International Journal of Science, Strategic Management and Technology
Volume **02** Issue **03** March-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

AI Resume Shortlisting System using Machine Learning

[1]M.Prabhakar,[2]Dr.V,Manimekalai, [1]Student, [2]Associate Professor, [1]B.Sc Computer Technology,
[1]Dr.N.G.P Arts and Science College,Coimbatore,Tamilnadu,India.
Email:[1]prabhakarchandru128@gmail.com,[2]manimekalaiv@drngpasc.ac.in

https://doi.org/10.55041/ijsmt.v2i3.083

Cite this Article: M.Prabhakar, (2026). AI Resume Shortlisting System using Machine Learning. International Journal of Science, Strategic Management and Technology, *02*(03). https://doi.org/10.55041/ijsmt.v2i3.083

ABSTRACT:

Recruitment is a critical activity for organizations seeking qualified employees for various job roles. In modern recruitment environments, companies often receive a large number of resumes for a single job opening, making manual resume screening a time-consuming and inefficient process. Human Resource (HR) professionals must analyze each resume individually to determine whether the candidate satisfies the job requirements. This process can lead to delays in recruitment and may also introduce inconsistencies due to subjective evaluation. This paper presents an **AI Resume Shortlisting System** that automates the initial screening stage of the recruitment process using machine learning and natural language processing techniques. The system analyzes candidate resumes and compares them with job descriptions to determine their relevance for a particular job role. Text preprocessing techniques are applied to extract meaningful information from resumes, and TF-IDF vectorization is used to convert textual data into numerical representations. Cosine similarity is then applied to measure the similarity between candidate profiles and job requirements. The proposed system is implemented as a web-based platform using Python, FastAPI, ReactJS, and MongoDB. The system enables HR professionals to upload multiple resumes simultaneously and automatically generates candidate evaluation results. Based on similarity scores, candidates are categorized into Shortlist, Hold, or Reject groups. The automated screening process significantly reduces recruitment time while maintaining consistent candidate evaluation.

Keywords — Artificial Intelligence, Resume Screening, Machine Learning, Natural Language Processing, TF-IDF, Recruitment Automation**.**

I. INTRODUCTION

Recruitment plays an important role in organizational development because selecting suitable employees directly influences productivity and business growth. Modern organizations frequently receive hundreds of resumes for a single job vacancy. Evaluating these resumes manually requires significant effort and time from HR professionals.

Traditional resume screening involves manually reviewing candidate information such as educational qualifications, technical skills, and work experience. This process is inefficient when dealing with large datasets of resumes. Moreover, manual evaluation may introduce human bias and inconsistencies in candidate selection.

Advances in Artificial Intelligence and Machine Learning have enabled the development of automated systems capable of analyzing large volumes of textual data. Machine learning algorithms can identify patterns in documents and support automated decision-making processes. These technologies provide new opportunities for improving recruitment systems by automating resume screening.

Natural Language Processing (NLP) enables computers to process and analyze textual documents such as resumes and job descriptions. NLP techniques such as tokenization, stop-word removal,

and vectorization allow systems to transform unstructured text into structured representations that can be analyzed by machine learning algorithms [1].

The proposed AI Resume Shortlisting System aims to automate the resume evaluation process by comparing candidate resumes with job descriptions. By calculating similarity scores between resumes and job requirements, the system identifies candidates who best match the desired job profile. This approach improves recruitment efficiency and reduces manual workload for HR departments.

II. RELATED WORK

Several studies have explored the application of machine learning and natural language processing techniques for automated resume analysis and candidate selection. Machine learning algorithms allow systems to learn patterns from data and make predictions based on historical information [2].

Natural Language Processing techniques are widely used for analyzing textual information in documents. These techniques enable computers to extract meaningful features from text and convert them into numerical representations suitable for machine learning models [3].

Data mining techniques are also used in recruitment systems to extract useful knowledge from large datasets. These methods help identify relationships between candidate skills and job requirements [4].

Recent developments in deep learning have further improved document analysis and text processing. Deep learning models can automatically learn hierarchical representations of textual data, enabling more accurate classification and information extraction [5].

Although many automated recruitment systems have been proposed, several existing systems rely only on keyword matching techniques. Such systems may fail to capture contextual relationships between candidate skills and job requirements. Therefore, integrating NLP preprocessing techniques with machine learning similarity algorithms can improve resume analysis and candidate ranking.

III. RESEARCH GAPS

Component Area	***Observed Limitation***	***Identified Gap***
Resume Screening	Manual HR evaluation	Automated AI-based screening
Candidate Ranking	No structured scoring mechanism	Machine learning-based ranking
Recruitment Systems	Slow processing of resumes	Automated high-speed evaluation
HR Decision Making	Subjective judgments	Data-driven candidate analyis

The proposed system addresses these research gaps by implementing an automated resume evaluation platform using machine learning techniques.

IV. SYSTEM OVERVIEW

The overall architecture of the AI Resume Shortlisting System is illustrated in Fig 1. The system enables HR users to log in and upload candidate resumes along with job descriptions. The system processes the uploaded resumes using machine learning techniques and stores evaluation results in a database. Generated reports help HR professionals analyze candidate suitability efficiently.

The system operates as a web-based platform consisting of several interconnected modules including authentication, resume parsing, candidate scoring, and report generation. HR users interact with the system through a user interface that allows them to upload resumes and view evaluation results.

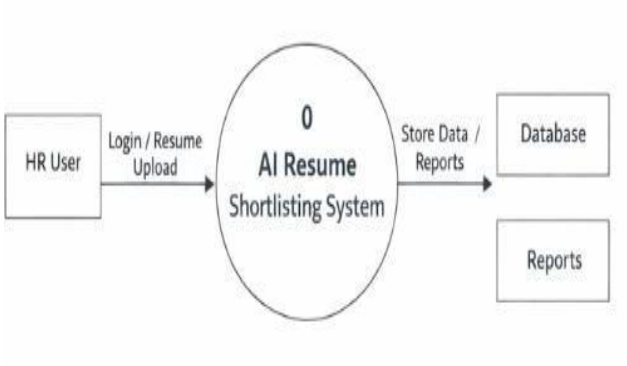

Fig 1. Level-0 Data Flow Diagram of the AI Resume Shortlisting System

V. EXPERIMENTAL METHODS

The system was implemented using modern web technologies and machine learning frameworks. Python was used as the primary programming language due to its extensive support for data processing and machine learning libraries.

The backend server was developed using the FastAPI framework, which provides efficient API development for web applications. MongoDB was used as the database for storing user credentials, candidate profiles, and evaluation results.

Machine learning algorithms were implemented using the Scikit-learn library. TF-IDF vectorization was applied to convert textual information into numerical vectors. Cosine similarity was then used to measure similarity between candidate resumes and job descriptions [6].

The frontend interface was developed using ReactJS to provide a responsive and user-friendly interface for HR users.

VI. WORKING PRINCIPLE

The proposed **AI Resume Shortlisting System** operates through a structured workflow designed to automate the resume screening process. The workflow consists of multiple stages including user authentication, resume submission, text processing, machine learning analysis, and report generation.

Each stage performs a specific task that contributes to the overall evaluation of candidate profiles.

The process begins with **HR user authentication**. When the HR user accesses the system, the login credentials are verified through the authentication module. The system checks the entered email and password with the stored records in the database to ensure that only authorized users can access the platform. Once authentication is successful, the HR user is redirected to the dashboard interface where recruitment operations can be performed.

After successful login, the HR user enters the **job description** for the required position. The job description typically contains details about required technical skills, qualifications, and experience. This information is used as the reference document against which candidate resumes will be evaluated.

Next, the HR user uploads multiple resumes simultaneously through the **bulk resume upload interface**. The system supports common document formats such as PDF and DOCX. Each uploaded resume is transferred to the backend server where the system begins the resume processing stage.

During the **resume parsing stage**, textual information is extracted from the uploaded resume files. The system uses document processing techniques to read the contents of each resume and convert them into plain text format. Important information such as candidate skills, educational qualifications, and work experience is captured during this stage.

Once the resume content is extracted, the system performs **text preprocessing operations** to prepare the data for machine learning analysis. The preprocessing stage includes removing special characters, converting text to lowercase, eliminating stop words, and performing tokenization. Tokenization divides the textual content into individual words or tokens that can be analyzed by the machine learning model. These preprocessing techniques improve the quality of textual data and ensure accurate analysis [1].

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **03** March-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

After preprocessing, the system performs **feature extraction** using the TF-IDF (Term Frequency– Inverse Document Frequency) technique. TF-IDF converts textual data into numerical vectors that represent the importance of words in a document relative to a collection of documents. This transformation allows machine learning algorithms to analyze textual information mathematically [2].

Once the textual features are generated, the system performs **similarity analysis** between the candidate resume vectors and the job description vector. Cosine similarity is used to measure the similarity between these vectors. Cosine similarity calculates the angle between two vectors in a multidimensional space and determines how closely the candidate profile matches the job requirements [3].

The similarity score obtained from this analysis represents the degree of relevance between a candidate's resume and the job description. Based on predefined threshold values, the system automatically categorizes candidates into three groups:

- **Shortlist** – Candidates with high similarity scores who strongly match the job requirements.
- **Hold** – Candidates with moderate similarity scores who may be considered for future evaluation.
- **Reject** – Candidates whose profiles do not sufficiently match the job requirements.

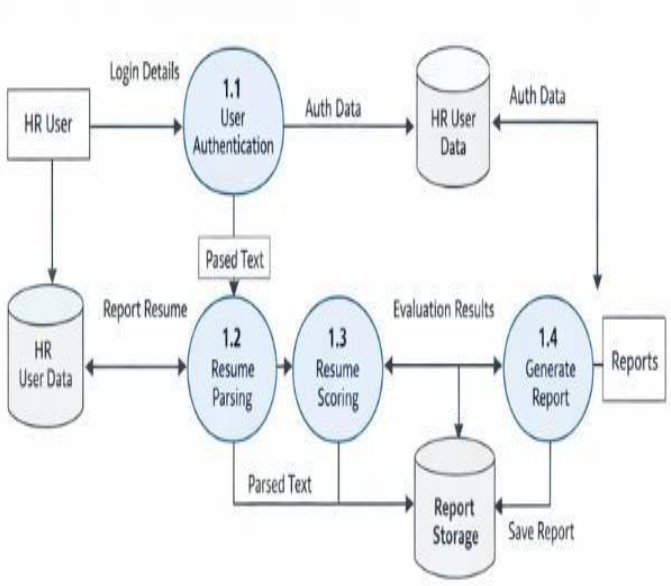

Fig 2. Level-1 Data Flow Diagram Showing Resume Processing Modules

After classification, the system generates a **candidate evaluation report** that includes candidate names, similarity scores, extracted skills, and decision status. The results are stored in the database and displayed to the HR user through the system interface.

To improve usability, the system also provides **visual representation of candidate scores** through graphical charts. These visualizations allow HR professionals to quickly compare candidate performance and identify top candidates for further recruitment stages.

The complete workflow ensures that the system performs resume analysis automatically while maintaining accuracy and consistency in candidate evaluation. By combining natural language processing and machine learning techniques, the AI Resume Shortlisting System significantly reduces manual effort and accelerates the recruitment screening process.

VII. RESULTS AND DISCUSSION

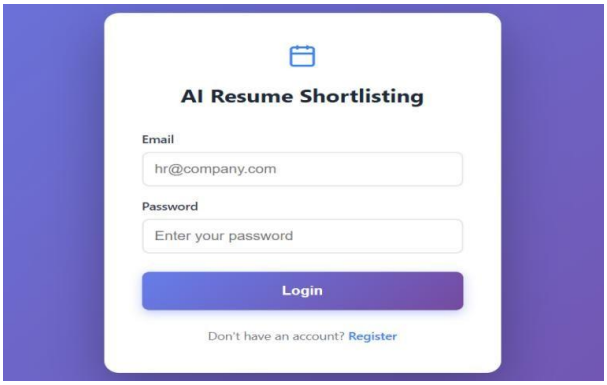

Fig 3. HR User Authentication Interface

The authentication module allows HR users to securely access the system by entering their registered credentials. This mechanism ensures that only authorized users can perform resume evaluation operations.

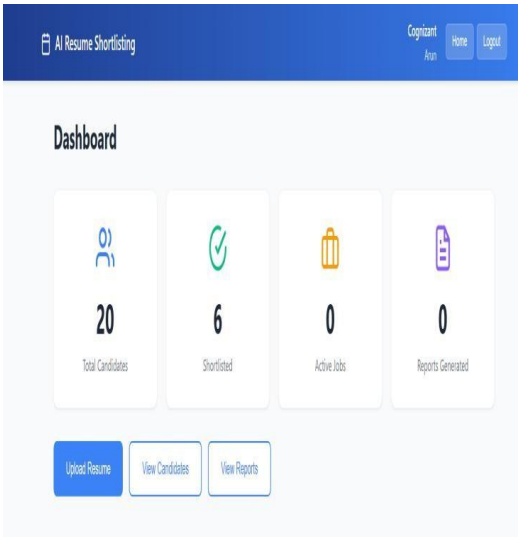

Fig 4. HR Dashboard Showing Candidate Statistics

International Journal of Science, Strategic Management and Technology
Volume **02** Issue **03** March-**2026** | ISSN: **3108-1762 (Online)** | Impact Factor: 3.8
An International, Peer-Reviewed, Open Access Scholarly Journal Indexed in recognized academic databases

After login, the HR user is redirected to the dashboard interface. The dashboard displays key statistics such as total candidates processed,shortlisted candidates, active jobs, and generated reports.

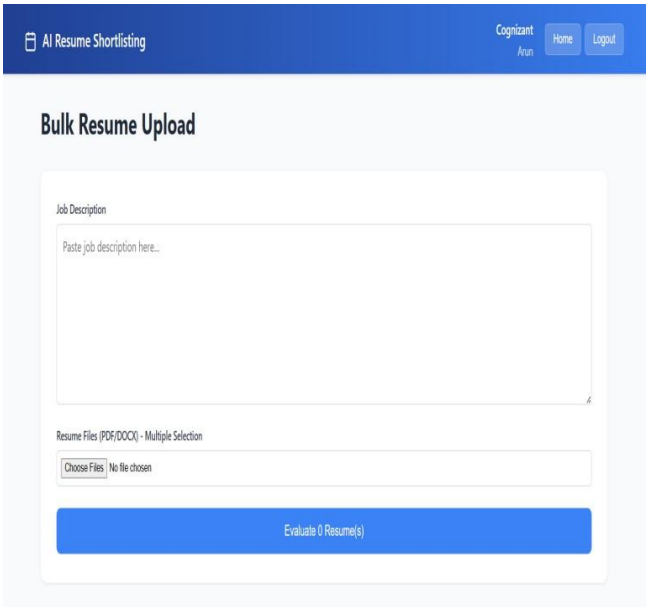

Fig 5. Bulk Resume Upload and Job Description Interface

The system allows HR users to upload multiple resumes simultaneously along with the job description. The uploaded resumes are processed automatically by the backend system.

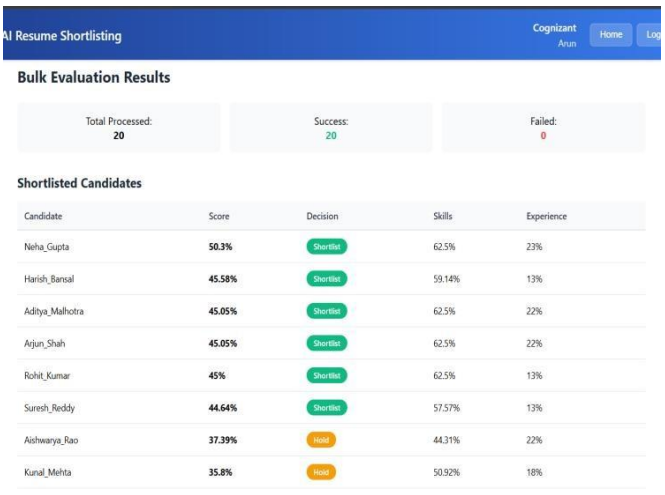

Fig 6. Candidate Evaluation Results

The evaluation results show candidate scores and classification decisions based on similarity analysis. The system successfully identifies candidates who best match the job requirements.

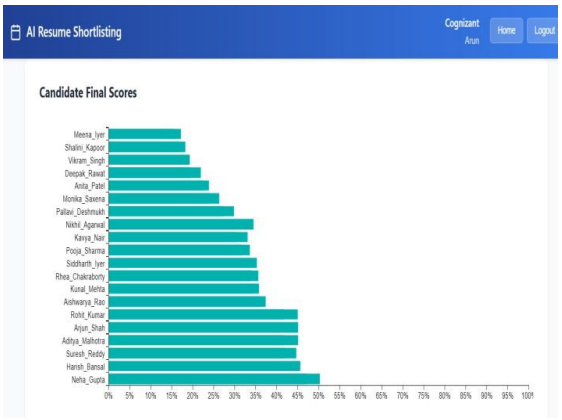

Fig 7. Visualization of Candidate Final Scores

Graphical visualization of candidate scores enables HR professionals to compare candidate performance and identify top candidates quickly. The experimental results demonstrate that the proposed system effectively automates resume screening and significantly reduces the time required for candidate evaluation.

VIII. CONCLUSION

This paper presented an AI Resume Shortlisting System that automates the resume screening process using machine learning and natural language processing techniques. The system analyzes candidate resumes and compares them with job descriptions to identify suitable candidates for specific job roles. By applying TF-IDF vectorization and cosine similarity algorithms, the system provides consistent and objective evaluation of candidate profiles. The automated screening process reduces manual effort and improves recruitment efficiency. The developed system demonstrates how artificial intelligence can support HR professionals in identifying suitable candidates quickly and accurately.

FUTURE ENHANCEMENTS

Future improvements to the system may include:

- Integration with online job portals for automatic resume collection.
- Implementation of deep learning models such as BERT for improved semantic analysis.
- Development of advanced analytics dashboards for recruitment insights.
- Deployment of the system on cloud infrastructure for scalability.
- Integration of automated interview scheduling systems.

REFERENCES

[1] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[2] K. P. Murphy, Probabilistic Machine Learning: An Introduction, MIT Press, 2022.

[3] D. Jurafsky and J. H. Martin, Speech and Language Processing, Pearson, 2023.

[4] J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, Morgan Kaufmann, 2011.

[5] R. S. Pressman and B. R. Maxim, Software Engineering: A Practitioner's

Approach, McGraw-Hill, 2015.

[6] F. Pedregosa et al., "Scikit-learn: Machine Learning in Python," Journal of

Machine Learning Research, 2011.

[7] T. Mikolov et al., "Efficient Estimation of Word Representations in Vector Space," ICLR, 2013.

[8] Python Software Foundation, Python Documentation.

[9] S. Ramírez, FastAPI Documentation.

[10] MongoDB Inc., MongoDB Documentation.

[11] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python,

O'Reilly Media, 2009.

[12] R. Feldman and J. Sanger, The Text Mining Handbook, Cambridge University Press, 2007.

[13] I. Sommerville, Software Engineering, Pearson Education.

[14] G. Salton and C. Buckley, "Term-Weighting Approaches in Automatic

Text Retrieval," Information Processing and Management.

[15] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press.