LONDON METROPOLITAN UNIVERSITY

Module Code & Module Title

CU6051NP Artificial Intelligence

75% Individual Coursework

Submission: Final Submission

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: 21/01/2026

Assignment Submission Date: 21/01/2026

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 system helps learners in selecting suitable courses based on interests, learning goals. It analyses course metadata and suggest the relevant course based on data patterns which reduces manual searching in Coursera platform. Machine Learning(ML) plays an important role in building recommendation systems by learning the patterns from the historical data and improving the 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 in this project.

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. These models are suitable for the Coursera dataset because the course popularity and the quality are influenced by multiple features rather than the single feature. 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. It is particularly useful when the learner history is unavailable as the system can still recommend related courses by analyzing similarities within the courses data. 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.

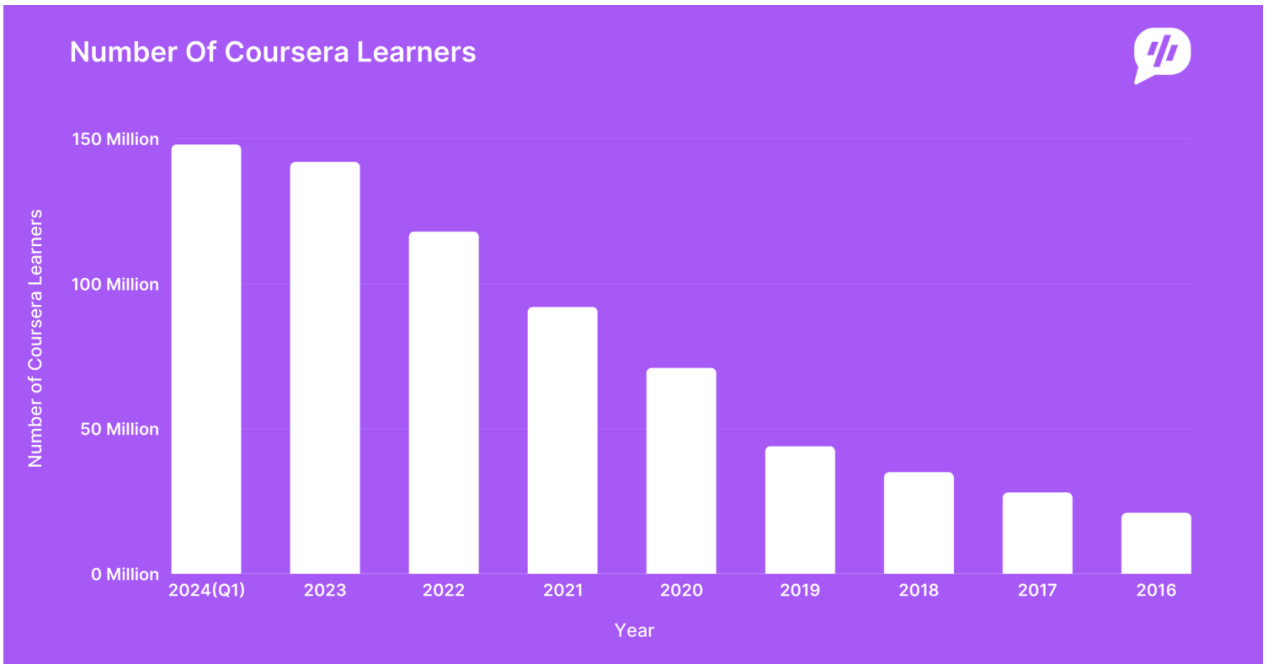


Figure 1: Number of Coursera Learner from 2016-2024 (Coursera, 2024)

The above bar graph shows the data of total numbers of coursera learner from 2016 -2024 across the world. It shows a slight increase in Coursera learners over the time with steady growth after 2020 due to adaption of online learning platforms during COVID 19. This growth shows the need for the effective recommendation system to support course selection.

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. As the number of courses and learners increases the intelligent recommendation system becomes important to manage the information overload.

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.

This project focuses on learning course recommendation approach that uses course metadata than learners history. This makes the system

suitable for the platforms like Coursera. It has limited interaction data of learners. Although the system does not fully analyse learners learning behoviour but it provides practical and scalable recommendations using widely available informations.

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 online learning platform such as Coursera.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.
- To reduce information overload by providing more relevant and meaningful course recommendations.

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 research work focuses on structured decision making for academic planning while more recent studies introduced machine learning, deep learning, and real-time personalisation.

This project follows horizontal research perspective. Here it focuses on reviewing and comparing different artificial intelligence and the machine learning approaches used in course recommendation systems. It shows how course recommendation systems have changed over the time and show common limitations. This approach is suitable for understanding the real world online learning platform like Coursera where the learners behaviour is different.

2.1.1 Research 1:

Detailed Study of Coursera Course 2024 Dataset and its Optimization.
A study was conducted on the Coursera Course 2024 dataset to analyse how course features support effective recommendation systems. It contains 6,645 course records with attributes such as course title, ratings, number of reviews, enrollment number, instructors and the organizations.These features shows both course popularity and quality of course which are important factors in course selection.

Exploratory data analysis was used to identify important features that effect interest of the learners. Enrollment number and average rating were strong features of course relevance while instructor and organization helps in grouping similar courses. Supervised learning models such as Logistic Regression, Decision Trees and Random Forest were tested using the dataset.Random Forest performs better in complex relationship between the course features. Logistic Regression is an baseline model to work on the recommendation system. It shows selecting few attributes can improve recommendation quality while keeping system simple and scalable.

2.1.2 Research 2: 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 important 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.

2.1.3 Research 3: 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.

2.1.4 Research 4: 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.

2.1.5 Research 5: 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

Detailed Study of Coursera Course 2024 Dataset and its Optimization.

The research offers valuable insights into optimizing coursera course datasets for quality course recommendation according to the popularity and the quality of courses. It shows that the Random Forest as a reliable classifier and provides the detailed framework for data preprocessing and feature selection in machine learning.

2.2.2 Review and Analysis of Research 2

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.

2.2.3 Review and Analysis of Research 3

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.

2.2.4 Review and Analysis of Research 4

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.

2.2.5 Review and Analysis of Research 5

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.

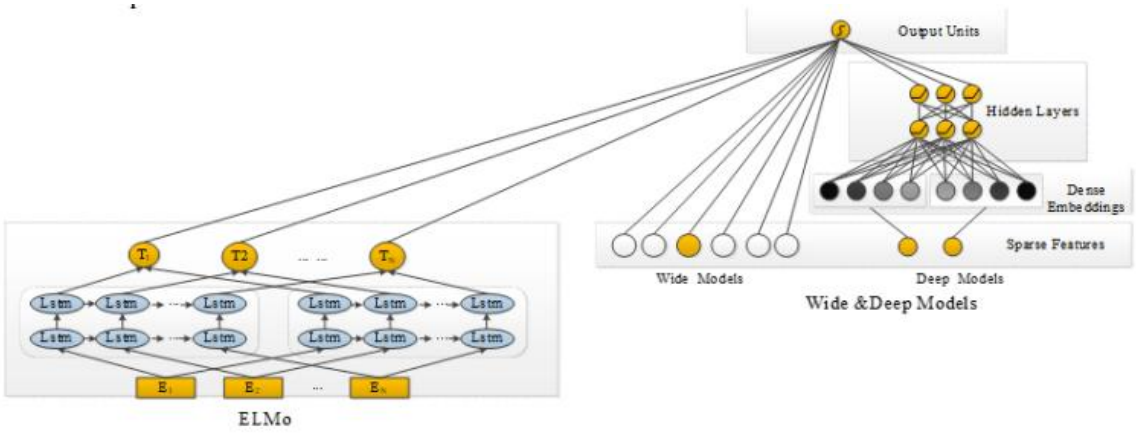


Figure 2: Online Learning Resource Recommendation Method Based on Wide & Deep and Elmo Model (Liu, et al., 2020)

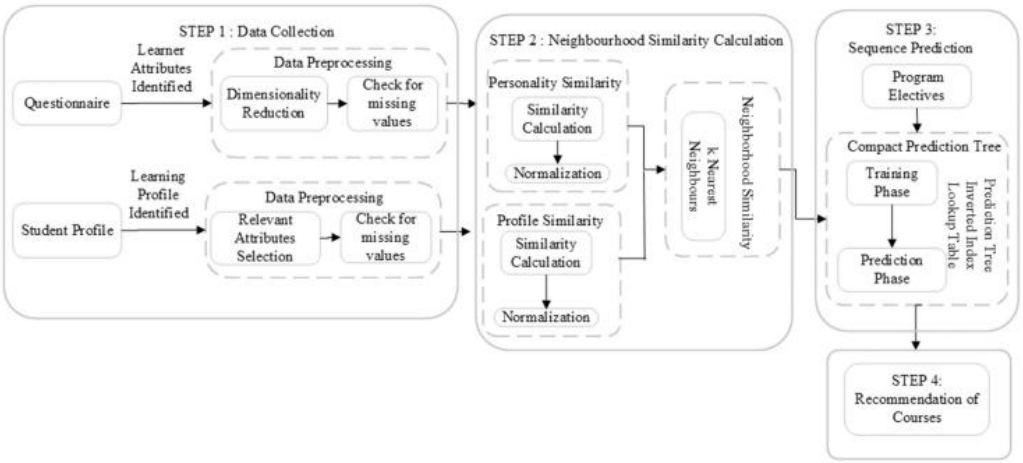


Figure 3: Course recommendation framework (George & La, 2024)

Therefore, it is clear that advanced deep learning models improve personalization but requires large datasets, high cost and reduce transparency from research articles. Simpler rule based system is interpretable but it lacks scalability and adaptability. This project adopts a balanced approach by combining both supervised and unsupervised machine learning methods which uses course metadata. It aims to get accuracy while remaining scalable and easy to interpret.

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 Overview of the Solution

This project develops an AI based course recommendation system to help learners choose suitable courses from large online learning platforms like Coursera. Many existing system is based on the popularity by applying machine learning algorithm that analyse course informations to generate more relevant course recommendations. This system uses machine learning to analyse course information to solve this problem.

The proposed solution uses the machine learning techniques to analyse course level information rather than relying on learner history. 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.

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 course recommendation system follows a modular architecture consisting of four main components: data preprocessing, feature engineering, recommendation engine, and output generation. This modular design allows each components to be improved which makes the system easier to maintain and scalable for the large online learning platforms.

At first the raw dataset undergoes preprocessing which includes handling missing values, normalising numerical features such as number of enrollment, ratings, and encoding categorical variables such as instructor names and organisations. After preprocessing, 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 input.

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.

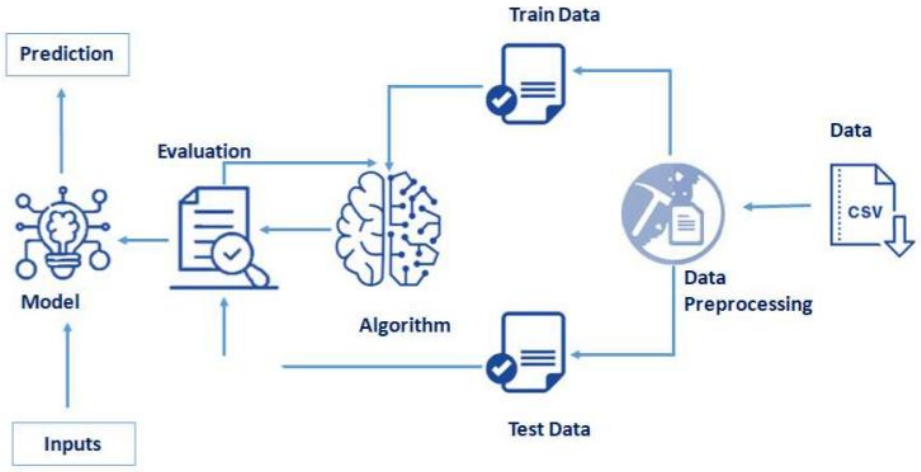


Figure 4: System Architecture

The overall workflow of the system consists of data collection and preprocessing in a machine learning model. The dataset is first cleaned to handle missing values and remove unnecessary attributesThe data is then split into training and testing sets. The training set is used to build the model using an algorithm, while the testing set is used to evaluate its performance. Once the model is trained and evaluated, it can take new user inputs to make predictions. Numerical features such as course ratings and number of enrollments are selected for modelling. Exploratory Data Analysis is another step that is used to understand the distribution of ratings, pattern of enrollment and their relationships. It helps to identify important attributes that help in course popularity and quality of the courses.

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:-

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

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.

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

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).

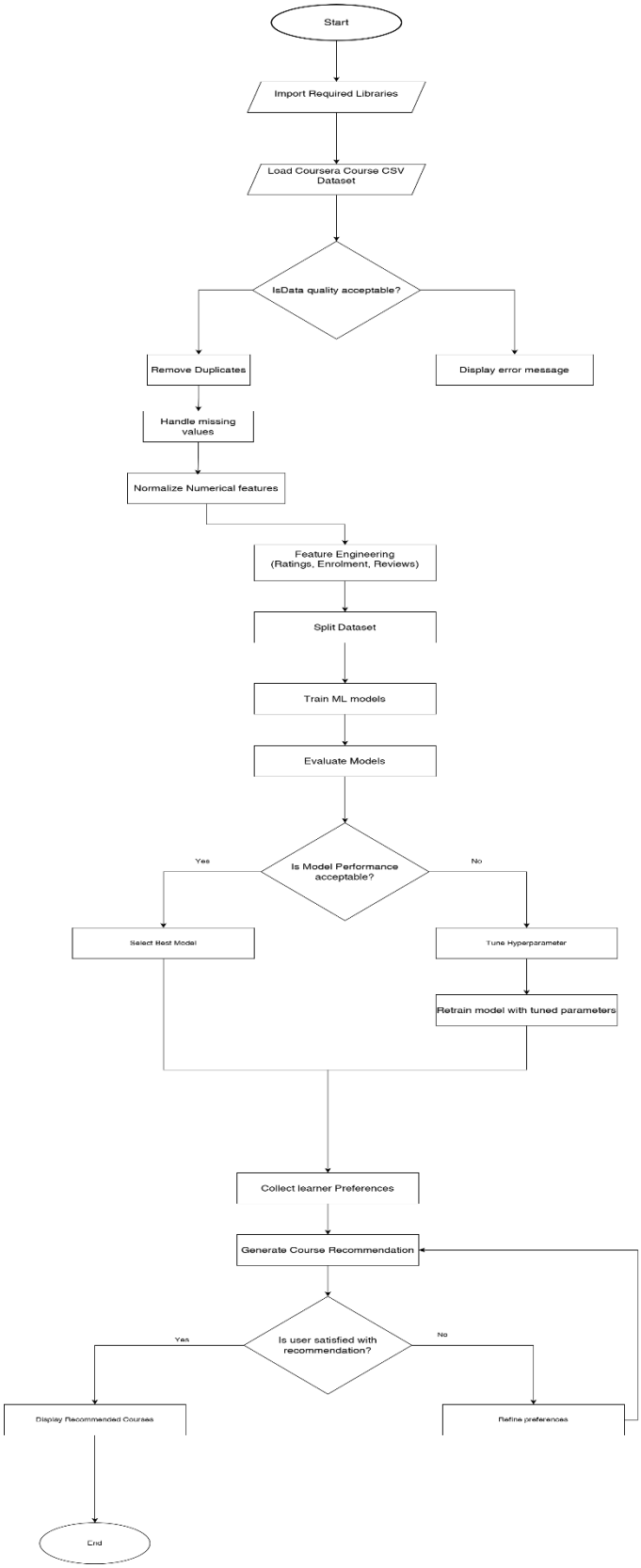


Figure 5: Flowchart of the system

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 6: 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 7: 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 8: 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

```
import pandas as pd
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.preprocessing import StandardScaler
from sklearn.metrics import roc_curve, auc
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, GradientBoostingClassifier, StackingClassifier
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report, ConfusionMatrixDisplay
```

Figure 9: 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.

3.6.2 Data Cleaning and Preprocessing

Data cleaning is performed to improve data quality and ensure reliable model training. It includes handling missing values, removing unnecessary columns and selecting meaningful numerical features. Feature scaling is applied to normalize the data which is important for clustering and classification algorithms.

3.6.3 Splitting Data for Training and Testing

The dataset is divided into training and testing subsets using the train test split method. The training set is used to train the machine learning algorithm while the testing set is used to evaluate model performance. This approach helps to prevent overfitting and ensures better performance.

3.6.4 Using Algorithm Model

Both supervised and the unsupervised learning algorithm are used in this project. K-Means Clustering is used to group similar courses. Logistic Regression, Decision Trees and the Random Forest are trained to predict course suitability.

3.6.5 Evaluating Algorithm Models

Model performance is evaluated using accuracy scores, confusion matrix, f1 score, recall. These evaluation metrics helps to compare the different algorithms and identifying the most effective model for the recommendation. Random Forest shows the highest accuracy due to the ensemble nature.

3.6.6 Feature Importance Analysis

Feature Importance analysis is performed using the Random Forest model which identifies features that help most in prediction. This analysis improves model interpretability and provides insight into features that helps in course recommendation decisions.

3.6.7 Visualization

Visualization are used to visualize data understanding and result interpretation. Graph and plots shows feature distributions, clustering result and model performance which helps to understand the system better.

3.7 Achieved Results

3.7.1 Load Dataset

The above figure 10 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

After the dataset has been stored in a DataFrame called data, then the

Figure 10: Loading Coursera_coursera_2024 CSV file

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 11:Diplaying first five rows

```
df.shape
```

```
(6646, 14)
```

```
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
```

```
df.isna().sum()
```

```
title                   0
enrolled                0
rating                  0
num_reviews          1392
Instructor              1
Organization            1
Skills                  0
Description            10
Modules/Courses        12
Level                 780
Schedule             1889
URL                     2
Satisfaction Rate    4449
enrolled_clean          0
rating_clean            0
reviews_clean           0
```

Figure 12: 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. After loading the data duplicate rows are removed to make sure the dataset does not contain repeated records. The df.isna(),sum() is used to count a number of missing null values of dataframe. This stage is important because duplicate data can affect model performance and give wrong results.

3.7.3 Data Cleaning and Preprocessing

4. Data Cleaning & Preprocessing

4.1 Remove unwanted column & duplicates

```
df = df.drop(columns=['Unnamed: 0'], errors='ignore')
df = df.drop_duplicates()
```

4.2 Function to clean numeric columns

```
def clean_numeric(col):
    return pd.to_numeric(
        col.astype(str)
           .str.replace(',', '')
           .str.replace('%', ''),
        errors='coerce'
    )
```

4.3 Clean required numeric columns

```
df['enrolled_clean'] = clean_numeric(df['enrolled'])
df['rating_clean'] = clean_numeric(df['rating'])
df['reviews_clean'] = clean_numeric(df['num_reviews'])
df['satisfaction_clean'] = clean_numeric(df['Satisfaction Rate'])
```

4.4 Check missing values

```
df[['enrolled_clean', 'rating_clean', 'reviews_clean','satisfaction_clean']].isna().sum()
```

```
enrolled_clean        1759
rating_clean          1437
reviews_clean         1393
satisfaction_clean    4449
dtype: int64
```

4.5 Handle missing values

```
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)
df['satisfaction_clean'].fillna(df['satisfaction_clean'].mean(), inplace=True)
```

```
df[['enrolled_clean', 'rating_clean', 'reviews_clean','satisfaction_clean']].isna().sum()
```

```
enrolled_clean        0
rating_clean          0
reviews_clean         0
satisfaction_clean    0
dtype: int64
```

Figure 13: 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 Exploratory Data Analysis

5. EXPLORATORY DATA ANALYSIS (EDA)

```
plt.figure(figsize=(6,4))
sns.histplot(df['rating_clean'], bins=20, kde=True)
plt.title("Distribution of Course Ratings")
plt.show()
```

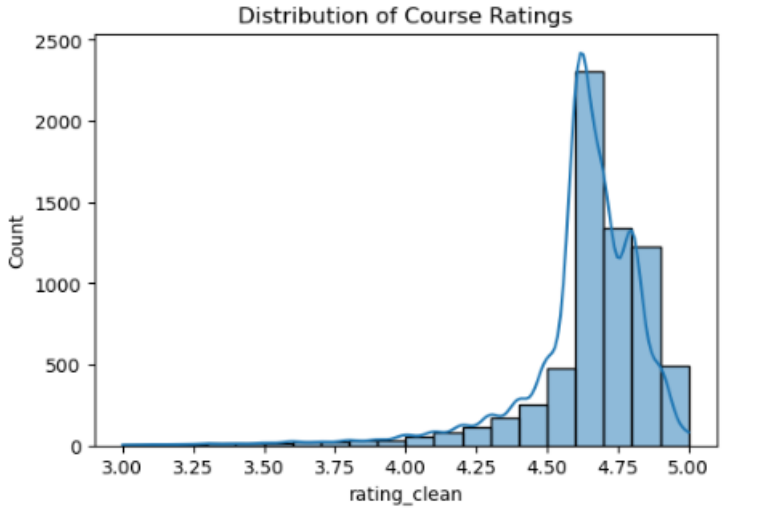


Figure 14: Distribution of Course Ratings

This figure shows the distribution of course ratings after cleaning the rating column. From the histogram figure it is clear that the most courses have ratings between 4.00 and 5.00 ratings. The bars are taller around this range which shows that the learners mostly give positive feedbacks in Coursera courses. The dataset has very few courses that falls below rating of 4.0 and very low ratings are rare. The smooth KDE line helps to understand the overall trend and the shape of the data. It explains why the datasets imbalanced when creating the target variable for high and low ratings of the courses. The model needs to be handled carefully during training because many courses are highly rated. Finally,this visualization helps to understand the user behaviour before building the recommendation model.

```
corr_data = df[['enrolled_clean', 'reviews_clean', 'rating_clean']]

plt.figure(figsize=(8,6))
sns.heatmap(
    corr_data.corr(),
    annot=True,
    cmap='coolwarm',
    fmt=".2f"
)
plt.title("Correlation Heatmap")
plt.show()
```

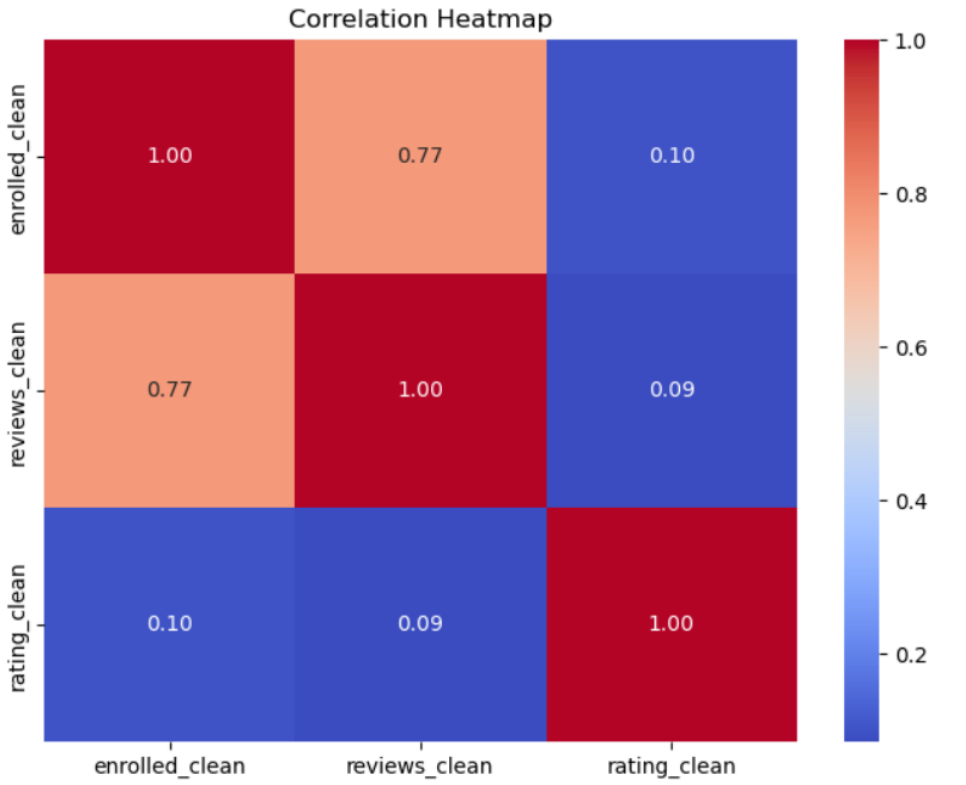


Figure 15: Correlation Heatmap of enrollment, ratings and review

This heatmap shows the linear correlation between numerical features like number of enrollment, number of reviews and rating of courses. Correlation value range from -1 to 1. +1 ,means strong positive relationship, 0 means no relationship and -1 means strong negative relationship. Here Enrollment and Reviews (0.77) shows strong positive correlation. The courses with more number of enrollments receives more reviews. Ratings and enrollments(0.10) has very weak correlation which means high enrollment does not guarantee a high ratings. Ratings and reviews(0.09) has very weak relationship. Therefore enrollment and review are strongly related to each other but course rating is independent which means quality is not determined by popularity.

```
plt.figure(figsize=(6,4))
sns.countplot(x='rating_label', data=df)
plt.title("Distribution of Course Rating Classes")
plt.xlabel("Rating Category (0 = Low, 1 = High)")
plt.ylabel("Number of Courses")
plt.show()
```

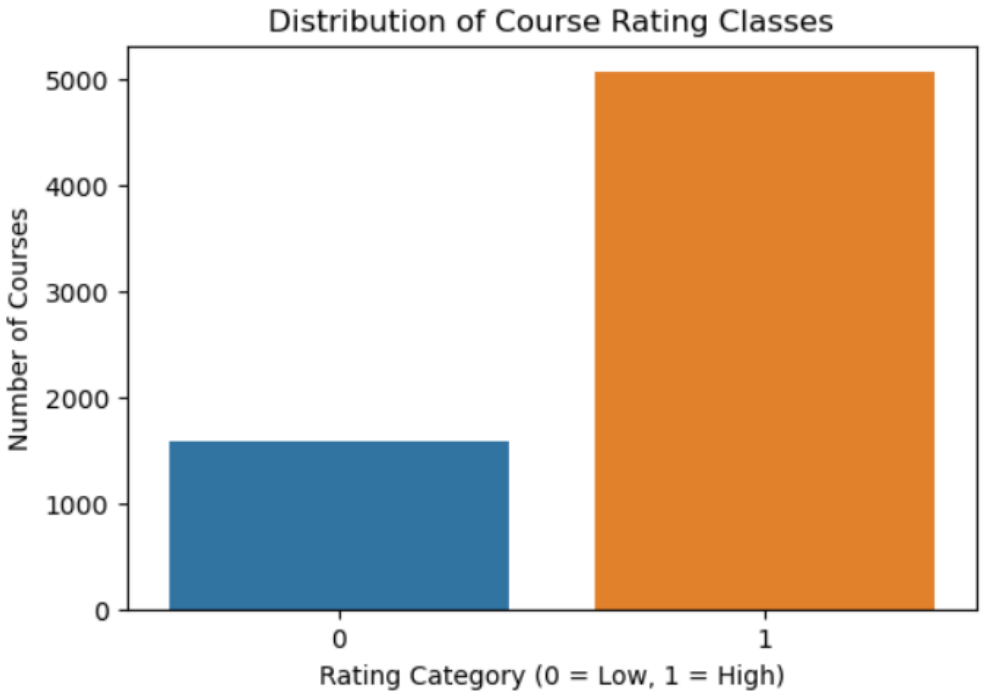


Figure 16: Distribution of Course Rating Classes

This bar chart shows the distribution of binary rating classes 0 and 1. 0 means low rating and 1 means high rating. Here high rated course is more than other course. Other course fall into the low rating category. The dataset is imbalanced. Class imbalance can bias machine learning model which requires class weight to be balanced in techniques of supervised model.

```
top_courses = df.sort_values('enrolled_clean', ascending=False).head(10)

plt.figure(figsize=(10,6))
sns.barplot(
    data=top_courses,
    x='enrolled_clean',
    y='title'
)
plt.title("Top 10 Courses by Enrollment")
plt.xlabel("Number of Enrollments")
plt.ylabel("Course Title")
plt.show()
```

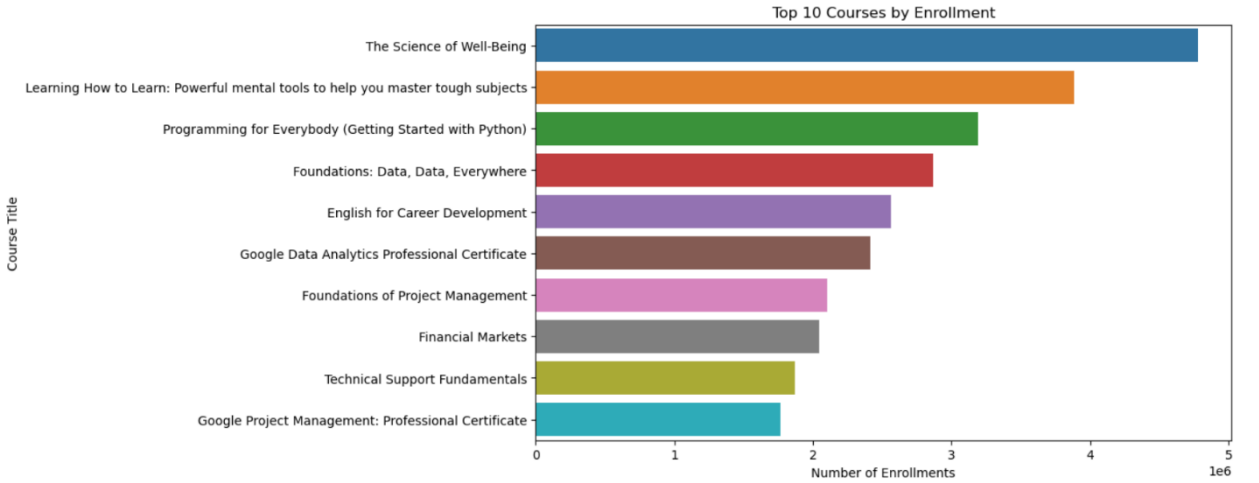


Figure 17:Top 10 Courses by Enrollment

This bar chart shows the top 10 most enrolled Coursera courses by enrollment of learners. A small number of courses can attract many millions of learners. Some of the most popular courses is related to science of well being, learning, programming, Data Analytics, English, Project Management and other Professional certificates. People prefer to learn related to their health and well being. Coursera follows a long tail distribution where some of the courses control the total enrollment of the learners.

```
plt.figure(figsize=(6,4))
sns.scatterplot(
    x='enrolled_clean',
    y='reviews_clean',
    data=df,
    alpha=0.4
)
plt.title("Relationship Between Enrolments and Reviews")
plt.xlabel("Enrolments")
plt.ylabel("Number of Reviews")
plt.show()
```

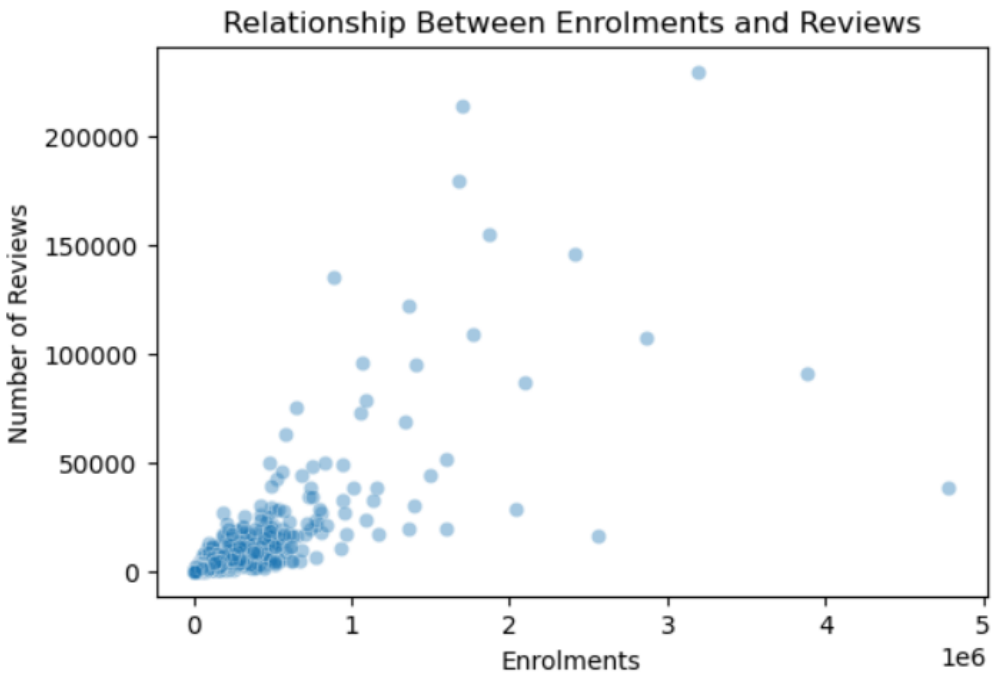


Figure 18: Scatterplot of Enrolments and Reviews

This is the scatter plot diagram which shows the relationship between enrollment and reviews. Each dot shows course where X-axis shows number of enrollments, Y-axis shows the number of reviews. It shows clear positive trend. As number of enrollments increases , review also increases. Some courses have high enrollments but few review.

```
plt.figure(figsize=(6,4))
sns.regplot(
    x='enrolled_clean',
    y='rating',
    data=df,
    scatter_kws={'alpha':0.3},
    line_kws={'color':'red'}
)
plt.title("Relationship between Enrolment Count and Course Rating")
plt.xlabel("Number of Enrollments")
plt.ylabel("Course Rating")
plt.show()
```

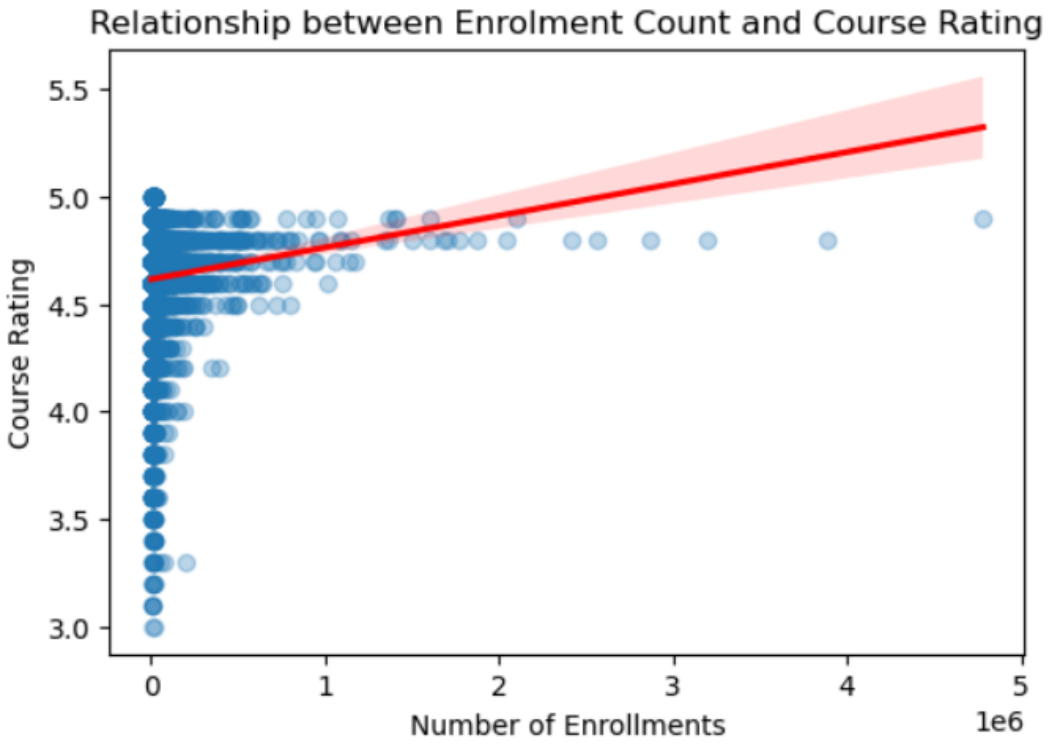


Figure 19: Relationship between Enrollment Count and Course Rating

This regression plot shows regression plot which is the combination of scatter plots and regression line. Here this diagram shows a slight upward slope which shows a weak positive relationship. Here most ratings of courses are between 4.5 and 5. Here high enrollment does not strongly affect the rating of the courses. Course ratings are independent which means despite the enrollment number it is not affected.

3.7.5 Unsupervised Learning : K Means Clustering

6.1 Feature Selection for K-Means

```
X_kmeans = df[['rating_clean', 'enrolled_clean', 'reviews_clean']]
```

6.2 Feature Scaling

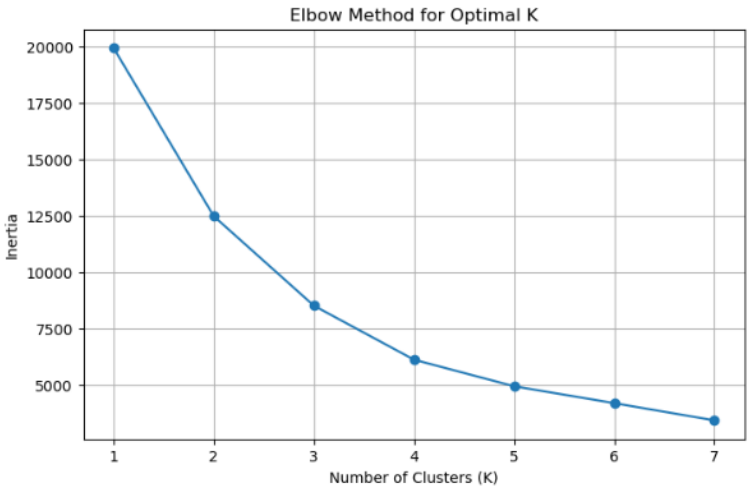
```
scaler= StandardScaler()
X_kmeans_scaled = scaler.fit_transform(X_kmeans)
```

6.3 Apply K-Means Clustering

```
inertia = []
K = range(1, 8)

for k in K:
    km = KMeans(n_clusters=k, random_state=42, n_init=10)
    km.fit(X_kmeans_scaled)
    inertia.append(km.inertia_)

plt.figure(figsize=(8,5))
plt.plot(K, inertia, marker='o')
plt.xlabel("Number of Clusters (K)")
plt.ylabel("Inertia")
plt.title("Elbow Method for Optimal K")
plt.grid(True)
plt.show()
```




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

Figure 20: 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.

```
plt.figure(figsize=(8,6))
plt.scatter(
    df['rating_clean'],
    df['enrolled_clean'],
    c=df['Cluster'],
    cmap='viridis',
    s=40
)
plt.xlabel("Rating")
plt.ylabel("Enrollment of Students")
plt.title("K-Means Clustering of Coursera Courses(2024)")
plt.show()
```

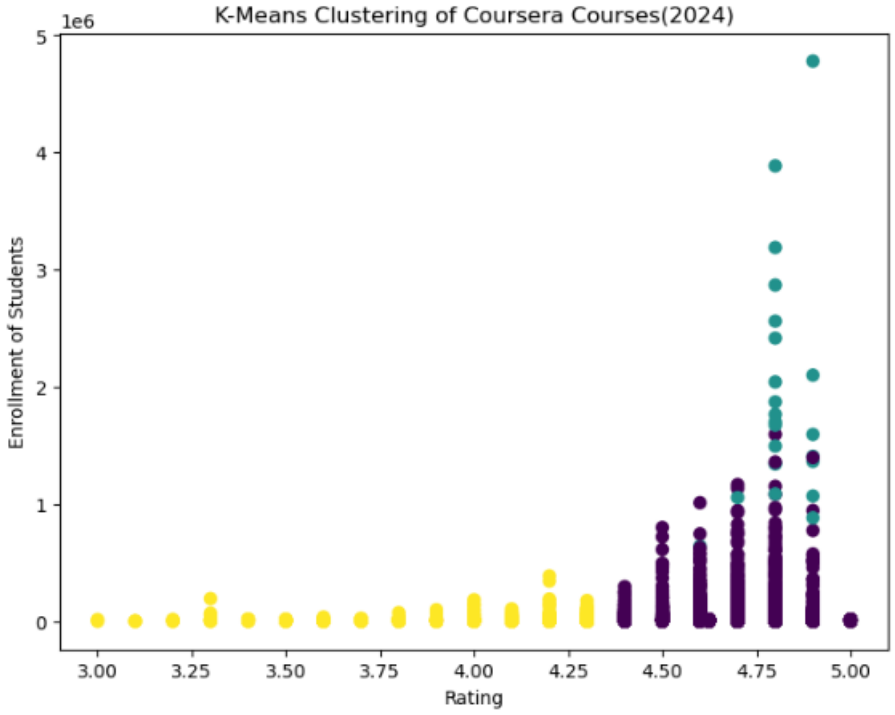


Figure 21:Scatter Plot of Rating and Enrolled Students

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>
```

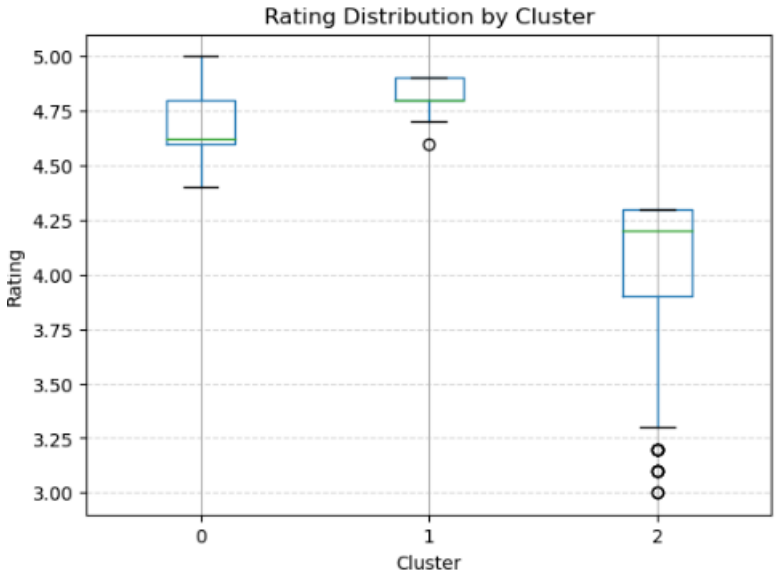


Figure 22: Boxplot of Rating by Cluster

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>
```

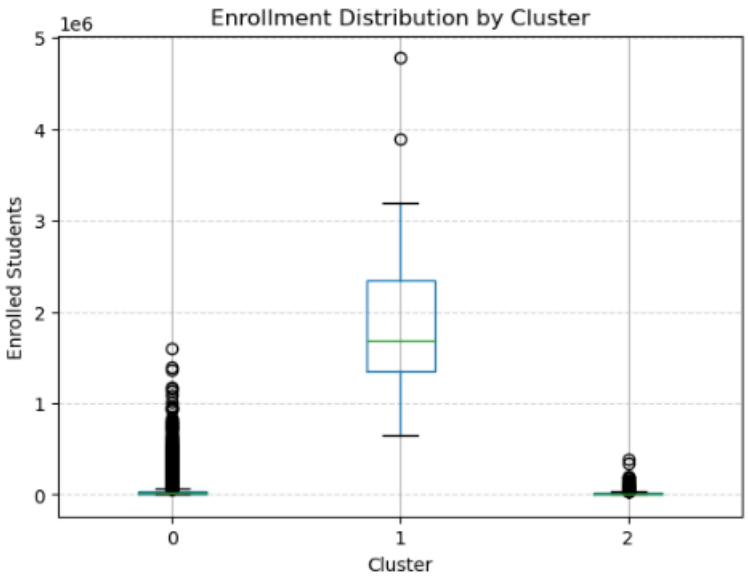


Figure 23: Boxplot of Enrollment by Cluster

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.

3.7.6 Supervised Learning

7. Supervised Learning Algorithm

7.1 Create Target Variable

```
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
```

7.2 Feature Selection

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

7.3 Train–Test Split

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

7.4 Feature Scaling

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

Figure 24: Creation of target variable for supervised learning

In this dataset, data is split into training and testing sets where they are split into 80- 20. Test_size means 20 % of 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.

- Logistic Regression

7.5 Logistic Regression

```
[42]: 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: 0.7731648616125151
              precision    recall  f1-score   support

           0       0.28      0.54      0.37       204
           1       0.93      0.81      0.86      1458

    accuracy                           0.77      1662
   macro avg       0.60      0.67      0.62      1662
weighted avg       0.85      0.77      0.80      1662
```

Figure 25:Logistic Regression Classification Accuracy and Performance Metrics

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.

```python
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

cm_log = confusion_matrix(y_test, y_pred_log)

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

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

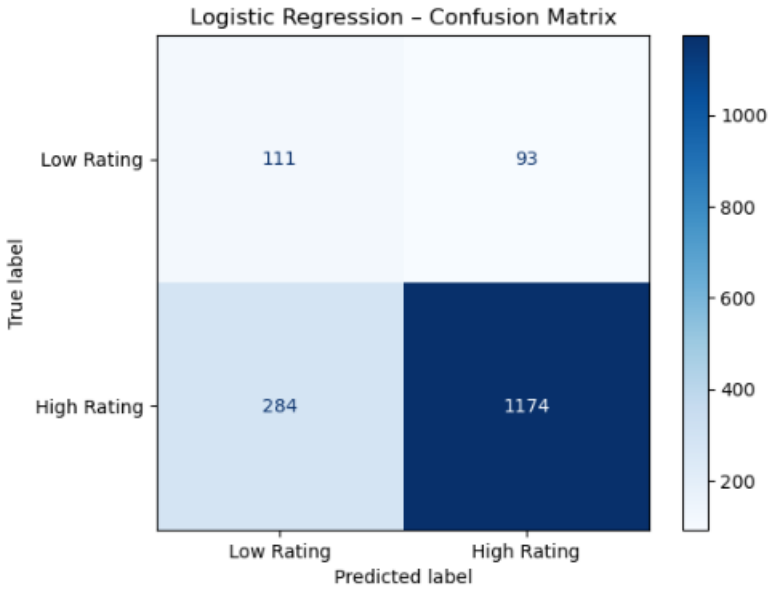


Figure 26: Confusion Matrix for Logistic Regression Model

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.

```
plt.figure(figsize=(6,4))
sns.scatterplot(
    x=X_test['enrolled_clean'],
    y=y_test,
    label='Actual',
    alpha=0.5
)

sns.scatterplot(
    x=X_test['enrolled_clean'],
    y=y_pred_log,
    label='Predicted',
    alpha=0.5
)

plt.title("Logistic Regression: Actual vs Predicted Ratings")
plt.xlabel("Enrollments")
plt.ylabel("Rating Class (0=Low, 1=High)")
plt.legend()
plt.show()
```

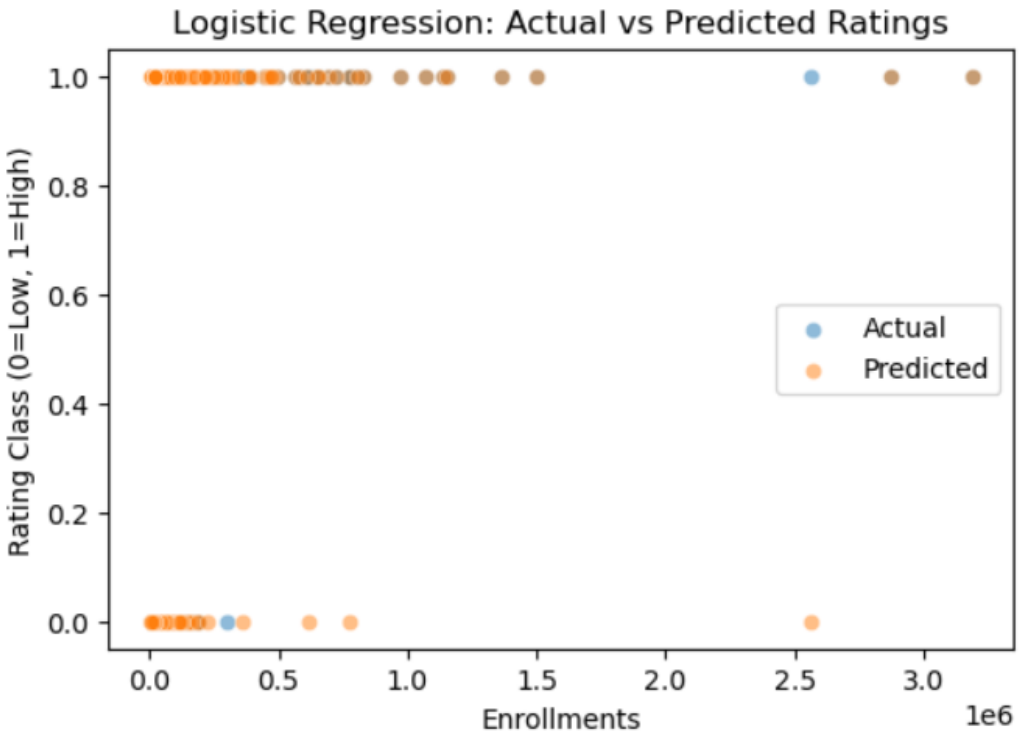


Figure 27:Logistic Regression: Actual vs Predicted Ratings

This diagram shows the logistic regression of actual and predicted scatter plot. Blue dot shows the actual class whereas the orange dots shows the predicted class by the model. Many predictions overlaps with the actual values. Some misclassifications are visible at lower enrollment. Therefore, logistic regression model performs good but is not perfect for some cases.

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

y_prob_log = log_model.predict_proba(X_test_scaled)[:,1]

fpr, tpr, _ = roc_curve(y_test, y_prob_log)
roc_auc = auc(fpr, tpr)

plt.figure(figsize=(6,5))
plt.plot(fpr, tpr, label=f"AUC = {roc_auc:.2f}")
plt.plot([0,1], [0,1], linestyle='--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve – Logistic Regression")
plt.legend()
plt.show()
```

```
/opt/conda/envs/anaconda-panel-2023.05-py310/lib/python3.11/site-packages/
  warnings.warn(
```

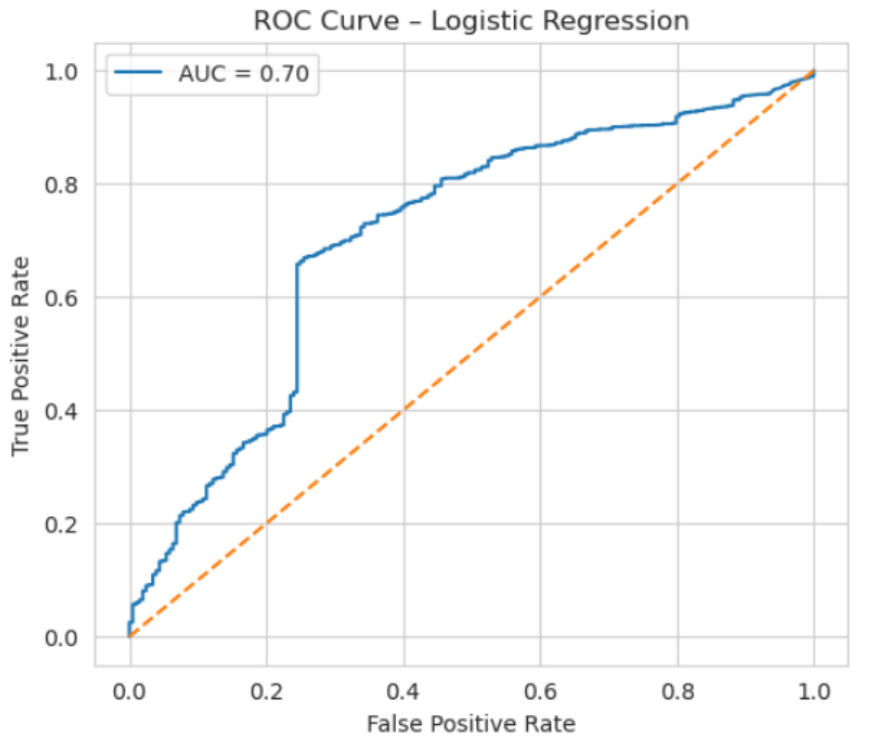


Figure 28:ROC Curve of Logistic Regression

This diagram shows the Receiver Operating Characteristic curve of logistic regression. It measures how well the model separates low rated courses and high rated courses. Area Under the Curve (AUC) is a common metric derived from the ROC curve that summarizes the model's overall performance. And the score of AUC is 0.70. The curve is clearly above the diagonal line. The model performs better than other models but still needs improvement.

- Decision Tree

7.6 Decision Tree Classifier

```
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))
print(classification_report(y_test, y_pred_dt))
```

```
Decision Tree Accuracy: 0.8730445246690735
              precision    recall  f1-score   support

           0       0.11      0.00      0.01       204
           1       0.88      0.99      0.93      1458

    accuracy                           0.87      1662
   macro avg       0.49      0.50      0.47      1662
weighted avg       0.78      0.87      0.82      1662
```

Figure 29: Decision Tree Classification Accuracy and Performance Metrics

Here, in this code it shows the accuracy score of logistic regression of the Decision Tree model. The accuracy score is 0.873 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 plot_tree

plt.figure(figsize=(16,6))
plot_tree(
    dt_model,
    feature_names=['Enrolled', 'Reviews'],
    class_names=['Low Rating', 'High Rating'],
    filled=True
)
plt.title("Decision Tree Visualization")
plt.show()
```

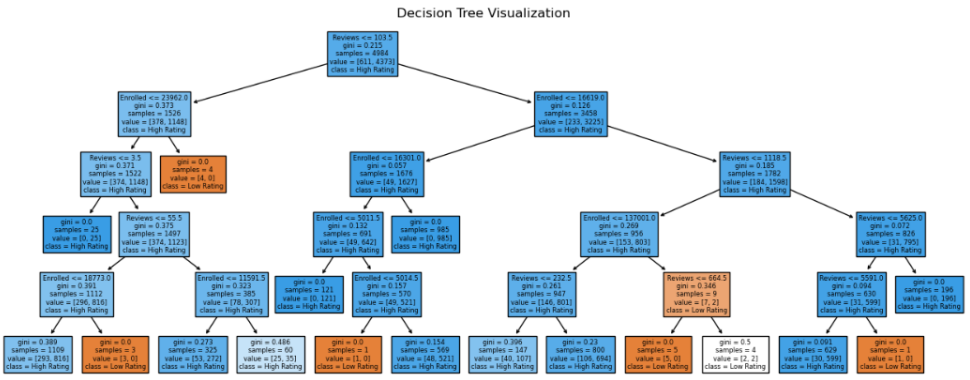


Figure 30:Visualization of the Decision Tree Model

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

7.7 Random Forest Classifier

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

   rf_model.fit(X_train, y_train)

   y_pred_rf = rf_model.predict(X_test)

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

```
Random Forest Accuracy: 0.8772563176895307
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       204
           1       0.88      1.00      0.93      1458

    accuracy                           0.88      1662
   macro avg       0.44      0.50      0.47      1662
weighted avg       0.77      0.88      0.82      1662
```

Figure 31: Random Forest Classification Accuracy and Performance Metrics

This code shows the accuracy of the Random 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.

```
rf_probs = rf_model.predict_proba(X_test)[:,1]

plt.figure(figsize=(6,4))
sns.histplot(rf_probs, bins=20, kde=True)
plt.title("Random Forest Prediction Probability Distribution")
plt.xlabel("Probability of High Rating")
plt.ylabel("Count")
plt.show()
```

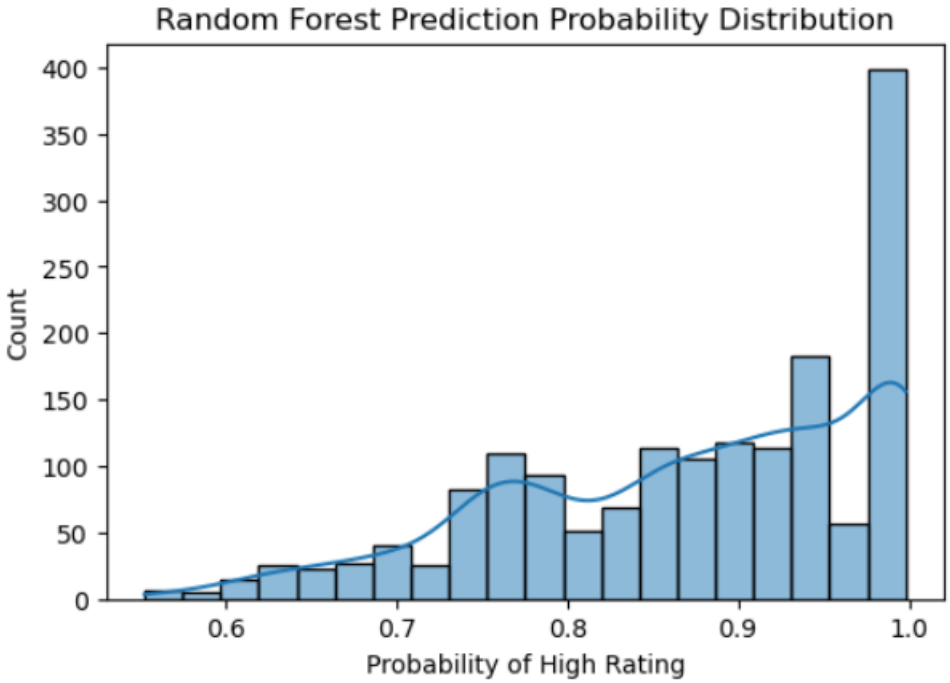


Figure 32: Random Forest Prediction Probability Distribution

This diagram shows the histogram of random forest model. It shows how correct the random forest is when predicting high rating. Many predictions are equals to 1 which means high confidence. Some few predictions fall under 0.5, The random forest model make strong and confident predictions not random ones.

Feature Importance – Random Forest

```
# Feature importance
feature_importance = pd.DataFrame({
    'Feature': X.columns,
    'Importance': rf_model.feature_importances_
}).sort_values(by='Importance', ascending=False)

plt.figure(figsize=(6,4))
sns.barplot(
    x='Importance',
    y='Feature',
    data=feature_importance
)
plt.title("Feature Importance from Random Forest")
plt.show()
```

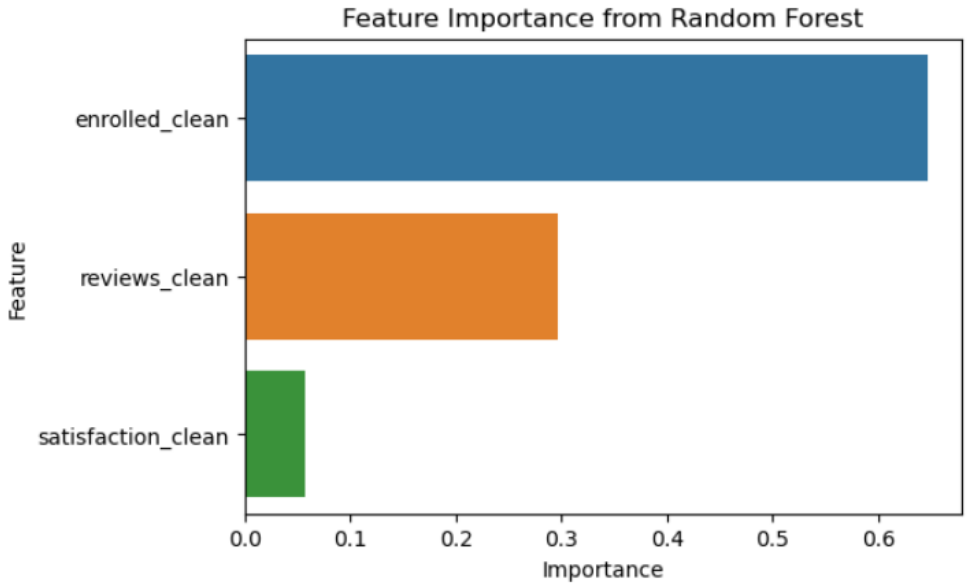


Figure 33: Bar graph of Random Forest Feature Importance

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. Enrolled_clean is the most important feature, reviews_clean is moderately important and satisfaction_clean is least important for recommendation of courses. 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.

7.8 Model Comparison

```
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	0.773165
1	Decision Tree	0.873045
2	Random Forest	0.877256

Figure 34: Accuracy Comparison of Supervised Learning Models

This tables compares the accuracy of three supervised learning models where the score of logistic regression is 0.77, decision tree is 0.873, Random forest is 0.877. Amon all these models Random forest performs the best which makes it the most suitable for supervised learning model for this project.

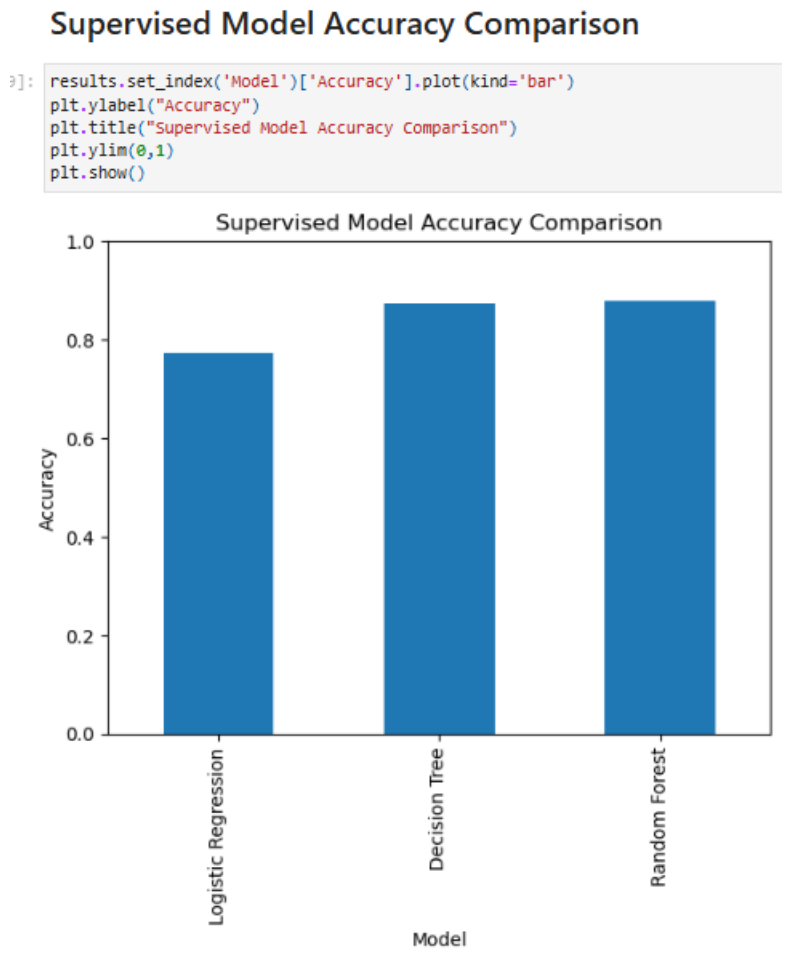


Figure 35: Comparison of Supervised Models in Bar graph

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.

3.7.7 Hybrid Model

8. Hybrid Model

```
X_hybrid = df[['enrolled_clean', 'reviews_clean', 'cluster']]
y_hybrid = df['high_rating']
```

```
Xh_train, Xh_test, yh_train, yh_test = train_test_split(
    X_hybrid, y_hybrid,
    test_size=0.25,
    random_state=42,
    stratify=y_hybrid
)
```

```
base_models = [
    ('rf', RandomForestClassifier(
        n_estimators=100, max_depth=5, random_state=42)),
    ('gb', GradientBoostingClassifier(
        n_estimators=100, learning_rate=0.1, random_state=42))
]

hybrid_model = StackingClassifier(
    estimators=base_models,
    final_estimator=LogisticRegression(max_iter=1000),
    passthrough=False
)

hybrid_model.fit(Xh_train, yh_train)
y_pred_hybrid = hybrid_model.predict(Xh_test)

acc_hybrid = accuracy_score(yh_test, y_pred_hybrid)
```

Figure 36:Hybrid Model Accuracy

The hybrid model uses enrolled_clean, review_clean and cluster as input features. It combines numerical data with information from unsupervised learning. The target variable is high_rating which is binary label that shows whether a course has high ratings. The dataset is split into training and testing set 25 % for testing and others remaining to maintain a balanced distribution of rating classes. Two ensemble modles Random Forest and Gradient Boosting are used to capture complex pattern which reduces overfitting. These base models are combined using Stacking Classifier and logistic regression is final estimator and learns how to combine their predictions. The trained hybrid model is evaluated using accuracy to measure its performance in classifying courses as high and low.Overall this hybrid model combines both unsupervised and multiple supervised models which results in more stable and reliable prediction as compare to other single model.

```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt

# Confusion matrix
cm_hybrid = confusion_matrix(y_test, y_pred_hybrid)

# Plot
plt.figure(figsize=(6,5))
disp = ConfusionMatrixDisplay(
    confusion_matrix=cm_hybrid,
    display_labels=["Low Rating", "High Rating"]
)
disp.plot(cmap="Purples", values_format="d")

plt.title("Hybrid Model – Confusion Matrix")
plt.grid(False)
plt.show()
```

```
<Figure size 600x500 with 0 Axes>
```

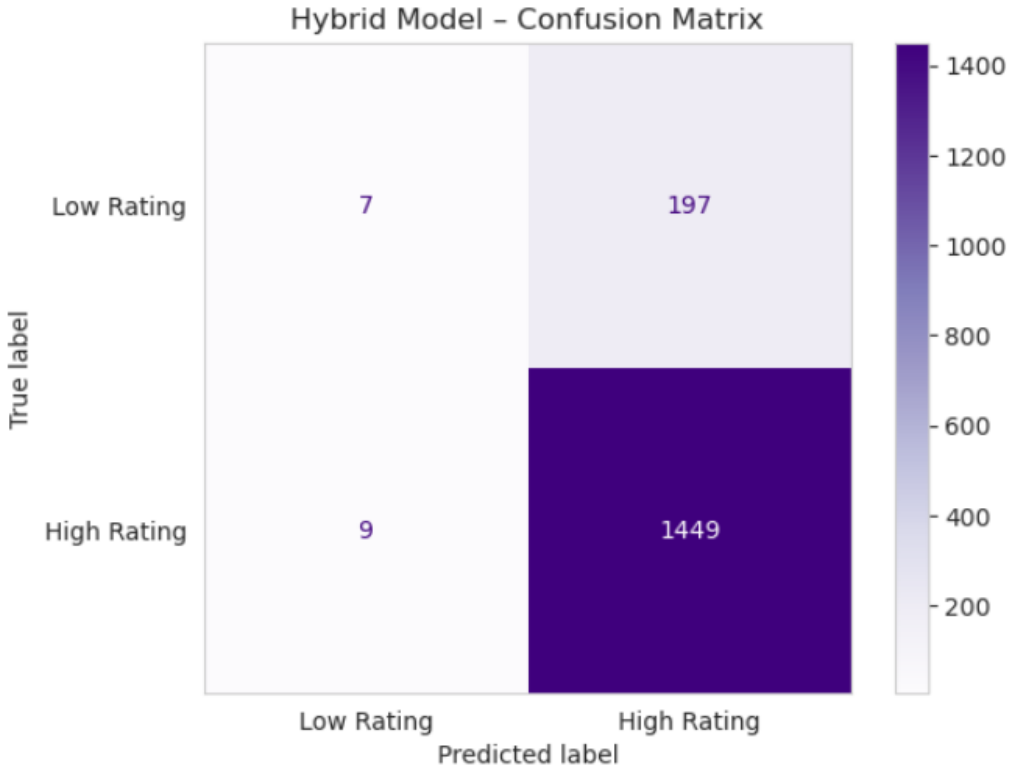


Figure 37: Confusion Matrix of Hybrid Model

This confusion matrix diagram shows how well hybrid model classifies courses into Low rating and High rating. The row shows the true class while column shows predicted class. Most values are in the dark color which means model correctly predicts the huge numbers of high rating courses. Only few low rating courses are misclassified as high ratings and few high rating courses are predicted as low rating. It shows the hybrid model perform strong especially in case of high rating courses which are majority of classes in the dataset.

```
]: hybrid_accuracy = accuracy_score(yh_test, y_pred_hybrid)
print("Hybrid Model Accuracy:", round(hybrid_accuracy, 4))

# Classification Report
print("\nHybrid Model Classification Report:\n")
print(classification_report(
    yh_test,
    y_pred_hybrid,
    target_names=["Low Rating", "High Rating"]
))
```

```
Hybrid Model Accuracy: 0.8761

Hybrid Model Classification Report:

              precision    recall  f1-score   support

  Low Rating       0.44      0.03      0.06       204
 High Rating       0.88      0.99      0.93      1458

    accuracy                           0.88      1662
   macro avg       0.66      0.51      0.50      1662
weighted avg       0.83      0.88      0.83      1662
```

Figure 38:Hybrid Model Accuracy and Performance Metrics

The hybrid model achieves an accuracy of 0.8761. It shows the strong performance of the model. It predicts high rating courses very well with high precision in 0.88, and recall 0.99 which means the most high rated courses are correctly identified. But the model performs poorly on low rated courses with low recall 0.03 because the dataset is imbalanced and contains more high rating courses. Therefore, the hybrid model is effective but biased toward predicting the high rating courses.

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

results
```

	Model	Accuracy
0	Logistic Regression	0.773165
1	Decision Tree	0.873045
2	Random Forest	0.877256
3	Hybrid Model	0.876053

Figure 39: Model Accuracy table

This table compares the accuracy of four supervised learning models which is logistic regression, decision tree, random forest and hybrid model. Logistic regression has the lowest accuracy which is expected as it is a simple linear model. Decision Tree and Random Forest perform better due to their ability to handle non linear data patterns. The hybrid model gives accuracy just like random forest but combining with other models that gives better performance while improving robustness and being scalable.

```
plt.figure(figsize=(8,4))
sns.set_style("whitegrid")

sns.barplot(
    x='Accuracy',
    y='Model',
    data=results
)

plt.xlim(0, 1)
plt.title("Model Accuracy Comparison")
plt.xlabel("Accuracy Score")
plt.ylabel("")

for i, acc in enumerate(results['Accuracy']):
    plt.text(acc + 0.01, i, f"{acc:.3f}", va='center')

plt.show()
```

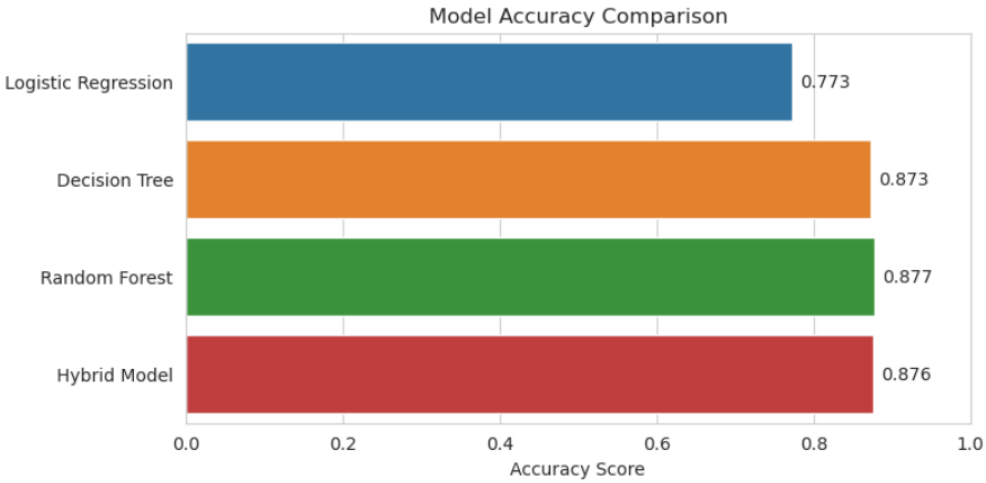


Figure 40: Model Accuracy Comparison in Bar Chart

The horizontal bar chart shows the comparison of accuracy score of various models. Each bar shows each model and the accuracy is labeled as well. Random Forest has the highest accuracy then Hybrid Model comes after the random forest. This visualization makes it easy to see ensemble model like Random Forest and Hybrid Model performs better than logistic regression model .

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 How the Application Addresses Real World Problems

This application addresses a real world problem faced by learners and education platforms which identifies high quality courses from a various available options. It analyses various factors such as enrollments, reviews and ratings. The models help to predict whether a course is likely to be a high rated or not. This helps students in making informed learning choices and helps in online learning platforms that improve course recommendations, monitors quality of courses and helps in decision making. The approach helps in feedback and popularity dta of courses to check the quality such as feedback and ratings.

4.3 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.

It was both challenging and amazing learning experience for me as I worked on this Artificial Intelligence project. During the start of the project I had basic understanding of AI how it is used in practical life. While developing this application , I faced issue in cleaning data , handling missing values , model selection, handling errors and evaluation of model. Most of the time The result came but it was confusing and some models produced same accuracies in all that help me recognize the pattern , overfitting and also that model behaviour also should be learned to make better performance. Through practice and improvement of the data I learned how to clean data properly, select relevant features and apply various machine learning algorithms effectively. I was able to understand the strength and weakness of supervised learning algorithm and unsupervised learning algorithms while building and comparing them. I learned gradient stacking classifiers as well and also includes them in other models for developing hybrid model which helps me to understand various multiple approaches. It helps in reducing overfitting and improve better prediction. This project also help me to visualize the mathematical equation and concepts in graph, plots, heatmaps, pie chart. This project helped me to connect theoretical knowledge with practical implementation. I learned to solve problem in better way by proper evaluation and clear interpretation of the results. Therefore , this experience increased my confidence in working with AI project and encouraged me to explore more advanced application project in future.

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.