MACHINE LEARNING

RECOMMENDER SYSTEM

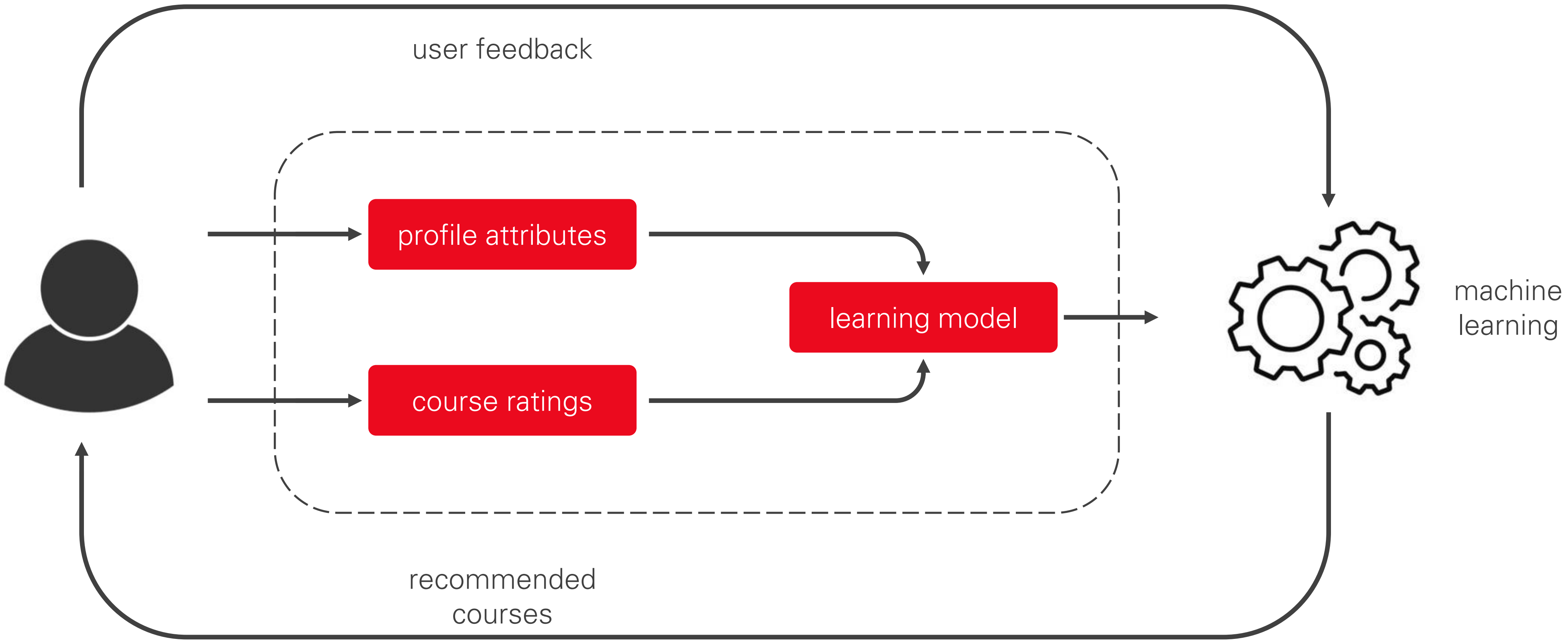
FEDERICO A. GORRINI

GitHub repository - *https://github.com/Federico-Gorrini-Ilz*

INTRODUCTION

In the age of online learning, where a vast array of courses are available across numerous platforms, finding the right courses tailored to an individual's preferences and learning needs can be complex.

A course recommender system plays a crucial role in personalizing this experience by providing learners with course recommendations based on their interests, past behavior, and other relevant factors. Such systems leverage machine learning techniques to analyze data and predict which courses will best align with each user's learning goals.



INTRODUCTION

At the core of a recommender system are several key components:

1. Users: The individuals who interact with the system, seeking to enhance their knowledge through various courses
2. Profile Attributes: These are the features that characterize a user, such as their past learning history, preferred course genres; e.g., data science, machine learning, or programming), and demographic information; e.g., age, education level, or professional background). These attributes help the system understand user preferences and make more accurate recommendations.
3. Course Ratings: Ratings or feedback given by users to courses after they complete them. These ratings are crucial for learning what works well for different types of learners and refining the model to improve future recommendations.
4. Learning Model: The underlying model or algorithm that drives the recommendation process. This could be a machine learning model such as collaborative filtering – which finds patterns in user behavior to recommend courses based on similar users; or content-based filtering – which suggests courses similar to those a user has liked in the past based on course attributes.
5. User Feedback: This refers to the data generated by users through their interactions with the system, such as course ratings, enrollments, and the amount of time spent on different courses. Feedback can also include explicit ratings – like "5 stars" or "thumbs up“ – or implicit signals – like clicks, course completions, or engagement level.
6. Recommended Courses: The output of the recommender system, which suggests a set of courses that best match a user’s profile and interests. These recommendations are dynamically adjusted based on new user interactions.
7. Machine Learning System: The computational engine that processes user data, learns from feedback, and generates recommendations. This system uses algorithms such as collaborative filtering, content-based filtering, or hybrid methods to match users with courses they are most likely to find valuable.

I. EXPLORATORY DATA ANALYSIS (EDA)

LAB 1

EXPLORATORY DATA ANALYSIS (EDA)

I. EXPLORATORY DATA ANALYSIS (EDA)

1. ANALYZING COURSE TITLES: IDENTIFYING KEY IT SKILLS VIA WORDCLOUD VISUALIZATION

In this exploratory analysis, a **WordCloud is used to highlight the most frequent terms** found in a collection of online course titles.

By organizing and filtering the data, we can identify core topics such as 'python', 'data', 'data science', 'machine learning', 'big data', 'data analysis', 'microservices', 'application', and 'database' as prominent themes.

This approach offers valuable insights into the prevailing trends in IT education, shedding light on the skills that are gaining traction in the digital learning space. It provides both learners and educators with a clearer understanding of the current educational landscape and the opportunities it presents.

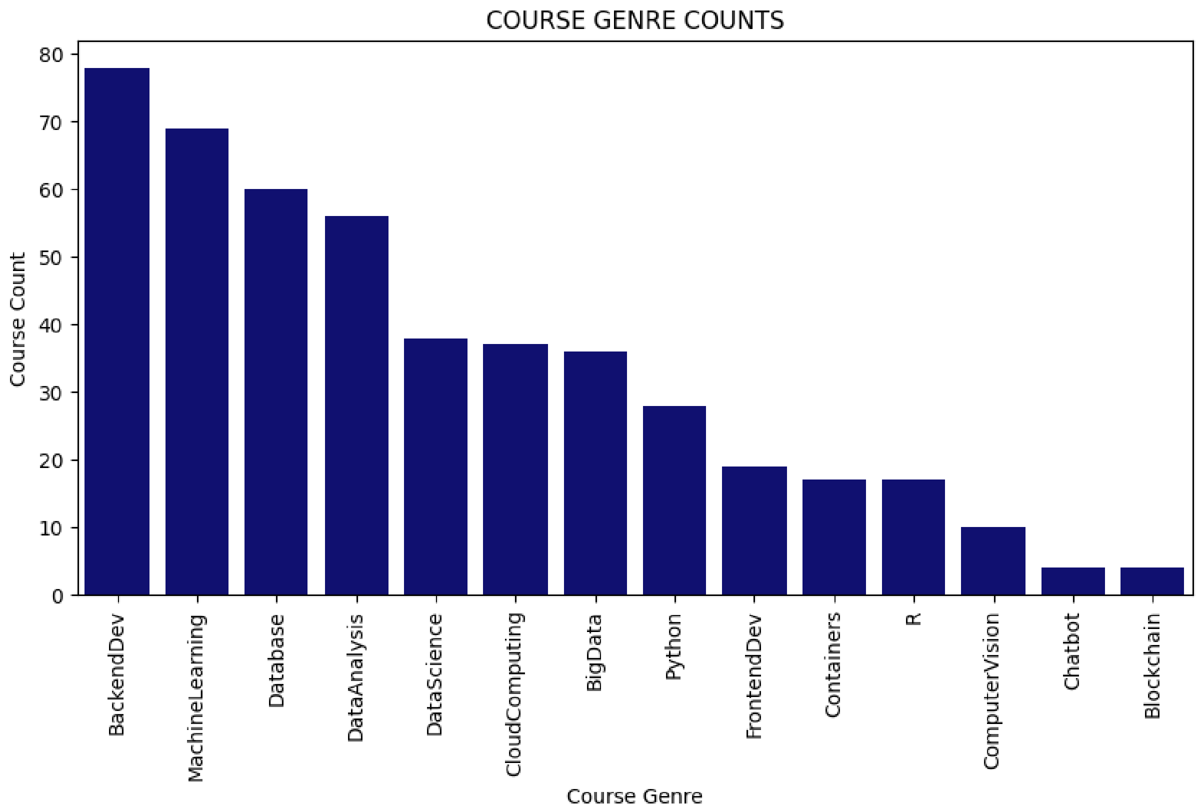
I. EXPLORATORY DATA ANALYSIS (EDA)

1. EXPLORING ONLINE COURSE CATEGORIES: IDENTIFYING POPULAR TRENDS AND TOPICS

In our examination of different course categories, we delved into the dataset to uncover which online learning subjects are most in-demand. By counting the number of courses in each category and visualizing the results using bar charts and tables, we were able to pinpoint clear trends in course popularity.

The findings revealed that courses in Backend Development, Machine Learning, and Database Management are among the top choices, while topics like Blockchain and Chatbots are less common.

This analysis provides useful insights for both students and instructors, offering a clearer picture of the prevailing trends in online education today.



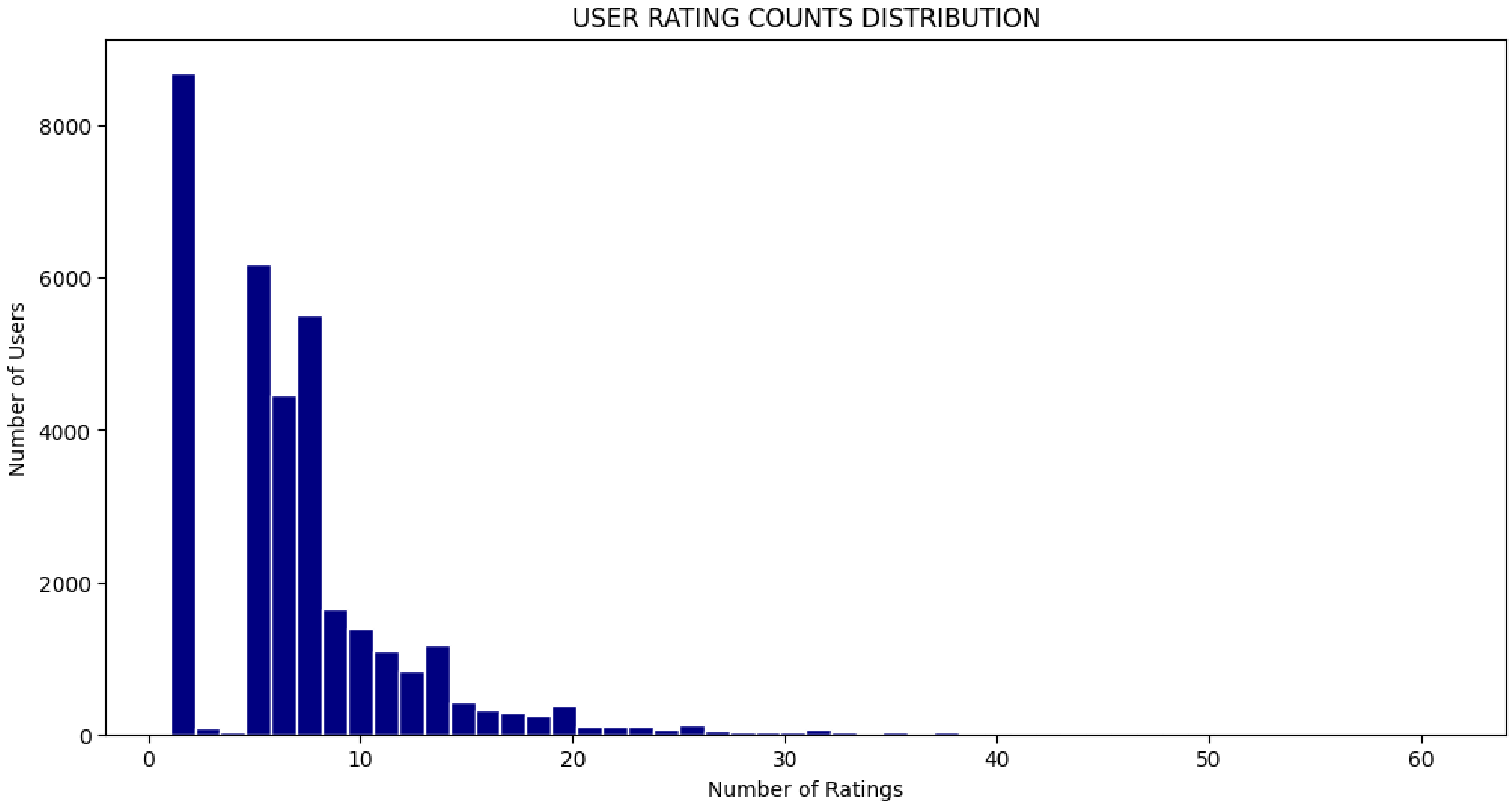
I. EXPLORATORY DATA ANALYSIS (EDA)

1. COURSE ENROLLMENT ANALYSIS: UNCOVERING TRENDS IN USER ENGAGEMENT AND INTERACTION

The dataset was examined to identify patterns in user engagement and interactions with online courses. **A total of 233,306 enrollment records**, associated with **5,000 distinct users**, were reviewed.

Through the aggregation of ratings submitted by users, it was found that most users provided only a limited number of ratings, while a smaller group contributed a much larger volume of feedback. This was visualized through a histogram, which displayed the distribution of rating counts among users.

The analysis of user engagement patterns offers valuable insights that can inform the refinement of course offerings and enhance the overall educational experience for learners.



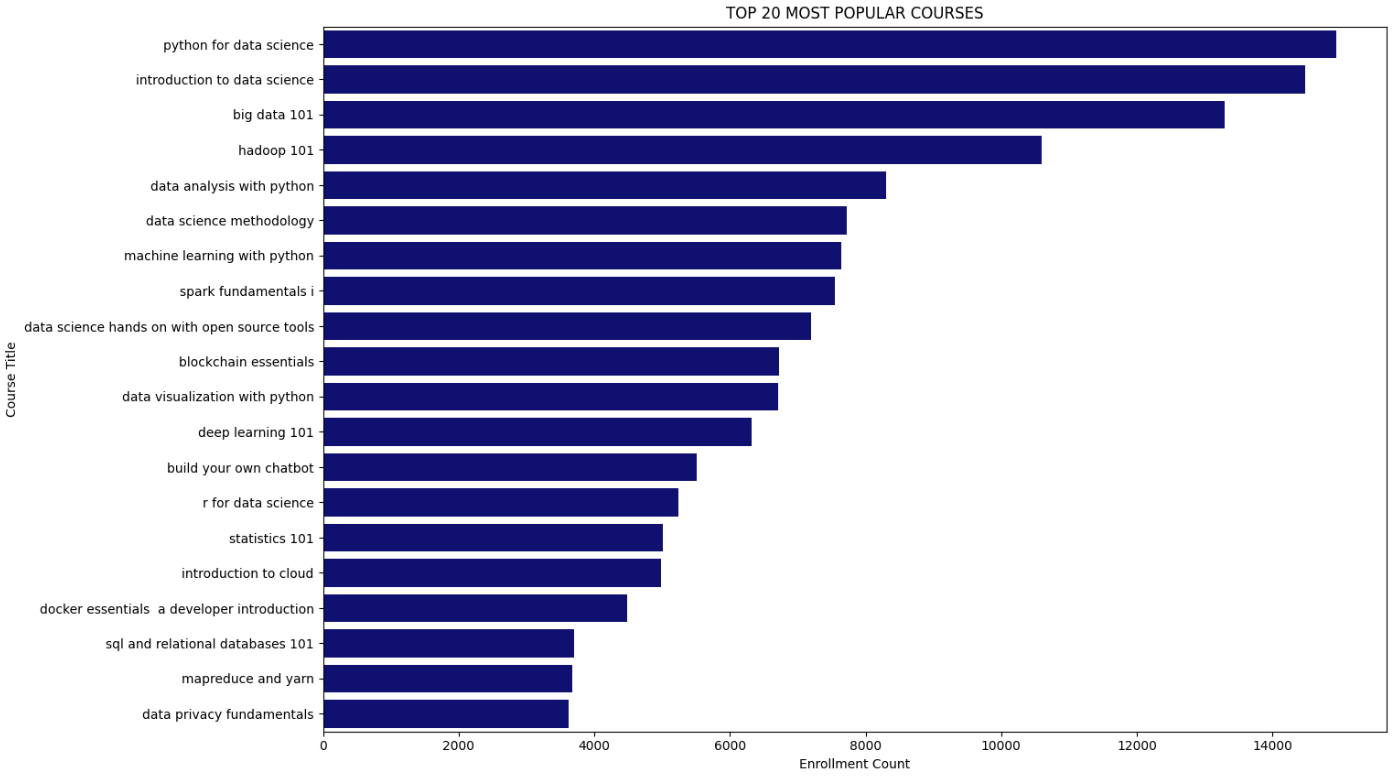
I. EXPLORATORY DATA ANALYSIS (EDA)

1. ANALYSIS: 20 MOST POPULAR COURSES

The objective of this analysis was to identify the top 20 courses according to enrollment figures. By sorting and examining the enrollment data, the courses with the highest participation were selected. The **top 20 courses captured 63.3% of all the enrollments**, being the top courses ‘python for data science’, ‘introduction to data science’, and ‘big data 101’.

	COURSE_ID	Ratings	TITLE
0	PY0101EN	14936	python for data science
1	DS0101EN	14477	introduction to data science
2	BD0101EN	13291	big data 101
3	BD0111EN	10599	hadoop 101
4	DA0101EN	8303	data analysis with python
5	DS0103EN	7719	data science methodology
6	ML0101ENv3	7644	machine learning with python
7	BD0211EN	7551	spark fundamentals i
8	DS0105EN	7199	data science hands on with open source tools
9	BC0101EN	6719	blockchain essentials
10	DV0101EN	6709	data visualization with python
11	ML0115EN	6323	deep learning 101
12	CB0103EN	5512	build your own chatbot
13	RP0101EN	5237	r for data science
14	ST0101EN	5015	statistics 101
15	CC0101EN	4983	introduction to cloud
16	CO0101EN	4480	docker essentials a developer introduction
17	DB0101EN	3697	sql and relational databases 101
18	BD0115EN	3670	mapreduce and yarn
19	DS0301EN	3624	data privacy fundamentals

percentage of the top course enrollments: 63.3%



II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

LAB 2

CONTENT-BASED RECOMMENDER SYSTEM USING USER PROFILES AND COURSE GENRES

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

2. FLOWCHART OF A CONTENT-BASED RECOMMENDER SYSTEM USING USER PROFILES AND COURSE GENRES

1. Raw Data: This stage refers to the original dataset, which contains information on users, courses, and their interactions or preferences. The raw data encompasses all necessary elements required for analysis.
2. Data Processing: In this step, the raw data is cleaned and prepared for further analysis. Tasks such as handling missing values, removing duplicates, and transforming the data into an appropriate format are carried out to ensure readiness for the next phases.
3. Cleaned Dataset: Once the data processing is completed, a refined dataset is produced. This dataset is now free from errors and inconsistencies, making it ready for feature engineering.
4. Feature Engineering: During this process, new features or representations are generated from the data to be used in training a machine learning model. Specifically, user profile vectors and course genre vectors are created. These vectors capture user preferences and the characteristics of various courses.
5. Features: The final output of feature engineering consists of user profile vectors and course genre vectors. These features serve as the inputs to the content-based recommender system. By comparing the user profile vectors with course genre vectors, personalized recommendations can be generated based on the individual interests of each user.

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

2. EVALUATION RESULTS OF THE USER PROFILE-BASED RECOMMENDER SYSTEM

- Hyperparameter Settings: A recommendation score threshold of 10.0 was set in the analysis to filter out low-scoring recommendations. This threshold was used to determine which courses were considered relevant enough to be recommended to users. Other hyperparameters, such as the methods for feature representation or similarity metrics, may also have been fine-tuned during the development of the recommender system.

- Average Number of New Courses Recommended per User: The average number of new course recommendations per user in the test dataset was calculated. This metric helps to evaluate the coverage and diversity of the recommender system. On average, approximately 61.82 courses were recommended to each user.

- Top 10 Most Frequently Recommended Courses: A table was created to list the top 10 courses most frequently recommended by the user profile-based recommender system. Each course is identified by its COURSE_ID, with the RECOMMENDATION_COUNT column showing the number of times it was recommended. These recommendations were generated by analyzing user profiles and course genre vectors, with the courses most aligned with user interests being recommended more frequently.

	COURSE_ID	RECOMMENDATION_COUNT
0	TA0106EN	608
1	GPXX0IBEN	548
2	excourse22	547
3	excourse21	547
4	ML0122EN	544
5	GPXX0TY1EN	533
6	excourse04	533
7	excourse06	533
8	excourse31	524
9	excourse73	516

LAB 3

CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

3. FLOWCHART OF A CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY

1. Raw Data: The initial dataset consists of details about various courses, such as their titles, descriptions, and other relevant attributes.
2. Data Processing: In this step, the raw data is preprocessed through techniques like breaking down the text into individual words (tokenization) and standardizing them to their base forms (lemmatization). These processes help simplify and normalize the text for further analysis.
3. Cleaned Dataset: The dataset is cleaned by removing stop-words (commonly used words with little significance) and filtering out irrelevant or noisy data points (outliers).
4. Feature Engineering: The cleaned dataset is converted into numerical representations of the courses. Specifically, Term Frequency-Inverse Document Frequency (TF-IDF) vectors are generated, which reflect the importance of words in each course's description relative to the entire dataset.
5. Features: The final set of features consists of the TF-IDF vectors created during feature engineering. These vectors are used to calculate the similarities between courses, enabling the system to recommend courses based on their content similarity.

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

3. EVALUATION RESULTS OF A CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY

- Hyper-parameter Settings: For the course similarity-based recommender system, a similarity threshold of 0.6 was established. This threshold defined the minimum level of similarity between courses required for them to be recommended to users.
- Average Number of New Courses Recommended per User: The average number of new or previously unseen courses recommended to each user in the test dataset was calculated to be approximately 0.987. This metric offers insight into the diversity of recommendations provided to users and helps evaluate the system's ability to introduce novel content.
- Top-10 Most Frequently Recommended Courses: The top 10 most frequently recommended courses across all users were identified. "excourse22" and "excourse62" emerged as the most recommended, each appearing 257 times, followed by "WA0103EN," which was recommended 101 times. Other courses, such as "TA0105" and "DS0110EN," also saw frequent recommendations, underscoring their relevance and popularity among users.

	COURSE_ID	RECOMMENDATION_COUNT
0	excourse62	257
1	excourse22	257
2	WA0103EN	101
3	TA0105	41
4	DS0110EN	38
5	excourse46	24
6	excourse47	24
7	excourse63	23
8	excourse65	23
9	TMP0101EN	17

LAB 4

CLUSTERING-BASED COURSE RECOMMENDER SYSTEM

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

3. FLOWCHART OF A CONTENT-BASED RECOMMENDER SYSTEM USING CLUSTERING-BASED RECOMMENDER SYSTEM

1. **Raw Data**: The raw data comprises the initial user profile feature vectors, containing information about users' preferences and interests across different course genres.

2. **Data Processing**: At this stage, preprocessing is performed to address missing values, outliers, and other data quality issues. Normalization techniques, such as the use of *StandardScaler*, are applied to ensure consistency in the scale and distribution of all features. This step is vital for the effective performance of clustering algorithms and other machine learning models.

3. **Cleaned Dataset**: Following the data processing, the dataset is standardized. Methods like *StandardScaler* are used to ensure that each feature has a mean of 0 and a standard deviation of 1, which is a common practice in many machine learning applications.

4. **Feature Engineering**: During this step, dimensionality reduction is performed using Principal Component Analysis (PCA). This technique transforms the original user profile features into a set of key components (eigenvectors), capturing the most significant variations in the data. The original features are projected onto these new components, effectively reducing the dimensionality of the data.

5. **Features**: The final output consists of the features transformed through PCA. These represent a lower-dimensional version of the original user profile data, where each feature is a combination of information from the original variables, making the data more compact and easier to work with.

II. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

4. EVALUATION RESULTS OF THE CLUSTERING-BASED RECOMMENDER SYSTEM

- Hyper-parameter Settings: The hyperparameters were fine-tuned to optimize the performance of the system. The K-means algorithm was used to determine the optimal number of clusters, with the elbow method applied to identify the ideal value. For Principal Component Analysis (PCA), components were selected that explained more than 90% of the variance in the dataset. This approach ensured that the recommender system could effectively group users based on their preferences while minimizing information loss during dimensionality reduction.
- Average Number of New Courses Recommended per User: The system's performance was evaluated, revealing that an average of 36.587 new or unseen courses were recommended to each user in the test dataset. This metric highlights the system’s capability to offer diverse recommendations, expanding the range of learning opportunities available to users beyond their past course selections.
- Top-10 Most Frequently Recommended Courses: An analysis of the most frequently recommended courses provided key insights into user preferences across different clusters. Courses such as "WA0101EN," "DB0101EN," and "DS0301EN" emerged as the most commonly recommended, indicating their widespread appeal among users from various clusters. These insights will inform future refinements in recommendation strategies, helping to better align course offerings with user preferences and learning goals.

	COURSE_ID	RECOMMENDATION_COUNT
0	WA0101EN	864
1	DB0101EN	857
2	DS0301EN	856
3	CL0101EN	852
4	ST0101EN	800
5	CO0101EN	783
6	RP0101EN	773
7	CC0101EN	769
8	DB0151EN	741
9	ML0120EN	738

III. COLLABORATIVE-FILTERING RECOMMENDER SYSTEM USING SUPERVISED LEARNING

LAB 5

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM USING KNN

III. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

5. FLOWCHART OF A KNN-BASED RECOMMENDER SYSTEM

1. Raw Data: The raw data consists of the initial dataset that records user-item interactions, such as user IDs, course IDs, and ratings or feedback (e.g., enrollments). This dataset serves as the foundation for subsequent analysis and recommendation modeling.
2. Data Processing: In this stage, essential tasks are performed, including loading the dataset, handling missing values, removing duplicate entries, and converting the data into a structured format. This step ensures the dataset is clean and prepared for further analysis, making it suitable for building the recommendation model.
3. Cleaned Dataset: After data processing, a refined dataset is obtained, where irrelevant or erroneous data points have been eliminated, and missing values have been appropriately addressed. This cleaned dataset is now in a structured and consistent format, ready to support the recommendation system.
4. Feature Engineering: Feature engineering involves creating new features or transforming existing ones to enhance the model's predictive power. This may include extracting crucial information such as user-item interactions, timestamps, or user demographics. Other tasks could include encoding categorical variables, scaling numerical features, and generating new interaction terms that reflect relationships between users and courses.
5. Features: The final set of features, used by the KNN-based recommender system for making predictions, captures the relationships between users and courses. These features enable the identification of patterns and allow for personalized recommendations. They may include user demographics, item characteristics, and historical interactions. In some cases, Principal Component Analysis (PCA) is applied to reduce the dimensionality of the features, retaining only the most significant components, which helps improve model performance.

$$RMSE = 0.2063$$

LAB 6

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM USING NMF

III. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

6. FLOWCHART OF A NMF-BASED RECOMMENDER SYSTEM

1. Raw Data: The raw data refers to the unprocessed dataset, which, in this case, consists of course ratings. This data is typically noisy and may contain missing values or inconsistencies, requiring preprocessing before it can be used for further analysis.
2. Data Processing: At this stage, the raw data undergoes cleaning and preparation for analysis. Tasks performed include handling missing values, removing duplicates, and reshaping the data into a more suitable format. For this example, Pandas was used to pivot the dataset, transforming it into a user-item matrix where each row corresponds to a user, and each column represents a course.
3. Cleaned Dataset: After data processing, a cleaned dataset is obtained, free from inconsistencies and ready for analysis. This dataset typically arranges users as rows and items (courses) as columns, with the ratings or interactions filling the respective cells.
4. Feature Engineering: Feature engineering involves the creation or transformation of features to improve the performance of machine learning models. In this case, new features might be generated, such as user-item interaction histories or latent factors extracted through techniques like Non-negative Matrix Factorization (NMF). These latent factors help capture underlying patterns in user behavior.
5. Features: Features are the variables that machine learning models use to make predictions. In collaborative filtering, these features can include user preferences, course attributes, or latent factors that represent users and items in a reduced-dimensional space. These features enable the model to understand similarities between users and courses, ultimately improving the accuracy of recommendations.

$$RMSE = 0.2048$$

LAB 7

COLLABORATIVE FILTERING BASED RECOMMENDER SYSTEM USING NEURAL NETWORK

III. CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING

7. FLOWCHART OF A NEURAL NETWORK EMBEDDING-BASED RECOMMENDER SYSTEM

1. Raw Data: The raw data refers to the initial, unprocessed dataset, which, in this case, consists of course ratings. This raw data often contains noise, missing values, or inconsistencies, all of which need to be addressed before it can be analyzed.
2. Data Processing: During this phase, the raw data is cleaned and organized to make it suitable for analysis. Tasks such as handling missing values, removing duplicates, and reshaping the data are performed. For example, Pandas was used to pivot the dataset into a user-item matrix, where each row represents a user and each column represents a course they have rated.
3. Cleaned Dataset: Once the data processing is complete, a cleaned and refined dataset is obtained. This dataset is free of inconsistencies and is structured for analysis. Typically, it consists of rows representing users, columns representing items (courses), and cells populated with ratings or other forms of interaction data.
4. Feature Engineering: Feature engineering focuses on the creation or transformation of features to enhance the performance of machine learning models. In the case of a neural network embedding-based system, user-item interaction embeddings are automatically learned during training. These embeddings capture latent patterns in user preferences and item characteristics without the need for traditional factorization techniques like Non-negative Matrix Factorization (NMF). The learned embeddings enable the system to identify similarities between users and courses, which improves the accuracy of personalized recommendations.
5. Features: Features are the variables or attributes used by the machine learning model to make predictions. In collaborative filtering, these features can include user preferences, course characteristics, or latent factors that represent users and items in a lower-dimensional space. This allows the model to understand similarities between users and courses more effectively, leading to more accurate recommendations.

$$RMSE = 0.1534$$

III. COLLABORATIVE FILTERING MODELS: COMPARATIVE

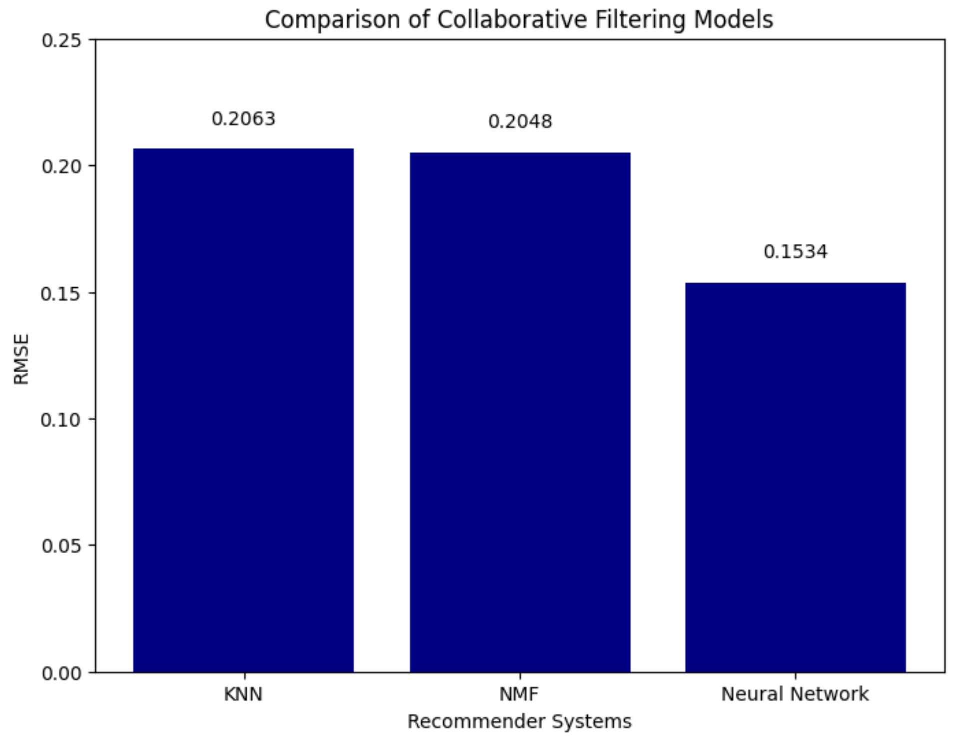
COURSE ENROLLMENT ANALYSIS: UNCOVERING TRENDS IN USER ENGAGEMENT AND INTERACTION

Based on the evaluation results, **the Neural Network Embedding-based recommender system achieved the lowest RMSE** (Root Mean Square Error) value of **0.1534**, reflecting its superior performance in predicting user-item interactions when compared to the other models.

Consequently, it can be concluded that the Neural Network Embedding-based recommender system **is the most effective collaborative filtering model** for this particular context.

Recommender systems based on KNN and NMF achieved a similar RMSE.

Recommender System (RS)	RMSE
KNN-based (RS)	0.2063
NMF-based (RS)	0.2048
NN Embedding-based (RS)	0.1534



IV. CONCLUSIONS

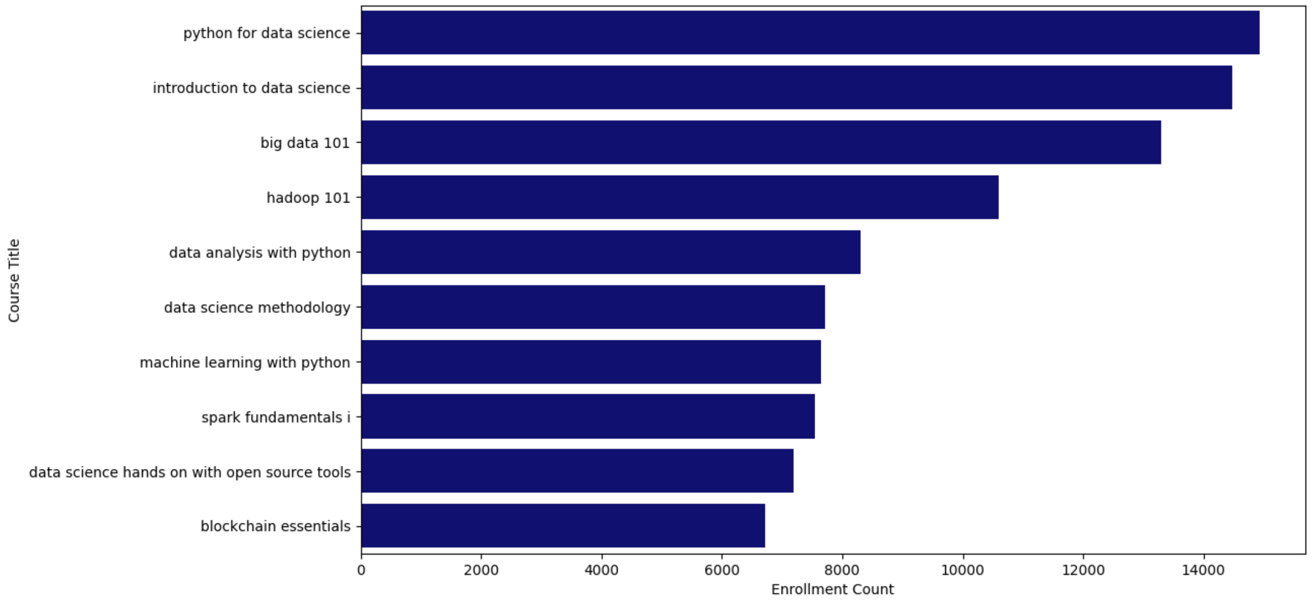
IV. CONCLUSION

CONCLUSION OF EXPLORATORY DATA ANALYSIS

Through the various analyses conducted, several key insights have emerged regarding both course offerings and user engagement on the online learning platform.

The examination of course genres revealed a broad spectrum of topics, with backend development, machine learning, and databases identified as the most popular areas, based on enrollment numbers. Furthermore, the analysis of user enrollments provided valuable insights into user behavior, indicating that a significant proportion of users completed courses rather than merely auditing them.

In addition, **the review of the top 20 most popular courses highlighted a notable emphasis on data-related subjects**, including courses such as Python for Data Science, Introduction to Data Science, and Big Data 101, reflecting the growing demand for skills in data analysis and machine learning.



IV. CONCLUSION

CONCLUSION OF CONTENT-BASED RECOMMENDER USING USER PROFILES AND COURSE GENRES

The content-based recommender system, leveraging user profiles and course genres, operates by generating course recommendations based on the vectors derived from user profiles and course genres. The process begins with the loading of user profile and course genre data, followed by the extraction of user interests, identification of courses unfamiliar to each user, computation of recommendation scores, and filtering of courses that fall below a predefined threshold.

Once the recommendation scores are calculated for all test users, the system's performance is assessed. This evaluation involves calculating the average number of courses recommended per user, as well as identifying the top 10 most frequently recommended courses across the user base.

On average, 61.82 courses are recommended to each user, reflecting a considerable volume of suggestions. Among the most frequently recommended courses are "TA0106EN," "GPXX0IBEN," and others, highlighting their popularity within the set of recommended courses.

	USER	COURSE_ID	SCORE
0	37465	RP0105EN	27.0
1	37465	GPXX06RFEN	12.0
2	37465	CC0271EN	15.0
3	37465	BD0145EN	24.0
4	37465	DE0205EN	15.0
...	...	...	...
53406	2087663	excourse88	15.0
53407	2087663	excourse89	15.0
53408	2087663	excourse90	15.0
53409	2087663	excourse92	15.0
53410	2087663	excourse93	15.0

	COURSE_ID	RECOMMENDATION_COUNT
0	TA0106EN	608
1	GPXX0IBEN	548
2	excourse22	547
3	excourse21	547
4	ML0122EN	544
5	GPXX0TY1EN	533
6	excourse04	533
7	excourse06	533
8	excourse31	524
9	excourse73	516

IV. CONCLUSION

CONCLUSION OF CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY

The content-based recommender system utilizing course similarity was successfully implemented and evaluated. **A similarity threshold of 0.60 was employed** to recommend courses based on users' interests and their prior course selections. By analyzing course content and calculating similarity scores, the system provides personalized recommendations that align with individual user preferences.

The evaluation yielded valuable insights, such as the average number of new courses recommended per user and the most frequently suggested courses. These results demonstrate the system's effectiveness in delivering relevant and engaging content. Additionally, the findings offer valuable direction for potential optimizations and enhancements aimed at improving the system's performance in future versions.

	USER	COURSE_ID	SCORE
0	37465	[]	[]
1	50348	[]	[]
2	52091	[ML0120EN, ML0120ENv2, ML0120ENv3]	[0.9828731898973628, 0.9828731898973628, 0.982...
3	70434	[]	[]
4	85625	[TMP0101EN, TA0105EN, BD0151EN]	[0.8894991799933215, 0.6598288790738579, 0.630...
...	...	...	...
995	2061096	[]	[]
996	2074313	[excourse62, excourse22]	[0.6475015976638527, 0.6475015976638527]
997	2074462	[]	[]
998	2082818	[]	[]
999	2087663	[]	[]

	COURSE_ID	RECOMMENDATION_COUNT
0	excourse62	257
1	excourse22	257
2	WA0103EN	101
3	TA0105	41
4	DS0110EN	38
5	excourse46	24
6	excourse47	24
7	excourse63	23
8	excourse65	23
9	TMP0101EN	17

IV. CONCLUSION

CONCLUSION OF CLUSTERING-BASED RECOMMENDER SYSTEM

The clustering-based course recommender system has shown strong performance in effectively grouping users based on their preferences and recommending relevant courses. By optimizing critical hyperparameters, such as the number of clusters and the selection of PCA components, the system successfully captures user interests while minimizing information loss.

Through thorough evaluation, it was found that **the system recommends an average of 36.587 new or previously unseen courses per user**, highlighting its capacity to provide diverse recommendations.
Furthermore, the identification of frequently recommended courses, including "WA0101EN," "DB0101EN," and "DS0301EN," provides valuable insights into user preferences and underscores the system's ability to highlight popular courses across various clusters.

	user	item	cluster
0	1502801	RP0105EN	0
1	1609720	CNSC02EN	1
2	1347188	CO0301EN	3
3	755067	ML0103EN	0
4	538595	BD0115EN	0
...	...	...	...
9397	1385217	EE0101EN	0
9398	1864644	DA0101EN	1
9399	435858	TMP0105EN	4
9400	1888188	DB0101EN	3
9401	708518	RP0101EN	2

	COURSE_ID	RECOMMENDATION_COUNT
0	WA0101EN	864
1	DB0101EN	857
2	DS0301EN	856
3	CL0101EN	852
4	ST0101EN	800
5	CO0101EN	783
6	RP0101EN	773
7	CC0101EN	769
8	DB0151EN	741
9	ML0120EN	738

IV. CONCLUSION

CONCLUSION OF PERFORMANCE OF THREE COLLABORATIVE FILTERING MODELS: KNN-BASED, NMF-BASED, AND NEURAL NETWORK EMBEDDING-BASED RECOMMENDER SYSTEMS

The comparative analysis of the KNN-based, NMF-based, and Neural Network Embedding-based collaborative filtering methods has provided key insights. It was observed that both the **KNN-based and NMF-based systems tend to produce higher RMSE values compared to the Neural Network Embedding method. This indicates that the Neural Network Embedding approach is more effective at identifying latent patterns in user-item interaction data**, leading to enhanced prediction accuracy.

The advantage of the Neural Network Embedding method lies in its capacity to capture complex, nonlinear relationships between users and items, allowing for the detection of subtle preferences. **However, despite its higher accuracy, this method often requires greater computational resources and longer training times due to its more complex architecture.**

Therefore, the selection of the optimal collaborative filtering technique should be based on the specific needs of the system, balancing the trade-off between predictive performance and resource efficiency.

Recommender System (RS)	RMSE
KNN-based (RS)	0.2063
NMF-based (RS)	0.2048
NN Embedding-based (RS)	**0.1534**

V. INNOVATIVE INSIGHTS

V. INNOVATIVE INSIGHTS

1. HYPER-PERSONALIZATION WITH DEEP LEARNING

Deep learning models enhance hyper-personalization by automatically learning complex patterns from user interaction data. This allows recommender systems to make more granular and context-aware recommendations based on real-time preferences, user mood, or specific tasks.

2. ACTIVE LEARNING AND REAL-TIME FEEDBACK

Active learning, where users provide explicit feedback on recommendations, can help refine the system's suggestions in real time. This dynamic feedback loop enables continuous improvement of recommendations as user interests evolve.

3. MULTIMODAL DATA FOR RICHER RECOMMENDATIONS

By incorporating multimodal data like text, video, and audio, recommender systems can provide richer, contextually-aware suggestions. This approach offers a more holistic view of user preferences, improving relevance and engagement.

4. CROSS-DOMAIN RECOMMENDATIONS

Cross-domain systems can recommend content across different categories, such as courses, books, or podcasts. This broadens the scope of recommendations, providing users with a richer array of learning resources tailored to their holistic interests.

5. CONTEXT-AWARE, TIME-SENSITIVE RECOMMENDATIONS

Context-aware systems consider factors like time of day or current activity, tailoring recommendations to the user’s specific context. Time-sensitive suggestions can help learners manage their schedules and preferences effectively.

V. INNOVATIVE INSIGHTS

6. SOCIAL LEARNING INTEGRATION

Social learning elements, such as peer recommendations or group-based preferences, can enhance collaborative filtering. Systems will consider user networks or communities to provide socially-informed recommendations that reflect shared learning goals.

7. DYNAMIC LEARNING PATHWAYS

Recommender systems could design dynamic learning pathways, adapting as users progress through courses. These pathways ensure continuous, relevant learning experiences and avoid redundancy by suggesting courses that match long-term learning goals.

8. REAL-TIME ADAPTATION VIA REINFORCEMENT LEARNING

Reinforcement learning allows systems to adapt in real time based on user feedback, improving recommendation quality. This continuous learning process ensures that suggestions evolve in line with changing user preferences.

9. GAMIFICATION TO BOOST ENGAGEMENT

Integrating gamification elements, like challenges and rewards, can motivate users to engage more deeply with content. Personalized, goal-driven recommendations can enhance user experience and encourage continuous learning.

GitHub repository - *https://github.com/Federico-Gorrini-Ilz*