THE UNIVERSITY OF
NEW SOUTH WALES

Project Proposal and Design

COMP9727: Recommender Systems

Name： Han Wang

ZID: z5525208

Table of Contents

1. Scope Definition

1.1 Recommendation system domain and target users

Project area: This project aims to develop a course recommendation system for online education platforms. By analyzing users' learning history, course preferences, ratings, and learning progress data, we aim to provide personalized course recommendations for college students and adult learners.

Target users: The target users of this system are college students, adult learners, and online education platform users. These users have different learning needs and preferences, so the system needs to provide customized recommendations based on each user's personal data.

1.2 Recommendation method and display method

Recommendation method: The system will generate a list of recommended courses using a hybrid recommendation model that combines content recommendation with collaborative filtering methods. Each time a user logs in, the platform will display up to five recommended courses.

User interface: The interface will provide the following functions:

Course recommendation list: Display the name, description and rating of the recommended course.

Course details: When a user selects a recommended course, the page will display detailed information about the course, including the specific content of the course, the background of the instructor, and reviews from other users.

Rating and feedback system: Users can rate the courses they have learned, and the system will optimize future recommendations based on these ratings.

1.3 Dynamic update and cold start problem

Dynamic update: The system will regularly update the recommendation list based on the user's new behavior data to ensure that the recommended content is always consistent with the user's latest interests and needs

Cold start problem:

(1) For new users: The system will use a content-based recommendation method to make preliminary recommendations based on the basic information provided by the user (such as interest areas, known course categories, etc.).

(2) For new courses: The system will generate recommendations based on the course content information (such as categories, keywords, etc.) to ensure that new courses can be effectively recommended.

1.4 Business model

In this recommendation system, the business model of recommended courses will be deeply integrated with the platform's advertising system. The platform provides personalized recommendations for course promotion through the recommendation system, which can not only increase the exposure of the course, but also increase user participation, and ultimately earn revenue through the promotion of courses and advertisements. Specifically, the system can provide the following business income methods:

Promotion course income: The platform gives priority to recommending courses or resources provided by advertisers, and advertisers pay for promotion fees.

Subscription recommendation: Through subscription services, the platform can provide more personalized and high-quality course recommendations to subscribed users to attract users to buy.

Partner promotion: The platform can cooperate with educational content providers to recommend relevant courses based on users' learning interests, and both parties share the benefits brought by the recommendation.

1.5 Ways to obtain user feedback

In order to improve the accuracy and personalization of the recommendation system, user feedback is crucial. User feedback will be obtained through the following channels:

Course rating system: Users rate courses (for example, 1-5 stars), and the rating data will be used to optimize subsequent recommendation algorithms.

Learning time and click data: By recording the learning time and click rate of users on the course, evaluate whether the course meets user needs, and adjust the recommended content.

Comments and recommendations: Users can comment and rate the recommended courses, and the system will optimize personalized recommendations based on these data.

User questionnaires: Regularly conduct user satisfaction surveys to understand users' evaluation of the recommendation system and collect feedback to improve the algorithm.

1.6 User interaction

(1) User first login:

System interaction: When a new user logs in for the first time, the system will display a simplified questionnaire asking the user about their learning interests, preferred course types (such as programming, data analysis, language learning, etc.), and previous learning experience.

System feedback: Based on the information provided by the user, the system provides a set of course recommendations through content recommendation methods (based on user interests) to help users quickly find courses that may be of interest.

(2) User browsing recommended courses:

System interaction: The user sees up to five recommended courses on the homepage. Each course includes the course name, introduction, rating, and learning progress.

User operation: The user clicks on a course to view detailed information. The course page displays the course outline, instructor introduction, comment area, and course resources (such as videos, articles, etc.).

System feedback: The user's click data will be input into the system as implicit feedback for future personalized recommendations.

(3) User rating and feedback:

System interaction: After the user finishes watching the course, the system will pop up a prompt box to invite the user to rate the course (1-5 stars) and leave a comment.

User operation: The user rates based on personal experience and fills in the comment.

System feedback: Based on the user's ratings and comments, the system adjusts the content of future recommended courses and increases the recommendation intensity for courses with high ratings.

(4) Dynamic recommendation and personalized update:

System interaction: The system regularly updates the recommendation list based on the user's learning progress and feedback. For example, if the user continuously selects a certain type of course (such as data science), the system will increase the recommendation weight of such courses.

User operation: Users can further refine the recommendation by setting preferences, for example, choosing whether to prefer certain course tags or instructors.

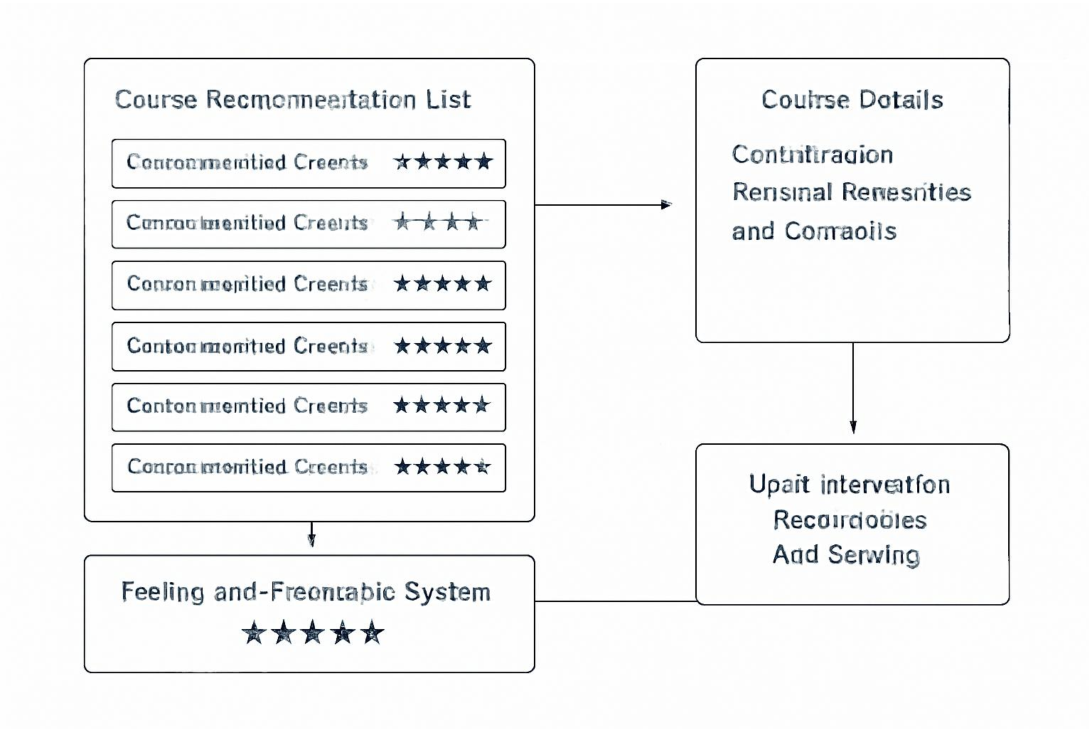
System feedback: The recommendation list is dynamically adjusted based on user preferences and feedback.

(5) Cold start problem handling:

System interaction: For new users or new courses, the system will give priority to content recommendation methods. New users can fill out a short interest survey, and the system will provide them with preliminary recommendations. New courses will be recommended based on course content and category.

1.7 Interface sketch

This is a sketch of the recommendation system interface, showing the main user interaction process:



Course recommendation list: Displays up to five recommended courses, including course name, introduction, rating, etc.

Course details: After clicking on a course, users can view course details, course outline, instructor introduction, and user comments.

Rating and feedback: Users can rate and provide feedback for courses they have taken, affecting subsequent recommendations.

User settings: Allow users to set preferences (e.g., learning areas, preferred instructors, whether to prioritize new courses, etc.).

Through the above simulation process, the system can achieve a more personalized and dynamically adjusted recommendation mechanism, and optimize the recommendation results through continuous user feedback. This interactive design ensures that every step of the user's operation has an impact on the recommendation system, thereby improving user experience and system effectiveness.

2. Dataset

2.1 Dataset description:

This study uses the open education platform course dataset from Hugging Face. This dataset contains some information about online education resources (such as courses, videos, articles, etc.), which is suitable for building recommendation systems and can provide basic data for course recommendation systems.

2.2 Dataset fields

To build a recommendation system from the fields in this dataset, we can focus on the following key fields:

1. resource_id: a unique identifier for a resource.

This field is used to distinguish different courses, articles or videos to ensure that all data resources are not repeated in the database.

2. title: a resource title.

Describe the topic of the course, article, or video in detail. This can help the recommendation system understand the course content and recommend courses that the user might like.

3. category: the subject or field to which the resource belongs.

Specific contents include language, humanities, history, mathematics, politics, etc. This part helps to classify courses and helps the system recommend relevant courses to users.

4. tags: tags associated with resources.

Tags further refine the content of the course, such as "language text classification", "artificial intelligence", "machine learning", etc. Content-based recommendation systems can use these tags to match corresponding courses

5. type: resource type (such as video, article, quiz, forum discussion, etc.).

Compared with text resources, users are generally more interested in video resources. This field can help the recommendation system recommend the corresponding resource type based on the user's personal preferences.

6. user_feedback: user feedback, such as ratings, likes, number of comments, etc.

User feedback data is the core part of collaborative filtering recommendation. The algorithm analyzes user feedback to infer which courses may be liked by users. These feedback data will be used in the collaborative filtering method to calculate the similarity between multiple users.

7. accessibility: the accessibility of resources (such as free, paid, subscription, etc.).

Resource pricing is an important factor that affects user choices. The platform can choose whether to change resource prices based on user subscriptions.

8. time_spent: the time spent by the user on the resource (if any).

If users spend more time on a course, it may indicate that the course is more attractive to them. Therefore, this data can be used as an important reference for recommending courses that users may be more willing to study in depth.

9. user_id: user ID, identifying the user who provided feedback for the resource.

This field can be used to associate users with their behaviors, build interest models for users, and then recommend personalized courses for them through collaborative filtering or content recommendation systems.

2.3 Limitations of the dataset

Although this dataset provides a lot of useful information for the recommendation system, it also has some limitations that may affect the effectiveness of the recommendation system:

1. Lack of detailed user interaction data:

This dataset does not contain the user's complete learning behavior data, such as specific learning progress, completion status, and specific interactions of the course (such as whether the course is completed, video viewing progress, etc.). The recommendation system mainly relies on behavioral data such as ratings and time spent, which may not fully reflect the user's interests and learning needs.

2. Scarce rating data:

User feedback for most courses may only include likes or viewing time, but lacks specific ratings (such as 1-5 star ratings). This scarce rating data will affect the recommendation effect based on collaborative filtering methods, especially in cold start situations, the recommendation effect for new users or new courses will be reduced.

3. Incomplete data:

The labels, descriptions, or titles of resources may be brief and may not fully reflect the actual content depth of the course. For example, the title and tags only provide general information about the course, but there is no in-depth course description, which may affect the accuracy of the content-based recommendation system.

4. Limited types of resources:

The dataset may lack certain types of learning resources (e.g., exercises, homework, discussions, etc.), which are very important on educational platforms. Different types of resources may have different appeals to different users, so the lack of diverse learning resources may affect the comprehensiveness of the recommendation.

5. Not considering dynamic updates of courses:

The course and resource information in this dataset may not take into account the updates of course content. For example, some courses may be outdated or updated, but this information is not reflected in the dataset in a timely manner. If the recommendation system does not update the content of the course in a timely manner, it may recommend courses that are no longer suitable for current needs.

2.4 Strategies for dealing with missing values

Missing values in data sets are a common problem, especially in recommendation systems, where missing user feedback data (such as course ratings, learning time, etc.) may lead to inaccurate recommendation results. For missing values in data, the following methods can be used:

1. Deletion method:

Delete records (rows or columns) containing missing values. This method is suitable for situations where the proportion of missing values is very small.

Applicable scenarios: When there are few missing values and deletion will not seriously affect the representativeness of the data set, using the deletion method is a simple and effective solution.

2. Mean/median filling method:

For numerical data (such as course ratings, learning time, etc.), the mean or median of the column data can be used to fill missing values.

Applicable scenarios: When the amount of missing data is small and the missing values are not randomly distributed, mean or median filling is a common and efficient method.

Improvement: For columns with biased data such as ratings, the median can be used for filling, because the median will not be affected by extreme values and can better reflect the real user feedback.

3. Interpolation method:

Interpolation method is a method of estimating missing values based on existing data, which is applicable to time series data (such as course learning progress, etc.). Applicable scenarios: Applicable to situations where missing data has regular changes, especially in the analysis of user behavior data and course participation.

4. Prediction filling method:

Use the values of other related fields to predict missing values. For example, you can use user historical behavior data to predict their ratings for a course, or use data from other similar users to fill in missing values.

Applicable scenarios: When the pattern of missing values has a certain relationship with other fields, you can use machine learning models (such as regression analysis, KNN, etc.) to fill them.

5. Use recommendation algorithms to deal with cold start problems:

For missing data of new users or new courses, you can use content recommendation or collaborative filtering methods to fill them. For example, for new users, you can use the basic information they provide (such as interest tags) to generate preliminary recommendations to avoid poor recommendation results due to missing data.

6. Strategies for handling noise data

Noise data refers to erroneous data that deviates from the true value, which may affect the effectiveness of the recommendation system. In practical applications, noise data may come from inaccurate ratings or accidental interactions of users. Strategies for dealing with noise data include:

Outlier detection and removal: Detect outliers through statistical methods (such as Z-score, IQR) and delete these values. Outliers may be caused by user misoperation, malicious ratings or system errors.

Applicable scenarios: Applicable to numerical data such as rating data and learning time. By setting reasonable thresholds, remove data that deviates from the normal range.

Improvement: When processing ratings, you can set reasonable upper and lower limits based on the distribution of ratings, such as removing extremely low or extremely high scores to avoid them from having excessive impact on the recommendation algorithm.

7. Smoothing: Smoothing ratings and other numerical data, such as weighted average or sliding average, to reduce data volatility.

Applicable scenarios: Applicable to user rating systems. When users rate courses too extreme, using smoothing can reduce the impact of noise on the recommendation system.

8. Weighted evaluation: When calculating recommendations, weight the user's ratings so that the ratings of users who interact frequently and have stable behavior have higher weights. This can reduce the impact of accidental behavior.

Applicable scenarios: Applicable to recommendation systems, especially in collaborative filtering methods, for some active users with frequent behaviors, give them a greater weight in their ratings, thereby reducing the influence of inactive users.

9. Feedback loop detection:

By detecting the consistency and stability of feedback, it can identify unreasonable feedback in user behavior (such as giving low scores to a large number of courses at once). The system can adjust its influence through the stability and consistency of user behavior.

Applicable scenarios: Applicable to comment and rating data, periodic analysis is performed on user behavior data to ensure the stability of the system recommendation results.

10. Combine multiple data sources to filter noise:

Combining different data sources (such as social media feedback, user historical behavior, other users' evaluations, etc.), verify and remove noise data from multiple angles.

Applicable scenarios: When the recommendation system can access multiple data sources, by integrating feedback from different information sources, noise data can be more accurately identified and eliminated.

3. Method

The fields extracted from Hugging Face - Educational Resources Dataset will be used to build a hybrid recommendation system. Based on the fields in the dataset, the recommendation system will combine content recommendation and collaborative filtering methods to improve personalized recommendation results.

3.1 Main methods

• Content-Based Filtering

• Collaborative Filtering

• Weighted Hybrid Model

The main reason for choosing these methods is that they can match the characteristics of the dataset and the needs of the target users well. The content recommendation method can provide effective recommendations for new users and users who lack sufficient behavioral data by analyzing the descriptions, tags, and categories of the courses; collaborative filtering can provide active users with accurate recommendations based on group behavior by analyzing the similarities between users; the hybrid recommendation model combines the advantages of both and provides a more flexible and personalized recommendation solution; and the weighted fusion strategy can improve the overall effect of the recommendation system by dynamically adjusting the weights of the methods.

3.2 Strategy refinement and implementation

3.2.1 Content-based recommendation

Reasons for selection:

• Dataset features: The dataset contains text information such as course labels, categories, titles, and descriptions. Content recommendation methods match courses with user interests by analyzing this information, and are suitable for recommendations based on course content and user historical behavior. Especially in the cold start phase where there is a lack of user feedback, content recommendation can make effective recommendations based on the interest information provided by users.

• Target user matching: Our target users include college students and working adult learners, who are usually interested in courses in specific fields or skills (such as programming, data analysis, etc.). Content recommendation can accurately recommend content that matches user interests based on the course categories and labels selected by users. Therefore, content recommendation methods can better meet the needs of these users and provide courses that they may be interested in.

Applicability:

• Since each course in the dataset has relevant labels and descriptions, the content of the course can be converted into a vector representation through natural language processing technologies such as TF-IDF or Word2Vec, and the similarity with the user's historical courses can be calculated. This method is particularly suitable for new users, even if their behavior data is insufficient, and preliminary recommendations can be made by providing the course categories they are interested in.Goal: By analyzing the match between course content and user interests, recommend content similar to courses that users have already studied or prefer.

(1) Implementation steps:

• Course feature extraction:

• Label and classification: Extract the subject and features of the course (such as programming, data science, language learning, etc.) from the category and tags fields. These fields will be used to generate keyword vectors for the course.

• Description analysis: Analyze the course title and description to get the core content information. Natural language processing (NLP) techniques such as TF-IDF or Word2Vec can be used to convert the course description and title into vector representation.

• Learning time and difficulty: Extract information about the course time and difficulty level from time_spent and category as features to help recommend courses suitable for users.

• User interest modeling

• Historical learning data: Use the user's historical behavior data (such as courses that have been studied and rated) to calculate the user's interests. Specifically, user_id and user_feedback (such as user ratings and comments) are used to build user interest profiles and identify user preference areas.

• Learning progress and duration: Combined with time_spent data, understand the user's involvement in a specific course type, and then predict similar courses that they may be interested in.

(2) Recommendation strategy:

• Calculate similarity: Use cosine similarity (or Euclidean distance, Jaccard similarity, etc.) to calculate the similarity between the user's historical learning courses and the courses that have not been learned. Recommend courses with high content similarity.

• Recommendation based on tags and categories: Based on the category and tags fields, recommend courses that match the user's historical preference categories or tags.

(3) Advantages:

• Can provide more accurate course recommendations for new users, even if there is a lack of sufficient historical data.

• Simple and easy to implement, especially suitable for user recommendations in the cold start phase.

3.2.2 Collaborative filtering recommendation

Reasons for selection:

Dataset features: Collaborative filtering relies on user behavior data, such as ratings, likes, learning time and other feedback information. In this dataset, although the user feedback data (such as course ratings) is not completely rich, user feedback (such as likes and learning time) can still be used as implicit feedback for recommendation.

Target user matching: For active users who already have a certain learning history (such as college students who have completed multiple courses or working adult learners), collaborative filtering can recommend courses of interest to users based on the behavior of similar users. By analyzing the behavior of multiple users, collaborative filtering can mine the similarity of interests between users and recommend courses that other similar users like to users.

Applicability:

The effectiveness of collaborative filtering in this dataset depends on the quantity and quality of user feedback data. Although some courses have less rating data, we can use implicit feedback data (such as the time and participation of users in the course) to process it. By constructing a user-course matrix, we can calculate the similarity between users and generate a personalized recommendation list.

(1) Objective: By analyzing the similarities between users and their feedback on courses (such as ratings, learning time), recommend courses that other similar users like to users.

Implementation steps:

(2) Data preparation:

• User-course rating matrix: Based on user_id and user_feedback data, construct a user-course rating matrix to record each user's rating or interactive behavior on the course (such as the number of likes, comments, etc.).

• Implicit feedback data: If there is no explicit rating data, the time_spent and accessibility fields can be used as implicit feedback data (for example, a course that users spend more time on may represent their preference for the course).

(3) Calculate user similarity:

• Use methods such as cosine similarity or Pearson correlation coefficient to calculate the similarity between users. The courses that the target user may be interested in can be inferred from the behavior of these similar users.

• For collaborative filtering, if there are enough user behaviors, you can choose project-based collaborative filtering, that is, analyze the similarity between courses and recommend other courses that are similar to the courses they have already taken to the user.

(4) Recommendation strategy:

• Recommend high-rated courses based on the ratings of similar users.

• For new users, use the KNN algorithm (nearest neighbor algorithm) to recommend courses that other users have studied that are most similar to the user's interests.

(5) Advantages:

• It can recommend courses that are not clearly marked with interests based on group behavior, which is suitable for platforms with sufficient user behavior data.

• It can mine the user's potential interests and recommend some courses that the user may not have actively searched for.

3.2.3 Weighted Fusion Strategy

Reasons for selection:

Dataset features: This dataset combines content data (such as course descriptions, tags, categories, etc.) and user behavior data (such as ratings, learning time, etc.). Therefore, the hybrid recommendation model can make full use of the advantages of these two data sources to provide more accurate and diverse recommendations.

Target user matching: For various types of users, the hybrid recommendation model can be adjusted according to different user needs and usage scenarios. For example, for new users, the system can rely more on content recommendations;

while for old users, it can rely more on collaborative filtering to adapt to their learning history and interest changes. The hybrid recommendation model can provide more flexible and comprehensive personalized recommendations, especially for complex user groups (such as college students, adult learners, etc.), because these users may be interested in various types of courses and their learning behaviors vary greatly.

Applicability:

By weighting the results of content recommendation and collaborative filtering recommendation, the hybrid recommendation model can dynamically adjust the recommendation strategy. For new users, content-based methods can be recommended first, while for active users, the effect of collaborative filtering is more prominent. Through weighted adjustment, the system can find a balance between diversity and accuracy to meet the personalized needs of different users.Goal: Combine the results of content-based recommendation and collaborative filtering for weighted fusion, comprehensively consider the advantages of both, and avoid the shortcomings of a single recommendation method.

(1) Implementation steps:

• Weighted strategy:

• Weighted fusion of the results of content-based recommendation and collaborative filtering recommendation, and assign different weights. For example:

Content recommendation weight: 60%

Collaborative filtering weight: 40%

• According to the score or ranking of each recommendation method, the final recommendation results are weighted and the final recommended course list is output.

• Dynamic weight adjustment:

• Dynamic weight adjustment based on user usage behavior For new users, content recommendation is given priority, and the course content helps build a user interest model. As user behavior data accumulates, the weight of collaborative filtering is gradually increased to enhance the recommendation effect based on group behavior.

• For active users, the weight of collaborative filtering is gradually increased to improve the personalized recommendation effect and ensure that the recommendation results can better reflect similar behaviors in the group.

(2) Adaptability of hybrid recommendation methods:

• For cold start users (new users or users with less historical data), the content-based recommendation method is preferred to provide new users with personalized recommendations based on course content to avoid the recommendation system being unable to provide any effective courses and help them quickly generate interest models.

• For active users, the recommendation effect of collaborative filtering is better. Collaborative filtering can better explore the user's potential interests and gradually introduce more recommendations based on similar user behaviors.

Therefore, the weight of collaborative filtering is gradually increased.

(3) Advantages:

• Combining the advantages of content recommendation and collaborative filtering, it overcomes the shortcomings of a single recommendation method and improves the accuracy and diversity of recommendations.

• Dynamic weight adjustment can continuously optimize the recommendation strategy according to user behavior and needs to improve the personalized recommendation effect.

4. Evaluate

4.1 Evaluation indicators:

• Accuracy: How many courses in the recommended results are actually clicked or studied by users.

• Recall: The proportion of courses that users are actually interested in in the recommended results.

• User satisfaction: Evaluate the satisfaction of recommendations through user feedback and surveys.

• Diversity and novelty: The diversity of recommended results and the proportion of recommended courses that users have never been exposed to.

4.2 Optimization strategy:

• A/B testing: Regularly verify the effectiveness of the recommendation system through A/B testing, and compare the recommendation quality of different recommendation algorithms (such as content recommendation and collaborative filtering).

• User feedback mechanism: Further optimize the recommendation model through user feedback (such as ratings, clicks, learning progress, etc.), and adjust the weights of content and collaborative filtering.

• Cold start problem handling: Provide new users with recommendations based on course content, quickly collect data through the user's first behavior, and gradually turn to collaborative filtering recommendations.

5. Time estimation

Phased estimation:

5.1 Project initiation, planning and team formation (week 1):

Determine the goals and tasks of the project.

Complete the assignment of team members and ensure that each member is clear about their tasks.

Discuss the project background and requirements, determine the domain, users and business goals of the recommendation system.

Submit a preliminary project plan and conduct the first progress check

5.2 Dataset selection and preparation (week 2):

Select a suitable dataset and perform data cleaning and preprocessing.

Deliver a data analysis report describing the characteristics, quality and limitations of the dataset, and analyze how to deal with missing values or noise in the data.

Perform data exploratory analysis (EDA) to determine whether the dataset meets the requirements for building a recommendation system

5.3 Method selection and design (end of week 2 to beginning of week 3):

Determine the recommendation method, such as content-based, collaborative filtering or hybrid methods, and describe the advantages and disadvantages of each method in detail.

Write a description of the method and decide how to implement it.

Possible preprocessing steps and feature engineering.

5.4 Model development and preliminary evaluation (week 3 to week 4):

Implement a preliminary prototype of the recommendation system, train it with relevant algorithms and output preliminary results.

Evaluate the effect of the preliminary model and determine its performance on actual data.

Preliminary evaluation of the performance and effect of the method to determine whether the model can provide effective recommendations without complex tuning.

5.5 User interface and interaction design (week 4):

Design a simple user interface for the recommendation system and describe the interaction process between the user and the system

Show the interface sketch or prototype, describe the user interaction process, and simulate the user feedback mechanism.

Whether the user interface meets expectations and can support the interaction of actual users.

5.6 Model evaluation and optimization (week 5):

Comprehensively evaluate the performance of the recommendation system based on evaluation indicators (such as accuracy, user satisfaction, etc.).

Adjust the recommendation model based on the evaluation results and perform necessary model optimization.

Evaluate the overall performance of the system and fix possible problems.

5.7 Report writing and submission (week 5):

Summarize the design plan and complete the report to ensure that the document content is clear and accurate.

Submit the project design document and project plan.

Check whether the report meets the course requirements and includes all necessary content.

5.8 Time Management Tools

Use Gantt Chart to visually display time estimates and key progress milestones. This allows you to clearly see the start and end time of each stage and whether each task can be completed on time.

Tasks can be assigned and tracked through project management tools (such as Trello, Asana, and Microsoft Project) to ensure that each member's tasks are carried out in an orderly manner.

5.9 Progress Monitoring and Adjustment

Conduct a progress review every week to check whether the project is progressing as planned. If there is any deviation from the plan, make timely adjustments and arrange the subsequent work time reasonably.

6. User research design

This study will collect user feedback through a simulated user interface. Users will experience the recommendation system during the test and fill out a short questionnaire to feedback the accuracy and satisfaction of the recommendation system.

6.1 Questionnaire Design:

- Design a questionnaire to collect user feedback quantitatively.

The questionnaire can collect user satisfaction with the recommended content through a Likert scale (1-5 points).

- Key questions include:

Accuracy of the recommendation system: Do you think the recommended courses match your interests? (1-5 points)

Diversity of the recommendation system: Do you think the recommended courses are rich in variety? (1-5 points)

Ease of use of the system: Do you think the recommendation system is easy to use? (1-5 points)

Learning value of recommended content: Do you think the recommended courses are helpful to you? (1-5 points)

6.2 Subject Selection

Select users with different backgrounds to participate in the user research to ensure that the research results are representative and universal.

6.3 User groups:

- New users (cold start problem): users who have not used the platform before and can only get course recommendations through the recommendation system. The research will focus on evaluating the effects of content recommendation and collaborative filtering, especially the initial experience of new users.

- Active users: users who have been using the platform for a while. The data of these users can be used to evaluate the personalized effect of the hybrid recommendation system (weighted fusion of content recommendation and collaborative filtering) on long-term users.

6.4 Sample size:

- Experimental group: at least 50-100 users in each group to ensure that the sample size is large enough for A/B testing and data analysis.

- Interview group: select 15-20 users for in-depth interviews to ensure that the data is representative and sufficient user feedback can be collected.

- Qualitative analysis process

We will collect detailed feedback from users on the recommendation system during the interviews, including their views on the accuracy of course recommendations, ease of use of the system, and personalized experience. During the data analysis process, we first use open coding to preliminarily classify user feedback, that is, label each user's feedback and identify different themes and patterns. For example, users may mention "accurate recommended content", "easy system operation" or "not rich enough recommended content". Then, use axial coding to further summarize these themes and group similar feedback into one category.

• Thematic analysis:

We will use thematic analysis to conduct in-depth analysis of open-ended questions and interview records to extract core themes in user feedback, such as "accuracy of course recommendations", "diversity" and "user interface design". These themes will provide direction for subsequent system optimization. We will also use qualitative analysis tools such as Nvivo or MAXQDA to encode and classify data to ensure the systematic and repeatable analysis.

• Report and summary of qualitative data:

After analyzing the qualitative data, we will summarize the main feedback from users on the system and organize it into a report, clearly indicating which aspects of improvement are most urgent. For example, some users may raise the issue of "insufficient content recommendations", which provides a basis for our subsequent system adjustments.

6.5 User characteristics:

• User's learning field preferences (such as programming, mathematics, language, etc.).

• User's learning progress and learning goals (such as entry-level or advanced learners).

• User's device usage (such as PC or mobile).

• Stratified random sampling

To ensure the representativeness of the sample, we will conduct stratified sampling among users with different backgrounds, needs, learning goals, and learning progress. Specifically, we will classify users according to their learning type (such as programming, data analysis, language learning, etc.), experience level (beginners, advanced, etc.), and device used (PC or mobile device). A certain proportion of users will be randomly selected from each category to ensure that the research sample can cover different types of users on the platform, thereby ensuring the wide applicability of the survey results.

• Specific sample selection

In the experimental group, we will select at least 50 to 100 users to ensure that the sample size is large enough to support A/B testing and data analysis. The interview group will select 15 to 20 users to ensure the in-depth interview and obtain valuable qualitative feedback. These users will be drawn from the active users, new users, and cold start users of the platform in order to obtain comprehensive feedback.

• User background: Users in each group will have different backgrounds and needs. For example, new users (cold start problems) may need more content recommendations, while active users can provide more feedback on collaborative filtering recommendations. In this way, we can fully compare the effects of content recommendation and collaborative filtering recommendation in different user groups, thus providing a strong basis for system optimization.

6.6 Data Collection and Analysis

(1) Data Collection Methods

• User Behavior Data:

Behavior data such as click-through rate, learning time, and course completion rate are collected from the platform database to reflect the interaction between users and the recommendation system.

Users' course ratings or course completion rates (such as learning time and participation) can be used as feedback data for recommendation quality.

• User Feedback Data:

Users' satisfaction with the recommendation system is obtained through online questionnaires, surveys, and in-depth interviews.

A questionnaire survey is conducted using Google Forms or the platform's own survey tool.

The interview and survey results will be recorded and converted into text data for qualitative analysis.

(2) Data analysis methods

• Quantitative analysis:

Descriptive statistics: Statistical analysis of data such as click-through rate, learning time and course completion rate to evaluate the overall effect of the recommendation system.

• Comparative analysis: Compare different recommendation methods through A/B testing and analyze the impact of data on user behavior.

• Correlation analysis: Study the relationship between recommendation accuracy and user satisfaction to determine whether the recommendation system meets user needs.

(3) Qualitative analysis:

• Conduct thematic analysis on open-ended questions in interviews and questionnaires to summarize user feedback and improvement suggestions on the recommendation system.

• Use Nvivo or MAXQDA to classify and encode user feedback and extract core ideas