**Resume Analysis System**

**Submitted for**

[**Artificial Intelligence and Machine Learning**](https://lms.bennett.edu.in/course/view.php?id=10303)[**CSET301**](https://lms.bennett.edu.in/course/view.php?id=10303)

Submitted by:

**(E23CSEU0377) Shivam Agrawal**

**(E23CSEU0379) Divyanshu Pathak**

Submitted to

**DR. YAJNASENI DASH**

**Jan-May 2025**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| Sr.No | Content | Page No |
| 1 |  |  |
| 2 |  |  |

**ABSTRACT**

The **Resume Analysis System** is an AI-powered solution designed to automate resume classification, skill extraction, and job matching. By leveraging Natural Language Processing techniques and machine learning models, the system processes resumes (PDF and text) to extract meaningful insights and assess their fit against job descriptions. It classifies resumes into predefined job categories, extracts technical and soft skills using pattern matching, and calculates a match score based on the weighted similarity between the resume and job description. This tool streamlines the recruitment process for HR teams by automatically identifying the most suitable candidates for specific roles. The project is built using Python, Flask, scikit-learn, and PyPDF2, with a clean and accessible web interface, offering an intelligent resume screening solution.  
  
  
  
**INTRODUCTION**

Recruiters often face the challenge of filtering through hundreds or thousands of resumes, making the process time-consuming and prone to bias. The Resume Analysis System automates this process by using artificial intelligence to evaluate resumes based on content relevance and alignment with job descriptions.

The system consists of three primary components:

* **Resume Classification**: Categorizes resumes into roles using Random Forest classifiers trained on TF-IDF vector representations.
* **Skill Extraction**: Detects key technical and soft skills using pattern matching against a curated keyword database.
* **Job Match Scoring**: Computes the match percentage between a resume and provided job descriptions using a weighted scoring system.

This intelligent platform assists both employers and candidates by speeding up the recruitment pipeline and improving job alignment

**METHODOLOGY**

The system is developed using a structured pipeline consisting of the following modules:

**1. Resume Text Extraction**

* Resumes uploaded in PDF format are parsed using **PyPDF2**.
* Extracted text is cleaned using regular expressions to remove special characters and standardize formatting.

**2. Resume Classification**

* Text is vectorized using **TF-IDF vectorization**.
* **Random Forest classifier** predicts the most suitable job category.
* The classifier is trained using the models stored in the models directory.

**3. Skill Extraction**

* A comprehensive dictionary of technical and soft skills is maintained in the application.
* Pattern matching using regular expressions identifies skills present in the resume.
* Extracted skills are presented as tags in the user interface.

**4. Job Description Matching**

* The system compares resumes against predefined job descriptions.
* Match scoring uses a weighted approach:
  + 50% weight for matching skills
  + 20% weight for education match
  + 30% weight for overall text similarity using cosine similarity
* Multiple job matches are provided with detailed breakdown of match components.

**5. User Interface**

* A web interface built with **Flask** and **HTML/CSS** allows users to upload resumes and view analysis results.
* Results include categorization, recommended job, contact information, skills, and job matches with scores.

**HARDWARE & SOFTWARE REQUIREMENTS**

**Hardware Requirements**

* **Computer**: Intel i5 or better, 8GB RAM
* **Storage**: At least 500MB
* **Internet**: Required for web interface access

**Software Requirements**

* **OS**: Windows 10 / Linux / macOS
* **Programming Language**: Python 3.8+
* **Libraries**:
  + Flask – Web framework for interface
  + PyPDF2 – PDF parsing
  + scikit-learn – ML algorithms and TF-IDF
  + NumPy – Numerical computations
  + Pandas – Data manipulation
  + Matplotlib & Seaborn – Visualization (for model training)
* **IDE**: VS Code / Jupyter Notebook
* **Web Browser**: Chrome, Firefox, or Edge for accessing the interface

**EXPERIMENTAL RESULTS**

**1. Classification Performance**

* Model successfully categorizes resumes into appropriate job roles.
* Resume categories include: **Software Developer, Data Scientist, Manager**, etc.

**2. Skill Extraction Efficiency**

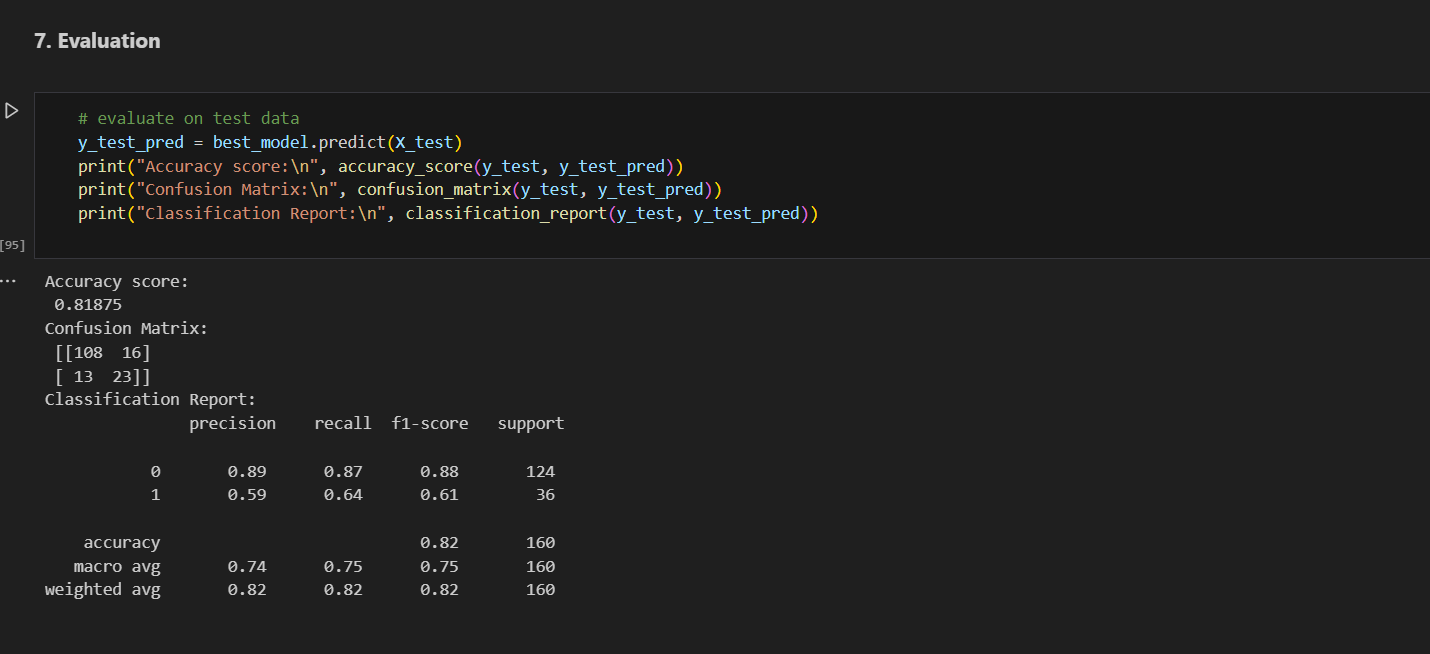
* System effectively identifies over **250+ distinct skills** from resumes.
* Skills are categorized and presented as **tags** in the UI.

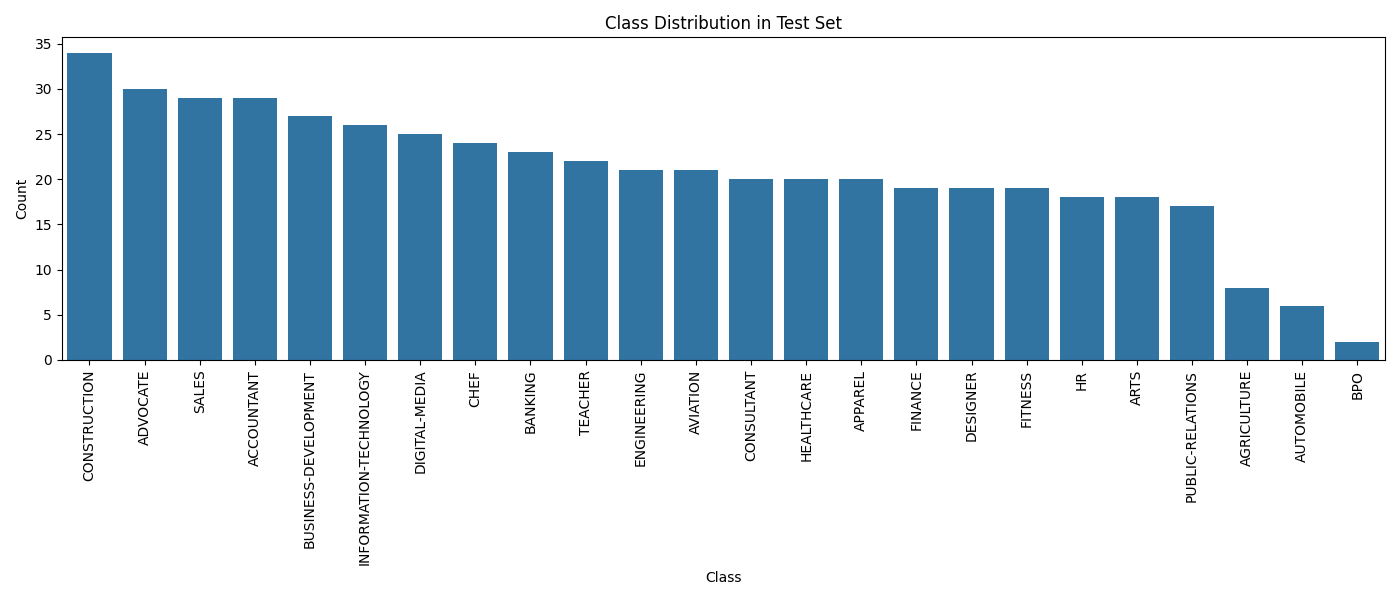
**3. Job Matching Accuracy**

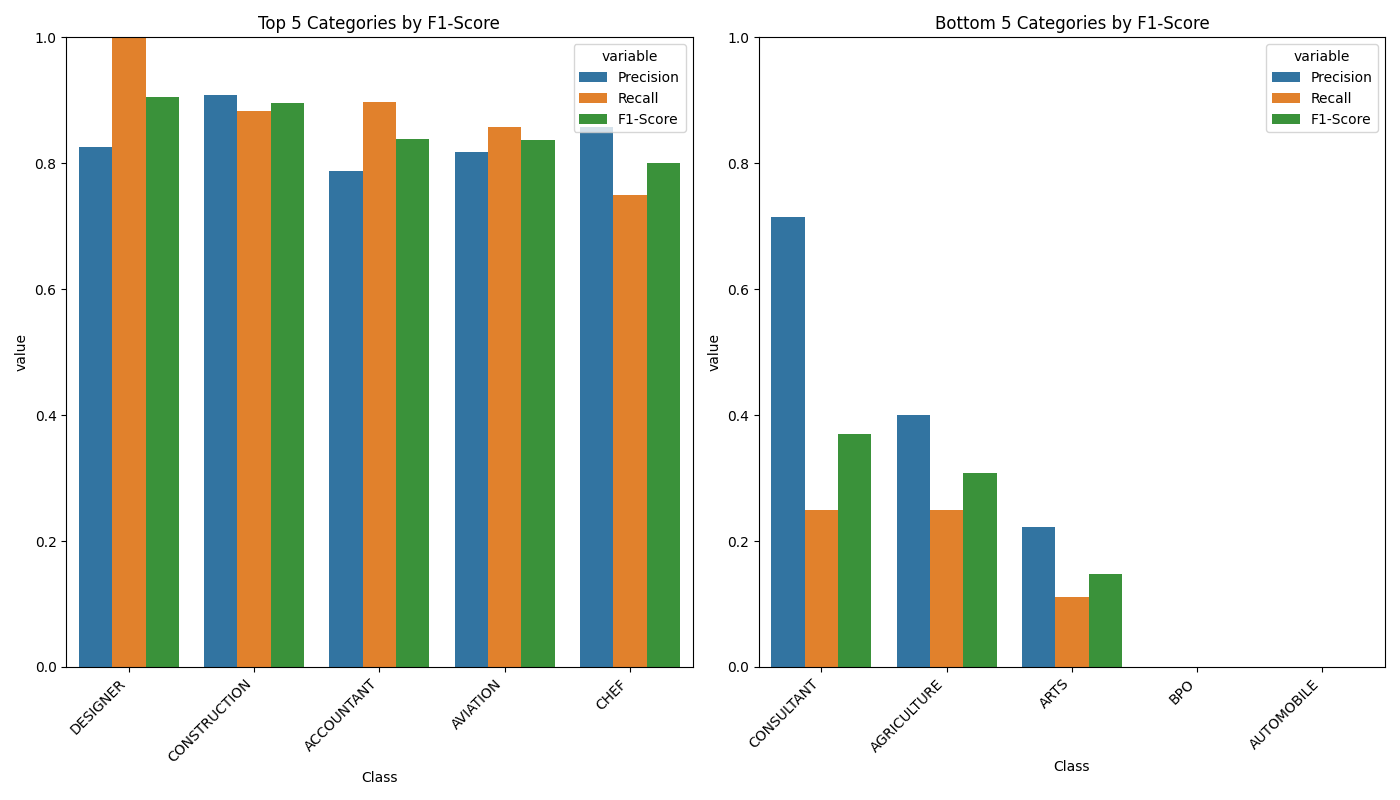
* Weighted scoring system:
  + 50% skills
  + 20% education
  + 30% content similarity
* Match scores correlate well with manual review.
* Top job matches are provided with **detailed breakdowns**.

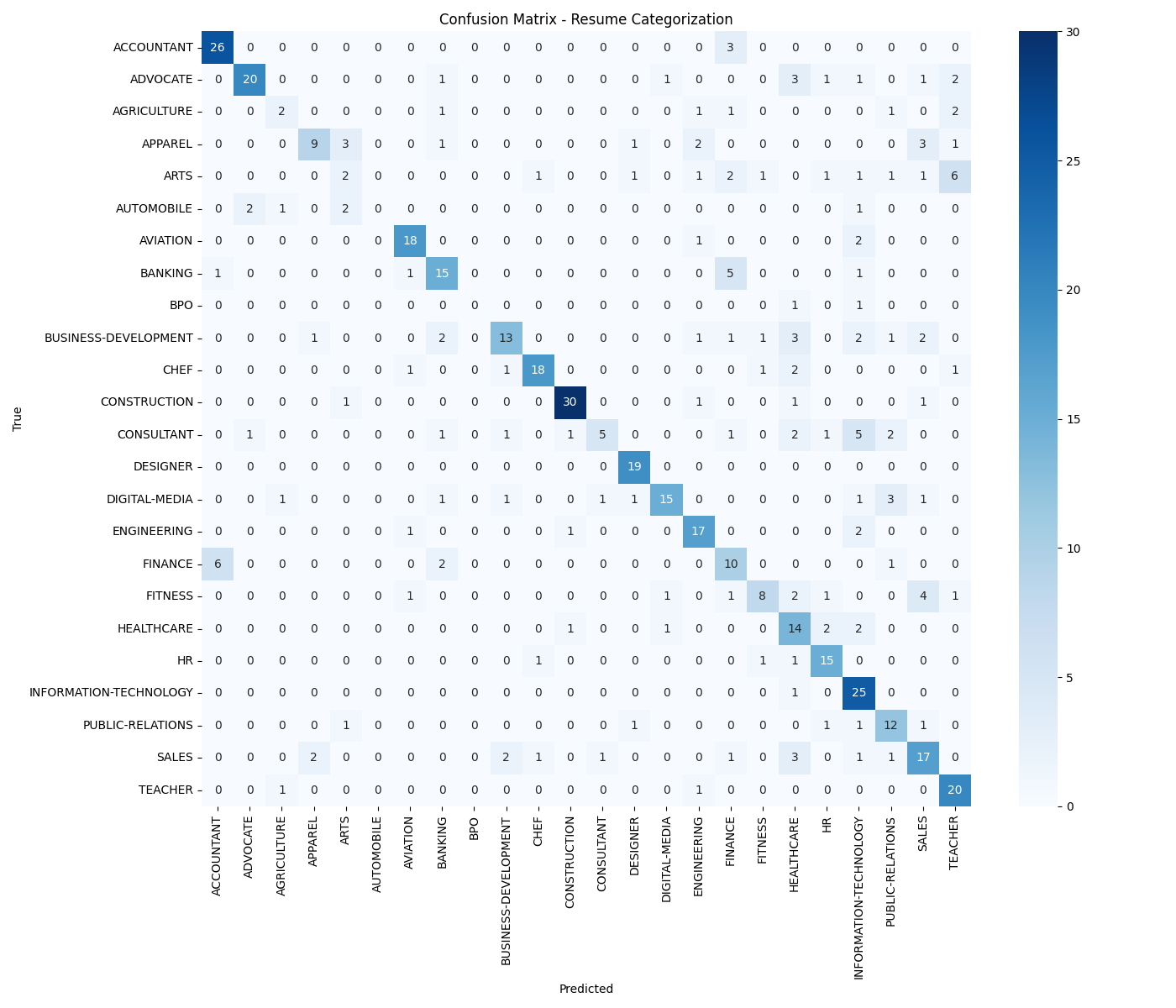
**4. System Performance**

* PDF parsing and analysis completed in **under 5 seconds on average**.
* Web interface provides **real-time feedback and results**.









**CONCLUSIONS**

The **Resume Analysis System** effectively automates key parts of the recruitment process. It:

* Provides accurate **job role classification** for submitted resumes.
* Extracts relevant **skills and education** information.
* Calculates **comprehensive match scores** against job descriptions.
* Presents results in an **intuitive, user-friendly interface**.

The solution demonstrates the practical application of NLP and machine learning techniques to solve real-world HR challenges. By automating the initial screening process, the system saves valuable time for hiring teams while providing **objective assessment** of resume-job fit.

**FUTURE SCOPE**

 **Advanced NLP Techniques**: Implement transformers (e.g., BERT) for better understanding.

 **Multilingual Support**: Process resumes in multiple languages.

 **Interview Question Generation**: Based on detected gaps or weak areas.

 **API Integration**: Connect with existing HR tools and platforms.

 **Resume Improvement Suggestions**: Automated feedback on formatting and content.

 **Custom Job Description Input**: Allow recruiters to define custom roles.

 **Resume Anonymization**: Hide personal details to reduce bias.

 **Visual Resume Analysis**: Analyze layout, color, and section balance.

 **Real-time Job Market Data**: Integrate with job APIs for relevant suggestions.

 **Mobile Application**: Lightweight app for resume scanning on-the-go.

**GITHUB LINK**

**https://github.com/ShivamAgrawal0813/resume\_analysis\_system**