



International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

AI-Powered Resume Analyzer For Efficient Candidate Shortlisting

Jaddu Jyothi¹, Jada Swapna², Jakka Ramya³, Kutcherlapati Venkata Abhishek Varma⁴, Kuchipudi Sasanth⁵, Mr.S.Chanti⁶

¹student, ²student, ³student, ⁴student, ⁵student, ⁶professor

^{1 2 3 4 5 6} Department of Computer Science and Engineering GMR Institute of Technology, Vizianagaram, Andhra Pradesh, India

ABSTRACT :

Resume screening is the process of reviewing candidates' profiles to find the most suitable person for a job role. Traditionally, this process is manual, time-consuming, and sometimes biased, especially when there are many applications. To overcome these challenges, we propose an AI-Powered Resume Analyzer that automates the resume screening process using machine learning and natural language processing (NLP) techniques. The system extracts and analyzes key information from resumes, such as skills, qualifications, and experience, and compares them with job requirements. Additionally, the system provides feedback to candidates, explaining why their resumes were not shortlisted based on specific factors, helping them improve their future applications. The tool also includes visualizations to display candidate rankings, making it easier for HR professionals to assess resumes. This solution reduces the time and effort involved in screening, ensures fair and unbiased evaluations, and enhances the overall recruitment process by offering valuable insights for decision-making.

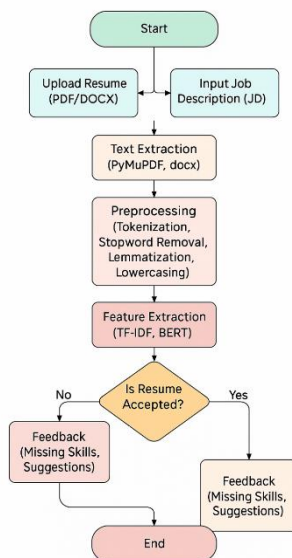
Keywords: Resume screening, Automated recruitment, Natural Language Processing (NLP), Machine Learning, Feedback System, Resume Visualization

INTRODUCTION

The recruitment process is a critical yet time-consuming task for organizations, often requiring the evaluation of numerous resumes to identify the most suitable candidates. Traditional methods of resume screening are prone to human biases, inefficiencies, and errors, making it challenging to ensure fair and accurate hiring decisions. To address these challenges, this project introduces an AI-Powered Resume Analyzer for Efficient Candidate Short listing. The proposed system leverages machine learning algorithms and natural language processing (NLP) techniques to automate the extraction of relevant information from resumes, such as skills, qualifications, and experience, and match them with job requirements. By employing advanced data analysis and ranking methods, the system aims to enhance the accuracy, speed, and reliability of candidate shortlisting. Additionally, it minimizes human intervention, reducing the chances of bias and promoting fairness in the recruitment process.

METHODOLOGY

ARCHITECTURE



Methodology1

B. Surendiran, T. Paturu, H. V. Chirumamilla and M. N. R. Reddy, "Resume Classification Using ML Techniques," 2023 International Conference on Signal Processing, Computation, Electronics, Power and Telecommunication (IConSCEPT), Karaikal, India, 2023, pp. 1-5,

Load Dataset & Preprocess Data:

Collect resumes from job portals, company databases, and email submissions in different formats (PDF, DOCX, TXT).. Perform text cleaning, tokenization, stopwords removal, and transform text into structured features using NLP techniques.

Feature Extraction & Vectorization:

Extract key attributes such as education, skills, and experience, and certifications and convert text into numerical representations using TF-IDF, Word Embeddings (Word2Vec, BERT, etc.)

Train & Evaluate Model:

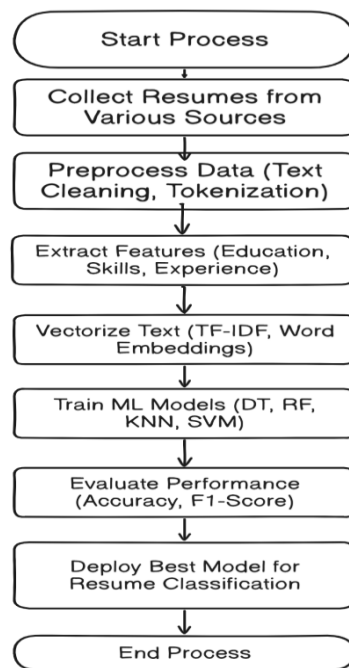
Train multiple ML classifiers (Decision Tree, Random Forest, KNN, SVM). Perform cross-validation and evaluate performance using Accuracy, Precision, and F1-score.

Decision & Feedback:

If a resume meets job criteria, classify it into relevant categories (Shortlisted, Rejected, Consider Later). If rejected, provide feedback on improving resume structure and keyword usage. If rejected, provide constructive feedback on resume structure, formatting, missing skills, or keyword optimization for better job matching.

Visualize Job Requirements & Deploy Model:

Display job trends using bar charts, heatmaps, or pie charts for better insights. Deploy the best-performing model for real-time resume classification in HR systems.



Methodology 2

H. Pendhari, S. Rodricks, M. Patel, S. Emmatty and A. Pereira, "Resume Screening using Machine Learning," 2023 International Conference on Data Science, Agents & Artificial Intelligence (ICDAAI), Chennai, India, 2023, pp. 1-5,

Data Collection and Preprocessing

Resumes are collected from various sources and preprocessed to remove noise, standardize ext, and extract key details such as skills, education, and experience. Job descriptions dergo a similar process to identify key requirements.

Machine Learning-Based Matching

Using NLP techniques, candidate profiles are compared with job descriptions. Machine learning models like Decision Trees, Random Forest, and SVM evaluate candidates based on skill relevance, experience, and education to determine suitability.

Machine Learning-Based Resume Evaluation:

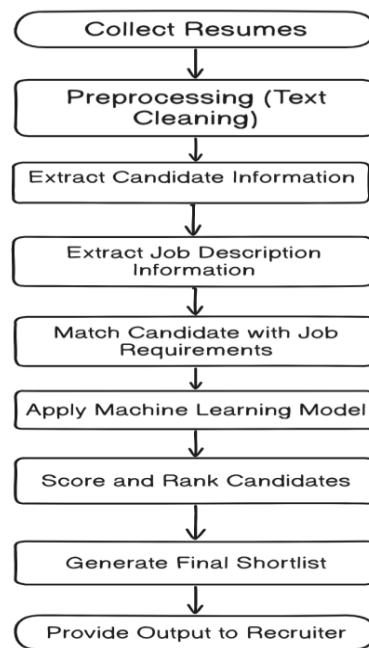
A machine learning model, such as Decision Trees, Random Forest, Support Vector Machines (SVM), or Neural Networks, is applied to automate classification and ranking. These models assess the relevance of a candidate's profile based on skill match, experience level, and educational background.

Candidate Ranking and Shortlisting:

Candidates are scored and ranked according to their suitability for the job. Higher-ranking candidates closely match the job requirements, while lower-ranking ones may need further review, reducing the recruiter's workload and ensuring that only the most relevant profiles move forward.

Recruiter Output and Decision Support:

Provides a structured output to recruiters, presenting a ranked list of candidates along with insights into their strengths and job alignment that streamlines the recruitment process, minimizes human bias, and enhances hiring accuracy through data-driven decision-making.



Methodology 3

E. Salakar, J. Rai, A. Salián, Y. Shah and J. Wadmare, "Resume Screening Using Large Language Models," 2023 6th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2023, pp. 494- 499, doi: 10.1109/ICAST59062.2023.10454984

Data Preprocessing

Extracts text from resumes (PDF files) using a PDF parser. Text is chunked into smaller sections or stored as a full document for efficient processing. Chunking ensures context retention while optimizing retrieval efficiency.

Data Transformation

Tokenization: Breaks text into words, phrases, or sentences.

Vector Embedding: Converts text into numerical representations that capture semantic meaning. Deep neural networks generate high-dimensional vector embeddings for improved resume analysis.

Vector Storage

Vector databases are used to store and retrieve embeddings efficiently. Unlike traditional databases, vector databases allow for semantic information retrieval and long-term memory integration. Enables advanced resume comparison and similarity calculations.

Vector Search & Ranking

- **Distance & Similarity Metrics:**

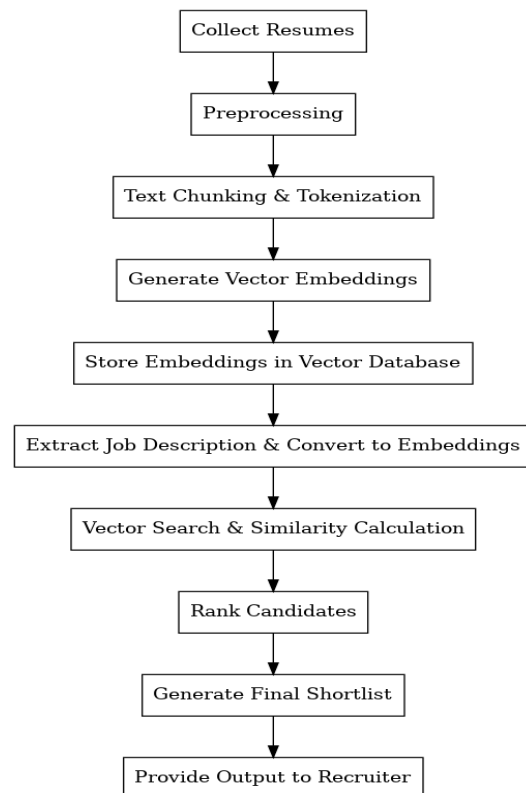
Euclidean Distance (L2) and Manhattan Distance (L1) measure dissimilarity. Cosine Similarity quantifies text-based relevance.

- **Indexing Algorithms:**

Product Quantization (PQ) compresses vectors to reduce memory usage. FAISS (Facebook AI Similarity Search) accelerates query processing with Voronoi cells. HNSW (Hierarchical Navigable Small Worlds) enables fast nearest neighbor search.

System Implementation & Results

Implemented a prototype system using ChromaDB. 135 resumes were tested using a sample query: "Find a Web developer experienced in JavaScript and Python to generate dynamic websites." The system generated vector embeddings for resumes and queries, ranking candidates based on semantic similarity. Candidate No. 15 was identified as the best match, proving the effectiveness of the approach.



Methodology 4

M. F. Mridha, R. Basri, M. M. Monowar and M. A. Hamid, "A Machine Learning Approach for Screening Individual's Job Profile Using Convolutional Neural Network," 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 2021, pp. 1-6, doi: 10.1109/ICSCT53883.2021.9642652.

Data Collection & Preprocessing

Collects job descriptions and resumes from online job portals and databases. Extracts relevant fields like job title, experience, skills, education, and location. Converts textual data into a structured format suitable for machine learning models. Eliminates heterogeneous terminologies by mapping similar job-related terms across different job sites.

Feature Engineering & Data Preparation

Tokenization: Converts resumes into numerical representations. **Vectorization:** Uses word embeddings (Word2Vec/GloVe) to represent job descriptions and resumes in vector format. **Feature Selection:** Identifies key attributes such as experience, education, and skills for ranking profiles.

Convolutional Neural Network (CNN) Model Training

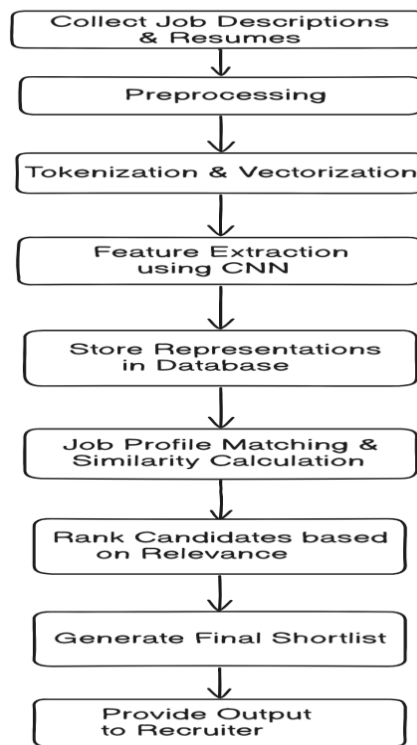
CNN Architecture: Input Layer: Processes text-based job profiles as matrix representations. Convolutional Layers: Extracts essential patterns in resumes. Fully Connected Layers: Determines whether a candidate is selected or rejected. Output Layer: Provides classification results. The model is trained using large datasets of labeled resumes (selected vs. rejected candidates).

Job Profile Matching & Ranking

Keyword Matching: Compares job descriptions with extracted resume features. **CV Ranking:** Calculates similarity scores to rank candidates based on job requirements. **Scoring Metrics:** Uses precision, recall, and F1-score to evaluate ranking accuracy.

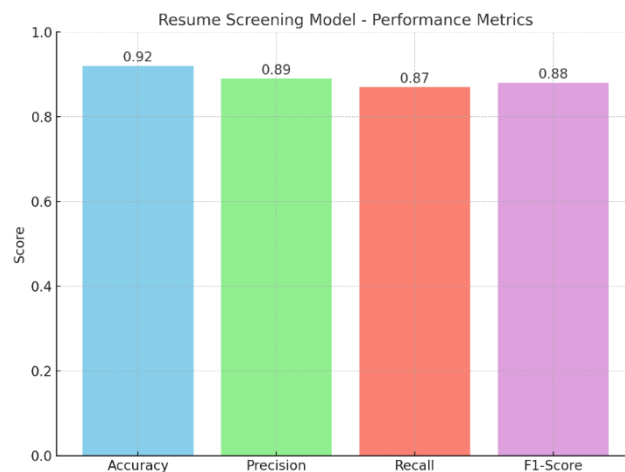
System Implementation & Evaluation

The system is tested on multiple job sites like BDJOBS, Careerjet, LinkedIn, Naukri, and Monster. **Performance Metrics:** Achieves 74% accuracy in job profile screening on BDJOBS. Demonstrates improved classification results over traditional keyword-based methods. This methodology ensures an efficient, automated, and intelligent approach to resume screening using deep learning and NLP techniques.



RESULTS AND DISCUSSIONS:

- Successfully Tested the resumes with job descriptions using:
 - TF-IDF and BERT for feature extraction
 - Random Forest & Ridge Classifiers for prediction
- The model is effectively able to screen resumes against job descriptions with a clear output for HR or recruiters.
- A score above 70% indicates that the candidate possesses most of the required skills, increasing their chances of shortlisting.



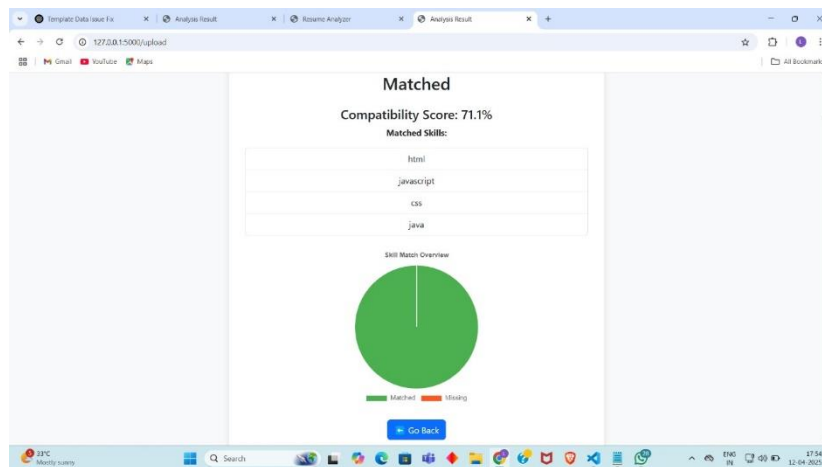
OUTPUTS:

OUTPUT1:

The screenshot shows a web browser window with the URL 127.0.0.1:5000. The page is titled "Resume Analyzer". It features a form with the following fields:

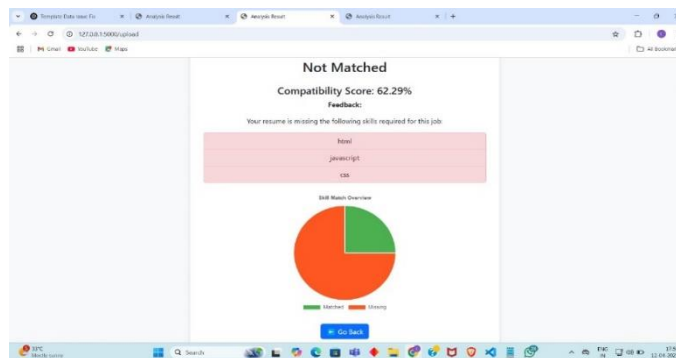
- Upload Resume (PDF):** A file input field with the text "Choose file" and a file named "0575@jyothi.pdf" selected.
- Paste Job Description:** A text area containing the text: "Looking for a Web Developer with strong UI design skills using HTML, CSS, Tailwind CSS and JavaScript. The candidate should be able to deliver visually appealing, responsive interfaces optimized for performance."
- Analyze:** A blue button with a right-pointing arrow.

The browser's taskbar at the bottom shows the system clock as 17:54 on 12-04-2025.



OUTPUT 2:

This screenshot is identical to the first one, showing the "Resume Analyzer" form with the file "0575@jyothi.pdf" and the job description text.



CHALLENGES FACED:

Limited RAM & Processing Power

- While training models or vectorizing text using BERT embeddings, system memory would get exhausted.
- Especially problematic when processing large resume datasets or running simultaneous tasks.

Dataset Creation

- Creating a labeled dataset was difficult due to:
- Lack of standardized resume formats
- Inconsistent structures and fields and manual labelling was time consuming.

SOLUTIONS:

Optimize Resource Usage

Used lightweight models like TF-IDF for quick testing.

Switched to batch-wise vectorization and caching where possible.

Cleared unused variables and optimized memory usage with Python's gc module.

Semantic Similarity

Implemented cosine similarity with TF-IDF initially.

Added BERT embeddings later to improve semantic relevance in JD-resume match.

FUTURE SCOPE:

Multilingual Resume Parsing

Expanding the model to process resumes in different languages would make it globally applicable, allowing companies to source talent worldwide without language barriers.

Skill Gap Analysis & Recommendations

Beyond screening, the tool can identify skill gaps and recommend personalized learning paths for rejected candidates, creating value even for those not shortlisted

CONCLUSION:

The development of our AI-powered Resume Analyzer marks a significant step toward transforming traditional recruitment processes through intelligent automation. By seamlessly combining Natural Language Processing and Machine Learning, our systems not only analyses and interprets resumes but also effectively compares them with job descriptions to determine candidate suitability. The integration of models like Random Forest for classification and Ridge Regression for compatibility scoring enabled our system to deliver a high confidence match, exemplified by a real time test score.

Throughout the project we navigated challenges such as limited computational resources and dataset inconsistencies. These were overcome through efficient memory management, dataset balancing, and the implementation of robust preprocessing and semantic similarity techniques. This project doesn't just showcase our technical proficiency; it represents a scalable, real-world solution with the potential to streamline hiring workflows, reduce bias, and accelerate decision-making in modern talent acquisition.