LONDON METROPOLITAN UNIVERSITY

informatics college • pokhara

Module Code & Module Title

CU6051NP Artificial Intelligence

75% Individual Coursework

Submission: Milestone 1

Academic Semester: Autumn Semester 2025

Credit: 15 credit semester long module

Student Name: Sharon Gurung

London Met ID: 23048933

College ID: NP04CP4A230114

Assignment Due Date: 07/01/2026.

Assignment Submission Date: 07/01/2026

Submitted To: Jeevan Prakash Pant

GitHub Link	*https://github.com/Norahs-00/Learning_recommendation_system.git*

I confirm that I understand my coursework needs to be submitted via MST under the relevant module page before the deadline in order for my coursework's milestone to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table of Figures

Table of Tables

Abstract

With the rapid growth of online learning platforms, learner can find it difficult to select courses that are of interest to them, that they have some prior knowledge of, and that are aligned with their learning goals, leading to poor learning outcomes and low learner satisfaction. In this project, A learning Course Recommendation System is proposed that considers learner input (interests and preferences) and course information (ratings, enrollment, reviews) to provide personalized course recommendations. Different recommendation strategies are applied to examine the relationship between learner preferences and available courses to generate appropriate course recommendations. The proposed system aims to improve recommendation relevance, learner engagement, and course selection efficiency in large-scale online learning environments.

Acknowledgement

I would like to thank my module leader Mr. Jeevan Prakash Pant Sir for his outstanding guidance, support, motivation and constructive feedback during the coursework project. My sincere appreciation to the Informatics College Pokhara for providing necessary resources and a supportive learning environment that allowed me to successfully complete this coursework. Finally, I would extend my gratitude toward friends and family for their continuous support and patience during the course of this project.

1. Introduction

1.1 Explanation of the Topic and AI Concepts Used

Artificial Intelligence is increasingly used in the education sector to support learning and decision making. One important application of the AI in education is recommendation systems which help user identify important information from huge datasets. Unlike the traditional recommendation systems used in e-commerce or entertainment platforms, educational recommendation systems focus on other factor like relevance of topic than popularity.Online learning platforms such as Coursera contains thousands of courses which can make learners difficult in choosing the course.AI based recommendation systems help reduce this difficulty by analysing courses features and suggesting the options that are related with learner needs (Tilahun & Sekeroglu, 2020).

Recommendation systems are used to help learners to choose suitable courses based on their interests, background, and learning goals in online learning platform. Machine Learning(ML) is a subfield of AI which plays an important role in building systems. ML models learn pattern from past historical data and improve their prediction over the time. Supervised learning algorithms are used when the results are labelled like if a course should be recommended based on enrollment data or ratings. Supervised Algorithms such as Logistic Regression, Decision Trees and Random Forests are selected to predict courses to learners. Logistic Regression is used as a standard model as it is simple to use. Decision Trees and Random Forest are used to show more complex relationships between course features. Unsupervised algorithm such as clustering is used to group similar courses. It allows the system to recommend group of related courses and features where information is limited. The hybrid recommendation method is known by combining supervised and unsupervised methods to enhance performance and robustness (Ren, et al., 2022).

The educational recommendation systems have been benefited from the use of Machine learning algorithms. First learners receive personalized suggestions that help them to match their learning objectives. Then, the system reduces time spent researching for suitable courses.Third, personalized recommendations can increase learner engagement and motivation to learn. Finally, It is scalable and suitable for handling huge datasets across online learning platforms. In this project, these methods are applied to develop a course recommendation system using the Coursera-course 2024 dataset. The system aims to show useful course recommendations by analysing course metadata such as ratings, enrollments, instructors and the organisations.

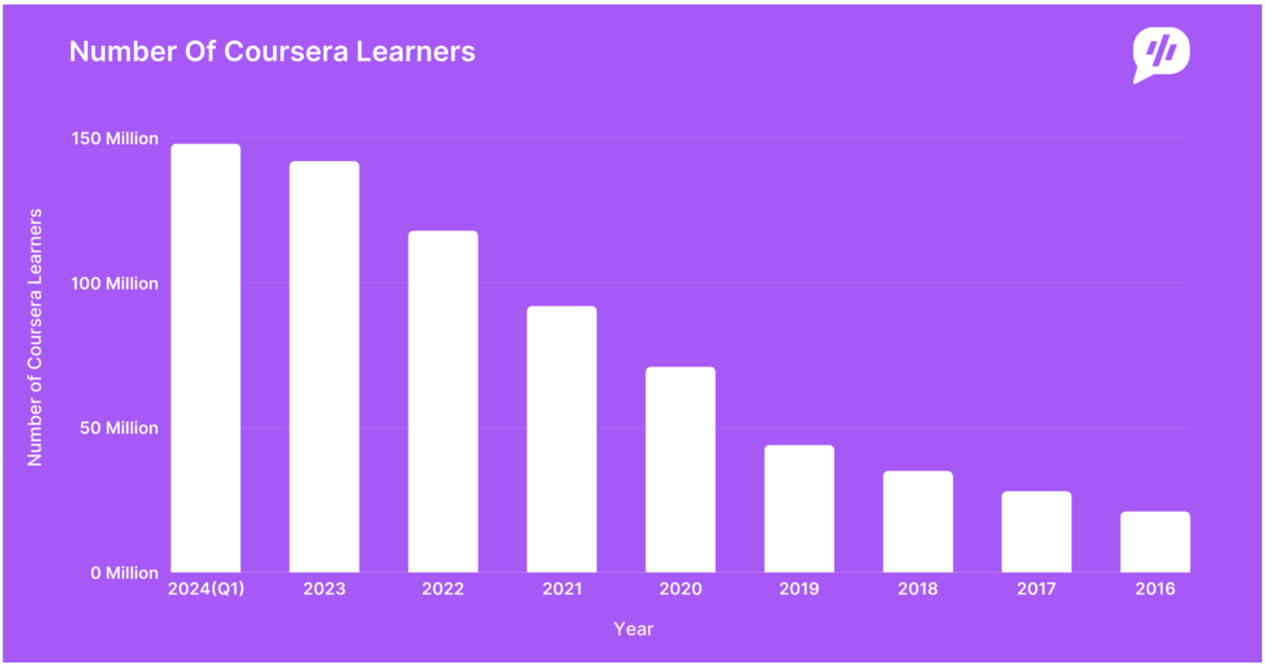
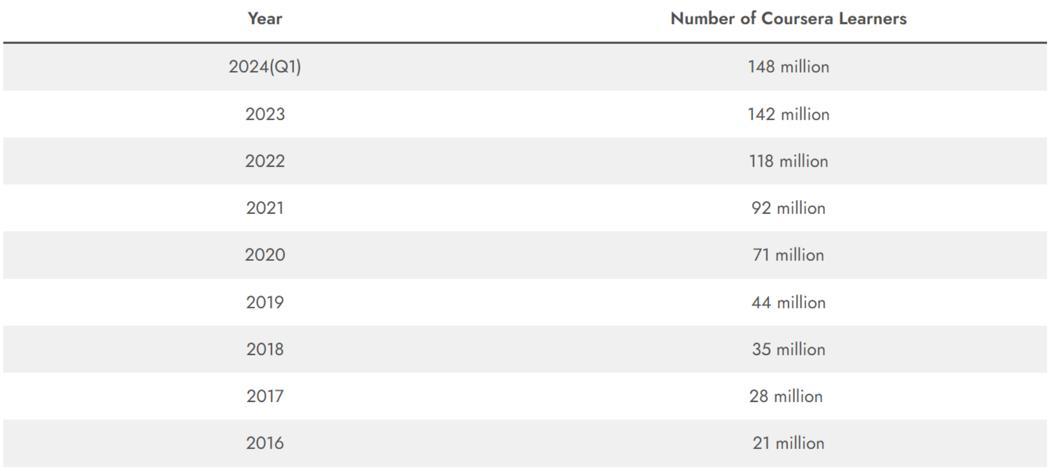


Figure 1: Number of Coursera Learner from 2016-2024 (Coursera, 2024)

The above bar graph shows the data of total numbers of coursera learner from 2016 -2024 across the world.



Year	Number of Coursera Learners
2024(Q1)	148 million
2023	142 million
2022	118 million
2021	92 million
2020	71 million
2019	44 million
2018	35 million
2017	28 million
2016	21 million

Table 1: Table of Total Coursera Learners from 2016-2024 (Coursera, 2024)

The above table shows the exact number of coursera learner who has enrolled in the platform over the years from 2016 to 2024. The gradual increase of learners from 2016 to 2024 can be seen due to covid 19 and growth of online learning platforms.

1.2 Explanation of the Chosen Problem Domain

Online learning platforms such as Coursera, Udemy, and EdX have significantly expanded access to education by offering a wide range of courses of various subjects. While the development of the online learning platform provides learner with many options but it also create a difficulty in identifying courses that match with their learning goal.This issue is commonly called information overload where learners struggle to make effective decisions sue to the huge volume of available options (Ren, et al., 2022).

The research shows that learner engagement and course completion rates in online learning platforms remains very low. Studies on Massive Open Online Courses (MOOCs)shows that the course completion rates often range between 5% and 15% which means that a huge proportion of learners do not complete the courses they enroll in (Peterson, 2013). Many learners joins for courses but fails to start or continue them which suggests that the course selection may not match with their expectations. Most existing recommendation systems on online learning systems rely on popularity based factors such as enrollment numbers, average ratings. These factors shows the course quality , which does not focuses on individual learner preferences, background knowledge or learning objectives. As a results learners are recommended courses that are popular but not suitable for their learning needs (Tilahun & Sekeroglu, 2020).

This project mainly focuses on addressing such problem by developing personalized course recommendation system. Personalization in education contains recommendations which is based on some factors such as difficulty of course, content focus and learner interests. The Coursera course 2024 dataset is a dataset which contains 6645 course records with features such as course title, ratings, enrollments, instructors and the organisations which is used as the foundation for this systems. The proposed system aims to reduce the information overload, improve course relevance and support decision making for learners by using machine learning algorithms. The system is designed to provide recommendations that balance popularity of the courses, quality and similarity of courses. This method helps improve learners satisfaction and supports more effective engagement with online learning platforms.

1.3 Aims and Objectives

The main aim of this project is to design and evaluate a machine learning based course recommendation system which helps learners in selecting suitable courses from platform such as Coursera. It focuses on analysing course metadata to create recommendations that matches with learner objectives and goals. Some of the objectives of this system are as following:-

- To analyse the Coursera course 2024 dataset using exploratory data analysis (EDA) to understand important features related to a course recommendation.
- To apply supervised machine learning algorithms such as Logistic Regression, Decision Trees, and Random Forests to predict the suitability of the course.
- To use unsupervised learning method such as clustering to group courses with similar interests.
- To develop a hybrid recommendation method that combines supervised and unsupervised methods to improve accuracy and scalability.
- To evaluate the performance of the course recommendations using proper metrics such as precision, recall, and F1 score.

2. Background

2.1 Research Work Done

Research on AI based course recommendation systems has changed over the time with each study trying to improve earlier limitations.Early foundational work focuses on structured decision making for academic planning while more recent studies introduced machine learning, deep learning, and real-time personalisation.These development shows the growing needs for the systems which can adapt to learner diversity and changing learner behaviours.

2.1.1 Research 1: Artificial Intelligence in Adaptive Education

The early research shows that the course selection was treated as a sequential decision making problem in the personalized course recommendation (Xu, et al., 2016). Their model uses dynamic programming and techniques which the system learned from historical student data to recommend suitable courses. It was significant because it shifted academic advising from static rule based method to data driven models. However this approach was designed mainly for traditional university environments where learning paths were structured. It was open to online learning platforms where learners often follow flexible and non-linear path which is limited.

2.1.2 Research 2: AI for Personalized Learning in Higher Education

An intelligent course advising system is developed which is based on expert systems and rule based reasoning (Tilahun & Sekeroglu, 2020). This model included the academic rules , curriculum and the student performance data to provide course recommendations. One of the important advantage of this approach is transparency as recommendations are created using defined rules. Inspite of the interpretability, the system has problems in handling large and various datasets. Rule based systems require manual updates and issues to adapt to change learner preferences. It makes them less suitable for large scale online learning platforms such as Coursera where learner behaviour differs significantly.

2.1.3 Research 3: Multi-Model Course Recommendation Framework

(Islam & Hosen , 2022) proposed a multi model machine learning framework which combines multiple perspective models to improve recommendation accuracy. Their work shows that hybrid approaches often performs better than a single model system s by gathering different aspects of learners and a course data. The framework relies on structured academic datasets. The lack of real time learner interaction data limits the effectiveness in dynamic online learning environments. It shows the requirements for the systems which can operate effectively using limited learner information. A Deep learning based multimodel recommendation framework was introduced which includes text content, learner behaviour, and features (Ren, et al., 2022). This system gathers both explicit and implicit learner preferences by using LSTM networks with attention mechanisms. Although this approach improved personalization and recommendation accuracy, it also introduced problems related to interpretability and complexity. Deep learning models needs large datasets and high processing power which may not be practical for all learning platforms.

2.1.4 Research 4: AI for Lifelong Learning

More recently Session based recommendation systems which adapt recommendations based on short term learner interactions. This research study shows it is effective for addressing cold start problems but may overlook long term learning goals and progression. Overall, these studies shows clear horizontal progression: from structured sequence optimisation (2016), to institutional(college)intelligence (2020), contextual prerequisite modeling (2022), multimodel deep learning (2022), and finally real-time session-aware personalization (2024).

2.2 Review and Analysis of Existing Work

2.2.1 Review and Analysis of Research 1

The work by (Xu, et al., 2016) proposed a personalized course sequence recommendation system that focuses on how the learners progresses over the time. One advantage of this work is that it considers course prerequisites and long term goals. This makes the system suitable for traditional university environments where courses has fixed structure. But, this approach has problems when applied to online learning platforms such as Coursera. Online learners often choose courses based on their interests without following fixed sequence because it may not work well in flexible learning environments.

2.2.2 Review and Analysis of Research 2

Tilahun and Sekeroglu (2020) has developed an academic advising system based on expert system and rule based methods. A main advantage of this approach is that it is easy to understand as recommendations are created using clear and predefined rules. This transparency makes the system useful for academic institutions that need to follow formal policies. But the system has limited flexibility. Rule based systems does not scale well when handling large datasets and they cannot easily adapt to change learner preferences. This makes them less suitable for large scale online learning platforms.

2.2.3 Review and Analysis of Research 3

The study by Islam and Hosen (2022) introduced a machine learning based course recommendation framework which combines multiple models to improve prediction accuracy. One of the benefit of this approach is its ability to manage complex academic constraints while producing better results than the single model systems. This shows that combining models can improve recommendation performance. However, the system mainly depend on structured academic data and does not use enough real learner interaction data which results in the limited performance in online learning platforms where learner behaviour is inconsistent and the data is often incomplete. Ren et al. (2022) introduced a deep learning based recommendation system that uses different types of data such as text, learner behaviour and contextual data. The main advantage of this approach is its ability to create highly personalized recommendations. But deep learning models need large datasets and highly computational resources which can be difficult to manage in many educational institutions. These models are often hard to explain which may reduce user trust and limit their practical use in education.

2.2.4 Review and Analysis of Research 4

More recently, Khan and Polyzou (2024) focused on session based recommendation system that analyse learners short term behaviour. This approach is useful for handling cold-start problems where little information is available about a learner (Abbakumov, 2014). It is especially helpful for online learning platforms with new users. But session based learning systems are mainly focused on immediate action and may ignores the learning goals. Continuous tracking of learner sessions also raises problems related to system complexity, scalability and data privacy.

Overall existing research shows great progress in improving course recommendation systems. But many approaches still face challenges related to scalability, adaptability and data dependency. These limitations shows the needs for a balanced recommendation system that uses both supervised and unsupervised machine learning algorithms. The proposed system aims to provide effective, scalable, and transparent recommendations which is suitable for online learning platforms by relying on simple and widely available course metadata.

2.3 Analytical Review of Existing Systems on the Problem Domain

In recent years, online learning platforms introduced developed different recommendation techniques to help learners select suitable courses. Early systems mainly relied on popularity based methods and collaborative filtering where recommendations were created using enrollment numbers, average ratings and basic learner course interaction data (Ziegler, et al., 2017).

As online platforms grew in size, hybrid recommendation systems were introduced. These systems combine content based filtering with collaborative signals which allows recommendations to be based on both course information and user interaction pattern. Many studies have shown that this approach improves recommendation relevance compared to using a single method (Buitrago & Chiappe, 2019). However, a hybrid systems still depend heavily on user interaction data which is often limited or unavailable on open platforms such as Coursera.

The researcher explores the Massive Open Online Courses(MOOCs) growth which shows knowledge aware and constraint based recommendation system to improve academic consistency. These systems consider curriculum structure, learner background, and prerequisite relationships. Although such systems improve recommendation validity it requires structured institutional data that is usually not available on open learning platforms (Tilahun & Sekeroglu, 2020). More recently , fairness aware and recommendation system have been proposed to address ethical concerns and improve transparency. While these systems increase but often reduce accuracy as simpler and more transparent models may not capture complex learning patterns (Liu, et al., 2020).

Learning platforms are now growing toward content aware and session based recommendation systems that adapt recommendations based on short term learner behaviour.These system are effective in handling cold-start problems and rapidly changing learner interests. But they introduce challenges related to scalability, real time processing and data privacy (George & La, 2024).

Overall, learning recommendation systems have progressed from static popularity related approaches to more adaptive and intelligent frameworks. Despite this progress, challenges related to personalization, scalability, fairness and explainability remained unresolved. These problems shows the need for a balanced recommendation system that can use structured course metadata while remaining scalable, interpretable and effective.

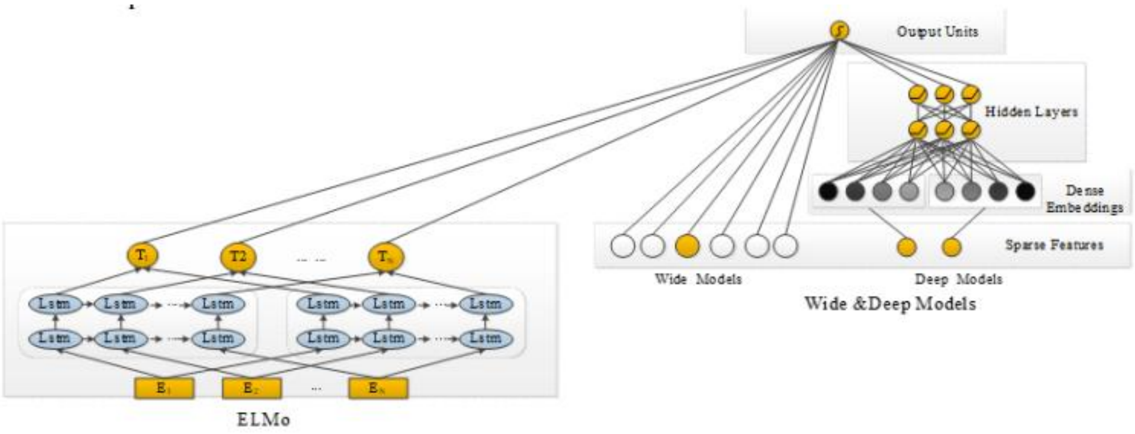


Figure 2: Online Learning Resource Recommendation Method Based on Wide & Deep and Elmo Model (Liu, et al., 2020)

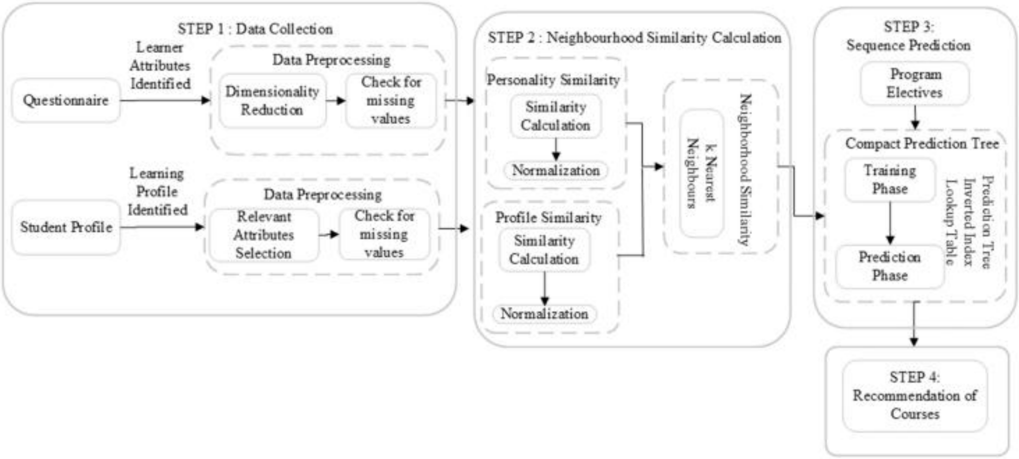


Figure 3: Course recommendation framework (George & La, 2024)

2.4 Dataset Description

The dataset used for this project is the Coursera-course 2024 dataset that consists of 6,645 records in CSV format. It includes features such as course title, enrollment numbers, average ratings, number of reviews, instructor names, and organisations. This dataset provides metadata to build learner profiles and generating recommendations. Enrollment and rating features helps to identify high-quality and widely varied courses, while instructor and organisation information enables similarity based filtering.The dataset includes the following features:

Feature Name	Description	Data Type
course_id	Unique identifier for each course	Integer
title	Course title	String
organization	Organization or university offering the course	String
instructor	Instructor name(s)	String
level	Course difficulty level	Categorical (String)
certificate_type	Type of certificate offered	Categorical (String)
enrolled	Number of learners enrolled	Integer
rating	Average course rating (out of 5)	Float
num_reviews	Number of learner reviews	Integer
duration_weeks	Estimated duration of the course in weeks	Integer
skills	Skills covered in the course	String
language	Language of instruction	String
url	Course webpage link	String

Table 2: Data Dictionary of the system

3. Solution

3.1 Explanation of the Solution

This project develops an AI based course recommendation system which helps learners select suitable courses from large online learning platforms. The proposed solution addresses the problems of existing systems that is based on popularity by applying machine learning algorithm that analyse course informations to generate more relevant course recommendations. The main goal is to reduce information overload to improve learner engagement, and support informed decision-making.

It uses the Coursera-course 2024 dataset, which contains structured information such as course titles, enrollment numbers, ratings, reviews, instructors, difficulty level and the organisations. The proposed solution focuses on content-based and hybrid recommendation technique since the individual learner interaction data is limited. These techniques are suitable for large scale platform and allow it to remain scalable.Data preprocessing is an important step of the proposed solution.This process includes handling missing values, removing duplicate records, normalizing numerical features and encoding categorical data.These steps reduces noise and improve the reliability of the machine learning models during training.

3.2 System Architecture and Workflow

The proposed learning recommendation system follows a modular architecture consisting of four main components: data preprocessing, feature engineering, recommendation engine, and output generation. Initially, the raw dataset undergoes preprocessing which includes handling missing values, normalising numerical features such as number of enrollment and ratings, and encoding categorical variables such as instructor names and organisations.

Next, feature engineering is performed to extract meaningful informations from the dataset. Course popularity, quality and organisation details are transformed into numbers to make system understand. These values show how course is relevant.The machine learning models uses features as the imput. The recommendation system uses more than one machine learning method to suggest courses in order. Supervised learning is used to predict if a course should be recommended or not. Unsupervised learning is used to group courses that are similar to each other. Both methods are combined in one system to improve accuracy and give more variety in results of recommendation systems.

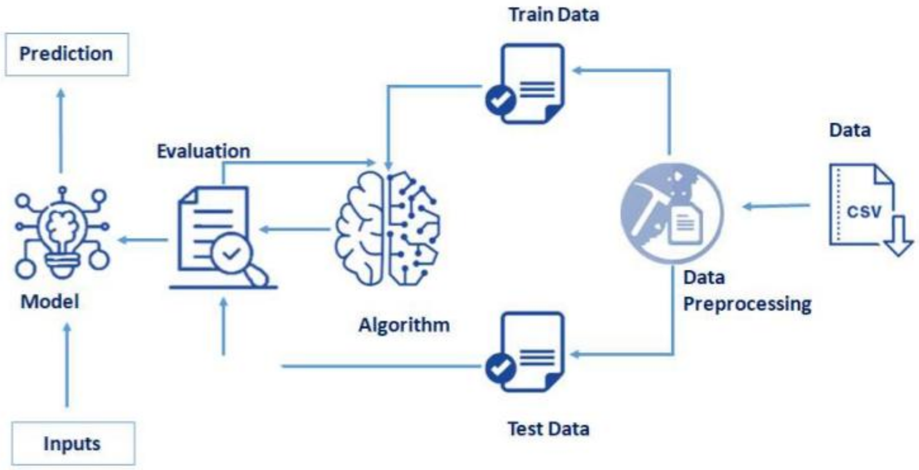


Figure 4: System Architecture

3.3 AI Algorithms Used

3.3.1 Supervised Learning Algorithms

Supervised learning algorithm is used to predict if a course should be recommended or not by using existing data such as ratings, enrollment levels, and review numbers. These values has known results so the model learns from them.

Logistic Regression is used as a basic classification method because it is easy to understand and simple to apply. It calculates the probability that a course is suitable for recommendation by finding the relationship between course features and the final decision which is either 0 or 1. The probability is calculated using the sigmoid function:-

$$P(y = 1 \mid x) = 1 / (1 + e\hat{} - (\beta 0 + \beta 1x1 + \beta 2x2 + \ldots + \beta nxn))$$

In this equation, $x_1, x_2, \ldots, x_n$ represent course features such as rating, number of enrollments, reviews, course level and duration of the course. The parameters β0, β1,…βn are learned weights from the model. The sigmoid function changes the output into values between 0 and 1. This equation gives the probability that a course should be recommended. If the probability of result is 1 then the course is considered suitable. If it is close to 0 then the course is not suitable. A threshold of (0.5) is used to make the final recommendation decision.

Decision Trees are used to handle more complex relationships between course features and recommendations. They split the data based on the feature values to create decision rules. Entropy is used to measre the impurity in the data. Information Gain shows how much uncertainity is reduced after splitting the data.

$$Entropy(S) = -\sum_{\{i=1\}_{i}^{\{c\}p}\backslash log_2} p_i$$

$$IG(S,A) = Entropy(S) - \sum_{\{v\}\backslash frac\{|S_v|\}\{|S|\}Entropy(S_v)}$$

Here, S shows the Coursera dataset, A represents a course feature such as rating or number of enrollment and Sv represents the subsets formed after the split. It helps to create rules for course recommendation such as recommending courses with higher ratings and enrollment numbers.

Random Forests is an advanced method which combine multiple decision trees together. Each tree is trained on different parts of the dataset. The final prediction is made by combining results from all decision trees which helps to improve prediction accuracy and reduce overfitting.

$$\widehat{}\{y\} = majority\ vote\left(T_{1(x)}, T_{2(x)}, \dots, T_{n(x)}\right)$$

In the above equation each T represents a decision tree prediction based on course features.The final prediction $\hat{y}$ is determined by using majority voting from all the trees.

3.3.2 Unsupervised Learning Algorithms

Unsupervised learning is used to identify hidden patterns in course data. It does not have labelled data. K-Means clustering is used to group courses that are similar based on features such as ratings, enrollment and course content. The main aim of K-Means is to reduce distance between courses and their centroid of clusters. It is measured by Within- Cluster sum of squares(WCSS).

$$J = \sum_{\{j=1\}}^{\{k\}\sum_{\{x_i \in c_j\}}^{2}\left\|x_i - c_j\right\|}$$

Where:

J = Total clustering error

k = Number of clusters

x_i = A course features

C_j = Cluster j

c_j = centroid (average course)

$||x_i - c_j||^2$ = squared distance

Here, $x_i$represents a course feature, $c_j$represents the centroid of cluster j, and k is the number of clusters. The algorithm groups courses with similar characteristics together.These clusters are useful to recommend groups of similar courses where the information is found limited. It helps solving cold-start problem which means hadling missing values found in dataset.

3.3.3 Evaluation Metrics

Evaluation Metrics is used to check how the recommendation system performs.

Precision shows the accuracy of recommended courses to the learners in recommendation system. A high precision value shows that the system never recommends unsuitable courses.

$$Precision = \frac{TP}{TP + FP}$$

Recall measures how many relevant courses are successfully recommended. A high recall value means that the system does not miss suitable courses.

$$Recall = \frac{TP}{TP + FN}$$

F1 score is the combination of precision and recall which provides balance evaluation of the performance. It is useful in recommendation of highly relevant courses for learner.

$$F1 = 2 \times \frac{(Precision * Recall)}{(Precision + Recall)}$$

3.3.4 Hybrid Recommendation Method

The system uses hybrid method by combining both supervised and unsupervised learning methods. It helps to change the limitations of using single method. Supervised learning algorithm rank course based on predicted relevance while clustering ensures that recommendation comes from various course groups. It improves accuracy and also adds variety to the recommendations. It prevents the system from only suggesting popular courses and improves in personalisation. The final recommendation score is calculated using a weighted combination of supervision prediction and clustering group to improve both relevance and varieties.

$$Score_{\{final\}} = \alpha \cdot Score_{\{supervised\}} + (1 - \alpha) \cdot Score_{\{cluster\}}$$

3.4 Pseudocode of the solution

```
START
IMPORT required libraries
   pandas for dataset handling
   numpy for numerical operations
   sklearn for preprocessing and machine learning
      StandardScaler for normalization
      train_test_split for data splitting
      LogisticRegression, DecisionTreeClassifier, RandomForestClassifier
      KMeans for clustering
      accuracy_score, precision_score, recall_score

LOAD Coursera-course-dataset (CSV file) into DataFrame
DATA PREPROCESSING
   REMOVE duplicate records
   HANDLE missing values
   SELECT relevant features (ratings, enrollments, reviews, course category)
   ENCODE categorical features (instructor, organization)
   NORMALIZE numerical features using StandardScaler

SUPERVISED LEARNING
   DEFINE target variable (Recommend =1, Not Recommend=0)
   SPLIT dataset into training and testing sets (80% training, 20% testing)

   TRAIN Logistic Regression model
   TRAIN Decision Tree model
   TRAIN Random Forest model

   EVALUATE supervised models
```

```
        CALCULATE accuracy and precision scores

UNSUPERVISED LEARNING
       APPLY K-Means clustering on course features
       GROUP similar courses into clusters
        IDENTIFY course similarity patterns
HYBRID RECOMMENDATION
   FOR each course:
      CALCULATE supervised relevance score

      IF supervised relevance score < predefined threshold THEN
         USE cluster-based recommendation
      ELSE
         USE supervised ranking
      END IF

   COMBINE supervised rankings with clustering results
   RANK courses based on relevance, similarity, and diversity

FINAL RECOMMENDATION
   INPUT learner preferences
   MATCH learner preferences with ranked courses
   OUTPUT Top-N recommended courses

END
```

3.5 Diagrammatical representations of the solution

3.5.1 Flowchart

Flowchart is the graphical representation of step-by-step operational workflow events or actions to make better decision. It is the best way for the beginners to create a program for general purpose. Oval shape represents start and stops of the program, parallelogram represent the input and output of the data, arrows show the direction, rectangle represent the tasks or process and diamond for decisions.

It starts with loading Coursera course 2024 dataset then followed by data preprocessing such as removing duplicate values, handling missing values, encoding categorical features into 0 and 1, and normalizing numerical attributes. It illustrates the use of supervised learning methods to predict known outcomes (course relevance) and unsupervised method including clustering to group similar courses. A decision shape determines whether supervised relevant scores that meet a defined threshold. If not then cluster based recommendation are applied. Finally, the system generates a list of recommended courses based on learner preferences, interest and learning goals. Flowcharts are commonly used in system design to clearly visualize algorithm, make complex workflow easier to understand and improve decision making (Charntaweekhun & Wangsiripitak, 2006).

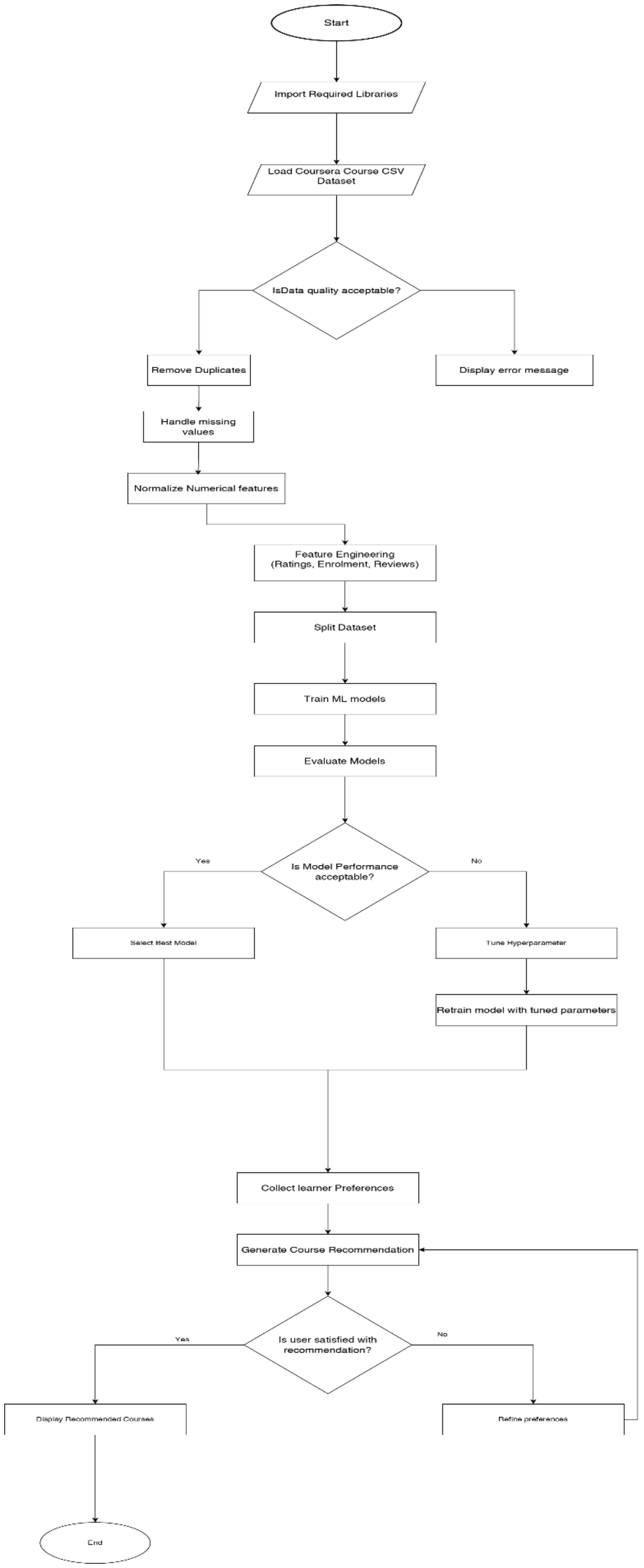

Start
Import Required Libraries
Load Coursera Course CSV Dataset
IsData quality acceptable?
Remove Duplicates
Display error message
Handle missing values
Normalize Numerical features
Feature Engineering (Ratings, Enrolment, Reviews)
Split Dataset
Train ML models
Evaluate Models
Yes
Is Model Performance acceptable?
No
Select Best Model
Tune Hyperparameter
Retrain model with tuned parameters
Collect learner Preferences
Generate Course Recommendation
Yes
Is user satisfied with recommendation?
No
Display Recommended Courses
Refine preferences
End

Figure 5: Flowchart of the system

3.5.2 State transition Diagram

A State Transition Diagram visually represents a finite state machine, utilized to model objects that have a limited number of states and their interactions with the external environment through state changes driven by events (ScienceDirect, 2012). It is made up of nodes that symbolize states and directed edges that show transitions marked with event names. The system transitions from the initial state to data loading, data preprocessing, model training, and recommendation generated states, before reaching the final output state where personalized course recommendations are delivered to learner.

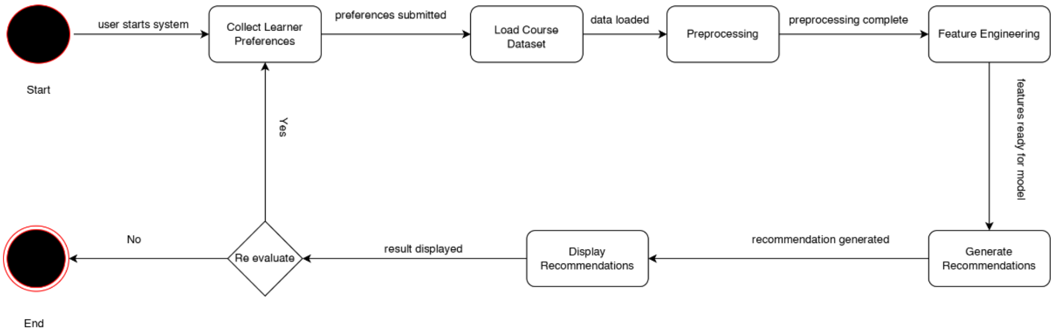


Figure 6: State transition Diagram

3.6 Explanation of the development process

The development process for recommending courses from Coursera datasets of various steps supported by various tools and technologies. This system will recommend according to the learner’s goals and needs.The technologies and libraries makes the task of data understanding, data handling, building model and making evaluation of the data.

Language used for the system:

Figure 7: Diagram of Python Logo

Python is a widely used general purpose high level programming language. It is a powerful and fastest growing object oriented programming language developed by Guido van Rossum . It is easy to understand, open source, user friendly,flexible. It also support many libraries like numpy, pandas, matplotlib and Scikit. It is suitable for handling data cleaning, analysis, data preprocessing, machine learning and data visualizations (VanderPlas, 2016).

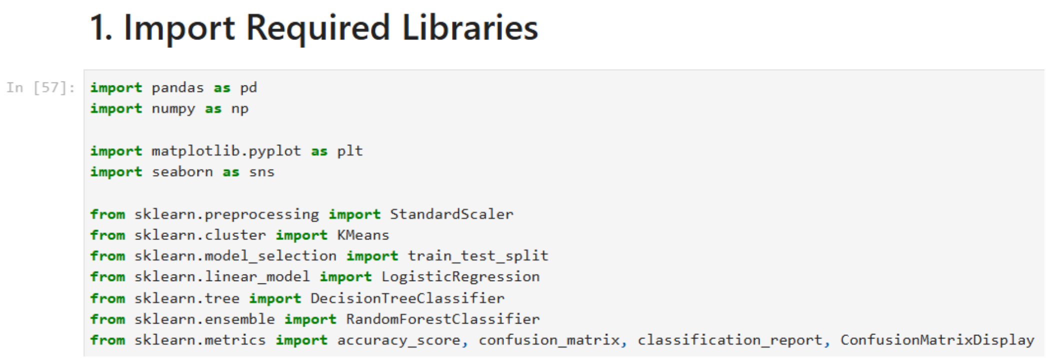
Platform IDE used for the system:

Figure 8: Diagram of Jupyter Notebook Logo

Jupyter Notebook is a user friendly tool for running Python code . It helps to write code in cells, run specific sections without restarting, and see results immediately below the code. Markdown feature is used to add explainations. It is suitable platform to work on several libraries like numpy, matplotlib, pandas. It is to shorten the gap between the user and the type of documentation and search that will help them do their work effectively.

Figure 9: Anaconda Navigator

Anaconda Navigator is a graphical interface which provides environment for Python programs without having to use command lines or to install packages and manage your environments. It is available for Windows, macOS, and Linux only.



1. Import Required Libraries

In [57]:

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, ConfusionMatrixDisplay
```

Figure 10: Import Libraries

The development process of this code involves the following steps:-

3.6.1 Import Libraries and early setup

This code imports all the required libraries for data analysis and machine learning in Python.These are the imported libraries and module used for a data required tasks.

1. Pandas : It is a powerful library used for data manipulation and data analysis. It provides structure like Dataframe and Series for handling the data efficiently. It is used for reading, writing and manipulating data in a table form.
2. Numpy : It is one of the libraries which is used for numerical computation in python. It helps to handle array, matrix and numerical related functions. It is used for handling numerical data and perform mathematical operations.
3. Matplotlib: It is a library that is used for plotting static data visualization of the data. It creates Bargraph, scatterplot, boxplot etc.
4. Seaborn: A data visualization library which provides high level graphics to create informative statistical graphics. It is used to create visual plots like heatmap and shows correlation between two variable.
5. Sklearn.preprocessing.StandardScaler: It is one of the preprocessing class from sci-kit to scale features by removing mean and scaling.
6. Sklearn.cluster.KMeans: It is one of the cluster class from sci-kit to group courses with similar interests.
7. Sklearn.model_selection.train_test_split: It is a function from the sci-kit library to split data into training and testing sets. It is used to prepare data for training and evaluation of machine learning models.

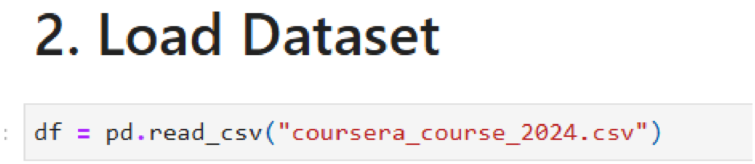
8. Sklearn.linear_model.LogisticRegression: A linear model for binary classification which predicts the probability of binary results if it is 0 or 1.
9. Sklearn.tree.DecisionTreeClassifier: A machine learning model which creates a decision tree based on input features.
10. Sklearn.ensemble.RandomForestClassifier: A machine learning model which is based on combination of different decision trees to improve classification accuracy.
11. Sklearn.metrics.accuracy_score,confusion_matrix,classification_report, ConfusionMatrixDisplay: It is used to calculate confusion matrix which summarize prediction results, classification report is used to generate text summary of classification performance.

3.7 Achieved Results

3.7.1 Load Dataset

The above figure shows that the dataset has been loaded after importing pandas library read)csv function that loads the dataset of coursera_course 2024 dataset needed for the system.

3.7.2 Data Understanding

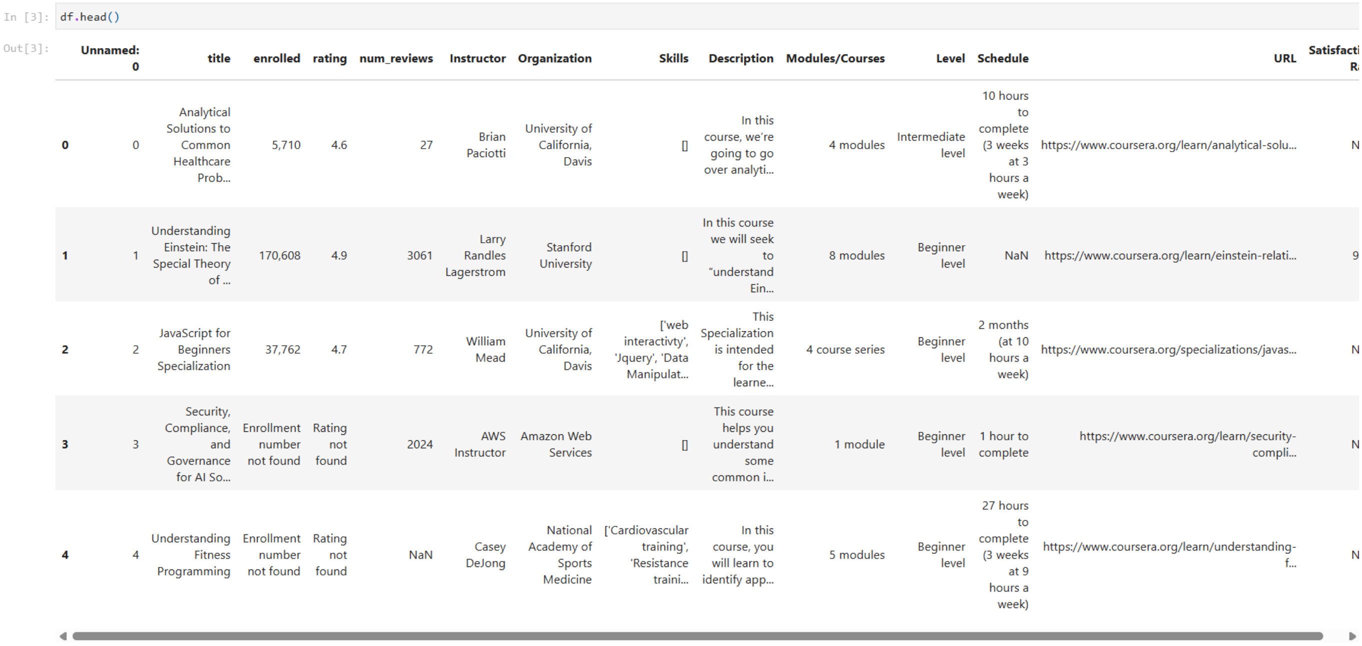


2. Load Dataset

```
df = pd.read_csv("coursera_course_2024.csv")
```

Figure 11: Loading Coursera_coursera_2024 CSV file

After the dataset has been stored in a DataFrame called data, then the first five rows are displayed by using data.head().

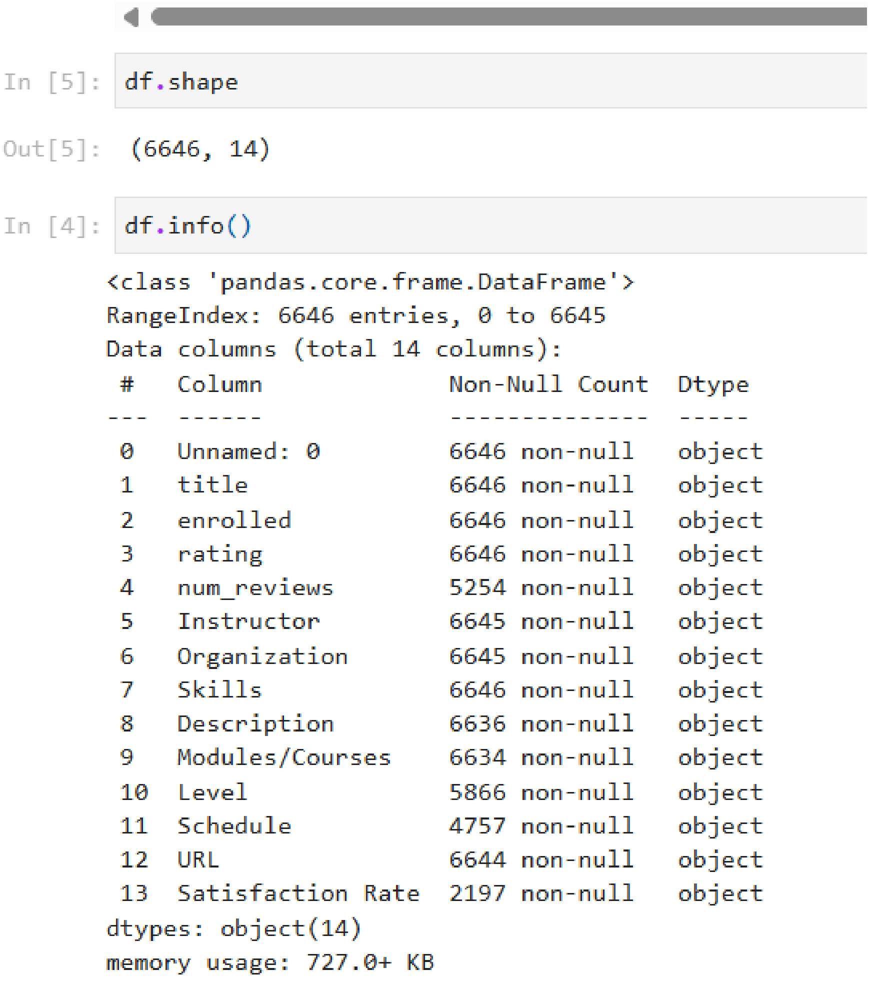


```
In [3]: df.head()
```

Out[3]:

	Unnamed: 0	title	enrolled	rating	num_reviews	Instructor	Organization	Skills	Description	Modules/Courses	Level	Schedule	URL	Satisfacti R
0	0	Analytical Solutions to Common Healthcare Prob...	5,710	4.6	27	Brian Paciotti	University of California, Davis	[]	In this course, we're going to go over analyti...	4 modules	Intermediate level	10 hours to complete (3 weeks at 3 hours a week)	https://www.coursera.org/learn/analytical-solu...	N
1	1	Understanding Einstein: The Special Theory of ...	170,608	4.9	3061	Larry Randles Lagerstrom	Stanford University	[]	In this course we will seek to "understand Ein...	8 modules	Beginner level	NaN	https://www.coursera.org/learn/einstein-relati...	9
2	2	JavaScript for Beginners Specialization	37,762	4.7	772	William Mead	University of California, Davis	['web interactivty', 'Jquery', 'Data Manipulat...	This Specialization is intended for the learne...	4 course series	Beginner level	2 months (at 10 hours a week)	https://www.coursera.org/specializations/javas...	N
3	3	Security, Compliance, and Governance for AI So...	Enrollment number not found	Rating not found	2024	AWS Instructor	Amazon Web Services	[]	This course helps you understand some common i...	1 module	Beginner level	1 hour to complete	https://www.coursera.org/learn/security-compli...	N
4	4	Understanding Fitness Programming	Enrollment number not found	Rating not found	NaN	Casey DeJong	National Academy of Sports Medicine	['Cardiovascular training', 'Resistance traini...	In this course, you will learn to identify app...	5 modules	Beginner level	27 hours to complete (3 weeks at 9 hours a week)	https://www.coursera.org/learn/understanding-f...	N

Figure 12:Diplaying first five rows

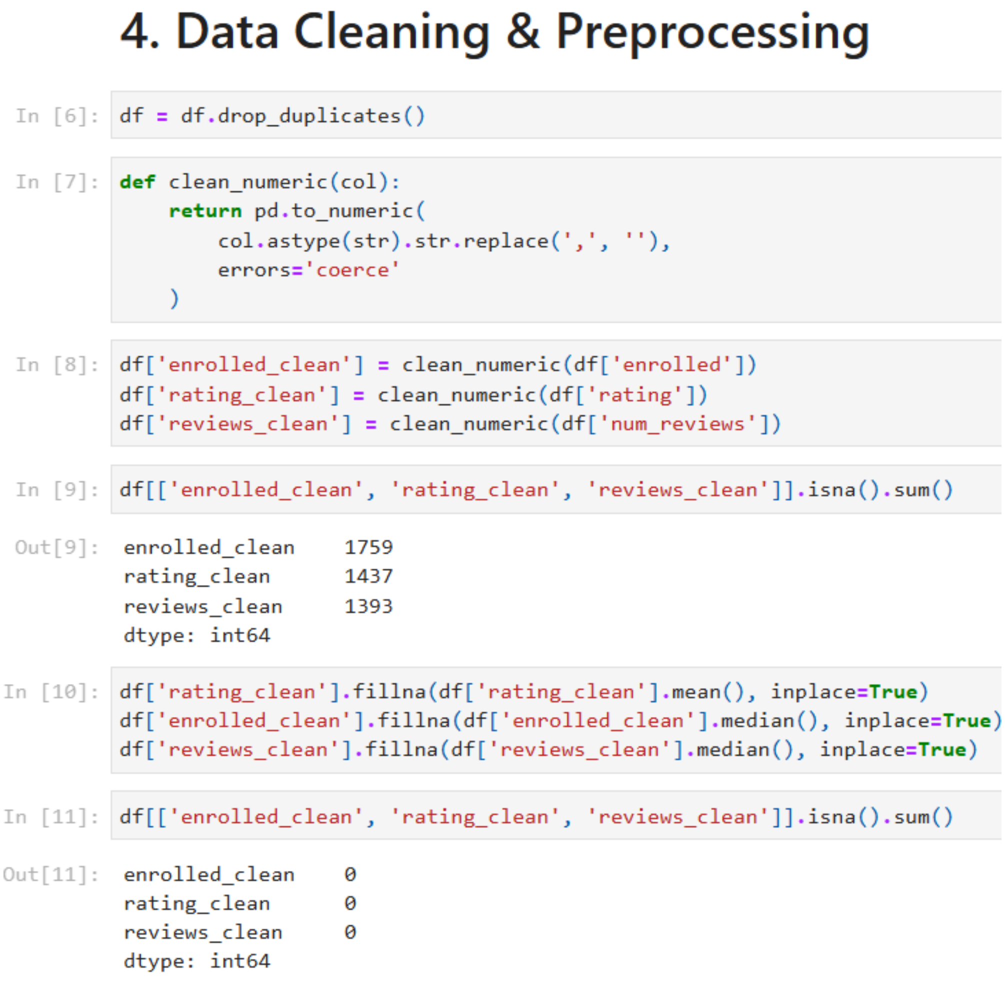


```
In [5]: df.shape

Out[5]: (6646, 14)

In [4]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6646 entries, 0 to 6645
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Unnamed: 0         6646 non-null   object
 1   title              6646 non-null   object
 2   enrolled           6646 non-null   object
 3   rating             6646 non-null   object
 4   num_reviews        5254 non-null   object
 5   Instructor         6645 non-null   object
 6   Organization       6645 non-null   object
 7   Skills             6646 non-null   object
 8   Description        6636 non-null   object
 9   Modules/Courses    6634 non-null   object
 10  Level              5866 non-null   object
 11  Schedule           4757 non-null   object
 12  URL                6644 non-null   object
 13  Satisfaction Rate  2197 non-null   object
dtypes: object(14)
memory usage: 727.0+ KB
```

Figure 13:Displaying the data shape and info of the dataset

The above figure shows the shape of the dataset using .shape method where there is 6646 rows and 14 rows.The df.info method is used to display the name of dataframe, index range from 0 to 6645, total number of column (14), name of the column and their datatype, not null counts and memory used by the dataframe.

3.7.3 Data Cleaning and Preprocessing



4. Data Cleaning & Preprocessing

```
In [6]: df = df.drop_duplicates()
```

```
In [7]: def clean_numeric(col):
            return pd.to_numeric(
                col.astype(str).str.replace(',', ''),
                errors='coerce'
            )
```

```
In [8]: df['enrolled_clean'] = clean_numeric(df['enrolled'])
        df['rating_clean'] = clean_numeric(df['rating'])
        df['reviews_clean'] = clean_numeric(df['num_reviews'])
```

```
In [9]: df[['enrolled_clean', 'rating_clean', 'reviews_clean']].isna().sum()

Out[9]: enrolled_clean    1759
        rating_clean      1437
        reviews_clean     1393
        dtype: int64
```

```
In [10]: df['rating_clean'].fillna(df['rating_clean'].mean(), inplace=True)
         df['enrolled_clean'].fillna(df['enrolled_clean'].median(), inplace=True)
         df['reviews_clean'].fillna(df['reviews_clean'].median(), inplace=True)
```

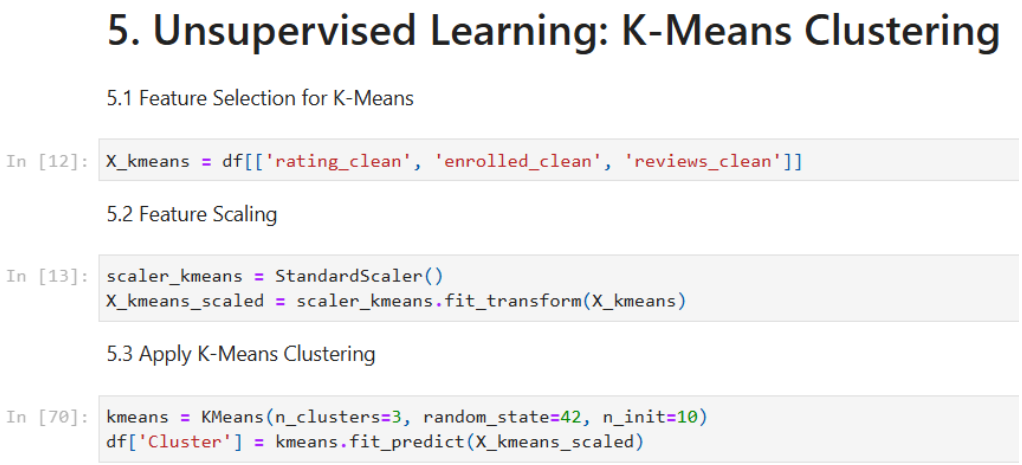
```
In [11]: df[['enrolled_clean', 'rating_clean', 'reviews_clean']].isna().sum()

Out[11]: enrolled_clean    0
         rating_clean      0
         reviews_clean     0
         dtype: int64
```

Figure 14: Dataset cleaning process before applying machine learning models

In the above figure, duplicate rows were removed from the dataset using drop_duplicates() method. Duplicate data affects the result of machine learning models by giving importance to repeating records. It helps to make dataset more accurate and reliable. Then , some columns such as enrolled students, rating and reviews are stored as text. A function clean_numeric function removes commas from the values and converts them into numerical values. If any value cannot be changed then it is changed into NaN. It is important step because machine learning models work with numerical data only. The cleaned function is needed to analyse and model training. After converting the data the number of missing values in each cleaned column was checked. This step help to understand how much data is missing and if it needs to be handled before applying machine learning algorithms. Missing values were handled using simple methods. The missing values of rating_clean were filled by using mean value as ratings follow average pattern. The enrolled and review cleans were filled using median value which helps to reduce the effect of extreme values.

Clean data helps improve model performance and produces more meaningful recommendation results.

3.7.4 Unsupervised Learning : K Means Clustering



5. Unsupervised Learning: K-Means Clustering

5.1 Feature Selection for K-Means

```
In [12]: X_kmeans = df[['rating_clean', 'enrolled_clean', 'reviews_clean']]
```

5.2 Feature Scaling

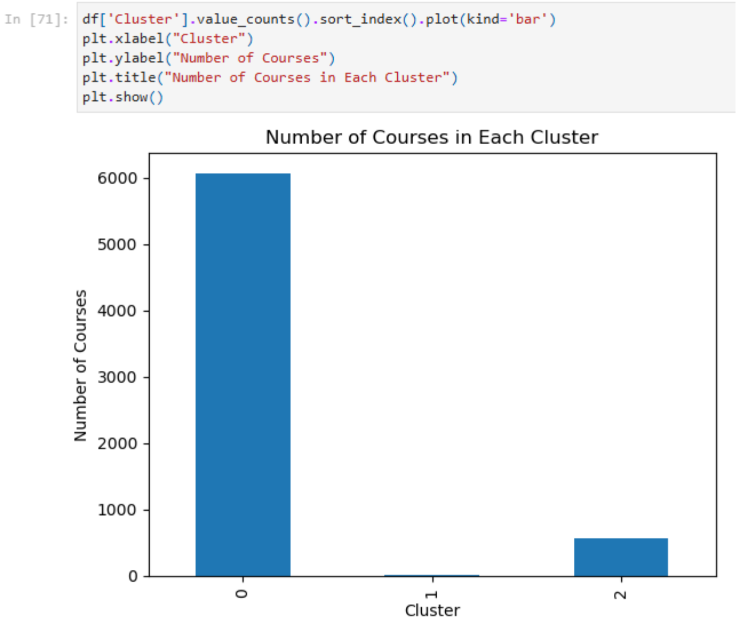
```
In [13]: scaler_kmeans = StandardScaler()
         X_kmeans_scaled = scaler_kmeans.fit_transform(X_kmeans)
```

5.3 Apply K-Means Clustering

```
In [70]: kmeans = KMeans(n_clusters=3, random_state=42, n_init=10)
         df['Cluster'] = kmeans.fit_predict(X_kmeans_scaled)
```

Figure 15: Displaying feature selection , scaling and applying K-Means Clustering

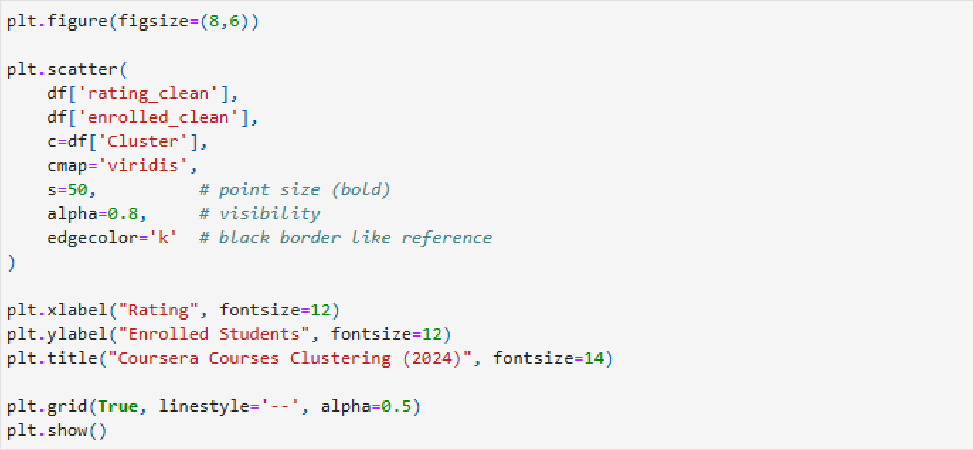
In this figure from all the column, column like rating_clean, enrolled_clean, reviews_clean are choosen as feature because they are the main factors which helps to recommend the suitable courses. Afte the feature which is X_kmeans is displayed then the features is applied by K- Means Clustering.



```
In [71]: df['Cluster'].value_counts().sort_index().plot(kind='bar')
         plt.xlabel("Cluster")
         plt.ylabel("Number of Courses")
         plt.title("Number of Courses in Each Cluster")
         plt.show()
```

Figure 16:Cluster Distribution of Bar Chart

The bar graph shows how many courses fall into each cluster created by the K means clustering algorithm. It helps to understand how the courses are distributed across the clusters. One cluster contain more courses compare to others which means many courses share similar features such as rating , enrollment numbers and review numbers.



```
plt.figure(figsize=(8,6))

plt.scatter(
    df['rating_clean'],
    df['enrolled_clean'],
    c=df['Cluster'],
    cmap='viridis',
    s=50,          # point size (bold)
    alpha=0.8,     # visibility
    edgecolor='k'  # black border like reference
)

plt.xlabel("Rating", fontsize=12)
plt.ylabel("Enrolled Students", fontsize=12)
plt.title("Coursera Courses Clustering (2024)", fontsize=14)

plt.grid(True, linestyle='--', alpha=0.5)
plt.show()
```

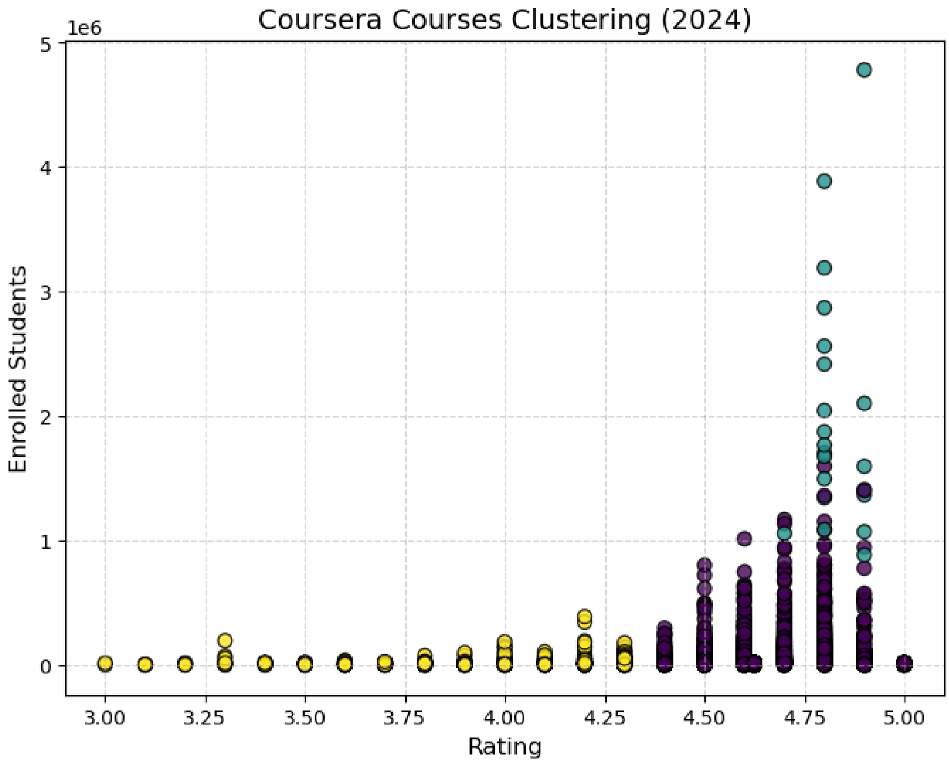
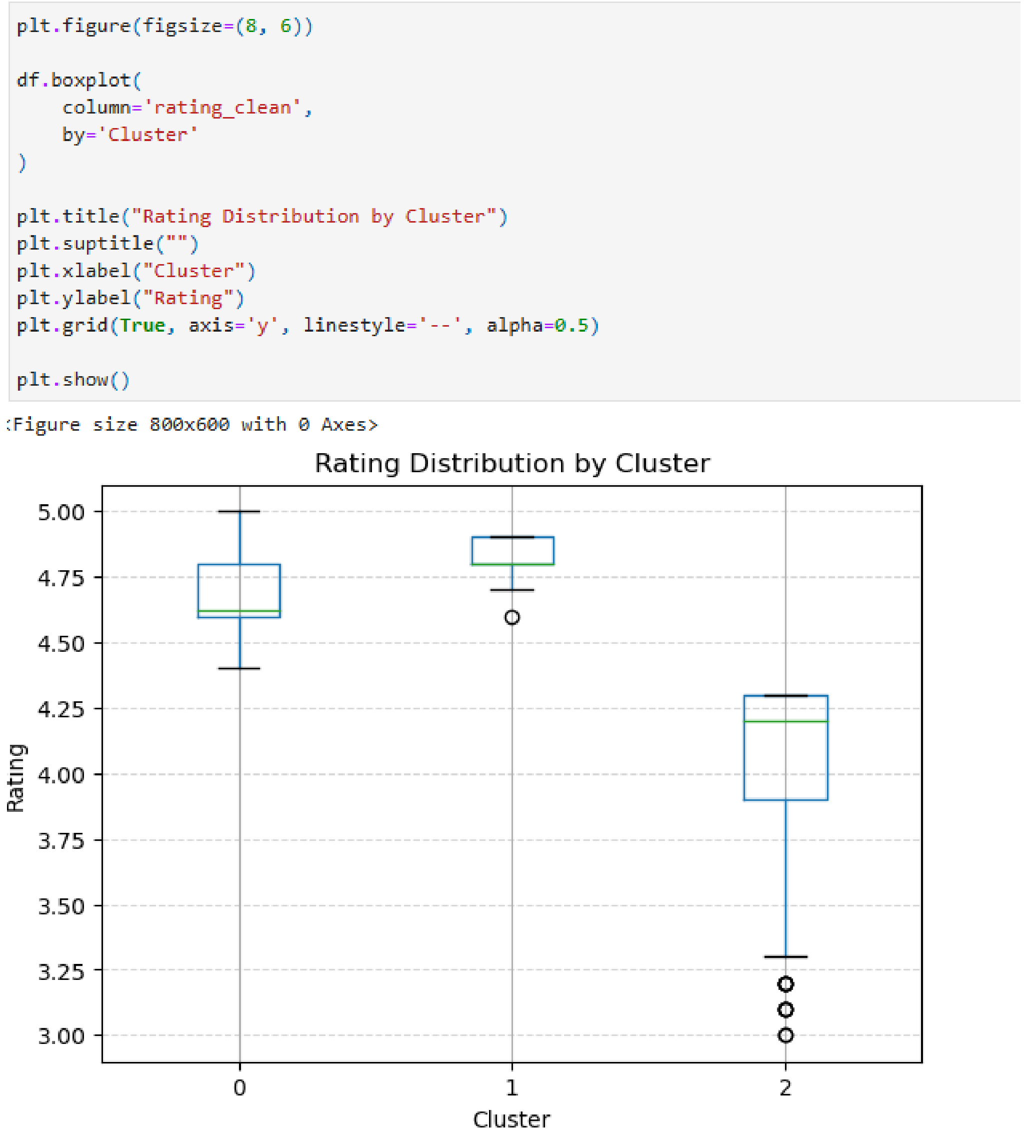


Figure 17: Scatter Plot of Rating and Enrolled Students

This scatter plot shows the relationship between course ratings and the number of enrolled students. Each dot represents a course and the color shows which cluster belong to which course. Course with higher enrollment and ratings appears together in the same cluster. This figure helps to visualize how K Means collects courses based on their quality and popularity.

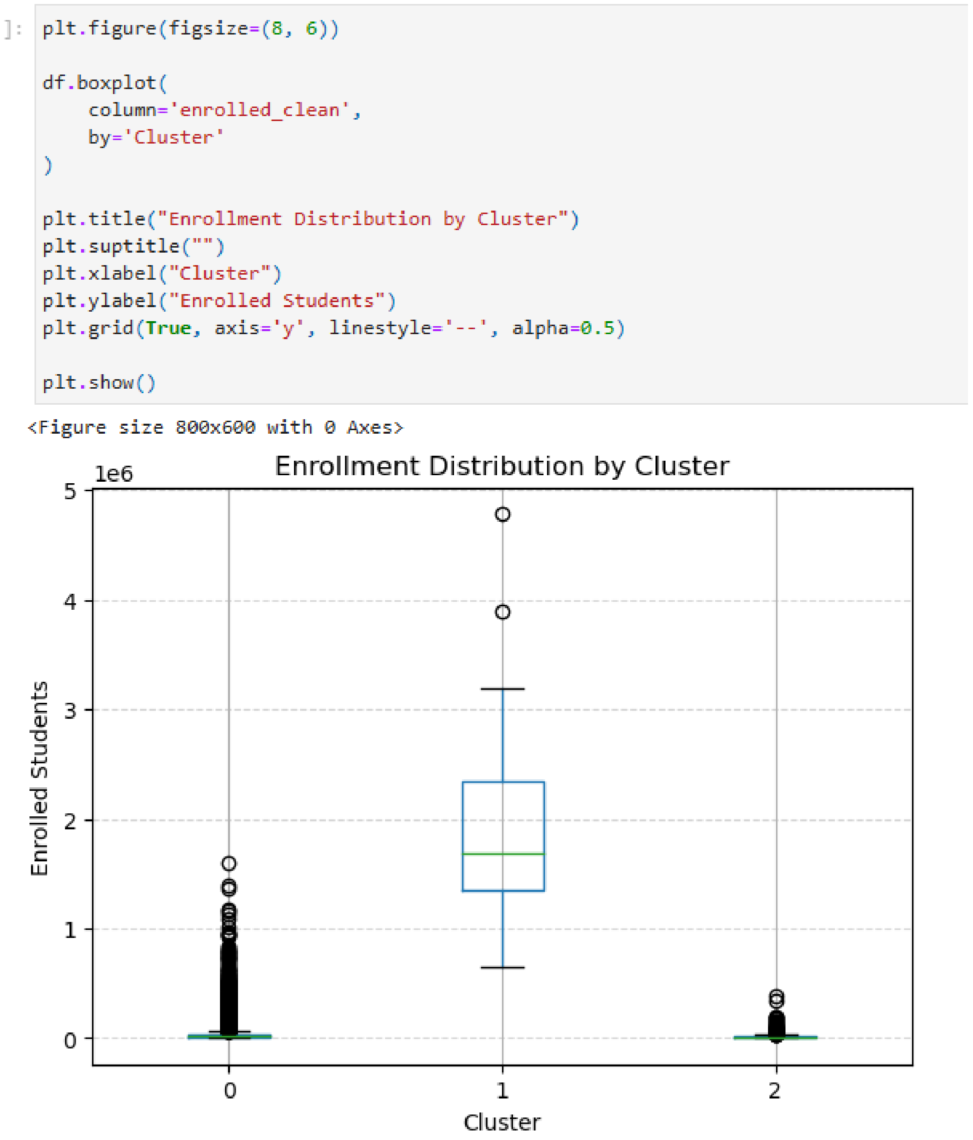


```
plt.figure(figsize=(8, 6))

df.boxplot(
    column='rating_clean',
    by='Cluster'
)

plt.title("Rating Distribution by Cluster")
plt.suptitle("")
plt.xlabel("Cluster")
plt.ylabel("Rating")
plt.grid(True, axis='y', linestyle='--', alpha=0.5)

plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```

Figure 18: Boxplot of Rating by Cluster

This boxplot compares the rating distribution across the different clusters. It shows median rating, how data is spread, and outliers for each cluster. One cluster has higher ratings which means those courses are better rated by learners. Another cluster shows lower or more different ratings as compared to higher ratings.



```
plt.figure(figsize=(8, 6))

df.boxplot(
    column='enrolled_clean',
    by='Cluster'
)

plt.title("Enrollment Distribution by Cluster")
plt.suptitle("")
plt.xlabel("Cluster")
plt.ylabel("Enrolled Students")
plt.grid(True, axis='y', linestyle='--', alpha=0.5)

plt.show()
```

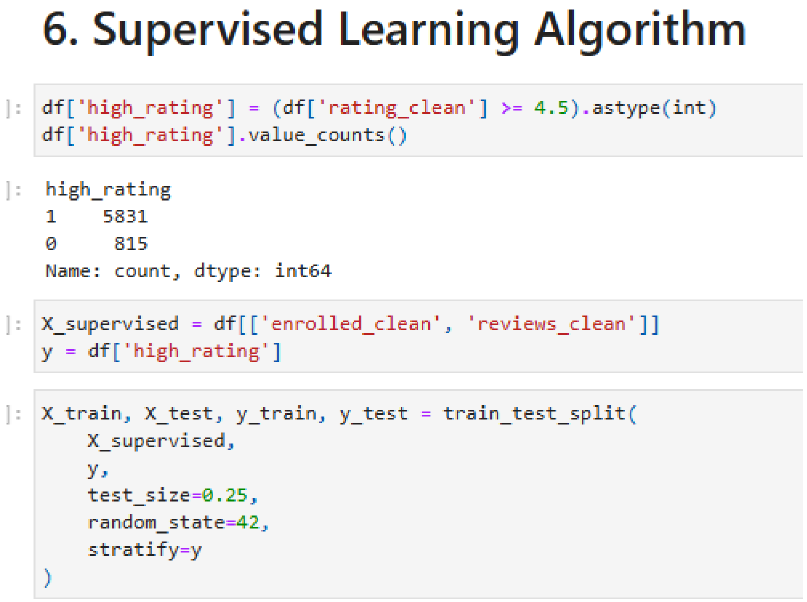
```
<Figure size 800x600 with 0 Axes>
```

Figure 19: Boxplot of Enrollment by Cluster

This boxplot shows how the number of enrollments varies between clusters. Some clusters includes courses with very high enrollment whereas other includes courses with lower enrollment numbers. The wide range and outliers shows that a few courses are well known as compared to other. This figure helps to explain how clustering separates highly popular courses from lesser popular course.

3.7.5 Supervised Learning

- Logistic Regression



6. Supervised Learning Algorithm

```
]: df['high_rating'] = (df['rating_clean'] >= 4.5).astype(int)
   df['high_rating'].value_counts()
```

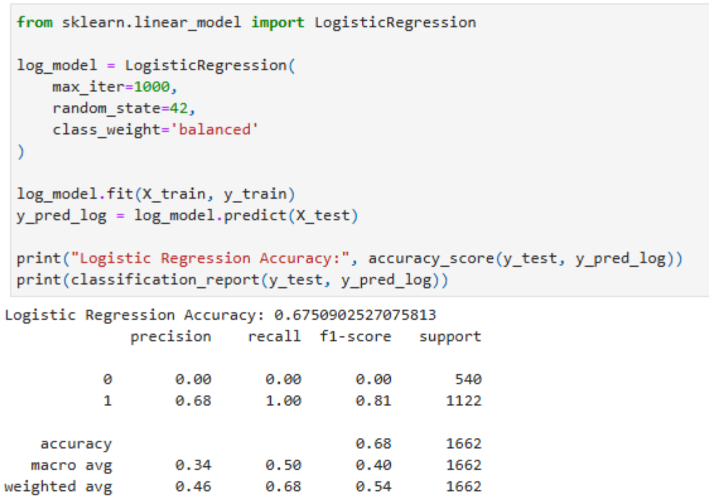
```
]:  high_rating
    1    5831
    0     815
    Name: count, dtype: int64
```

```
]: X_supervised = df[['enrolled_clean', 'reviews_clean']]
   y = df['high_rating']
```

```
]: X_train, X_test, y_train, y_test = train_test_split(
       X_supervised,
       y,
       test_size=0.25,
       random_state=42,
       stratify=y
   )
```

Figure 20: Train Test split

In this dataset, data is split into training and testing sets where they are split into 80- 20. Test_size means 20 % od the data will be used for testing and 80% for training the data. It controls the randomness of the data splitting process, random_state equals to 42 which is commonly used to fix the randomness seed for consistent results.

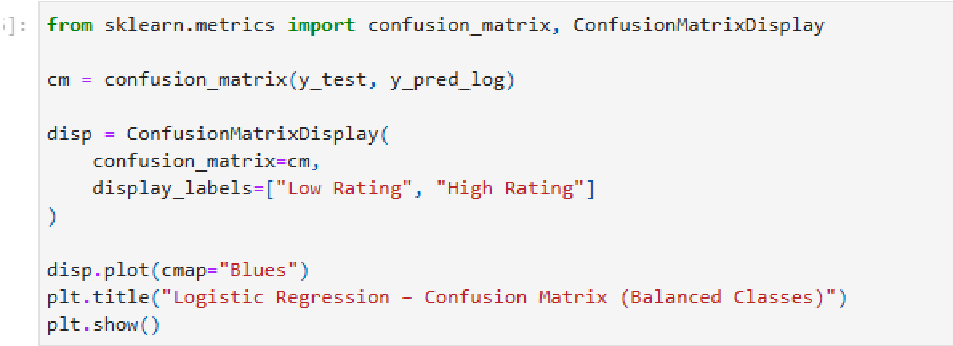


```
from sklearn.linear_model import LogisticRegression

log_model = LogisticRegression(
    max_iter=1000,
    random_state=42,
    class_weight='balanced'
)

log_model.fit(X_train, y_train)
y_pred_log = log_model.predict(X_test)

print("Logistic Regression Accuracy:", accuracy_score(y_test, y_pred_log))
print(classification_report(y_test, y_pred_log))
```

```
Logistic Regression Accuracy: 0.6750902527075813
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       540
           1       0.68      1.00      0.81      1122

    accuracy                           0.68      1662
   macro avg       0.34      0.50      0.40      1662
weighted avg       0.46      0.68      0.54      1662
```

Figure 21: Using Logistic Regression Accuracy

In this logistic regression model, it is used to train and evaluate recommendations. At first it trains the model using the input features which is X_train and labels which is y_train. It makes recommendation on the test data X_test which is stored in y_pred_log. It calculates the model by showing how many prediction were made. It shows classification report to evaluate the models performance. The logistic regression accuracy score is 0.675.



```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay

cm = confusion_matrix(y_test, y_pred_log)

disp = ConfusionMatrixDisplay(
    confusion_matrix=cm,
    display_labels=["Low Rating", "High Rating"]
)

disp.plot(cmap="Blues")
plt.title("Logistic Regression – Confusion Matrix (Balanced Classes)")
plt.show()
```

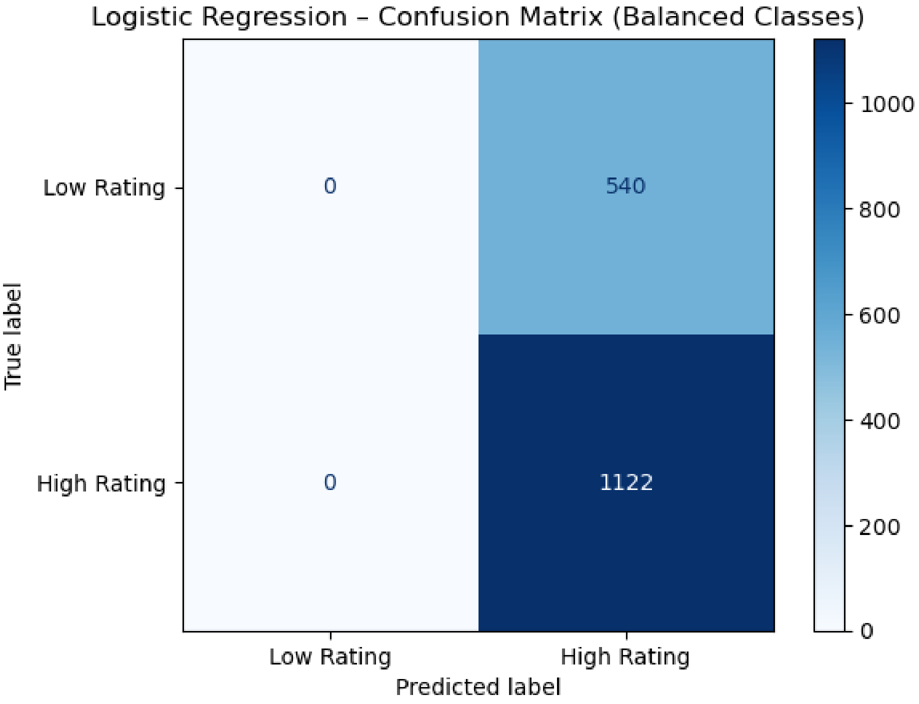
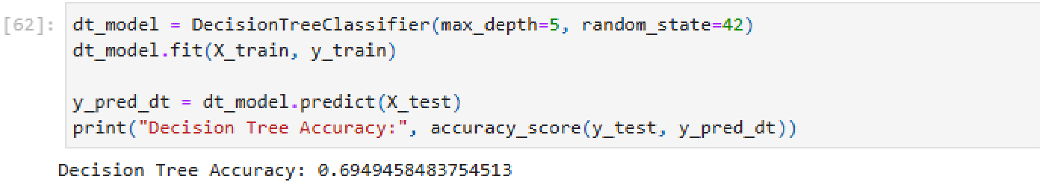


Figure 22: Heatmap of Logistic Regression

This heatmap figure shows the confusion matrix of Logistic regression model. It compares the predicted course rating with the actual ratings. Most courses are predicted as high ratings which means the model is biased toward majority class. It predicts many high rated courses but fails to identify low rated courses. This shows that logistic regression model which is simple and fast but it does not perform well when classes are imbalanced. Dark color shows higher prediction rate. The model struggles to separate low rating course clearly .

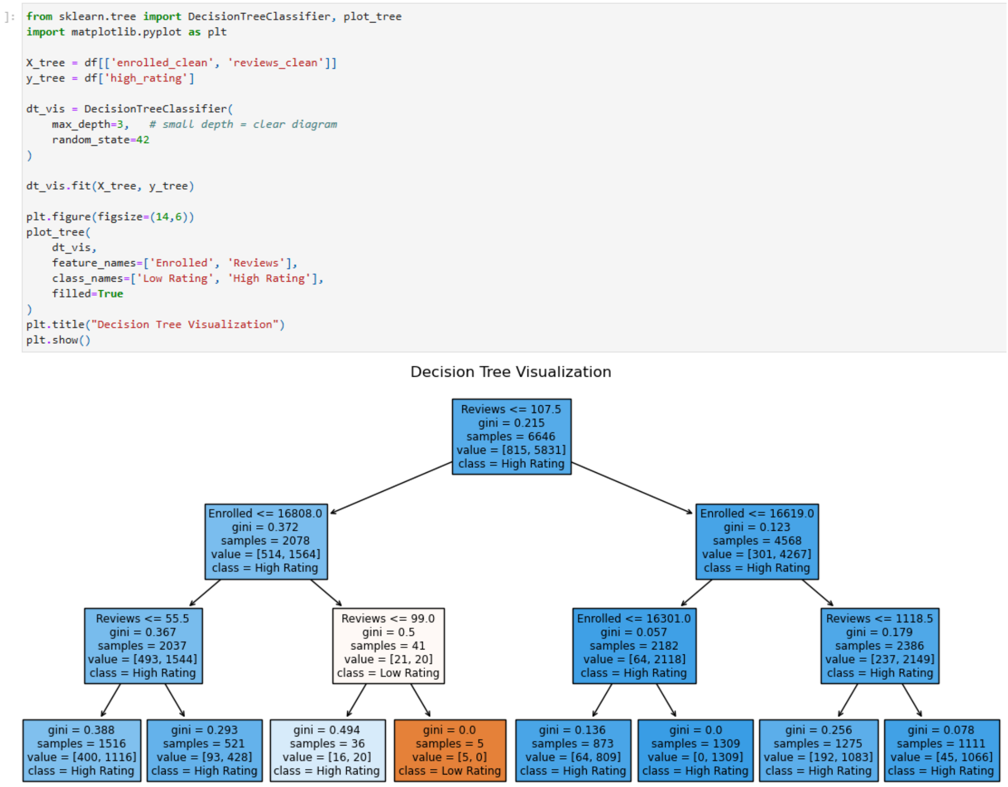
- Decision Tree



```
[62]: dt_model = DecisionTreeClassifier(max_depth=5, random_state=42)
      dt_model.fit(X_train, y_train)

      y_pred_dt = dt_model.predict(X_test)
      print("Decision Tree Accuracy:", accuracy_score(y_test, y_pred_dt))

    Decision Tree Accuracy: 0.6949458483754513
```

Figure 23: Using Decision Tree Classifier

Here, in this code it shows the accuracy score of logistic regression of the Decision Tree model. The accuracy score is 0.69 which is slightly better than logistic regression model. It shows that decision trees can work on more complex pattern in the data.



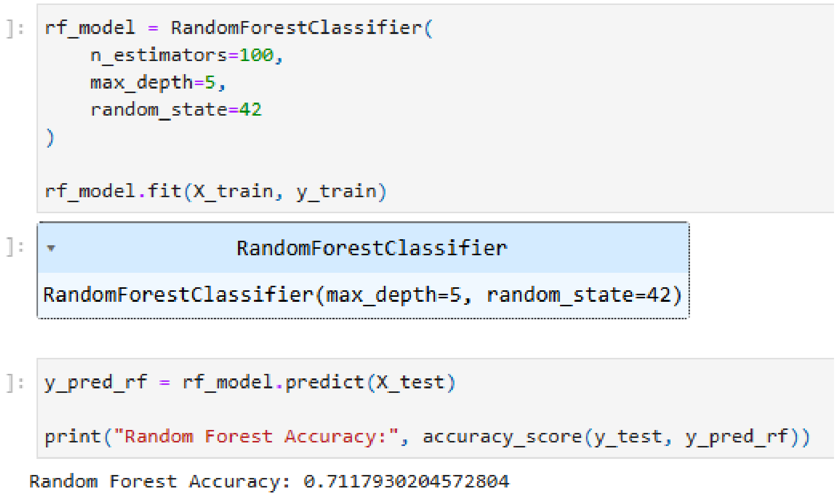
```
]: from sklearn.tree import DecisionTreeClassifier, plot_tree
   import matplotlib.pyplot as plt

   X_tree = df[['enrolled_clean', 'reviews_clean']]
   y_tree = df['high_rating']

   dt_vis = DecisionTreeClassifier(
       max_depth=3,   # small depth = clear diagram
       random_state=42
   )

   dt_vis.fit(X_tree, y_tree)

   plt.figure(figsize=(14,6))
   plot_tree(
       dt_vis,
       feature_names=['Enrolled', 'Reviews'],
       class_names=['Low Rating', 'High Rating'],
       filled=True
   )
   plt.title("Decision Tree Visualization")
   plt.show()
```

Decision Tree Visualization
Reviews <= 107.5 gini = 0.215 samples = 6646 value = [815, 5831] class = High Rating
Enrolled <= 16808.0 gini = 0.372 samples = 2078 value = [514, 1564] class = High Rating
Enrolled <= 16619.0 gini = 0.123 samples = 4568 value = [301, 4267] class = High Rating
Reviews <= 55.5 gini = 0.367 samples = 2037 value = [493, 1544] class = High Rating
Reviews <= 99.0 gini = 0.5 samples = 41 value = [21, 20] class = Low Rating
Enrolled <= 16301.0 gini = 0.057 samples = 2182 value = [64, 2118] class = High Rating
Reviews <= 1118.5 gini = 0.179 samples = 2386 value = [237, 2149] class = High Rating
gini = 0.388 samples = 1516 value = [400, 1116] class = High Rating
gini = 0.293 samples = 521 value = [93, 428] class = High Rating
gini = 0.494 samples = 36 value = [16, 20] class = High Rating
gini = 0.0 samples = 5 value = [5, 0] class = Low Rating
gini = 0.136 samples = 873 value = [64, 809] class = High Rating
gini = 0.0 samples = 1309 value = [0, 1309] class = High Rating
gini = 0.256 samples = 1275 value = [192, 1083] class = High Rating
gini = 0.078 samples = 1111 value = [45, 1066] class = High Rating

Figure 24: Diagram of decision tree

The above diagram shows how the decision tree makes decisions based on the features like number of enrollments and number of reviews. Here each node splits the data based on a given condition and the final leaf nodes gives the predicted ratings of the course. It clearly shows decision rules which make the models easy to understand. It is useful to explain recommendation to learners.

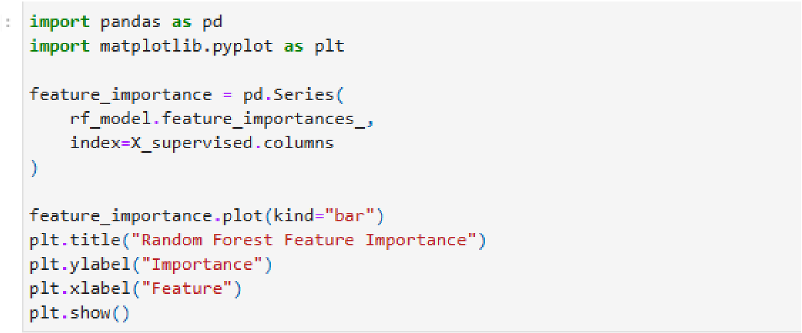
- Random Forest



```
]: rf_model = RandomForestClassifier(
       n_estimators=100,
       max_depth=5,
       random_state=42
   )

   rf_model.fit(X_train, y_train)

]: RandomForestClassifier
   RandomForestClassifier(max_depth=5, random_state=42)

]: y_pred_rf = rf_model.predict(X_test)

   print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))

  Random Forest Accuracy: 0.7117930204572804
```

Figure 25: Using RandomForest Classifier

This code show sthe accuracy of the Randomm Forest model which is about 0.71 score . It combines multiple decision trees and takes a majority votes. It performs better than Decision Tree and Logistic Regression. It reduces overfitting and gives correct predictions.



```
import pandas as pd
import matplotlib.pyplot as plt

feature_importance = pd.Series(
    rf_model.feature_importances_,
    index=X_supervised.columns
)

feature_importance.plot(kind="bar")
plt.title("Random Forest Feature Importance")
plt.ylabel("Importance")
plt.xlabel("Feature")
plt.show()
```

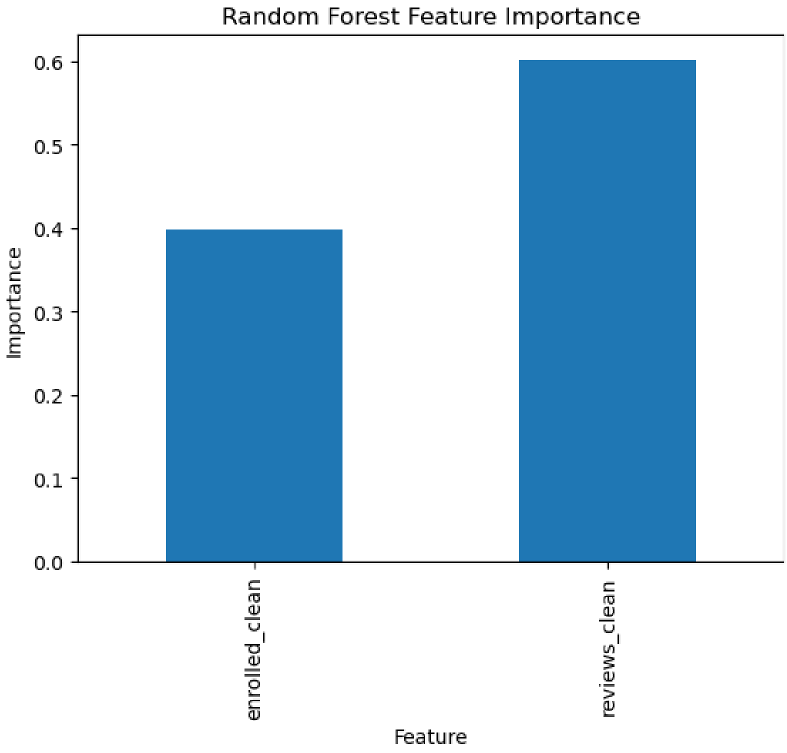
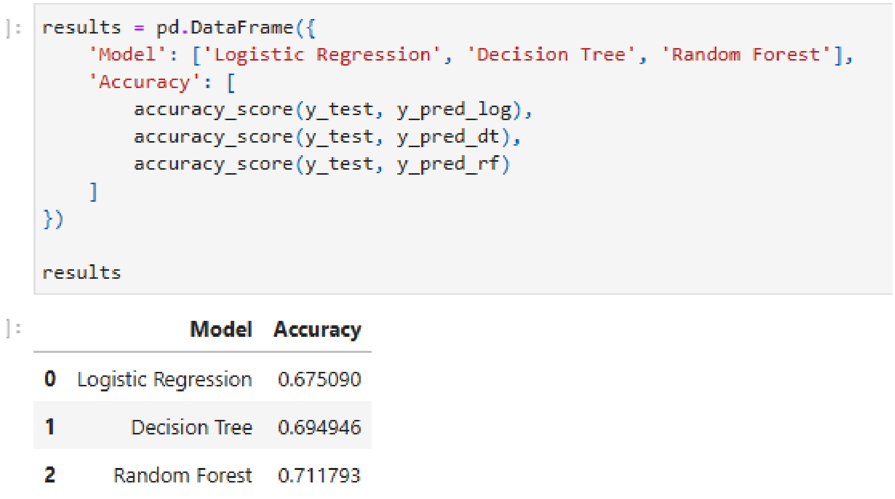


Figure 26: Bar graph of Random Forest Feature Importance

This bar graph shows the importance of each features used in the Random Forest model . Features importance shows how much each feature helps to the final recommendation made by the model. Features like number of enrollments and number of review have higher importance as compared to other features. These two features plays a major role in recommending if the course should be rate as high or low. Courses with more reviews and the high enrollment numbers are most likely to be recommended. It calculates features importance by combining the results from different decision trees.It helps to predict course ratings. It helps to reduce overfitting and improve correct prediction. It helps to make us understand the feature that is more useful and focus on that useful data and improve the quality and correct data for the recommendation of the courses.

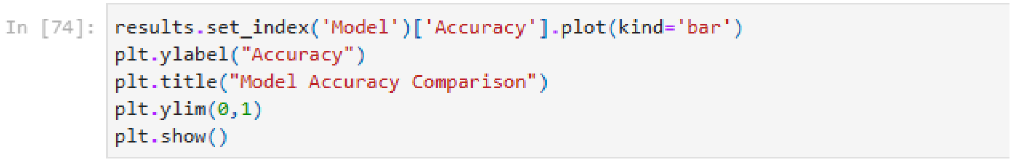


```
results = pd.DataFrame({
    'Model': ['Logistic Regression', 'Decision Tree', 'Random Forest'],
    'Accuracy': [
        accuracy_score(y_test, y_pred_log),
        accuracy_score(y_test, y_pred_dt),
        accuracy_score(y_test, y_pred_rf)
    ]
})

results
```

```
               Model  Accuracy
0  Logistic Regression  0.675090
1        Decision Tree  0.694946
2        Random Forest  0.711793
```

Figure 27: Accuracy among 3 models

This tables compares the accuracy of three supervised learning models where the score of logistic regression is 0.675, decision tree is 0.695, Random forest is 0.71. Amon all these models Random forest performs the best which makes it the most suitable for supervised learning model for this project.



```
In [74]: results.set_index('Model')['Accuracy'].plot(kind='bar')
         plt.ylabel("Accuracy")
         plt.title("Model Accuracy Comparison")
         plt.ylim(0,1)
         plt.show()
```

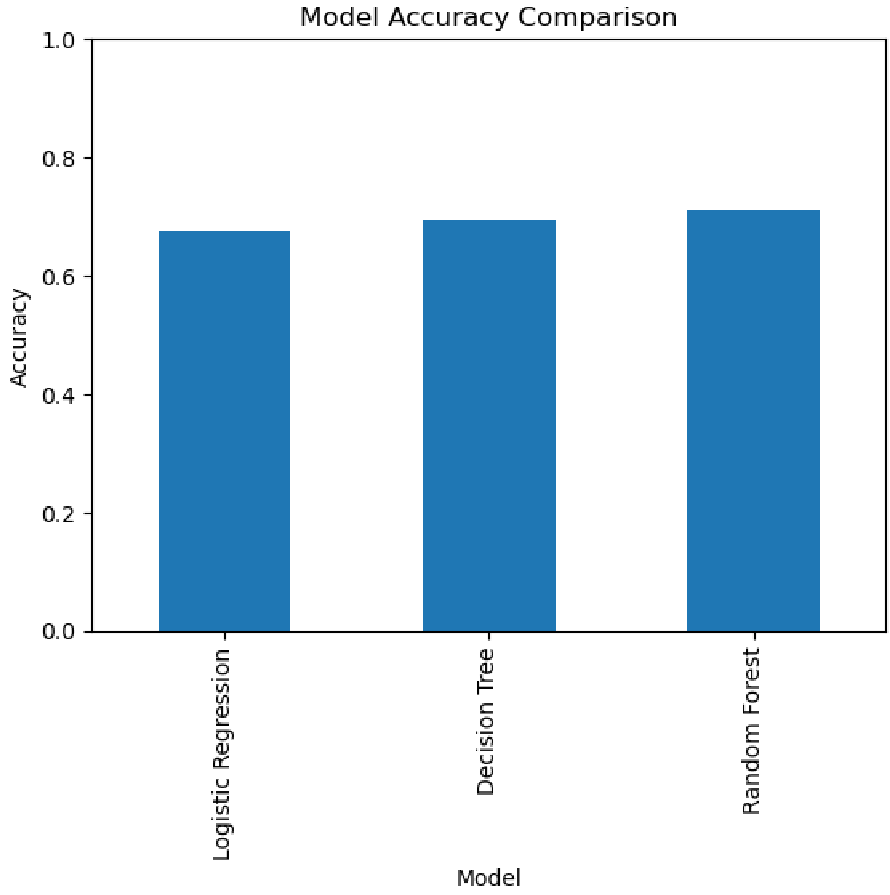


Figure 28: Displaying Bar graph of model accuracy comparison

This bar graph compares the accuracy of all the three supervised learning models which are Logistic Regression, Decision Tree and Random Forest. Accuracy measures how often the model correctly predicts the rating of course. It shows that the Random Forest has the highest accuracy then Decision Tree and at last Logistic Regression which has the lowest accuracy among the three models. This shows that ensemble methods like Random Forest that performs better because they combine multiple decision tree and has less chance of getting errors. This comparison shows that the Random Forest is chosen as best supervised learning model. It provides better performance which handles noise and imbalanced data effectively. This results helps the use of more advanced model for course recommendation tasks in future.

4. Conclusion

4.1 Analysis of the Work Done

This coursework has presented the design of an AI based course recommendation system to improve personalised learning on online learning platform such as Coursera, Udemy and EdX. The project explores how recommendation systems have changed from basic rule based systems to more advanced hybrid techniques which adapt better to learner needs.

While working on this project, most of the existing recommendation system performs well and has advanced techniques. But still challenges like scalability, cold-start problem like missing values from dataset are still present in modern systems. Course rating and enrollment number helps to find popular and best quality courses. It also supports similarity based recommendation to organisation and instructor related information. But due to lack of detailed information of learner interaction, collaborative filtering is not enough so hybrid approach is considered suitable to make it accurate and better for decision making.The Coursera Course 2024 dataset contains 6645 records was analysed and cleaned using various techniques. Exploratory Data Analysis helps to find relationship between ratings, reviews, enrollment numbers which helps in the selection of the algorithm. Supervised learning algorithm such as Logistic Regression, Decision Tree and Random Forest are selected along with unsupervised learning algorithm like clustering . Evaluation metrics which includes precision, recall, F1 score were used to make the system performance better.Therefore, the system shows how AI can improve course recommendation better by making them more personalised, scalable and efficient. It helps learner choose suitable courses , reduce dropout rates and helps platform in providing better learning experiences. In future improvement can be made using learner interaction data and applying more advanced learning models.

4.2 Further Work

This project works well from the beginning but it can still be improved in many ways. One improvement can be made using course reviews and student comments to understand what learners actually think about a course. It can also look at learner behaviour like what kind of course they click , browse or are spending their time on to give better suggestion. Another improvement is to explain recommendations clearly so learners know why a course is suggested. Many problems like data privacy and fairness should be taken very seriously so the system is safe and trusted. In the future, the system could be turned into a real application with a simple border which users can easily use. It also includes courses from various platforms instead of one platform like Coursera so the learners have more option in one places.

5. References

Abbakumov, D., 2014. The solution of the “cold start problem” in e-Learning. *Procedia - Social and Behavioral Sciences,* pp. 1225-1231.

Buitrago, M. & Chiappe, A., 2019. Representation of knowledge in digital educational environments: A systematic review of literature.. *Australasian Journal of Educational Technology,* 4(35), pp. 46-62.

Byjus, 2022. *An introduction to Msword.* [Online]
Available at: https://byjus.com/govt-exams/microsoft-word/
[Accessed 6 January 2026].

Celik, B. & Cagiltay, K., 2024. Uncovering MOOC Completion: A Comparative Study of Completion Rates from Different Perspective. *Open Praxis,* 29 August, 16(3), p. 445–456.

Charntaweekhun , K. & Wangsiripitak, . S., 2006. Visual Programming using Flowchart. *2006 International Symposium on Communications and Information Technologies, Bangkok, Thailand,* pp. 1062-1065.

Coursera, 2024. *Coursera Reports Fourth Quarter and Full Year 2023 Financial Results.* [Online]
Available at: https://investor.coursera.com/news/news-details/2024/Coursera-Reports-Fourth-Quarter-and-Full-Year-2023-Financial-Results/default.aspx
[Accessed 2 January 2026].

George, G. & La, . A. . M., 2024. PERKC: Personalized kNN With CPT for Course Recommendations in Higher Education. *IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES,* Volume 17, pp. 885-890.

Islam, M. S. & Hosen , A. S. M. S., 2022. Personalized Course Recommendation System: A Multi-Model Machine Learning Framework for Academic Success. *Digital ,* 14(5), p. 2907.

Khan, M. A. Z. & Polyzou, A., 2024. Session-Based Course Recommendation Using Learner Interaction Sequences. *Electronics,* 3762(18), p. 13.

Liu, J., Zhang, H. & Liu, Z., 2020. Research on Online Learning Resource Recommendation Method Based on Wide & Deep and Elmo Model. *Journal of Physics: Conference Series,* 1437(1).

Peterson, R., 2013. *Why Do Students Drop Out of MOOCs?.* [Online]
Available at:
https://www.nas.org/blogs/article/why_do_students_drop_out_of_moocs
[Accessed 10 December 2025].

Ren, X. et al., 2022. A Deep Learning Framework for Multimodal Course Recommendation Based on LSTM+Attention. *Sustainability,* 14(12), p. 2907.

ScienceDirect, 2012. *State Transition Diagram.* [Online] Available at: https://www.sciencedirect.com/topics/computer-science/state-transition-diagram
[Accessed 13 December 2025].

Shah, D., 2021. *By The Numbers: MOOCs in 2021.* [Online] Available at: https://www.classcentral.com/report/mooc-stats-2021/
[Accessed 10 December 2025].

Tilahun, L. A. & Sekeroglu, B., 2020. An Intelligent Course Advising System Based on Expert Systems. *SN Computer Science,* 1(4), p. 1–15.

VanderPlas, J., 2016. *Python Data Science HandbookEssential Tools for Working with Data.* First Edition ed. s.l.:O'Reilly Media.

Xu, J., Xing, T. & Schaar, M. v. d., 2016. Personalized Course Sequence Recommendations. *IEEE Transactions on Signal Processing,* 64(20), p. 5340–5352.

Ziegler, N. et al., 2017. Interdisciplinary Research at the Intersection of CALL, NLP, and SLA: Methodological Implications From an Input Enhancement Project. *Journal of Learning Analytics,* 4(1), pp. 1-15.