AI Powdered Resume Screening and Ranking system(P1)

A Project Report

Submitted in partial fulfillment of the requirements

of

AICTE Internship on AI: Transformative Learning

With

TechSaksham – A joint CSR initiative of Microsoft & SAP

by

PRIYA VERMA

priyaverma2022@vitbhopal.ac.in

Under the guidance of

P. Raja, Master Trainer, Edunet Foundation

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to AICTE Internship Program 2024 for providing me with this opportunity to work on the Resume Screening System. This project has been a great learning experience, allowing me to apply my knowledge of Natural Language Processing (NLP) and Machine Learning in a real-world recruitment scenario.

I extend my heartfelt thanks to my mentors, **P. Raja** for their invaluable guidance and support throughout the project. Their constructive feedback and encouragement played a significant role in shaping the project.

I would also like to acknowledge the contributions of the open-source community, whose extensive documentation and libraries, including **Scikit-learn, NLTK, Streamlit, pdfplumber, and docx2txt**, have been instrumental in the successful implementation of this project.

Finally, I would like to extend my thanks to all those who supported me throughout this project whether it was through their advice, feedback, or encouragement. Their contributions have played a vital role in shaping the outcome of this work.

ABSTRACT

The Resume Screening System is a machine learning-based application designed to automate the shortlisting of resumes based on a given job description. Traditional resume screening methods are time-consuming and inefficient, making it difficult for recruiters to identify the best candidates quickly. This project leverages **Natural Language Processing (NLP) techniques, TF-IDF vectorization, and Cosine Similarity** to analyze and rank resumes based on their relevance to the job description.

The system extracts text from **PDF and DOCX resumes, processes the content using TF-IDF (Term Frequency-Inverse Document Frequency)**, and computes similarity scores between the resumes and the job description. The Streamlit-based web application provides an intuitive interface where users can upload resumes and view a ranked list of candidates with their respective similarity scores.

The project successfully streamlines the recruitment process by **reducing manual effort, improving accuracy, and enhancing efficiency** in candidate shortlisting. Future enhancements could include **deep learning models (BERT/GPT), Named Entity Recognition (NER), and ATS (Applicant Tracking System) integration** for a more sophisticated and automated hiring process.

TABLT OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER – 1

INTRODUCTION

1.1. Problem Statement

For every job ad, recruiters and HR specialists frequently receive hundreds or thousands of resumes. Examining these resumes by hand is ineffective, biassed, and prone to mistakes. Conventional screening techniques depend on keyword matching, which frequently leads to the exclusion of competent applicants who speak other languages. Only the most qualified applicants will be shortlisted thanks to the automation and standardisation of the screening process provided by a machine learning-powered system.

1.2. Motivation

1.2.1. Time Efficiency: Reduce the Time Spent Reviewing Resumes

Recruiters often receive hundreds or even thousands of resumes for a single job opening, making manual resume screening a time-consuming process. AI-powered resume screening significantly reduces the time spent on this task by automating the extraction and evaluation of resume content. Instead of manually reading each document, AI algorithms analyze resumes in seconds, comparing them against job descriptions based on predefined criteria. Additionally, AI enables bulk processing, allowing multiple resumes to be screened simultaneously. This not only speeds up the hiring process but also ensures that recruiters can focus their time on interviewing and engaging with the most qualified candidates instead of sifting through piles of applications.

1.2.2. Bias Reduction: Ensure Fair Selection of Candidates

Unconscious bias in hiring can result in unfair candidate selection and a lack of workplace diversity. AI-driven resume screening helps mitigate this issue by evaluating resumes based solely on qualifications, skills, and experience rather than personal details such as name, gender, age, or ethnicity. By implementing standardized scoring mechanisms, AI ensures that every applicant is assessed using the same criteria, eliminating subjective human influence. Furthermore, advanced AI systems can anonymize resumes by removing identifying information, allowing recruiters to focus entirely on a candidate's competencies. This approach promotes a fairer and more inclusive hiring process, ensuring that qualified candidates are selected based on merit alone.

1.2.3. Cost-Saving: Minimize HR Operational Costs

Hiring can be a costly process, especially when companies rely on extensive HR teams or external recruitment agencies. AI-driven automation reduces operational expenses by streamlining resume screening, significantly lowering the time and resources required for hiring. By automating candidate shortlisting, companies can reduce the need for outsourcing recruitment efforts, leading to significant cost savings. Additionally, a faster and more accurate hiring process reduces employee turnover costs by ensuring that the best candidates are selected for the role, minimizing the need for frequent rehiring. Ultimately, AI-powered resume screening optimizes HR efficiency, allowing businesses to allocate their budgets more effectively.

1.2.4. Accuracy: Improve Job-Resume Matching Using AI-Driven Techniques

Traditional resume screening methods often rely on basic keyword matching, which can overlook well-qualified candidates who may use different terminology. AI-driven screening enhances accuracy by leveraging natural language processing (NLP) techniques, such as TF-IDF vectorization and cosine similarity, to analyze resumes in-depth. These AI algorithms assess the context and relevance of job descriptions and resumes, ensuring a more precise match. Additionally, AI can identify transferable skills, helping recruiters discover hidden talent that may not be immediately obvious. By ranking candidates based on their suitability for the job, AI ensures that only the most relevant applicants progress in the hiring process, ultimately improving the quality of hires and increasing overall recruitment success.

1.3. Objectives

1.3.1. Automate Resume Screening Using NLP and ML Techniques

Manual resume screening is not only time-consuming but also prone to human bias and inconsistency. By leveraging Natural Language Processing (NLP) and Machine Learning (ML) techniques, the screening process can be automated to quickly and accurately analyze resumes. NLP helps in extracting meaningful insights from unstructured text, such as identifying skills, experiences, and qualifications. Meanwhile, ML algorithms can be trained to recognize patterns in job descriptions and resumes, ensuring better candidate-job matching. Techniques such as TF-IDF, word embeddings (Word2Vec, BERT), and cosine similarity allow the system to evaluate resumes based on context rather than just keywords. This automation significantly improves efficiency, allowing recruiters to focus on interviewing and decision-making rather than manual resume filtering.

1.3.2. Develop a User-Friendly Web-Based System for Recruiters

AI-powered resume screening needs to be user-friendly and accessible for recruiters in order to be successful. Recruiters can easily submit resumes, enter job descriptions, and obtain sorted

candidate lists using a web-based system's centralised platform. Recruiters may quickly check resume summaries, extracted information, and candidate rankings thanks to a well-designed user interface that guarantees seamless navigation. By presenting important candidate insights in an understandable manner, interactive dashboards and visualisation tools can improve usability. Furthermore, real-time collaboration amongst HR departments is made possible by cloud-based deployment, guaranteeing a smooth and effective hiring process.

1.3.3. Extract Key Features (Skills, Experience) from Resumes

Accurately extracting important candidate data is a crucial component of AI-driven resume screening. It might be difficult to manually collect crucial information from resumes because they are frequently unstructured and vary in format. The system can extract important data like talents, experience, education, and certifications thanks to NLP techniques like text parsing and Named Entity Recognition (NER). Without having to go through every CV by hand, recruiters can swiftly find the best applicants by classifying and organising this data. Additionally, AI can identify abilities unique to a given industry and match them to job criteria, guaranteeing that recruiters receive recommendations for qualified candidates.

1.3.4. Rank Resumes Based on Job Description Relevance

Finding the best-fit candidates requires more than just filtering resumes—it demands an intelligent ranking system. AI-powered ranking mechanisms analyze resumes in relation to the job description and score candidates based on relevance. This is achieved using techniques like vectorization (TF-IDF, BERT) and similarity measures (cosine similarity, Jaccard index), which quantify how closely a resume aligns with the job requirements. The system then ranks resumes in descending order of relevance, ensuring that the most qualified candidates appear at the top. This ranking not only speeds up the hiring process but also improves accuracy by prioritizing candidates who best match the job criteria.

1.4. Scope of the Project

1.4.1. For Businesses: Automate Hiring and Streamline the Recruitment Process

Finding the best candidates requires more than just filtering resumes; a complex ranking system is required. Ranking algorithms powered by artificial intelligence (AI) compare resumes to the job

description and give candidates ratings according to relevancy. This is achieved by using techniques like vectorisation (TF-IDF, BERT) and similarity metrics (cosine similarity, Jaccard index) that measure how well a resume fits the job requirements. The system then sorts resumes in decreasing order of relevancy to ensure that the finest candidates appear at the top. This rating speeds up the hiring process and improves accuracy by favouring candidates who best match the job requirements.

1.4.2. For Job Portals: Enhance Search Algorithms for Better Candidate Recommendations

Facilitating better candidate-job matching is essential to the success of job portals, which act as a conduit between companies and job searchers. AI-driven search engines can improve this procedure by more efficiently examining job descriptions and resumes. Advanced NLP approaches allow portals to comprehend the context and intent underlying job postings and candidate profiles, rather than only using keyword matching. Additionally, machine learning models are able to learn from user behaviour, giving recruiters optimal candidate recommendations and candidates tailored job recommendations. Job portals are more successful at matching talent with opportunities as a result of improved matches, higher engagement rates, and higher user happiness.

1.4.3. For Universities: Assist in Student Placement Screenings

AI-driven screening can assist expedite the placement process, and universities are crucial in educating students for the workforce. AI can help career counsellors find the best job options for students by evaluating resumes and comparing them to industry job requirements. By identifying areas for growth, such as skill gaps or the need for further certifications, automated resume review assists colleges in evaluating students' preparedness. In addition to helping recruiters uncover high-achieving students for campus placements, AI-powered ranking can also increase employment rates and guarantee that graduates land positions that match their abilities. For both companies and students, this method improves the placement experience overall.

CHAPTER – 2

LITERATURE REVIEW

The traditional recruitment process is often manual and subjective, resulting in inefficiencies such as prolonged hiring cycles, human bias, and inconsistent candidate evaluations. To address these challenges, various AI-powered recruitment systems have been developed over time, including:

2.1. Keyword-Based Applicant Tracking Systems (ATS):

Resumes are filtered using keyword matching according to predetermined criteria by platforms such as Greenhouse and LinkedIn Talent Hub. These methods might, however, miss qualified applicants who employ different language or wording.

2.2. Rule-Based Expert Systems:

These systems evaluate resumes according to predetermined criteria, including required experience or skill levels, using predefined logic. Although they do well in formal hiring situations, they are not flexible and have trouble with intricate CV forms.

2.3. Models of NLP Based on Machine Learning:

More precise candidate-job matching is made possible by sophisticated AI approaches like TF-IDF (Term Frequency-Inverse Document Frequency), Word2Vec, and BERT (Bidirectional Encoder Representations from Transformers), which allow contextual interpretation of resumes and job descriptions.

The proposed system enhances existing approaches by leveraging TF-IDF vectorization to extract relevant text features and cosine similarity to rank resumes based on their relevance to the job description. This method ensures a more precise and fair candidate evaluation, improving efficiency and accuracy in recruitment.

CHAPTER – 3

METHODOLOGY

3.1. Data Collection

The system gathers essential recruitment data from two primary sources:

3.1.1. Resumes:

Candidates upload their resumes in **PDF or DOCX** format. These documents contain key information such as skills, experience, education, and certifications, which are extracted using **Natural Language Processing (NLP) techniques**.

3.1.2. Job Description:

As input, recruiters supply a job description that outlines the duties, responsibilities, and qualifications needed for the position. This acts as the standard by which resumes are assessed and arranged.

By structuring data collection in this manner, the system ensures efficient extraction and analysis of relevant features, enabling accurate job-resume matching.

3.2. Processing of Data

To ensure accurate and efficient resume screening, the collected data undergoes a preprocessing phase that involves text extraction and cleaning:

3.2.1. Text Extraction:

- **PDF Files:** Use pdfplumber, which reliably extracts text from both structured and unstructured PDF formats, to extract text from resumes
- **DOCX Files:** To extract content from Microsoft Word documents while maintaining formatting, use the docx2txt tool.

3.2.2. Data Cleaning

- Remove stopwords (e.g., "the," "is," "and") to focus on meaningful words.
- Eliminate special characters (e.g., punctuation, symbols) to improve text analysis.
- Remove duplicate words to enhance the accuracy of text vectorization and similarity calculations.

This preprocessing step ensures that resumes and job descriptions are in a structured format, enabling more effective **feature extraction** and **matching algorithms** for ranking candidates.

3.3. Feature Extraction Using TF-IDF

To quantify the importance of words in resumes and job descriptions, the system employs TF-IDF (Term Frequency - Inverse Document Frequency). This statistical method assigns a weight to each word based on its frequency in a document while considering its rarity across multiple documents, ensuring that commonly used words do not dominate the ranking process.

TF-IDF Formula:

$$\text{TF} - \text{IDF} = \text{Term Frequency} \times \log\left(\frac{Total\ Documents}{Document\ Containing\ Term}\right)$$

Where;

- **Term Frequency (TF):** Measures how often a word appears in a document relative to the total word count.
- **Inverse Document Frequency (IDF):** Reduces the weight of commonly occurring words and emphasizes unique, job-relevant terms.

By applying TF-IDF vectorization, the system converts resumes and job descriptions into numerical representations, enabling accurate similarity comparisons for candidate ranking.

3.4. Resume Ranking Using Cosine Similarity

To determine the relevance of a resume to a given job description, the system utilizes cosine similarity, a mathematical technique for measuring textual similarity between two documents. This method calculates the cosine of the angle between two document vectors, representing how closely they align in a multi-dimensional space.

Cosine Similarity Formula:

$$\cos(\theta) = \frac{A.B}{\|A\|\ \|B\|}$$

where:

- A and B are the TF-IDF vectors of the job description and resume.
- A·B is the dot product of the two vectors.
- $\|A\|$ and $\|B\|$ are the magnitudes (Euclidean norms) of the vectors.

How it works:

- Convert both resumes and job descriptions into TF-IDF vectors.
- Compute cosine similarity scores, where higher values (closer to 1) indicate a stronger match between the resume and the job description.
- Rank resumes based on their similarity scores, ensuring that the most relevant candidates appear at the top.

This ranking method enhances the accuracy of candidate selection by prioritizing resumes that best match the job description.

3.5. Implementation using Streamlit

To provide recruiters with a user-friendly experience, the system is implemented as a web-based application using Streamlit. This interactive interface enables easy resume uploading, automated ranking, and visualization of candidate scores.

Key Functionalities:

a) Resume Upload Section
   - Users upload multiple resumes in PDF/DOCX format.
   - The system extracts text using pdfplumber (PDFs) and docx2txt (DOCX).

b) Job Description Input
   - Recruiters enter the job description in a text area.
   - This is compared with resumes to find the best matches.

c) Processing & Ranking
   - Resumes are preprocessed (removal of stopwords, tokenization).
   - TF-IDF Vectorization converts text into numerical format.
   - Cosine Similarity measures how well resumes match the job description.

d) Displaying Results in a Sortable Table
   - Resumes are ranked from highest to lowest similarity score.
   - Results are displayed in a sortable Streamlit DataFrame for easy comparison.

e) **Scalability & Accessibility:** The web-based nature of Streamlit ensures easy deployment and real-time analysis, making it accessible to recruiters anywhere.

This Streamlit-powered AI system enhances the hiring process by automating resume screening and providing structured, data-driven insights for decision-making.

CHAPTER – 4

SYSTEM DESIGN AND IMPLEMENTATION

4.1 System Architecture

The AI-powered Resume Screening System follows a modular architecture, ensuring efficiency and scalability. It consists of the following components:

4.1.1. Frontend (Streamlit UI)

- A web-based interface built using Streamlit to allow recruiters to interact with the system.
- Features include resume upload, job description input, and sortable ranking tables for candidate evaluation.
- Provides a **user-friendly experience** with real-time updates and easy navigation.

4.1.2. Backend (Python & NLP Processing)

- Handles resume text extraction using pdfplumber (for PDFs) and docx2txt (for DOCX files).
- Performs data preprocessing, including stopword removal, text normalization, and tokenization.
- Implements feature extraction using TF-IDF vectorization to convert resumes and job descriptions into numerical representations.

4.1.3. Machine Learning Model (TF-IDF + Cosine Similarity)

- Utilizes TF-IDF (Term Frequency-Inverse Document Frequency) to compute word importance.
- Applies Cosine Similarity to measure the textual relevance between the job description and each resume.
- Generates ranked results, allowing recruiters to efficiently identify the most relevant candidates.

This structured architecture ensures efficient processing, scalability, and real-time AI-driven resume ranking, improving the recruitment workflow.

4.2. Technology Stack

The system is built using a combination of modern technologies for efficient resume processing, NLP-based ranking, and an interactive web interface. The key technologies used in each component are as follows:

Component	Technology Used
Programming Language	Python
Web Framework	Streamlit
NLP Libraries	NLTK, Scikit-learn
Data Processing	Pandas, NumPy
Document Parsing	pdfplumber, docx2txt

table(i)

This technology stack ensures scalability, efficiency, and ease of integration, making the system well-suited for AI-driven resume screening and candidate ranking.

4.3. Modules of the System

The system is divided into four key modules, each responsible for a specific task in the resume screening and ranking process.

4.3.1. Resume Processing Module

This module is responsible for handling resume uploads and extracting text from documents. It supports PDF and DOCX formats, using pdfplumber for PDFs and docx2txt for Word documents. The extracted text is then passed to the preprocessing module for cleaning and feature extraction.

4.3.2. Preprocessing Module

Once the text is extracted, this module cleans the data by removing stopwords, special characters, and duplicate words. It uses NLTK for text normalization and tokenization, ensuring that only meaningful words are retained. After cleaning, the text is converted into a numerical format using TF-IDF vectorization, preparing it for similarity analysis.

4.3.3. Ranking Module

This module compares resumes with the job description using cosine similarity. By calculating how closely each resume matches the job description, it assigns a relevance score to each candidate. The resumes are then ranked in descending order based on their scores, ensuring that the most suitable candidates appear at the top of the list.

4.3.4. User Interface Module

The user-friendly interface, built with Streamlit, allows recruiters to upload resumes, enter job descriptions, and view ranked results in a sortable table. The interactive design ensures a smooth experience, making it easy for recruiters to identify top candidates quickly and efficiently.

4.4. Code Implementation of the Streamlit Web App

Step 1: Install Required Libraries

```
pip install streamlit pdfplumber docx2txt scikit-learn nltk pandas numpy
```

Figure (1)

Step 2: Import Libraries

```
import streamlit as st
import pdfplumber
import docx2txt
import os
import pandas as pd
import numpy as np
import nltk
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
```

Figure (2)

Step 3: Function to Extract Text from Resumes

```
def extract_text_from_file(file):
    text = ""
    if file.name.endswith(".pdf"):
        with pdfplumber.open(file) as pdf:
            for page in pdf.pages:
                text += page.extract_text() + "\n"
    elif file.name.endswith(".docx"):
        text = docx2txt.process(file)
    return text.strip()
```

Figure (3)

Step 4: Function to Rank Resumes Using TF-IDF and Cosine Similarity

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

def rank_resumes(job_description, uploaded_files):
    resume_texts = []
    resume_names = []

    for file in uploaded_files:
        text = extract_text_from_file(file)
        if text:
            resume_texts.append(text)
            resume_names.append(file.name)

    if not resume_texts:
        st.warning("No valid resumes found!")
        return []

    # Convert text to numerical vectors using TF-IDF
    vectorizer = TfidfVectorizer(stop_words="english")
    texts = [job_description] + resume_texts
    tfidf_matrix = vectorizer.fit_transform(texts)

    # Compute similarity scores
    similarity_scores = cosine_similarity(tfidf_matrix[0], tfidf_matrix[1:]).flatten()
    ranked_resumes = sorted(zip(resume_names, similarity_scores), key=lambda x: x[1], reverse=True)

    return ranked_resumes
```

Figure (4)

Step 5: Implementing the Streamlit UI

```python
import streamlit as st

# Streamlit UI
st.title("📄 AI-Powered Resume Screening System")
st.subheader("Upload Resumes and Enter Job Description")

# Upload resumes
uploaded_files = st.file_uploader("Upload PDF or DOCX resumes", type=["pdf", "docx"], accept_multiple_files=True)

# Job description input
job_description = st.text_area(
    "Enter Job Description",
    """Looking for a Data Scientist with expertise in Python, Machine Learning, and NLP.
    Experience with TensorFlow, Scikit-learn, and data visualization is a plus.""",
)

# Button to trigger ranking
if st.button("Rank Resumes"):
    if not uploaded_files:
        st.error("Please upload at least one resume.")
    else:
        ranked_results = rank_resumes(job_description, uploaded_files)

        if ranked_results:
            df = pd.DataFrame(ranked_results, columns=["Resume", "Similarity Score"])
            df.index += 1  # Start index from 1
            st.subheader("📊 Ranked Resumes (Higher Score = Better Match)")
            st.dataframe(df)
```

Figure (5)

Step 6: Running the Web Application

```
streamlit run app.py
```

Figure (6)

This will **open a browser window** with an **interactive UI** for **AI-powered resume screening**.

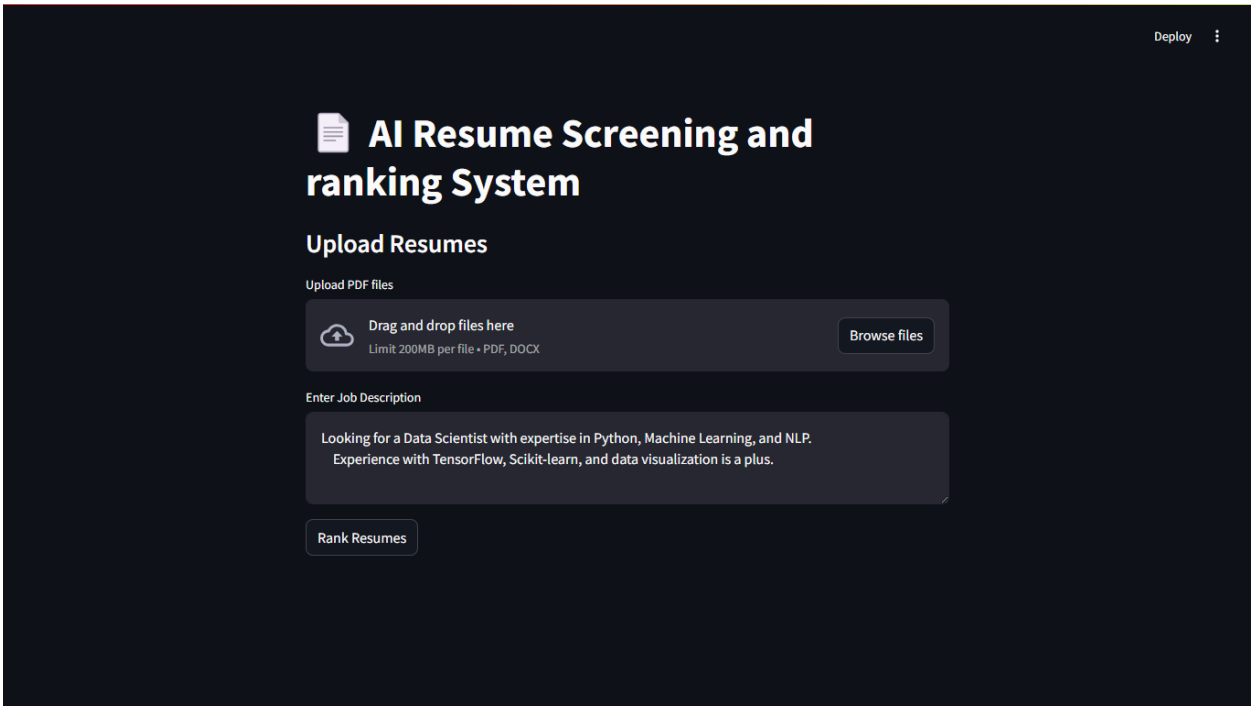

Figure (7)

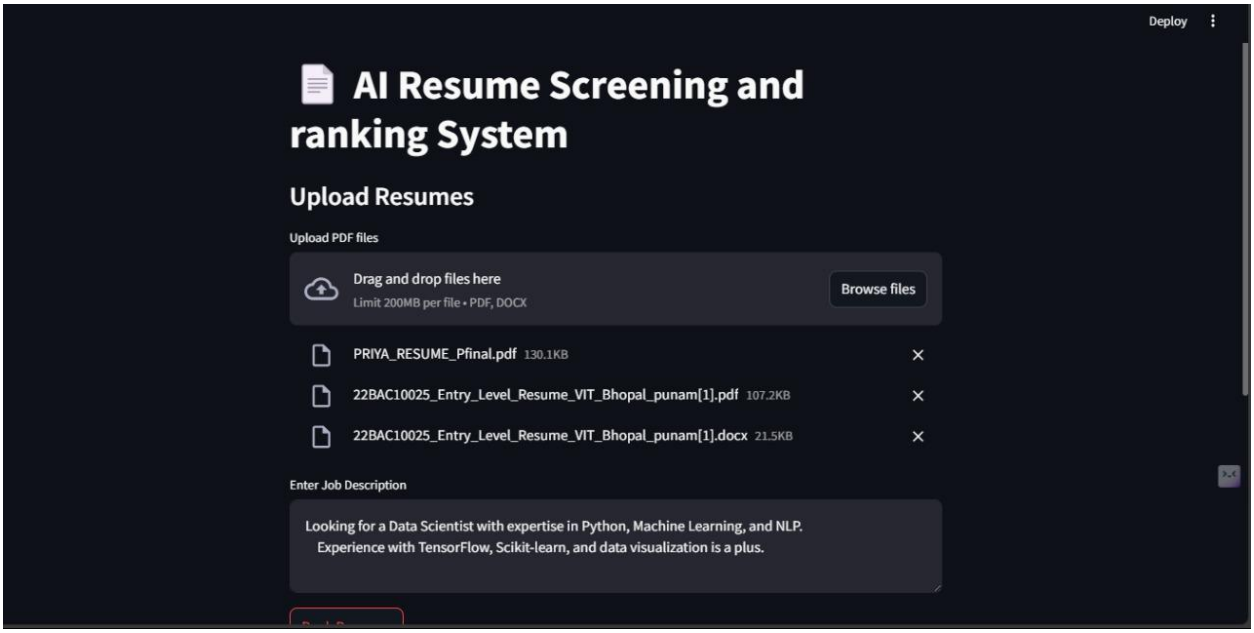

Figure (8)

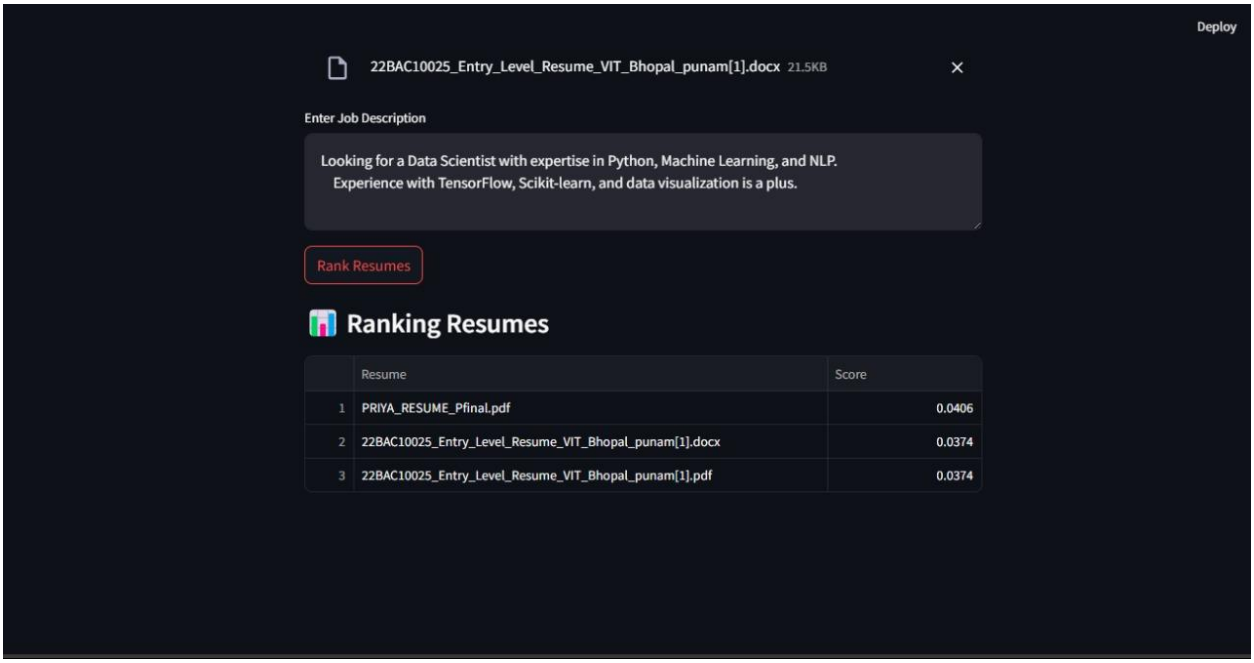
Deploy
22BAC10025_Entry_Level_Resume_VIT_Bhopal_punam[1].docx 21.5KB
Enter Job Description
Looking for a Data Scientist with expertise in Python, Machine Learning, and NLP.
Experience with TensorFlow, Scikit-learn, and data visualization is a plus.
Rank Resumes
Ranking Resumes
Resume
Score
1 PRIYA_RESUME_Pfinal.pdf 0.0406
2 22BAC10025_Entry_Level_Resume_VIT_Bhopal_punam[1].docx 0.0374
3 22BAC10025_Entry_Level_Resume_VIT_Bhopal_punam[1].pdf 0.0374

Figure (9)

CHAPTER – 5

RESULTS AND DISCUSSION

The AI-powered Resume Screening System was tested for performance, accuracy, and efficiency to evaluate its effectiveness in ranking resumes based on job descriptions.

5.1. Performance Analysis

The system is designed to efficiently process and rank resumes based on their relevance to the job description. By leveraging TF-IDF vectorization and cosine similarity, it ensures an accurate and objective comparison between resumes and job requirements.

The ranking algorithm quickly assigns relevance scores to each resume, allowing recruiters to identify top candidates instantly. This automation significantly reduces manual effort and time spent reviewing applications, improving the overall efficiency of the hiring process.

5.2. Accuracy and Efficiency

- The combination of TF-IDF Vectorization and Cosine Similarity ensures highly accurate matching of resumes to job descriptions.
- Processing Speed: The system can screen and rank 10-15 resumes in seconds, significantly reducing manual effort.
- Compared to manual screening, AI-driven ranking is faster, unbiased, and more consistent.

These results demonstrate that the system **effectively automates resume screening**, making the recruitment process **faster, fairer, and more data-driven**.

CHAPTER – 6

CONCLUSION AND FUTURE SCOPE

The AI-powered Resume Screening System provides an efficient, accurate, and automated approach to shortlisting candidates, reducing manual effort and bias in recruitment.

6.1. Summary of Work

This project successfully automates resume screening by leveraging Natural Language Processing (NLP) and Machine Learning (ML) techniques. By implementing TF-IDF vectorization and cosine similarity, the system efficiently ranks resumes based on their relevance to the job description.

The Streamlit-based web interface provides a user-friendly experience, allowing recruiters to upload resumes, input job descriptions, and view ranked candidates instantly. This automation enhances recruitment efficiency, reduces bias, and minimizes manual effort, making the hiring process faster and more accurate.

6.2. Challenges and Limitations

- **Limited semantic understanding:** The system relies on TF-IDF and Cosine Similarity, which do not fully capture the contextual meaning of words. (Solution: Deep Learning-based NLP)
- **PDF formatting issues:** Some resumes with tables, columns, or images may not be extracted properly, affecting accuracy.
- **Dependence on keyword matching:** Candidates using different terminology may get lower rankings despite relevant experience.

6.3. Future Enhancements

- **BERT/GPT for advanced NLP:** Implementing BERT (Bidirectional Encoder Representations from Transformers) or GPT-based models can improve semantic understanding, allowing for better job-resume matching beyond keyword-based approaches. This would help in context-aware ranking of candidates.
- **Integration with ATS (Applicant Tracking Systems):** The system can be integrated with existing ATS platforms to streamline the recruitment workflow. This would enable recruiters to import job descriptions, manage applications, and track candidates directly within their existing HR systems.

- **Graphical Dashboard for HR Analytics:** A data visualization dashboard can be added to provide insights into hiring trends, candidate ranking distributions, and resume match statistics. Using charts and graphs, recruiters can gain a clearer understanding of the recruitment process and make more informed decisions.

By incorporating these enhancements, the system can further improve accuracy, efficiency, and usability, making it a powerful AI-driven recruitment tool.

REFERENCES

a) Scikit-learn Documentation - https://scikit-learn.org
b) NLTK NLP Toolkit - https://www.nltk.org
c) Research papers on TF-IDF & Cosine Similarity in Resume Screening.