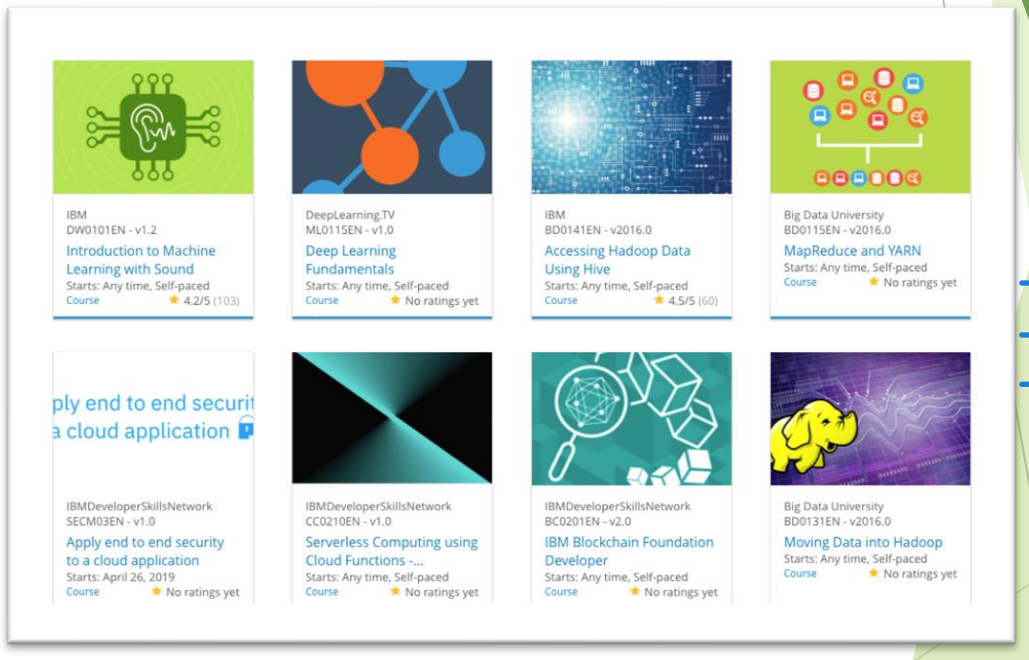
Build a Personalized Online Course Recommender System with Machine Learning

Vasantan Govinda Raju

15 Feb 2025



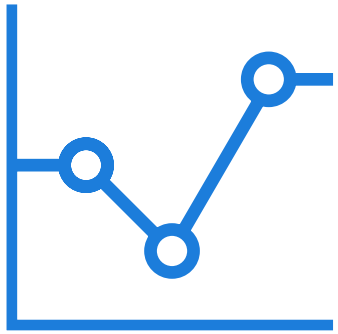
Outline

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

Introduction

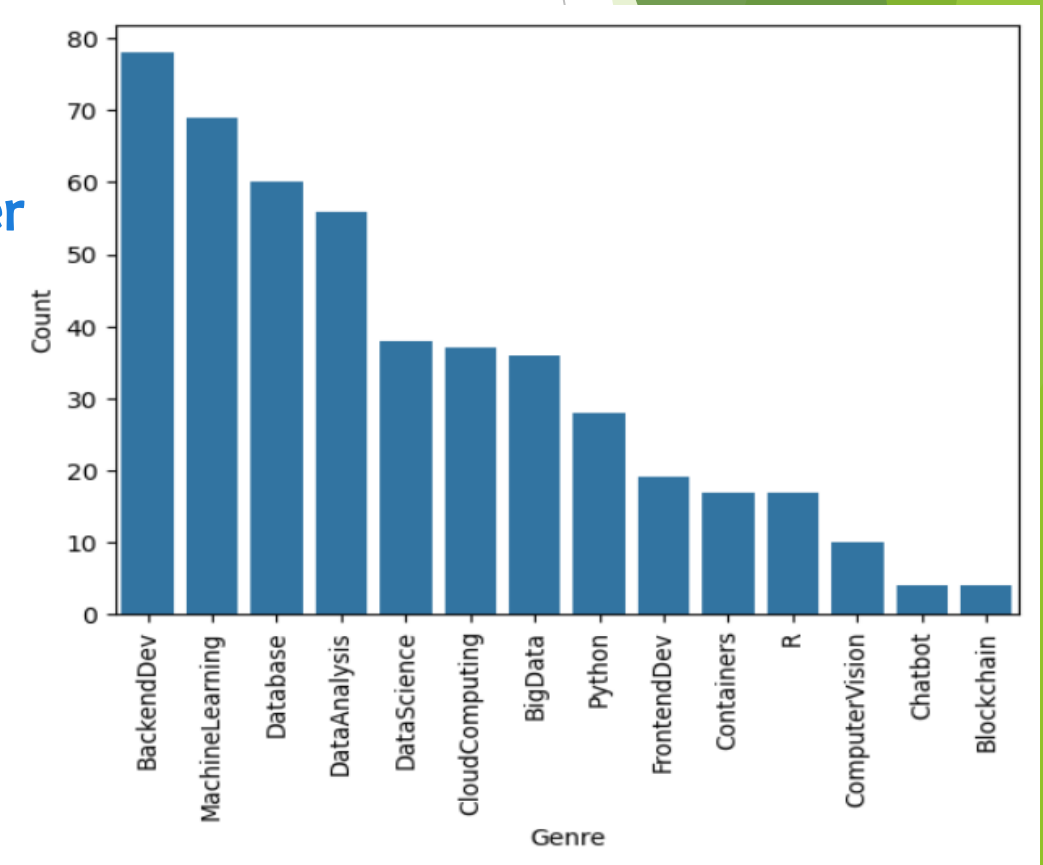
- Background & Context
  - Online learners struggle to find the right courses.
  - A personalized recommender system can improve engagement.
- Problem Statement
  - Goal: Suggest relevant courses using machine learning.
  - Approach: Use content-based & collaborative filtering.

Exploratory Data Analysis



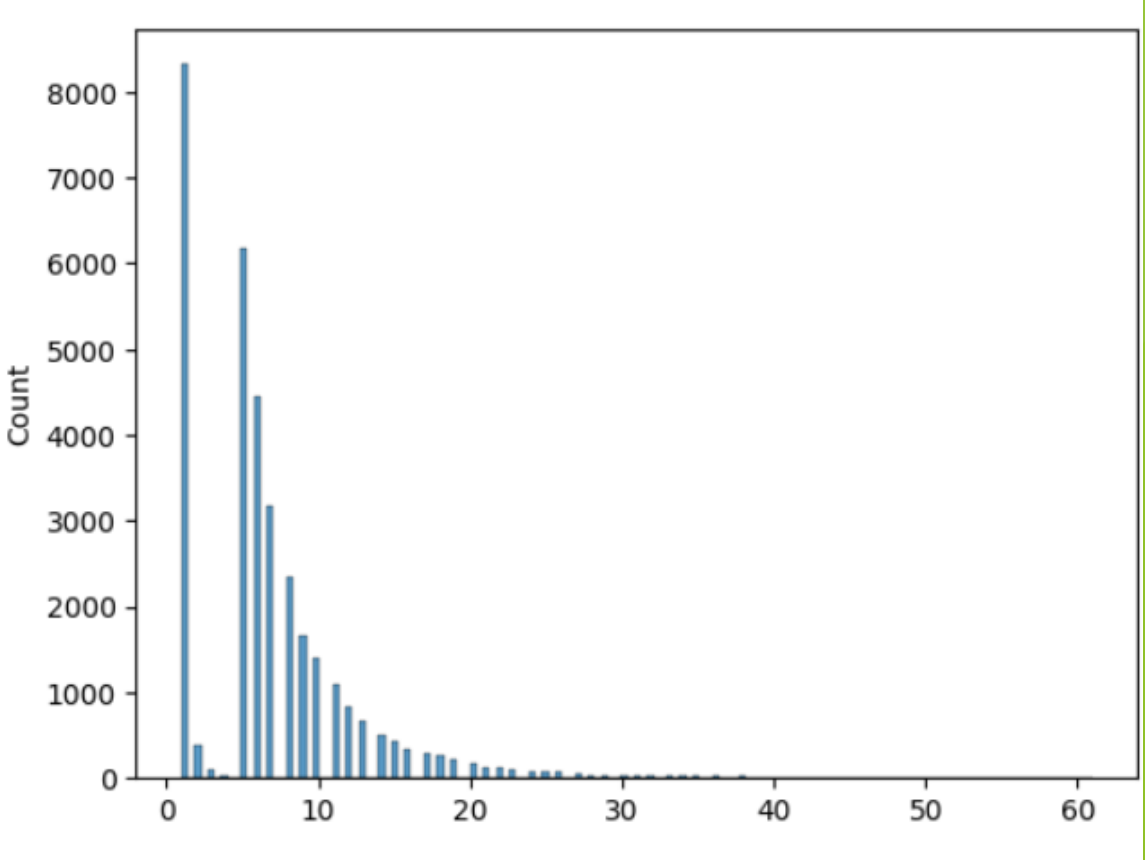
Course Counts Per Genre

- Backend Development has the most courses, followed by Machine Learning and Databases.
- Niche fields like Chatbots, Blockchain, and Computer Vision have fewer courses.
- The trend shows high availability in popular tech fields and limited options in specialized areas.
- This helps in better course recommendations based on user interest.



Course Enrollment Distribution

- Most users enroll in only a few courses, with a sharp drop as the number of enrollments increases.
- A small number of users enroll in many courses, creating a long-tail effect.
- This suggests that most learners are selective, while a few explore widely.
- Helps in personalized recommendations by understanding user engagement patterns

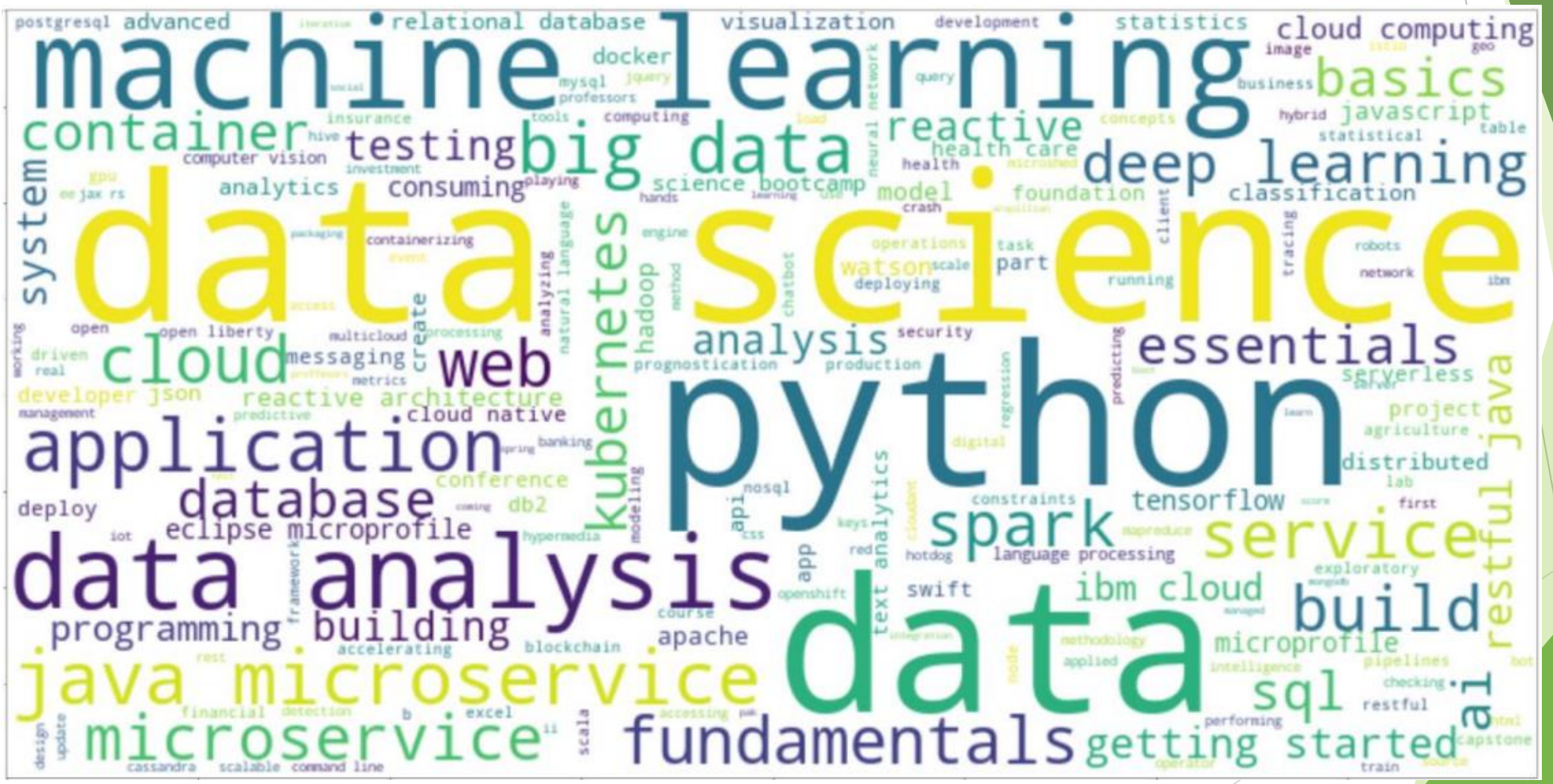


20 Most Popular Courses

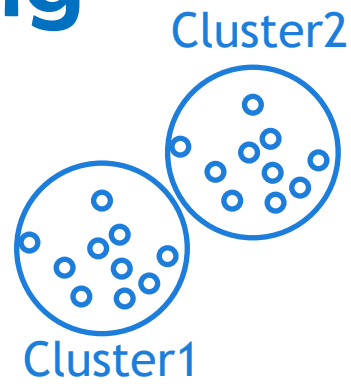
- The list showcases the top 20 courses based on enrollment or rating.
- Python for Data Science ranks highest, followed by Introduction to Data Science and Big Data 101.
- Data Science, Machine Learning, and Cloud Computing dominate the top spots.
- This insight helps refine recommendations by prioritizing highly sought-after courses.

	COURSE_ID	Rating	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

Word Cloud of Course Titles

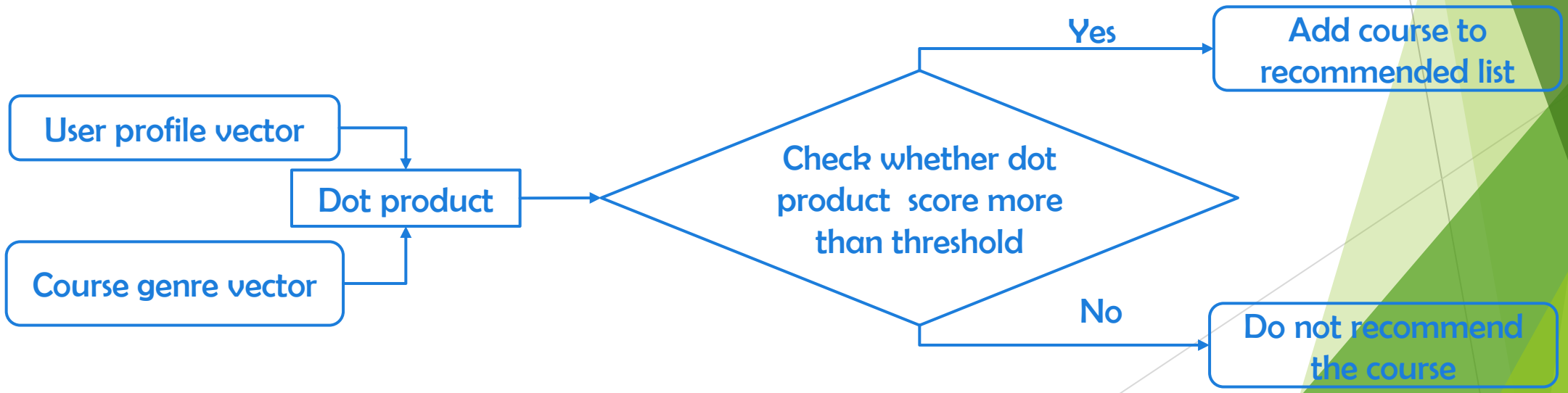


Content-based Recommender System using Unsupervised Learning



Flowchart of Content-Based Recommender System Using User Profile and Course Genres

- The recommender system compares user preferences with course genres using vector representations.
- Dot product calculation measures similarity between the user profile and course genres.
- If the similarity score exceeds a threshold, the course is recommended; otherwise, it is not recommended.



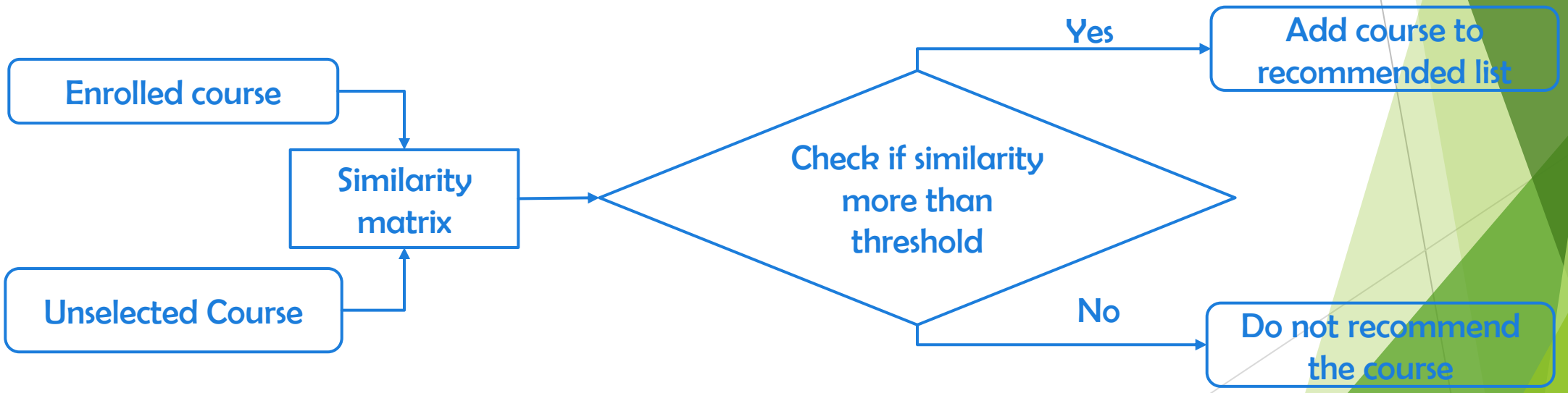
Evaluation Results of User Profile-Based Recommender System

- Threshold score: 10.0
- On average, 60.82 new courses have been recommended per test user.
- The top-10 commonly recommended courses across all test users are listed

```
COURSE_ID
TA0106EN      17390
excourse22    15656
excourse21    15656
GPXX0IBEN     15644
ML0122EN      15603
excourse06    15062
excourse04    15062
GPXX0TY1EN    14689
excourse73    14464
excourse72    14464
Name: count, dtype: int64
```

Flowchart of Content-Based Recommender System Using Course Similarity

- Uses a similarity matrix to compare enrolled and unselected courses
- Recommends courses if similarity exceeds the threshold

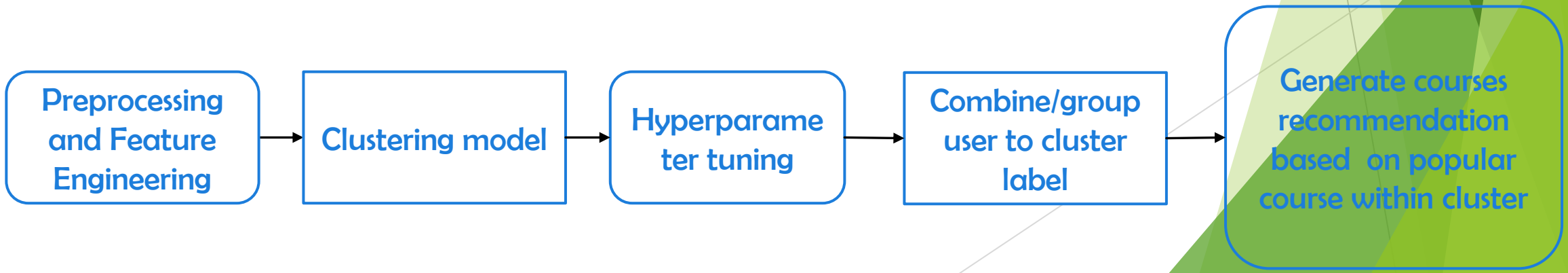


Evaluation Results of Course Similarity Based Recommender System

- Threshold for similarity : 0.6

Flowchart of Clustering-Based Recommender System

- User profile data is cleaned and transformed into numerical features relevant for clustering, such as course preferences, activity history, and demographics.
- An unsupervised learning algorithm (e.g., K-Means) groups similar users into clusters based on these features.
- The clustering model is optimized by adjusting parameters like the number of clusters to ensure accurate groupings.
- Each user is assigned to a specific cluster. Generate course based on popular course within cluster



Evaluation Results of Clustering-Based Recommender System

- Number of cluster : 14
- PCA component : 9

On average, how many new/unseen courses have been recommended per user

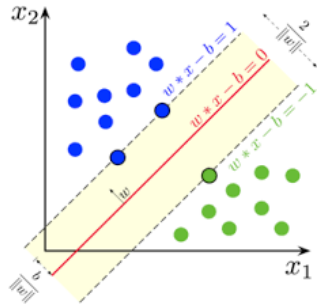
```
s = 0
for r in user_recommendations.values:
    s+=r[1:].sum()
avg=s/len(user_recommendations)
print(avg)
```

```
6.126869413881597
```

Top-10 commonly recommended courses

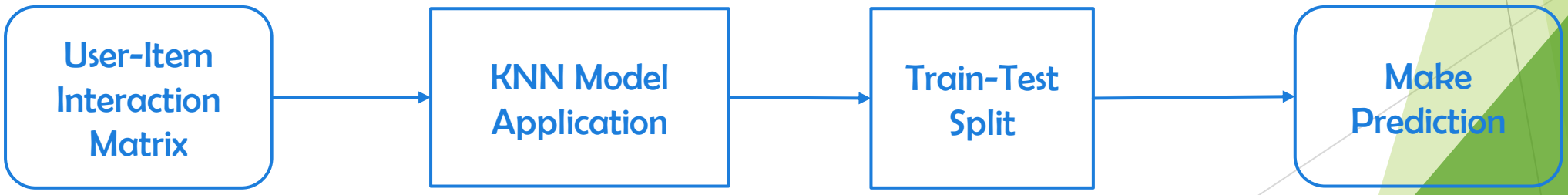
BD0101EN	20610
DS0101EN	18950
BD0111EN	17991
ST0101EN	15630
BC0201EN	10346
PY0101EN	9404
DS0105EN	8720
DAI101EN	8684
CL0101EN	8624
ML0115EN	8623

Collaborative-filtering Recommender System using Supervised Learning



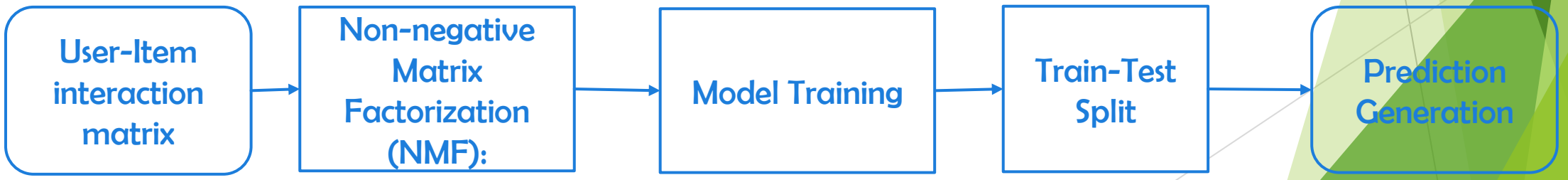
Flowchart of KNN Based Recommender System

- Captures the history of course enrollments by users.
- Implements the K-Nearest Neighbors algorithm to find similar users or courses based on enrollment patterns.
- Divides the data into training and testing sets to evaluate performance.
- Recommends courses to users based on similar users' enrollments.



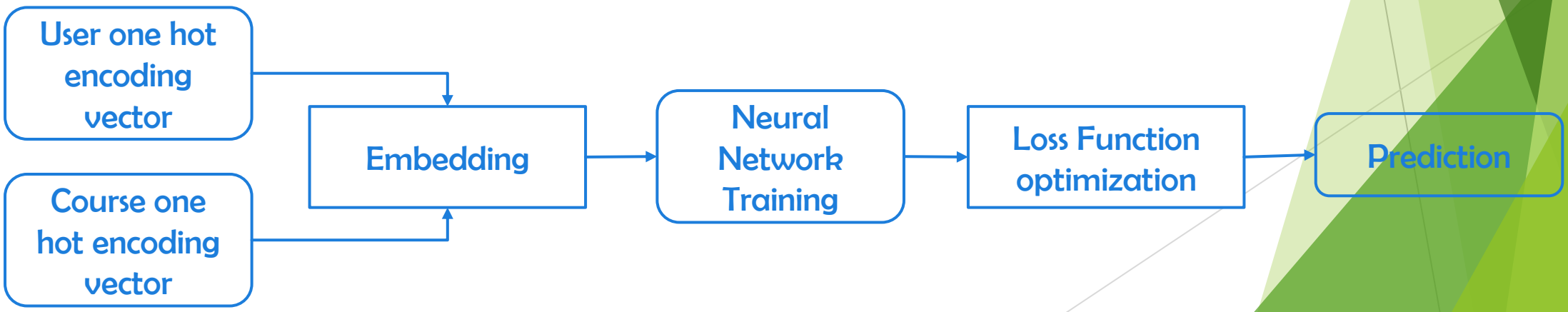
Flowchart of NMF Based Recommender System

- Constructed User-Item Interaction Matrix from course enrollments.
- Non-negative Matrix Factorization (NMF): Decomposes the matrix into user and item feature matrices.
- Model Training: Implemented using Scikit Surprise with hyperparameter tuning.
- Train-Test Split: Data is divided to validate predictions.
- Prediction Generation: Reconstructs missing values for personalized course recommendations.



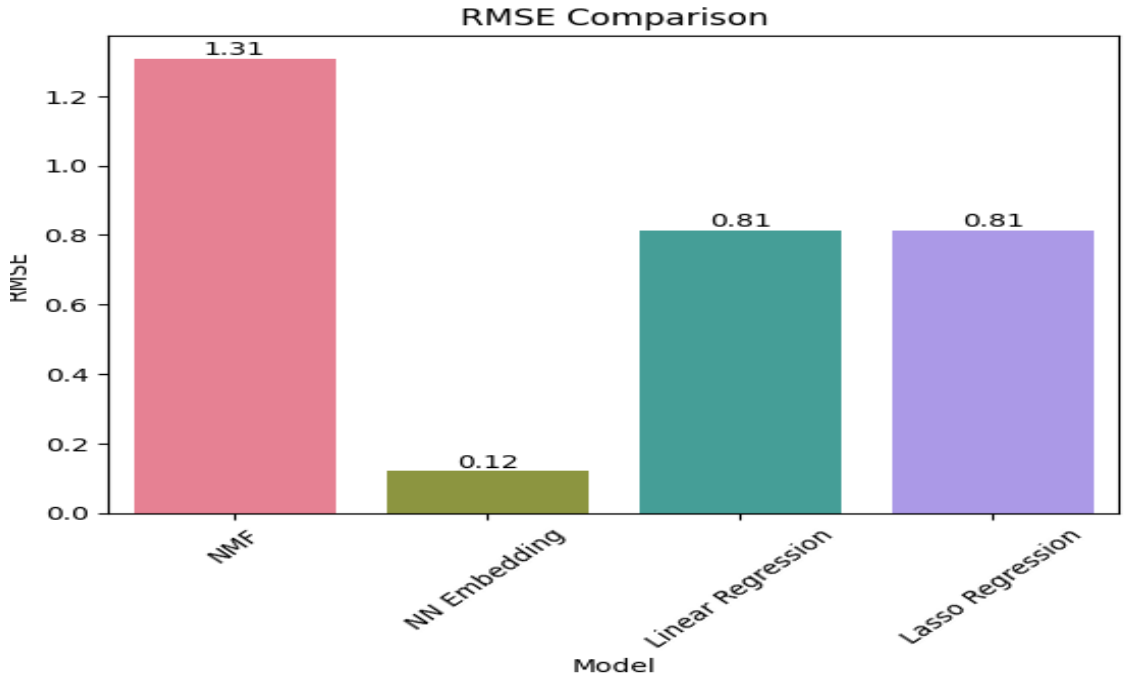
Flowchart of Neural Network Embedding Based Recommender System

- User one-hot encoding vector and Course one-hot encoding vector are created.
- Both vectors are passed into an Embedding layer that transforms them into dense, low-dimensional vectors.
- The embeddings are fed into a Neural Network Training process.
- A Loss Function optimization step minimizes the prediction error.
- Finally, the trained model generates Predictions of course recommendations for users.



Compare The Performance of Collaborative-Filtering Models

- This bar chart compares the RMSE of collaborative-filtering models: NMF, Neural Network Embedding, Linear Regression, and Lasso Regression.



Note: KNN model performance is not included as the training could not be completed due to memory constraints, causing a system crash.

Conclusions

- Developed multiple collaborative filtering models for personalized course recommendations.
- Neural Network Embedding model had the lowest RMSE, excelling in handling sparse data and complex interactions.
- NMF and regression models showed promising results.
- KNN model could not be evaluated due to memory constraints.
- Future work: Optimize KNN implementation, explore hybrid models, and enhance user profiling for better recommendations.

Appendix

- All material, notebook link available at

  https://github.com/Vasantan/IBM-Machine-Learning