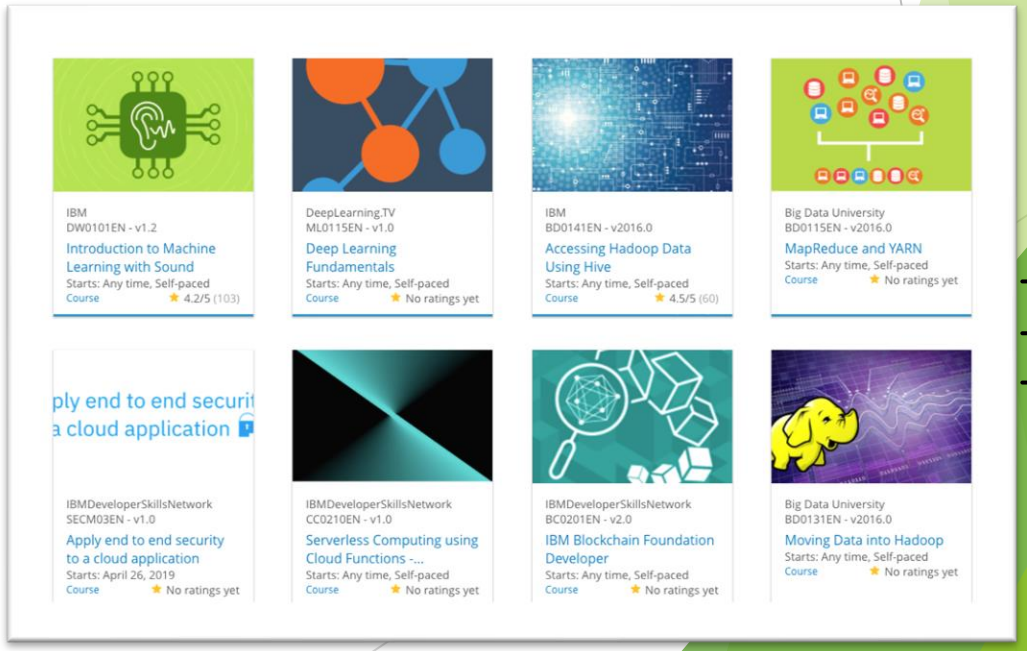
Building a Personalized Online Course Recommender System with Machine Learning

Artur Szczypta
05.01.2025



Outline

- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

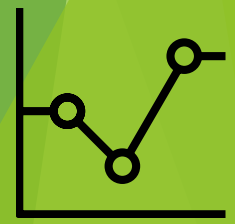
Introduction and Background

Increase of market demand for e-learning and online entertainment has lead to higher demand for recomender systems. Due to growing numer of customers and higher diversity of product it has become necessary to use machine learning to train recomender systems.

We'll explore their training processes, evaluation, and performance on a selected dataset. In folowing machine learning approaches:

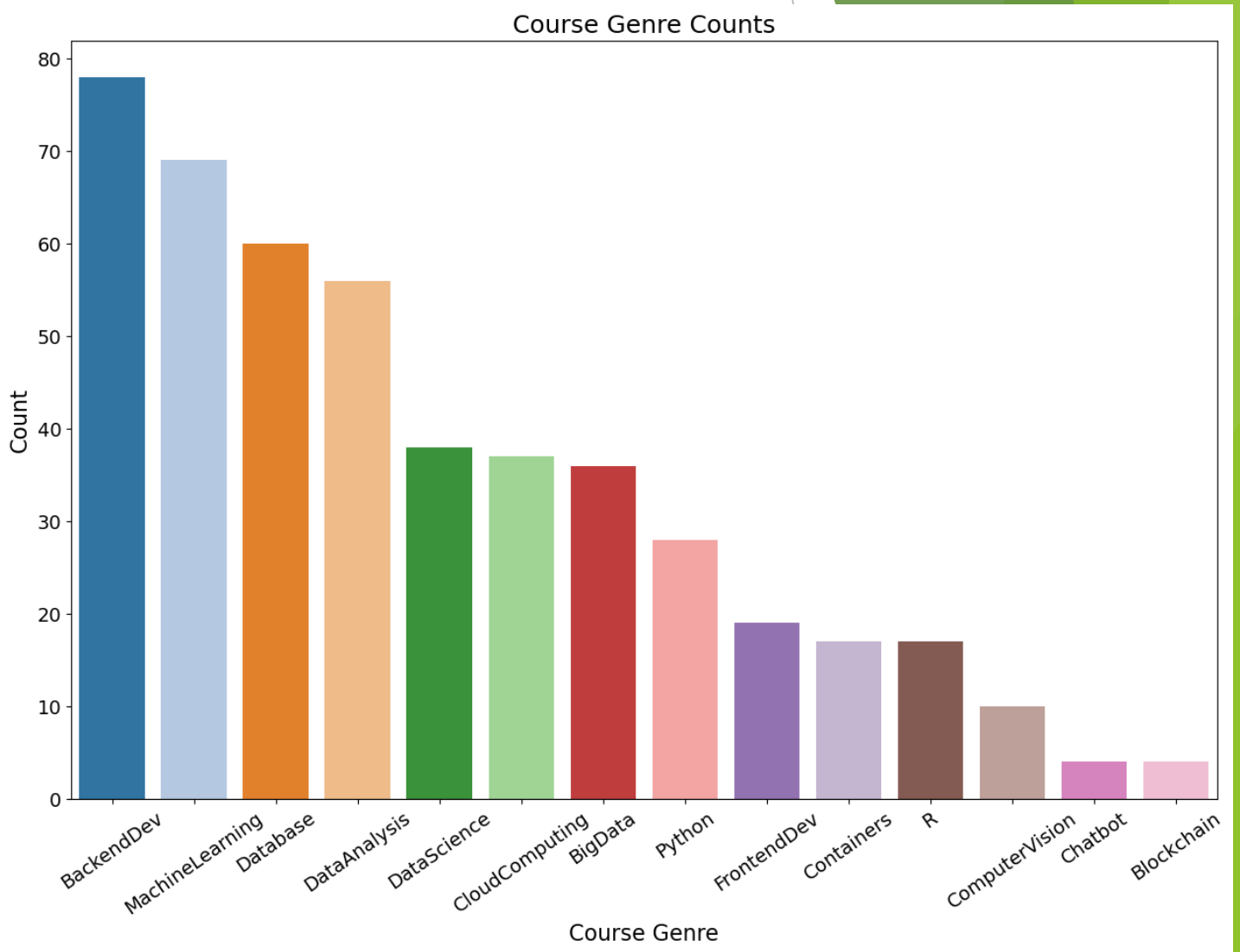
- K-Nearest Neighbors (KNN): A simple, instance-based algorithm for classification and regression.
- Non-negative Matrix Factorization (NMF): A dimensionality reduction technique used for uncovering latent structures in data.
- Neural Networks: Powerful models capable of capturing complex, non-linear patterns, widely used in deep learning tasks.

Exploratory Data Analysis



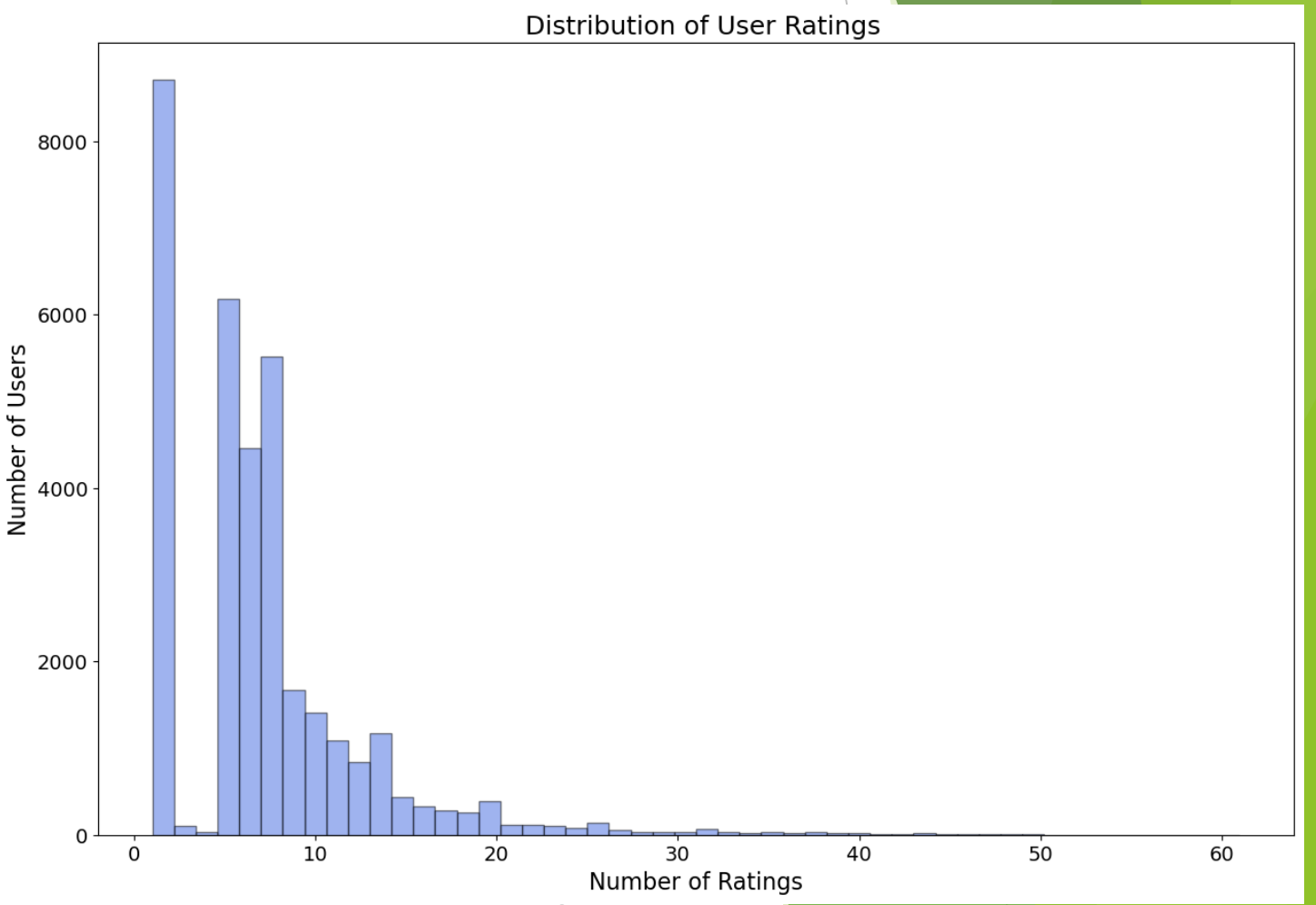
Most popular genres

To analyze course genres, the dataset was examined to uncover the popularity of various online topics. Course counts for each genre were calculated and visualized using a bar chart and table, highlighting trends in online education. The findings showed that Backend Development, Machine Learning, and Database courses are highly popular, while Blockchain and Chatbot courses are less common. This analysis offers valuable insights for learners and educators navigating current trends in online learning.



Course enrollment distribution

To analyze course enrollments, the dataset was studied to uncover user engagement patterns with online courses. By aggregating rating counts per user, it was revealed that the dataset contains 233,306 enrollment records from 5,000 unique users.
A histogram of rating counts showed varying engagement levels, with most users giving few ratings and a smaller group providing many. This analysis highlights the distribution of user interactions, offering insights to optimize course offerings and improve user experiences.

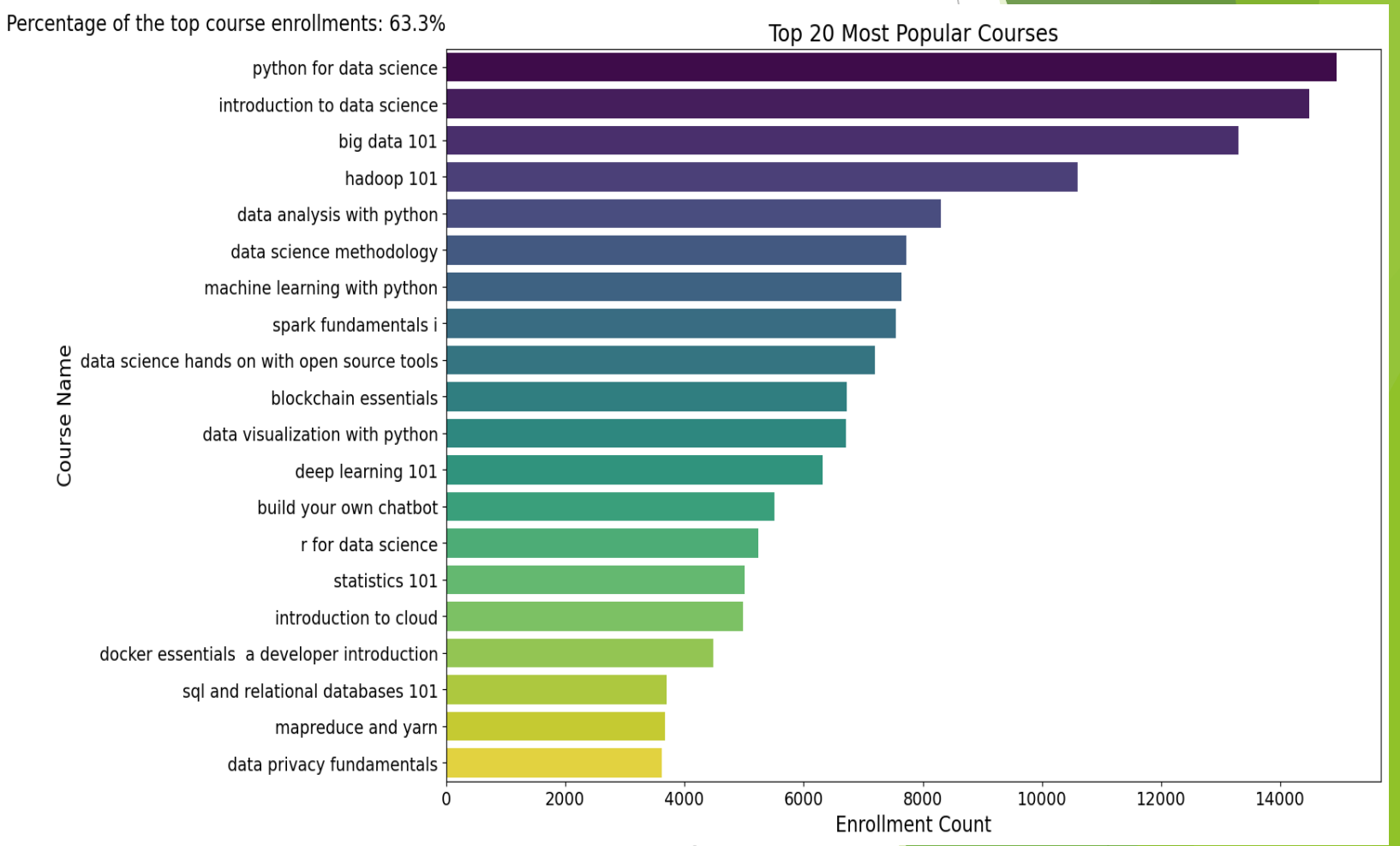


20 Most popular courses

20 most popular courses account for 63.3% of course enrolment.

All of these courses contain basic or introductory content.

Within these courses most prevelant topic is data science.

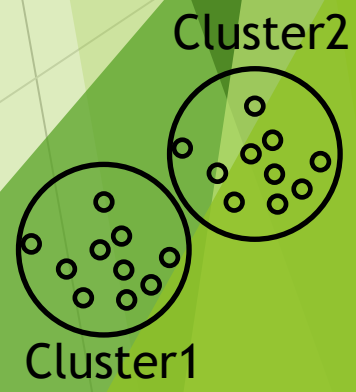


Visualising popular IT skills in courses through WordCloud analysis

This exploratory analysis uses WordCloud visualization to highlight prevalent keywords in online course titles.

Aggregated data reveals dominant themes like Python, data science, machine learning, AI, big data, TensorFlow, containers, and cloud computing. These insights offer a clear view of trending IT skills, helping learners and educators navigate the digital learning landscape.

Content-based Recommender System using Unsupervised Learning



Flowchart of content-based recommender system using user profile and course genres

1. **Raw Data:** The initial dataset contains user profiles, course genres, and potential interactions or ratings. This serves as the foundation for analysis.

2. **Data Processing:** This step cleans and prepares the raw data for analysis by addressing missing values, removing duplicates, and reformatting data as needed.

3. **Cleaned Dataset:** After processing, a cleaned dataset is produced, free of inconsistencies and ready for feature engineering. It includes all relevant user and course information.

4. **Feature Engineering:** New features are created to represent the data effectively. For this task, user profile vectors and course genre vectors are developed to capture user preferences and course characteristics.

5. **Final Features:** The output of feature engineering consists of user profile and course genre vectors. These features serve as the input for the content-based recommender system.

Evaluation results of user profile-based recommender system

• Hyperparameter Settings

A recommendation score threshold of 10.0 was set to filter out low-scoring recommendations, ensuring only highly relevant courses are suggested. Additional adjustments were made to hyperparameters, including feature representation methods and similarity metrics, to optimize the recommender system.

• Average New Courses Recommended per User

The system's performance was evaluated by calculating the average number of new courses recommended to each user in the test dataset. On average, approximately 62 courses per user were suggested, reflecting the system's coverage and diversity.

• Top 10 Most Frequently Recommended Courses

A table highlights the 10 most recommended courses based on the user profile-based recommender system. Each entry includes a COURSE_ID and its RECOMMENDATION_COUNT, indicating how often the course was suggested. These recommendations are derived from user profiles and course genre vectors, with higher-scoring courses aligned to user 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

1. **Raw Data:** The initial dataset contains details about various courses, including titles, descriptions, and other attributes.

2. **Data Processing:** Preprocessing involves tokenizing text into individual words and lemmatizing them to their base forms.

3. **Cleaned Dataset:** The dataset is cleaned by removing stopwords (common words with little meaning) and outliers (irrelevant or noisy data points).

4. **Feature Engineering:** The cleaned data is transformed into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency) vectors, which represent the importance of words within each course in the dataset.

5. **Features:** The resulting TF-IDF vectors serve as the features for each course. These features are used to calculate course similarities and generate recommendations based on content similarity.

Evaluation results of content-based recommender system using course similarity

• Hyperparameter Settings

A similarity threshold of 0.6 was set for the course similarity-based recommender, determining the minimum similarity required for a course to be recommended.

• Average New Courses Recommended per User

On average, 0.987 new or unseen courses were recommended per user in the test dataset, reflecting the system's ability to suggest diverse and novel content.

• Top 10 Most Recommended Courses

The most frequently recommended courses included "excourse 22" and "excourse 62," each appearing 257 times, followed by "WA 0103 EN" with 101 recommendations. Other popular courses, such as "TA 0105" and "DS 0110 EN," were also highlighted, showcasing their relevance to users. These insights help evaluate the system's performance and guide future improvements.

```
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

1. **Raw Data:** The original user profile feature vectors represent users' interests across various course genres, such as Machine Learning, Data Science, and Cloud Computing.

2. **Data Processing:** Raw data is preprocessed to address missing values, outliers, or quality issues. Techniques like StandardScaler are applied to normalize features, ensuring consistent scale and distribution for effective clustering.

3. **Cleaned Dataset:** After processing, the dataset is standardized using StandardScaler, resulting in features with a mean of 0 and a standard deviation of 1 - ideal for many machine learning algorithms.

4. **Feature Engineering:** Principal Component Analysis (PCA) is applied to reduce dimensionality while retaining essential information. PCA identifies key components that explain the most variance in the data, transforming the original features into a simplified representation.

5. **PCA-Transformed Features:** The final output is a lower-dimensional feature set from PCA. These features combine information from the original dataset, capturing its most significant characteristics.

Evaluation results of clustering-based recommender system

• Hyperparameter Settings

Using the K-means algorithm, the ideal number of clusters was determined via the elbow method. For PCA, components explaining over 90% of variance were selected, balancing effective user grouping with minimal information loss.

• Average New Courses Recommended per User

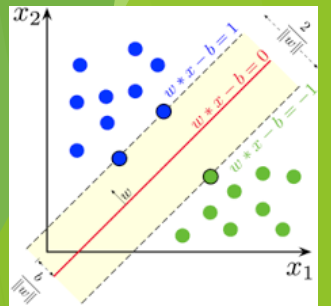
On average, the system recommended about 37 new courses per user, showcasing its ability to diversify suggestions and introduce a wide range of learning opportunities.

• Top 10 Most Recommended Courses

Popular courses like "WA 0101 EN," "DB 0101 EN," and "DS 0301 EN" were frequently recommended, highlighting user preferences across clusters. These insights guide further refinement of recommendation strategies to better match user interests and goals.

```
Average recommended courses per user: 37
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



Flowchart of KNN based recommender system

1. **Raw Data:** The original dataset includes user-item interactions, such as user IDs, item IDs, and ratings, forming the foundation for analysis.

2. **Data Processing:** This step cleans the data by handling missing values, removing duplicates, and structuring it for analysis, ensuring readiness for modeling.

3. **Cleaned Dataset:** The cleaned dataset is free from irrelevant or erroneous entries, with missing values addressed and data structured, serving as the basis for the recommendation model.

4. **Feature Engineering:** New or transformed features are created to enhance system performance, including user-item interactions, timestamps, and demographics. Tasks like encoding, scaling, or creating interaction terms may be applied.

5. **PCA Transformed Features:** Final features include relationships between users and items, transformed through PCA for dimensionality reduction. These features enable the KNN-based recommender to identify similarities and provide personalized recommendations.

Flowchart of NMF based recommender system

1. **Raw Data:** The initial, unprocessed data includes course ratings, often containing noise, missing values, or inconsistencies.

2. **Data Processing:** The raw data is preprocessed to remove duplicates, handle missing values, and format it for analysis. We used Pandas to pivot the data into a user-item matrix.

3. **Cleaned Dataset:** The cleaned dataset is free from inconsistencies, structured with users as rows, items as columns, and ratings or interactions in the cells.

4. **Feature Engineering:** New features or transformations are created to enhance model performance, such as user-item interactions or latent factors generated by the NMF model.

5. **Features:** Features are the variables used by machine learning models for predictions, including user preferences, item attributes, and latent factors representing users and items in a reduced-dimensional space.

Flowchart of Neural Network Embedding based recommender system

1. **Raw Data:** The unprocessed course ratings data, which may contain noise, missing values, or inconsistencies.

2. **Data Processing:** Preprocessing tasks, including handling missing values, removing duplicates, and formatting the data into a user-item matrix using Pandas.

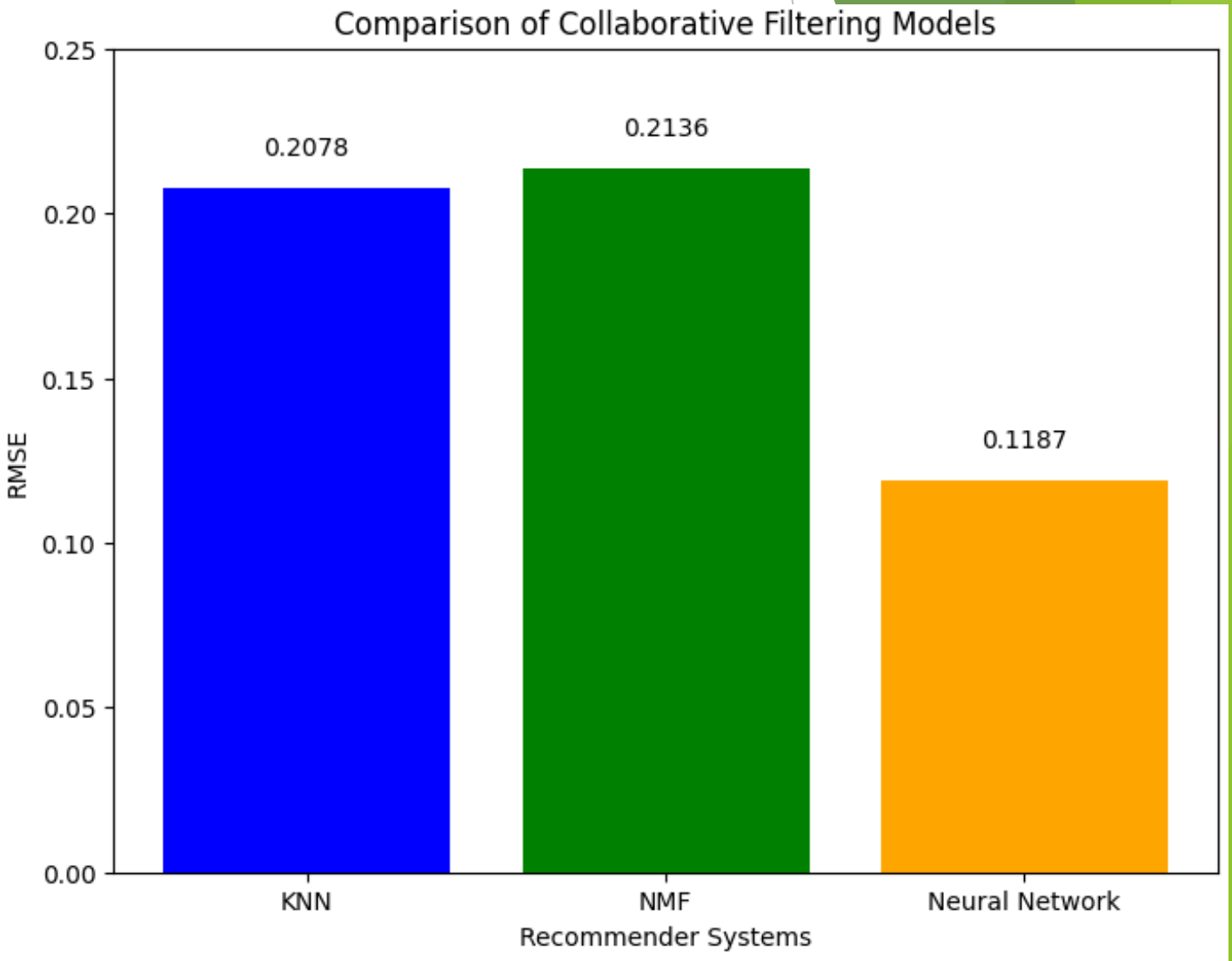
3. **Cleaned Dataset:** The cleaned dataset is free from inconsistencies and structured with users as rows, items as columns, and ratings in the cells.

4. **Feature Engineering:** Creating or transforming features, such as user-item interactions or latent factors from the NMF model, to improve model performance.

5. **Features:** The variables used for machine learning predictions, including user preferences, item attributes, or latent factors in a lower-dimensional space.

Comparison of Models

The Neural Network Embedding-based recommender system achieved the lowest RMSE of 0.1187, outperforming the other models in predicting user-item interactions. Therefore, it is considered the most effective model for collaborative filtering in this scenario.



Conclusions

Exploratory Data Analysis:

- Popular Course Genres: Backend development, machine learning, and databases are the most popular genres, reflecting user demand.
- Data-Driven Insights: A significant portion of users completed courses, highlighting the importance of using data insights to optimize course offerings.

Content-based Recommender System Using User Profile and Course Genres:

- Recommendation Process: The system generates recommendations based on user profiles and course genre vectors, resulting in an average of 61.82 recommended courses per user.
- Popular Courses: "TA0106EN" and "GPXX0IBEN" were the most frequently recommended, indicating user interest in these courses.

Content-based Recommender System Using Course Similarity:

- Effective Personalization: The course similarity-based system uses a 0.6 similarity threshold to recommend personalized content based on user interests and previous selections.
- Valuable Insights: The evaluation highlights the effectiveness of the system in suggesting relevant content, with recommendations informing future improvements.

Conclusions

Clustering-Based Recommender System:

- User Grouping: The system effectively groups users based on preferences and recommends new courses, averaging 37 unseen courses per user.
- Popular Course Recommendations: Courses like "WA 0101 EN" and "DB 0101 EN" were frequently recommended, reflecting user interest and system accuracy.

Performance of Three Collaborative Filtering Models:

- Superior Performance of Neural Network Embedding: The Neural Network Embedding-based system outperformed KNN and NMF in prediction accuracy due to its ability to capture complex patterns.
- Computational Trade-off: Although more accurate, the Neural Network Embedding system requires more resources and time, suggesting a need for balancing performance with efficiency.

Appendix

GutHub:
https://github.com/ArturSzczypta/IBM-Machine-Learning-Professional-Certificate/tree/main/Machine%20Learning%20Campstone