Four Weeks Industrial Training Project Report

on

AI Resume Analyzer

Submitted in the partial fulfillment of the requirement for the award of a degree of

Bachelor of Technology

in

Computer Science and Engineering

Batch

(2023-2027)

Submitted To:

Dr, Riddhi kapoor

Submitted By:

Tarangini dave

12300387

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DAV UNIVERSITY JALANDHAR-PATHANKOT NATIONAL

HIGHWAY, NH44, SARMASTPUR PUNJAB-144001

ABSTRACT

This project introduces an AI-based Resume Analyzer built to automate, simplify, and enhance the traditional candidate screening process. The system applies Natural Language Processing (NLP) and Machine Learning algorithms to extract meaningful information such as technical and soft skills, work experience, education details, certifications, project highlights, and keywords from resume documents. It then intelligently compares the extracted data with a given job description to assess relevance, detect skill gaps, and determine how well the candidate aligns with the job requirements. This ensures a faster and more structured evaluation process, reducing the effort spent on manual screening. The analyzer generates a comprehensive score, highlights key strengths and weaknesses, and provides personalized improvement suggestions to help candidates refine their resumes. For recruiters, it offers a data-driven and unbiased approach to shortlisting, helping them identify the most suitable applicants quickly and accurately. By improving precision, reducing human error, and supporting large-scale recruitment, this AI-based solution contributes to a more efficient, consistent, and intelligent hiring workflow.

ACKNOWLEDGEMENT

I express my gratitude to all those who helped us in various stages of the development of this project. First, I would like to express my sincere gratitude and indebtedness to Dr. Rahul Hans (Coordinator) of DAV University for allowing me to undergo the summer training of 45 days at Sensation Software Solution . I am also thankful to all faculty members of the Department of Computer Science and Engineering for their true help, inspiration, and for helping me with the preparation of the final report and presentation. Last but not least, I pay my sincere thanks and gratitude to all the staff members of Sensation Software Solution for their support and for making our training valuable and fruitful.

DECLARATION

I Tarangini dave, hereby declare that the work that is being presented in this project/training titled “Resume Analyzer” by me, in partial fulfillment of the requirements for the award of a Bachelor of Technology (B. Tech) degree in Computer Science and Engineering, is an authentic record of my work carried out under the guidance of Mr. Shivam (Course Instructor).

To the best of my knowledge, the matter embodied in this report has not been
submitted to any other university/institute for the award of any degree or diploma.

Tarangini dave

12300387

Sensation Solutions is one of the best Companies, well defined to offer customer-centric B2B, B2C services & indigenous products for various industry verticals such as Digital Services, IT Services, Gaming, Travel, Business Consultancy & Legal Services. We have a team of experts that leaves no stone unturned to provide the best solutions.

In the year 2013, Sensation Solutions was born with the aim of empowering local businesses to to expand and register their presence globally. With each passing year, we grew exponentially and the reason behind the tremendous growth, To the point process, our Customer interaction, Development Strategies and of course our dedication to build something amazing par excellence. Today we have reached the stage we are in stands in league of top market leaders in the IT & Digital Services world that specializes in providing the top notch services. You name any IT service, we are able to provide it.

- We believe in **understanding the requirements** of our clients in order to develop a perfect solution which can solves their business problems.
- Our budget friendly approach always makes our client relaxed.
- Getting a project done with full **Confidentiality** has been always a top priority for us.
- We have gained appreciation and trust of our customers due to our end to end services under a roof.

+91- 98158-10580, +91-82647-42696

E-Mail: hr@sensationsolutions.com

Website: https://sensationsolutions.com/

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

In today’s competitive job market, the recruitment process has become increasingly challenging due to the large volume of applications submitted for every job role. Recruiters often spend significant time manually reading, sorting, and analyzing resumes to identify suitable candidates. This traditional method of screening resumes is not only time-consuming but can also lead to bias, inconsistency, and errors, especially when handling thousands of applicants. To overcome these challenges, Artificial Intelligence (AI) has emerged as a powerful tool in modern recruitment systems. An AI-based Resume Analyzer provides an intelligent and automated solution to evaluate resumes accurately, efficiently, and objectively. The AI Resume Analyzer uses advanced technologies such as Natural Language Processing (NLP), Ma chine Lea:r ning (ML), and Text Classification to extract meaningful information from resume documents. These technologies allow the system to identify important elements such as skills, education, work experience, certifications, achievements, and keywords related to the job description. Once the information is extracted, the system analyzes and matches it with predefined job requirements. This enables the tool to generate a relevance score, detect skill gaps, and classify applicants based on their suitability for the position. The main purpose of developing this system is to improve the hiring process by reducing manual screening time and increasing accuracy in candidate selection. It ensures a faster and more data-driven recruitment workflow by automating repetitive tasks and minimizing human intervention. Additionally, the AI Resume Analyzer promotes fairness by offering unbiased
evaluation, free from :h uman preferences or fatigue.

1.1 Key Features of the AI Resume Analyzer:

The system includes several intelligent features, including:

- Automated Resume Parsing: Extracts text from PDF, Word, or image-based resumes and converts it into structured information.
- Keyword Matching: Identifies important keywords from the job description and compares them with those found in the resume.
- Skill Extraction and Classification: Detects both technical and soft skills and categorizes them appropriately.

:

1.2 Tools Used: ：

- Python – Used as the primary programming language for implementing machine learning and NLP functions.
- NLTK (Natural Language Toolkit) – Used for text preprocessing such as tokenization, stemming, stop-word removal, and keyword extraction.
- Scikit-Learn – Used for machine learning model development, training, testing, and generating similarity scores.
- Pandas – Used for data management, dataset handling, and structured storage of extracted resume information.
- TF-IDF Vectorizer – Used to compare resume content with job descriptions and calculate keyword relevance.
- Flask / Streamlit – Used to build the user interface, allowing users to upload resumes and view analysis results.
- Matplotlib / Seaborn – Used for data visualization to present match scores, missing skills, and resume analytics.
- GitHub – Used for code management, version control, and project documentation

Overall, the AI Resume Analyzer represents a step toward smarter recruitment systems. With automation, consistency, scalability, and intelligent decision-making, it transforms the hiring process into a more organized, accurate, and efficient system. As businesses continue to adopt digital and AI-powered tools, systems like this will play a crucial role in shaping the future of employment and talent acquisition.

CHAPTER 2: OBJECTIVES

2.1 Objectives:

- **Automate Resume Screening and Filtering**
    - Minimize manual processing time required to review large volumes of resumes.
    - Automatically filter candidates based on predefined job criteria such as experience, skills, domain, and education level.
    - Speed up the recruitment cycle by performing initial screening in seconds.
- **Improve Accuracy and Consistency in Applicant Evaluation**
    - Apply standardized evaluation rules to ensure all resumes are assessed fairly and uniformly.
    - Detect relevant keywords, measurable achievements, and technical competencies using AI-driven pattern matching.
    - Reduce human error, fatigue, and subjective decision-making.
- **Perform Data-Driven Pre-Existing Resume Analysis**
    - Analyze resumes based on historical screening patterns and previously shortlisted candidates.
    - Compare candidate data with existing resume databases to determine standard deviations in skill sets or profile strength.
    - Identify common trends, candidate improvement gaps, and industry requirements.
    - Allow recruiters to refine job expectations based on real applicant behavior.
- **Enhance Candidate-Job Relevance Matching**
    - Evaluate resumes against specific job descriptions using similarity algorithms like TF-IDF and: cosine similarity.
    - Generate a job suita bility score based on content relevance, keyword density, and experience match.

2.2 Findings:

- **Reduced Resume Screening Time**

  The AI system processed resumes significantly faster than manual screening

  Time reduction increased further when working with large datasets.

- **Improved Resume-to-Job Match Accuracy**

  Similarity scores based on TF-IDF and cosine similarity effectively

  Candidates with relevant skill keywords and experience consistently ranked

- **Impact of ATS-Friendly Formatting**

  Resumes with simple structure and clear section headings were parsed

  Lower keyword extraction accuracy

  Missing or misinterpreted text blocks

- **Grammar and Readability Insights**
  - Grammar errors
  - Spelling mistakes
  - Informal phrasing

2.3 Outputs:

- **Extracted structured information including:**
  - Candidate Name
  - Email and Contact Details
  - Education and Qualifications
  - Work Experience and Job Titles
  - Technical and Soft Skills
  - Certifications and Achievements
- **Keyword Extraction Output**

  Identified relevant job-specific keywords from the resume.

  Highlighted missing or weak keywords based on job description compariso**n**.

CHAPTER 3: HARDWARE AND SOFTWARE REQUIREMENTS

3.1 Hardware Requirements:

- **Processor:** Intel Core i5 (8th Gen or higher) or AMD Ryzen 5 or above
- **RAM:** Minimum 8 GB
- **Storage:** At least 100 GB HDD or SSD (SSD preferred for faster performance)
- **Graphics:** Integrated graphics is sufficient
- **Operating System:** Windows 10 or higher / Ubuntu or other Linux-based systems
- **Display:** Minimum resolution of 1366×768
- **Internet:** Stable internet connection for dataset download and dashboard hosting

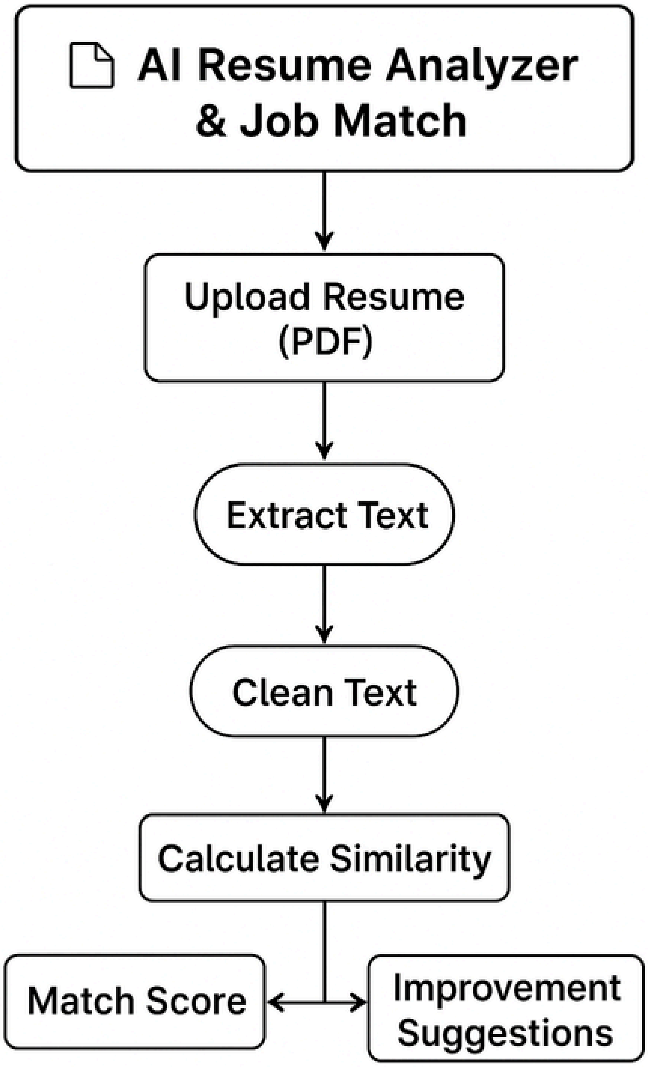
3.2 Software Requirements:

- **Python (version 3.8 or above):** Programming language used for analysis
- **Jupyter Notebook:** Forwritingand testing code
- **Pandas:** For data manipulation
- **NumPy:** Fornumericaloperations
- **Matplotlib and Seaborn:** Fordata visualization
- **Plotly (optional):** Forinteractivevisualizations
- **Streamlit:** Forbuildingdashboards
- **VS Code or Anaconda:** Development environment
- **Web Browser (Chrome/Edge):** To view and interact with the dashboard

CHAPTER 4: SYSTEM DESIGN

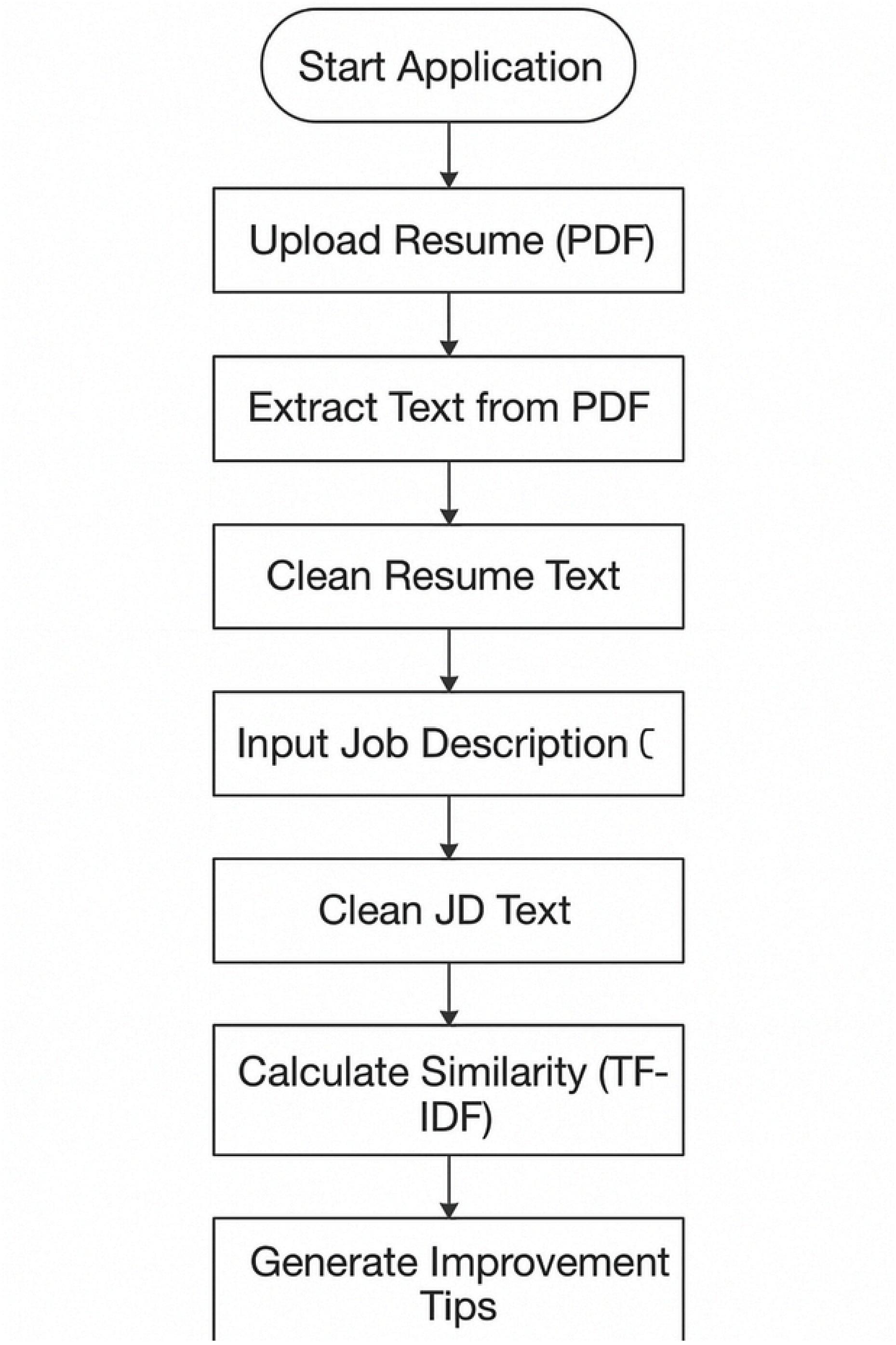
4.1 DFD

A Data FlowDiagram(DFD) represents the flow of data within information systems. Data FlowDiagrams(DFD) provide a graphical representation of the data flow of a system thatbothtechnicaland non-technical users can understand. The models enable softwareengineers,customers, and users to work together effectively during the analysisandspecificationof requirements.



4.2 Flow Chart

Flowcharts are graphical representations of data or algorithms for better visual understanding of the code. It displays step-by-step solutions to a problem, algorithm, or process. It is a pictorial way of representing steps that are preferred by most beginner- level programmers to understand algorithms of computer science, thus it contributes to troubleshooting the issues in the algorithm. A flowchart is a picture of boxes that indicate the process flow sequentially. Since a flowchart is a pictorial representation of a process or algorithm, it's easy to interpret the process.



CHAPTER 5: METHODOLOGY

5.1 Data Input and Acquisition

- The system accepts two inputs:
- Resume File (PDF Format) uploaded through the interface.
- Job Description (Plain Text) entered by the user.
- Streamlit handles user input interaction and file uploading functions.

5.2 Resume Text Extraction

- The uploaded resume is processed using the pdfplumber library.
- Each page is scanned, and text is extracted and combined into a single document.
- Process:
  - Load PDF → Read all pages → Extract text → Append into a continuous string.

5.3 Text Preprocessing & Cleaning

- Both resume and job description texts undergo preprocessing to normalize
- Steps involved:
  - Lowercasing of text
  - Removal of special characters (non-alphanumeric symbols)
  - Removing noise and irrelevant patterns using Regular Expressions (RegEx)
- Purpose:
  - To ensure text consistency and better feature extraction performance.

5.4 Feature Extraction using TF-IDF

- TF-IDF Vectorization converts the cleaned text inputs into numerical
- The TfidfVectorizer() from Scikit-learn is used to transform both resume

5.5 Similarity Computation

- The similarity between the resume and job description is calculated using Cosine Similarity.
- Purpose:
  - To determine how closely the resume content matches the expected profile in the job description.
- Output:
  - A matching score (%) representing resume-job relevance.

5.6 Keyword Gap Analysis and Suggestions

- NLP pattern matching identifies missing keywords by comparing:
- Extracted resume tokens
- Extracted job description keywords
- The system generates personalized recommendations such as:
- Missing technical skills
- Resume structure improvements
- Content expansion if resume is too short

5.7 Result Visualization and User Interaction

The processed output is displayed interactively through the Streamlit interface. Results include:
Match Percentage
Keyword-based recommendations
Optional extracted resume text for transparency
This enhances user understanding and allows them to interpret how the system derived its evaluation score.

5.8 System Feedback Loop

Finally, the system encourages iterative refinement. Users can revise their resumes based on the suggestions provided, re-upload them, and compare changes in match score. This feedback cycle improves resume quality progressively and increases chances of passing real ATS systems. The loop can be summarized as: User Uploads Resume → System Analyzes → Suggestions Provided → Resume Updated → Re-Testing This iterative pipeline transforms the tool into a practical resume optimization system.

CHAPTER 6: RESULT ANALYSIS

6.1 Match Score Analysis

- The similarity score ranged between resumes depending on keyword relevance and content quality.
    - Higher scores were observed when:
        - The resume contained job-specific skills and terminology.
        - Technical keywords matched exactly with job description requirements.
        - Sections like Projects, Experience, and Skills were clearly mentioned.
    - Lower scores were observed when:
        - Resume used vague or generic descriptions.
        - Required skills or terms were missing or differently phrased.
        - The resume lacked measurable achievements or detailed content.

6.2 Keyword Detection and Missing Keyword Reporting

- The system successfully identified keywords from job descriptions.
    - Correct keyword matches were detected when:
        - The exact or similar term existed in the resume.
        - The wording aligned with job-specific language.
    - Missing keyword suggestions occurred when:
        - Resume lacked technical skills (e.g., Python, SQL, Machine Learning).
        - Soft skills or domain-specific terminology were not included.

6.3 Resume Content Quality Evaluation

The system evaluated the richness and completeness of content.

- Strength indicators included:
    - Detailed descriptions of responsibilities and achievements.
    - Well-structured sections such as Skills, Projects, Experience, and Education.
- Weakness indicators included:
    - Short resumes with minimal text.
    - Lack of real experience or project-based evidence

6.4 ATS Readiness Feedback

The system provided actionable suggestions on resume formatting and structure.

- Positive ATS-ready features detected:
    - Clear section headings.
    - Keyword alignment with job requirements.
    - Consistent use of standard resume terminology.

6.5 User Experience and Interaction Performance

- The UI allowed users to interact with the system easily.
  - Positive findings:
    - Uploading PDFs and pasting job descriptions was user-friendly.
    - Instant feedback improved user engagement.
  - Areas noted for future enhancement:
    - Support for more document types (DOCX, TXT).
    - Advanced filtering or scoring breakdown visualization.

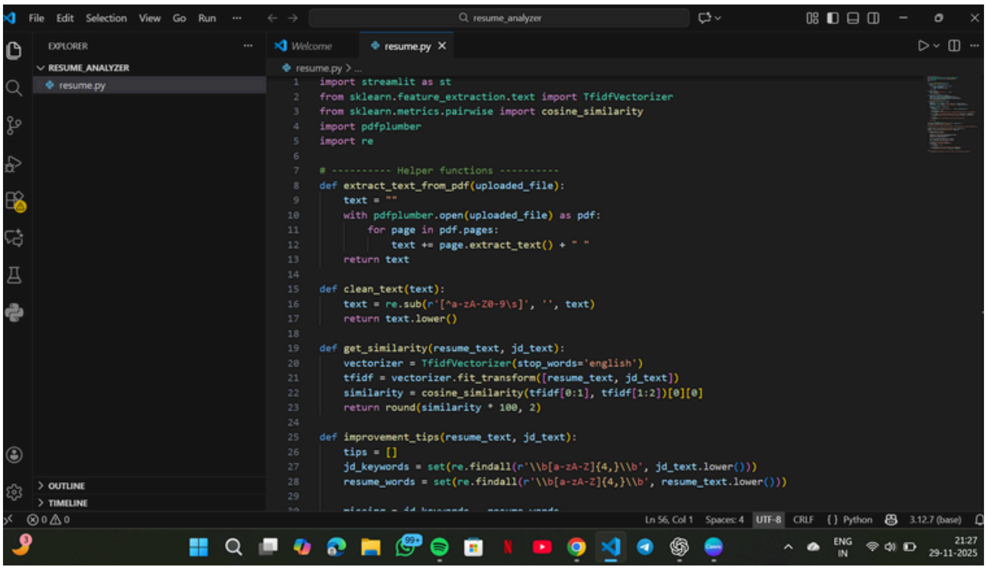
6.6 Consistency and Accuracy Observations

- The system produced consistent scoring across repeated tests.
  - Consistency strengths:
    - Same resume and job input always returned the same similarity score.
    - Keyword analysis remained stable across execution cycles.
  - Areas of accuracy improvement:
    - Synonym detection (e.g., "ML" vs "Machine Learning").
    - Semantic understanding beyond direct word matching.

6.7 Overall Performance

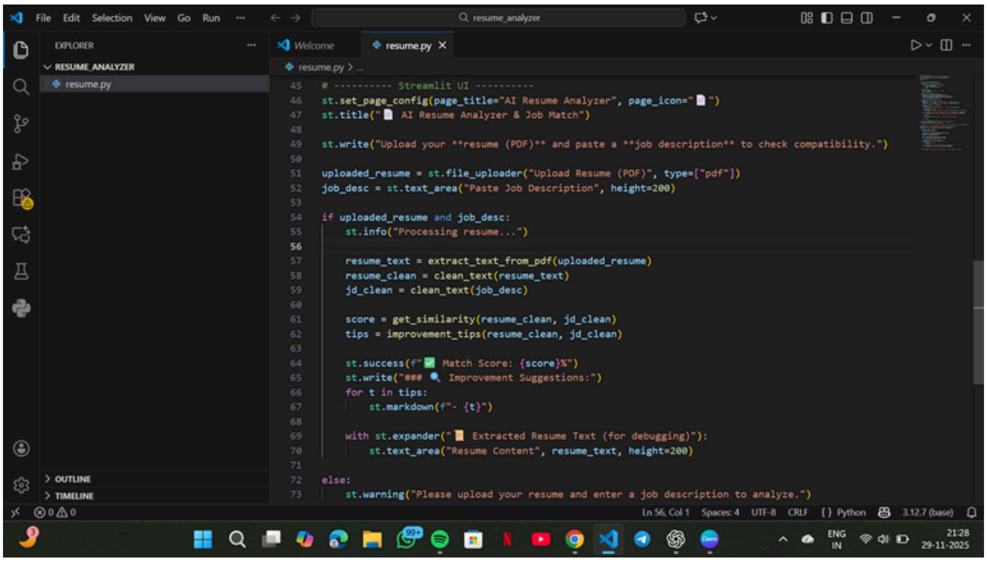
- The AI Resume Analyzer reliably:
  - Extracted text from resumes.
  - Cleaned and processed data using NLP.
  - Compared content using TF-IDF and cosine similarity.
  - Generated customized suggestions for optimization.

SCREENSHOTS:



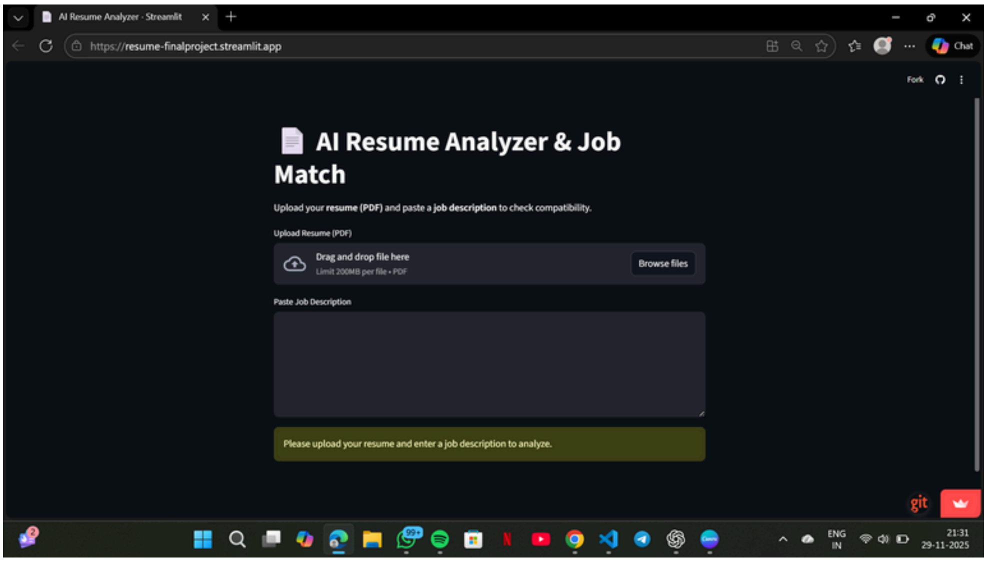
```python
import streamlit as st
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import pdfplumber
import re

# ---------- Helper functions ----------
def extract_text_from_pdf(uploaded_file):
    text = ""
    with pdfplumber.open(uploaded_file) as pdf:
        for page in pdf.pages:
            text += page.extract_text() + " "
    return text

def clean_text(text):
    text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    return text.lower()

def get_similarity(resume_text, jd_text):
    vectorizer = TfidfVectorizer(stop_words='english')
    tfidf = vectorizer.fit_transform([resume_text, jd_text])
    similarity = cosine_similarity(tfidf[0:1], tfidf[1:2])[0][0]
    return round(similarity * 100, 2)

def improvement_tips(resume_text, jd_text):
    tips = []
    jd_keywords = set(re.findall(r'\\b[a-zA-Z]{4,}\\b', jd_text.lower()))
    resume_words = set(re.findall(r'\\b[a-zA-Z]{4,}\\b', resume_text.lower()))

    missing = jd_keywords - resume_words
```

Fig. 6 .4. 1



```python
# ---------- Streamlit UI ----------
st.set_page_config(page_title="AI Resume Analyzer", page_icon="📄")
st.title("📄 AI Resume Analyzer & Job Match")

st.write("Upload your **resume (PDF)** and paste a **job description** to check compatibility.")

uploaded_resume = st.file_uploader("Upload Resume (PDF)", type=["pdf"])
job_desc = st.text_area("Paste Job Description", height=200)

if uploaded_resume and job_desc:
    st.info("Processing resume...")

    resume_text = extract_text_from_pdf(uploaded_resume)
    resume_clean = clean_text(resume_text)
    jd_clean = clean_text(job_desc)

    score = get_similarity(resume_clean, jd_clean)
    tips = improvement_tips(resume_clean, jd_clean)

    st.success(f"✅ Match Score: {score}%")
    st.write("### 🔍 Improvement Suggestions:")
    for t in tips:
        st.markdown(f"- {t}")

    with st.expander("📜 Extracted Resume Text (for debugging)"):
        st.text_area("Resume Content", resume_text, height=200)

else:
    st.warning("Please upload your resume and enter a job description to analyze.")
```

Fig. 6 .4. 2


AI Resume Analyzer · Streamlit
https://resume-finalproject.streamlit.app
Chat
Fork
git
ENG IN
21:31
29-11-2025

📄 AI Resume Analyzer & Job Match

Upload your **resume (PDF)** and paste a **job description** to check compatibility.

Upload Resume (PDF)

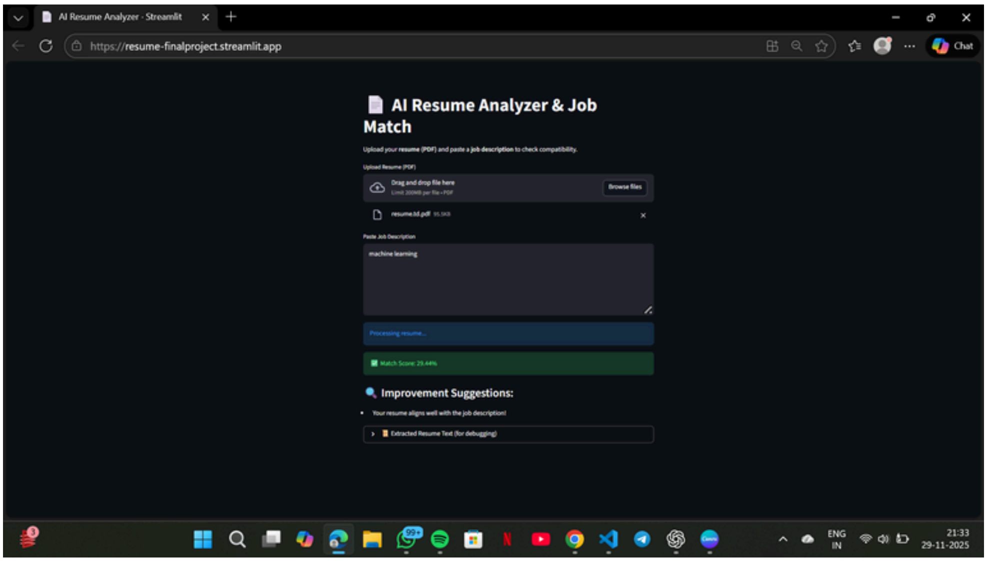
Drag and drop file here
Limit 200MB per file • PDF

Browse files

Paste Job Description

Please upload your resume and enter a job description to analyze.

Fig. 6.4.3


AI Resume Analyzer · Streamlit
https://resume-finalproject.streamlit.app
Chat
ENG IN
21:33
29-11-2025

📄 AI Resume Analyzer & Job Match

Upload your **resume (PDF)** and paste a **job description** to check compatibility.

Upload Resume (PDF)

Drag and drop file here
Limit 200MB per file • PDF

Browse files

resume.td.pdf 95.5KB

Paste Job Description

machine learning

Processing resume...

✅ Match Score: 29.44%

🔍 Improvement Suggestions:

- Your resume aligns well with the job description!

› Extracted Resume Text (for debugging)

Fig. 6.4.4

CHAPTER 7: CONCLUSION AND FUTURE SCOPE

7.1 CONCLUSION:

The AI Resume Analyzer successfully applies NLP and machine learning to evaluate how well a resume matches a job description. By extracting and processing text, computing similarity scores, and identifying missing keywords, the system provides meaningful feedback to help users improve their resumes for better job relevance and ATS compatibility. With its simple Streamlit interface and automated analysis, the tool proves effective in supporting job seekers in enhancing resume quality and increasing employability

7.2 FUTURE SCOPE:

The AI Resume Analyzer has strong potential for further enhancement and real-world application. In future development, the system can integrate advanced NLP models such as BERT, GPT, or transformer-based semantic analysis to improve accuracy and understand contextual meaning rather than only keyword matching. Support for multiple file formats like DOCX, TXT, and LinkedIn profile data extraction can also be added to enhance usability. Additionally, the system may evolve into a full career-assistance platform by incorporating features such as automated resume formatting, ATS-friendly template generation, personalized skill gap analysis, and job role recommendations based on the candidate’s profile. Integration with industry databases, recruitment platforms, and HR systems can further make the tool scalable and beneficial at institutional and corporate levels. Machine learning feedback loops can also be implemented, allowing the system to continuously learn from user behavior and recruiter standards, improving precision over time. Ultimately, the future scope of this system holds promise for transforming the traditional job application process into a more efficient, data-driven, and personalized experience.

CHAPTER 8: REFERENCE:

- **Plotly – Interactive Data Visualization**

  https://github.com/taranginidave/AI_RESUME_ANALYZER

- **Streamlit – Build Interactive Dashboards**

  https://resume-finalproject.streamlit.app/