**INSTUTUENT FOR ADVANCED COMPUTING AND**

**SOFTWARE DEVELOPMENT**

AKURDI, PUNE - 411044

Documentation on

**“Resume Screening Using NLP”**

PG-DBDA AUG 2024

Submitted By-

**Group No.14**

**Prathamesh Patil (248529)**

**Rajan Jadhav (248533)**

**Mrs. Priti Take**  **Mr. Rohit Puranik**

Project Guide Centre Coordinator

**DECLARATION**

I, the undersigned hereby declare that the project report titled "Resume Screening Using NLP" written and submitted by me to Institute For Advanced Computing And Software Development Akurdi Pune, in the fulfilment of requirement for the award of degree of Post Graduate Diploma In Big Data Analytics (PG DBDA) under the guidance of Mrs.Priti Takeis my original work I have not copied any code or content from any source without proper attribution, and I have not allowed anyone else to copy my work. The project was completed using Python and MS Excel and libraries.

The project was developed as part of my academic coursework. I also confirm that the project is original, and it has not been submitted previously for any other academic or professional purpose.

Place: Signature:

Date: Name: Prathamesh Patil/Rajan Jadhav

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to Mrs. Priti Take, Project Guide, for providing me with the guidance and support to complete this academic project. Their valuable insights, expertise, and encouragement have been instrumental in the success of this project. I would also like to thank my fellow classmates for their support and cooperation during the project. Their feedback and suggestions were helpful in improving the quality of the project. I would like to extend my gratitude to Mr. Rohit Puranik, Centre Coordinator, for providing me with the necessary resources and facilities to complete this project. Their support has been crucial in the timely completion of this project.

Finally, I would like to thank my family and friends for their constant encouragement and support throughout the project. Their belief in me has been a constant source of motivation and inspiration. Thank you all for your support and guidance in completing this academic project.

**ABSTRACT**

Recruitment is a time-consuming process that requires hiring managers to manually screen a large number of resumes to find the best candidates for job positions. Traditional resume screening methods are inefficient, subjective, and prone to human bias. This project introduces an **AI-powered Resume Screening System** using **Natural Language Processing (NLP)** to automate and optimize the hiring process.

The system takes **resume PDFs and job descriptions as input**, extracts textual information using **text preprocessing techniques** (tokenization, stopword removal, and lemmatization), and converts the text into numerical vectors using **TF-IDF (Term Frequency-Inverse Document Frequency).** The **cosine similarity** algorithm is then applied to measure the relevance of resumes to job descriptions.

The project is implemented using **Python, NLTK, Pandas, and Scikit-Learn**. The system is deployed using **Streamlit**, allowing users to upload resumes and receive an **automated job role prediction** along with the **most relevant job descriptions.** Evaluation metrics such as **accuracy, precision, recall, and F1-score** are used to assess the effectiveness of the LSTM model. Visualizations, including **heatmaps and similarity score distributions**, provide insights into resume-job matching accuracy.

This automated approach significantly reduces manual effort, improves hiring efficiency, and ensures objective candidate selection. This project demonstrates the potential of **NLP and AI** in revolutionizing recruitment, making it faster, smarter, and more reliable.

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **Topic** | **Page No.** |
| **1.** | **Chapter 1 Introduction**  1.1Description  1.2 Objective of the project  1.3 Scope of the project  1.4 Limitation of the project | 1  1  1  1 |
| **2.** | **Chapter 2 Project Description**  2.1 Project work flow diagram  2.2 Data collection  2.3 Studying the data  2.4 Studying the model  2.5 Implementing the model  2.6 Validating the model  2.7 Creating the tool | 2  2  3  3  3  3  3-4 |
| **3.** | **Chapter 3 Model Description**  3.1 Important Terms | 5 |
| **4.** | **Chapter4 Data Flow**  4.1Data flow of project | 7 |
| **5.** | **Chapter5 Tools used in project**  5.1 Pandas:  5.2 Numpy  5.3 Sklearn  5.4 OS Module  5.5 PyPDF: PDF Processing in Python  5.6 NLTK (Natural Language Toolkit) in Python  5.7 Matplotlib  5.8 re Module  5.9 Seaborn: Data Visualization in Python | 8  8  8  9  9  9  9  9  9 |
| **6.** | **Chapter 6 Project Requirements**  6.1 Hardware Requirements  6.2 Software Requirements | 13  13 |
| **7.** | **Chapter 7 Future Scope**  7.1 Future Scope | 14 |
| **8.** | **Chapter 8 Conclusion**  8.1 Conclusion | 15 |
| **9.** | **Chapter 9 References**  9.1 References | 16 |

**Chapter 1**

**Introduction**

**Introduction:**

* 1. **Description:**

The resume screening using NLP project automates the process of shortlisting candidates by analyzing and matching resumes with job descriptions. It extracts and preprocesses text from PDF resumes, converts it into numerical representations using TF-IDF, and applies cosine similarity to find the best-matching resumes for each job role. The system is deployed using Streamlit, allowing users to upload resumes and job descriptions for automated matching. This approach improves efficiency, reduces manual effort, and ensures a more objective hiring process.

* 1. **Objective of project:**

The objective of this project is to develop an automated resume screening system using Natural Language Processing (NLP) techniques. The system extracts text from resumes, processes it using TF-IDF, and applies cosine similarity to match resumes with relevant job descriptions. By automating this process, the system aims to reduce the time and effort required for manual resume screening, minimize human bias, and ensure a more efficient and objective candidate selection process. The goal is to help recruiters quickly identify the most suitable candidates for job positions based on their skills, experience, and qualifications.

* 1. **Scope of the project:**

The project is designed to automate resume shortlisting in the hiring process by analyzing text data from resumes and job descriptions. It allows recruiters to upload PDF resumes and job descriptions, which are then processed and matched using NLP techniques. The system provides a similarity score for each resume, helping recruiters identify the best candidates. The project is applicable across various industries, including IT, finance, healthcare, and marketing.

* 1. **Limitations of the Project:**

Despite its advantages, the project has certain limitations. The system relies on text-based resume analysis, meaning it may not effectively process resumes with complex designs, images, or graphical elements. Additionally, the accuracy of resume-job matching depends on the quality of job descriptions and resumes provided. The TF-IDF method primarily focuses on word frequency rather than semantic meaning, which may affect relevance ranking. The system also does not consider non-textual factors such as soft skills, cultural fit, or personality traits, which are important in the hiring process.

**Chapter 2**

**Project Description**

**Project Description :**

**2.1 Project work flow Diagram:**

**Data Collection**

**Pre-Processing**

**Shortlisted Resume**

**Prediction**

**Text Extraction**

Project Workflow Diagram

**2.2 Data collection:**

Data collection involves gathering resumes in **PDF format** and job descriptions in **CSV format** from various sources. Resumes contain **candidate details, skills, experience, and education**, while job descriptions define the **requirements, qualifications, and responsibilities** of a role. The quality and quantity of the collected data directly impact the performance of the system. Ensuring diverse and well-structured data helps improve the **accuracy of the resume-job matching process**. Additionally, publicly available datasets, company databases, and job portals can be used to enhance the dataset for better training and evaluation.

**2.3 Studying the Data:**

Once the data is collected, it is analyzed to understand its **structure, format, and key features**. Resumes are inspected for **text quality, missing values, and common patterns**, while job descriptions are examined for **keywords and required skills**. Exploratory Data Analysis (EDA) is performed to identify **trends, most frequent words, and data inconsistencies**. Data visualization techniques, such as **word clouds and bar charts**, help in better understanding. This step ensures that the collected data is **clean, well-structured, and relevant**, which is essential for building an accurate resume screening model.

**2.4 Studying the Model:**

Before implementing the model, different **resume screening techniques** are explored. Traditional methods like **TF-IDF and cosine similarity** are studied for text-based matching. Various NLP techniques for **feature extraction and text comparison** are considered to enhance resume-job matching. The importance of preprocessing, **vectorization methods, and text similarity scoring** is analyzed to ensure the best approach is selected. Understanding these models helps in choosing the most **efficient, scalable, and accurate** method for automating resume screening while minimizing errors and biases in candidate selection.

**2.5 Implementing the Model:**

The model is implemented by **preprocessing text**, extracting important features, and applying **TF-IDF vectorization**. Cosine similarity is used to compare resumes with job descriptions and rank them based on relevance. Python libraries like **NLTK, Scikit-Learn, and Pandas** are used for text processing, while **Streamlit** is used to build an interactive tool. The model is optimized by tuning **hyperparameters and improving data quality**. The implementation ensures that the system can **process multiple resumes efficiently** and return the most relevant candidates for a given job description.

**2.6 Validating the Model:**

Once implemented, the model is validated using **accuracy, precision, recall, and F1-score** to measure its effectiveness. A dataset with known resume-job pairs is used for evaluation, and the model’s predictions are compared with expected results. The **similarity score distributions** and **confusion matrix** help assess the model's performance. If inaccuracies are found, adjustments are made to **improve text preprocessing, TF-IDF settings, or similarity thresholds**. The validation step ensures that the system provides **reliable and relevant resume recommendations** before deployment.

**2.7 Creating the Tool:**

The final step involves developing a **user-friendly application** for recruiters. The system is deployed using **Streamlit**, allowing users to **upload resumes and job descriptions** for automated matching. The tool provides **real-time analysis**, displaying the **best-matching resumes for each job** along with similarity scores. The UI is designed for ease of use, with interactive **filters, job role selection, and visualization charts**. Once tested, the tool can be integrated into **company hiring platforms** or used as a standalone application, streamlining the resume screening process and reducing hiring time.

**Chapter 3**

**Model Description**

**3.1 Model Description :**

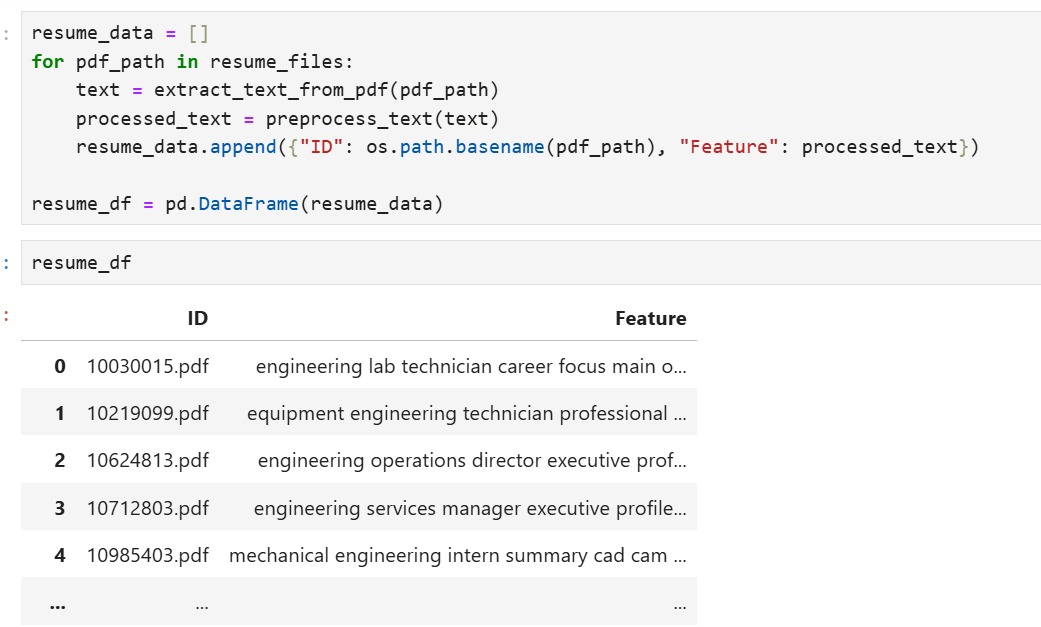
The resume screening system is built using Natural Language Processing (NLP) techniques to automate the shortlisting of candidates based on job descriptions. The model follows a structured pipeline that involves text preprocessing, feature extraction, and similarity computation to match resumes with relevant job roles.

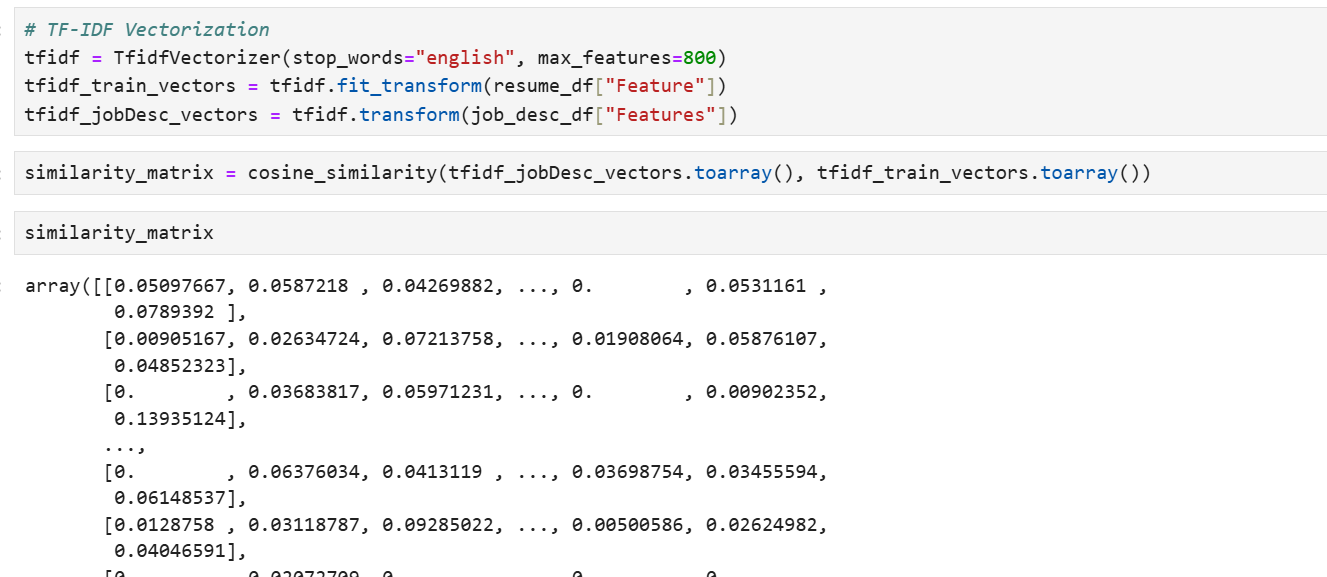
The first step involves extracting text from resumes and job descriptions, which are then preprocessed using techniques such as tokenization, stopword removal, and lemmatization. This ensures that only meaningful words are retained for analysis. After preprocessing, the text is converted into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF). This method assigns weights to words based on their importance, ensuring that commonly used words have lower significance while job-specific terms are given more weight.

Once the resumes and job descriptions are vectorized, cosine similarity is used to compute the relevance of each resume to the given job descriptions. Cosine similarity measures the angle between two document vectors, providing a numerical score that represents how similar the resume is to a job description. The model ranks resumes based on these similarity scores and selects the top matches for each job role.

To enhance efficiency, the system is implemented using Python and utilizes libraries such as NLTK for NLP tasks, Scikit-Learn for TF-IDF and similarity computation, and Pandas for data handling. The final application is built using Streamlit, allowing recruiters to upload resumes and job descriptions, view similarity scores, and shortlist candidates quickly.

The model provides a fast and objective approach to resume screening, eliminating manual effort and bias in candidate selection.

****

****

**Chapter 4**

**Data Flow**

**4.Data Flow of project :**

The system ranks resumes and displays the best matches with similarity scores.

Cosine similarity compares resume vectors with job description vectors.

TF-IDF converts text into numerical representations.

The system extracts text from resumes (PDFs) and preprocesses the text.

User uploads resumes & job descriptions.

**Chapter 5**

**Tools Used In Project**

**Tools used in project:**

Following libraries used in the project:

**5.1 Pandas:**

Pandas is a popular open-source library for data manipulation and analysis in Python. It was created by Wes McKinney in 2008 and has since become a widely used tool in data science and analytics.

Pandas is built on top of two other Python libraries, NumPy and Matplotlib. NumPy provides efficient numerical operations on arrays of data, while Matplotlib provides powerful visualization tools. Pandas combines these libraries and adds its own data structures and tools to create a comprehensive data analysis package.

One of the key data structures in Pandas is the DataFrame. A DataFrame is a two-dimensional labeled data structure with columns of potentially different types. It can be thought of as a spreadsheet or a SQL table. Each column in a DataFrame is a Series, which is a one-dimensional labeled array that can hold any data type.

Pandas provides a wide range of tools for data cleaning, manipulation, and analysis. Some of the key functions include:

**5.2 Numpy:**

NumPy (Numerical Python) is a fundamental library for scientific computing in Python. It provides support for large, multi-dimensional arrays and matrices, along with a vast collection of mathematical functions to operate on these arrays efficiently

**5.3 Sklearn:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy. SciPy and Matplotlib.

Scikit-learn (sklearn) is a popular Python library for machine learning. It provides a wide range of functions for model evaluation.

**5.4 OS Module:**

The os module in Python provides functions to interact with the operating system. It allows developers to perform operations like file handling, directory management, environment variable access, and process control. The os module is especially useful for automation, file organization, and system-level scripting in various applications, including resume screening using NLP, where file handling and directory management are required.

**5.5 PyPDF: PDF Processing in Python:**

PyPDF (previously known as PyPDF2) is a Python library used for reading, extracting, merging, splitting, and modifying PDF files. It is widely used in resume screening projects for extracting text from PDF resumes.

**5.6 NLTK (Natural Language Toolkit) in Python**

NLTK is a comprehensive Python library used for working with human language data (text), often referred to as Natural Language Processing (NLP). It provides tools to process, analyze, and understand text, making it essential for text analysis in resume screening projects.

**5.7 Matplotlib:**

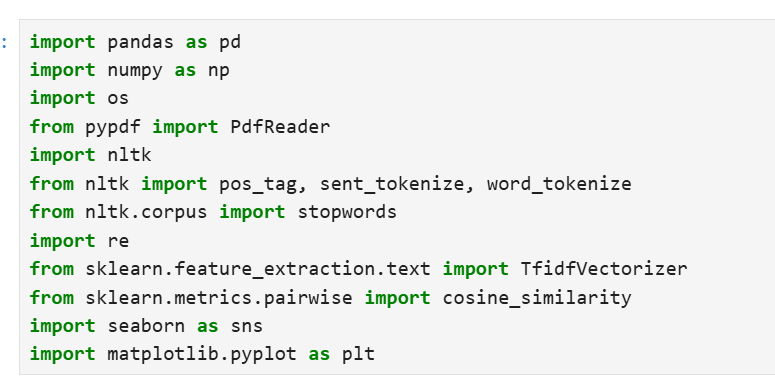
Matplotlib is a popular data visualization library for Python. It provides a wide range of functions for creating static, animated, and interactive visualizations in Python. Matplotlib is widely used in data science, scientific computing, and engineering applications.

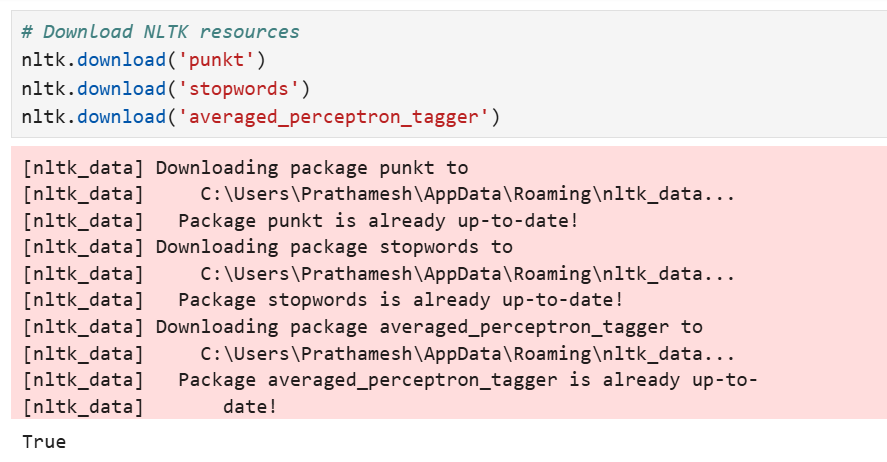
**5.8 re Module:**

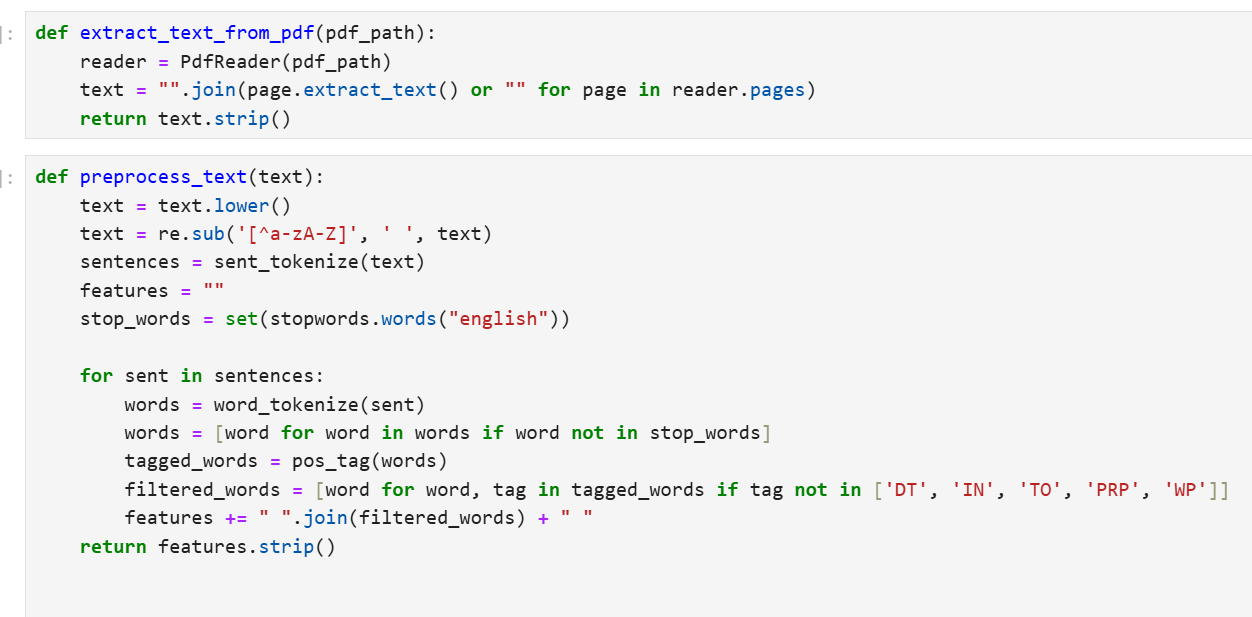
The re module in Python provides support for working with regular expressions (regex), which are sequences of characters that define search patterns. Regular expressions are used for text search, manipulation, and pattern matching, making them extremely useful in resume screening for extracting specific information from resume text (e.g., phone numbers, email addresses, or skills).

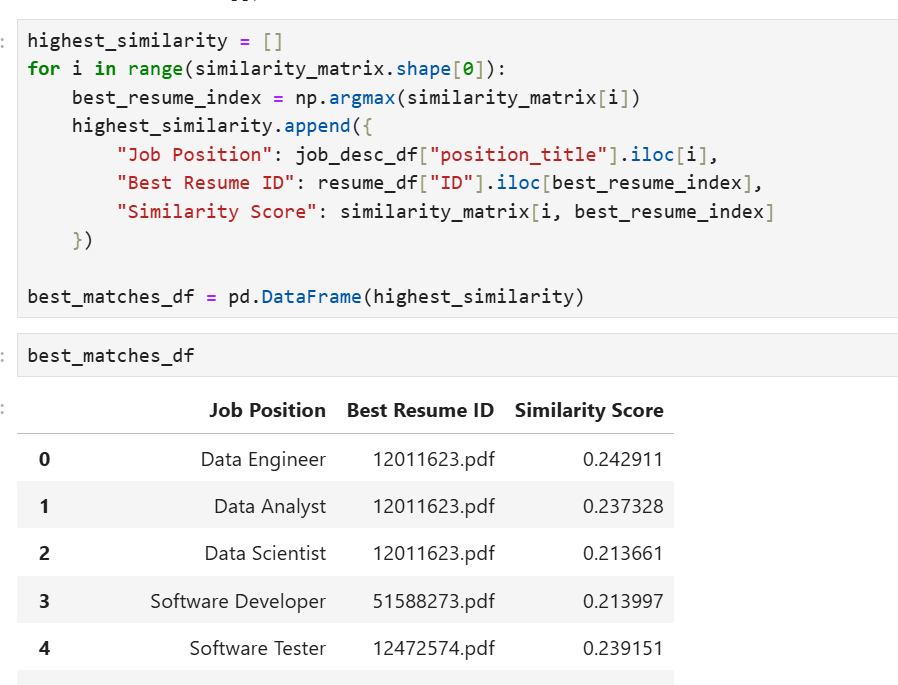
**5.9 Seaborn: Data Visualization in Python**

Seaborn is a powerful Python data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics. Seaborn is particularly useful for visualizing complex datasets and relationships, such as those found in resume screening projects, where visual representations of features like experience, skill distribution, and performance can be insightful.

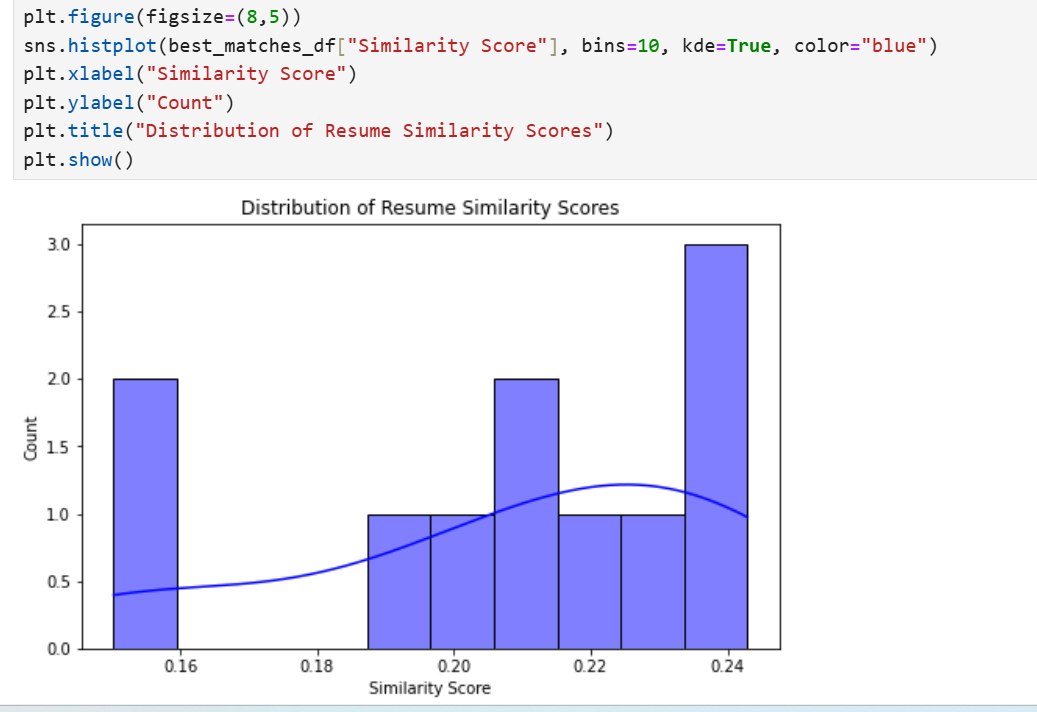
****

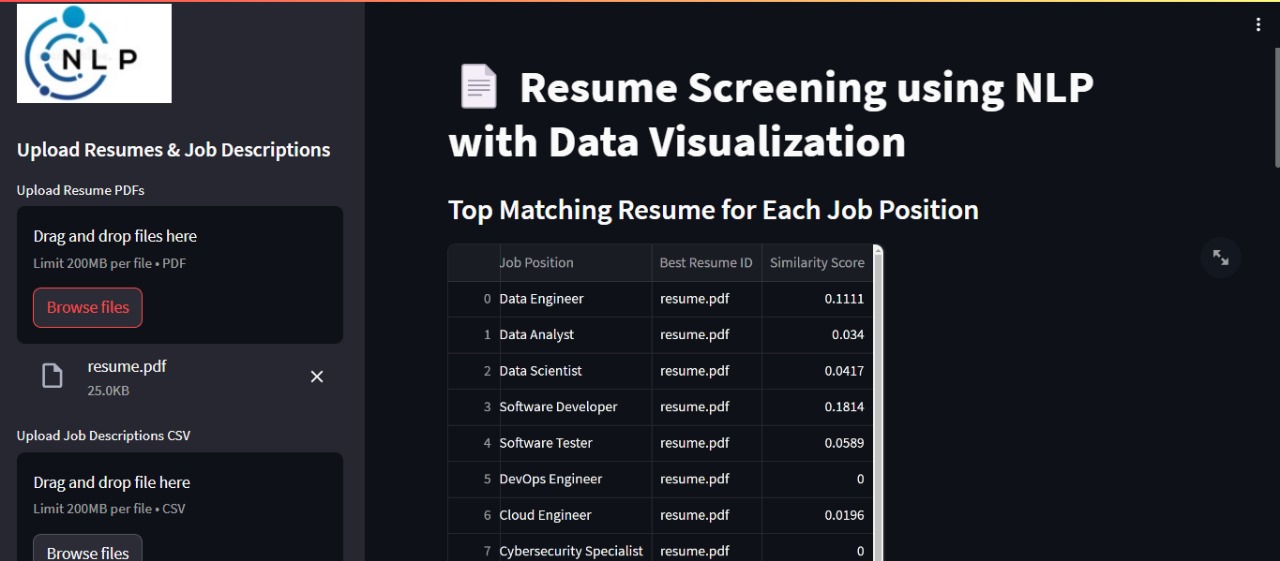












**Chapter 6**

**Project Requirements**

**Project Requirements** :

**6.1 Hardware Requirement**

1. 500 GB hard drive (Minimum requirement)

2. 8 GB RAM (Minimum requirement)

3. PC x64-bit CPU

4. Internet with minimum speed 5 Mbps

**6.2 Software Requirement**

1. Operating System (Windows/Mac/Linux)

2. Python with version 3.9 or above

3. Python Libraries

Pandas 1.5.3

Numpy

NLTK

Re

Os

seaborn

Matplotlib 3.7.0

Sklearn 0.0.post1

4. For coding VS code/Anaconda/spider

5. Web browser like Google Crome / Mozilla Firefox

6. Microsoft Excel (Ms office 2016 or above)

**Chapter 7**

**Future Scope**

**7.1 Future Scope**

The future scope of this project could involve several improvements and expansions, including:

The proposed system can be improved in a number of ways. For example, the system can be trained on a larger dataset of resumes to improve its performance. Additionally, the system can be extended to predict other information from resumes, such as salary expectations and years of experience. Explore the use of different machine learning algorithms and hyperparameters to improve the performance of the system. Train the system on a larger and more diverse dataset of resumes to improve its generalizability. Extend the system to predict other information from resumes, such as salary expectations and years of experience. Develop a user interface for the system to make it more accessible to recruiters and other practitioners.

The future scope of Resume Screening using NLP is vast, as advancements in Natural Language Processing (NLP), machine learning, and automation will continue to improve the efficiency and accuracy of the process.

**Chapter 8**

**Conclusion**

**8.1 Conclusion**

This paper proposed a resume screening system based on NLP and Python. This suggests that NLP can be used to effectively screen resumes for job openings. The proposed system can be used by recruiters to save time and resources, and to be more confident that they are identifying the best candidates for the job. Implications for practice -The proposed system has a number of implications for practice. First, it can help recruiters to save time and resources by automating the resume screening process. Second, it can help to reduce bias in resume screening by evaluating resumes based on their qualifications, rather than on irrelevant factors such as the candidate's name, gender, or age. Third, it can help recruiters to identify more qualified candidates by considering a wider range of factors, such as the candidate's skills and experience, as well as their education.

Overall, the proposed system is a promising new tool for resume screening. It has the potential to help recruiters to be more efficient and effective in their search for the best candidates for the job.

**Chapter 9**

**References**

**9.1 References :**

[1] Swapnil Sonar, Resume Parsing with Named Entity Clustering Algorithm, IEEE Research, may2012, http://www.slideshare.net/swapnilmsonar/resu meparsing-with-named-entity-clustering algorithm

[2] Machine Learning Methods for Solving Complex Ranking and Sorting Issues in HumanResourcing”, Institute of Electrical and Electronics Engineers (IEEE) 2017.

[3] Intelligent Hiringwith Resume Parserand Ranking us ing Natural Language Processing and Machine Learning”, International Journal of Innovative Research in Computer and Communication Engineering(IJIRCCE)

[4] Wei, K., Huang, J., Fu, S., 2007. A survey of e commerce recommender systems, in: 2007 international conference on service systems and service management, IEEE.

[5] Paparrizos, I., Cambazoglu, B.B., Gionis, A., 2011. Machine learned job recommendation, in: Proceedings of the fifth ACM Conference on resume scanning.

[6] Ramos, J., et al., 2003. Using tf-idf to determine word relevance in document queries, in: Proceedings of the first instructional conference on resume screening.

[7] Resnick, P., Varian, H.R., 1997. Recommender systems. Communications of the ACM 40, 56– 59.

[8] Rish, I., et al., 2001. An empirical study of the naïve bayes classifier, in: IJCAI 2001 workshop on empirical methods in artificial intelligence .

[9] Deep Learning-Based Resume Screening Model for Efficient Candidate Selection Ali, Awais, et al. Information Processing & Management 60.2 (2023): 102954.

[10] Resume Screening with Transformer-Based Models Kumar, Rahul, et al. Applied Intelligence 53.2 (2023): 1049-1062.