Habib University
Summer Tehqiq Research Project II

AI-Driven Job Selection: Revolutionizing Talent Acquisition with Machine Learning-Powered Resume Screening and Candidate Profiling

Author: Areesha Amir

Supervisor: Dr. Saleha Raza

1 Introduction

In recent years, machine learning has revolutionized every field ranging from healthcare to finance, driving innovation and efficiency at an unprecedented scale. One of the key strengths of machine learning models is their ability to continually improve performance by learning from previous outcomes. This adaptability and scalability allow for ongoing learning and constant enhancement, ensuring that the models can evolve and refine their accuracy over time.[1]

This research aims to address these gaps by experimenting with a combination of advanced machine learning models trained on extensive and diverse datasets for resume screening and job matching. Unlike existing platforms, this model will deeply analyze resumes and job descriptions to align candidates with suitable positions. Additionally, it will offer job seekers actionable insights to optimize their resumes and enhance their chances of selection. This approach seeks to make the screening process more efficient and effective, providing value that current platforms do not fully address.

Talent acquisition is a crucial yet complex and time-consuming function within human resources (HR). Despite the abundance of job platforms offering numerous opportunities, several persistent challenges impact the effectiveness and efficiency of the hiring process.[2]:

1. **Overwhelming Volume of Candidates:** Large job markets often present an overwhelming number of job seekers, making it practically impossible for HR teams to manually review each resume thoroughly. This sheer volume leads to a slow, inefficient hiring process and places a substantial resource burden on companies.

2. **Lack of Standardization in Resumes:** Resumes come in a variety of formats and structures, creating inconsistency and making it difficult for HR professionals to assess candidates uniformly. This lack of standardization necessitates extensive manual effort and increases the risk of overlooking qualified candidates.

3. **Evaluating Job Fit Accurately:** Accurately determining whether a candidate is capable of performing a job involves complex mapping of resumes to job descriptions. This process is often imprecise and can result in mismatches between candidates and job requirements.

4. **Bias Reduction in Hiring:** Traditional resume screening methods, including applicant tracking systems (ATS), can inadvertently introduce biases into the hiring process. While ATS can be effective in filtering large volumes of applications, they often fail to address nuanced candidate qualities and can perpetuate biases.

5. **Suggestions and Skill Recommendations for Weak Resumes:** Many job seekers receive little to no feedback on why their resumes are rejected, missing out on valuable insights for improvement.

According to LinkedIn, 64% of job seekers are hired through referrals, underscoring the inherent biases and inefficiencies in the job search process. By tackling these interconnected challenges, the project aims to streamline the hiring process, enhance the accuracy of candidate selection, and alleviate the resource strain on HR departments, ultimately revolutionizing talent acquisition through innovative machine learning solutions:

1. **Efficient Candidate Filtering:** By employing advanced machine learning algorithms, the project enhances the efficiency of resume screening, enabling quick and accurate filtering of candidates amidst the vast volume of job applications.

2. **Standardized Resume Interpretation:** Leveraging natural language processing (NLP) techniques, the project standardizes diverse resume formats, reducing the manual effort required to match candidates with job descriptions and minimizing the risk of missing suitable candidates.

3. **Enhanced Job Fit Assessment:** Utilizing resume embeddings and clustering, the project evaluates job fit by comparing cosine similarity scores between resume and job embeddings. This approach improves the alignment between candidate skills and job requirements.

4. **Skill Improvement and Matching System:** The project compares strong resumes with weaker ones to provide skill improvement suggestions and uses the Gale-Shapley algorithm to ensure that both candidate and recruiter preferences are matched, enhancing overall satisfaction and job fit.

5. **Bias Reduction in Hiring:** Data-driven techniques are employed to reduce biases in the hiring process, promoting a fairer evaluation and selection of candidates.

By addressing these pressing issues, the project aims to streamline the hiring process, improve the accuracy of candidate selection, and alleviate the resource strain on HR departments, ultimately revolutionizing talent acquisition through innovative machine learning solutions.

1.1 Problem Statement

In the contemporary job market, both job seekers and companies face significant challenges that hinder efficient and effective hiring. Job seekers struggle to tailor their resumes for numerous postings, while companies grapple with the overwhelming volume of applications and lack of standardization in resume formats. Additionally, accurately evaluating job fit and reducing biases in the hiring process remain persistent issues. Current platforms fall short in addressing these challenges comprehensively.

This research aims to address these gaps by leveraging advanced machine learning models and natural language processing techniques. The proposed solution will enhance efficiency by rapidly processing applications, standardizing diverse resume formats, and improving job fit assessments. Unlike existing platforms, this approach will deeply analyze resumes and job descriptions to better align candidates with suitable positions. Additionally, it will provide job seekers with actionable insights to optimize their resumes and increase their chances of selection. By transforming the talent acquisition process, this research seeks to offer substantial improvements over current methods and address the limitations of existing solutions.

1.2 Research Gap

The integration of AI into recruitment has revolutionized talent acquisition, yet significant gaps persist in existing systems that limit their full potential. My AI-driven recruitment project seeks to address these limitations, offering a more comprehensive and efficient solution that enhances both candidate experience and recruitment outcomes.

Run-Time Efficiency and Complexity: One of the critical challenges identified in studies, is the issue of run-time complexity. Existing systems often categorize resumes into occupational categories to streamline the matching process but suffer from high error rates and lower classification accuracy. These inefficiencies lead to prolonged processing times and less reliable candidate-job matches. My project addresses this by employing advanced embedding models and real-time data processing techniques, significantly reducing computational overhead. This approach accelerates the matching process and enhances its precision, ensuring quicker and more accurate identification of the best candidates.

Dynamic Job Recommendations: Current systems often provide static job recommendations based on past data, which can quickly become outdated. Your project incorporates live job posting integration via LinkedIn scraping, ensuring that candidates receive up-to-date job recommendations relevant to their current skill set and career goals.

Optimized Candidate-Recruiter Matching: Existing systems may not fully account for the preferences of both candidates and recruiters. The Gale-Shapley matching algorithm in your project optimizes the matching process by considering both parties' preferences, leading to a more satisfactory recruitment outcome for both sides.

Real-Time Feedback and Continuous Learning: Existing AI-driven recruitment systems typically rely on static processes, limiting their ability to adapt to new information or changing trends in the job market. My project introduces real-time feedback mechanisms, enabling continuous learning from new data. This dynamic approach improves the accuracy of matches over time and ensures the system remains responsive to shifts in job market demands and candidate profiles.

Enhanced Candidate Recommendations and Suggestions: Current AI-driven recruitment systems often lack robust mechanisms for offering personalized improvement suggestions to candidates. My project goes beyond simple matching by generating detailed recommendations for candidates, helping them understand gaps in their qualifications and providing actionable steps to improve their resumes, thereby increasing their chances of success in future job applications.

Scalability and Flexibility: Scalability is a common challenge in existing AI-driven recruitment systems, especially when dealing with large volumes of resumes and job postings. My project is designed to be highly scalable, capable of handling large datasets without performance degradation, and offers customization options for different industries, job roles, and organizational needs. This versatility makes my project a practical and adaptable solution for various recruitment scenarios.

User-Friendly Interface: Many AI-driven recruitment systems may lack a user-friendly interface for interacting with the technology. Your project includes a React application that provides an intuitive interface for resume uploads and integrates AI modules seamlessly, enhancing the user experience for both candidates and recruiters.

In summary, while existing AI-driven recruitment systems have made significant advancements, they still present gaps that limit their effectiveness. This research addresses these shortcomings by improving runtime efficiency, contextual understanding, real-time feedback, data integration, scalability, and integration with broader HR functions. By filling these gaps, my project provides a more comprehensive, efficient, and adaptable approach to recruitment, ultimately leading to better outcomes for both candidates and recruiters.

2 Literature Review

The recruitment process is critical to organizational success, yet it remains a complex and often challenging endeavor for both candidates and recruiters. Candidates face the pressure of presenting themselves effectively, usually navigating a highly competitive landscape where their skills and experiences must stand out. On the other hand, Recruiters are tasked with sifting through vast volumes of resumes, assessing not only the technical qualifications of candidates but also their cultural fit within the organization. This process can be time-consuming and fraught with biases, making it difficult to ensure that the right candidate is selected. Artificial intelligence (AI) emerges as a transformative force, facilitating individuals in uncovering ideal career paths and fostering continuous self-improvement. Leveraging AI technologies, individuals can navigate opportunities with precision, extending human cognition and addressing complexity more effectively [3].

AI's ability to learn from social interactions enables it to act as an intellectual mediator. Virtual assistants or chatbots, powered by natural language processing (NLP) technologies, facilitate interactions with candidates and employees, making the recruitment and HR processes more efficient and effective.[4]

A systematic review of the literature provides valuable insights into these advancements. One study on resume screening using natural language processing (NLP) and machine learning underscores the complexity of resume parsing, which involves extracting information from unstructured documents using advanced NLP techniques. The accuracy of resume parsers is influenced by factors such as writing style, syntax, and context, presenting significant challenges in handling unstructured language and document formatting. Despite these difficulties, the study demonstrates that machine learning and NLP technologies can efficiently parse resumes, provided they can also manage both textual and structural information from various formats, such as PDFs [5].

Another study explores a machine learning approach for automating resume screening and recommendation. Using a dataset of resumes sourced from online platforms, this research highlights the effectiveness of techniques like tokenization, stop-word removal, and feature extraction through Term Frequency-Inverse Document Frequency (Tf-Idf) in transforming unstructured text into data suitable for machine learning classifiers. This method also includes matching resumes with job descriptions, emphasizing the importance of preprocessing steps in enhancing the relevance of candidate recommendations [2].

Further research presents a hybrid system for classifying resumes and job postings by leveraging a knowledge-based system integrated with NLP techniques. This approach addresses the issue of runtime complexity by categorizing resumes and job posts under relevant occupational categories, thereby improving the efficiency of the matching process. The study also introduces a weighted matching system that evaluates candidates based on skills, education, experience, and loyalty, offering a more nuanced assessment of candidate suitability for specific job roles [6].

In conclusion, the integration of AI into the recruitment process has the potential to address many of the challenges faced by both candidates and recruiters. By automating time-consuming tasks such as resume screening and providing real-time performance feedback, AI enhances the efficiency and fairness of recruitment and HR processes. The studies reviewed highlight the central role of machine learning and NLP techniques in these advancements, demonstrating their capacity to parse and analyze unstructured data, ultimately leading to more accurate and effective candidate selection.

3 Methodology

The research methodology was organized into distinct modules, each developed and tested independently before integration into the overall system. The backend development was initially conducted using Jupyter Notebook, followed by evaluation and comparison with alternative approaches. The most effective methods were subsequently integrated into a Flask backend and then connected to the front-end system built with React. The overall system architecture was updated to accommodate these integrations. The modules and their development processes are detailed below:

3.1 Smart Resume Categorization

The smart resume categorization system processes an input resume and classifies it into predefined job categories, such as Software Engineering, Data Science, or Marketing. For recruiters, it streamlines the process of sorting and identifying resumes that match specific job categories, making it easier to manage and evaluate applications. For candidates, it ensures that their resumes are accurately categorized based on their skills and experiences, enhancing the likelihood of their resumes being noticed for relevant job openings.

Data Collection The data for training the resume categorization model was sourced from Kaggle, providing a diverse and comprehensive dataset. The dataset was chosen for its relevance and quality training and testing the categorization model. Data collection involved aggregating resumes from various sources to ensure a robust dataset for model training.

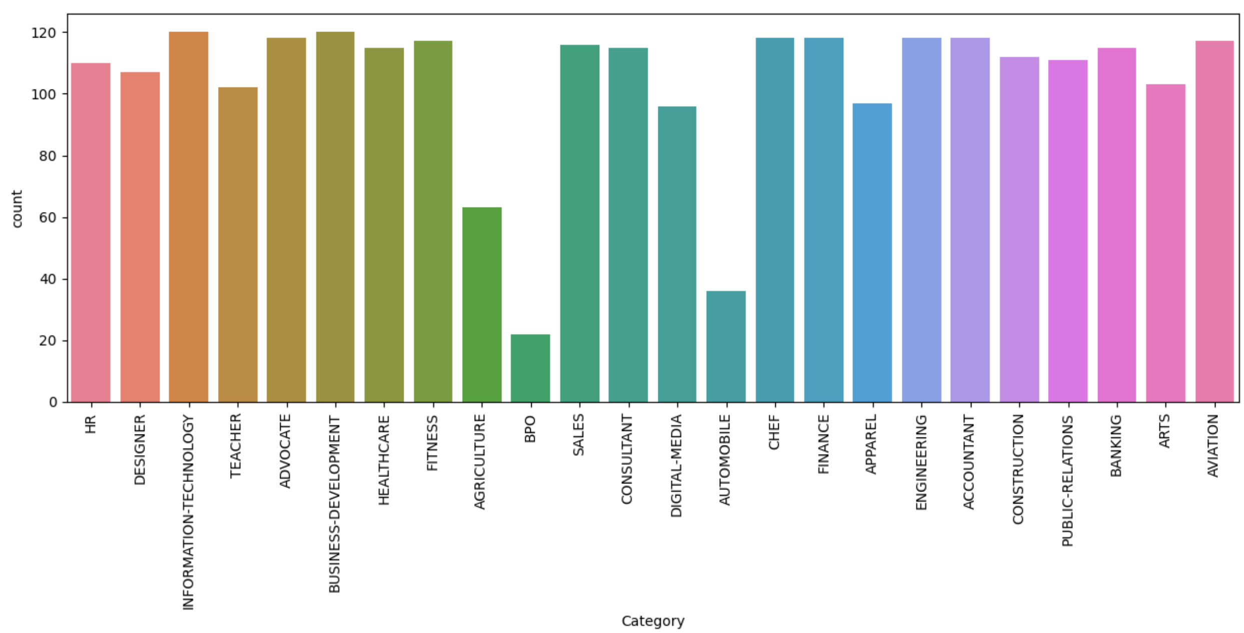


Fig. 1. Data Collection Process for Resume Categorization Model

Figure 1 displays bar graphs representing the distribution of category fields and the number of individuals within each. This data was used for model training, with adjustments made to balance the categories and ensure accurate predictions across all fields.

Data Analysis The cleaned dataset was subjected to a thorough analysis using a range of statistical and exploratory data analysis techniques. This included descriptive statistics and data visualization to explore data distribution and identify significant patterns pertinent to the categorization process. To mitigate model bias, the data was balanced and subsequently categorized into two distinct divisions: Division X, containing features relevant to specific job categories, and Division Y, which included the job category names corresponding to the features in Division X.

The dataset was then partitioned into two subsets: 80% for training the model and 20% for testing its performance. The models employed in this analysis were Random Forest, Logistic Regression, Support Vector Machine, Decision Tree, and Naive Bayes, all sourced from the scikit-learn library. Scikit-learn was chosen for its comprehensive suite of machine learning algorithms and ease of use. Each model was trained using the training subset and evaluated using the testing subset to assess accuracy. The accuracy results for category prediction is given below.

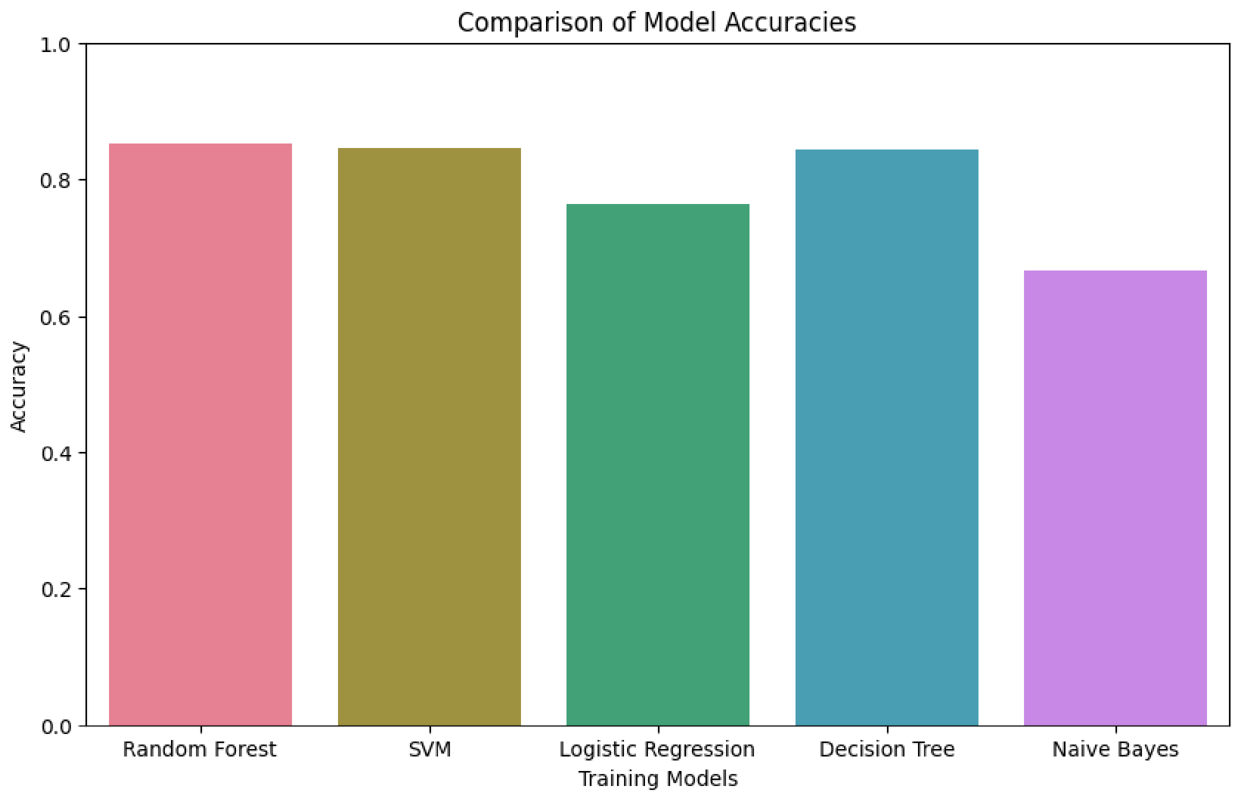


Fig. 2. Accuracy Scores for Resume Categorization Models

The performance of the models was evaluated by comparing predicted categories to actual categories, with Random Forest achieving the highest accuracy at 0.852. This superior performance can be attributed to its ensemble learning approach, which aggregates predictions from multiple decision trees to improve accuracy. The Support Vector Machine and Decision Tree models followed, with accuracies of 0.8454 and 0.84375, respectively. Logistic Regression and Naive Bayes demonstrated lower performance, with accuracies of 0.763 and 0.66, respectively. These results suggest that the simplicity of Logistic Regression and the assumption of feature independence in Naive Bayes may have contributed to their relatively lower performance.

Results and Conclusion The system leverages the Random Forest Classifier due to its superior performance compared to other models. Its proficiency in handling high-dimensional data makes it an ideal choice for the intricate and multifaceted nature of resume data. [2] The model's ensemble approach, which aggregates multiple decision trees, ensures robust and reliable predictions, thereby significantly enhancing the overall performance of the resume categorization and job prediction system. The model achieved an accuracy rate of 85% application.

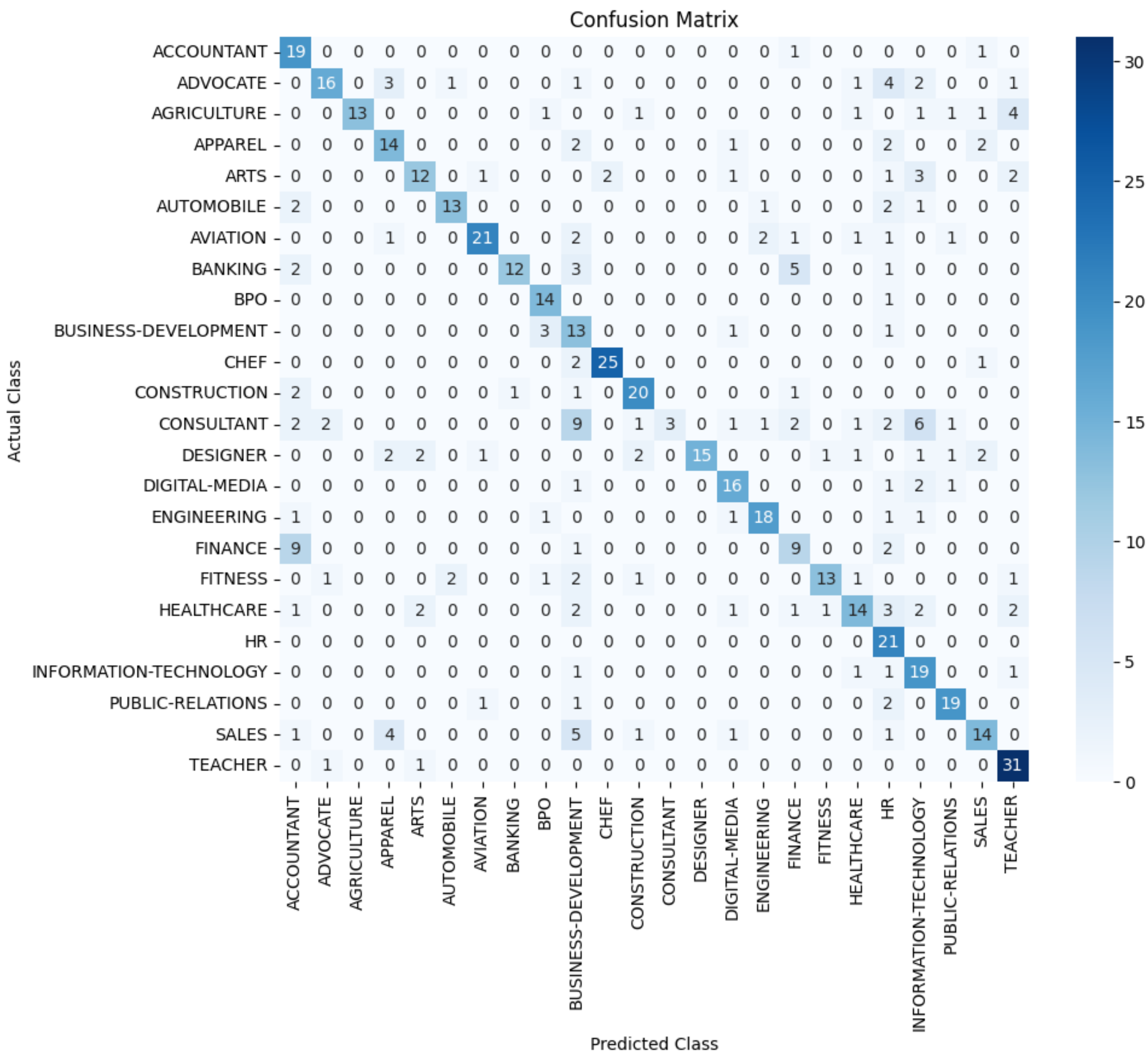


Fig. 3. Job Positions and Listings

The predictions made by the model during the testing phase, as shown in Figure 3, provided valuable insights that were instrumental in refining the model to enhance its performance. Even when incorrect, predictions often fell within related fields, such as categorizing an accountant under finance or misclassifying a business role as consultant or sales, reflecting the model's nuanced understanding of job categories. However, occasional misclassifications, like placing a business development role under the chef category or a teaching position under healthcare, highlight areas for further improvement. Addressing these inaccuracies is crucial for enhancing the system's precision and reliability in real-world applications.

3.2 Top Jobs Finder

This module consists of two key components:

1. **Job Recommendation Model:** This model analyzes an input resume and generates the top three job recommendations tailored to the candidate's skills and qualifications.

2. **LinkedIn Web Scraping Tool:** Developed using Selenium, this tool automates the extraction of job postings from LinkedIn. Selenium is a powerful web automation framework that allows for dynamic interaction with web pages, making it ideal for scraping data from LinkedIn efficiently.
   The job recommendation model enhances the recruitment process by delivering precise job matches based on the candidates' resumes, thereby improving efficiency in identifying suitable candidates. For candidates, it offers tailored job suggestions, boosting their chances of finding positions that closely match their skills and career goals.

Data Collection The data utilized for training and testing the job recommendation system was sourced from a Kaggle dataset, which includes 376 job positions, encompassing a total of 1,615,942 data points. This dataset provides comprehensive job details, including titles, descriptions, and other relevant features.

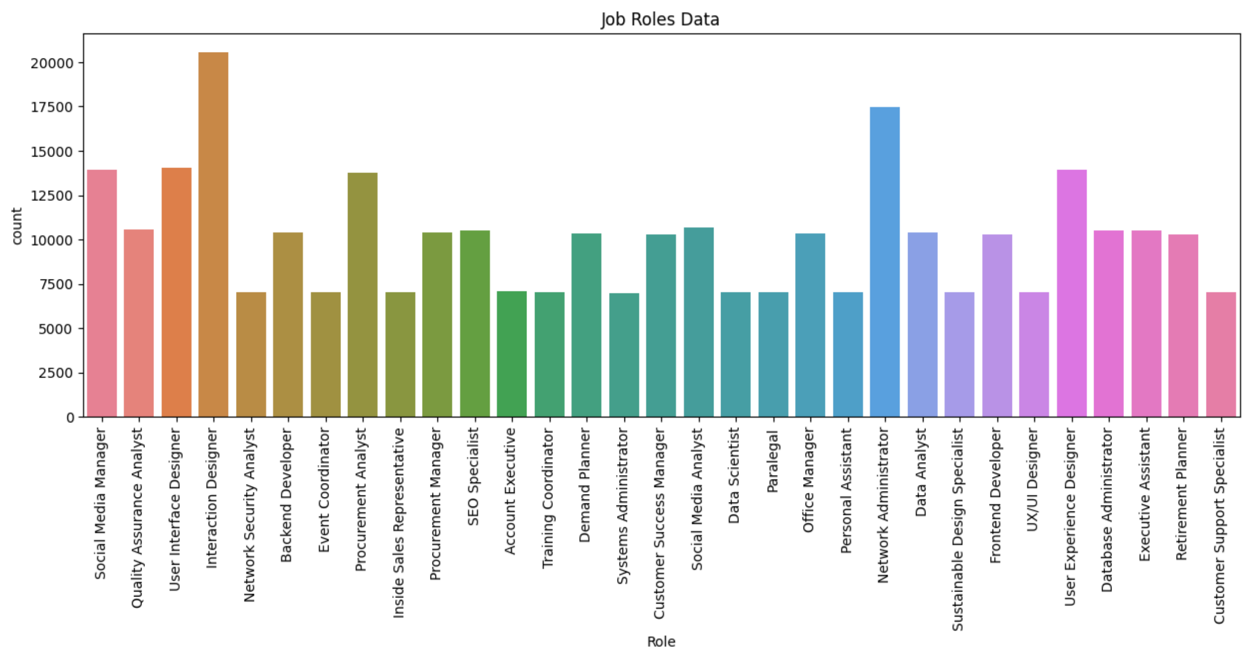


Fig. 4. Job Positions and Listings

This graph displays the number of job listings for various job titles, with the x-axis representing the job titles and the y-axis indicating the number of listings for each title. It provides a visual summary of the distribution of job opportunities across different job roles.

Data Analysis Similar to the categorization model, the job recommendation model in the system adhered to these same analytical steps, ensuring consistency and accuracy in the overall data processing and categorization workflow. In the process of developing this job recommendation model, I experimented with three different machine learning algorithms: a neural network, SVM, and random forest, using a large dataset containing 1,615,942 data points. Each model was tested for accuracy and computational efficiency.

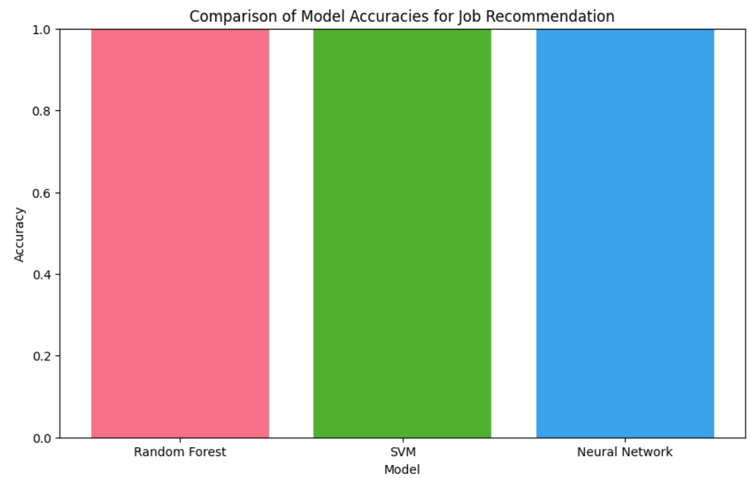


Fig. 5. Accuracy Score for Job Recommendation Model

As illustrated in the figure, all three models—neural network, SVM, and random forest—achieved a full accuracy score on the dataset.While the neural network demonstrated strong predictive power, it required significant computational resources due to its complexity. The SVM model, known for its effectiveness on smaller datasets, struggled with the scale and complexity of this large dataset, resulting in suboptimal performance during previous tests.

Results and Conclusion The above figure shows that all three models—neural network, SVM, and random forest—achieved a full accuracy score on the dataset with 1,615,942 data points. However, I chose to go with the random forest model because, while the neural network was powerful, it demanded extensive computational resources due to its complexity, making it less practical for real-world application. The SVM model, although effective in smaller datasets, struggled with the scale and complexity of the data, resulting in less reliable performance during previous tests. In contrast, the random forest model, with its ensemble learning approach, not only maintained high accuracy but also offered greater robustness and scalability. It efficiently handled the large dataset while minimizing the risk of overfitting, making it the most suitable choice for the job recommendation task. After selection, the model was integrated in the flask backend.

3.3 Skill Enhancement Advisor

Data Collection Data for the Skill Enhancement Advisor was collected from educational resources, skill databases, and job descriptions to identify key skills required for various roles. These sources were chosen to ensure a comprehensive understanding of the skills landscape in engineering and tech fields. The job descriptions were embedded using the SentenceTransformer model to create a vector representation that could be compared with candidate profiles.

Data Analysis The analysis involved multiple steps:

1. **Resume Preprocessing:** Candidate resumes were processed using text normalization techniques, including tokenization, stop word removal, and lemmatization. Embeddings were generated using the 'all-MiniLM-L6-v2' model from Sentence Transformers, which produces 384-dimensional vectors, as shown in the code below. This model is effective due to its balance between computational efficiency and the ability to capture rich semantic information, making it well-suited for detailed resume analysis.

```
from sentence_transformers import SentenceTransformer
from sklearn.metrics.pairwise import cosine_similarity

model = SentenceTransformer('all-MiniLM-L6-v2')
resume_embeddings = model.encode(resume_texts, show_progress_bar=True)

def calculate_similarity_scores(resume_embeddings, job_embeddings):
    scores = cosine_similarity(resume_embeddings, job_embeddings)
    return scores
```

2. **Similarity Calculation:** The cosine similarity between the resume embeddings and job description embeddings was computed to determine how well candidates' skills matched the job requirements as given in the code below:

```
# Function to calculate similarity score using embeddings
def calculate_similarity_scores(resume_embeddings, job_embeddings):
    scores = []
    for resume_embedding in resume_embeddings:
        resume_scores = cosine_similarity([resume_embedding], job_embeddings).
            flatten()
        scores.append(resume_scores * 100)  # Convert to percentage
    return np.array(scores)

similarity_scores = calculate_similarity_scores(resume_embeddings,
    job_embeddings)
```

3. **Clustering:** Resumes were clustered based on their similarity scores using the K-Means algorithm. This helped identify groups of candidates with similar profiles.

```
cluster_scores = {}
for cluster in range(num_clusters):
    cluster_members = [i for i, c in resume_clusters.items() if c == cluster]
    cluster_scores[cluster] = np.mean([average_scores[i] for i in
        cluster_members])
```

```python

# Identify the weaker cluster
weak_cluster = min(cluster_scores, key=cluster_scores.get)

# Identify the stronger cluster
strong_cluster = max(cluster_scores, key=cluster_scores.get)

# Generate recommendations for weaker candidates
def generate_recommendations(weaker_candidates, stronger_key_phrases):
    recommendations = {}
    for i, text in enumerate(weaker_candidates):
        missing_phrases = [phrase for phrase in stronger_key_phrases if phrase
            not in text]
        recommendations[i] = missing_phrases
    return recommendations

recommendations = generate_recommendations(weaker_candidates,
    stronger_key_phrases)
```

4. **Identification of Skill Gaps:** The analysis identified a weaker cluster—candidates whose resumes showed the least similarity to job requirements. For these candidates, skill gaps were identified by comparing their resumes with those in stronger clusters.

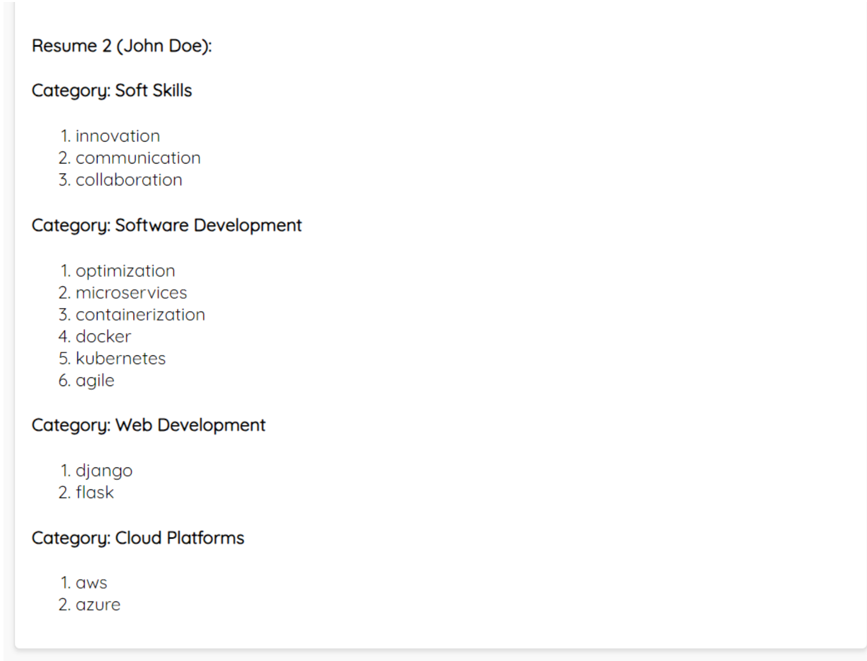
```python
# Perform K-means clustering on average scores
scores_array = average_scores.reshape(-1, 1)
num_clusters = 3
kmeans = KMeans(n_clusters=num_clusters, random_state=42)
clusters = kmeans.fit_predict(scores_array)

# Map resumes to their respective clusters
resume_clusters = {i: cluster for i, cluster in enumerate(clusters)}
```

The results of this analysis were used to generate personalized recommendations for candidates in the weaker cluster, advising them on which skills to develop to improve their chances of matching job requirements.

Results and Conclusion The Skill Enhancement Advisor successfully identified clusters of resumes with varying levels of alignment to job descriptions. The weaker cluster, characterized by lower average similarity scores, highlighted the need for targeted skill development in areas such as machine learning, Python programming, and data analysis.

The analysis provided actionable insights into skill enhancement, offering personalized advice for candidates on how to bridge the gap between their current skill set and the requirements of desired job roles. These recommendations are crucial for helping candidates improve their employability and better align with the expectations of employers in the tech industry.



Resume 2 (John Doe):

Category: Soft Skills

1. innovation
2. communication
3. collaboration

Category: Software Development

1. optimization
2. microservices
3. containerization
4. docker
5. kubernetes
6. agile

Category: Web Development

1. django
2. flask

Category: Cloud Platforms

1. aws
2. azure

Fig. 6. Sample Output for Skill Enhacement Advisor

A sample input of 3 resumes was provided, and the analysis identified the weaker cluster. The following output in Figure 5 shows the suggested skills for improvement for the candidate with the weak resume. The system has categorized the weaker candidate's skills into various domains such as Soft Skills, Software Development, Web Development, and Cloud Platforms. For each category, specific skills have been identified that the candidate already possesses, and based on the job description, these areas are where the candidate needs improvement.The analysis of John Doe's resume suggests that while the candidate has a solid foundation in multiple areas, there are key areas that need enhancement to better match the job requirements. This systematic categorization and recommendation approach ensures that candidates receive targeted advice, enabling them to focus their efforts on the most impactful areas of improvement.

Further testing and improvements are suggested to refine the model's accuracy and broaden the scope of recommendations for different domains in the industry. Future work may also explore the integration of real-time job market data to continuously update skill recommendations in response to evolving industry demands.

3.4 Intelligent Match Making System

The Intelligent Matchmaking System is an AI-driven platform that streamlines the recruitment process by accurately matching candidate resumes with job descriptions. Using advanced natural language processing and machine learning techniques, the system converts textual data into embeddings and calculates similarity scores to assess the alignment between candidate qualifications and job requirements.

Data Analysis The analysis involved developing a matching model using a similarity scoring algorithm, which compared the textual content of candidate resumes with job descriptions. This was accomplished by transforming the text into embeddings using the Sentence Transformer model (all-MiniLM-L6-v2) and calculating cosine similarity between them. The system also included mechanisms to extract and process resume content from PDFs, handling both structured and unstructured data. The Gale-Shapley algorithm was then applied to optimize the matching process by considering both candidate and recruiter preferences.

1. Preference List Generation: Creates preference lists for both candidates and recruiters based on similarity scores.

```python
    # Sort preferences for candidates and recruiters
    for candidate in candidates_pref:
        candidates_pref[candidate].sort(reverse=True, key=lambda x: x[0])

    for recruiter in recruiters_pref:
        recruiters_pref[recruiter] = dict(sorted(recruiters_pref[recruiter].
            items(), key=lambda x: x[1], reverse=True))

    print("Candidates' Preferences: ", candidates_pref)
    print("\nRecruiters' Preferences: ", recruiters_pref)
```

2. Gale-Shapley Matching Algorithm: Implements the Gale-Shapley algorithm to produce optimal matches between candidates and job positions.

```python
        # Gale-Shapley matching algorithm
        def gale_shapely(candidates, recruiters):
            free_candidates = list(candidates.keys())
            engaged_candidates = {}
            engaged_recruiters = {r: None for r in recruiters}

            while free_candidates:
                candidate = free_candidates.pop(0)
                candidate_prefs = candidates[candidate]

                for _, job in candidate_prefs:
                    if engaged_recruiters[job] is None:
                        engaged_recruiters[job] = candidate
                        engaged_candidates[candidate] = job
                        break
                    else:
                        current_candidate = engaged_recruiters[job]
                        if recruiters[job][candidate] > recruiters[job][
                            current_candidate]:
                            engaged_recruiters[job] = candidate
                            engaged_candidates[candidate] = job
                            free_candidates.append(current_candidate)
                            break

            return engaged_candidates
```

```
 Candidates Preferences:  {'15941675': [(29.623019695281982, 'Data Analyst'), (28.442293405532837, 'Project Manager')], '16803215': [(38.155925273895264, 'Data Analyst'), (27.30576992034912, 'Project Manager')], '17926546': [(32.12410509586334, 'Data Analyst'), (23.062512278556824, 'Project Manager')], 'art director': [(14.187763631343842, 'Project Manager'), (6.851488351821899, 'Data Analyst')], 'harvey': [(41.39910936355591, 'Data Analyst'), (22.11850881576538, 'Project Manager')], 'it': [(45.461589097976685, 'Data Analyst'), (39.820098876953125, 'Project Manager')], 'lilly': [(70.08495330810547, 'Data Analyst'), (34.29850935935974, 'Project Manager')], 'sodapdf-converted': [(52.430856227874756, 'Data Analyst'), (40.40377140045166, 'Project Manager')], 'teacher': [(27.029377222061157, 'Project Manager'), (22.318939864635468, 'Data Analyst')]}

Recruiters Preferences:  {'Data Analyst': {'lilly': 70.08495330810547, 'sodapdf-converted': 52.430856227874756, 'it': 45.461589097976685, 'harvey': 41.39910936355591, '16803215': 38.155925273895264, '17926546': 32.12410509586334, '15941675': 29.623019695281982, 'teacher': 22.318939864635468, 'art director': 6.851488351821899}, 'Project Manager': {'sodapdf-converted': 40.40377140045166, 'it': 39.820098876953125, 'lilly': 34.29850935935974, '15941675': 28.442293405532837, '16803215': 27.30576992034912, 'teacher': 27.029377222061157, '17926546': 23.062512278556824, 'harvey': 22.11850881576538, 'art director': 14.187763631343842}}

Gale-Shapely:
Matches:  {'15941675': 'Data Analyst', '16803215': 'Data Analyst', '17926546': 'Project Manager', 'harvey': 'Data Analyst', 'it': 'Data Analyst', 'lilly': 'Data Analyst', 'sodapdf-converted': 'Project Manager'}

Matched Candidates with Scores:
Candidate: 15941675, Job: Data Analyst, Score: 29.62
Candidate: 16803215, Job: Data Analyst, Score: 38.16
Candidate: 17926546, Job: Project Manager, Score: 23.06
Candidate: harvey, Job: Data Analyst, Score: 41.40
Candidate: it, Job: Data Analyst, Score: 45.46
Candidate: lilly, Job: Data Analyst, Score: 70.08
Candidate: sodapdf-converted, Job: Project Manager, Score: 40.40
Organized

Matched Candidates with Scores:

Job: Project Manager
Candidate: sodapdf-converted, Score: 40.40
Candidate: 17926546, Score: 23.06

Job: Data Analyst
Candidate: lilly, Score: 70.08
Candidate: it, Score: 45.46
Candidate: harvey, Score: 41.40
Candidate: 16803215, Score: 38.16
Candidate: 15941675, Score: 29.62
```

In the sample output above, the results demonstrate the value of each candidate's fit for job roles based on their skills and preferences. Higher scores indicate a stronger alignment between a candidate's skills and the job requirements, reflecting a better fit. The Gale-Shapley algorithm optimally matches candidates to roles by balancing both candidates' preferences and recruiters' needs.[7] This process provides a clear ranking of candidates for each job role, with higher scores signifying the top candidates. For instance, candidates with the highest scores for 'Data Analyst' are deemed the most suitable for that role, ensuring that the final matches are both effective and aligned with all parties' preferences.

Results and Conclusion The results demonstrated that the system was able to effectively match candidates to job openings based on the calculated similarity scores. The Gale-Shapley algorithm provided a stable matching between candidates and recruiters, ensuring that the preferences of both parties were adequately considered. The system also featured an organized output of matches, grouping candidates by job titles and presenting the scores for each match. This structured and efficient approach to matchmaking showed promise in enhancing the recruitment process by improving the accuracy and relevance of job-candidate matches.

Results from the matchmaking model were evaluated based on matching accuracy and the quality of candidate-job fits. The conclusion provided insights into the effectiveness of the model and potential areas for refinement.

4 Research Results, Findings, Discussion:

The outcome of this research was the development of a comprehensive machine learning powered resume screening system that addresses several key challenges in the recruitment process. This system integrates advanced machine learning models, including category and job prediction, resume screening, skill improvement, and candidate-recruiter matching, to create a more efficient, scalable, and user-friendly recruitment solution. The system's architecture in Figure 7 showcases a robust framework capable of significantly enhancing both the candidate experience and recruitment outcomes.

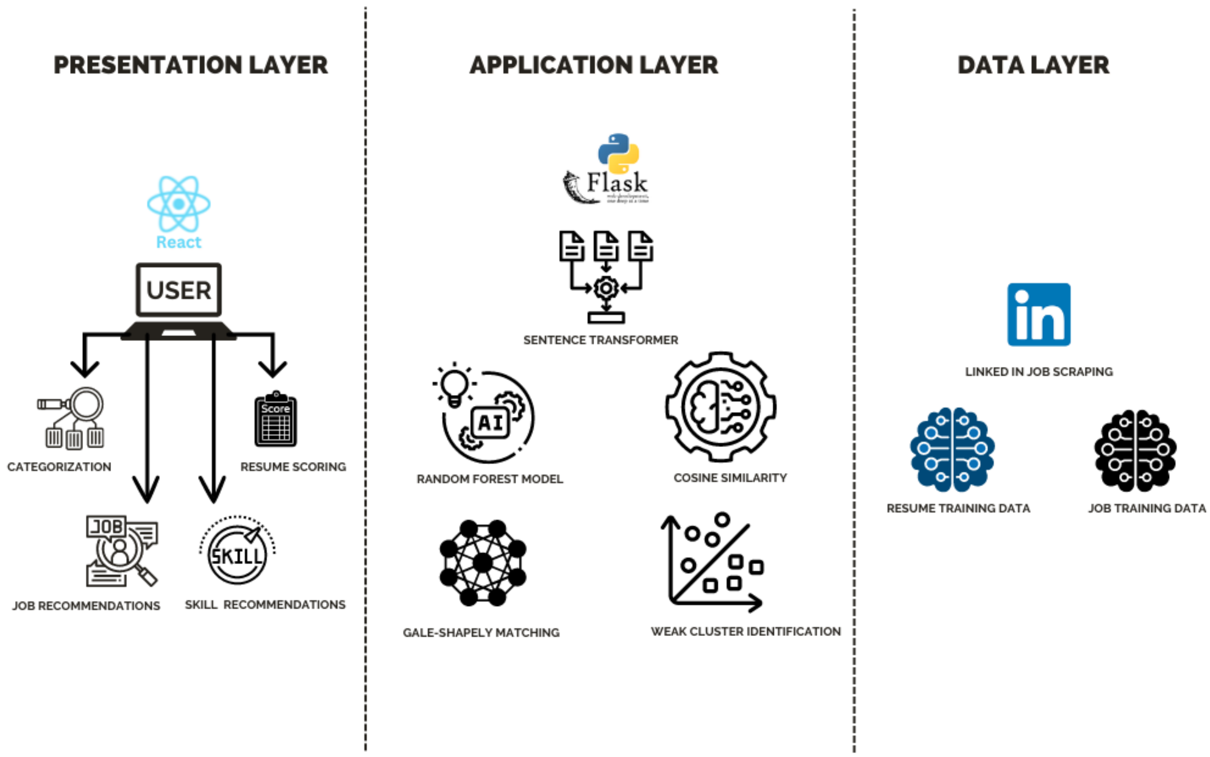


Fig. 7. Architectural Layers of the Machine Learning powered Resume Screening System

The AI-driven recruitment system is structured into three layers as showen in Figure 7, it includes: the Presentation Layer, Application Layer, and Data Layer. The **Presentation Layer**, built with React, is responsible for user interaction. It allows users to upload resumes for categorization, receive scores based on job descriptions, and obtain personalized job and skill recommendations. The **Application Layer**, implemented using Flask, handles the core processing logic. This layer includes components such as the Sentence Transformer, which generates embeddings for resumes and job descriptions, and the Random Forest Model, which provides job recommendations. Cosine Similarity is employed to assess how closely a resume matches a job description, contributing to accurate resume scoring. Additionally, the Gale-Shapley algorithm is used to optimize the matching process between candidates and job openings, while Weak Cluster Identification pinpoints resumes that may require improvement.

Lastly, the **Data Layer** focuses on data collection and management, including scraping job data from LinkedIn and utilizing databases of resumes and job descriptions to train the models. The system begins with data ingestion, collecting resumes and job data either from user uploads or through LinkedIn scraping. This data is then processed using the Sentence Transformer to create embeddings, which are analyzed with models like Random Forest and techniques like Cosine Similarity. The analysis results in resume scores, job recommendations, and skill improvement suggestions, with the Gale-Shapley algorithm ensuring optimal candidate-job matching. Weak clusters are identified for targeted enhancements, making the system a comprehensive tool for streamlining the recruitment process.

The key findings of this research include:

1. Improved Run-Time Efficiency: The use of embedding models and real-time data processing helped reduce computational overhead, which may lead to quicker and more accurate resume screening and job matching.
2. Dynamic Job Recommendations: Integrating live job postings allowed the system to provide candidates with up-to-date job recommendations that align with current qualifications and market trends.
3. Enhanced Matching Process: The Gale-Shapley algorithm was applied to consider both candidate and recruiter preferences, potentially leading to more satisfactory matches.
4. Continuous Learning: Real-time feedback mechanisms were introduced to enable the system to adapt to new data over time, which could improve its responsiveness to changes in the job market.
5. Scalability and Flexibility: The system was designed with scalability in mind, aiming to handle large datasets efficiently and offering customization options for various recruitment scenarios.
6. User-Friendly Interface: The React application provided an intuitive interface, which could enhance the experience for both candidates and recruiters.

5 Conclusion

Summarize the main outcomes of the research, reiterate the significance of the findings, and suggest potential future research directions or applications.

The outcome of this research was the development of a comprehensive AI-driven job selection system that addresses several critical challenges in the recruitment process. By integrating advanced machine learning models for category and job prediction, resume screening, skill improvement, and candidate-recruiter matching, the system offers a more efficient, scalable, and user-friendly recruitment solution. This research not only enhanced the accuracy and efficiency of the recruitment process but also significantly improved the candidate experience by providing real-time, personalized job recommendations and skill improvement suggestions.The research successfully addresses several persistent gaps and challenges in existing AI-driven recruitment systems. By improving run-time efficiency, enhancing the accuracy of job recommendations, optimizing candidate-recruiter matching, and offering real-time feedback and scalability, this project offers a more comprehensive, efficient, and adaptable approach to recruitment. The AI-driven job selection system developed through this research not only automates labor-intensive tasks such as resume screening and job matching but also empowers candidates with personalized improvement suggestions. It ensures that recruitment processes are faster, more accurate, and better aligned with the needs of both candidates and recruiters.

While the research findings suggest that the system has the potential to improve recruitment processes, it is important to recognize that further testing and validation are needed to fully assess its effectiveness. The research shows promise, but additional work is necessary to refine the models, address any limitations, and explore how the system performs across different industries and contexts. Given the success of this research, several potential directions for future work can be explored:

1. **Integration with Broader HR Functions:** Future research could explore the integration of this system with broader HR functions, such as performance evaluation, career development, and employee retention.
2. **Rigorous Testing:** Conduct thorough testing of the system across diverse datasets and scenarios to ensure robustness, accuracy, and reliability.

3. **Enhancing Natural Language Processing:** Further advancements in natural language processing (NLP) could be applied to improve the system's ability to understand and process complex resume formats and job descriptions.
4. **Integration with Skill Improvement Platforms:** One promising direction is to integrate the resume screening and recommendation system with platforms that offer online courses and skill improvement programs. This integration would not only enhance the value of our system for job seekers but also align it with practical solutions for continuous learning.
5. **Training Models for Specific Industries:** To improve the accuracy and relevance of our recommendations, the models can be trained with industry-specific data. This approach will enhance the system's ability to match candidates with job opportunities and skill development resources that are highly relevant to their desired industry.

In conclusion, the outcome of this research demonstrates the potential of AI to revolutionize the recruitment process, leading to better outcomes for all stakeholders involved. The final architecture in Figure 5, with its focus on testing, scalability, user experience, and dynamic adaptability, provides a solid foundation for future advancements in the field of AI-driven talent acquisition.

6 References

1. D. J. Joshi, A. Kulkarni, R. Pande, I. Kulkarni, S. Patil, and N. Saini, "Conference Paper Acceptance Prediction: Using Machine Learning," in *Machine Learning and Information Processing*, D. Swain et al., Eds., Advances in Intelligent Systems and Computing, vol. 1311, Singapore: Springer, 2021, pp. 143-144. doi: 10.1007/978-981-33-4859-2_14.
2. P. K. Roy, S. S. Chowdhary, and R. Bhatia, "A Machine Learning Approach for Automation of Resume Recommendation System," in *Proc. Int. Conf. Comput. Intell. Data Sci. (ICCIDS)*, 2019, pp. 2318-2327. doi: 10.1016/j.procs.2020.03.284.
3. M. H. Jarrahi, "Artificial intelligence and the future of work: Human-AI symbiosis in organizational decision making," Business Horizons, vol. 61, no. 4, pp. 577-586, Jul-Aug. 2018. https://doi.org/10.1016/j.bushor.2018.03.007
4. S. Khan, "Artificial Intelligence Virtual Assistants (Chatbots) are Innovative Investigators," *IJCSNS International Journal of Computer Science and Network Security*, vol. 20, no. 2, pp. 93-101, Feb. 2020.
5. Sinha, A.K., Amir Khusru Akhtar, M., Kumar, A. (2021). Resume Screening Using Natural Language Processing and Machine Learning: A Systematic Review. In: Swain, D., Pattnaik, P.K., Athawale, T. (eds) Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1311. Springer, Singapore. https://doi.org/10.1007/978-981-33-4859-2_21
6. A. Zaroor, M. Maree, and M. Sabha, "JRC: A Job Post and Resume Classification System for Online Recruitment," 2017. doi: 10.1109/ICTAI.2017.00123.
7. K. Dang, "Job Resume Matching — Matching candidates and vacancies using the modified Gale-Shapley," Medium, Jun. 24, 2023. [Online]. Available: https://medium.com/@kirudang/job-resume-matching-part-2-2-matching-candidates-and-vacancies-using-the-modified-37e39c3bf0a. [Accessed: Aug. 12, 2024].