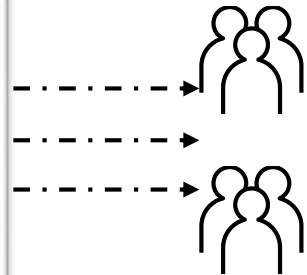
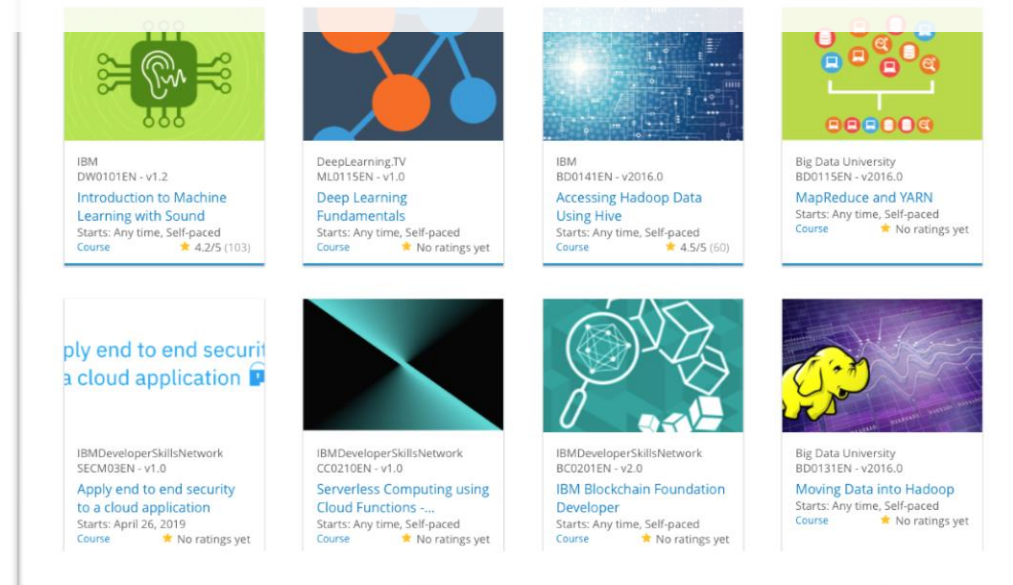


Build a Personalized Online Course Recommender System with Machine Learning

Fahd Seddik
12-9-2022



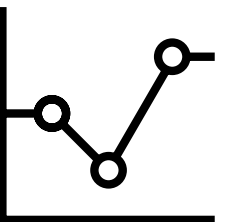
Outline

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

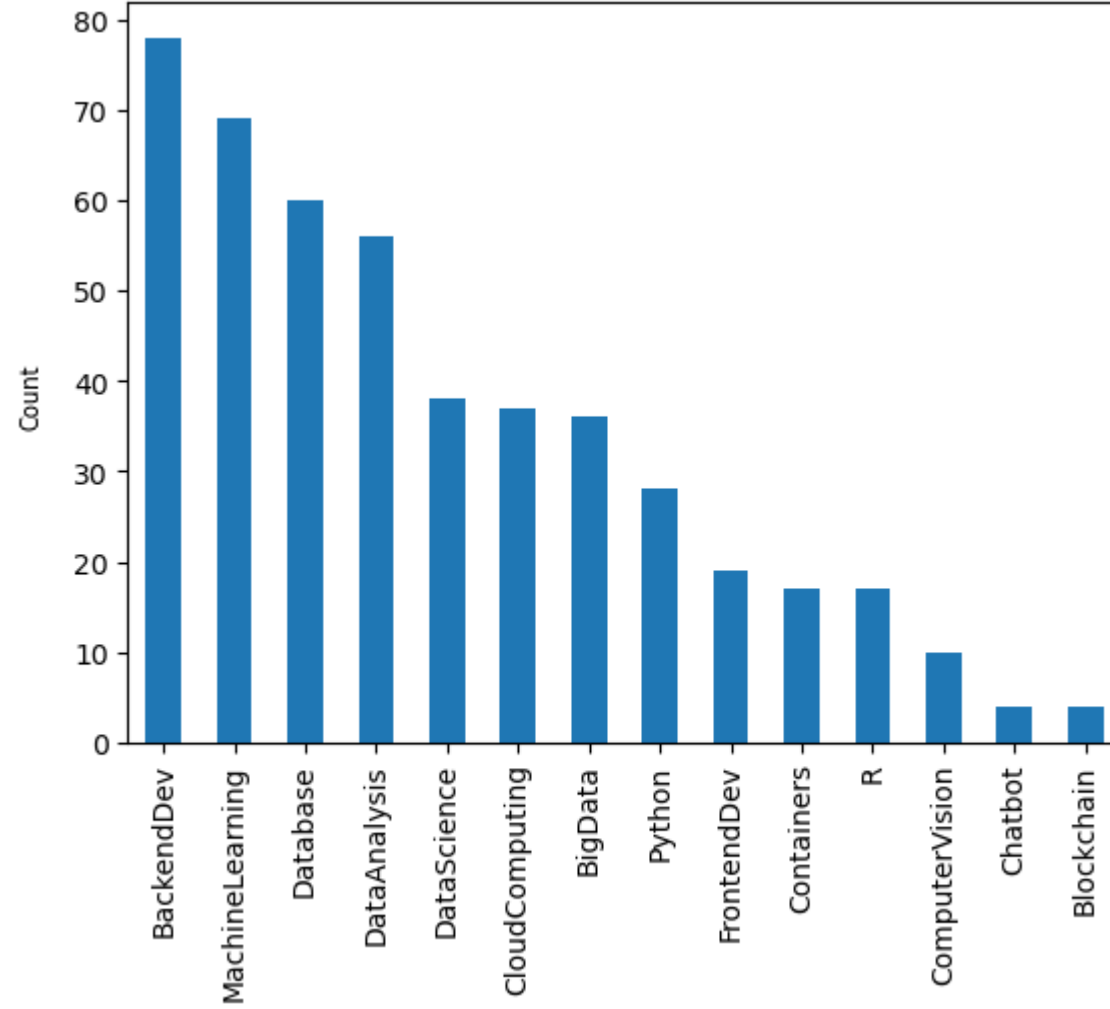
Introduction

- 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

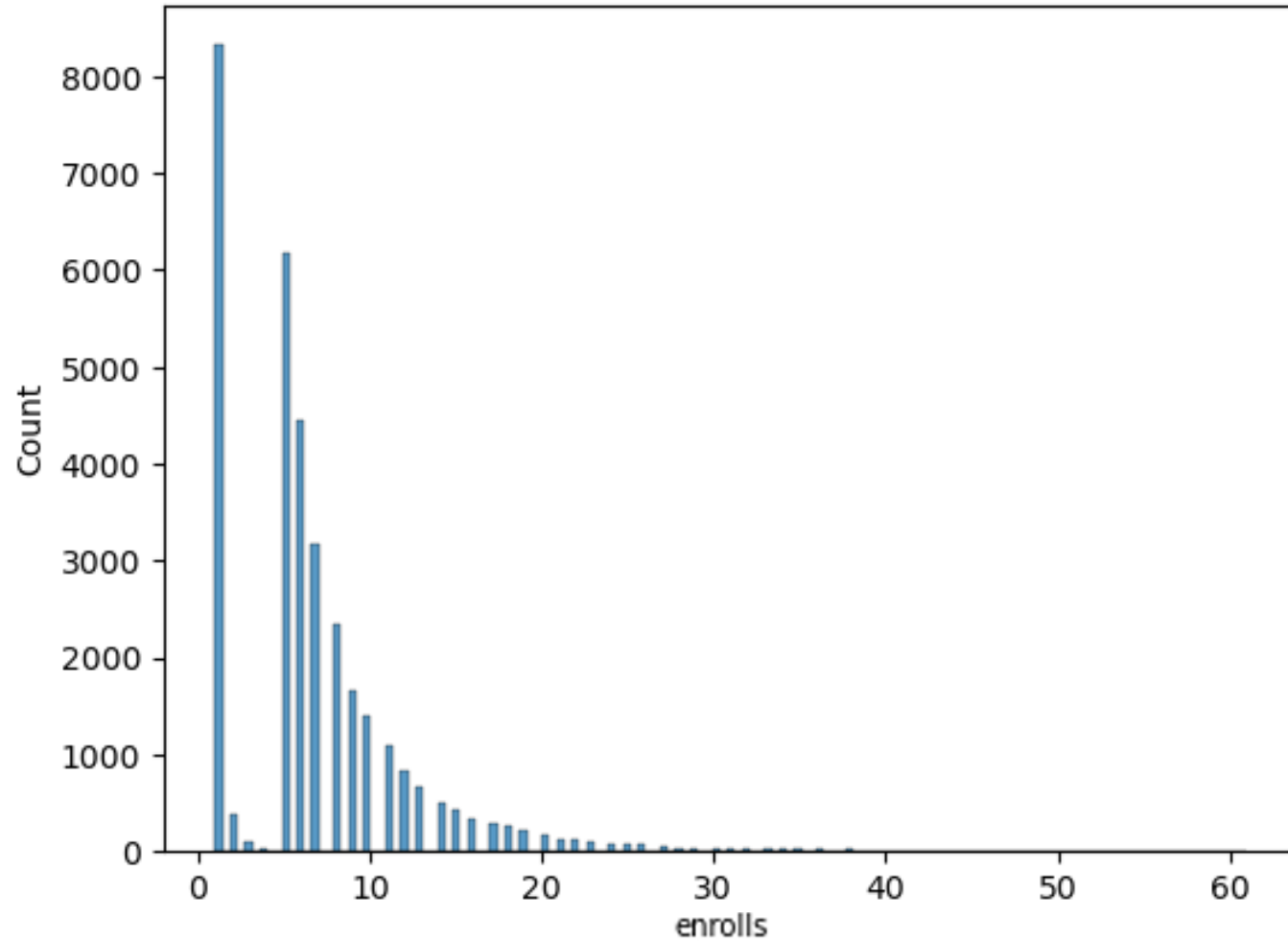
Exploratory Data Analysis



Course counts per genre



Course enrollment distribution



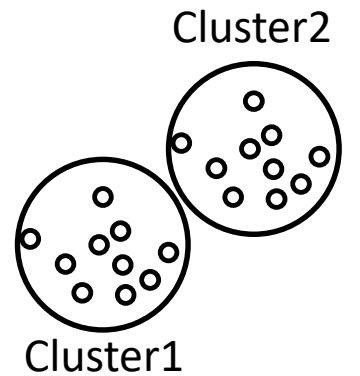
20 most popular courses

	TITLE	Enrolls
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

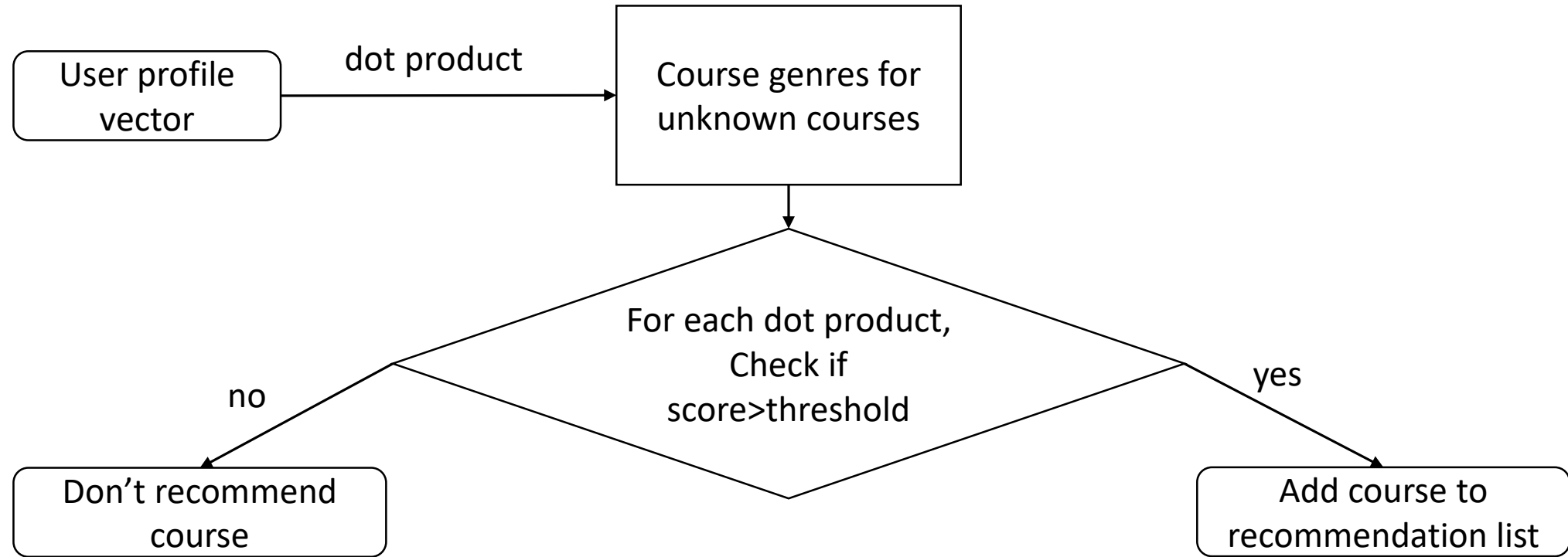
Word cloud of course titles



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

Score_threshold = 10.0

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

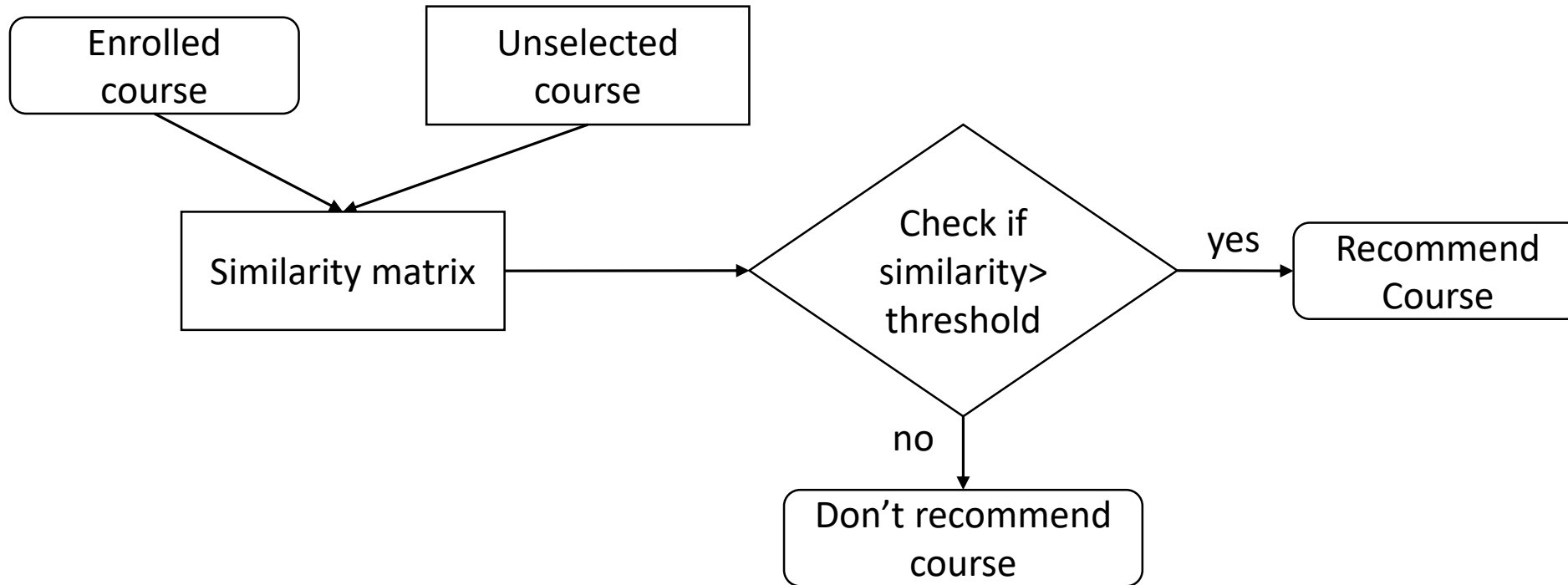
```
res_df[ 'SCORE' ].mean()
```

```
18.62679972290352
```

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

COURSE_ID	
TA0106EN	608
GPXX0IBEN	548
excOURSE22	547
excOURSE21	547
ML0122EN	544
excOURSE06	533
excOURSE04	533
GPXX0TY1EN	533
excOURSE31	524
excOURSE73	516

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)

```
s = 0
for i in range(len(res_df['COURSE_ID'])):
    s+=len(res_df['COURSE_ID'].iloc[i])
avg = s/len(res_df['COURSE_ID'])
```

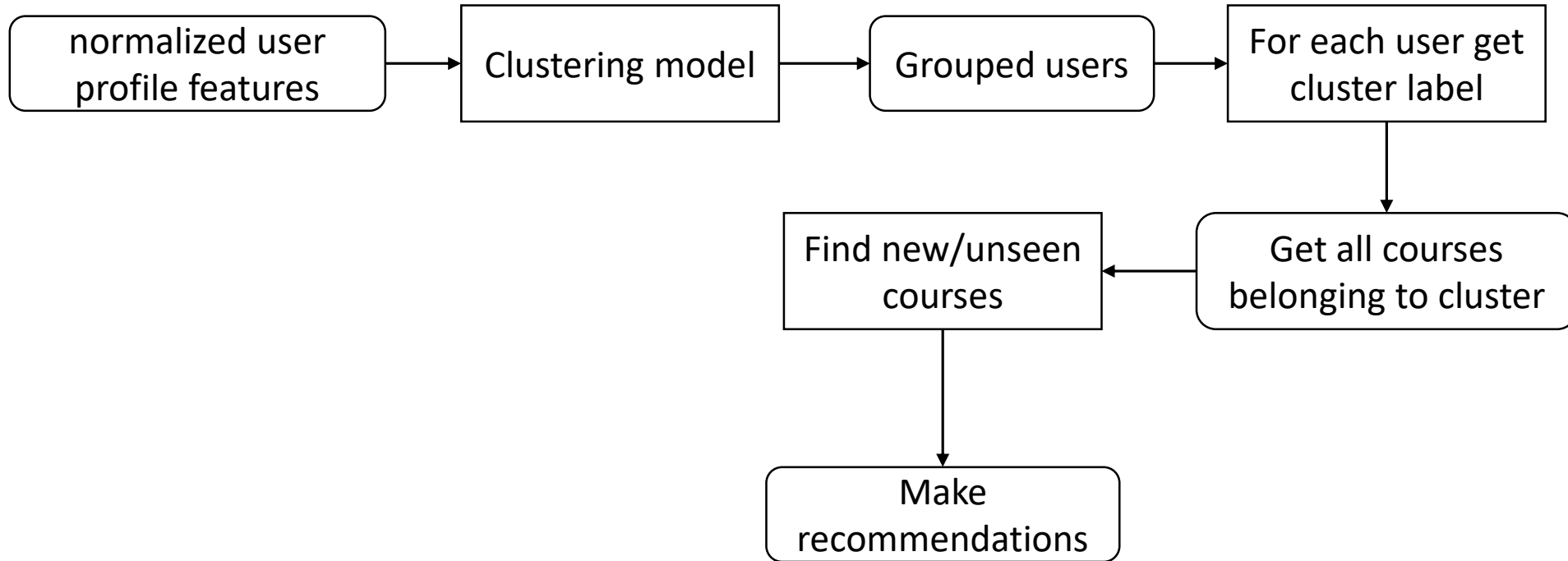
avg

11.377

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

excercise22	579
excercise62	579
DS0110EN	562
excercise65	555
excercise63	555
excercise72	551
excercise68	550
excercise67	539
excercise74	539
BD0145EN	506

Flowchart of clustering-based recommender system



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)

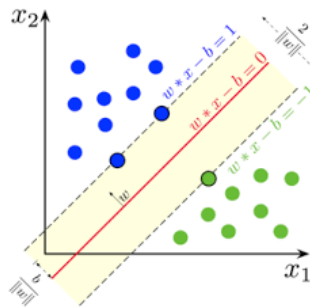
```
: s = 0
  for r in user_recommendations.value:
      s+=r[1:].sum()
  avg=s/len(user_recommendations)
  print(avg)

5.733
```

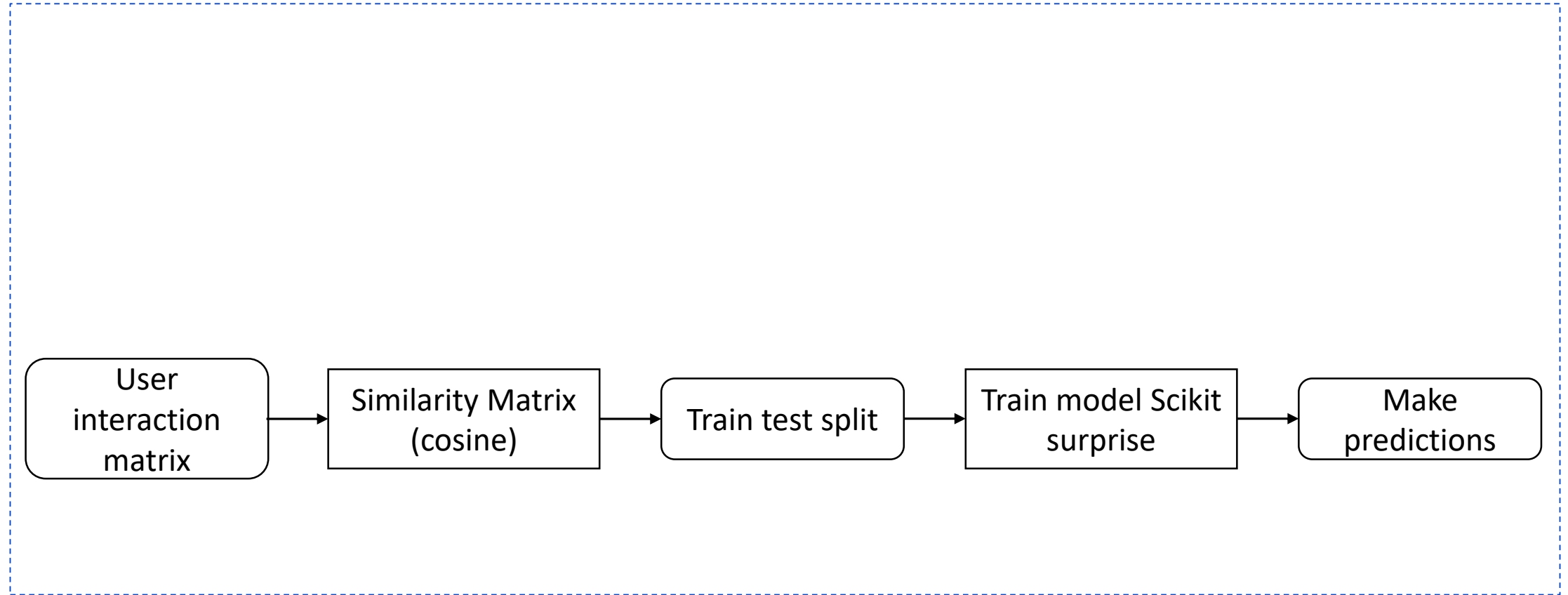
What are the most frequently recommended courses? Return the top-10 commonly recommended courses

DS0103EN	579
DA0101EN	532
BD0111EN	456
DS0101EN	444
BD0101EN	428
PY0101EN	386
DS0105EN	319
ML0101ENv3	299
BC0101EN	296
ML0115EN	286

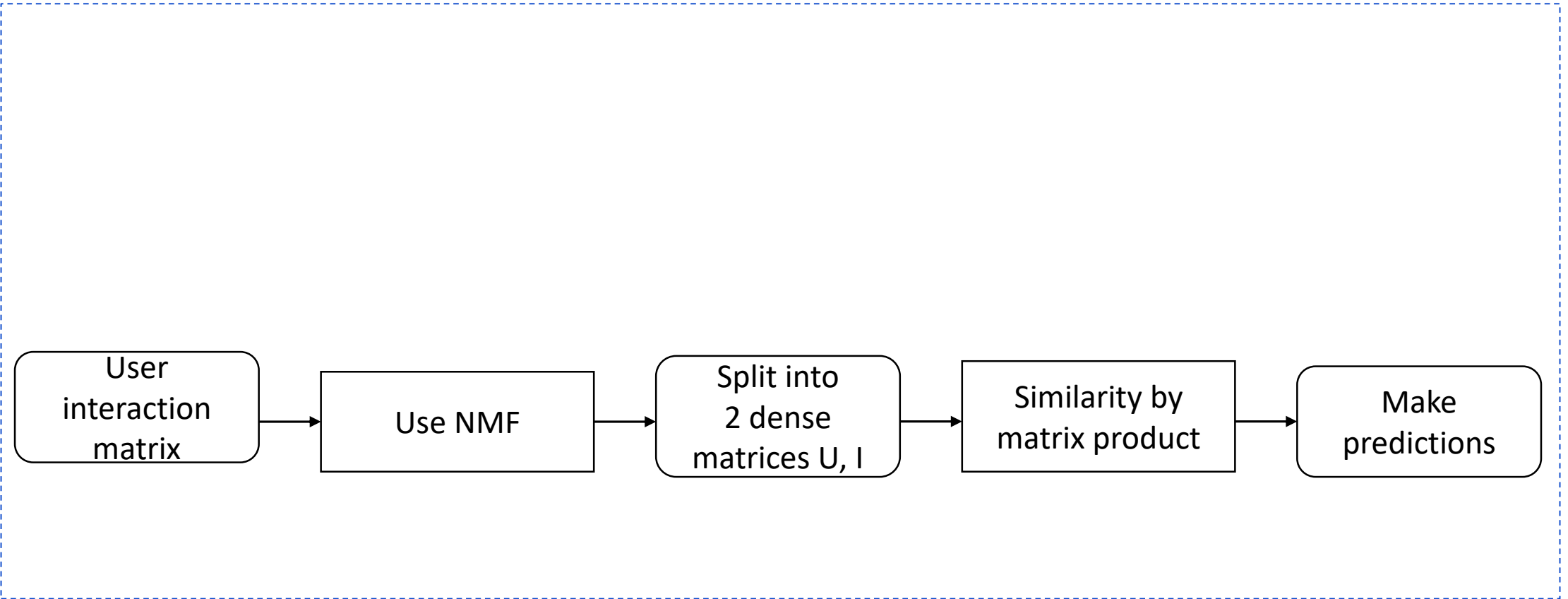
Collaborative-filtering Recommender System using Supervised Learning



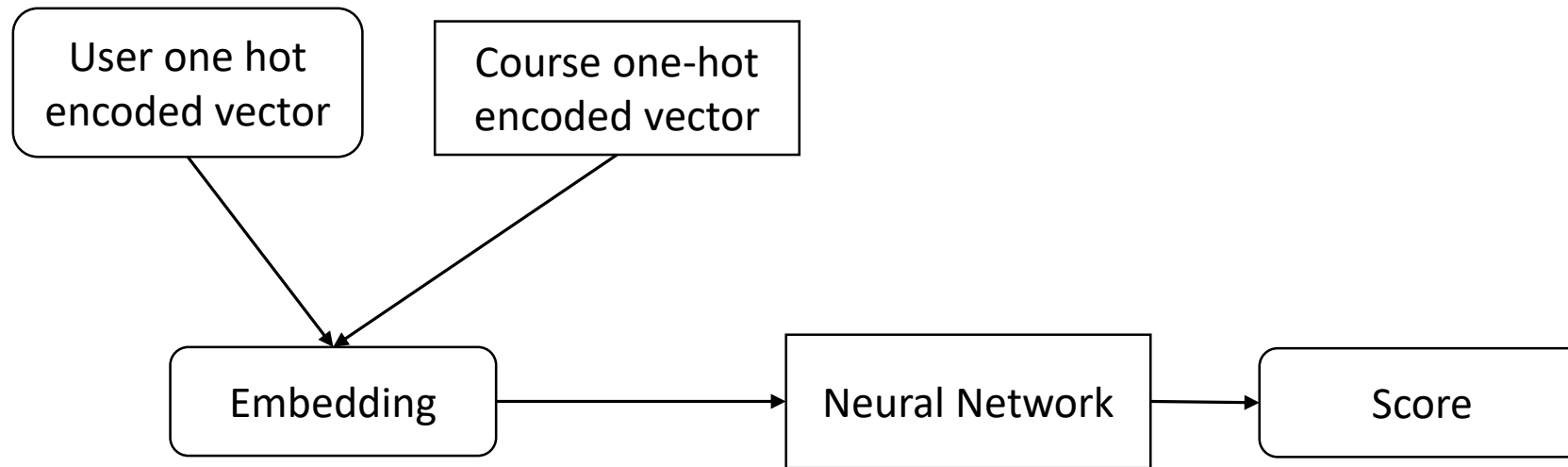
Flowchart of KNN based recommender system



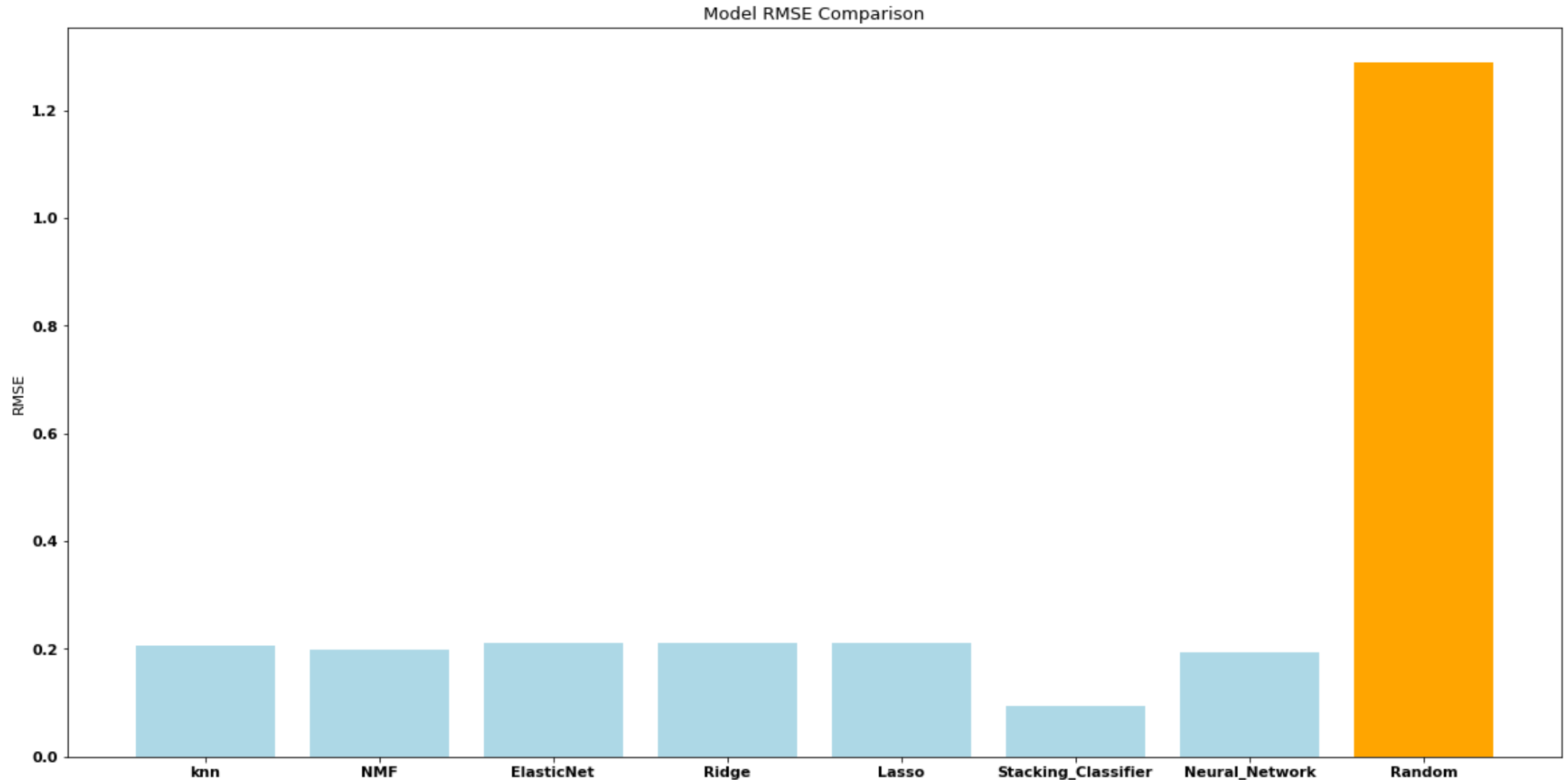
Flowchart of NMF based recommender system



Flowchart of Neural Network Embedding based recommender system



Compare the performance of collaborative-filtering models



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

<https://drive.google.com/drive/folders/1OKcl56MRC4ShCSxeGA7x3dQPyf84i6rm?usp=sharing>