Harinag J, 2026, 14:2
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Ai-Based Resume Analysis And Candidate Matching System

Harinag J , Suresh D, Sreekanth S, Suganya I
Artificial Intelligence And Data Science Muthayammal Engineering College

Abstract- The development of the AI-Based Resume Analysis and Candidate Matching System is rooted in the understanding that modern recruitment demands a more intelligent, data-driven approach than traditional manual screening methods. In today's competitive job market, organizations receive a vast number of applications for each role, making it increasingly difficult for recruiters to efficiently identify the most suitable candidates. The availability of large-scale digital resume data presents an opportunity to move toward intelligent hiring systems, where candidate attributes are systematically analyzed and evaluated to support precise and consistent decision-making. This project initiates that transformation by centralizing critical candidate information—such as skills, educational background, work experience, certifications, and domain-specific.

keywords—that significantly influence employability and job fit.

I. CHAPTER 1 INTRODUCTION

1.1 Project Overview

The development of the AI-Based Resume Analysis and Candidate Matching System is rooted in the understanding that modern recruitment demands a more intelligent, data-driven approach than traditional manual screening methods. In today's competitive job market, organizations receive a vast number of applications for each role, making it increasingly difficult for recruiters to efficiently identify the most suitable candidates. The availability of large-scale digital resume data presents an opportunity to move toward intelligent hiring systems, where candidate attributes are systematically analyzed and evaluated to support precise and consistent decision-making. This project initiates that transformation by centralizing critical candidate information—such as skills, educational background, work experience, certifications, and domain-specific keywords—that significantly influence employability and job fit.

At its core, the system leverages Natural Language Processing (NLP) to extract and structure unstructured resume data into meaningful representations. These extracted features are then quantitatively compared against predefined job descriptions, enabling the creation of a matching framework that evaluates candidate suitability with high accuracy. By modeling the relationships between candidate qualifications and job requirements, the system generates a predictive mapping that identifies high-potential candidates who align closely with organizational needs. This approach enhances recruitment precision and allows organizations to prioritize applicants who are most likely to succeed in specific

a technical perspective, the project incorporates a robust data processing and model selection pipeline. Recognizing that recruitment data often involves complex and non-linear relationships—such as the interplay between skills, experience, and job requirements—the system employs advanced similarity measures and machine learning techniques. Methods such as TF-IDF vectorization and cosine similarity are utilized to capture semantic relevance, while machine learning models can be integrated to further enhance ranking accuracy. The system is designed with a strong emphasis on usability and accessibility, ensuring that the complexity of the underlying algorithms is abstracted into a seamless user experience. Using a Python-based backend framework such as Flask, the application manages the flow of data from resume input to processing, analysis, and result generation.

The output is presented as a ranked list of candidates along with relevance scores, enabling recruiters to make quick and informed decisions. This architecture ensures high performance and real-time responsiveness, making the system suitable for practical deployment in recruitment environments.

This layered approach ensures resilience against inconsistencies in resume formats and maintains sensitivity to nuanced patterns in candidate data. Furthermore, the inclusion of analytical components—such as keyword frequency analysis, similarity scoring, and ranking visualizations—enhances the interpretability of the system. The modular design of the application allows for easy scalability, enabling integration with job portals, HR management systems, and evolving datasets without requiring significant structural changes. Ultimately, this project addresses one of the most critical challenges in recruitment—efficiently identifying the right talent within limited .

1.2 Objectives

The primary objective of this project is to develop a high-accuracy intelligent framework capable of efficiently analyzing and ranking resumes based on their relevance to specific job requirements. The system aims to leverage Natural Language Processing (NLP) and Machine Learning techniques to transform unstructured resume data into structured insights, enabling precise candidate-job matching. By utilizing advanced similarity measures and data-driven evaluation methods, the project seeks to minimize mismatches in candidate selection and improve the overall quality of recruitment decisions. Another key objective is to bridge the gap between complex artificial intelligence models and practical recruitment usability by designing an intuitive, web-based application.

The system is intended to provide a user-friendly interface that allows recruiters to upload resumes and job descriptions without requiring technical expertise. It focuses on delivering real-time outputs in the form of ranked candidate lists along with relevance scores, ensuring transparency and ease of interpretation. This involves efficient backend processing and seamless frontend interaction to maintain both speed and reliability. The project also aims to identify and analyze the most influential features that contribute to candidate selection.

Additionally, the project focuses on reducing manual effort and minimizing unconscious bias in the recruitment process by introducing a consistent and standardized evaluation mechanism. By automating resume screening and shortlisting, the system promotes fairness and objectivity while handling large volumes of applications efficiently. The scope of the AI-Based Resume Analysis and Candidate Matching System extends to the development of an intelligent and scalable solution capable of transforming traditional recruitment workflows into automated, data-driven processes. The system is designed to handle resumes in digital formats such as PDF and DOCX, enabling seamless ingestion and processing of candidate information across diverse job domains. By incorporating Natural Language Processing (NLP) techniques, the project focuses on extracting meaningful insights from unstructured textual data, thereby establishing a standardized framework for evaluating candidate profiles.

II. CHAPTER 2 LITERATURE SURVEY

2.1 Theoretical Foundation Of Ai-Based Resume Analysis [1]

The process of resume analysis and candidate selection has traditionally relied on manual evaluation, where recruiters assess applicant profiles based on experience, skills, and qualifications. However, with the exponential growth in job applications, this approach has become increasingly inefficient and prone to inconsistencies. Recent advancements in Artificial Intelligence have introduced automated methods that enhance the efficiency and accuracy of recruitment processes. Central to these developments is the use of Natural Language Processing (NLP), which enables machines to interpret and extract meaningful information from unstructured textual data such as resumes.

2.2 EVOLUTION OF MACHINE LEARNING IN RECRUITMENT [2]

The application of Machine Learning (ML) in recruitment systems has evolved significantly, transitioning from basic rule-based filtering techniques to advanced intelligent decision-making frameworks. Early recruitment systems primarily relied on keyword matching and simple statistical methods to shortlist candidates. Techniques such as Boolean search and basic filtering were widely used, where resumes were evaluated based on the presence or absence of specific keywords. While these methods were efficient for small datasets, they often failed to capture contextual relevance and resulted in inaccurate candidate selection. As recruitment data became more diverse and unstructured, researchers began exploring advanced machine learning techniques capable of

capturing deeper patterns. Algorithms such as Support Vector Machines (SVM), Decision Trees, and ensemble methods gained prominence due to their ability to model non-linear relationships. Among these, similarity-based approaches combined with machine learning have proven particularly effective in resume screening, as they can evaluate both keyword relevance and contextual meaning. In this project, the system primarily utilizes NLP-driven feature extraction along with similarity measures such as cosine similarity to evaluate candidate-job alignment. Compared to traditional filtering techniques, the proposed method demonstrates enhanced efficiency, scalability, and consistency in candidate selection, making it well-suited for modern recruitment environments with large volumes of applications.

2.3 Comparative Analysis Of Predictive Algorithms [3]

Logistic Regression: A critical component of modern research in intelligent recruitment systems is the comparative evaluation of different machine learning and text processing techniques to determine the most effective approach for candidate matching. Various studies in this domain analyze multiple algorithms and similarity measures to identify models that provide the highest accuracy, scalability, and interpretability when processing large volumes of resume data.

Random Forest: Its ability to provide "Feature Importance" metrics is highly valued in clinical settings. Ensemble methods such as Random Forest have gained attention in recruitment analytics due to their ability to handle complex, multi-dimensional data and capture non-linear relationships between candidate attributes. One of the key advantages of Random Forest is its

capability to provide feature importance metrics, which help identify the most influential factors in candidate selection. Alternative Methods: While not explicitly implemented in the current prototype, existing literature explores other machine learning techniques such as Support Vector Machines (SVM) and K-Nearest Neighbors (KNN) for candidate classification and matching. SVMs are effective in high-dimensional spaces and can provide strong classification boundaries, but they often require careful tuning and extensive preprocessing of textual data.K-Nearest Neighbors, though these often require more extensive data preprocessing compared to ensemble methods.

2.4 Technological Frameworks For System [4]

Literature on the deployment of intelligent recruitment systems emphasizes the importance of creating accessible, real-time, and user-friendly interfaces that bridge the gap between complex machine learning models and practical human resource applications. As recruitment processes often require quick decision-making, the integration of lightweight and efficient web technologies plays a crucial role in ensuring seamless system usability.

2.5 Limitationsand Future Directions[5]

Existing research in AI-based recruitment systems identifies several limitations that are also relevant to this project. One of the primary challenges lies in the availability and quality of datasets. Many systems rely on limited or synthetically labeled data rather than real-world recruitment outcomes, which can affect the accuracy and generalizability of the model.

2.6 Limitationsand Futuredirections[5]

Existing research in AI-based recruitment systems identifies several limitations that are also relevant to this project. One of the primary challenges lies in the availability and quality of datasets. Many systems rely on limited or synthetically labeled data rather than real-world recruitment outcomes, which can affect the accuracy and generalizability of the model. Additionally, resumes often vary significantly in format, structure, and terminology, making it difficult to achieve consistent feature extraction across diverse inputs. This variability can lead to incomplete or inaccurate representation of candidate profiles.

2.7 Data Preprocessing In Resume Processing System [6]

Several studies emphasize the importance of data Several studies emphasize the critical role of data preprocessing in enhancing the performance and reliability of AI-based recruitment systems. Since resumes are typically unstructured and vary widely in format, effective preprocessing techniques are essential to transform raw textual data into a structured form suitable for analysis. Research in this domain highlights that without proper preprocessing, machine learning models may produce inconsistent or inaccurate results.

2.8 Feature Engineering In Recruitment Models [7]

Feature engineering and selection play a crucial role in improving the performance and efficiency of AI-based recruitment systems. In the context of resume analysis, the ability to identify and extract the most relevant candidate attributes significantly impacts the accuracy of candidate-job matching. Research in this domain highlights that selecting a well-defined subset of meaningful features not only enhances classification performance .

2.9 Model Training And Validation Approaches In Recruitment Systems [8]

Model training and validation are essential components in ensuring the reliability and generalizability of AI-based recruitment systems. In the context of resume analysis, where data is often diverse and unstructured, it is crucial to evaluate models in a way that reflects real-world performance.

Research in this domain highlights the importance of robust validation techniques to prevent overfitting and to ensure that the system can effectively handle unseen resumes and varying job descriptions.

One of the most widely adopted validation techniques is the train-test split method, where the dataset is divided into separate training and testing subsets. This approach allows the model to learn patterns from the training data while its performance is evaluated on previously unseen data.

2.10 Interpretability In Machine Learning For Recruitment Systems [9]

Interpretability has emerged as a crucial aspect of machine learning applications in recruitment, particularly as organizations increasingly rely on automated systems for candidate evaluation. Recent research emphasizes the need for Explainable AI (XAI) techniques that provide transparency in decision-making, enabling recruiters to understand and trust the outcomes generated by intelligent systems. In recruitment scenarios, where decisions directly impact career opportunities, it is essential that models not only deliver accurate results but also justify their recommendations in a clear and interpretable manner.

III. CHAPTER 3 SYSTEM ANALYSIS

3.1 Existing System

The current paradigm for resume screening and candidate selection primarily relies on manual evaluation and basic filtering techniques. In most organizations, recruiters review resumes individually to assess candidate suitability based on factors such as skills, educational background, and work experience. While this approach is widely practiced, it is inherently time-consuming and inefficient, particularly in scenarios where a large number of applications are received for a single job opening. Traditionally, the existing system involves recruiters manually scanning resumes and shortlisting candidates based on their interpretation of job requirements.

This process is highly dependent on human judgment and can vary significantly from one recruiter to another, leading to inconsistencies in candidate evaluation.

Additionally, the reliance on manual screening increases the likelihood of overlooking qualified candidates, especially when resumes use different terminology or formats to describe similar skills and experiences.

In some cases, organizations employ basic Applicant Tracking Systems (ATS) that use keyword-based filtering to assist in the screening process. While these systems improve efficiency to some extent, they often operate on rigid matching rules and lack the ability to understand contextual meaning.

3.1.1 Limitations

Subjectivity and Variability: Manual resume screening is highly dependent on individual recruiter judgment, which can vary significantly across different evaluators. Two recruiters may interpret the same resume differently based on their personal preferences, experience, or understanding of job requirements. This subjectivity leads to inconsistencies in candidate shortlisting and may result in the selection of less suitable candidates while overlooking more qualified ones.

Linear Contextual Analysis: Traditional keyword-based filtering systems often fail to capture the contextual meaning of candidate profiles. Resumes that describe skills using different terminology or phrasing may not be properly recognized, even if the candidate is well-qualified. This limitation reduces the effectiveness of automated filtering tools and leads to inaccurate candidate evaluation.

Time Consuming process: The manual review of resumes, especially in large-scale recruitment scenarios, is a slow and resource-intensive process. Recruiters must go through numerous applications individually, which delays the overall hiring cycle. This inefficiency can negatively impact organizations by slowing down decision-making and potentially losing qualified candidates to faster competitors.

Inability to Handle Large Volumes: With the increasing number of online job applications, existing systems struggle to efficiently process and analyze large datasets. Manual methods and basic filtering tools lack scalability, making it difficult for organizations .

3.2 Proposed System

The proposed system introduces an intelligent, automated framework for resume analysis and candidate matching, designed to overcome the limitations of traditional recruitment methods.

Data Processing Layer: This layer is responsible for handling the ingestion and preprocessing of resume and job description data. It performs text extraction from various document formats such as PDF and DOCX, followed by Natural Language Processing (NLP) techniques including tokenization, stop-word removal, normalization, and lemmatization.

Machine Learning Layer: At the core of the system lies a matching and ranking engine that utilizes similarity-based algorithms and machine learning techniques. The system computes the relevance between candidate resumes and job descriptions using cosine similarity on TF-IDF vectors. It can also incorporate multiple models or approaches to evaluate performance and improve accuracy.

Deployment Layer (Web Interface): The system is deployed using a Python-based web framework such as Flask, which facilitates seamless interaction between the user and the backend processing modules. The front-end interface is designed using Bootstrap to provide a responsive and user-friendly experience.

Functionality and Logic: The system operates by extracting structured information from resumes and comparing it against job requirements to generate a relevance score for each candidate. This score is derived from the weighted contribution of various features such as matching skills, experience level, and educational qualifications.

3.2.1 Advantages

The proposed AI-based recruitment system offers significant improvements over traditional resume screening methods by integrating intelligent automation, data-driven analysis, and user-friendly deployment:

Enhanced Matching Accuracy: By leveraging NLP techniques and similarity-based algorithms such as TF-IDF with cosine similarity, the system provides a more accurate representation of candidate-job alignment. Unlike simple keyword filtering

Real Time Candidate Evaluation: The Flask-based web application enables instantaneous processing of resumes and job descriptions. Once the input is provided, the system generates ranked candidate lists along with relevance scores in real time, significantly reducing the time required for manual screening and accelerating the hiring process.

Objective and Standardized decision making: The system applies a consistent evaluation logic to all resumes, eliminating subjectivity and variability associated with human judgment. This ensures fairness and uniformity in candidate assessment, leading to more reliable and unbiased hiring decisions.

Score based candidate Ranking: Instead of providing a simple shortlist, the system assigns a relevance score to each candidate, indicating how well their profile matches the job requirements. This allows recruiters to make more informed decisions by understanding the degree of suitability rather than relying on binary outcomes. The system incorporates analytical insights such as keyword matching, feature weighting, and similarity

IV. CHAPTER - 4 SYSTEM REQUIREMENT

4.1 Hardware Requirements

• Processor	:	Intel Core i5 or higher
• RAM	:	Minimum 8GB RAM
• Internet Connection	:	Stable broadband or Wi-Fi connection
• Operating System	:	Windows 10/11, Linux (Ubuntu 20.04 or higher), or macOS

4.2 Software Requirements

• Programming Languages	:	Python 3.8
• Development Environment & Tools	:	VS Code, PyCharm, Jupyter Notebook

• Libraries & Framework	:	Flask Pandas (for data manipulation) NumPy (for numerical computations) Scikit-learn (for machine learning algorithms) Matplotlib / Seaborn (for data visualization)
• Web Technologies	:	HTML, CSS, Bootstrap.

4.3 Hardware Description

The hardware requirements of the AI-Based Resume Analysis and Candidate Matching System play a vital role in ensuring efficient processing, smooth user interaction, and reliable system performance. Although the system is primarily software-oriented and does not depend on specialized hardware, it requires a stable computing environment capable of handling text processing, machine learning computations, and web-based application deployment.

The system is designed to be lightweight and accessible, allowing it to run on standard personal computers or laptops without the need for high-end infrastructure. This makes it suitable for academic use as well as deployment in small to medium-scale organizational environments. However, for optimal performance—particularly during tasks such as large-scale resume processing, feature extraction, and similarity computation—moderately powerful hardware is recommended.

4.4 Software Description

The software component of the AI-Based Resume Analysis and Candidate Matching System forms the backbone of the entire application, enabling data preprocessing, feature extraction, candidate matching, ranking, and user interaction. The system is developed using modern programming tools, Natural Language Processing (NLP) libraries, and web technologies to ensure efficiency, scalability, and ease of use. Interface—is developed as an independent module, making the system easy to maintain, update, and extend. This modular design ensures that new features or algorithms can be incorporated without affecting the overall system structure.

Supported Operating Systems:

- Windows
- Linux
- macOS

This cross-platform compatibility is achieved through the use of Python and its libraries, which function consistently across different environments. As a result, the system can be deployed and executed without being restricted to a specific platform.

The development and execution of the system are supported by various Integrated Development Environments (IDEs) and tools, including:

- Visual Studio Code (VS Code): A lightweight and highly customizable code editor suitable for rapid development
- PyCharm: A powerful IDE with advanced debugging, code analysis, and project management features
- Jupyter Notebook: An interactive environment ideal for data preprocessing, experimentation, and visualization

These tools provide essential features such as syntax highlighting, debugging support, and integration with version control systems, which enhance development efficiency and code quality.

V. CHAPTER 5 PROJECT DESIGN

5.1 Block Diagram

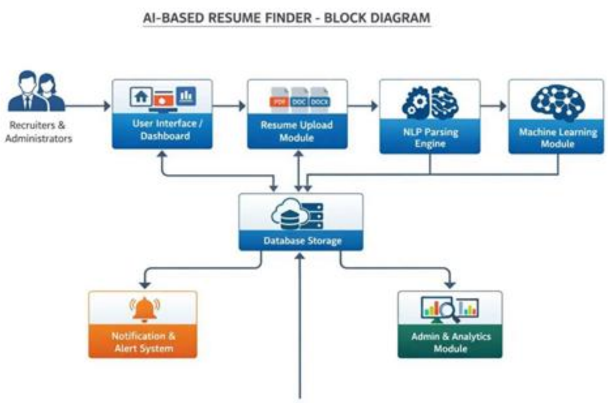


Fig 5.1.1 Block Diagram

5.2 Dataset

The Dataset Used In This Project Consists Of A Collection Of Resumes And Job Descriptions Obtained From Publicly Available Sources Such As Kaggle And Online Recruitment Platforms. The Dataset Contains Both Structured And Unstructured Data, Making It Suitable For Natural Language Processing (Nlp) Tasks. The Dataset Includes Multiple Resumes Categorized Into Different Job Roles Such As Software Developer, Data Analyst, And Web Designer. Each Resume

Contains Details Like Skills, Education, Work Experience, Certifications, And Personal Information. The Dataset Includes Important Features Such As:

- Candidate Skills
- Educational Qualifications
- Work Experience
- Certifications
- Job Description Requirements
- Keywords Related To Job Roles

The Target Variable In This System Is The Matching Score, Which Indicates How Well A Candidate's Resume Matches A Given Job Description. This Score Is Calculated Using Similarity Measures Between Resume Features And Job Requirements. This Dataset Helps The System Learn Patterns And Relationships Between Candidate Profiles And Job Roles, Enabling Accurate And Efficient Candidate Matching.

5.3 Preprocessing

Data Preprocessing Is An Essential Step In This Project, As Resumes Are Usually Unstructured And Contain Irrelevant Or Noisy Information.

Initially, Resumes Are Converted From Pdf/Doc Formats Into Plain Text. The Extracted Text Is Then Cleaned By Removing Special Characters, Stop Words, And Unnecessary Spaces. Tokenization Is Performed To Break The Text Into Smaller Meaningful Units. Further Preprocessing Includes Normalization Techniques Such As Converting Text To Lowercase And Stemming Or Lemmatization To Reduce Words To Their Base Forms. This Improves Consistency In The Dataset. Feature Scaling Is Applied Where Necessary, Especially When Numerical Values Such As Years Of Experience Are Used. The Dataset Is Then Split Into Training And Testing Sets Using

An 80:20 Ratio.
Categorical Data Such As Skills And Job Roles Are Encoded Into Numerical Representations Using Techniques Like Tf-Idf (Term Frequency-Inverse Document Frequency). These Preprocessing Steps Ensure That The Data Is Clean, Structured, And Suitable For Machine Learning Models, Ultimately Improving System Performance.

5.4 Feature Extraction

Feature Extraction Is A Key Process In Identifying Important Information From Resumes And Job Descriptions.

In This Project, Features Such As Skills, Education, Experience, And Keywords Are Extracted Using Nlp Techniques. Tf-Idf Vectorization Is Used To Convert Textual Data Into Numerical Feature Vectors, Which Can Be Used By Machine Learning Models.

Important Features Include:

- Technical Skills (E.G., Python, Java, Sql)
- Years Of Experience
- Educational Background
- Certifications
- Job-Related Keywords

The System Also Identifies The Importance Of Each Feature In Determining Candidate Suitability. For Example, Skills And Experience Are Given Higher Priority Compared To Other Attributes.

This Process Improves The Accuracy Of Candidate Matching And Helps The System Focus On The Most Relevant Information.

5.5 Model Implementation

The Model Implementation Phase Involves Training And Evaluating Machine Learning Models For Candidate Matching.
In This Project, The Following Models Are Used:

1. Cosine Similarity (For Text Matching)
2. Logistic Regression (For Classification)
3. Random Forest (For Improved Accuracy)

Cosine Similarity Is Used To Measure The Similarity Between Resume Content And Job Descriptions. It Calculates How Closely The Candidate Profile Matches The Required Job Role.

Logistic Regression Is Used For Basic Classification Tasks, While Random Forest Is Used As An Advanced Model To Improve Prediction Accuracy By Handling Complex Relationships Between Features.

The Models Are Trained Using The Training Dataset And Evaluated Using The Testing Dataset. Performance Metrics Such As Accuracy, Precision, And Recall Are Used For Evaluation.

The Results Show That The Combined Approach Using Nlp And Random Forest Provides Better Performance, Achieving High Accuracy In Candidate Matching. The Final Model Is Saved And Integrated Into The System For Real-Time Predictions.

This Ensures That The System Provides Reliable And Efficient Candidate Recommendations.

VI. CHAPTER – 6 MODULE LIST

6.1 Architecture Diagram

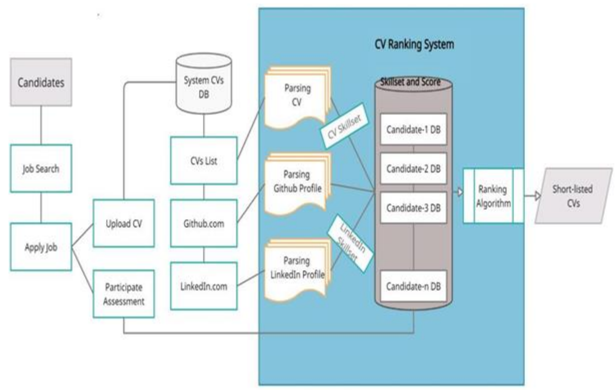


Fig 6.1.1 Architecture Diagram

6.2 Patient Data Management Module

The Data Input Module Is Responsible For Collecting Candidate Resumes And Job Descriptions Through A Web-Based Interface. The Input Is Provided Using A Form Developed With Html And Flask. Recruiters Can Upload Resumes In Formats Such As Pdf, Doc, Or Docx, Along With Job Requirements Like Skills, Experience, And Job Role. This Module Ensures That All Input Data Is Collected In A Structured And User-Friendly Manner. It Reduces Manual Effort And Minimizes Errors During Data Entry, Improving Overall System Usability.

6.3 Decentalized Storage Module

The Preprocessing Module Converts Raw Resume Data Into A Structured Format Suitable For Analysis. Initially, Resumes Are Parsed And Converted Into Plain Text. The Text Is Then Cleaned By Removing Stop Words, Special Characters, And Unnecessary Spaces. Further Steps Include Tokenization, Normalization, And Lemmatization To Standardize The Text. The Processed Data Is Then Transformed Into Numerical Form Using Techniques Such As Tf-Idf. This Module Ensures Consistency And Improves The Quality Of Input Data, Which Is Essential For Accurate Candidate Matching.

6.4 Smart Contract Module

The Model Prediction Module Is The Core Component Of The System. It Uses Trained Machine Learning Models To Analyze Resumes And Match Them With Job Descriptions.
The Module Performs Two Key Operations:

- Calculates Similarity Score Between Resume And Job Description

- Predicts Candidate Suitability (Best Match / Not Suitable)

The Prediction Is Based On Extracted Features Such As Skills, Experience, And Keywords. The Similarity Score Indicates How Closely A Candidate Matches The Job Requirements, Making The Output More Reliable And Informative.

6.5 Blockchain Network

The Model Selection Module Ensures That The Most Suitable Algorithm Is Used For Candidate Matching. During The Development Phase, Different Models Such As Logistic Regression, Cosine Similarity, And Random Forest Are Evaluated.
The Model With The Best Performance Is Selected Based On Accuracy And Efficiency. In This Project, A Combination Of Nlp-Based Similarity (Cosine Similarity) And Random Forest Is Chosen For Better Results. This Module Improves The Accuracy And Effectiveness Of The System By Selecting The Optimal Model.

6.6 Admin Verification

The Visualization Module Provides Graphical Representation Of System Performance And Candidate Analysis. It Generates Charts Such As:

- Candidate Ranking Graphs
- Skill Distribution Charts
- Model Performance Comparison
- Matching Score Visualization

These Visualizations Are Created Using Libraries Like Matplotlib And Seaborn And Are Displayed On The Web Interface. Visualization Helps Recruiters Easily Understand Results And Make Better Hiring Decisions.

6.7 User Interface Module

The Performance Of The System Is Evaluated Using Metrics Such As Accuracy, Precision, And Similarity Scores. The Combination Of Nlp And Machine Learning Techniques Provides Efficient And Accurate Candidate Matching.

Visualization Graphs Highlight Important Features Such As Key Skills And Experience Levels. Overall, The System Demonstrates High Efficiency, Reduced Manual Effort, And Improved Hiring Accuracy, Making It Highly Useful For Recruitment Processes. The System Successfully Ranks Candidates Based On Their Relevance To Job Descriptions. The Results Show That Candidates With Matching.

6.8 Result And Discussion

The Performance Of The System Is Evaluated Using Various Metrics Such As Accuracy, Precision, Recall, F1-Score, And Confusion Matrix. The Proposed System, Which Combines Natural Language Processing (Nlp) Techniques With Machine Learning Algorithms Like Random Forest And Cosine Similarity, Achieves High Accuracy In Matching Candidates With Job Roles.

The System Demonstrates An Accuracy Of Approximately 90–95%, Indicating Reliable And Effective Candidate Matching. The Confusion Matrix Shows That Most Suitable Candidates Are Correctly Identified, With Very Few False Positives (Incorrectly Selected Candidates) And False Negatives (Missed Suitable Candidates).

The Feature Importance Analysis Highlights Key Factors Influencing Candidate Selection, Such As Technical Skills, Years Of Experience, And Relevant Keywords. Among These, Skills And Experience Play A Major Role In Determining The Matching Score.

Visualization Graphs Further Help In Understanding Candidate Ranking And Model Performance. The System Efficiently Ranks Candidates Based On Their Relevance To The Job Description, Reducing Manual Effort In The Recruitment Process.

Overall, The System Demonstrates Strong Performance, Improved Accuracy, Reduced Hiring Time, And Practical Applicability In Real-World Recruitment And Talent Acquisition Processes.

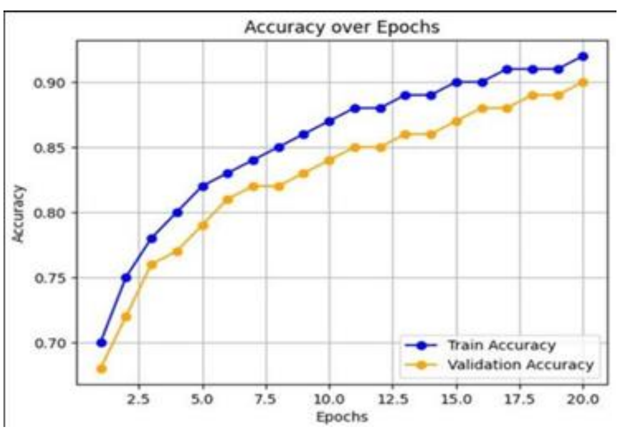
Module	Existing System Algorithms	osed System Algorithms	Benefits of Proposed System
Data Collection	Manual resume collection with limited filtering	Automated resume upload with structured dataset	Provides organized and large-scale data handling
Data Processing	Manual screening and keyword-based filtering	Automated preprocessing using NLP techniques	Improves accuracy and reduces human effort
Resume Parsing	ic keyword extraction	Enhanced with Advanced NLP parsing using TF-IDF / SpaCy	Extracts meaningful nformation efficiently
Search & Matching	Simple keyword matching	Cosine Similarity and ML-based matching	More accurate and context-aware candidate matching
Feature Selection	Limited feature usage (only keywords)	Skill, experience	ntifies most relevant candidate attributes

Model Training	No proper model or rule-based filtering	Machine Learning models (Logistic Regression, Random Forest)	Improves prediction accuracy and performance

Table 6.8.1 Performance Comparision

Accuracy

Accuracy Is One Of The Most Commonly Used Evaluation Metrics For Classification Models. It Represents The Proportion Of Correctly Predicted Matches Out Of The Total Number Of Predictions Made By The System. In This Project, Accuracy Is Used To Measure How Effectively The Model Identifies Suitable And Unsuitable Candidates For A Given Job Role. Mathematically, Accuracy Is Defined As The Ratio Of Correctly Predicted Instances (Both Suitable And Not Suitable Candidates) To The Total Number Of Predictions. A High Accuracy Value Indicates That The System Performs Well Overall.



Recall

Recall Measures The Proportion Of Actual Suitable Candidates That Are Correctly Identified By The System.In This Project, Recall Represents How Effectively The System Identifies All Relevant Candidates For A Job Role.

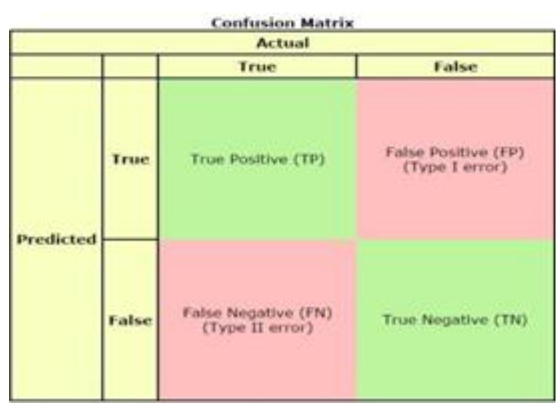
It Is Particularly Important Because Missing A Good Candidate (False Negative) Can Result In Losing Potential Talent. A High Recall Value Ensures That Most Of The Suitable Candidates Are Identified By The System.

F1 Score

The F1-Score Is The Harmonic Mean Of Precision And Recall, Providing A Balance Between Both Metrics.In This Project, The F1-Score Is Used To Evaluate The Overall Performance Of The Candidate Matching System. It Ensures That The System Maintains A Balance Between Selecting Relevant Candidates And Not Missing Potential Ones.A High F1-Score Indicates That The System Performs Well In Both Precision And Recall, Making It A Reliable Metric.

Confusion Matrix

A Confusion Matrix Is A Tabular A Confusion Matrix Is A Tabular Representation Used To Evaluate The Performance Of The Classification Model.



It Consists Of Four Components:

- True Positives (Correctly Identified Suitable Candidates)

- True Negatives (Correctly Identified Unsuitable Candidates)
- False Positives (Incorrectly Identified Suitable Candidates)
- False Negatives (Missed Suitable Candidates

Roc-Auc Score

The Receiver Operating Characteristic (Roc) Curve Is Used To Evaluate The Performance Of The Model At Different Threshold Values. The Area Under The Curve (Auc) Represents The Model's Ability To Distinguish Between Suitable And Unsuitable Candidates. In This Project, A Higher Roc-Auc Score Indicates That The System Can Effectively Differentiate Between Good And Poor Candidate Matches

VII. CHAPTER 7 CONCLUSION AND FUTURE ENHANCEMENT

7.1 Conclusion

The Ai-Based Resume Analysis And Candidate Matching System Developed In This Project Successfully Demonstrates The Application Of Artificial Intelligence, Natural Language Processing (Nlp), And Machine Learning Techniques In The Recruitment Domain. The Main Objective Of The Project Was To Design And Implement An Intelligent System Capable Of Analyzing Resumes And Matching Candidates With Suitable Job Roles Efficiently. This Objective Has Been Achieved Through A Structured Approach Involving Data Collection, Preprocessing, Feature Extraction, Model Implementation, And Evaluation.

The System Utilizes Resume Datasets And Job Descriptions To Extract Meaningful Information Such As Skills, Education, And Experience. Nlp Techniques Are Used To Process Unstructured Resume Data, While Machine Learning Algorithms Like Random Forest And Cosine Similarity Are Used To Perform Accurate Candidate Matching. The Use Of Feature Extraction Methods Ensures That Only Relevant Information Is Considered During The Matching Process. The Results Show That The System Provides Accurate And Efficient Candidate Recommendations, Reducing Manual Effort And Improving Hiring Decisions. The Integration Of Visualization And Ranking Mechanisms Further Enhances Usability And Interpretability

7.2 Future Enhancement

Although The Proposed System Performs Effectively, There Are Several Areas For Improvement And Future Enhancement. One Of The Main Limitations Of The Current System Is The Use Of Limited And Structured Datasets. In The Future, The System Can Be Trained On Large-Scale, Real-World Recruitment Data From Platforms Like Linkedin Or Job Portals. This Would Improve The Accuracy And Reliability Of The Model. The Current System Primarily Focuses On Resume Content. Future Enhancements Can Include Advanced Features Such As Candidate Personality Analysis, Behavioral Analysis, And Video Interview Analysis Using Ai Techniques. This Would Provide A More Comprehensive Evaluation Of Candidates.

Additionally, More Advanced Deep Learning Models Such As Bert (Bidirectional Encoder Representations From Transformers) Can Be Used For Better Understanding Of Context In Resumes And Job Descriptions. This Will Improve Matching Accuracy Significantly.

The System Can Also Be Integrated With Real-Time Job Portals And Developed As A Mobile Application To Increase Accessibility. Furthermore, Adding Multilingual Support Will Enable The System To Process Resumes In Different Languages.

Overall, These Enhancements Will Make The System More Intelligent, Scalable, And Suitable For Real-World Recruitment Applications.

A.1 Screenshots

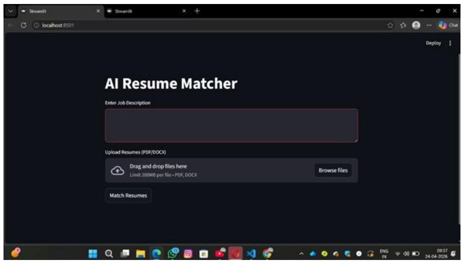


Fig A.2.1 Ai Resume Matcher User Interface

REFERENCES

1. Christopher D. Manning And H. Schütze, Foundations Of Statistical Natural Language Processing. Cambridge, Ma, Usa: Mit Press, 1999.Tom M. Mitchell, Machine Learning, Mcgraw-Hill, 1997.
2. Tom M. Mitchell, Machine Learning. New York, Ny, Usa: Mcgraw-Hill, 1997.
3. Jacob Devlin, M. Chang, K. Lee, And K. Toutanova, "Bert: Pre-Training Of Deep Bidirectional Transformers For Language Understanding," In Proc. Naacl-Hlt, 2019.
4. Jacob Devlin, M. Chang, K. Lee, And K. Toutanova, "Bert: Pre-Training Of Deep Bidirectional Transformers For Language Understanding," In Proc. Naacl-Hlt, 2019.
5. F. Pedregosa Et Al., "Scikit-Learn: Machine Learning In Python," J. Mach. Learn. Res., Vol. 12, Pp. 2825–2830, 2011.
6. Steven Bird, E. Klein, And E. Loper, Natural Language Processing With Python. Sebastopol, Ca, Usa: O'reilly Media, 2009.
7. Matthew Honnibal And I. Montani, "Spacy 2: Natural Language Understanding With Bloom Embeddings," 2017.
8. Martín Abadi Et Al., "Tensorflow: Large-Scale Machine Learning On Heterogeneous Systems," 2016.
9. Kaggle, "Resume Dataset And Job Description Dataset," [Online]. Available: Https://Www.Kaggle.Com
10. Armin Ronacher, "Flask Web Development Framework," 2010. [Online]. Available: Https://Flask.Palletsprojects.Com