Course Recommender System

with Unsupervised Machine Learning

Bo-Yan Huang
2024 1/16

Introduction - Background

In the era of digital education, our project aims to transform the learning landscape through an advanced recommender system. The mission is to enhance user experiences by seamlessly connecting learners with new, relevant courses, shaping personalized educational paths.

This initiative not only strives to improve user satisfaction but also anticipates a positive impact on company revenue. By facilitating user engagement with diverse courses, we envision a symbiotic relationship between learner contentment and business growth.

Introduction - Challenges

1. Limited Course Discoverability: Users face challenges in discovering new and relevant courses that align with their interests and learning goals, leading to a suboptimal learning experience.

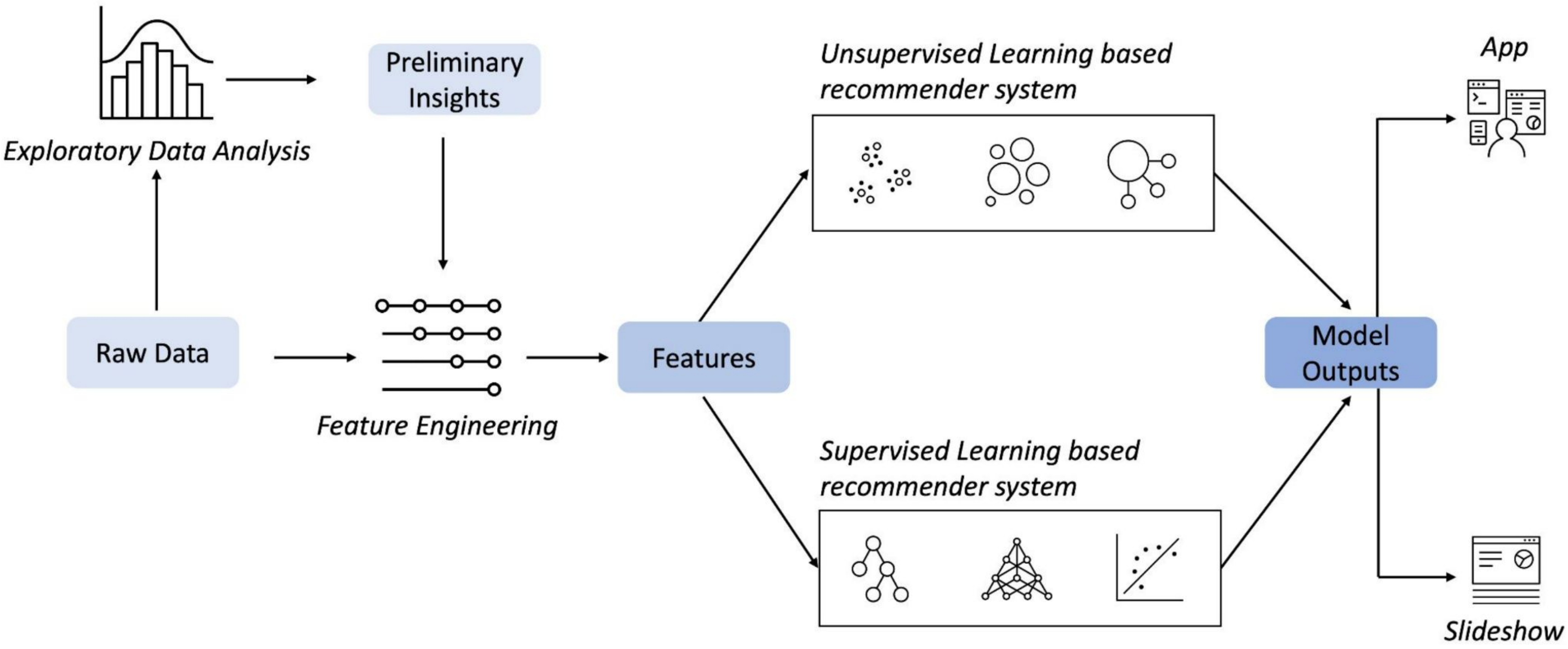
2. Underutilized Learning Paths: The absence of personalized recommendations may result in users not fully realizing the potential of a structured learning path, hindering their educational progression and engagement.

3. Revenue Growth Opportunities: The current lack of an effective recommender system may be limiting the company's revenue potential, as user interactions with courses could be suboptimal.

Introduction - Hypothesis

1. Relevance: Implementing a recommender system will significantly improve course recommendations, enhancing user satisfaction.
2. Engagement: The recommender system will boost user engagement, creating a more active and committed user base.
3. Learning Path Optimization: The system will optimize users' learning paths for a more effective educational journey.
4. Revenue Impact: Enhanced user engagement will positively affect company revenue by increasing enrollments and interactions.
5. Model Performance: In the Proof of Concept phase, exploring machine learning models aims to identify superior performance for effective online implementation.

Machine Learning Workflow



Exploratory Data Analysis

1. Statistical Overview
2. Keyword Identification using WordCloud
3. Find Popular Course Genres
4. Summary Statistics and Visualizations for Enrollment Data

Columns/Features of the Data:

```
COURSE_ID          object
TITLE              object
Database            int64
Python              int64
CloudComputing      int64
DataAnalysis        int64
Containers          int64
MachineLearning     int64
ComputerVision      int64
DataScience         int64
BigData             int64
Chatbot             int64
R                   int64
BackendDev          int64
FrontendDev         int64
Blockchain          int64
dtype: object
```

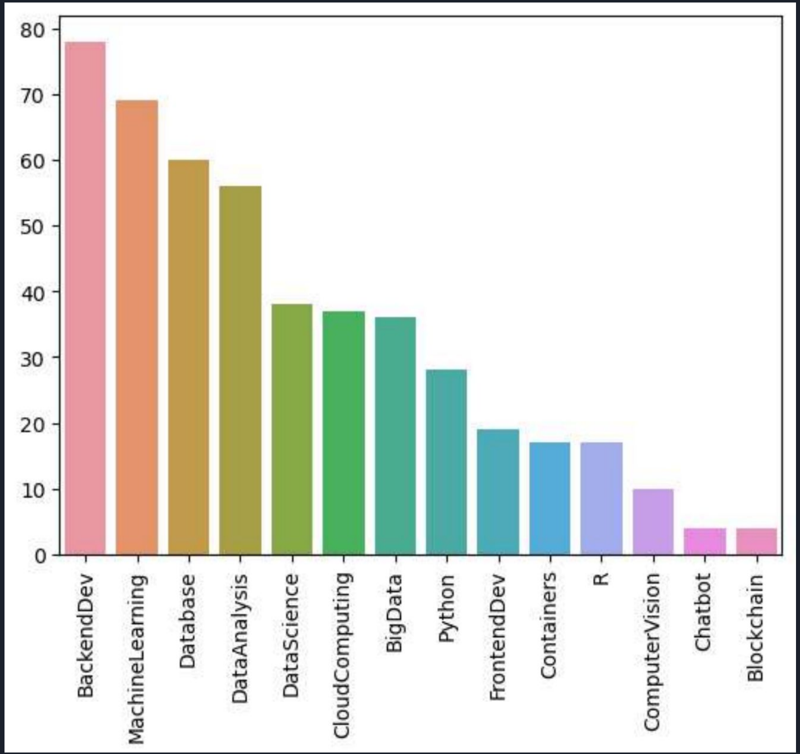
Course dataframe columns

	Python	**Database**	**Machine Learning**
user1	1.0	0	1.0
user2	0	1.0	1.0
...	...	...	...

User profile vectors

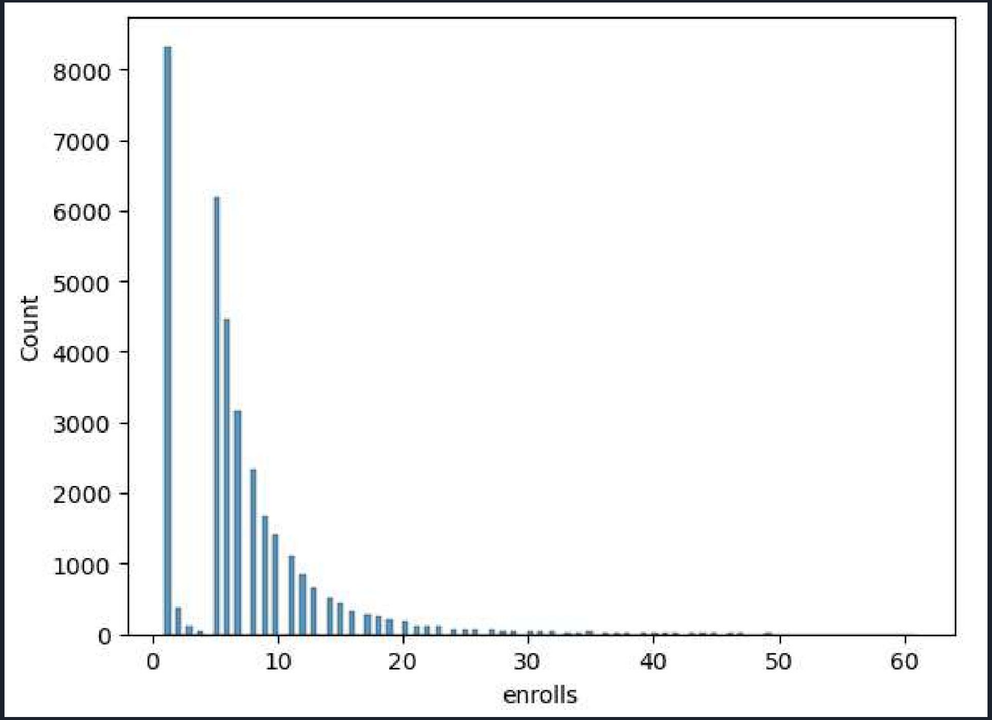
Course counts per genre

Backend Development, Machine Learning, and Database emerge as the most widely embraced genres. In contrast, Blockchain, Chatbot, and Computer Vision draw less attention."



Course enrollment distribution

The provided histogram depicts the distribution of user rating counts. The majority of users either refrained from rating any courses or did so infrequently. However, a small number of exceptional students gave ratings for more than 40 courses.



20 most popular courses

The table below displays the top 20 widely-adopted courses. Nine out of the top 10 courses pertain to the data topic, with the 4th course being the sole representative of the software engineering topic.

	TITLE	Enrolls
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199
9	blockchain essentials	6719
10	data visualization with python	6709
11	deep learning 101	6323
12	build your own chatbot	5512
13	r for data science	5237
14	statistics 101	5015
15	introduction to cloud	4983
16	docker essentials a developer introduction	4480
17	sql and relational databases 101	3697
18	mapreduce and yarn	3670
19	data privacy fundamentals	3624

Flowchart of clustering-based recommender system

Raw data → User/Item feature engineering → Group users → Recommend courses

Raw data

Course genres dataframe: `course_id, title, [genre_x, genre_y,...]`

User dataframe: `user_id, [genre_interest_x, genre_interest_y,...]`

User/Item feature engineering

Extract features from users such as enrollments or genres

Apply PCA on user profile feature vectors to reduce dimensions

Group users

Divide users into several groups based on their enrollments or genres

Recommend courses

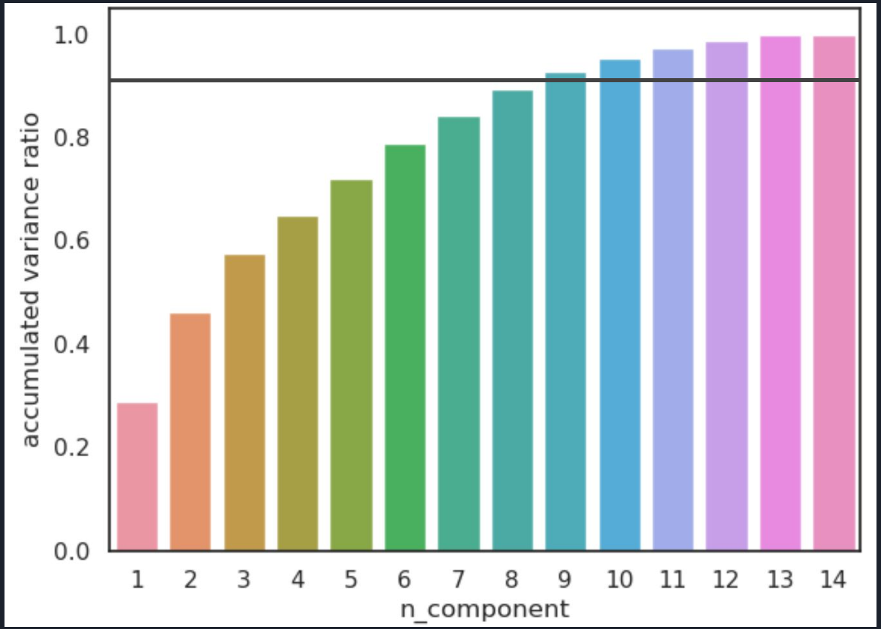
Recommend popular courses according to the preferences of other users within the same group.

Flowchart of clustering-based recommender system

User/Item feature engineering

Extract features from users such as enrollments or genres

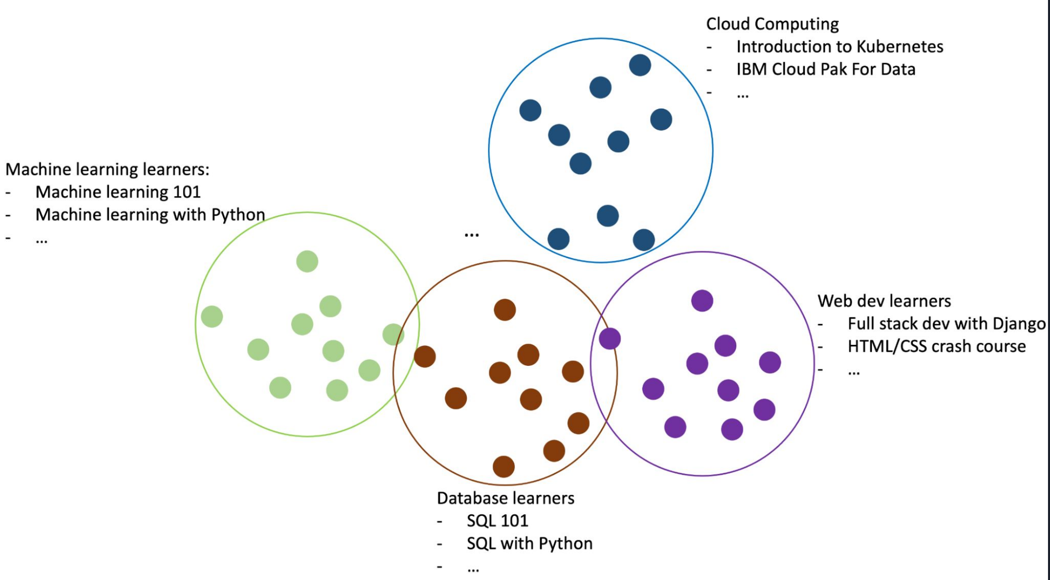
Apply PCA on user profile feature vectors to reduce dimensions



Flowchart of clustering-based recommender system

Group users

Divide users into several groups based on their enrollments or genres



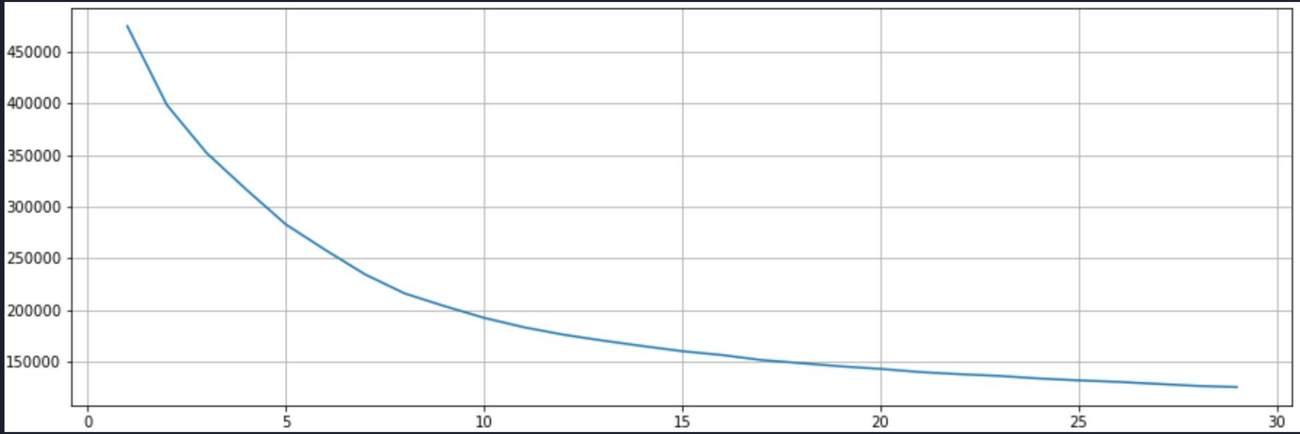
Comparing Clustering Approaches

Method name	**K-means**	**Mean-shift**	**Hierarchical clustering**	**DBSCAN**
Parameters	Number of clusters	Bandwidth	Number of clusters	Neighborhood Size
Scalability	Very large n_samples medium n_clusters with MiniBatch code	Not scalable with n_samples	Large n_samples and n_clusters	Very large n_samples, medium n_clusters
General use Case	General purpose even cluster size, flat geometry, not too many clusters	Many clusters, uneven cluster size, non-flat geometry	Many clusters, possibly connectivity constraints	Non-flat geometry, uneven cluster sizes, outlier detection
Applications	Find few clusters of roughly the same size	Can identify number of clusters, often used in video	Clusters may be of different size, does not identify outliers	Often used in computer vision applications

K-means

Inertia curve

Sum of squared distance

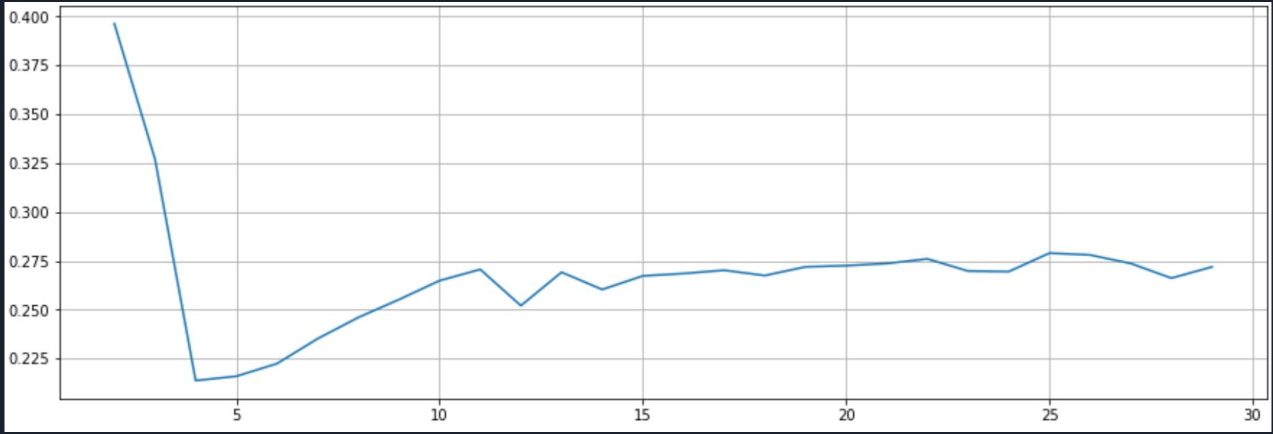


Number of cluster k

K-means

Silhouette score curve

Silhouette score

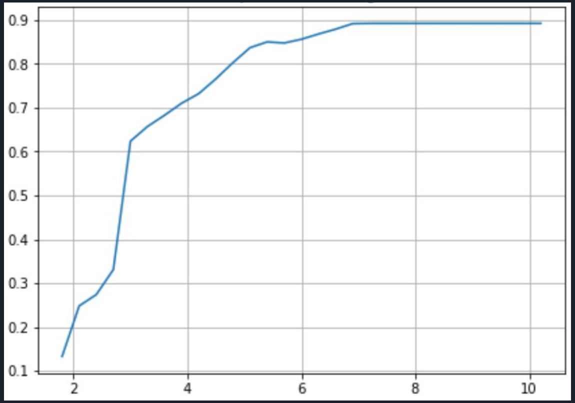


Number of cluster k

DBSCAN

Silhouette score curve

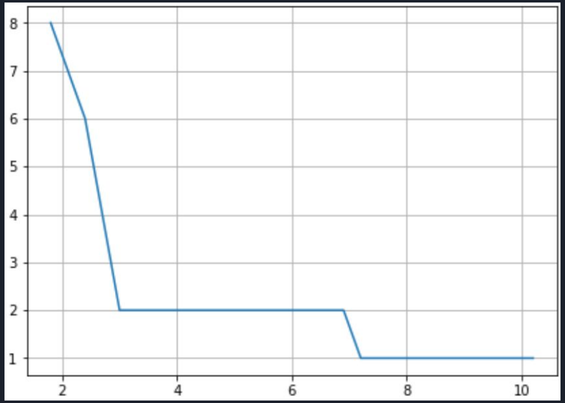
Silhouette score



Neighborhood size(epsilon)

Silhouette score curve

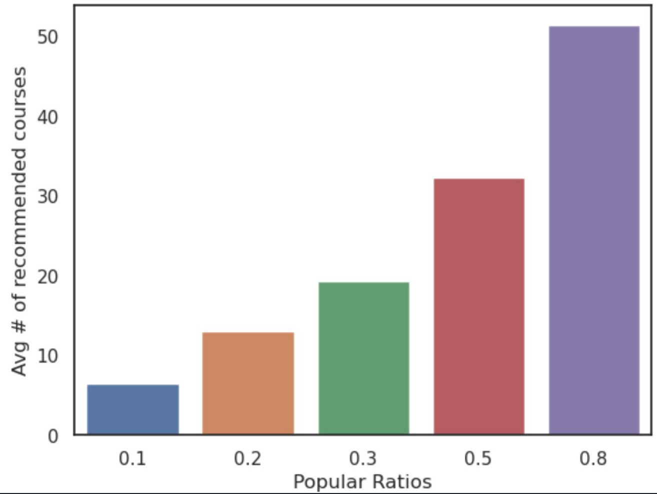
Silhouette score



Neighborhood size(epsilon)

Evaluation results of clustering-based recommender system

In the lower left chart, we adjust the popular ratio range from 0.1 to 0.8, while keeping PCA features at 9 and n_clusters at 30 for the KNN classifier. To achieve around 10 recommended courses per user, we specifically set the popular ratio to 0.2. The popular ratio is defined as the ratio of all enrollments within the user's cluster.



	COURSE_ID	recommended_count
0	DS0101EN	9372
1	PY0101EN	9255
2	BD0101EN	8882
3	BD0111EN	8042
4	ML0115EN	7553
5	DS0103EN	7293
6	ML0101ENv3	7235
7	DS0105EN	6951
8	DA0101EN	6853
9	BD0211EN	6782

Conclusions

- We can group users based on the genre in their profiles and suggest courses that are popular within the same cluster
- Applying PCA to user profile feature vectors can decrease dimensions, consequently reducing computational power requirements
- The K-means method is preferred for our project.
- We have the flexibility to fine-tune the number of recommended courses by adjusting the popular ratio parameter.

Outlook

Possible issues:

- **Cluster Interpretability:** Understanding and interpreting the meaning of clusters can be difficult. Without clear domain knowledge, it might be challenging to explain why certain items or users are grouped together.
- **Dynamic Nature of Data:** User preferences and item popularity can change over time. Unsupervised models might struggle to adapt to dynamic shifts in user behavior and preferences.
- **Lack of Personalization:** Traditional clustering methods might not capture individual user preferences well, leading to less personalized recommendations.
- **Evaluation Metrics:** Selecting appropriate evaluation metrics for unsupervised recommendation systems is challenging. Defining what constitutes a "good" clustering can be subjective.

Outlook

Possible solutions:

- **Cluster Interpretability:** Gather user-generated course ratings and construct an interpretable supervised machine learning model, such as a decision tree, to elucidate the characteristics of the identified clusters
- **Dynamic Nature of Data:** Periodically retrain the model with updated data to adapt to changes in user preferences and item popularity over time
- **Lack of Personalization:** Combine clustering with user-specific features or collaborative filtering methods to enhance the personalization of recommendations
- **Evaluation Metrics:** Define and use appropriate evaluation metrics based on the specific goals of the recommendation system. Incorporate user feedback and conduct A/B testing to assess the real-world impact of recommendations

Appendix

Data source:
https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/user_profile.csv

Courses:
https://www.coursera.org/learn/ibm-unsupervised-machine-learning/home

Jupyter notebooks:
https://github.com/r95222023/IBM-Machine-Learning-Professional-Certificate/tree/main/Unsupervised%20Machine%20Learning