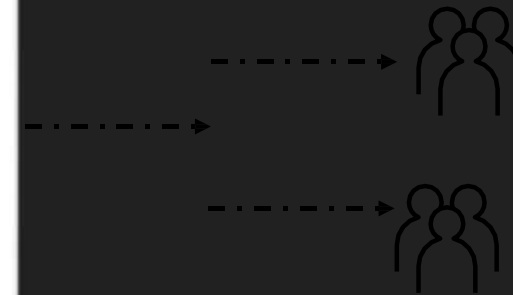
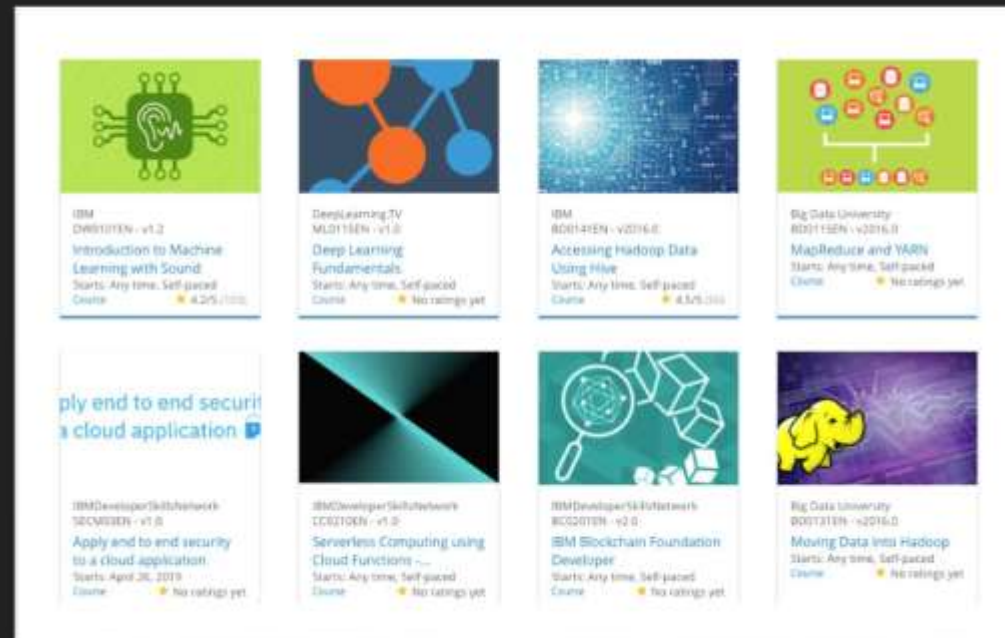


Build a Personalized Online Course Recommender System with Machine Learning



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

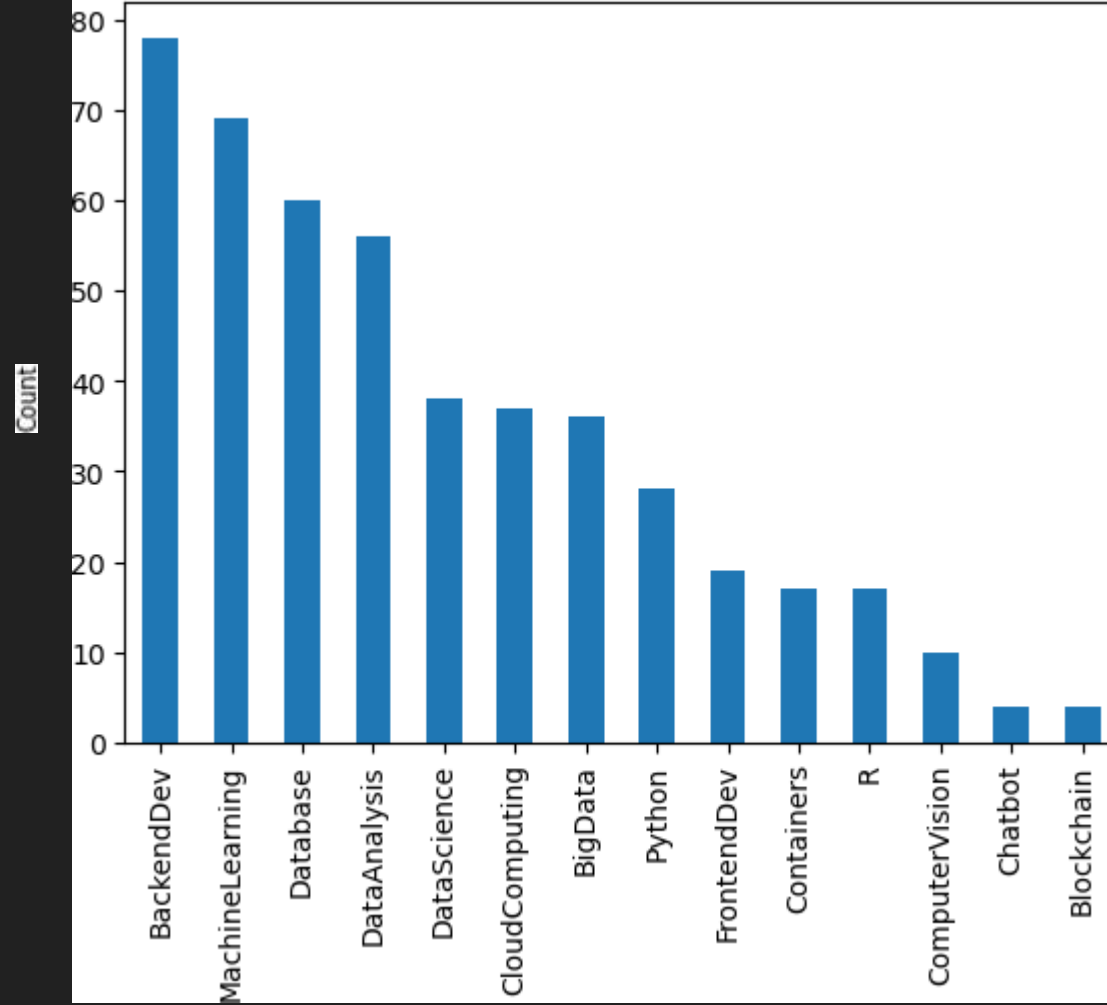
A course recommendation system helps to:

- Provide personalized course recommendations
- Identify courses that align closely with individual interests
- Our goal is to identify the most suitable courses for users by considering their interests, the interests of their friends, and the courses they are currently enrolled in.

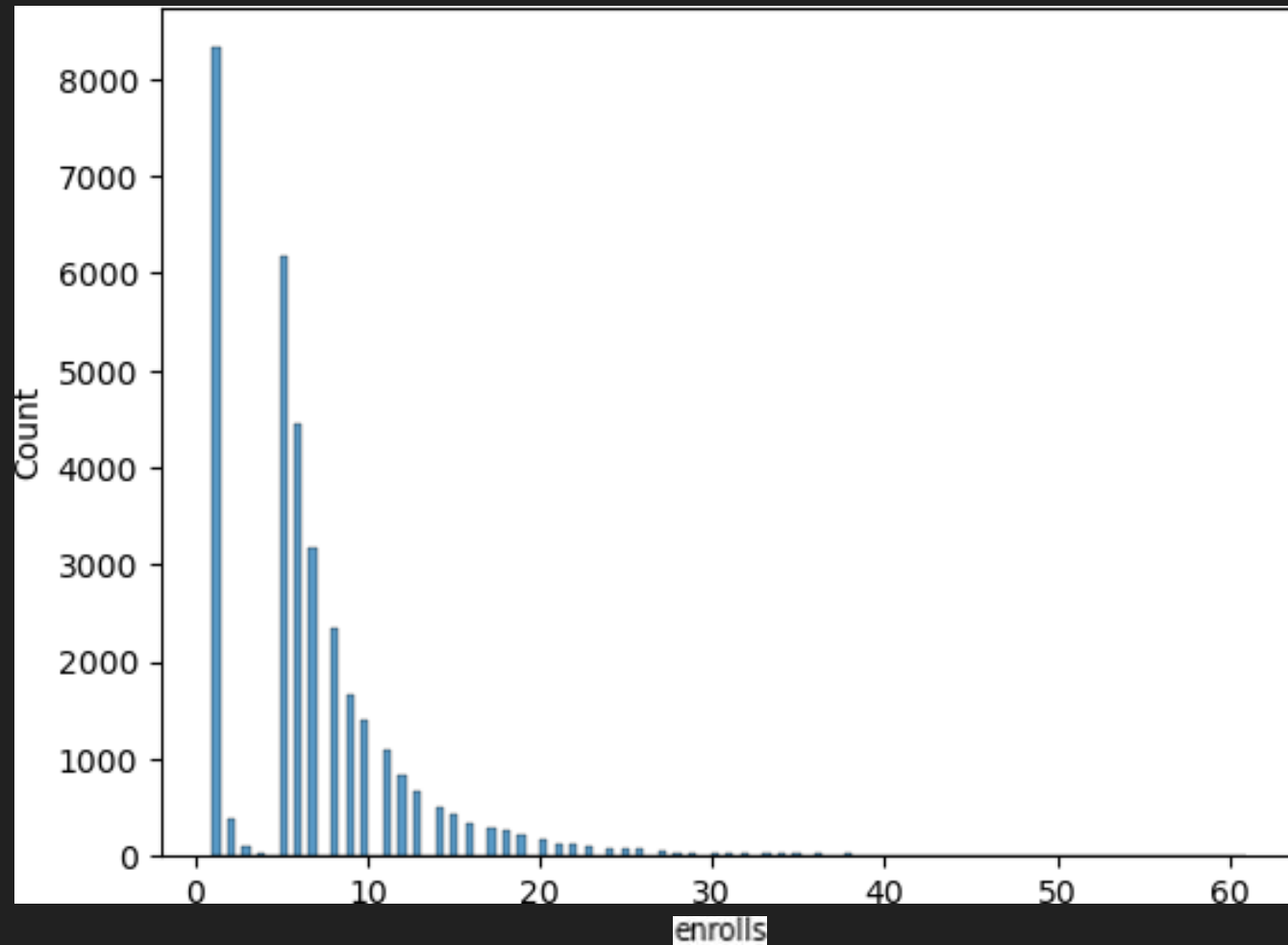
Obstacles

- There are many options to build the recommendation system.
- Each approach has different assumptions and ways in itself to be built.

Course counts per genre

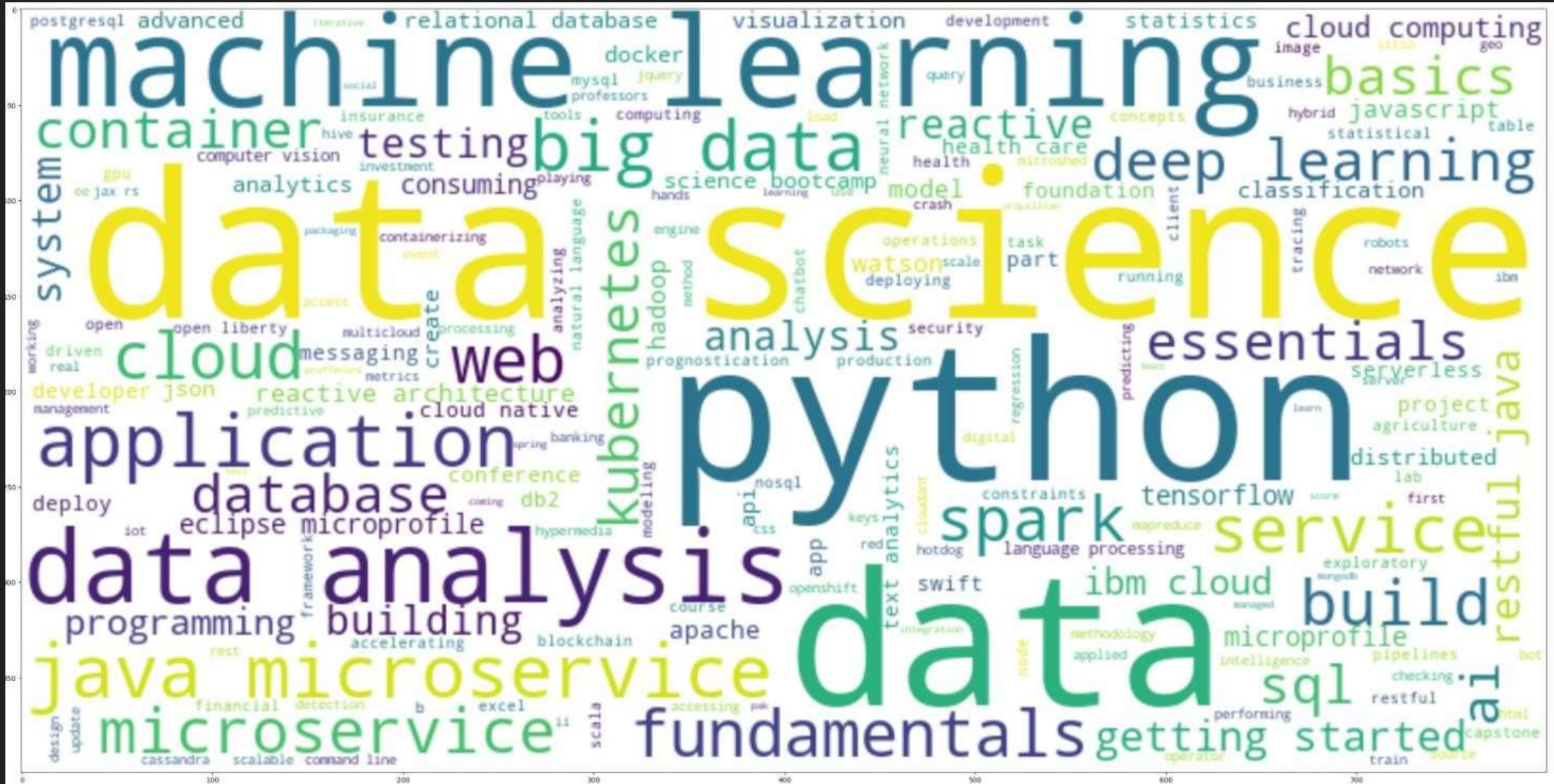


Course enrollment distribution

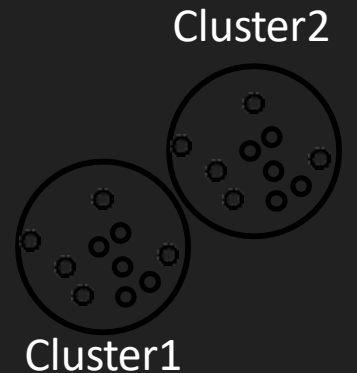


20 most popular courses

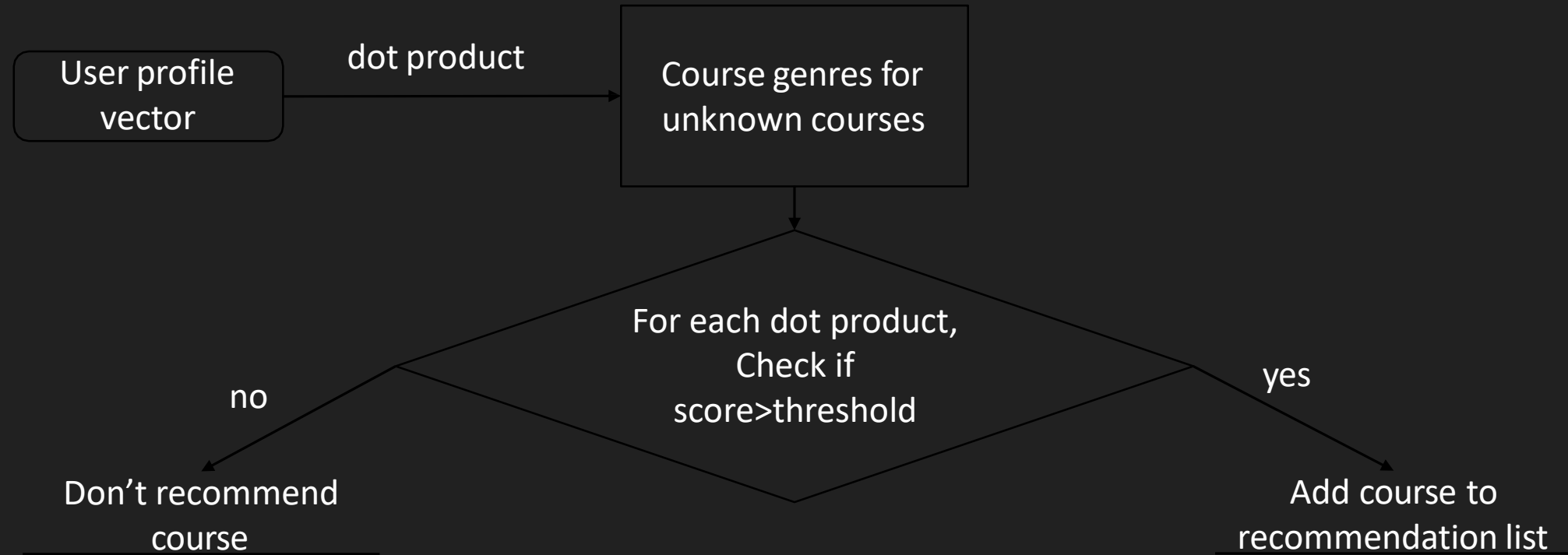
	TITLE	Enrolls
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199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



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)

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

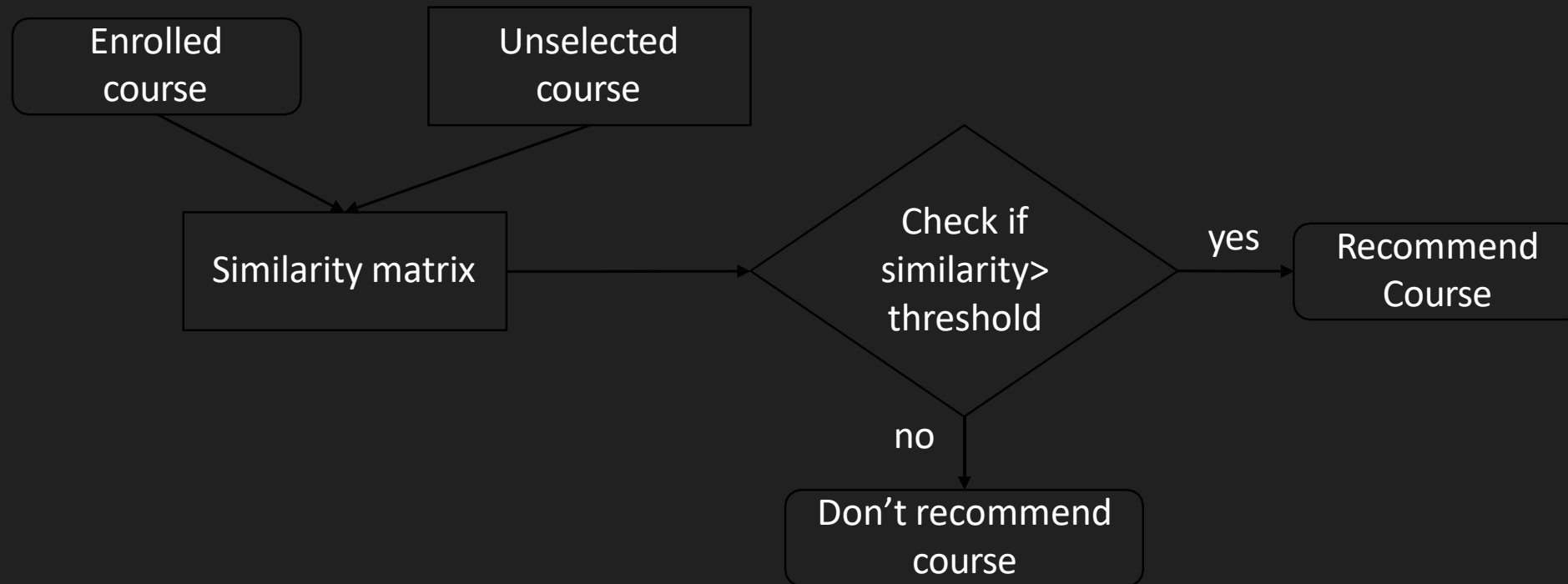
```
[46]: 18.62679972290352
```

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

```
[47]: res_df.groupby('COURSE_ID').size().sort_values(ascending=False)[:10]
```

```
[47]: COURSE_ID
      TA0106EN      608
      GPXX0IBEN      548
      excourse22      547
      excourse21      547
      ML0122EN      544
      excourse06      533
      excourse04      533
      GPXX0TY1EN      533
      excourse31      524
      excourse73      516
      dtype: int64
```

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'])
```

[34]

```
avg
```

[35]

```
... 11.377
```

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

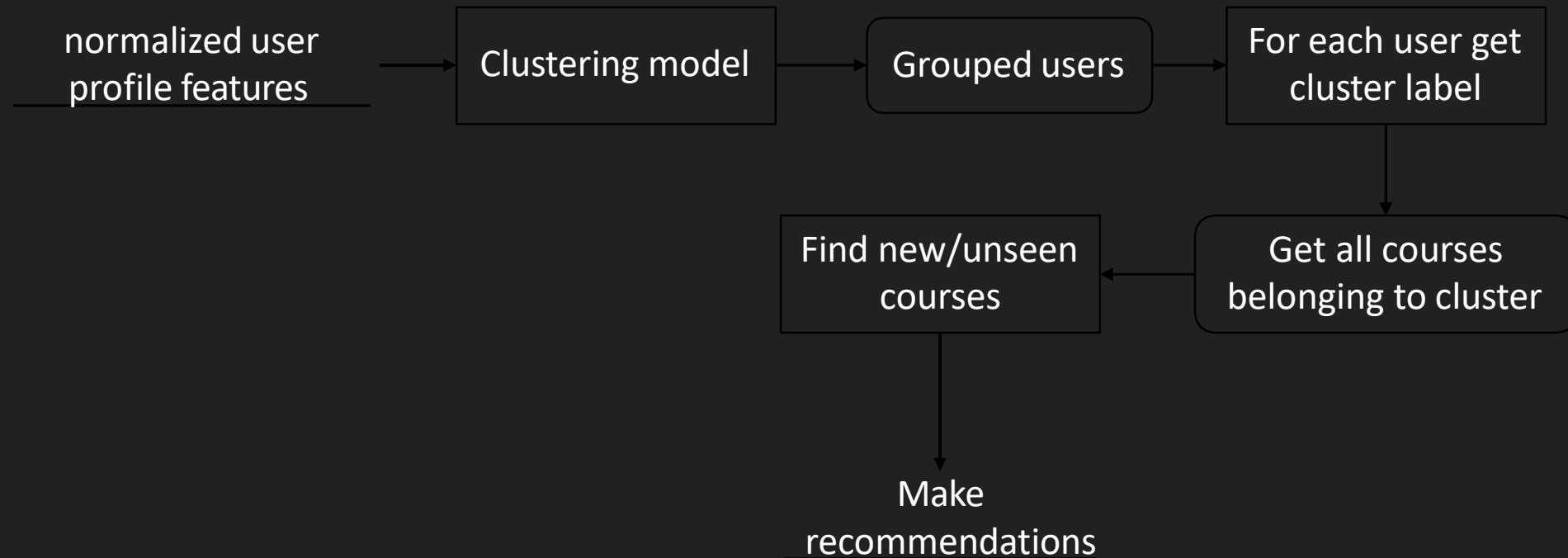
▷ ▾

```
pd.Series({k: v for k, v in sorted(dict(zip(recc,tally)).items(),
|   key=lambda item: item[1]))).sort_values(ascending=False)[:10]
```

[39]

```
... excourse22    579
    excourse62    579
    DS0110EN     562
    excourse65    555
    excourse63    555
    excourse72    551
    excourse68    550
    excourse67    539
    excourse74    539
    BD0145EN     506
    dtype: int64
```

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)

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

```
... 5.737
```

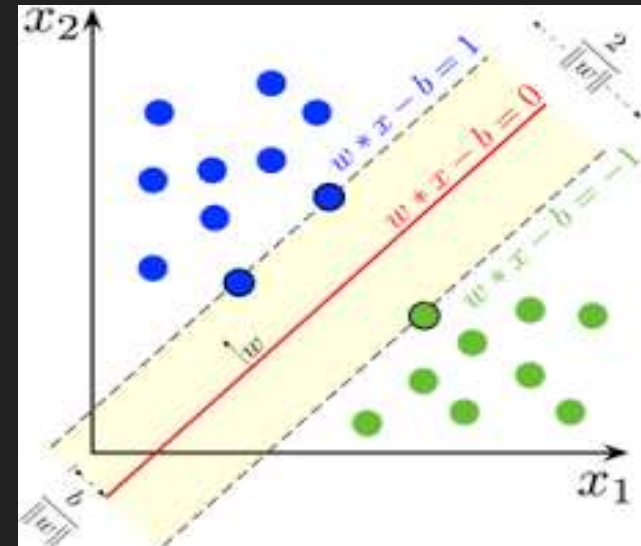
What are the most frequently recommended courses? Return the top-10 commonly recommended courses

```
user_recommendations.iloc[:,1:].sum().sort_values(ascending=False).iloc[:10]
```

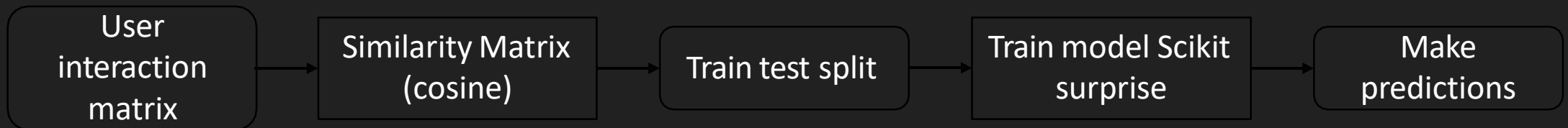
```
[47]
```

```
... DA0101EN    530
     DS0103EN    522
     DS0101EN    449
     BD0101EN    429
     PY0101EN    386
     ML0115EN    359
     DS0105EN    349
     BD0111EN    349
     BC0101EN    325
     ML0101ENV3  313
     dtype: int64
```

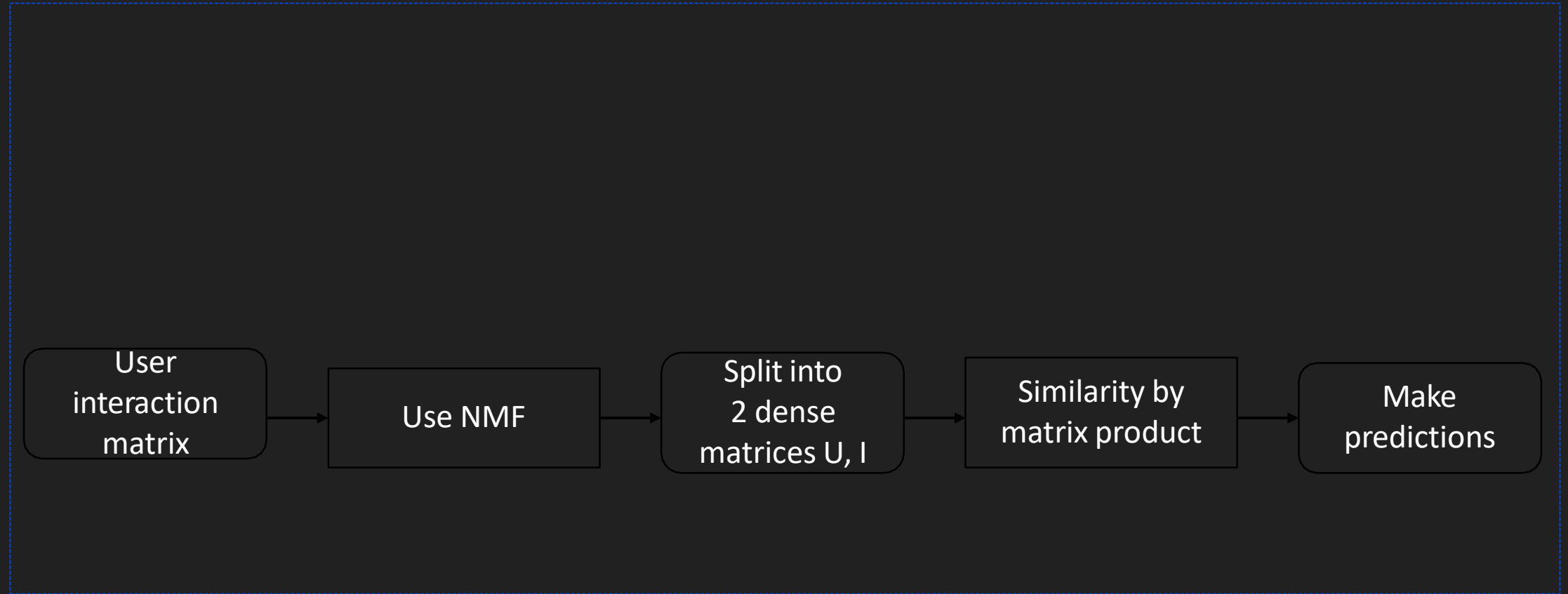
Collaborative-filtering Recommender System using Supervised Learning



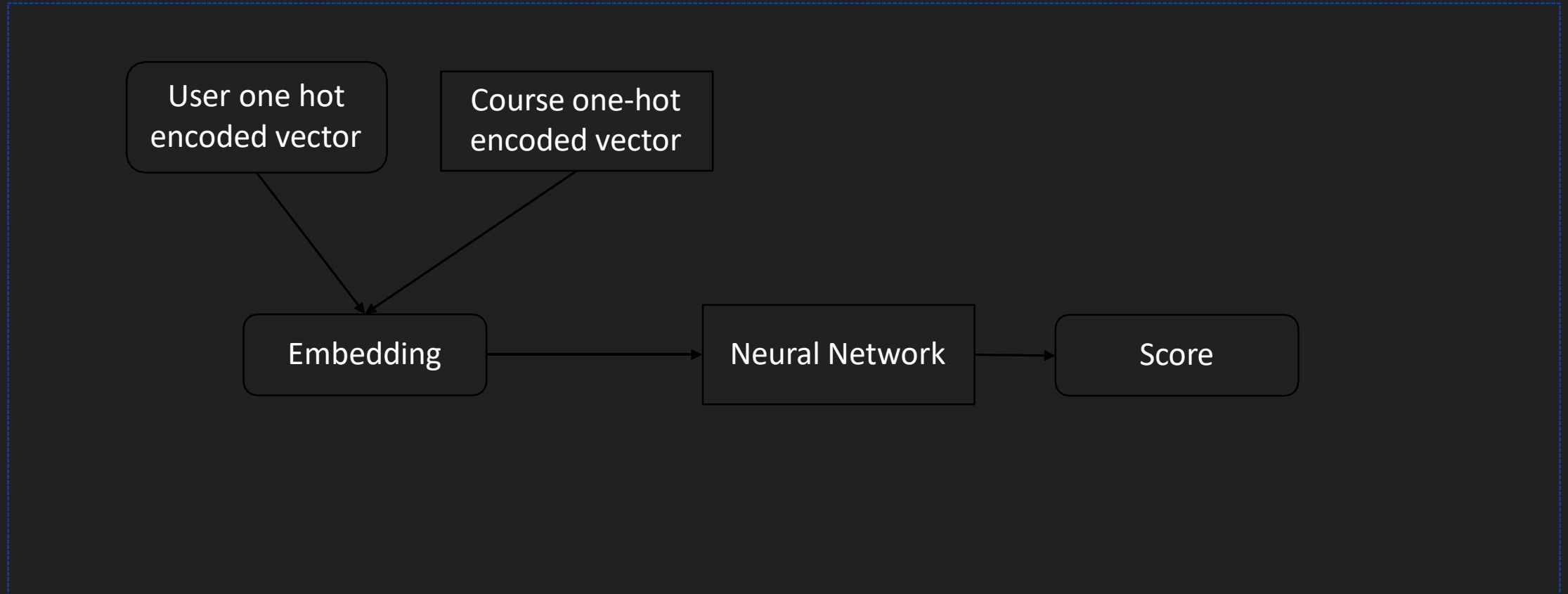
Flowchart of KNN based recommender system



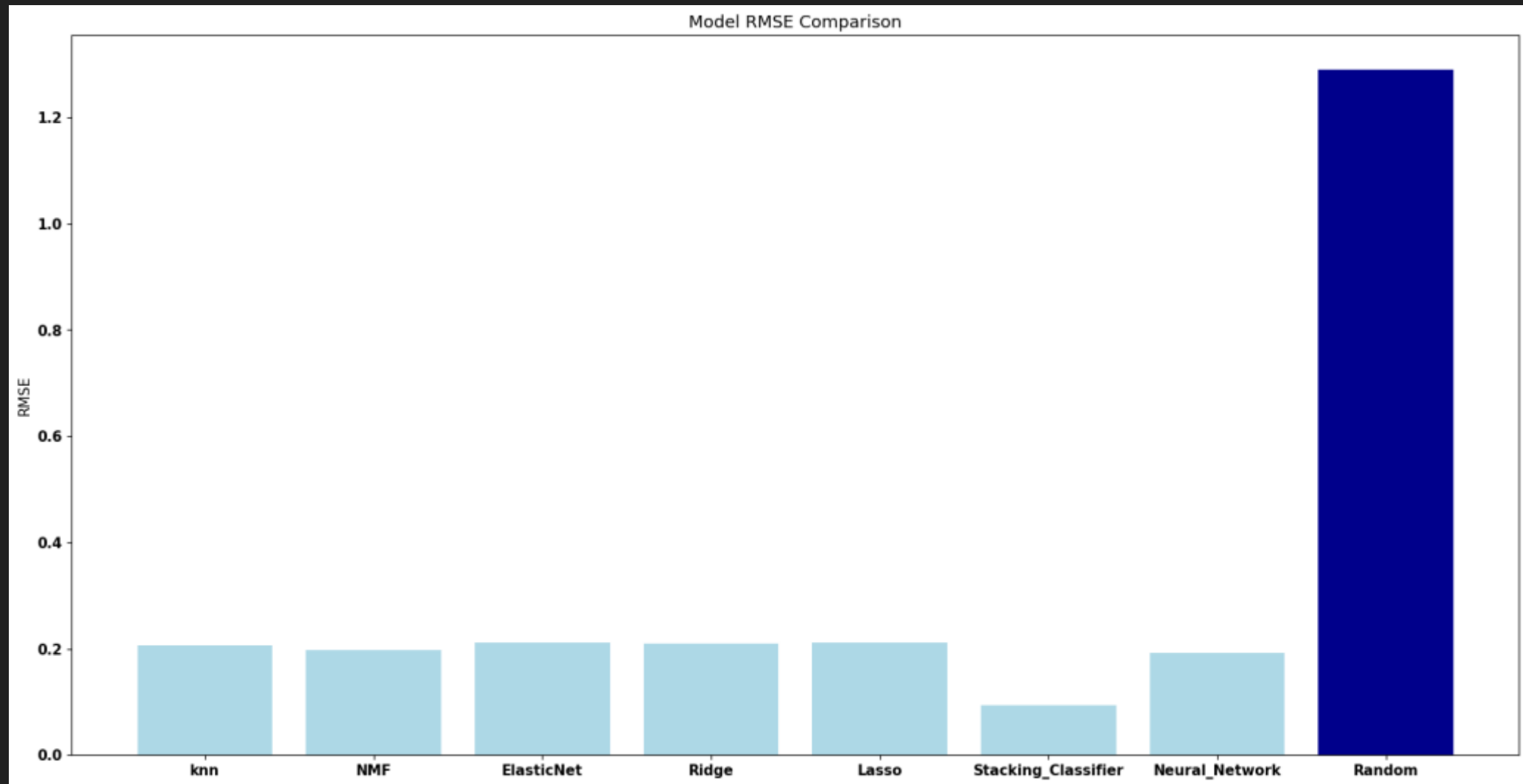
Flowchart of NMF based recommender system



Flowchart of Neural Network Embedding based recommender system



Compare the performance of collaborative-filtering models



Conclusions

- All the model presented have similar results.
- User profile based has the highest number of recommendations
- Stacking Classifier has best performance with 'RMSE' = 0.094647

Appendix

- All materials link
- <https://github.com/ElCahuamo/Coursera-certificates.git>