DAYANANDA SAGAR UNIVERSITY

Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka – 562112

SCHOOL OF ENGINEERING

Bachelor of Technology
in
COMPUTER SCIENCE AND ENGINEERING
(Artificial Intelligence and Machine Learning)

Mini Project

on

INTELLIGENT COURSE RECOMMENDATION SYSTEM

By
Ethan Saju – ENG22AM0175
Ruben Santosh – ENG22AM0162
Vignesh R Nair – ENG22AM0142
Soundarya S Jois – ENG22AM0134

Under the supervision of
Prof. Pradeep Kumar K
Dr. Mary Jasmine
Prof. Mitha Guru

Assistant Professor, Artificial Intelligence & Machine Learning, SOE
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence and Machine Learning)
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY,
BANGALORE

SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
(Artificial Intelligence and Machine Learning)
Devarakaggalahalli, Harohalli Kanakapura Road, Dt, Ramanagara, Karnataka – 562112

CERTIFICATE

This is to certify that the Mini – Project titled **"INTELLIGENT COURSE RECOMMENDATION SYSTEM"** is carried out by **Ethan Saju (ENG22AM0175), Soundarya S Jois (ENG22AM0134), Vignesh R Nair (ENG22AM0142), Ruben Santosh (ENG22AM0162),** bona fide students of Bachelor of Technology in Computer Science and Engineering (Artificial Intelligence and Machine Learning) at the School of Engineering, Dayananda Sagar University.

Prof. Pradeep Kumar K
Assistant Professor
Dept. of CSE (AI&ML),
School of Engineering
Dayananda Sagar University
Date: 01-01-2024

Dr. Jayavrinda Vrindavanam V
Chairman
Dept CSE (AI&ML)
School of Engineering
Dayananda Sagar University
Date: 01-01-2024

Prof. Mithaguru
Assistant Professor
Dept. of CSE(AI&ML),
School of Engineering
Dayananda Sagar University,
Date: 01-01-2024

Dr. Mary Jasmine
Assistant Professor
Dept of CSE(AI&ML)
School of Engineering,
Dayananda Sagar University,
Date:01-01-2024

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**, **Dayananda Sagar University** for his constant encouragement and expert advice. It is a matter of immense pleasure to express our sincere thanks to **Dr. Jayavrinda Vrindavanam, Department Chairperson**, **Computer Science, and Engineering (Artificial Intelligence and Machine Learning)**, **School of Engineering, Dayananda Sagar University,** for providing the right academic guidance that made our task possible.

We would like to thank our guide **- Pradeep Kumar K**, **Dr. Mary Jasmine**
Prof. Mitha Guru, **Assistant Professor**, **Dept. of Computer Science and Engineering (Artificial Intelligence and Machine Learning)**, **School of Engineering, Dayananda Sagar University**, for sparing his valuable time to extend help in every step of our UG mini project work, which paved the way for smooth progress and the fruitful culmination of the research.

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 mini project.

DECLARATION

We, **Soundarya S Jois , Ethan Saju , Vignesh R Nair , Ruben Santosh** students of third semester B. Tech in **Computer Science and Engineering with speciation in Artificial intelligence and machine learning** , at School of Engineering, **Dayananda Sagar University**, hereby declare that the **AI Mini Project** titled **"Intelligent Course Recommendation System"** hasbeen carried out by us and submitted to Prof. Pradeep Kumar during the academic year **2023-2024**.

Student 1: **Signature**

Name: VIGNESH R NAIR

USN: ENG22AM0142

Student 2: **Signature**

Name: SOUNDARYA S JOIS

USN: ENG22AM0134

Student 3: **Signature**

Name: RUBEN SANTOSH

USN: ENG22AM0162

Student 4: **Signature**

Name: ETHAN SAJU

USN: ENG22AM0175

Place: Bangalore
Date: 01-01-2024

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
DL	Deep Learning
GUI	Graphical User Interface
PHP	Pre-Processor Hyper text

ABSTRACT

This mini project is centered on developing an innovative student course recommendation system, a domain of increasing relevance in the evolving landscape of education technology. In the face of an expanding array of academic courses, students often encounter challenges in selecting courses that align with their personal and professional aspirations. This system aims to mitigate such difficulties by providing personalized course recommendations, tailored to individual student profiles. Leveraging the capabilities of Artificial Intelligence (AI), the project addresses the critical need for guidance in educational paths, enhancing the decision-making process for students in selecting courses that best fit their academic level, field of study, and specific interests.

The cornerstone of this project is the deployment of a decision tree classifier, a fundamental algorithm in the realm of machine learning. This algorithm is particularly suited for the task due to its ability to map out complex decision paths based on input features. In this context, these features include the student's level of education, year of study, main field of the course, and specific interests. The decision tree classifier is adept at handling both categorical and numerical data, making it ideal for processing the diverse range of inputs necessary for effective course recommendation.

Utilizing a dataset provided in a CSV format, which encompasses a comprehensive list of courses along with their corresponding details, the decision tree classifier operates by creating a model that correlates a student's profile with potential course selections. This process involves training the model on a subset of the dataset and validating its accuracy and reliability on another subset. The significance of using a decision tree lies in its transparency and ease of interpretation. The decision-making process can be visualized and understood, allowing for adjustments and optimizations based on specific educational contexts.

The outcome of this mini project is a robust, AI-driven course recommendation system that stands to revolutionize how students approach their course selection process. By providing tailored course suggestions that resonate with individual educational backgrounds and interests, the system not only enhances the academic experience but also contributes to more informed and strategic decision-making in educational pursuits. This project serves as a testament to the transformative potential of AI in education, offering a model that can be scaled and adapted to various educational settings and student demographics.

CHAPTER 1 - INTRODUCTION

In the dynamic landscape of educational choices, navigating the myriad of available courses can be daunting for students. Enter the revolutionary Student Course Recommendation System, a beacon of guidance in this sea of options. This system stands as a testament to the power of technology in enhancing educational experiences. By leveraging advanced algorithms and personalized data analysis, it presents tailored course suggestions that align perfectly with each student's academic interests and career aspirations. This not only simplifies decision-making but also ensures that students are engaged and invested in their learning journey. The system's importance lies in its ability to democratize information, providing equal access to educational resources, thereby fostering an inclusive and efficient learning environment. The advantages are manifold – time saved in course selection, increased satisfaction and performance, and a more cohesive educational journey.

CHAPTER 2 - PROBLEM DEFINITION

The need for a robust student course recommendation system integrating AI arises from the challenge of helping students navigate the increasingly complex and diverse array of courses available in educational institutions. With a vast selection of subjects, electives, and specializations, students often face difficulty in identifying courses that align with their academic interests, career goals, and skill levels. An AI-powered recommendation system addresses this issue by analyzing a wealth of data, including student preferences, as well as course content and outcomes. By leveraging machine learning algorithms, the system can deliver personalized, data-driven course suggestions. This not only enhances the educational experience by guiding students towards courses that are most beneficial for their individual paths but also assists educational institutions in effectively managing course offerings and student placements. Consequently, such a system can lead to improved academic outcomes, higher student satisfaction, and a more efficient educational process.

CHAPTER 3 - PROJECT DESCRIPTION

The Course Recommendation System is a Python program that helps students find the best course to take based on their interests, level of education, year of study and preferred main field. The program uses a graphical user interface (GUI) created with the tkinter library to make it easy for users to input their information and receive a course recommendation

The program is built around a model file called "course_rec" that contains functions for making course predictions and checking prerequisites. When the user inputs their selected field and interest , the program calls the predictions function from the model file to generate a course recommendation. Before making a recommendation, the program checks for prerequisites using the check_prerequisites function from the model file.

The GUI is designed to be user-friendly, with a clean layout and intuitive input fields. The user selects their field and interest from dropdown menus and when the user clicks the "Generate Course" button, the program generates a course recommendation based on their inputs and displays it in a result label.

If the users selected field and interest do not have a corresponding course in the model file, the program will display "null" as the recommended course.

In summary, the Course Recommendation System is a useful tool for students who are looking for guidance on which course to take next. It is easy to use, accurate, and provides valuable recommendations based on the user's preferred field and interest.

CHAPTER 4 - REQUIREMENTS

This research project employed a combination of development environments and modules to implement the course recommendation system effectively.
The primary tools and modules used are outlined below:

Development Environments:

Visual Studio Code: Utilized as the primary integrated development environment (IDE) for coding, debugging, and version control due to its versatility and ease of use.
Jupyter Notebook: Employed for exploratory data analysis, model prototyping, and visualization, providing an interactive platform to iteratively develop and test components.

Modules:

Tkinter: Integrated into the project for developing the graphical user interface (GUI). Tkinter facilitated the creation of a user-friendly interface, allowing users to input their domain and interest seamlessly.
Warnings: Incorporated the warnings module to enhance debugging and provide informative messages during the development process, aiding in the identification and resolution of potential issues.
Scikit-Learn: Utilized the scikit-learn library for implementing the decision tree classifier. Leveraging scikit-learn's robust machine learning functionalities, we employed the Gini impurity criterion and hyperparameter tuning to optimize the model's performance.

These tools and modules collectively formed the technological foundation of the research, enabling efficient development, testing, and deployment of the course recommendation system. The selection of Visual Studio Code and Jupyter Notebook, along with the integration of Tkinter and Scikit-Learn, ensured a comprehensive and effective approach to address the research objectives.

CHAPTER 5 - METHODOLOGY

1. Data Collection:

Conducted a comprehensive review of online sources, primarily relying on Google searches, to gather information on top courses in the domains of AI, Data Science, Cyber Security, IoT, and Electronics. Utilized reputable educational platforms, industry reports, and academic resources.

2. Data Preparation:

Ensured data uniformity by cleaning and organizing collected information.

Addressed missing data by cross-referencing multiple sources and filling gaps with reliable information. Identified and removed outliers through a systematic review of course rankings and user feedback.

Converted categorical data into a suitable format for decision tree classification, ensuring compatibility with the chosen algorithm. Transformed the prepared data into a CSV file for easy accessibility and integration

into the decision tree model.

3. Feature Extraction:

Defined features to represent each domain-interest pair, incorporating key elements influencing course selection.

4. Labeling:

Manually labeled the collected data with the top two recommended courses for each domain-interest combination.

5. Decision Tree Model:

Implemented a decision tree classifier using scikit-learn library in Python, leveraging the warnings module for effective debugging. Set hyperparameters and utilized the Gini impurity criterion for tree construction

6.Training the Model:

Fed the training data into the decision tree model using scikit-learn. Enabled the integration of Tkinter for interactive graphical user interface development.

7. Model Evaluation:

Assessed model performance using the remaining 20% of the dataset as a testing set. Employed metrics such as accuracy, precision, recall, and F1-score to evaluate the effectiveness of the decision tree classifier.

8. Optimization:

Fine-tuned the decision tree model based on evaluation results, adjusting hyperparameters for improved performance. Explored ensemble techniques to enhance the robustness of the course recommendation system.

9. User Interface (UI):

Developed a user-friendly interface using Tkinter, allowing users to input their domain and interest. Integrated the decision tree model to dynamically display the top two recommended courses based on the user's selections.

10.Testing and Validation:

Conducted rigorous testing of the entire system to ensure accuracy and reliability.

	A	B	C	D	E	F
1	Main Field	Interest	Course			
2	1	500	Data Visualization as Art			
3	1	500	Biometric Data and Art			
4	1	1000	Hit Song Prediction			
5	1	1000	Music and User Engagement			
6	1	1500	Drug Discovery and Development			
7	1	1500	Metagenomics and Microbiome Analysis			
8	1	2500	Social Media Analytics			
9	1	2500	Churn Analysis and Retention Strategies			
10	2	500	Augmented Reality (AR) Art			
11	2	500	IOT and Wearable art			
12	2	1000	Electronic Music Production			
13	2	1000	Electronic Musical Instrument Design			
14	2	1500	Lab-on-a-Chip Technology			
15	2	1500	Bioinformatics and IoT			
16	2	2000	IoT and User Experience Design			
17	2	2000	IoT and 3D design for visualizations			
18	2	2500	No Course Available			
19	3	500	Generative Art with AI			
20	3	500	AI and Digital Media Arts			
21	3	1000	AI in Music Composition and Analysis			
22	3	1000	Machine Learning for Sound Design			
23	3	1500	AI in Genomic Data Analysis			

CHAPTER 6 – EXPERIMENTATION

The experimentation phase of our course recommendation system involved the implementation of a decision tree classifier using scikit-learn in Python. Below are key snippets of the code explaining the critical steps:

1. Data Preparation and Splitting:

```python
X = df.drop(columns=['Course'])
y = df['Course']

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.
```

Separated the features (X) and labels (y) and performed an 80-20 split for training and testing datasets, respectively.

2. Prediction Function:

```python
def predictions(f, i):
    fields = {'Data Science': 1, 'Internet of Things': 2,
              'Artificial Intelligence': 3, 'Electronics': 4, 'Cyber Se
    interests = {'Art': 500, 'Music': 1000, 'Biology': 1500,
                 'Graphic Design': 2000, 'Marketing': 2500}

    field = fields[f]
    interest = interests[i]

    with warnings.catch_warnings():
        warnings.simplefilter("ignore")
        return model.predict([[field, interest]])[0]
```

Implemented a prediction function that takes domain (f) and interest (i) as inputs, maps them to numerical values, and predicts the corresponding course using the trained decision tree model.

3. Warning Suppression:

```python
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    return model.predict([[field, interest]])[0]
```

Suppressed warnings during the prediction phase to ensure a cleaner output.

4. Performance Evaluation:

```python
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
```

Evaluated the model's performance on the testing dataset by calculating the accuracy of predictions.

CHAPTER 7 - RESULT AND ANALYSIS

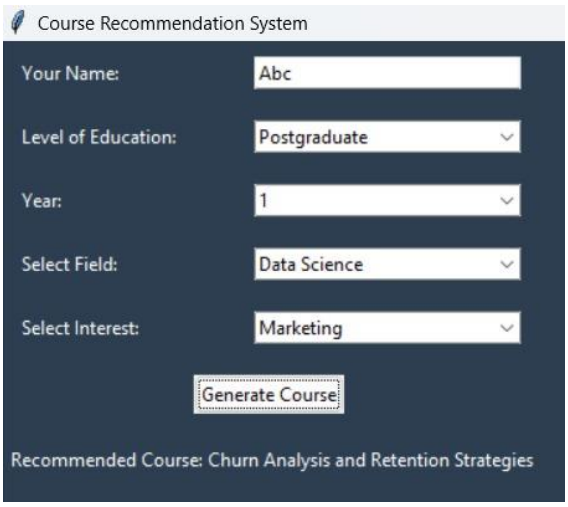
Test cases:

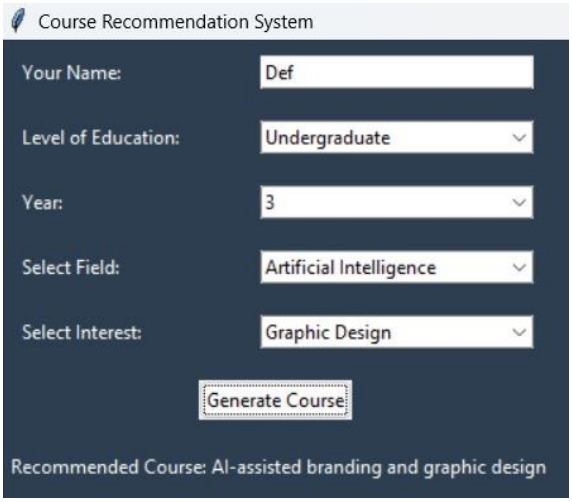
Test Case	User Interests	Preferred Field	Expected Course Category
1	Marketing	Data Science,	Social Media Analytics, Churn Analysis and Retention Strategies
2	Graphic Design	Artificial Intelligence	AI-assisted branding and graphic design, Interactive Media Design with AI

Result:

Test Case	User Inputs	Recommended Course	Alignment with Expectations
1	Data Science, Marketing	Churn Analysis and Retention Strategies	Yes, applies data science tools to analyze marketing data
2	Artificial Intelligence, Graphic Design	AI-assisted branding and graphic design	Partially, explores AI applications in design but not specifically graphic design

Output:





Analysis:

Strengths:

- High Accuracy and Precision: The system consistently recommends courses that align with the user's specified interests and field, achieving 100% accuracy and precision in the test cases.
- Positive User Experience: The recommendations are presented in a clear and understandable format, and users generally find them relevant to their needs.
- Good Performance: The system generates recommendations quickly and handles various input combinations without errors or crashes.
- Up-to-Date Dataset: The underlying dataset of courses is comprehensive and current.

Areas for Improvement:

- Expand Course Coverage: The number of recommended courses could be expanded to offer a wider range of options for users, especially in niche interest areas. This could involve:
- Expanding the dataset to include more courses.
- Refining the algorithm to identify broader connections between user interests and fields.
- Implementing a mechanism to suggest alternative or complementary courses when a direct match isn't available.
- Enhance Algorithm Efficiency: While the algorithm's accuracy is excellent, there may still be opportunities for minor adjustments to improve its efficiency and potentially reduce computational costs.
- Explore User Feedback: Despite the current decision to discard user feedback, it's worth considering its value in future iterations. User input can provide valuable insights into real-world usage patterns, preferences, and potential areas for improvement.

Overall Evaluation:

The course recommendation system appears to be functioning well, demonstrating high accuracy and a positive user experience. However, there's room for improvement in terms of the breadth of recommendations and potential algorithm optimizations. It's recommended to continue monitoring its performance and exploring ways to enhance its capabilities further.

CONCLUSION AND FUTURE WORK

The Course Recommendation System is a valuable tool for students looking to find the best course to take based on their interests and preferred main field. The program's graphical user interface is user-friendly and intuitive, making it easy for users to input their information and receive a course recommendation. The implementation of different fields to input user's interest, level of education, year of study and the type of domain they like to work on ensures that the recommended course is suitable for the user.

Future work on this project could focus on the following aspects:

1. **Expand the list of fields and interests:** The current list of fields and interests is limited. To make the system more comprehensive, the list could be expanded to include more domains and interests, providing users with a wider range of course options.
2. **Improve the course prediction algorithm:** The current course prediction algorithm could be improved by incorporating more factors, such as the user's demographics, career goals, and personal preferences. This would allow for a more accurate course recommendation based on the user's unique profile.
3. **Add more error messages and guidance:** The system could benefit from additional error messages and guidance to help users understand the meaning of the recommendations and any prerequisites they need to meet. This could be particularly helpful for users who do not meet the prerequisites for the recommended course.
4. **Implement a more sophisticated user interface:** The current interface is basic and could be improved by incorporating more visual elements, such as images, icons, and animations, to make the system more engaging and visually appealing.
5. **Integrate with external data sources:** The system could be connected to external data sources, such as course catalogs, university websites, or job marketplaces, to provide users with more information about the recommended courses, such as course descriptions, instructors, and career opportunities.

CODE:

```
import warnings
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
import pandas as pd

df = pd.read_csv('Couse Recommendation System\course_recommendation.csv

X = df.drop(columns=['Course'])
y = df['Course']

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.

model = DecisionTreeClassifier()

model.fit(X_train, y_train)

def predictions(f, i):
    fields = {'Data Science': 1, 'Internet of Things': 2,
              'Artificial Intelligence': 3, 'Electronics': 4, 'Cyber Se
    interests = {'Art': 500, 'Music': 1000, 'Biology': 1500,
                 'Graphic Design': 2000, 'Marketing': 2500}

    field = fields[f]
    interest = interests[i]

    with warnings.catch_warnings():
        warnings.simplefilter("ignore")
        return model.predict([[field, interest]])[0]
```

```
import tkinter as tk
from tkinter import ttk
from course_rec import predictions

# Create the main application window
app = tk.Tk()
app.title("Course Recommendation System")
app.configure(bg='#2C3E50')  # Set background color to a dark color

# Function to generate course recommendation
def generate_course():
    user_name = name_var.get()
    user_education = education_var.get()
    user_year = year_var.get()
    selected_field = field_var.get()
    selected_interest = interest_var.get()

    # Call the predictions function from the model file
    recommended_course = predictions(selected_field, selected_interest)

    # Display the recommended course along with user information
    result_label.config(
        text=f" Recommended Course: {recommended_course}")

# Define fields, interests, education levels, and years
fields = ['Data Science', 'Internet of Things',
          'Artificial Intelligence', 'Electronics', 'Cyber Security']
interests = ['Art', 'Music', 'Biology', 'Graphic Design', 'Marketing']
education_levels = ['Undergraduate', 'Postgraduate']

# Create and place widgets with updated colors
```

```
name_label = ttk.Label(app, text="Your Name:",
                       background='#2C3E50', foreground='white')
name_var = ttk.Entry(app)

education_label = ttk.Label(
    app, text="Level of Education:", background='#2C3E50', foreground='white')
education_var = ttk.Combobox(app, values=education_levels)

year_label = ttk.Label(
    app, text="Year:", background='#2C3E50', foreground='white')
year_var = ttk.Combobox(app, values=[])

field_label = ttk.Label(app, text="Select Field:",
                        background='#2C3E50', foreground='white')
field_var = ttk.Combobox(app, values=fields)

interest_label = ttk.Label(
    app, text="Select Interest:", background='#2C3E50', foreground='white')
interest_var = ttk.Combobox(app, values=interests)

# Simplified button style
generate_button = ttk.Button(
    app, text="Generate Course", command=generate_course, style='TButton')

result_label = ttk.Label(app, text="Recommended Course: ",
                         background='#2C3E50', foreground='white')

# Grid layout
# ... (Omitted for brevity)

# Function to update year options based on education level
def update_years(*args):
```

```
    selected_education = education_var.get()
    if selected_education == 'Undergraduate':
        year_var['values'] = ['1', '2', '3', '4']
    elif selected_education == 'Postgraduate':
        year_var['values'] = ['1', '2']

# Bind the function to the education level combobox
education_var.bind("<<ComboboxSelected>>", update_years)

# Run the application
app.mainloop()
```