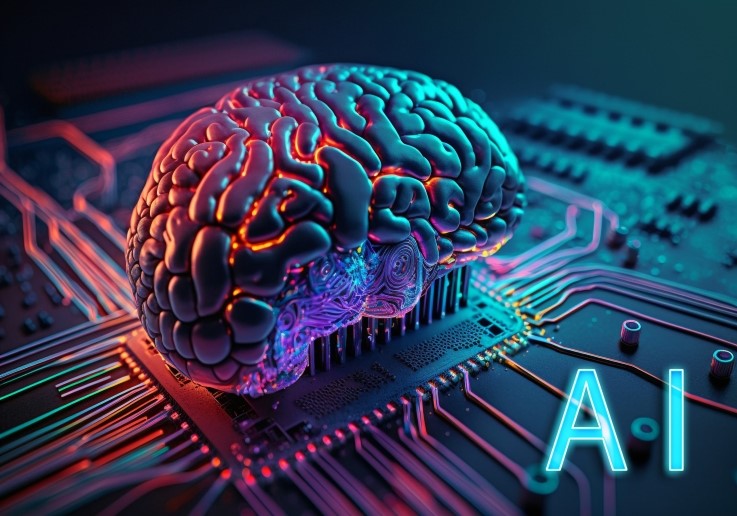
ABSTRACT:

Resume Analyzer

AI PROJECT



The proposed project aims to develop an Analyzer that Analyze Resume using AI and machine learning techniques

MEMBER:

Abdullah Tahir (CS-1004)

# Proposal for AI Project: Resume Analyzer

## Introduction

# The proposed project aims to develop a Resume Analyzer system using AI and machine learning techniques. The backend will be implemented using Python, and the frontend will use React.js. The system will allow users to upload their resumes in various formats (PDF, DOCX, etc.). Once uploaded, the system will process the resume, extracting key information such as name, contact details, skills, experience, and education. It will analyze the content and provide feedback on resume strength, offer suggestions for improvement, and match the profile against relevant job listings, leveraging a pre-trained machine learning model for accurate predictions.

# Features

 **Resume Parsing:** Extract key information from resumes, such as name, contact details, skills, education, and work experience.

 **File Upload:** Upload resumes in various formats (PDF, DOCX, etc.) for analysis.

 **Resume Strength Analysis:** Analyze the content and provide feedback on the strength of the resume.

 **Job Matching:** Match the resume profile with relevant job listings based on skills and experience.

 **API Integration:** Two backend routes for processing:

* Home page for user interaction and resume upload.
* Prediction API to analyze the uploaded resume and generate results.

 **Model Training:** Train the model using a dataset of job listings and resumes for accurate matching and feedback generation.

**Technologies and Libraries import** streamlit as st

import pickle

import re

import nltk

from PyPDF2 import PdfReader

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.neighbors import KNeighborsClassifier

from sklearn.multiclass import OneVsRestClassifier

from sklearn.metrics import accuracy\_scoreFrontend

## Procedure

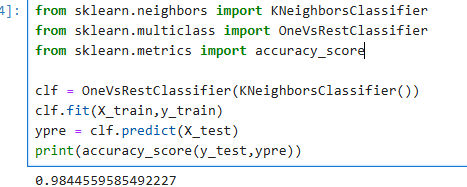
1. **Model Training:**
   * Load a dataset of resumes and job listings.
   * Preprocess the data (normalize text, extract relevant features).
   * Define and train a model (e.g., a text classification or matching model) using TensorFlow or scikit-learn.
   * Evaluate the model on the testing set and save it for deployment.
2. **Integration and Testing:**
   * Connect the frontend to the backend API.
   * Test the resume analysis flow with uploaded resumes and form submissions.
   * Display results, such as resume strength, feedback, and job matches, on the frontend.

**Working Flow:**

1. **Model Training:**
   * Load resume and job listing dataset.
   * Preprocess the data (normalize text, extract features).
   * Train a model on the resume data.
   * Validate on test data.
   * Save the trained model for deployment.
2. **Frontend Interaction:**
   * User uploads a resume or fills out a form with resume details.
   * The input is processed and sent to the backend API.
3. **Backend Processing:**
   * Receive the resume or form data.
   * Preprocess the data (extract features, normalize text).
   * Load the pre-trained model and analyze the resume.
   * Return feedback and job matching results.
4. **Result Display:**
   * Show the analysis results (resume strength, suggestions, job matches) on the React.js interface.

**Accuracy**

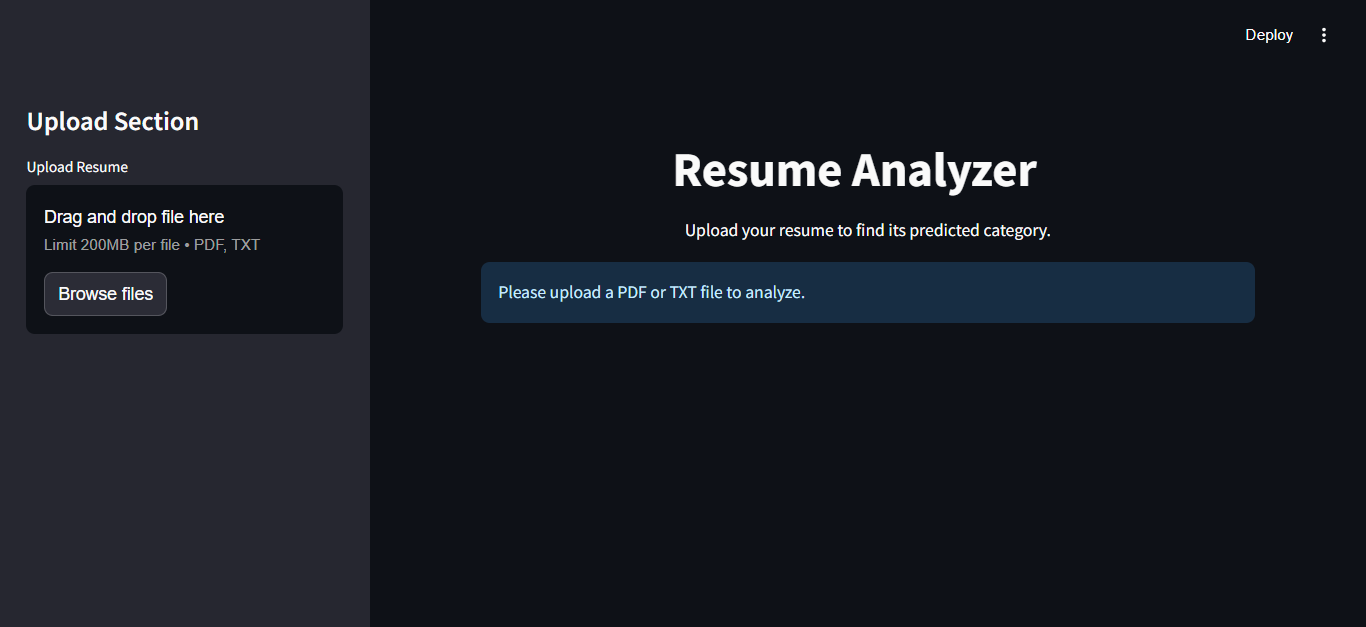
I trained this data on 80 per and the accuracy after testing on 20 per data is



## Conclusion

This project provides an interactive system for resume analysis. The integration of AI with user-friendly frontend and backend technologies ensures a robust and scalable solution. The use of a comprehensive resume and job listing dataset guarantees accurate analysis and reliable job matching, making this system suitable for real-world applications such as job recruiting and career development.

**Final look**

****