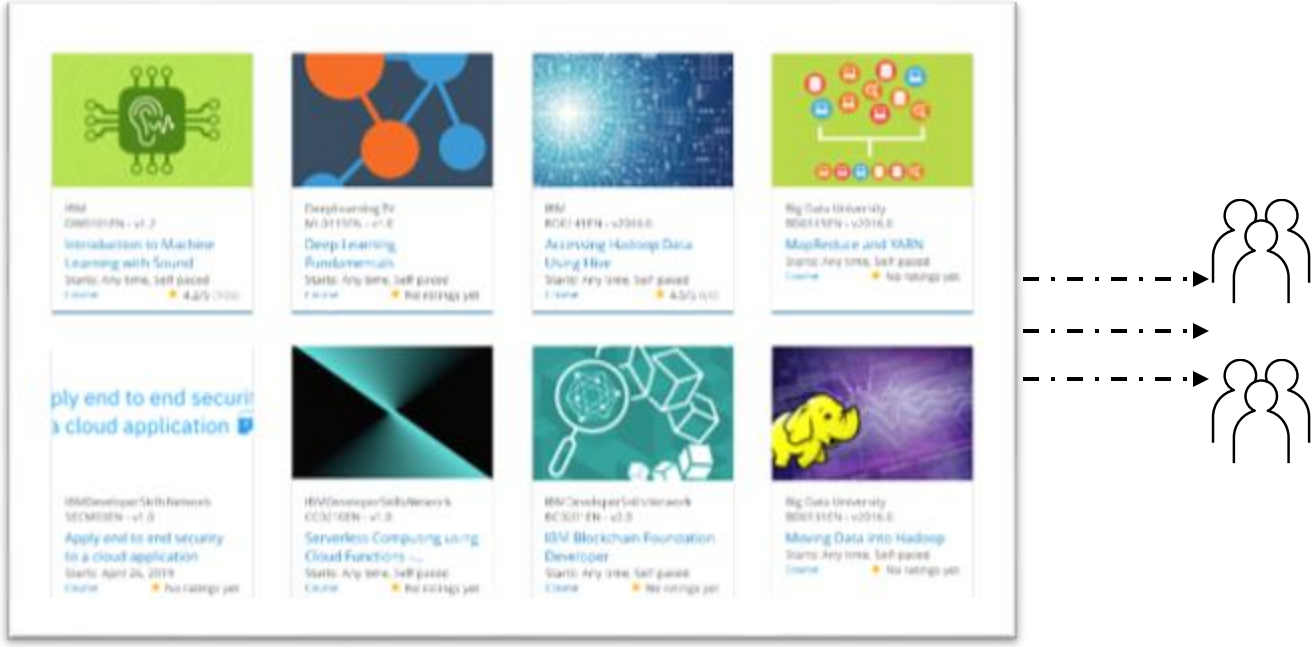
Build a Personalized Online Course Recommender System with Machine Learning

Fariel Shafee
March 5, 2025



Outline

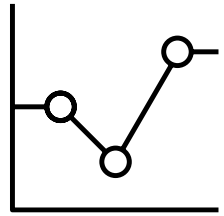
- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

Introduction

- Several methods are used to create a course recommendation system for students
- Popular courses are found and student recommendations are created using clusters, course similarity and student interest
- Several clustering algorithms were tested for efficacy. These include KNN and NNM.
- A neural net is also trained to discover hidden patterns.

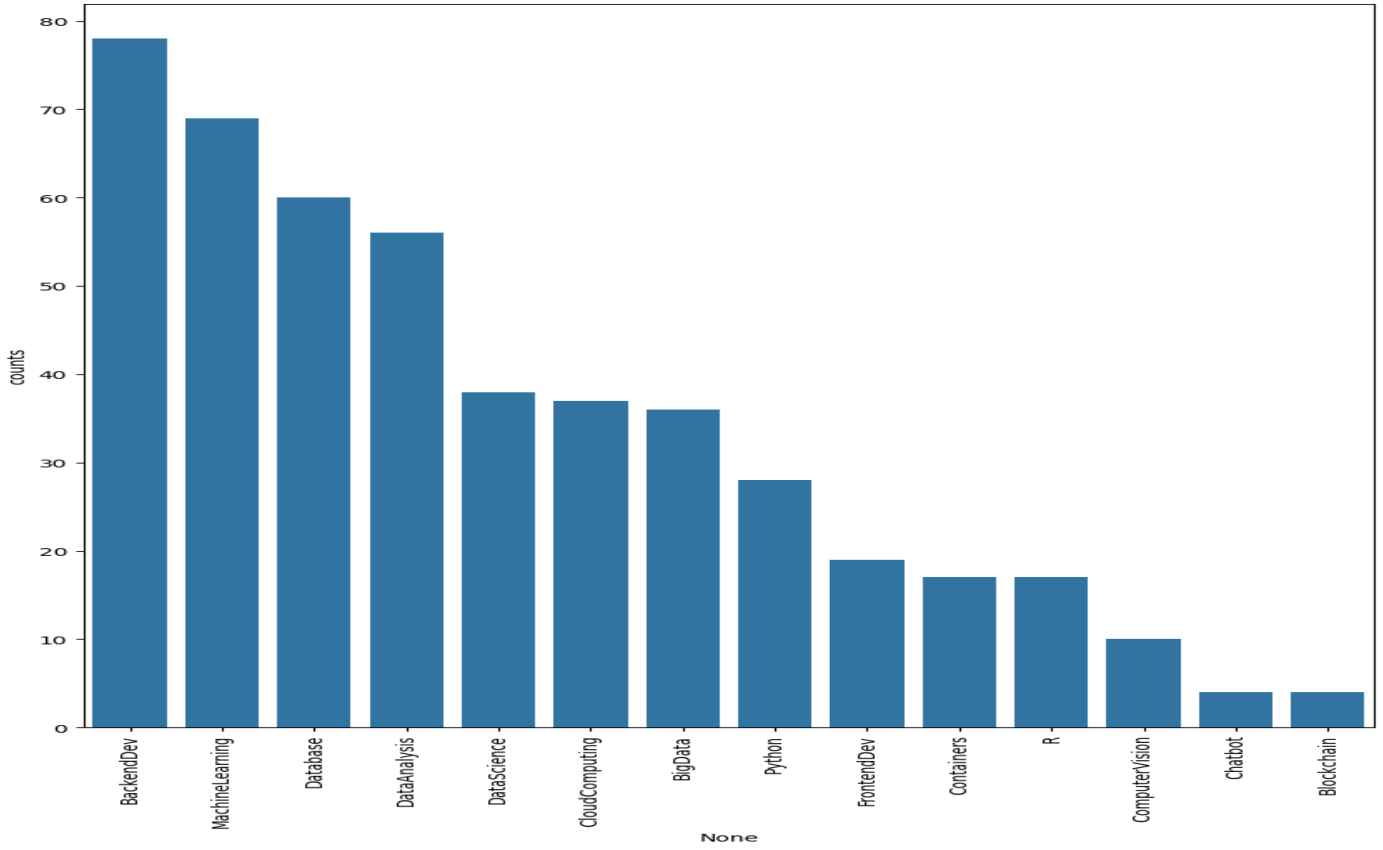
- Hypothesis: a course recommendation system is able to successfully find the popular courses that might align with the student's interest.

Exploratory Data Analysis



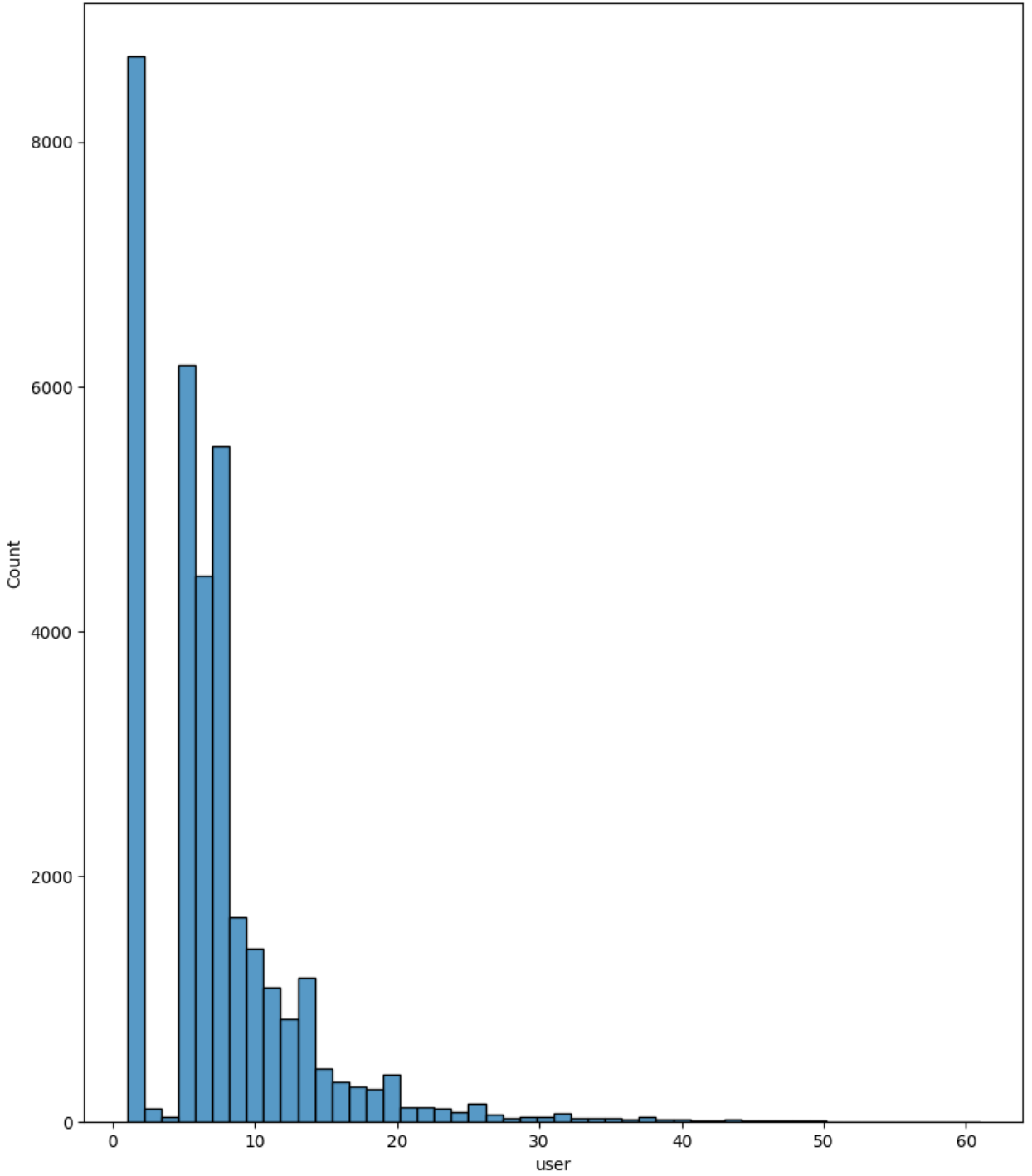
Course counts per genre

- Bar chart shows count of genres in the dataset.
- Each course was split into genres
- Prevalence of genres was analysed



Course enrollment distribution

- Histogram presenting number of courses Enrolled into by number of students.
- Most students enrolled in just one course.



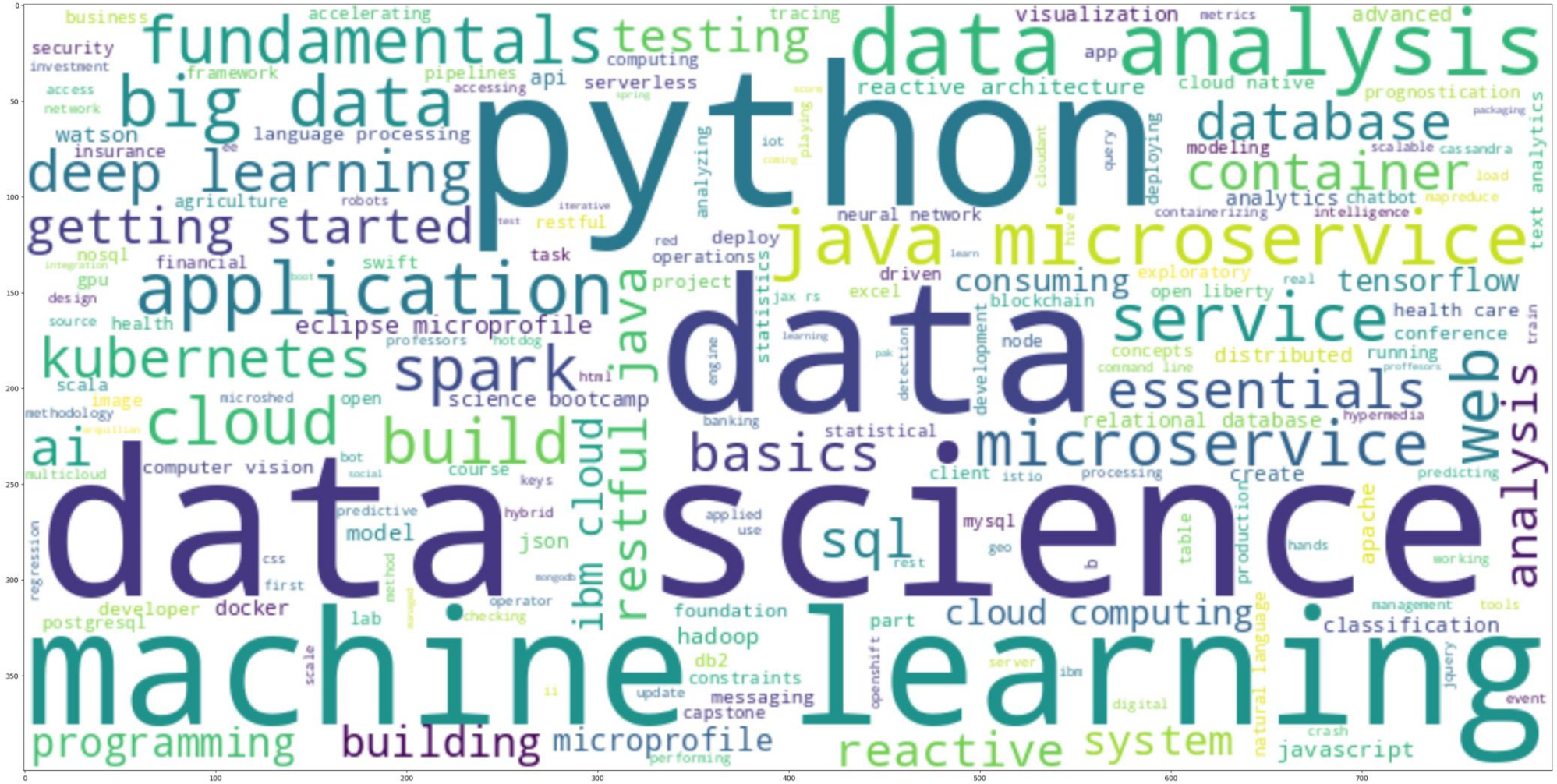
20 most popular courses

	course	ratings	COURSE_ID	TITLE
0	PY0101EN	14936	PY0101EN	python for data science
1	DS0101EN	14477	DS0101EN	introduction to data science
2	BD0101EN	13291	BD0101EN	big data 101
3	BD0111EN	10599	BD0111EN	hadoop 101
4	DA0101EN	8303	DA0101EN	data analysis with python
5	DS0103EN	7719	DS0103EN	data science methodology
6	ML0101ENv3	7644	ML0101ENv3	machine learning with python
7	BD0211EN	7551	BD0211EN	spark fundamentals i
8	DS0105EN	7199	DS0105EN	data science hands on with open source tools
9	BC0101EN	6719	BC0101EN	blockchain essentials
10	DV0101EN	6709	DV0101EN	data visualization with python
11	ML0115EN	6323	ML0115EN	deep learning 101
12	CB0103EN	5512	CB0103EN	build your own chatbot
13	RP0101EN	5237	RP0101EN	r for data science
14	ST0101EN	5015	ST0101EN	statistics 101
15	CC0101EN	4983	CC0101EN	introduction to cloud
16	CO0101EN	4480	CO0101EN	docker essentials a developer introduction
17	DB0101EN	3697	DB0101EN	sql and relational databases 101
18	BD0115EN	3670	BD0115EN	mapreduce and yarn
19	DS0301EN	3624	DS0301EN	data privacy fundamentals

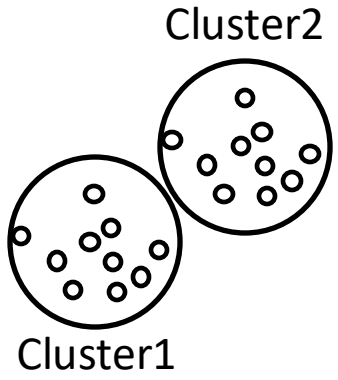
Word cloud of course titles

Frequencies of student interest are easily visualized using a word cloud.

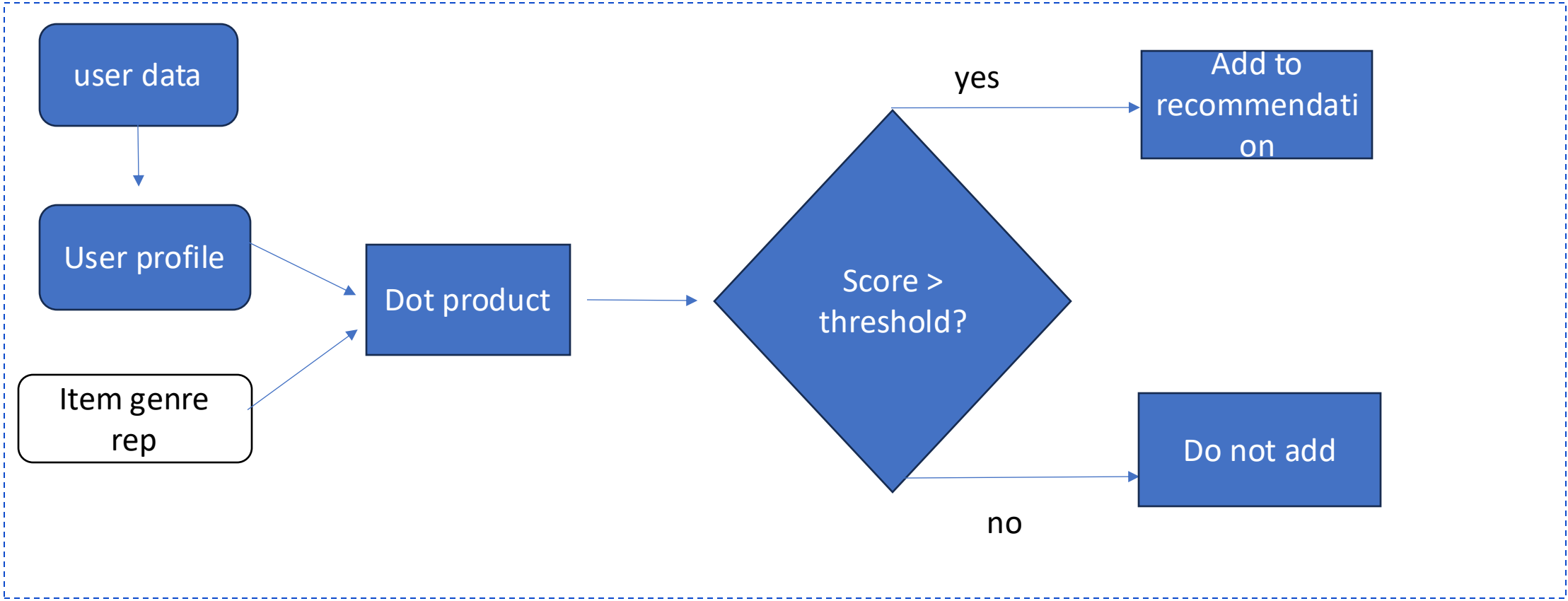
These interests are later used to find popular but similar courses



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

Cutoff threshold is 10

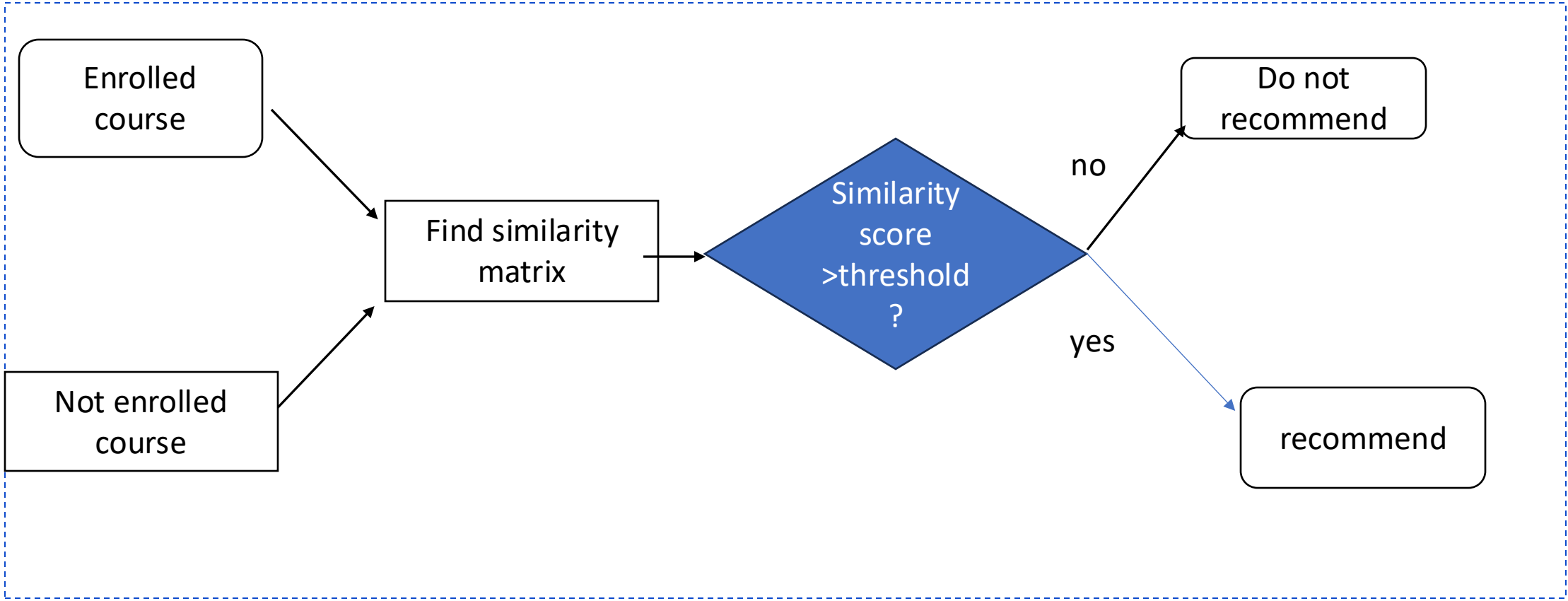
```
average recommendation is
60.82471217772012

# find average number of recommendation

total_counts = res_df['USER'].count()
total_users =
len(res_df['USER'].value_counts())
ave = total_counts/total_users
print('average recommendation is', ave)
```

0	2	ML0201EN	43.0
1	2	GPXX0ZG0EN	43.0
2	2	GPXX0Z2PEN	37.0
3	2	DX0106EN	47.0
4	2	GPXX06RFEN	52.0
5	2	GPXX0SDXEN	37.0
6	2	CC0271EN	58.0
7	2	WA0103EN	49.0
8	2	DX0108EN	43.0
9	2	GPXX0PICEN	14.0

Flowchart of content-based recommender system based on course similarity



Evaluation results of course similarity based recommender system

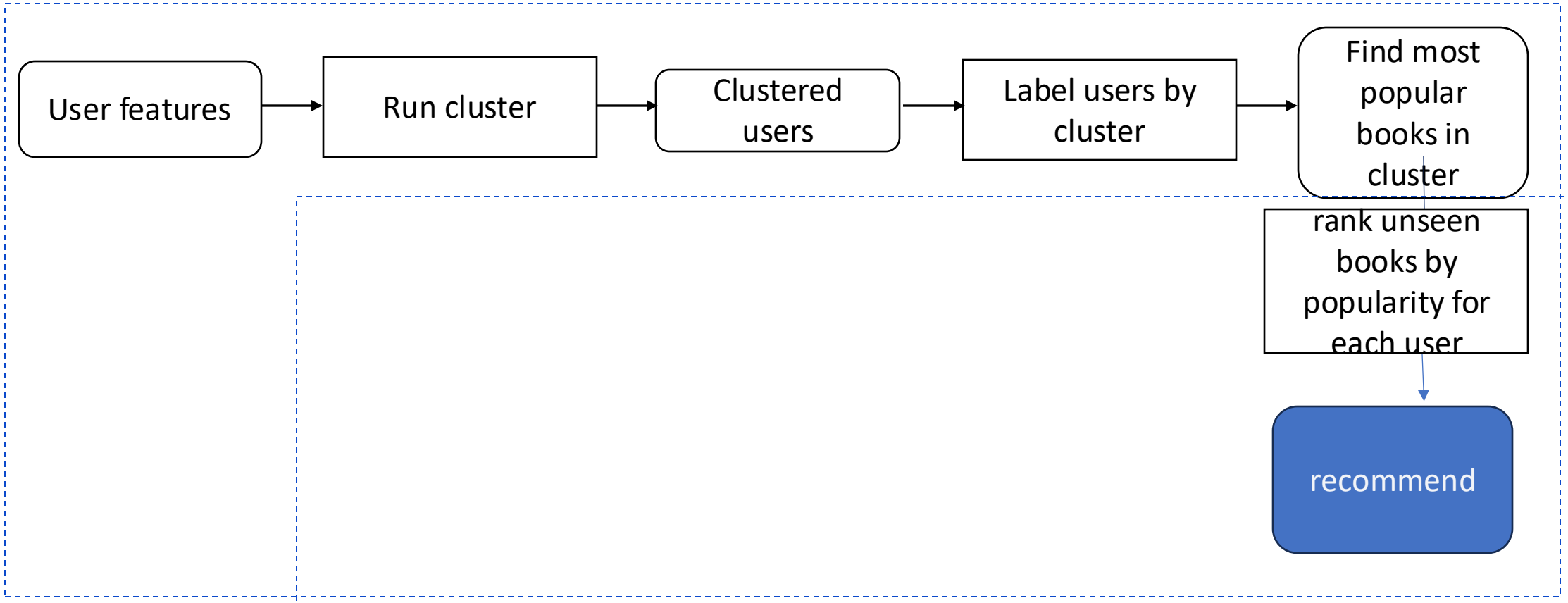
Threshold, alpha, is .6

```
average recommendations
1.076398926285360 4
```

```
s = 0
total = len(res_df['USER'])
for i in range(total):
    number = len(res_df['COURSE_ID'].iloc[i])
    s+= number
average= s/total
print('average recommendations  ', average)
```

2	[WA0103EN]	[0.6311528416041716]
4	[WA0101EN]	[0.6311528416041716]
5	[WA0101EN]	[0.63115416041716]
7	[]	[]
8	[]	[]
9	[]	[]
12	[]	[]
16	[excourse22, excourse62]	[0.6475015976638527, 0.6475015976638527]
17	[TMP0101EN, TA0105EN, BD0151EN]	[0.8894991799933215, 0.6598288790738579, 0.630…
19	[]	[]

Flowchart of clustering-based recommender system



Evaluation results of clustering-based recommender system

Threshold value = 10

Optimal cluster = 9

```
recommendations, total =
generate_course_recommendation
s(test_users_labelled,
courses_cluster_grouped)

s = 0
total = 0
for r,m in
recommendations.items():
    s += len(m)
    total = total + 1
print('average number of
recommendations is',  s/total)
```

```
average number of recommendations
is 87.6848765523141
```

CB0101EN: 33679

ML0120ENv3: 33597

DJ0101EN: 33539

BD0153EN: 33516

DS0201EN: 33496

EE0101EN: 33478

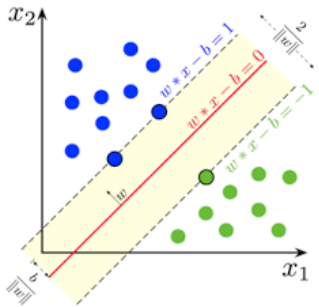
IT0101EN: 33444

ML0111EN: 33439

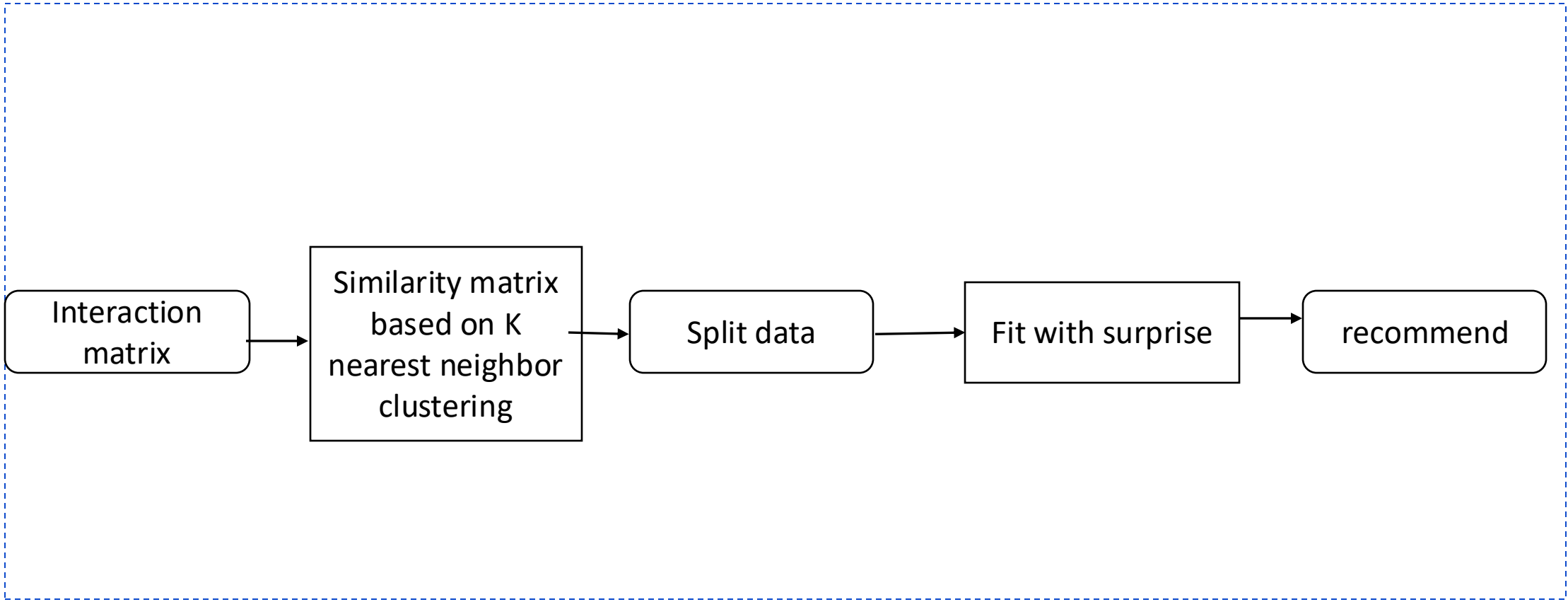
DE0205EN: 33417

CC0250EN: 33404

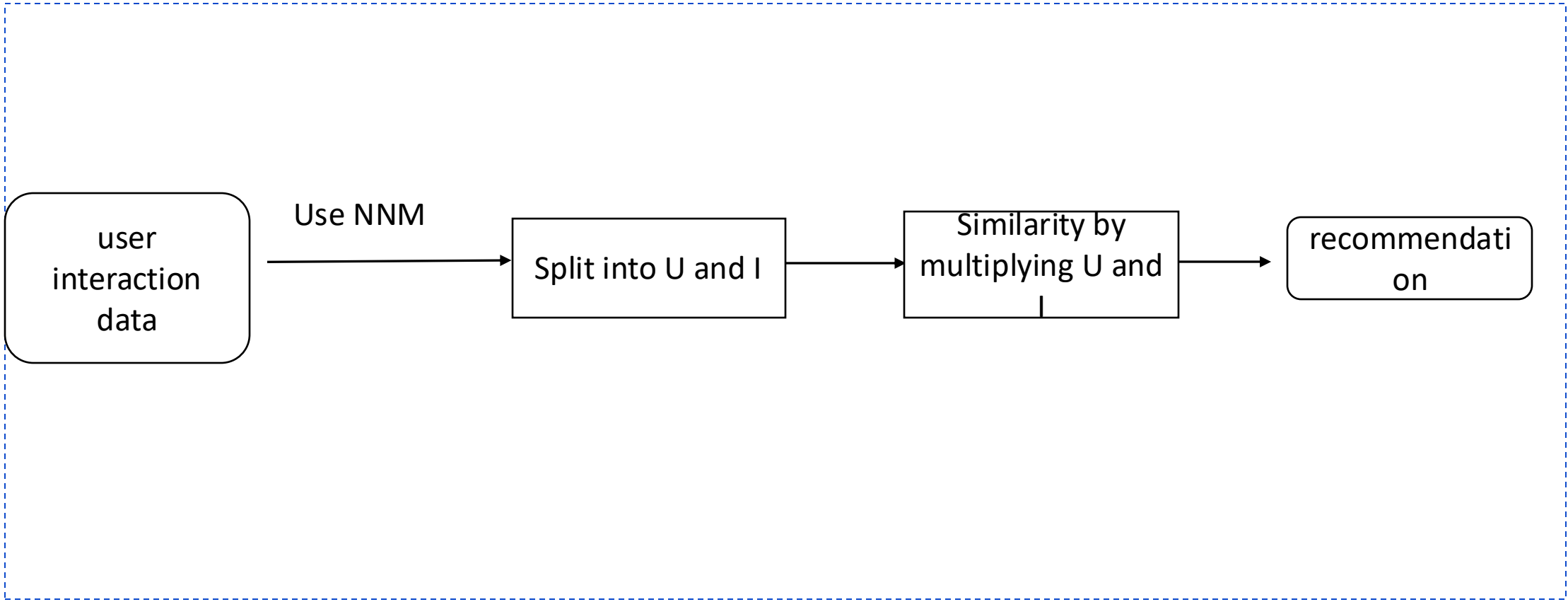
Collaborative-filtering Recommender System using Supervised Learning



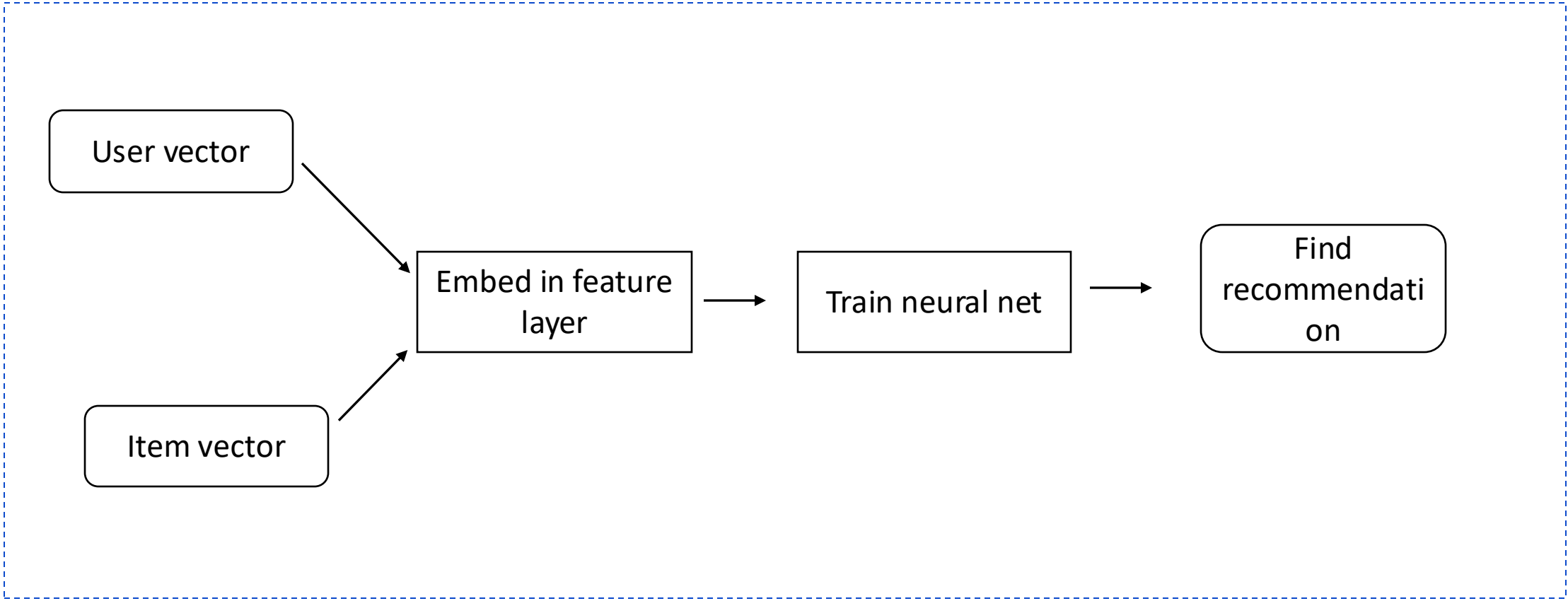
Flowchart of KNN based recommender system



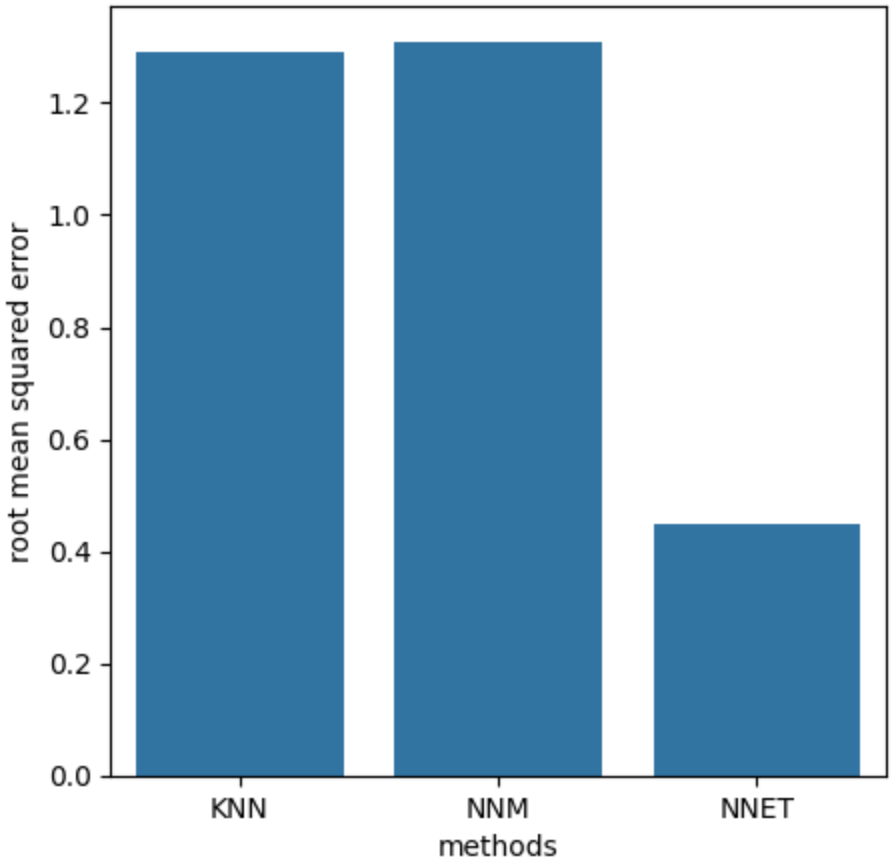
Flowchart of NNM based recommender system



Flowchart of Neural Net based recommender system



Compare the performance of collaborative-filtering models



Our root mean squared error for the neural net was much lower than those for K Nearest Neighbors or NNMs

Conclusions

- Our models are capable of recommending courses of interest to students
- Similarity based recommendation systems gave very few recommendations on an average, especially for people who are not already enrolled in many courses.
- On the other hand, using a neural network was the most efficient method for prediction.

- …

Appendix

Check

https://builtin.com/data-science/recommender-systems for a well written introduction to Recommender systems.

- …