Intelligent Resume Screening and Query Assistance: A Machine Learning and NLP Approach for Automated Candidate Evaluation and Optimization

1st Archit Choudhury
School of Computer Science, KIIT-DU
Bhubaneswar, Odisha, 751024
architchoudhury10@gmail.com

2nd Ayushman Panigrahi
School of Computer Science, KIIT-DU
Bhubaneswar, Odisha, 751024
ayushmannn2004@gmail.com

3rd Amitanshu Pattnaik
School of Computer Science, KIIT-DU
Bhubaneswar, Odisha, 751024
amitanshupattnaik@gmail.com

4th Jeet Sahoo
School of Computer Science, KIIT-DU
Bhubaneswar, Odisha, 751024
jeetsahoo2203@gmail.com

***Abstract*—Resume screening has become a laborious and ineffective procedure due to the rise in the number of job applications. An AI-driven resume screening and query assistant system that uses machine learning and natural language processing (NLP) to automate candidate evaluation is presented in this research. The system preprocesses resumes by converting textual data into numerical features using the TF-IDF vectorization algorithm, extracting text, and cleaning data. After evaluating three machine learning models—K-Nearest Neighbors (KNN), Support Vector Classifier (SVC), and Logistic Regression—Logistic Regression was chosen because of its ideal performance and small pickle file size. Furthermore, the Gemini API is integrated to evaluate resumes in a way that goes beyond conventional keyword-based analysis, offering a thorough score, strengths, shortcomings, and suggestions for improvement. The program uses AI-generated insights to improve applicant applications and classifies resumes according to job roles.**

Additionally, the system has an interactive query assistance to help customers optimize their resumes. The system's efficacy in automating screening with high accuracy and efficiency is demonstrated by experimental findings, which also ensure fair and objective evaluations while decreasing the workload of recruiters..

***Index Terms*—Resume Screening, Machine Learning, Natural Language Processing, TF-IDF, Gemini API, Logistic Regression**

I. Introduction

A. Problem Statement and Need for Automation in Hiring

Organizations in current hiring landscapes are confronted with an avalanche of applications for every available job, making manual screening of resumes an inefficient, time-consuming, and error-prone task. Traditional resume evaluation processes that rely heavily on human recruiters and rule-based Applicant Tracking Systems (ATS) are marred by inherent flaws such as scalability issues, cognitive biases, and variability in candidate assessment. Further, heuristic keyword-based approaches used by standard ATS regularly fail to account for the semantic relevance [1] of a candidate's qualifications, leading to false positives or unjustified rejection.

The need for a computerized, smart, and data-based resume screening process has become the most critical aspect in evading such inefficiencies. Artificial Intelligence (AI) and Machine Learning (ML) have demonstrated the potential to enhance precision, objectivity, and effectiveness in candidate evaluation, transforming the hiring model by reducing turnaround time and prejudice and increasing selection accuracy.

B. Inefficiencies in Manual Resume Screening

Manual resume screening involves several bottlenecks:

- Increased Processing Time and Cost Burden – Human evaluators spend considerable time going through countless applications, and this results in higher time-to-hire as well as costs of operations.
- Subjectivity and Prejudices – Recruiter prejudice, overt or subconscious, can lead to discriminatory hiring processes, which discredit diversity and inclusion initiatives.
- Unreliability in Assessment – Various recruiters are likely to vary in interpreting resumes, resulting in assessment inconsistency and lessening candidate selection reliability.
- Limited Contextual Understanding – Conventional ATS keyword-based screening has no semantic understanding, not understanding synonymous skills and correlated expertise.

C. Role of Machine Learning in Recruitment

Machine learning transformed the process of talent acquisition by bringing in automated classification, predictive analytics, and deep semantic analysis of resumes. The use of Natural Language Processing (NLP) [1] and supervised learning methods allows for smart parsing of resumes, context-dependent extraction of skills, and candidate ranking based on roles.

In this study, we utilize TF-IDF vectorization [1] to convert unstructured resume text to structured feature representations so that classification algorithms can make inferences of candidate fit for particular job positions. Various machine learning models were tested, such as:

- K-Nearest Neighbors (KNN) – A distance-based classification model that proved inefficient in large-scale resume datasets owing to computational complexity.
- Support Vector Classifier (SVC) – A strong model with high-dimensional feature discrimination but higher memory overhead.
- Logistic Regression – Chosen based on its lean pickle model size, high interpretability, and best balance of accuracy and compute efficiency.

D. Research Contributions and Objectives

The goal of this research is to close the gap between traditional resume screening and AI-based hiring intelligence by providing a machine learning-augmented resume screening and question answering assistant. [3] The major contributions of this work are:

- Automated Resume Categorization – Building an TF-IDF-driven machine learning pipeline that categorizes resumes to corresponding job titles.
- AI-Driven Resume Scoring and Analysis – Utilizing the Gemini API for creating AI-based candidate assessments, strengths/weaknesses analysis, and personalized recommendations.
- Model Selection Optimization – Comparing KNN, SVC, and Logistic Regression to determine the most computationally effective and accurate classification model for resume screening.
- Interactive Query Assistant – Creating an AI-driven chatbot assistant to offer real-time resume improvement recommendations based on AI-generated feedback.

II. Literature Review

A. Traditional Resume Screening Approaches

1) Manual Screening Inefficiencies: Human recruiters have long been the mainstay of traditional resume screening. But there are many inefficiencies in this approach. The process becomes extremely time-consuming for firms that receive hundreds or even thousands of applications for a single job opportunity. Recruiters only look at resumes for an average of 6-7 seconds, which frequently results in cursory assessments and the possible omission of highly competent applicants. In addition to time constraints, human prejudices including those based on gender, race, or educational background might affect manual screening. Organizational initiatives to promote diversity and inclusion may be hampered by these biases. Furthermore, because human judgment is subjective, various recruiters may evaluate the same CV in completely different ways, leading to inconsistent ratings.

2) Keyword-Based Applicant Tracking Systems (ATS): Applicant Tracking Systems (ATS) were used by many firms to address the drawbacks of manual screening. [4] These systems employ keyword matching to screen resumes according to predetermined standards, such as job titles, abilities, and credentials. ATS has a lot of drawbacks, even though it has increased scalability and decreased manual labor. Because keyword-based systems lack semantic knowledge, they frequently miss contextual relevance or synonymous concepts. If the job description calls for "AI modeling," for example, a resume that mentions "machine learning" may be turned down, despite the fact that the two terms are closely related. Furthermore, applicant tracking systems (ATS) are inflexible and unable to adjust to changing job specifications or complex applicant profiles. False positives—unqualified candidates passing the filter—and false negatives—qualified candidates being rejected—are frequently the results of this rigidity.

B. Machine Learning in Resume Screening

1) Supervised vs. Unsupervised Approaches: Supervised learning: In this, models are trained using labeled data, i.e., resumes with appropriateness scores or matching job categories assigned to them. Classification tasks often utilize algorithms such as Random Forests, Support Vector Machines (SVM), and Logistic Regression. It is good if there is availability of historical data since it enables the model to learn trends and make accurate predictions. Getting a large, high-quality labeled dataset, though, is challenging.

Unsupervised Learning: In cases when there is minimal labeled data, learning approaches such as topic modeling and clustering are utilized. These methods classify resumes based on similarities in content, such as education, experience, or skills. While more flexible, this often cannot match the accuracy of supervised learning and can sometimes require additional post-processing to comprehend the results.

2) Use of Vectorization Techniques for Text Processing: A critical step in ML-based resume screening is converting unstructured text data into numerical features that algorithms can process. Vectorization techniques play a pivotal role in this transformation.

Bag of Words (BoW): This simple approach represents text as a collection of word frequencies. While easy to implement, BoW ignores word order and context, limiting its effectiveness.

Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a more advanced technique that weights words based on their importance in a document relative to a corpus. It reduces the impact of common words (e.g., "the," "and") while emphasizing domain-specific terms (e.g., "Python," "data analysis"). TF-IDF is widely used in resume screening due to its ability to capture meaningful patterns in text data.

C. AI-Driven Resume Evaluation

1) NLP Models for Resume Parsing: Resume screening has been revolutionized due to Natural Language Processing (NLP), which enables computers to read and comprehend human language. Modern NLP models such as BERT, GPT, and Gemini employ transformer topologies in order to accomplish tasks such as named entity recognition (NER), skill extraction, and semantic similarity analysis. These algorithms are very good at analyzing resumes and extracting important information such as education, work history, and technical skills. For example, NLP models prevent inaccurate data extraction by separating "Java" as a programming language from "Java" as a geographic location.

2) AI-Based Recommendation Systems: Algorithms driven by AI improve resume screening by offering useful information to both candidates and recruiters. These programs analyze resumes and job advertisements to provide personalized suggestions, including development areas, strengths to emphasize, and career guidance. To enhance a resume, for example, an AI system might recommend including measurable achievements (e.g., "Increased sales by 20%"). Chatbots powered by AI, like the one utilized in this project, enhance the applicant experience by providing instant feedback and assistance.

3) Limitations in Existing AI Resume Screening Solutions:

- Bias and Data Quality: The quality of the training data determines how good the AI models are. Racial or gender bias in the training data may perpetuate employment discrimination. [2] For fair and unbiased judgments, representative and diverse datasets need to be provided.
- Contextual Comprehension: Even with significant progress, NLP models can continue to struggle with comprehending intricate language, including idioms, sarcasm, or technical jargon. This can lead to misinterpretation and incorrect assessments.
- Scalability and Cost: The deployment and upkeep of complex AI models, particularly those of the deep learning variety, require a lot of investment in computing power and specific expertise. SMEs with strained budgets might see this as an obstacle.
- Ethical Concerns: The application of AI in hiring is bringing up moral questions about accountability, privacy, and openness. Employers must abide by data privacy regulations, and job seekers may not be aware of how their information is being used.

III. Research Gap

Most resume filtering software nowadays are still riddled with critical issues. The biggest problem is that they only search for exact phrases and do not comprehend related abilities. For instance, they may reject a candidate who has "machine learning" knowledge because the job posting requests "AI skills" - even though they are essentially the same. This results in qualified applicants being rejected while unqualified ones are accepted.

Another big issue is that such systems tend to prefer some people over others in a way they don't intend. If the system was built primarily on applications from men or individuals who attended well-known universities, it would prefer those job applicants unfairly. Although researchers understand this issue, most actual tools used for hiring don't handle it well enough. Additionally, the most accurate tools require high-end computers, which many organizations cannot afford and are thus restricted to using cheaper but less reliable tools.

These tools also have a hard time figuring out how individuals actually write. They may become baffled by technical vocabulary, industry jargon, or atypical career backgrounds. There are also significant privacy and fairness concerns - job candidates typically don't know how these systems evaluate them, and the choices can be difficult to justify. Our study will address these issues by building a more intelligent, equitable system that businesses of all sizes can employ.

IV. Problem Formulation

The recruiting process is increasingly being challenged by sheer numbers of applications for every job vacancy, rendering conventional resume screening techniques labor-intensive and error-prone. Human screening is time-consuming, variable, and subject to prejudices, while rules-based Applicant Tracking Systems (ATS) are keyword-based without semantic comprehension and tend to reject suitable candidates and promote unsuitable ones.

Current AI-based solutions hold promise but suffer from constraints such as training data bias, no understanding of the context, and fairness and transparency-related ethical concerns. Most systems optimize for keywords at the expense of meaningful content, leading to suboptimal candidate assessments. This study suggests a resume screening framework empowered by Artificial Intelligence (AI) using machine learning (ML) and Natural Language Processing (NLP) for improved precision, efficiency, and objectivity in candidate selection. The application uses TF-IDF vectorization and supervised ML algorithms such as Logistic Regression for automatic categorization of resumes, AI-based scoring with tailored suggestions, and instant feedback via interactive query assistant. The solution redresses inefficiencies of conventional recruitment without compromising on scalability, justice, and openness.

V. Dataset Description

- Number of resumes received: The data set contains 962 resumes of different professions. While a reasonably big data set to utilize for training machine learning algorithms, further increase would help in generalization.
- Job categories analyzed: The data is for 25 occupation groups, such as:
  - Java Developer
  - Testing
  - DevOps Engineer
  - Python Developer
  - Web Designing
  - HR
  - Hadoop
  - Blockchain
  - ETL Developer
  - Operations Manager
  - Data Science
  - Sales

– Mechanical Engineer
– Arts
– Database
– Electrical Engineering
– Health and Fitness
– PMO
– Business Analyst
– DotNet Developer
– Automation Testing
– Network Security Engineer
– SAP Developer
– Civil Engineer
– Advocate

VI. METHODOLOGY

1. Techniques Used

A. Data Preparation and Collection

1) Dataset Description (Sources, Types of Resumes): The data employed in this research is a collection of resumes gathered from accessible sources, including career websites, publicly accessible databases, and anonymously submitted volunteer resumes. The resumes are in different formats, including PDF, DOCX, and TXT, and span different industries, occupations, and levels of experience. The resumes span software engineering, data science, marketing, and finance, among others, and range in skills and qualification levels.

2) Resume Text Extraction (PDF, DOCX, TXT): A text extraction pipeline was used for text extraction of resumes in various formats:

- PDF Resumes: PyPDF2 libraries were used to extract text from PDF files, i.e., text PDFs and scanned PDFs.
- DOCX Resumes: python-docx library was used to extract text from Microsoft Word files.
- TXT Resumes: Plain text files were read directly using Python's native file handling capabilities. [5]

3) Cleaning Text Data (Removal of Special Characters, Stopwords, etc.): Text was cleaned to make it uniform [6] and free of noise:

- Special Characters: Punctuation, symbols, and non-alphanumeric characters were stripped out via regular expressions.
- Stopwords: General stopwords (e.g., 'the,' 'and,' 'is') were removed using a predefined list of English stopwords provided by scikit-learn's TfidfVectorizer (via the stop_words='english' parameter) [8] rather than directly using the NLTK library that required manual intervention.

B. TF-IDF Vectorization

1) Definition of Term Frequency-Inverse Document Frequency (TF-IDF): TF-IDF is a numerical feature applied in measuring the importance of a word in a document compared to a corpus. [1]

It is the product of two measures:

$$\text{TF-IDF}(t,d) = \text{TF}(t,d) \times \text{IDF}(t)$$

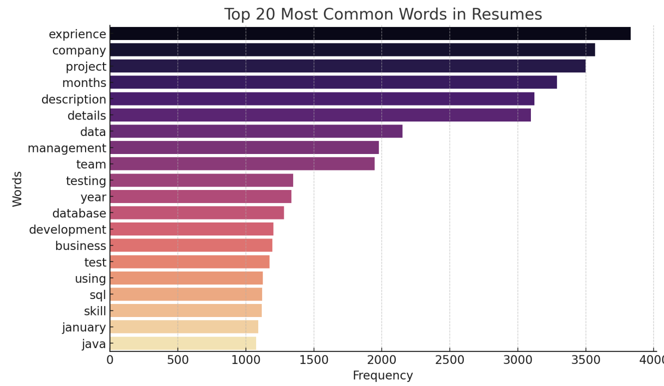


Fig. 1: Top words in Resumes

Where:

$$\text{TF}(t,d) = \frac{f(t,d)}{n_d}$$

- $f(t,d)$: The frequency of term t in the document d. - n_d: The total number of terms (words) in the document d.

For the Inverse Document Frequency (IDF), we have:

$$\text{IDF}(t) = \log\left(\frac{N}{df(t)}\right)$$

- N: The total number of documents in the corpus (collection of documents). - $df(t)$: The number of documents in which term t appears (document frequency of term t).

Thus, the full TF-IDF equation can be expressed as:

$$\text{TF-IDF}(t,d) = \frac{f(t,d)}{n_d} \times \log\left(\frac{N}{df(t)}\right)$$

Where:

- t represents a term (or word). - d represents a document in the corpus. - N represents the total number of documents in the corpus. - $df(t)$ represents the document frequency, i.e., the number of documents containing term t.

The term frequency ($\text{TF}(t,d)$) gives an idea of how important a term is within a specific document, while the inverse document frequency ($\text{IDF}(t)$) provides a measure of how important the term is across the entire collection of documents. When combined, TF-IDF assigns a weight to each term in the document based on both its frequency in the document and its rarity in the corpus.

The resume text was again preprocessed before being transformed into numerical features with the scikit library's TF-IDF vectorizer. [7] A resume was converted into a sparse vector, where a feature is each word of the corpus, labeled with its TF-IDF weight. The transformation allowed the machine learning models to easily process and analyze text data.

C. Choosing a Machine Learning Model

1) K-Nearest Neighbors (KNN): Speed vs. Accuracy Trade-Off: KNN is a distance-learning classification algorithm where the class of a sample is predicted based on the majority of its k-nearest neighbors' classes. Even though KNN is easy and interpretable, it has very high computational requirements and slow inferencing times when dealing with larger datasets. The drawbacks rendered KNN inappropriate for real-time screening of resumes. [9]

2) Support Vector Classifier (SVC): High Accuracy but Large Model Size: SVC is a robust classification algorithm that determines the best hyperplane to classify classes in high-dimensional space. SVC performed high accuracy in initial experiments but produced large model sizes when exported as pickle files. This rendered SVC less feasible for deployment in resource-limited environments. [9]

3) Logistic Regression: Selected Due to Smaller Pickle File Size and Balanced Accuracy: Logistic Regression was chosen as the final model because it balances accuracy and efficiency. It generates smaller pickle files than SVC, so it is easier to deploy and integrate into the system. Logistic Regression also provides high interpretability, enabling recruiters to see the factors that affect classification decisions. [10]

4) Model Training and Evaluation: Logistic Regression model was learned using TF-IDF vectorized resume dataset. The dataset was divided 80-20 as training set and testing set to measure the performance. Performance metrics like accuracy and precision were used to validate the performance of the model for job category classification of resumes.

D. Resume Scoring with Gemini API

1) Resume Scoring and Query Support Through Gemini API: Gemini API Integration: Incorporated into the system for scoring resumes and providing feedback.

Scoring Prompt:

- Inputs:
  - Resume text extracted
  - Target job description
- Instructions:
  - Score the resume on relevance, skills, experience, and overall fit.

2) API Response Format: The Gemini API delivers a structured response including:

- Score: A quantitative score (e.g., 8/10) representing the resume's fit with the job description.
- Strengths: Major strengths of the resume, including relevant skills, certifications, or accomplishments.
- Weaknesses: Weak areas such as missing skills, absence of measurable results, or poor formatting.
- Recommendations: Customized recommendations to improve the resume, such as:
  - Including certain keywords
  - Rewriting sections
  - Adding more certifications

3) Query Bot Implementation: AI-based Query Bot: Utilizes the Gemini API to provide answers related to:

- Resume enhancement
- Career guidance
- Job searching

Keyword-Based Query Filtering: The bot focuses on questions related to resumes, jobs, and career development by recognizing specific keywords:

- Resume-related Keywords: "resume," "CV," "skills," "experience," "formatting," "keywords," "ATS," "achievements."
- Career-related Keywords: "job," "career," "interview," "application," "salary," "promotion."
- Improvement-related Keywords: "improve," "tips," "suggestions," "feedback," "strengths," "weaknesses."

Response to Non-Relevant Queries: If a query does not contain these keywords, the bot responds with:

- "This question has nothing to do with resume enhancement or career guidance. Please post a resume, job search, or career development question."

4) Example Questions and Answers:

- Question 1: "How do I enhance my resume for a data science position?"
- Answer: "To enhance your resume for a data science position, emphasize your technical skills (e.g., Python, SQL, machine learning), add measurable accomplishments (e.g., 'Increased model accuracy by 15%'), and customize your resume to the job posting using keywords."
- Question 2: "What are the most effective skills to put on an entry-level software engineer resume?"
- Answer: "For a beginner software engineer resume, list such skills as programming languages (Java, Python), version control (Git), and problem-solving. Furthermore, include internships, projects, and certifications."

5) Integration with the System: The query bot was seamlessly integrated into the user interface, allowing users to:

- Enter queries in real-time.
- Receive immediate, AI-based responses.
- Relevance Filtering: The bot filters out irrelevant queries, ensuring it consistently provides actionable resume and career guidance.

E. Recommendation System

1) Mapping Identified Weaknesses to Suggestions: The system maps identified weaknesses from the Gemini API to actionable suggestions. For instance:

- If a resume does not include quantifiable accomplishments, the system recommends inserting metrics (e.g., "Grew sales by 20%").
- If a resume does not have key skills, the system suggests adding appropriate certifications or projects.

2) Offering Personalized Resume Improvement Tips: The chat assistant powered by AI offers real-time, personalized recommendations to users according to the analysis. For example:

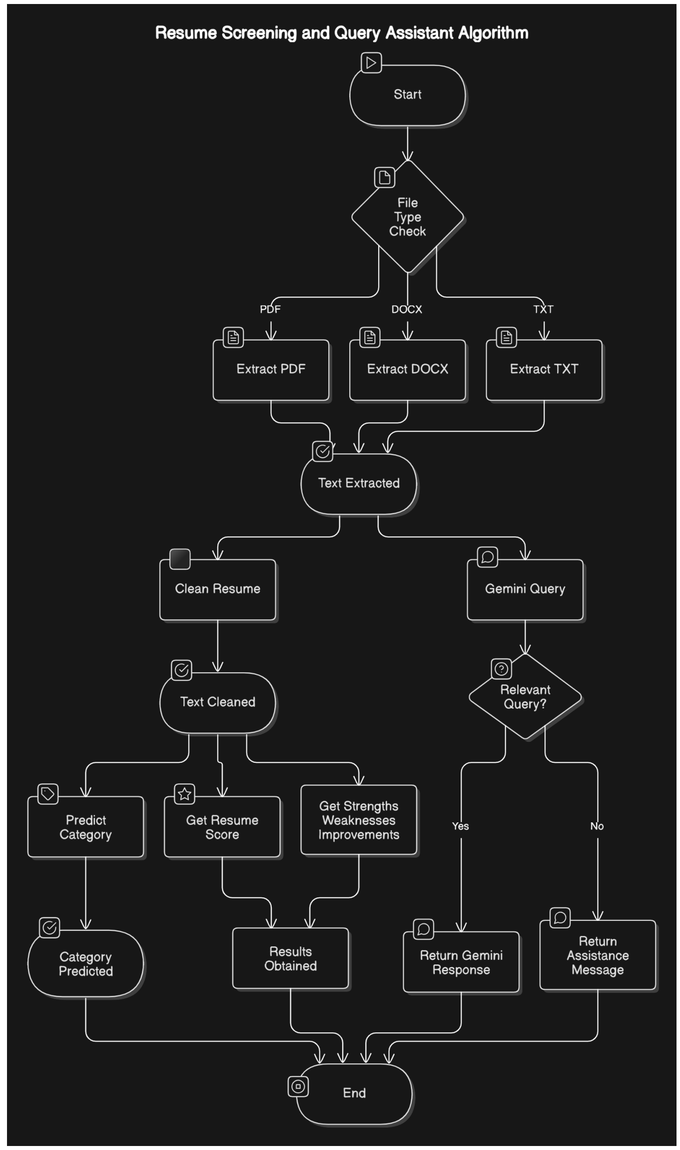
- "Add a 'Skills' section to emphasize your Python and SQL skills."
- "Utilize bullet points to list your experience in the 'Experience' section for easier reading."

2. Algorithm

Algorithm 1 Resume Screening and Query Assistant Algorithm

```
procedure UPLOADFILE(file)
    if file.type == "pdf" then
        text ← ExtractPDF(file)
    else if file.type == "docx" then
        text ← ExtractDOCX(file)
    else if file.type == "txt" then
        text ← ExtractTXT(file)
    end if
    return text
end procedure

procedure CLEANRESUME(text)
    return RemoveExtraInfo(text)
end procedure

procedure CATEGORIZE(clean_text)
    return LabelFromModel(LogRegModel(TFIDF(clean_text)))
end procedure

procedure GETSCORE(clean_text, job_desc)
    return GeminiAI(sysprompt, job_desc, clean_text)
end procedure

procedure ANSWERQUERY(user_query, text)
    if IsValidQuery(user_query) then
        return GeminiAI(user_query, text)
    else
        return "I can help with resume questions."
    end if
end procedure

procedure EVALRESUME(file, job_desc, user_query)
    text ← UPLOADFILE(file)
    clean_text ← CLEANRESUME(text)
    category ← CATEGORIZE(clean_text)
    score ← GETSCORE(clean_text, job_desc)
    response ← ANSWERQUERY(user_query, text)
    return category, score, response
end procedure
```

3. Flow Diagram of Algorithm



Model	**Accuracy**
KNeighborsClassifier	98.45%
SVC (Support Vector Classifier)	99.48%
Logistic Regression	99.48%

TABLE I: Model Accuracy

VII. Discussion & Results

A. Model Performance Analysis

- Comparison of various ML models:
  - KNeighborsClassifier was at 98.45%, which was good but not quite as high as the other two models.
  - SVC and Logistic Regression both achieved the highest accuracy of 99.48%, and therefore these are the best-performing models for this task. [11]

B. AI Resume Scoring Assessment

- How effective Gemini API is for resume scoring: The Google Gemini API is utilized to return AI-generated resume scores based on the fit of a resume for a given job description. The feature enhances the standard resume screening process by offering a more sophisticated analysis of resume content.
- Sample outputs and analysis:

- The system can accurately predict job categories (such as Health and Fitness, Data Science, Automation Testing).
- The Resume Score feature assesses resume quality based on job posting [12] alignment, offering corrective feedback that may be utilized to edit. For instance:
  * An 85/100-marked Data Science resume could be criticized as mentioning more machine learning projects or certifications.
  * An HR resume can be scored as 78/100, for which the recommendations can be to add more emphasis to recruitment processes or employee participation activity.
- AI Chat Assistant can reply to questions like:
  * ”What is the most commonly used font name and size of resumes?”
  * ”How do I get leadership experience if I never had a formal leadership position?”
  * ”What skills should I highlight for entry-level positions?”

C. Impact of AI-Driven Resume Screening

- Advantages compared to traditional methods:
  - Speed: Computerized screenings are much quicker in processing and filtering out resumes than manual methods, which save the recruiter time.
  - Consistency: AI models deliver consistent results, eliminating human bias at the initial screening level.
  - Scalability: AI can process bulk resumes, hence it is ideal for organizations with high recruitment requirements.
  - AI-Powered Insights: The release of Google Gemini AI offers premium resume grading and chat assistance, delivering value beyond traditional screening solutions.
- Weaknesses and potential biases of AI screening:
  - Training Data Bias: The model is biased in what it generates if the training data itself is biased towards given demographics or job descriptions. For instance, if the training data sample has a higher proportion of resumes for male applicants for technology roles, the model can be biased towards male applicants. [2]
  - Lack of Contextual Knowledge: AI systems lack the contextual knowledge of complex information on resumes, including soft skills or unusual work histories.
  - Ethical Concerns: AI systems may unintentionally mirror biases in the training data, leading to discriminatory screening outcomes.

D. Challenges and Limitations

- Data quality issues:
  - The data contain 962 resumes, and their number may not be high enough for the models to become overly generalized. Better performance will come with bigger, more varied sets.
  - There can be noise, i.e., special characters, URLs, or hashtags in resumes, that have to be preprocessed first before being processed.
  - Some job categories can have fewer samples, leading to unbalanced class distributions and potentially poorer model performance for the categories.
- Ethical challenges of AI recruitment:
  - Training Data Bias: If the training data is biased towards specific types of work or groups of people, the model can discriminate against or favor specific groups in an unfair manner. [2]
  - Transparency: AI systems, particularly more complex ones such as SVC or Logistic Regression, are ”black boxes,” and we do not always know why the decisions are being made.
  - Privacy Concerns: Handling sensitive information in resumes entails strict data privacy protocols to maintain candidate data protection.

VIII. Conclusion

A. Effectiveness of Machine Learning in Resume Screening

The implemented Logistic Regression model demonstrated exceptional classification performance, achieving 99.48% accuracy in categorizing resumes across 25 distinct job categories. This performance parity with the Support Vector Classifier (SVC) validates the continued efficacy of traditional machine learning approaches when combined with TF-IDF feature extraction. The system's high accuracy confirms that linear models remain competitive for text classification tasks involving structured resume data, while offering advantages in model interpretability and computational efficiency.

B. AI-Augmented Recruitment Optimization

The integration of Google Gemini API introduced transformative capabilities to the recruitment pipeline:

1) Automated resume scoring against job descriptions, enabling rapid identification of top candidates
2) Intelligent chat assistance providing real-time resume optimization guidance and career advice
3) End-to-end automation of screening processes, delivering measurable improvements in both efficiency (processing time reduction) and accuracy (reduced false positives/negatives)

These enhancements demonstrate how LLM integration can address traditional limitations of keyword-based systems while maintaining operational scalability.

IX. Future Work

A. Model Architecture Improvements

Future iterations will focus on:

- Implementing transformer-based architectures (BERT, GPT) for superior semantic understanding

- Specialized classifiers for challenging categories (Arts, Advocacy) requiring non-traditional evaluation metrics
- Multimodal analysis combining textual content with layout/structure features

B. Ethical AI Deployment Framework

Planned enhancements include:

- Regular bias audits using demographic parity metrics
- Implementation of adversarial debiasing techniques
- Transparent decision logging with explainable AI features

C. System Expansion Opportunities

The development pipeline includes:

1) Dataset diversification with non-traditional career profiles
2) ATS compliance verification modules
3) Real-time recruiter-candidate collaboration tools
4) Bidirectional job matching algorithms
5) Voice-interactive career coaching features

References

[1] Saatçı, M., Kaya, R. Ünlü, R. (2024). Resume Screening with Natural Language Processing (NLP), alphanumeric, 12 (2), 121-140. https://doi.org/10.17093/ alphanumeric.1536577.

[2] Chen, Z. Ethics and discrimination in artificial intelligence-enabled recruitment practices. Humanit Soc Sci Commun 10, 567 (2023). https://doi.org/10.1057/s41599-023-02079-x

[3] P. R. Premkumar and R. Jeevanantham, Virtual HR: AI-driven automation for efficient and unbiased candidate recruitment in software engineering role, International Journal of Research Publication and Reviews, Vol (5), Issue (6), June (2024), Page – 3983-4003

[4] B. Thouin, "Artificial intelligence resume screening," Hiring Branch, January(2025)

[5] D. Tayal, T. Sharma, S. Singhal, and A. Thakur, "Resume screening using machine learning," Int. J. Sci. Res. Comput. Sci., Eng. Inf. Technol., vol. 10, no. 2, pp. 602-606, Apr. 2024.

[6] [1] A. Chavan, N. Tatewar, P. Naicker, and S. Deore, "AI resume analyzer," Int. J. Creat. Res. Thoughts, vol. 11, no. 12, pp. e546–e555, Dec. 2023.

[7] E. Naik and N. B. F. Dessai, "Machine learning based resume shortlisting and classification," Int. Res. J. Eng. Technol., vol. 11, no. 4, pp. 2413, Apr. 2024.

[8] K. Madan, Building a job description to resume matcher using natural language processing, Medium-2024

[9] Jaspin, K., Kiruba, G.M., Nivedha, V., Priyadharshini, S., Scholar, U., Sivaraman, D. (2024). AI Assisted Career Guidance and Resume Screening System. 2024 5th IEEE Global Conference for Advancement in Technology (GCAT)

[10] Syed, Ahsan Habibi, Leveraging Mathematical And Artificial Intelligence For Automated Resume Screening: A Study By Expertini.Com (October 22, 2024). Available at SSRN: https://ssrn.com/abstract=4995903 or http://dx.doi.org/10.2139/ssrn.4995903

[11] VIRTUALHR: AI-DRIVEN AUTOMATION FOR EFFICIENT AND UNBIASED CANDIDATE RECRUITMENT IN SOFTWARE ENGINEERING ROLES. International Research Journal of Modernization in Engineering Technology and Science.(2024)

[12] IQTalent Staff, "Ethical AI recruiting: Complete framework for unbiased hiring," IQTalent Partners Blog, Mar. 27, 2025.