IBM

Machine Learning Capstone

Massimiliano Marchesiello

09 September 2024

IBM

Build a Personalised Online Course Recommender System with Machine Learning

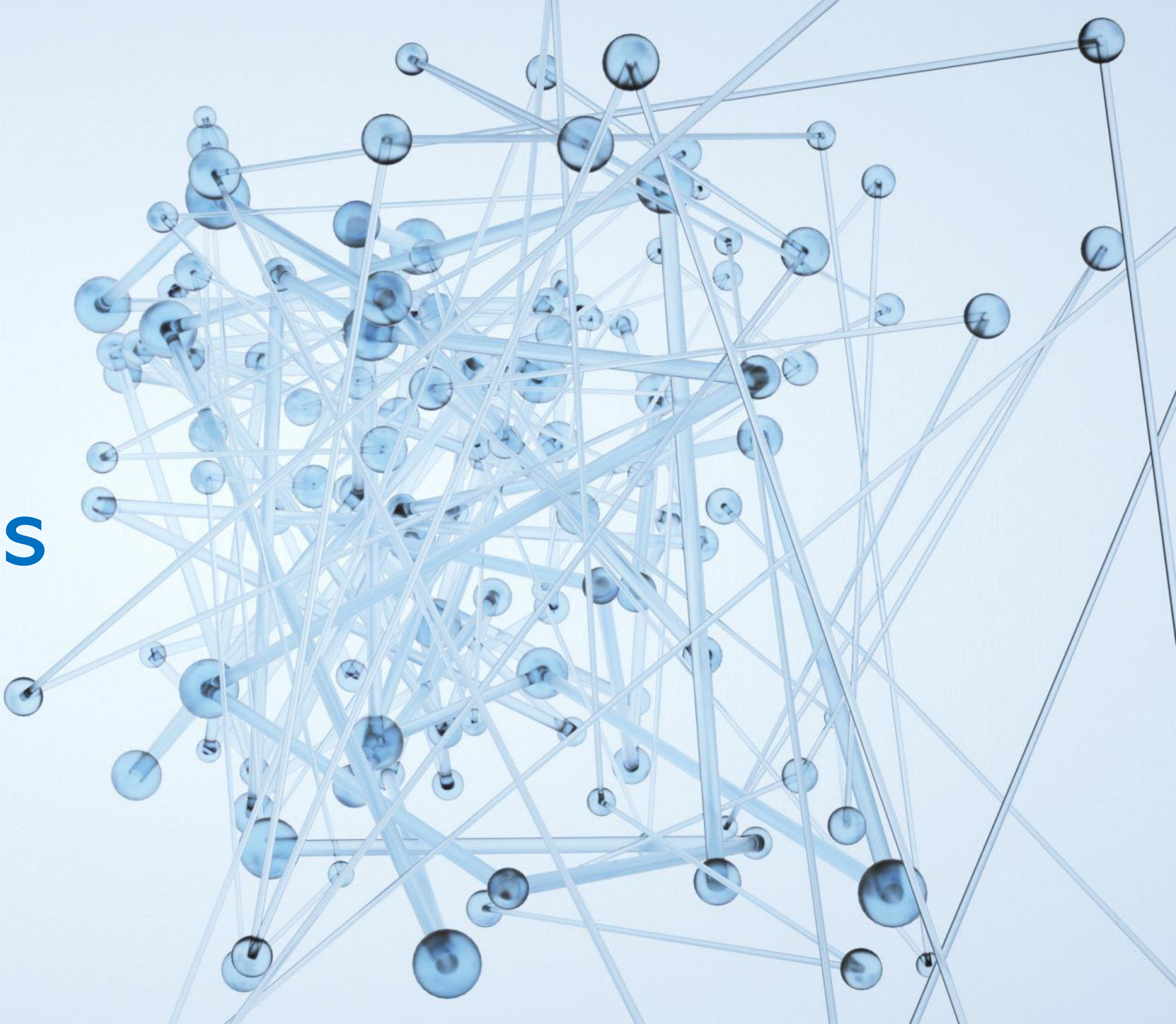
Table of Contents

1. Exploratory Data Analysis
2. Content-based Recommender System using Unsupervised Learning
3. Collaborative-filtering Recommender System using Supervised Learning
4. Collaborative-filtering Recommender System using Supervised Learning

IBM

Exploratory Data Analysis

1

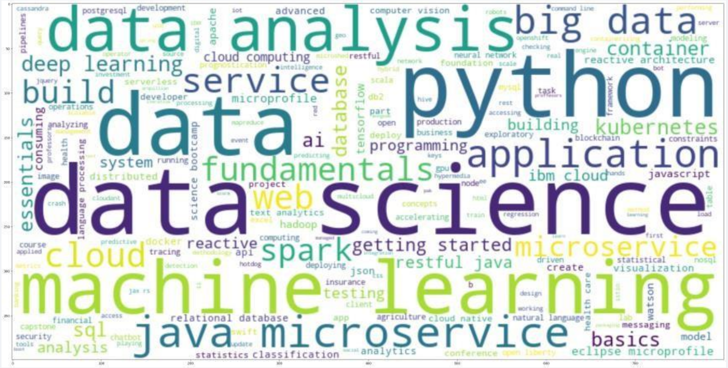


Exploring Online Course Titles: Unveiling Popular IT Skills through WordCloud Analysis

Through this exploratory data analysis, a WordCloud visualization is employed to reveal the most common keywords from a dataset of online course titles.

By systematically grouping and filtering these titles, key topics such as Python, data science, machine learning, big data, AI, TensorFlow, containers, and cloud computing emerge as central themes.

This analysis offers valuable insights into the current trends in IT skills and digital learning areas, helping both learners and educators better navigate the evolving educational landscape and the opportunities it presents.



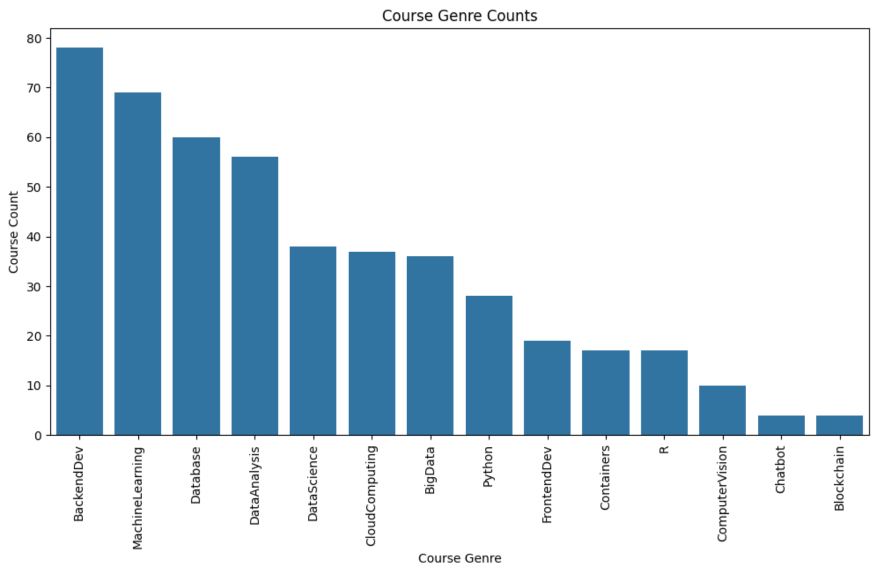
Analysing Online Course Genres: Unveiling Popular Topics and Trends

During the analysis of course categories, the dataset was explored to determine the popularity of various online learning topics.

By calculating the number of courses within each category and presenting the findings through both bar charts and tables, clear trends in the most in-demand subjects emerged.

The analysis highlighted that Backend Development, Machine Learning, and Database courses are among the most popular, whereas Blockchain and Chatbot courses appear less frequently.

This investigation offers valuable insights for both learners and educators, helping them grasp the current trends in online education.



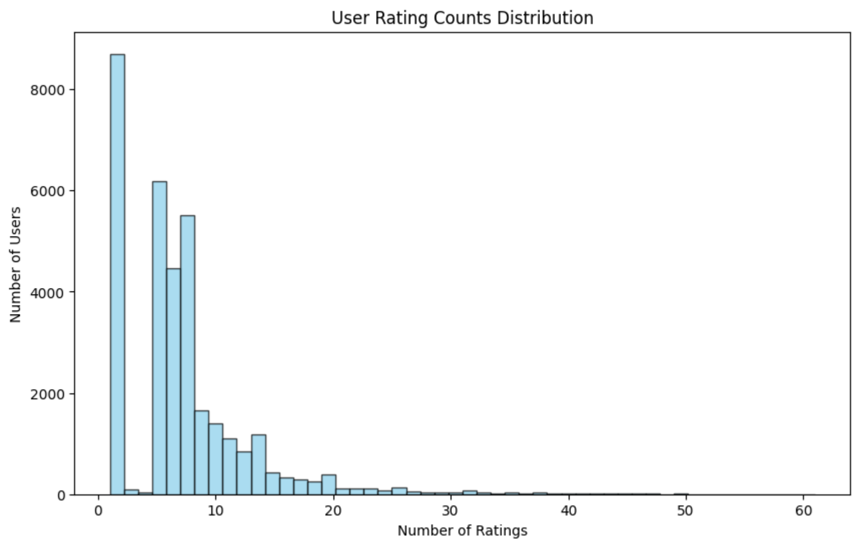
Analysing Course Enrollments: Understanding User Engagement and Interactions

In analysing course enrollments, the dataset was reviewed to uncover patterns in user engagement and interactions with online courses.

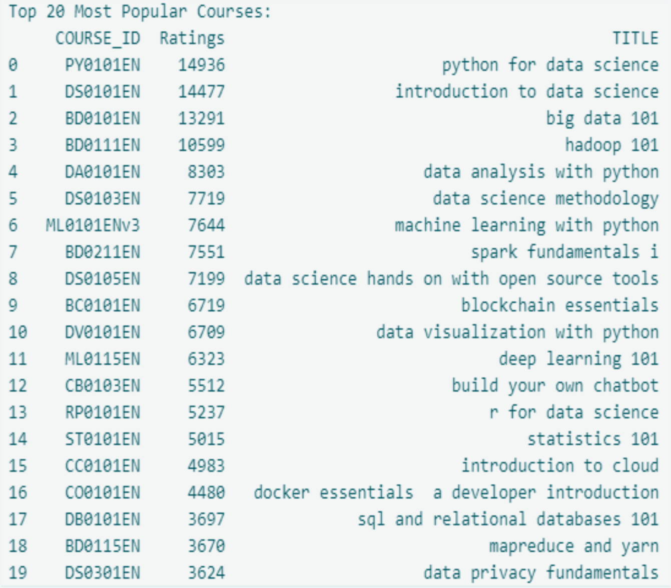
By aggregating the number of ratings submitted by each user, it was observed that the dataset includes 233,306 enrollment records from 5,000 unique users.

A histogram of user rating counts reveals diverse engagement levels, with most users providing relatively few ratings and a smaller group contributing significantly more.

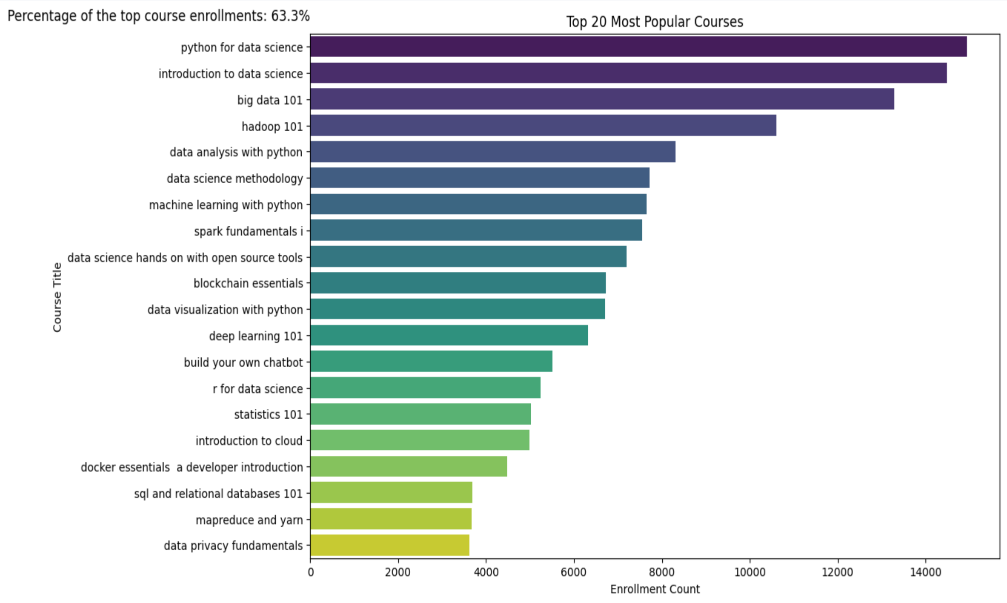
This analysis highlights the distribution of user interactions with online courses, offering valuable insights to enhance course offerings and improve the overall learning experience.



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
```

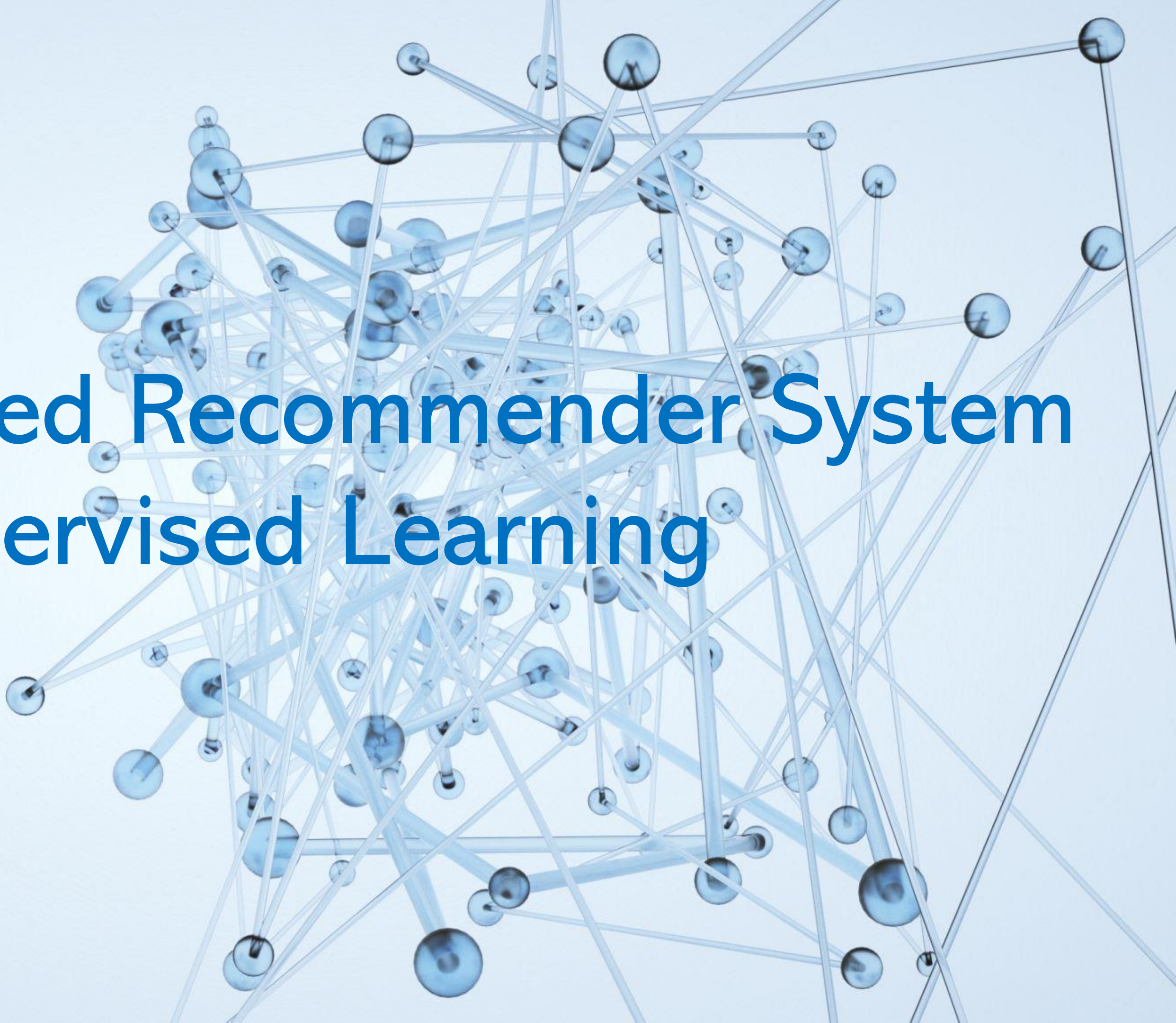


Percentage of the top course enrollments: 63.3%

The goal of this task was to identify the 20 most popular courses based on enrollment numbers.

By analysing and sorting the enrollment data, the top 20 courses were determined.

The final list highlights the course titles along with their respective enrollment figures, offering valuable insights into learner preferences and the most in-demand courses on the online learning platform.

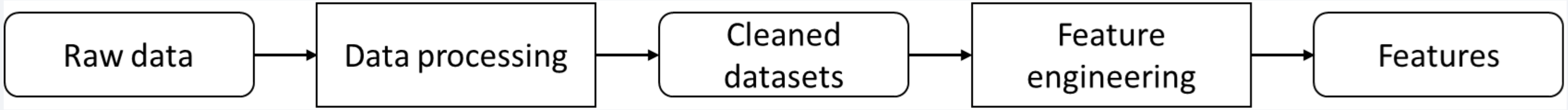


IBM

Content-based Recommender System using Unsupervised Learning

2

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



Let's break down the flowchart based on our previous discussion:

1. Raw Data:

This refers to the original dataset that contains information about users, courses, and their interactions or preferences. In this case, the raw data includes all the necessary elements for analysis.

2. Data Processing:

This step focuses on cleaning and preparing the raw data for further analysis. Key tasks during this stage include handling missing values, eliminating duplicates, and transforming the data into a usable format for subsequent steps.

3. Cleaned Dataset:

After completing the data processing, we obtain a refined dataset that is free from errors and inconsistencies, making it suitable for feature engineering.

4. Feature Engineering:

This process involves generating new features or representations from the data that can be utilised to train a machine-learning model. In this context, we create user profile vectors and course genre vectors as features. These vectors represent the preferences or interests of users, as well as the characteristics of different courses.

5. Features:

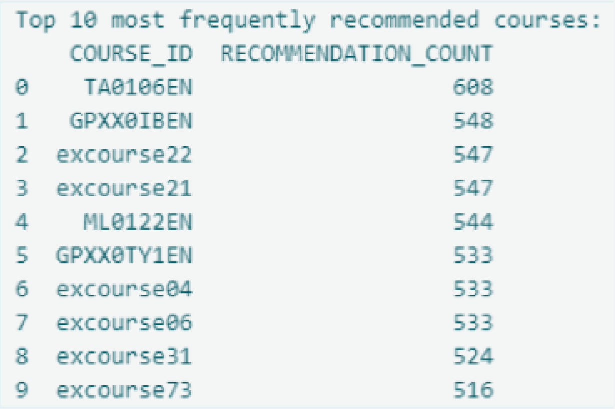
The final output of feature engineering is a set of features, which include user profile vectors and course genre vectors. These serve as inputs to the content-based recommender system. By comparing the user profile vectors with course genre vectors, the system can provide personalised recommendations based on each user’s specific interests.

Evaluation results of user profile-based recommender system

❑ Hyperparameter Settings:
In our analysis we established a recommendation score threshold of 10.0 to exclude low-scoring recommendations. This threshold determines which courses are deemed sufficiently relevant to be suggested to users. Additionally, we may have fine-tuned other hyperparameters, such as feature representation methods or similarity metrics, during the design of the recommender system.

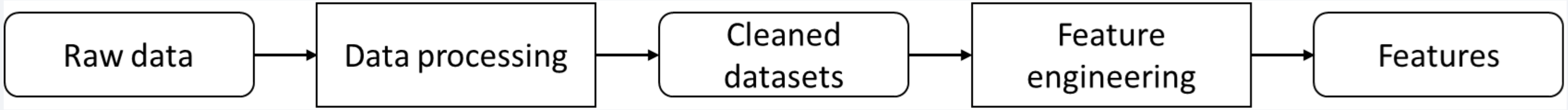
❑ Average Number of New Courses Recommended per User:
We computed the average number of new course recommendations per user in the test dataset. This measure helps assess both the coverage and diversity of the recommender system. In our case, the average stood at approximately 61.82 courses per user.

❑ Top 10 Most Frequently Recommended Courses:
The table lists the top 10 courses most frequently recommended by the user profile-based recommender system. Each entry corresponds to a course, identified by its COURSE_ID, and the number of times it was recommended, represented in the RECOMMENDATION_COUNT column. These recommendations are generated by analysing user-profiles and course genre vectors, with the courses most aligned with users' 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 a content-based recommender system using course similarity



Let's break down the flowchart based on our discussion:

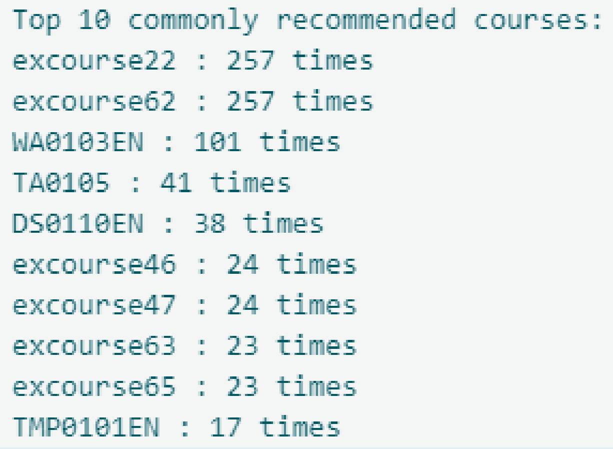
1. Raw Data:
This represents the initial dataset, containing details about various courses, such as their titles, descriptions, and other essential attributes.
2. Data Processing:
This step focuses on preprocessing the raw data, which involves tasks like tokenisation and lemmatisation. These processes break down the text into individual words and convert them to their root forms.
3. Cleaned Dataset:
Once the data is processed, the dataset undergoes cleaning, which involves removing stopwords (commonly used words with little semantic importance) and filtering out outliers (irrelevant or noisy data points).
4. Feature Engineering:
At this stage, the cleaned dataset is transformed into numerical features that represent the courses. Specifically, Term Frequency-Inverse Document Frequency (TF-IDF) vectors are generated for each course based on the words in their descriptions and their relative importance in the dataset.
5. Features:
This refers to the final set of features, represented by the TF-IDF vectors created during the feature engineering step. These vectors are used to measure the similarities between courses and to generate recommendations based on content similarity.

Evaluation results of a 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.

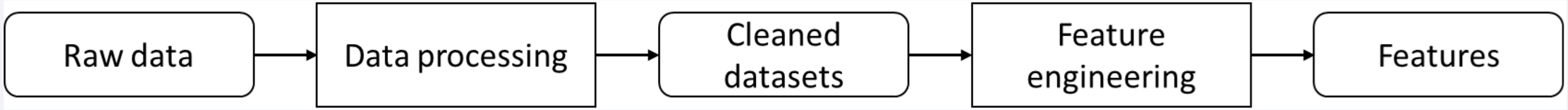
❑ Average Number of New Courses Recommended per User:
The average number of new or unseen courses recommended per user in the test user dataset was approximately 0.987. This metric provides insight into the diversity of recommendations provided to users and helps assess the system's effectiveness in 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 optimisations 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 a content-based recommender system using course similarity



Let's break down the flowchart based on our discussion:

In our case, each user profile vector consists of features that correspond to various course categories, such as Machine Learning, Data Science, and Cloud Computing.

1. Raw Data:

This refers to the original user profile feature vectors, which include details about users' preferences and interests across different course genres.

2. Data Processing (Normalization):

At this stage, the raw data is preprocessed to handle missing values, outliers, and other quality issues. Normalization techniques, such as StandardScaler, are applied to ensure that all features have a consistent scale and distribution, which is critical for the effective performance of clustering algorithms and other machine learning models.

3. Cleaned Dataset (Standardization):

After processing, the dataset is standardized using methods like StandardScaler. This ensures that each feature has a mean of 0 and a standard deviation of 1—standard practice for many machine learning models.

4. Feature Engineering (PCA):

In this step, feature engineering is used to reduce dimensionality. Principal Component Analysis (PCA) is applied to transform the original user profile features into a set of key components (eigenvectors) that capture the most important variations in the data. The original features are projected onto these new components.

5. Features (PCA Transformed):

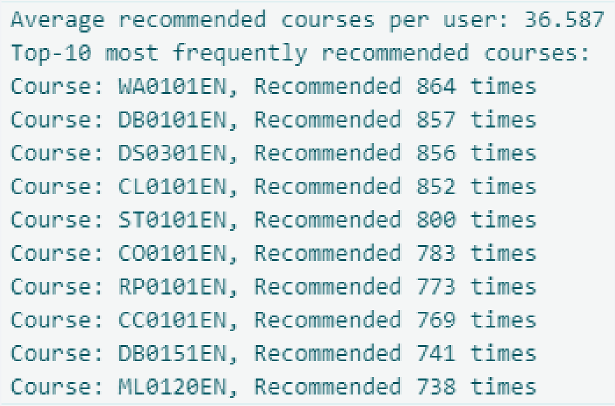
The final output consists of the transformed features obtained through PCA. These represent a lower-dimensional version of the original user profile data, where each feature combines information from the original variables into a more compact form.

Evaluation results of clustering-based recommender system

❑ Hyperparameter Settings:
We carefully fine-tuned the hyperparameters to enhance system performance. Using the K-means algorithm, we identified the optimal number of clusters by applying the elbow method. For Principal Component Analysis (PCA), we selected components that accounted for over 90% of the variance in the data. This method ensured that our recommender system could effectively group users based on their preferences while minimising information loss through dimensionality reduction.

❑ Average Number of New Courses Recommended per User:
When evaluating the system's performance, we found that it recommended an average of 36.587 new or previously unseen courses per user in the test dataset. This metric highlights the system’s ability to provide diverse recommendations and offer users a wide array of learning opportunities beyond their prior experiences.

❑ Top 10 Most Frequently Recommended Courses:
Our analysis of the most frequently recommended courses provided valuable insights into user preferences within each cluster. Courses like "WA0101EN," "DB0101EN," and "DS0301EN" emerged as the top recommendations, reflecting their widespread popularity among users across various clusters. These findings will help us refine our recommendation strategies and better tailor course offerings to align with user preferences and learning objectives.

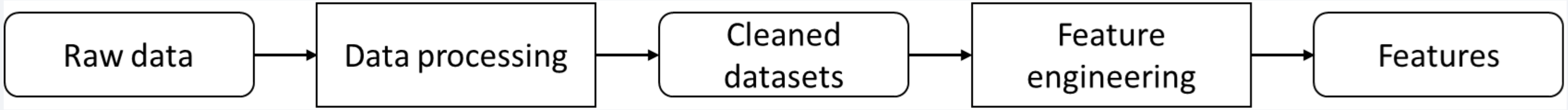


```
Average recommended courses per user: 36.587
Top-10 most frequently recommended courses:
Course: WA0101EN, Recommended 864 times
Course: DB0101EN, Recommended 857 times
Course: DS0301EN, Recommended 856 times
Course: CL0101EN, Recommended 852 times
Course: ST0101EN, Recommended 800 times
Course: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

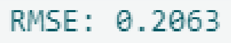
Collaborative-filtering Recommender System using Supervised Learning

3

Flowchart of KNN-based recommender system

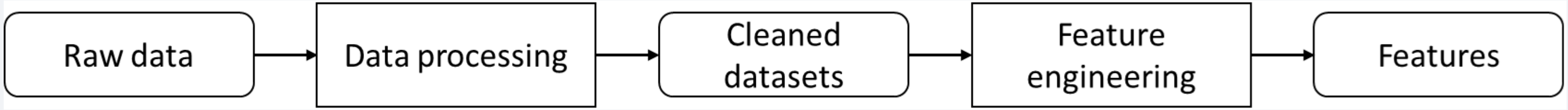


1. Raw Data:
The raw data refers to the initial dataset that captures user-item interactions, such as user IDs, course IDs, and ratings or feedback (e.g., enrollments). This dataset forms the basis for further analysis and recommendation modelling.
2. Data Processing:
Data processing involves essential tasks such as loading the dataset, addressing missing values, removing duplicates, and converting the data into a structured format. This step ensures the dataset is clean and prepped for analysis, making it suitable for building a recommendation model.
3. Cleaned Dataset:
After data processing, the result is a refined dataset where irrelevant or erroneous data has been removed, and missing values have been addressed. The cleaned dataset is now in a structured, consistent format, ready to serve as the foundation for the recommendation system.
4. Feature Engineering:
Feature engineering entails the creation of new features or the transformation of existing ones to improve the model's predictive power. This process may include extracting key information like user-item interactions, timestamps, or user demographics. It can also involve tasks such as encoding categorical variables, scaling numerical features, and creating new interaction terms that capture relationships between users and courses.
5. Features (PCA Transformed):
These are the variables used by the KNN-based recommender system to make predictions. Features capture relationships between users and courses, enabling the system to identify patterns and provide personalised recommendations. Examples include user demographics, item characteristics, and historical user-item interactions. Additionally, Principal Component Analysis (PCA) may be applied to reduce the dimensionality of the features, retaining the most important components and improving model performance.



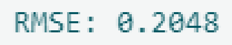
```
RMSE: 0.2063
```

Flowchart of NMF based recommender system



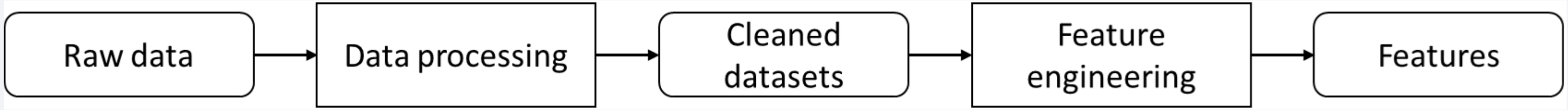
Raw data → Data processing → Cleaned datasets → Feature engineering → Features

1. Raw Data:
This refers to the unprocessed data, which, in our case, consists of course ratings. Raw data often contains noise, missing values, or inconsistencies and requires processing before it can be used for analysis.
2. Data Processing:
In this stage, we clean and prepare the raw data for further analysis. Tasks include handling missing values, removing duplicates, and reshaping the data into a more usable format. In our example, we used Pandas to pivot the dataset, transforming it into a user-item matrix where each row corresponds to a user and each column to a course.
3. Cleaned Dataset:
Once data processing is complete, we obtain a cleaned dataset, free of inconsistencies and ready for analysis. This dataset typically represents users as rows and items (courses) as columns, with the corresponding ratings or interactions populating the cells.
4. Feature Engineering:
Feature engineering involves creating or transforming features to enhance the performance of machine learning models. In this context, we might generate features such as user-item interaction histories or latent factors extracted by models like NMF (Non-negative Matrix Factorization) to represent underlying patterns in user behaviour.
5. Features:
Features are the variables used by machine learning models to make predictions. In collaborative filtering, these can include user preferences, course attributes, or latent factors that represent users and items in a reduced-dimensional space. These features help the model understand similarities between users and courses, improving the accuracy of recommendations.



```
RMSE: 0.2048
```

Flowchart Of Neural Network Embedding-based recommender system



Raw data

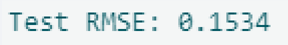
Data processing

Cleaned datasets

Feature engineering

Features

1. Raw Data:
Raw data refers to the initial, unprocessed dataset—in our case, this includes the course ratings data. This raw form of data typically contains noise, missing values, or inconsistencies that need to be addressed before analysis.
2. Data Processing:
In this phase, we clean and structure the raw data to make it suitable for analysis. Key tasks include managing missing values, removing duplicates, and reshaping the dataset as needed. For instance, we used Pandas to pivot the data into a user-item matrix, where users are mapped to the courses they have rated.
3. Cleaned Dataset:
After the data processing step, we obtain a refined dataset that is free of inconsistencies and ready for analysis. Typically, this dataset is organised with rows representing users, columns representing items (courses), and cells populated with ratings or other interaction data.
4. Feature Engineering:
Feature engineering focuses on creating or transforming features to improve the performance of machine learning models. In the context of a neural network embedding-based system, the model automatically learns user-item interaction embeddings during training. These embeddings capture latent patterns in user preferences and item characteristics without the need for traditional factorization techniques like NMF. The learned embeddings allow the model to identify similarities between users and courses, improving the accuracy of personalized recommendations.
5. Features:
Features are the attributes or variables that the machine learning model uses to make predictions. In collaborative filtering, these can include user preferences, course characteristics, or latent factors that represent users and items in a lower-dimensional space, allowing the model to better understand similarities and make more accurate recommendations.

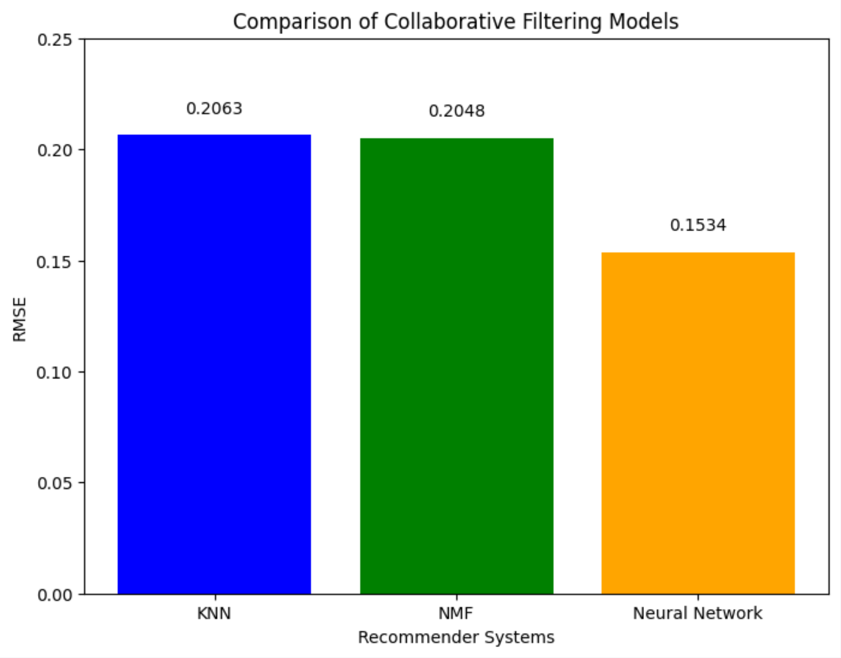


```
Test RMSE: 0.1534
```

Comparison of Collaborative Filtering Models

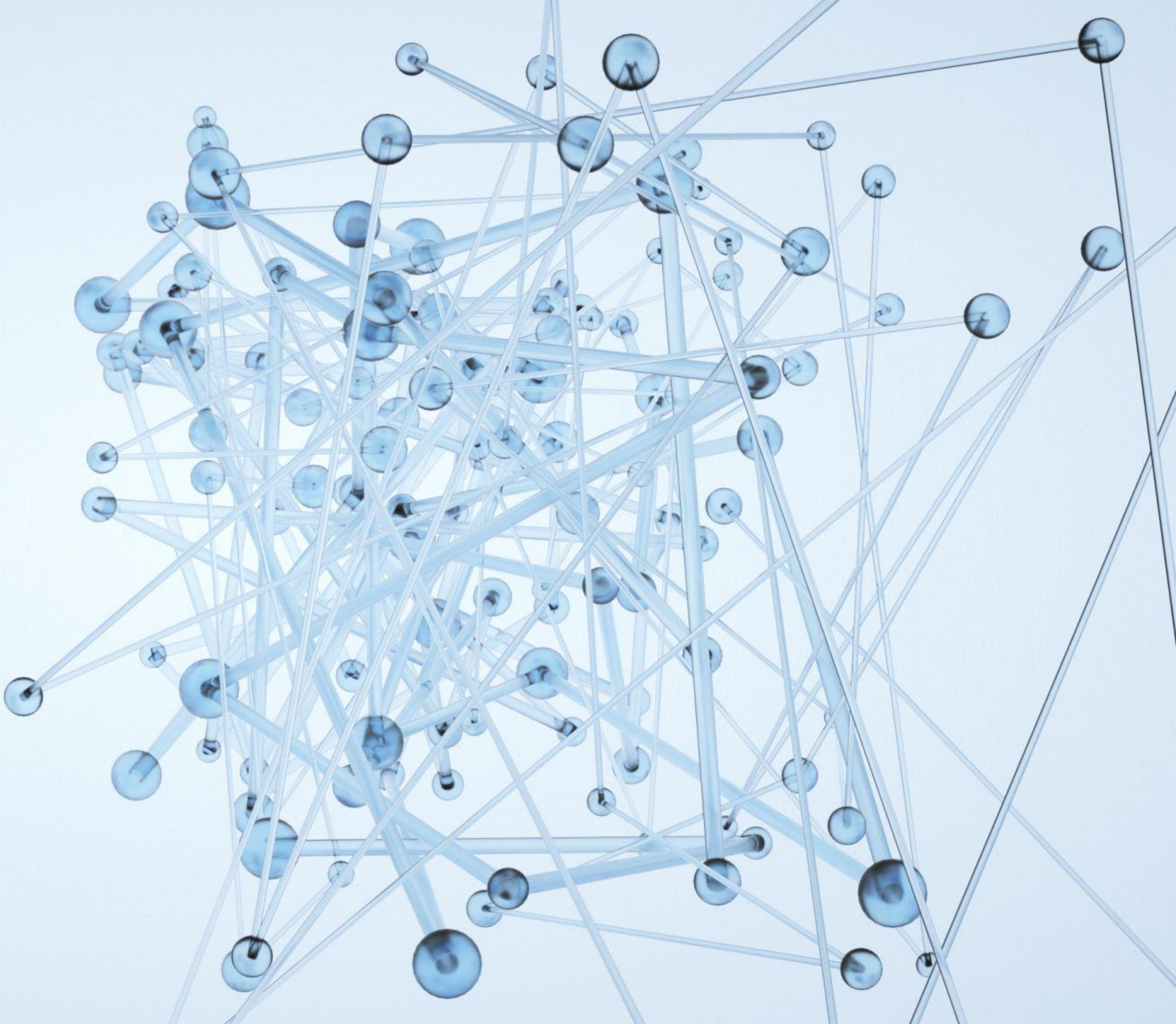
According to the evaluation results, the Neural Network Embedding-based recommender system demonstrated the lowest RMSE value of 0.1534, indicating superior performance in predicting user-item interactions compared to the other models.

As a result, we conclude that the Neural Network Embedding-based recommender system is the most effective model for collaborative filtering in this context.

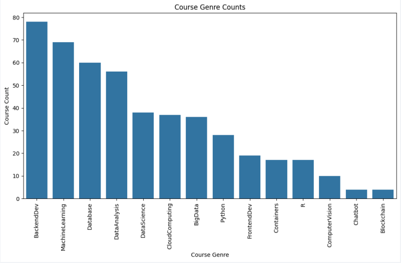


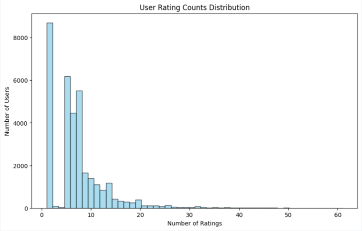
Conclusions

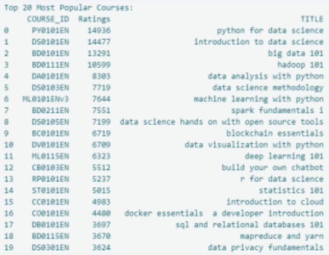
4



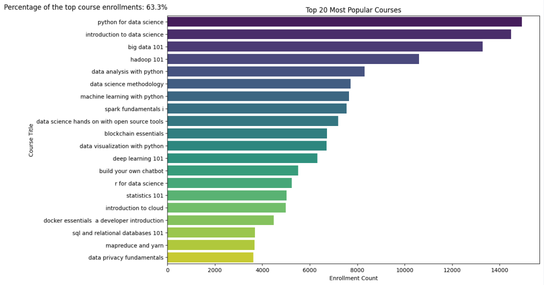
Conclusion of Exploratory Data Analysis







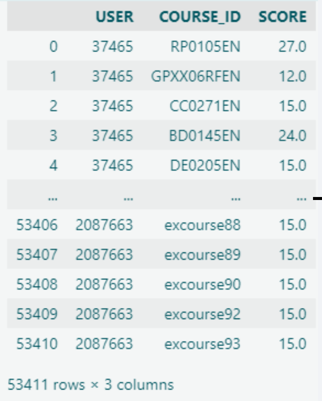
```
Top 20 Most Popular Courses:
     COURSE_ID  Ratings                                    TITLE
0     PY0101EN    14936                  python for data science
1     DS0101EN    14477             introduction to data science
2     BD0101EN    13291                             big data 101
3     BD0111EN    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
```



From the various analyses conducted, several key insights emerge regarding the course offerings and user engagement on the online learning platform. First, the examination of course genres revealed a wide range of topics, with backend development, machine learning, and databases standing out as the most popular based on enrollment figures. Additionally, the analysis of user enrollments provided insights into behaviour, showing that a substantial number of users completed courses rather than simply auditing them. Moreover, the review of the top 20 most popular courses highlighted a strong focus on data-centric topics, such as Python for Data Science, Introduction to Data Science, and Big Data 101, reflecting the increasing demand for skills in data analysis and machine learning. Overall, these findings emphasise the need for a diverse course selection that appeals to various interests while underscoring the value of data-driven insights in refining course offerings and boosting user engagement on the platform.

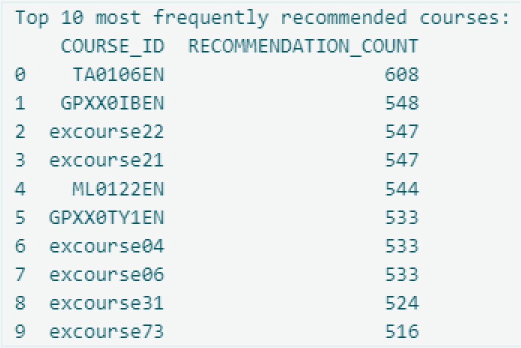
Conclusion of Content-based Recommender Using User Profile and Course Genrs

The content-based recommender system utilising user profiles and course genres begins by setting up the task of generating course recommendations based on the vectors from user profiles and course genres. The process involves loading user profile and course genre dataframes, extracting user interests, identifying courses unfamiliar to each user, computing recommendation scores, and filtering out courses that fall below a defined threshold. Once the recommendation scores are computed for all test users, the system's performance is evaluated. This evaluation includes calculating the average number of courses recommended per user and identifying the top 10 most frequently recommended courses across the user base. On average, 61.82 courses are recommended per user, reflecting a significant volume of suggestions. Among the most frequently recommended courses are "TA0106EN," "GPXX0IBEN," and others, indicating their popularity within the recommended course set.



	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

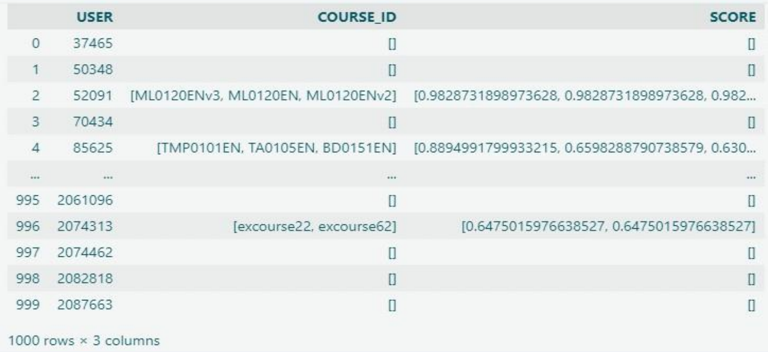
53411 rows × 3 columns



```
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
```

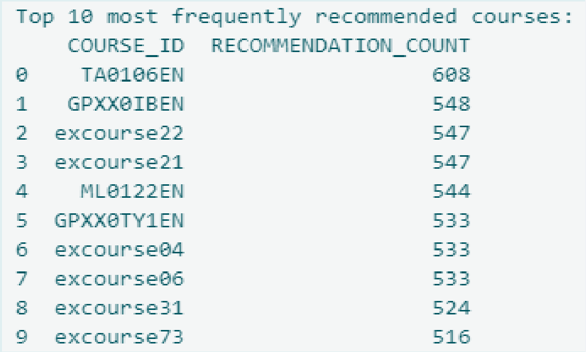
Conclusion of Content-based Recommender System Using Course Similarity

The conclusion from the content-based recommender system, which uses course similarity, is that the system was successfully implemented and evaluated. It leverages a similarity threshold of 0.6 to recommend courses based on users' interests and previously selected courses. By analysing course content and calculating similarities, the system delivers personalised recommendations tailored to each user. The evaluation provided important insights, such as the average number of new courses recommended per user and the most frequently recommended courses. These findings demonstrate the system's effectiveness in suggesting relevant and engaging content. Additionally, the evaluation results offer guidance for potential optimisations and improvements to enhance the system's performance in future iterations. Overall, this conclusion highlights the importance and efficacy of using course similarity-based approaches to deliver personalised recommendations in educational platforms.



	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	[]	[]

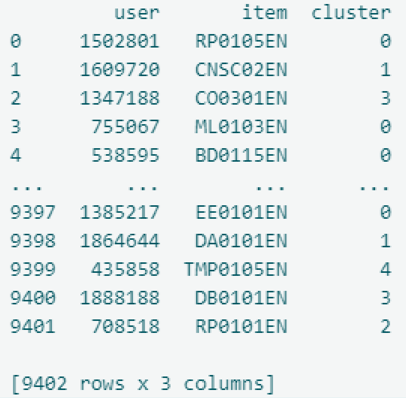
1000 rows × 3 columns



```
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
```

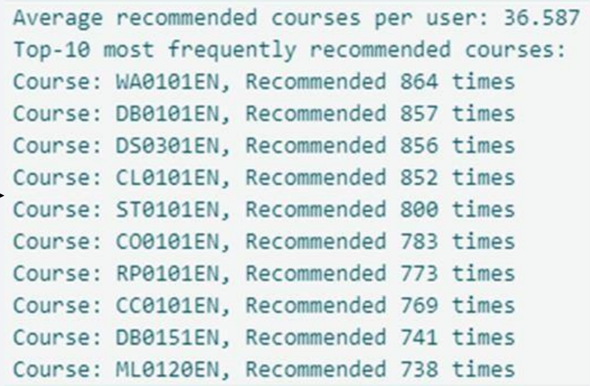
Conclusion of clustering-based recommender system

In conclusion, our clustering-based course recommender system demonstrates strong performance in effectively grouping users by their preferences and recommending relevant courses. By optimising key hyperparameters, such as the number of clusters and the selection of PCA components, the system accurately captures user interests while minimising information loss. Through careful evaluation, we found that the system recommends an average of 36.587 new or previously unseen courses per user, showcasing its ability to provide diverse recommendations. Additionally, the identification of frequently recommended courses like "WA0101EN," "DB0101EN," and "DS0301EN" offers valuable insights into user preferences and highlights the system's ability to promote popular courses across various clusters. Overall, this recommender system proves to be a valuable tool for enhancing user engagement and satisfaction by delivering personalised course suggestions aligned with individual learning goals.



```
         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.

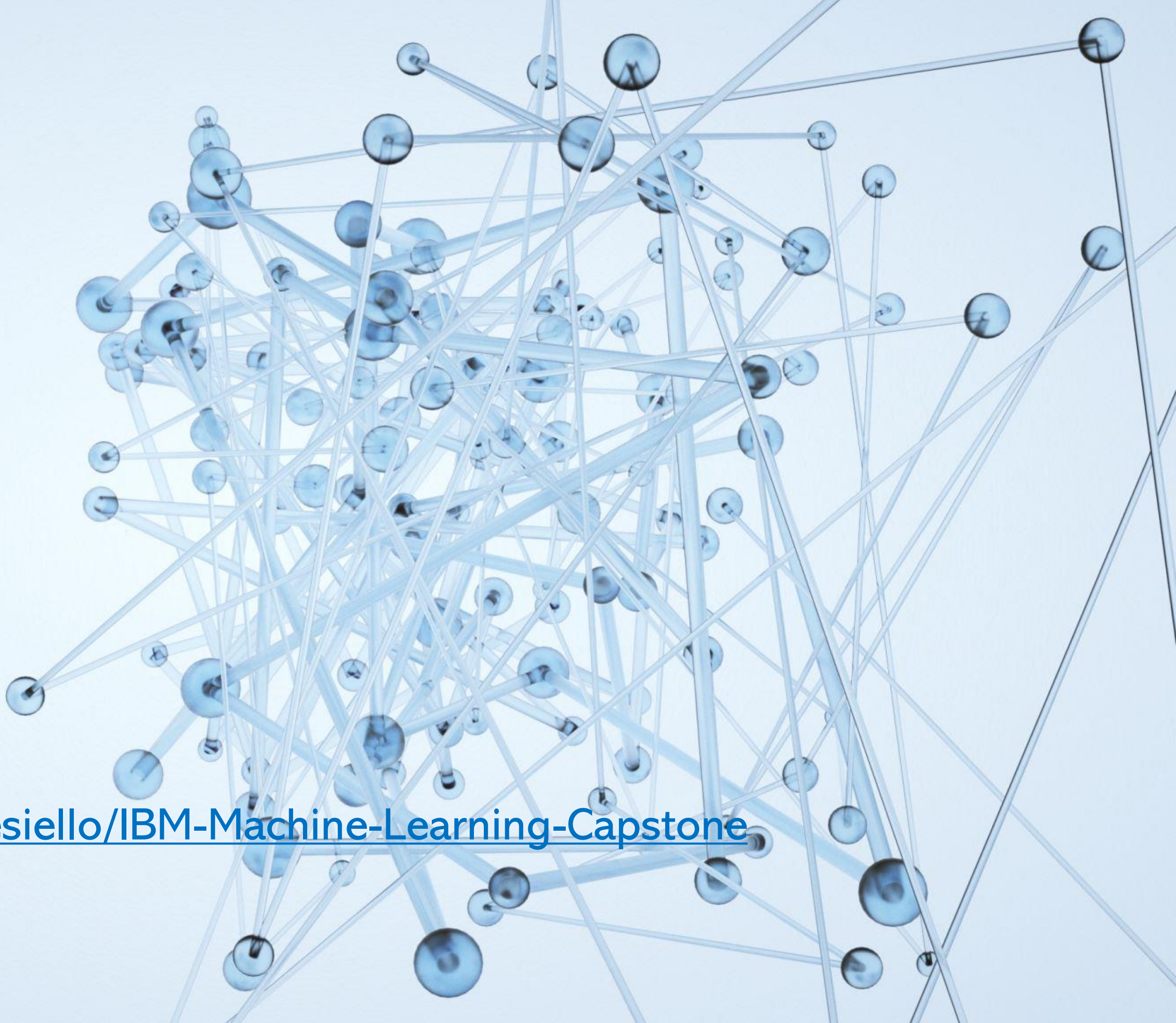
The comparative analysis of KNN-based, NMF-based, and Neural Network Embedding-based collaborative filtering methods reveals several important insights. Notably, the KNN-based and NMF-based systems tend to yield higher RMSE values when compared to the Neural Network Embedding method, which indicates that the latter is more effective at identifying latent patterns in user-item interaction data, resulting in improved prediction accuracy.

The strength of the **Neural Network Embedding** method lies in its ability to capture complex, nonlinear relationships between users and items, which allows it to discern subtle patterns in preferences. Despite its superior accuracy, this approach often requires more computational power and longer training times due to its more intricate architecture.

As a result, choosing the best collaborative filtering technique should be guided by the system's specific needs, where a balance between predictive performance and resource efficiency is essential.

	KNN	NMF	NNE
RMSE	0.2063	0.2048	0.1534

Appendix



https://github.com/Marchesiello/IBM-Machine-Learning-Capstone