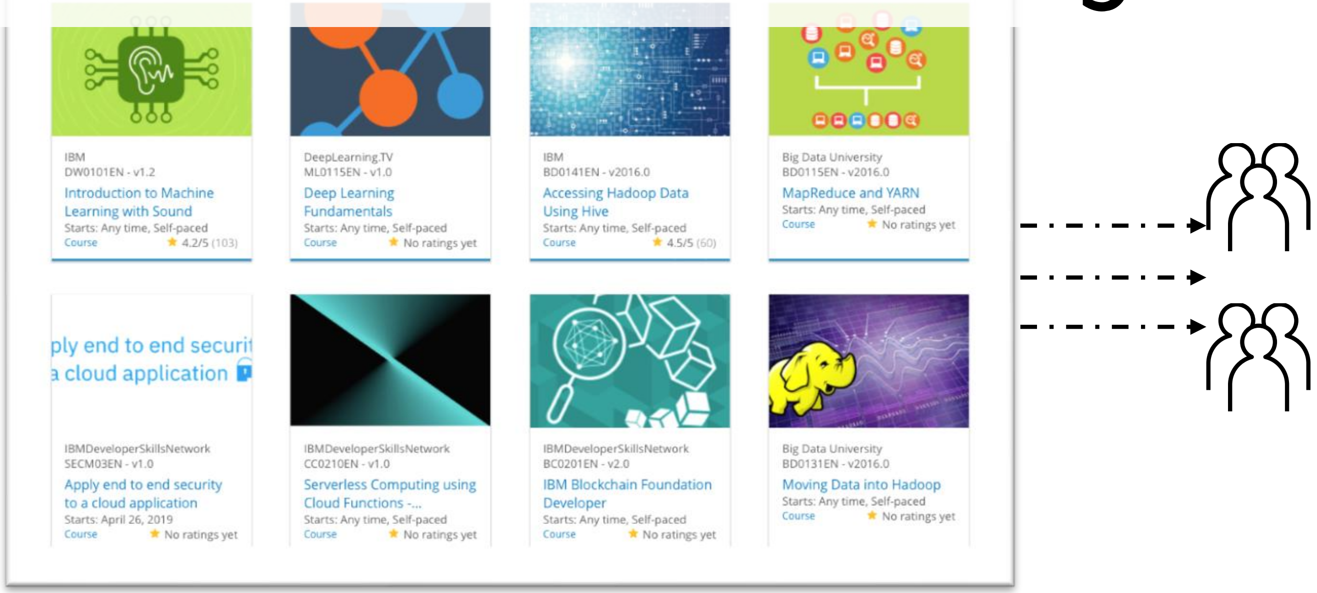
Build a Personalized Online Course Recommender System with Machine Learning

To Nguyen Ngoc Nguyen
21-7-2023



IBM
DW0101EN - v1.2
Introduction to Machine Learning with Sound
Starts: Any time, Self-paced
Course ★ 4.2/5 (103)

DeepLearning.TV
ML0115EN - v1.0
Deep Learning Fundamentals
Starts: Any time, Self-paced
Course ★ No ratings yet

IBM
BD0141EN - v2016.0
Accessing Hadoop Data Using Hive
Starts: Any time, Self-paced
Course ★ 4.5/5 (60)

Big Data University
BD0115EN - v2016.0
MapReduce and YARN
Starts: Any time, Self-paced
Course ★ No ratings yet

ply end to end securit
a cloud application

IBMDeveloperSkillsNetwork
SECM03EN - v1.0
Apply end to end security to a cloud application
Starts: April 26, 2019
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
CC0210EN - v1.0
Serverless Computing using Cloud Functions -...
Starts: Any time, Self-paced
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
BC0201EN - v2.0
IBM Blockchain Foundation Developer
Starts: Any time, Self-paced
Course ★ No ratings yet

Big Data University
BD0131EN - v2016.0
Moving Data into Hadoop
Starts: Any time, Self-paced
Course ★ No ratings yet

Outline

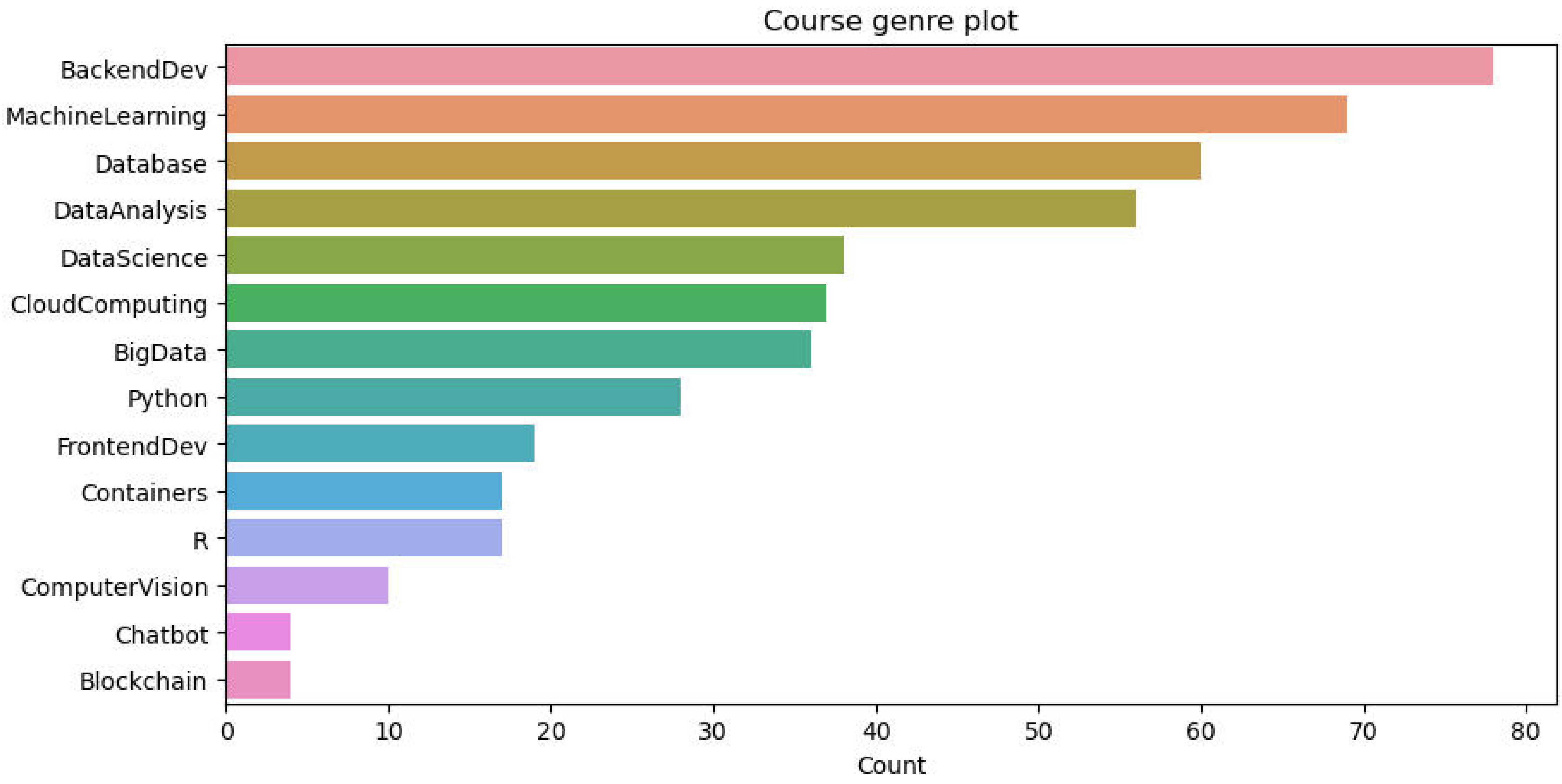
- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

Introduction

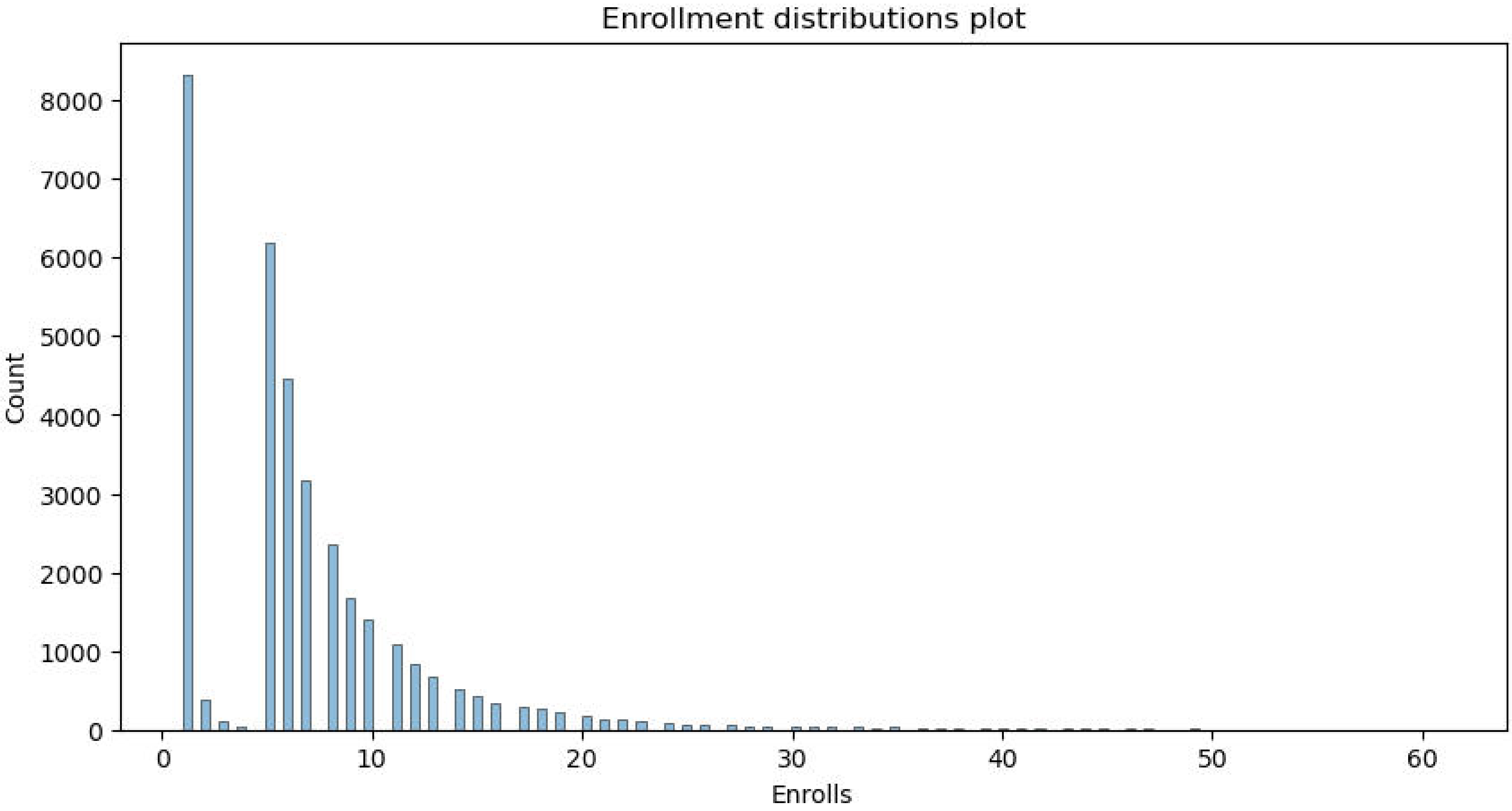
- A course recommendation system will help in:
  - Finding better courses
  - Finding courses that well suits each person's interests
  - We aim to find the best courses to recommend to users based on their interests, their friend's interests, and the courses they are enrolled in.
- Obstacles
  - We have many approaches
  - Each approach has different assumptions

Exploratory Data Analysis

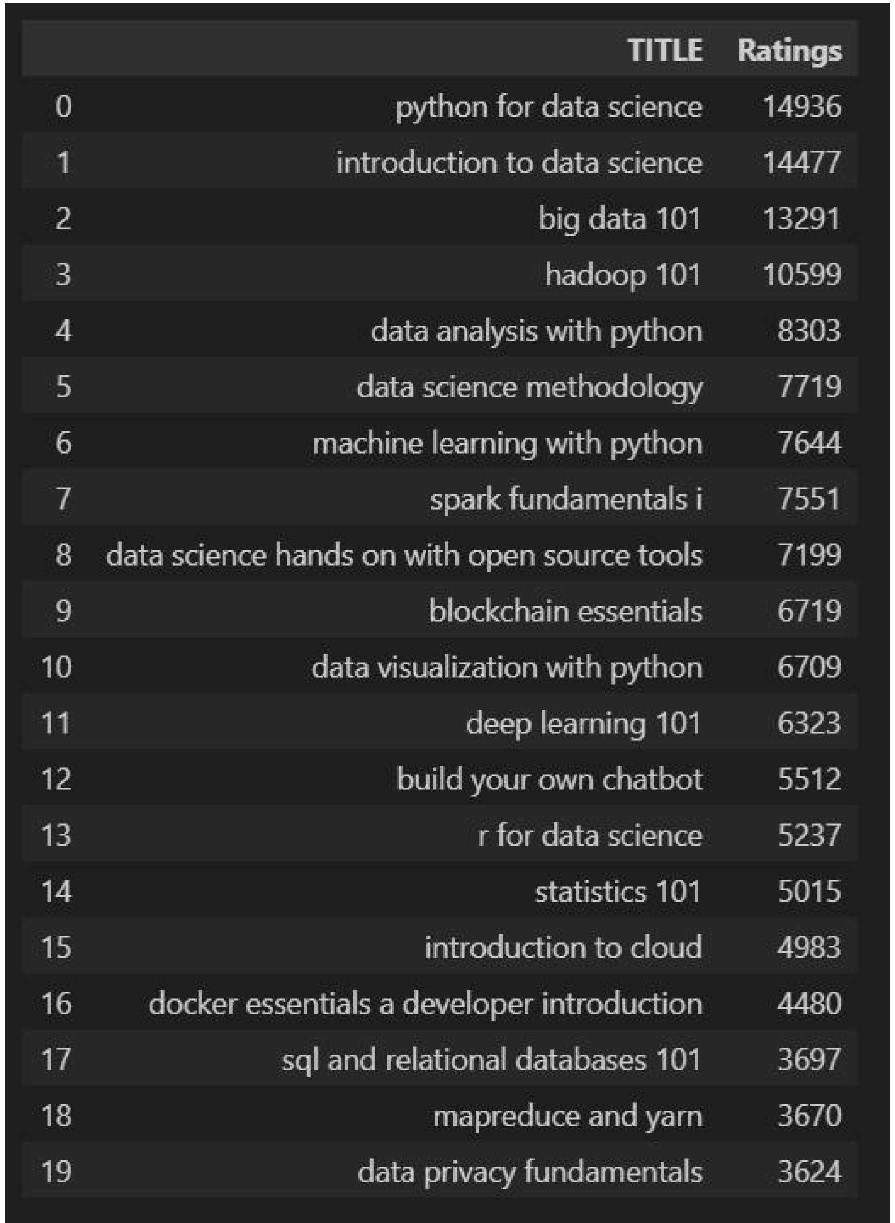
Course counts per genre



Course enrollment distribution

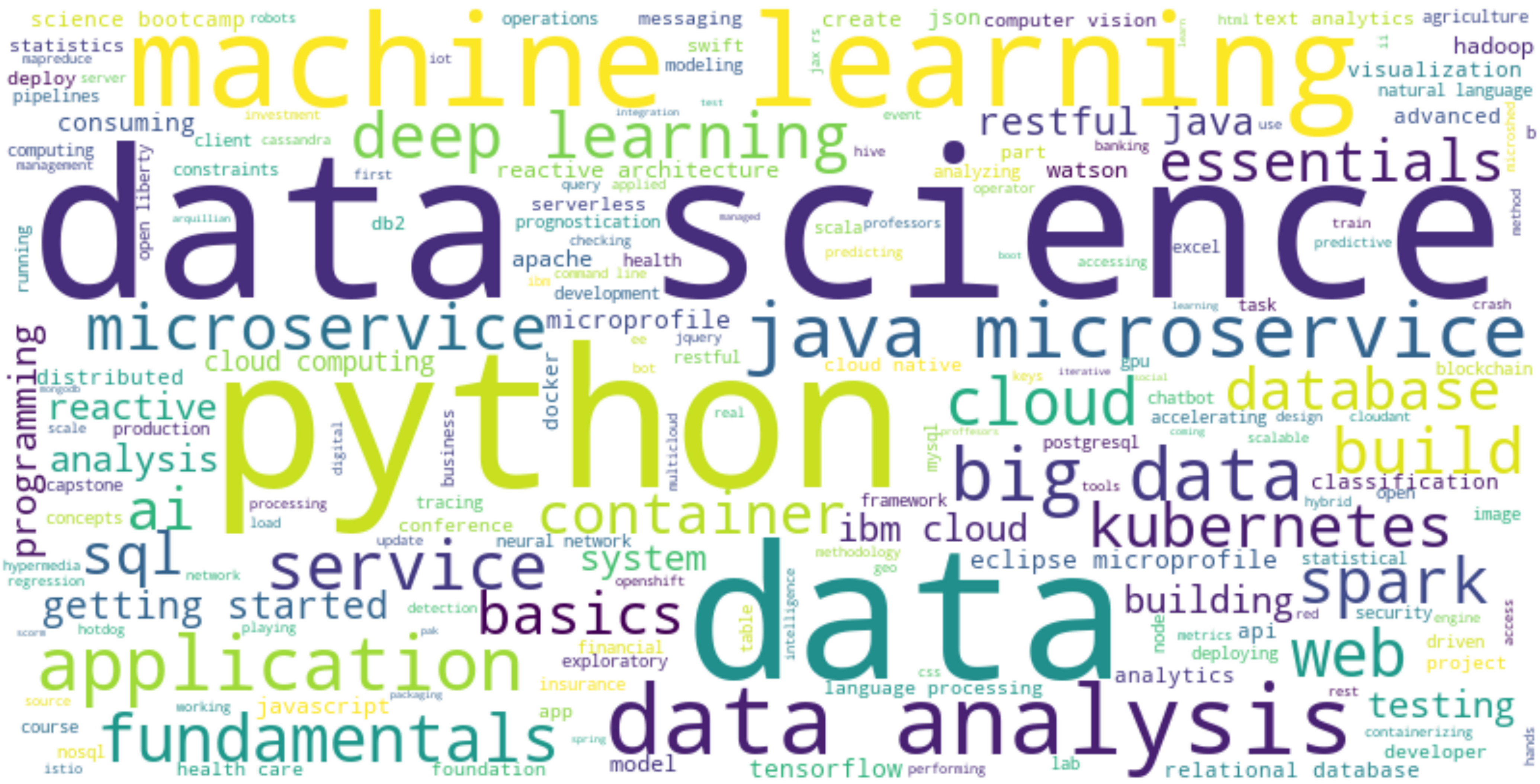


20 most popular courses

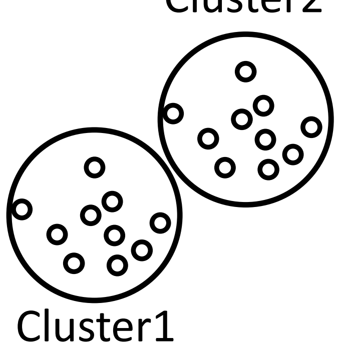


	TITLE	Ratings
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199
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

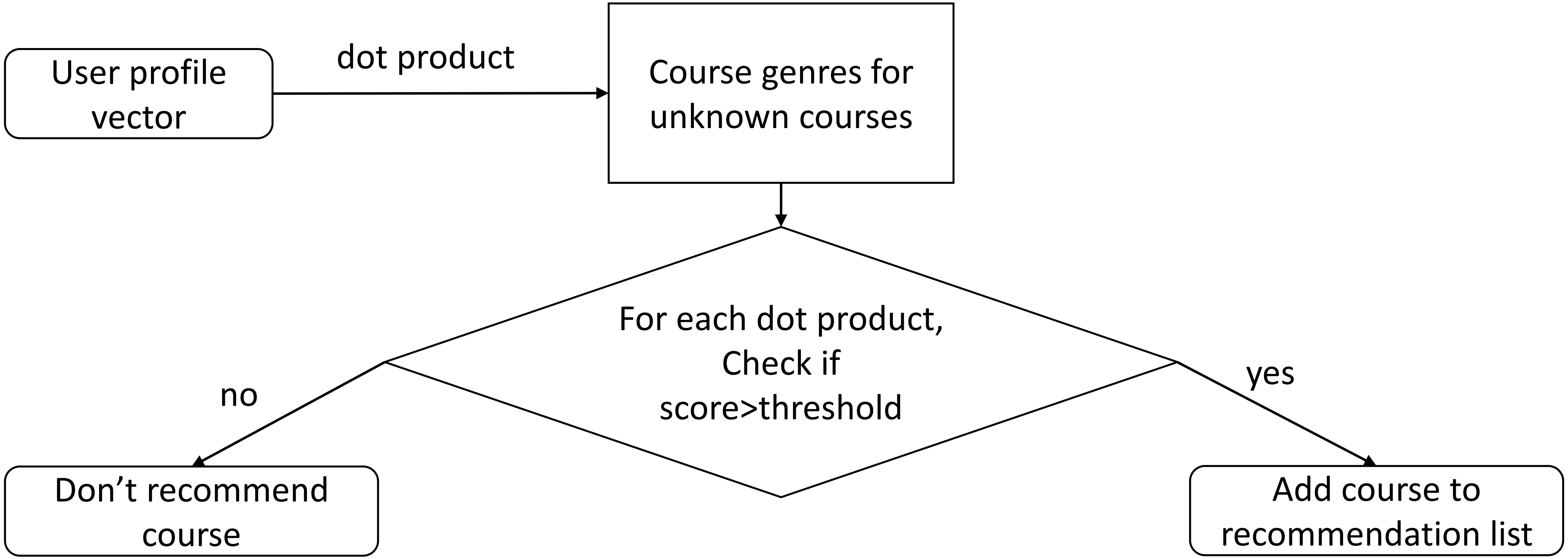
Word cloud of course titles



Content-based Recommender System using Unsupervised Learning



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



User profile vector

dot product

Course genres for unknown courses

For each dot product, Check if score>threshold

no

yes

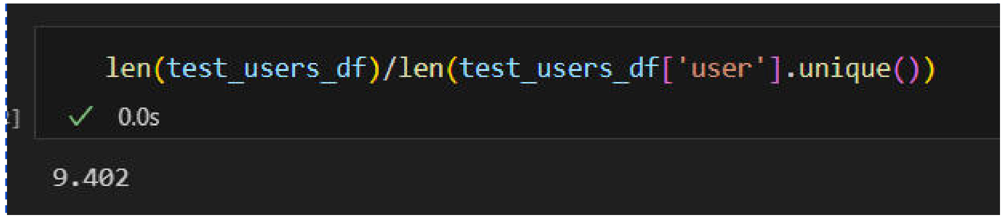
Don’t recommend course

Add course to recommendation list

Evaluation results of user profile-based recommender system

Score_threshold = 10.0

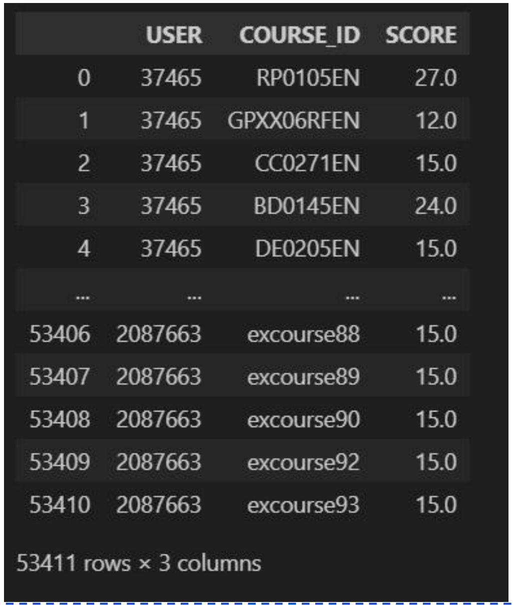
On average, how many new/unseen courses have been recommended per user (in the test user dataset)



```
len(test_users_df)/len(test_users_df['user'].unique())
✓ 0.0s

9.402
```

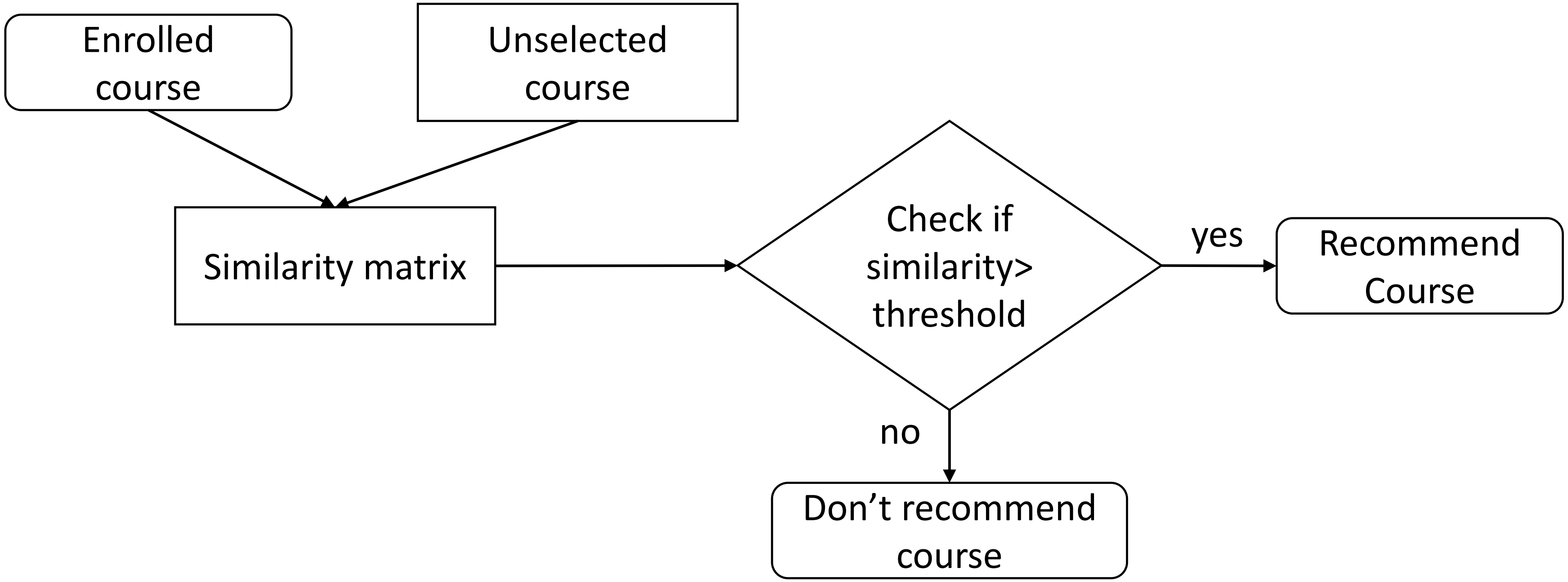
What are the most frequently recommended courses? Return the top-10 commonly recommended courses across all users



	USER	COURSE_ID	SCORE
0	37465	RP0105EN	27.0
1	37465	GPXX06RFEN	12.0
2	37465	CC0271EN	15.0
3	37465	BD0145EN	24.0
4	37465	DE0205EN	15.0
...	...	...	...
53406	2087663	excourse88	15.0
53407	2087663	excourse89	15.0
53408	2087663	excourse90	15.0
53409	2087663	excourse92	15.0
53410	2087663	excourse93	15.0

53411 rows × 3 columns

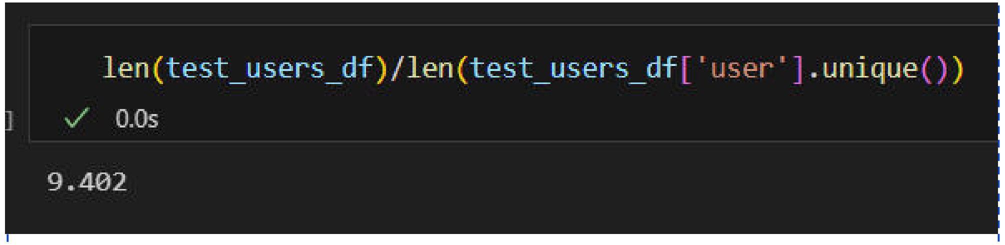
Flowchart of content-based recommender system using course similarity



Evaluation results of course similarity based recommender system

Threshold = 0.6

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

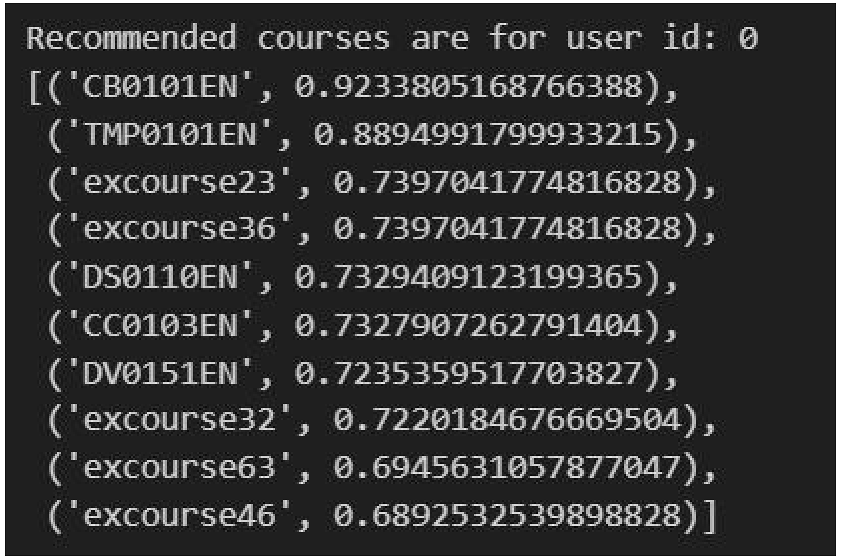


```
len(test_users_df)/len(test_users_df['user'].unique())
```

✓ 0.0s

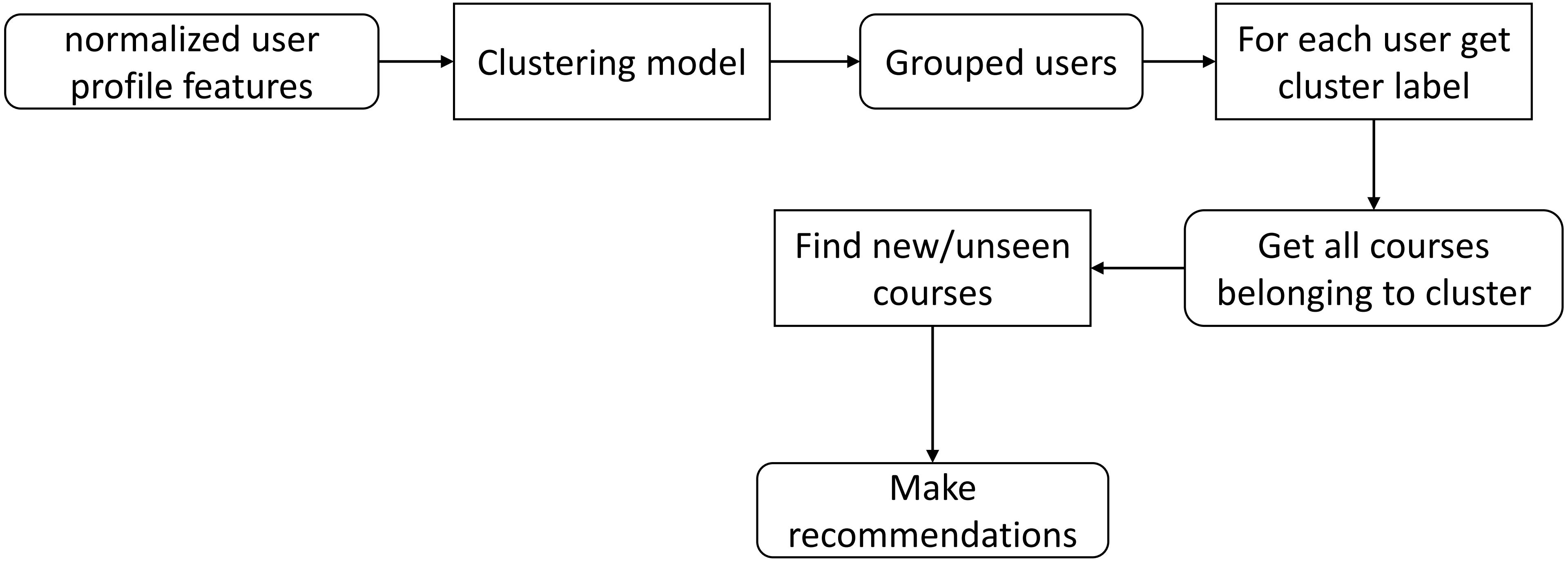
```
9.402
```

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



```
Recommended courses are for user id: 0
[('CB0101EN', 0.9233805168766388),
 ('TMP0101EN', 0.8894991799933215),
 ('excourse23', 0.7397041774816828),
 ('excourse36', 0.7397041774816828),
 ('DS0110EN', 0.7329409123199365),
 ('CC0103EN', 0.7327907262791404),
 ('DV0151EN', 0.7235359517703827),
 ('excourse32', 0.7220184676669504),
 ('excourse63', 0.6945631057877047),
 ('excourse46', 0.6892532539898828)]
```

Flowchart of clustering-based recommender system



normalized user profile features

Clustering model

Grouped users

For each user get cluster label

Get all courses belonging to cluster

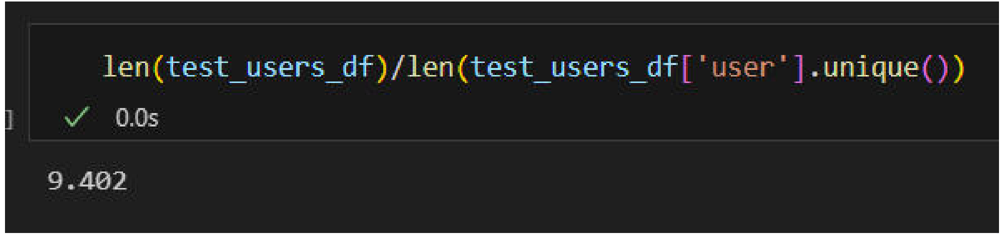
Find new/unseen courses

Make recommendations

Evaluation results of clustering-based recommender system

Number of clusters = 20

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

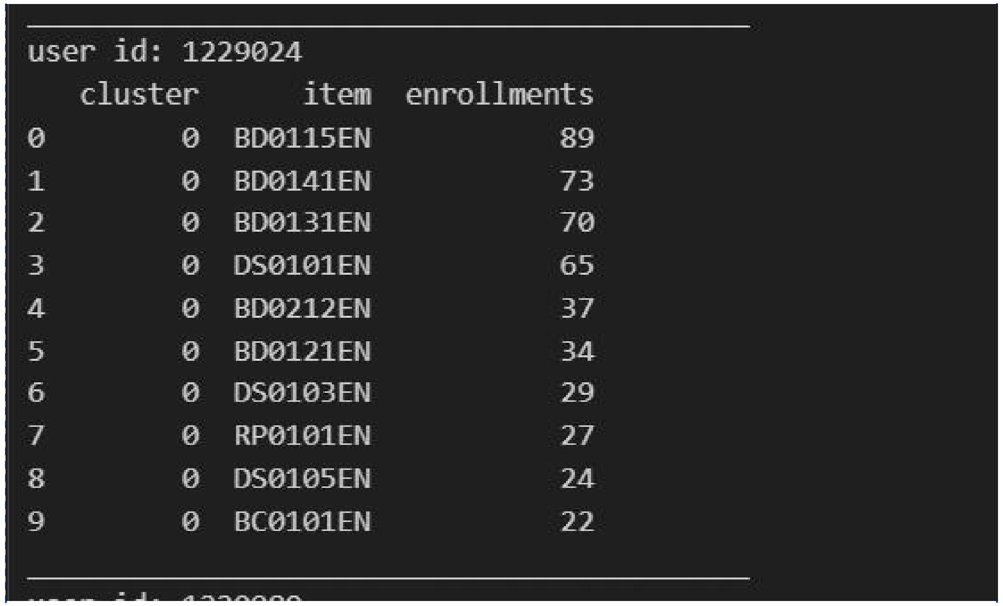


```
len(test_users_df)/len(test_users_df['user'].unique())
```

✓ 0.0s

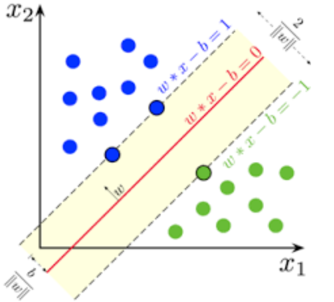
```
9.402
```

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

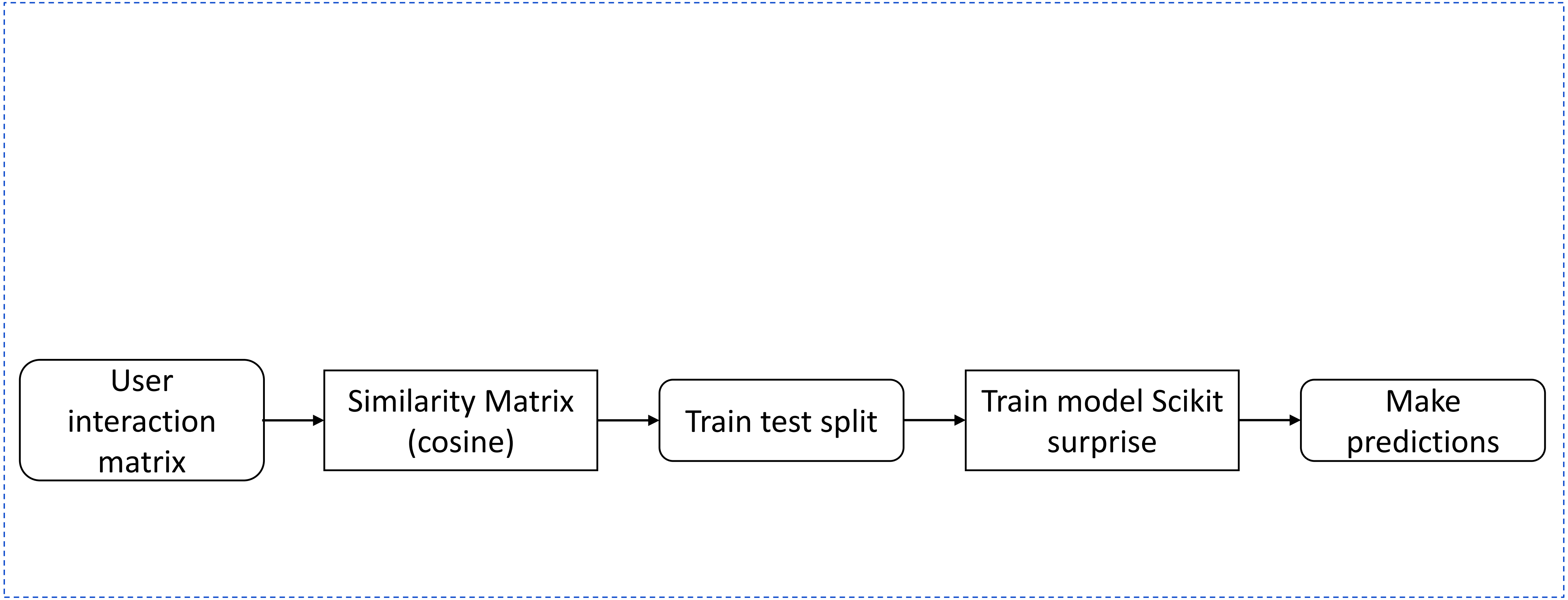


```
user id: 1229024
   cluster      item  enrollments
0        0  BD0115EN           89
1        0  BD0141EN           73
2        0  BD0131EN           70
3        0  DS0101EN           65
4        0  BD0212EN           37
5        0  BD0121EN           34
6        0  DS0103EN           29
7        0  RP0101EN           27
8        0  DS0105EN           24
9        0  BC0101EN           22
```

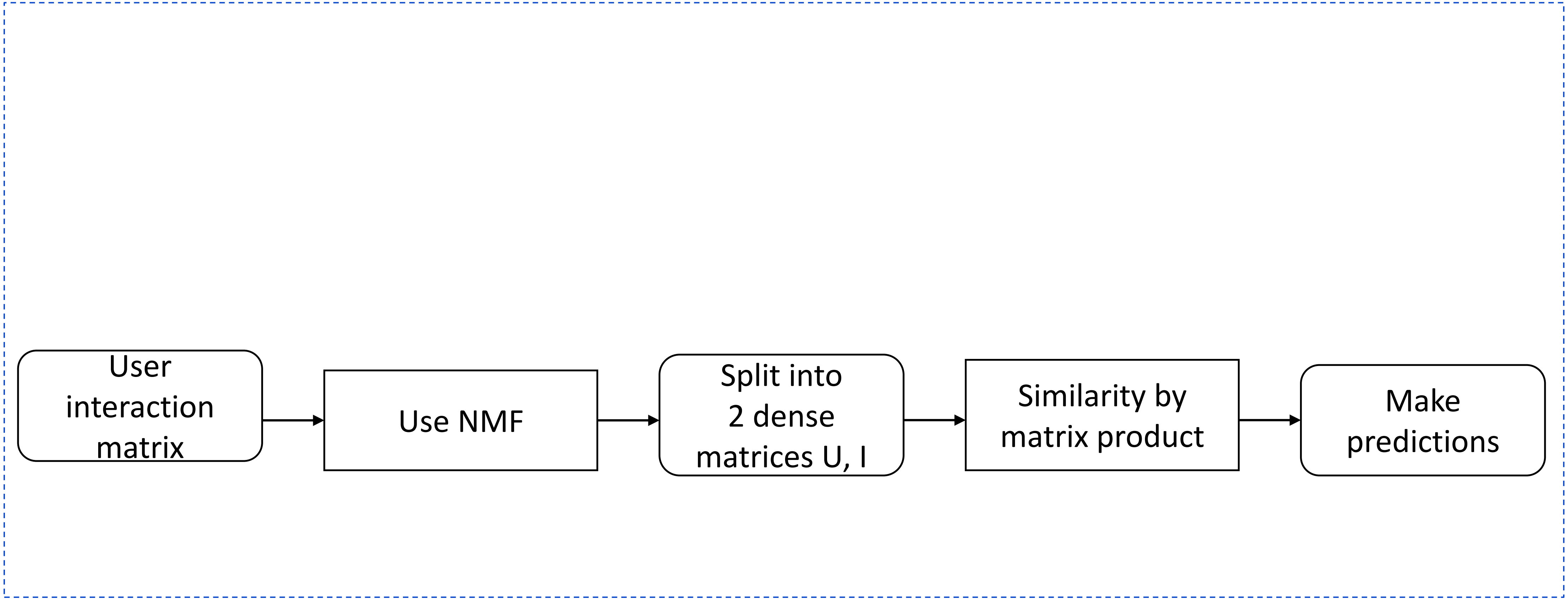
Collaborative-filtering Recommender System using Supervised Learning



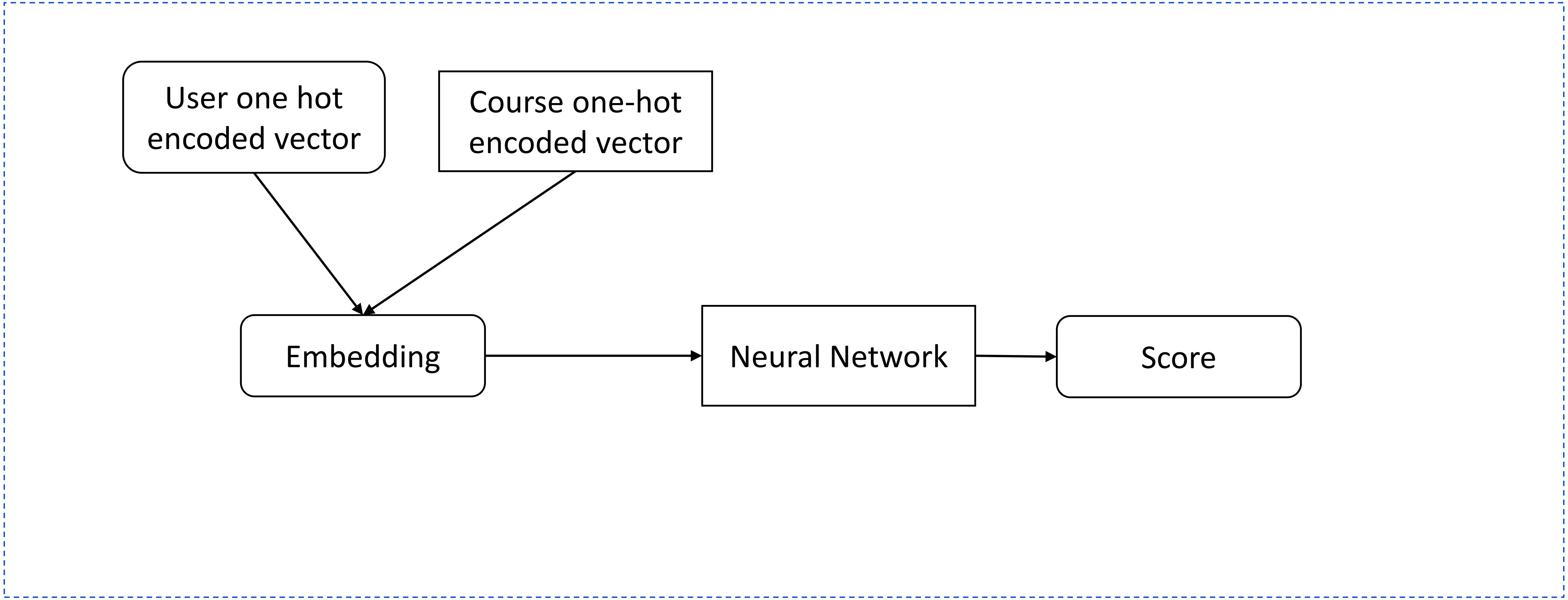
Flowchart of KNN based recommender system



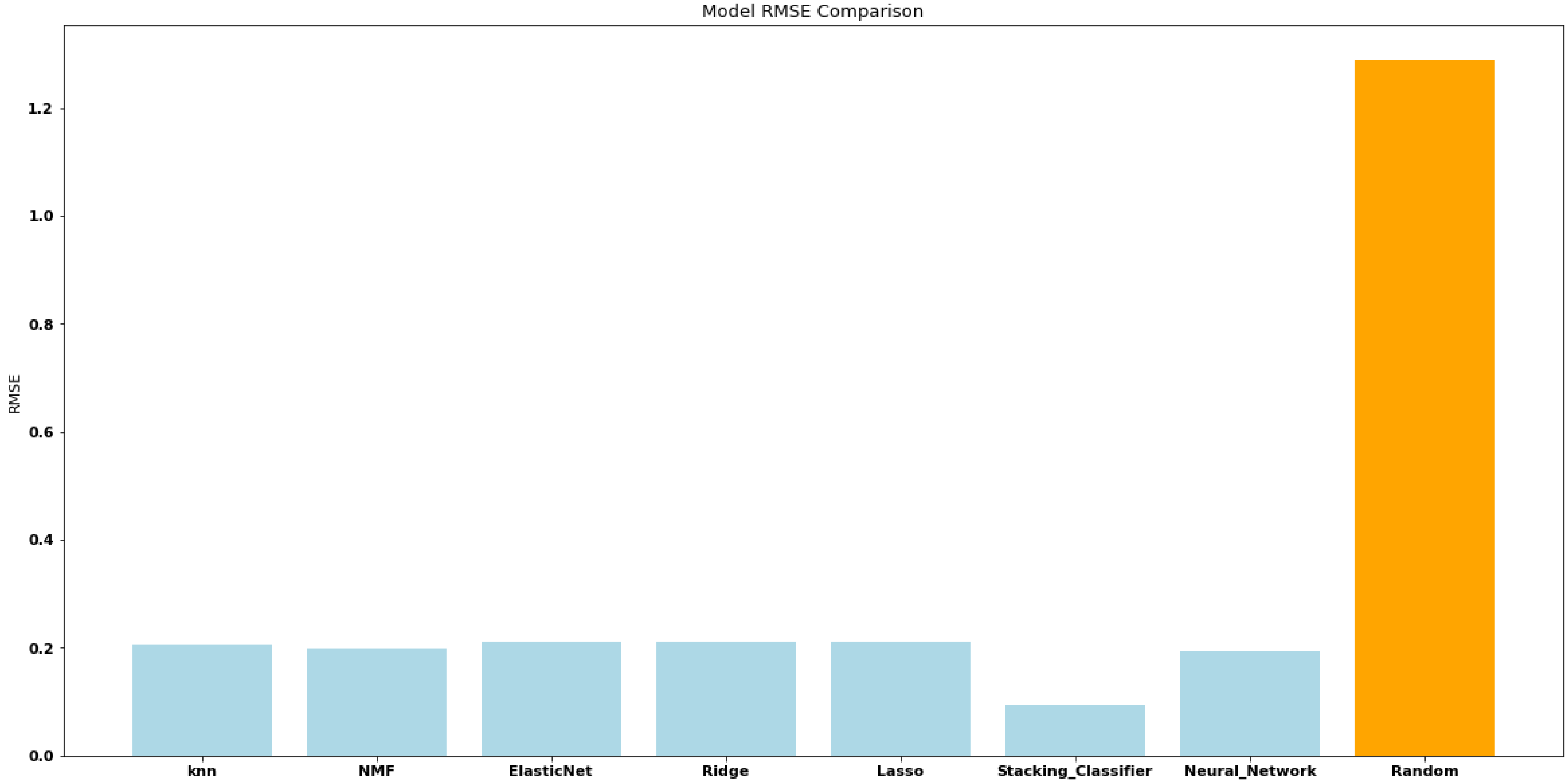
Flowchart of NMF based recommender system


User interaction matrix
Use NMF
Split into 2 dense matrices U, I
Similarity by matrix product
Make predictions

Flowchart of Neural Network Embedding based recommender system



Compare the performance of collaborative-filtering models



Conclusions

- Similar performance of models
- User profile based has highest number of recommendations
- Stacking Classifier has best performance
- Similarity matrix's high complexity
- NMF as a solution

Appendix

All materials link: **here**