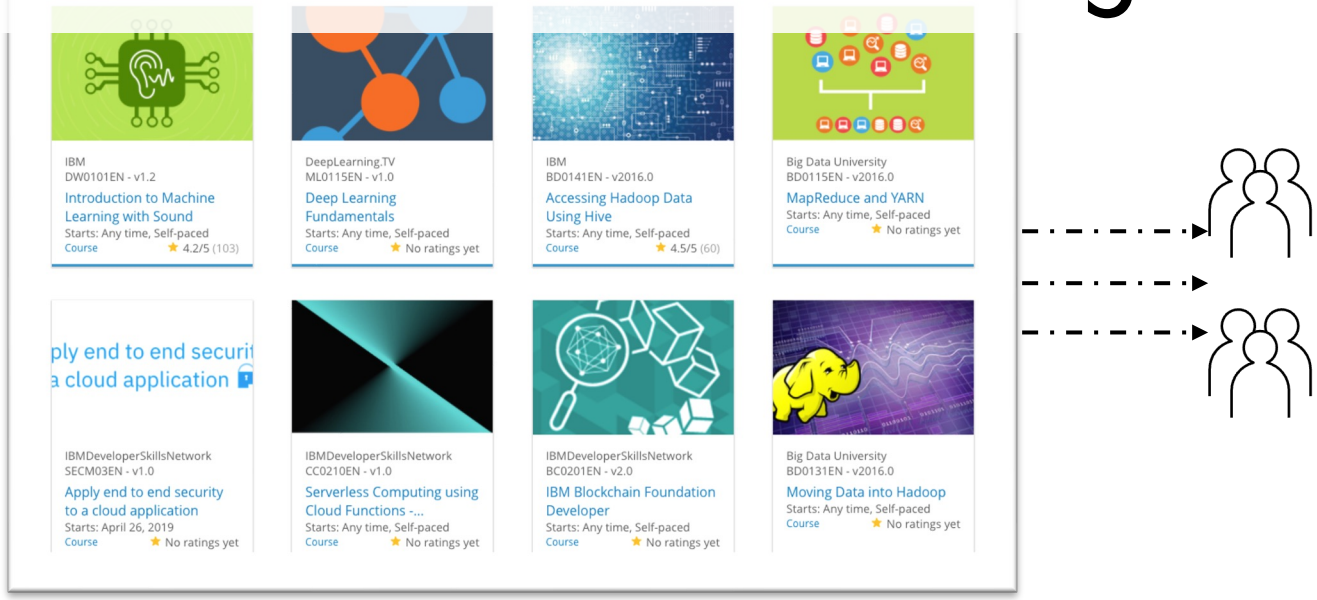
Build a Personalized Online Course Recommender System with Machine Learning

Bruno Tenorio
May 2024



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

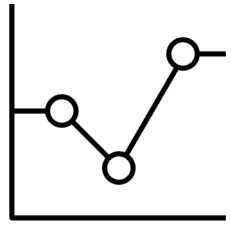
- AI Training Room is a Massive Open Online Courses (MOOCs) startup that grows rapidly and reaches millions of learners in a very short period.
- The main goal of this project is to improve learners' learning experience via helping them quickly find new interesting courses and better paving their learning paths.
- This project is currently at the Proof of Concept (PoC) phase so our main focus at this moment is to explore and compare various machine learning models and find one with the best performance in off-line evaluations.

The analysis is based on two provided datasets: *course ratings, course processed* and *course genres*

The jupyter notebooks that support this analysis are given below in the following order:

- P1_EDA
- P2_Content_Similarity_BoW
- P3_Content_User_Profile_Recommender_System
- P4_Content_clustering_Recommender_System
- P5_Recommender_System_with_Surprise_library
- P6_NN_Colaborative_Recommender_System

Exploratory Data Analysis



Course genre Dataset

- The cell shows the first 5 rows of the course genre dataset.
- The dataset contains information about 307 courses.
- We can see that we have 14 course genres spread among some popular topics related to machine learning, databases, app developments, etc.
- The **COURSE_ID** and **TITLE** are str datatypes and all the course genres are binary/int datatypes. Any genre column with value 1 means the course is associated with the course genre while 0 means the course is not.

	COURSE_ID	TITLE	Database	Python	CloudComputing	DataAnalysis	Containers	MachineLearning	ComputerVision	DataScience	BigData	Chatbot	R	BackendDev	FrontendDev	Blockchain
0	ML0201EN	robots are coming build iot apps with watson ...	0	0	0	0	0	0	0	0	0	0	0	1	1	0
1	ML0122EN	accelerating deep learning with gpu	0	1	0	0	0	1	0	1	0	0	0	0	0	0
2	GPXX0ZG0EN	consuming restful services using the reactive ...	0	0	0	0	0	0	0	0	0	0	0	1	1	0
3	RP0105EN	analyzing big data in r using apache spark	1	0	0	1	0	0	0	0	1	0	1	0	0	0
4	GPXX0Z2PEN	containerizing packaging and running a sprin...	0	0	0	0	1	0	0	0	0	0	0	1	0	0

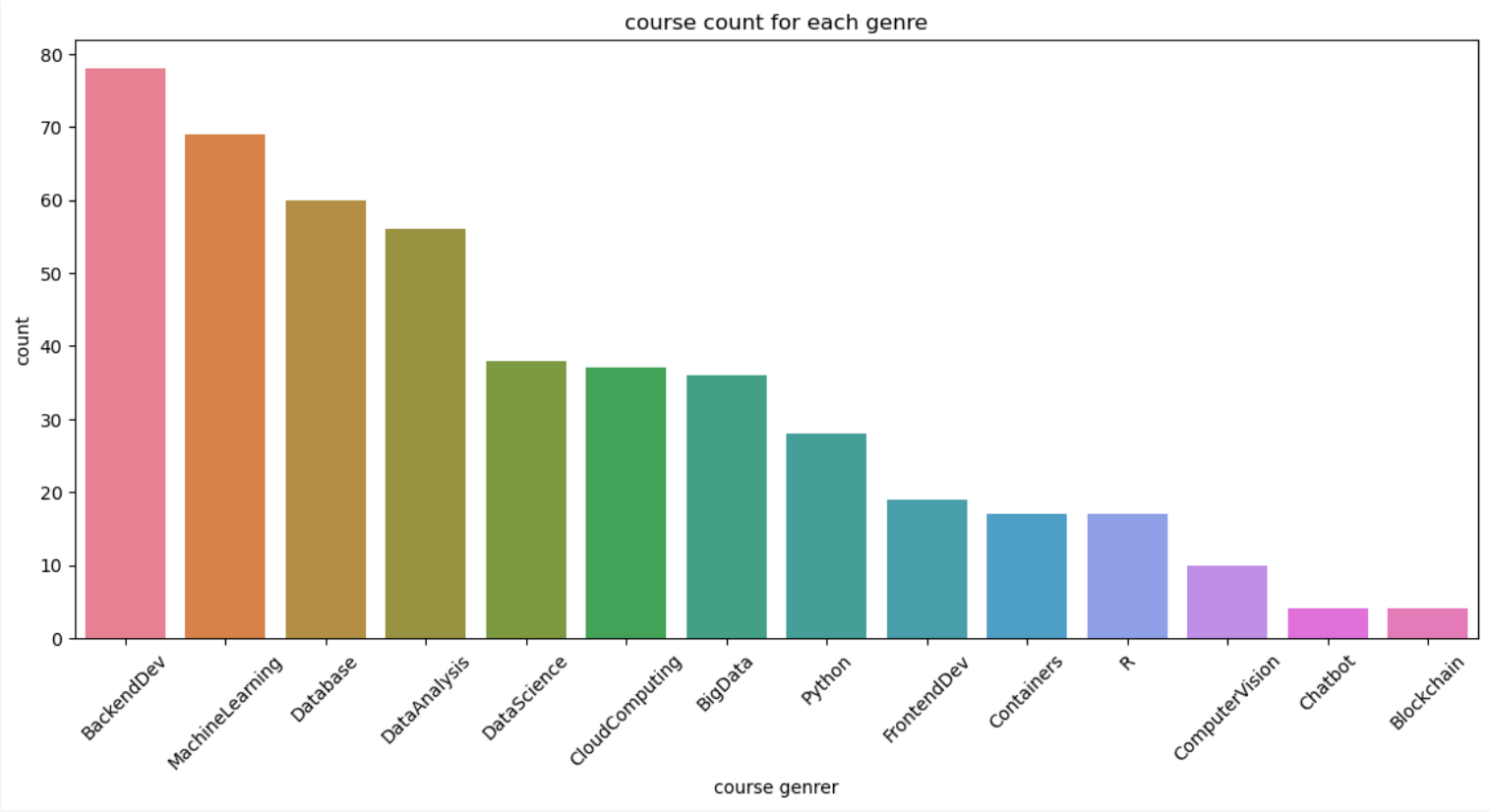
Course ratings Dataset

- The course ratings dataset contains three columns, **user** representing a unique user id, **item** representing a course id, and **rating** representing the ratings given by the user.
- The dataset contains 233306 rows (enrollments).
- The rating columns have only three categorical values: 3, 4, and 5.

	user	item	rating
0	1889878	CC0101EN	5
1	1342067	CL0101EN	3
2	1990814	ML0120ENv3	5
3	380098	BD0211EN	5
4	779563	DS0101EN	3
5	1390655	ST0101EN	5
6	367075	DS0301EN	3
7	1858700	CC0101EN	4
8	600100	BD0211EN	3
9	623377	DS0105EN	3

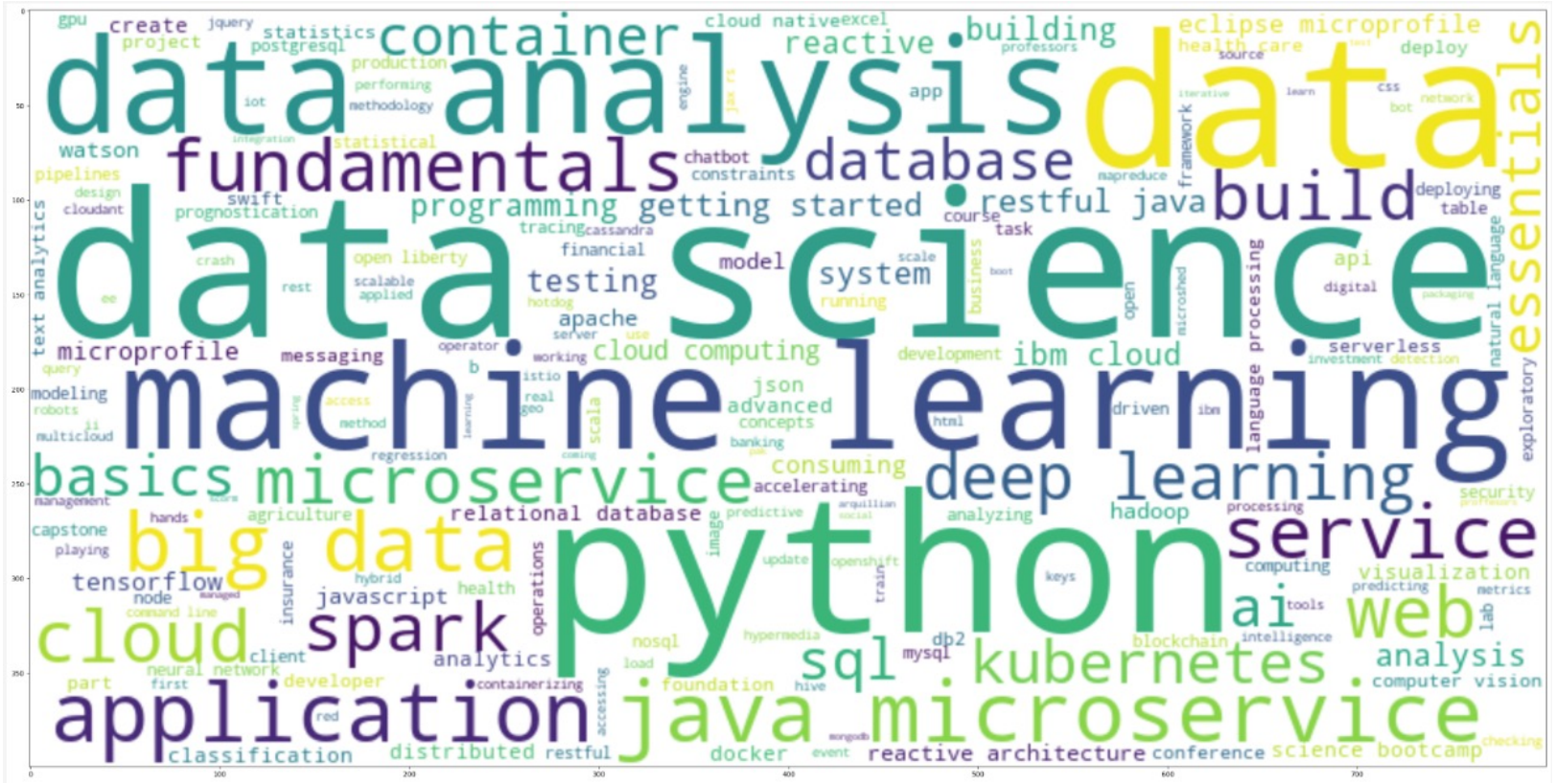
Course counts per genre

The bar plot shows the most common genres from all courses.



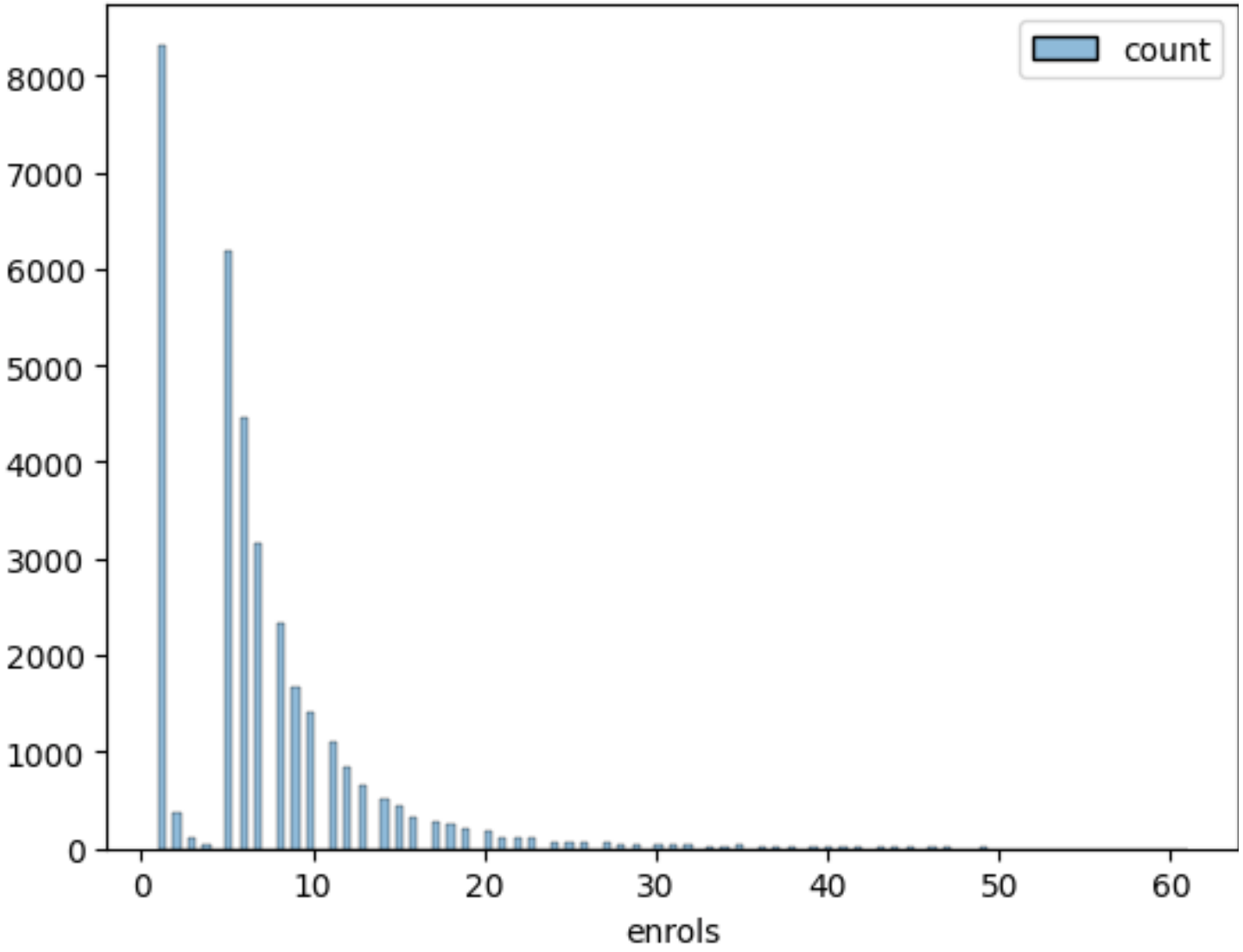
Word cloud of course titles

- The word cloud shows the most common words weighted from their frequence of appearance in the course titles.



Course enrollment distribution

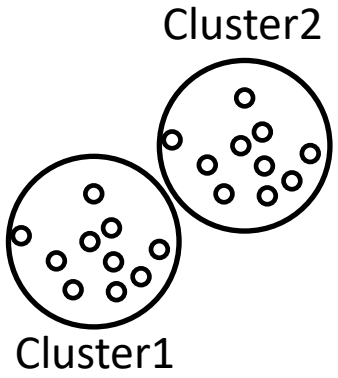
- The enrollment distributions exemplify how many users rated just 1 item or how many rated 10 items, etc.
- We see that about 8000 users rated only one course.



20 most popular courses

	COURSE_ID	enrolls	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

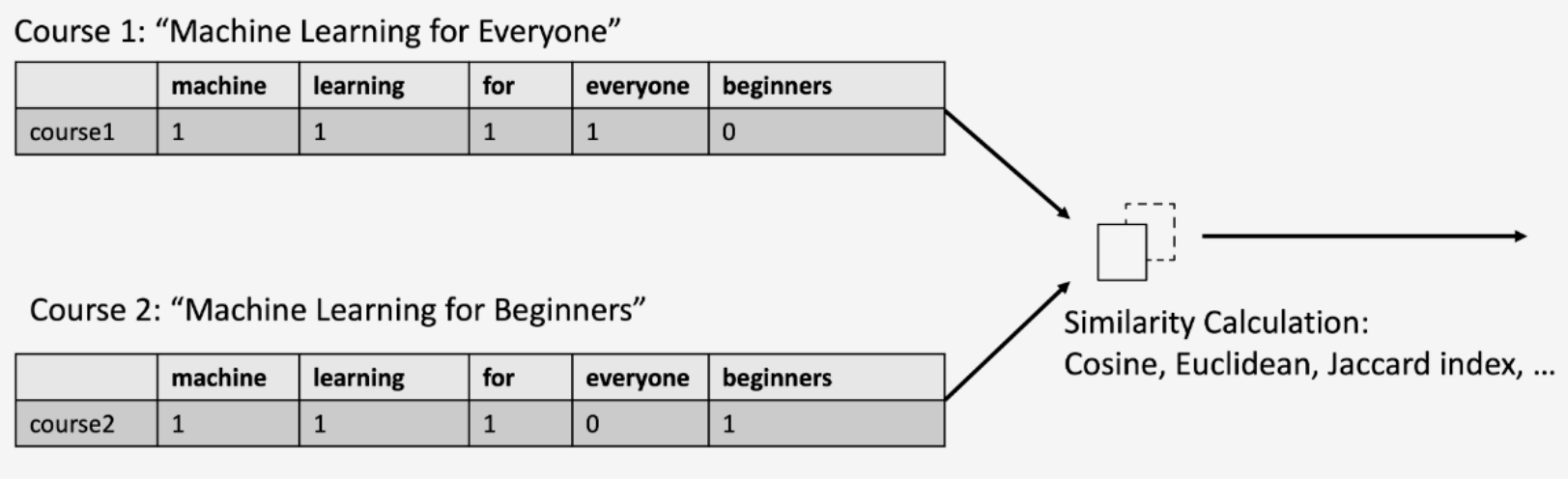
Content-based Recommender System using Unsupervised Learning



Flowchart of content-based recommender system using course similarity

See on notebook: **P2_Content_Similarity_BoW**

- Content-based recommender system is based on the similarity score computed by each pair of courses.



Evaluation results of content-based recommender system using course similarity

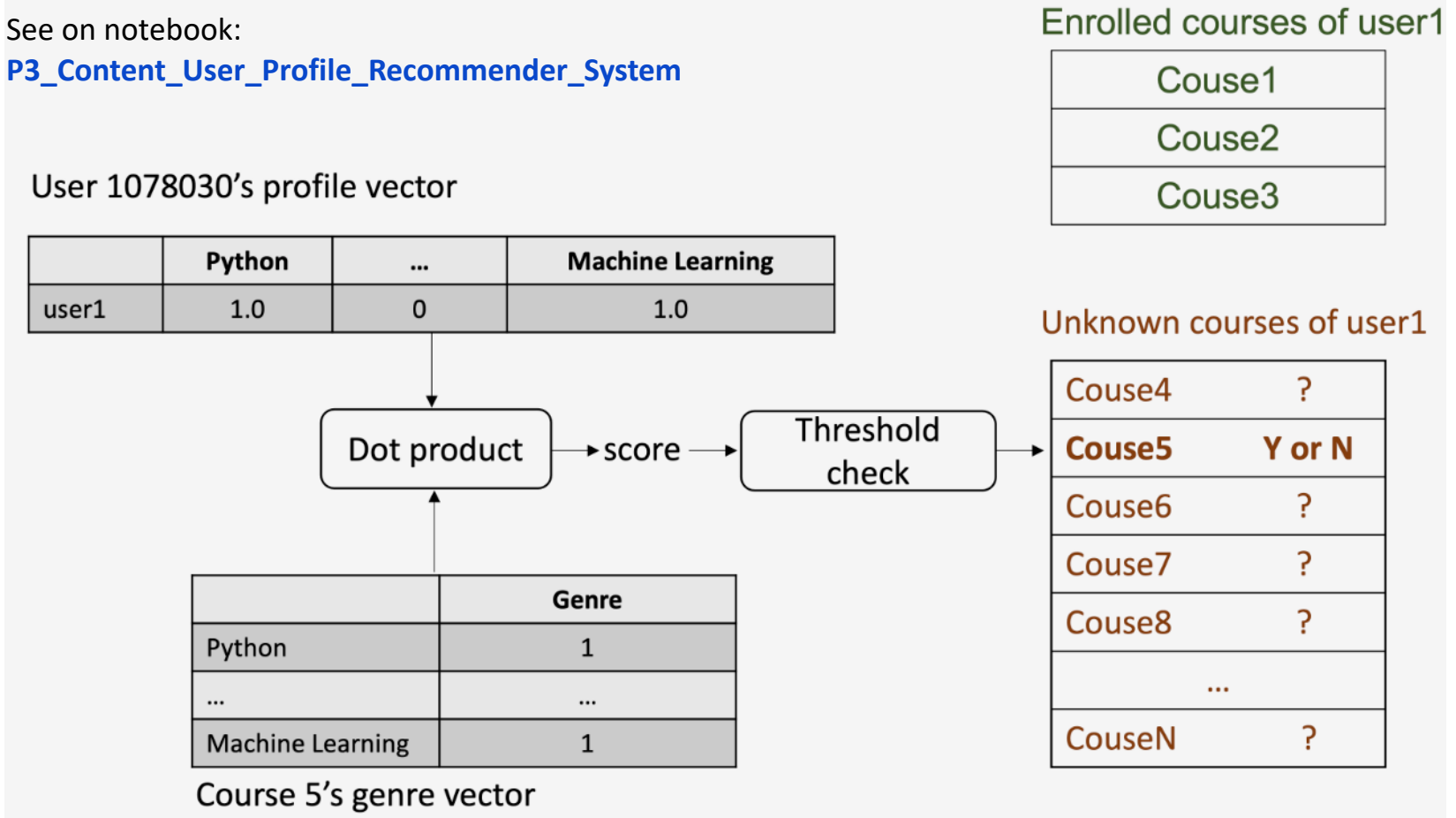
- With a similarity score threshold of 0.5, on average, around **1** new/unseen course has been recommended per user (in the test user dataset).
- The list of the 10 most recommended courses is listed below.

See on notebook: **P2_Content_Similarity_BoW**

	COURSE_ID	Count	TITLE
0	TMP107	245	data science bootcamp with python
1	DS0110EN	143	data science with open data
2	DA0151EN	65	data analysis using r 101
3	DX0106EN	61	data science bootcamp with r for university proffesors
4	DS0201EN	61	end to end data science on cloudpak for data
5	TMP0106	58	data science bootcamp
6	DS0107	58	data science career talks
7	WA0103EN	58	watson analytics for social media
8	CB0101EN	36	build your own chatbots
9	TMP0101EN	29	text analysis

Flowchart of content-based user-profile recommender system

See on notebook:
P3_Content_User_Profile_Recommender_System



- The user profile matrix can be combined with the Course genre matrix to create a user profile data frame.
- User profile vector dot products with Course genre matrix to generate a score.
- Using a threshold check, we can use decide on recommending a course or not.

Evaluation results of user profile-based recommender system

- With a score threshold for new course recommendations set to 20, on average, around **17** new/unseen courses have been recommended per user (in the test user dataset) with a user-profile-based recommender system.
- The list of the 10 most recommended courses is listed below.

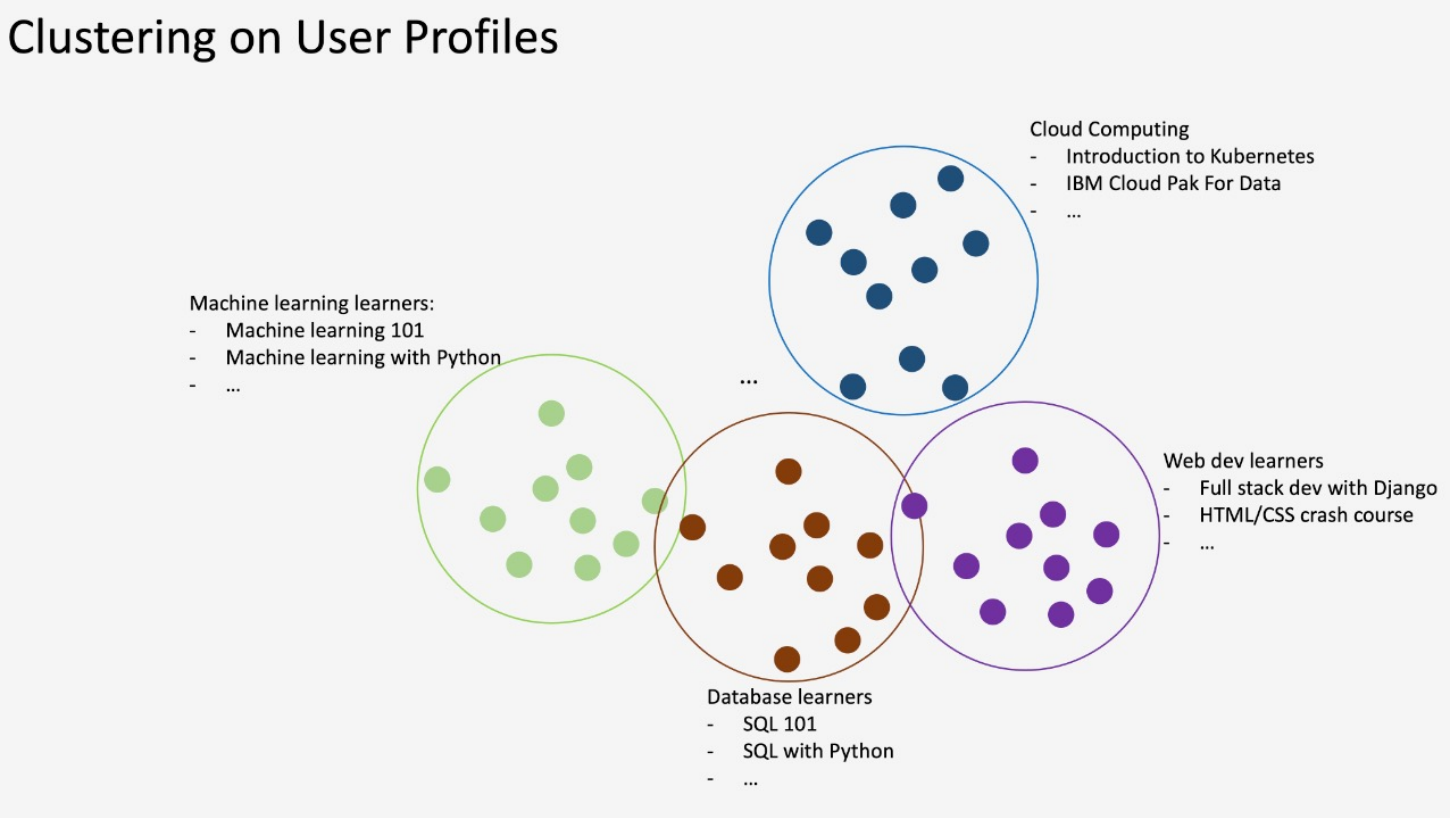
See on notebook:
P3_Content_User_Profile_Recommender_System

	COURSE_ID	Count	TITLE
0	TA0106EN	379	text analytics at scale
1	ML0122EN	351	accelerating deep learning with gpu
2	RP0105EN	343	analyzing big data in r using apache spark
3	TMP0105EN	341	getting started with the data apache spark ma...
4	SC0103EN	306	spark overview for scala analytics
5	ML0101EN	304	machine learning with python
6	BD0212EN	299	spark fundamentals ii
7	DX0108EN	251	data science bootcamp with python for universi...
8	TMP107	251	data science bootcamp with python
9	BD0143EN	245	using hbase for real time access to your big data

Flowchart of clustering-based recommender system

- A clustering-based recommender system consists of grouping similar content in clusters.
- A user who interacts with an item belonging to a cluster will be recommended items from the cluster.

See on notebook:
P4_Content_clustering_Recommender_System



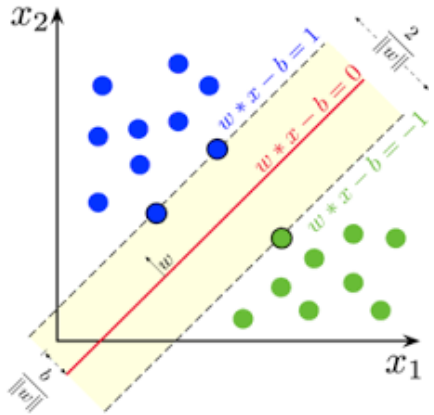
clustering-based recommender system

- With PCA we reduced the features' dimensionality to 8 and optimized K-Means with 8 clusters.
- On average, 11 new/unseen courses have been recommended per user (in the test user dataset) with a clustering-based recommender system.
- The top 10 commonly recommended courses are:

See on notebook:
P4_Content_clustering_Recommender_System

	COURSE_ID	Count	TITLE
0	DS0103EN	677	data science methodology
1	BD0111EN	628	hadoop 101
2	BD0211EN	617	spark fundamentals i
3	ML0115EN	579	deep learning 101
4	DA0101EN	571	data analysis with python
5	DS0105EN	567	data science hands on with open source tools
6	BD0101EN	564	big data 101
7	PY0101EN	533	python for data science
8	DS0101EN	533	introduction to data science
9	ST0101EN	510	statistics 101

Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system

- A KNN based recommender system using course enrollments history works similarly K-Means clustering.
- Finding nearest neighbors are based on similarity measurements among users or items with big similarity matrices.

See on notebook:
P5_Recommender_System_with_Surprise_library

User-Item interaction matrix

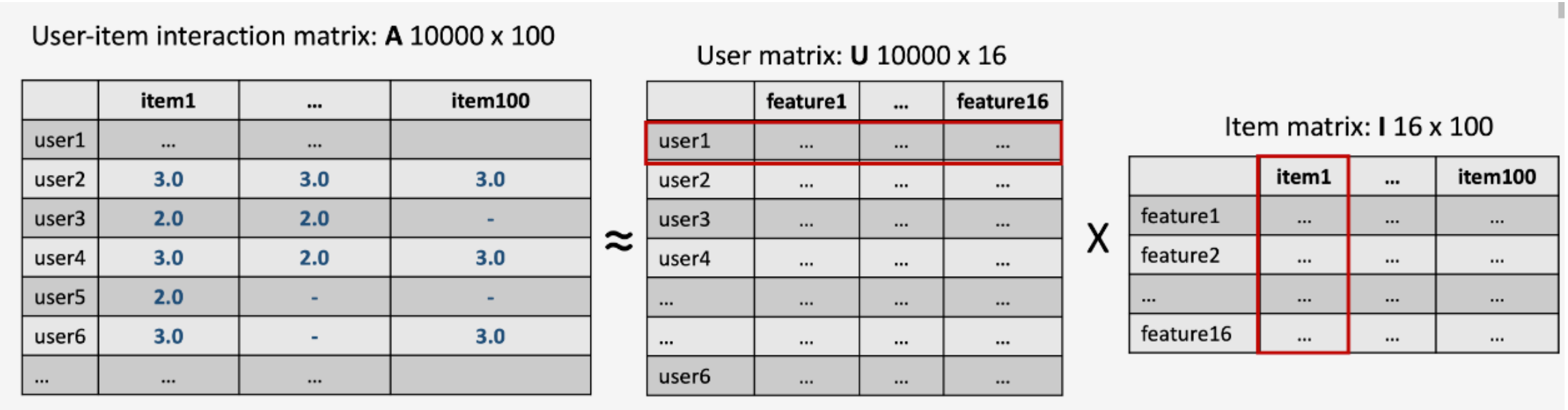
	Machine Learning With Python	Machine Learning 101	Machine Learning Capstone	SQL with Python	Python 101
...	...	...	...	...	
user2	3.0	3.0	3.0	3.0	3.0
user3	2.0	3.0	3.0	2.0	
user4	3.0	3.0	2.0	2.0	3.0
user5	2.0	3.0	3.0		
user6	3.0	3.0	?		3.0
...	...	...	...	...	

Similar users

Flowchart of NMF based recommender system

- The main idea is to decompose the big and sparse user-interaction into two smaller dense matrices, one represents the transformed user features and another represents the transformed item features.
- The idea here is when we multiply the row j of U and column k of matrix I, we can get an estimation to the original rating.

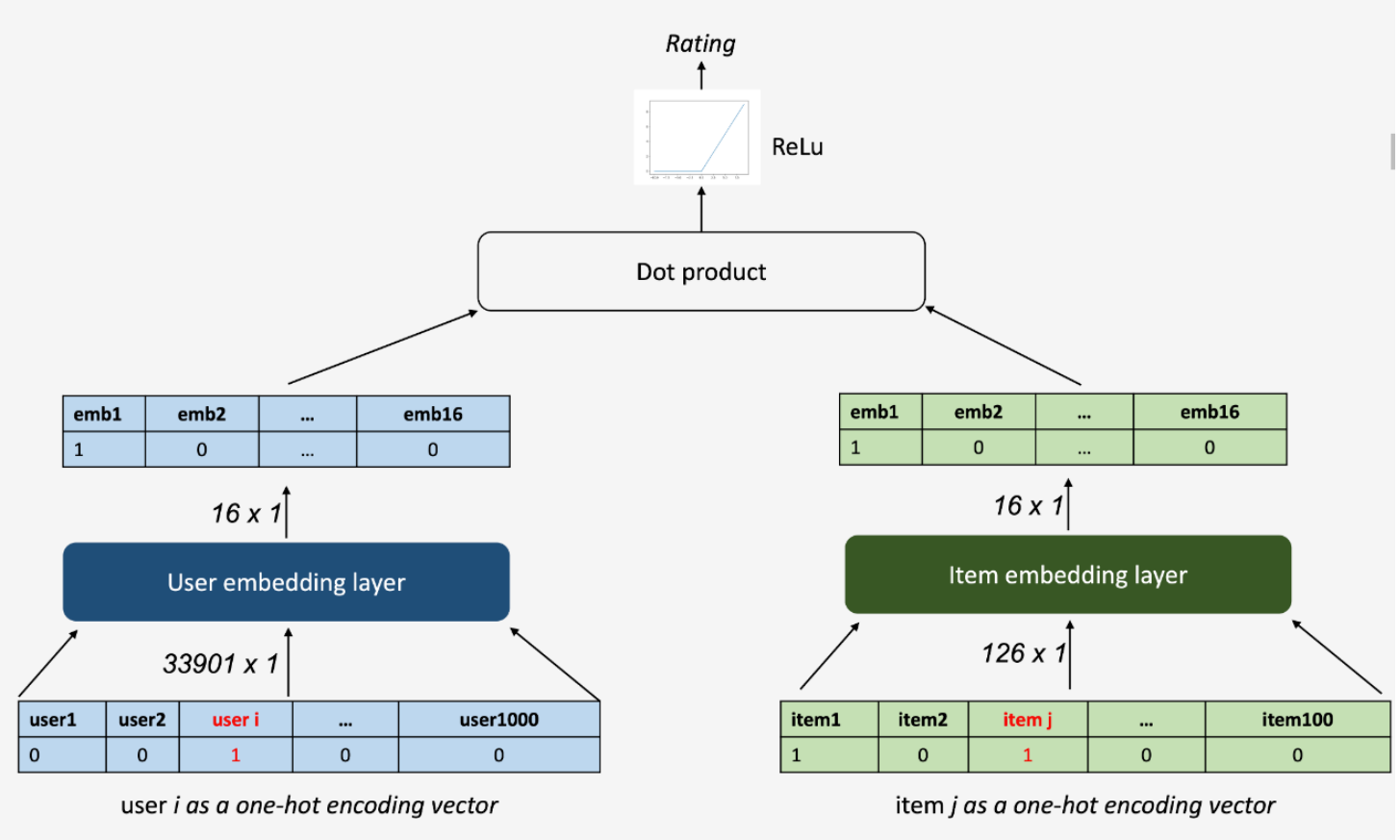
See on notebook:
P5_Recommender_System_with_Surprise_library

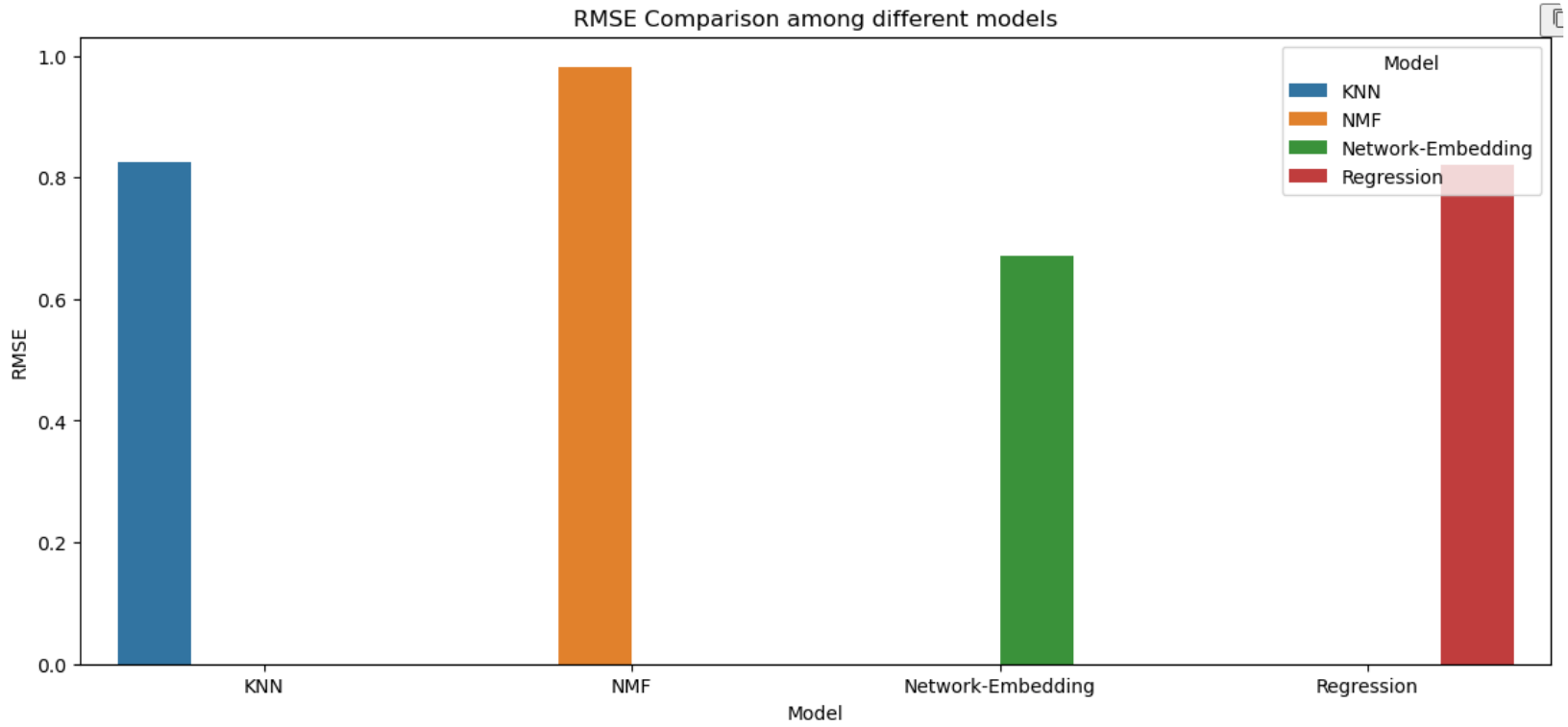


Flowchart of Neural Network Embedding based recommender system

- Non-negative Matrix Factorization decomposes the user-item interaction matrix into user matrix and item matrix, which contain the latent features of users and items.
- Neural networks can also be used to extract the latent user and item features.
- The embedding layer outputs two embedding vectors, which are similar to Non-negative matrix factorization. Then we could simply dot product the user and item embedding vector to output a rating estimation.

See on notebook:
P6_NN_Colaborative_Recommender_System





Compare the performance of collaborative-filtering models

- As we can see, Network-Embedding shows the lowest RMSE score among all tested models.

Conclusions

- The EDA was concluded where we extracted the most popular genders and courses.
- From content-based recommender system using user profiles and course genres, we have made recommendations based on user or course similarities.
- From the Collaborative-filtering Recommender System using Supervised Learning, we predicted the ratings on a user-item interaction matrix test set.
- The network-embedding model has given the best evaluation metric score on the test set.