A PROJECT REPORT

on

"Resume Screening & Placement Prediction Model"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

AKSHAT	2105944
PUSHKAR	2105957
ISHIKA	2105966
RAJ ARYAN	2105986

UNDER THE GUIDANCE OF
Sourajit Behera

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

November 2024

KIIT Deemed to be University

School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is to certify that the project entitled

"Resume Screening & Placement Prediction Model"

submitted by

AKSHAT	2105944
PUSHKAR	2105957
ISHIKA	2105966
RAJ ARYAN	2105986

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2023-2024, under our guidance.

Date: 23/11/2024

Sourajit Behera
Project Guide

Acknowledgements

We are profoundly grateful to **Sourajit Behera** of KIIT University for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion.

AKSHAT
PUSHKAR
ISHIKA
RAJ ARYAN

ABSTRACT

The Resume Screening and Placement Prediction System leverages machine learning algorithms and natural language processing (NLP) to automate candidate evaluation and placement prediction. Developed with techniques like K-Nearest Neighbours (KNN) and cosine similarity, the system processes resume to identify key skills, experience, and qualifications, matching candidates with suitable job roles. OCR is employed to extract text from PDF resumes, enabling a seamless analysis pipeline. By using advanced feature extraction and keyword identification, the model predicts the likelihood of placement, helping recruiters streamline the hiring process. This tool enhances recruitment efficiency, ensuring a data-driven approach to candidate selection.

Keywords:

1. Resume Screening
2. Placement Prediction
3. Optical Character Recognition (OCR)
4. Machine Learning
5. K-Nearest Neighbors
6. Tesseract OCR
7. Candidate Evaluation
8. Cosine Similarity
9. Feature Score
10. Recruitment Automation

Contents

List of Figures

Chapter 1

Introduction

In recent years, advancements in artificial intelligence and machine learning have transformed traditional recruitment processes, making them more efficient and data-driven. One prominent application is the development of Resume Screening and Placement Prediction Systems, which aim to automate and optimize candidate evaluation. These systems use natural language processing (NLP) and machine learning algorithms to analyze resumes, extract relevant skills and qualifications, and predict candidate suitability for specific roles.

Resume screening traditionally requires extensive manual effort and time, especially in large-scale recruitment scenarios. By integrating techniques like K-Nearest Neighbors (KNN), cosine similarity, and Optical Character Recognition (OCR) for PDF processing, this system efficiently processes and ranks resumes based on predefined criteria. Such an approach enhances decision-making for recruiters by providing a consistent and objective evaluation of candidates, significantly reducing the recruitment time. With potential applications in corporate hiring, academic placements, and talent acquisition, automated resume screening and placement prediction represent a valuable tool for modern recruitment.

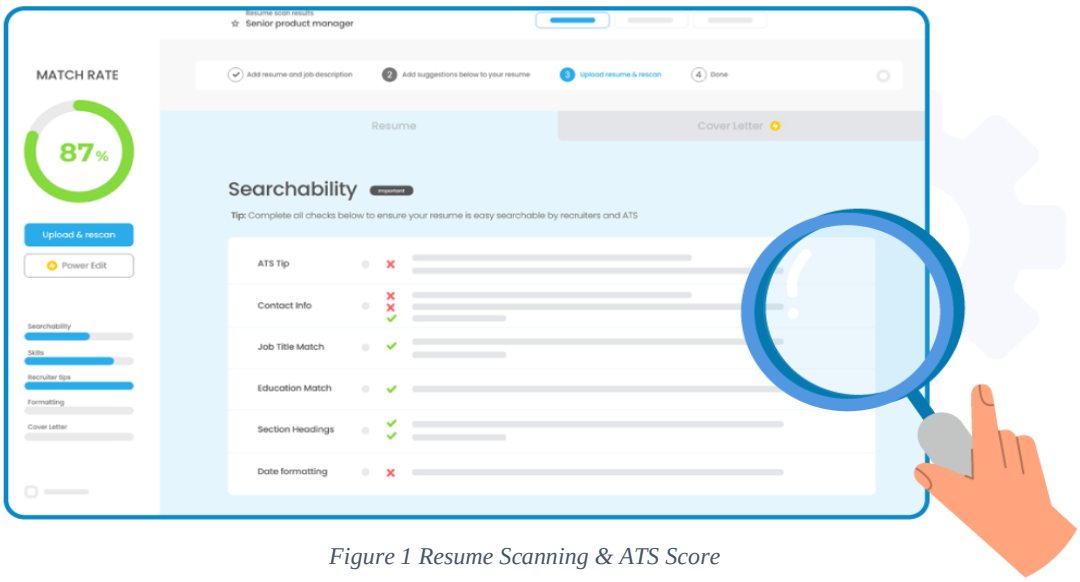

Figure 1 Resume Scanning & ATS Score

Key components of the Resume Screening and Placement Prediction System include machine learning algorithms for candidate evaluation, OCR for extracting text from PDF resumes, and NLP techniques for parsing and analyzing resume content. The system architecture is designed to handle a large volume of resumes efficiently, applying K-Nearest Neighbors (KNN) and cosine similarity measures to match candidates with the most suitable roles based on their skills and experience. This approach allows for consistent, data-driven predictions of candidate placement potential.

To address challenges in automated resume screening, such as variations in resume formatting, language ambiguity, and keyword relevance, the system employs advanced text processing techniques. Feature extraction and keyword identification ensure robust matching between candidate qualifications and job requirements, improving the system's accuracy and reliability. Additionally, a secure web interface can be integrated to enable easy uploading of resumes, real-time screening results, and authorized access to candidate data, ensuring privacy and preventing unauthorized system modifications.

Chapter 2

Literature Review

Automated Resume Screening and Placement Prediction have become pivotal in recruitment, offering efficiency and reducing bias compared to manual processes. Research highlights using techniques like named entity recognition and cosine similarity, paired with algorithms such as K-Nearest Neighbors (KNN), to assess candidate suitability. OCR is critical for extracting text from varied resume formats, while multi-stage frameworks improve accuracy by combining machine learning with industry-specific knowledge. Literature underscores the growing impact of these AI-driven systems in high-volume hiring and campus placements, emphasizing their value in modern recruitment.

1. **Resume Acquisition and Text Extraction**
   - The Resume Screening and Placement Prediction system begins by acquiring resumes, often in formats like PDFs or Word documents. These are uploaded to the system, where Optical Character Recognition (OCR) is applied to convert text from images or non-text-based files.
   - Preprocessing techniques, including text normalization, and standardization, enhance the extracted text quality, facilitating accurate analysis in the next stages.
2. **Skill and Keyword Identification**
   - This step involves identifying and extracting relevant skills, qualifications, and keywords from resumes. Techniques such as tokenization, part-of-speech tagging, and named entity recognition (NER) are used to capture crucial information on skills and experiences.
   - Challenges in keyword identification include handling different resume formats, varying terminologies, and ambiguities in skills representation, which require robust text-processing techniques for consistent results.
3. **Candidate-Job Matching**
   - After extracting keywords and skills, the system matches candidates with job requirements using similarity measures such as cosine similarity and machine learning algorithms like K-Nearest Neighbors (KNN).
   - Matching challenges arise from variations in job descriptions and candidate profiles, necessitating adaptable algorithms that account for diverse experience levels and industry-specific terminology.

4. **Placement Prediction**
   - The system predicts placement potential based on matching scores, providing insights into a candidate's suitability for specific roles. This involves training models on historical data to determine likelihoods of success in given roles.
5. **Applications and Practical Implementations.**
   - Automated resume screening is applied in various sectors, from corporate recruitment to campus placements, saving time and reducing hiring biases. This technology is commonly integrated into applicant tracking systems (ATS) for efficient candidate management.
   - Practical implementations include large-scale hiring scenarios, where the system quickly narrows down candidate pools, and for campus placements, where it streamlines matching students with roles based on their skill profiles.
6. **Challenges and Future Directions:**
   - Despite its progress, automated resume screening faces challenges, such as handling varied resume structures, language nuances, and ethical considerations like ensuring fairness and transparency in the hiring process.
   - Future research is directed towards developing robust algorithms that can better understand resume language, integrating multi-modal data like social profiles, and addressing ethical concerns to foster responsible AI use in recruitment.

Chapter 3

Problem Statement / Requirement Specification

This project aims to develop an efficient and reliable Resume Screening and Placement Prediction system that addresses challenges such as varied resume formats, inconsistent language, and the need for unbiased evaluation. By leveraging advancements in natural language processing (NLP) and machine learning, the system will automate candidate screening, accurately match candidate profiles with job requirements, and handle large volumes of resumes. Ethical considerations, including fairness in selection and data privacy, are central to the project, ensuring responsible technology development. The goal is to enhance recruitment efficiency and candidate-job matching accuracy, contributing to the adoption of AI-driven hiring tools in real-world applications.

3.1 Project Planning

3.1.1 Effort Estimation

This project was completed over a span of 2.5 months, with the majority of the effort dedicated to data collection, preprocessing, and model development. Significant time was spent ensuring data quality and performing feature engineering to optimize the model's performance. The literature review and report writing were completed within a reasonable timeframe, providing a strong foundation for the project and ensuring clear communication of the findings. The effort estimation aligned well with the actual project duration, demonstrating effective time management and efficient allocation of resources throughout the development process.

3.1.2 Project Schedule (Gantt Chart)

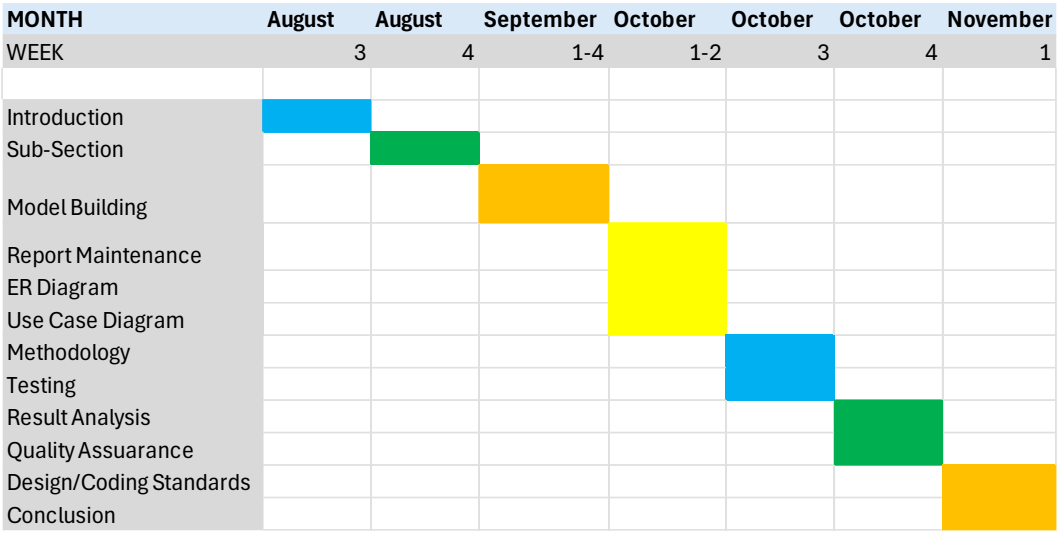

Figure 2 Gantt Chart

3.1.3 Staffing

In the Resume Screening and Placement Prediction project, tasks were allocated based on team members' areas of expertise to ensure efficiency and high-quality deliverables:

- Data Collection and Preprocessing: A separate team focused on collecting and processing resumes, ensuring that high-quality and relevant data was gathered. They worked closely to preprocess and clean the data, preparing it for model training.
- Model Development and Evaluation: Some members specialized in machine learning, focusing on the development, training, and evaluation of the prediction models. Their expertise ensured the algorithms were optimized for accuracy and efficiency.
- Project Reporting and Documentation: A dedicated team handled the documentation and reporting. They worked on compiling the project's methodologies, progress, and findings, ensuring clear communication of the project's objectives and outcomes.

3.1.5 Risk Management

Identified Risk:

1. Data Quality Concerns: Variations in resume formats and quality could affect data extraction.
   Mitigation: Standardize preprocessing techniques to ensure consistent data input.

2. Algorithm Performance Constraints: The models may struggle with diverse or incomplete resume data.
   Mitigation: Test multiple resume samples.
3. Technical Constraints: Issues with software, hardware, or code implementation could hinder functionality.
   Mitigation: Conduct thorough testing across environments to identify issues as early.
4. Project Timeline Constraints: Delays in data collection or model training could impact the schedule.
   Mitigation: Monitor progress regularly and allocate buffer time for unexpected delays.
5. Expertise Gaps: Lack of experience in machine learning could affect system performance.
   Mitigation: Provide team training and encourage collaboration to address knowledge gaps.

Time Management:

1. Agile Approach: Adopt an agile project management methodology to develop and test the system iteratively, providing flexibility to accommodate changing requirements and minimize delays.
2. Task Prioritization: Focus on high-priority tasks such as data preprocessing, feature engineering, and model training, ensuring critical components are completed first to maintain momentum.
3. Continuous Monitoring: Regularly track the project's progress, identify potential roadblocks early, and adjust timelines or strategies as needed to ensure the project stays on schedule for timely delivery.

3.2 Project Analysis

3.2.1 Purpose

The purpose of this document is to define the software specifications for building a Resume Screening and Placement Prediction System. The system will leverage machine learning models to analyze and predict job placement outcomes based on input resume data.

3.2.2 Scope

Users can upload resumes to extract their roles. A machine learning model will analyze this data to predict placement success, offering insights and recommendations for recruiters and candidates.

3.2.3 Functional Requirement

User Interface:

1. Resume Upload: Users will upload resume containing relevant candidate information via the web interface.
2. Placement Prediction: User will give the input such as cgpa & number of internships to predict whether he/she will be placed or not.

Recognition Process:

1. KNN Model: The system shall use the KNN model to analyze uploaded resumes and assess their suitability for desired job positions, as it demonstrated higher accuracy compared to other models.
2. OCR for text recognition: The system shall employ OCR techniques to extract text from uploaded resume files, ensuring accurate and efficient data extraction for analysis.

3.2.4 Non-Functional Requirements

Performance

1. The system shall provide rapid results for resume analysis and job placement prediction.
2. KNN model should be optimized to ensure high accuracy and efficiency in processing resumes.

Usability

The web interface shall be intuitive, with clear instructions for uploading resumes, making it accessible to non-technical users.

Reliability

The system shall handle unexpected errors gracefully, ensuring minimal disruption to the recognition process and user experience.

Conclusion

This project analysis defines the functional and non-functional requirements for developing a Resume Screening and Placement Prediction System, utilizing different machine learning models.

3.3 System Design

3.3.1 Design Constraints

Hardware

The system for resume screening will primarily be run in a cloud environment, specifically Google Colab. Therefore, hardware constraints related to physical devices are less significant. However, it is essential to note that a stable internet connection is required to interact with the cloud-based resources effectively. The system will process resume data uploaded in PDF format, and Google / Colab provides the necessary computational resources, such as CPU and GPU, for model training and evaluation.

System: AMD Ryzen 7 5700U 1.8GHz
SSD: 512 GB
Display: 14”
Ram: 16 GB
OS: Ubuntu / Windows 10,11
Cloud Platform: Google Colab
Internet: Stable connection for accessing Google Colab and loading datasets

Software

Machine Learning Libraries: Google Colab supports Python libraries such as TensorFlow and scikit-learn for training machine learning models. Additionally, libraries like PyPDF2 and Tesseract OCR can be used to extract text from resumes in PDF format.

Data Analysis Tools: Libraries like Pandas and NumPy will be utilized for data manipulation, ensuring the extracted resume data is pre-processed and ready for machine learning.

Visualization Tools: Matplotlib and Seaborn will be used for visualizing the data distribution, model performance, and training results directly in the Colab environment.

Report Writing Software: MS Word, MS Excel for charts.

3.3.2 System Architecture

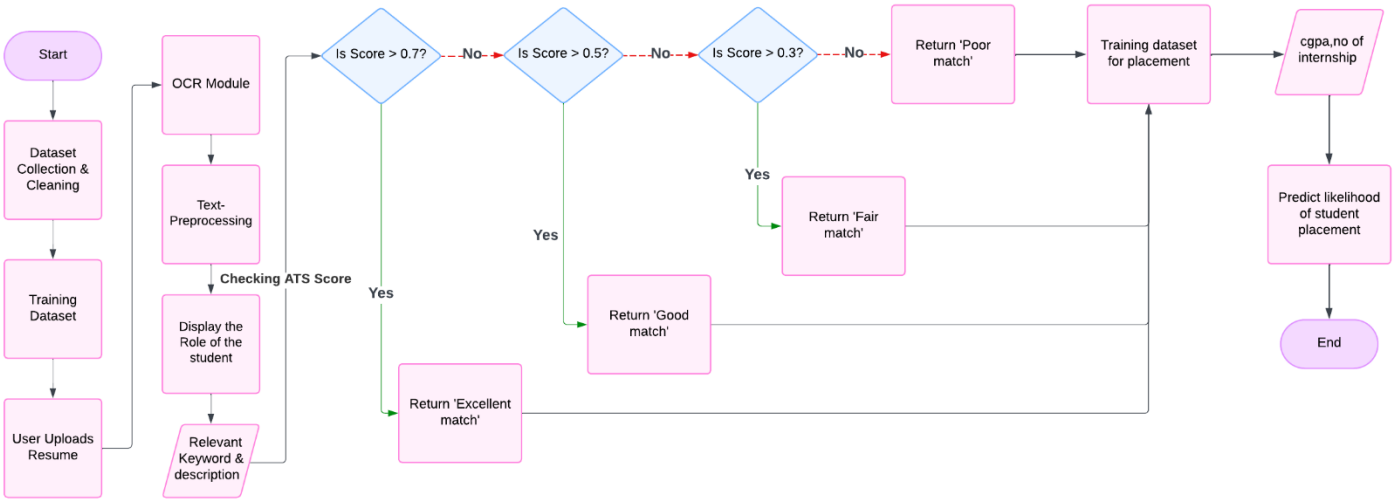

Figure 3 System Architecture

3.3.3 Use Case Diagram

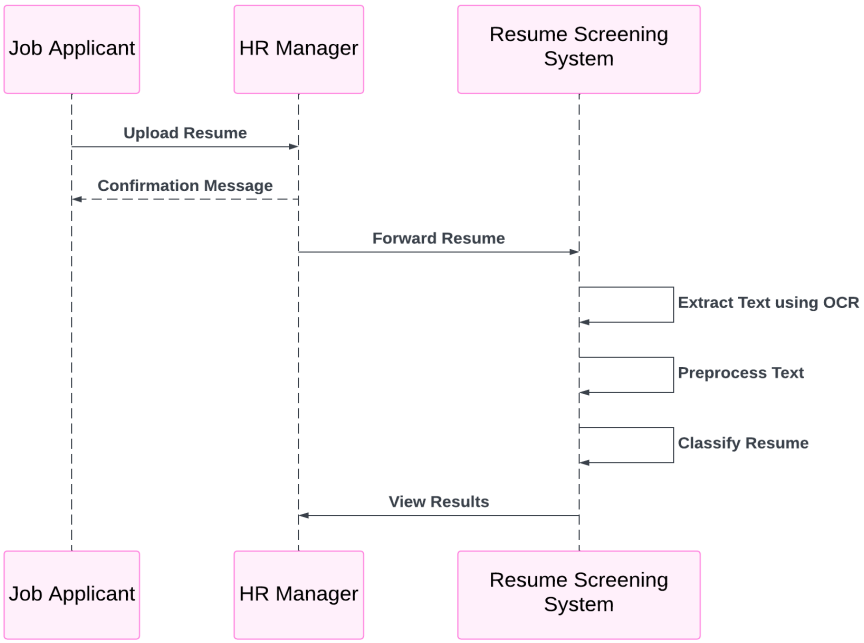

Figure 4 Use Case Diagram

Chapter 4

Implementation

4.1 Methodology Proposal:

The proposed methodology integrates principles from Natural Language Processing and Machine Learning to automate the resume screening process. This approach leverages advanced algorithms and techniques to extract key features from resumes and predict placement suitability based on predefined criteria. Below is an outline of the steps adopted for completing the project work.

4.1.1 Data Collection

Resume Dataset Collection: A large collection of resumes in PDF format is gathered from various publicly available datasets, career websites, and generated datasets to train the machine learning models.
Diversity in Data: The collected dataset is curated to include a diverse range of resumes in terms of experience levels, industries, formats, and educational qualifications to ensure that the model generalizes well to various types of resumes.

4.1.2 Preprocessing

- Text Extraction: Resumes in PDF format is processed using tools like Tesseract OCR to extract text.
- Data Cleaning: Irrelevant content like headers and special characters is removed.
- Feature Extraction: Key features such as cgpa, no of internship are extracted using method like TF-IDF.

4.1.3 Model Selection

K-Nearest Neighbors- KNN model is a simple, non-parametric machine learning algorithm used for classification and regression tasks. The KNN model analyzes resumes by comparing them to a dataset of previously labeled resumes or job descriptions. It classifies resumes based on their similarity to the "K" nearest neighbors in the feature space.

Logistic Regression (LR): Logistic Regression is a supervised machine learning algorithm commonly used for binary classification tasks. LR uses a linear equation to estimate probabilities, applying the logistic function to ensure outputs are between 0 and 1. The result indicates the likelihood of placement success or fit for a specific job role.

Random Forest Classifier: It is a powerful and versatile machine learning algorithm that builds multiple decision trees during training and combines their outputs to improve classification accuracy. Combines predictions from multiple decision trees to make a final decision based on majority voting.

Support Vector Classifier (SVC): It is a machine learning algorithm from the Support Vector Machine family, used for classification tasks. It works by finding a hyperplane that best separates data points of different classes.

4.1.5 Model Training

- Model Training: Train the KNN model, optimize "k", and use cross-validation for the best performance.
- Evaluation & Optimization: Evaluate model performance with metrics like accuracy, precision, and recall.
- Integration: Deploy the trained KNN model for real-time resume screening and placement prediction.

4.1.6 Model Evaluation

Model evaluation for the resume screening and placement prediction system involves assessing accuracy, precision, recall, and F1 score to evaluate prediction performance. Cross-validation is used to validate the model's generalization across different resume datasets.

4.2 Testing or Verification Plan

Alpha Testing:

- Conducted internally by the development team before releasing the software to external users.
- Focuses on identifying and fixing issues within the software before it reaches a wider audience.

4.3 Screenshot or Result Analysis

```
Please upload your resume in PDF format:
Choose files  No file chosen        Upload widget is only available when the cell has been executed
Saving Sample Resumes5.pdf to Sample Resumes5.pdf
Given Resume is for: Sales
```

Figure 5 Upload Resume & predict job role

```
Enter relevant keywords for filtering (comma-separated): Sales, Business Development, Lead Generation, Customer Relationship Management, CRM, Negotiation, Account Management,
Enter the job description of the company: Key Responsibilities:  Sales Leadership: Lead and motivate a sales team to achieve sales targets. Business Development: Identify and
ATS Score (Cosine Similarity): 0.58
Interpretation: Good match
Relevant Missing Keywords: None

Your resume already covers the key skills and qualifications from the job description.
```

Figure 6 Checking ATS score

```
# Calculate accuracy for each model
for model_name, model in models.items():
    predictions = model.predict(X_test)
    accuracy = accuracy_score(y_test, predictions)
    model_accuracies[model_name] = accuracy

# best model based on accuracy
best_model_name = max(model_accuracies, key=model_accuracies.get)
best_model = models[best_model_name]
print(f"The best model is: {best_model_name} with accuracy: {model_accuracies[best_model_name]}")
```

```
The best model is: KNN with accuracy: 0.8417508417508418
```

Figure 7 Selecting Best Model

```
Enter the student's CGPA: 7
Enter the number of internships: 2
The student is likely to get placed.
```

Figure 8 Predicting Placement

Chapter 5

Standards Adopted

5.1 Design Standards

5.1.1 IEEE Standards

- IEEE 802 Series: Ensures compatibility with network protocols for communication between systems.
- IEEE 610.12: Used for documenting software processes throughout development.
- IEEE 1016: Employed for detailed software architecture documentation.

5.1.2 ISO Standards

- ISO 9001: Guarantees consistent quality in software, including testing and validation.
- ISO 14001: Promotes environmentally responsible development practices.

5.1.3 UML Diagram

Use case diagrams represent user interactions and system functionalities for a clear overview of the system.

5.2 Coding Standards

- Adopt consistent naming conventions for variables and functions.
- Organize code into modular components with clear responsibilities.
- Document code thoroughly with inline comments.
- Implement error handling to gracefully manage exceptions and failures.
- Follow a consistent code formatting style and adhere to best practices for readability and maintainability

5.3 Testing Standards

- ISO/IEC/IEEE 29119: Provides a framework for comprehensive software testing.
- IEEE 829: Ensures structured test documentation for thorough validation of system functionality.

Chapter 6

Conclusion

6.1 Conclusion

The development of the resume screening and placement prediction system, leveraging Python, machine learning algorithms, and a web-based interface, represents a significant advancement in recruitment technology. By utilizing KNN and NLP techniques, the system offers an efficient solution for automating resume evaluation, predicting placement outcomes, and streamlining the recruitment process.

Python, coupled with libraries like scikit-learn, provides powerful tools for feature extraction, model training, and prediction. The system preprocesses resume, extracts relevant features such as skills, experience, and education, and then uses machine learning models to predict placement likelihood with high accuracy.

The integration of a user-friendly web interface enhances accessibility, allowing HR professionals and recruiters to easily upload and analyze resumes. This interface offers real-time predictions and insights, improving decision-making in the hiring process.

In conclusion, the combination of Python, machine learning, and web development in the resume screening and placement prediction system demonstrates the effective application of technology in solving HR challenges, enhancing recruitment efficiency, and improving hiring outcomes.

6.2 Future Scope

1. Improved Model Accuracy: Enhance prediction accuracy using advanced machine learning algorithms, such as deep learning and ensemble methods.
2. Real-time Analytics: Integrate real-time data processing for up-to-date resume evaluations and placement predictions.
3. Cloud Integration: Use cloud-based services for scalable storage, computing, and processing, ensuring efficient performance as data grows.
4. Mobile Application: Develop a mobile app to offer recruiters on-the-go access to resume screening results and placement predictions.

References

[1] PDF and OCR Processing in Python [online]
https://pypi.org/project/PyPDF2/
[2] Resume Screening using ML [online: kaggle]
https://shorturl.at/2hGWY
[3] Resume Dataset [online: kaggle]
https://shorturl.at/wT3yH
[4] Placements Prediction [online: kaggle]
https://shorturl.at/pXv1m
[5] PDF to OCR Tutorials [online: GFG]
https://shorturl.at/NGwij
[6] Extracting Text from PDF Files [online: Medium]
https://shorturl.at/RPl3f
[7] Machine Learning Tutorial [online: GFG]
https://tinyurl.com/pd8zc57b
[8] Machine Learning Models [online: JavaPoint]
https://tinyurl.com/mrb5mc5d
[9] Training Datasets for ML [online: encord]
https://tinyurl.com/2p8xjvrv
[10] Google Colab [online]
https://tinyurl.com/4dprw5vf
[11] Campus-Placement-Prediction [online: GitHub]
https://tinyurl.com/3reh7kaj
[12] Lucid Chart [online]
For Use case diagram and system architecture

ORIGINALITY REPORT

16%	12%	5%	12%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to KIIT University Student Paper	4%
2	www.coursehero.com Internet Source	2%
3	Submitted to Caledonian College of Engineering Student Paper	1%
4	Submitted to Middlesex University Student Paper	1%
5	Submitted to University of West London Student Paper	1%
6	Submitted to Liverpool John Moores University Student Paper	1%