Introduction to

Machine Learning Capstone

by: Amanatullah Pandu Zenklinov

IBM

25/Apr/2024

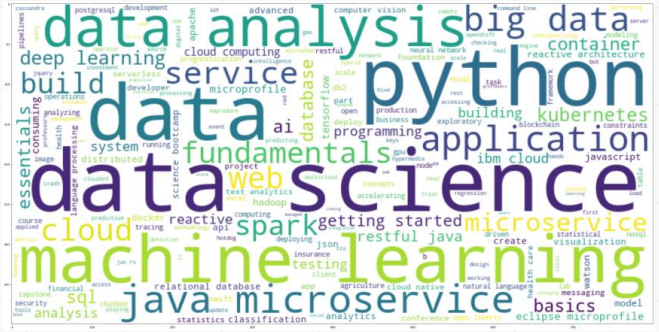
Table of contents

01

Exploratory Data Analysis

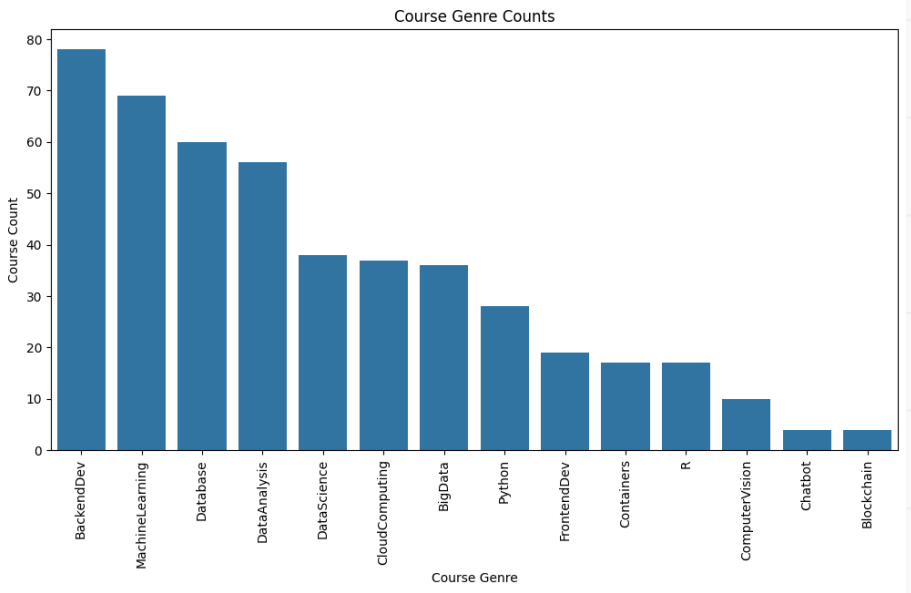
Exploring Online Course Titles: Unveiling Popular IT Skills through WordCloud Analysis

In this exploratory data analysis, the utilization of WordCloud visualization unveils prevalent keywords within a dataset of online course titles. By aggregating and filtering course titles, dominant themes such as Python, data science, machine learning, big data, AI, TensorFlow, containers, and cloud computing are identified. This analysis provides valuable insights into the trending IT skills and subject areas covered by the online courses, guiding learners and educators in understanding the current landscape of digital learning opportunities.



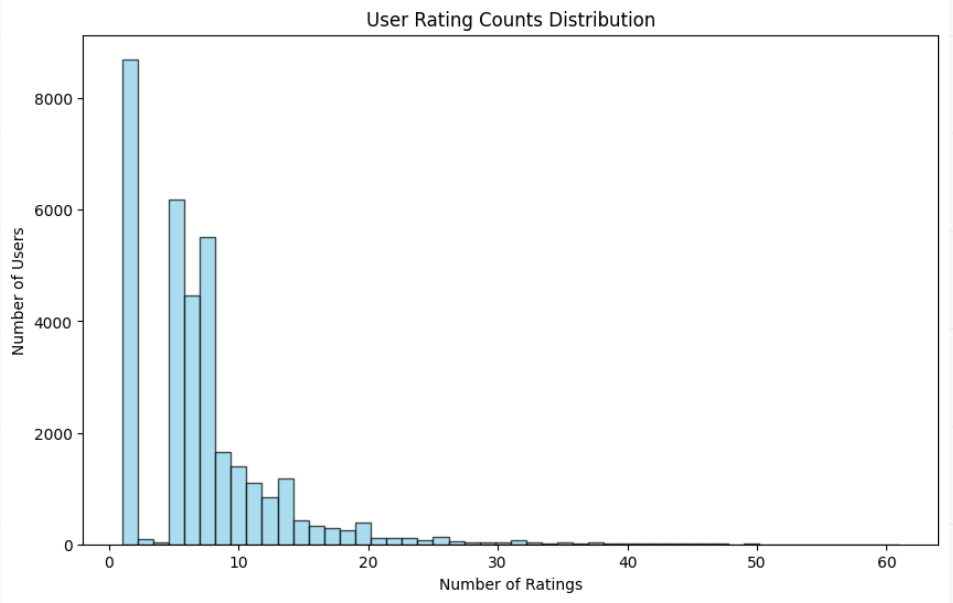
Analyzing Online Course Genres: Unveiling Popular Topics and Trends

In the process of analyzing course genres, the dataset was examined to identify the popularity of various online course topics. By calculating course counts for each genre and visualizing the results through a bar chart and table, insights into the most sought-after subjects were revealed. The analysis revealed that Backend Development, Machine Learning, and Database courses are among the most prevalent, while Blockchain and Chatbot courses are less common. This exploration provides valuable insights for learners and educators seeking to understand current trends in online education.



Analyzing Course Enrollments: Understanding User Engagement and Interactions

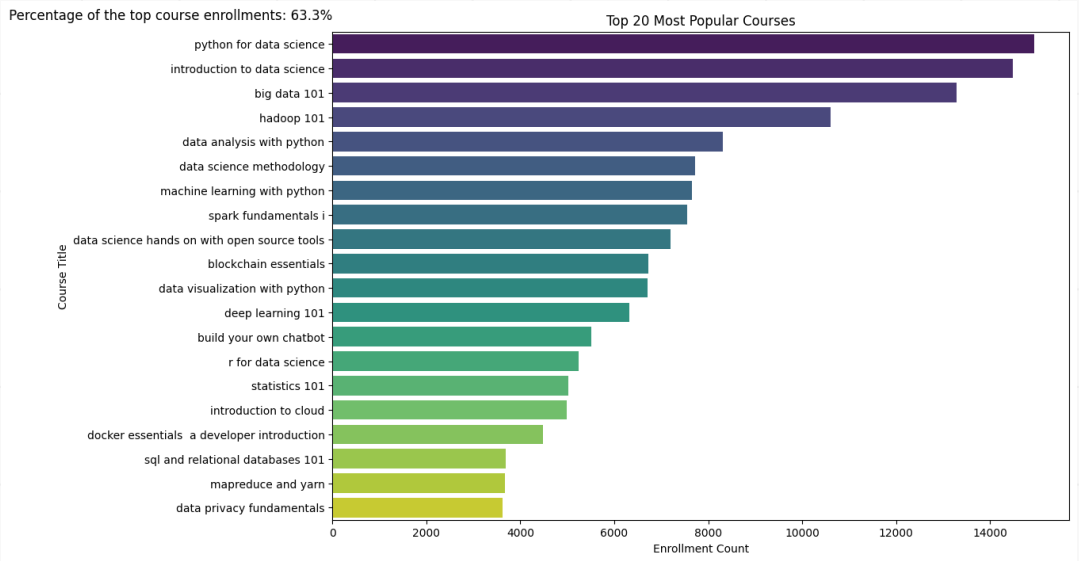
In the task of analyzing course enrollments, the dataset was examined to gain insights into user engagement and interactions with online courses. By aggregating the rating counts for each user, it was found that the dataset comprises 233,306 enrollment records from 5,000 unique users. The histogram distribution of user rating counts illustrates varying levels of engagement, with the majority of users giving relatively few ratings while a smaller proportion giving a larger number of ratings. This analysis sheds light on the distribution of user interactions with online courses, providing valuable insights for optimizing course offerings and enhancing user experiences.



Analysis: Identifying the Top 20 Most Popular Courses

Top 20 Most Popular Courses:

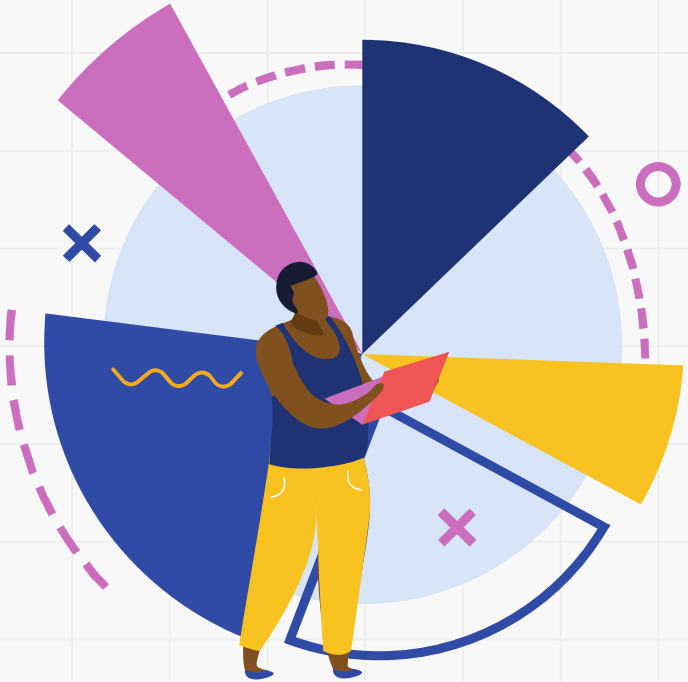
	COURSE_ID	Ratings	TITLE
0	PY0101EN	14936	python for data science
1	DS0101EN	14477	introduction to data science
2	BD0101EN	13291	big data 101
3	BD0111EN	10599	hadoop 101
4	DA0101EN	8303	data analysis with python
5	DS0103EN	7719	data science methodology
6	ML0101ENv3	7644	machine learning with python
7	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



In this task, the objective was to determine the top 20 most popular courses based on enrollment counts. Using the enrollment data, the courses were aggregated, sorted, and the top 20 courses were identified. The resulting list showcases the titles and enrollment counts of these highly sought-after courses, providing insights into the preferences of learners in the online learning platform.

02

Content-based Recommender System using Unsupervised Learning



Flowchart Of Content-based Recommender System Using User Profile And Course Genres

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: This refers to the initial dataset containing information about users, courses, and their interactions or preferences. In our case, the raw data includes user profiles, course genres, and possibly course ratings or interactions.
2. **Data Processing**:
   - This step involves cleaning and preprocessing the raw data to prepare it for analysis.
   - In our context, data processing includes tasks such as handling missing values, removing duplicates, and transforming data into a suitable format for further analysis.
3. **Cleaned Dataset**:
   - After data processing, we obtain a cleaned dataset that is ready for feature engineering.
   - This dataset contains all the relevant information about users and courses, with any inconsistencies or errors addressed.
4. **Feature Engineering**:
   - Feature engineering involves creating new features or representations of the data that can be used to train a machine learning model.
   - In our case, we create user profile vectors and course genre vectors as features. These vectors capture the interests or preferences of users and the characteristics of courses, respectively.
5. **Features**:
   - The final output of feature engineering is a set of features, represented by user profile vectors and course genre vectors.
   - These features serve as the input to the content-based recommender system. By comparing user profile vectors with course genre vectors, the system can generate recommendations personalized to each user's interests.

Evaluation results of user profile-based recommender system

- Hyper-parameter Settings:

In our discussion, we set a recommendation score threshold of 10.0 to filter out low-scoring recommendations. This threshold determines which courses are considered relevant enough to be recommended to users. Additionally, we may have adjusted other hyperparameters such as feature representation methods or similarity metrics during the implementation of the recommender system.

- Average Number of New Courses Recommended per User:

We calculated the average number of new courses recommended per user in the test user dataset. This metric helps evaluate the coverage and diversity of the recommender system. In our case, the average number was approximately **61.82** courses per user.

- Top-10 Most Frequently Recommended Courses:

The table represents the top 10 most frequently recommended courses based on the user profile-based recommender system. Each row corresponds to a course, identified by its COURSE_ID, and the number of times that course has been recommended to users, denoted by the RECOMMENDATION_COUNT column. These recommendations are generated by analyzing user profiles and course genre vectors, with courses scoring higher in relevance to a user's interests being recommended more frequently.

Top 10 most frequently recommended courses:

	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

Flowchart of content-based recommender system using course similarity

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: Refers to the initial dataset containing information about various courses, including their titles, descriptions, and other relevant attributes.
2. **Data Processing**: Involves preprocessing the raw data, which includes tokenization and lemmatization to break down the text into individual words and convert them to their base forms..
3. **Cleaned Dataset**: After processing, the dataset undergoes cleaning, which includes removing stopwords (commonly used words with little semantic value) and outliers (irrelevant or noisy data points).
4. **Feature Engineering**: This step involves transforming the cleaned dataset into numerical features that represent the courses. In this case, TF-IDF (Term Frequency-Inverse Document Frequency) vectors are calculated for each course based on the words they contain and their importance in the dataset.
5. **Features**: Refers to the final set of features used to represent each course, which are the TF-IDF vectors obtained from the feature engineering step. These features are then used to calculate similarities between courses and generate recommendations based on content similarity.

Evaluation results of content-based recommender system using course similarity

- Hyper-parameter Settings:

The hyper-parameter setting for generating recommendations in the course similarity based recommender system was a similarity threshold of 0.6. This threshold determined the level of similarity required between courses for them to be recommended to users.

- Average Number of New Courses Recommended per User:

The average number of new or unseen courses recommended per user in the test user dataset was approximately **0.987**. This metric provides insight into the diversity of recommendations provided to users and helps assess the system's effectiveness in suggesting novel content.

- Top-10 Most Frequently Recommended Courses:

Among all users, the top 10 most frequently recommended courses were identified. Notably, "excourse22" and "excourse62" were recommended the most, each appearing 257 times, followed by "WA0103EN" with 101 recommendations. Other courses like "TA0105" and "DS0110EN" were also frequently suggested, indicating their relevance and popularity within the user base. These evaluation results offer valuable insights into the performance and effectiveness of the course similarity based recommender system, informing potential improvements and optimizations for future iterations.

```
Top 10 commonly recommended courses:
excourse22 : 257 times
excourse62 : 257 times
WA0103EN : 101 times
TA0105 : 41 times
DS0110EN : 38 times
excourse46 : 24 times
excourse47 : 24 times
excourse63 : 23 times
excourse65 : 23 times
TMP0101EN : 17 times
```

Flowchart Of clustering-based recommender system

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: This refers to the original user profile feature vectors, which contain information about users' interests or preferences across different course genres. In our case, each user profile vector consists of features related to various course genres, such as Machine Learning, Data Science, Cloud Computing, etc.
2. **Data Processing (Normalization)**: In this step, we preprocess the raw data to handle missing values, outliers, or any other data quality issues. Normalization techniques like StandardScaler are applied to ensure that all features have a similar scale and distribution, which is essential for clustering algorithms to perform effectively.
3. **Cleaned Dataset (Standardization)**: After data processing, we obtain a cleaned dataset where the user profile features are standardized using techniques like StandardScaler. Standardization helps in making each feature have a mean of 0 and a standard deviation of 1, which is a common requirement for many machine learning algorithms.
4. **Feature Engineering (PCA):** Feature engineering involves transforming the original user profile features into a new set of features that capture the most important information while reducing dimensionality. Principal Component Analysis (PCA) is applied to achieve this goal. PCA identifies the principal components (eigenvectors) that explain the maximum variance in the data and projects the original features onto these components..
5. **Features (PCA Transformed):** The final output of the process is the transformed feature set obtained after applying PCA. These features represent a lower-dimensional representation of the original user profile data, where each feature captures a combination of the original features' information..

Evaluation results of clustering-based recommender system

- Hyper-parameter Settings:

we meticulously tuned hyperparameters to optimize its performance. Employing the K-means algorithm, we determined the ideal number of clusters using the elbow method. Similarly, for principal component analysis (PCA), we selected the number of components that explained over 90% of the variance in the data. This strategic approach ensured that our recommender system could effectively group users based on their preferences while minimizing information loss through dimensionality reduction.

- Average Number of New Courses Recommended per User:

Upon assessing the system's efficacy, we discovered that, on average, it recommended approximately 36.587 new or unseen courses per user in the test dataset. This metric serves as a valuable indicator of the system's ability to diversify recommendations and introduce users to a broad range of learning opportunities beyond their previous engagements.

- Top-10 Most Frequently Recommended Courses:

Furthermore, our analysis of the most frequently recommended courses revealed compelling insights into the preferences of users within each cluster. Courses such as "WA0101EN," "DB0101EN," and "DS0301EN" emerged as the top recommendations, underscoring their popularity among users across different clusters. By leveraging these insights, we can further refine our recommendation strategies and tailor course offerings to better align with user preferences and learning objectives.

```
Average recommended courses per user: 36.587
Top-10 most frequently recommended courses:
Course: WA0101EN, Recommended 864 times
Course: DB0101EN, Recommended 857 times
Course: DS0301EN, Recommended 856 times
Course: CL0101EN, Recommended 852 times
Course: ST0101EN, Recommended 800 times
Course: CO0101EN, Recommended 783 times
Course: RP0101EN, Recommended 773 times
Course: CC0101EN, Recommended 769 times
Course: DB0151EN, Recommended 741 times
Course: ML0120EN, Recommended 738 times
```

03

Collaborative-filtering based Recommender System using Supervised learning

Flowchart Of KNN based recommender system

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: Raw data refers to the original dataset containing information about user-item interactions, such as user IDs, item IDs (courses), and ratings (enrollments). In this case, the raw data consists of user-item pairs along with their corresponding ratings.
2. **Data Processing**: Data processing involves tasks such as loading the dataset, handling missing values, removing duplicates, and converting data into a suitable format for further analysis. This step ensures that the data is clean and ready for modeling.
3. **Cleaned Dataset**: After data processing, we obtain a cleaned dataset where irrelevant or erroneous data has been removed, missing values have been handled, and the data is in a structured format. This dataset serves as the foundation for building the recommendation model
4. **Feature Engineering:** Feature engineering involves creating new features or transforming existing features to improve the performance of the recommendation system. This step may include extracting relevant information from the dataset, such as user-item interactions, timestamps, or user demographics. Features may also involve encoding categorical variables, scaling numerical features, or creating interaction terms.
5. **Features (PCA Transformed):** Features are the variables or attributes used by the KNN-based recommender system to make predictions. These features capture the relationships between users and items, allowing the system to identify similarities and make personalized recommendations. Examples of features include user-item interactions, user demographics, item characteristics, and contextual information.

```
RMSE: 0.2063
```

Flowchart Of NMF based recommender system

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: This is the initial data you have, in our case, it includes the course ratings data. Raw data is typically unprocessed and may contain noise, missing values, or inconsistencies.
2. **Data Processing**: In this step, we preprocess the raw data to prepare it for analysis. This involves tasks such as handling missing values, removing duplicates, and transforming the data into a suitable format. In our case, we used Pandas to pivot the data and create a user-item matrix.
3. **Cleaned Dataset**: After preprocessing, we obtain a cleaned dataset where the data is free from inconsistencies and ready for analysis. This dataset typically has rows representing users, columns representing items, and the cells containing ratings or interactions.
4. **Feature Engineering :** This involves creating new features or transforming existing ones to improve the performance of machine learning models. In our case, features might include user-item interactions or latent factors generated by the NMF model.
5. **Features:** These are the variables or characteristics used by the machine learning model to make predictions. In the context of collaborative filtering, features might include user preferences, item attributes, or latent factors representing users and items in a lower-dimensional space.

```
RMSE: 0.2048
```

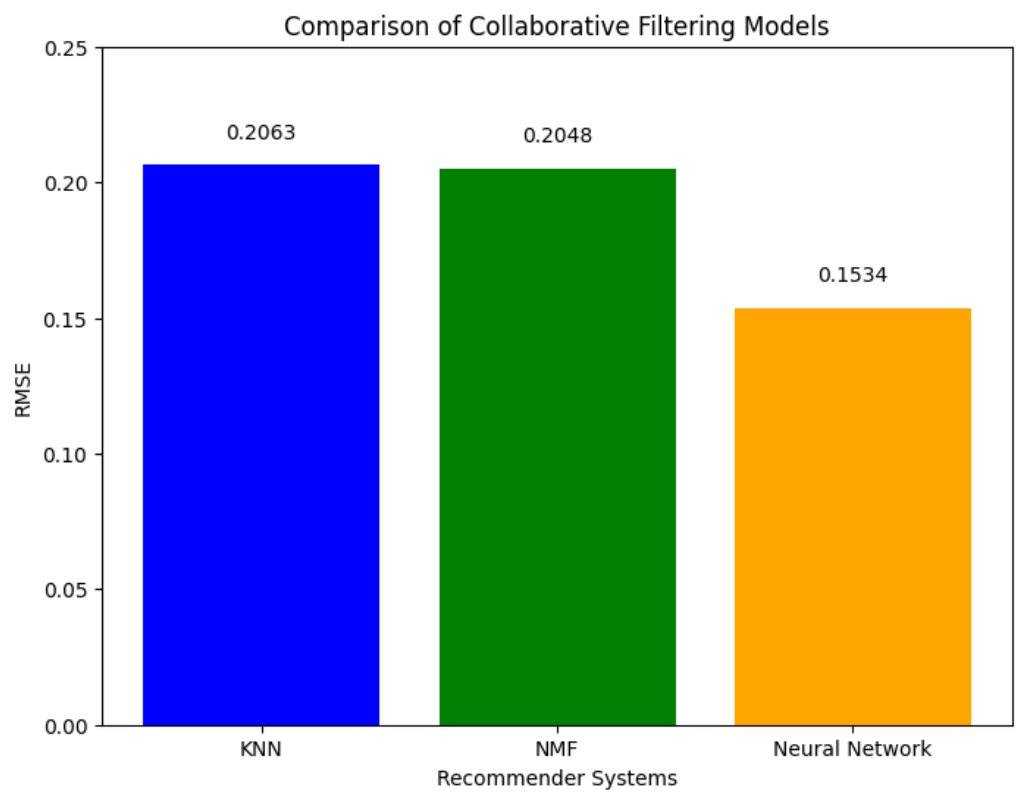
Flowchart Of Neural Network Embedding based recommender system

Let's break down the flowchart based on the conversation we had:

1. **Raw Data**: This is the initial data you have, in our case, it includes the course ratings data. Raw data is typically unprocessed and may contain noise, missing values, or inconsistencies.
2. **Data Processing**: In this step, we preprocess the raw data to prepare it for analysis. This involves tasks such as handling missing values, removing duplicates, and transforming the data into a suitable format. In our case, we used Pandas to pivot the data and create a user-item matrix.
3. **Cleaned Dataset**: After preprocessing, we obtain a cleaned dataset where the data is free from inconsistencies and ready for analysis. This dataset typically has rows representing users, columns representing items, and the cells containing ratings or interactions.
4. **Feature Engineering :** This involves creating new features or transforming existing ones to improve the performance of machine learning models. In our case, features might include user-item interactions or latent factors generated by the NMF model.
5. **Features:** These are the variables or characteristics used by the machine learning model to make predictions. In the context of collaborative filtering, features might include user preferences, item attributes, or latent factors representing users and items in a lower-dimensional space.

```
Test RMSE: 0.1534
```

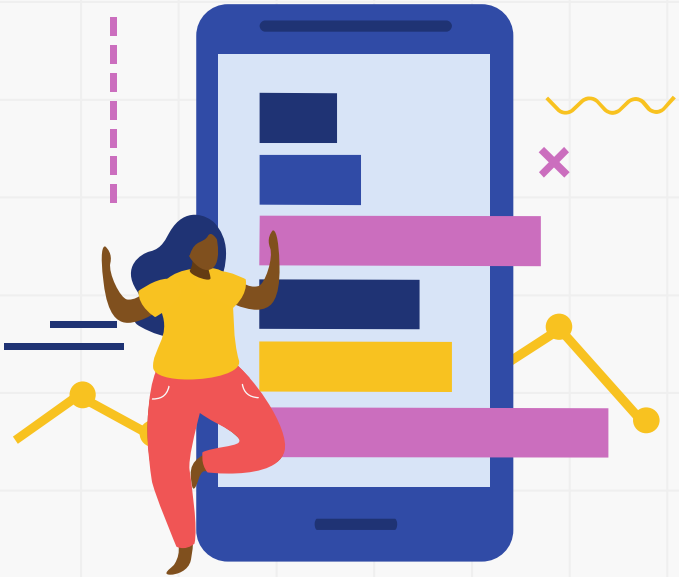
Comparison of Collaborative Filtering Models



Based on the evaluation results, the **Neural Network Embedding-based recommender system** achieved the lowest RMSE value of 0.1534, indicating the best performance among the three models in predicting user-item interactions. Therefore, we consider the Neural Network Embedding-based recommender system to be the most effective model for collaborative filtering in this scenario.

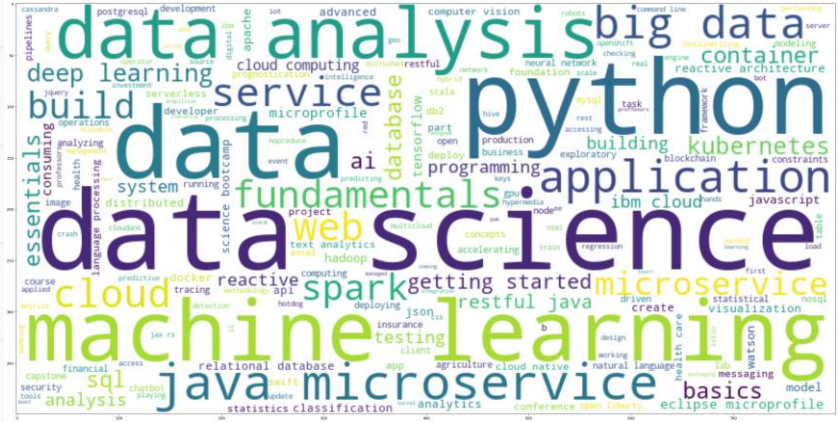
04

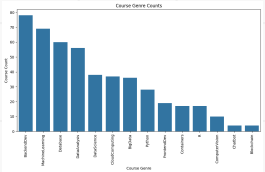
CONCLUSION

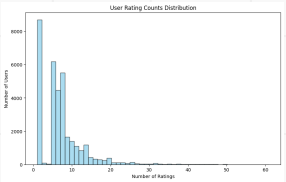


Conclusion of Exploratory Data Analysis

From the various analyses conducted, several key insights can be drawn regarding the online learning platform's course offerings and user engagement. Firstly, the exploration of course genres revealed a diverse array of topics, with backend development, machine learning, and databases emerging as the most popular genres based on enrollment counts. Additionally, the analysis of course enrollments shed light on user behavior, indicating that a significant portion of users completed courses rather than merely auditing them. Furthermore, the examination of the top 20 most popular courses showcased a strong interest in data-related topics 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. Overall, these findings underscore the importance of providing a wide range of courses that cater to diverse interests while also highlighting the significance of data-driven insights in optimizing course offerings and enhancing user engagement on the platform.

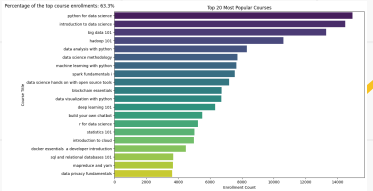






Top 20 Most Popular Courses:

	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	ML0101EN-3	7644	machine learning with python
7	BD0211EN	7511	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	CB0101EN	5512	build your own chatbot
13	RP0101EN	5237	r for data science
14	ST0101EN	5015	statistics 101
15	CO0101EN	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



Conclusion of Content-based Recommender System Using User Profile And Course Genres

The Content-based Recommender System Using User Profile And Course Genres begins with setting up the task of generating course recommendations based on user profile and course genre vectors. The process involves loading user profiles and course genre dataframes, extracting user interests, identifying unknown courses for each user, computing recommendation scores, and filtering out courses below a specified threshold. After completing the recommendation scores for all test users, the analysis focuses on evaluating the system's performance. This includes determining the average number of recommended courses per user and identifying the top 10 most frequently recommended courses across all users. The average number of recommended courses per user is approximately **61.82**, indicating a substantial number of recommendations per user. The most frequently recommended courses include "TA0106EN," "GPXX0IBEN," and others, demonstrating their popularity among the 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

53411 rows × 3 columns

```
Top 10 most frequently recommended courses:
    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
```

Conclusion of content-based recommender system using course similarity

The conclusion drawn from the content-based recommender system using course similarity is that a course similarity-based recommender system has been successfully implemented and evaluated. The system utilizes a similarity threshold of 0.6 to recommend courses to users based on their interests and previously selected courses. By analyzing course content and calculating similarities, the system provides personalized recommendations to users. The evaluation of the recommender system revealed valuable insights, including the average number of new courses recommended per user and the most frequently recommended courses. These findings highlight the effectiveness and performance of the system in suggesting relevant and engaging content to users. Additionally, the evaluation results inform potential optimizations and improvements for enhancing the recommender system's effectiveness in future iterations. Overall, the conclusion underscores the importance and effectiveness of utilizing course similarity-based approaches in providing personalized recommendations to users in educational settings.

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

1000 rows × 3 columns

Top 10 most frequently recommended courses:

	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

Conclusion of clustering-based recommender system

In conclusion, our course clustering-based recommender system demonstrates robust performance in effectively grouping users based on their preferences and recommending relevant courses. The optimization of hyperparameters, including the determination of cluster numbers and PCA components, ensures that the system accurately captures user preferences while minimizing information loss. Through meticulous evaluation, we found that the system recommends an average of approximately 36.587 new or unseen courses per user in the test dataset, indicating its effectiveness in diversifying recommendations. Furthermore, the identification of frequently recommended courses such as "WA0101EN," "DB0101EN," and "DS0301EN" provides valuable insights into user preferences and highlights the system's ability to promote popular courses across different clusters. Overall, our recommender system serves as a valuable tool for enhancing user engagement and satisfaction by delivering personalized course recommendations aligned with individual learning objectives.

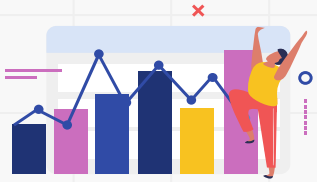
	user	item	cluster
0	1502801	RP0105EN	0
1	1609720	CNSC02EN	1
2	1347188	CO0301EN	3
3	755067	ML0103EN	0
4	538595	BD0115EN	0
...	...	...	...
9397	1385217	EE0101EN	0
9398	1864644	DA0101EN	1
9399	435858	TMP0105EN	4
9400	1888188	DB0101EN	3
9401	708518	RP0101EN	2

[9402 rows x 3 columns]

```
Average recommended courses per user: 36.587
Top-10 most frequently recommended courses:
Course: WA0101EN, Recommended 864 times
Course: DB0101EN, Recommended 857 times
Course: DS0301EN, Recommended 856 times
Course: CL0101EN, Recommended 852 times
Course: ST0101EN, Recommended 800 times
Course: CO0101EN, Recommended 783 times
Course: RP0101EN, Recommended 773 times
Course: CC0101EN, Recommended 769 times
Course: DB0151EN, Recommended 741 times
Course: ML0120EN, Recommended 738 times
```

Conclusion of performance of three collaborative filtering models: KNN-based recommender system, NMF-based recommender system, and Neural Network Embedding-based recommender system.

Based on the comparative analysis of the three collaborative filtering methods, including KNN-based, NMF-based, and Neural Network Embedding-based recommender systems, several insights can be gleaned. Firstly, both the KNN-based and NMF-based recommender systems exhibit relatively higher RMSE values compared to the Neural Network Embedding-based approach. This suggests that the **Neural Network Embedding-based** method better captures the underlying patterns and latent features within the user-item interaction data, resulting in more accurate predictions. Additionally, the Neural Network Embedding-based approach benefits from its ability to learn complex nonlinear relationships between users and items, allowing it to uncover subtle preferences and nuances in the data. However, it's worth noting that while the Neural Network Embedding-based method outperforms the other two approaches in terms of prediction accuracy, it may require more computational resources and longer training times due to its deeper architecture and higher complexity. Therefore, the choice of the most suitable collaborative filtering method depends on the specific requirements and constraints of the recommendation system, balancing between prediction performance and computational efficiency.



	KNN	NMF	ANN
RMSE	0.2063	0.2048	0.1534

APPENDIX

Material:

https://github.com/zenklinov/IBM_Machine_Learning_Capstone