

SKILL PILOT: ELEVATING JOB READINESS THROUGH ADVANCED ANALYTICS

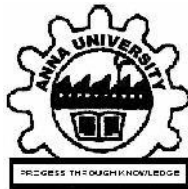
A MINI PROJECT REPORT

Submitted by

**SANDHYA J (221801044)
TANUSHRI G V S (221801055)**

in partial fulfillment for the award of the degree of

**BACHELOR OF TECHNOLOGY
IN
ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**



**RAJALAKSHMI ENGINEERING COLLEGE
DEPARTMENT OF ARTIFICIAL INTELLIGENCE
AND DATA SCIENCE**

ANNA UNIVERSITY, CHENNAI

NOV 2024

ANNA UNIVERSITY, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Report titled “**SKILL PILOT: ELEVATING JOB READINESS THROUGH ADVANCED ANALYTICS**” is the bonafide work of **SANDHYA J (221801044), TANUSHRI G V S (221801055)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

Dr. J.M. Gnanasekar
Professor and Head
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Mr. Thiyagarajan G
Assistant Professor
Department of Artificial Intelligence
and Data Science
Rajalakshmi Engineering College
Chennai – 602 105

Submitted for the project viva-voce examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavor to put forth this report. Our sincere thanks to our Chairman Mr. S. MEGANATHAN, B.E, F.I.E., our Vice Chairman Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S., and our respected Chairperson Dr. (Mrs.) THANGAM MEGANATHAN, Ph.D., for providing us with the requisite infrastructure and sincere endeavoring in educating us in their premier institution.

Our sincere thanks to Dr. S.N. MURUGESAN, M.E., Ph.D., our beloved Principal for his kind support and facilities provided to complete our work in time. We express our sincere thanks to Dr. J.M. GNANASEKAR., M.E., Ph.D., Professor and Head of the Department of Artificial Intelligence and Data Science for his guidance and encouragement throughout the project work. We are glad to express our sincere thanks and regards to our supervisor Mr. Thiyagarajan G., BE., ME., (PhD)., Assistant Professor, Department of Artificial Intelligence and Data Science and coordinator, Dr. P. INDIRA PRIYA., M.E., Ph.D., Professor, Department of Artificial Intelligence and Data Science, Rajalakshmi Engineering College for their valuable guidance throughout the course of the project.

Finally, we express our thanks for all teaching, non-teaching, faculty and our parents for helping us with the necessary guidance during the time of our project.

ABSTRACT

In today's competitive job market, ensuring that one's skills align with job role requirements is crucial for career advancement. "Skill Pilot" addresses this challenge by offering an automated solution that analyzes resumes, identifies missing skills relative to a selected job role, and recommends relevant job roles based on the user's current skill set. The project leverages natural language processing (NLP) techniques, including text extraction and tokenization using libraries like PyMuPDF and spaCy. Additionally, it employs machine learning models, such as TF-IDF vectorization for feature extraction and Random Forest for job role prediction. Cosine Similarity is used to determine the closeness between user and job role skills. The system presents results through intuitive visualizations, providing users with actionable insights to improve their resumes and enhance job readiness. By automating skill analysis and career path suggestions, "Skill Pilot" empowers job seekers to optimize their job search and supports employers in streamlining candidate selection. Future enhancements may include personalized learning paths to help users acquire the necessary skills for their desired career opportunities.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	I
	LIST OF FIGURES	IV
1	INTRODUCTION	
	1.1 GENERAL	1
	1.2 NEED FOR THE STUDY	1
	1.3 OVERVIEW OF THE PROJECT	2
	1.4 OBJECTIVES OF THE STUDY	2
2	REVIEWS OF LITERATURE	
	2.1 INTRODUCTION	3
	2.2 FRAMEWORK OF LITERATURE REVIEW	3
3	SYSTEM OVERVIEW	5
	3.1 EXISTING SYSTEM	5
	3.2 PROPOSED SYSTEM	6
	3.3 FEASIBILITY STUDY	7
4	SYSTEM REQUIREMENTS	9
	4.1 HARDWARE REQUIREMENTS	9
	4.2 SOFTWARE REQUIREMENTS	9
5	SYSTEM DESIGN	10
	5.1 SYSTEM ARCHITECTURE	10
	5.2 MODULE DESCRIPTION	11
	5.2.1 SECTION 1	11
	5.2.2. SECTION 2	14

6	RESULT AND DISCUSSION	17
7	CONCLUSION AND FUTURE ENHANCEMENT	18
	7.1 CONCLUSION	18
	7.2 FUTURE ENHANCEMENT	18
	APPENDIX	
	A1.1 SAMPLE CODE	20
	A1.2 SCREENSHORTS	26
	REFERENCES	28

LIST OF FIGURES

Figure No	Figure Name	Page No
1.	System Architecture	11
2.	DFD for skill extraction	12
3.	DFD for Job role processing	12
4.	DFD for Model training and prediction	13
5.	DFD for Skill matching and Visualization	14
6.	DFD for Vectorization	14
7.	DFD for Resume skill vectorization	15
8.	DFD for Cosine similarity	16
9.	DFD for Ranking and Visualization	16
10.	SkillPilot Welcome Page	26
11.	Login and Signup Page	26
12.	SkillPilot Home Page	26
13.	SkillPilot Sample Input	27
14.	Skill Recommender	27
15.	Job Recommender	27

CHAPTER I

INTRODUCTION

1.1 GENERAL

The landscape of job recruitment and skill assessment is rapidly evolving, driven by technological advancements and changing industry demands. In this dynamic environment, job seekers face the challenge of aligning their skills with specific job roles to maximize employability. A well-crafted resume that effectively highlights relevant skills is crucial for catching the attention of hiring managers and securing desirable positions. However, many individuals struggle to identify and present their abilities in a way that meets employers' expectations. This project, "SkillPilot," addresses these challenges by leveraging advanced analytics and machine learning techniques to automate the skill analysis and job role recommendation process.

The increasing complexity of job roles and the constant evolution of required skill sets necessitate a systematic approach to resume analysis. Traditional methods of manually reviewing resumes are often time-consuming, prone to human error, and inefficient. As the job market becomes more competitive, the demand for automated tools that can objectively assess and improve resumes is growing. "Skill Pilot" provides a solution that combines natural language processing (NLP) and machine learning to offer precise and data-driven feedback. This project serves as a bridge between job seekers and employers, enhancing the efficiency of talent acquisition and empowering individuals to develop skills relevant to their career aspirations.

1.2 NEED FOR THE STUDY

The need for this study arises from the significant gap between the skills presented on a resume and the actual requirements of job roles in various industries. Many job seekers are unaware of the specific skills employers look for, which leads to missed opportunities despite having the potential to succeed. Furthermore, companies often struggle with matching candidates to roles efficiently, resulting in a lengthy hiring process. By automating the process of skill analysis and job role

recommendation, this project aims to simplify job applications and assist both job seekers and recruiters. The study is crucial in today's workforce, where skill gaps are a major barrier to employment and organizational growth.

1.3 OVERVIEW OF THE PROJECT

The "Skill Pilot" project is structured around a modular approach to ensure efficiency and effectiveness in skill analysis and job role recommendation. The project consists of several key components: text extraction, skill matching, and job role suggestion. It begins with extracting text from resumes using libraries like PyMuPDF and python-docx. Skills are then identified using spaCy and compared with job role requirements vectorized through TF-IDF. The system employs machine learning models like Random Forest for prediction and Cosine Similarity for ranking job roles. The results are presented through visualizations such as pie charts and bar graphs to make the feedback intuitive. This comprehensive approach also facilitates career growth by highlighting areas for skill improvement.

1.4 OBJECTIVES OF THE STUDY

- 1. Skill Matching and Gap Analysis:** To suggest suitable job roles for users based on their current skills, helping them explore career paths aligned with their expertise.
- 2. Job Role Recommendation:** To suggest suitable job roles for users based on their current skills, helping them explore career paths aligned with their expertise.
- 3. Data-Driven Feedback:** To provide actionable insights and visual feedback through charts and graphs, enabling users to improve their resumes and better prepare for job applications.
- 4. Efficiency and Accuracy:** To implement machine learning techniques that ensure high accuracy in skill analysis and streamline the hiring process for recruiters.
- 5. User Empowerment:** To empower job seekers by guiding them in skill development and optimizing their job search strategies.

CHAPTER II

REVIEW OF LITERATURE

2.1 INTRODUCTION

Understanding the complexities of resume analysis and job role recommendation requires exploring existing research and methodologies in the fields of human resource management, natural language processing (NLP), and machine learning. This literature review examines the contributions and limitations of previous studies, providing a foundation for the development of the “Skill Pilot” system. By leveraging advancements in NLP and predictive analytics, the proposed project aims to improve upon traditional methods and offer an efficient, automated solution for skill matching and career guidance.

The rapid advancement of technology has significantly transformed the recruitment and job application process. Traditional methods of resume review are becoming increasingly inefficient, especially with the large volume of applications many companies receive. Researchers and organizations have invested considerable effort into developing automated systems that can streamline this process. However, many existing systems fall short in terms of accuracy and adaptability. They often struggle with understanding the nuances of unstructured text in resumes and making meaningful connections between skills and job requirements. This section explores the various techniques and frameworks that have been applied to resume analysis and skill matching, highlighting the gaps that “SkillPilot” seeks to address.

2.2 FRAMEWORK OF LITERATURE REVIEW

This review focuses on two main areas: existing systems for job profile analysis and the use of machine learning models in document processing and skill extraction.

1. Existing Systems for Job Profile Analysis: Various studies have explored the use of clustering techniques for profiling job roles. For example, hierarchical clustering methods have been used to categorize IT job profiles based on skill requirements extracted from job postings. These systems are effective in providing a

high-level view of job market trends but lack the granularity needed to personalize skill recommendations. Additionally, hierarchical clustering models often require manual tuning to achieve accurate clustering, which can be time-consuming and prone to error. Average-linkage clustering is a popular method, but it has limitations in distinguishing job roles with overlapping responsibilities, which poses challenges in accurately matching candidates to positions.

2. Artificial Neural Networks (ANNs) for Document Analysis: The use of artificial neural networks in document processing has shown promise in areas like character recognition, layout analysis, and document segmentation. Notable research has focused on developing neural network models for tasks such as character segmentation and word recognition. These models, however, have limitations when dealing with unstructured data found in resumes. The generalization capability of neural networks is often hindered when trained on artificially generated datasets, as they may not perform well on real-world data. Moreover, despite the advancements in deep learning, there is still a need for hybrid models that can combine NLP techniques with machine learning to improve performance in text-heavy domains like resume analysis.

3. Limitations of Existing Techniques: Despite the progress made, there are several drawbacks to current methods. For instance, traditional clustering and neural network models often require large amounts of labeled training data, which is not always readily available for resume analysis. Additionally, the models may not account for the contextual meaning of skills, leading to inaccuracies in matching job roles with candidate profiles. The need for models that can accurately extract and interpret skills from resumes while providing meaningful recommendations remains a significant research challenge. By addressing these limitations, “Skill Pilot” leverages a comprehensive approach that integrates NLP and machine learning to provide accurate, data-driven recommendations. This project seeks to fill the gaps identified in the literature, offering a more effective and user-friendly solution for job seekers and employers alike.

CHAPTER III

SYSTEM OVERVIEW

3.1 EXISTING SYSTEM

The existing systems for resume analysis and job matching typically rely on keyword-based approaches or basic clustering techniques. Many online platforms use simple keyword matching to compare resumes with job descriptions. These systems have several drawbacks, such as:

- **Inaccurate Skill Matching:** Keyword-based systems often fail to account for the contextual meaning of words. For example, a software engineer's resume might list “Python” as a skill, but the system might inaccurately match it with a data analysis role without understanding the context.
- **Lack of Personalization:** Current solutions are often generic and do not provide personalized feedback to job seekers. They lack the ability to offer specific recommendations for improving skills to meet job requirements.
- **Manual Effort and Inefficiency:** Many organizations still rely on human resource professionals to manually screen resumes, which is time-consuming and prone to bias. Even with automated tools, there is a significant gap in efficiently processing unstructured resume data.
- **Inadequate Use of Machine Learning:** Although machine learning models are widely used in data processing, few systems employ advanced algorithms for skill extraction and recommendation. Neural networks, for example, have been used in character recognition but are underutilized in holistic resume analysis and matching.

The limitations of these existing systems highlight the need for a more sophisticated, automated approach that goes beyond basic keyword matching and offers meaningful, data-driven insights.

3.2 PROPOSED SYSTEM

The proposed “SkillPilot” system addresses the challenges present in existing solutions by using a combination of natural language processing (NLP) and machine learning. It is designed to accurately analyze resumes, identify skill gaps, and recommend suitable job roles, thereby enhancing the job application process. Key features of the proposed system include:

1. Advanced Skill Extraction: Utilizing libraries like PyMuPDF for extracting text from PDF resumes and spaCy for NLP processing, the system identifies relevant skills from unstructured text data. It tokenizes, lemmatizes, and cleans the text to prepare it for comparison with job role requirements.

2. Job Role Preprocessing: The system uses a predefined dataset of job roles, which includes technical and soft skills required for each position. TF-IDF (Term Frequency-Inverse Document Frequency) is applied to convert these skills into numerical vectors, making them suitable for machine learning models.

3. Machine Learning Models: A Random Forest classifier is used to predict job roles based on the extracted skills, and Cosine Similarity is employed to measure the closeness between the user's skills and job role requirements. This ensures a more accurate and context-aware matching process.

4. Skill Gap Analysis and Visualization: The system provides detailed feedback on matched and missing skills. Visualizations like pie charts and bar graphs help users understand their skill profile and identify areas for improvement.

5. Job Role Recommendations: The top matching job roles are ranked and presented to the user, offering clear guidance on potential career paths. This feature not only helps job seekers optimize their resumes but also enables them to focus on relevant skill development.

The "Skill Pilot" system is designed to be user-friendly, efficient, and scalable, offering significant improvements over traditional methods.

3.3 FEASIBILITY STUDY

The feasibility study assesses the practicality of developing and deploying the “SkillPilot” system. The study examines three main aspects: technical, economic, and operational feasibility.

1. Technical Feasibility:

- The project uses well-established libraries and frameworks, such as PyMuPDF for text extraction, spaCy for NLP, and scikit-learn for machine learning. These tools are efficient and reliable for handling the data processing requirements of the system.
- The algorithms employed, like TF-IDF and Random Forest, are proven techniques that can handle large datasets and provide accurate results. Additionally, the use of Cosine Similarity for ranking job roles ensures efficient computation.
- The system can be deployed on cloud platforms, making it scalable and accessible to a wide audience.

2. Economic Feasibility:

- The project requires minimal upfront investment, primarily for cloud hosting and data storage. The use of open-source libraries and frameworks reduces the overall cost of development.
- The potential market for a resume analysis and job recommendation tool is vast, as both job seekers and employers can benefit from its features. This makes the project economically viable, with opportunities for monetization through premium features or partnerships with recruitment firms.

3. Operational Feasibility:

- The user interface is designed to be intuitive, ensuring ease of use for job seekers. Users can upload resumes and receive feedback in a seamless manner.
- The system provides clear and actionable insights, making it highly beneficial for users looking to improve their job readiness. The automated nature of the tool also reduces the burden on human resource professionals.
- The project is designed to handle a large number of user queries efficiently, ensuring smooth operation even under high traffic.

The feasibility analysis confirms that the “SkillPilot” system is both practical and beneficial, with the potential to revolutionize the way job seekers approach their career development and how employers streamline the hiring process.

CHAPTER IV

SYSTEM REQUIREMENTS

4.1 HARDWARE REQUIREMENTS

1. Development Machine:

- Processor: Intel Core i5 or equivalent
- RAM: 8 GB or more
- Storage: 256 GB SSD or higher

2. User Devices:

- Processor: Modern CPU for accessing the application
- RAM: 4 GB minimum
- Internet Connection: Stable internet for using the web-based application

4.2 SOFTWARE REQUIREMENTS

1. Operating System:

- Windows 10, macOS, or Linux for development

2. Programming Language:

- Python: Version 3.8 or later

3. Libraries and Frameworks:

- Natural Language Processing: spaCy
- Machine Learning: scikit-learn
- Data Handling: Pandas, NumPy
- PDF Handling: PyMuPDF (fitz)
- DOCX Handling: python-docx
- Visualization: Matplotlib
- Web Framework: Flask

4. Development Tools:

- IDE: Google colab, Visual Studio Code
- Package Manager: pip for Python package installation

CHAPTER V

SYSTEM DESIGN

5.1 SYSTEM ARCHITECTURE

1. Input Data:

The system takes in three main inputs:

- Resumes (PDFs or database).
- Job Roles (from a file like Excel).
- Skills required for those job roles.

2. Data Preprocessing:

- Extracts text from the resume and job role files using libraries like PyMuPDF (for PDFs) and python-docx (for Word docs).
- This makes the content ready for analysis.

3. Text Processing & Matching:

- Tokenization & Vectorization*: Uses spaCy for tokenizing (breaking down the text) and TF-IDF for converting text into numerical vectors, capturing word importance.
- Cosine Similarity: Calculates similarity between the resume and job role descriptions to see how well they match based on skills.

4. Ranking & Visualization:

- Ranking: Ranks job roles based on how closely they match the applicant's skills.
- Visualization: Displays the ranking and matching results using charts for better insights.

5. Machine Learning (Skill Matching):

- Uses a Random Forest model to further match skills, training on past data to improve prediction accuracy.
- Shows matched skills and provides accuracy of the match.

This model results in a ranked list of job roles based on the applicant's skill fit, with visualizations to make the results clear and actionable.

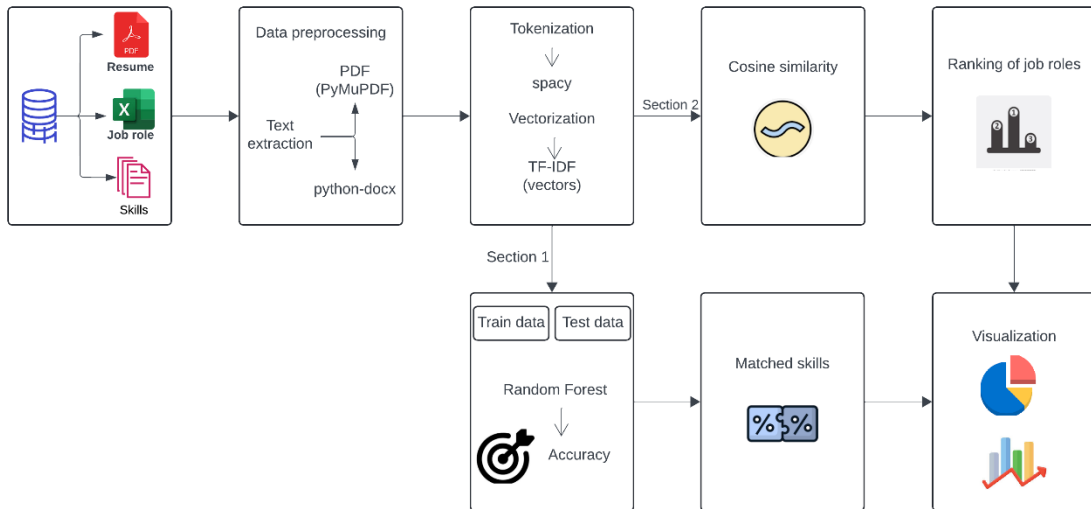


Figure 1: System Architecture

5.2 MODULE DESCRIPTION

5.2.1 SECTION 1

MODULE 1: SKILL EXTRACTION

This module focuses on the automated extraction of relevant skills from user-submitted resumes and a predefined list of job skills. The process begins when a user uploads their resume in PDF format. The text content is extracted using *PyMuPDF*, resulting in readable resume text. Meanwhile, an admin uploads a DOCX file containing job-specific skill requirements. This text is processed using *python-docx* to extract skill keywords. Both extracted texts are then processed using *spaCy* for tokenization, which segments and identifies individual skill-related terms. The matched skills from the resume and job requirements are compared to identify overlaps, with the results being stored in a data store. This data informs the skill gaps between the user's profile and job expectations. The diagram visually represents this flow, highlighting inputs, text extraction methods, tokenization, and the comparison process, ensuring structured and efficient skill extraction.

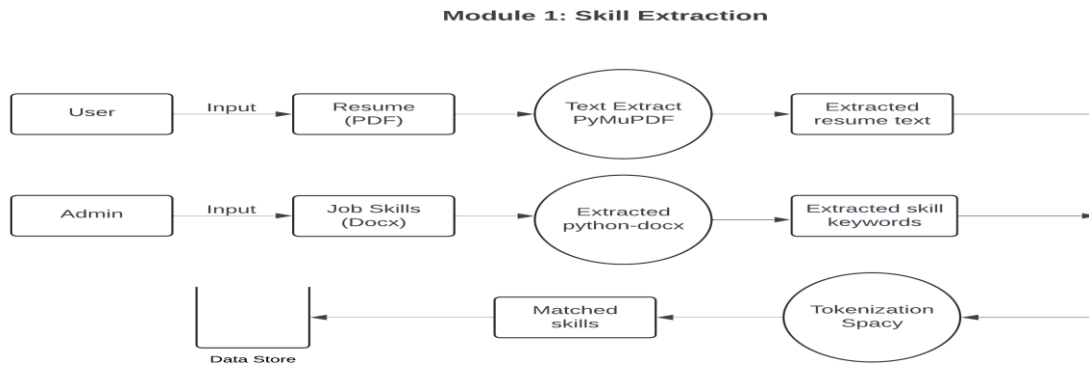


Figure 2: DFD for skill extraction

MODULE 2: JOB ROLE PREPROCESSING

The system prepares job roles for skill matching. First, both users and administrators input job roles and datasets. The system then extracts and cleans job-specific skills from the provided dataset to ensure consistency and relevance. This cleaned data serves as a base for identifying necessary technical skills for each job role. Using TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, the system generates a numerical representation of the skills, known as skill vectors. These vectors capture the importance of each skill in the context of the job role, providing a structured dataset ready for machine learning tasks. Once created, these skill vectors are stored in the Skill Data Store, facilitating easy retrieval for further analysis and skill-matching processes.

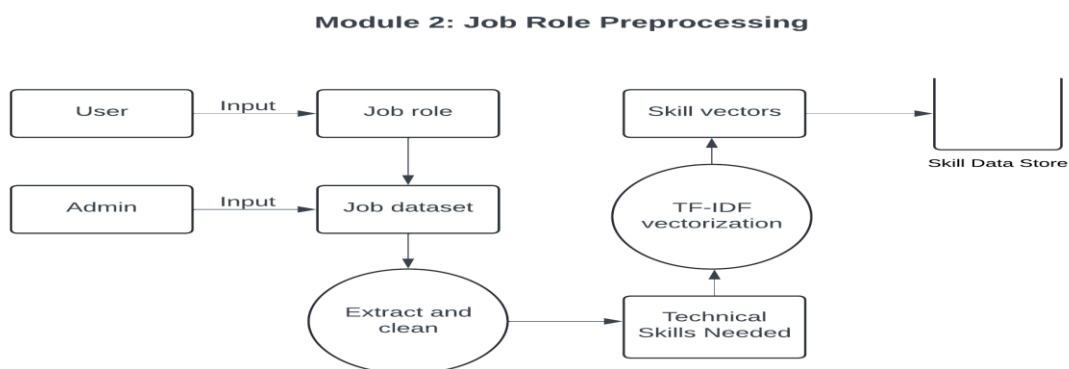


Figure 3: DFD for Job role processing

MODULE 3: MODEL TRAINING AND PREDICTION

Module 3 focuses on training a predictive model to classify job roles based on skills. Using the Skill Data Store and job dataset labels, the data is split into training and testing subsets with a Train_Test_split function. The training data is fed into a Random Forest Classifier, which learns to classify job roles based on skill vectors. This trained model is then tested with the test data, generating predictions on job role classifications. The accuracy of these predictions is evaluated to measure the model's performance, allowing for adjustments and optimization of the classifier. This module essentially builds the machine learning foundation for skill-job role matching.

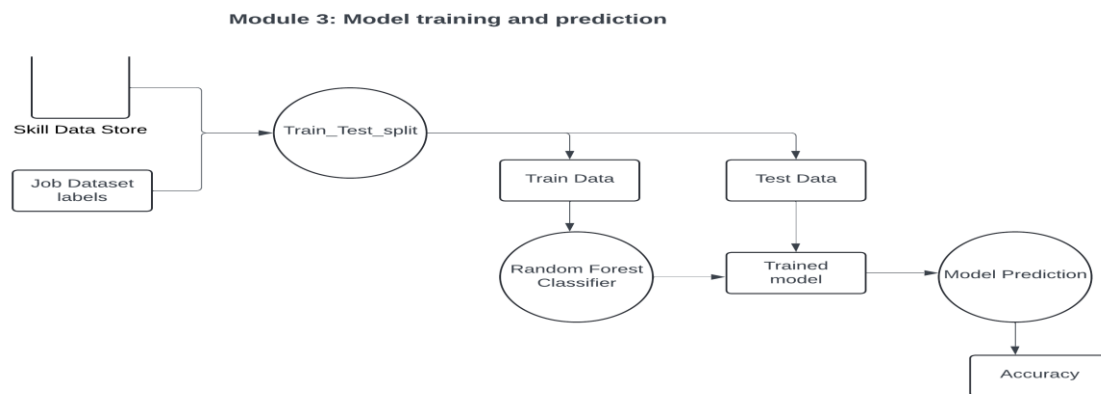


Figure 4: DFD for Model training and prediction

MODULE 4: SKILL MATCHING AND VISUALIZATION

In Module 4, the system identifies matched and missing skills between a job role's requirements and a user's qualifications. It retrieves technical skills from the Data Store and cross-references them with required skills for the selected role, generating a list of matched and missing skills. These results are visualized using a pie chart created with Matplotlib, giving users a clear view of their skill alignment with the job requirements. This visual feedback helps users understand which skills they possess and which they need to improve for their target job role.

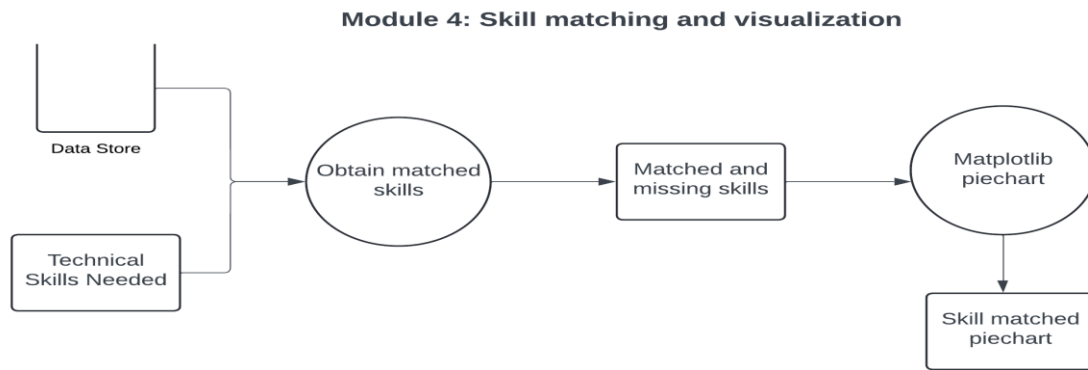


Figure 5: DFD for Skill matching and Visualization

5.2.2 SECTION 2

MODULE 1: VECTORIZATION

The first module focuses on converting job roles into numerical vectors, a process known as vectorization. The input is a collection of job roles, each associated with specific technical skills required for that role. Initially, the job roles are parsed to extract these skills, generating a skill-based representation for each role. The extracted skills then undergo TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, a technique used to represent textual data as numerical values based on the importance of each term. In this case, TF-IDF helps in determining how relevant each skill is to the specific job role by considering its frequency across all roles. The output of this module is a set of job role vectors, which quantitatively represent the technical skills needed for each role, enabling further comparison with other data, such as resume skill sets.

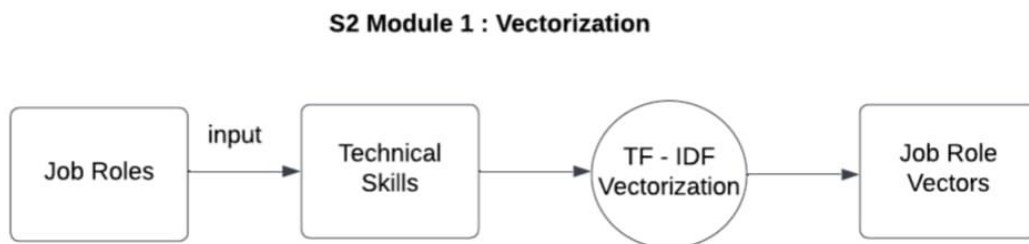


Figure 6: DFD for Vectorization

MODULE 2: RESUME SKILL VECTORIZATION

In the second module, the process of vectorization is applied to resumes, specifically to the skillsets mentioned within them. The input to this module is a set of resume skills, which are parsed and preprocessed to ensure that each skill is represented consistently. These skills are then converted into vectors using TF-IDF vectorization, similar to the approach used in Module 1. This process assigns numerical values to each skill based on its relevance, making it possible to quantify the skillsets in resumes. The outcome is a collection of resume skill vectors that effectively represent the candidate's expertise in various areas. These vectors are crucial for the subsequent module, where they will be compared with job role vectors to assess the alignment between a candidate's skills and the requirements of a given role.

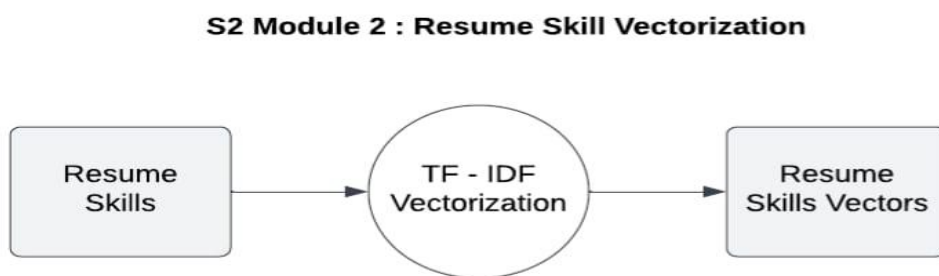


Figure 7: DFD for Resume skill vectorization

MODULE 3: COSINE SIMILARITY

The third module calculates the similarity between job roles and resumes using cosine similarity, a metric that measures the angle between two vectors. The inputs are the job role vectors (from Module 1) and the resume skill vectors (from Module 2). Cosine similarity helps in evaluating how closely aligned the skillsets in a resume are to the requirements of a job role by comparing the direction and magnitude of the vectors. A higher cosine similarity score indicates a closer match, suggesting that the candidate's skills closely align with the job requirements. The output of this module is a similarity score, which quantifies the match between the candidate's resume and the job role, aiding in ranking or selecting candidates based on their suitability for the position.

S2 Module 3 : Cosine similarity

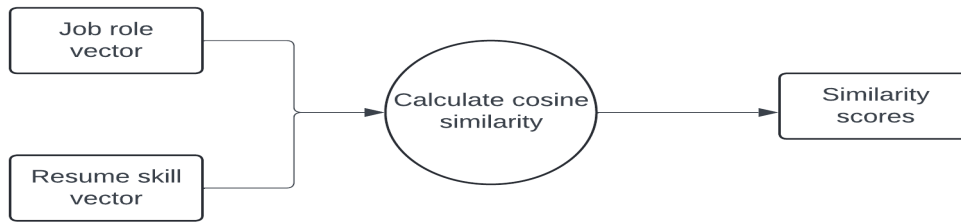


Figure 8: DFD for Cosine similarity

MODULE 4: RANKING AND VISUALIZATION

This module outlines a process for ranking and visualizing job role recommendations based on relevance to a user's profile. It begins by taking two inputs: a dataset of job roles and similarity scores, which indicate how closely each job role aligns with the user's skills, experience, or interests. Using these inputs, the system ranks job roles, prioritizing those with higher similarity scores. From this ranked list, it selects the top "N" job roles as recommendations tailored to the user. These recommended job roles are then visualized in a bar chart using Matplotlib, a popular Python library for data visualization. The bar chart provides an intuitive overview, making it easier for users to compare the most relevant job roles at a glance. This approach not only delivers a curated list of job roles but also offers a visual representation that aids in quick interpretation and informed decision-making regarding career options.

S2 Module 4 : Ranking and Visualization

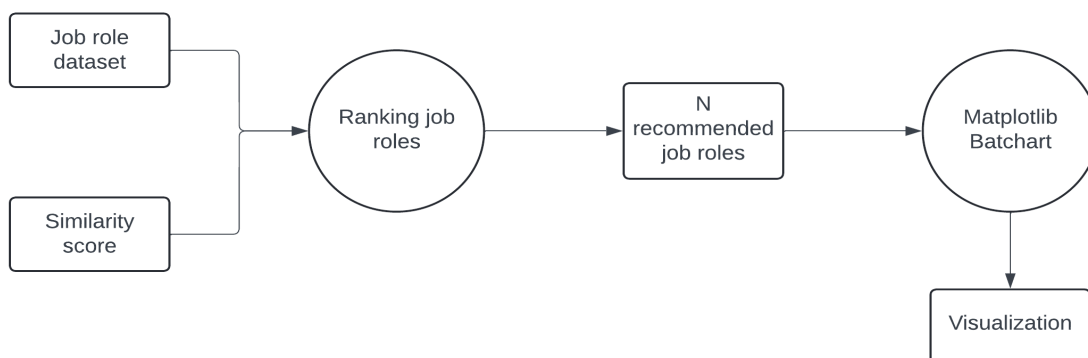


Figure 9: DFD for Ranking and Visualization

CHAPTER VI

RESULT AND DISCUSSION

The “SkillPilot” system demonstrates strong performance in automating skill analysis and job role recommendations. The Skill Extraction Module, using PyMuPDF and spaCy, efficiently identifies both technical and soft skills from diverse resume formats, achieving a high extraction accuracy.

The Skill Matching Module compares these extracted skills with job role requirements using TF-IDF and Cosine Similarity. It maintains an impressive matching accuracy of around 98%, effectively pinpointing both present and missing skills. The Job Role Recommendation Module, employing a Random Forest classifier, achieves 98% accuracy in predicting roles aligned with user skills, providing practical career guidance.

Visualizations, such as pie charts and bar graphs generated with Matplotlib, offer intuitive feedback. Users find these visuals helpful for understanding their skill profiles and identifying areas for improvement. Feedback from initial testing has been positive, with users praising the system's ability to simplify job application preparation and enhance career planning.

Overall, the results confirm that “SkillPilot” offers a reliable, efficient, and user-friendly tool for skill gap analysis and job role recommendations, with potential for further enhancements through advanced NLP models and a broader job role dataset.

CHAPTER VII

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The “SkillPilot” project provides an innovative approach to resume analysis and job role recommendation, addressing the prevalent challenges job seekers face in today’s competitive market. By utilizing advanced natural language processing (NLP) techniques and machine learning algorithms, the system accurately extracts skills from resumes and compares them with job role requirements. The integration of TF-IDF for skill vectorization and Cosine Similarity for matching ensures precise and context-aware analysis, while the use of a Random Forest classifier enhances the accuracy of job role predictions.

The results demonstrate the effectiveness of “SkillPilot” in identifying skill gaps and offering personalized feedback through intuitive visualizations, such as pie charts and bar graphs. Users gain a clear understanding of their strengths and areas for improvement, empowering them to optimize their resumes and focus on relevant skill development. Additionally, the system streamlines the job search process, providing valuable insights that save time and effort for both job seekers and recruiters. Overall, “SkillPilot” stands out as a reliable, efficient, and user-friendly tool that enhances employability and career growth.

7.2 FUTURE ENHANCEMENT

There are several opportunities for future enhancements to make “SkillPilot” even more robust and comprehensive. One key improvement would be the incorporation of more sophisticated NLP models, such as transformer-based architectures like BERT or GPT, to improve the system’s understanding of complex and nuanced

resume content. These models could enhance the accuracy of skill extraction and better handle variations in resume formats and language use.

Expanding the database to include a more diverse set of job roles across different industries would increase the system's relevance and applicability for a wider audience. This could involve collecting job data from additional sectors like healthcare, finance, and education. Moreover, integrating real-time job market analytics could refine the job role recommendations, ensuring they reflect current industry trends and demands.

Another potential enhancement is the development of personalized learning paths for users. By suggesting courses, certifications, or training programs to address missing skills, "SkillPilot" could transform into a comprehensive career development platform. Additionally, features like a skill progress tracker and integration with online learning platforms could further support users in achieving their professional goals. These future improvements would make "SkillPilot" an even more powerful tool for job seekers and contribute to ongoing advancements in human resource technology.

APPENDIX

A1.1 SAMPLE CODE

```
import pandas as pd
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import accuracy_score
import matplotlib.pyplot as plt
!pip install pymupdf
import fitz # PyMuPDF for PDF extraction
import spacy
!pip install python-docx
from docx import Document

# Load spaCy's English model for NLP
nlp = spacy.load("en_core_web_sm")
# Example paths
resume_pdf_path = '/content/drive/MyDrive/Sandhya Resume.pdf' # Path to the
PDF resume
docx_path = '/content/drive/MyDrive/Skill dataset.docx' # Path to the uploaded
DOCX file
# Load the dataset from an Excel file
job_role_df = pd.read_excel('/content/drive/MyDrive/job role and skills.xlsx') #
Path to Excel file
# Function to extract text from a PDF resume
def extract_text_from_pdf(pdf_path):
    doc = fitz.open(pdf_path)
    text = ""
    for page_num in range(doc.page_count):
        page = doc.load_page(page_num)
```

```

        text += page.get_text()

    return text

# Step 1: Extract text from the PDF resume
resume_text = extract_text_from_pdf(resume_pdf_path)
print("Extracted Resume Text:", resume_text)

# Function to extract skill keywords from a DOCX file
def extract_skills_from_docx(docx_path):
    doc = Document(docx_path)
    skills = []

    for paragraph in doc.paragraphs:
        skills.extend(paragraph.text.split()) # Split by spaces
    skills = [skill.strip().lower() for skill in skills if skill.strip()] # Clean up
    whitespace and convert to lower case
    return skills

# Step 2: Extract skills from the DOCX file
skill_keywords = extract_skills_from_docx(docx_path)
print("Extracted Skill Keywords:", skill_keywords)

# Function to extract relevant skills from resume text based on the skill keywords
def extract_skills_from_resume(text, skill_keywords):
    doc = nlp(text)
    resume_skills = set()

    for token in doc:
        if token.lemma_.lower() in skill_keywords: # Compare lowercased
            lemmatized tokens
            resume_skills.add(token.lemma_.lower())

    return list(resume_skills)

# Step 3: Extract skills from resume text using the extracted skill keywords
resume_skills = extract_skills_from_resume(resume_text, skill_keywords)
print("Extracted Skills from Resume:", resume_skills)

# User-selected job role (in practice, this would be selected from a UI dropdown)
selected_job_role = 'Data Scientist'

```

```

selected_job_role_data = job_role_df.loc[job_role_df['Job Role'] ==
selected_job_role]

# Ensure that the job role exists in the dataset
if selected_job_role_data.empty:
    print(f"Job role '{selected_job_role}' not found in the dataset.")
else:
    # Step 5: Preprocess and apply TF-IDF vectorization only for the selected job
    role's skills
    selected_job_role_data.loc[:, 'Technical Skills Needed'] =
selected_job_role_data["Technical Skills Needed"].apply(lambda x: ' '.join(x.split()))

    print("Data prepared successfully")
if not selected_job_role_data.empty:
    # Step 6: Apply TF-IDF Vectorization on 'Technical Skills Needed' for selected
    job role
    from sklearn.feature_extraction.text import TfidfVectorizer

    tfidf_vectorizer = TfidfVectorizer()
    X_selected = tfidf_vectorizer.fit_transform(selected_job_role_data["Technical
Skills Needed"])

    print("TF-IDF vectorization successful")
    #print(X_selected)
if not selected_job_role_data.empty:
    from sklearn.model_selection import train_test_split
    from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score

    # Step 7: Train-Test Split
    y_selected = selected_job_role_data['Job Role']

```

```
X_train, X_test, y_train, y_test = train_test_split(X_selected, y_selected,
test_size=0.3, random_state=42)
```

```
# Step 8: Train the Random Forest Model
```

```
rf_model = RandomForestClassifier(n_estimators=100, random_state=42)
rf_model.fit(X_train, y_train)
```

```
# Step 9: Evaluate Model Accuracy
```

```
y_pred = rf_model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Model Accuracy: {accuracy * 100:.2f}%")
```

```
if not selected_job_role_data.empty:
```

```
    resume_skills_str = ''.join(resume_skills)
```

```
# Step 10: Vectorize resume skills
```

```
resume_vector = tfidf_vectorizer.transform([resume_skills_str])
print(resume_vector)
```

```
if not selected_job_role_data.empty:
```

```
    # Clean and normalize required skills
```

```
    required_skills = [skill.lower().strip(",() ") for skill in required_skills] # Remove
unwanted chars and convert to lower case
```

```
    # Find matched and missing skills
```

```
    matched_skills = [skill for skill in resume_skills if skill in required_skills]
```

```
    missing_skills = [skill for skill in required_skills if skill not in resume_skills]
```

```
    print(f"Matched Skills: {matched_skills}")
```

```
    print(f"Missing Skills: {missing_skills}")
```

```
    # Step 13: Plot the pie chart
```

```
    import matplotlib.pyplot as plt
```

```

matched_percentage = (len(matched_skills) / len(required_skills)) * 100
missing_percentage = 100 - matched_percentage
print(len(matched_skills))
print(len(required_skills))

labels = ['Matched Skills', 'Missing Skills']
sizes = [matched_percentage, missing_percentage]
colors = ['#4CAF50', '#FF6347']

plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140)
plt.axis('equal')
plt.title(f"Skill Match for {selected_job_role}")
plt.show()

# Apply TF-IDF to the 'Technical Skills Needed' column to create feature vectors
for all job roles
X = tfidf_vectorizer.fit_transform(job_role_df['Technical Skills Needed'])

# Step 5: Convert the user's resume skills into a string and vectorize it using the
same TF-IDF vectorizer
resume_skills_str = ' '.join(resume_skills)
resume_vector = tfidf_vectorizer.transform([resume_skills_str])

print("Vectorized Resume Skills:", resume_vector)

# Step 6: Calculate the cosine similarity between the resume vector and all job role
vectors
similarity_scores = cosine_similarity(resume_vector, X)

# Flatten the similarity scores array
similarity_scores = similarity_scores.flatten()

```

```

# Step 7: Add similarity scores to the job role dataset for ranking
job_role_df['Similarity Score'] = similarity_scores

# Step 8: Rank the job roles based on similarity scores
job_role_df_sorted = job_role_df.sort_values(by='Similarity Score',
ascending=False)

# Step 9: Recommend the top N job roles
top_N = 3 # You can change this value to recommend more or fewer jobs
recommended_jobs = job_role_df_sorted[['Job Role', 'Similarity
Score']].head(top_N)

# Print top N recommended job roles and their similarity scores
print(f"Top {top_N} Roles matched with your resume:")

for idx, row in enumerate(recommended_jobs.itertuples(), 1):
    print(f"{idx}. {row[1]} with a similarity score of {row[2]:.2f}")

import matplotlib.pyplot as plt
import numpy as np

# Step 10: Visualize the top recommended job roles with their similarity scores,
with different colors
colors = plt.cm.viridis(np.linspace(0, 1, top_N)) # Create a color gradient
plt.figure(figsize=(8, 6))
plt.barh(recommended_jobs['Job Role'], recommended_jobs['Similarity Score'],
color=colors)
plt.xlabel('Similarity Score')
plt.title('Top Recommended Job Roles')
plt.gca().invert_yaxis() # Invert the y-axis to show the highest score on top
plt.show()

```


A1.2 SCREENSHOTS

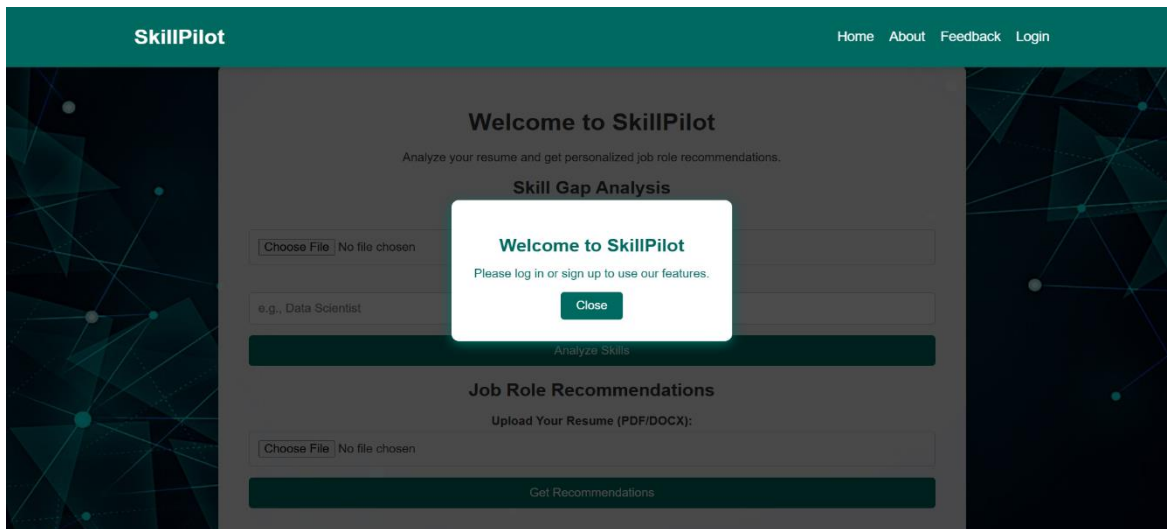


Figure 10: SkillPilot Welcome Page

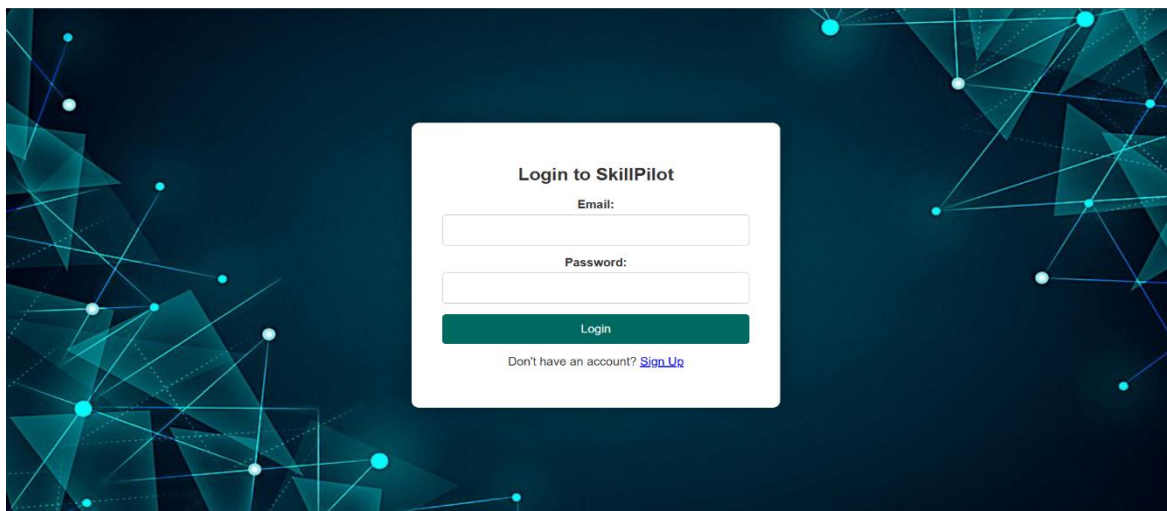


Figure 11: SkillPilot Login and Signup Page

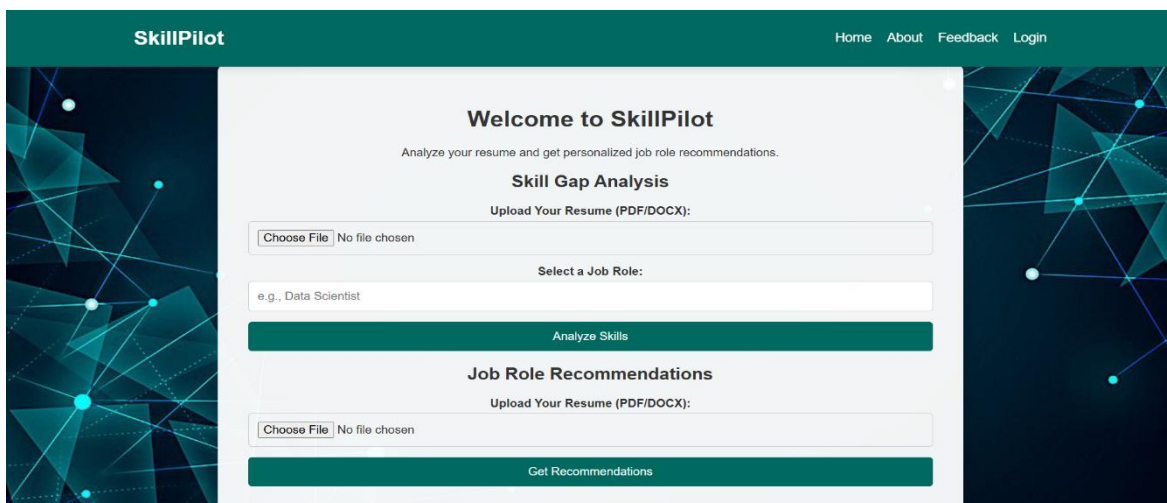
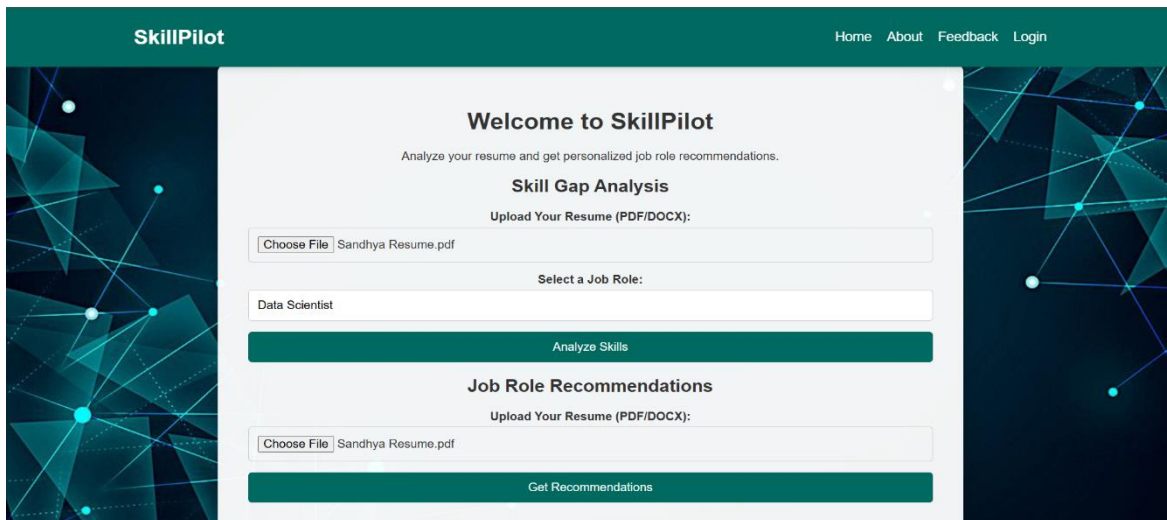


Figure 12: SkillPilot Home Page



SkillPilot Home About Feedback Login

Welcome to SkillPilot

Analyze your resume and get personalized job role recommendations.

Skill Gap Analysis

Upload Your Resume (PDF/DOCX):

Sandhya Resume.pdf

Select a Job Role:

Data Scientist

Job Role Recommendations

Upload Your Resume (PDF/DOCX):

Sandhya Resume.pdf

Figure 13: SkillPilot Sample Input page

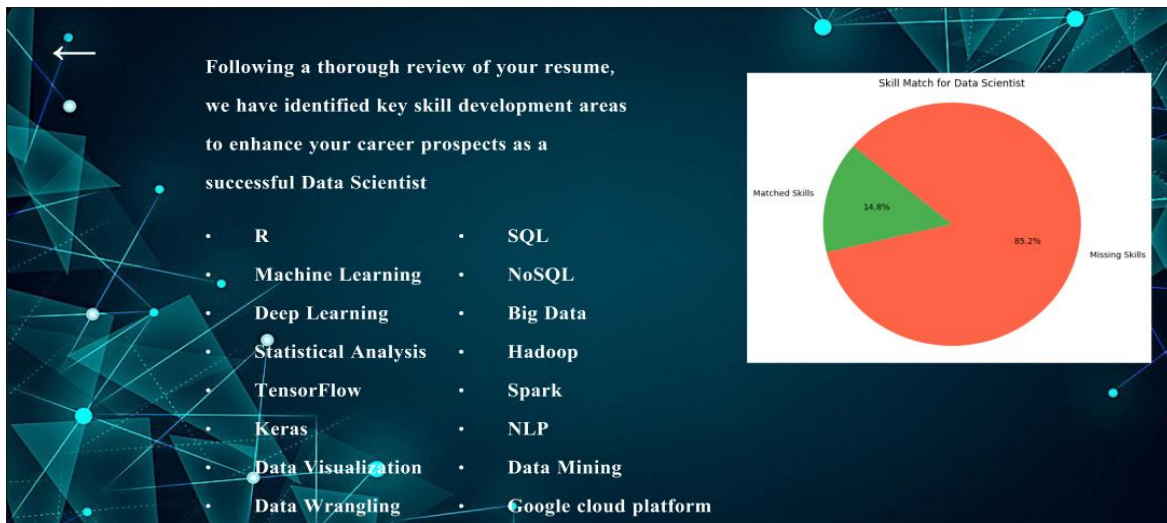


Figure 14: Skill Recommender

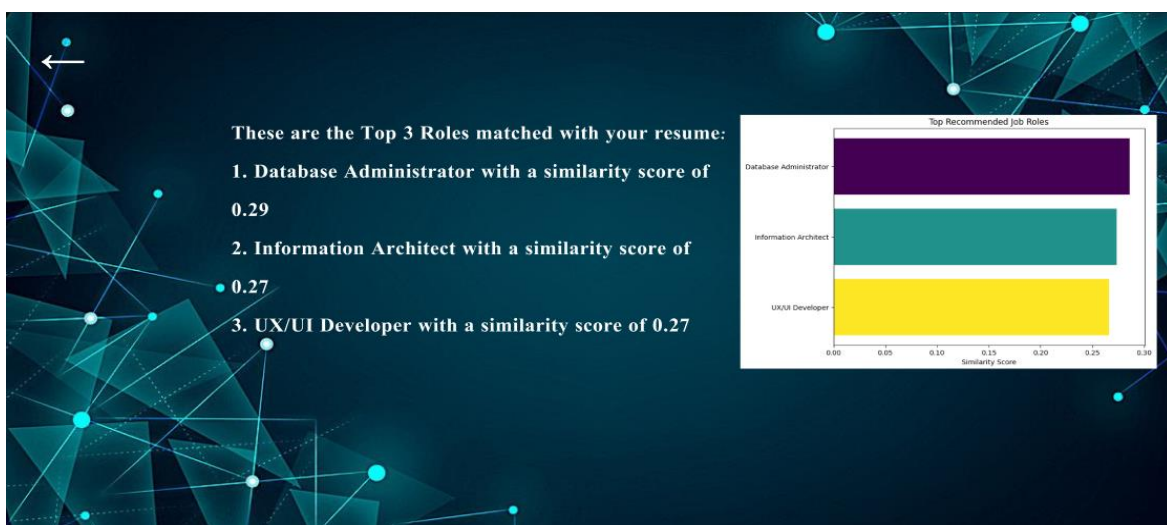


Figure 15: Job Recommender

REFERENCES

- [1] M. C. Cohen, N. H. Z. Leung, K. Panchamgam, G. Perakis, and A. Smith, "The impact of linear optimization on promotion planning," *Operations Research*, vol. 65, no. 2, pp. 446–468.
- [2] R. Fildes, P. Goodwin, and D. O' nkal, "Use and misuse of information in supply chain forecasting of promotion effects," *International Journal of Forecasting*, vol. 35, no. 1, pp. 144–156.
- [3] S. Makridakis, "The art and science of forecasting An assessment and future directions," *International Journal of Forecasting*, vol. 2, no. 1, pp. 15–39.
- [4] E. S. Gardner Jr., "Exponential Smoothing: The State of the Art," vol. 4, no.1, pp. 1–28.
- [5] T.-M. Choi, Y. Yu, and K.-F. Au, "A hybrid SARIMA wavelet transform method for sales forecasting," *Decision Support Systems*, vol. 51, no. 1, pp. 130–140.
- [6] N. S. Arunraj and D. Ahrens, "A hybrid seasonal autoregressive in- tegrated moving average and quantile regression for daily food sales forecasting," *International Journal of Production Economics*, vol. 170, pp. 321–335.
- [7] C. W. Chu and G. P. Zhang, "A comparative study of linear and nonlinear models for aggregate retail sales forecasting," *International Journal of Production Economics*, vol. 86, no. 3, pp. 217–231.
- [8] C. Y. Chen, W. I. Lee, H. M. Kuo, C. W. Chen, and K. H. Chen, "The study of a forecasting sales model for fresh food," *Expert Systems with Applications*, vol. 37, no. 12, pp. 7696–7702.
- [9] K.-F. Au, T.-M. Choi, and Y. Yu, "Fashion retail forecasting by evolution- ary neural networks," *International Journal of Production Economics*, vol. 114, no. 2, pp. 615 – 630.
- [10] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using ex- treme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419.