Smart Hire+ An Intelligent Resume Screening and Candidate Ranking Framework Using Natural Language Processing and Machine Learning

Kodiboina Navyasri Durga[1], Mr. S. Lakshmana Rao[2]
[1]*M. Tech Student, Department of Computer Science and Engineering, Gonna Institute of Information Technology and Sciences, Visakhapatnam, Andhra Pradesh, India*
[2]*Assistant Professor and HOD, Department of Computer Science and Engineering, Gonna Institute of Information Technology and Sciences, Visakhapatnam, Andhra Pradesh, India*

***Abstract*—The increasing volume of job applications received through online recruitment platforms has made manual resume screening a challenging and time-consuming task for organizations. Traditional recruitment approaches often rely on manual evaluation or keyword-based filtering techniques, which may lead to inconsistent candidate assessment and inefficient hiring decisions. To address these challenges, this paper presents SmartHire+, an intelligent resume screening and candidate ranking framework that integrates Natural Language Processing (NLP), Machine Learning (ML), and Learning-to-Rank techniques to automate candidate evaluation and recruitment decision support. The proposed framework extracts and processes information from candidate resumes and job descriptions using text extraction, Optical Character Recognition (OCR), and text preprocessing techniques. Multiple similarity analysis methods, including Term Frequency–Inverse Document Frequency (TF-IDF), Bag-of-Words (BoW), Word2Vec, and Sentence-BERT (SBERT), are employed to measure the relevance between candidate profiles and job requirements. In addition, feature engineering techniques such as skill matching, domain matching, and keyword overlap analysis are incorporated to enhance candidate assessment. Several machine learning algorithms, including Logistic Regression, Random Forest, XGBoost, and Gradient Boosting, are utilized for candidate classification, while a LambdaMART-based Learning-to-Rank model is implemented to prioritize applicants according to their suitability for specific job roles. Experimental results demonstrate the effectiveness of the proposed framework in intelligent recruitment applications. Among the evaluated classification models, Random Forest and Gradient Boosting achieved the highest accuracy of 75.50%. Furthermore, the ranking framework achieved an average Normalized Discounted Cumulative Gain (NDCG) score of 0.9895, indicating excellent candidate prioritization performance. The system also incorporates a skill gap analysis module that identifies missing competencies and generates career and course recommendations to support professional development. A**
Flask-based web application was developed to provide recruiters with an interactive platform for real-time resume analysis and candidate evaluation. The obtained results demonstrate that SmartHire+ effectively improves recruitment efficiency, candidate selection accuracy, and data-driven hiring decision-making.

***Index Terms*—Resume Screening, Candidate Ranking, Natural Language Processing, Machine Learning, Learning-to-Rank, LambdaMART, Word2Vec, Sentence-BERT, Skill Gap Analysis, Recruitment Automation.**

I. INTRODUCTION

The rapid growth of digital recruitment platforms has transformed the way organizations identify, evaluate, and hire potential candidates. With the increasing number of job applications submitted through online portals, recruiters are often required to review hundreds of resumes for a single position. This large volume of applications makes manual resume screening a time-consuming and resource-intensive process. Moreover, traditional recruitment methods frequently rely on human judgment and keyword-based filtering techniques, which may result in inconsistent evaluations and the omission of qualified

candidates. Consequently, there is a growing demand for intelligent recruitment systems capable of automating candidate assessment while maintaining accuracy and efficiency. Recent advancements in Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP) have created new opportunities for improving recruitment processes. NLP techniques enable the extraction and interpretation of meaningful information from unstructured textual documents such as resumes and job descriptions, while machine learning algorithms facilitate automated decision-making based on historical patterns and candidate attributes. These technologies have been increasingly adopted in recruitment applications to support candidate matching, resume classification, and recommendation generation. However, many existing systems primarily depend on keyword matching approaches and often fail to capture semantic relationships between candidate qualifications and job requirements.

Another significant challenge in modern recruitment is candidate prioritization. While identifying suitable applicants is important, recruiters also require mechanisms to rank candidates according to their relevance and suitability for specific job roles. Conventional screening systems typically focus on classification tasks and provide limited support for ranking candidates based on multiple evaluation criteria. Furthermore, most recruitment platforms do not offer comprehensive skill gap analysis or personalized recommendations that can assist candidates in improving their qualifications and employability. To address these limitations, this paper proposes SmartHire+, an intelligent resume screening and candidate ranking framework that integrates NLP, machine learning, and Learning-to-Rank techniques. The framework utilizes multiple similarity analysis methods, including TF-IDF, Bag-of-Words (BoW), Word2Vec, and Sentence-BERT (SBERT), to evaluate the compatibility between candidate resumes and job descriptions. In addition, machine learning algorithms are employed for candidate assessment, while a LambdaMART-based ranking model is used to prioritize applicants according to their relevance scores. The framework further incorporates a skill gap analysis module that identifies missing competencies and generates career and course recommendations.

The proposed system is implemented as a Flask-based web application that enables recruiters to upload resumes, provide job descriptions, and obtain real-time candidate evaluation results. Experimental analysis demonstrates the effectiveness of the framework in recruitment applications, achieving a classification accuracy of 75.50% and an average NDCG score of 0.9895 for candidate ranking. These results indicate the capability of the proposed approach to improve recruitment efficiency, enhance candidate selection accuracy, and support intelligent hiring decisions. The remainder of this paper is organized as follows. Section II presents a review of existing research related to resume screening and intelligent recruitment systems. Section III describes the proposed methodology and system architecture. Section IV discusses the experimental results and performance evaluation. Finally, Section V concludes the paper and outlines future research directions.

II. LITERATURE REVIEW

The application of Artificial Intelligence (AI), Machine Learning (ML), and Natural Language Processing (NLP) in recruitment has attracted considerable attention in recent years due to the increasing demand for efficient talent acquisition systems. Traditional recruitment processes largely depend on manual resume screening, which requires significant time and effort from recruiters. As the volume of job applications continues to grow, researchers have explored automated approaches for candidate evaluation, resume analysis, and recruitment decision support [13], [14].

Early resume screening systems primarily relied on keyword matching techniques to compare candidate resumes with job descriptions. These systems identified applicants based on the occurrence of predefined keywords and skills. Although keyword-based methods reduced manual effort, they often failed to recognize semantic relationships between terms and were unable to accurately evaluate candidate suitability when different expressions were used to describe similar competencies [10], [13].

To overcome these limitations, researchers introduced Natural Language Processing techniques for resume analysis. Methods such as Term Frequency–Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW) became widely used for representing textual documents in numerical form and measuring document similarity. These approaches improved

automated candidate matching by considering the significance and frequency of terms within resumes and job descriptions. However, they still exhibited limitations in understanding contextual meaning and semantic relationships among words [10], [11].

The development of word embedding models further enhanced recruitment-related text analysis. Techniques such as Word2Vec enabled the transformation of textual information into dense vector representations that capture semantic similarities between words. By representing related terms in similar vector spaces, Word2Vec improved candidate-job matching performance and facilitated more meaningful resume evaluation. Nevertheless, traditional word embedding models generally produce static representations and may not fully capture contextual information within complex recruitment documents [1].

Recent advancements in transformer-based language models have significantly improved semantic text understanding. Models such as BERT and Sentence-BERT (SBERT) generate contextual embeddings that capture the meaning of words and sentences within their surrounding context. These models have demonstrated superior performance in various NLP tasks, including document similarity analysis, information retrieval, and recommendation systems. Their ability to understand semantic relationships makes them highly suitable for resume screening and candidate evaluation applications [2], [3].

In addition to candidate matching, machine learning techniques have been increasingly utilized to automate recruitment decision-making. Classification algorithms such as Logistic Regression, Random Forest, XGBoost, and Gradient Boosting have been employed to predict candidate suitability based on extracted features. These approaches enable recruitment systems to learn patterns from historical hiring data and improve prediction accuracy. Ensemble learning methods, in particular, have demonstrated strong performance in handling complex recruitment datasets and candidate evaluation tasks [7] – [9].

Another emerging area of research focuses on candidate ranking rather than simple classification. Learning-to-Rank techniques, including LambdaMART and LightGBM Ranker, have been successfully applied in recommendation systems and information retrieval applications. These methods generate ordered candidate lists by considering multiple evaluation features simultaneously, thereby supporting more effective recruitment decisions. Ranking-based approaches better reflect real-world hiring scenarios, where recruiters prioritize applicants according to relevance rather than merely categorizing them as suitable or unsuitable [4] – [6].

Despite these advancements, many existing recruitment systems continue to focus primarily on candidate matching and classification while providing limited support for candidate ranking, skill gap identification, and personalized recommendations. Furthermore, several solutions rely on a single similarity analysis technique, which may not adequately capture both lexical and semantic aspects of candidate-job relationships. To address these limitations, the proposed SmartHire+ framework integrates multiple NLP techniques, machine learning models, and a Learning-to-Rank mechanism within a unified recruitment platform. In addition, the framework incorporates skill gap analysis and recommendation modules to provide comprehensive recruitment decision support and professional development guidance.

III. PROPOSED METHODOLOGY

3.1 Proposed Smart Hire+ Framework

The proposed SmartHire+ framework is designed to automate the recruitment process by integrating Natural Language Processing (NLP), Machine Learning (ML), and Learning-to-Rank techniques within a unified candidate evaluation platform. The framework aims to assist recruiters in efficiently screening resumes, identifying suitable candidates, generating ranking scores, and providing skill-based recommendations. By reducing the dependency on manual resume evaluation, the proposed system improves recruitment efficiency, consistency, and decision-making accuracy. The framework operates through a multi-stage processing pipeline that begins with the submission of candidate resumes and job descriptions. Resumes are uploaded in PDF format through a web-based interface, while recruiters provide the corresponding job requirements. The system extracts textual information from resumes using document processing and Optical Character Recognition (OCR) techniques when necessary. The extracted content is subsequently preprocessed to

remove noise and generate standardized textual representations suitable for analysis. After preprocessing, multiple feature extraction and similarity analysis techniques are employed to evaluate the relationship between candidate profiles and job descriptions. The framework utilizes TF-IDF, Bag-of-Words (BoW), Word2Vec, and Sentence-BERT (SBERT) models to capture both lexical and semantic similarities. These techniques generate candidate-job matching features that serve as inputs to the machine learning and ranking modules.

The candidate evaluation stage incorporates several machine learning algorithms, including Logistic Regression, Random Forest, XGBoost, and Gradient Boosting, to analyze candidate suitability. In addition, a LambdaMART-based Learning-to-Rank model is employed to prioritize applicants according to their relevance scores. This ranking mechanism enables recruiters to focus on the most qualified candidates and supports efficient recruitment decision-making. To further enhance recruitment support, the framework includes a Skill Gap Analysis module that compares candidate competencies with job requirements. Missing skills are automatically identified, and personalized career and course recommendations are generated to assist candidates in improving their qualifications. The final results, including similarity scores, candidate rankings, identified skills, missing competencies, and recommendations, are presented through an interactive dashboard. The integration of resume screening, candidate ranking, skill gap identification, and recommendation generation within a single framework makes Smart Hire+ a comprehensive solution for intelligent recruitment and talent acquisition. The proposed approach not only improves candidate selection accuracy but also provides valuable insights that support both recruiters and job seekers throughout the hiring process.

3.2 System Architecture

The system architecture of the proposed SmartHire+ framework illustrates the interaction among the major modules responsible for resume processing, candidate evaluation, ranking, and recommendation generation. The architecture is designed to provide an end-to-end recruitment solution that automates candidate screening while ensuring accurate and efficient decision support for recruiters. The process begins when a recruiter uploads a candidate resume and provides a corresponding job description through the Flask-based web interface. The uploaded resume is forwarded to the text extraction module, where textual information is extracted using PDF processing techniques. In cases where scanned resumes are encountered, Optical Character Recognition (OCR) is employed to convert image-based content into machine-readable text. The extracted content then undergoes preprocessing operations, including text normalization, removal of special characters, and elimination of irrelevant words, to generate clean textual data.The preprocessed resume and job description data are subsequently supplied to the feature engineering and similarity analysis module. Multiple text representation techniques, including TF-IDF, Bag-of-Words (BoW), Word2Vec, and Sentence-BERT (SBERT), are utilized to compute similarity scores and capture both lexical and semantic relationships between candidate profiles and job requirements. These similarity measures, together with additional candidate-related features, form the basis for candidate assessment.

The generated features are processed by machine learning models to evaluate candidate suitability. Furthermore, a LambdaMART-based Learning-to-Rank engine utilizes the extracted features to prioritize candidates according to their relevance scores. This ranking mechanism enables recruiters to efficiently identify highly qualified applicants from a large pool of candidates. In parallel, the skill gap analysis module compares candidate competencies with the skills specified in the job description. Missing skills are identified and forwarded to the recommendation engine, which generates career guidance and course recommendations aimed at improving candidate qualifications. Finally, all analytical outputs, including similarity scores, candidate rankings, identified skills, missing competencies, and recommendations, are presented through the results dashboard for recruiter review.

The modular architecture of SmartHire+ ensures scalability, maintainability, and seamless integration of advanced NLP and machine learning techniques, making the framework suitable for intelligent recruitment and talent acquisition applications.

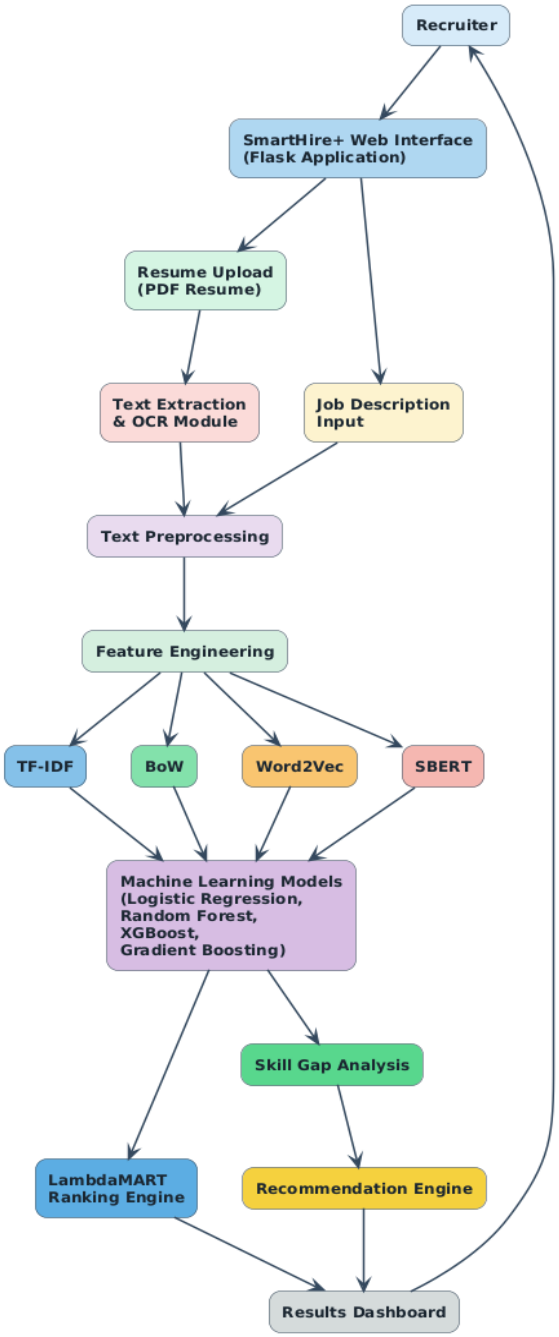

Fig. 1. Proposed Smart Hire+ System Architecture for intelligent resume screening, candidate ranking, skill gap analysis, and recommendation generation.

3.3 Resume Text Extraction and Preprocessing

Resume text extraction and preprocessing constitute a critical stage of the proposed SmartHire+ framework, as the quality of extracted information directly influences candidate evaluation and ranking performance. Since resumes are generally available in unstructured PDF formats, they cannot be directly utilized by machine learning algorithms. Therefore, an effective mechanism is required to transform resume documents into structured textual representations suitable for further analysis. The proposed framework begins by extracting textual content from uploaded resumes using PDF processing techniques. The extraction process retrieves relevant candidate information, including educational qualifications, technical skills, certifications, work experience, projects, and professional achievements. In situations where resumes are stored as scanned documents or image-based PDFs, Optical Character Recognition (OCR) is employed to convert visual content into machine-readable text. This approach ensures compatibility with diverse resume formats commonly encountered in recruitment environments.

Once the textual information is extracted, preprocessing operations are performed to improve data quality and eliminate inconsistencies. Initially, all extracted text is converted to lowercase to maintain uniformity across documents. Special characters, punctuation symbols, numerical values, and irrelevant formatting elements are removed to reduce noise within the textual data. Additional cleaning procedures are applied to eliminate redundant spaces and normalize document content. The cleaned text is subsequently processed to retain meaningful information required for candidate assessment. Text normalization enhances consistency among resumes originating from different sources and formatting styles. The resulting preprocessed text provides a standardized representation of candidate profiles, enabling reliable comparison with job descriptions during subsequent analysis stages. The output generated from the preprocessing module serves as the foundation for feature engineering and similarity analysis. By transforming unstructured resume documents into clean and structured textual data, the proposed framework improves the effectiveness of machine learning models and ranking algorithms, thereby contributing to more accurate candidate evaluation and recruitment decision-making.

3.4 Feature Engineering and Similarity Analysis

Feature engineering and similarity analysis play a vital role in the proposed SmartHire+ framework by transforming textual information into meaningful numerical representations that facilitate candidate

evaluation and ranking. The primary objective of this stage is to quantify the relationship between candidate resumes and job descriptions through the extraction of informative features and the application of multiple similarity measurement techniques. After preprocessing, the textual content obtained from resumes and job descriptions is converted into machine-readable representations using various Natural Language Processing techniques. The framework employs multiple similarity analysis methods to capture different aspects of candidate-job compatibility. Term Frequency–Inverse Document Frequency (TF-IDF) is utilized to identify important terms within documents and measure their relevance across candidate profiles and job requirements. This technique enables the framework to evaluate document similarity based on significant keywords and domain-specific terminology.

In addition to TF-IDF, the Bag-of-Words (BoW) model is employed to represent textual documents as collections of individual words. This approach provides lexical information regarding the frequency of terms appearing within resumes and job descriptions, thereby contributing to candidate matching and relevance estimation. Although BoW does not capture semantic relationships, it remains useful for identifying direct keyword overlaps between candidate qualifications and job requirements. To enhance semantic understanding, Word2Vec embeddings are incorporated into the framework. Word2Vec generates dense vector representations of words that preserve contextual and semantic relationships. Consequently, the system can identify related skills and competencies even when exact keyword matches are absent. Furthermore, Sentence-BERT (SBERT) is utilized to generate contextual sentence embeddings that capture deeper semantic meaning from complete resume and job description texts. This capability significantly improves similarity estimation by considering contextual information rather than relying solely on keyword occurrence.

Beyond textual similarity measures, additional candidate-related features are engineered to strengthen the evaluation process. These include skill matching scores, domain relevance indicators, keyword overlap measures, and resume characteristics. Such features provide supplementary information regarding candidate suitability and contribute to more comprehensive assessment and ranking. The similarity scores and engineered features generated during this stage are subsequently supplied to the machine learning and ranking modules. By combining lexical, semantic, and domain-specific information, the proposed framework achieves a more robust representation of candidate qualifications and job requirements. This multi-dimensional feature extraction strategy enhances candidate matching accuracy and supports effective recruitment decision-making.

3.5 Machine Learning Models

To evaluate classification performance, multiple supervised machine learning algorithms were implemented and compared. Logistic Regression was utilized as a baseline model due to its simplicity, interpretability, and effectiveness in binary classification tasks. The model estimates the probability of candidate suitability based on the extracted feature set and provides a computationally efficient approach for recruitment analysis. Random Forest was incorporated as an ensemble learning technique that combines multiple decision trees to improve predictive performance and reduce overfitting. By aggregating the outputs of several trees, the model captures complex relationships among candidate attributes and generates more robust classification results. Its ability to handle high-dimensional feature spaces makes it particularly suitable for resume evaluation applications.

In addition, XGBoost and Gradient Boosting models were implemented to further enhance classification accuracy. These boosting-based algorithms iteratively improve prediction performance by focusing on previously misclassified instances during the training process. Their capability to model nonlinear relationships and complex feature interactions enables more effective candidate suitability assessment compared with conventional classification techniques. The performance of the implemented machine learning models was evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score.

Experimental results demonstrated that Random Forest and Gradient Boosting achieved the highest classification accuracy of 75.50%, indicating their effectiveness in identifying suitable candidates within the recruitment dataset. The comparative analysis of

multiple algorithms ensured the selection of reliable models for candidate evaluation and provided valuable insights into their suitability for intelligent recruitment applications. The outputs generated by these machine learning models are subsequently utilized by the candidate ranking module to further prioritize applicants according to their relevance and overall suitability for specific job roles.

3.6 Candidate Ranking Using LambdaMART

Candidate ranking is a critical component of the proposed SmartHire+ framework, as recruiters often need to prioritize applicants according to their relevance rather than simply classify them as suitable or unsuitable. To address this requirement, the framework incorporates a Learning-to-Rank approach based on the LambdaMART algorithm, which is widely recognized for its effectiveness in ranking and recommendation applications. The ranking process utilizes the similarity scores and engineered features generated during previous stages, including TF-IDF similarity, Bag-of-Words similarity, Word2Vec similarity, Sentence-BERT similarity, skill matching scores, and domain relevance indicators. These features collectively represent the degree of compatibility between candidate profiles and job requirements. The LambdaMART model learns ranking patterns from the training data and assigns a ranking score to each candidate based on their overall relevance.

Unlike conventional classification models that produce binary predictions, LambdaMART generates an ordered list of candidates by optimizing ranking quality. This approach enables the framework to place highly relevant candidates at the top positions of the recommendation list, thereby improving recruiter efficiency and reducing the effort required to identify qualified applicants. The ranking mechanism is particularly beneficial in large-scale recruitment scenarios where numerous candidates apply for the same position. To evaluate ranking effectiveness, the Normalized Discounted Cumulative Gain (NDCG) metric was employed. NDCG measures the quality of ranked outputs by comparing the generated ranking with the ideal ranking order. Experimental results demonstrated that the proposed ranking framework achieved an average NDCG score of 0.9895, indicating excellent ranking performance and a high degree of alignment with ideal candidate prioritization. The obtained results confirm the effectiveness of the LambdaMART-based ranking strategy in supporting intelligent recruitment decision-making and candidate selection.

3.7 Skill Gap Analysis and Recommendation Module

The Skill Gap Analysis and Recommendation Module extends the functionality of the proposed SmartHire+ framework beyond candidate screening and ranking by providing valuable insights into candidate competencies and professional development opportunities. The objective of this module is to identify discrepancies between candidate skills and job requirements and generate personalized recommendations that can assist both recruiters and job seekers. The analysis begins by extracting skills from candidate resumes and job descriptions using a predefined skills repository. The extracted skill sets are compared to determine the extent of alignment between candidate qualifications and the competencies required for a specific job role. Skills present in both documents are identified as matching skills, while skills specified in the job description but absent from the candidate profile are classified as missing skills. This comparison enables the framework to accurately identify competency gaps that may affect candidate suitability.

Based on the identified gaps, the recommendation module generates personalized suggestions aimed at improving candidate employability. Career recommendations are produced by analyzing the candidate's existing skill set and identifying job roles that closely align with their qualifications. Additionally, course recommendations are provided for missing skills, enabling candidates to pursue targeted learning opportunities and strengthen their professional profiles. The integration of skill gap analysis and recommendation generation enhances the practical value of the SmartHire+ framework by providing actionable insights rather than merely ranking candidates. This functionality benefits recruiters by offering a deeper understanding of candidate capabilities while simultaneously supporting candidates in identifying areas for improvement. Consequently, the proposed framework serves not only as a recruitment support system but also as a tool for continuous skill development and career advancement.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experimental Setup

The experimental evaluation of the proposed SmartHire+ framework was conducted to assess its effectiveness in automated resume screening, candidate ranking, and skill gap analysis. The implementation was carried out using a combination of cloud-based and local development environments to support model training, performance evaluation, and application deployment. This setup enabled comprehensive experimentation under conditions that closely resemble real-world recruitment scenarios. The model development and training phase was performed using the Google Colab platform, which provided the computational resources required for data preprocessing, feature extraction, similarity analysis, machine learning model training, and candidate ranking. Google Colab facilitated efficient experimentation with multiple machine learning algorithms and ranking techniques while providing access to widely used Python-based data science libraries.

The deployment phase was implemented using Visual Studio Code and the Flask web framework. A web-based application was developed to allow recruiters to upload resumes, provide job descriptions, and obtain real-time candidate evaluation results. The application served as an interface between users and the backend analytical modules responsible for resume processing, candidate assessment, and recommendation generation. The proposed framework was implemented using Python along with several supporting libraries. Resume text extraction was performed using PDF processing tools and Optical Character Recognition techniques. Text preprocessing and feature engineering were carried out using Natural Language Processing libraries, while Scikit-learn was utilized for machine learning model development. Word2Vec embeddings were generated using Gensim, semantic similarity analysis was performed using Sentence-BERT, and candidate ranking was implemented through the LightGBM-based LambdaMART framework.

The dataset utilized in this study consisted of resumes, job descriptions, and relevance information collected from publicly available recruitment-related sources and subsequently organized for experimental analysis. The dataset was used to evaluate candidate matching performance, classification accuracy, ranking quality, and recommendation effectiveness. Multiple similarity analysis techniques, including TF-IDF, Bag-of-Words, Word2Vec, and Sentence-BERT, were employed to generate features for candidate evaluation. To assess the performance of the proposed framework, standard classification metrics such as accuracy, precision, recall, and F1-score were used for evaluating machine learning models. In addition, the Normalized Discounted Cumulative Gain (NDCG) metric was employed to measure the quality of candidate ranking generated by the LambdaMART model. These evaluation measures provided a comprehensive assessment of both classification effectiveness and ranking performance. The experimental setup was designed to validate the practical applicability of the SmartHire+ framework and to demonstrate its capability to support intelligent recruitment decision-making through automated candidate evaluation, ranking, and recommendation generation.

4.2 Model Performance Analysis

The effectiveness of the proposed SmartHire+ framework was evaluated using multiple machine learning algorithms to determine their capability in candidate classification and recruitment decision support. The experimental analysis focused on comparing the performance of Logistic Regression, Random Forest, XGBoost, and Gradient Boosting models using the extracted similarity features and candidate-related attributes. The evaluation was conducted using standard performance metrics, including accuracy, precision, recall, and F1-score, to provide a comprehensive assessment of model effectiveness. The experimental results indicate that all implemented machine learning models achieved satisfactory classification performance. Logistic Regression obtained an accuracy of 74.50%, demonstrating its ability to effectively classify candidate profiles based on the extracted features. The model provided a reliable baseline for performance comparison and produced consistent classification results across the evaluation dataset.

Among the evaluated models, Random Forest achieved the highest classification accuracy of 75.50%. The ensemble learning mechanism enabled the model to capture complex feature interactions and

improve prediction reliability. Similarly, the Gradient Boosting model also achieved an accuracy of 75.50%, indicating strong predictive capability and effective learning of candidate-job relationships. The XGBoost model achieved an accuracy of 74.00%, which remained competitive while providing efficient training and classification performance. The comparative performance of the machine learning models is presented in Table I.

Table I - Performance Comparison of Machine Learning Models

Model	Accuracy (%)
Logistic Regression	74.50
Random Forest	75.50
XGBoost	74.00
Gradient Boosting	75.50

To obtain deeper insights into classification effectiveness, precision, recall, and F1-score metrics were also analyzed. Precision evaluates the correctness of positive predictions, recall measures the ability of the model to identify relevant candidates, and F1-score provides a balanced assessment of both metrics. The classification report generated during experimentation is summarized in Table II.

Table Ii - Classification Performance Metrics

Class	Precision	Recall	F1-Score
Class 0	0.73	0.67	0.70
Class 1	0.41	0.48	0.44
Macro Average	0.57	0.58	0.57
Weighted Average	0.63	0.61	0.62

The results indicate that the proposed framework is capable of effectively distinguishing between relevant and non-relevant candidates based on the extracted recruitment features. The ensemble-based models demonstrated superior performance compared with conventional classification approaches, highlighting their suitability for intelligent recruitment applications. The achieved accuracy and classification metrics confirm that the combination of similarity analysis techniques and machine learning algorithms provides a reliable foundation for candidate evaluation. Overall, the experimental findings validate the effectiveness of the proposed SmartHire+ framework in automating candidate assessment and supporting recruitment decision-making. The strong performance achieved by Random Forest and Gradient Boosting models demonstrates the capability of the framework to accurately evaluate candidate suitability and contribute to efficient talent acquisition processes.

4.3 Candidate Ranking Results

Candidate ranking is a crucial component of the proposed SmartHire+ framework, as recruiters often require an ordered list of applicants based on their relevance to a specific job role. While classification models determine candidate suitability, ranking models provide additional support by prioritizing candidates according to their overall qualifications and compatibility with job requirements. To achieve this objective, the proposed framework incorporates a Learning-to-Rank approach based on the LambdaMART algorithm implemented through the LightGBM Ranker. The ranking model utilizes multiple features generated during similarity analysis and candidate evaluation, including TF-IDF similarity, Bag-of-Words similarity, Word2Vec similarity, Sentence-BERT similarity, skill matching information, and domain relevance indicators. These features collectively represent the relationship between candidate profiles and job descriptions, enabling the ranking model to accurately estimate candidate relevance. Based on the generated ranking scores, applicants are arranged in descending order of suitability, allowing recruiters to focus on the most qualified candidates.

To evaluate ranking effectiveness, the Normalized Discounted Cumulative Gain (NDCG) metric was employed. NDCG is widely used in ranking systems because it measures how closely the generated ranking matches the ideal ranking order. The proposed framework achieved an average NDCG score of 0.9895, indicating excellent ranking quality and demonstrating the ability of the model to consistently place highly relevant candidates at the top positions of the recommendation list. The candidate ranking output generated during experimentation, where applicants are ordered according to their ranking scores. The results demonstrate that candidates with stronger alignment to job requirements obtained higher ranking scores and appeared at the top of the ranked list. The

top-ranked candidates obtained during the evaluation process are summarized in Table III.

Table III - Top-Ranked Candidates Generated By The Proposed Framework

Resume ID	Domain	Relevance Score	Ranking Score
data_science_4.pdf	Data Science	3	8.2781
data_science_3.pdf	Data Science	3	8.2339
data_science_7.pdf	Data Science	3	7.4160
data_science_9.pdf	Data Science	3	7.2805
data_science_5.pdf	Data Science	3	7.1924

The ranking results confirm that the proposed Learning-to-Rank framework effectively differentiates candidate profiles and prioritizes applicants according to their suitability. The high NDCG score achieved during experimentation further validates the reliability of the ranking model and demonstrates its practical applicability in recruitment environments where efficient candidate prioritization is essential.

Overall, the candidate ranking module significantly enhances the recruitment process by reducing manual screening effort and enabling recruiters to identify highly qualified candidates more efficiently. The integration of LambdaMART within the SmartHire+ framework provides a robust mechanism for intelligent candidate prioritization and contributes substantially to data-driven recruitment decision-making.

4.4 Skill Gap Analysis Results

The Skill Gap Analysis module was incorporated into the proposed SmartHire+ framework to identify differences between candidate competencies and job requirements. In addition to evaluating candidate suitability, this module provides valuable insights into missing skills that may affect employment opportunities. The objective is to assist recruiters in understanding candidate strengths and limitations while simultaneously enabling applicants to recognize areas requiring improvement. The skill gap analysis process begins by extracting skills from both candidate resumes and job descriptions using the predefined skills repository integrated within the framework. The extracted skill sets are compared to determine the degree of alignment between candidate qualifications and job requirements. Skills appearing in both the resume and job description are categorized as matching skills, whereas skills specified in the job description but absent from the resume are identified as missing skills.

Experimental results demonstrated that the proposed framework effectively identified candidate competencies and highlighted skill deficiencies. Candidates possessing a greater number of required skills generally achieved higher similarity scores and ranking positions. Conversely, applicants lacking important technical competencies obtained lower suitability scores, indicating the influence of skill alignment on recruitment outcomes. The automated identification of missing skills enables recruiters to perform more informed evaluations and supports transparent candidate assessment. A sample skill gap analysis result generated by the SmartHire+ framework. The dashboard displays identified skills, missing competencies, and overall candidate evaluation results obtained from the analysis process. To further illustrate the effectiveness of the proposed module, a sample skill comparison result is presented in Table IV.

Table – IV Sample Skill Gap Analysis Results

Category	Skills
Skills Extracted from Resume	SQL, Java, HTML, CSS, JavaScript, React
Skills Required in Job Description	SQL, Java, HTML, CSS, JavaScript, React, Python
Matching Skills	SQL, Java, HTML, CSS, JavaScript, React
Missing Skills	Python

As observed from Table IV, the candidate possesses most of the required technical skills specified in the job description. However, Python was identified as a missing competency, indicating a potential area for improvement. Such insights enable recruiters to understand candidate readiness for specific roles while providing applicants with actionable feedback regarding their professional development nee ds.In addition to identifying skill deficiencies, the framework generates personalized recommendations based on the detected gaps. Career recommendations

are produced according to the candidate's existing skill profile, while course recommendations are provided to assist candidates in acquiring the missing competencies. These recommendations extend the functionality of the framework beyond traditional resume screening and contribute to continuous learning and career development. Overall, the experimental results confirm that the Skill Gap Analysis module effectively identifies competency gaps and generates meaningful recommendations. This capability enhances the practical usefulness of the SmartHire+ framework by supporting both recruitment decision-making and candidate skill development, thereby contributing to a more intelligent and comprehensive talent acquisition process.

4.5 User Interface Implementation

To demonstrate the practical applicability of the proposed SmartHire+ framework, a web-based user interface was developed using the Flask framework along with HTML, CSS, and JavaScript technologies. The interface serves as a communication layer between recruiters and the underlying machine learning modules, enabling users to perform resume screening, candidate evaluation, and skill analysis through a simple and interactive environment. The primary objective of the user interface is to simplify the recruitment workflow while providing recruiters with meaningful analytical insights in real time. The developed application consists of two major interfaces: the Resume Upload Interface and the Candidate Analysis Dashboard. The Resume Upload Interface allows recruiters to upload candidate resumes in PDF format and enter the corresponding job description. Once the required inputs are provided, the system automatically initiates the analysis process and forwards the data to the backend processing modules. The interface incorporates input validation mechanisms to ensure that all necessary information is available before candidate evaluation begins.

The Candidate Analysis Dashboard presents the results generated by the SmartHire+ framework in a structured and visually organized manner. The dashboard displays similarity scores obtained from TF-IDF, Bag-of-Words, Word2Vec, and Sentence-BERT models, along with the overall matching score calculated by the framework. These scores provide recruiters with a clear understanding of the relationship between candidate qualifications and job requirements. In addition to similarity analysis results, the dashboard displays the skills identified from the candidate resume and highlights missing competencies detected through the skill gap analysis module. This information enables recruiters to quickly assess candidate strengths and determine whether additional skills are required for a particular role. The dashboard further supports recruitment decision-making by presenting recommendation outputs generated by the framework. The developed user interface provides an intuitive and responsive environment that integrates resume screening, candidate ranking, and skill analysis functionalities within a single platform. The seamless interaction between the frontend and backend modules ensures efficient processing and real-time result generation, making the framework suitable for practical deployment in recruitment environments.

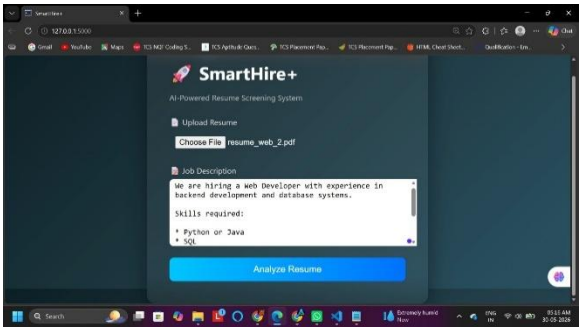

Fig. 2. Resume upload interface of the proposed SmartHire+ framework.

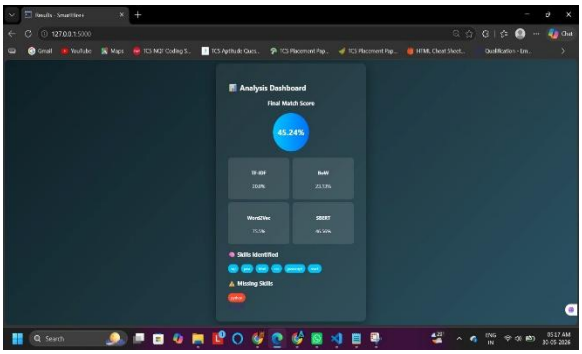

Fig. 3. Candidate analysis dashboard displaying similarity scores, identified skills, missing competencies, and evaluation results.

The successful implementation of the web application demonstrates the practicality of the proposed

framework and validates its ability to support intelligent recruitment decision-making through an accessible and user-friendly interface.

4.6 Discussion

The experimental results demonstrate that the proposed SmartHire+ framework effectively integrates Natural Language Processing, Machine Learning, and Learning-to-Rank techniques to support intelligent recruitment decision-making. The combination of multiple similarity analysis methods and advanced ranking algorithms enabled the framework to accurately evaluate candidate suitability and prioritize applicants according to their relevance to specific job requirements. The obtained results validate the effectiveness of the proposed methodology in addressing the limitations of conventional resume screening approaches.The comparative analysis of machine learning models revealed that ensemble-based algorithms achieved superior classification performance. Both Random Forest and Gradient Boosting obtained the highest accuracy of 75.50%, indicating their ability to effectively capture relationships between candidate features and job requirements. These results suggest that ensemble learning techniques are well suited for recruitment applications where candidate profiles often contain diverse and complex information. The satisfactory performance achieved by all evaluated models further confirms the reliability of the extracted features and similarity measures used within the framework.

The candidate ranking module produced particularly significant results. The LambdaMART-based ranking model achieved an average NDCG score of 0.9895, indicating excellent ranking quality and a strong alignment between generated rankings and ideal candidate ordering. This outcome demonstrates the capability of the proposed framework to consistently place highly relevant candidates at the top of the recommendation list. Such performance is highly beneficial in practical recruitment environments, where recruiters often need to review large numbers of applications within limited time constraints. The results also highlight the importance of combining lexical and semantic similarity analysis techniques. Traditional keyword-based approaches may fail to recognize related concepts expressed using different terminology. By integrating TF-IDF, Bag-of-Words, Word2Vec, and Sentence-BERT representations, the framework was able to capture both surface-level textual similarities and deeper contextual relationships between resumes and job descriptions. This multi-technique approach contributed significantly to improved candidate matching and evaluation accuracy.

Another important contribution of the framework is the Skill Gap Analysis module. The experimental results demonstrated that the system successfully identified missing competencies and provided meaningful recommendations based on candidate profiles. This functionality extends the scope of the framework beyond candidate screening by offering actionable insights for professional development. As a result, the system benefits both recruiters and job seekers by supporting informed hiring decisions and continuous skill improvement. The successful deployment of the Flask-based web application further demonstrates the practical applicability of the proposed framework. The developed interface enables recruiters to perform resume analysis, candidate ranking, and skill assessment through a simple and interactive platform. The integration of analytical outputs within a unified dashboard improves usability and facilitates efficient recruitment management.

Overall, the findings indicate that the proposed SmartHire+ framework provides an effective solution for intelligent resume screening, candidate ranking, and skill gap analysis. The combination of machine learning, semantic text analysis, and ranking techniques enhances recruitment efficiency, reduces manual screening effort, and supports more accurate candidate selection. These results confirm the potential of AI-driven recruitment systems to improve talent acquisition processes and support data-driven hiring practices in modern organizations.

V. CONCLUSION AND FUTURE WORK

This paper presented SmartHire+, an intelligent recruitment framework designed to automate resume screening, candidate evaluation, ranking, and skill gap analysis through the integration of Natural Language Processing, Machine Learning, and Learning-to-Rank techniques. The proposed framework addresses the challenges associated with traditional recruitment processes by reducing manual screening effort and providing data-driven support for candidate selection.

By combining multiple similarity analysis methods, including TF-IDF, Bag-of-Words, Word2Vec, and Sentence-BERT, the system effectively captures both lexical and semantic relationships between candidate resumes and job descriptions.

Experimental evaluation demonstrated the effectiveness of the proposed approach in recruitment applications. Among the evaluated classification models, Random Forest and Gradient Boosting achieved the highest accuracy of 75.50%, indicating reliable candidate classification performance. Furthermore, the LambdaMART-based ranking model achieved an average NDCG score of 0.9895, demonstrating excellent candidate prioritization capability and validating the suitability of Learning-to-Rank techniques for intelligent recruitment systems. The integration of skill gap analysis and recommendation generation further enhanced the functionality of the framework by providing actionable insights for candidate development and career improvement.

A Flask-based web application was successfully developed to provide recruiters with an interactive platform for resume analysis and candidate evaluation. The developed system enables real-time recruitment support through candidate ranking, skill assessment, and recommendation generation, thereby improving recruitment efficiency and decision-making accuracy. The obtained results confirm that the proposed SmartHire+ framework provides a practical and scalable solution for modern talent acquisition and intelligent hiring processes.

Future research can focus on integrating advanced transformer-based language models and Large Language Models (LLMs) to improve semantic understanding of resumes and job descriptions. Additional candidate assessment factors such as interview performance, technical assessments, and behavioral evaluation may also be incorporated to provide a more comprehensive representation of candidate suitability. Furthermore, multilingual resume analysis, cloud-based deployment, real-time recruitment platform integration, and explainable Artificial Intelligence (XAI) techniques can be explored to enhance system scalability, transparency, and applicability across diverse recruitment environments. These enhancements have the potential to further improve the effectiveness of intelligent recruitment systems and support more accurate talent acquisition strategies.

REFERENCES

[1] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in Proc. Int. Conf. Learn. Representations (ICLR), 2013, pp. 1–12.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” in Proc. North American Chapter Assoc. Comput. Linguistics (NAACL-HLT), 2019, pp. 4171–4186.

[3] N. Reimers and I. Gurevych, “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks,” in Proc. Conf. Empirical Methods Natural Language Processing (EMNLP), 2019, pp. 3982–3992.

[4] C. Burges, R. Ragno, and Q. Le, “Learning to Rank with Nonsmooth Cost Functions,” in Advances in Neural Information Processing Systems, vol. 19, 2007, pp. 193–200.

[5] Q. Wu, C. Burges, K. Svore, and J. Gao, “Adapting Boosting for Information Retrieval Measures,” Information Retrieval Journal, vol. 13, no. 3, pp. 254–270, 2010.

[6] G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in Advances in Neural Information Processing Systems, vol. 30, pp. 3146–3154, 2017.

[7] L. Breiman, “Random Forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[8] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[9] J. H. Friedman, “Greedy Function Approximation: A Gradient Boosting Machine,” Annals of Statistics, vol. 29, no. 5, pp. 1189–1232, 2001.

[10] S. Robertson and H. Zaragoza, “The Probabilistic Relevance Framework: BM25 and Beyond,” Foundations and Trends in Information Retrieval, vol. 3, no. 4, pp. 333–389, 2009.

[11] F. Pedregosa et al., “Scikit-learn: Machine Learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.

[12] R. Smith, “An Overview of the Tesseract OCR Engine,” in Proc. Ninth Int. Conf. Document Analysis and Recognition, 2007, pp. 629–633.
[13] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. Sebastopol, CA, USA: O’Reilly Media, 2009.
[14] D. Jurafsky and J. H. Martin, Speech and Language Processing, 3rd ed. Stanford, CA, USA: Stanford University, 2023.
[15] A. Géron, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 3rd ed. Sebastopol, CA, USA: O’Reilly Media, 2022.