

DAYANANDA SAGAR UNIVERSITY

KUDLU GATE, BANGALORE – 560068



Bachelor of Technology in COMPUTER SCIENCE AND ENGINEERING

Major Project Phase-II Report

“Personal Career Recommendation System”

By

Ch Revanth – ENG19CS0069

Ch Venkata Yaswanth –ENG19CS0072

Divya B – ENG19CS0087

K Sai Sumanth – ENG19CS0138

Under the supervision of

Prof. Veena M

Assistant Professor Dept. of Computer Science and Engineering

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING,
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE**

(2022-2023)



DAYANANDA SAGAR UNIVERSITY

School of Engineering
Department of Computer Science & Engineering
Kudlu Gate, Bangalore – 560068
Karnataka, India

CERTIFICATE

This is to certify that the Phase-II project work titled “**PERSONAL CAREER RECOMMENDATION SYSTEM**” is carried out by **Chaganti Revanth (ENG19CS0069), CH Venkata Yaswanth (ENG19CS0072), Divya.B (ENG19CS0087), K Sai Sumanth Reddy (ENG19CS0138)**, a bonafide students of Bachelor of Technology in Computer Science and Engineering at the School of Engineering, Dayananda Sagar University, Bangalore in partial fulfillment for the award of degree in Bachelor of Technology in Computer Science and Engineering, during the year **2022-2023**.

Prof. Veena M
Assistant Professor
Dept. of CS&E,
School of Engineering
Dayananda Sagar
University

Dr. Girisha G S
Chairman CSE
School of Engineering
Dayananda Sagar
University

**Dr. Udaya Kumar
Reddy K R**
Dean
School of Engineering
Dayananda Sagar
University

Date:

Date:

Date:

Name of the Examiner

- 1.
- 2.

Signature of Examiner

DECLARATION

We, **Chaganti Revanth (ENG19CS0069), CH Venkata Yaswanth (ENG19CS0072), Divya.B (ENG19CS0087), K Sai Sumanth Reddy (ENG19CS0138)**, are student's of eight semester B.Tech in **Computer Science and Engineering**, at School of Engineering, **Dayananda Sagar University**, hereby declare that the Major Project Stage-II titled "**PERSONAL CAREER RECOMMENDATION SYSTEM**" has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Technology in Computer Science and Engineering** during the academic year **2022-2023**.

Student

Signature

Name1: Chaganti Revanth
USN: ENG19CS0069

Name2: CH Venkata Yaswanth
USN: ENG19CS0072

Name3: Divya B
USN: ENG19CS0087

Name4: K Sai Sumanth Reddy
USN: ENG19CS0138

Place: Bangalore

Date:

ACKNOWLEDGEMENT

It is a great pleasure for us to acknowledge the assistance and support of many individuals who have been responsible for the successful completion of this project work.

First, we take this opportunity to express our sincere gratitude to School of Engineering & Technology, Dayananda Sagar University for providing us with a great opportunity to pursue our Bachelor's degree in this institution.

*We would like to thank **Dr. Udaya Kumar Reddy K R, Dean, School of Engineering & Technology, Dayananda Sagar University** for his constant encouragement and expert advice.*

*It is a matter of immense pleasure to express our sincere thanks to **Dr. Girisha G S, Department Chairman, Computer Science and Engineering, Dayananda Sagar University**, for providing right academic guidance that made our task possible.*

*We would like to thank our guide **Prof. Veena M, Assistant Professor, Dept. of Computer Science and Engineering, Dayananda Sagar University**, for sparing her valuable time to extend help in every step of our project work, which paved the way for smooth progress and fruitful culmination of the project.*

*We would like to thank our **Project Coordinator Dr. Meenakshi Malhotra and Dr. Pramod Kumar Naik** as well as all the staff members of Computer Science and Engineering for their support.*

We are also grateful to our family and friends who provided us with every requirement throughout the course.

We would like to thank one and all who directly or indirectly helped us in the Project work.

TABLE OF CONTENTS

	Page,no
LIST OF ABBREVIATIONS	V
LIST OF FIGURES	VI
LIST OF TABLES.....	VI
ABSTRACT	VII
CHAPTER 1 INTRODUCTION.....	1
1.1. INTRODUCTION.....	2
1.2 LANGUAGE USED IN OUR PROJECT.....	3
1.3 SCOPE.....	3
1.4 NOVELTY OF IDEA & SOCIETAL/ ENVIRONMENTAL IMPACT.....	3
CHAPTER 2 PROBLEM DEFINITION	5
CHAPTER 3 LITERATURE SURVEY.....	7
CHAPTER 4 PROJECT DESCRIPTION.....	10
CHAPTER 5 REQUIREMENTS	15
5.1. FUNCTIONAL REQUIREMENTS	15
5.2. NON - FUNCTIONAL REQUIREMENTS.....	15
5.3. HARDWARE AND SOFTWARE REQUIREMENTS.....	17
CHAPTER 6 METHODOLOGY	19
CHAPTER 7 EXPERIMENTATION.....	24
CHAPTER 8 RESULT AND DISCUSSION.....	31
8.1 RESULTS.....	32
REFERENCES.....	35
APPENDIX.....	36
SAMPLE CODE.....	36
FUNDING AND PAPER PUBLISHING.....	55
GITHUB LINK.....	56

NOMENCLATURE USED

SVM	Support Vector Machine
RF	Random Forest
XG Boost	Extreme Gradient Boost

LIST OF FIGURES

Fig. No.	Description of the figure	Page No.
4.1	Example of encoding	6
4.2	An example of Decision Tree	14
4.3	An example of SVM	15
4.4	An example of Random Forest	23
6.1	Use-case Diagram	24
6.2	Process Flow Diagram	25
7.3.1	Example of Decision Tree	29
7.3.2	Example of SVM	30
7.3.3	Example of Random Forest	31
8.1-8.3	Visualization of data	35
8.2	Web Page	36
8.3	Final Result	36
8.4	Course Recommendation	37

LIST OF TABLES

Table No.	Description of the table	Page No.
8.1	Test cases for the project	32

ABSTRACT

The increasing number of courses in education nowadays enables students to find their study programs of their interest. But some students end up with confusion to choose from variety of courses and not able to take the right decision for their career .

The Students Career Guidance software will help them out to choose the career path using data mining. Many students are open to suggestions or forced by others like parents, relatives but after they select a course that aren't really interested in them and then they are faced with different problems and as a result they don't excel in it or do it with interest.

This software intends to create a system that will recommend a course based on some basic details about the students like personal information, academic details, hobbies, co-curricular activities, extracurricular activities and other activities interests, background, aims of the specific person..

This Project can be used by the undergraduate students for choosing their suitable job role in their interested domain. It can also help pre-university students to analyze their abilities, so that they can enhance themselves if required, to suit themselves for the job role perfectly.

The output of our project is to develop a software system which will be able to predict the right career option for the students. This project will convert to patent and will be published

CHAPTER 1

INTRODUCTION

CHAPTER 1 INTRODUCTION

INTRODUCTION

Nowadays, students are often facing a dilemma in deciding to choose a career in their life. There are several factors that influence the students when choosing their career path such as their personal aptitudes, educational achievement and their environment. Upon completing their first degree or undergraduate students at the university, students are normally starting to consider a career path which may suit their skill and potential the best. However, many students made wrong decision on selecting their career due to the lack of experience, help and advice from friends and relatives, parents and lecturers, or career counselling. It is necessary to deliver career guidance in several ways like courses, trainings and seminars that offer group experiences in future career planning and group or individual counselling activities. The factor that may cause students not successful in their career is due to wrong choice of a job that suits with them. It requires decision-making process at an early stage. So this system gives recommendation for student about their career based on their abilities.

OBJECTIVES

The main objectives of the project are the following :

- To recommend a suitable career option (job role) to the users.
- To make recruitment system more secure and easier.
- To make the process of job hunt easier for freshers as well as experienced candidates.
- Help in analyzing themselves about their abilities.

Programming Language Used

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

SCOPE

- This Career prediction system mainly focuses on Computer Science and engineering domains.
- This work could be used by the job seekers.
- Could make recruitment process easier.

Novelty of idea & Societal/ environmental impact

- The system could be used to help individuals make more informed decisions about their future careers.
- The system could be used to help identify individuals who may be at risk for future unemployment or underemployment.
- The system could be used to help individuals who are struggling in their current careers to find new career paths that may be a better fit for them.

CHAPTER 2

PROBLEM DEFINITION

CHAPTER 2 PROBLEM DEFINITION

PROBLEM DEFINITION

The main problem of difficulty in choosing the right career by students is they do not know how to make decisions and also may be due to lack of knowledge and information about jobs and careers.

In order to help students in career selection, it is essential to build Career Recommendation System with a capacity to meet all the needs where it provide direction and guidance to students in choosing a career that suits with their skills and abilities.

Hence, we proposed the problem statement “**Personal Career Recommendation System**”.

OBJECTIVES

To achieve this project, we have the following steps:

- Pre-process the data
- User submits input data about themselves to the system
- The system uses machine learning algorithms to analyze the data and make predictions about the user's future career.
- To ensure that the system outputs the predictions to the user

Purpose

A career prediction system using machine learning can be used to predict an individuals future career path based on their skills, interests, and previous experience. This type of system is used to provide guidance to individuals who are exploring different career option, or to help identify potential career paths for individuals who are seeking a change.

CHAPTER 3

LITERATURE REVIEW

CHAPTER 3 LITERATURE REVIEW

Title: Career Recommendation for students based on deep learning and machine learning. – Hindawi.2022

In this paper the author had proposed a Career Recommendations for students based on Deep Learning and Machine Learning. A hybrid CNN model is proposed for employment suggestion of undergrads. And he concluded that by improving the activation function, pooling strategy, and loss function in the algorithm, the quality of model prediction is greatly improved.[1]

Title: Educational Career Recommendation System using Machine Learning. – 2021

In this paper, the author had built a educational Career Recommendation System Using Machine Learning. The issues of cold beginning, trust, and security are addressed in this approach to deal with utilize the earlier data from every school or college. The suggestion framework is done using Python since it is simple and productive to place into impact calculations on select working frameworks.[2]

Title: Career choice prediction based on campus big data, the potential behaviour of college students- 2020

In this paper, the author had created a career path Recommendation on frame work. A novel career path framework for personalized job and skills suggest youngs, focusing on students and young professionals is presented. And he concluded that the framework can run on cloud infrastructure using apache spark or Hadoop cluster which can handle more number of jobs description and more users.To increase the system capability at any time,new machines can be added to the existing cluster without affecting the running application.[3]

Title: A career path recommendation framework-a novel career path framework.

In this paper, the author had proposed a Student carrer prediction Student suggestion and it was proposed to help student career prediction in concluding there capacities in which they are solid and weak. And he used random forest as it contains various choice trees different subsets of the given datasets and taken the ordinary to deal with the accuracy of that datasets. It tends to be utilized for both Arrangements and Regression issue in ML[4]

Title: Student Career Prediction.-2022

In this paper, the author had proposed a Career Choice Prediction Based on Campus Big Data—Mining the Potential Behavior of College Students Proposed a prototypical cluster center generation approach to use the priori information from each college A noval regularization object was presented by overcome any barrier between this present reality models and prototypical bunch.[5]

CHAPTER 4

PROJECT DESCRIPTION

CHAPTER 4 PROJECT DESCRIPTION

Data Extraction

The first step in our project is to collect the data of students' marks, interests, skills. We have collected the data students from kaggle website and stored it in csv files.

Data Preprocessing

The next step in the project is processing of the data extracted. Data collected from various means will be in an unorganized format and there may be lot of null values, invalid data values and unwanted data. Cleaning all these data and replacing them with appropriate or approximate data and removing null and missing data and replacing them with some fixed alternate values are the basic steps in pre processing of data. The data of students details is directly sent for data cleaning. We read the data from all the columns, then combine and store it in a common file for easy processing and access.

The next step in the project is processing of the data extracted. The data of pollution and population is obtained year wise, hence it is directly sent for data cleaning. The temperature data is obtained day wise and the data is stored in separate files for each month from 2009 to 2020. We read the data from all the files, then combine and store it in a common file for easy processing and access.

Encoding

Encoding is a technique of converting categorical variables into numerical values so that it could be easily fitted to a machine learning model.

A machine learning algorithm needs to be able to understand the data it receives. For example, categories such as “small”, “medium”, and “large” need to be converted into numbers. To solve that, we can for example convert them into numeric labels with “1” for small, “2” for medium, and “3” for large.

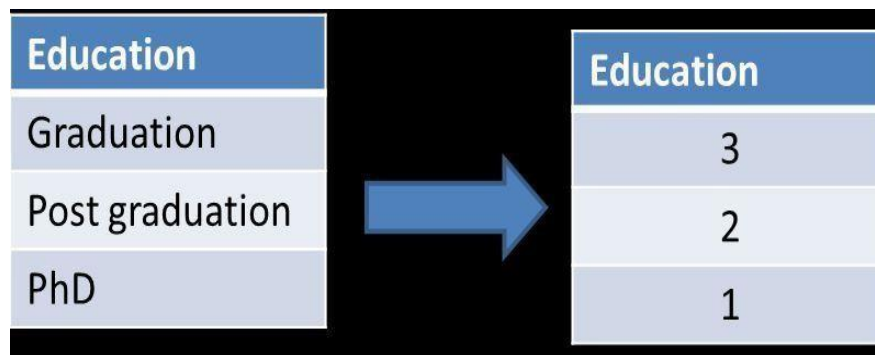


Figure 4.1 An example of Encoding

Decision Tree

Decision Tree is the most powerful and popular tool for classification and prediction. A Decision tree is a flowchart-like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.

A decision tree is a graphical representation of possible solutions to a problem based on certain conditions. It is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

A decision tree can be used in a career prediction system to predict an individual's future career path. The decision tree can be used to identify the most likely career paths based on the individual's skills, interests, and personality. The decision tree can also be used to identify the most likely career paths based on the individual's current career path.

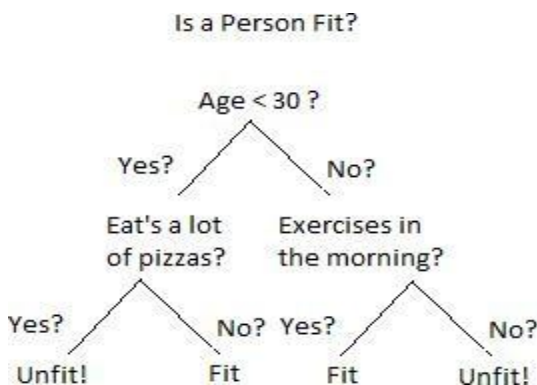


Figure 4.2 An example of Decision Tree

Support Vector Machine (SVM)

There are many different types of machine learning algorithms that could be used for career prediction, but support vector machines (SVMs) are a popular choice. SVMs are a type of supervised learning algorithm, which means they learn from labeled training data. In the context of career prediction, this would mean using data about people's past careers (or other relevant information) to train the SVM to predict future careers.

SVMs are known for being effective in high-dimensional spaces, which makes them well-suited for this task. They are also relatively robust to overfitting, meaning they can generalize well from the training data to new data.

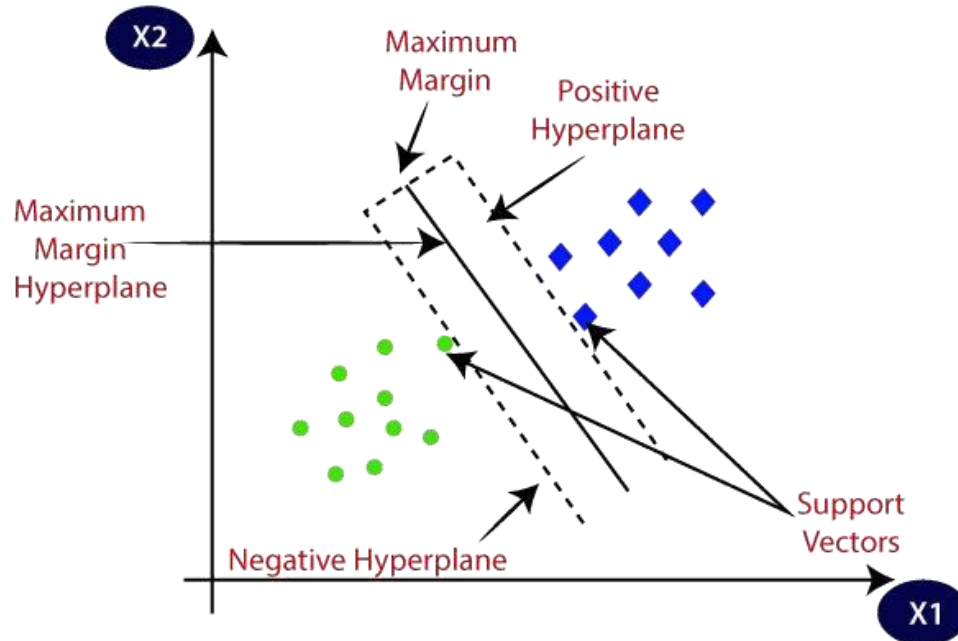


Figure 4.3 Support Vector Machine

Random forest

One advantage of using a random forest in a career prediction system is that it can handle a large number of input features without requiring them to be individually scaled. This is because the random forest algorithm is based on a decision tree, which is not sensitive to feature scaling. In addition, random Forests are also resistant to overfitting, meaning that they can still make accurate predictions on unseen data even if the training data is not representative of the entire population.

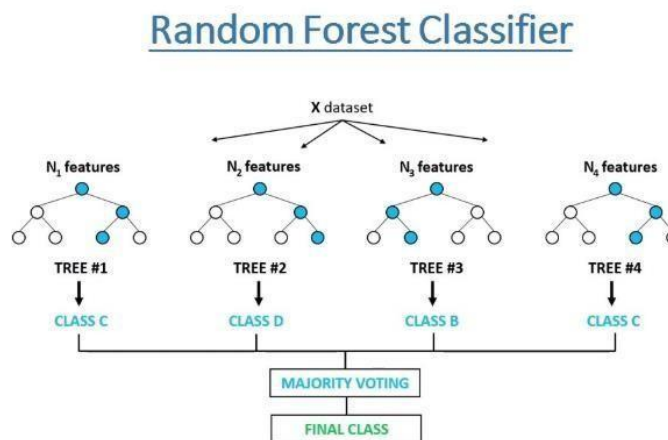


Figure 4.4 Support Vector Machine

XG BOOST

XGBOOST was introduced by TIANQI CHEN, a great machine learning library. It extends Gradient Boosting Machines, which has been conceded as one of the most effective supervised literacy algorithms. XGBoost stands for “Extreme Gradient Boosting,” according to the abstract. Gradient Boosting Machines are one of the stylish performing algorithms in supervised literacy, and XGBoost is an extension of them. XGBoost stands for “Extreme Gradient Boosting,” according to the abstract. The model and parameters are the abecedarian aspects of XGBoost, as it's a supervised literacy strategy. The model is a fine model that's used to prognosticate issues grounded on input values, and the parameters must be learned from the data set.

CHAPTER 5 REQUIREMENTS

CHAPTER 5 REQUIREMENTS

5.1 Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed; the general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

- The system firstly includes the collection of data
- Next, Pre-processing (cleaning) of the training data and testing data.
- Based on the train and test data, efficiency of the model will be analysed.
- The system will ask the user to enter the necessary details.
- The system will predict the career.

5.2 Non-Functional Requirements

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces.

User Non-Functional Requirements

System shall allow for users to get prediction for career within almost two mouse clicks.
System should ensure that features that require a user to be logged in.

System Non-Functional Requirements

System should be able to run with core functionality from computer system.

Performance

The performance of the system is mainly dependent on the internal working of the model accuracy.

Efficiency

Efficiency is a crucial aspect that is required in our project, because based on the accuracy of the results the future of the user has to be decided. Thus, its efficiency and accuracy is very important.

Reliability

The system must be reliable i.e., it must produce accurate results for different inputs as well as, the prediction must not change when the system is run on the same input data again.

Modifiability

In this field, lot of new algorithms are being discovered, the data is changing day by day, hence it very important that our system is modifiable and can adapt to the changes in the future so that it doesn't lose its functionality and its purpose in the future. The project is adaptable to changes, since the data is already processed and cleaned, hence any new algorithm can be implemented upon the same dataset.

5.3 Software/Hardware Requirements

Software Requirements

- Jupyter notebook
- Google collab
- Python
- Sikit Learn
- Matplotlib

Hardware Requirements

- Processor-i3 4th gen or high.
- RAM - 4GB or high.
- HDD – 256GB.

CHAPTER 6

METHODOLOGY

CHAPTER 6 METHODOLOGY

Among modeling languages Unified Modeling Language (UML) has become most popular. UML is commonly used in the design and implementation of any system and software architectures. To achieve functional and non-functional requirements of the system, UML model helps. In order to initiate the programming phase of building software, UML tools help in the creation of source code from UML diagram.

Introduction

In software industries, Object Oriented Development process is widely used. Object-Oriented Programming has heavily contributed toward a standardized method of modeling known as the Unified Modeling Language (UML). UML has become synonym for software modeling. UML is commonly used to model the software architecture as per the requirements and it includes a set of graphic notation techniques to create visual models of software-intensive systems. With the help of different UML diagrams for building the software, source code can be easily generated. The correctness of source code depends on the UML specification which needs to be standard, complete, precise, and unambiguous. A good UML specification leads to clearly defined semantics and an efficient code can be generated. The project is based upon the predicting temperature.

Personal Career Recommendation System Architecture

The system is developed in python. The system takes input from the datasets and produces the result.

The system building process consists of following sequential steps:

1. Fetching the dataset
2. Cleaning the dataset
3. Selection of the features of dataset
4. Train Model
5. Use the model to predict results.

Use-Case Diagram

An interaction between a user and a system is described by use case diagram. Use case diagrams describe what a system does from the standpoint of an external observer. The emphasis is on what a system does rather than how. Use case diagrams are closely connected to scenarios. A scenario is an example of what happens when someone interacts with the system. A use case diagram is a collection of actors, use cases, and their communications. For initial development we can use this use case. In this use case diagram, we can see following use cases and actor. Use cases are self-explanatory and they represent the main functions of Personal Career Recommendation System.

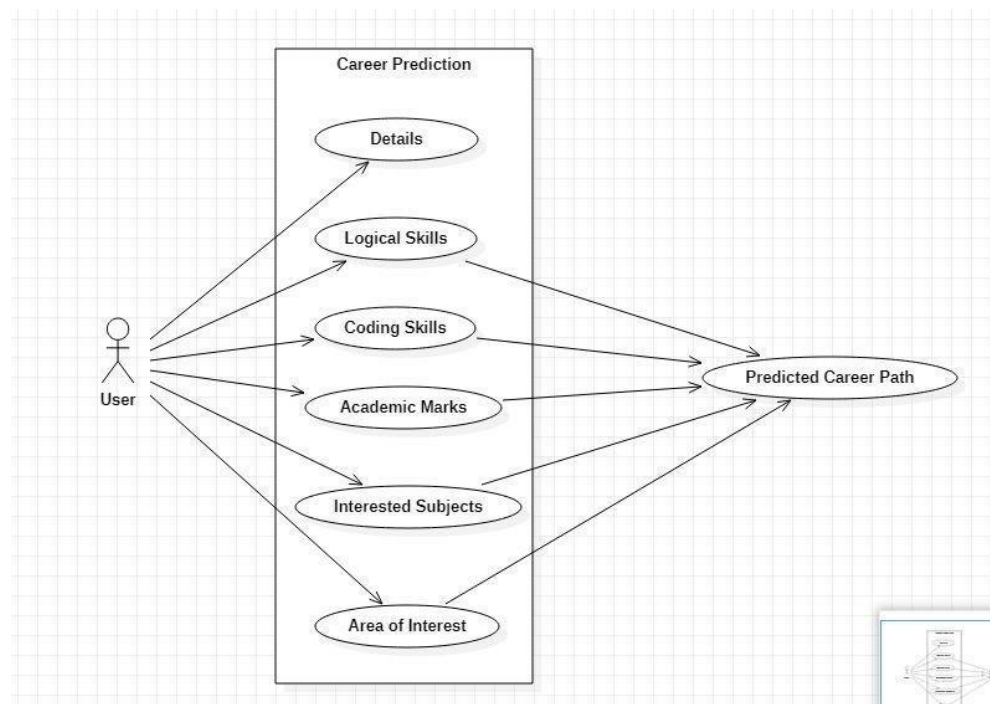


Fig 6.1 Use-case Diagram

Process Flow Diagram

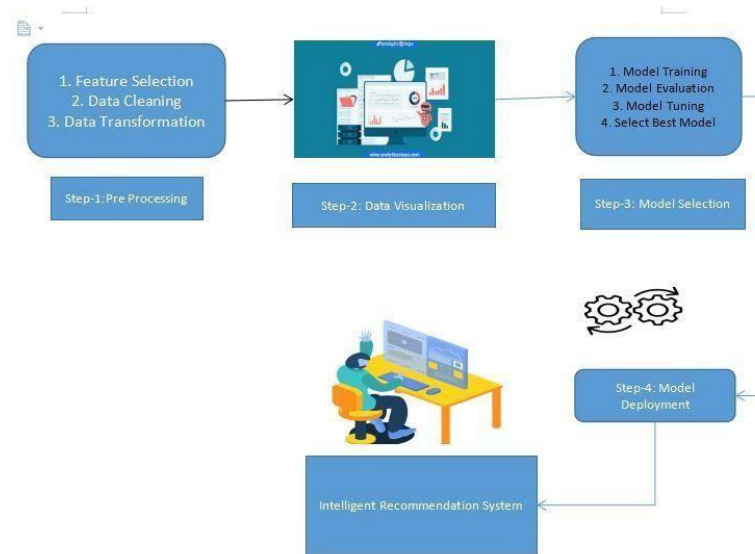


Fig 6.2 Process flow Diagram

Figure 6.4 shows the process flow when we were choosing the model for statistical analysis. First, we import the data, and then we process and clean the data. Corresponding to the data we fit a statistical model, and then we evaluate the model and cross-validate the model to ensure its functionality. We implement the selected model upon the data set. Finally, we produce a report based on the result.

CHAPTER 7

EXPERIMENTATION

CHAPTER 7 EXPERIMENTATION

Experimentation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. In other words, an implementation is a realization of a technical specification or algorithm as a program, software component, or other computer system through programming and deployment. Many implementations may exist for a given specification or standard.

Implementation is one of the most important phases of the Software Development Life Cycle (SDLC). It encompasses all the processes involved in getting new software or hardware operating properly in its environment, including installation, configuration, running, testing, and making necessary changes. Specifically, it involves coding the system using a particular programming language and transferring the design into an actual working system.

7.1 Steps for implementation

- Installation of Hardware and Software utilities.
- Testing the developed system with sampled data.
- Detection and correction of errors.

7.2 Implementation issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

7.3 Algorithms

7.3.1 Decision Tree

A decision tree which is also known as prediction tree refers a tree structure to mention the sequences of decisions as well as consequences.

Considering the input $X = (X_1, X_2, \dots, X_n)$, the aim is to predict a response or output variable Y . Each element in the set (X_1, X_2, \dots, X_n) is known as input variable. It is possible to achieve the prediction by the process of building a decision tree which has test points as well as branches.

At each test point, it is decided to select a particular branch and traverse down the tree. Ultimately, a final point is reached, and it will be easy to make prediction. In a decision tree, all the test points exhibit testing specific input variables (or attributes), and the developed decision tree is represented by the branches. Because of flexibility as well as simple visualization, decision trees are mostly probably deployed in data mining applications for the purpose of classification.

In the decision tree, the input values are considered as categorical or continuous. A structure of test points (known as nodes) and branches is established by the decision tree by which the decision being made will be represented. Leaf node is the one which do not have further branches. The returning value of leaf nodes is class labels while in some cases they return the probability scores.

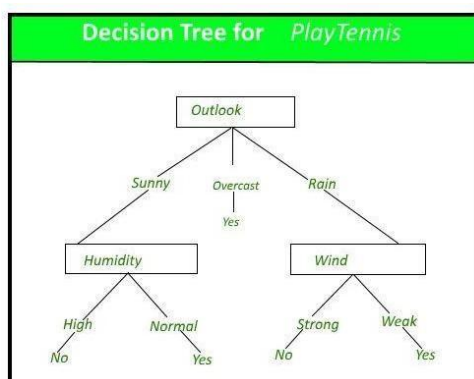


Fig 7.3.1 Example of Decision Tree

7.3.2 Support Vector Machine(SVM)

Support Vector Machine (SVM) is a relatively simple Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well.

Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line. In SVM, we plot each data item in the dataset in an N-dimensional space, where N is the number of features/attributes in the data. Next, find the optimal hyperplane to separate the data. So by this, you must have understood that inherently, SVM can only perform binary classification (i.e., choose between two classes).

However, there are various techniques to use for multi-class problems. Support Vector Machine for Multi-CLASS Problems To perform SVM on multi-class problems, we can create a binary classifier for each class of the data. The two results of each classifier will be :

The data point belongs to that class OR

The data point does not belong to that class.

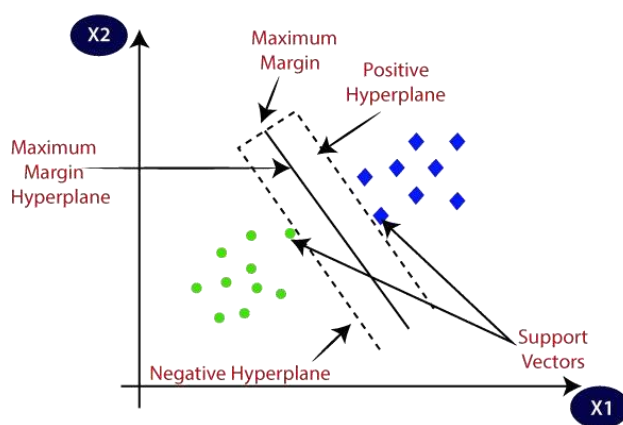


Fig 7.3.2 Example of SVM

7.4 Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

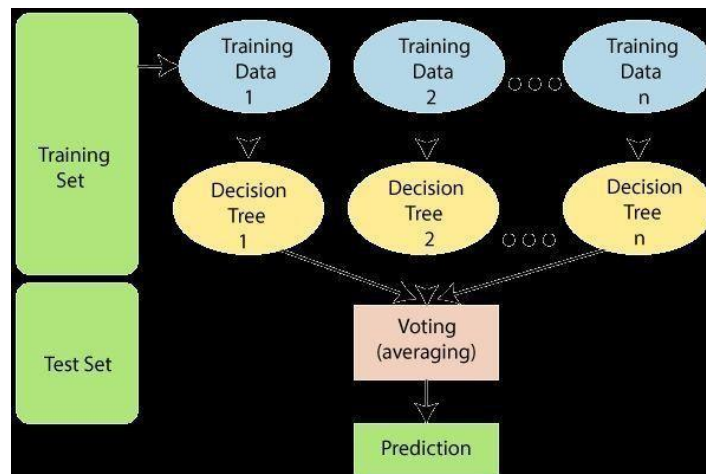


Fig 7.4.1 Example of Random Forest

7.5 XG Boost

XGBoost is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning method that combines the predictions of multiple weak models to produce a stronger prediction. XGBoost stands for “Extreme Gradient Boosting” and it has become one of the most popular and widely used machine learning algorithms due to its ability to handle large datasets and its ability to achieve state-of-the-art performance in many machine learning tasks such as classification and regression.

One of the key features of XGBoost is its efficient handling of missing values, which allows it to handle real-world data with missing values without requiring significant pre-processing. Additionally, XGBoost has built-in support for parallel processing, making it possible to train models on large datasets in a reasonable amount of time.

XGBoost can be used in a variety of applications, including Kaggle competitions, recommendation systems, and click-through rate prediction, among others. It is also highly customizable and allows for fine-tuning of various model parameters to optimize performance.

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.

7.5 Experiment Approach

For the experimentation, we have taken the mldata dataset.

Here, we have considered different features for the prediction of career such as level of coding skills, Number of hackathons attended, speaking skills, listening skills. These can be used to predict the best option.

CHAPTER 8

TESTING AND RESULTS

CHAPTER 8 TESTING AND RESULTS

Testing

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Testing Process

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested.

Testing Objectives

The main objectives of testing process are as follows.

- Testing is a process of executing a program with the intent of finding an error.
- A good test case is one that has high probability of finding undiscovered error.
- A successful test is one that uncovers the undiscovered error.

Test Cases

The test cases provided here test the most important features of the project.

Test cases for the project

Sl No	Test Input	Expected Results	Observed Results	Remarks
1	Unstructured data	Predicted values	Error	This test case failed because the uncleaned data produces errors and affects the algorithm functioning
2	Structured data	Predicted values	Predicted values	The test passed since the cleaned data can be used by the algorithm without any errors.

Table No:8.1

8.1 Results

This section describes the results obtained from our project “**Personal Career Recommendation System**”. The snapshots are shown below for each module.

From the ML dataset, we will visualise the data

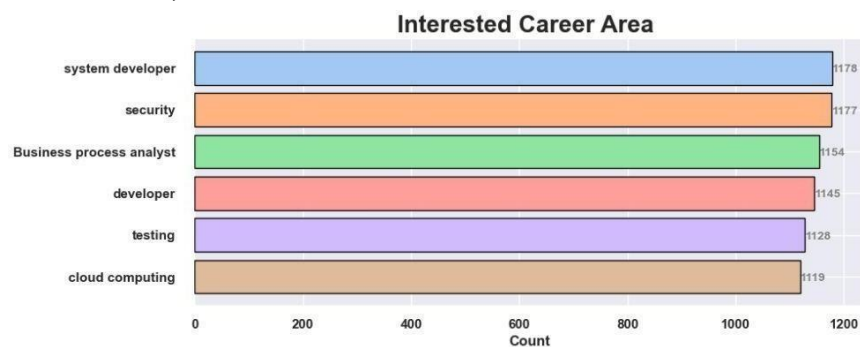


Fig 8.2 (a). Intrested domian

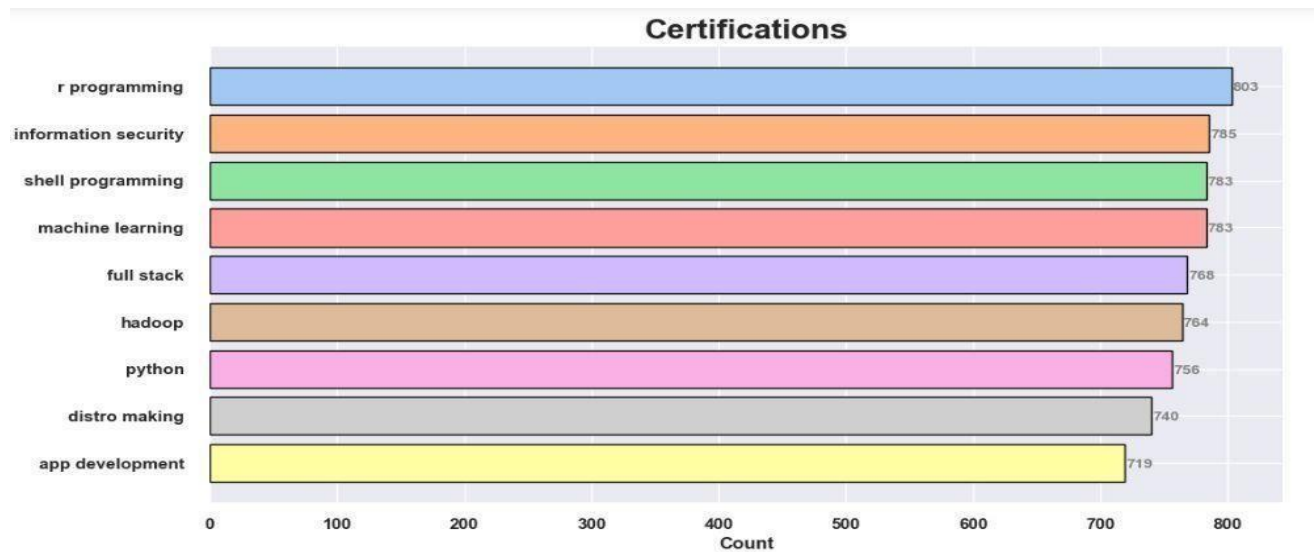


Fig 8.2 (b).Certifications

Results Of exploratory data analysis

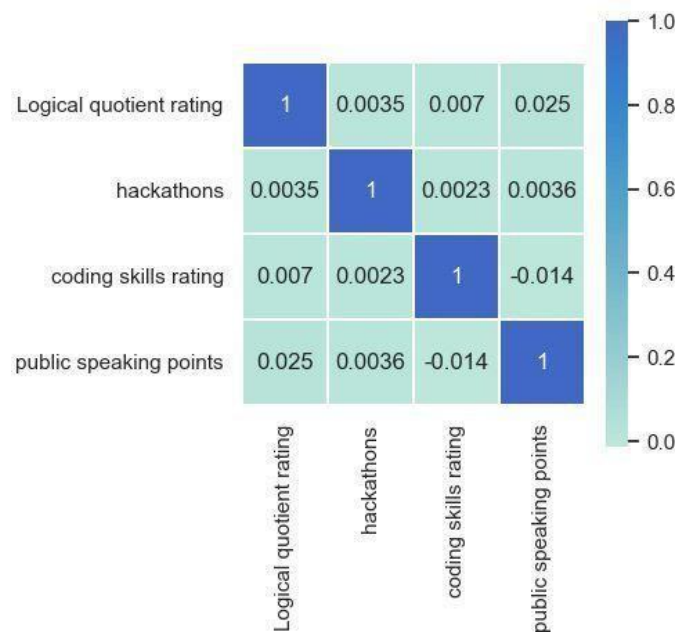


Fig 8.2 (b).Exploratory data analysis

Web Page Results

A web page is created as a front end, so that it would be user friendly. The outcome is as follows.

Fig 8.2 Web page

Recommendation Result

Fig.8.3 Final Result

Course Recommendation Result

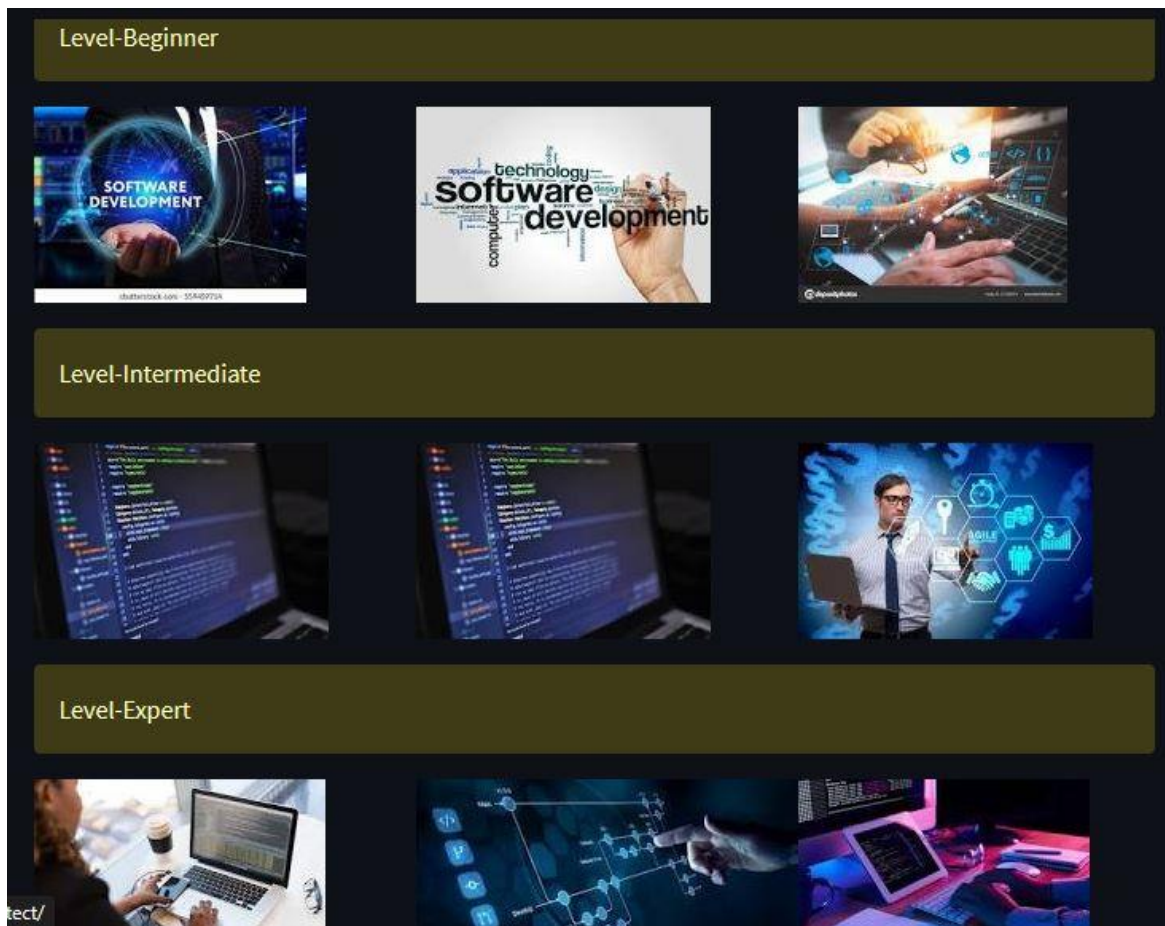


Fig.8.4 Course Recommendation

REFERENCES

- [1] “Wan, Qing, and Lin Ye. "Career Recommendation for College Students Based on Deep Learning and Machine Learning." Scientific Programming 2022 (2022).
- [2] Razak, Tajul Rosli, et al. "Career path recommendation system for UiTM Perlis students using fuzzy logic." 2014 5th International Conference on Intelligent and Advanced Systems (ICIAS). IEEE, 2014.
- [3] VidyaShreeram, N., and A. Muthukumaravel. "Student Career Prediction Using Decision Tree and Random Forest Machine Learning Classifiers." I3CAC 2021: Proceedings of the First International Conference on Computing, Communication and Control System, I3CAC 2021, 7-8 June 2021, Bharath University, Chennai, India. European Alliance for Innovation, 2021.
- [4] Deshpande, Samruddhi, et al. "Prediction of Suitable Career for Students using Machine Learning." International Research Journal of Engineering and Technology 8.02 (2021).
- [5] Chaudhary, Dileep, et al. "Student Future Prediction Using Machine Learning." International Journal of Scientific Research in Computer Science, Engineering and Information Technology (2019): 1104-1108.

APPENDIX

Sample code:

```
#Importing Libraries
```

```
import pandas as pd
```

```
import numpy as np
```

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

```
import seaborn as sns
```

```
import pickle
```

```
import time
```

```
import streamlit as st
```

```
from db import *
```

```
pickleFile=open("weights.pkl","rb")
```

```
regressor=pickle.load(pickleFile)
```

```
df = pd.read_csv('./data/mldata.csv')
```

```
df.head()
```

```
df['workshops'] = df['workshops'].replace(['testing'],'Testing')
```

```
df.head()
```

```
print(df.columns.unique)
```

```
n = df['Suggested Job Role'].unique()
print(len(n))
```

```
print('The shape of our training set: %s professionals and %s
features'%(df.shape[0],df.shape[1]))
```

```
## Binary Encoding for Categorical Variables
```

```
newdf = df
newdf.head(10)
```

```
cols = df[["self-learning capability?", "Extra-courses did", "Taken inputs from seniors
or elders", "worked in teams ever?", "Introvert"]]
```

```
for i in cols:
```

```
    print(i)
```

```
    cleanup_nums = {i: {"yes": 1, "no": 0}}
```

```
    df = df.replace(cleanup_nums)
```

```
print("\n\nList of Categorical features: \n" ,
df.select_dtypes(include=['object']).columns.tolist())
```

```
## Number Encoding for Categorical
```

```
mycol = df[["reading and writing skills", "memory capability score"]]
```

```

for i in mycol:

    print(i)

    cleanup_nums = {i: {"poor": 0, "medium": 1, "excellent": 2}}

    df = df.replace(cleanup_nums)


category_cols = df[['certifications', 'workshops', 'Interested subjects', 'interested career
area ', 'Type of company want to settle in?',
                    'Interested Type of Books']]

for i in category_cols:

    df[i] = df[i].astype('category')

    df[i + "_code"] = df[i].cat.codes


print("\n\nList of Categorical features: \n" ,
df.select_dtypes(include=['object']).columns.tolist())


## Dummy Variable Encoding


print(df['Management or Technical'].unique())
print(df['hard/smart worker'].unique())


df = pd.get_dummies(df, columns=["Management or Technical", "hard/smart worker"],
prefix=["A", "B"])

df.head()


df.sort_values(by=['certifications'])

```

```
print("List of Numerical features: \n" ,
df.select_dtypes(include=np.number).columns.tolist())
```

```
category_cols = df[['certifications', 'workshops', 'Interested subjects', 'interested career
area ', 'Type of company want to settle in?', 'Interested Type of Books']]
```

```
for i in category_cols:
```

```
    print(i)
```

```
Certifi = list(df['certifications'].unique())
```

```
print(Certifi)
```

```
certi_code = list(df['certifications_code'].unique())
```

```
print(certi_code)
```

```
Workshops = list(df['workshops'].unique())
```

```
print(Workshops)
```

```
Workshops_code = list(df['workshops_code'].unique())
```

```
print(Workshops_code)
```

```
Certi_1 = list(df['certifications'].unique())
```

```
certi_code = list(df['certifications_code'].unique())
```

```
C = dict(zip(Certi_1, certi_code))
```

```
Workshops = list(df['workshops'].unique())
```

```
print(Workshops)

Workshops_code = list(df['workshops_code'].unique())

print(Workshops_code)

W = dict(zip(Workshops,Workshops_code))


Interested_subjects = list(df['Interested subjects'].unique())

print(Interested_subjects)

Interested_subjects_code = list(df['Interested subjects_code'].unique())

ISC = dict(zip(Interested_subjects,Interested_subjects_code))


interested_career_area = list(df['interested career area '].unique())

print(interested_career_area)

interested_career_area_code = list(df['interested career area _code'].unique())

ICA = dict(zip(interested_career_area,interested_career_area_code))


Typeofcompany = list(df['Type of company want to settle in?'].unique())

print(Typeofcompany)

Typeofcompany_code = list(df['Type of company want to settle in?_code'].unique())

TOCO = dict(zip(Typeofcompany,Typeofcompany_code))


Interested_Books = list(df['Interested Type of Books'].unique())

print(Interested_subjects)

Interested_Books_code = list(df['Interested Type of Books_code'].unique())

IB = dict(zip(Interested_Books,Interested_Books_code))
```



```
Range_dict = {"poor": 0, "medium": 1, "excellent": 2}
print(Range_dict)
```

```
A = 'yes'
B = 'No'
col = [A,B]
for i in col:
    if(i=='yes'):
        i = 1
    print(i)
```

```
f=[]
A = 'r programming'
clms = ['r programming',0]
for i in clms:
    for key in C:
        if(i==key):
            i = C[key]
            f.append(i)
print(f)
```

```
C = dict(zip(Certifi,certi_code))
```

```
print(C)
```

```
import numpy as np
```

```
array = np.array([1,2,3,4])
```

```
array.reshape(-1,1)
```

```
def inputlist(Name,Contact_Number,Email_address,
    Logical_quotient_rating, coding_skills_rating, hackathons,
    public_speaking_points, self_learning_capability,
    Extra_courses_did, Taken_inputs_from_seniors_or_elders,
    worked_in_teams_ever,Introvert, reading_and_writing_skills,
    memory_capability_score, smart_or_hard_work, Management_or_Technical,
    Interested_subjects, Interested_Type_of_Books,certifications, workshops,
    Type_of_company_want_to_settle_in, interested_career_area):
```

```
Afeed = [Logical_quotient_rating, coding_skills_rating, hackathons,
public_speaking_points]
```

```
input_list_col =
[self_learning_capability,Extra_courses_did,Taken_inputs_from_seniors_or_elders,wo
rked_in_teams_ever,Introvert,reading_and_writing_skills,memory_capability_score,s
mart_or_hard_work,Management_or_Technical,Interested_subjects,Interested_Type_
of_Books,certifications,workshops,Type_of_company_want_to_settle_in,interested_ca
reer_area]
```

```
feed = []
```

```
K=0
```

```
j=0
for i in input_list_col:
    if(i=='Yes'):
        j=2
        feed.append(j)

        print("feed 1",i)

    elif(i=="No"):
        j=3
        feed.append(j)

        print("feed 2",j)

    elif(i=='Management'):
        j=1
        k=0
        feed.append(j)
        feed.append(K)

        print("feed 10,11",i,j,k)

    elif(i=='Technical'):
        j=0
        k=1
```

```
feed.append(j)
```

```
feed.append(K)
```

```
print("feed 12,13",i,j,k)
```

```
elif(i=='Smart worker'):
```

```
    j=1
```

```
    k=0
```

```
    feed.append(j)
```

```
    feed.append(K)
```

```
print("feed 14,15",i,j,k)
```

```
elif(i=='Hard Worker'):
```

```
    j=0
```

```
    k=1
```

```
    feed.append(j)
```

```
    feed.append(K)
```

```
print("feed 16,17",i,j,k)
```

```
else:
```

```
    for key in Range_dict:
```

```
        if(i==key):
```

```
            j = Range_dict[key]
```

```
            feed.append(j)
```

```
print("feed 3",i,j)
```

```
for key in C:
```

```
    if(i==key):
```

```
        j = C[key]
```

```
        feed.append(j)
```

```
print("feed 4",i,j)
```

```
for key in W:
```

```
    if(i==key):
```

```
        j = W[key]
```

```
        feed.append(j)
```

```
print("feed 5",i,j)
```

```
for key in ISC:
```

```
    if(i==key):
```

```
        j = ISC[key]
```

```
        feed.append(j)
```

```
print("feed 6",i,j)
```

```
for key in ICA:
```

```
if(i==key):  
    j = ICA[key]  
    feed.append(j)  
  
    print("feed 7",i,j)
```

```
for key in TOCO:  
    if(i==key):  
        j = TOCO[key]  
        feed.append(j)  
  
        print("feed 8",i,j)
```

```
for key in IB:  
    if(i==key):  
        j = IB[key]  
        feed.append(j)  
  
        print("feed 9",i,j)
```

```
t = Afeed+feed  
output = regressor.predict([t])
```

```
return(output)
```

```
def main():
```

```
    html1="""
```

```
        <div style="text-align:center;">
```

```
            <h1>        Personal Career Recommendation System        </h1>
```

```
        </div>
```

```
    """
```

```
    st.markdown(html1,unsafe_allow_html=True)
```

```
    col1, col2, col3 = st.columns(3)
```

```
    with col1:
```

```
        st.image("./images/image1.png")
```

```
    with col2:
```

```
        st.image("./images/image2.png")
```

```
    with col3:
```

```
        st.image("./images/image3.png")
```

```
st.sidebar.title("Your Information")
```

```
Name = st.sidebar.text_input("Full Name")
```

```
Contact_Number = st.sidebar.text_input("Contact Number")
```

```
Email_address = st.sidebar.text_input("Email address")
```

```
if not Name and Contact_Number and Email_address:
```

```
    st.sidebar.Error("Please fill out your name and EmailID")
```

```
if Name and Contact_Number and Email_address:
```

```
    st.sidebar.success("Thanks!")
```

```
Logical_quotient_rating = st.slider(
```

```
    'Rate your Logical quotient Skills', 0,10)
```

```
st.write(Logical_quotient_rating)
```

```
coding_skills_rating =
```

```
st.slider( 'Rate your Coding Skills',
```

```
0,10) st.write(coding_skills_rating)
```

```
hackathons = st.slider(
```


'Enter number of Hackathons participated',0,10)

st.write(hackathons)

public_speaking_points =

st.slider('Rate Your Public Speaking',

0,10,1)

st.write(public_speaking_points)

self_learning_capability =

st.selectbox('Self Learning

Capability',

('Yes', 'No')

)

Extra_courses_did =

st.selectbox('Extra courses',

('Yes', 'No')

)

Taken_inputs_from_seniors_or_elders =

st.selectbox('Took advice from seniors or elders',

('Yes', 'No')

)

```
worked_in_teams_ever =
```

```
    st.selectbox('Team Co-ordination
```

```
    Skill',
```

```
    ('Yes', 'No')
```

```
    )
```

```
Introvert =
```

```
    st.selectbox('Introvert',
```

```
    ('Yes', 'No')
```

```
    )
```

```
reading_and_writing_skills =
```

```
    st.selectbox('Reading and writing skills',
```

```
    ('poor','medium','excellent')
```

```
    )
```

```
st.write('You selected: **{}**'.format(reading_and_writing_skills))
```

```
memory_capability_score =
```

```
    st.selectbox('Memory capability score',
```

```
    ('poor','medium','excellent')
```

```
    )
```

```
st.write('You selected: **{}**'.format(memory_capability_score))
```

```
smart_or_hard_work =
```

```

st.selectbox('Smart or Hard Work',
('Smart worker', 'Hard Worker')
)
st.write('You selected: ** {} **'.format(smart_or_hard_work))

```

```

Management_or_Technical =
st.selectbox('Management or Techninical',
('Management', 'Technical')
)
st.write('You selected: ** {} **'.format(Management_or_Technical))

```

```

Interested_subjects =
st.selectbox('Interested Subjects',
('programming', 'Management', 'data engineering', 'networks', 'Software
Engineering', 'cloud computing', 'parallel computing', 'IOT', 'Computer Architecture',
'hacking')
)
st.write('You selected: ** {} **'.format(Interested_subjects))

```

```

Interested_Type_of_Books =
st.selectbox('Interested Books Category',
('Series', 'Autobiographies', 'Travel', 'Guide', 'Health', 'Journals', 'Anthology',
'Dictionaries', 'Prayer books', 'Art', 'Encyclopedias', 'Religion-Spirituality', 'Action and
Adventure', 'Comics', 'Horror', 'Satire', 'Self help', 'History', 'Cookbooks', 'Math',
'Biographies', 'Drama', 'Diaries', 'Science fiction', 'Poetry', 'Romance', 'Science',
'Trilogy', 'Fantasy', 'Childrens', 'Mystery')

```

```
)
st.write('You selected: **{**'.format(Interested_Type_of_Books))
```

```
certifications =
```

```
st.selectbox('Interested_Type_of_Books',
('information security', 'shell programming', 'r programming', 'distro making',
'machine learning', 'full stack', 'hadoop', 'app development', 'python')
)
st.write('You selected: **{**'.format(certifications))
```

```
workshops =
```

```
st.selectbox('Workshops Attended',
('Testing', 'database security', 'game development', 'data science', 'system designing',
'hacking', 'cloud computing', 'web technologies')
)
st.write('You selected: **{**'.format(workshops))
```

```
Type_of_company_want_to_settle_in =
```

```
st.selectbox('Type of Company You Want to Settle In ',
('BPA', 'Cloud Services', 'product development', 'Testing and Maintainance Services',
'SaaS services', 'Web Services', 'Finance', 'Sales and Marketing', 'Product based',
'Service Based')
)
```

```

st.write('You selected: **{}**'.format(Type_of_company_want_to_settle_in))

interested_career_area =

st.selectbox('Interested Career Area',
('testing', 'system developer', 'Business process analyst', 'security', 'developer', 'cloud
computing')
)

st.write('You selected: **{}**'.format(interested_career_area))

result=""

if st.button("Predict"):

    result=inputlist(Name,Contact_Number,Email_address,Logical_quotient_rating,
coding_skills_rating, hackathons,
                    public_speaking_points, self_learning_capability,Extra_courses_did,
                    Taken_inputs_from_seniors_or_elders,worked_in_teams_ever, Introvert,
                    reading_and_writing_skills,memory_capability_score,
smart_or_hard_work,
                    Management_or_Technical,Interested_subjects,
Interested_Type_of_Books,
                    certifications, workshops, Type_of_company_want_to_settle_in,
interested_career_area)

my_bar = st.progress(0)

for percent_complete in range(100):

```

```

time.sleep(0.05)

my_bar.progress(percent_complete + 1)


st.success("Predicted Career Option : "
          "{}".format(result))

create_table()

add_data(Name,Contact_Number,Email_address,Logical_quotient_rating,
coding_skills_rating, hackathons,
        public_speaking_points, self_learning_capability,Extra_courses_did,
        Taken_inputs_from_seniors_or_elders,worked_in_teams_ever, Introvert,
        reading_and_writing_skills,memory_capability_score, smart_or_hard_work,
        Management_or_Technical,Interested_subjects, Interested_Type_of_Books,
        certifications, workshops, Type_of_company_want_to_settle_in,
interested_career_area)


html3="""

<div style="color:red; margin:80px; text-align:center;">

    Developed by TEAM-76

</div>

"""

st.markdown(html3,unsafe_allow_html=True)

if __name__=='__main__':

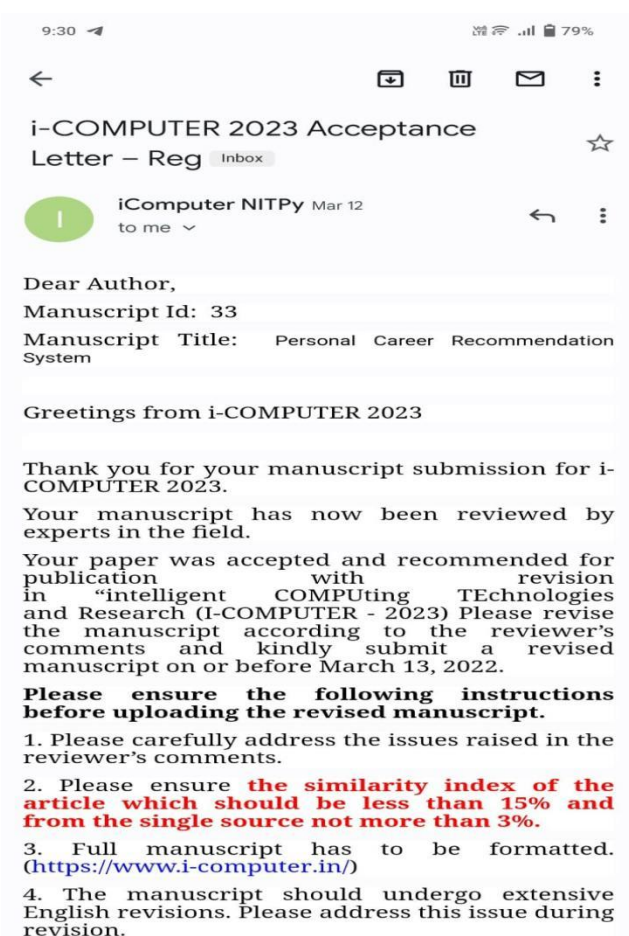
    main()

```

FUNDING AND PAPER PUBLISHING:

Our team applied for a publication at the renowned university of NIT Puducherry and got an acceptance letter for publishing our full-length paper in a journal as part of the iComputer conference. Our paper was formatted according to the template provided by the organizers of the conference. Our team attended the conference as per the instructed schedule and presented our paper at the conference as well.

We have presented the paper to **International Conference on Intelligent Computing Technology & Research (i-COMPUTER 2023)** held on 24th & 25th March 2023, Department of Computer Science & Engineering, NIT Puducherry, Karaikal. The paper will be published.



GITHUB LINK:

https://github.com/CSE-DSU/Team_76_Personal-Career-Recommendation-System.git