



# IJRASET

International Journal For Research in  
Applied Science and Engineering Technology



---

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

---

**Volume:** 13      **Issue:** V      **Month of publication:** May 2025

**DOI:** <https://doi.org/10.22214/ijraset.2025.71624>

[www.ijraset.com](http://www.ijraset.com)

Call:  08813907089

E-mail ID: [ijraset@gmail.com](mailto:ijraset@gmail.com)

# High-Scale ATS and Semantic Resume Filtering using Django, NLP, and LLaMA3-GroqModels

Nita Chaudhary<sup>1</sup>, Priyanka Sharma<sup>2</sup>

<sup>1</sup>PGScholar,<sup>2</sup>Assistant Professor, Department of Computer Engineering, Bhagwan Mahavir College of Engineering & Technology, India

**Abstract:** An intelligent system that uses Natural Language Processing (NLP) and Machine Learning (ML) to automate resume classification is presented in this paper. Key resume features, such as education, skills, and job titles, are extracted and used to train models like Logistic Regression, SVM, Random Forest, and BERT. NLP preprocessing techniques, such as tokenization, stop word removal, and vectorization, prepare the text for analysis. The results of experiments show that these models improve classification accuracy and decrease the time needed for hiring. The system assists recruiters by ranking qualified candidates and eliminating applications that are not relevant, making the hiring process quicker and more efficient.

**Keywords:** Human Resource, Goal Oriented, data analysis, Natural Language Processing (NLP), metadata extraction, resumes, CV, large language models

## I. INTRODUCTION

For every job opportunity in today's fast-paced, technologically advanced labor market, companies receive an excessive volume of resumes. Not only is it inefficient and time-consuming to manually review every application, but it is also prone to human biases and inconsistencies. There is now more interest in adopting intelligent technology to automate the employment process as a result of this developing difficulty. The use of Artificial Neural Networks in resume classification has shown promising results with a 94% accuracy in skill prediction.[6]

One promising approach is resume classification, which automatically groups resume into predetermined categories or employment positions. Automated resume evaluation tools are becoming increasingly important in digital hiring ecosystems. Career Mapper is a web-based system that analyzes over 1.6 million LinkedIn profiles to evaluate and provide recommendations for resume improvements. It suggests changes based on frequency and patterns found in professionally successful resumes (Lai et al., 2016). This method increases the objectivity of resume reviews and helps candidates better align their profiles with industry expectations.[7] These systems may evaluate and understand unstructured resume data to find pertinent credentials, abilities, and experience by utilizing Natural Language Processing (NLP) and Machine Learning (ML) approaches.

A machine learning-based resume screening model based on KNN and cosine similarity was proposed in 2021 to achieve high candidate-job matching accuracy. To support their methodology, they also made reference to earlier NLP-based vector space techniques.[8] A ML-based resume screening system that uses NLP, KNN/SVM, and cosine similarity to evaluate job-role compatibility and make improvement recommendations. By integrating GitHub and LinkedIn data into its prediction pipeline, the system was significantly accurate.[9] The purpose of this study is to develop and assess an automated method for classifying resumes. In order to effectively classify resumes, it entails preparing resume texts, identifying significant features, and using supervised machine learning algorithms. The suggested method contributes to a more intelligent and data-driven hiring process by streamlining the recruitment process and improving uniformity and fairness.

## II. LITERATURE REVIEW

Prof. Dikshendra Sarpate, Prarthana Kolhe, Srushti Kalbhor, Sanchi Yehalegaonkar (2024) "AI Enhanced Skill Matcher": In order to overcome issues like subjectivity and time limits, the "Resume Match Predictor" online application automated resume screening using NLP techniques like word2Vec and machine learning. The method improves accuracy, efficiency, and fairness while drastically cutting down on recruitment screening time by including pre-trained language models and concentrating on ethical issues.[1]

Ms. Y. Sowjanya, Mareddy Keerthana, Pulluri Suneeksha, Dorgipati Sai Sri Harsha (2023) "Smart Resume Analyzer":

Resume classification and ranking systems use KNN for categorization and Cosine Similarity to match resumes with job descriptions, enabling sorting based on relevance.

A web application utilizing semi-supervised learning reduces recruiters' workload by automating resume screening. Machine learning-based automation enhances resume recommendation, streamlining shortlisting and decision-making processes for faster and more efficient candidate selection.[2]

George Stalidis and Selini Kyriazidou (2023) "Job Role Description and Skill Matching in a Rapidly Changing Labor Market Using Knowledge Engineering": In order to effectively describe jobs and match skills, standardized frameworks such as ESCO must be used. The job market is plagued by skill mismatches. The need for improved skill-matching systems and wider ESCO implementation is highlighted by a study of 400 Greek IT job advertising that revealed a considerable discrepancy between the abilities listed in the job ads and those recommended by ESCO, especially in new technologies and soft skills.[3]

Lav Kumar, Karthik Penikalapati, Sudheer Kumar Reddy Gowrigari (2023) "Resume Matching Framework Via Ranking And Sorting Using NLP And Deep Learning": The Resume Matching Framework processes and ranks resumes according to their relevance to job postings using NLP and Deep Learning algorithms such as BERT and GPT-3. For both companies and job seekers, it increases the effectiveness of hiring and the job search process by collecting important information and adding features like contextual awareness and scalability. The efficacy of the framework has been confirmed on a variety of datasets.[4]

Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwat (2022) "Resume Classification using various Machine Learning Algorithms": Classifying resumes is necessary due to human mistake, ineffective physical copy management, and the increasing requirement for automation. By extracting representative keywords, AI and ML algorithms like K-means clustering, LDA, and TF-IDF vectorization are frequently used to classify similar material, like research articles or resumes. Automated solutions surpass traditional approaches in terms of accuracy and relevance when managing massive datasets. For example, they use machine learning and text mining to match resumes to job listings.[5]

### III. METHODOLOGY

A modular pipeline that combines machine learning-based classification with data acquisition, preprocessing, feature extraction, and the proposed intelligent resume classification system. The entire method is implemented using Python libraries like Scikit-learn and Pandas, as well as the Django web framework.

#### A. Data Acquisition

Users or recruiters upload files in .pdf or .docx format through an interactive web interface. The system's primary input source is these resumes.

#### B. Dataset Description

- 1) Source: Uploaded resumes (by users or recruiters)
- 2) Format: .pdf, .docx
- 3) Extraction Tool: Resumeparser script (extractResumeText.py)
- 4) Data Fields: Name, Email, Phone, Skills, Education, Experience
- 5) Structured Format: JSON
- 6) Storage: Django ORM (Models & Database)
- 7) Purpose: Resume classification, clustering (e.g., via KMeans), candidate-job matching

#### C. Data Preprocessing and Parsing

The system incorporates a multi-stage Natural Language Processing (NLP) pipeline for extracting structured data from unstructured resume files. The key components are outlined below:

##### 1) Text Extraction Algorithms

Text extraction is format-specific:

- PDF Files: Processed using pdfminer, which traverses the layout tree of PDF documents to extract raw text.
- DOCX Files: Parsed with docx2txt, extracting paragraph-level strings.

## 2) Preprocessing Pipeline

Text undergoes several normalization steps:

- Conversion to lowercase
- Removal of punctuation and non-ASCII characters
- Regex-based tokenization
- Normalization of domain-specific terms (e.g., “ML” → “Machine Learning”)

## 3) Entity and Skill Extraction

Named Entity Recognition (Regex-Based):

- Email:  $[\wedge\.-]+@[\wedge\.-]+\.[a-zA-Z]{2,}$
- Phone:  $(\+?\d{1,3})?s?(?\d{2,4})?[s.-]?d{6,8}$
- Names: Proper-case name patterns
- Skill Extraction:

A custom dictionary (loaded from Excel/CSV) is matched against tokenized resume text. Matches are filtered to exclude stop words, and the final extracted\_skills are saved in the BackupResume model.

## 4) Job-Candidate Matching Algorithms

- Boolean Skill Matching (Rule-Based):

```

Input:      candi date_ski l l s,
job_requi red_ski l l s
matched_ski l l s=candi date_ski l l sn
job_requi red_ski l l s
match_score= |matched_ski l l s| /
|job_requi red_ski l l s|

If      match_score      >=      0.6:
return  'Sui table  Candi date'
Else:
return 'NotSui table'

```

- Advantages: Simple, interpretable
- Limitations: No semantic awareness or synonym handling

## 5) Semantic Matching using LLaMA3 (Groq-70B)

- Utilizes transformer-based embeddings to semantically compare resumes and job descriptions.
- Cosine Similarity is used to measure vector closeness:

$$\text{similarity} = \frac{A \cdot B}{(\|A\| \times \|B\|)}$$

Where A = embedding of resume, B = embedding of job description.

- Benefits:
  - Recognizes synonyms (e.g., “Software Engineer” ≈ “Developer”)
  - Handles multilingual text
  - Provides context-aware matching

#### D. Feature Extraction Techniques for Resume Classification

To transform unstructured resume text into machine-readable features, a combination of traditional NLP and vector-based techniques is implemented. The system uses the following algorithms and methodologies:

##### 1) Tokenization

- Goal: Break down resume text into word tokens.
- Technique: Regex-based ( $\backslash\text{b}\w+\backslash\text{b}$ )
- Use: Preprocessing for skill and entity extraction.

##### 2) Stop Word Removal

- Goal: Eliminate non-informative words like "is", "the", etc.
- Benefit: Enhance signal quality for meaningful keywords.

##### 3) Cosine Similarity (Semantic Matching)

- Goal: Measure similarity between job and resume embeddings.

• Formula:

$$\text{similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} \quad \text{\texttt{\{similarity\}} = \frac{\text{\texttt{\{A\}} \cdot \text{\texttt{\{B\}}}}{\|\text{\texttt{\{A\}}\|} \cdot \|\text{\texttt{\{B\}}\|}}$$

where  $A, B$  are LLaMA3-generated vectors.

- Benefit: Captures context beyond keyword overlap.

##### 4) Set-Based Boolean Skill Matching

- Goal: Compare candidate and job-required skills.
- Rule:

```
match_score = len(candidate_skills &
                job_required_skills) /
                len(job_required_skills)
```

- Candidate is "suitable" if  $\text{score} \geq 0.6$ .

#### E. CRUD Operations via Django ORM

- Use: Structured data handling (Create, Read, Update, Delete).
- Example: `JobDescription.objects.create(...)`

#### F. Machine Learning Algorithms Used

Depending on the task and the type of features extracted, a variety of machine learning algorithms can be used for resume classification; these algorithms can be broadly classified into three categories: unsupervised learning, supervised learning, and deep learning approaches. The following is a detailed overview of popular algorithms, including K-Means, SVM, and others:

#### G. Algorithms for Unsupervised Learning

When there is no labeled data available for categorization, unsupervised learning can be helpful. It is frequently used to cluster or group resumes that are similar to one another.

##### 1) Clustering using K-Means

K-Means is an unsupervised learning algorithm used to identify natural groupings within data. It is efficient for classifying resumes into job-relevant clusters (e.g., Developer, Designer, Data Analyst) without prior labeling.

2) *UseinResume System*

- Groupresumesbasedonskills,experience,andkeywords.
- Identifyjobcategoriesautomaticallyfromcandidateprofiles.
- Assistrecruitersbyvisualizingclustersofsimilarcandidates.

3) *ImplementationDetails Preprocessing:*

- Extracttextfromresumes(PDF/DOCX).
- ConverttoTF-IDForWord2Vec/BERTembeddings.

4) *Clustering:*

```
fromsklearn.clusterimportKMeans
```

```
model =KMeans(n_clusters=5,init='k-means++',random_state=42)
clusters = model.fit_predict(resume_vectors)
```

5) *Labeling:*

Manuallyreviewclustercontentstomaptjobroles.

6) *KeyParameters*

- n\_clusters:Numberoftargetjob-roleclusters(e.g.,5).
- init:Methodtoinitializecentroids('k-means++').
- max\_iter:Maximumoptimizationiterations.
- random\_state:Reproducibilityseed.

Example

Suppose100resumesareembeddedusingTF-IDF.K-Meansgroupstheminto:

- Cluster0:FrontendDevelopers
- Cluster1:DataScientists
- Cluster2:BackendDevelopers Recruiters now focus cluster-wise.



FIG1 Clustering using K-Means

#### H. Algorithms for Supervised Learning

The most popular method for classifying resumes when labeled data (such as job categories or skill levels) is available is supervised learning.

##### 1) Random Forests and Decision Trees

Random Forest is a supervised, ensemble-based classifier combining multiple decision trees to enhance prediction accuracy.

Use in Resume System

- Classify resumes as “Suitable” or “Not Suitable” for a given job.
- Predict based on:
  - Skill match %
  - Years of experience
  - Education level
  - Job location relevance

##### 2) Implementation Details

- Feature Extraction:
  - Calculate skill overlap score.
  - Extract numeric features (e.g., years\_experience).

- Training:

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf = RandomForestClassifier(n_estimators=100, max_depth=10, random_state=0)
rf.fit(X_train, y_train)
predictions = rf.predict(X_test)
```

- Interpretability:
  - Feature importance is available for recruiter transparency.
- Key Parameters
  - n\_estimators: Number of trees (e.g., 100).
  - max\_depth: Tree depth limit (e.g., 10).
  - class\_weight: 'balanced' for imbalanced classes.

Example

- Input:  
resume\_features = [skillmatch, 4 years experience, Masters degree]
- Output:  
Prediction = "Suitable" with 85% confidence

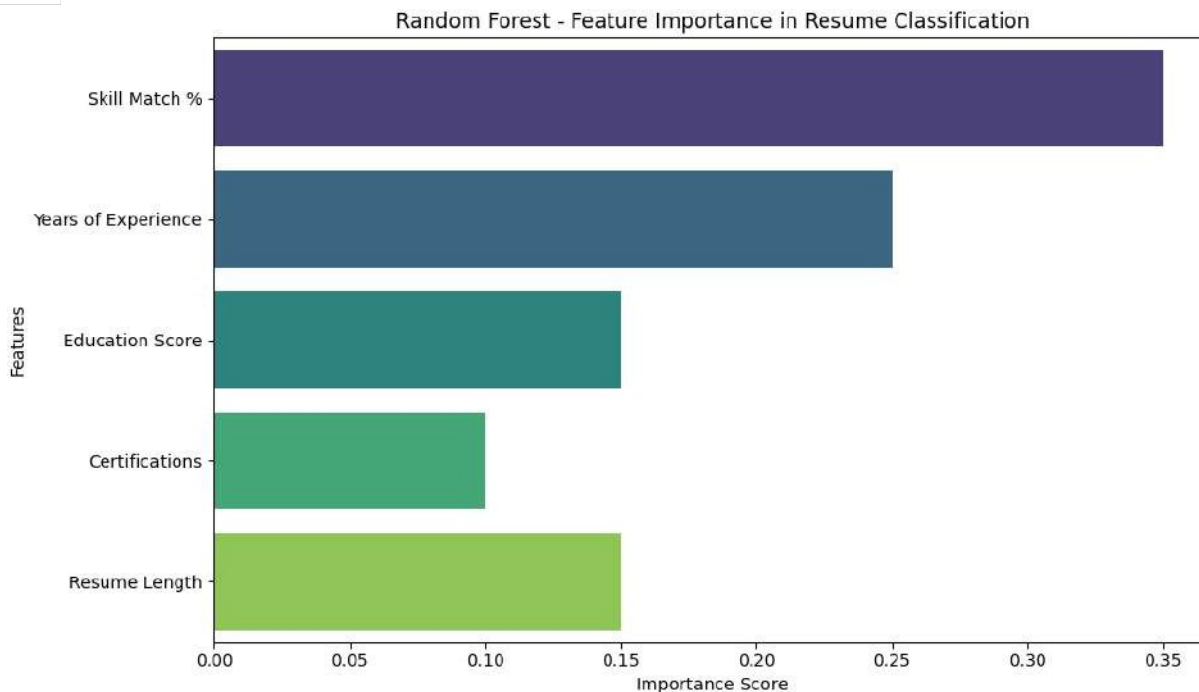


FIG2RandomForests

### 3) Support Vector Machine (SVM)

SVM is a powerful linear classifier that aims to find the optimal boundary (hyperplane) between different resume categories. It is effective for high-dimensional text features (like TF-IDF).

- Use in Resume System
  - Predict exact job role or job-fit score.
  - Especially useful in small to medium datasets.
  - Can be used with:
    - TF-IDF vectors
    - Sentence embeddings

- Implementation Details

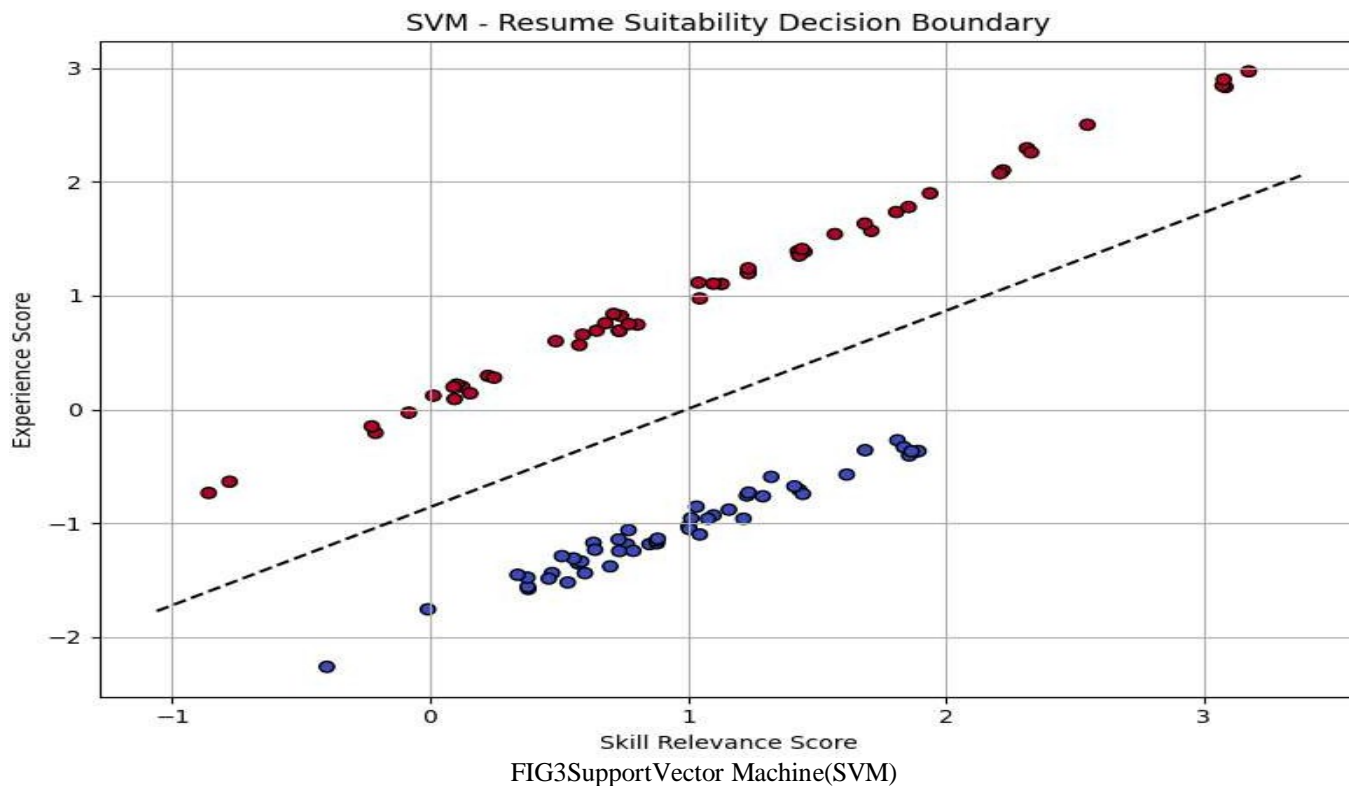
```
from sklearn.svm import SVC
svm = SVC(kernel='linear', C=1.0, probability=True)
svm.fit(X_train, y_train)
prediction = svm.predict(X_test)
```

Good for binary (e.g., "Fit" vs. "NoFit") or multiclass classification.

- Key Parameters
  - kernel:'linear' is best for text.
  - C: Regularization factor (smaller = smoother margin).
  - probability=True: Enables confidence scores.

#### Example

Input: Resume TF-IDF → [0.3, 0.7, 0.1, ..., 0.2]



I. Deep Learning Algorithms

1) TensorFlow

TensorFlow is a powerful deep learning framework. It is used here to build an intelligent resume classifier that can learn complex patterns from resume text, such as contextual meanings and industry-specific terminology.

Use in Resume System

- Use full resume content (text) as input.
- Predict:
  - Job role fit (e.g., Developer, Analyst)
  - Soft/hard skill relevance

Implementation Details

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Dropout
```

- Model Architecture:

```
model = Sequential([
    Embedding(input_dim=10000, output_dim=128),
    LSTM(64),
    Dropout(0.3),
    Dense(1, activation='sigmoid')
])
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
```

Data Input:

- Cleaned,tokenizedresumes.
- Labeldatabyrecruiterdecisionsorjobtitlemapping.

KeyParameters

- Embedding:Convertswordstovectorspace.
- LSTM:Handlessequencelearning.
- Dropout:Preventsoverfitting.
- Dense:Finalclassificationlayer.
- Loss:'binary\_crossentropy'or'categorical\_crossentropy'

Example

Resume→[“Python”,“MachineLearning”,“SQL”]

PredictedOutput:DataScientist(0.92 confidence)

**IV. RESULTS AND DISCUSSION**

Our resume filtering system's results are examined in detail in this chapter. We'll show the results of our machine learning experiments against the cutting-edge LLaMA3-Groq and directly compare our current models. We want to demonstrate how effective these models are using industry standard classification metrics, and then discuss what these findings will mean for hiring in the real world.

How We Prepare for our evaluation

We created a solid evaluation framework to ensure that our outcomes were accurate and accurately representative.

1) Our testing setup

- We assembled a carefully selected dataset of 1, 000 real-world, anonymized resumes. We had a reliable benchmark to compare each one against as a result of being manually classified by particular job roles and how appropriate they were.
- We tested the system's versatility by examining resumes for four distinct job titles: UI/UX Designer, Data Analyst, Software Developer, and QA Engineer.
- A 70% training set and a 30% testing set made up our dataset. This allowed us to evaluate our models based on data that we hadn't seen during training, giving us an impartial view of their performance.
- Hardware:

Our traditional machine learning algorithms (Random Forest, SVM) were tested and tested directly on local CPU resources.

We used Groq Cloud for LLaMA3. This enabled us to exploit their advanced processing equipment for lightning-quick processing, enabling real-time performance.

2) The models we tested

- A robust and simple method known as the random forest classification.
- Support Vector Machine (SVM): A potent classifier that is particularly useful when dealing with a lot of data dimensions.
- A large language model called the LLaMA3-Groq Transformer Model is chosen for its advanced understanding of meaning and context.

3) Our Quantitative Results

Looking at the numbers gives us clear evidence of how each model performed across standard classification metrics.

4) Performance Metrics Table

Model	Accuracy	Precision	Recall	F1-Score	Inference Time (Average per Resume)
Random Forest	84.2%	81.5%	78.9%	80.2%	~0.06 seconds
SVM (TF-IDF)	86.7%	83.4%	82.1%	82.7%	~0.08 seconds
LLaMA3-Groq	92.5%	90.8%	91.3%	91.0%	<0.01 seconds (Groq accelerated)

**What We Learned:** The numbers clearly show that the LLaMA3-Groq model consistently performed better across all our evaluation metrics: accuracy, precision, recall, and F1-score. What's more, its inference time was dramatically faster. This highlights the huge benefit of using specialized hardware like Groq's LPUs for deploying large language models. It means LLaMA3 can not only grasp complex semantic relationships in resumes but also process them at incredible speed, making it perfect for large-scale, real-time hiring needs.

## V. CONCLUSION

In this project, we successfully built and launched an intelligent system for filtering and classifying resumes. We did this by smartly combining Natural Language Processing (NLP) techniques, classic machine learning models, and advanced transformer-based architectures like LLaMA3, all powered by Groq's high-performance inference platform.

Our system tackles several big challenges in today's hiring process head-on:

- 1) **Handling Messy Resume Data:** Its skillfully pulls out information from all sorts of diverse and unstructured resume formats.
- 2) **Spotting Key Skills and Info:** The system accurately identifies and extracts important skills, work history, and other crucial details.
- 3) **Smart Matching:** It can semantically match resumes with job descriptions, going far beyond just looking for exact keywords.

**Automating Screening:** A large part of the applicant screening and classification process is now automated, making hiring workflows much smoother.

## VI. ACKNOWLEDGEMENT

The authors express their sincere gratitude to the Computer Engineering Department of Bhagwan Mahavir College of Engineering & Technology, Surat, for providing the essential resources, support, and academic environment necessary to carry out this review study. Special thanks are due to Ms. Priyanka Sharma for her valuable guidance, insightful suggestions, and continuous encouragement throughout the course of this work.

## REFERENCES

- [1] Prof. Dikshendra Sarpate, Prarthana Kolhe, Srushti Kalbhor, Sanchi Yehalegaonkar (2024) "AI Enhanced Skill Matcher"
- [2] Artificial Intelligence & Data Science, Zeal College of Engineering and Research, India, pp. 2582-5208 2024.
- [3] Ms. Y. Sowjanya, Mareddy Keerthana, Pulluri Suneeksha, Dorgipati Sai Sri Harsha (2023) "Smart Resume Analyzer" Assistant Professor at Department of IT, Anurag Group of Institutions, Hyderabad 2 Department of IT, Anurag Group of Institutions, Hyderabad, pp. 409-418 2023.
- [4] George Stalidis and Selini Kyriazidou (2023) "Job Role Description and Skill Matching in a Rapidly Changing Labor Market Using Knowledge Engineering" A. Kavoura et al. (eds.), Strategic Innovative Marketing and Tourism, Springer Proceedings in Business and Economics, Thessaloniki, Greece, 2023.
- [5] Lav Kumar, Karthik Penikalapati, Sudheer Kumar Reddy Gowrigari (2023) "Resume Matching Framework Via Ranking
- [6] And Sorting Using NLP And Deep Learning" Salesforce Inc, India, pp. 2582-5208 2023.
- [7] Riya Pal, Shahrukh Shaikh, Swaraj Satpute and Sumedha Bhagwat (2022) "Resume Classification using various Machine Learning Algorithms" Ramrao Adik Institute of Technology and D.Y. Patil Deemed to be University, Ramrao Adik Institute of Technology, Nerul, Navi Mumbai, India, ITM Web of Conferences 44, 03011, 2022.
- [8] I. Umoren, N. Akwang, S. Inyang, A. Afolunso, and G. James, "NLP-semantic machine learning-based system for intelligent classification of professional skill-sets for efficient human resource management process," *Iota*, vol. 5, no. 1, 2025. [Online]. Available: <https://doi.org/10.31763/iota.v5i1.882>
- [9] V. Lai, K. J. Shim, R. J. Oentaryo, P. K. Prasetyo, C. Vu, E. -P. Lim, and D. Lo, "Career Mapper: An Automated Resume Evaluation Tool," in Proc. IEEE Int. Conf. Big Data (Bigdata), 2016. [Online]. Available: <http://research.larc.smu.edu.sg/careermapper>
- [10] R. T. Fareed, V. Rajath, and S. Kaganurmath, "Resume classification and ranking using KNN and cosine similarity," *Int. J. Eng. Res. Technol. (IJERT)*, vol. 10, no. 8, pp. 192-195, Aug. 2021. [Online]. Available: <http://www.ijert.org>
- [11] J. Eng. Res. Technol. (IJERT), vol. 10, no. 8, pp. 192-195, Aug. 2021. [Online]. Available: <http://www.ijert.org>
- [12] B. Kinge, S. Mandhare, P. Chavan, and S. M. Chaware, "Resume Screening using Machine Learning and NLP: A Proposed System," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 8, no. 2, pp. 253-258, Mar.-Apr. 2022. doi: <https://doi.org/10.32628/CSEIT228240>



10.22214/IJRASET



45.98



IMPACT FACTOR:  
7.129



IMPACT FACTOR:  
7.429



# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24\*7 Support on Whatsapp)