed similarity gauge that merges both factors - collaborative filtering item resemblance as well as semantic affinity of items. Such an amalgamation strikes a harmonious balance between two techniques enabling the system to proffer recommendations predicated on both collaborative filtering & item semantics. Comparative experiments performed with conventional algorithms showcase how FKGCF overtakes them considerably in augmenting suggestion precision, recall rate along with elevating overall value of accuracy implied by metric parameters like F1 score, etcetera.

In [48], a novel approach to Course Recommendation Systems (CRS) is introduced, incorporating rule-based classification techniques. The CRS system aids students in choosing courses intelligently by proposing alternative selections that correlate with their academic standings and interests. By scrutinizing former grades and chosen electives, unsuitable picks are circumvented to enable sensible decisions. The recommendations founded on algorithms classify the options into theoretical, conceptual, programming or logical criteria creating better proposals compared to those from experimental design tactics based on haphazard methods.

Empirical statistics demonstrate a rise in overall marks among users who avail of the CRS facilities as opposed to non-users.

The effectiveness of a rule-driven approach used in CRS is demonstrated by the better performance of students who receive instruction through CRS-based subjects compared to those taught with non-CRS methods. Additionally, customized recommendations based on their academic record and interests further support this claim. Therefore, selecting electives via CRS provides an easy solution for individuals to align themselves with appropriate courses that match their educational background and career goals.

In article [49] authors presented an integrated approach to predict academic success based on multiple goals. This process consists of collecting and organizing relevant data to implement an algorithmic structure. The authors used various ML methods like: K-nearest neighbor, support vector machines (SVMs), decision trees, logistic regression, linear discriminant analysis and naive Bayes analysis. Integration of these methods gave an accuracy of 88.5%. When testing SVM separately, the accuracy level reached up to 90.3% when student performance levels were separated. This methodology is different in that it can identify students at risk. Also give a forecast for passing grades after completing your studies. After the forecast, the system also gives immediate support without any downtime or delay. This results in significant improvements over time and maintains a high retention rate of over 93%.

The paper [50] proposes a method for suggesting suitable majors to high school graduates based on their test scores. This approach uses a 3D graph that illustrates potential pathways classified by labels and analysis of JEE or CET data. Machine learning techniques such as collaborative filtering and K-nearest neighbor (KNN) were used to develop this recommendation system. But according to the results, KNN is preferable among all methods since it can classify feature space data using distance functions through proximity metrics. Furthermore, collaborative filtering determines user interests by approximating ratings or product preferences derived from past actions undertaken by other users – an approach commonly used within recommender systems.

The deployment outcomes of content-based collaborative filtering demonstrate its effectiveness in suggesting products to consumers by comparing their preferences. This recommendation engine establishes a user profile and gains insight into their interests through product attributes and ratings, enabling the provision of personalized recommendations.

The focus of the study [51] is on the design and methodology employed in providing course recommendations to learners, resulting in a Unified E-Learning Course Advisory System. The E-Dirassa system places emphasis on both courses and students through its four-pane interface and tailored criteria for every user role. A course comprises various elements such as title, definition, type, status, instructors along with prospective commencement or conclusion dates that aid it define itself accurately. Based on how well they match the learner profiles of current students, learner profiles—which are created using personal information like name, level, and specialty—are used to recommend courses to prospective students. Utilizing machine learning techniques, the system selects articles based on word similarity in content-based recommendations and predictions items based on similar user actions in collaborative filtering.

The efficacy of the system depends on how well the machine learning models predict courses. For content-based recommendation, the study assesses a number of textual similarity metrics, including cosine similarity, Pearson correlation coefficient, Euclidean distance, Jaccard coefficient, and Dice index. Stability, sensitivity to uncommon terms, and sensitivity to popular terms are some of the evaluation's criteria. According on the research, the Dice index provides reliable and accurate recommendations and is the most appropriate metric for evaluating course descriptions. The study also covers static profile-based collaborative filtering for new learners and course-based collaborative filtering for active learners, offering a thorough method for individualized course recommendations in E-Dirassa.

In order to enhance students' learning experiences, the study [52] investigates several adaptive methodologies and problem transformation approaches used to machine learning algorithms for multi-label learning. One-vs-All, Binary Relevance, Label Powerset, Classifier Chains, and an adaptive Multi-Label-KNN classifier are among the techniques. Several classification algorithms, including Support Vector Classifier, Logistic Regression, Random Forest, Gaussian Naive Bayes, and Decision Tree, are used to assess these methods. In terms of F1-measure and Hamming loss, the experimental results show that the Label Powerset with Support Vector Classifier and adaptive Multi-Label-KNN algorithms perform best, demonstrating their efficacy in forecasting suggested actions for course improvements. Moreover, the feature importance analysis emphasizes the key components influencing each recommended action, providing valuable data for strategies targeted at course improvement.

ULEARN employs machine learning techniques to enhance recommendation accuracy, by utilizing the learner model, course content model and ratings of learning objects as elaborated in [53]. Personalization approaches such as those mentioned above are employed to predict and evaluate learners' preferences. The customized recommendations offered cater specifically to each student's unique needs and desires. Through its tailored approach, ULEARN guarantees that all individuals receive instructional resources specially crafted for them.

The core of ULEARN's machine learning method lies in its adaptive engine, which includes a cooperative filtering system. Its objective is to provide an exceptional educational journey tailored for each student; our state-of-the-art technology utilizes diverse teaching techniques and resources seamlessly. By analyzing individual adaptability profiles, this potent tool proficiently identifies the best ways to boost academic achievements.

A survey [54] conducted to evaluate the effectiveness of ML methods. According to the results, the use of machine learning-based decision-making strategies resulted in an accuracy rate that exceeded 90%, as well as remarkable outcomes in resource allocation in various scenarios. Additionally, several models displayed precision levels reaching up to 95%, even in dynamic contexts. Consequently, this study showcases how accurately machine learning algorithms manage energy at the edge. Some methods attain an accuracy rate of 80% or more, effectively predicting and optimizing energy usage. A fundamental discovery from the survey highlights how dataset quantity and quality impacts machine learning model precision. The importance of appropriate data compilation and preprocessing in maximizing model performance is highlighted by research indicating a rise in precision rates when using diverse, extensive datasets. Additionally, examining trade-offs between complexity and accuracy reveals some simpler models can achieve similar levels of precision as their intricate counterparts.

Paper [55] examines the Random Forest Classifier versus the Decision Tree Classifier in job recommendation systems. The Decision Trees approach is a widely respected and versatile method in supervised learning. It is capable of efficiently handling both categorical and continuous output variables, which makes it highly effective. Additionally, the attribute selection process is straightforward and user-friendly due to simple data set partitioning processes and learning algorithms. During testing with the sklearn module, pre-processing data has become easier as test sets can be readily distinguished from training sets resulting in an excellent accuracy rate of 84.61% made possible by decision tree algorithm techniques. On a different note, Random Forest classifiers have demonstrated exceptional performance through their remarkable ability to handle massive amounts of information without requiring any initial fine-tuning while simultaneously avoiding bootstrap method overfitting issues. Implementing feature subset strategies further increased overall efficiency in multiple decision trees leading up to an astounding precision percentile reaching up to 90%.

The college major recommendation system, developed by researchers at [56], was created using data collection, preprocessing and visualization techniques coupled with machine learning methods. The relevant job market trend datasets were obtained from Kaggle and MBA research. A combination of advanced hyperparameter optimization along with popular algorithms such as Random Forest (RF), Support Vector Machines (SVM) and Decision Trees (DT) ensured a high accuracy rate of 95%, which was achieved using the RFC algorithm. In order to improve the prediction results, as in [55], the researchers analyzed the relevance of the function. To address larger areas, the authors included percentages of coursework entrance exam scores. To fine-tune hyperparameters along with complex training datasets that significantly improve successful results. Ultimately, the results highlight the need for careful algorithm design coupled with responsible governance. At each stage, important insights provide critical guidance for creating effective educational solutions in large-scale settings. "Technical Acumen" is of paramount importance and is demonstrated through proper practical application. Only this approach will maximize the intelligent use case without any grammatical or spelling errors.

In the study [57] the authors developed a job recommendation system that uses machine learning. The required data was extracted from employment websites. The team used techniques such as scraping, feature selection, heatmap visualization, and histograms to gather relevant information for accurate recommendations. To identify important features, they used classification tools such as Stochastic Gradient Descent or K Nearest Neighbor classifier. All this contributes to the generation of suitable sentences in the data set. After the research results, it was found that the random forest classifier has an accuracy of 94.54%. This means that random forest is a reliable method for job recommendation systems that use machine learning approaches. Based on these results, along with additional scientific journals related to the development of a job recommendation algorithm in artificial intelligence and machine learning discovered during online browsing sessions, the researchers propose combining F1 scores with random forest, leading to higher accuracy and recall performance compared to traditional approaches. The study's findings imply that anyone looking for career guidance might find this methodology useful. These algorithms may be included by recruiting firms into recommendation systems. Customers would receive more tailored recommendations regarding potential roles as a result.

A mixed technique is used by the graduate program recommendation system in [58]. It combines the multi-class Support Vector Machine (SVM) with the K-Nearest Neighbors (KNN) algorithm. Using data from the web, this system makes recommendations for institutions based on factors including price, location, acceptance rate, and rating. The KNN algorithm uses predefined parameters to identify appropriate colleges. The SVM gains the ability to determine which postgraduate program is best for every user. With an accuracy rate of 61.6%, empirical results demonstrate that this system performs better than multilayer perceptrons, C4.5 algorithms, and Naive Bayes.More precisely, the KNN algorithm, when limited to four neighbors (K), performs better than other methods in terms of yielding results. It does this by recommending colleges with comparable core curricula in an effort to increase student choice. In comparison to current alternative methods, integrating SVM+KNN yields suggestions that are noticeably better overall".

The system's prediction accuracy at the core universities was found to be 58% during user research. Most concepts were rated as "Relevant" or "Likely Relevant" by appraisers, indicating that they were effective. It is clear from evaluating nDCG ratings for overall ranking performance that this system more accurately and prominently displays the top choices, underscoring its ability to offer well-tailored recommendations.

To increase students' options, the KNN algorithm finds similar universities by using the core curriculum. Four neighbors (K) are ideal since they yield better results. In performance comparison tests, the SVM+KNN technique outperforms other recommendation systems statistically. The predictive accuracy rate for suggested core colleges is 58%, according to a user research, and appraisers find that most recommendations are "Relevant" or "Likely Relevant," further confirming the effectiveness of the method. Furthermore, recommendations that were judged beneficial also show up at the top half based on nDCG analysis measuring overall ranking effectiveness, which highlights their precision and relevance criteria. This illustrates the system's potential to provide precise predictions that accurately meet customized requirements.

Authors [59] used a two-part recommendation technique in another analysis. Initially, they computed a collection of potential items for every user, greatly diminishing the processing burden. Second, they developed a model to forecast the likelihood that a user will interact with a specific object. In order to learn these probabilities, they employed Gradient Boosting Decision Trees (GBDT), optimizing for log loss. They trained their model and assessed it against the data from the previous week ingeniously, even in the lack of a well-defined training set, which allowed them to properly track their progress. With scores on the public and secret leaderboards of 675985.03 and 2035964.16, respectively, the team's solution achieved an impressive second place in the competition.

The method's successful candidate selection procedure and efficient interaction probability learning are its main selling points. By choosing strong candidates based on several similarity measures, they were able to tackle the problem of a large number of user-item combinations with a significant reduction in calculation time. The model's capacity to correctly anticipate user-item interactions was demonstrated by their successful usage of XGBoost for learning interaction probabilities. They faced obstacles like as limited submission attempts and data anonymization, but their careful feature engineering and model mixing techniques enabled them to emerge victorious in the competition.

The article [60] describes experiments on cooperative recommendation systems with GNNs, namely GraphSAGE and Temporal Graph Networks (TGN), in combination with baselines such as LightGCN and Gradient Boosted Classifier (GBC). These techniques were used on a MEDLINE dataset that highlighted author collaborations. The results demonstrated that TGN is effective at learning complex dependencies and temporal interactions, especially when using publication titles as node characteristics. It also outperformed other approaches in terms of AUC and AP. The models' practical value was further demonstrated by external evaluations incorporating user ratings.TGN showed that it could make recommendations that were adequate even with little adjustments.

Additionally, the study identified areas for improvement, such as investigating more effective hyperparameter tuning techniques, investigating distribution-based representations for authors' features to better manage uncertainty, and examining the effects of penalizing publications with lengthy author lists on recommendation performance. The customisation of the models could be improved through the collection of user text comments. However, given the possibility of declining response rates, this could present difficulties. The study's overall findings for GNN-based cooperative recommendation systems are encouraging, although more work and research is still needed.

Important steps in the requirements formulation and application evaluation processes are automated by a proposed technique [61]. In-depth feedback is also given to individuals who are denied. This strategy makes the work of the human resources department easier and ensures that the correct individuals are chosen by providing a systematic approach to candidate selection. The system's use of data mining algorithms, such as naive Bayes, for skill matching in the hiring process demonstrates its commitment to accuracy and efficacy.

The interactions and results show how the system works, with features including job recommendations based on CV analysis, applicant status checks, and options for rejected candidates to submit feedback. The implementation details provide an organized procedure for arranging placements, registering candidates and companies, and viewing results.

The suggested system [62] introduces a customized recommender system for companies and job seekers, thereby addressing the shortcomings of conventional recruiting techniques. The method seeks to match people with appropriate job opportunities based on their profiles and preferences by utilizing the abundance of online information. The system links recruiters and job seekers through web services integration, enabling effective recommendations. Better candidate-job matching is made possible by the application of Professional Social Recommender (PSR) approaches, which improve the recruiting process. The system enhances suggestion accuracy and relevance by means of text field filtering and a hybrid method that blends collaborative, content-based, and knowledge-based filtering.

The system architecture centralizes all recruiting-related operations and data by incorporating subsystems for database administration and workflow management. The system offers a user-friendly interface for recruiters and job searchers to manage their profiles and suggestions, including features like job recommendation with PSR and see recommendations. All things considered, the system tackles the problems associated with information overload in online recruiting and provides a complete solution for effective candidate-job matching.

The paper [63] proposes a hybrid recommendation system for suggesting lifelong learning courses to LinkedIn users based on their profiles. The system combines ontology-based semantic filtering with machine learning (ML) clustering algorithms like DBSCAN and K-Means. The ontology represents job sectors, skills, and knowledge areas hierarchically, enabling the system to determine related job sectors and skills for each user. The ML algorithms are used to cluster job sectors and skills, enriching the ontology and improving recommendation quality. The system's performance was evaluated using various configurations, showing promising results compared to a baseline hybrid recommender system. The proposed system offers a flexible architecture for improving recommendations in a competitive work environment, focusing on updating and developing users' skills.

The results indicate that the DBSCAN-based configuration and the semantic rule-based configuration performed best, showing improvements in recall and serendipity measures compared to a baseline system. The system's ability to predict related job sectors and skills, along with course recommendations, demonstrates its potential in providing personalized lifelong learning suggestions for LinkedIn users. The study compares favorably with similar works, highlighting its unique approach in determining related positions and predicting skills for improvement or development.

Together, these studies demonstrate diverse approaches and methodologies, leveraging machine learning and innovative techniques to offer tailored recommendations and guidance across various domains.

As you can see, there are many studies in the area of recommendation systems based on machine learning. According to a study [9], the authors provided a full overview of recommendation systems using machine learning and reinforcement learning. The study highlights the existence of reinforcement-based recommendation systems in different domains, but from a career perspective, they are not too much. In this study, information regarding the benefits and disadvantages of each algorithm was obtained. This paper provides explanations and introduces reinforcement learning algorithms and their types. So, in our study we want to make a comprehensive analysis of machine learning and reinforcement learning algorithms

2.2 Reinforcement Learning

Reinforcement Learning (RL) is an evolving area that aims to maximize an agent's rewards by guiding its actions in a changing environment. This process involves observing the environment, deciding on actions, receiving rewards or penalties, and using these experiences to improve decision-making strategies see Figure 2.9. The four main components of reinforcement learning are the agent, environment, reward, and action. The two main categories of algorithms in this discipline are model-free and model-based. While model-based algorithms rely on this knowledge, model-free algorithms use the dynamics of the environment to determine the best course of action. By providing a strong basis for determining optimal actions in uncertain and intricate situations, reinforcement learning can aid in developing algorithms [64].

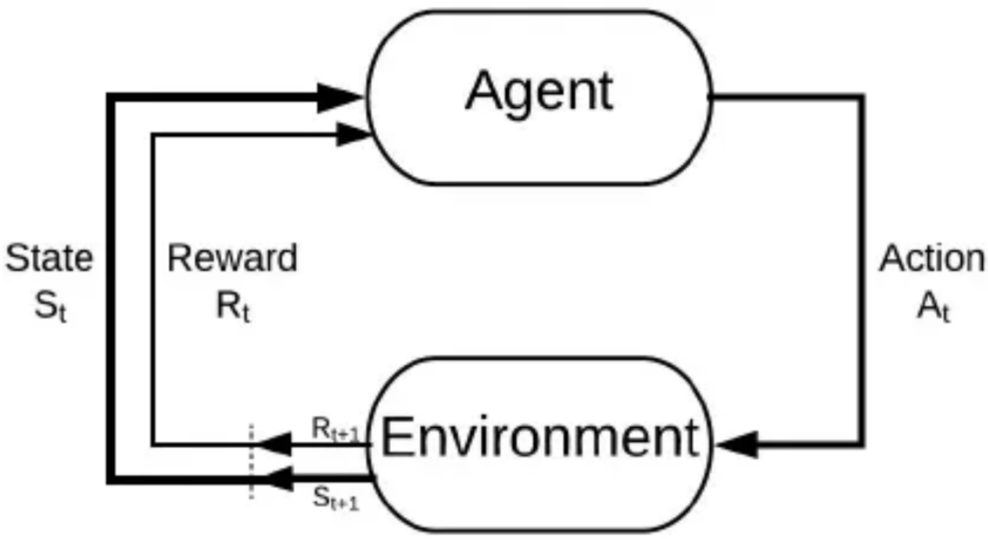

Figure 2.9 – Reinforcement Learning [64]

Systems that recommend careers largely rely on reinforcement learning approaches, such Q-learning and Deep Q Network. These techniques enable the system to dynamically recommend the best course of action in response to user feedback, resulting in highly relevant and customized career counseling [65] [66].

By taking into account individual preferences and prior experiences, Q-learning, a method to reinforcement learning is a powerful method for enhancing career suggestions by enabling adaptive personalization of offered employment opportunities based on changing conditions and user input [67]. It does not rely on models, can maximize career suggestions [64]. The recommendation process is guided by the program's incentive system, which also allows it to learn from previous interactions. Q-learning enables adaptive personalization of offered employment possibilities in response to changing conditions or user input as the user's aims and interests vary over time [68]. By incorporating the program's incentive system, the recommendation process becomes more guided and responsive, enhancing its ability to provide tailored suggestions that align with the user's evolving aims and interests [67]. Figure 2.10 illustrates the Q-Learning algorithmic approach.

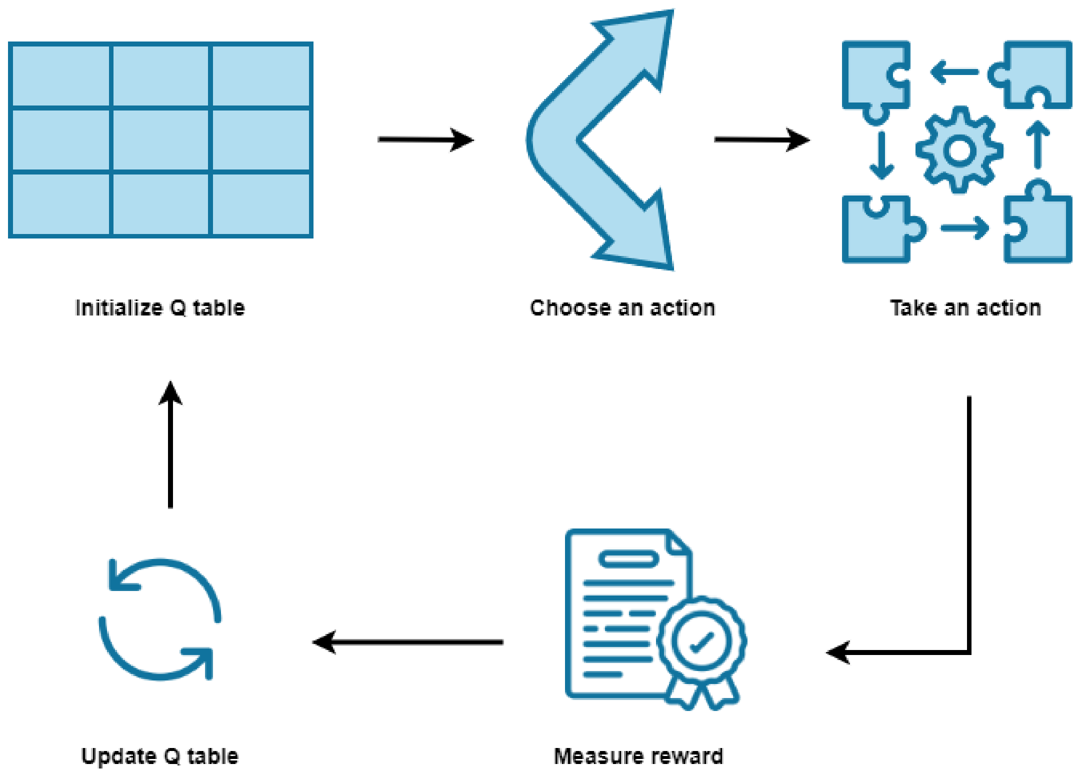

Figure 2.10 – Q-learning algorithm

- **Initialize Q Table:** At the outset, the Q-table, representing the state-action space, is initialized with random values. This table serves as the repository of the learned knowledge, storing the expected rewards for each possible action in each possible state.
- **Choose an Action (randomly possible job role):** The agent, situated within the environment, selects an action based on its current policy. In the context of this study, the action corresponds to recommending a job role to a candidate. Initially, this action selection is random, reflecting the agent's exploration of the solution space.
- **Take an Action:** The selected action, representing a job recommendation, is executed within the environment, which in this case is a dataset containing information about job roles and candidate profiles. This step simulates the real-world application of the recommendation algorithm.
- **Get Reward:** Following the execution of the recommended job role, the agent receives a reward from the environment. This reward is a numerical value that reflects the success or suitability of the recommended job role for the candidate. It serves as feedback for the agent's decision-making process.
- **Update Q Table:** Based on the observed reward and the Q-learning for-

mula, the Q-value associated with the current state-action pair is updated. This update reflects the agent's updated understanding of the effectiveness of recommending the particular job role to candidates with similar profiles. Over time, through repeated iterations and interactions with the environment, the Q-table converges towards an optimal policy for job role recommendation.

The amalgamation of reinforcement learning and deep learning in Deep Q-Network provides accurate predictions and recommendations. The Deep Q Network is able to deliver customized job recommendations by analyzing a user's preferences, abilities, and interests through the use of reinforcement learning techniques. These suggestions are predicated on the total benefit derived from doing particular things (such following a particular career route) in a particular setting (the labor market). Because of this approach's capacity for ongoing learning and modification, the recommendations are guaranteed to remain applicable in the ever-changing labor market. A job suggestion system utilizing the Deep Q Network has the potential to completely transform the way people go about their professional lives. Through the application of reinforcement learning and deep neural networks, the Deep Q Network can offer users precise and customized career recommendations based on their own goals, hobbies, and skill sets [69] [70]. In figure 2.11 you can see algorithmic approach of DQN.

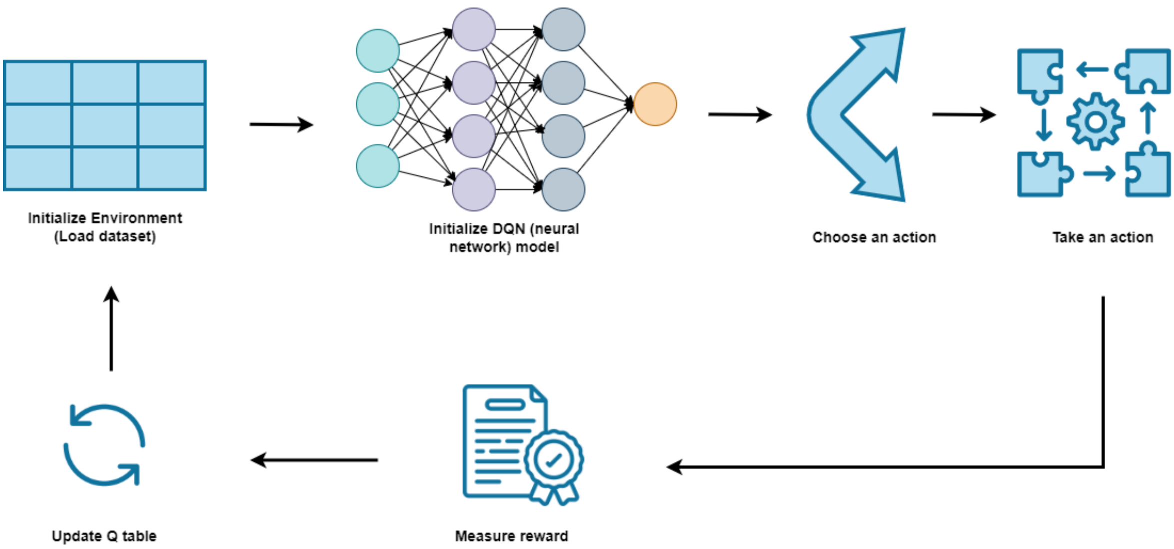

Figure 2.11 – DQN algorithm

- **Initialize the DQN Model:** Following environment creation, a neural network architecture is defined to approximate Q-values. The current state, representing a candidate profile, is observed from the dataset. This state is then input into the DQN model to obtain Q-values for all possible actions, corresponding to job recommendations.
- **Choose an Action (randomly possible job role):** The DQN agent selects an action based on its current policy. Initially, this action selection is

random, enabling exploration of the solution space and gradual refinement of the policy.

- **Take an Action:** Subsequently, the selected action (job recommendation) is executed within the environment, which in this case is the job recommendation dataset. This step simulates the real-world application of the recommendation algorithm.
- **Get Reward:** Upon executing the recommended job role, the DQN agent receives a reward from the environment. This reward is a numerical value reflecting the success or suitability of the recommended job role for the candidate. It serves as feedback for the agent's decision-making process.
- **Update Q Table:** The Q-value for the current state-action pair is updated based on the observed reward and the Q-learning formula. This update reflects the agent's updated understanding of the effectiveness of recommending that particular job role to candidates with similar profiles. Through repeated iterations and interactions with the environment, the DQN agent learns to optimize its policy for job role recommendation.

Chapter 3

Methodology

In this section, we will write about our methoology part. As highlighted in the Literature Review section, the majority of studies in the field of recommender systems primarily utilize machine learning (ML) methodologies. However, we intentionally diverged from this prevalent trend by exploring reinforcement learning (RL) techniques. This decision was influenced by the findings and methodologies outlined in a specific study [9].

Our methodology involved a deliberate shift towards reinforcement learning, as depicted in the methodology flow chart 3.1. This decision was driven by the desire to conduct a comparative analysis between these two paradigms—machine learning and reinforcement learning—in order to determine which model demonstrated superior accuracy and efficacy within the specific context of our task.

By deviating from the conventional approach and exploring reinforcement learning, we aimed to provide a fresh perspective on the development of recommendation systems. This deliberate divergence allowed us to explore the potential advantages and limitations of RL techniques in comparison to traditional ML approaches, ultimately seeking to enhance the effectiveness and performance of our recommendation system.

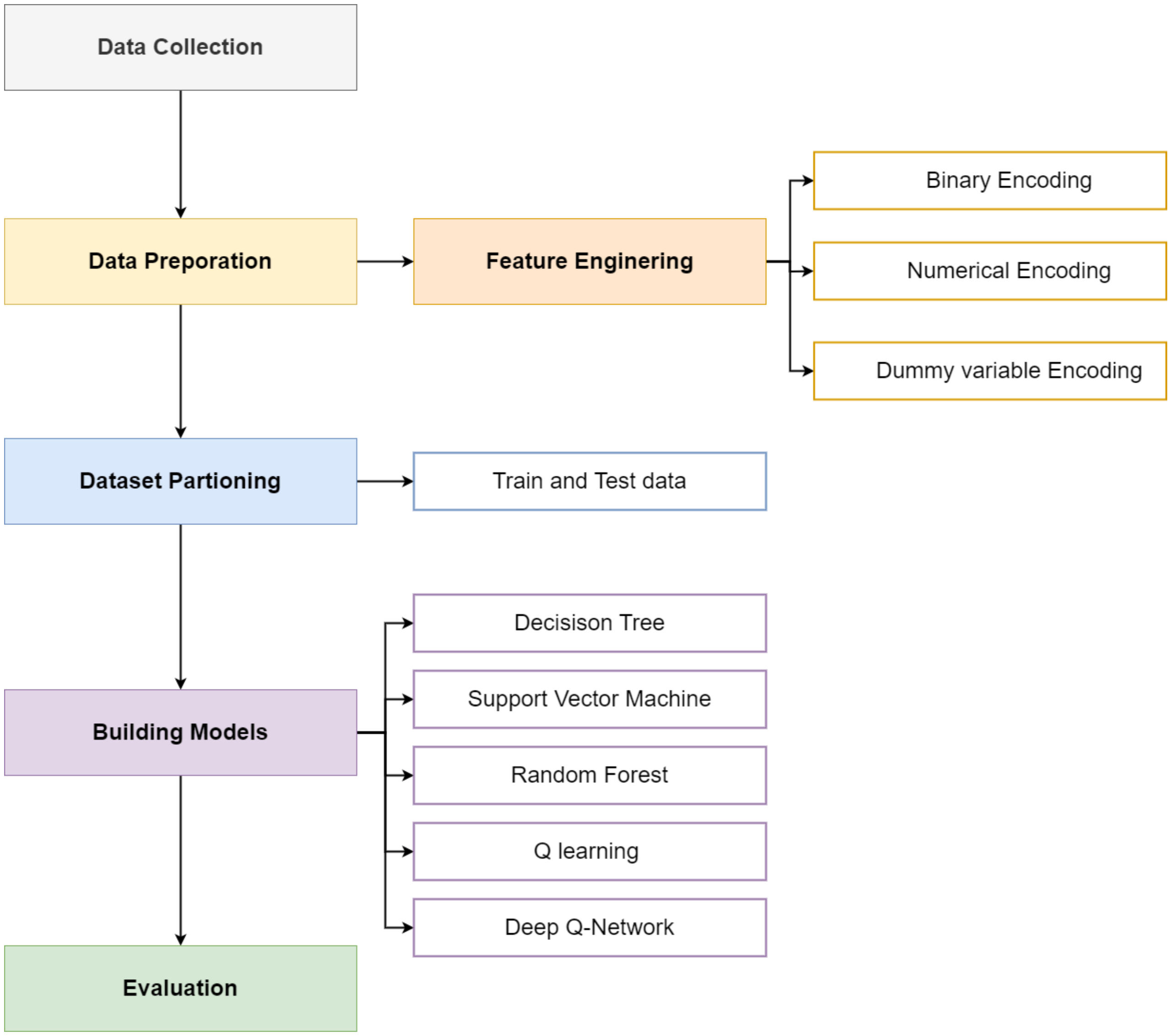

Figure 3.1 – Methodology flowchart

3.1 Dataset Source and Description

The dataset used in this study was sourced from public data available on GitHub [71], containing information about 6901 professionals and comprising 18 features as illustrated in Table 3.1. These features consist of both numerical and categorical variables, each providing valuable insights into the characteristics of the professionals.

The dataset includes 4 numerical features, such as Logical Quotient Rating, Hackathons participated in, Coding Skills Rating, and Public Speaking Points. These numerical features are likely to play a significant role in the analysis, as they provide quantifiable metrics for the skills and abilities of the professionals.

Additionally, the dataset contains 14 categorical features, which offer information in the form of categories or labels. Some of these categorical features require conversion to numerical values for analysis. These conversions include binary responses (Yes/No), ordinal values (e.g., poor, medium, excellent), and categorical values with multiple levels.

Table 3.1 – Description of Dataset Features

Feature	**Description**	**Attribute**
Numerical	Logical Quotient Rating	0-9
	Hackathons	0-9
	Coding Skills Rating	0-9
	Public Speaking Points	0-9
Categorical	Self-Learning Capability	Yes or No
	Extra-Courses Did	Yes or No
	Taken Inputs From Seniors Or Elders	Yes or No
	Worked In Teams Ever?	Yes or No
	Introvert	Yes or No
	Reading And Writing Skills	Poor, medium, excellent
	Memory Capability Score	Poor, medium, excellent
	Management Or Technical	Management Or Technical
	Hard/Smart Worker	Hard Worker or Smart Worker
	Certifications	R Programming, Information Security, Shell Programming, Machine Learning, Full Stack, Hadoop, Python, Distro Making, App Development
	Workshops	Database Security, System Designing, Web Technologies, Hacking, Testing, Data Science, Game Development, Cloud Computing
	Interested Subjects	Software Engineering, Iot, Cloud Computing, Programming, Networks, Computer Architecture, Data Engineering, Hacking, Management, Parallel Computing
	Interested Career Area	System Developer, Security, Business Process Analyst, Developer, Testing, Cloud Computing
	Type Of Company Want To Settle In?	Service Based, Web Services, Bpa,Testing And Maintainance Services, Product Based, Finance, Cloud Services, Product Development, Sales And Marketing, Saas Services

3.1.1 Data preparation

To make sure the dataset was prepared for analysis and modeling, a number of important actions were conducted during the data preparation phase. Among these actions were:

- Loaded Data: The dataset was imported into a Pandas DataFrame from a CSV file in the first phase. This made it simple to use Python for data manipulation and analysis.
- Checked Missing Values: Next, any null or missing values were looked for in the dataset. It was crucial to find and deal with missing values since they can affect how accurate the analysis and modeling process is.
- Analyzed Categorical Values: The dataset's unique categorical values were then examined. Comprehending the distinct values of categorical variables is crucial for the purposes of feature engineering and modeling.
- Examined Data Balance: It is important to look into the data balance between various classes when doing classification activities. A biased model may be more accurate for the majority class but perform badly for minority classes due to an imbalance in the dataset. As a result, creating a strong and impartial model requires making sure that the distribution of classes is equal.

We made sure the dataset was clear, balanced, and well-understood by carrying out these crucial procedures during the data preparation phase, which provided the groundwork for the analysis and modeling phases that followed.

3.1.1.1 Feature Engineering

As shown in Table 3.2, we performed essential changes on the categorical variables during the Feature Engineering stage in order to get them ready for incorporation into predictive models. This step is crucial because it makes sure the data is in a format that the models can process and interpret correctly. The categorical variables must undergo certain changes in order for the predictive models to function better and produce more accurate predictions.

This stage's transformations are intended to change the categorical variables into a format that may be used with predictive models. Since many machine learning methods require numerical input, this frequently entails transforming category data into numerical representations.

- Binary Encoding: To translate binary responses ("yes" or "no"), we used binary coding (1 or 0). This helped improve our models' understanding and use of binary inputs, allowing us to integrate them directly into our models.
- Numerical Encoding: For categorical variables (such as "poor," "average," and "excellent"), we used numeric coding that had ordinal values. To preserve natural order, we ensured that ordinal categories were preserved in their numerical equivalents (0, 1, 2). This was done so that our models could

effectively understand and use the ordinal value of these variables by having numerical representations.

- Dummy Variable Encoding: For categorical attributes, we used dummy variable coding. This technique resulted in generating additional binary columns that corresponded to distinct feature categories. By incorporating these binary indicators, the models were able to gain better insights into the various categorical data sets and facilitate their interpretation for predictive analysis purposes.

Table 3.2 – Features After Feature Engineering

Feature	Description	Attribute
Numerical	Logical Quotient Rating	0-9
	Hackathons	0-9
	Coding Skills Rating	0-9
	Public Speaking Points	0-9
Binary Encoding	Self-Learning Capability	1 or 0
	Extra-Courses Did	1 or 0
	Inputs From Seniors	1 or 0
	Worked In Teams Ever?	1 or 0
	Introvert	1 or 0
Numerical Encoding	Reading And Writing Skills	0, 1, 2
	Memory Capability Score	0, 1, 2
Dummy variables	Management Or Technical	A_Management, B_Technical
	Hard/Smart Worker	A_Hard Worker, B_Smart Worker
Categorical	Certifications	R Programming, Information Security, Shell Programming, Machine Learning, Full Stack, Hadoop, Python, Distro Making, App Development
	Workshops	Database Security, System Designing, Web Technologies, Hacking, Testing, Data Science, Game Development, Cloud Computing
	Interested Subjects	Software Engineering, Iot, Cloud Computing, Programming, Networks, Computer Architecture, Data Engineering, Hacking, Management, Parallel Computing
	Interested Career Area	System Developer, Security, Business Process Analyst, Developer, Testing, Cloud Computing
	Type Of Company Want To Settle In?	Service Based, Web Services, Bpa,Testing And Maintainance Services, Product Based, Finance, Cloud Services, Product Development, Sales And Marketing, Saas Services

The dataset comprises of 13 numerical and 5 categorical features.

To ensure the categorical data was understandable and usable by the models, some adjustments were necessary. The transformation of ordinal categories into numerical counterparts and binary variables into numeric representations helped enhance our ability to identify hidden patterns and correlations in the data. Moreover, we boosted model performance and prediction accuracy through more comprehensive comprehension of various categories present in the information set by adding dummy variable encoding for features with multiple levels.

3.1.2 Dataset Partitioning

The training set and the testing set were the two distinct sets generated during the dataset splitting phase. The purpose of this part is to make the process of developing and assessing the prediction models easier. Eighty percent of the dataset was reserved for training, and twenty percent was set aside for testing. (see Figure 3.2).

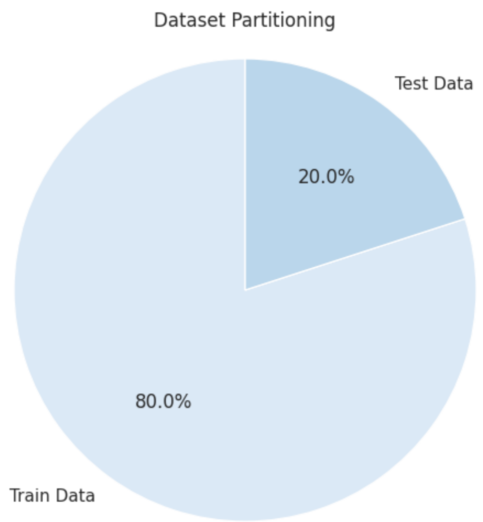

Figure 3.2 – Dataset Partioning

The training set is used to train the models in order to aid them in identifying underlying relationships and patterns in the data. On the other hand, the testing set is utilized to assess the performance of the trained models with unknown data. As a result, evaluating how effectively the models generalize to novel and untested situations is much simpler.

One ratio that's often used to partition datasets is the 80/20 split. It finds a compromise between the amount of data needed for testing to produce a reliable performance rating and the amount needed for efficient model training. We successfully trained and assessed the predictive models by segmenting the dataset in this manner, proving their dependability and ability to generate accurate predictions on fresh data.

3.2 Building model

In order to forecast job roles based on the attributes that were extracted from the dataset, we go into great length in this part regarding the selection and use of several classifiers, including machine learning and reinforcement learning techniques. This study aims to assess and compare various algorithms' accuracy in identifying and categorizing different job types by extracting extensive data from the dataset through feature engineering.

By carefully selecting appropriate classifiers, we conducted a thorough examination of the dataset with the aim of achieving an improved understanding in predicting job roles. Utilizing distinctive advantages and features associated with each approach required coordinated efforts. The decision tree classifier is perfect for comprehending the underlying decision-making process because of its interpretability and simplicity. However, because the support vector machine (SVM) classifier is so good at handling high-dimensional data and challenging decision limits, it might be useful to find little trends in the dataset.

To create a more dependable prediction model, the random forest classifier was also incorporated due to its capacity to manage noise and overfitting. Using reinforcement learning methods, namely Q-learning and the Deep Q-Network (DQN) model, the applicability of reinforcement learning was investigated in this scenario, where the prediction job may be described as a sequential decision-making process.

Below you can see formulas of each classifier:

1. Decision Tree Classifier:

   Based on the input attributes, the Decision Tree Classifier is a straightforward but effective technique for making judgments. The decision tree model uses a tree structure and a set of features to branch based on feature values, guiding it toward a final prediction or option.

$$\textbf{Decision} = \textbf{Tree(Features)} \tag{3.1}$$

   - The forecast or result from the decision tree model is represented by the decision.
   - Tree(features) describes how the decision tree's input qualities are used to make judgments. The tree structure branches based on feature values so that a decision can be made.

2. Random Forest Classifier:

   The Random Forest Classifier is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification tasks or the mean prediction for regression tasks. The predicted output can be calculated as:

   The Random Forest Classifier is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification tasks or the mean prediction for regression tasks. The predicted output can be calculated as:

   The Random Forest Classifier is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification tasks or the mean prediction for regression tasks. The predicted output can be calculated as:

$$\mathbf{\hat{Y}} = \frac{\mathbf{1}}{\mathbf{N}} \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{f_i(X)} \tag{3.2}$$

Where:

- $\hat{Y}$ is the predicted output.
- N is the number of trees in the forest.
- $f_i(X)$ is the prediction of the i-th tree in the forest for input X.

3. Support Vector Machine Classifier:

   The Support Vector Machine classification finds the optimal hyperplane to separate data points into different classes. It maximizes the margin between classes, making it robust to outliers. The decision function of an SVM classifier can be expressed as:

$$\mathbf{f(x)} = \mathrm{sign}(\mathbf{w^T x + b}) \tag{3.3}$$

- $f(x)$ is the decision function that predicts the class label of a new data point x.
- w is the weight vector perpendicular to the hyperplane.
- b is the bias term.
- x is the input data vector.

4. Q-Learning Reinforcement Learning Model:

   Q-Learning is a model-free reinforcement learning algorithm that learns the optimal action-selection policy for any given Markov decision process $Q(s, a)$ is updated iteratively using the Bellman equation:

$$\mathbf{Q(s,a)} = (\mathbf{1} - \alpha) \cdot \mathbf{Q(s,a)} + \alpha \cdot [\mathbf{R(s,a)} + \gamma \cdot \max_{\mathbf{a'}} \mathbf{Q(s',a')}] \tag{3.4}$$

- $Q(s, a)$ is the Q-value for state s and action a.
- α is the learning rate, controlling the impact of new information.
- $R(s, a)$ is the immediate reward received after taking action a in state s.
- γ is the discount factor, determining the importance of future rewards.
- $\max_{a'} Q(s', a')$ is the maximum Q-value for the next state s' and possible actions a' from that state.

5. Deep Q-Network Model:

For value approximation and action selection, the Deep Q-Network uses intricate neural network designs. Deep neural networks, in contrast to the others, do not have a single formula and are instead utilized to approximate Q-values and select actions depending on those values. You can see the difference between Q-Learning and Deep Q-Network in figure 3.3

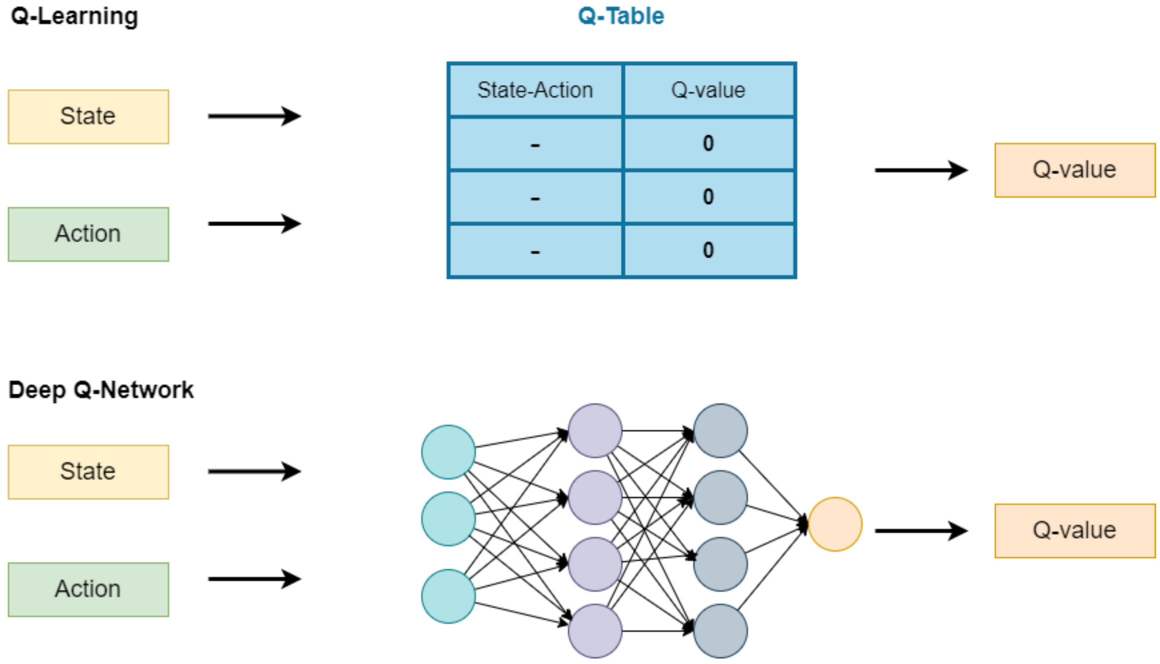

Figure 3.3 – Q-Learning vs Deep Q-Network

Chapter 4

Results

This study aimed to compare the performance of reinforcement learning (RL) and machine learning (ML) models in a job recommender system. Random Forest, Decision Tree, Support Vector Machine (SVM), Deep Q-Network (DQN), and Q-learning are the five models that were compared. Figure 4.1 displays the performance scores that each model achieved, providing a comparative examination of their individual performance. Specifically, the outcomes offer strong proof of these models' potential in the field of job counseling.

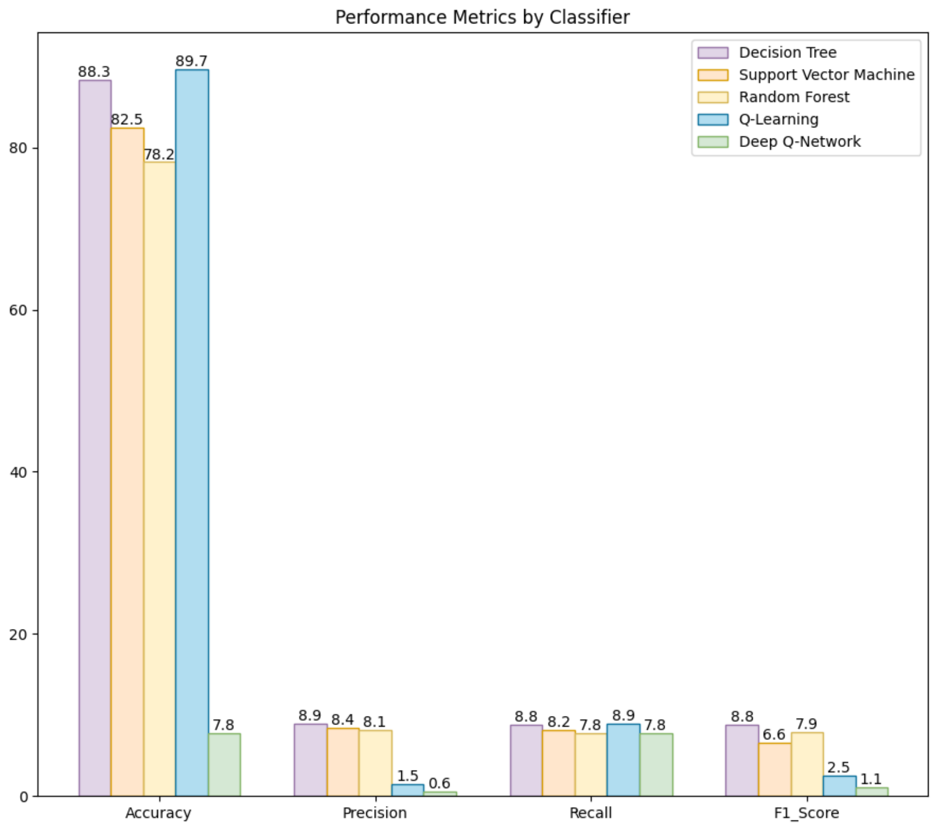

Figure 4.1 – Performance Scores for Different Classifier Models

The job recommendation system's performance was evaluated using various classifiers, including Decision Tree, Support Vector Machine, Random Forest, Q Learning, and Deep Q-Network. The evaluation metrics used to assess the classifiers included accuracy, precision, recall, F1 score, and runtime. Table 4.1 below provides a detailed summary of the performance scores and training and testing timeframes for each model.

Table 4.1 – Performance Metrics

Classifier	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**Runtime**
Decision Tree	88.3%	8.9%	8.8%	8.8%	0.257 sec
SVM	82.5%	8.4%	8.2%	6.6%	9.389 sec
Random Forest	78.2%	8.1%	7.8%	7.9%	2.395 sec
Q Learning	89.7%	1.5%	8.9%	2.5%	12 sec
DQN	7.8%	0.6%	7.8%	1.1%	1080 sec

The Decision Tree classifier exhibited commendable performance, achieving an accuracy of 88.3%. This high accuracy rate underscores the classifier's capability to accurately classify job recommendations based on input data features. The precision, recall, and F1 score metrics further affirm the classifier's effectiveness, with respective values of 8.9%, 8.8%, and 8.8%. Notably, the Decision Tree demonstrated efficient processing, with a runtime of a mere 0.257 seconds.

Support Vector Machine (SVM) showcased promising performance, yielding an accuracy of 82.5%. Despite slightly lower precision, recall, and F1 score values compared to the Decision Tree classifier (8.4%, 8.2%, and 6.6%, respectively), SVM demonstrated robust capabilities in classifying job recommendations. However, this efficacy came at the cost of increased computational overhead, with SVM incurring a runtime of 9.389 seconds.

The Random Forest classifier, leveraging ensemble learning techniques, achieved an accuracy of 78.2%. Its performance metrics closely mirrored those of the Decision Tree classifier, with precision, recall, and F1 score values of 8.1%, 7.8%, and 7.9%, respectively. Remarkably, Random Forest exhibited computational efficiency, boasting a runtime of 2.395 seconds.

Q Learning, an exemplar of reinforcement learning algorithms, delivered impressive results with an accuracy of 89.7%. Its ability to discern effective job recommendation strategies through trial and error was evident. However, Q Learning showcased relatively lower precision (1.5%) balanced by a high recall rate (8.9%). The resultant F1 score of 2.5% reflects the trade-off between precision and recall. Notably, Q Learning required a moderate computational investment, with a runtime of 12 seconds.

Deep Q-Network (DQN), harnessing the power of deep learning, exhibited the lowest accuracy among the classifiers, standing at 7.8%. This suboptimal performance was echoed in its precision, recall, and F1 score metrics, which registered at 0.6%, 7.8%, and 1.1%, respectively. Furthermore, DQN incurred the highest computational overhead, necessitating a runtime of 1080 seconds. This substantial computational demand underscores the challenges associated with deep learning-based approaches in the context of job recommendation systems.

The comprehensive evaluation of these classifiers offers invaluable insights into their effectiveness and efficiency within the job recommendation system framework. Q Learning emerged as the top-performing classifier, boasting the highest accuracy and demonstrating potential in personalized job recommendations. Conversely, while Deep Q-Network exhibited substantial computational demands, its performance fell short of expectations, highlighting the intricate challenges associated with deep learning-based methodologies in this domain.

The Decision Tree and Random Forest classifiers showcased commendable performance, striking a balance between accuracy and computational efficiency. SVM, while effective, exhibited higher computational costs, suggesting a need for optimization in resource-constrained environments.

Chapter 5

Conclusions and future work

5.1 Conclusions

In this study, the functionality of job recommending system was evaluated by implementing machine learning and reinforcement learning models. Q-learning, Deep Q-Network, Decision Tree, Random Forest and Support Vector Machine were tested for their ability to generate reliable employment suggestions using a dataset comprising various factors pertaining to talents, interests and career preferences. The selection criterion was primarily based on these models' applicability in generating accurate recommendations. Our evaluation was based on key metrics such as accuracy, precision, recall, F1 score, and runtime.

The results revealed important insights into the strengths and limitations of each classifier. Q Learning emerged as the top-performing classifier in terms of accuracy, demonstrating its potential for learning effective job recommendation strategies through trial and error. However, its relatively low precision indicates the need for further optimization to reduce false positives.

The Decision Tree and Random Forest classifiers demonstrated competitive performance, achieving a balance between accuracy and computational efficiency. These classifiers are well-suited for scenarios where interpretability and runtime efficiency are crucial considerations.

Support Vector Machine showed promising accuracy but incurred higher computational costs compared to other classifiers. Further optimization may be required to enhance its efficiency for large-scale job recommendation tasks.

Deep Q-Network, despite its potential in other domains, exhibited limited effectiveness in the job recommendation context. Its low accuracy and high computational overhead suggest challenges associated with applying deep learning-based approaches to this task.

While reinforcement learning techniques, such as Q Learning, yielded the highest accuracy, traditional machine learning methods still demonstrated strong performance. The Decision Tree and Random Forest classifiers, representing ML techniques, provided competitive accuracy while maintaining efficiency.

The novelty of this research lies in its focus on reinforcement learning, driven by its adaptive nature. While many studies rely on machine learning, our approach addresses the issue of students making career choices based on their interests by providing personalized guidance, reinforcement learning encourages students to explore options aligned with their preferences.

Reinforcement learning's ability to adapt in real-time addresses the problem of poor choices resulting from limited career guidance. It ensures that recommendations remain relevant in a constantly changing job market, mitigating the risk of outdated advice. By focusing on reinforcement learning, the research aims to bridge the gap in career guidance by offering a dynamic and personalized approach. This directly addresses the issues highlighted in the problem statement, ultimately aiming to improve the quality of career decision-making among students.

By providing valuable insights into the performance of different classifiers, this study contributes to the advancement of job recommendation systems, ultimately enhancing the effectiveness of matching job seekers with suitable employment opportunities.

5.2 Future work

Because advising is at the intersection of technological innovation and career guidance, there are several opportunities to further improve its effectiveness and impact. In the dynamic educational technology landscape, the following key areas are emerging as future research focuses:

- Introduction of a recommendation system with consistent assistance in choosing elective courses: The evolving nature of academic programs requires an expanded role for the recommendation system to provide ongoing support in the selection of elective courses. Strengthening algorithms that help students not only select initial career paths but also navigate the diverse landscape of elective courses can go a long way toward facilitating a more personalized and well-rounded educational path.

- Implementation of a real-time recommendation system: The future trajectory of a recommendation system depends on its ability to adapt in real-time to dynamic changes in industry requirements and emerging labor market trends. Integrating reinforcement learning to provide real-time updates and responsiveness to an ever-changing professional environment will be critical to ensuring the system remains up-to-date and effective.

- Development of a website or application for interaction with students: To increase student engagement, it is necessary to create a special platform, be it a website or a mobile application. This will allow students to enter preferences, respond to dynamic questionnaires and receive personalized career advice. Additionally, features for tracking academic progress and accessing resources related to selected careers can improve the overall user experience.

- Integration of a user feedback mechanism: A fundamental aspect of improvement is the integration of a user feedback mechanism. Collecting ideas and experiences from students who have used the recommendation system will provide valuable data for iterative improvements. Creating continuous feedback ensures adaptation to individual needs, which ultimately improves user satisfaction.

- Cooperation with educational institutions and industry partners: Strengthening connections with educational institutions and industry partners is critical to aligning the recommendation system with real-world requirements. The collaboration provides access to current data, industry knowledge and the expertise of scientific advisors. This partnership ensures that the recommendation system meets academic requirements and industry expectations.

Following these directions, the recommendation system is designed to not only effectively guide students in their career paths, but also to make a meaningful contribution to the continuous development of educational technologies. As collaborative efforts evolve, researchers, educators, and industry professionals will collectively shape a future in which career guidance is personalized and aligned with the changing needs of students and professional environments.

Bibliography

[1] Alexandra Budjanovcanin and Chris Woodrow. Regretting your occupation constructively: A qualitative study of career choice and occupational regret. *Journal of Vocational Behavior*, 136, 08 2022. doi: 10.1016/j.jvb.2022.103743. URL https://doi.org/10.1016/j.jvb.2022.103743.

[2] Melodie Chamandy and Patrick Gaudreau. Career doubt in a dual-domain model of coping and progress for academic and career goals. *Journal of Vocational Behavior*, 110:155–167, 02 2019. URL https://www.sciencedirect.com/science/article/abs/pii/S0001879118301453.

[3] Aksinya Titova. Kazakhstanis drop out of university en masse. *MK-kz*, 04 2022. URL https://mk-kz.kz/social/2022/04/20/kazakhstancy-massovo-brosayut-uchebu-v-vuzakh.html.

[4] Baigarin M. The right choice of profession depends on career guidance. *News kazinform*, 05 2022. URL https://www.inform.kz/ru/pravil-nyy-vybor-professii-zavisit-ot-proforientacii_a3936986.

[5] Sushma Koushik, Chandana M., Lavanya V., Suhas Y., and Harshitha V. Educational career recommendation system using machine learning. *IJARCCE*, 08:1–104, 01 2021. doi: 10.17148/IJARCCE.2021.10809. URL https://ijarcce.com/wp-content/uploads/2021/09/IJARCCE.2021.10809.pdf.

[6] Na Chen, Ying Li, and Gang Zheng. Research on personalized recommendation algorithms for the web with knowledge representation. *SPIE. Digital library*, 06 2023. doi: 10.1117/12.2681370. URL https://www.spiedigitallibrary.org/conference-proceedings-of-spie/12707/2681370/Research-on-personalized-recommendation-algorithms-for-the-web-with-knowled 10.1117/12.2681370.short.

[7] Fei Wang and Xiaoxia Liu. Personalized book recommendation algorithm based on fusion of content and improved collaborative filtering. *International Conference on Advanced Algorithms and Signal Image Processing*, 10 2023. doi: 10.1117/12.3006117. URL https://www.researchgate.net/publication/374614170_Personalized_book_recommendation_algorithm_based_on_fusion_of_content_and_improved_collaborative_filtering.

[8] Hui Li. Research on the design of oral english system based on con-

tent recommendation. *European Union Digital Library*, 03 2023. doi: 10.4108/eai.2-12-2022.2328066. URL https://eudl.eu/doi/10.4108/eai.2-12-2022.2328066.

[9] CHALLENGES IN CAREER RECOMMENDATION SYSTEMS A COMPREHENSIVE REVIEW OF APPROACHES. Assemay imankulova, arailym serikbay, meraryslan meraliyev. *SDU Bulletin*, 03 2024. doi: 10.47344/sdubnts.v64i1.1148. URL https://journals.sdu.edu.kz/index.php/nts/article/view/1148.

[10] Hmood Al-Dossari, Ziyad Al-Qahtani, Fawaz Abu Nughaymish, Mohammed Alkahlifah, and Asma Alqahtani. A machine learning approach to career path choice for information technology graduates. *English Technology Applied Science.*, 10:6589—-6596, 12 2020. URL https://etasr.com/index.php/ETASR/article/view/3821/2417.

[11] Suleiman Ali Alsaif, Minyar Sassi Hidri, Hassan Ahmed Eleraky, Imen Ferjani, and Rimah Amami. Learning-based matched representation system for job recommendation. *Computers, MDPI*, 11, 11 2022. URL https://www.mdpi.com/2073-431X/11/11/161.

[12] Dahanke Ajay, Shinde Nilesh, Dhagate Anirudh, and Shaikh Huzaif. An intelligent career guidance system using machine learning. *IRJET*, 09:1684–1691, 03 2022. URL https://www.irjet.net/archives/V9/i3/IRJET-V9I3310.pdf.

[13] John Idakwo, Dr. Agbogun Joshua Babatunde, and Dr. Taiwo Kolajo. Development of a hybrid students' career path recommender system using machine learning techniquesg. *FUW Trends in Science Technology Journal*, 07: 236–242, 12 2022. URL https://www.ftstjournal.com/uploads/docs/73%20Article%2033a%20pp%20236-242.pdf.

[14] Poonam Parate, Mukesh Barapatre, Harish Kalode, Shruti Harkut, Shivam Kamdi, and Arpit Alne. Machine learning base student future orientation and recommendation system. *IJARSCT*, 04, 04 2021. URL https://ijarsct.co.in/Paper1049.pdf.

[15] Adewale Opeoluwa Ogunde and Joy Oseremen Idialu. A recommender system for selecting potential industrial training organizations. *Engineering Reports, Wiley Online Library*, 01, 10 2019. doi: 10.1002/eng2.12046. URL https://doi.org/10.1002/eng2.12046.

[16] M C B Natividad, B D Gerardo, , and R P Medina. A fuzzy-based career recommender system for senior high school students in k to 12 education. *IOP Conf. Ser.: Mater. Sci. Eng. 482 012025*, 2019. URL https://iopscience.iop.org/article/10.1088/1757-899X/482/1/012025.

[17] Roshan G. Belsare and Dr. V. M. Deshmukh. Employment recommendation system using matching, collaborative filtering and content based recommendation. *IJARSCT*, 07:215–220, 2018. URL https://ijcat.com/archieve/volume7/issue6/ijcatr07061003.pdf.

[18] Akshay Gugnani, Vinay Kumar Reddy Kasireddy, and Karthikeyan Ponnalagu. Generating unified candidate skill graph for career path recommendation. *IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 328–333, 2018. doi: 10.1109/ICDMW. URL https://akshaygugnani.github.io/pdf/IEEE_ICDMW_Skill%20Extraction_paper.pdf.

[19] Ahsan Zubair, Asim Shahzad, Muhammad Zeeshan, Jamaluddin Mir, Muhammad Zubair Tahir, Aamer Khan, Muhammad Sibt ul Hassan, Muhammad Naeem, and Mehmood Ahmed. Career selection for students in the information technology sector using a hierarchical fuzzy-based ontological recommender system approach. *Webology*, 19: 8218–8230, 2022. URL https://www.webology.org/data-cms/articles/20220819063701pmwebology%2019%20(1)%20-%20178.pdf.

[20] Lamyae El Youbi El Idrissi, Ismail Akharraz, and Abdelaziz Ahaitouf. Personalized e-learning recommender system based on autoencoders. *MDPI*, 2023. doi: 10.3390/asi6060102. URL https://www.mdpi.com/2571-5577/6/6/102.

[21] Phuwadol Viroonluecha and Thongchai Kaewkiriya. Improving university programme recommender system based on mbti by using gradient boosted trees and firefly algorithm. *International Journal of the Computer, the Internet and Management*, 09 2018. URL https://www.researchgate.net/publication/335061480_Improving_University_Programme_Recommender_System_Based_on_MBTI_by_Using_Gradient_Boosted_Trees_and_Firefly_Algorithm_International_Journal_of_the_Computer_the_Internet_and_Management.

[22] Gregorius Ryan, Pricillia Katarina, and Derwin Suhartono. Mbti personality prediction using machine learning and smote for balancing data based on statement sentences. *MDPI*, 02 2023. doi: 10.3390/info14040217. URL https://www.mdpi.com/2078-2489/14/4/217.

[23] Roshan G. Belsare and Dr. V. M. Deshmukh. Employment recommendation system using matching, collaborative filtering and content based recommendation. *International Journal of Computer Applications Technology and Research*, 7:215–220, 2018. URL https://ijcat.com/archieve/volume7/issue6/ijcatr07061003.pdf.

[24] Joy Dhar and Asoke Kumar Jodder. An effective recommendation system to forecast the best educational program using machine learning classification algorithms. *International Information and Engineering Technology Association*, 25:559–568, 09 2020. doi: 10.18280/isi.250502. URL https://www.semanticscholar.org/paper/Machine-Learning-based-Ideal-Job-Role-Fit-and-S.-S./c77deb31bdac0d043f0b427456e3eb01facb2ff9.

[25] Qing Wan and Lin Ye. Career recommendation for college students based on deep learning and machine learning. *Hindawi*, 08 2023. doi: 10.1155/2023/9789507. URL https://www.hindawi.com/journals/sp/2023/9789507/.

[26] Ch.Suneetha, M.Jahnavi, Yashwanth, and Naonit Kumar Goutam. Course recommendation system using machine learning. *Hindawi*, 8, 06 2023. doi: 10.1155/2023/9789507. URL https://www.ijsdr.org/papers/IJSDR2306255.pdf.

[27] Tribhuwan Kumar, Sakthidasan Sankaran, Mahyudin Ritonga, Shazia Asif, Sathiya Kumar, Shoaib Mohammad, Sudhakar Sengan, and Evans Asenso. Fuzzy logic and machine learning-enabled recommendation system to predict suitable academic program for students. *Hindawi*, 08 2022. doi: 10.1155/2022/5298468. URL https://www.hindawi.com/journals/mpe/2022/5298468/.

[28] Brijmohan Daga, Juhi Checker, Anne Rajan, and Sayali Deo. Computer science career recommendation system using artificial neural network. *International Journal of Computer Trends and Technology*, 68:84–88, 03 2020. doi: 10.14445/22312803/IJCTT-V68I3P117. URL https://www.ijcttjournal.org/2020/Volume-68%20Issue-3/IJCTT-V68I3P117.pdf.

[29] N. VidyaShreeram and Dr. A. Muthukumaravel. Student career prediction using decision tree and random forest machine learning classifiers. *EAI*, 06 2021. doi: 10.4108/eai.7-6-2021.2308621. URL https://eudl.eu/pdf/10.4108/eai.7-6-2021.2308621.

[30] Yosua F Simanjuntak and Atoni Wibowo. Recommendation system for online job vacancy using machine learning models. *Journal of Theoretical and Applied Information Technology*, 101, 01 2023. URL https://www.jatit.org/volumes/Vol101No2/28Vol101No2.pdf.

[31] Asmita Kamble, Shreyash Tambe, Hritik Bansode, Sudhanva Joshi, and Samruddhi Raut. Job recommendation system for daily paid workers using machine learning. *International Journal for Research in Applied Science Engineering Technology*, 11, 04 2023. URL https://www.ijraset.com/best-journal/job-recommendation-system-for-daily-paid-workers-using-machine-learning.

[32] Timothy Dicky, Alva Erwin, and Heru Purnomo Ipung. Developing a scalable and accurate job recommendation system with distributed cluster system using machine learning algorithm. *Journal of Applied of Information, Communication and Technology*, 7, 10 2020. URL https://www.semanticscholar.org/paper/Developing-a-Scalable-and-Accurate-Job-System-with-Dicky-Erwin/6b3d3ea4d56e73bade5040aea120626b9e2911ed.

[33] G.Mahalakshmi, A.Arun Kumar, B.Senthilnayaki, and J.Duraimurugan. Job recommendation system based on skill sets. *International Journal of Creative Research Thoughts*, 10, 08 2022. URL https://ijcrt.org/papers/IJCRT2208099.pdf.

[34] Amber Nigam, Aakash Roy, Hartaran Singh, and Aabhas Tonwer. Job recommendation through progression of job selection. *International Conference on Cloud Computing and Intelligence Sys-*

tems, 05 2019. URL https://www.semanticscholar.org/paper/Job-Recommendation-through-Progression-of-Job-Nigam-Roy/921eed17ddc38bdc026a887ef63fc6b091777f0a.

[35] Abdullah Jawaid Man-Fai Leung1, *, Sai-Wang, Chun-Hei Kwok, and Shing Yan. A portfolio recommendation system based on machine learning and big data analytics. *Data Science in Finance and Economics*, 3:152—-165, 05 2023. doi: 10.3934/DSFE.2023009. URL https://www.aimspress.com/article/doi/10.3934/DSFE.2023009.

[36] Rohan Lohia, Sumit Bhatnagar, Mr. Vibhor Sharma, and Mr. Nitesh Kumar. Recommendation system using machine learning. *International Journal of Innovative Research in Science, Engineering and Technology*, 10, 06 2021. doi: 10.15680/IJIRSET.2021.1006141. URL https://www.ijirset.com/upload/2021/june/141_A_Study_NC1.pdf.

[37] Qing Wan and Lin Ye. Career recommendation for college students based on deep learning and machine learning. *Hindawi*, 2022. doi: 10.1155/2022/3437139. URL https://www.semanticscholar.org/paper/Career-Recommendation-for-College-Students-Based-on-Wan-Ye/c28e983776be853df7bf20c28dc64a9bcfa4aae9.

[38] Jihad Zahir and Abderrahim El Qadi. A recommendation system for execution plans using machine learning. *Mathematical and Computational Applications*, 06 2016. URL https://www.semanticscholar.org/reader/4ecba007221548b1352f6274fe21c2d6a1bef263.

[39] Agnes van Belle, Eike Dehling, and Daniel Foster. Improving candidate to job matching with machine learning. *SemanticScholar*, 2018. URL https://dshcmorg.files.wordpress.com/2018/08/dshcm_2018_paper_5_job_matching.pdf.

[40] Yixin Bao, Yanghua Peng, Chuan Wu, and Zongpeng Li. Online job scheduling in distributed machine learning clusters. *SemanticScholar*, 06 2018. URL https://dshcmorg.files.wordpress.com/2018/08/dshcm_2018_paper_5_job_matching.pdf.

[41] Ch.Suneetha, M.Jahnavi, M.Yashwanth, and Naonit Kumar Goutam. Course recommendation system using machine learning. *International Journal of Scientific Development and Research*, 8, 06 2023. URL https://www.ijsdr.org/papers/IJSDR2306255.pdf.

[42] P Ramya, S G Balakrishnan, and M Kannan. Recommendation system to improve students performance using machine learning. *IOP Conference Series: Materials Science and Engineering*, 2020. URL https://iopscience.iop.org/article/10.1088/1757-899X/872/1/012038.

[43] Blessing C. Uzo and Collins N. Udanor. Towards an efficient machine learning algorithm for a graduate study elective course recommendation system. *International Journal of Mechatronics, Electrical and Computer Technology*,

11:5036–5042, 2021. URL https://aeuso.org/includes/files/articles/Vol11_Iss41_5036-5042_Towards_an_Efficient_Machine_Learni.pdf.

[44] Zeinab Shahbazi and Yung-Cheol Byun. Agent-based recommendation in e-learning environment using knowledge discovery and machine learning approaches. *MDPI*, 04 2022. URL https://www.semanticscholar.org/reader/460edd398c293f7b5305bd245b2f1c788708a3e0.

[45] Lakshmi prasanna and DR.D.Haritha. Smart career guidance and recommendation system. *IJEDR*, 7, 2019. URL https://web.archive.org/web/20200809154837/https://www.ijedr.org/download.php?file=IJEDR1903111.pdf.

[46] S. Bhaskaran and Raja Marappan. Design and analysis of an efcient machine learning based hybrid recommendation system with enhanced density-based spatial clustering for digital e-learning applications. *Complex Intelligent Systems*, 09 2021. URL https://link.springer.com/content/pdf/10.1007/s40747-021-00509-4.pdf.

[47] Gongwen Xu, Guangyu Jia, Lin Shi, and Zhijun Zhang. Personalized course recommendation system fusing with knowledge graph and collaborative filtering. *Hindawi*, 09 2021. URL https://www.semanticscholar.org/reader/90d6a6b7950d16624cc39c7f739507fb64bac803.

[48] M. Rekha Sundar, G. Shreya, and T. Jawahar. Course recommendation system. *International Journal of Computer Applications*, 175, 11 2020. URL https://www.ijcaonline.org/archives/volume175/number29/sundari-2020-ijca-920823.pdf.

[49] Jovana Jović, Emilija Kisić, Miroslava Raspopović Milić, Dragan Domazet, and Kavitha Chandra. Prediction of student academic performance using machine learning algorithms. *CEUR Workshop Proceedings*, 09 2022. URL https://ceur-ws.org/Vol-3454/article_4.pdf.

[50] Mr. Roshan S. Bhanuse and Mr. Ganesh Yenurkar. Recommendation of branch of engineering using machine learning. *International Research Journal of Engineering and Technology*, 7, 03 2020. URL https://www.irjet.net/archives/V7/i3/IRJET-V7I31039.pdf.

[51] Jallal Talaghzi, Mostafa Bellafkih, and Abdellah Bennane. A combined e-learning course recommender system. *International Journal of Emerging Technologies in Learning*, 18, 06 2023. URL https://www.semanticscholar.org/paper/Combined-E-Learning-Course-Recommender-System-Talaghzi-Bellafkih/e80b78e5c1577086b32ec524168f766b007136c0.

[52] Nacim Yanes and Ayman Mohamed Mostafa. A machine learning-based recommender system for improving students learning experiences. *IEEE Access*, 11 2020. URL https://ieeexplore.ieee.org/ielx7/6287639/8948470/09249379.pdf.

[53] Shaimaa M. Nafea, Francois Siewe, and Ying He. Ulearn: Personalized course learning objects based on hybrid recommendation approach. *International Journal of Information and Education Technology*, 8, 12 2018. URL https://www.ijiet.org/vol8/1151-IE002.pdf.

[54] Mushtaq Hussain, Wenhao Zhu, Wu Zhang, and Syed Muhammad Raza Abidi. Student engagement predictions in an e-learning system and their impact on student course assessment scores. *International Journal of Information and Education Technology*, 09 2018. URL https://www.semanticscholar.org/reader/519ce9c91d8a9fd5dfbb220dde6cb049935779b1.

[55] S.Saranya, K. Harini, J.Srinithi, T.Arjun, and P. Senthil Kumar. A classification-based predictive analysis for online job recommendation system. *International Journal of Creative Research Thoughts*, 8, 03 2020. URL https://ijcrt.org/papers/IJCRT2003303.pdf.

[56] Yara Zayed, Yasmeen Salman, and Ahmad Hasasneh. A recommendation system for selecting the appropriate undergraduate program at higher education institutions using graduate student data. *MDPI*, 12 2022. URL https://www.semanticscholar.org/paper/A-Recommendation-System-for-Selecting-the-Program-Zayed-Salman/45e858f8bbdffbeb844179d47267d897f0ab09fc.

[57] Binny Parida, Prashanta Kumar Patra, and Sthita Pragyan Mohanty. Use of social network for recommending job by applying machine learning techniques. *International Research Journal Of Engineering And Technology*, 7, 06 2020. URL https://www.semanticscholar.org/paper/A-Recommendation-System-for-Selecting-the-Program-Zayed-Salman/45e858f8bbdffbeb844179d47267d897f0ab09fc.

[58] Alisha Baskota and Yiu-Kai Ng. A graduate school recommendation system using the multi-class support vector machine and knn approaches. *IEEE International Conference on Information Reuse and Integration*, 06 2018. URL https://www.semanticscholar.org/paper/A-Graduate-School-Recommendation-System-Using-the-Banskota-Ng/29d307869d9d05aa946c835354fae3d4cfc05572.

[59] Andrzej Pacuk, Piotr Sankowski, Karol Wegrzycki, and Adam Witkowski. Recsys challenge 2016: job recommendations based on preselection of offers and gradient boosting. *SemanticScholar*, 09 2015. URL https://www.semanticscholar.org/reader/6b4c3f5156b2976c8163cec6ab3fae0609fd74b9.

[60] Jie Zhu and Ashraf Yaseen. A recommender for research collaborators using graph neural networks. *Frontiers in Artificial Intelligence*, 5, 2022. URL https://www.frontiersin.org/articles/10.3389/frai.2022.881704/full.

[61] Kumar R. Candidate job recommendation system. *International Jour-*

nal of Scientific Research in Computer Science and Engineering, 6:12–15, 12 2018. URL https://www.frontiersin.org/articles/10.3389/frai.2022.881704/full.

[62] Punitavathi D, Shinu V, Siva Kumar S, and Vidhya Priya S P. Online job and candidate recommendation system. *International Research Journal Of Multidisciplinary Technovation*, 03 2019. URL https://www.semanticscholar.org/reader/57d45d71643e4edb2f0b3c0807735221e070546b.

[63] María Cora Urdaneta-Ponte, Amaia Méndez-Zorrilla, and Ibon Oleagordia-Ruiz. Lifelong learning courses recommendation system to improve professional skills using ontology and machine learning. *MDPI*, 04 2021. URL https://www.semanticscholar.org/reader/66b5207e0d7daf8dfbb5bf2960d76b3828864661.

[64] Chathurangi Shyalika. A beginners guide to q-learning. *Meduim*, 11 2019. URL https://towardsdatascience.com/a-beginners-guide-to-q-learning-c3e2a30a653c.

[65] Tanya V. Yadalam, Vaishnavi M. Gowda, Vanditha Shiva Kumar, Disha Girish, and Namratha M. Career recommendation systems using content based filtering. *5th International Conference on Communication and Electronics Systems (ICCES)*, pages 660–665, 06 2020. doi: 10.1109/ICCES48766.2020.9137992. URL https://ieeexplore.ieee.org/document/9137992.

[66] Snehal Joshi, Manasi Jadhav, Pratiksha Londase, and Sakshi Nikat. Career recommendation system. *iJRASET*, 04 2023. doi: 10.22214/ijraset.2023.50888. URL https://www.ijraset.com/research-paper/career-recommendation-system.

[67] Dai Jing, Gu Xiaoqing, and Jiawen Zhu. Personalized recommendation in the adaptive learning system: the role of adaptive testing technology. *ERIC*, 06 2023. doi: 10.1177/07356331221127303. URL https://eric.ed.gov/?id=EJ1379477.

[68] Chengqian Gao, Ke Xu, Kuangqi Zhou, Lanqing Li, Xueqian Wang, Bo Yuan, and Peilin Zhao. Value penalized q-learning for recommender systems. *arxiv*, 2022. doi: 10.1145/3477495.3531796. URL https://arxiv.org/pdf/2110.07923.pdf.

[69] Yu Lei, Zhitao Wang, Wenjie Li, and Hongbin Pei. Social attentive deep q-network for recommendation. *ACM Digital library*, 06 2019. doi: 10.1145/3331184.3331302. URL https://dl.acm.org/doi/10.1145/3331184.3331302.

[70] Ying Sun, Fuzhen Zhuang1, Hengshu Zhu, Qing He, and Hui Xiong. Cost-effective and interpretable job skill recommendation with deep reinforcement learning. *ACM Digital library*, 04 2021. doi: 10.1145/3442381.3449985. URL https://dl.acm.org/doi/pdf/10.1145/3442381.3449985.

[71] Dataset. URL https://raw.githubusercontent.com/hrugved06/Career-Prediction-System/main/data/mldata.csv.