Mini Project Report

On

"JOB RECOMMENDATION SYSTEM USING MACHINE LEARNING"

CSE VI Semester Mini Project

Report Session: 2024-25

Submitted to:

Mr MUKESH KUMAR

(Department of CSE)

Submitted By:

NAMAN SINGH RANA

RollNo:2118823

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GRAPHIC ERA HILL UNVERSITY, DEHRADUN

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the project report entitled **"Job Recommendation System Using Machine Learning"** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineering of the Graphic Era (HILL University), Dehradun shall be carried out by the under the mentorship of Mr Mukesh Kumar**, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Hill University),Dehradun.

Name : NAMAN SINGH RANA

Univ. Roll No: 2118823

Table of Contents

Job Recommendation System

Overview

The Python script you provided focuses on analyzing job offers and making recommendations based on their content. By leveraging popular libraries such as NumPy, pandas, NLTK, and scikit-learn, the script processes job offer data from a CSV file named `ALL_Offers.csv`.

Key Components

1. Data Import and Cleaning

- **Data Import**: The script begins by importing job offer data from the specified CSV file into a pandas DataFrame. This step ensures that we have a structured dataset to work with.

- **Text Cleaning Function (`cleaning`)**: To prepare the textual data for analysis, the script defines a custom function called `cleaning`. This function performs several essential tasks:

    - Removes non-alphanumeric characters.
    - Converts text to lowercase.
    - Tokenizes the text (splits it into individual words).
    - Removes common stopwords (such as “the,” “and,” “in,” etc.).
    - Applies stemming (reducing words to their root form).

2. Feature Engineering

- **New Column Creation (`new_col`)**: The script creates a new column called `new_col` by concatenating the cleaned `job_title` and `description` columns. This combined text feature allows us to analyze both job titles and descriptions together.

3. Text Vectorization and Similarity Calculation

- **TF-IDF Vectorization**: The script employs the `TfidfVectorizer` from scikit-learn. This technique converts the text data in `new_col` into a matrix of TF-IDF (Term Frequency-Inverse Document Frequency) features. TF-IDF captures the importance of each word in relation to the entire dataset.

- **Cosine Similarity**: The script calculates the cosine similarity between different job offers based on their TF-IDF vectors. Cosine similarity measures the angle between two vectors and provides a value between -1 and 1. Higher values indicate greater similarity.

4. Recommendation System

- **Recommendation Function**: The script defines a function (let's call it `recommendation`). This function takes a job title as input and returns a list of recommended job titles. Here's how it works:
    - For the given input job title, it computes the cosine similarity scores with all other job titles.
    - It selects the job titles with the highest similarity scores as recommendations.
    - Note: The current implementation returns only the first recommendation due to a limitation in the loop structure. Future improvements should address this.

5. Data Persistence

- **Pickle Serialization**: To save the processed data for later use, the script utilizes the `pickle` module. It serializes and stores the DataFrame and the similarity matrix on disk.

Dependencies

Make sure the following libraries are installed:

- NumPy
- pandas
- NLTK
- scikit-learn
- pickle

Known Issues and Limitations

- The script assumes the presence of specific columns (`job_title` and `description`) in the CSV file. Robust error handling for missing columns should be added

- The recommendation function currently returns only one recommendation. Enhancements could provide a more comprehensive list of similar job titles.

Future Work

- Implement robust error handling to handle missing files, columns, or unexpected data.
- Optimize the recommendation function to return multiple relevant job titles.
- Explore advanced NLP techniques for text cleaning and feature extraction.

Conclusion

This script lays the groundwork for analyzing job offers and suggesting similar positions based on their descriptions. With further refinement, it can become a valuable tool for job seekers and recruiters alike. 🚀