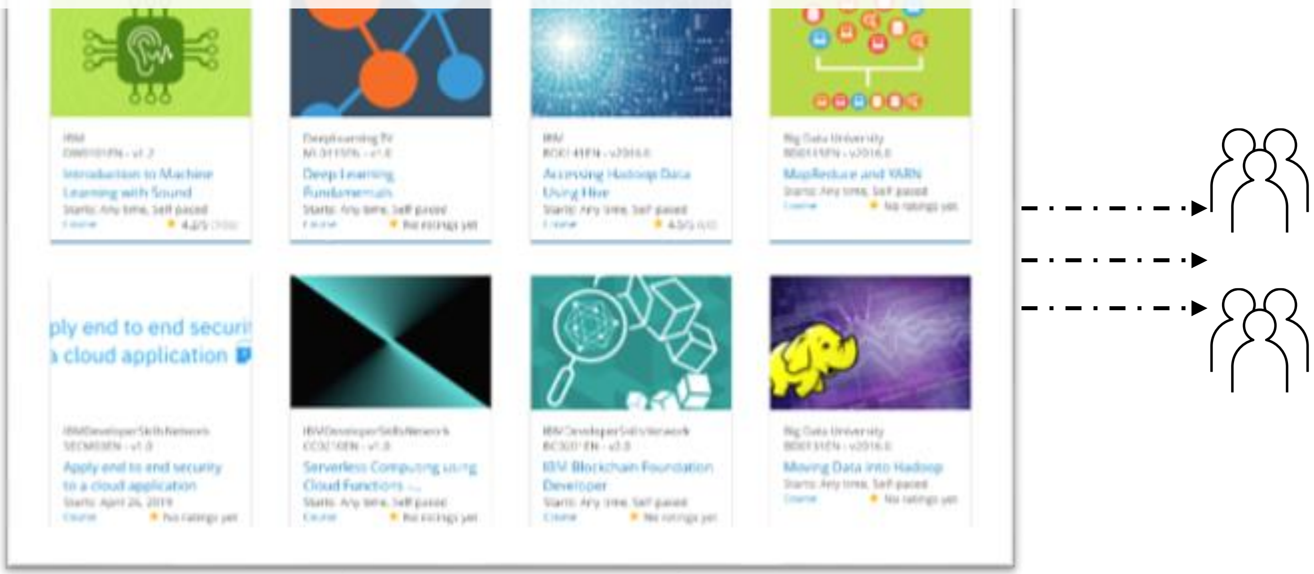
# Build a Personalized Online Course Recommender System with Machine Learning

Pedro Antonio Vázquez
González
2025/01/10



# 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

**Project background and context**

- This project explores how recommendation systems can personalize user experiences through various machine learning approaches.
- The main focus is on analyzing content-based methods and collaborative filtering, comparing their performance and effectiveness.
- It aims to address the problem of information overload by helping users find relevant and personalized courses.

**Problem states and hypotheses**

- **Problem:**

How can relevant courses be identified for individual users or groups using their history and course attributes?

- **Hypotheses:**

1. Content-based systems are effective in recommending courses by analyzing similarities between users and courses.

2. Collaborative filtering can capture patterns in user behavior to enhance personalization.

3. Combining different approaches can improve recommendation accuracy and mitigate individual method limitations.

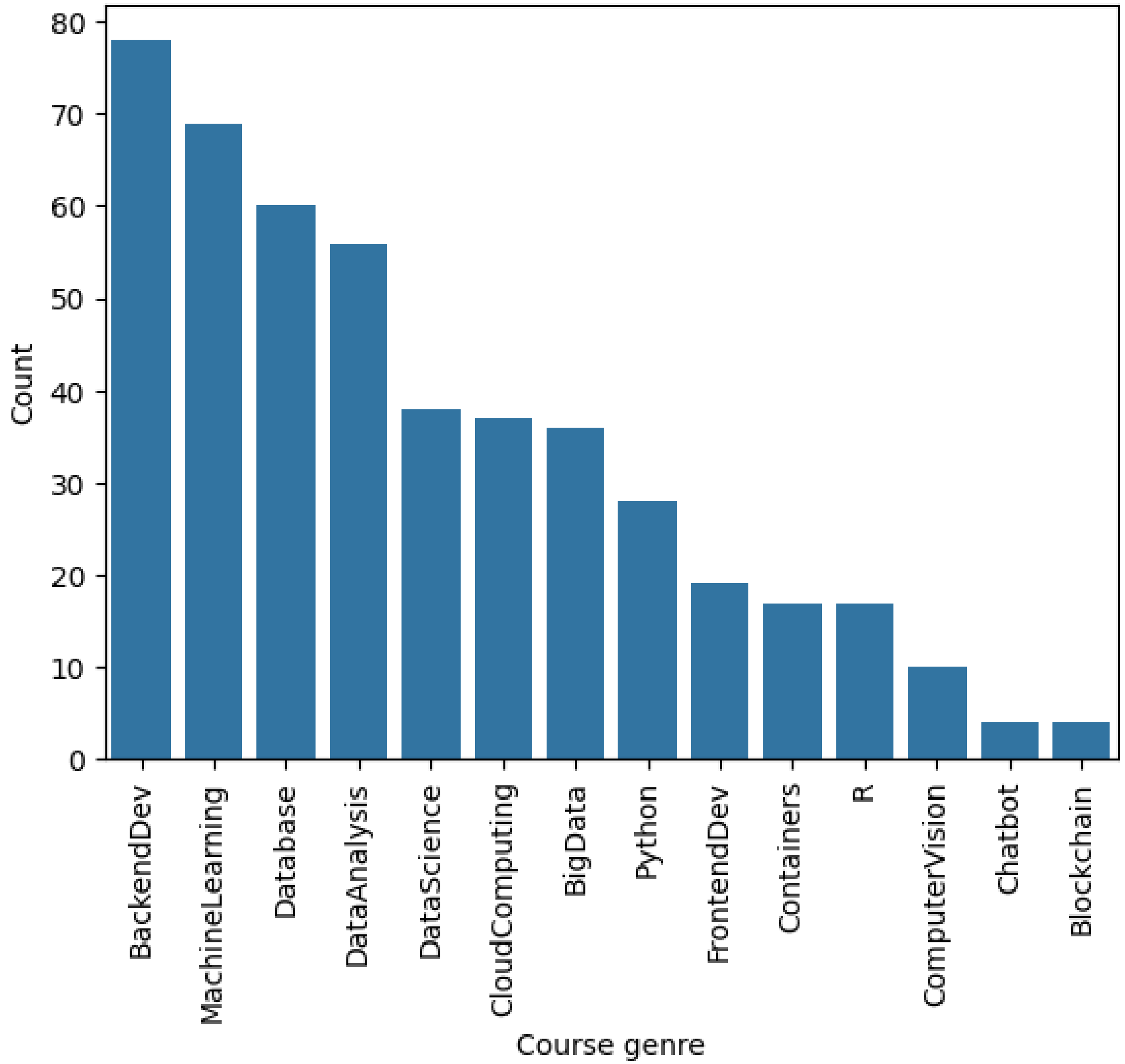
# Exploratory Data Analysis

# Course counts per genre

**Key observations:**

- Backend Development has the highest number of courses (78), indicating strong coverage in this area.
- Machine Learning and Database follow closely, reflecting their popularity and demand in the field.
- On the other hand, genres such as Chatbot and Blockchain have significantly fewer courses (4 each), suggesting niche or emerging topics.

This visualization helps highlight areas of focus and potential gaps in course offerings.

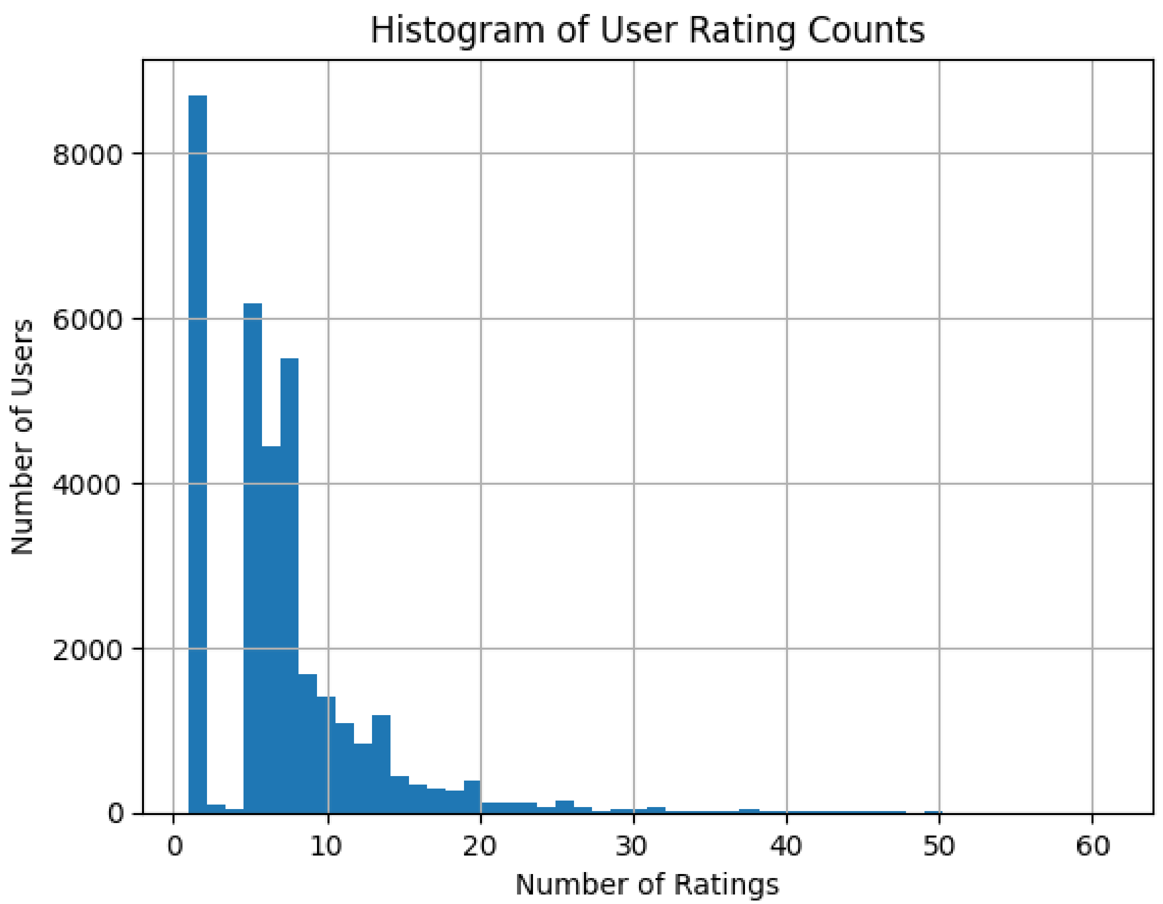


# Course enrollment distribution

**Key insights from the data:**

- The majority of users enrolled in 6 courses or fewer, as indicated by the median value (6) and the interquartile range (2-9).
- A smaller proportion of users enrolled in a significantly higher number of courses, with the maximum being 61.
- The data shows a right-skewed distribution, reflecting that a few highly engaged users account for the upper end of enrollments.

This analysis helps identify patterns in user engagement and may guide decisions on course content and marketing strategies.



# 20 most popular courses

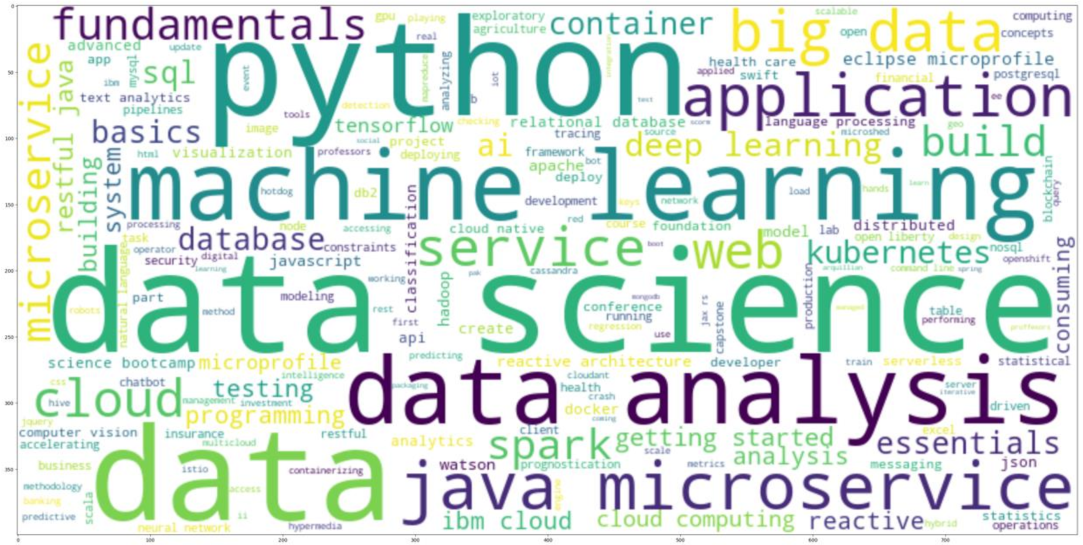
| | TITLE | Ratings |
|---|---|---|
| 0 | python for data science | 14936 |
| 1 | introduction to data science | 14477 |
| 2 | big data 101 | 13291 |
| 3 | hadoop 101 | 10599 |
| 4 | data analysis with python | 8303 |
| 5 | data science methodology | 7719 |
| 6 | machine learning with python | 7644 |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

**Key insights:**

- **“Python for Data Science”** and **“Introduction to Data Science”** lead the list, with over 14,000 ratings each, reflecting their relevance in the current data-driven job market.
- Courses focused on foundational topics like **Big Data 101**, **Hadoop 101**, and **Data Analysis with Python** also rank highly, showing demand for technical skills.
- Emerging topics like **Blockchain Essentials** and **Deep Learning 101** are gaining popularity, highlighting growing interest in these areas.

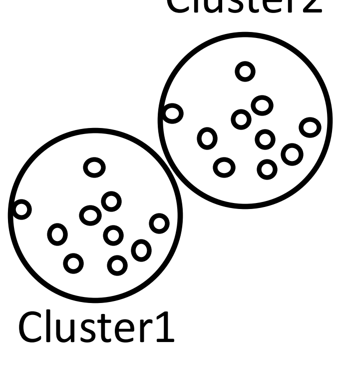
This information can help prioritize content updates and marketing efforts based on user preferences.

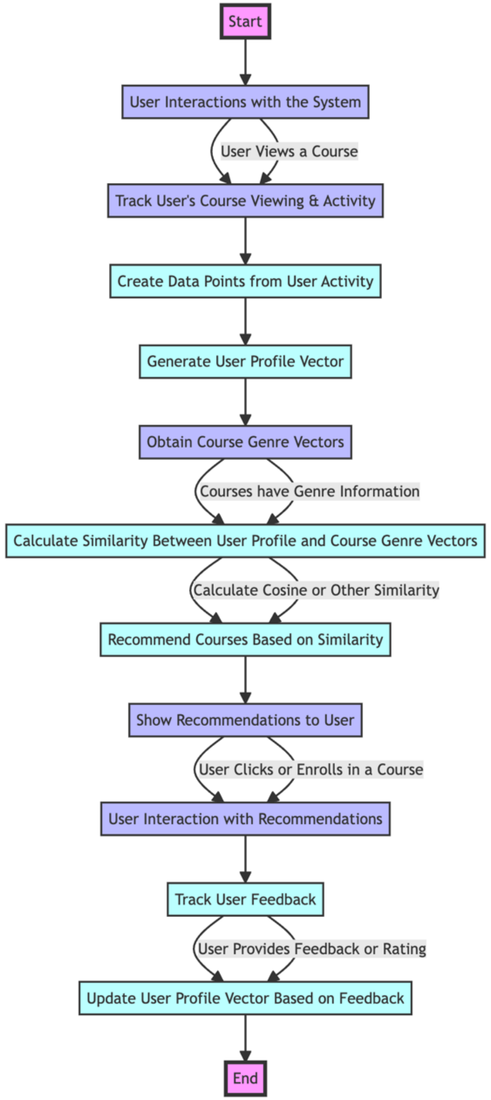
# Word cloud of course titles




The word cloud highlights the most relevant topics in data science and technology, with larger words like ‘Python,’ ‘data,’ and ‘machine learning’ showing core concepts. Smaller words reveal specialized areas like ‘visualization,’ ‘neural networks,’ and ‘SQL,’ emphasizing the diversity and breadth of the field.

# Content-based Recommender System using Unsupervised Learning





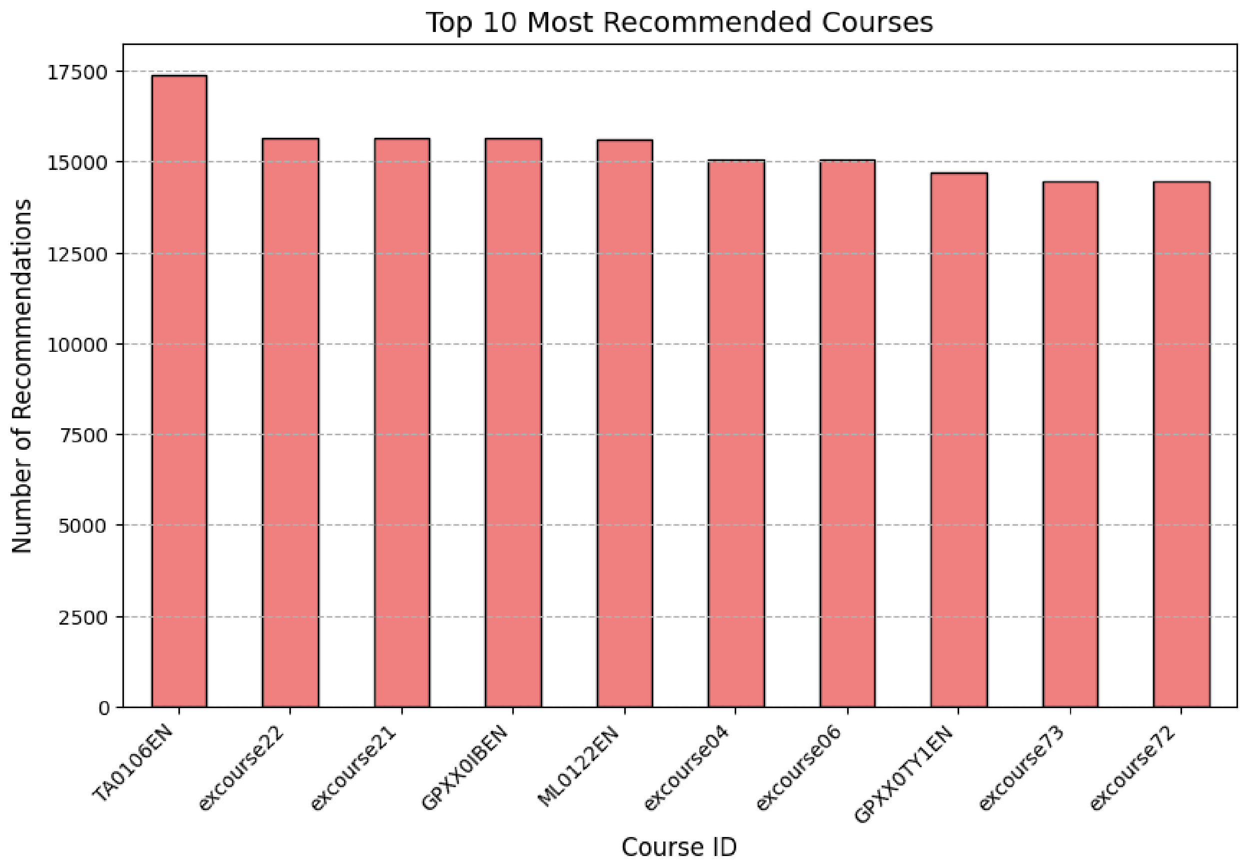

# Flowchart of content-based recommender system using user profile and course genres

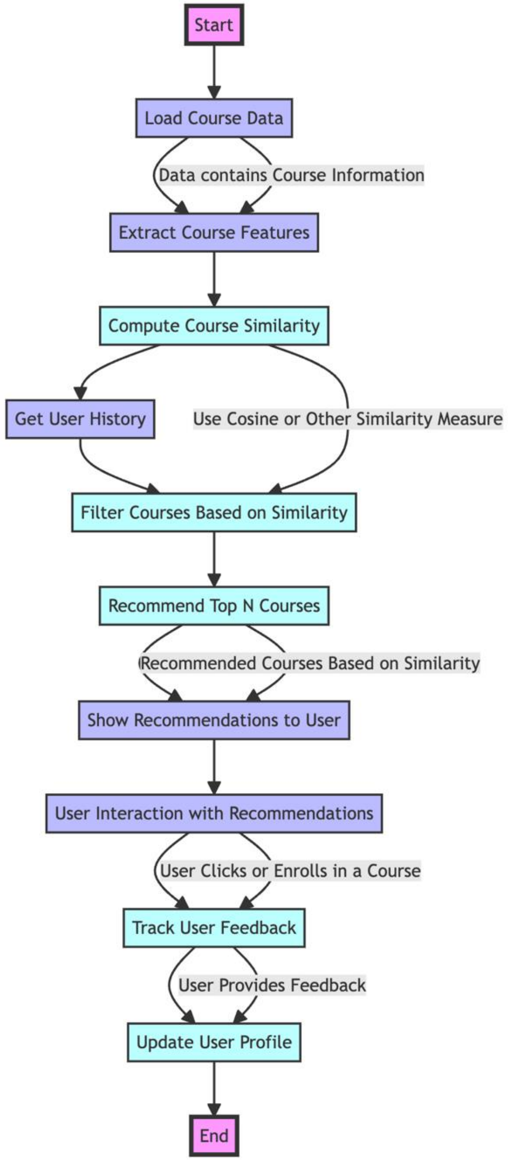
The content-based recommender system uses user interactions with courses to create a **user profile**. This profile is compared with **course genres** to suggest relevant content. As the user interacts more, recommendations are adjusted and improved, making the system more accurate and personalized.

# Evaluation results of user profile-based recommender system

- **Recommendation Score Threshold**: 12 (Only courses with a score of 12 or higher are recommended).
- **Course Similarity Threshold**: Similarity scores are calculated based on the dot product between user profile vectors and course genre vectors.

On average, each user was recommended 60.82 new/unseen courses.

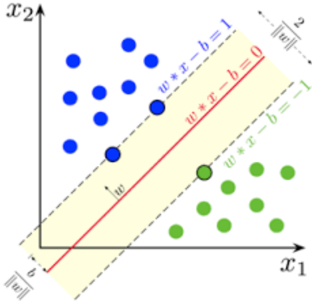


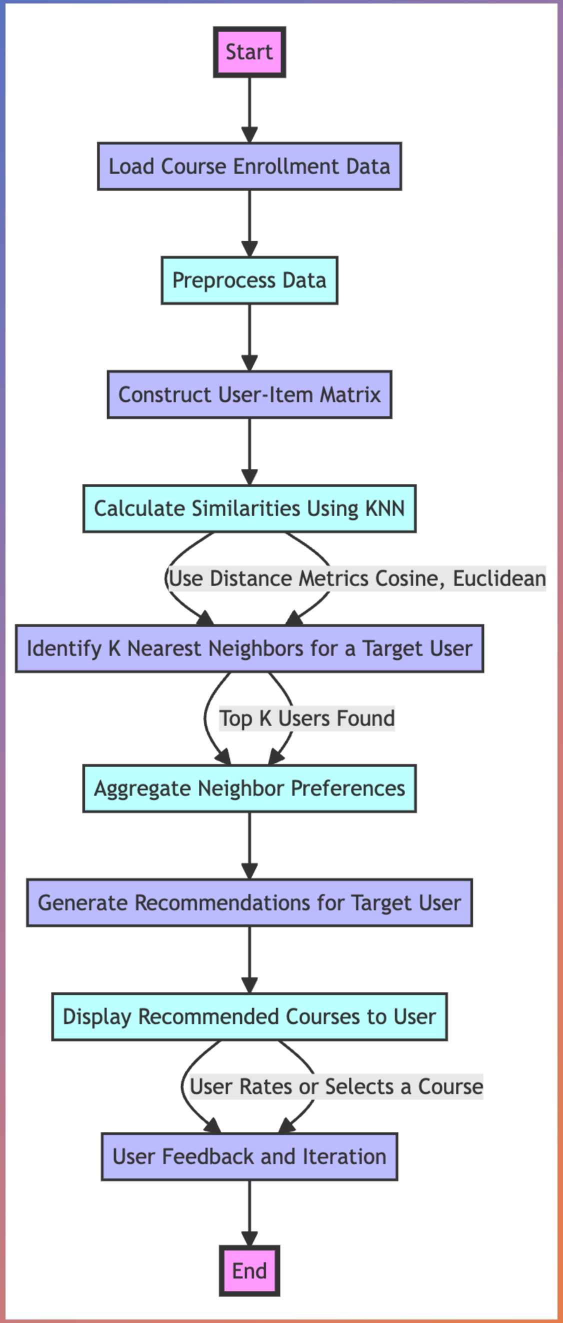



# Flowchart of content-based recommender system using course similarity

The flowchart describes the process of a course similarity-based recommender system. First, course data such as ID, title, genre, and description is loaded. Then, the course features are extracted, and the similarity between courses is calculated using a measure like cosine similarity. Next, the user history (courses they have taken) is retrieved, and courses similar to that history are filtered. Finally, the most similar courses are recommended, and the user can interact with the recommendations. User feedback is used to update their profile.

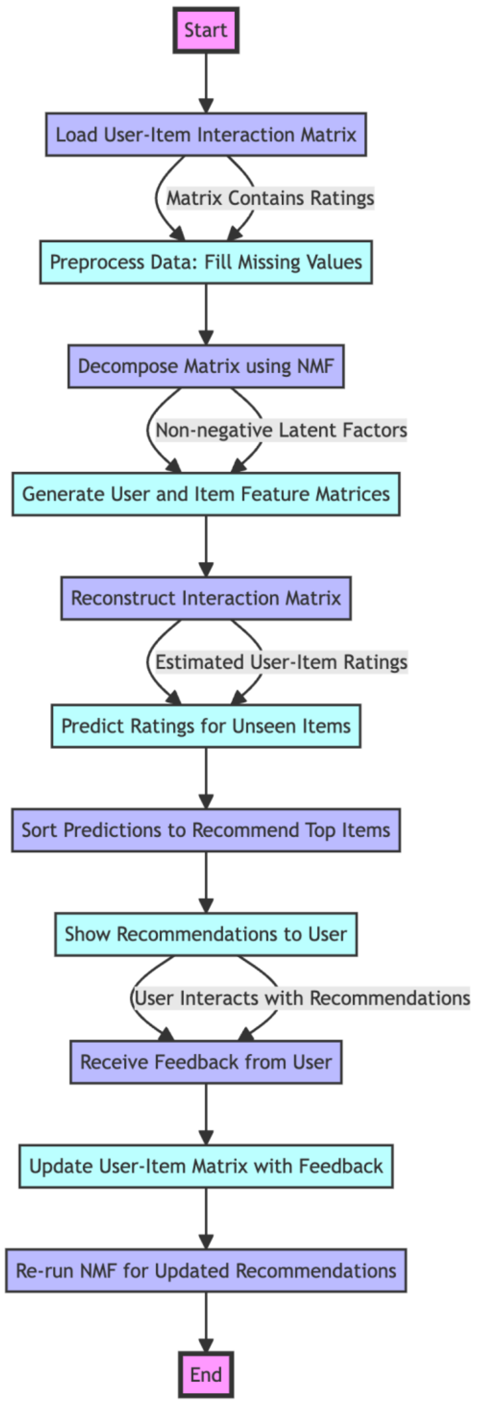
# Collaborative-filtering Recommender System using Supervised Learning





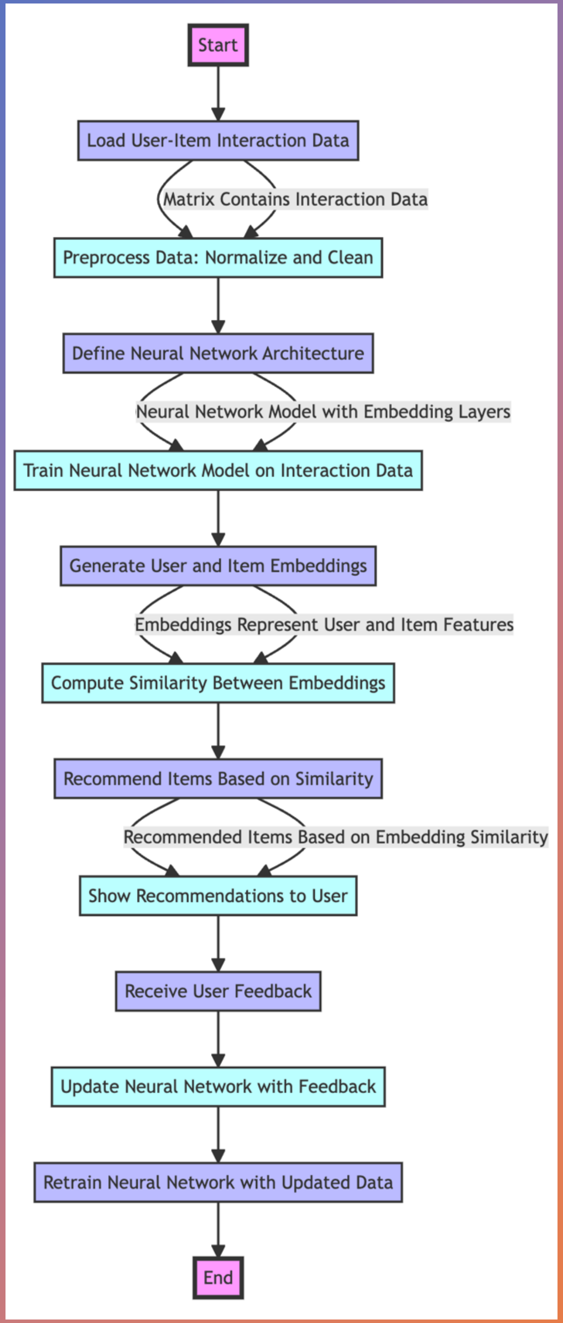

# Flowchart of KNN based recommender system

The flowchart illustrates a KNN-based recommender system using course enrollment history. User-course interactions are processed into a matrix, and similarities between users are calculated with metrics like cosine distance. For a target user, the top K nearest neighbors' preferences are used to recommend courses. User feedback refines future recommendations."




# Flowchart of NMF based recommender system

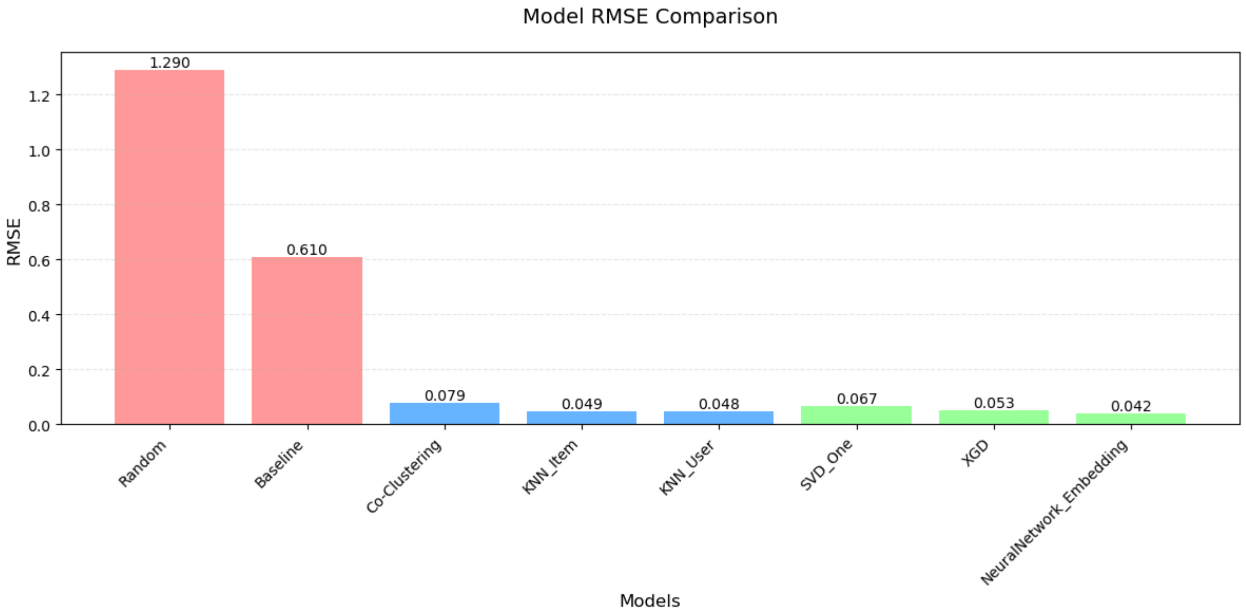
The flowchart depicts the process of an **NMF (Non-negative Matrix Factorization)**-based recommender system. The system starts by loading the user-item interaction matrix, which contains available ratings. The data is then preprocessed to handle missing values. Using NMF, the matrix is decomposed into user and item feature matrices. These matrices are used to reconstruct the interaction matrix and predict ratings for unseen items. The predictions are sorted to recommend the most relevant items to the user. Finally, the system incorporates user feedback to update the original matrix and refine the recommendations.




# Flowchart of Neural Network Embedding based recommender system

The flowchart illustrates a **Neural Network Embedding**-based recommender system. It starts by loading user-item interaction data, followed by preprocessing. A neural network is then trained to generate embeddings for users and items, calculating similarity to recommend relevant items. The model is updated with user feedback and retrained.

# Compare the performance of collaborative-filtering models




**Key Observations:**

- Neural Network Embedding results in the lowest RMSE of 0.042.
- Collaborative filtering methods overall perform much better than the baseline.
- KNN and NMF exhibit comparable performance.
- As anticipated, the random baseline performs poorly.

# Conclusions

Throughout this project, we explored various approaches to recommender systems, including course similarity-based systems, clustering methods like K-means, KNN-based methods, and neural networks. During the process, workflows were implemented to integrate similarity calculations, user profile construction, and personalized recommendation generation. These systems demonstrated how machine learning techniques and data analysis, especially neural networks, can transform user interactions on educational platforms.

In conclusion, Neural Network Embedding emerged as the most promising technique, showcasing its strength in providing highly accurate recommendations. While other methods like KNN and NMF also performed well, it was clear that the use of neural networks significantly improved performance. This project not only reinforced technical skills in Python and related libraries but also deepened the understanding of how advanced techniques like neural networks can optimize recommender systems, making them more efficient and effective.