MACHINE LEARNING BASED SYSTEM FOR PERSONALIZED JOB ROLE RECOMMENDATIONS AND INDUSTRY SPECIFIC TAILORING

A PROJECT REPORT

submitted by

VIGNESHWARAN G 2116210701307

VISHVA A 2116210701314

VINOTH RAJ G 2116210701509

in partial fulfillment for the award of the

degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE

ANNA UNIVERSITY, CHENNAI

MAY 2024

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

BONAFIDE CERTIFICATE

Certified that this Project report titled **"MACHINE LEARNING BASED SYSTEM FOR PERSONALIZED JOB ROLE RECOMMENDATIONS AND INDUSTRY SPECIFIC TAILORING "** is the bonafide work of **"VIGNESHWARAN G -2116210701307, VISHVA A -2116210701314, VINOTH RAJ G -2116210701509"** 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.

SIGNATURE

Dr.P.SHANMUGAM,**M.Tech.,Ph.D**,

PROJECT COORDINATOR

Associate Professor

Department of Computer Science and Engineering

Rajalakshmi Engineering College

Chennai - 602 105

Submitted to Project Viva-Voce Examination held on ____________________

Internal Examiner **External Examiner**

ABSTRACT

The system collects job-related information from industry information website, including factors such as job title, salary estimates, company ratings, location, industry, and more. By analyzing this data, we aim to generate accurate recommendations tailored to the user's skills, interests, and career goals. Key components of our system include Data Scraping where we scrape relevant job data from industry information website to build our recommendation system, Feature Engineering where we preprocess and transform raw data, handling missing values, encoding categorical variables, and normalizing numerical features and Machine Learning Techniques where we employ TF-IDF (Term Frequency-Inverse Document Frequency) vectorization to capture the importance of words in job descriptions and user preferences. This enables the system to find similar job opportunities. Our system not only recommends relevant job roles but also facilitates connections with industry experts. By combining data scraping, machine learning, and cloud deployment, we create an AI-powered solution that benefits both job seekers and employers.

ACKNOWLEDGMENT

We thank the almighty god for the successful completion of the project. Our sincere thanks to our chairman **Mr. S. MEGANATHAN,B.E., F.I.E.,** for his sincere endeavor in educating us in his premier institution. We would like to express our deep gratitude and sincere thanks to our beloved Chairperson **Dr.(Mrs).THANGAM MEGANATHAN,Ph.D.,**for her enthusiastic motivation which inspired us a lot in completing this project and we extend our gratitude to our Vice Chairman **Mr.ABHAY SHANKAR MEGANATHAN,B.E., M.S.,** for providing us with the requisite infrastructure.

We also express our sincere and warmful gratitude to our college Principal **Dr. S.N.MURUGESAN,M.E.,Ph.D.**,and **Dr. P. KUMAR,M.E.,Ph.D.**, **Director of Computing and Information Science** and **Head Of the Department of Computer Science and Engineering** and **Project coordinator Dr.P.SHANMUGAM,M.Tech.,Ph.D.**,for his encouragement and guiding us throughout project towards successful completion of project and to our parents, friends, all faculty members,supporting staff for direct and indirect involvement in successful completion of project for their encouragement and support.

VIGNESHWARAN G

VISHVA A

VINOTH RAJ G

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The "Machine Learning Based System for Personalized Job Role Recommendations and Industry Specific Tailoring" project aims to revolutionize the job search process by leveraging advanced machine learning algorithms to provide highly personalized job role recommendations. This system will analyze individual profiles, including skills, experiences, and preferences, to match users with the most suitable job roles. Furthermore, it will tailor recommendations to align with specific industry demands, ensuring that both job seekers and employees benefit from a more precise and efficient matching process. By enhancing the relevance and accuracy of job recommendations, this project seeks to streamline career development and optimize workforce placement.

1.1 PROBLEM STATEMENT

The project "Machine Learning Based System for Personalized Job Role Recommendations and Industry Specific Tailoring" aims to develop an intelligent platform that leverages machine learning algorithms to provide personalized job role recommendations to individuals. By analyzing user profiles, including their skills, experiences, and preferences, the system will generate tailored job suggestions that align with their career aspirations and industry demands. Additionally, the platform will incorporate industry-specific tailoring, ensuring that recommendations are not only personalized but also relevant to the current market trends and opportunities within specific sectors. This system seeks to enhance job search efficiency and satisfaction by bridging the gap between job seekers' qualifications and the dynamic needs of industries.

1.2 SCOPE OF THE WORK

In the proposed model, the scope of this work encompasses the development and implementation of a sophisticated AI-driven platform designed to enhance career guidance. This system will leverage machine learning algorithms to analyze a wide array of user data, including educational background, professional experience, skills, and personal preferences. By doing so, it aims to deliver personalized job role recommendations that are finely tuned to each user's unique profile. Additionally, the project will incorporate industry-specific tailoring, ensuring that the recommendations are not only personalized but also aligned with current market demands and trends within various sectors. This comprehensive approach aims to bridge the gap between job seekers and suitable employment opportunities, thereby optimizing the job search process and increasing the chances of successful career placements.

1.3 AIM AND OBJECTIVES OF THE PROJECT

The aim of the proposed system is to develop an advanced algorithmic framework that leverages machine learning techniques to provide customized job role suggestions to individuals. This system will analyze user profiles, including skills, experience, and career goals, to deliver highly personalized job recommendations. Additionally, the system will incorporate industry-specific tailoring, ensuring that the recommendations are not only personalized but also aligned with the unique demands and trends of various sectors, thereby enhancing the relevance and accuracy of job matches for users.

The primary objective of our proposed system is to develop an intelligent platform that leverages machine learning algorithms to provide highly customized job role recommendations for individuals based on their unique skills, experiences, and career aspirations. By integrating industry-specific tailoring, the system aims to align candidate profiles with the evolving demands of various sectors, enhancing the accuracy and relevance of job matches. This personalized approach not only streamlines the job search process for users but also assists employers in identifying the most suitable candidates, ultimately fostering a more efficient and effective employment ecosystem.

1.4 RESOURCES

To develop a machine learning-based system for personalized job role recommendations and industry-specific tailoring, key resources include comprehensive datasets such as job listings, employee profiles, and industry trends from sources like LinkedIn, Glassdoor, and Indeed. Natural language processing (NLP) tools like spaCy and NLTK can help in analyzing job descriptions and resumes. Machine learning frameworks like TensorFlow, PyTorch, and scikit-learn are essential for building recommendation algorithms. Access to cloud computing platforms such as AWS, Google Cloud, or Azure will facilitate scalable data processing and model training. Additionally, industry-specific reports and market analysis from sources like Gartner or McKinsey can provide valuable insights for tailoring recommendations to specific fields.

1.5 MOTIVATION

The project "Machine Learning Based System for Personalized Job Role Recommendations and Industry Specific Tailoring" seeks to revolutionize the way individuals navigate the job market. By harnessing the power of machine learning, this project aims to provide tailored job role suggestions based on an individual's skills, experience, and preferences. Furthermore, by incorporating industry-specific insights, the system aims to bridge the gap between job seekers and employers, fostering better matches and enhancing career satisfaction. Ultimately, this project endeavors to empower individuals to make informed career decisions and unlock opportunities aligned with their unique talents and aspirations.

CHAPTER 2

2.1 LITERATURE SURVEY

[1] **“Resume Parser and Job Recommendation System using Machine Learning, 2024” by Ashish Virendra Chandak; Hardik Pandey; Gourav Rushiya; Harsh Sharma**, proposed a technique which consists of two pivotal components: a Resume Parser and a Job Recommender. The Resume Parser is implemented in Python which consist of libraries such as spaCy and PDFMiner to extract vital resume details. The extracted skills form the foundation for generating job recommendations.

[2] **”Job Recommendation System using LinkedIn User Profiles, 2024” by Govinda K; Meghanath Reddy K; Ananya Haldar”** investigates the recommender system for LinkedIn user profiles to generate job recommendations, highlighting the key factors and approaches that influence the recommendation process and there is potential for bias in LinkedIn's job recommendation system.

[3] **“Applying Machine Learning Algorithm to Optimize Personalized Education Recommendation System , 2024" authors Wangmei Chen, Zepeng Shen, Yiming Pan, Kai Tan and Cankun Wang** present an education system personalized and optimized by machine learning algorithms can provide customized learning materials and recommendations based on

each student's learning history, interests and abilities to improve learning outcomes, and machine learning algorithms can provide real-time feedback on student performance and adjust learning plans based on feedback.

[4] **"Job offers recommender system based on virtual organizations, 2022" authors Alfonso González-Briones, Pablo Chamoso, Juan Pavón, Fernando De La Prieta, Juan M. Corchado,** present a recommender system for a business and employment oriented social network, on which users are recommended job offers and other user profiles to follow. The presented system is based on virtual organizations of agents, and uses artificial neural networks to determine whether job offers and users should be recommended or not.

[5] **"Exploring Large Language Model for Graph Data Understanding in Online Job Recommendations, 2024" authors Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu and Enhong Chen** presents a meta-path prompt constructor that aids LLM recommender in grasping the semantics of behavior graphs for the first time and design a corresponding path augmentation module to alleviate the prompt bias introduced by path-based sequence input. By facilitating this capability, our framework enables personalized and accurate job recommendations for individual users.

[6] **“A Job Recommendation Model Based on a Two-Layer Attention Mechanism, 2024" by Yu Mao ,Shaojie Lin and Yuxuan Cheng** present a recurrent neural network model based on a two-layer attention mechanism. The model first improves the entity representation of recruiters and applicants through user behavior, company-related knowledge and other information.

2.2 PROPOSED SYSTEM

DATASET:

The dataset for a machine learning-based job role recommendation system consists of several key components. It includes detailed profiles of job seekers, such as their educational background, skills, work experience, and career objectives. Each profile is labeled with a current job title or role. Additionally, the dataset contains a comprehensive list of job descriptions, including required skills, qualifications, and responsibilities. Interaction data, such as job applications, interview invitations, and employment outcomes, is also included to capture the relationship between job seekers and job roles. Demographic information like age, gender, and location might be present to provide insights into diversity and regional job market trends. The dataset is designed to be used for training machine learning models to predict the most suitable job roles for individuals, enhance job matching accuracy, and improve employment rates by aligning candidates' profiles with appropriate job opportunities.

MODEL ARCHITECTURE:

The machine learning-based job role recommendation system utilizes a hybrid architecture combining content-based filtering and collaborative filtering. It employs a neural network model with user and job profile embeddings. Initially, a data preprocessing step involves cleaning and vectorizing job descriptions and user resumes using NLP techniques like TF-IDF or word embeddings (e.g., Word2Vec, BERT). The neural network comprises an embedding layer to transform textual data into dense vectors, followed by a few dense layers for feature extraction. User-job interaction data feeds into a collaborative filtering module, often leveraging matrix factorization or deep learning techniques like neural collaborative filtering (NCF). These models predict user-job suitability scores. Finally, a ranking layer sorts job recommendations based on these scores. The system continually updates its recommendations by incorporating real-time user feedback and new job postings, ensuring personalized and dynamic job suggestions.

TRAINING AND TESTING:

The training and testing process for our proposed model involves several key steps. Firstly, data collection from various sources such as job listings, resumes, and industry reports is essential to build a robust dataset. Next, data preprocessing techniques are applied to clean and prepare the data for analysis. Then, machine learning algorithms like classification and clustering are employed to model the relationships between job roles, skills, and industry preferences. The system is trained on a portion of the data and tested on another to evaluate its performance and fine-tune parameters. Continuous iteration and validation are crucial to ensure the system provides accurate and personalized job recommendations tailored to individual preferences and industry requirements.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

In this section, we would like to show how the general outline of how all the components end up working when organized and arranged together. It is further represented in the form of a flow chart below.

3.2 SYSTEM ARCHITECTURE DIAGRAM

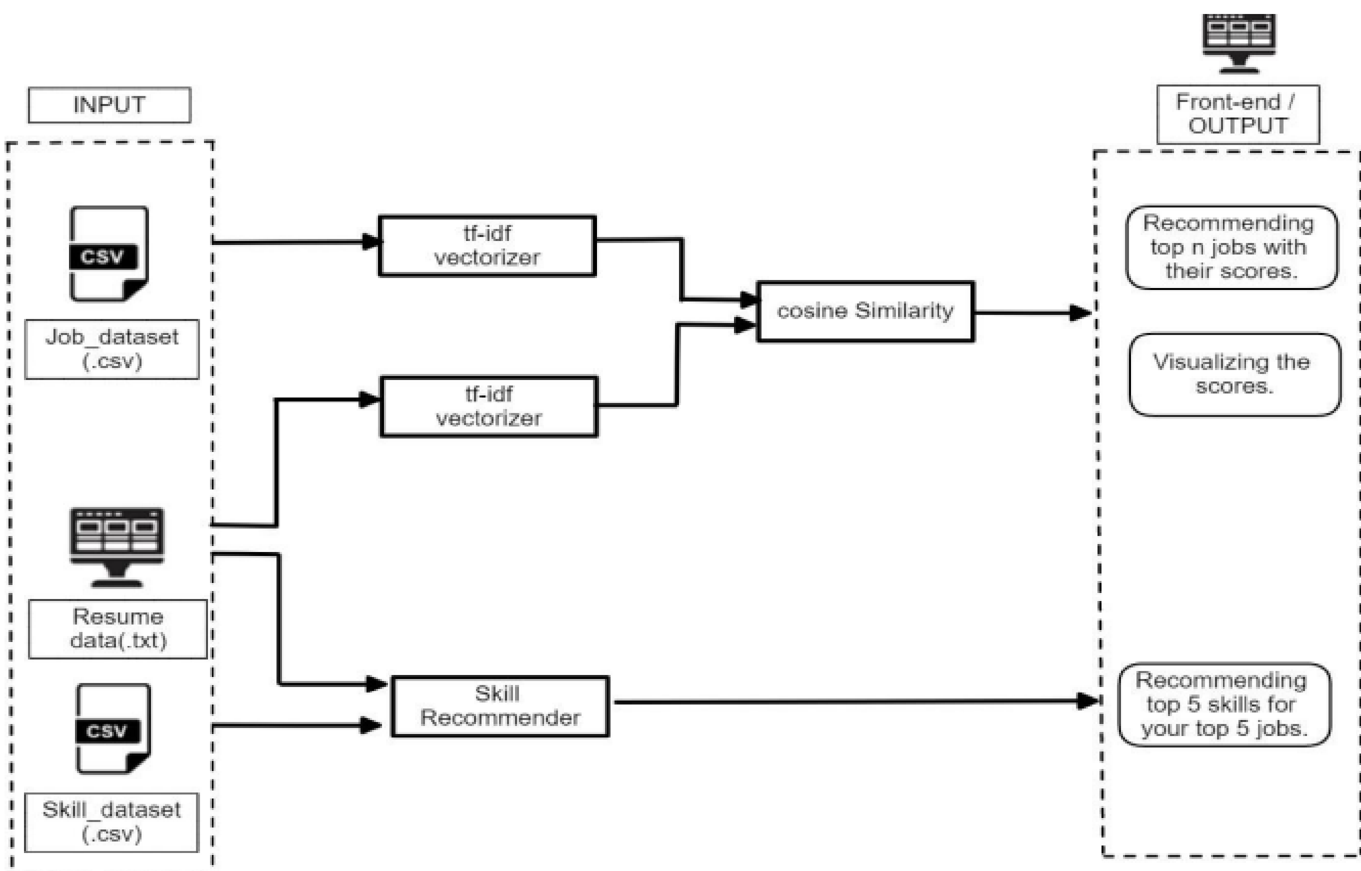

Fig 3.2.1: System Architecture

3.3 DEVELOPMENTAL ENVIRONMENT

3.3.1 HARDWARE REQUIREMENTS

The hardware requirements may serve as the basis for a contract for the system's implementation. It should therefore be a complete and consistent specification of the entire system. It is generally used by software engineers as the starting point for the system design.

Table 3.3.1 Hardware Requirements

COMPONENTS	SPECIFICATION
PROCESSOR	Intel Core i5
RAM	8 GB RAM
GPU	NVIDIA GeForce GTX 1650
MONITOR	15" COLOR
HARD DISK	512 GB
PROCESSOR SPEED	MINIMUM 1.1 GHz

3.3.2 SOFTWARE REQUIREMENTS

The software requirements document is the specifications of the system. It should include both a definition and a specification of requirements. It is a set of what the system should rather be doing than focus on how it should be done. The software requirements provide a basis for creating the software requirements specification.The software requirements are description of features and functionalities of the target system. Requirements convey the expectations of users from the software product.

Table 3.3.2 Software Requirements

S.NO	REQUIREMENT
1	Jupyter Notebook
2	StreamLit API
3	TensorFlow

3.4 DESIGN OF THE ENTIRE SYSTEM:

3.4.1 SEQUENCE DIAGRAM:

A sequence diagram simply depicts the interaction between the objects in a sequential order. An sequence diagram is used to show the interactive behavior of a system. The sequence diagram for Disease prediction in plants and recommendation of fertilizer is attached in the below figure 3.4.1.

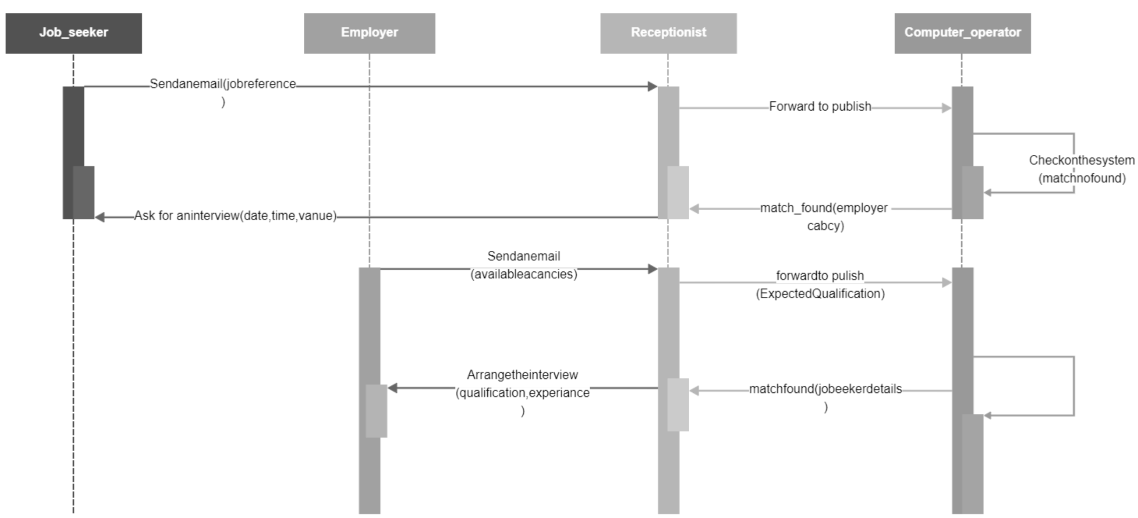

Fig 3.4.1: Sequence Diagram

CHAPTER 4

PROJECT DESCRIPTION

4.1 METHODOLODGY

The methodology for our proposed model involves several key steps. Firstly, comprehensive data collection from various sources such as job postings, resumes, and industry reports is conducted. Next, feature engineering is employed to extract relevant information from the collected data. Following this, machine learning algorithms such as collaborative filtering, natural language processing, and clustering techniques are utilized to develop personalized job role recommendation models. These models are then fine-tuned and validated using techniques like cross-validation and A/B testing. Finally, the system undergoes iterative improvements based on user feedback and performance evaluation metrics to ensure its effectiveness and relevance in providing tailored job recommendations for individuals across diverse industries.

4.2 MODULE DESCRIPTION

Gather relevant datasets containing job descriptions, candidate profiles, industry trends, and skill requirements. Identify and acquire diverse datasets from job boards, HR platforms, and industry reports. Clean and preprocess data to remove noise, standardize formats, and enhance compatibility. Perform exploratory data analysis to identify patterns and insights. Extract meaningful features from the collected data to represent job roles, candidate profiles, and industry-specific requirements. Identify relevant features such as skills, experience, education, and industry keywords. Design algorithms to convert unstructured data (e.g., text descriptions) into structured features. Explore techniques like word embeddings, TF-IDF, and semantic analysis for feature representation.

Develop predictive models to recommend personalized job roles based on candidate profiles and industry trends. Customize job recommendations based on individual candidate preferences, skills, and career aspirations. Adapt job recommendations and skill development paths to align with the unique requirements of different industries. Adapt job recommendations and skill development paths to align with the unique requirements of different industries. Deploy the system in production environments and integrate it with existing HR platforms, job portals, and career development tools. Monitor system performance, user interactions, and data quality to ensure long-term effectiveness and reliability.

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 OUTPUT

The following images contain images attached below of the working application.

TF-IDF OUTPUT

```
...   [array([[0.01004203]]),
       array([[0.00277372]]),
       array([[0.01380723]]),
       array([[0.0042945]]),
       array([[0.01483838]]),
       array([[0.00927136]]),
       array([[0.]]),
       array([[0.0074321]]),
       array([[0.02078868]]),
       array([[0.00356108]]),
       array([[0.00450561]]),
       array([[0.00479991]]),
       array([[0.00466231]]),
       array([[0.]]),
       array([[0.00863563]]),
       array([[0.00862199]]),
       array([[0.00314284]]),
       array([[0.00507078]]),
       array([[0.01038314]]),
       array([[0.02121474]]),
       array([[0.01357133]]),
       array([[0.00945792]]),
       array([[0.02001338]]),
       array([[0.0120537]]),
       array([[0.02072318]]),
      ...
       array([[0.00336819]]),
       array([[0.01963864]]),
       array([[0.01232667]]),
       array([[0.0100778]]),
       ...]
```

Fig 5.1.1: TF-IDF OUTPUT

COUNT VECTORIZER OUTPUT

```
[array([[0.03279129]]),
 array([[0.0151255]]),
 array([[0.04519045]]),
 array([[0.03608178]]),
 array([[0.07871211]]),
 array([[0.03961453]]),
 array([[0.]]),
 array([[0.02699992]]),
 array([[0.09709831]]),
 array([[0.03507393]]),
 array([[0.0192281]]),
 array([[0.0295519]]),
 array([[0.01495413]]),
 array([[0.]]),
 array([[0.03435365]]),
 array([[0.03141437]]),
 array([[0.02201542]]),
 array([[0.02848725]]),
 array([[0.01609661]]),
 array([[0.02789883]]),
 array([[0.04364098]]),
 array([[0.03802288]]),
 array([[0.03926796]]),
 array([[0.03261062]]),
 array([[0.05052016]]),
...
 array([[0.02091405]]),
 array([[0.05092753]]),
 array([[0.04012087]]),
 array([[0.02923214]]),
 ...]
```

Fig 5.1.2: Output of count vectorizer

Text conversion :

Extraction of the text from the document provided to the model by getting technical and skill related words from the document by text processing Figure 5.1.3

```
Summary
Motivated and determined individual from a Data Science and Machine Learning
data analysis and machine learning in multiple data science projects. My
logical and analytical skills, as well as recognizing the importance teamwork
for opportunities where I can deploy data analytic skills.

Technical Skills
● Proficient: Python, Pandas, Keras-Tensorflow, Machine Learning, Deep Learning, Tableau
● Familiar with: SQL

Xccelerate Full-time Data Science and Machine Learning Immersive Bootcamp

Graduate from Xccelerate's Full-time Data Science Bootcamp, developing various data
both individually and in a team. Some of the Projects:

House Ownership Scheme Market Analysis (Web-scraping, Pandas)
● Deployed web-scraping technique on various websites and obtained valuable
● Deployed Pandas visualization tools to analyze housing estate information.
● Identified significant insights in the public housing estate market, i.e. locations

Flight Ticket Price Prediction (Machine Learning, Pandas)
● Performed techniques such as Pandas and Regular Expression on a large dataset
phase.
● Trained and deployed various regression models to conduct price prediction.
● Models could predict prices with more than 75%.

Facial Detection and Recognition System (Deep Learning)
● Deployed facial detection technique and captured faces from real-life picture
● Developed and optimized model with convolution neural network (CNN) for image predict
● Model achieved more than 85% on gender prediction and 83% on emotion

Work Experience
● Laboratory Technician at Smith and Associates Far East Ltd
Conduct quality control on electronic components via visual inspection and testing
Detect counterfeit electronic components and report relevant discrepancies

● Technician at Crestron Asia Limited 11/ 2015
Perform testing and repairing of company's products

Education

● Bachelor of Engineering in Chemical Engineering, University of Birmingham
● Brockenhurst College A-Levels: Mathematics, Chemistry, Physics, Chinese,

Language: Native in Cantonese, fluent in English and familiar with Mandarin
```

Fig 5.1.3: TEXT CONVERSION

Natural language processing:

The proposed model is evaluated and the testing and training accuracy graph is obtained. Spliting the dataset into training and validation sets (e.g., 80-20 split). Training the model using the training set, adjusting hyperparameters to optimize performance. Employ techniques such as dropout and batch normalization to prevent overfitting. The training and testing accuracy rate of the model is attached in the below figure 5.1.4

```
key_word = get_key_word(NLP_Processed_CV)
key_word
```

```
(('data', 'science'), 4)
(('deep', 'learning'), 2)
(('electronic', 'component'), 2)
(('facial', 'detection'), 2)
(('full-time', 'data'), 2)
(('housing', 'estate'), 2)
(('machine', 'learning'), 6)
(('prediction', 'model'), 2)
(('price', 'prediction'), 2)
(('science', 'machine'), 2)

['data science',
 'deep learning',
 'electronic component',
 'facial detection',
 'full-time data',
 'housing estate',
 'machine learning',
 'prediction model',
 'price prediction',
 'science machine']
```

Fig 5.1.4: Natural Language Processing

Job role recommendation

The output of the derived models with combined score from each of the modules to provide accurate output which provides insights on which role the context provide in the resume matches with jobs available in the database will be provide, so that the user can decide from various options available

	JobID	title	KNN	TF-IDF	CV	Final
0	641	Senior Machine Learning Engineer/Expert (Plann...	0.333333	0.333333	0.189947	0.856614
3	7357	Associate Data Scientist/ Data Scientist - Top...	0.27341	0.275167	0.278731	0.827308
7	7259	Data Insights Analyst (Big Data- Machine Learn...	0.249845	0.252081	0.325352	0.827278
2	3942	Software Engineer Trainee (AI / GIS)	0.275054	0.276773	0.248643	0.80047
10	857	Associate Data Scientist	0.214888	0.217617	0.310602	0.743106
8	7312	Artificial Intelligence (AI) R&D Engineer	0.242279	0.244644	0.238304	0.725226
9	5717	Data Science Manager - Machine Learning/AI | 2...	0.239048	0.241464	0.212282	0.692794
4	12794	Vice President, Big Data Analytics & Machine L...	0.270547	0.272368	0.142904	0.685819
12	3247	Machine Learning Engineer/ Data Scientist	0.19946	0.202323	0.259108	0.66089
14	7582	Data Scientist Team Lead	0.178896	0.181858	0.299642	0.660395

Fig 5.1.5: Job Role Recommendation Output

5.2 RESULT

The implementation of a Disease prediction in plants and recommendation of fertilizers using CNN and Random forest algorithm represents a remarkable journey by achieving 98% accuracy in identifying disease caused plants. Also the proposed system is computationally efficient because of the use of statistical image processing and machine learning model. The created system, which employs advanced image processing techniques and Machine learning algorithms.

The Proposed model enables farmers to make data-driven decisions that optimize yield potential while minimizing the risk of disease outbreaks and nutrient deficiencies. Moreover, the automated nature of the system streamlines the decisionmaking process, allowing for timely interventions and ultimately contributing to enhanced agricultural productivity and sustainability. As technology continues to advance, the integration of CNNs and Random Forests holds immense potential for revolutionizing crop management practices and empowering farmers with the tools needed to navigate the complexities of modern agriculture.

CHAPTER 6

CONCLUSION AND FUTURE ENHANCEMENT

6.1 CONCLUSION

The proposed model presents a significant advancement in the realm of career guidance and employment matching. Through the utilization of machine learning algorithms, it offers tailored recommendations for job roles based on individual skills, preferences, and industry requirements. This innovative system not only streamlines the job search process for individuals but also enhances workforce efficiency by aligning candidates with positions that best suit their capabilities. With its potential to revolutionize recruitment processes and optimize workforce utilization, this project marks a pivotal step towards a more personalized and effective approach to employment placement in diverse industries.

6.2 FUTURE ENHANCEMENT

Explainability and Transparency: Incorporate methods to explain how the recommendations are generated, increasing user trust and understanding of the system.

Personalization and Customization: Allow users to provide feedback and customize their preferences, such as preferred job locations, salary range, company size, etc., to tailor recommendations more accurately to individual needs.

APPENDIX

TFIDF=COUNTV=KNN.py:

```
import pdfplumber
import pandas as pd
import numpy as np
import nltk
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.corpus import stopwords
from string import punctuation
from nltk.corpus import wordnet as wn
from nltk.stem import WordNetLemmatizer
from nltk.probability import FreqDist
from heapq import nlargest
from collections import defaultdict
import pandas as pd
from nltk.collocations import *
 nltk.download('punkt')
 nltk.download('stopwords')
 nltk.download('wordnet')

with pdfplumber.open("/content/drive/MyDrive/test.pdf") as pdf:
    text=""
    pages = pdf.pages
    for page in pages:
        text += page.extract_text(x_tolerance=2)
        print(text)
 #Import CV
f=open('test.txt','r', errors = 'ignore')
```

```
text=f.read()

# Locations
f1=open('hk_districts.txt','r', errors = 'ignore')
text1=f1.read()
locations = word_tokenize(text1.replace("\n", " "))

# Additional stopwords
f2=open('stopwords.txt','r', errors = 'ignore')
text2=f2.read()
stopwords_additional = word_tokenize(text2.replace("\n", " "))

def nlp(x):
    word_sent = word_tokenize(x.lower().replace("\n",""))
    _stopwords = set(stopwords.words('english') + list(punctuation)+list("●")+list('–
')+list('’')+locations+stopwords_additional)
    word_sent=[word for word in word_sent if word not in _stopwords]
    lemmatizer = WordNetLemmatizer()
    NLP_Processed_CV = [lemmatizer.lemmatize(word) for word in word_tokenize("
".join(word_sent))]
#    return " ".join(NLP_Processed_CV)
    return NLP_Processed_CV

def get_key_word(x):
    finder = BigramCollocationFinder.from_words(x)
    lst = []
    Key_word_from_CV = []
    for i in sorted(finder.ngram_fd.items()):
```

```
    # if a double word appears more than once, then print it out.
        if i[1] > 1:
            print(i)
            lst.append(i[0])
        else:
            pass

    # print("************************")
    for j in lst:
    #     print(" ".join(j))
        Key_word_from_CV.append(" ".join(j))
    return Key_word_from_CV
NLP_Processed_CV = nlp(text)
  NLP_Processed_CV=' '.join(NLP_Processed_CV)
NLP_Processed_CV
key_word = get_key_word(NLP_Processed_CV)
key_word


df.info()


df.shape()


#Import scraped data
df = pd.read_csv("whole-v7-nlp.csv")
df.head(2)


  from sklearn.metrics.pairwise import cosine_similarity
```

```
  from sklearn.feature_extraction.text import TfidfVectorizer
  tfidf_vectorizer = TfidfVectorizer(stop_words='english')
  tfidf_jobid = tfidf_vectorizer.fit_transform(df['All'])
# # print(tfidf_jobid)
  user_tfidf = tfidf_vectorizer.transform(df2['All'])

 cos_similarity_tfidf = map(lambda x: cosine_similarity(user_tfidf,x),tfidf_jobid)
  output2 = list(cos_similarity_tfidf)
  output2
 from sklearn.metrics.pairwise import cosine_similarity
  from sklearn.feature_extraction.text import TfidfVectorizer
  def TFIDF(scraped_data, cv):
      tfidf_vectorizer = TfidfVectorizer(stop_words='english')

      # TF-IDF Scraped data
      tfidf_jobid = tfidf_vectorizer.fit_transform(scraped_data)

        TF-IDF CV
      user_tfidf = tfidf_vectorizer.transform(cv)

#     # Using cosine_similarity on (Scraped data) & (CV)
      cos_similarity_tfidf = map(lambda x: cosine_similarity(user_tfidf,x),tfidf_jobid)

      output2 = list(cos_similarity_tfidf)
      return output2

  output2 = TFIDF(df['All'], df2['All'])
  output2
```

```
df2 = pd.DataFrame()
# append columns to an empty DataFrame
df2['title'] = ["I"]
df2['job highlights'] = ["I"]
df2['job description'] = ["I"]
df2['company overview'] = ["I"]
df2['industry'] = ["I"]

# Compare with the key words from CV only
df2['All'] = " ".join(key_word)

# Compare with the entire CV
  df2['All'] = " ".join(NLP_Processed_CV)
  df2

def get_recommendation(top, df_all, scores):
    recommendation = pd.DataFrame(columns = ['JobID', 'title', 'industry', 'job description','score'])
    count = 0
    for i in top:
#        recommendation.at[count, 'ApplicantID'] = u
        recommendation.at[count, 'JobID'] = df.index[i]
        recommendation.at[count, 'title'] = df['title'][i]
        recommendation.at[count, 'industry'] = df['industry'][i]
#        recommendation.at[count, 'location'] = df['location'][i]
        recommendation.at[count, 'job description'] = df['job description'][i]
        recommendation.at[count, 'score'] =  scores[count]
```

```
        count += 1
    return recommendation

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

def TFIDF(scraped_data, cv):
    tfidf_vectorizer = TfidfVectorizer(stop_words='english')

    # TF-IDF Scraped data
    tfidf_jobid = tfidf_vectorizer.fit_transform(scraped_data)

    # TF-IDF CV
    user_tfidf = tfidf_vectorizer.transform(cv)

    # Using cosine_similarity on (Scraped data) & (CV)
    cos_similarity_tfidf = map(lambda x: cosine_similarity(user_tfidf,x),tfidf_jobid)

    output2 = list(cos_similarity_tfidf)
    return output2

output2 = TFIDF(df['All'], df2['All'])
output2

top = sorted(range(len(output2)), key=lambda i: output2[i], reverse=True)[:100]
list_scores = [output2[i][0][0] for i in top]
TF=get_recommendation(top,df, list_scores)
TF
```

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

def count_vectorize(scraped_data, cv):
    # CountV the scraped data
    count_vectorizer = CountVectorizer()
    count_jobid = count_vectorizer.fit_transform(scraped_data) #fitting and transforming
the vector

    # CountV the cv
    user_count = count_vectorizer.transform(cv)
    cos_similarity_countv = map(lambda x: cosine_similarity(user_count, x),count_jobid)
    output3 = list(cos_similarity_countv)
    return output3
output3 = count_vectorize(df['All'], df2['All'])
output3

from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics.pairwise import cosine_similarity

count_vectorizer = CountVectorizer()
count_jobid = count_vectorizer.fit_transform(df['All']) #fitting and transforming the
vector
count_jobid

cos_similarity_countv = map(lambda x: cosine_similarity(user_count, x),count_jobid)
output3 = list(cos_similarity_countv)
```

```
output3

top = sorted(range(len(output3)), key=lambda i: output3[i], reverse=True)[:100]
list_scores = [output3[i][0][0] for i in top]
cv=get_recommendation(top, df, list_scores)
cv

from sklearn.neighbors import NearestNeighbors

def KNN(scraped_data, cv):
    tfidf_vectorizer = TfidfVectorizer(stop_words='english')

    n_neighbors = 100
    KNN = NearestNeighbors(n_neighbors, p=2)
    KNN.fit(tfidf_vectorizer.fit_transform(scraped_data))
#     NNs = KNN.kneighbors(tfidf_vectorizer.transform(cv), return_distance=True)
    NNs = KNN.kneighbors(tfidf_vectorizer.transform(cv))
    top = NNs[1][0][1:]
    index_score = NNs[0][0][1:]

    knn = get_recommendation(top, df, index_score)
    return knn

knn = KNN(df['All'], df2['All'])
knn

# Scale it
from sklearn.preprocessing import MinMaxScaler
```

```
slr = MinMaxScaler()
final[["KNN", "TF-IDF", 'CV']] = slr.fit_transform(final[["KNN", "TF-IDF", 'CV']])

# Multiply by weights
final['KNN'] = (1-final['KNN'])/3
final['TF-IDF'] = final['TF-IDF']/3
final['CV'] = final['CV']/3
final['Final'] = final['KNN']+final['TF-IDF']+final['CV']
final.sort_values(by="Final", ascending=False)
# final.to_csv('final.csv', index=False)
final2 = final.sort_values(by="Final", ascending=False).copy()
final2

# final2.merge(df, on="JobID")
final_salary = df.merge(final2, on="JobID")
final_salary
```

REFERENCES

[1] **“Resume Parser and Job Recommendation System using Machine Learning, 2024”** by Ashish Virendra Chandak; Hardik Pandey; Gourav Rushiya; Harsh Sharma

[2] **“Job Recommendation System using LinkedIn User Profiles, 2024”** by Govinda K; Meghanath Reddy K; Ananya Haldar

[3] **“Applying Machine Learning Algorithm to Optimize Personalized Education Recommendation System , 2024”** authors Wangmei Chen, Zepeng Shen, Yiming Pan, Kai Tan and Cankun Wang

[4] **“Job offers recommender system based on virtual organizations, 2022”** authors Alfonso González-Briones, Pablo Chamoso, Juan Pavón, Fernando De La Prieta, Juan M. Corchado

[5] **“Exploring Large Language Model for Graph Data Understanding in Online Job Recommendations, 2024”** authors Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu and Enhong Chen

[6] **“A Job Recommendation Model Based on a Two-Layer Attention Mechanism, 2024"** by Yu Mao ,Shaojie Lin and Yuxuan Cheng