

CV ANALYSIS AND OPTIMIZING THE RECRUITMENT PROCESS IN THE IT INDUSTRY USING MACHINE LEARNING TECHNIQUES

Project Final (Draft) Report

De Silva S.R.

(IT20216900)

Bachelor of Science (Hons) Degree in Information Technology

Specializing in Data Science

Department of Information Technology

Faculty of Computing

Sri Lanka Institute of Information Technology

Sri Lanka

September 2023

CV ANALYSIS AND OPTIMIZING THE RECRUITMENT PROCESS IN THE IT INDUSTRY USING MACHINE LEARNING TECHNIQUES

De Silva S.R.

(IT20216900)

Bachelor of Science (Hons) Degree in Information Technology
Specializing in Data Science

Department of Information Technology
Faculty of Computing


Sri Lanka Institute of Information Technology
Sri Lanka

September 2023

DECLARATION

I declare that this is our own work, and this proposal does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgment is made in the text.

Also, I hereby grant to Sri Lanka Institute of Information Technology, the nonexclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Name	Student ID	Signature
De Silva S.R.	IT20216900	

The supervisor/s should certify the proposal report with the following declaration.

The above candidates are carrying out research for the undergraduate Dissertation under my supervision.

.....

Signature of the supervisor:

(Dr. Anuradha Karunasena)

.....

Date:

ABSTRACT

Choosing the right individuals is vital to an organization's success and expansion. However, the traditional recruitment approach, which involves manual procedures like screening resumes and academic qualifications, and assessing technical and professional skills, is not only time-consuming but also ineffective. In order to meet the demands of employers, it is essential to adopt an efficient and reliable approach to assess the skills and abilities of candidates. Our proposed solution aims to optimize the hiring process by implementing a system for the IT industry that can effectively identify the most suitable candidate/s for a given job role. To achieve this goal, this research component will analyze the data present in a candidate's academic transcript. This analysis will extract important features such as module names and grades, which will then be categorized into different types and visualized through graphical representation and also get a matching score based on the positions and these skill categories. By doing this, we will be able to identify which candidate possesses the necessary technical skills required for the job role, thus making the recruitment process more efficient and beneficial for the company.

To accomplish this, we will employ Natural Language Processing and machine learning techniques to perform the necessary processing steps. This will enable our system to accurately evaluate a candidate's skills and abilities, providing employers with a more thorough understanding of their potential employees. Overall, our proposed solution offers a promising approach to enhancing the recruitment process and identifying the most effective candidates for a given job role.

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to everyone who supported me throughout this project to make it successful. Firstly, I would like to express my heartfelt appreciation to my supervisor, Dr. Anuradha Karunasena, for her unwavering guidance and support. Also, I would like to express my gratitude to my co-supervisor, Dr. Lakmini Abeywardhana, for her support in completing this research project.

I would also like to express my gratitude to the RP team of Sri Lanka Institute of Information Technology, for giving us the opportunity along with the support and guidance to complete this project. Last but not least, I also sincerely thank my project members, who worked with me and helped me throughout this research to make it a successful project.

Shanali de Silva,

Faculty of Computing,

Sri Lanka Institute of Information Technology.

TABLE OF CONTENTS

DECLARATION	i
ABSTRACT.....	ii
ACKNOWLEDGEMENT	iii
LIST OF FIGURES.....	vi
LIST OF TABLES	viii
LIST OF ABBREVIATIONS	ix
1. INTRODUCTION.....	1
1.1 Background.....	1
1.1.1 Component Overview	3
1.2 Literature Review	5
1.3 Research Gap.....	10
2. RESEARCH PROBLEM.....	13
3. RESEARCH OBJECTIVES	15
3.1. Main Objectives.....	15
3.2. Specific Objectives	15
3.2.1. Extract Module keywords and phrases from module outlines	15
3.2.2. Extracting Academic Transcripts and Skill Area Categorization	15
3.2.3. Graphical representation of skill areas	15
4. METHODOLOGY.....	16
4.1. Overview	16
4.1.1. System Diagram	17
4.1.2. Tools and Technologies	18
4.1.3. Requirements.....	21
4.2. Implementation and Testing	22
4.2.1. Implementation	22

4.2.1.2. Extracting the data from academic transcripts	25
4.2.1.4. Skill area categorization and graphical representation of Skill Areas	31
4.2.1.5. Academic Transcript Score Prediction Model	44
4.2.2. Testing	50
4.2.2.1. Skill area Categorization	50
4.2.2.2. Score based on skill proficiency	53
4.3. COMMERCIALIZATION	56
5. RESULTS & DISCUSSION	58
5.1. Results	58
5.1.1. Extracting module keywords and phrases using n-grams	58
5.1.2. Extracting data from academic transcripts	58
5.1.3. Name entity recognition model	59
5.1.4. Model Building for Skill Area Categorization	60
5.1.5. Graphical representation of skill areas	62
5.1.6. Academic Transcript Score Prediction Model	66
5.2. Research Findings and Discussion	67
5.2.1. Research Findings	67
5.2.2. Discussion	69
6. CONCLUSION AND FUTURE WORK	71
7. REFERENCES	73
8. APPENDICES	75
8.1. Appendix 1 – Survey Questionnaire	75
8.2. Appendix 2 – Turnitin Report	76

LIST OF FIGURES

Figure 1: Survey result of importance of analyzing an academic transcript in a recruitment system.	3
Figure 2: Survey result of the importance of identifying courses or programs that are relevant to the position that a candidate has applied.....	4
Figure 3: High-level system diagram of analyzing academic transcript component.....	17
Figure 4: Extracting module keywords and phrases from module outlines overview diagram....	22
Figure 5: Module descriptions Dataset	22
Figure 6: First 5 rows of the dataset.....	23
Figure 7: Datatypes of the dataset.....	23
Figure 8: Dataset pre-processing	23
Figure 9:Preprocessed Dataset.....	24
Figure 10: Academic Transcript Extraction Overview Diagram	25
Figure 11:Academic transcript pre-processing and extracting data.....	25
Figure 12: Name entity recognition model overview diagram	26
Figure 13:Dataset for custom NER model.....	27
Figure 14: Dataset preprocessing for custom NER model.....	28
Figure 15:Conversion of objects to JSON format.....	28
Figure 16: Training the custom NER model	29
Figure 17: Dataset annotations.....	29
Figure 18: Training the Model	30
Figure 19: docBin object to save the model.....	30
Figure 20: Skill area categorization and graphical representation of skill areas overview diagram	31
Figure 21:Survey results of identifying the high-level skill areas	32
Figure 22: Dataset for skill area prediction.....	33
Figure 23: Visualization of skill area prediction dataset.....	34
Figure 24: EDA of skill area prediction dataset.....	34
Figure 25: Logistic regression algorithm for skill area prediction	36
Figure 26: Naive Bayes algorithm for skill area prediction.....	37
Figure 27: XGBoost algorithm for skill area prediction.....	38
Figure 28: Random Forest algorithm for skill area prediction	39
Figure 29: SVM algorithm for skill area prediction	40
Figure 30: Fine-tuned SVM model	41
Figure 31: Skill area prediction.....	43
Figure 32: Skill area categorization	43
Figure 33: Academic Transcript Score Prediction Model Overview Diagram	44
Figure 34:Dataset for academic transcript score prediction model	44
Figure 35: Drop duplicates and full missing values.....	45
Figure 36: Job role conversion to numeric using One hot encoder	45
Figure 37: Job description conversion to numeric using Countvectorizer	45
Figure 38: Splitting into training and testing sets	46
Figure 39: Random Forest Regressor model to predict score.....	46

Figure 40: Linear Regressor model to predict score	47
Figure 41: Support Vector Regressor model to predict score	47
Figure 42: Gradient Boost Regressor model to predict score	48
Figure 43: Academic Transcript user interface	49
Figure 44: Extracted module keywords and phrases from module outline descriptions	58
Figure 45: Extracted text from an academic transcript using OCR	58
Figure 46: Custom NER losses	59
Figure 47: Tested output of custom NER.....	59
Figure 48: Pre-trained NER model losses and score	59
Figure 49: Tested output of pre-trained NER.....	59
Figure 50: Confusion matrix of the new SVM model	60
Figure 51: Classification report of the new SVM model.....	61
Figure 52: Bar chart representation of skill areas	62
Figure 53: Pie chart representation of skill areas.....	62
Figure 54: Radar graph representation of skill areas	63
Figure 55: Score based on the skill proficiency and the position that a candidate has applied	66

LIST OF TABLES

Table 1: Comparison between existing studies and the proposed system	12
Table 2: Model accuracies of score prediction model.....	60
Table 3: Mean square errors of academic transcript score prediction model.....	66

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
CV	Curriculum Vitae
PDF	Portable Document Format
NER	Name Entity Recognition
SVM	Support Vector Machine
TF-IDF	Term Frequency - Inverse Document Frequency
TSRM	Technical Skills Recognition Model
SMOTE	Synthetic Minority Oversampling Technique
RE	Regular Expressions
OCR	Optical Character Recognition
NLTK	Natural Language Tool Kit
IT	Information Technology

1. INTRODUCTION

1.1 Background

The recruitment process in Sri Lanka traditionally involves manual steps like reviewing resumes, shortlisting candidates, conducting interviews, and conducting background checks. Despite significant technological advancements in the country, many companies still rely on these manual recruitment methods.

[1]As of 2019, the Information Technology and Business Process Management (IT-BPM) industry in Sri Lanka employed over 113,000 individuals and generated a substantial revenue of US\$ 1.5 billion. The primary goal of this industry was to create more opportunities, particularly for young talent, in its knowledge-based workforce. This endeavor not only aligned with the country's ambitions to secure a larger share of global IT spending but also aimed to achieve an impressive US\$ 5 billion in export revenues by the year 2025. Moreover, a thriving IT-BPM industry was expected to yield positive ripple effects, with each high-skilled job in this sector potentially generating up to 2.5 additional jobs within local industries.

Given the significant growth and demand in the IT-BPM sector, the IT industry is rapidly changing, they have a high demand for new talent. Traditional recruitment process which manually screens candidates can be time and labor-consuming and resource-intensive intensive which makes it challenging for businesses to keep pace with the industry's rapid growth. Furthermore, it can be challenging to select the best qualified individuals for particular roles due to the absence of conventional procedures for evaluating crucial components such as technical skills and personality traits. To address these challenges and facilitate more successful and efficient hiring methods, a more streamlined and systematic approach to recruitment is required. Such a system could efficiently match the industry's expanding employment needs with a diverse pool of potential candidates, streamlining the hiring process and ensuring that the right skills are readily available to support the

industry's growth trajectory. Additionally, an automated recruitment system can aid in identifying and attracting the next generation of talent vital for sustaining and advancing the IT-BPM sector's contributions to Sri Lanka's economic development and global competitiveness.

[2]According to a recent study by ICTA, the IT industry in Sri Lanka is growing at an annual rate of 16% and is expected to create over 200,000 direct and indirect job opportunities by 2022. This growth is attributed to the country's highly skilled IT workforce, which has received top-notch education and training and is competitive in the global IT market. Finding qualified individuals for these positions and matching the demand for IT jobs, however, continue to be difficult tasks. Despite the abundance of talent, there is a shortage of competent workers, which fuels intense rivalry for the best employees. A mismatch between employers' expectations and candidates' talents may make it difficult for businesses to find the best employees.

Therefore, in order to keep up with the industry's expansion and efficiently match job demand with skilled candidates, a more streamlined and automated system will not only help companies find the right employees but also contribute to the country's economic growth and competitiveness in the global IT market. The growth of the IT industry in Sri Lanka highlights the importance of modernizing recruitment practices to bridge the gap between job seekers and employers, ensuring a brighter future for both.

1.1.1 Component Overview

The research component, analyzing academic transcripts, specifically targets a particular university and aims to extract and analyze the information from the academic transcripts which will be used to evaluate the suitability of candidates for specific job roles.

An academic transcript is a crucial document that provides a comprehensive record of a student's educational journey at an institution of higher learning. It serves as an official and detailed account of the courses taken, grades earned, and credits achieved. By analyzing this, we can identify patterns and trends in a candidate's performance and provide valuable insights into their strengths and weaknesses.

According to the survey conducted among IT employees, it is clear that the majority of them consider that a recruitment system should analyze academic transcripts during the recruitment process as important.

4. How important do you think it is for a recruitment system to analyze academic transcripts during the recruitment process?
112 responses

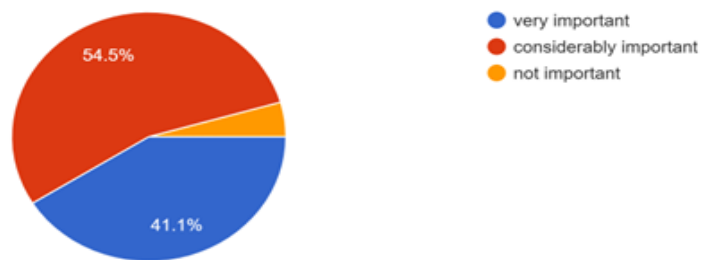


Figure 1: Survey result of importance of analyzing an academic transcript in a recruitment system.

The component focuses on identifying the skill areas of a candidate which helps the organization to make informed hiring decisions by providing a thorough and accurate

picture of the candidate's academic background. Thus, by understanding the candidate's skills and knowledge, the organization finds the right match for their job roles, which increases the chance of a successful hire. This, in turn, will lead to higher productivity and efficiency, which will be beneficial for the organization.

5. How important do you think it is for a recruitment system to be able to identify courses or programs that are particularly relevant to the position being recruited for?

112 responses

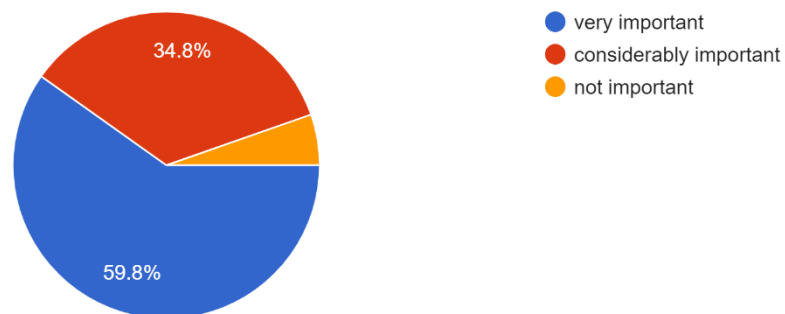


Figure 2: Survey result of the importance of identifying courses or programs that are relevant to the position that a candidate has applied.

According to the above survey results, it is clear that majority of the of the of the employees in the IT industry consider that a recruitment system should identify courses or programs that are particularly relevant to the position being recruited as highly important.

Overall, this academic transcript analysis component is an essential part of an online recruitment system. It helps organizations to simplify the hiring process, ensure the accuracy and relevance of the analysis, and find the right fit for their job openings. It provides a competitive edge for the organization by increasing the likelihood of finding the right candidate and simplifying the hiring process, especially in IT companies in Sri Lanka.

1.2 Literature Review

The success and growth of an organization depend on hiring the right employees. The typical recruiting strategy, which relies on manual tasks like reviewing academic records and resumes and evaluating technical and professional abilities, is time-consuming and inefficient. It is crucial to use a reliable and efficient method to evaluate candidates' skills and talents in order to satisfy company criteria. Some solutions have been proposed and have been implemented to automate the recruitment process. The following are some of the solutions that have been proposed.

Academic transcripts have been extremely important when hiring a candidate for a certain job position. For the problems faced when analyzing an academic transcript in a manual recruitment process, some solutions have been proposed and have been implemented. Authors have developed systems based on analyzing the academic transcript when hiring a candidate. The following are some research done to address those problems.

[3]According to the study done by a group of students in 2020 which focuses on candidate selection for an interview for the position of Software engineer, this research conducts a resume analysis and a personality prediction, and the researchers examined applicants' GitHub accounts using text mining tools and discovered that the number of repositories, commits, and followers were strong indicators of job performance. Apart from that they analyzed the academic transcripts identified the modules and categorized them into technical skills and soft skills and then the candidate with the most technical skills will be selected and their respective grade for each module will be compared. The candidate who has grades above C+ will be selected and others will be rejected for the job role. One of the critical aspects of this pre-screening process, as highlighted in recent research, is the analysis of academic transcripts. Academic transcripts provide valuable information about a candidate's educational background, including their performance in relevant courses. Analyzing this information can help identify a candidate's knowledge, skills, and capabilities related to the job.

Here, we will look into the methodology of academic transcript analysis. It focuses on extracting essential insights into a candidate's educational history.

Data Extraction: This step involves extracting data from the academic transcript. An academic transcript typically contains information about the candidate's courses, grades, and possibly other relevant details.

Text Pre-processing: To prepare the transcript data for analysis, text pre-processing is performed. This step includes cleaning the text by removing unnecessary characters, punctuation marks, tokenization, stop-word removal, and converting the text to lowercase.

Feature Extraction: This step involves converting the pre-processed text into a structured format that machine learning algorithms can work with. Techniques like TF-IDF vectorization are used to represent text data as numerical features.

Machine Learning Classification: Once the features are extracted, various supervised machine learning algorithms are employed to classify and evaluate the candidate's academic performance. Algorithms such as Naïve Bayes, Logistic Regression, and Support Vector Machines (SVM) are used to determine the knowledge level and skills of each candidate.

Final Evaluation: The results obtained from the academic transcript analysis contribute to the overall assessment of a candidate's suitability for the Software Engineer role. Candidates with strong academic backgrounds relevant to the job requirements are given higher consideration.

In addition to academic transcript analysis, the proposed solution also encompasses the analysis of other candidate data sources, such as CVs, GitHub profiles, LinkedIn profiles, phone call transcripts, and recommendation letters. Each of these components provides unique insights into a candidate's qualifications and personality traits.

[4]The research paper titled "Analyzing Competences in Software Testing: Combining Thematic Analysis with Natural Language Processing (NLP)" addresses the lack of

analysis of competencies required in software testing education for fresh graduates in computer science. The authors use NLP techniques to analyze job descriptions and course syllabi to identify the competencies required in software testing. The research methodology employed in this study combines both qualitative and data-driven approaches. The primary data sources include job advertisements from prominent job portals in the USA and Canada and course descriptions from 20 leading academic institutions where software testing is taught. The analysis is conducted using Natural Language Processing (NLP) techniques and thematic analysis to extract relevant keywords and identify competencies. A central focus of this study is the examination of course syllabi from academic institutions. These syllabi provide insights into the competencies covered in software testing courses. The goal is to determine whether these courses align with the competencies demanded by employers in the industry. This analysis helps evaluate the effectiveness of current academic offerings in preparing students for careers in software testing.

[5]The research paper titled “DevFlair: Automated Pre-screening of Software Engineering Candidates using NLP and ML.” proposes a system that automates the pre-screening process of a software engineer candidate. Traditionally, evaluating job candidates relied heavily on manual review of resumes and CVs, a time-consuming and potentially biased process. The introduction of automated pre-screening processes marked a significant shift in this paradigm. These processes leverage technology to analyze a candidate's qualifications, skills, and even personality traits efficiently. One crucial aspect of candidate evaluation is assessing their personality traits. Personality plays a vital role in determining how well an individual fits into a team and the work environment. Several research papers, including "DevFlair," have tackled the challenge of automating personality prediction. They use techniques like NLP to analyze social media profiles and predict personality traits such as Openness, Conscientiousness, Extraversion, Agreeableness, and Neuroticism (commonly known as the Big Five traits). This automated approach provides a deeper understanding of a candidate beyond what's

mentioned in their CV. One challenge in predicting personality traits is acquiring the necessary data. LinkedIn, a professional networking platform, contains valuable information about users, including their career history and personal summaries. However, LinkedIn's terms of service prohibit data scraping, making it difficult to collect datasets for research. To address this limitation, the researchers turned to Facebook data, specifically users' status updates, which were annotated with gold-standard personality labels. While Facebook data may not perfectly reflect professional aspects, it provides linguistic cues that correlate with personality traits. The heart of automated pre-screening lies in machine learning algorithms. Researchers employ various ML models, such as Logistic Regression, Support Vector Machines (SVM), and FastText, to predict personality traits. These models learn patterns from the data and make predictions based on linguistic features extracted from text. The researchers notably use FastText, a library developed by Meta's AI Research Team, which outperforms other models in predicting certain personality traits. In addition to personality prediction, assessing technical skills is vital for positions like Software Engineering. The researchers introduce a Technical Skills Recognition Model (TSRM) that employs a questionnaire to evaluate candidates' proficiency in specific technical skills like Java, JavaScript, and Python. This approach ensures a more accurate assessment of a candidate's technical abilities, going beyond what's typically found in a CV. While this research focuses on Software Engineering candidates, the framework's principles can extend to other job roles and domains. Future research could involve customizing the framework for diverse positions and incorporating domain-specific expertise. Furthermore, expanding the analysis to include data from additional platforms like GitLab and Stack Overflow would provide a more comprehensive candidate evaluation.

In the research paper titled [6]Machine Learning Approach for predicting career suitability, Career Progression, and Attrition of IT Graduates, the authors explored the field of Information Technology (IT) career development, focusing on Sri Lanka's IT industry. The primary aim was to help both fresh graduates entering the IT workforce and

existing IT employees in achieving their career goals while addressing some critical challenges in the industry. One of the main challenges identified was that IT graduates often struggle to find the most suitable career path and skills alignment upon entering the industry. Additionally, IT employees face obstacles in career progression, leading to potential attrition issues for employers. To tackle these challenges, the researchers collected data from IT professionals in Sri Lanka through surveys. They used various machine learning classification algorithms, including XGBoost, Random Forest, Support Vector Machine, K-Nearest Neighbors, Decision Tree, and Naive Bayes, to build predictive models for career suitability, initial salary, career and salary progression, professional course recommendations, employee attrition, and remedial actions. The models' performance was evaluated using accuracy, precision, recall, and F1-Score metrics. Notably, XGBoost emerged as the top-performing algorithm for several predictions, demonstrating the potential of machine learning in addressing career-related challenges in the IT industry. This research approach effectively addressed the pressing issues in the IT sector. For fresh graduates, the study provided a valuable career mentoring system that predicts suitable job roles and initial salaries based on their skills and qualifications, offering them guidance and direction in a competitive job market. Additionally, existing IT employees benefited from predictions related to career progression, salary advancement, and professional course recommendations, aiding them in achieving their career goals. Furthermore, the study addressed employers' concerns about attrition by predicting employee attrition and suggesting remedial actions to retain valuable staff. The comprehensive methodology involved data collection, preprocessing, model training, evaluation, and the development of a web-based application to deliver these predictions. This research not only contributes to the IT industry in Sri Lanka but also demonstrates the power of data-driven solutions in overcoming career-related challenges, making it a valuable addition to the field of career development and predictive analytics.

1.3 Research Gap

The research component presented in this study aims to address a significant research gap that distinguishes it from the four previously discussed research papers. conducting bridges several critical gaps in the existing literature, offering a comprehensive and unique approach to analyzing academic transcripts for IT job roles. Firstly, while the previous researchers primarily focused on the software engineering field, my research extends its scope to encompass all IT-related job roles. This broader perspective allows for a more inclusive and adaptable approach to candidate selection, addressing the diverse skill sets and qualifications required in the IT industry. By examining academic transcripts for various IT positions, this research component aims to develop a robust framework that can be applied across multiple job categories, benefiting both employers and job seekers.

The existing literature survey valuable insights into the use of machine learning algorithms and data-driven approaches to address career-related challenges in the Information Technology (IT) industry, particularly in Sri Lanka. While the surveyed research offers significant contributions in predicting career suitability, salary ranges, career progression, professional course recommendations, and employee attrition, it primarily focuses on general aspects of the IT workforce, without delving into the specifics of individual job roles. Furthermore, the literature does not specifically explore the comprehensive analysis of academic transcripts and their alignment with high-level technical skill areas, a key component in my research. This is a crucial gap, as understanding the direct correlation between academic qualifications and specific IT job roles can provide a more personalized and targeted approach to career development.

Secondly, this research component seeks to deep into the academic transcript analysis by specifically mapping course descriptions and module outlines from a particular university. This level of detail enables the identification of high-level technical skill areas that are being covered in the curriculum. By connecting these skills with the requirements of the industry, this research component aims to provide a clearer understanding of how well

academic programs align with real-world job demands. This mapping process offers a valuable tool for universities to enhance their course offerings and tailor their programs to better prepare students for their future careers in the IT sector.

Furthermore, this research introduces a novel aspect by incorporating graphical representation and skill proficiency scoring derived from academic transcripts. This graphical representation not only makes the data more accessible but also allows for quick visual comparisons between candidates. Moreover, the introduction of skill proficiency scores provides a quantitative measure of a candidate's preparedness for a particular job role. This innovative approach offers employers a more efficient and objective means of evaluating candidates, enhancing the overall hiring process. In summary, my research strives to fill these gaps by expanding the scope of academic transcript analysis, emphasizing the alignment of educational programs with industry needs, and introducing visual representations and skill proficiency scores to facilitate improved decision-making in the hiring process for IT job roles.

This research aims to bridge this gap by focusing on a more granular analysis of academic transcripts, specifically tailoring the study to the IT industry, including job roles beyond just software engineers. By examining and categorizing academic transcripts based on high-level technical skill areas, as derived from the module outlines or course syllabi of a specific university, my research seeks to provide a comprehensive and detailed understanding of the skill proficiencies gained through academic education. Furthermore, my study aims to graphically represent these skill areas, allowing for a visual and intuitive interpretation of the academic curriculum's alignment with real-world job requirements. Lastly, this research component introduces a scoring mechanism to assess skill proficiency, offering a quantifiable measure of an individual's readiness for specific IT job roles. In summary, the research gap lies in the absence of a tailored and fine-grained analysis of academic transcripts, which my research aims to fill by providing a more customized and skill-focused approach to career development in the IT sector.

Table 1: Comparison between existing studies and the proposed system

Research	Based In Sri Lanka	For all the job roles IT industry	Analyzing the academic transcript	Considering the module outlines	Graphical representation of expertise areas of a candidate	Score based on skill proficiency and job role
Research [3]	✓	✗	✓	✗	✗	✗
Research [4]	✗	✗	✗	✓	✗	✗
Research [5]	✗	✗	✗	✗	✗	✗
Research [6]	✓	✗	✗	✗	✗	✗
Proposed Component	✓	✓	✓	✓	✓	✓

2. RESEARCH PROBLEM

In today's fast-paced world, landing a job in the field of information technology (IT) is highly competitive. Employers want to make sure they hire the best candidates and job seekers want to stand out. However, the current methods of selecting candidates often rely on traditional resumes and interviews, which may not always provide a complete picture of a person's skills and qualifications. Thus, this research problem the aim is to address how to improve the process of candidate selection for IT jobs. The industry needs means and ways to understand what skills candidates have and how well they match the job requirements. Therefore, it is vital to address these problems because it can help both employers and job seekers make more informed decisions.

One of the key challenges in this research is how to effectively analyze academic transcripts. These transcripts contain a wealth of information about a person's educational history, including the courses they've taken and the grades they've earned. However, the problem is that academic transcripts are often lengthy and full of complex information. Thus, it is vital to figure out how to extract the most important insights from these transcripts, such as identifying technical skills. Solving this problem will help us create a fair and reliable method for assessing a candidate's qualifications based on their academic record.

Once the challenge of analyzing academic transcripts is tackled, the next research problem is how to categorize and visualize the skills that were identified. A system should be created that not only recognizes the skills but also organizes them into clear categories. For instance, we need to differentiate between technical skills like programming skills and project management skills. Furthermore, we want to represent this data visually, using

graphs and charts that are easy to understand. This is crucial because it will make it simpler for employers to see at a glance what a candidate brings to the table and how their skills compare to job requirements.

The fourth research problem that needs to be addressed is the need to go beyond the software engineering field. Previous research has mainly focused on this specific job category, but the IT industry offers a wide range of roles, from cybersecurity experts to data analysts. We aim to create a framework that can analyze academic transcripts for all these different IT job roles. Each role may require a unique set of skills, and we want to ensure our research applies to them all. By doing this, we can help job seekers find the right opportunities and assist employers in identifying the most suitable candidates for various IT positions.

The final research problem involves developing a scoring system for skill proficiency based on academic transcripts. Once the skills are identified and categorized, there should be a way to quantify how proficient a candidate is in each skill. This scoring system will be crucial for employers to make informed decisions about candidate suitability. However, creating a fair and accurate scoring system is a complex task. Certain factors must be considered such as the difficulty of courses, the grades obtained, and the relevance of each skill to the job. By addressing this research problem, we can provide a valuable tool that simplifies the hiring process and ensures a more equitable evaluation of candidates' skills and qualifications.

In conclusion, these five research problems form the core of our study, aiming to enhance the candidate selection process for IT jobs by analyzing academic transcripts comprehensively, categorizing and visualizing skills, extending the framework beyond

software engineering, and developing a skill proficiency scoring system. These solutions will benefit both employers and job seekers in the dynamic and competitive IT job market.

3. RESEARCH OBJECTIVES

3.1. Main Objectives

The fundamental objective is to analyze academic transcripts as a means to gain a comprehensive understanding of the specific skills and knowledge that candidates have acquired throughout their degree program.

3.2. Specific Objectives

3.2.1. Extract Module keywords and phrases from module outlines

This objective aims to extract the module keywords and phrases from module outlines for a specific university. These extracted keywords and phrases are used to combine with the modules taken from the academic transcript and categorize the skill areas.

3.2.2. Extracting Academic Transcripts and Skill Area Categorization

The goal of this objective is to extract data from a transcript of a candidate. The purpose is to identify skill areas. This categorization process is crucial for our research as it allows us to evaluate the candidates' skills and proficiency levels in each area. Additionally, the aim is to create a scoring system that considers the candidate's skills, proficiency, and the specific requirements of a job position. This score will provide a measure of the candidate's suitability for the role serving as a tool for employers.

3.2.3. Graphical representation of skill areas

Once the skill areas are categorized, it is visually represented using graphs and charts. These visual representations are carefully designed to help employers easily understand and evaluate a candidate's skill level.

By leveraging these graphical representations, the aim is to present the candidate's skills in an appealing way that allows employers to quickly assess the breadth and depth of their

abilities. This visual representation serves as a tool for employers, in making informed decisions during the hiring process helping them match a candidate's skills with the specific requirements of a job role.

4. METHODOLOGY

4.1. Overview

This research component bridges the gap between academic qualifications and professional suitability. Firstly, module keywords and phrases are systematically extracted from module outlines at a specific university using n-grams, providing a comprehensive foundation for data analysis. Next, data is extracted from academic transcripts via Optical Character Recognition (OCR) techniques, allowing for the identification of module names and associated grades using a custom Named Entity Recognition (NER) model. Following this, skill areas are categorized utilizing machine learning models, offering a nuanced understanding of a candidate's skill profile. Further enhancing this assessment, a proficiency-based score is computed, considering both the categorized skills and specific job role requirements. Finally, these skill areas are visually represented through graphical visualization techniques, providing employers with an intuitive means to understand a candidate's suitability. This comprehensive framework not only streamlines the candidate evaluation process but also promotes data-driven decision-making in academic and employment contexts, ultimately contributing to more efficient talent acquisition and placement procedures.

4.1.1. System Diagram

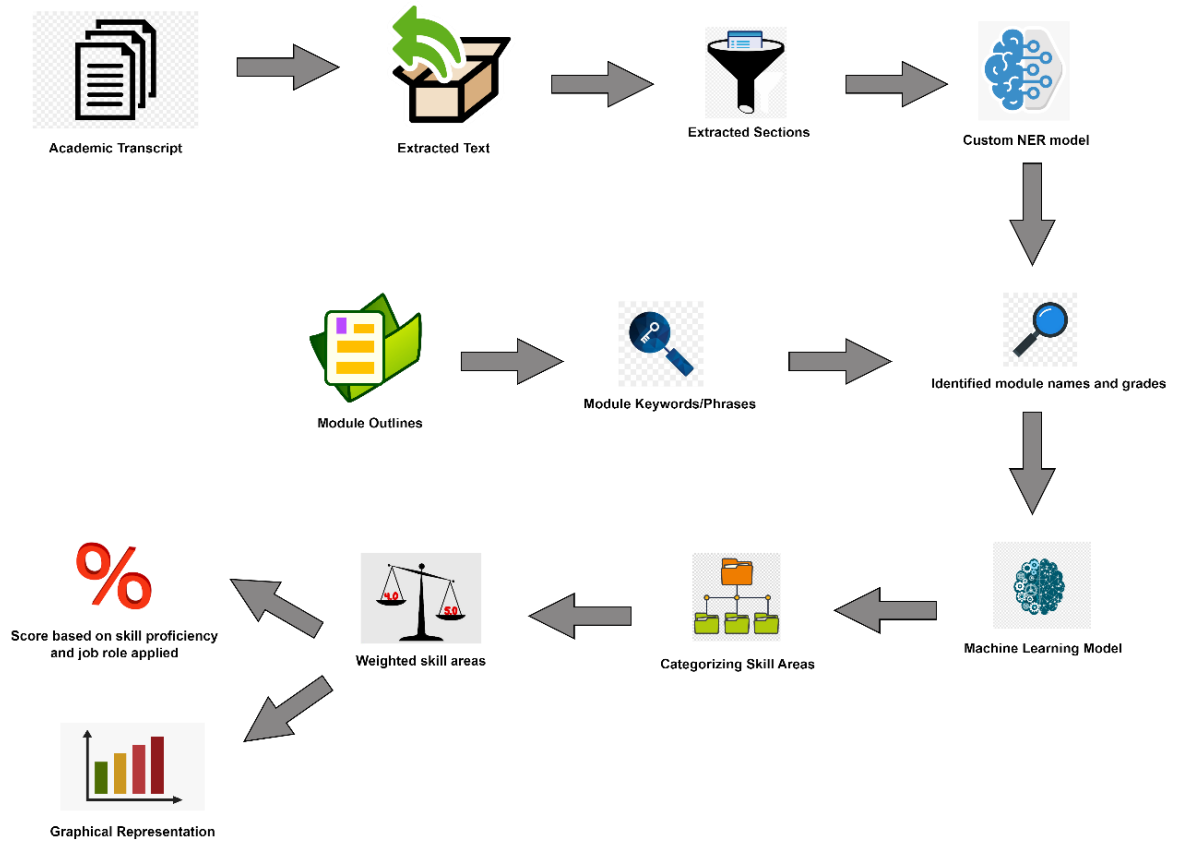


Figure 3: High level system diagram of analyzing academic transcript component.

4.1.2. Tools and Technologies

Python was mainly used for the development of this system since Python provides a variety of libraries which is helpful to achieve the set target.

4.1.2.1 Libraries used

NLTK - NLTK (Natural Language Toolkit) is a comprehensive library for natural language processing tasks. It provides tools for tokenization, stemming, lemmatization, and more.

OCR pytesseract - PyTesseract is a Python wrapper for Google's Tesseract OCR (Optical Character Recognition) engine. It is used for extracting text from images in your code.

Poppler - Poppler is a library for rendering and working with PDF files. It provides a set of tools and libraries for parsing, rendering, and manipulating PDF documents.

Spacy - Spacy is a popular natural language processing library in Python. It provides tools for text processing, including tokenization, part-of-speech tagging, named entity recognition, and more.

Numpy - NumPy is a fundamental library for numerical computing in Python. It offers support for arrays and matrices, along with mathematical functions to operate on them efficiently.

Pandas - The re module is Python's built-in regular expression library. It allows you to work with regular expressions for pattern matching and text manipulation.

Matplotlib - Matplotlib is a versatile library for creating static, animated, and interactive visualizations in Python.

Scikit-learn - A comprehensive machine learning library in Python that provides tools for various machine learning tasks such as classification, regression, clustering, and more.

Gensim - Gensim is a library for topic modeling and document similarity analysis. It includes Word2Vec and other models for word embeddings.

Glob - The glob module is used for file path pattern matching and retrieval. It allows you to find files that match a specified pattern.

Re (Regular Expressions)- The re module is Python's regular expression library. It is used for pattern matching and text extraction based on regular expressions.

Swifter - Swifter is a library that optimizes the execution of certain operations on Pandas DataFrame by using parallel processing.

IDE: VS Code

VS Code, a Microsoft-developed source code editor, is a versatile and lightweight tool that is suitable for various programming tasks, including coding, debugging, and version control. Its key benefits are its extensibility, allowing users to add new functionalities through its vast library of extensions, and its clean and user-friendly interface. Additionally, it has built-in support for Git, making it easier for developers to manage version control from within the editor.

Model Implementation: Python using Google Colab

Python is a user-friendly programming language that is particularly useful for quickly developing models and natural language processing. Google Colab is a cloud-based platform that offers high-performance computing resources and already installed machine learning libraries, reducing the need for manual installation and ensuring compatibility with other frameworks.

Frontend: Flask

Flask is a Python web framework that enables the creation of a user interface for Python models, streamlining the interaction with them. By using Flask, a web application can be built that facilitates the input of data from users, which is then processed by the Python

model. Finally, the output can be displayed back to the user via the Flask interface. This provides an accessible and user-friendly way to utilize Python models in various applications such as artificial intelligence, data analysis, and machine learning.

Database: Azure Cosmos DB

Azure Cosmos DB is a database service designed for modern app development that supports both NoSQL and relational data models. It was chosen as the database solution for storing unstructured data obtained from various sources such as CVs, academic transcripts, and social media profiles.

Deployment: Azure App Service

4.1.3. Requirements

User Requirements

- The hiring manager should be able to input the academic transcripts of a candidate.
- The company should be able to see a graphically represented skill set of candidates.
- The company should be able to

Functional Requirements

- **Text extraction**- The system should be able to extract text accurately from academic transcript documents.
- **Skill area categorization** – Should be able to classify modules into different skill areas and analyze them.
- **Graphical representation** – should graphically represent the categorized skill areas.
- **Generate a score** – Should give a score based on the skill proficiency and job role.

Non-Functional Requirements

- **Reliability** - The system must be highly reliable without interruptions or breakdowns during the translation process.
- **Accuracy** - The system is intended to give reliable outputs since users rely on it for acquiring knowledge.
- **User Friendliness** – The system should be able to give users a simple environment in which to perform what they desire.
- **Performance** – The system should be able to work properly and provide quick and accurate results.
- **Compliance** – All applicable laws and rules should be followed by the system. The candidate's private and secure information should be protected.

4.2. Implementation and Testing

4.2.1. Implementation

4.2.1.1. Extracting module keywords and phrases from module outlines

In this section, methodology related to extracting module keywords and phrases will be discussed.

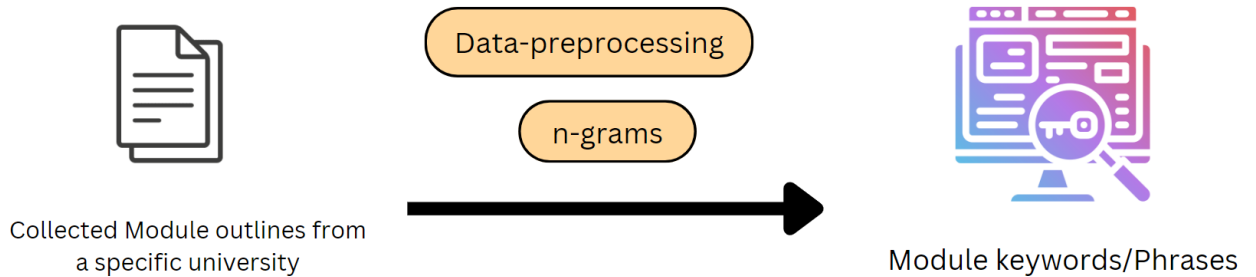


Figure 4: Extracting module keywords and phrases from module outlines overview diagram.

Dataset and Data Exploratory Analysis

Assuming that this research component is built for a specific university, the module outlines from a specific university were collected, and created a dummy dataset by using the module title, and module description.

A	B
1 Module Title	Module Description
2 MATHEMATICS FOR COMPUTING	Logic Control, Number Systems, Differentiation, Integration, Functions, Counting, Graph Theory, Matrices, Finite-State machines
3 Introduction to Programming	Introduction to fundamental programming concepts, specifically in the procedural programming paradigm. The topics include data types, control structures, functions, pointers, arrays, files, recursion and the mechanics of testing, and debugging. The students will also get a hands on experience to develop applications using the C language in LINUX operation system.
4 Communication Skills	Speech test, spot tests and a midterm examination. The final examination will be a comprehensive exam based on the topics covered during the semester.
5 Introduction to Computer Systems	Essentials of computer systems and computer networks in order to prepare students for advanced courses. It covers fundamentals of computer organization, combinational & sequential logic circuits, data communication and computer networks. By the end of the course, students will be able to explain how computer systems are organized, layered communication models, and the basic operation of computer networks.
6 English for Academic Purposes	Necessary English language skills and all the other academic practices that you require to pursue university studies
7 Object Oriented Concepts	Understand and apply the basic concepts of Object Oriented Programming, Design solutions by identifying the classes and relationships (Object Oriented Analysis and Design), Implement a solution to the given problem using the C++ Language
8 Software Process Modeling	The objective of this module is to introduce Software Engineering concepts. The topics discussed in this module include requirement specification, design, implementation, verification, validation, and evolution of software artifacts. Further, students will be able to appreciate how software engineering concepts help to produce and maintain software products and software-intensive systems.
9 Information Systems and Data Modeling	This is a mandatory module for the 1st year students in the B.Sc. Sp. (Hons) in Information Technology streams. The objective of this unit is to provide an overall understanding of Information Systems and Data Modeling. The students will mainly understand all the aspect of real world information systems including Business Process Modeling with their relationship to Data Modeling, which is introduced through relational database systems.
10 Internet and Web Technologies	Explain the concepts and technologies associated with the Internet and related applications. Identify the effective use of social media for organizations and individual users. Explain fundamentals of e-Commerce application domain along with security and privacy concerns and supportive web technologies. Apply modern markup languages and presentation technologies to design web interfaces. Implement web applications using client and server side scripting languages. Apply standards, UI design principles and best practices to enhance usability of a web application development. Explain the importance of web standards, digital content rights, usability and accessibility initiatives in web related applications.
11 Engineering Mathematics	This unit enables the student to understand the Engineering Mathematics with special emphasis on studying and applying mathematical techniques in engineering / IT systems. The objective of this unit is to provide the students with skills necessary to solve various mathematical problems encountered in IT and engineering - electrical/electronic/computer engineering in particular.
12 Network Fundamentals	The aim of this module is to cover essentials of data communications and computer network concepts to prepare students for advanced networking courses. It covers essentials of computer networks and the Internet technologies. By the end of the course, students will be able to build simple LANs, perform basic configurations for routers and switches, and implement IP addressing schemes.
Object Oriented Programming	Apply object oriented concepts in Java Language. Validate the knowledge in concurrency programming. Synthesize suitable set of design patterns for an application development. Justify usage of software

Figure 5: Module descriptions dataset

The dataset contained 157 records of module data, from 1st-year 1st-semester modules up to 4th-year 2nd-semester modules.

	Module Title	Module Description
0	MATHEMATICS FOR COMPUTING	Logic Control, Number Systems, Differentiation...
1	Introduction to Programming	Introduction to fundamental programming concept...
2	Communication Skills	Speech test, spot tests and a midterm examinat...
3	Introduction to Computer Systems	Essentials of computer systems and computer ne...
4	English for Academic Purposes	Necessary English language skills and all the ...

Figure 6: First 5 rows of the dataset

Module Title	object
Module Description	object
dtype: object	

Figure 7: Datatypes of the dataset

Data Pre-processing

In Natural Language Processing (NLP), a crucial step is cleaning up the text data, and we achieve this through data pre-processing. By using the NLTK library, stop words are removed, and any unwanted characters are taken out. Additionally, all words are changed to lowercase to ensure consistency throughout the text. This pre-processing is vital because it sets the stage for more effective NLP tasks and model training, by enhancing the overall robustness and accuracy of the research framework.

```
[ ] # Create a list to store the words in each row
words_per_row = []

# Iterate over the rows and split the module description into words
for desc in module_desc_column:
    if isinstance(desc, str):
        words = desc.lower().split() # Split the description into lowercase words

        # Remove stop words
        stop_words = set(stopwords.words("english"))
        words = [word for word in words if word not in stop_words]

        # Remove unwanted characters using regular expressions
        words = [re.sub(r"^[a-zA-Z0-9]", "", word) for word in words]

        # Remove empty strings
        words = [word for word in words if word]

        words_per_row.append(words) # Add the words to the list

# Print the lists of words per row
for words in words_per_row:
    print(words)
```

Figure 8: Dataset pre-processing

```
[
  'logic', 'control', 'number', 'systems', 'differentiation', 'integration', 'functions', 'counting', 'graph', 'theory', 'matrices', 'finitestate', 'machines',
  'introduction', 'fundamental', 'programming', 'concepts', 'specifically', 'procedural', 'programming', 'paradigm', 'topics', 'include', 'data', 'types', 'control', 'structures', 'functions', 'pointers', 'arrays', 'files', 'recursion',
  'speech', 'test', 'spot', 'tests', 'midterm', 'examination', 'final', 'examination', 'comprehensive', 'exam', 'based', 'topics', 'covered', 'semester',
  'essentials', 'computer', 'systems', 'computer', 'networks', 'order', 'prepare', 'students', 'advanced', 'courses', 'covers', 'fundamentals', 'computer', 'organization', 'combinational', 'sequential', 'logic', 'circuits', 'data',
  'necessary', 'english', 'language', 'skills', 'academic', 'practices', 'require', 'pursue', 'university', 'studies',
  'understand', 'apply', 'basic', 'concepts', 'object', 'oriented', 'programming', 'design', 'solutions', 'identifying', 'classes', 'relationships', 'object', 'oriented', 'analysis', 'design', 'implement', 'solution', 'given', 'prob',
  'objective', 'module', 'introduce', 'software', 'engineering', 'concepts', 'topics', 'discussed', 'module', 'include', 'requirement', 'specification', 'design', 'implementation', 'verification', 'validation', 'evolution', 'software',
  'mandatory', 'module', 'ist', 'year', 'students', 'bsc', 'sp', 'hons', 'information', 'technology', 'streams', 'objective', 'unit', 'provide', 'overall', 'understanding', 'information', 'systems', 'data', 'modeling', 'students', 'f',
  'explain', 'concepts', 'technologies', 'associated', 'internet', 'related', 'applications', 'identify', 'effective', 'use', 'social', 'media', 'organizations', 'individual', 'users', 'explain', 'fundamentals', 'e-commerce', 'applic',
  'unit', 'enables', 'student', 'understand', 'engineering', 'mathematics', 'special', 'emphasis', 'studying', 'applying', 'mathematical', 'techniques', 'engineering', 'systems', 'objective', 'unit', 'provide', 'students', 'skills',
  'aim', 'module', 'cover', 'essentials', 'data', 'communications', 'computer', 'network', 'concepts', 'prepare', 'students', 'advanced', 'networking', 'courses', 'covers', 'essentials', 'computer', 'networks', 'internet', 'technol',
  'apply', 'object', 'oriented', 'concepts', 'java', 'language', 'validate', 'knowledge', 'concurrency', 'programming', 'synthesize', 'suitable', 'set', 'design', 'patterns', 'application', 'development', 'justify', 'usage', 'softwa',
  'design', 'develop', 'maintain', 'database', 'cater', 'user', 'requirements', 'module', 'covers', 'conceptual', 'database', 'design', 'logical', 'database', 'design', 'schema', 'refinement', 'sql', 'database', 'programming', 'furt',
  'routing', 'switching', 'theory', 'configurations', 'top', 'ip', 'operation',
  'major', 'components', 'operating', 'systems', 'practice', 'utilities', 'unit', 'system', 'administration', 'students', 'also', 'apply', 'knowledge', 'learn', 'lectures', 'tutorial', 'labs', 'complete', 'units', 'programming', 'as',
  'object', 'oriented', 'analysis', 'design', 'using', 'unified', 'modeling', 'language', 'tools', 'object', 'oriented', 'software', 'engineering', 'applying', 'design', 'patterns', 'software', 'testing', 'supportive', 'processes',
  'essential', 'digital', 'electronic', 'concept', 'computer', 'networking', 'engineering', 'graduate', 'module', 'give', 'understanding', 'knowledge', 'signals', 'processing', 'keep', 'operation', 'computer', 'systems', 'network',
  'provide', 'understanding', 'theory', 'behind', 'descriptive', 'statistics', 'inferential', 'statistics', 'data', 'analysis', 'interpretation', 'outputs', 'obtained', 'using', 'statistical', 'software', 'basic', 'theory', 'covers',
  'understand', 'basic', 'analogue', 'electronic', 'engineering', 'concepts', 'objective', 'unit', 'provide', 'students', 'knowledge', 'necessa', 'understand', 'various', 'analog', 'electronic', 'devices', 'operate', 'provide', 'sk',
  'oracle', 'database', 'components', 'architecture', 'creating', 'controlling', 'database', 'installing', 'oracle', 'database', 'software', 'database', 'creation', 'managing', 'oracle', 'instanceconfiguring', 'oracle', 'network',
  'routing', 'concepts', 'static', 'routing', 'dynamic', 'routing', 'switched', 'networks', 'switch', 'configuration', 'vlans', 'access', 'control', 'lists', 'dhcp', 'nat', 'ipv4', 'device', 'discovery', 'management', 'maintenance',
  'design', 'develop', 'database', 'solutions', 'real', 'world', 'applications', 'use', 'relational', 'query', 'languages', 'database', 'programming', 'applications', 'evaluate', 'query', 'plans', 'recommen', 'solutions', 'speed',
  'introduction', 'cyber', 'security', 'security', 'controls', 'risk', 'management', 'malicious', 'software', 'user', 'authentication', 'access', 'control', 'introduction', 'cryptography', 'cryptographic', 'hashes', 'symmetric', 'cr',
  'operating', 'systems', 'structure', 'process', 'management', 'cpu', 'scheduling', 'basic', 'concepts', 'process', 'synchronization', 'background', 'memory', 'management', 'deadlocks', 'real', 'time', 'systems', 'system', 'charact',
  'structured', 'analysis', 'design', 'information', 'systems', 'students', 'learn', 'analyze', 'business', 'problems', 'design', 'information', 'systems', 'structured', 'approach', 'students', 'also', 'gain', 'hands-on', 'experience',
  'module', 'provides', 'students', 'knowledge', 'problem', 'solving', 'difficult', 'often', 'ineffective', 'well', 'practical', 'tools', 'analyzing', 'effectively', 'solving', 'complex', 'problems', 'future', 'business', 'analysts',
  'provides', 'introduction', 'network', 'design', 'management', 'field', 'network', 'covers', 'fundamental', 'concepts', 'theory', 'evolution', 'applications', 'recent', 'technological', 'advancements',
  'module', 'covers', 'fundamental', 'concepts', 'mobile', 'application', 'development', 'applied', 'using', 'mobile', 'technologies', 'module', 'introduces', 'students', 'different', 'features', 'mobile', 'applications', 'platform',
  'module', 'introduces', 'students', 'fundamental', 'data', 'structures', 'stacks', 'queues', 'linked', 'lists', 'trees', 'addition', 'offers', 'depth', 'coverage', 'different', 'algorithms', 'techniques', 'designing', 'algorithm',
  'final', 'presentation', 'viva', 'final', 'report', 'writing', 'system', 'testing', 'activity', 'git', 'report', 'progress', 'evaluation', 'informal', 'progress', 'evaluation', 'er', 'diagram', 'activity', 'interface', 'designing',
  'introductory', 'course', 'discrete', 'mathematics', 'goal', 'course', 'introduce', 'students', 'ideas', 'techniques', 'discrete', 'mathematics', 'widely', 'used', 'science', 'engineering', 'course', 'teaches', 'students', 'techni',
  'course', 'survey', 'computer', 'algorithms', 'examines', 'fundamental', 'techniques', 'algorithms', 'design', 'analysis', 'develops', 'problemsolving', 'skills', 'required', 'programs', 'study', 'involving', 'computer', 'science',
  'course', 'covers', 'fundamentals', 'technologies', 'data', 'communications', 'related', 'network', 'infrastructure', 'business', 'environment', 'topics', 'include', 'business', 'imperatives', 'distributed', 'systems', 'systems',
  'module', 'aim', 'students', 'gain', 'practical', 'experience', 'developing', 'managing', 'achieving', 'objectives', 'it', 'industry', 'based', 'project', 'involves', 'identifying', 'project', 'goals', 'gathering', 'requirements',
  'activity', 'diagram', 'object', 'diagram', 'state', 'diagram', 'class', 'diagram',
  'introduction', 'lan', 'concepts', 'ipv6', 'lan', 'design', 'scaling', 'vlans', 'spanning', 'tree', 'protocol', 'stp', 'etherchannel', 'hsrp', 'dynamic', 'routing', 'single', 'area', 'ospf', 'eigrp',
  'understanding', 'system', 'network', 'administration', 'apply', 'theory', 'practice', 'using', 'command', 'line', 'interfaces', 'understand', 'complex', 'ideas', 'relate', 'specific', 'problems', 'questions', 'understanding', 'ta',
  'module', 'introduce', 'realtime', 'operating', 'systems', 'basics', 'objectives', 'unit', 'educate', 'students', 'fundamentals', 'operating', 'systems', 'scheduling', 'mechanisms', 'processor', 'reserves', 'capabilities', 'cover',
  'module', 'designed', 'introductory', 'course', 'embedded', 'system', 'design', 'students', 'following', 'unit', 'learn', 'requirement', 'analysis', 'embedded', 'systems', 'design', 'implement', 'closedloop', 'automatic', 'control',
  'introduction', 'management', 'introduction', 'business', 'analysis', 'introduction', 'project', 'management', 'planning', 'forecasting', 'team', 'building', 'project', 'initiation', 'selection', 'project', 'estimation', 'planning',

```

Figure 9:Preprocessed Dataset

Furthermore, WordNet lemmatizer is used which is a tool that simplifies words to their base forms. A list is created to hold these simplified words for each row of text. While going through the rows, the lemmatizer processes the words, converting them into their basic forms. This step ensures that words with similar meanings are treated consistently, enhancing the accuracy and clarity of our analysis. It's similar to simplifying complex math problems to better understand their solutions.

Keywords and phrase extraction using n-grams

Once the preprocessing was completed, the TF-IDF vectorizer was applied to these sentences to capture the importance of individual words. The most significant words for each sentence were identified and presented. Additionally, n-grams (word combinations) were extracted from the sentences using a similar TF-IDF approach, highlighting meaningful word sequences. These steps collectively improved the understanding and representation of the module description, aiding in the research's accuracy and comprehensibility.

4.2.1.2. Extracting the data from academic transcripts

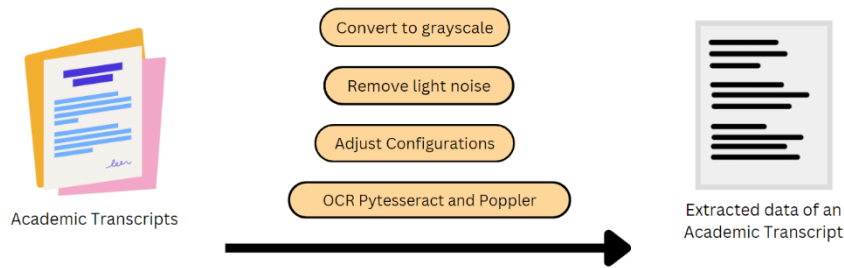


Figure 10: Academic Transcript extraction overview diagram

Dataset and Data Exploratory Analysis

Academic transcripts from graduates of a particular university who specialized in various IT fields were gathered. These transcripts serve as essential data sources, providing insights into the educational achievements and qualifications of individuals with diverse IT expertise.

Data Pre-processing and extracting data

When a file is uploaded, it is temporarily saved, and then optical character recognition (OCR) technology is applied to convert the content from PDF format into text format. This involves making adjustments to the image quality, converting it to grayscale, and eliminating any background noise in order to improve the accuracy of extracting the text. The extracted text is then stored in a text file, for analysis.

```
def extract_text():
    uploaded_file = request.files['file']

    if uploaded_file.filename != '':
        # Save the uploaded file temporarily
        uploaded_file.save('files/academic_transcript/uploaded_file.pdf')

    pdf_paths = glob.glob("files/academic_transcript/uploaded_file.pdf")
    text_file_path = "files/academic_transcript/extracted_text.txt"

    pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe' # Provide the path to your Tesseract executable

    with open(text_file_path, "w", encoding="utf-8") as text_file:
        for pdf_path in pdf_paths:
            pages = convert_from_path(pdf_path, dpi=300) # Adjust DPI as needed

            for pageNum, imBlob in enumerate(pages):
                im = Image.frombytes('RGB', imBlob.size, imBlob.tobytes()) # Corrected line
                im = im.convert('L') # Convert to grayscale
                im = im.point(lambda x: 0 if x < 200 else x) # Threshold to remove light noise
                text = pytesseract.image_to_string(im, lang='eng', config='--psm 6') # Adjust configuration
                text_file.write(text)
```

Figure 11: Academic transcript pre-processing and extracting data

4.2.1.3. Name entity recognition model to identify module names and grades

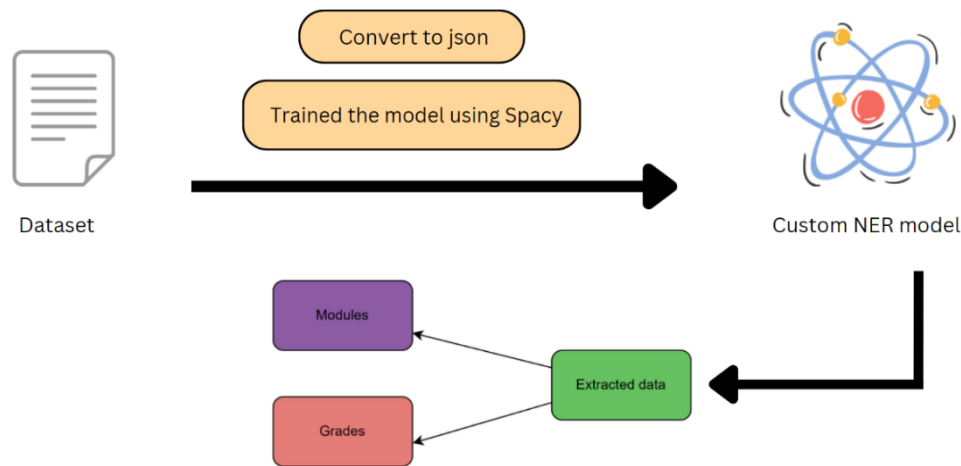


Figure 12: Name entity recognition model overview diagram

Named Entity Recognition (NER) models are a type of AI tool used in natural language processing. These models are designed to identify and categorize specific pieces of information within text data, such as names of people, places, organizations, dates, and more. NER models use various techniques, including deep learning, to analyze the context and structure of text and pinpoint these entities accurately. Deep learning, a subset of machine learning, involves training complex neural networks to make sense of patterns in data.

Custom NER model

Dataset and Data Exploratory Analysis

To build the custom Named Entity Recognition (NER) model, a dummy dataset was created. This dataset consisted of 8298 records, each containing specific columns such as "Code," "Module Title," "Semester," "Period," "Credits," and "Grade." These columns mimic the structure of academic transcripts, providing a foundation for training the NER model to recognize and extract essential information, such as module names and associated grades, from textual data resembling academic records.

	A	B	C	D	E	F
1	Code	Module Title	Semester	Period	Credits	Grade
2	EC1430	Data Communications & Computer Network	1	16-Apr	4	B+
3	EL1200	English Language Skills - I	1	16-Apr	4	B+
4	IT1000	Computer Fundamentals	1	16-Apr	4	A
5	IT1010	Introduction to Programming Environments	1	16-Apr	5	A+
6	MA140	Mathematics for Information Technology	1	16-Apr	4	A-
7	EL1210	English Language Skills- II	2	16-Apr	3	A
8	IT1020	Software Technology - I (Data Structures)	2	16-Apr	4	A+
9	IT1030	Database Management Systems - I	2	16-Apr	4	A+
10	IT1040	Internet Technology and Applications	2	16-Apr	4	A
11	MA1410	Foundations of Computer Science	2	16-Apr	4	A-
12	CG2000	Computer Graphics & Multimedia	1	16-Oct	4	B+
13	IT2000	Software Technology - II (OOP)	1	16-Oct	4	B+
14	IT2020	Database Management Systems - II	1	16-Oct	4	B+
15	IT2200	Software Engineering- I	1	16-Oct	4	B+
16	MA220	Probability & Statistics	1	16-Oct	4	A
17	EC2440	Data Communications & Computer Network	2	16-Oct	4	A
18	IT2010	Systems Programming and Design	2	16-Oct	4	A
19	IT2210	Software Engineering - II	2	16-Oct	4	B+
20	IT2220	Information Technology Project	2	16-Oct	4	A-
21	IT2400	Design and Analysis of Algorithms	2	16-Oct	4	A
22	IT3050*	Employability Skills Development - Seminar	1	16-Oct	1	B
23	SE3010	Software Engineering Process & Quality Man	1	16-Oct	4	B
24	SE3020	Distributed Systems	1	16-Jun	4	A
25	SE3030	Software Architecture	1	16-Jun	4	A-

Figure 13:Dataset for custom NER model

Data preprocessing for model training

In data preprocessing of the custom Named Entity Recognition (NER) model, a new Spacy model was loaded, and the custom NER component was integrated into the pipeline. Distinct entity labels, including MODULE_CODE, MODULE_TITLE, SEMESTER, PERIOD, CREDITS, and GRADE, were defined to represent the different elements in the academic transcripts. The dataset containing the relevant information was then imported, and text along with entity annotations were structured into a compatible format, typically in JSON (JavaScript Object Notation) objects, which served as a standardized way to represent the annotated data.

```

- load a new spacy model

[ ] nlp = spacy.blank("en") # load a new spacy model

- Load the blank English language model in spaCy

[ ] # Load the blank English language model in spaCy
nlp = spacy.blank("en")

- Create a new entity type for your custom NER

[ ] # Create a new entity type for your custom NER
ner = nlp.create_pipe("ner")
nlp.add_pipe("ner")

<spacy.pipeline.ner.EntityRecognizer at 0x7f4ab38f5ee0>

- Adding the Labels

[ ] ner.add_label("CODE")
ner.add_label("MODULE_TITLE")
ner.add_label("SEMESTER")
ner.add_label("PERIOD")
ner.add_label("CREDITS")
ner.add_label("GRADE")

1

```

Figure 14: Dataset preprocessing for custom NER model

```

- Define the Entities

[ ] for index, row in df.iterrows():
    code = str(row["Code"])
    module_title = str(row["Module Title"])
    semester = str(row["Semester"])
    period = str(row["Period"])
    credits = str(row["Credits"])
    grade = str(row["Grade"])

    text = code + " " + module_title + " " + semester + " " + period + " " + credits + " " + grade

    entities = []
    current_pos = 0
    entities.append((current_pos, current_pos + len(code), "CODE"))
    current_pos += len(code) + 1
    entities.append((current_pos, current_pos + len(module_title), "MODULE_TITLE"))
    current_pos += len(module_title) + 1
    entities.append((current_pos, current_pos + len(semester), "SEMESTER"))
    current_pos += len(semester) + 1
    entities.append((current_pos, current_pos + len(period), "PERIOD"))
    current_pos += len(period) + 1
    entities.append((current_pos, current_pos + len(credits), "CREDITS"))
    current_pos += len(credits) + 1
    entities.append((current_pos, current_pos + len(grade), "GRADE"))

    TRAIN_DATA.append((text, {"entities": entities}))

```

Figure 15: Conversion of objects to JSON format

Training the model to identify module names and their corresponding grades

The NER model was then trained using this annotated training data, with multiple iterations and mini-batch training to optimize its performance. Finally, the trained NER model was saved for future use, allowing it to accurately recognize and extract module-related information from textual data.

Train the NER model

```
# Train the NER model
n_iter = 10
optimizer = nlp.begin_training()
for i in range(n_iter):
    random.shuffle(TRAIN_DATA)
    losses = {}
    batches = minibatch(TRAIN_DATA, size=compounding(4.0, 32.0, 1.001))
    for batch in batches:
        examples = []
        texts, annotations = zip(*batch)
        for i in range(len(texts)):
            doc = nlp.make_doc(texts[i])
            example = Example.from_dict(doc, annotations[i])
            examples.append(example)
        nlp.update(examples, sgds=optimizer, losses=losses)
    print("Losses:", losses)
```

Figure 16: Training the custom NER model

Pre-trained NER model

Dataset and data preprocessing

The dataset, initially in a text format, was annotated and transformed into structured JSON objects, as illustrated below.

	✓ MODULE_TITLE	2 GRADE
IT1000	Computer Fundamentals	1
	16-Apr 4	A

Figure 17: Dataset annotations

Model training

Initially, a blank Spacy model is loaded, and a DocBin object is created to store the training data. The training data, obtained from a JSON file contains text annotations and entity labels. The text and entity annotations are processed, and the entities are added to the DocBin object. This training data is then saved to a file. Next, a configuration file is initialized for the NER pipeline, followed by training the NER model using the training data. The trained model is saved and loaded for testing purposes.

```
] from spacy.util import filter_spans

#for training_example in training_data:
text = training_data['annotations'][0][0]
labels = training_data['annotations'][0][1]['entities']
doc = nlp.make_doc(text)
ents = []
for start, end, label in labels:
    span = doc.char_span(start, end, label=label, alignment_mode="contract")
    if span is None:
        print("Skipping entity")
    else:
        ents.append(span)
filtered_ents = filter_spans(ents)
doc.ents = filtered_ents
doc_bin.add(doc)

doc_bin.to_disk("train.spacy") # save the docbin object
```

Figure 18: docBin object to save the model

```
[ ] ! python -m spacy train config.cfg --output ./ --paths.train ./train.spacy --paths.dev ./train.spacy
```

Figure 19: Training the model

4.2.1.4. Skill area categorization and graphical representation of skill areas

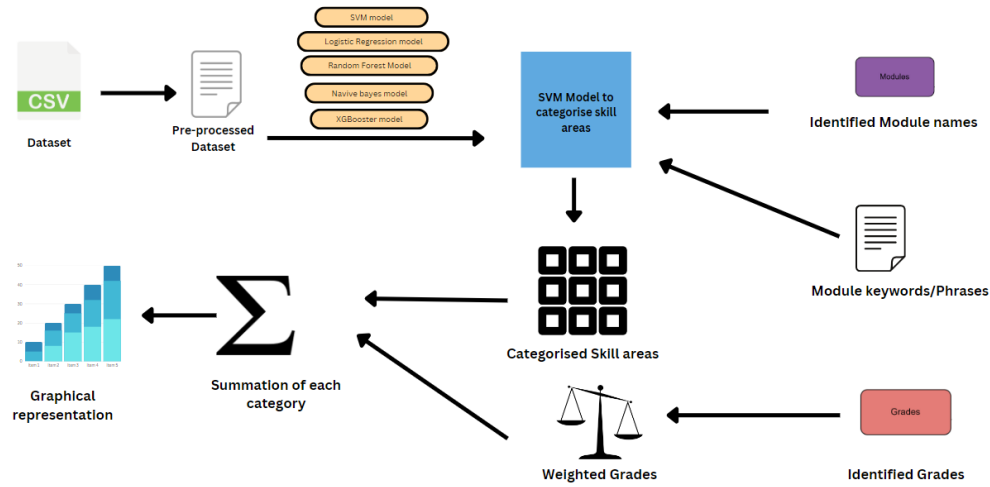


Figure 20: Skill area categorization and graphical representation of skill areas overview diagram

Dataset and Data Exploratory Analysis

The dataset used for predicting skill areas comprises three main columns: "Module Title," "Module Keywords," and "Skill Area," encompassing a total of 7776 records. The identification of the ten skill areas within this dataset was informed by comprehensive research drawn from various online sources [7] [8]. Furthermore, the population who was considered for this survey were the IT employees from the industry. From the survey below, we can see that the majority of the respondents have selected the first 10 categories.

Out of the list below please pick the skill areas that you think has a high demand in current time Period in the IT industry

38 responses

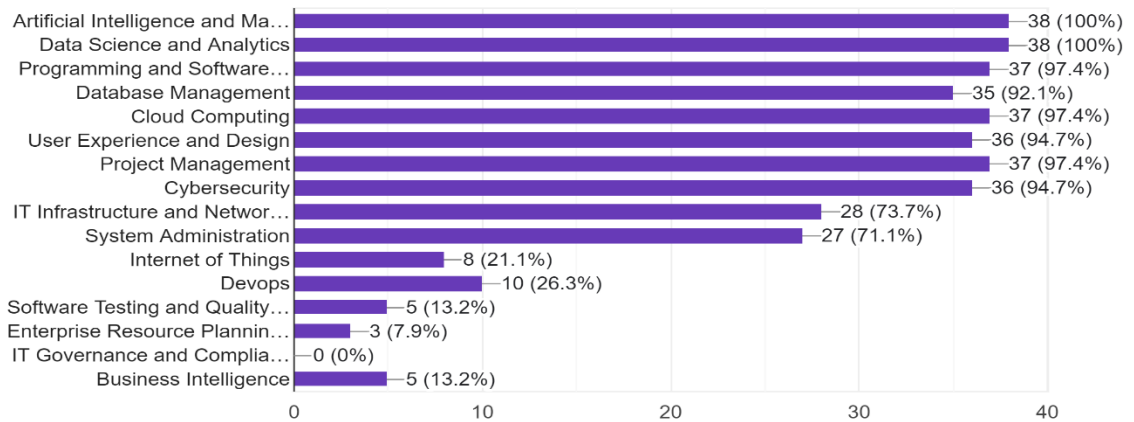


Figure 21: Survey results of identifying the high level skill areas

Thus, the categories are,

- System Administration
- IT Infrastructure and Networking
- Data Science and Data Analytics
- Artificial Intelligence and Machine Learning
- Cloud Computing
- Cybersecurity
- Database Management
- Project Management
- Programming and Software Development
- User Experience, and Design.

These skill categories were derived from real-world industry insights and serve as the basis for our skill area prediction model, enabling us to categorize module titles and keywords effectively in alignment with these crucial skill domains.

The exploratory data analysis phase involved a series of essential steps to gain insights from the dataset. Initially, the dataset was visualized, allowing us to better understand its structure and patterns. We also conducted checks for missing values within the dataset, ensuring data completeness and reliability. Furthermore, we examined the data types and shapes of the dataset's elements to ensure consistency and compatibility throughout. These fundamental exploratory tasks provided a solid foundation for subsequent data processing and analysis, helping us make informed decisions and draw meaningful conclusions within our research.

	A	B	C
1	Module Title	Module Keywords	Skill Area
2	Secure Coding in Perl	Perl programming language, Secure coding practices, Code vulnerabilities, Code security, Perl security	Programming and Software Development
3	Data Visualization for Project Management	Data visualization, Project management, Visual representation, Data analysis, Project tracking	Data Science and Analytics
4	Database Capacity Planning and Forecasting	Database capacity planning, Database forecasting, Performance optimization, Resource allocation	Database Management
5	Secure Coding in PHP	Secure coding, PHP programming, Web application security, Code vulnerabilities, Security best practices	Programming and Software Development
6	Database Audit and Compliance Procedures	Database audit, Compliance procedures, Data governance, Data security, Regulatory requirements	Database Management
7	Agile Project Communication and Collaboration Tools and Platforms	Agile project management, Project communication, Collaboration tools, Agile methodologies	Project Management
8	Cloud Microservices Architecture and Containerization	Cloud microservices, Containerization, Cloud architecture, Scalability, Service-oriented architecture	Cloud Computing
9	Database Migration and Upgrades	Database migration, Database upgrades, Data transfer, Schema conversion, Data synchronization	Database Management
10	Project Cost Tracking Techniques and Tools	Project cost tracking, Cost management, Budgeting, Expense tracking, Financial analysis	Project Management
11	Web Application Vulnerability Scanning and Assessment	Web application security, Vulnerability scanning, Penetration testing, Risk assessment	Cybersecurity
12	Cloud Backup and Recovery Solutions	Disaster recovery, Data protection, Cloud storage	Cloud Computing
13	Secure Configuration of Routers and Switches	Network security, Router configuration, Switch configuration, Access control, Firewall settings	Cybersecurity

Figure 22: Dataset for skill area prediction

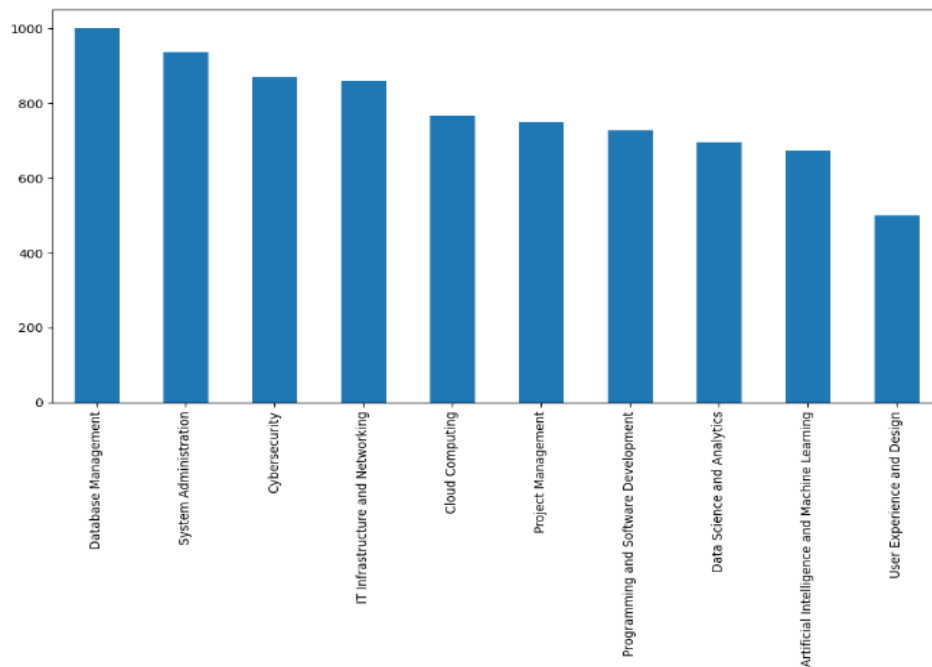


Figure 23: Visualization of skill area prediction dataset

```

Exploring the dataset

[ ] #Explore the data types of the columns
df.dtypes

Skill Area      object
Module Title    object
Module Keywords  object
dtype: object

[ ] #Checking for null values
df.isnull().sum()

Skill Area      0
Module Title    0
Module Keywords  0
dtype: int64

[ ] df.shape

(7775, 3)

```

Figure 24: EDA of skill area prediction dataset

Data preprocessing for model training

In the data preprocessing phase, several key tasks were performed to prepare the dataset for analysis. Initially, stopwords, which are common and non-informative words in English, were removed to focus on more meaningful content. Next, text data within the "Module Title" and "Module Keywords" columns were converted to lowercase for uniformity and to reduce text variations. Additionally, special characters and symbols were eliminated from the text, ensuring that only relevant alphanumeric characters and spaces remained. These data preprocessing steps collectively contributed to cleaning and standardizing the dataset, making it more suitable for subsequent analysis and modeling within the research framework.

Model Building for Skill Area Prediction

In this phase of the research methodology, Advanced natural language processing techniques were used to analyze textual data, and multi-class text classification models of supervised learning machine learning models were used to predict the skill area.

Logistic regression model

Initially, a Word2Vec model was trained on the provided text data, enabling us to understand the relationships between words and represent them as numerical vectors. These vectors were then utilized to create document-level representations by averaging the word vectors within each document. Next, the class labels were encoded for classification tasks and split the data into training and testing sets. A logistic regression classifier was trained using the training data to predict class labels for the test set. The accuracy of the classification model was assessed, yielding a performance metric that gauges the model's effectiveness in categorizing text data accurately, which is crucial for our research's objectives.

```

import gensim
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression

# Train Word2Vec model
sentences = [text.split() for text in x_train]
word2vec_model = gensim.models.Word2Vec(sentences, vector_size=100, window=5, min_count=1, workers=4)

# Create document vectors using Word2Vec
document_vectors = []
for text in sentences:
    vectors = [word2vec_model.wv[word] for word in text if word in word2vec_model.wv]
    if vectors:
        document_vectors.append(sum(vectors) / len(vectors))
    else:
        document_vectors.append([0] * 100)

import numpy as np

# Create document vectors using Word2Vec
document_vectors = [np.mean([word2vec_model.wv[word] for word in sentence], axis=0) for sentence in sentences]

# Convert document vectors to numpy array
X_train = np.array(document_vectors)

# Encode class labels
label_encoder = LabelEncoder()
y_train_encoded = label_encoder.fit_transform(y_train)

# Split into training and testing data
X_train, X_test, y_train_encoded, y_test_encoded = train_test_split(X_train, y_train_encoded, test_size=0.2, random_state=42)

# Train a logistic regression classifier
classifier = LogisticRegression()
classifier.fit(X_train, y_train_encoded)

# Predict and evaluate
X_test_vectors = []
for text in x_test:
    vectors = [word2vec_model.wv[word] for word in text.split() if word in word2vec_model.wv]
    if vectors:
        X_test_vectors.append(sum(vectors) / len(vectors))
    else:
        X_test_vectors.append([0] * 100)
X_test_vectors = np.array(X_test_vectors)

y_pred_encoded = classifier.predict(X_test_vectors)
y_pred = label_encoder.inverse_transform(y_pred_encoded)

accuracy = accuracy_score(y_pred, y_test)
print(f'Accuracy: {accuracy:.4f}')

```

Figure 25: Logistic regression algorithm for skill area prediction

Naïve Bayes Model

Multinomial Naive Bayes classifier was used to analyze and classify textual data. Initially, class labels were encoded for classification purposes, ensuring the model understands the target categories. Next, a processing pipeline was constructed that included text vectorization using CountVectorizer, followed by term frequency-inverse document frequency (TF-IDF) transformation and, finally, the Multinomial Naive Bayes classifier. The pipeline was then fitted to the training data, allowing the model to learn and understand the relationships between text features and their corresponding categories. Next, the trained model was used to predict class labels for the test data. The accuracy of our classification model was assessed, providing a measure of its effectiveness in correctly categorizing textual data, which is vital for our research's objectives.

```
[ ] from sklearn.naive_bayes import MultinomialNB
    from sklearn.pipeline import Pipeline
    from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
    from sklearn.metrics import accuracy_score
    from sklearn.preprocessing import LabelEncoder

    # Encode class labels
    label_encoder = LabelEncoder()
    y_train_encoded = label_encoder.fit_transform(y_train)

    # Create the pipeline
    naivebayes = Pipeline([
        ('vect', CountVectorizer()),
        ('tfidf', TfidfTransformer()),
        ('clf', MultinomialNB()),
    ])

    # Fit the pipeline
    naivebayes.fit(x_train, y_train_encoded)

    # Predict and evaluate
    y_pred_encoded = naivebayes.predict(x_test)

    # Decode predicted labels
    y_pred = label_encoder.inverse_transform(y_pred_encoded)

    accuracy = accuracy_score(y_pred, y_test)
    print(f'Accuracy: {accuracy}')
```

Figure 26: Naive Bayes algorithm for skill area prediction

XGBoost Model

Apart from logistic regression and Naïve Baiyes, the XGBoost classifier was used to analyze and classify textual data. Initially, class labels were encoded to facilitate classification tasks, ensuring the model understands the target categories. Then a processing pipeline was built that included text vectorization using CountVectorizer, followed by term frequency-inverse document frequency (TF-IDF) transformation, and finally, the XGBoost classifier. The pipeline was fitted to the training data, enabling the model to learn the intricate relationships between text features and their corresponding categories. Next, this trained XGBoost model was used to predict class labels for the test data. The accuracy of our classification model was assessed, providing a measure of its effectiveness in accurately categorizing textual data, which is crucial for our research objectives.

```
[ ] from sklearn.preprocessing import LabelEncoder
    from xgboost import XGBClassifier
    from sklearn.pipeline import Pipeline
    from sklearn.feature_extraction.text import CountVectorizer, TfidfTransformer
    from sklearn.metrics import accuracy_score

    # Encode class labels
    label_encoder = LabelEncoder()
    y_train_encoded = label_encoder.fit_transform(y_train)

    xgboost = Pipeline([
        ('vect', CountVectorizer()),
        ('tfidf', TfidfTransformer()),
        ('clf', XGBClassifier())
    ])

    xgboost.fit(x_train, y_train_encoded)

    y_pred_encoded = xgboost.predict(x_test)

    # Decode predicted labels
    y_pred = label_encoder.inverse_transform(y_pred_encoded)

    accuracy = accuracy_score(y_pred, y_test)
    print(f'Accuracy: {accuracy}')
```

Figure 27: XGBoost algorithm for skill area prediction

Random Forest Model

Next, Random Forest classifier was used to analyze and classify textual data. Initially, the dataset was then split into training and testing subsets. To prepare the textual data for analysis, we utilized CountVectorizer to convert it into a numerical format that the machine learning model can understand. Subsequently, we trained the Random Forest model on the training data, enabling it to learn patterns and relationships within the text data. Using this trained model, we predicted skill areas for the test data. Finally, the model's performance was evaluated by calculating its accuracy, providing a measure of its effectiveness in correctly categorizing the textual data, which is fundamental to our research objectives.

```
[ ] from sklearn.feature_extraction.text import CountVectorizer
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score

    # Preprocess the data
    nltk.download('stopwords')
    STOPWORDS = set(stopwords.words('english'))

    def clean_text(text):
        text = text.lower()
        text = re.sub('[/(){}[\]\|@,;]', ' ', text)
        text = re.sub('[^0-9a-z #+_-]', '', text)
        text = ' '.join(word for word in text.split() if word not in STOPWORDS)
        return text

    df['Module Title'] = df['Module Title'].apply(clean_text)
    df['Module Keywords'] = df['Module Keywords'].apply(clean_text)

    # Split the dataset into training and testing
    x = df['Module Title'].str.lower() + ' ' + df['Module Keywords'].str.lower()
    y = df['Skill Area']
    x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

    # Initialize CountVectorizer
    countvect = CountVectorizer()

    # Prepare the feature matrix
    x_train_encoded = countvect.fit_transform(x_train.tolist())
    x_test_encoded = countvect.transform(x_test.tolist())

    # Train the Random Forest model
    random_forest = RandomForestClassifier(n_estimators=100, random_state=42)
    random_forest.fit(x_train_encoded, y_train) # Use y_train instead of y_train_encoded

    # Predict on the test set
    y_pred = random_forest.predict(x_test_encoded)

    # Evaluate the model
    accuracy = accuracy_score(y_pred, y_test)
    print(f'Accuracy: {accuracy}')
```

Figure 28: Random Forest algorithm for skill area prediction

Support Vector Machine Model

Finally, the SVM model was used to analyze and classify textual data. Initially, the issue of class imbalance in the dataset was addressed by employing the Random Oversampling technique, a method that generates synthetic samples to balance the classes. We initially split the dataset into training and testing sets, ensuring the stratification of classes to maintain their distribution. The text data was then transformed into numerical vectors using CountVectorizer. To address the class imbalance, the Synthetic Minority Over-sampling Technique (SMOTE) was used on the training data, creating additional synthetic samples for the minority class. Subsequently, we trained a Support Vector Machine (SVM) model on the resampled training data. Using this model, the predictions were made on the test data and evaluated its performance. Afterwards, the accuracy was calculated as a measure of the model's effectiveness in correctly classifying the data and visualizing the confusion matrix to gain insights into its performance. Furthermore, a classification report was generated to provide a comprehensive assessment of the model's performance in categorizing textual data, which is crucial for the research objectives.

```
[ ] from imblearn.over_sampling import RandomOverSampler

# Splitting the dataset into training and testing
from sklearn.model_selection import train_test_split
x = df['Module Title'].str.lower() + ' ' + df['Module Keywords'].str.lower()
y = df['Skill Area']
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, stratify=y, random_state=42)

# Initialize CountVectorizer
countvect = CountVectorizer()

# Transform text data into vectors
x_train_vect = countvect.fit_transform(x_train)
x_test_vect = countvect.transform(x_test)

# Create an instance of RandomOverSampler
random_oversampler = RandomOverSampler(sampling_strategy='auto', random_state=0)

# Apply SMOTE to the vectorized data
X_train = pd.DataFrame(x_train_vect)
x_resample, y_resample = SMOTE().fit_resample(x_train_vect, y_train)

# Initialize and train the SVM model
svm_model = SVC(kernel='rbf', random_state=0)
svm_model.fit(x_resample, y_resample)

# Predict and evaluate
prediction = svm_model.predict(x_test_vect)
accuracy = accuracy_score(y_test, prediction)
print("Accuracy:", accuracy)

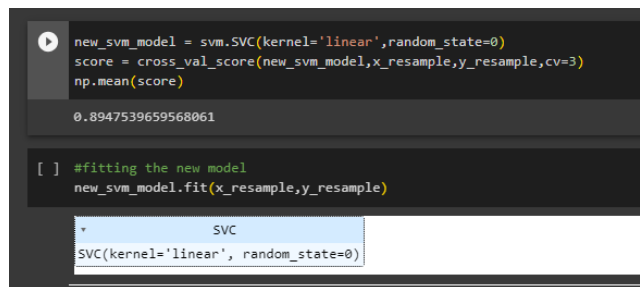
# Visualize confusion matrix
sns.heatmap(confusion_matrix(y_test, prediction), annot=True, fmt="g")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# Display classification report
print(classification_report(y_test, prediction))
```

Figure 29: SVM algorithm for skill area prediction

Considering the accuracies of all models, The Support Vector Machine (SVM) model was selected as the best option because it consistently achieved the highest accuracy in our classification tasks. This means that SVM was the most reliable choice for accurately categorizing skill areas from textual data, making it the preferred model for the research.

In the process of fine-tuning the SVM model, to improve its performance we used a linear kernel and applied cross-validation for robust evaluation. The mean score of the cross-validation results was calculated to assess the model's effectiveness. Next, the new SVM model was fitted on the resampled training data, ensuring it was optimized for the research's objectives. Finally, the model was saved to a file using the Pickle library, making it available for future use in skill area prediction tasks.



```
new_svm_model = svm.SVC(kernel='linear', random_state=0)
score = cross_val_score(new_svm_model, x_resample, y_resample, cv=3)
np.mean(score)

0.8947539659568061

[ ] #fitting the new model
new_svm_model.fit(x_resample, y_resample)

+ SVC
SVC(kernel='linear', random_state=0)
```

Figure 30: Fine-tuned SVM model

Skill area categorization and graphical representation

In the research methodology, several critical steps were taken to process academic transcripts and predict skill areas. Initially, the extracted text from the academic transcripts was divided into sections based on academic years using regular expressions. The custom Named Entity Recognition (NER) model that was built was used to extract module titles and their corresponding grades from these sections.

Each identified module title was combined with its corresponding module keywords and phrases which were extracted from the module outline using n-grams and passed into the SVM model to categorize them into different skill areas.

Then, the weighted grades were calculated for each module title. A grading scheme was defined, assigning weights to grades from 'A+' to 'E'.

- A+ : 10
- A : 9
- A- : 8
- B+ : 7
- B : 6
- B- : 5
- C+ : 4
- C : 3
- C- : 2
- D+ : 1
- D : 1
- E : 1

These weighted grades were used to evaluate the academic performance associated with each skill area. Next, a table was constructed, summarizing skill areas, module titles, and their corresponding weighted grades.

Furthermore, the research analyzed the distribution of weighted grades across different skill areas. The skill areas were grouped, and the total weighted grades for each category were calculated. These categories were sorted in descending order based on their total

weighted grades, providing insights into the relative importance of various skill areas within the academic transcripts.

In order to present a concise overview of the skill areas a graphical representation of the skill categories was used, which serves as a valuable tool for hiring managers, offering a clear and concise view of candidates' skill profiles.

```
# Split the tuple to access the model and vectorizer
model, countvect = classification_model

# Read module titles from the first CSV file
module_titles_df = pd.read_csv('files/academic_transcript/extracted_data.csv')

# Extract module titles from the DataFrame
module_titles = module_titles_df['Module Title']

# Read module titles and module keywords from the second CSV file
module_keywords_df = pd.read_csv('files/academic_transcript/Module Keywords.csv')

# Create a dictionary to map module titles to their corresponding module keywords
module_title_to_keywords = dict(zip(module_keywords_df['Module Title'], module_keywords_df['Module Keywords']))

# Define a function to predict skill areas
def predict_skill_area(module_titles, module_title_to_keywords):
    predicted_categories = []
    for module_title in module_titles:
        module_keywords = module_title_to_keywords.get(module_title, "")
        combined_text = module_title.lower() + ' ' + module_keywords.lower()
        module_title_vector = countvect.transform([combined_text])
        predicted_category = model.predict(module_title_vector)
        predicted_categories.append(predicted_category[0]) # Assuming you want a single prediction
    return predicted_categories

# Predict skill areas using the function
predicted_skill_areas = predict_skill_area(module_titles, module_title_to_keywords)

# Map the module titles from the extracted data to the module titles in the module outline
module_titles_df['Category'] = predicted_skill_areas
```

Figure 31: Skill area prediction

```
# Predict skill areas using the function
predicted_skill_areas = predict_skill_area(module_titles, module_title_to_keywords)

# Map the module titles from the extracted data to the module titles in the module outline
module_titles_df['Category'] = predicted_skill_areas

# Define the grade weighting dictionary
grade_weighting = {
    'A+': 10,
    'A': 9,
    'A-': 8,
    'B+': 7,
    'B': 6,
    'B-': 5,
    'C+': 4,
    'C': 3,
    'C-': 2,
    'D+': 1,
    'D': 1,
    'E': 1
}

# Calculate the weighted grades based on the grade weighting dictionary
module_titles_df['Weighted Grade'] = module_titles_df['Grade'].map(grade_weighting)

# Calculate the weighted grades based on the grade weighting dictionary
module_titles_df['Weighted Grade'] = module_titles_df['Grade'].map(grade_weighting)

# Group the skill area table by the 'Category' column and calculate the sum of the 'Weighted Grade' column for each category
category_totals = module_titles_df.groupby('Category')['Weighted Grade'].sum()

# Sort the categories by their total weighted grades in descending order
category_totals = category_totals.sort_values(ascending=False)
```

Figure 32: Skill area categorization

4.2.1.5. Academic Transcript score prediction model

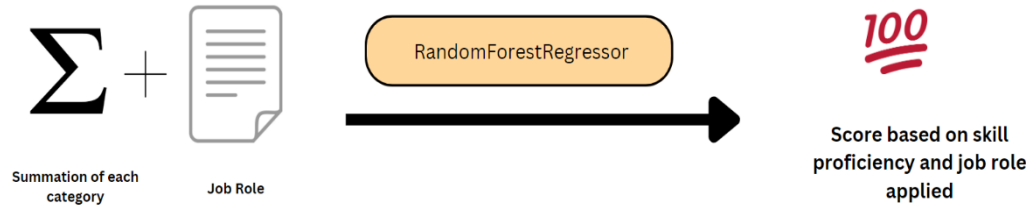


Figure 33: Academic Transcript score prediction model overview diagram

Dataset

In the context of this research, a dataset comprising 11,849 records was used. The columns included "job role" and "job description" to provide insight into different employment positions. Additionally, the dataset encompassed skill categories such as "Programming and Software Development," "Data Science and Analytics," "Database Management," "Cloud Computing," "Project Management," "Cybersecurity," "IT Infrastructure and Networking," "Artificial Intelligence and Machine Learning," "System Administration," and "User Experience and Design." These skill categories were assessed along with a "Final Score" column, representing academic performance.

Job role	Job Description	Programming and Software Development	Data Science and Analytics	Database Management	Cloud Computing	Project Management	Cybersecurity	IT Infrastructure and Networking	Artificial Intelligence and Machine Learning	System Administration	User Experience and Design	Final Score
Technical Project Manager	As a Technical Project Manager, you will oversee the planning, execution, and delivery of technology projects. Your role involves coordinating team members, setting	39.32	13.22	34.69	48.57	90.6	15.01	23.63	24.3	54.94	44.81	85.49
Identity and Access Management (IAM) Analyst Team Lead	Lead a team of IAM analysts to safeguard organizational data by managing user identities, access permissions, and security policies.	59.39	56.1	43.69	36.31	17.82	75.59	19.11	32.49	56.93	16.63	83.32
API Developer	Design and develop Application Programming Interfaces (APIs) that enable seamless communication between different software applications.	97.95	77.06	67.42	50.87	69.63	6.31	63.32	79.3	22.01	23.43	97.81
AI Platform Specialist (Senior)	As a Senior AI Platform Specialist, you'll be responsible for configuring and optimizing AI platforms, ensuring they run efficiently and support data-driven decision-making.	30.84	54.79	11.12	33.17	30.1	13.31	29.61	76.35	26.7	8.6	93.28
Cloud Business Analyst	Analyze cloud technology trends and their impact on business operations, helping organizations make informed decisions regarding cloud adoption and optimization.	69.34	22.03	14.72	96.12	28	18.94	6.11	1.27	27.28	21.04	73.09
AI Platform Architect (Senior)	Architect advanced AI solutions, creating frameworks that support machine learning models and data pipelines for complex AI projects.	68.49	56.1	16.15	1.7	0.85	11.42	32.35	89.86	0.19	56.15	86.31

Figure 34:Dataset for academic transcript score prediction model

Data preprocessing and feature engineering

In data preprocessing and feature engineering, duplicate entries based on job roles were removed. Then, to address any missing values within the dataset, a consistent approach was adopted by replacing them with zero values. Additionally, to facilitate the model's understanding of job roles, a one-hot encoding technique was applied to the "Job role" column, converting categorical job roles into a numerical format. Simultaneously, to convert the "job description" column into a numerical format, a text preprocessing method known as CountVectorizer was employed.

```
# Remove duplicate rows
data.drop_duplicates(inplace=True)

# Fill null values with 0
data.fillna(0, inplace=True)
```

Figure 35: Drop duplicates and full missing values

```
# Define a column transformer for one-hot encoding the "Job Role" column
column_transformer = ColumnTransformer(
    transformers=[
        ('encoder', OneHotEncoder(), ['Job role'])
    ],
    remainder='passthrough'
)

# Transform the X data including CountVectorizer encoding
X_encoded = column_transformer.fit_transform(X)

# After the transformation
print("X_encoded shape:", X_encoded.shape)
```

Figure 36: Job role conversion to numeric using One hot encoder

```
# Define the CountVectorizer for the "Job Description" column
vectorizer = CountVectorizer()

# Fit and transform the "Job Description" column
job_description_matrix = vectorizer.fit_transform(data['Job Description'])

# Convert the result to a DataFrame
job_description_df = pd.DataFrame(job_description_matrix.toarray(), columns=vectorizer.get_feature_names_out())

# Concatenate the new DataFrame with the original X_encoded
X_encoded = pd.concat([pd.DataFrame(X_encoded, columns=column_transformer.get_feature_names_out()), job_description_df], axis=1)
```

Figure 37: Job description conversion to numeric using Countvectorizer

Model building

The dataset was divided into two distinct sets: a training set and a testing set, enabling the assessment of model performance. Four regression models were considered for this task: the Random Forest Regressor, Linear Regression, Support Vector Regression (SVR), and Gradient Boost Regressor. Initially, the dataset underwent feature transformation, involving the addition of polynomial features and standardization to enhance the models' predictive capabilities. Next, the models were trained, and the models' performance was evaluated by two metrics, Mean Squared Error (MSE), which quantifies the average squared difference between predicted and actual scores, and the R-squared (R2) score, which serves as an indicator of how effectively the models elucidate the variations in the data.

```
[ ] # Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_encoded, y, test_size=0.2, random_state=42)
```

Figure 38: Splitting into training and testing sets

```
[ ] # Train the RandomForestRegressor model
rf_regressor = Pipeline([
    ('poly', PolynomialFeatures(degree=2)), # Add polynomial features
    ('scaler', StandardScaler(with_mean=False)), # Standardize features without centering
    ('regressor', RandomForestRegressor(n_estimators=100, random_state=42))
])

rf_regressor.fit(X_train, y_train)

# Predict
y_pred = rf_regressor.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")
```

Figure 39: Random Forest Regressor model to predict score

```

# Train the LinearRegressor model
linear_regressor = Pipeline([
    ('poly', PolynomialFeatures(degree=2)), # Add polynomial features
    ('scaler', StandardScaler()), # Standardize features
    ('regressor', LinearRegression())
])

linear_regressor.fit(X_train, y_train)

# Predict
y_pred = linear_regressor.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")

```

Figure 40: Linear Regressor model to predict score

```

# Train the SupportVectorRegressor model
svm_regressor = Pipeline([
    ('poly', PolynomialFeatures(degree=2)), # Add polynomial features
    ('scaler', StandardScaler()), # Standardize features
    ('regressor', SVR(kernel='linear', C=1.0))
])

svm_regressor .fit(X_train, y_train)

# Predict
y_pred = svm_regressor .predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")

```

Figure 41: Support Vector Regressor model to predict score

```

# Create a GradientBoostingRegressor pipeline
gb_regressor = Pipeline([
    ('poly', PolynomialFeatures(degree=2)), # Add polynomial features if needed
    ('scaler', StandardScaler()), # Standardize features if needed
    ('regressor', GradientBoostingRegressor(n_estimators=100, random_state=42))
])

# Train the GradientBoostingRegressor model
gb_regressor.fit(X_train, y_train)

# Predict on the test data
y_pred = gb_regressor.predict(X_test)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse:.2f}")
print(f"R-squared (R2) Score: {r2:.2f}")

```

Figure 42: Gradient Boost Regressor model to predict score

4.2.1.6. Frontend Development

The frontend of this research component was built using HTML, CSS, and JavaScript. Flask was used as the engine to connect the frontend with the backend(Machine learning models and NLP algorithms).

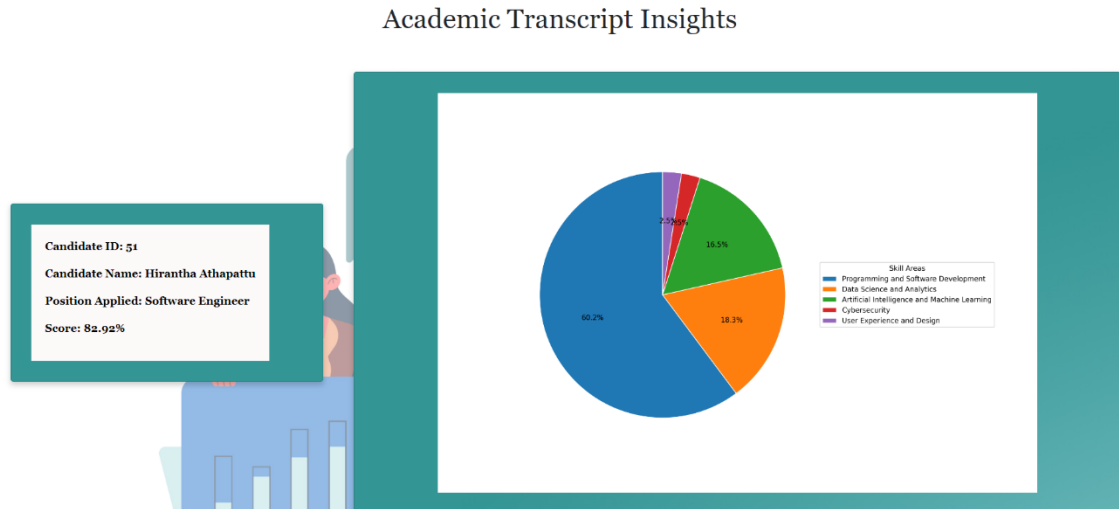
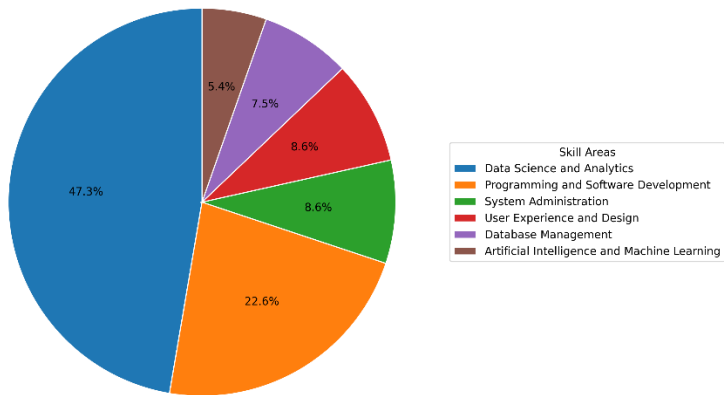
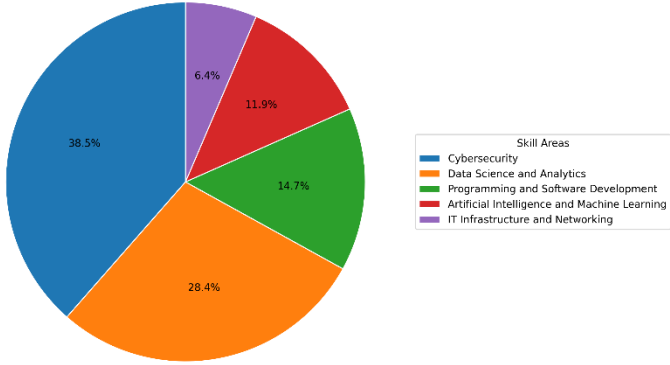


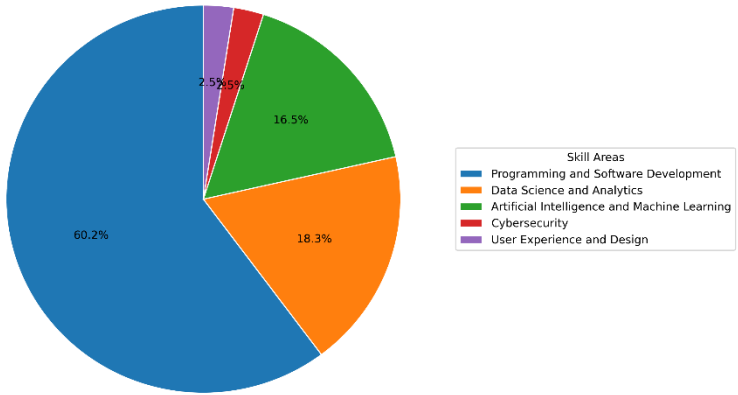
Figure 43:Academic Transcript user interface

4.2.2. Testing

4.2.2.1. Skill area Categorization

Test Case 01															
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart														
Input	An academic transcript of a data science specialized candidate														
Expected output	Categorized skill areas with the higher weight for Data Science and Analytics.														
Actual Output	 <p>A pie chart titled 'Skill Areas' showing the distribution of skill areas for a data science specialized candidate. The chart is divided into six segments with the following percentages: Data Science and Analytics (47.3%, blue), Programming and Software Development (22.6%, orange), System Administration (8.6%, green), User Experience and Design (8.6%, red), Database Management (7.5%, purple), and Artificial Intelligence and Machine Learning (5.4%, brown). A legend on the right lists the skill areas with their corresponding colors.</p> <table border="1"><thead><tr><th>Skill Area</th><th>Percentage</th></tr></thead><tbody><tr><td>Data Science and Analytics</td><td>47.3%</td></tr><tr><td>Programming and Software Development</td><td>22.6%</td></tr><tr><td>System Administration</td><td>8.6%</td></tr><tr><td>User Experience and Design</td><td>8.6%</td></tr><tr><td>Database Management</td><td>7.5%</td></tr><tr><td>Artificial Intelligence and Machine Learning</td><td>5.4%</td></tr></tbody></table>	Skill Area	Percentage	Data Science and Analytics	47.3%	Programming and Software Development	22.6%	System Administration	8.6%	User Experience and Design	8.6%	Database Management	7.5%	Artificial Intelligence and Machine Learning	5.4%
Skill Area	Percentage														
Data Science and Analytics	47.3%														
Programming and Software Development	22.6%														
System Administration	8.6%														
User Experience and Design	8.6%														
Database Management	7.5%														
Artificial Intelligence and Machine Learning	5.4%														
Test Status	Pass														

Test Case 02													
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart												
Input	An academic transcript of a cyber security specialized candidate												
Expected output	Categorised skill areas with the higher weight for Cyber-security												
Actual output	 <p>A pie chart titled 'Skill Areas' showing the distribution of skills. The chart is divided into five segments: Cybersecurity (blue, 38.5%), Data Science and Analytics (orange, 28.4%), Programming and Software Development (green, 14.7%), Artificial Intelligence and Machine Learning (red, 11.9%), and IT Infrastructure and Networking (purple, 6.4%). A legend to the right of the chart lists the skill areas with their corresponding colors.</p> <table border="1"> <thead> <tr> <th>Skill Area</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Cybersecurity</td> <td>38.5%</td> </tr> <tr> <td>Data Science and Analytics</td> <td>28.4%</td> </tr> <tr> <td>Programming and Software Development</td> <td>14.7%</td> </tr> <tr> <td>Artificial Intelligence and Machine Learning</td> <td>11.9%</td> </tr> <tr> <td>IT Infrastructure and Networking</td> <td>6.4%</td> </tr> </tbody> </table>	Skill Area	Percentage	Cybersecurity	38.5%	Data Science and Analytics	28.4%	Programming and Software Development	14.7%	Artificial Intelligence and Machine Learning	11.9%	IT Infrastructure and Networking	6.4%
Skill Area	Percentage												
Cybersecurity	38.5%												
Data Science and Analytics	28.4%												
Programming and Software Development	14.7%												
Artificial Intelligence and Machine Learning	11.9%												
IT Infrastructure and Networking	6.4%												
Test Status	Pass												

Test Case 03													
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart												
Input	An academic transcript of a software engineering specialized candidate												
Expected output	Categorised skill areas with the higher weight for Programming and Software development												
Actual output	 <p>A pie chart titled 'Skill Areas' showing the distribution of skills. The largest slice is 'Programming and Software Development' at 60.2%, followed by 'Data Science and Analytics' at 18.3%, 'Artificial Intelligence and Machine Learning' at 16.5%, 'Cybersecurity' at 2.5%, and 'User Experience and Design' at 2.5%.</p> <table border="1"> <thead> <tr> <th>Skill Area</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>Programming and Software Development</td> <td>60.2%</td> </tr> <tr> <td>Data Science and Analytics</td> <td>18.3%</td> </tr> <tr> <td>Artificial Intelligence and Machine Learning</td> <td>16.5%</td> </tr> <tr> <td>Cybersecurity</td> <td>2.5%</td> </tr> <tr> <td>User Experience and Design</td> <td>2.5%</td> </tr> </tbody> </table>	Skill Area	Percentage	Programming and Software Development	60.2%	Data Science and Analytics	18.3%	Artificial Intelligence and Machine Learning	16.5%	Cybersecurity	2.5%	User Experience and Design	2.5%
Skill Area	Percentage												
Programming and Software Development	60.2%												
Data Science and Analytics	18.3%												
Artificial Intelligence and Machine Learning	16.5%												
Cybersecurity	2.5%												
User Experience and Design	2.5%												
Test Status	Pass												

4.2.2.2. Score based on skill proficiency

Test Case 01	
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart. The score is given based on the summation of weighted grades of all skill categories
Input	An academic transcript of a data science specialized candidate who applied for the Data Scientist position
Expected output	A score greater than 70%
Actual Output	<div>Position Applied: Data Scientist Score: 71.51%</div>
Test Status	Pass

Test Case 02	
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart. The score is given based on the summation of weighted grades of all skill categories
Input	An academic transcript of a data science specialized candidate who applied for the Software Engineer position
Expected Output	A score between 50% to 75%
Actual output	<div> Position Applied: Software Engineer Score: 59.03% </div>
Test Status	Pass

Test Case 03	
Description	When the academic transcript is uploaded, the skill areas should be categorized and shown in a pie chart. The score is given based on the summation of weighted grades of all skill categories
Input	An academic transcript of a cyber security specialized candidate who applied for the Cyber Security Engineer position
Expected output	A score greater than 70%
Actual output	<div> Position Applied: Cyber Security Engineer Score: 77.43% </div>
Test Status	Pass

4.3. COMMERCIALIZATION

When a candidate is hired, especially in Sri Lanka, the process takes a long time. The company examines resumes, shortlists them, interviews them, and conducts background checks. In the long run, this can be inefficient and may tend to lower the productivity of the company as well. As the information technology business is continually evolving, there is a great demand for new employees. And when organizations take time to respond and undergo the manual recruiting procedure, they tend to lose applicants who can be highly beneficial to the organization.

The "Academic Transcript Analysis" software component represents a pioneering solution poised to revolutionize the recruitment landscape. Tailored for a prominent university, this innovative tool is designed to analyze the academic records of recent graduates, providing organizations with an invaluable resource to identify top-tier talent. Let's delve into how this cutting-edge technology can be commercialized to address the pressing needs of companies.

Streamlined Talent Identification: In today's competitive job market, the ability to swiftly identify high-potential candidates is paramount. Our software, powered by an advanced algorithm, offers organizations the capability to evaluate a candidate's academic performance, discern their strengths and weaknesses, and pinpoint specific skill areas effortlessly and accurately. This streamlining of the candidate selection process translates to significant time savings for recruiters, who can now efficiently identify the most promising graduates from a pool of applicants with limited professional experience.

Enhanced Efficiency and Informed Decision-Making: When our academic transcript analysis software is adopted by businesses, it presents a golden opportunity to significantly boost their recruitment processes. By automating the assessment of academic records, companies can cut down on the time and energy spent on manual screening and evaluations. This isn't just about saving money but also about empowering organizations

to make smarter hiring choices based on solid data and insights. Ultimately, this leads to hiring talent that's a better fit for the organization's specific requirements and goals.

Empowering the Future of Talent Acquisition: In essence, the commercialization of the academic transcript analysis software component of this system, is a pivotal step towards shaping the future of talent acquisition. Its user-friendly interface, adaptability, reliability, and reliance on data-driven techniques provide organizations with the tools they require to thrive in the ever-changing job market. By using this groundbreaking solution, IT companies can spot and bring on board candidates who possess the skills and knowledge necessary to excel in their respective fields. This, in turn, will boost their competitive position and drive their organizations towards new horizons.

Overall, this system is an efficient, reliable, and cost-effective solution that can help companies of all sizes streamline their recruitment process and identify the best candidates for the job. With its advanced CV analysis and recruitment software, our solution can be a valuable asset to any company looking to optimize its hiring process.

5. RESULTS & DISCUSSION

5.1. Results

5.1.1. Extracting module keywords and phrases using n-grams

```
Row 6 - Top Words: ['oriented', 'object', 'solution', 'given', 'class', 'design', 'identifying', 'relationship', 'implement', 'language']
Row 7 - Top Words: ['software', 'engineering', 'verification', 'artifact', 'produce', 'validation', 'appreciate', 'softwareintensive', 'specification', 'mai']
Row 8 - Top Words: ['modeling', 'information', 'system', 'data', 'mainly', 'list', 'mandatory', 'sp', 'relational', 'stream']
Row 9 - Top Words: ['web', 'application', 'explain', 'usability', 'standard', 'language', 'technology', 'related', 'apply', 'side']
Row 10 - Top Words: ['engineering', 'mathematical', 'unit', 'special', 'emphasis', 'electrical', 'electronic', 'computer', 'studying', 'encountered', 'mathematics']
Row 11 - Top Words: ['essential', 'computer', 'network', 'course', 'cover', 'scheme', 'addressing', 'router', 'lan', 'perform']
Row 12 - Top Words: ['concurrency', 'application', 'programming', 'apply', 'development', 'synthesize', 'validate', 'justify', 'suitable', 'java']
Row 13 - Top Words: ['database', 'sql', 'design', 'schema', 'handsonexperience', 'refinement', 'conceptual', 'administrative', 'logical', 'cater']
Row 14 - Top Words: ['tcp', 'switching', 'routing', 'ip', 'configuration', 'operation', 'theory', 'evaluation', 'every', 'everybody']
Row 15 - Top Words: ['assignment', 'unix', 'utility', 'complete', 'lab', 'tutorial', 'lecture', 'major', 'system', 'administration']
Row 16 - Top Words: ['oriented', 'object', 'software', 'unified', 'languageuml', 'supportive', 'design', 'modeling', 'applying', 'pattern']
Row 17 - Top Words: ['logic', 'computer', 'signal', 'keep', 'combinational', 'electronic', 'graduate', 'number', 'sequential', 'covering']
Row 18 - Top Words: ['statistical', 'statistic', 'output', 'theory', 'different', 'using', 'software', 'data', 'behind', 'understood']
Row 19 - Top Words: ['electronic', 'analog', 'device', 'understand', 'provide', 'analogue', 'necessa', 'amplifier', 'operational', 'circuitry']
Row 20 - Top Words: ['oracle', 'database', 'backup', 'managing', 'instanceconfiguring', 'undo', 'creation', 'installing', 'controlling', 'creating']
Row 21 - Top Words: ['routing', 'static', 'ipv4', 'nat', 'switched', 'discovery', 'dhcp', 'list', 'vlans', 'switch']
Row 22 - Top Words: ['database', 'solution', 'plan', 'query', 'application', 'performance', 'propose', 'address', 'speed', 'recommend']
Row 23 - Top Words: ['security', 'cryptography', 'control', 'introduction', 'trusted', 'multilevel', 'symmetric', 'public', 'asymmetric', 'malicious']
Row 24 - Top Words: ['system', 'io', 'structure', 'process', 'management', 'cpu', 'synchronization', 'deadlock', 'secondary', 'protection']
Row 25 - Top Words: ['structured', 'information', 'design', 'analysis', 'system', 'also', 'student', 'computer', 'aided', 'handson', 'matter']
Row 26 - Top Words: ['problem', 'solving', 'effectively', 'solver', 'defining', 'accommodating', 'collaborating', 'ineffective', 'varied', 'difficult']
Row 27 - Top Words: ['network', 'advancement', 'recent', 'technological', 'evolution', 'field', 'provides', 'theory', 'introduction', 'fundamental']
```

Figure 44: Extracted module keywords and phrases from module outline descriptions

5.1.2. Extracting data from academic transcripts

The text extracted from the OCR engine is shown as follows.

```
YEAR 1
171010 Introduction to Programming 1 Apr - 2019 4 A
171020 Introduction to Computer Systems 1 Apr - 2019 4 C+
171030 Mathematics for Computing 1 Apr - 2019 4 B
171040 Communication Skills 1 Apr - 2019 3 A-
171050 Object Oriented Concepts 2 Oct - 2019 2 A
171060 Software Process Modeling 2 Oct - 2019 3 A
| 171080 English for Academic Purposes 2 Oct - 2019 3 B
| 171090 Information Systems & Data Modeling 2 Oct - 2019 4
| 171100 Internet & Web Technologies 2 Oct - 2019 4 A-
- Year 1 Credits=31.00 Year 1 Grade Points=108.10 Year 1 GPA=3.49
STATUS: Completed YEAR 1
Dean's List Recognition 2nd Semester
YEAR 2
172020 Software Engineering 1 Apr - 2020 4 A
172030 Object Oriented Programming 1 Jun - 2020 4 A
172040 Database Management Systems 1 Apr - 2020 4 A
172050 Computer Networks 1 Jun - 2020 4 A
172060 Operating Systems and System Administration 1 Apr - 2020 4 B+
| 172010 Mobile Application Development 2 Oct - 2020 4 B+
172070 Data Structures & Algorithms 2 Oct - 2020 4 A
172080 IT Project 2 Oct - 2020 4 A
172090 Professional Skills 2 Oct - 2020 2 B+
172100* Employability Skills Development - Seminar 2 Oct - 2020 Non-Credit C+
172110 Probability & Statistics 2 Oct - 2020 3 A
*Year 2 Credits=37.00 Year 2 Grade Points=141.00 Year 2 GPA=3.81
```

Figure 45: Extracted text from an academic transcript using OCR

5.1.3. Name entity recognition model

5.1.3.1. Custom NER model

The custom NER model was trained with 10 epochs and had a loss of 2.2673915734306062e-07.

```
Losses: {'ner': 1080.9641413251313}
Losses: {'ner': 81.69892219707003}
Losses: {'ner': 26.13685030728511}
Losses: {'ner': 32.97153806471643}
Losses: {'ner': 2.39907462387285}
Losses: {'ner': 22.977387887101763}
Losses: {'ner': 4.599581692964621e-06}
Losses: {'ner': 1.9882955234192774e-08}
Losses: {'ner': 5.26225924263131e-09}
Losses: {'ner': 2.2673915734306062e-07}
```

Figure 46: Custom NER losses

```
IT2010 Mobile Application Development MODULE_TITLE
2 SEMESTER
Oct - 2020 PERIOD
4 CREDITS
B+ GRADE
IT2070 CODE
Data Structures & Algorithms MODULE_TITLE
2 SEMESTER
Oct - 2020 PERIOD
4 CREDITS
A GRADE
IT2080 CODE
IT Project MODULE_TITLE
2 SEMESTER
Oct - 2020 PERIOD
4 CREDITS
A GRADE
IT2090 CODE
Professional Skills MODULE_TITLE
2 SEMESTER
Oct - 2020 PERIOD
2 CREDITS
B+ GRADE
```

Figure 47: Tested output of custom NER

5.1.3.2. Pre-trained NER model

The pre-trained Named Entity Recognition (NER) model that was built using the Spacy library was trained with 200 epochs, 128 as the batch size, a learning rate of 0.001, and a loss of 84.75 with a score of 1.00.

E	#	LOSS	TOK2VEC	LOSS_NER	ENTS_F	ENTS_P	ENTS_R	SCORE
0	0	0.00	47.20	0.00	0.00	0.00	0.00	
1	200	258.80	1839.42	99.42	99.47	99.37	0.99	
2	400	13.28	35.24	99.61	99.61	99.61	1.00	
4	600	21.16	30.47	99.64	99.56	99.71	1.00	
6	800	34.70	43.84	99.68	99.66	99.71	1.00	
8	1000	38.80	52.18	99.78	99.81	99.76	1.00	
12	1200	92.93	73.50	99.66	99.66	99.66	1.00	
16	1400	329.55	91.60	99.83	99.85	99.81	1.00	
20	1600	174.63	81.22	99.88	99.90	99.85	1.00	
26	1800	76.73	59.74	99.85	99.85	99.85	1.00	
33	2000	72.33	61.78	99.90	99.90	99.90	1.00	
42	2200	76.58	60.44	99.88	99.81	99.95	1.00	
53	2400	131.30	81.35	99.90	99.90	99.90	1.00	
63	2600	194.17	81.45	99.90	99.90	99.90	1.00	
74	2800	296.60	67.01	99.88	99.81	99.95	1.00	
84	3000	127.52	75.52	99.90	99.90	99.90	1.00	
95	3200	198.12	88.36	99.88	99.81	99.95	1.00	
105	3400	489.22	91.21	99.90	99.90	99.90	1.00	
116	3600	976.41	84.75	99.88	99.81	99.95	1.00	

✓ Saved pipeline to output directory
model-last

Figure 48: Pre-trained NER model losses and score

1T4070*	Preparation for the Professional World	MODULE_TITLE	1 Jun - 2022 2	A GRADE
1T4010	Research Project 2 Dec -	MODULE_TITLE	2022 16	B+ GRADE
174011	Database Administration and Storage Systems	MODULE_TITLE	2 Nov - 2022 4	A- GRADE

Figure 49: Tested output of pre-trained NER

Thus, the custom NER model was chosen over the pre-trained custom NER model as the loss that was obtained after running 10 epochs was low in the custom NER model.

5.1.4. Model Building for Skill Area Categorization

The SVM model demonstrated the highest accuracy among the evaluated models, achieving an impressive accuracy rate of 0.8810. The Random Forest model closely followed with an accuracy of 0.8784, while the XGBoost model achieved an accuracy of 0.8623. Additionally, the Logistic Regression model yielded an accuracy of 0.7524, and the Naïve Bayes model achieved an accuracy of 0.7980. The hyper-parameters were tuned for a better result. Based on these results, the SVM model was determined to be the most effective and reliable model for skill area prediction.

Table 2: Model accuracies of score prediction model

Model	Accuracy
Logistic Regression	75.24%
Naïve Bayes	79.80%
XGBoost	86.23%
Random Forest	87.84%
Support Vector Machine	89.47%

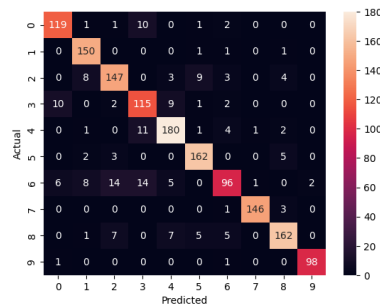


Figure 50: Confusion matrix of the new SVM model

	precision	recall	f1-score	support
Artificial Intelligence and Machine Learning	0.88	0.89	0.88	134
Cloud Computing	0.88	0.98	0.93	153
Cybersecurity	0.84	0.84	0.84	174
Data Science and Analytics	0.77	0.83	0.80	139
Database Management	0.88	0.90	0.89	200
IT Infrastructure and Networking	0.90	0.94	0.92	172
Programming and Software Development	0.83	0.66	0.74	146
Project Management	0.99	0.97	0.98	150
System Administration	0.92	0.87	0.89	187
User Experience and Design	0.98	0.98	0.98	100
accuracy			0.88	1555
macro avg	0.89	0.89	0.88	1555
weighted avg	0.88	0.88	0.88	1555

Figure 51: Classification report of the new SVM model

5.1.5. Graphical representation of skill areas

After categorizing the skills of a candidate, they are visually presented using a graphical representation. In order to visualize, 3 graphical representations were considered. The bar chart, Pie Chart, and Radar graph as shown below.

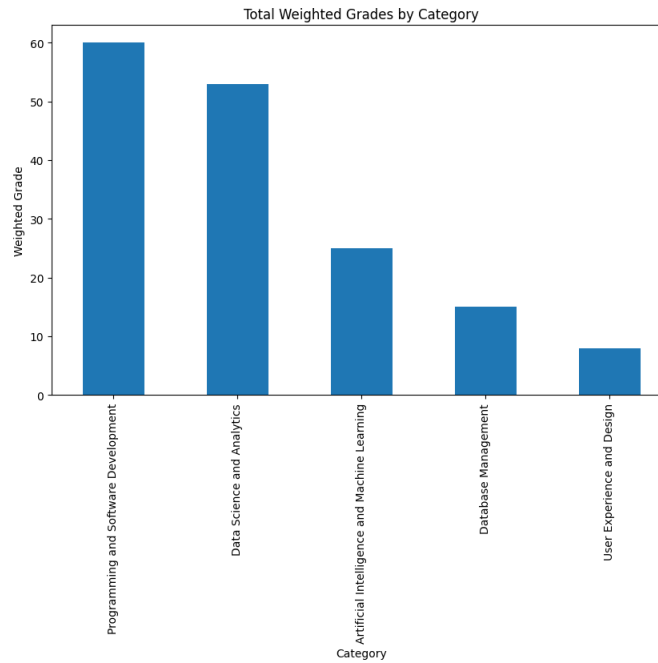


Figure 52: Bar chart representation of skill areas

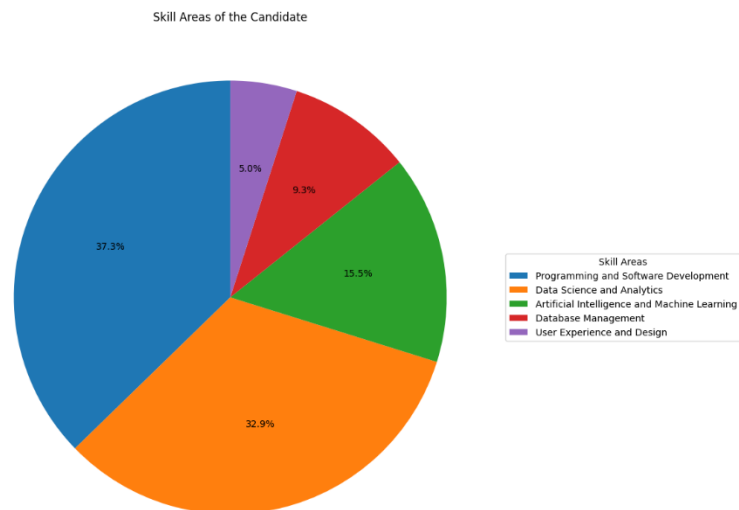


Figure 53: Pie chart representation of skill areas

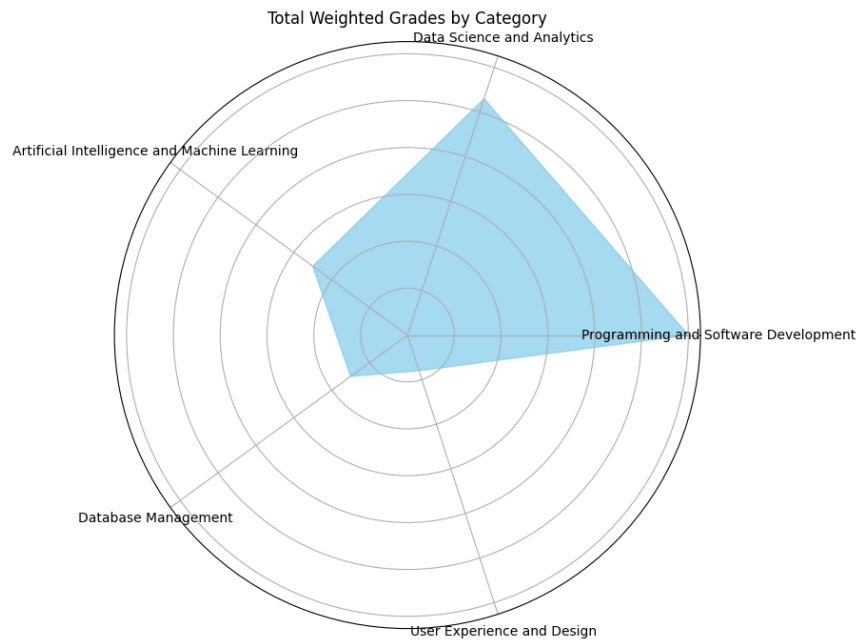


Figure 54: Radar graph representation of skill areas

Here a pie chart is used to visualize the skill areas. The following are some justifications for why this approach was used.

- **Compositional View:** Pie charts excel at showing the relative proportions of different categories within a dataset. In the context of skill areas, a pie chart can provide a quick, at-a-glance view of how various skills contribute to the whole.
- **Simplicity:** Pie charts are straightforward to understand so that the hiring managers can make well-defined decisions.
- **Percentage Representation:** Each "slice" of the pie can be labeled with the percentage it represents, giving the hiring managers an immediate sense of the distribution. As we

need to show the audience the percentages as a whole, a pie chart can be more appropriate.

- Limited Categories: Pie charts work best when you have a relatively small number of categories or skill areas to display. If you have numerous skills, a bar chart or radar graph may become cluttered and less effective.

Reasons to Eliminate the Bar Chart:

- Not Suitable for Composition: Bar charts are better suited for comparing different categories or values, rather than showing the composition of a whole.
- Less Efficient for Few Categories: Since there graph represents a maximum of 10 categories, it would be ideal to display it in a pie chart rather than a bar graph.
- Less Intuitive for Percentages: [9]When using a bar chart, the hiring may need to rely on labels or tooltips to understand the proportion of each skill area relative to the whole, making it less intuitive than a pie chart. Thus, the hiring managers won't be able to get an understanding at a glance as the bar chart does not represent each category.

Reasons to Eliminate the Radar Graph:

- Complexity: Radar graphs are most suitable for comparing multiple data points across different categories simultaneously. [10]After inputting all the measurements onto the chart, the filled-in area can create distortions in the data since the shaded region becomes a visual indicator of magnitude and favorable performance, potentially leading to misinterpretation. If you only want to represent

the distribution of skills without emphasizing the relationships between them, a radar graph can make the data appear unnecessarily complex.

- **Difficulty in Interpretation:** Radar graphs can be challenging for some viewers to interpret, especially if they are not familiar with this type of chart. [10] And also, the distance on the radii is quite difficult to read thus it can be difficult to quantify. They require more cognitive effort to understand compared to a pie chart, which is more straightforward.
- **Not Ideal for Skill Representation:** Radar graphs are often used for multi-dimensional data with varying degrees of performance across multiple attributes. Since the primary objective is to display the distribution of skills, a pie chart will be a more appropriate choice.

Due to these reasons, a pie chart was used to graphically represent the skill areas.

5.1.6. Academic transcript score prediction model

The MSE and R-squared(R2) for each regression model is as follows:

Table 3: Mean square errors of academic transcript score prediction model

Model	MSE	R2 score
Random Forest Regressor	35.21	0.65
Logistic Regressor	48.44	0.54
Support Vector Regressor	60.98	0.47
Gradient boost Regressor	30.67	0.71

Among the various models considered, we can see that the Gradient Boost Regressor has the lowest MSE and thus is taken as the best choice and was selected to predict the score. The model was saved and loaded to ensure its availability for predicting academic scores based on job roles. Candidate's job role, job descriptions, and weighted grades for specific skill categories were taken and finally, by using the model, a score was generated based on the skill proficiency and the position the candidate applied for

Position Applied: Software Engineer

Score: 82.92%

Figure 55: Score based of the skill proficiency and the position that a candidate has applied

5.2. Research Findings and Discussion

5.2.1. Research Findings

This research is a journey to bridge the gap between academic qualifications and professional suitability, aiming to make the process of evaluating job candidates more data-driven and efficient. A comprehensive approach was taken, beginning with the extraction of module keywords and phrases from university course outlines, forming the foundation for our data analysis. Through the use of Optical Character Recognition (OCR) techniques, the extraction of valuable data from academic transcripts was a success. Next, identifying module names and associated grades using a custom Named Entity Recognition (NER) model was also successful.

One of the key findings of this research component was the effectiveness of different machine learning models in categorizing skill areas based on module titles and keywords. We compared several models, including Logistic Regression, Naïve Bayes, XGBoost, Random Forest, and Support Vector Machine (SVM). The results indicated that the SVM model consistently outperformed the others, achieving an accuracy rate of 89.47%. This finding underscores the importance of choosing the right model for the task, as the SVM model excelled in accurately categorizing skills, which is crucial for evaluating job candidates effectively.

In addition to skill categorization and graphical representation, our research ventured into the realm of generating a score based on skill proficiency and job role. The key insight from this part of this research component was the successful development of a predictive model that could generate scores reflecting a candidate's skill proficiency relative to the requirements of a particular job role. The Random Forest Regressor, a powerful machine learning algorithm, was our tool of choice for this task. By inputting a candidate's skill

profile and the skill requirements of a job role into the model, we could produce a numerical score that quantified how well the candidate's skills aligned with the job's demands.

Furthermore, A graphical representation is used to represent the candidate's skill distribution. A pie chart was used as it was an excellent choice for displaying the relative proportions of different skill categories within a candidate's profile.

This innovative approach not only provided a more objective and data-driven assessment of candidates but also demonstrated the potential for automation in the hiring process. It offered a systematic and standardized way to evaluate job applicants, enabling employers to make more informed decisions about candidate suitability. Overall, the research findings in this area emphasized the potential for data-driven academic transcript analysis to revolutionize the recruitment and hiring processes, enhancing objectivity and efficiency in talent acquisition.

5.2.2. Discussion

In this research, a holistic approach was developed to bridge the gap between academic qualifications and professional suitability. Initially, module keywords and phrases from university module outlines are extracted, Optical Character Recognition (OCR) techniques to extract data from academic transcripts, and Custom Named Entity Recognition (NER) model to identify module names and grades. Next, we categorize skill areas using machine learning models and generate proficiency-based scores. Finally, a graphical representation of these skill areas was visualized to assist employers in evaluating candidates.

This research offers several significant social benefits. Firstly, it enhances education by enabling institutions to adapt their curricula to align with industry demands effectively. This leads to graduates who are better prepared for the job market. Secondly, our work contributes to increased employability. By categorizing skills and generating skill profiles, we empower graduates to understand their strengths and areas for improvement, thereby improving their job prospects. Lastly, institutions can make data-driven decisions about curriculum development, course offerings, and career counseling, benefiting both students and educators.

To use this research ethically, it is recommended to protect graduates' data privacy. Data extracted from academic transcripts should be anonymized and stored securely. Informed consent should be obtained from individuals before using their academic data for analysis, ensuring transparency and clear communication about data usage. Additionally, algorithms categorizing skills should be designed to avoid bias and ensure fair representation of all skill areas.

Ethical concerns revolve around avoiding bias and discrimination in hiring and educational opportunities, protecting privacy to prevent data breaches, maintaining academic integrity, and accurately representing skills. Upholding these ethical principles ensures that our research positively impacts education and employment while minimizing potential pitfalls and biases.

In simpler terms, this research helps universities improve their programs to match job market needs, making it easier for students to find jobs. But it's important to use this information carefully and protect students' privacy. We must make sure that the system is fair and doesn't discriminate against anyone. This way, our research can make education and employment better for everyone.

6. CONCLUSION AND FUTURE WORK

In conclusion, this research represents a significant leap towards modernizing the traditional recruitment process in Sri Lanka, particularly in the thriving IT industry. By harnessing the power of Machine Learning, NLP, and Data Analysis, a systematic approach was created to evaluate job candidates through their academic transcripts. This innovation addresses the challenges faced by companies in identifying the best-suited talent amidst rapid industry growth and high demand for new skills.

This approach revolves around the thorough analysis of academic transcripts, unearthing hidden insights from these documents. Through the extraction of module keywords, OCR technology, and custom NER models, we have developed a tool that empowers employers to identify a candidate's strengths and weaknesses accurately. This approach not only streamlines the hiring process but also enhances the likelihood of finding the perfect match for a job role. In the context of Sri Lanka's IT industry, where the talent pool is vast but finding the right fit remains a challenge, our research provides a competitive edge and simplifies hiring, particularly for IT companies.

In terms of future work, while it's currently tailored for one specific university, the system could be extended to analyze academic transcripts from all universities in Sri Lanka. By scaling up to encompass a wider range of educational institutions, we can provide a more comprehensive and inclusive platform for both job seekers and employers across the country. This expansion would enable a more holistic evaluation of candidates, considering diverse academic backgrounds and enhancing the overall efficiency of the recruitment process, not only for IT companies but across various industries in Sri Lanka.

Furthermore, this research isn't just about efficiency; it's about empowering individuals by helping them understand their skill profiles better. By categorizing and visually representing skill areas, this approach benefits both candidates and employers. Candidates gain insights into their abilities, enabling them to make informed career choices, while employers can quickly assess a candidate's suitability. As the IT industry in Sri Lanka

continues to thrive and evolve, this research ensures that the bridge between academic qualifications and job suitability is solidified, ultimately contributing to more efficient talent acquisition and placement procedures.

7. REFERENCES

- [1] "SLASSCOM," 2019/20. [Online]. Available: <https://slasscom.lk/wp-content/uploads/2021/06/State-of-the-industry-report.pdf>.
- [2] "ICTA," 08 August 2019. [Online]. Available: <https://www.icta.lk/news/sri-lanka-aiming-200000-ict-workforce-by-2022/>.
- [3] R. Gajanayake , M. Hiras, P. Gunathunga, E. Supun, A. Karunasena and P. Bandara, "Candidate Selection for the Interview using GitHub Profile and User Analysis for the Position of Software Engineer," in *IEEE*, 2020.
- [4] T. Rahman, J. Nwokeji, . R. Matovu, S. Frezza, H. Sugnanam and A. Pisolkar, "Analyzing Competences in Software Testing: Combining Thematic Analysis with Natural Language Processing (NLP)," in *IEEE*, 2021.
- [5] R. Jayasekara , K. Kariyawasam, S. Jayasinghe, K. Kudarachchi, D. Rajapaksha and S. Thelijjagoda, "DevFlair: A Framework to Automate the Pre screening Process of Software Engineering Job Candidates," in *IEEE*, 2022.
- [6] B. Bannaka, D. Dhanasekara, M. Sheena, A. Karunasena and N. Premadasa, "Machine Learning approach for predicting career suitability, career progression, and attrition of IT graduates," in *IEEE*, 2021.
- [7] R. Day, "Skillsoft global knowledge," 17 August 2020. [Online]. Available: <https://www.globalknowledge.com/us-en/resources/resource-library/articles/the-10-most-important-it-skills-for-2020/#gref>.
- [8] L. Stevens-Huffman, "ITCareerFinder," 18 May 2023. [Online]. Available: <https://www.itcareerfinder.com/brain-food/blog/entry/top-10-it-skills-in-demand-for-2023.html>.
- [9] J. Freeman, "edraw," [Online]. Available: <https://www.edrawsoft.com/chart/difference-bar-pie-chart.html>.
- [10] "Tibco," [Online]. Available: <https://www.tibco.com/reference-center/what-is-a-radar-chart#:~:text=Radar%20Charts%20Can%20Cause%20Occlusion,data%20points%20can%20become%20occluded..>

[1] "harver.com," [Online]. Available: <https://harver.com/blog/time-to-hire/>.
1]

[1] "www.icta.lk," [Online]. Available: [https://www.icta.lk/news/sri-lanka-aiming-](https://www.icta.lk/news/sri-lanka-aiming-200000-ict-workforce-by-2022/)
2] 200000-ict-workforce-by-2022/.

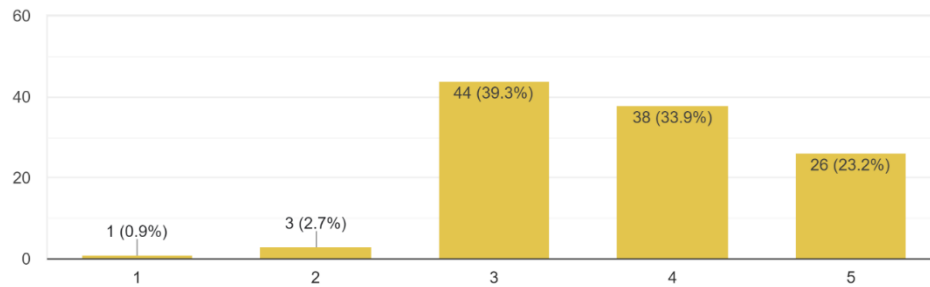
[1] "www.browserstack.com," [Online]. Available:
3] <https://www.browserstack.com/guide/automation-testing-tutorial>.

8. APPENDICES

8.1. Appendix 1 – Survey Questionnaire

6. On a scale of 1 to 5 how important do you think academic performance is when considering a candidate for a job?

112 responses



8.2. Appendix 2 – Turnitin Report

Turnitin Originality Report

[Document Viewer](#)

Processed on: 11-Sep-2023 20:40 +0530
ID: 2041891077
Word Count: 12084
Submitted: 3

Final report draft By Shanali de Silva

Similarity Index		Similarity by Source	
11%		Internet Sources:	5%
		Publications:	2%
		Student Papers:	9%

include quoted	include bibliography	exclude small matches	mode: quickview (classic) report	print	download
4% match (student papers from 03-Sep-2023) Submitted to Sri Lanka Institute of Information Technology on 2023-09-03					
1% match (Internet from 15-Jul-2023) https://www.coursehero.com/file/202995999/IT17027298pdf/					
<1% match (student papers from 07-Oct-2021) Submitted to Sri Lanka Institute of Information Technology on 2021-10-07					
<1% match (student papers from 21-Oct-2020) Submitted to Sri Lanka Institute of Information Technology on 2020-10-21					
<1% match (Internet from 12-Nov-2022) https://www.coursehero.com/file/103994291/COSC3337-Python-Resourcesdocx/					
<1% match (Internet from 01-Mar-2023) http://dl.lib.uom.lk					
<1% match (Internet from 18-Apr-2022) http://dl.lib.uom.lk					
<1% match (Internet from 25-Jun-2022) http://dl.lib.uom.lk					
<1% match (Internet from 07-Sep-2022) https://www.researchgate.net/publication/355927910_Analyzing_Competences_in_Software_Testing_Combining_Thematic_Analysis_with_Natural_Language_Processing_NLP					
<1% match (Internet from 03-Feb-2022) https://www.researchgate.net/publication/346478456_Perceived_Work_Uncertainty_and_Creativity_During_the_COVID-					