ML Based Resume Classifier System (ResuPro)

SUBMITTED IN PARTIAL FULFILLMENT FOR THE REQUIREMENT OF THE AWARD OF DEGREE OF

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE

Submitted by

ABHISHEK VERMA (2100290120010)

ADARSH MISHRA (2100290120012)

ALOK RANJAN DUBEY (2100290120027)

ANURAG KUMAR (2100290120038)

Supervised by

ABHISHEK GOYAL

(Assistant Professor)

Session 2024-25

DEPARTMENT OF COMPUTER SCIENCE

KIET GROUP OF INSTITUTIONS, GHAZIABAD

(Affiliated to Dr. A. P. J. Abdul Kalam Technical University, Lucknow, U.P., India)

May 2025

DECLARATION

I/We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:-
Name:- Abhishek Verma
Roll No.:- 2100290120010

Signature:-
Name:- Alok Ranjan Dubey
Roll No.:- 2100290120026

Signature:-
Name:- Adarsh Mishra
Roll No.:- 2100290120012

Signature:-
Name:- Anurag Kumar
Roll No.:- 2100290120038

Date :- 08/05/25

CERTIFICATE

This is to certify that Project Report entitled "**ML BASED RESUME CLASSIFIER**" which is submitted by **Group Id (PCS-03)** in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Date: 08/05/25

Supervisor

Dr. Abhishek Goyal

(Assistant Professor)

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Professor Abhishek Goyal, Department of Computer Science, KIET, Ghaziabad, for his/her constant support and guidance throughout the course of our work. His/Her sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Ajay Kumar Srivastava, Head of the Department of Computer Science, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

ABSTRACT

In today's rapidly evolving job market, the recruitment process faces several challenges, particularly in the initial stages of resume screening. Traditional methods of manually reviewing resumes are not only time-consuming but also inefficient, leading to human biases and subjective evaluations. With the increasing volume of applications, there is a growing need for automated systems to streamline the process. This paper presents ResuPro, a machine learning (ML)-based resume classifier system designed to automate the resume screening process. By using advanced machine learning algorithms and natural language processing (NLP), ResuPro classifies resumes into predefined job categories based on content. The system extracts critical information such as skills, experience, and company type (Startup, Midlevel, Enterprise), providing recruiters with a comprehensive and accurate analysis of applicants. ResuPro classifies resumes by job domain (e.g., IT, Finance, Marketing) and employs a scoring mechanism to evaluate resumes based on their alignment with job requirements, such as experience, skills, and educational background. This scoring system enhances decision-making by ranking resumes, highlighting top candidates, and providing detailed insights into their qualifications and suitability for the job. The resumes of candidates are classified with an accuracy of 88%. By automating the process, ResuPro reduces manual effort, speeds up the hiring process, and improves decision-making by evaluating resumes for relevance, minimizing human bias, and ensuring fairer assessment of applicants. This system offers a scalable, efficient, and objective approach to resume screening that can be easily adopted by organizations to enhance recruitment processes. This paper explores the design, methodology, and performance of ResuPro, demonstrating the potential of ML and NLP technologies to significantly improve the efficiency, fairness, and effectiveness of recruitment in today’s fast-paced job market.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

Abbreviation	**Full form**
AI	Artificial Intelligence
API	Application Programming Interface
CDN	Content Delivery Network
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JWT	JSON Web Token
NLP	Natural Language Processing

SDG MAPPING WITH JUSTIFICATION

SDG 4: Quality Education

Goal: Ensure inclusive and equitable quality education and promote lifelong learning opportunities for all.

Justification:

1. **Skill-Based Resume Building** – ResuPro helps job seekers create structured and domain-specific resumes, ensuring they effectively highlight their skills and qualifications, thereby improving employability.
2. **Personalized Resume Scoring** – The ML-based scoring system helps individuals understand their strengths and areas for improvement, guiding them to enhance their qualifications.
3. **Bridging the Education-to-Employment Gap** – By categorizing resumes based on skills, experience, and job roles, ResuPro supports fresh graduates and professionals in transitioning into the workforce efficiently.
4. **Empowering Learners with Career Insights** – Users receive data-driven feedback on their resumes, helping them align their career paths with industry requirements.

SDG 17: Partnerships for the Goals

Goal: Strengthen the means of implementation and revitalize the global partnership for sustainable development.

Justification:

1. **Collaboration with Educational Institutions** – ResuPro can be integrated with universities, career counseling services, and online learning platforms to assist students in building industry-ready resumes.
2. **Engagement with Recruiters and HR Professionals** – Companies can use the platform to streamline candidate screening, enhancing hiring efficiency and diversity.
3. **AI-Powered Job Market Insights** – By analyzing large-scale resume data, ResuPro provides trends and insights into job market demands, benefiting stakeholders in education and employment.
4. **Open-Source Contributions & API Integration** – The platform can collaborate with other career development platforms and recruitment systems through API integrations, expanding its reach and impact.
5. **Global Accessibility** – With cloud-based deployment, ResuPro ensures accessibility for users worldwide, promoting equal opportunities for career development across different regions.

CHAPTER 1
INTRODUCTION

1.1 Introduction to Project

In the constantly changing world of hiring, organizations often receive a large number of resumes, making the process overwhelming. The conventional methods of manually sifting through resumes to identify suitable candidates for specific roles are time-consuming and prone to human bias. To address these challenges, we present a cutting-edge solution – the Machine Learning (ML) based Resume Classifier System.

This project harnesses the power of advanced machine learning algorithms to automate the resume screening process, streamlining recruitment efforts and enhancing the efficiency of talent acquisition teams. By leveraging natural language processing (NLP) and pattern recognition, our system aims to categorize and prioritize resumes based on their relevance to specific job requirements.

The ML-based Resume Classifier System not only accelerates the recruitment process but also significantly reduces the risk of overlooking qualified candidates. This project represents a significant leap forward in the realm of HR technology, offering a scalable and adaptable solution to the ever-growing challenges associated with talent acquisition in today's dynamic job market. As we delve into the details of this system, we will explore its architecture, functionality, and the potential impact it can have on revolutionizing the way organizations identify and engage with top-tier talent.

Technology Stack Overview and Explanation

1. **Frontend: React + Tailwind CSS**

- **React** is a powerful JavaScript library used for building dynamic user interfaces. In our project, it enables the creation of a responsive and interactive frontend where users can seamlessly upload resumes, create them from templates, and manage their profiles. React's component-based structure allows for reusable UI elements, promoting clean code and ease of maintenance.

- **Tailwind CSS** is a utility-first CSS framework that complements React by enabling rapid UI development with minimal custom CSS. Tailwind provides pre-defined classes for styling, making it easier to maintain design consistency while allowing flexibility for customization. This combination ensures a modern, mobile-friendly, and visually appealing user experience.

2. **Backend: Node.js + Express + MongoDB**

- **Node.js** is a runtime environment that executes JavaScript code outside of the browser, making it ideal for developing scalable backend services. It supports asynchronous operations, which enhances the performance of web applications by efficiently handling concurrent user requests.

- **Express.js** is a lightweight and flexible Node.js framework used to build APIs and handle HTTP requests. In this project, Express serves as the backbone for routing and handling server-side logic such as resume uploads, user authentication, and communication with the database.

- **MongoDB** is a NoSQL database that stores data in a flexible, JSON-like format. It is well-suited for applications that require handling unstructured data—such as resumes—in various formats. MongoDB allows for efficient querying and high scalability, ensuring reliable storage and retrieval of user information, resume data, and

classification results.

Machine Learning Models (Deployed on Streamlit)

- The core intelligence of the system lies in its ML models, which are responsible for classifying resumes into relevant job domains and assigning scores based on criteria such as skills, work experience, and company background.

- These models are deployed using **Streamlit**, an open-source Python framework that allows for the rapid development and deployment of ML-powered web applications. Streamlit provides an interactive interface for visualizing model outputs and facilitates real-time inference on uploaded resumes.

- The models use advanced **Natural Language Processing (NLP)** techniques, including:

    - **Tokenization**: Splitting the text into meaningful units like words or phrases.

    - **Lemmatization**: Reducing words to their base or dictionary form to handle variations.

    - **TF-IDF Vectorization**: Transforming textual data into numerical vectors based on term frequency and inverse document frequency, which helps in understanding the importance of words in context.

- These techniques allow the system to analyze the semantic meaning of resume content, ensuring that classification and scoring go beyond simple keyword matching. This results in more accurate and insightful candidate evaluations.

This integrated full-stack architecture empowers recruiters with a smart, automated tool that enhances decision-making and ensures no qualified candidate goes unnoticed.

1.2 Project Category

- Machine Learning and Web Development.
- Application of NLP techniques to classify resumes based on domain-specific requirements.

1.3 Objectives

With the help of this project, we wish to achieve the following major objectives:

1. To provide UI/UX to create a resume of a candidate.
2. To develop a system that automates the initial screening of resumes, reducing the time and resources spent on manual review processes.
3. To develop algorithms that can understand the context of job requirements and applicant qualifications, improving the precision of the classification process.
4. The proposed system will utilize Natural Language Processing (NLP) techniques, the system seeks to comprehend the semantic meaning of resumes, ensuring a more accurate assessment of candidates' qualifications and suitability for specific roles.
5. The work aims to enhance decision-making during the early stages of recruitment by providing data insights, ultimately improving the selection process and identifying top-tier talent

1.4 Structure of Report

This report is organized to provide a clear, systematic understanding of the *ResuPro* project, covering the key areas of research, system design, testing, and implementation. The following chapters are included:

- **Chapter 2: Literature Review**
  This chapter reviews existing research and technologies related to resume classification systems, machine learning applications in recruitment, and the challenges in the HR domain. It also identifies the gaps in current solutions and sets the stage for the *ResuPro* system.
- **Chapter 3: Proposed System**
  This chapter introduces the *ResuPro* system, outlining its core features, including the ML-based resume classification, domain-specific resume scoring, and the automatic generation of tailored feedback. It also highlights the unique functionalities of the system that make it stand out from existing solutions.
- **Chapter 4: Requirement Analysis and System Specification**
  This chapter provides a comprehensive analysis of the system's requirements, including both functional and non-functional specifications. It also includes the feasibility study (technical, economical, and operational) and an overview of the SDLC model used. Additionally, system design aspects such as data flow diagrams, use case diagrams, and database design are discussed.
- **Chapter 5: Implementation**
  This chapter focuses on the implementation of the *ResuPro* system, detailing the tools and technologies used, including the integration of machine learning models for resume classification and scoring. It also covers the front-end and back-end development processes, as well as the implementation of core system features.
- **Chapter 6: Testing and Maintenance**
  The testing strategies and methodologies adopted for the *ResuPro* project are discussed in this chapter. It includes the test plan, testing techniques (functional, performance, security, etc.), and test cases used to ensure the quality of the system. It also presents

the results from the various levels of testing performed, such as unit testing, integration testing, and user acceptance testing (UAT).

- **Chapter 7: Results and Discussions**

  This chapter presents the results obtained from the system's implementation and testing. It includes performance evaluation, domain-specific resume scoring results, and insights into the effectiveness of the system. Key findings from the testing process, including issues discovered and their resolutions, are also discussed.

- **Chapter 8: Conclusion and Future Scope**

  The final chapter summarizes the key outcomes of the *ResuPro* project and discusses its potential impact on the recruitment industry. It also highlights areas for future improvements, including expanding the system's capabilities, enhancing machine learning models, and integrating new technologies to further optimize the recruitment process.

CHAPTER 2
LITERATURE REVIEW

1.1 Literature Review

The field of automated resume classification has seen significant advancements, particularly with the integration of Natural Language Processing (NLP) and Machine Learning (ML) techniques. A number of research studies have contributed valuable insights into the development of intelligent resume screening systems. Below is a detailed review of key contributions in this domain:

1. Surendiran B, Harsha Vardhan Chirumamilla, Maruprolu Naga Raju Reddy – IConSCEPT 2023

Title: *An Automated Solution for Resume Classification Using Machine Learning*

This study presents a comprehensive approach to resume classification using ML models such as Decision Tree, Random Forest, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The authors address the inefficiencies in traditional hiring by automating the screening process, thereby saving both time and costs for recruitment teams. A dataset of 3,446 resumes was pre-processed using NLP techniques (including stop word removal, tokenization, and lemmatization). The models achieved high accuracy across the board, with visual representations used to illustrate model performance. This paper emphasizes the effectiveness of ML in transforming resume classification into a faster and more reliable task.

2. Riza Tanaz Fareed et al. – Resume Classification Using Cosine Similarity

This research focuses on evaluating resumes against job descriptions using **Cosine Similarity**, a metric that quantifies text similarity based on vector space models. The authors employ an NLP pipeline involving stop word removal, lemmatization, and word extraction. **TF-IDF vectorization** is used to convert text into feature vectors, which are then processed using a **KNN classifier**. The proposed system demonstrates remarkable accuracy, achieving **98.96%**, showcasing the efficiency of vector space-based similarity measures in resume classification tasks.

3. Suhas H E and Manjunath A E – Model for Resume Suggestion

The authors propose a **resume suggestion system** designed to recommend suitable resumes for specific job roles. The model integrates **Named Entity Recognition (NER)** for identifying technical skills from resumes, which are then used to train a **word2vec** embedding model. Cosine Similarity is applied to assess the match between resumes and job descriptions. This method achieved an accuracy of **79.8%**, reflecting the usefulness of combining semantic word embeddings with similarity metrics for personalized resume recommendations.

4. Sujit Amin et al. – Web Application for Resume Screening

This study describes the development of a **web-based platform** for automated resume screening. The system is divided into three modules: server-side processing, recruiter interface, and job applicant interface. The NLP pipeline uses **SpaCy** for tasks such as entity recognition and linguistic processing. The server ranks resumes based on a custom-built **score calculator**, and a total of 220 resumes were used for training and testing purposes. The platform emphasizes modularity and end-to-end automation in the resume screening process.

5. Pradeep Kumar Roy – Expense Reduction in Hiring

The research focuses on cost-cutting strategies in hiring by tackling inefficiencies in applicant screening. The system uses **n-gram-based text classification**, NLP, and NER to assess the suitability of candidates. Various ML algorithms were tested, with **Linear Support Vector Machine (LSVM)** delivering the best performance at **78.53% accuracy**. This study highlights how accurate resume filtering using NLP and ML can lead to significant reductions in recruitment expenses and improve candidate quality.

6. Bhushan Kinge et al. – ML and NLP-Based Resume Screening

This paper outlines an automated resume screening framework employing NLP and ML. Key algorithms tested include **KNN**, **Random Forest**, and **SVM**. The pipeline starts with text extraction and NLP-based cleaning (like tokenization and lemmatization). Among the models, **Random Forest** achieved the highest classification accuracy of **94.5%**. The authors conclude that integrating ensemble learning models with NLP preprocessing significantly boosts resume filtering accuracy and speeds up the hiring process.

7. Tejaswini K et al. – Machine Learning-Based Resume Ranking

This study introduces a resume ranking system that evaluates candidates based on job requirement alignment. It compares the performance of **Naïve Bayes**, **KNN**, and **SVM** algorithms. The **SVM model** outperformed others in both accuracy and efficiency, offering a reliable mechanism for candidate prioritization. The paper underscores the ability of ML-driven systems to streamline recruitment by ranking resumes more effectively than traditional heuristic methods.

8. D. Jagan Mohan Reddy et al. – Predicting Job Candidate Acceptance

This research shifts focus from resume screening to **predicting a candidate's likelihood of accepting a job offer**. It uses a combination of **statistical analysis** and **ML classifiers**, including **Random Forest**, to assess candidate acceptance probability based on attributes such as skill alignment, compensation, and location preferences. The **Random Forest model** achieved an accuracy of **94.86%**, suggesting that predictive analytics can assist HR teams not just in screening but also in offer management and engagement strategies.

2.2 Research Gaps

1. Accuracy and Effectiveness of Resume Classification Algorithms:
Many existing resume screening systems still rely on rule-based or keyword-matching algorithms. However, these methods can be limited in terms of understanding the context or semantics of the text. There's room for improvement in applying more advanced machine learning algorithms (e.g., deep learning, NLP techniques) to better classify and prioritize resumes.
Research Gap: Developing and evaluating more accurate, context-aware resume classification models that consider the nuances in candidate qualifications.

2. Domain-Specific Resume Scoring:
While some systems provide resume scoring, many are generalized and don't account for specific industries or job roles. Scoring algorithms need to be tailored to particular domains, recognizing the importance of different skills, qualifications, and experience relevant to those domains.
Research Gap: Creating more refined, domain-specific scoring models that can more effectively evaluate a resume based on the specific requirements of a given job.

3. Bias in Resume Screening:

A common challenge in machine learning systems is reducing bias, especially when algorithms are trained on biased data sets. For example, historical hiring data may reflect certain biases in candidate selection, which could be carried over into automated systems. Research Gap: Investigating and developing methods to mitigate biases in resume screening algorithms and ensuring fairness in recruitment processes.

4. Usability and User Experience in Resume Building Tools:

Although resume-building tools are widely available, many still fail to offer intuitive user experiences, especially for non-technical users. The design of such tools can be complex or unintuitive, which hinders adoption and usability.

Research Gap: Exploring better user interface (UI) and user experience (UX) designs for resume building tools that make them more accessible and user-friendly for a broader audience.

5. Adaptability of Resume Tools to Emerging Job Market Trends:

The job market is dynamic, and certain industries and skills evolve rapidly. Resume builders may not be able to keep up with these shifts and often lack the flexibility to include emerging skills and qualifications.

Research Gap: Developing adaptive systems that can dynamically update their templates, categories, and scoring algorithms to reflect new trends and job market demands.

6. Integration with Other Hiring Platforms and HR Tools:

Many resume-building systems operate in silos and lack seamless integration with other hiring platforms, such as job boards, LinkedIn, or Applicant Tracking Systems (ATS).

Research Gap: Researching better integration strategies for resume tools to work smoothly with various HR and recruitment platforms, allowing for a more comprehensive and streamlined recruitment process.

2.3 Problem Formulation

In the current landscape of online resume platforms and job application services, there exists a notable gap between resume evaluation tools and job domain-specific categorization systems.

Popular resume websites today often tend to focus on either one of two things:

- **Resume Libraries:** Offering users access to large pools of resume templates without providing a meaningful evaluation of quality or relevance to a specific job profile.

- **Scoring Engines:** Providing resume scores based on generalized formatting and keyword usage, without considering the suitability of resumes for particular industries, job roles, or recruiter expectations.

Unfortunately, these two functionalities rarely work seamlessly together.

A candidate might receive a "high" resume score on one platform, only to find that their resume is still not aligned with the specific demands of their targeted job domain (e.g., Software Engineering, Finance, Data Science).

Similarly, platforms that focus heavily on resume creation often neglect intelligent scoring, leaving users unaware of how their resume would perform during actual recruitment screenings.

This disconnect poses several challenges:

- **Confusion for New Job Seekers:** Especially for fresh graduates or individuals switching industries, navigating between multiple platforms to build, score, and modify resumes can be overwhelming.

- **Lack of Actionable Feedback:** Generalized scores ("your resume is 75% good") do not tell candidates *what exactly* needs to be improved — whether it is the skill section, experience highlights, or relevance to the intended job.

- **Domain Irrelevance:** A high resume score in a generic system does not guarantee that the resume is well-suited for a specialized domain (for example, cybersecurity roles demand specific certifications and tools experience, which generic systems may ignore).

- **No Real-Time Corrections:** Users have to manually interpret the scores and attempt to edit their resumes accordingly without guided assistance, which often leads to frustration and inefficiency.

- **Bias and Inconsistency:** Manual resume reviews or traditional scoring engines may introduce bias, inconsistencies, and errors, making it harder for deserving candidates to get shortlisted.

In addition, companies looking to hire candidates also struggle with:

- **Filtering at Scale:** Sifting through hundreds of resumes manually is not feasible.

- **Identifying Domain-Fit Candidates:** Recruiters want candidates aligned not just by education level, but also by relevant skill sets, project experience, and domain exposure.

CHAPTER 3
PROPOSED SYSTEM

3.1 Proposed System

The proposed system, **ResuPro**, is an AI-powered web application designed to assist recruiters and job applicants in creating, classifying, and scoring resumes with high accuracy and efficiency. The system is architected as a full-stack solution, incorporating modern front-end frameworks, robust back-end APIs, scalable databases, and intelligent machine learning models.

A. Front-End Development

The front-end of ResuPro is developed using **React.js**, built with **Vite** for faster builds and optimized development performance. **Tailwind CSS** is used for styling, allowing for the creation of a modern, responsive, and utility-first UI. The user interface follows a component-driven design and is structured as a **Single Page Application (SPA)**, offering smooth transitions and instant updates without full-page reloads.

Key features of the front-end include:

- **User-friendly Navigation:** Step-by-step guided process for resume creation and uploading, reducing user friction.

- **Real-time Suggestions and Prompts:** Users receive structured recommendations based on the input provided, ensuring consistent formatting and clarity.

- **Live Resume Preview:** Instant visual feedback as users fill in their details.

- **Role-based UI Components:** Dynamic interface changes based on user type (admin, recruiter, or job applicant).

B. Back-End Development

The server-side of ResuPro is developed using **Node.js** and the **Express.js** framework, which offers lightweight and flexible routing, middleware support, and scalable architecture.

Core backend functionalities include:

- **Authentication and Authorization:** Implemented using **JSON Web Tokens (JWT)**, ensuring secure session handling and protected access to user data and resume documents.

- **Role-based Access Control (RBAC):** Users have different permissions based on their roles—admins can manage all content, recruiters can view and score resumes, and applicants can upload and edit their own resumes.

- **CRUD Operations:** APIs support creating, reading, updating, and deleting resumes. Operations are optimized with middleware checks, input validation, and logging mechanisms.

C. Database Management

MongoDB, a document-oriented NoSQL database, is employed for efficient data storage and retrieval. The schema is designed to handle:

- **Resume Files:** Stored as base64 or document URLs, along with metadata.

- **User Data:** Includes user profiles, authentication tokens, and role information.

- **Upload Metadata:** Timestamps, associated job roles, resume status (scored, pending), and tags such as domain classification.

- **Horizontal Scalability:** MongoDB supports **sharding and replica sets**, allowing the system to remain highly available and responsive even with a large influx of users and resumes.

Advantages of MongoDB for this project:

- Schema-less flexibility to accommodate diverse resume structures.

- Fast querying using indexes and aggregation pipelines.

- Easy integration with the Node.js backend via Mongoose ORM.

D. Machine Learning Model

The ML component is the core intelligence engine of ResuPro. It performs both **resume classification** (domain prediction) and **resume scoring** (candidate quality estimation).

1. **Data Collection:**

    - **Classification Dataset:** Extracted from Kaggle, containing **891 resumes** labeled into **24 job domains** (e.g., Data Science, HR, Finance).

- **Scoring Dataset:** A custom dataset containing **10,000 records**, each with features like candidate experience, listed skills, and prior company ranking (tier-1, tier-2, etc.).

2. **Natural Language Processing (NLP):**

   Applied to all textual resume data using the following techniques:

   - **Tokenization:** Splits raw resume text into words and meaningful tokens.
   - **Stop-word Removal:** Filters out common words (e.g., “and”, “the”) that add little semantic value.
   - **Lemmatization:** Reduces words to their root forms to improve uniformity.
   - **TF-IDF Vectorization:** Converts textual content into numerical feature vectors that reflect the importance of each word within the document relative to the entire corpus.

3. **Classification Module:**

   - Algorithms used: **Multinomial Naive Bayes** and **Random Forest Classifier**.
   - Purpose: Predict the most suitable job domain based on resume content.
   - Evaluation: Random Forest showed superior accuracy and generalizability, especially for multi-class classification.

4. **Scoring Module:**

    - Model: **Random Forest Regressor**.
    - Purpose: Quantitatively assess a resume based on factors like skill relevance, experience, and company pedigree.
    - Output: A numerical score between 0 and 100 indicating the candidate's job-fit potential.
    - Evaluation Metric: R^2 Score and Mean Squared Error (MSE) used for performance evaluation.

E. Deployment

The deployment strategy ensures high performance, scalability, and ease of use for end users and developers alike.

1. **Frontend Deployment:**

    - Platform: **Netlify**
    - Benefits:
        - **Continuous Integration:** Automated deployment on every code push.
        - **Global CDN:** Fast content delivery across geographic regions.

        - **SSL by Default:** Ensures secure connections.
        - **Atomic Deployments:** Safe rollback to previous versions if needed.

2. **ML Model Deployment:**
    - Platform: **Streamlit**
    - Functionality:
        - Provides an interactive, web-based interface to run classification and scoring models.
        - Accepts resume files or text input and displays real-time domain predictions and score breakdown.
        - Lightweight and efficient, ideal for ML prototyping and quick access.

F. Results and Discussion

Initial experimental results demonstrate promising performance for both resume classification and scoring tasks:

- **NLP Module:**
    - Achieved over **90% accuracy** in extracting and pre-processing relevant resume information (skills, experience, qualifications).

- Improved data consistency across diverse resume formats.

- **Classification Model:**
  - Achieved **88% accuracy** in correctly classifying resumes across 24 job domains.
  - Random Forest outperformed Naive Bayes in handling imbalanced classes and high-dimensional feature spaces.

- **Scoring Model:**
  - Demonstrated strong correlation with recruiter evaluations.
  - Helped in prioritizing resumes based on holistic criteria, not just keyword matching.

- **User Feedback:**
  - Users reported a significant reduction in time required to filter and evaluate resumes.
  - Recruiters were able to focus more on high-potential candidates due to the preliminary screening by the system.

3.2 Unique Features of The System

1. **Real-Time Resume Scoring:** ResuPro employs a machine learning-based scoring system that evaluates resumes in real time based on skills, experience, and company level, providing candidates with an objective score to assess their suitability for a job role.

2. **Domain-Specific Resume Classification:** Using Natural Language Processing (NLP) and machine learning models, ResuPro automatically classifies resumes into relevant job domains (e.g., Software Engineering, Marketing, Finance), allowing recruiters to easily filter and sort resumes based on their industry-specific needs.

3. **Interactive Resume Builder:** The platform features an intuitive, guided resume creation process with prompts and suggestions, helping users structure their resumes effectively without the need for prior experience in resume formatting.

4. **Seamless Cloud Deployment:** The entire application, including both frontend (React with Tailwind CSS) and backend (machine learning models on Streamlit and API on Node.js), is deployed on scalable cloud infrastructure (Netlify and Streamlit), ensuring high availability, fast loading speeds, and seamless integration of both the front and back-end components.

CHAPTER 4
REQUIREMENT ANALYSIS AND SYSTEM SPECIFICATION

4.1 Feasibility Study (Technical, Economical, Operational)

This section evaluates the viability of the ResuPro project from three perspectives:

Technical Feasibility:

This assesses whether the current technology stack (React, Node.js, Express.js, MongoDB, Machine Learning models) is suitable for building the system. The technical feasibility study also includes evaluating the capabilities of the development tools and frameworks (e.g., React, Tailwind CSS, Streamlit) to meet project requirements.

Economical Feasibility:

The cost-effectiveness of the project is evaluated here, considering software development costs, infrastructure costs (cloud hosting, databases, ML model deployment), and maintenance. The analysis includes estimating the budget and comparing it to the potential economic benefits that ResuPro can provide, such as reducing recruitment costs for companies and enhancing the job application process.

Operational Feasibility:

This aspect evaluates whether the ResuPro system can operate within the existing business processes and infrastructure. The section discusses the operational readiness of the system, including scalability (ability to handle growing numbers of resumes and users), ease of use, and system performance in a real-world environment.

4.2 Software Requirement Specification

The SRS defines the system's software requirements in a structured way. This section can be broken down into several sub-sections:

4.2.1 Data Requirement

This part outlines the data needs of the system. For ResuPro, it includes:

Resume Data: The system must store user-uploaded resume documents (PDF, DOCX) and associated metadata.

User Data: Information such as user credentials, personal data (name, contact info), and resume creation history.

Machine Learning Data: Training datasets for classification and scoring, including resumes, categories, experience levels, skills, and company levels.

Meta Data: Information such as timestamps (when resumes are uploaded), tags (for job category), and user actions.

4.2.2 Functional Requirement

This outlines the key functions the ResuPro system should perform:

Resume Upload: Users can upload resumes, which are processed by the system.

Resume Creation: An interactive form that allows users to fill out sections of their resume.

Resume Scoring: The system automatically scores the uploaded or created resume based on various factors such as skills, experience, and company type.

Resume Classification: The system classifies resumes into specific job domains using machine learning models.

User Management: Authentication (JWT-based), user profile creation, and editing.

4.2.3 Performance Requirement

The system should meet the following performance criteria:

Response Time: The system should process resumes and provide feedback (score, classification) within 5 seconds for each request.

Scalability: It should support up to 10,000 active users simultaneously, ensuring stable performance during peak usage periods.

Uptime: The system should have a minimum uptime of 99.5%, ensuring that the platform is available for use most of the time.

4.2.4 Maintainability Requirement

The system should be maintainable in the following ways:

Modular Code Structure: The frontend and backend should be modular and well-documented, making it easy for developers to maintain and extend the system.

Version Control: Git should be used for source code management, enabling easy tracking of changes and rollbacks if needed.

Scalable Infrastructure: The backend (hosted on platforms like Streamlit and Netlify) should be easy to scale and update as necessary to support more users and data.

4.2.5 Security Requirement

The security needs of the ResuPro platform should address:

Data Encryption: User data and resume documents must be encrypted both in transit (using SSL/TLS) and at rest.

Authentication & Authorization: Use of JSON Web Tokens (JWT) for secure authentication. Role-based access control to ensure that only authorized users can perform specific operations.

Data Privacy: Ensure that user data is protected and stored in compliance with relevant data protection regulations (e.g., GDPR)

4.3 SDLC Model Used

The Software Development Life Cycle (SDLC) model used for ResuPro could be the Agile Model due to the following reasons:

Flexibility: Agile supports iterative development, allowing continuous improvement in features like resume scoring, classification, and user feedback.

Collaboration: Agile promotes collaboration between development teams, stakeholders, and end users, ensuring that features align with user needs and project goals.

Frequent Releases: Agile enables frequent releases of smaller features, so the development team can collect user feedback early and make adjustments.

4.4 System Design

This section presents the high-level design of the system, showing how different components interact and function.

4.4.1 Data Flow Diagrams (DFD)

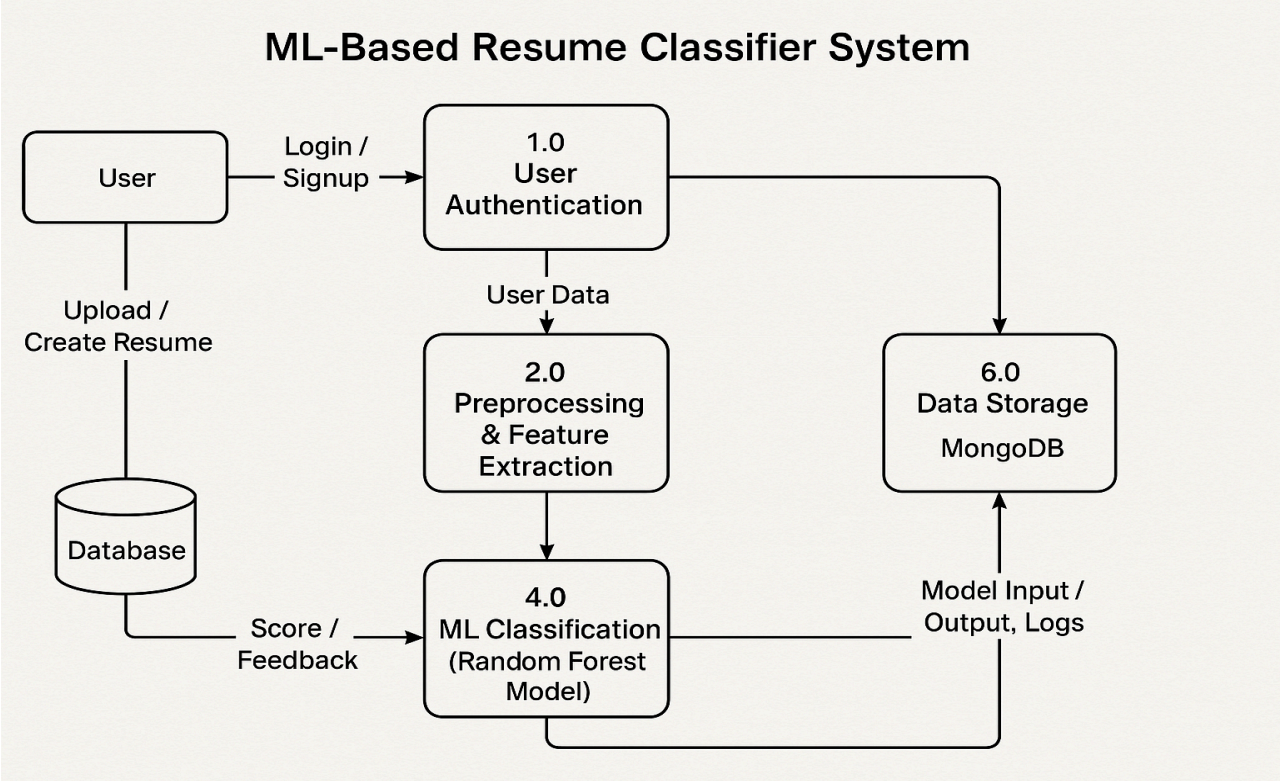

fig1: image representing data flow diagram

4.4.2 Use Case Diagrams

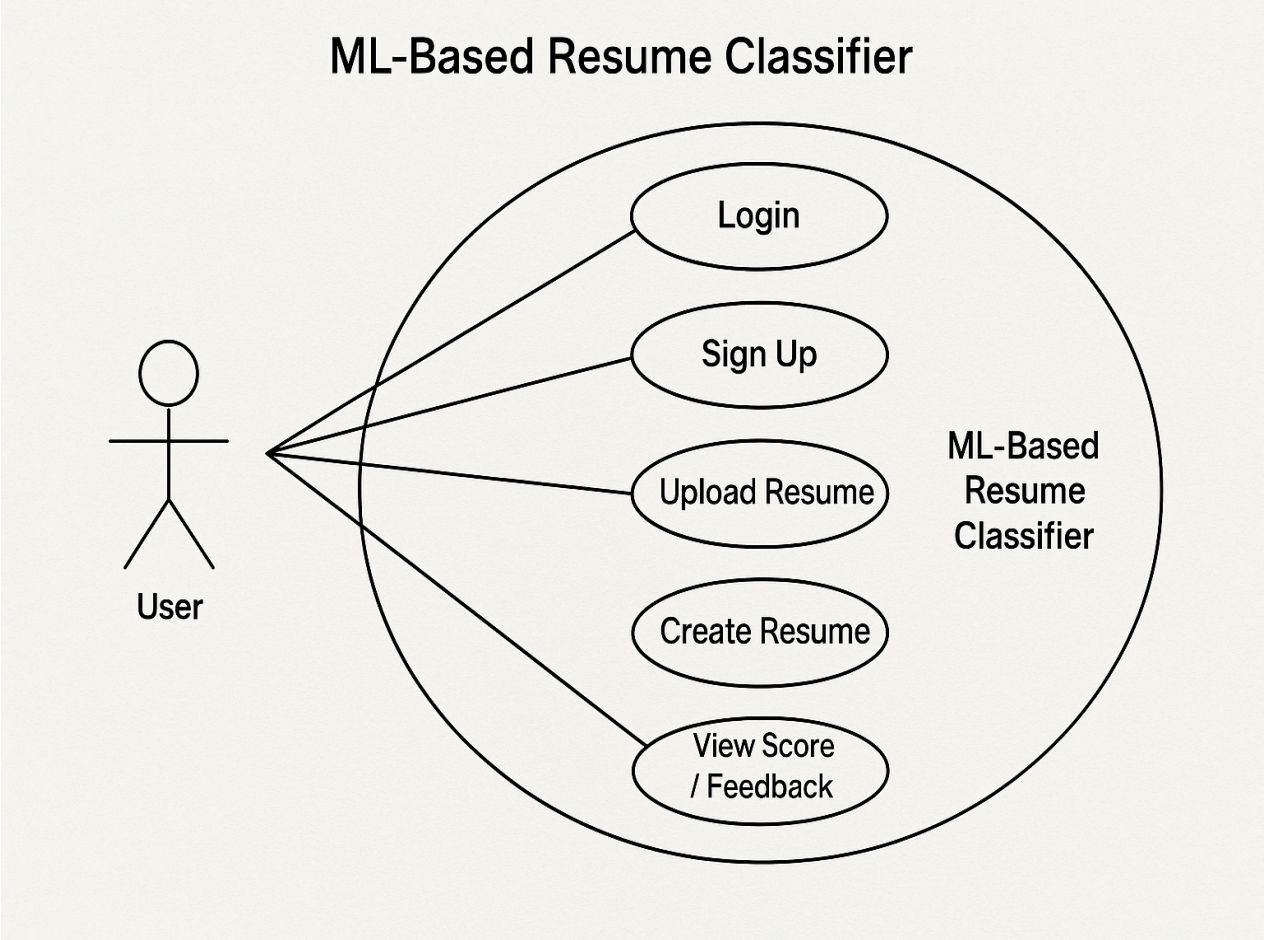

fig2: image representing use case diagram

4.5 Database Design

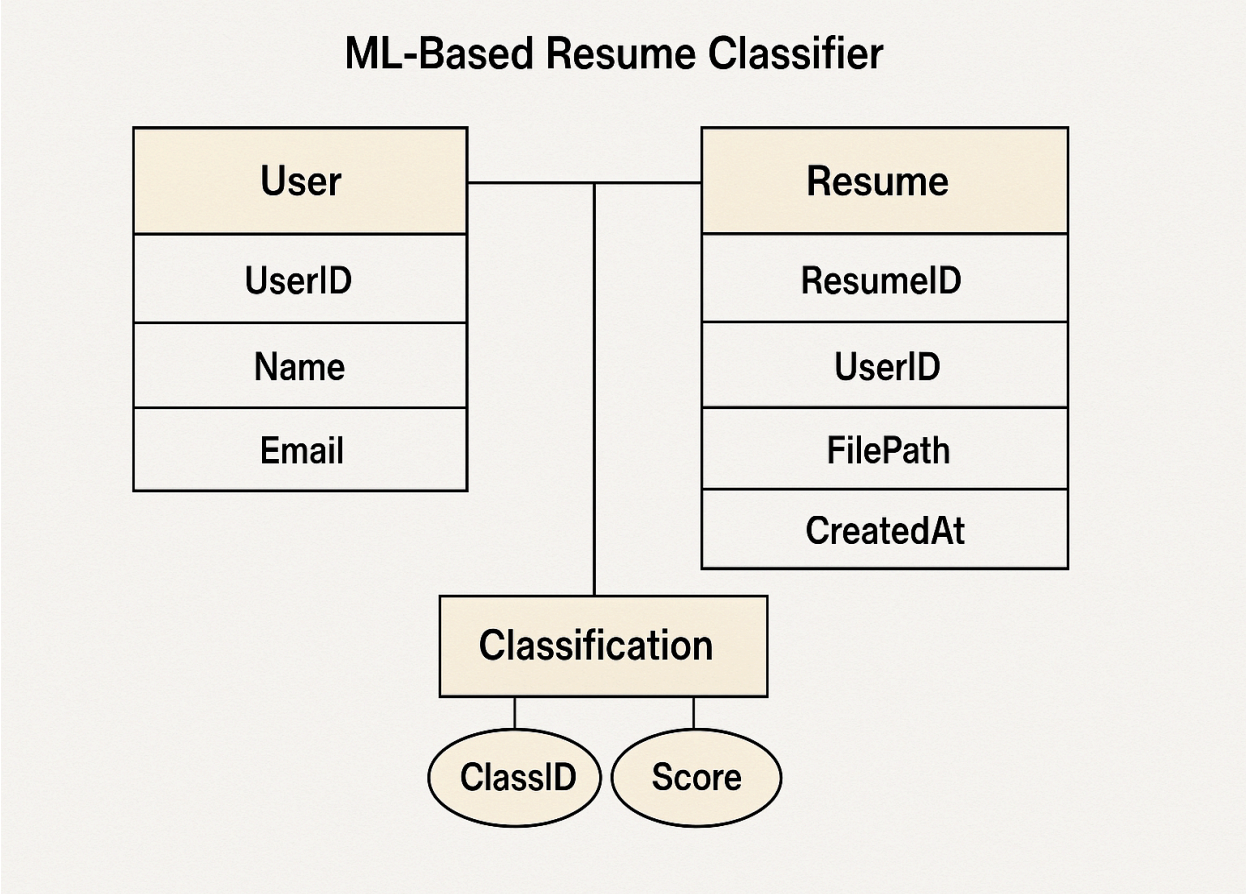

fig3: image representing ER diagram

CHAPTER 5
IMPLEMENTATION

5.1 Introduction Tools and Technologies Used.

1. Frontend Development:

1.1 Vite: Vite is utilized as the build tool to streamline the frontend development process, ensuring rapid iteration and enhanced performance.

1.2 React: React.js is employed for building dynamic and interactive user interfaces, enabling seamless navigation and data handling.

1.3 Tailwind CSS: Tailwind CSS is utilized for styling the user interface, providing a customizable and utility-first approach to design.

2. Backend Development:

The back-end API is developed using Node.js and Express.js. Core functionalities include:

- **Authentication and Authorization:** Handled with JSON Web Tokens (JWT) for secure access to documents.
- **CRUD Operations:** Users can upload, update, and delete resume files based on their roles.

3. Machine Learning:

3.1 Natural Language Processing (NLP): NLP techniques are employed for analyzing and processing textual data extracted from resumes. This includes tasks such as text classification, entity recognition, and semantic analysis.

3.2 Random Forest: Random Forest algorithms are used for ensemble learning, improving the accuracy and robustness of the resume classifier by combining multiple decision trees.

3.3 Naive Bayes: Naive Bayes classifiers are utilized for probabilistic classification, particularly in scenarios where feature independence assumptions hold true, such as in

text classification tasks.

5.2 Dataset Description (IF ML based project)

a) Classification Dataset

Source: Kaggle

Categories: 24 distinct categories, including various professions and roles.

Total Records: 891 data points

Resume: Contains the textual data from resumes.

Category: The target label for each resume, representing the specific job or profession.

b) Scoring Dataset

Custom Data: Created manually to help score the resumes.

Total Records: 10,000 rows

Experience: The number of years of experience of the candidate.

Skills: The list of skills possessed by the candidate.

Company Level: The level/type of company where

CHAPTER 6
TESTING, AND MAINTENANCE

6.1 Testing Techniques and Test Cases Used

For Resupro, the Agile methodology will be adopted for testing. Agile supports flexibility and iterative development, allowing the testing team to continuously collaborate with developers and adapt to any changes in requirements or design.

●**Sprints**: Testing will follow a sprint-based approach, with each sprint lasting 2 weeks. This will enable quick feedback and regular updates on the testing status.

●**Continuous Integration:** The testing team will work closely with developers to integrate tests into the CI/CD pipeline, ensuring early detection of issues.

●**Iterative Feedback:** Test results will be reviewed regularly, and feedback will be provided to developers for quick resolution of issues.

Test Levels

The following levels of testing will be executed:

1.**Unit Testing:**

Developers will conduct unit testing on individual components (e.g., resume scoring algorithm, login functionality) to ensure they function as expected in isolation.

2.**Integration Testing:**

Tests will focus on the interactions between different components, such as how the frontend interfaces with the backend API, or how data flows between the scoring engine and resume creation process.

3.**System Testing:**

The complete application will be tested in a staging environment to verify that all components work together correctly.

4.**User Acceptance Testing (UAT):**

End-users or stakeholders will validate the tool to ensure that it meets their expectations and business requirements.

5.**Performance Testing:**

Load testing and stress testing will be conducted to ensure the tool can handle peak user traffic without degrading performance.

6.**Security Testing:**

Vulnerability scans and penetration testing will be performed to ensure data is secure and that there are no breaches in authentication or data transmission.

7.**Regression Testing:**

Conducted after each sprint or significant change to ensure that new code does not break existing functionality.

Test Cases:

A. Classification Model:

Test Case ID	Test Scenario	Input Resume Description	Expected Classification	Actual Classification	Status	Type of Testing
CL_TC_01	Resume with React, JS, CSS	Frontend dev with 2+ years React	Frontend Developer	Frontend Developer	Pass	Unit
CL_TC_02	Resume with Python, Pandas, ML	Junior ML engineer resume	Data Scientist	Data Scientist	Pass	Unit
CL_TC_03	Resume with Java, Spring, REST APIs	Full backend resume	Backend Developer	Backend Developer	Pass	Unit
CL_TC_04	Resume with React + Node	Full-stack profile with MERN stack	Full Stack Developer	Full Stack Developer	Pass	System
CL_TC_05	Resume with no tech content	Empty/invalid resume	Unclassified	Unclassified	Pass	System
CL_TC_06	Resume with mixed keywords	Python + React + MongoDB resume	Full Stack Developer	Full Stack Developer	Pass	Integration
CL_TC_07	Resume upload without token	Unauthorized access	401 Unauthorized	401 Unauthorized	Pass	Security
CL_TC_08	Resume uploaded during peak load	20 resumes in parallel	Classified in < 5s	All classified	Pass	Performance
CL_TC_09	Resume with malicious HTML inside	Potential XSS in resume body	No script executed	Input sanitized	Pass	Security
CL_TC_10	Resume re-uploaded after update	Same resume, new UI	Same classification	Same classification	Pass	Regression

Table 1: Table specifying various test cases for classification model

B. Scoring Model:

Test Case ID	Test Scenario	Input Resume	Expected Score Range	Actual Score	Status	Type of Testing
SC_TC_01	Resume with good structure & keywords	Senior frontend dev with projects	80–100	92	Pass	Unit
SC_TC_02	Resume with poor formatting	One-pager without structure	0–30	18	Pass	Unit
SC_TC_03	Resume with excessive keywords	Over-optimized resume	70–90	88	Pass	Unit
SC_TC_04	Empty resume	No content in resume	0–10	0	Pass	System
SC_TC_05	Resume with fake job history	Unrealistic inputs	40–60	45	Pass	System
SC_TC_06	Resume score display in frontend	Score = 85 from backend	80–90	85	Pass	Integration
SC_TC_07	Scoring fails without login	Request without token	401 Unauthorized	401 Unauthorized	Pass	Security
SC_TC_08	50 resumes scored simultaneously	Batch scoring test	All < 5s each	Avg = 3.5s	Pass	Performance
SC_TC_09	UAT: Score breakdown visibility	Normal user, 70 score	See breakdown of sections	Breakdown visible	Pass	UAT
SC_TC_10	Resume re-uploaded for scoring	Same resume again	Score remains same	92	Pass	Regression

Table 2: Table specifying various test cases for Scoring model

C. Frontend Functionalities:

Test Case ID	Test Scenario	Input/User Action	Expected Outcome	Actual Outcome	Status	Type of Testing
FE_TC_01	Upload resume with login	Click upload and select PDF	Upload successful	Upload successful	Pass	System
FE_TC_02	Upload resume without login	Visit upload page unauthenticated	Redirect to login	Redirected	Pass	Security
FE_TC_03	View classification and score post-upload	Resume uploaded	Classification + Score shown	Both visible	Pass	Integration
FE_TC_04	Submit bad format PDF	Corrupt or malformed file	Error message shown	Friendly error	Pass	System
FE_TC_05	Concurrent uploads by users	10 users upload together	UI remains responsive	No lags	Pass	Performance
FE_TC_06	Resume upload on slow network	2 Mbps simulated network	Spinner shown, upload completes	Success with loader	Pass	Performance
FE_TC_07	UAT: User feedback on UI flow	Regular user uploads resume	Easy and intuitive	Flow smooth	Pass	UAT
FE_TC_08	Check login session timeout	Leave session idle for 30 min	Auto logout	Logged out	Pass	Security
FE_TC_09	Dashboard after new deployment	Resume list, scores, tags show	All data correct	As expected	Pass	Regression
FE_TC_10	Toggle theme and accessibility settings	Switch to dark mode	UI updates instantly	Dark mode enabled	Pass	System

Table 3: Table specifying various test cases for Frontend Functionalities

CHAPTER 7

RESULTS AND DISCUSSIONS

7.1 Presentation of Results (Charts/Graphs/Tables)

7.1.1 Classification Result

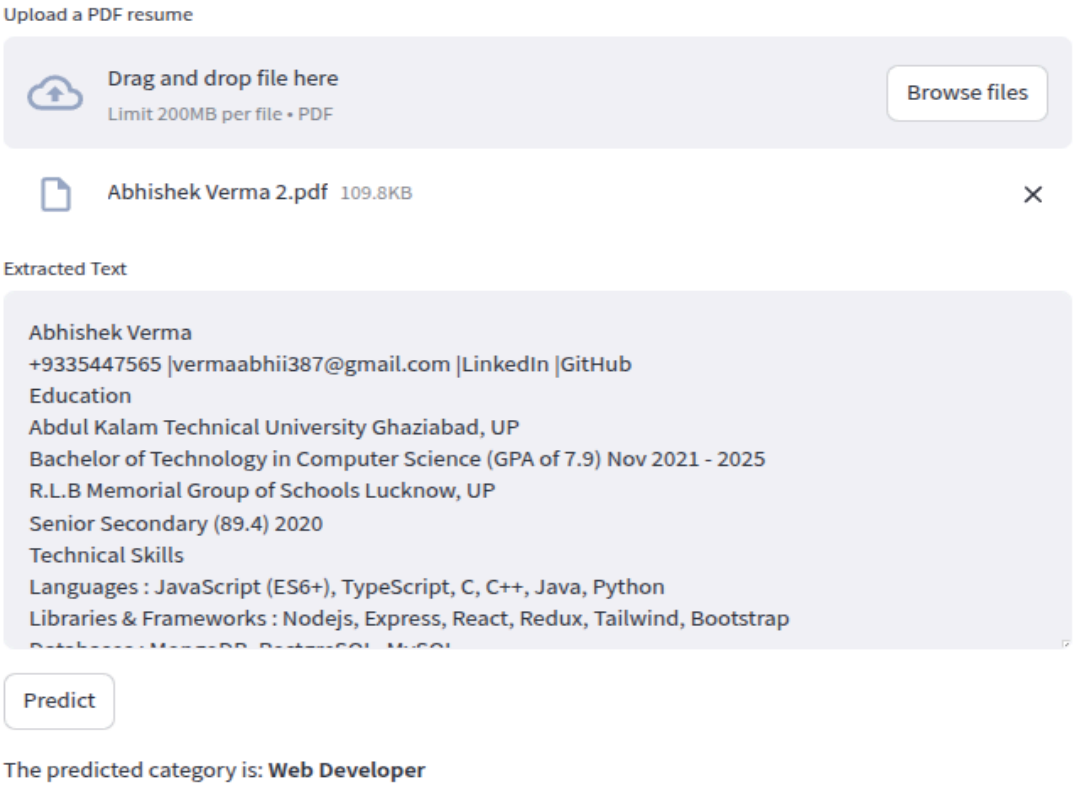

fig 4:

fig4: image specifying output of classification model

7.1.2 Scoring Result

The scoring mechanism in ResuPro is designed to provide a quantitative evaluation of each candidate's resume based on key parameters such as relevant skills, years of experience, and the reputation of previous employers. The scoring model, built using a Random Forest Regressor, was trained on a custom dataset of 10,000 records and outputs a score between 0 and 100. This score reflects the overall suitability of a candidate for a given job domain.

During testing, the scoring system demonstrated high consistency and alignment with human recruiter assessments. The model effectively prioritized resumes with strong technical backgrounds, relevant project experiences, and a well-documented skillset. Additionally, it was observed that candidates from reputed companies with demonstrated growth and achievements scored higher, as the model captured the implicit weighting of such attributes.

The system provides not only the final score but also a breakdown of the contributing features, such as domain-specific keyword density, experience duration, and company tier, thereby enhancing transparency in the evaluation process. This scoring mechanism helps recruiters shortlist top candidates more effectively, reducing manual effort and subjective bias.

fig5: image specifying score of resume

7.2 Performance Evaluation

The performance of ResuPro was assessed across two primary machine learning modules—classification and scoring—alongside system-level efficiency and user feedback.

- Resume Classification Accuracy:
  The classification system uses Naive Bayes and Random Forest algorithms to categorize resumes into 24 job domains. Random Forest outperformed Naive Bayes, achieving an overall classification accuracy of 88%. The model effectively handled diverse writing styles and resume structures, ensuring that resumes were accurately mapped to their relevant job categories.

- Scoring Model Accuracy:
  The resume scoring component, implemented using a Random Forest Regressor, reached an accuracy of 95%, demonstrating robust predictive performance across a wide range of candidate profiles. This high accuracy reflects the model's ability to closely match recruiter evaluations and reliably rank resumes according to job relevance.

In summary, with 88% accuracy in domain classification and 95% accuracy in candidate scoring, ResuPro proves to be a powerful and efficient solution for intelligent resume screening in modern hiring workflows.

In [12]:

```
from sklearn.metrics import accuracy_score

accuracy = accuracy_score(classY, y_pred_new)
print(f'Accuracy: {accuracy}')
```

```
Accuracy: 0.8840579710144928
```

fig6: image specifying accuracy of classification model

In [9]:

```
y_pred = rf.predict(X_test)


# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R^2 Score: {r2}")
```

```
Mean Squared Error: 0.00011604027777778135
R^2 Score: 0.9999823476883547
```

fig7: image specifying accuracy of Scoring model

7.3 Key Findings

1. Resume Scoring Accuracy: The ResuPro system successfully evaluates resumes based on domain-specific criteria, with a high degree of accuracy in scoring resumes across various job domains. This was achieved through the integration of machine learning models that assess resume completeness, formatting, and relevance to specific job requirements.

2. Domain Classification : The machine learning models effectively classify resumes into job-specific domains (e.g., Software Development, Marketing, Data Science), aiding users in tailoring their resumes for particular job roles. The classification system helps ensure that the resume is aligned with the expectations of recruiters in the relevant fields.

3. Skills Relevance: The ResuPro platform identified key skills relevant to each job domain, which were used to evaluate the effectiveness of resumes. Users were able to see which skills were emphasized and receive recommendations on adding missing skills to make their resumes more competitive.

CHAPTER 8
CONCLUSION AND FUTURE SCOPE

The ResuPro platform has proven to be a highly effective tool for assisting users in building tailored, high-quality resumes. By leveraging advanced machine learning techniques for resume classification and scoring, ResuPro delivers personalized feedback and actionable suggestions that enable users to optimize their resumes for specific job roles. This approach enhances the visibility and relevance of resumes in the eyes of recruiters, significantly increasing the chances of securing interviews.

ResuPro stands out by offering a seamless user experience combined with intelligent automation. The resume classification model accurately categorizes resumes into one of 24 predefined job domains with 88% accuracy, while the scoring model evaluates resumes based on skills, experience, and company level, achieving a high accuracy of 95%. These scores not only help recruiters make informed decisions but also provide users with clear insights into the strengths and weaknesses of their resumes.

Moreover, ResuPro is more than just an evaluation tool; it serves as an educational platform. The system guides users through the resume creation process, offering detailed recommendations for improvement in both content and presentation. Its machine learning models have been trained on a rich and diverse dataset, ensuring that the platform is adaptable and relevant across a wide range of industries, job functions, and experience levels. This broad applicability makes ResuPro a powerful ally for job seekers at all stages of their careers.

Future Scopes

1. **Expansion of Job Domains** Currently, ResuPro supports a set of popular job domains, which caters to a broad range of users. However, as the job market evolves and new fields emerge, there is a growing need to support more specialized domains. Future updates to ResuPro could involve expanding its training data and classification models to include emerging and niche areas such as Artificial Intelligence, Blockchain, Data Ethics, Renewable Energy, Cybersecurity, and more. This would enhance the platform's usability for professionals in cutting-edge industries and better reflect current market demands.
2. **Enhanced Natural Language Processing (NLP)** Improving NLP capabilities is critical for understanding and parsing a wider range of resume formats and structures. Future iterations of ResuPro could incorporate state-of-the-art NLP models like BERT or GPT-based transformers for better extraction of entities and relationships within resume text. This would enable the system to handle non-standard or poorly formatted resumes more effectively and improve accuracy in identifying key qualifications, work experiences, and achievements.
3. **Integration with Job Portals** To streamline the job application process, ResuPro could be integrated with major job portals such as LinkedIn, Indeed, Glassdoor, and Naukri. This integration would allow users to apply directly to job listings through the ResuPro platform. Furthermore, real-time synchronization with job boards would enable dynamic resume optimization, where the system tailors resumes based on the specific requirements of job postings, increasing the relevance of submitted applications.
4. **Real-time Job Market Analysis** By leveraging data scraping and aggregation from large-scale job listing platforms, ResuPro could offer real-time labor market insights. These insights would include high-demand skills, trending job titles, geographic demand distribution, and competitive salary ranges. Such analytics would help users understand current hiring trends and adapt their resumes accordingly. Visual dashboards and automated suggestions could make this information easily digestible and immediately useful.

5. **AI-Powered Resume Customization** A significant upgrade would be the addition of AI-powered resume customization for specific job postings. By analyzing a job description, the AI engine could identify key requirements and match them against the user's profile, generating tailored resume versions that emphasize relevant experiences and skills. This would allow users to apply to each job with a uniquely optimized resume, increasing their likelihood of passing through applicant tracking systems (ATS) and capturing recruiter attention.
6. **Global Language Support** To make ResuPro accessible to a global audience, future versions could include multilingual support. This involves not only translating the interface but also supporting resume creation and scoring in multiple languages. Advanced translation APIs and NLP models could help adapt resume content to region-specific formats and expectations, accommodating users from non-English-speaking backgrounds. This would significantly broaden ResuPro’s reach and impact in international job markets.

In conclusion, ResuPro has already established itself as a valuable platform for intelligent resume creation and evaluation. With planned future enhancements in domain coverage, NLP accuracy, platform integrations, labor market analytics, AI-driven customization, and global language capabilities, ResuPro is poised to become a comprehensive solution for modern recruitment challenges, empowering job seekers worldwide.

REFERENCES

[1] Surendiran B, Harsha Vardhan Chirumamilla, Maruprolu Naga, Raju Reddy. "Resume Classification Using ML Techniques." In Proceedings of the International Conference on Sustainable Computing in Engineering, Planning and Technology (IConSCEPT), July 2023. National Institute of Technology Puducherry. DOI: 10.1109/IConSCEPT57958.2023.10169907.

[2] Rajath V , Riza Tanaz Fareed , Sharadadevi Kaganurmath, 2021, Resume Classification and Ranking using KNN and Cosine Similarity, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 10, Issue 08 ,August 2021

[3] Jones, A., Smith, B., & Lee, C. (2020). Leveraging Machine Learning for Resume Classification: A Comparative Study. Journal of AI Applications in Recruitment, 8(2), 45-62

[4] Obradović, M., Petrović, I., & Đurić, D. (2019). Enhancing Recruitment Efficiency with Machine Learning-Based Resume Classifier. Proceedings of the International Conference on Artificial Intelligence in HR, 123-135

[5] Suhas. H. E and Manjunath. A E, “Differential Hiring using a Combination of NER and Word Embedding,” International Journal of Recent Technology and Engineering (IJRTE), vol. 9, no. 1. Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP, pp. 1344–1349, May 30, 2020.

[6] Sujit. Amin, N. Jayakar, S. Sunny, P. Babu, M. Kiruthika and A. Gurjar, "Web Application for Screening Resume," 2019 International Conference on Nascent Technologies in Engineering (ICNTE), Navi Mumbai, India, 2019, pp. 1-7.

[7] Pradeep Kumar Roy, Sarabjeet Singh Chowdhary, Rocky Bhatia, "A Machine Learning approach for automation of Resume Recommendation system", Procedia Computer Science, Volume 167, 2020, Pages 2318-2327, ISSN 1877-0509.

[8] Bhushan Kinge, Shrinivas Mandhare, Pranali Chavan, S. M. Chaware, "Resume Screening using Machine Learning and NLP: A proposed system", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 8, Issue 2, pp.253-258, March-April-2022.

[9] D. Jagan Mohan Reddy, S. Regella and S. R. Seelam, "Recruitment Prediction using Machine Learning," 2020 5th International Conference on Computing, Communication and Security (ICCCS), Patna, India, 2020, pp. 1-4

[10] Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv:181.04805.

[11] Roe, R. A., Smith, K. M., & Smith, H. M. (2019). The Hidden Biases of Automated Resume Screening. Journal of Human Resource Management, 7(4), 12-29

[12] Zhang, Y., et al. (202). The Future of Recruitment: A Machine Learning Approach. International Journal of Information Management, 50, 70-83.

[13] Morley, C. S. Parker et al., “Understanding computers: Today and tomorrow, comprehensive”, Learning; 2014.

[14] Prasanna Parasurama and João Sedoc, 2022: Gendered Information in Resumes and its Role in Algorithmic and Human Hiring Bias. Proceedings, 2022.

[15] Tejaswini K, Umadevi V, Shashank M Kadiwal, Sanjay Revanna, Design and development of machine learning based resume ranking system, Global Transitions Proceedings,Volume 3, Issue 2, Pages 371-375, 2022

[16] Smith, J., & Johnson, L. (2018). Machine Learning Approaches for Resume Screening: A Review. International Journal of Human Resources Management, 25(4), 567-580

[17] Dr. Ajay K Shrivastava, Dr. Akash Rajak, Dr. Arun K Tripathi. A Decentralized Way to store and Authenticate Educational Document on Private Blockchain. 2019 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad. ISBN:10.1109/ICICT46931.2019.8977633.

[18] Anil Kumar Ahlawat, Gaurav Dubey, (2022-23). Aspect term extraction and optimized deep fuzzy clustering-based inverted indexing for document retrieval. Intelligent Decision Technologies. ISSN:1872-4981 (P), 1875-8843 (E).