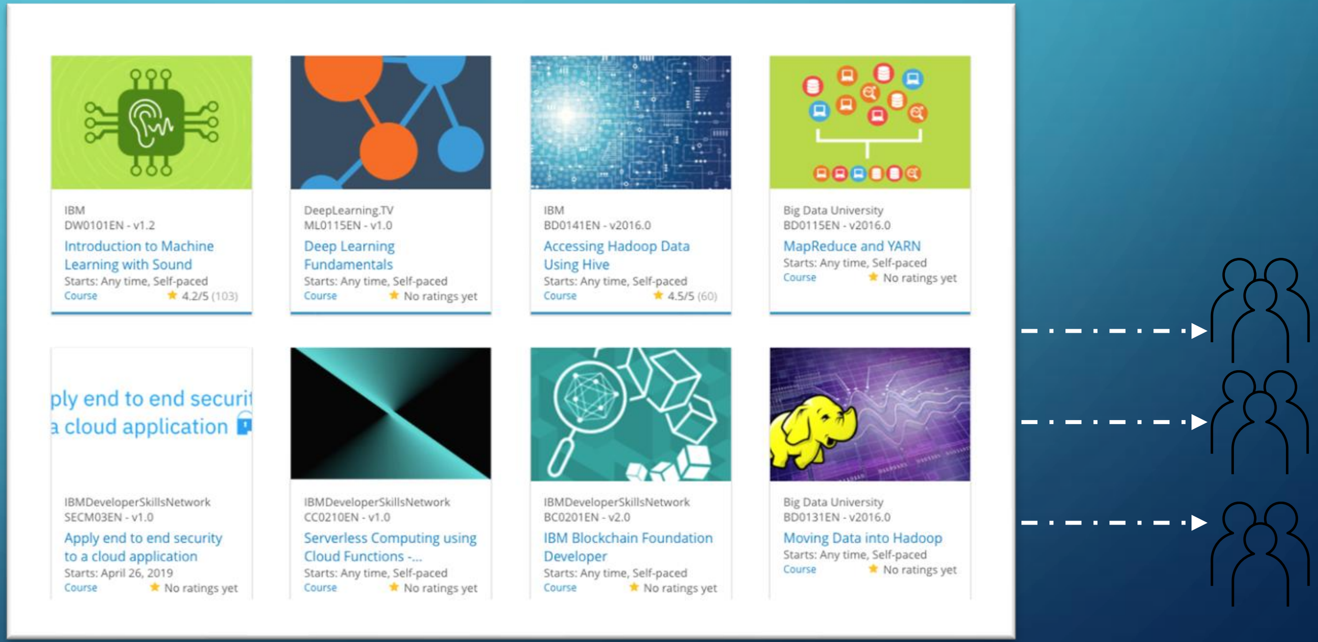
Build a Personalized Online Course Recommender System with Machine Learning



Carlos Madariaga
18/06/25

Outline

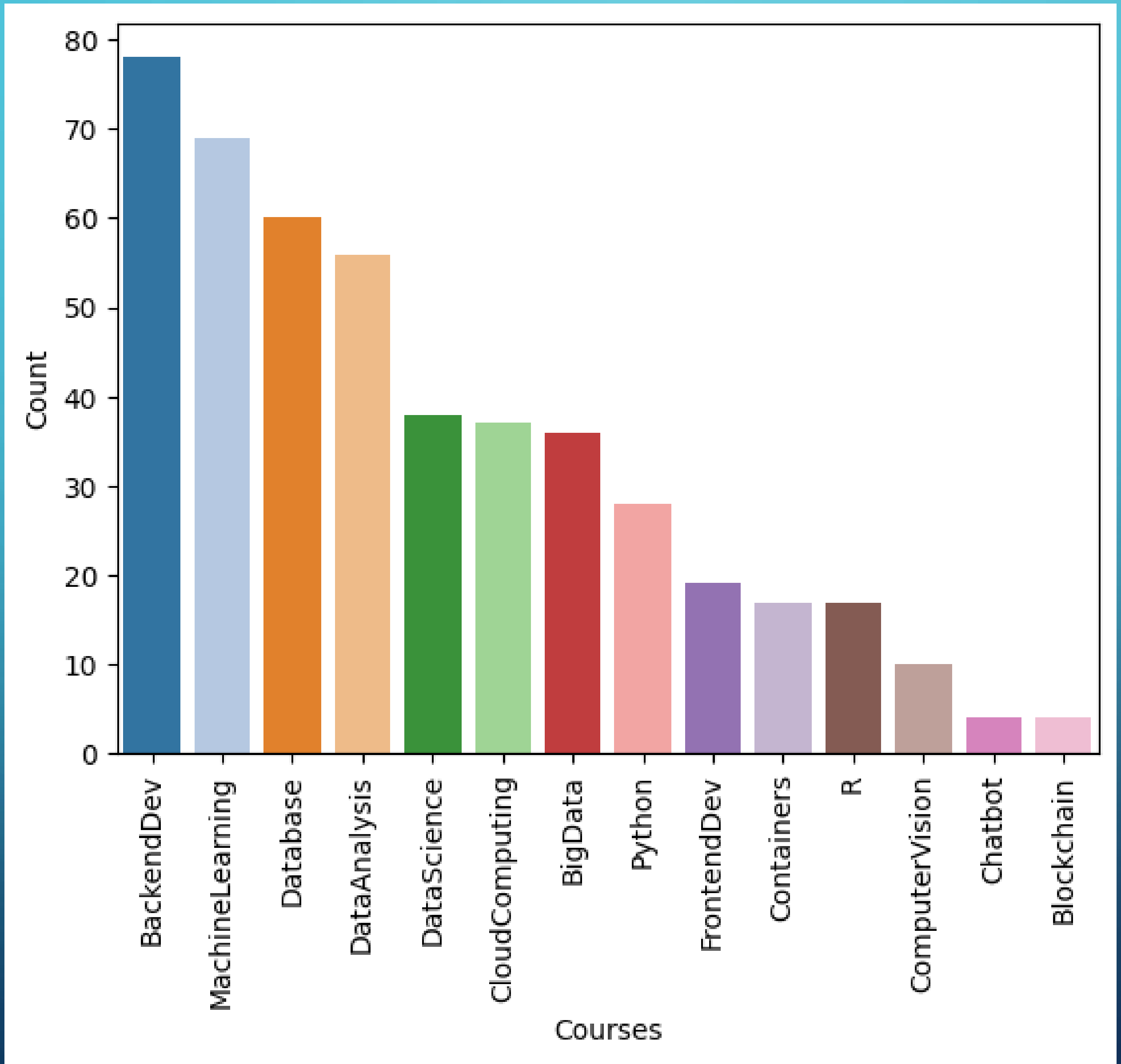
- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

Introduction

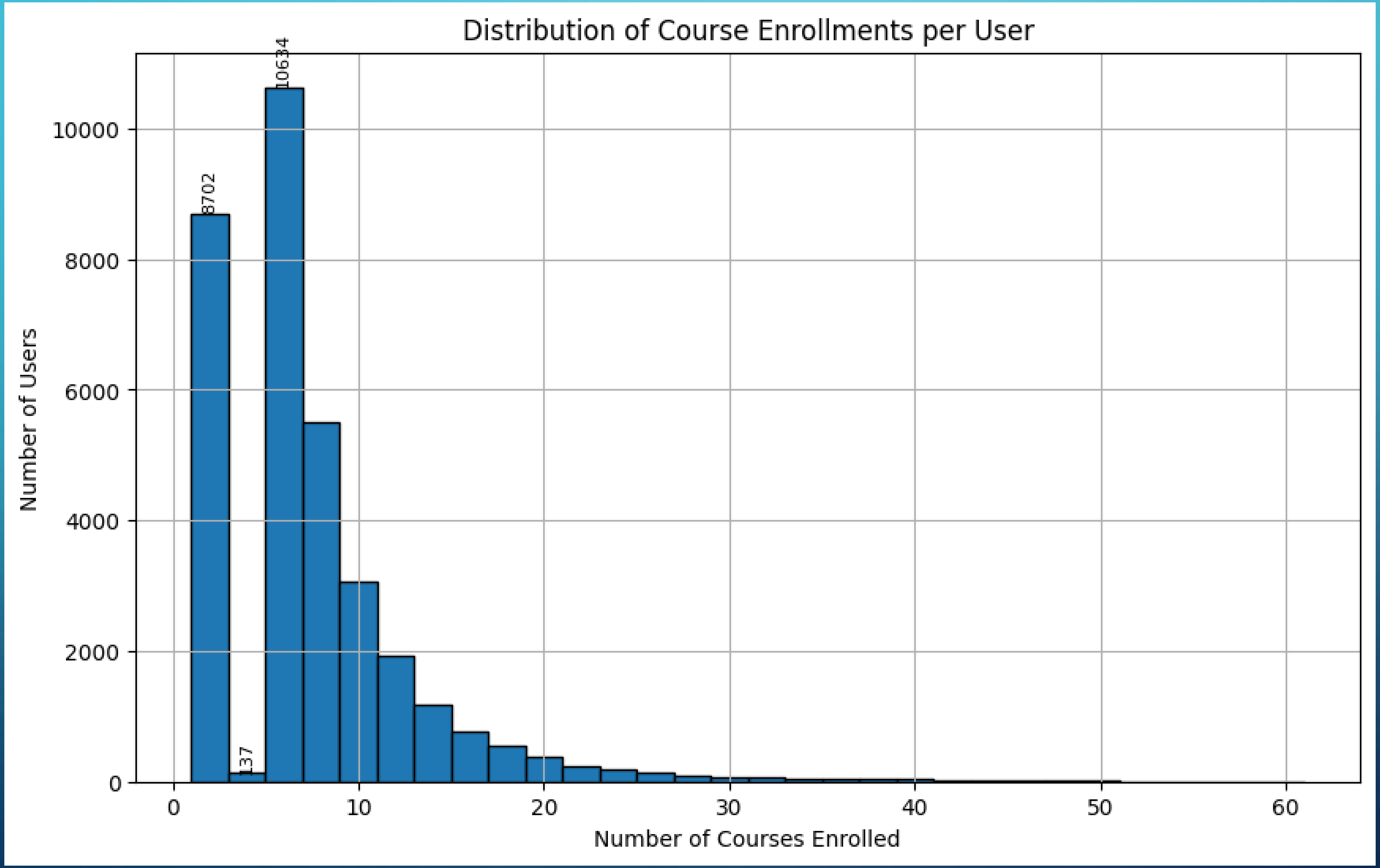
- **Project background and context:** This project develops a personalized course recommender system using machine learning techniques applied to user enrollments and course metadata, such as genres and titles.
- **Problem states and hypotheses:** The goal is to recommend relevant courses that users have not yet seen, based on their past enrollments and preferences.
- We hypothesize that:
  - H1: Users prefer courses similar in genre and title to those they've already taken.
  - H2: Similar users (based on course history) tend to enroll in similar future courses.
  - H3: Combining user clustering or latent factors improves recommendation diversity and relevance.

EXPLORATORY DATA ANALYSIS

COURSE COUNTS PER GENRE



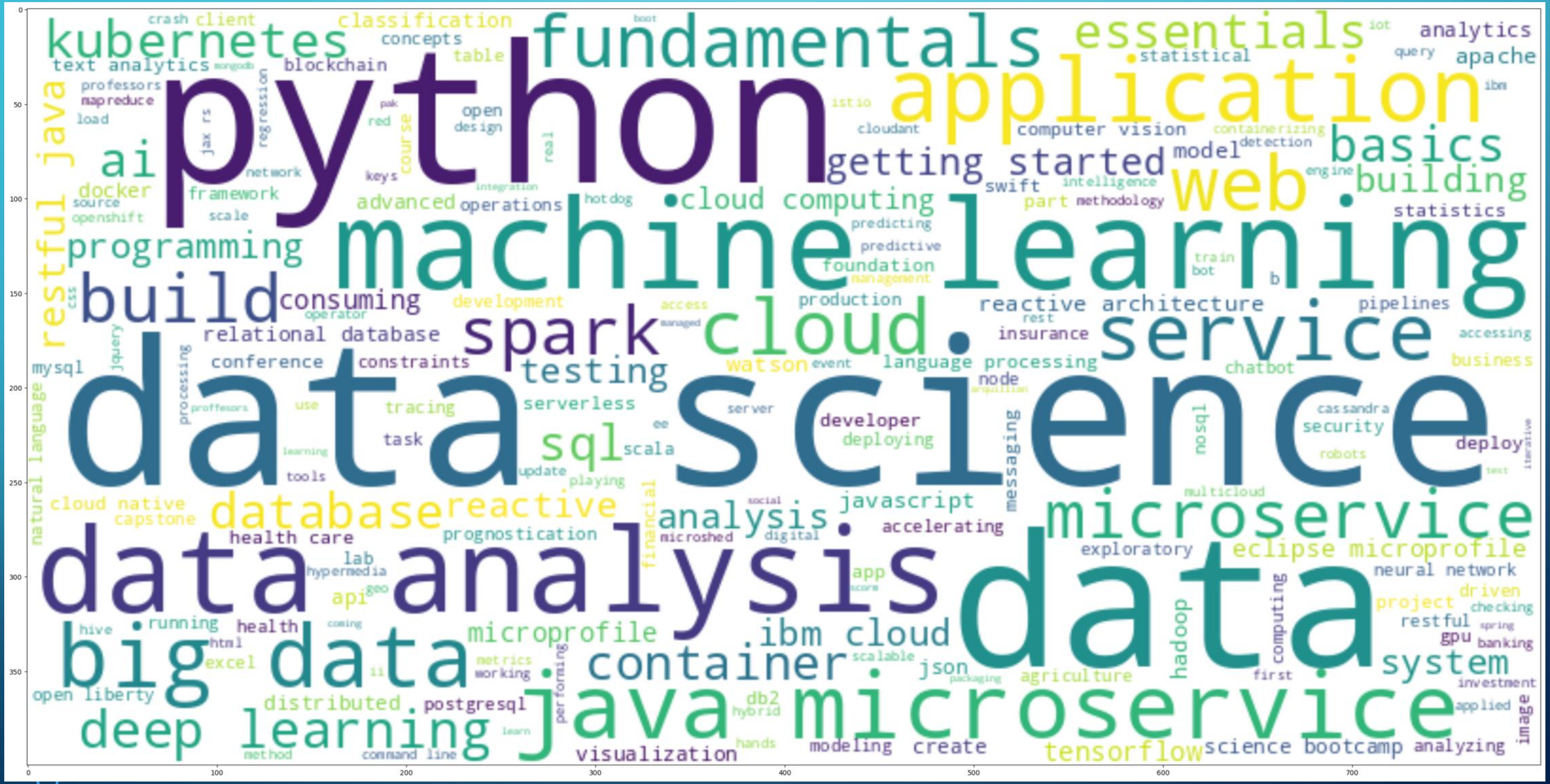
COURSE ENROLLMENT DISTRIBUTION



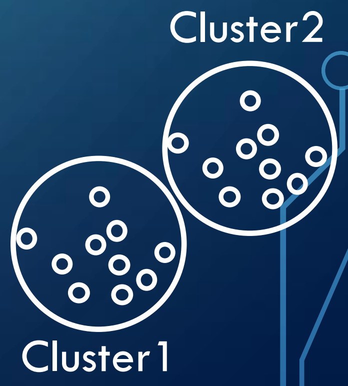
20 MOST POPULAR COURSES

	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

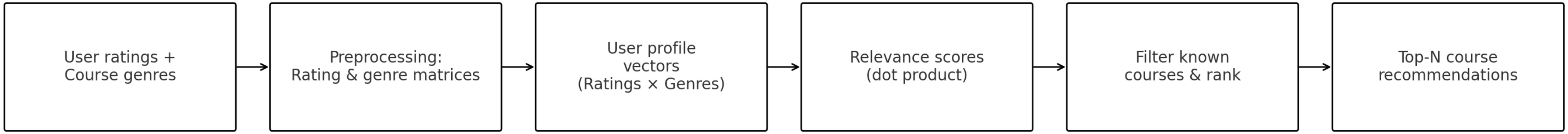
WORD CLOUD OF COURSE TITLES



CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING



FLOWCHART OF CONTENT-BASED RECOMMENDER SYSTEM USING USER PROFILE AND COURSE GENRES



EVALUATION RESULTS OF USER PROFILE-BASED RECOMMENDER SYSTEM

Place your hyper-parameter settings, such as recommendation score or course similarity thresholds, etc.

score threshold = 10.0 // number of courses threshold = top 20

On average, how many new/unseen courses have been recommended per user (in the test user dataset)

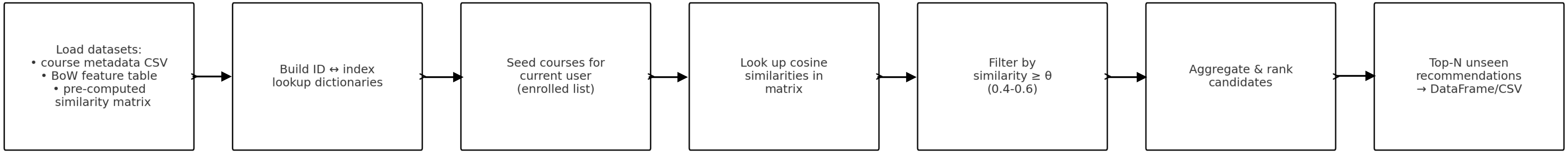
60.82 courses recommended on average

(limited to 20 at the end)

What are the most frequently recommended courses? Return the top-10 commonly recommended courses across all users

TA0106EN	**17390**
excourse21	**15656**
excourse22	**15656**
GPXX0IBEN	**15644**
ML0122EN	**15603**
excourse04	**15062**
excourse06	**15062**
GPXX0TY1EN	**14689**
excourse73	**14464**
excourse72	**14464**

FLOWCHART OF CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY



EVALUATION RESULTS OF COURSE SIMILARITY BASED RECOMMENDER SYSTEM

Your hyper-parameter settings, such as a score or similarity threshold

Similarity threshold ≥ θ (0.4 – 0.6) // number of courses threshold = top 20

On average, how many new/unseen courses have been recommended per user (in the test user dataset)

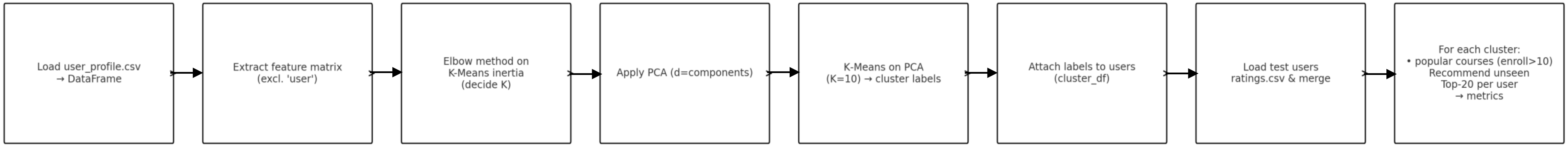
49.68 courses recommended on average

(limited to 20 at the end)

What are the most frequently recommended courses? Return the top-10 commonly recommended courses

excourse32	**27188**
excourse68	**26358**
excourse67	**25932**
excourse74	**25639**
excourse23	**25556**
excourse36	**25556**
excourse38	**24510**
excourse33	**24339**
excourse04	**24277**
DS0110EN	**23909**

FLOWCHART OF CLUSTERING-BASED RECOMMENDER SYSTEM



EVALUATION RESULTS OF CLUSTERING-BASED RECOMMENDER SYSTEM

Your hyper-parameter settings, such as a score or similarity threshold

K=10 // popularity threshold = 10 // max recommendations = 20

On average, how many new/unseen courses have been recommended per user (in the test user dataset)

74.65 courses recommended on average

(limited to 20 at the end)

What are the most frequently recommended courses? Return the top-10 commonly recommended courses

ML0109EN: 32852

CB0105ENv1: 32538

SC0103EN: 31822

CO0301EN: 31557

PA0101EN: 31282

CL0101EN: 30883

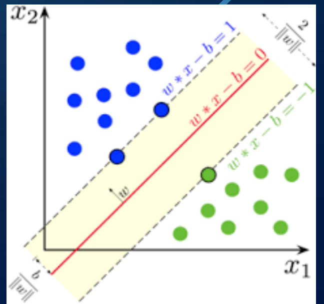
ML0120ENv2: 30612

BD0131EN: 30544

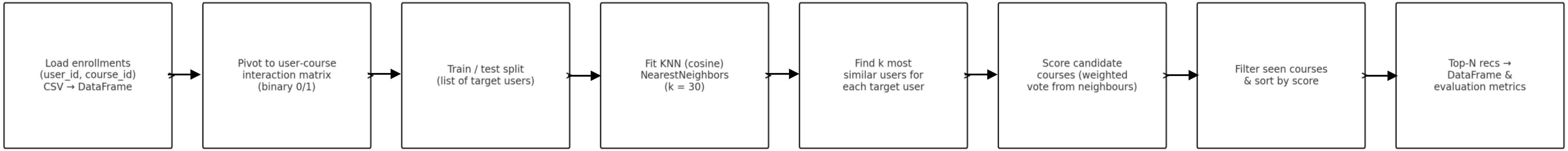
DS0301EN: 30257

BD0153EN: 29428

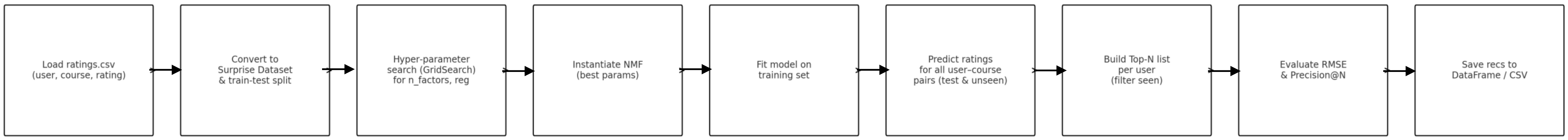
COLLABORATIVE-FILTERING RECOMMENDER SYSTEM USING SUPERVISED LEARNING



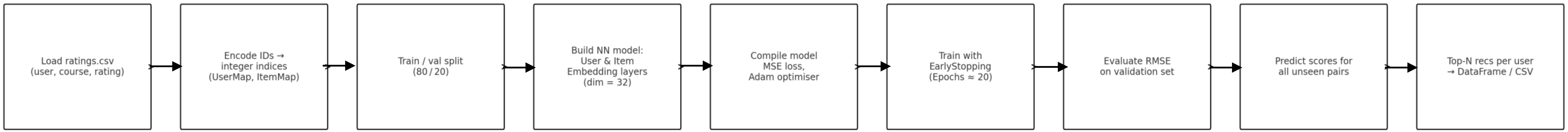
FLOWCHART OF KNN BASED RECOMMENDER SYSTEM



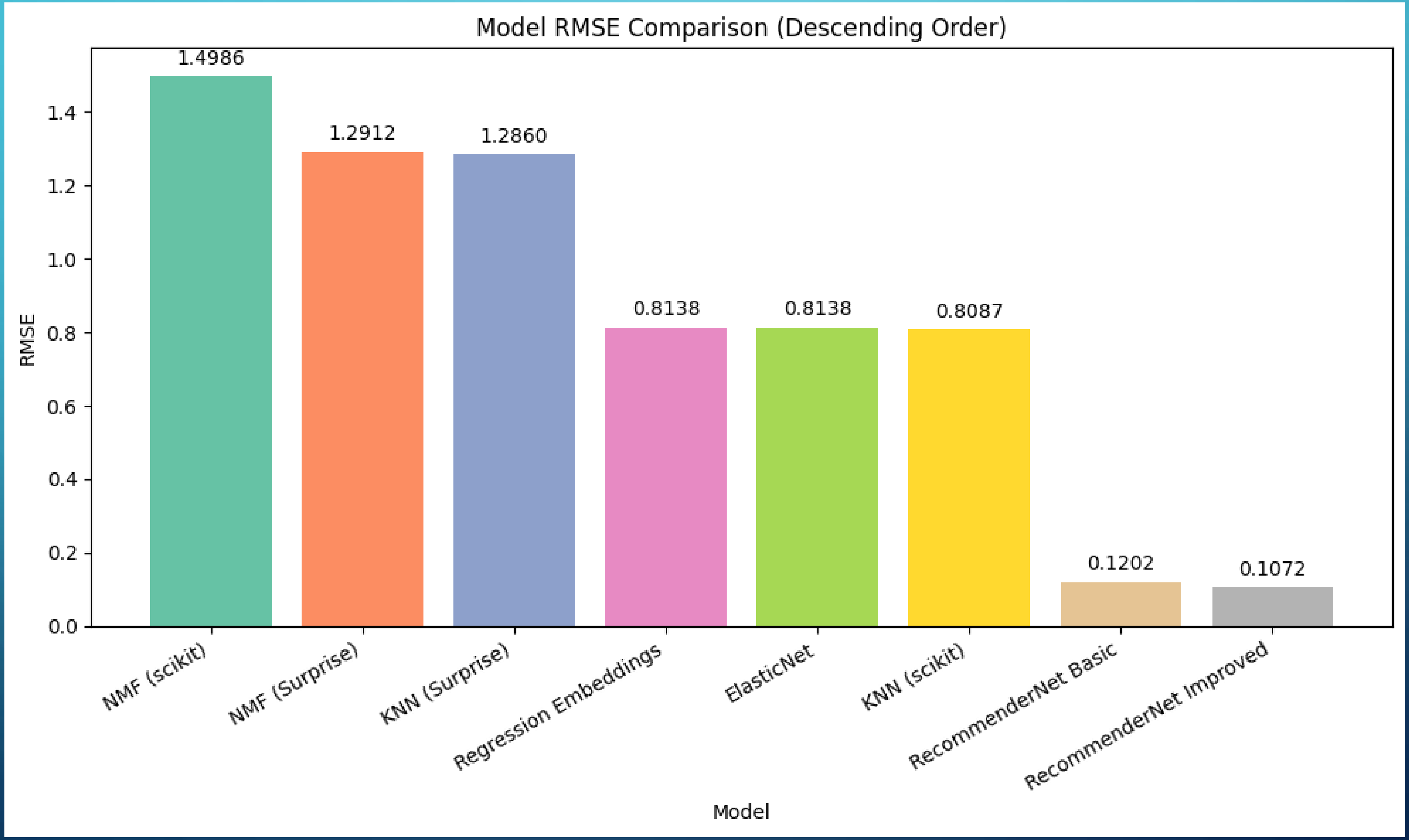
FLOWCHART OF NMF BASED RECOMMENDER SYSTEM



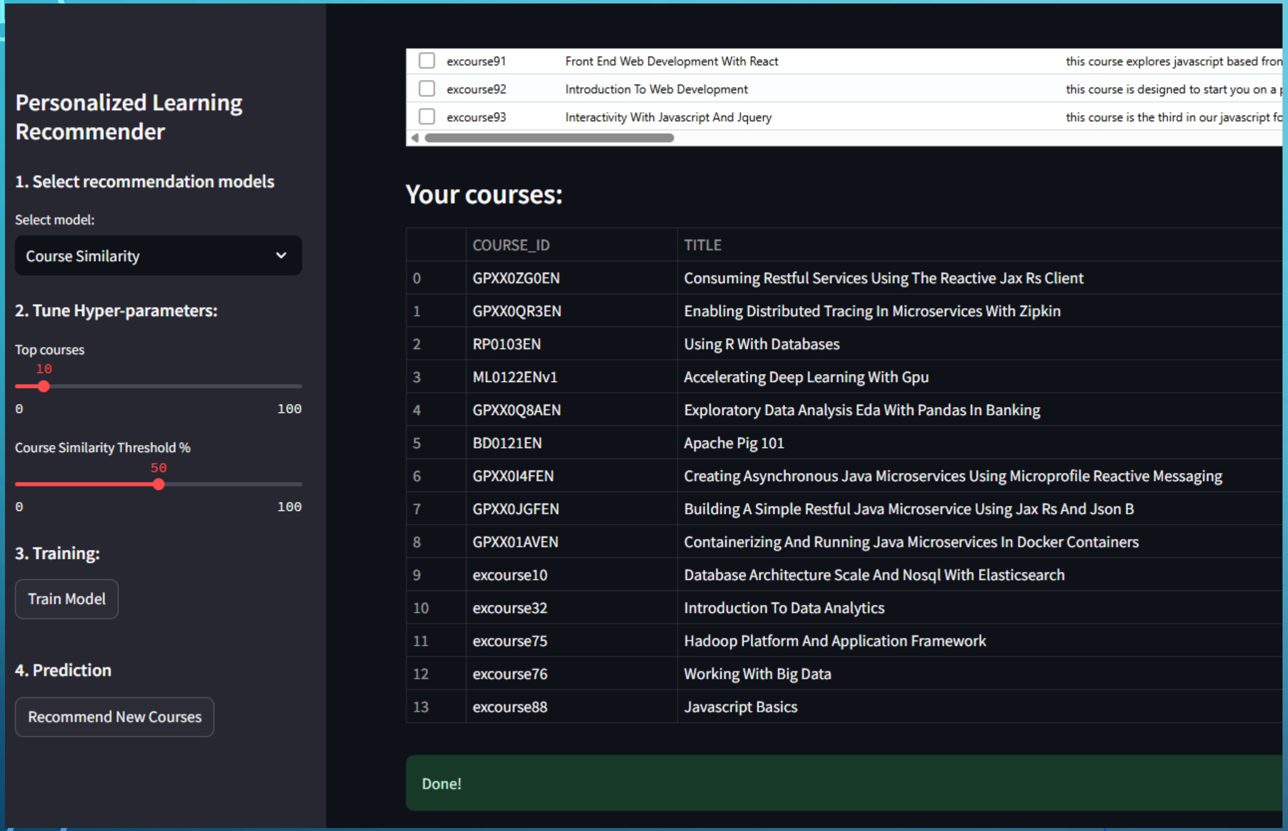
FLOWCHART OF NEURAL NETWORK EMBEDDING BASED RECOMMENDER SYSTEM

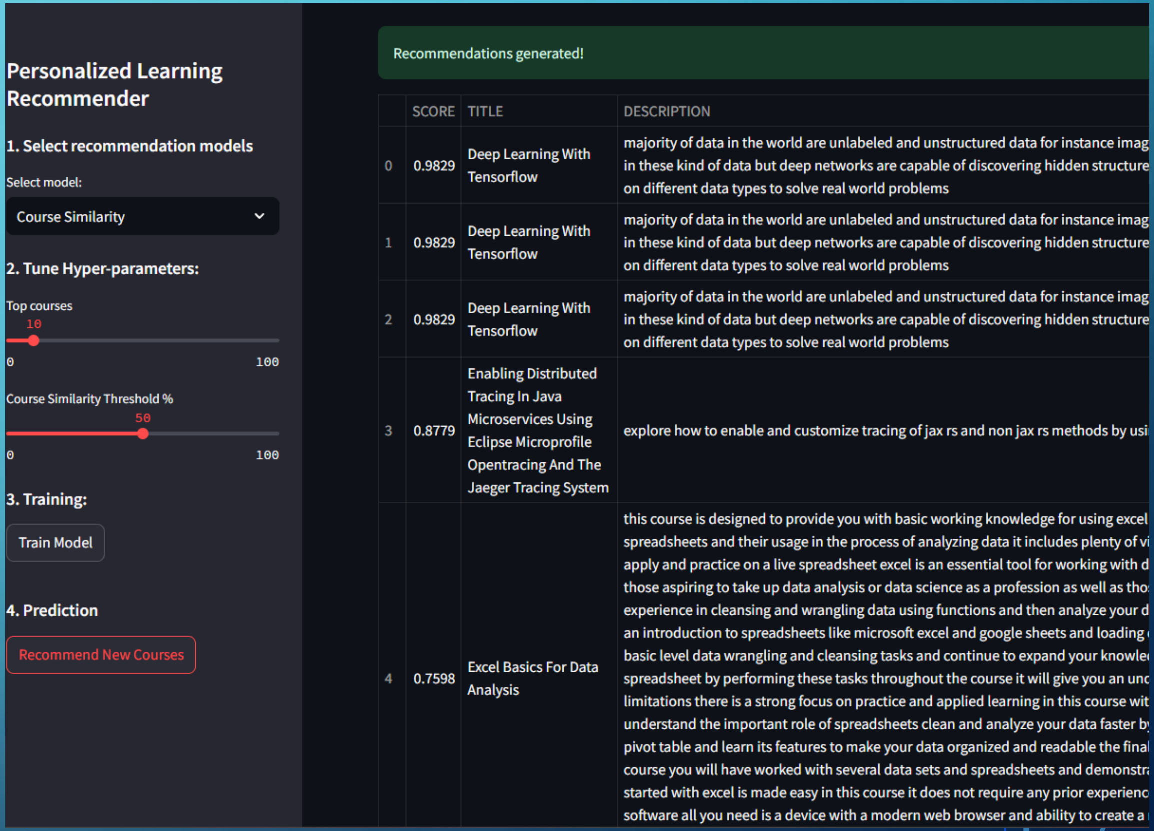


COMPARE THE PERFORMANCE OF COLLABORATIVE-FILTERING MODELS



OPTIONAL: BUILD A COURSE RECOMMENDER SYSTEM APP WITH STREAMLIT





A published Streamlit App URL for a live demo: **http://141.63.130.224:8501**

Conclusions

1. **Clear performance hierarchy**
   Classical latent-factor methods (NMF & K-NN from both scikit-learn and Surprise) sit on the left with RMSE between **1.50 and 0.81**.
   Deep-learning approaches (RecommenderNet) dominate the right side, slashing RMSE to **≈0.11–0.12**.
2. **RecommenderNet improvements matter**
   The “Improved” variant shaves roughly **0.013** RMSE off the “Basic” version (0.1072 vs 0.1202). Given RMSE’s squared-error nature, that is a meaningful lift.
3. **Traditional models vary widely**
   - **NMF (scikit)** performs worst (1.50) – almost 40 % higher error than the Surprise NMF implementation (1.29).
   - K-NN fares better than NMF but still lags elastic-net and deep models.
4. **Regularised linear ensemble is mid-pack**
   ElasticNet and the “Regression Embeddings” baseline (both ≈0.814) cut error roughly in half vs. NMF, yet still cannot match the neural models.
5. **Overall conclusion**
   Switching from matrix-factorisation or memory-based methods to a neural embedding architecture yields an *order-of-magnitude* reduction in rating prediction error on this dataset. Fine-tuning that network (the “Improved” variant) bags the last bit of accuracy and leads the leaderboard.

Appendix

- GitHub URL: https://github.com/carlosmada22/IBM_machine_learning
- Streamlit app URL: http://141.63.130.224:8501/Asset 4
- …

GRADING CRITERIA

- ✓ Uploaded your completed presentation in PDF format (2 pts)
- ✓ Completed the required Introduction slide (4 pt)
- ✓ Completed the required Exploratory Data Analysis slides (8 pts)
- ✓ Completed the required content-based recommender system using user profile and course genres slides (6 pts)
- ✓ Completed the required content-based recommender system using course similarity slides (6 pts)
- ✓ Completed the required content-based recommender system using user profile clustering slides (6 pts)
- ✓ Completed the required KNN-based collaborative filtering slide (6 pts)
- ✓ Completed the required NMF-based collaborative filtering slide (6 pts)
- ✓ Completed the required neural network embedding based collaborative filtering slide (6 pts)
- ✓ Completed the required collaborative filtering algorithms evaluation slides (6 pts)
- ✓ Completed the required Conclusion slide (6 pts)
- ✓ Applied your creativity to improve the presentation beyond the template (4 pts)
- ✓ Displayed any innovative insights (4 pts)