



# SUPERIOR UNIVERSITY LAHORE

## GOLD CAMPUS

**NAME:** MUHAMMAD MOIZ RAZA

**ROLL NO:** SU92-BSAIM-F24-061

**PROGRAM:** ARTIFICIAL INTELLIGENCE

**SEMESTER:** 3<sup>rd</sup> SEMESTER

**SECTION:** BSAI-3A

**SUBJECT:** ARTIFICIAL INTELLIGENCE

**Submission Title:** PROJECT

✓ COURSE PREDICTOR

**Submitted To:** Sir RASHIK ALI

## 1. Introduction

The aim of this project is to build a **Course Recommendation System** that suggests the most suitable course to a user based on their personal preferences. The system uses **Machine Learning** to make accurate predictions.

The recommendation is based on the following user inputs:

- Interests
- Category
- Preferred Difficulty Level
- Past Experience
- Preferred Language

The output of the model is the **Recommended Course**.

## 2. Dataset Description

Dataset File Name: recomdata.csv

Total Records: **5000**

Total Columns: **6**

**Dataset Properties:**

- All columns are of **categorical (object) type**
- **No missing (null) values**
- Dataset is clean and well-structured

## 3. Libraries Used

The following Python libraries were used in this project:

Library	Purpose
pandas	Data loading and manipulation
matplotlib	Data visualization
seaborn	Heatmap and correlation visualization
scikit-learn	Model training, preprocessing, and evaluation
Pickle	Saving the trained model

## 4. Data Loading

The dataset was loaded using pandas:

- df.head()
- df.tail()
- df.shape
- df.columns
- df.describe()
- df.info()

These functions helped in understanding:

- Number of rows and columns
- Data types of each column
- Summary statistics
- Overall structure of the dataset

## 5. Exploratory Data Analysis (EDA)

EDA was performed to understand the data distribution and relationships between features.

**Purpose of EDA:**

- To understand data patterns
- To analyze feature relationships
- To detect outliers
- To study data distribution

## 6. Data Preprocessing

### **Handling Missing Values**

Although the dataset had no missing values, a function was created to handle missing values in case they appear in future

### **Encoding Categorical Data**

Since Machine Learning models cannot work with text data directly, **Label Encoding (Factorization)** was applied

## 7. Feature and Target Selection

```
X = df.drop("recommended_course", axis=1)
```

```
y = df["recommended_course"]
```

X → Input Features

y → Output / Target Variable

## 8. Train-Test Split

The dataset was divided into training and testing sets to evaluate model performance

```
X_train, X_test, y_train, y_test = train_test_split(  
    X, y, test_size=0.2, random_state=42  
)
```

**80% data** used for training

**20% data** used for testing

## 9. Model Used

### Random Forest Classifier

Random Forest is an **ensemble learning algorithm** that combines multiple decision trees to improve prediction accuracy and reduce overfitting.

## 10. Prediction and Model Evaluation

- The model achieved **high accuracy**
- The predictions are reliable and stable
- The model generalizes well on unseen data

## 11. Model Saving

The trained model was saved using **pickle** so it can be reused later for deployment

## 12. FLASK IMPLEMENTATION

After train a model. Now we make a front end on flask with stylish look in which we enter our experience and interest then we get our recommended course for GUI we use flask.

### **STRUCTURE:**

1. app.py
2. Template folder
  - Index.html
  - Index.html
  - Result.html
3. Static folder
  - Style.css
  - Script.js
4. Requirements.txt
5. Model\_train.ipynb file
6. Model.pkl
7. Venv file

**App.py:** This is our main entry point of flask application

**Template/index.html:** This is our home page of website

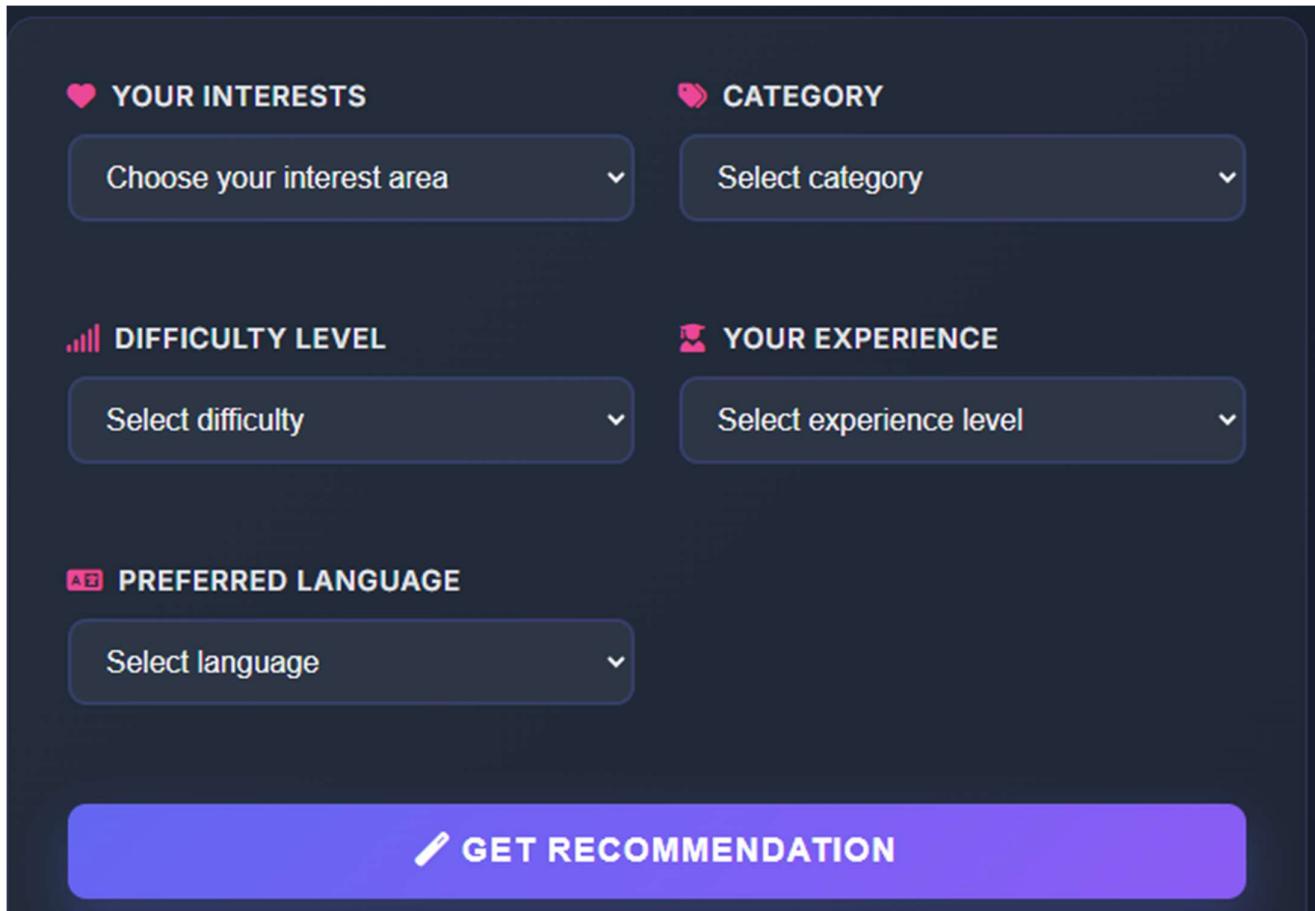
**Template/result.html:** this is another page in which our predicted score shows

**Style.css:** This is my css file of Gui in which all style and theme of my Website Written

**Requirement.txt:** In this file our libraries which is use in our project is written it is use to install libraries in virtual env

**Model\_train.ipynb file:** In this File we train our model from dataset

**Model.pkl:** In this file our trained model is saved.



The recommendation is based on the following user inputs:

- Interests
- Category
- Preferred Difficulty Level
- Past Experience
- Preferred Language

YOUR INTERESTS

C++ ▾

CATEGORY

data science ▾

DIFFICULTY LEVEL

advanced ▾

YOUR EXPERIENCE

none ▾

PREFERRED LANGUAGE

English ▾

GET RECOMMENDATION

We put all his fields that is necessary for recommendation

Your Perfect Course Match!

Based on your preferences, here's our AI-powered recommendation

RECOMMENDED COURSE

**Big Data advanced English**

Confidence Score: **87.0%**

YOUR PREFERENCES

INTERESTS C++	CATEGORY data science	DIFFICULTY advanced
EXPERIENCE none	LANGUAGE English	

This is result page of our project in which he recommend best suitable course.

## **Result**

- The dataset was clean and well-balanced.
- Proper EDA improved understanding of the data.
- Label encoding successfully converted categorical data.
- Random Forest performed very well with **high prediction accuracy**.
- The model was successfully trained and stored for future use.

## **Future Scope**

The system can be further improved by:

- Using Deep Learning models
- Deploying it as a web application using Flask or Streamlit
- Adding real-time user feedback
- Using collaborative filtering
- Connecting it to a real online course database

## **Conclusion**

- This project successfully demonstrates how **Machine Learning can be used to build an intelligent Course Recommendation System**. The system analyzes user preferences and suggests the most suitable course using a trained Random Forest model.
- All key machine learning steps were implemented:
  - Data loading
  - Exploratory Data Analysis
  - Data preprocessing
  - Train-test split
  - Model training
  - Model evaluation
  - Model saving
- This system can be effectively used in:
  - E-learning platforms
  - Online education portals

- Skill development websites

THE END