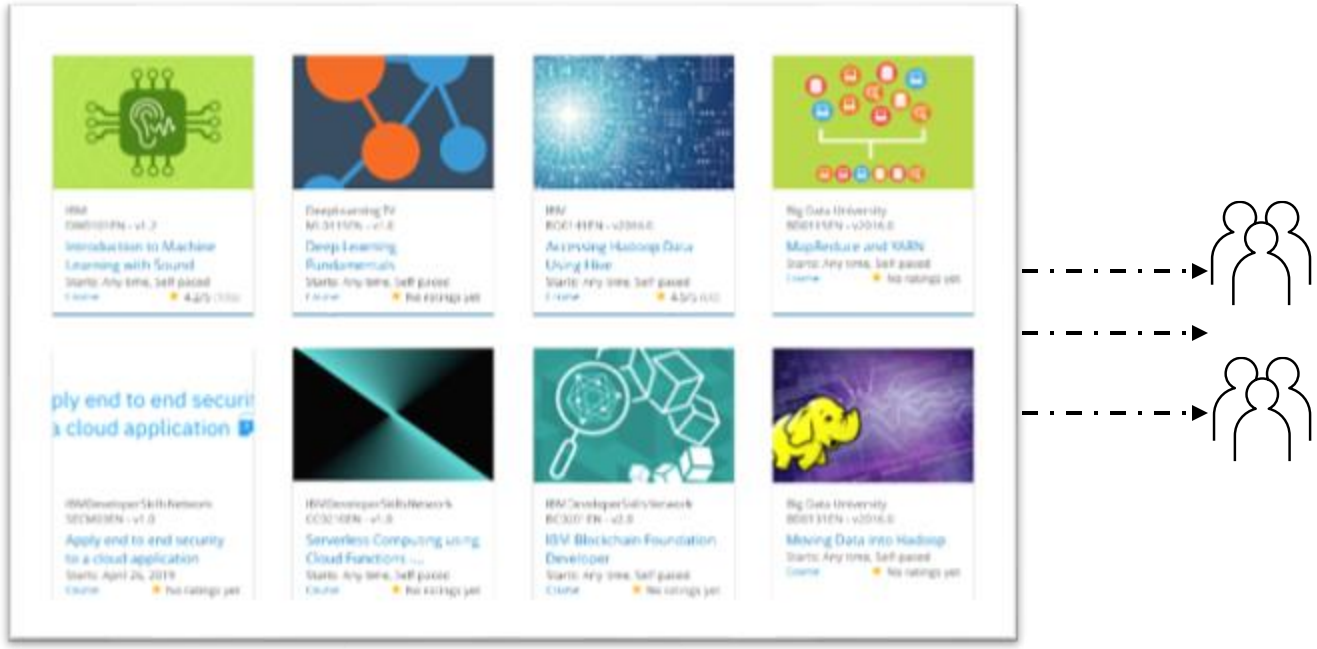
Build a Personalized Online Course Recommender System with Machine Learning

<Name>Wayne LI



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

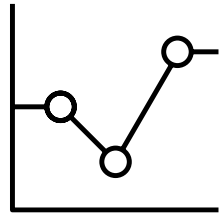
- Project background:

These days, many online service like shopping websites, streamline, Ytube, social media all use recommendation systems to provides their users better experience and give the users something they want or need. The techniques behind the scene are actually related to the machine learning or nueral network methods. In this project, I learn to use some ways to build a recommendation system. It can be based on supervised,unsupervised learning or neural network. It can also be content-based or collaborative-based depend on your goal and targets.

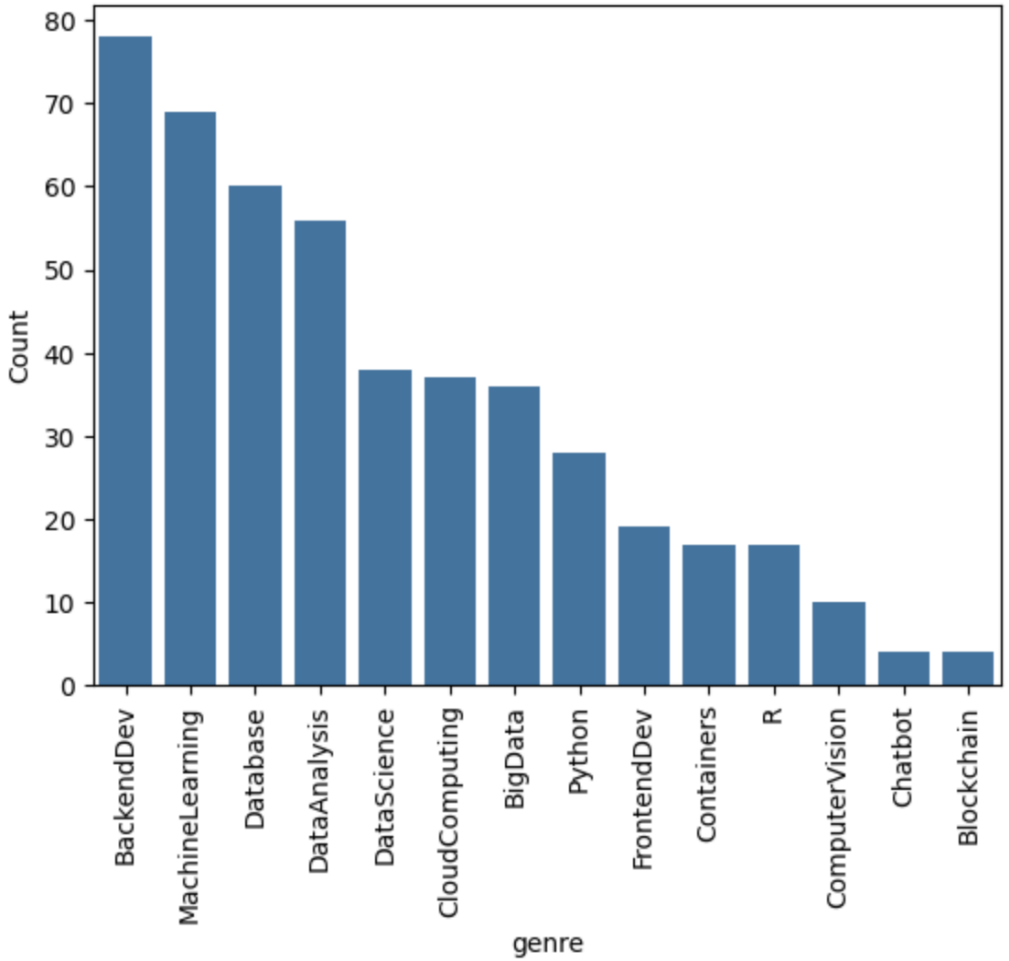
- Problem states and hypotheses

In this project, I practice using recommendation system to predict the courses users may be interested in. I use content-based according to the profile,interest of user and the content or feature of certain classes. I compare the similarity between them and do best recommendation. I also use collaborative-filtering systems to recommend using the history rating of the users.

Exploratory Data Analysis



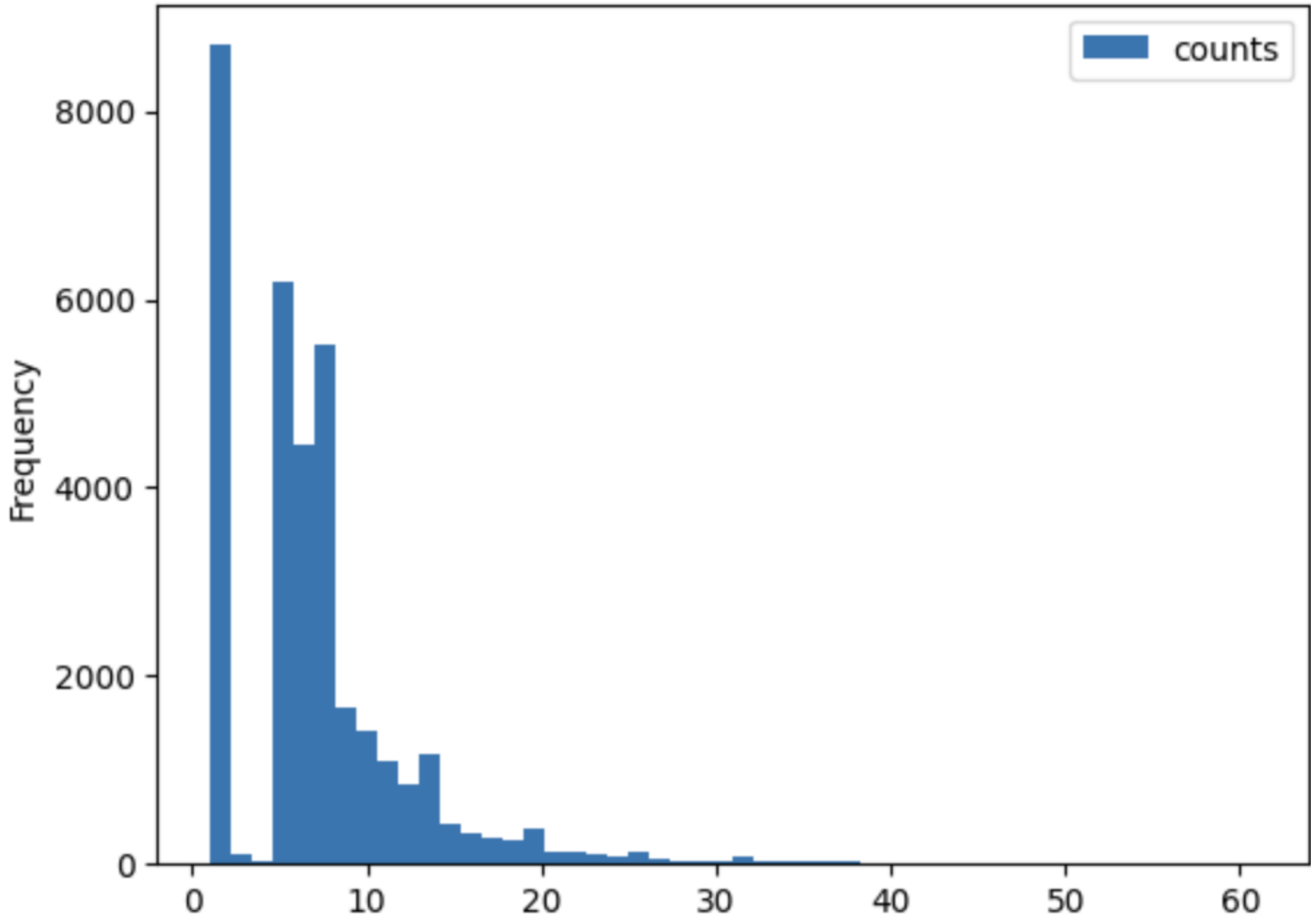
Course counts per genre



From this plot, we see that the most courses provided are related to backend,machinelearning and Database. This might implicitly show that these categories are popular genre to students.

Course enrollment distribution

This is a histogram for student enrollments. It shows that most students enroll in less than ten courses.



20 most popular courses

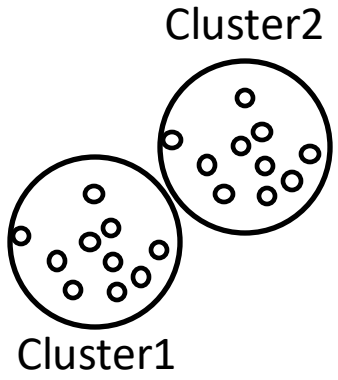
	TITLE	Ratings
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199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

The most popular courses are about data science, machine learning, big data,data analysis,database,etc.

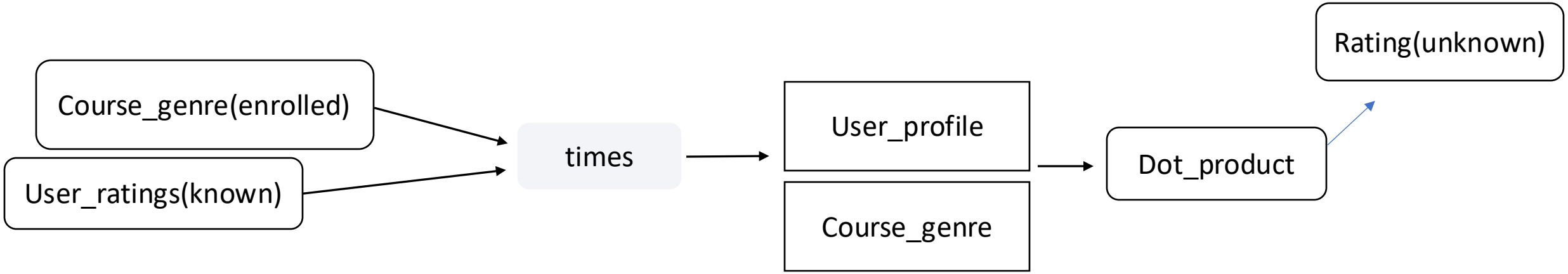
Word cloud of course titles

The wordcloud is generated from course titles. The size is related to the frequency the word appears. It shows many courses mention data science, machine learning,python,data analysis,big data.

Content-based Recommender System using Unsupervised Learning



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



Steps:

- Calculate user_profile(interest/like category)/ or by using history rating times history enrolled course genre
- Next,calculate the dot product of the user_profile(content) and the course genre that you want to rate
- Use a score to filter and choose the course above this rating score as the recommended courses

result

	USER	COURSE_ID	SCORE
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
...	...	...	...
1500419	2102680	excourse62	15.0
1500420	2102680	excourse69	14.0
1500421	2102680	excourse77	14.0
1500422	2102680	excourse78	14.0
1500423	2102680	excourse79	14.0

Evaluation results of user profile-based recommender system

Place your hyper-parameter settings, such as recommendation score or course similarity thresholds, etc.

Score_threshold=10

On average, 61 new/unseen courses have been recommended per user (in the test user dataset)

```
res_user=res_df.USER.unique().tolist()
num = res_df['USER'].value_counts().tolist()
np.mean(num)
```

✓ 0.0s

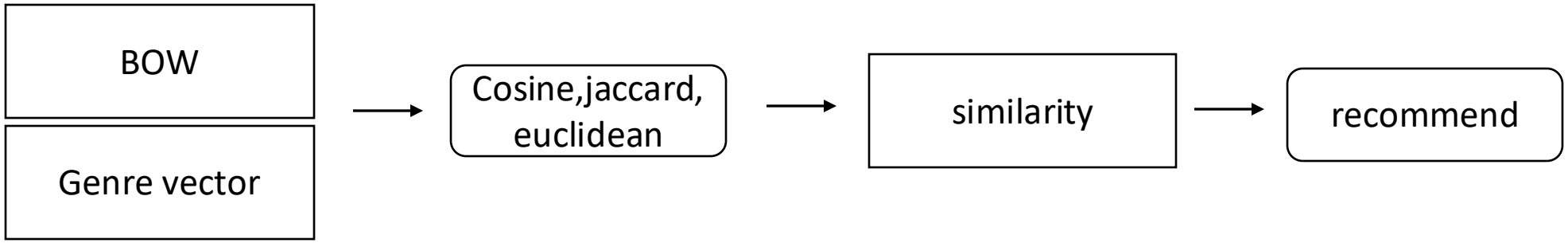
```
60.82471217772012
```

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

```
COURSE_ID
TA0106EN      17390
excourse21    15656
excourse22    15656
GPXX0IBEN     15644
ML0122EN      15603
excourse04    15062
excourse06    15062
GPXX0TY1EN    14689
excourse73    14464
excourse72    14464
dtype: int64
```

Flowchart of content-based recommender system using course similarity

- Plot a flowchart which should clearly illustrate how you implemented the course similarity based recommender system
- Briefly explain the flowchart in the slide note



Steps:

- First find the data of the courses like genre or topics and use it to calculate the similarity between courses
- The main idea is that we prefer courses with high similarity
- Then we also set some thresholds to filter out the recommended courses

Evaluation results of course similarity based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

Threshold=0.3 if two courses have similarity larger than it then recommend

On average, 25 new/unseen courses have been recommended per user (in the test user dataset)

```
course_id=res_df.COURSE_ID.tolist()
num=[len(num) for num in course_id ]
np.mean(num)
```

✓ 0.0s

```
24.917701542727354
```

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

	course	counts
15	excourse32	18857
14	excourse68	18034
17	excourse36	17893
16	excourse23	17893
12	excourse67	17875
26	excourse38	17808
18	excourse33	17783
25	excourse04	17388
19	DS0110EN	17188
27	excourse09	17082

Flowchart of clustering-based recommender system

Plot a flowchart which should clearly illustrate how you performed user profile clustering based recommender system

- Briefly explain the flowchart in the slide note

I use user profile/interest/content to do cluster, similar user will be in the same group then I recommend each user the most popular course in their group

Steps:

- We have user to course,course to course, we also have last user to user similarity. This time we use cluster to group similar users together and recommend users in the same group with popular items in the group.
- We can use a variety of cluster methods like kmeans,dbscan,hierachy cluster,etc
- In this case, I use kmeans

Evaluation results of clustering-based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

I use kmeans n_cluster=10 based the elbow point on the inertia graph

On average, 33 new/unseen courses have been recommended per user (in the test user dataset)

```
from col (variable) result: dict
id=list(result.values())
num=[len(i) for i in id]
np.mean(num)
```

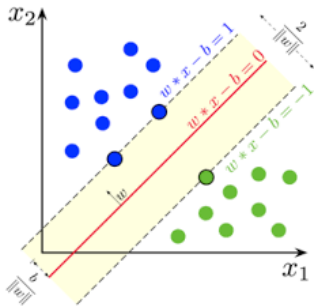
✓ 0.0s

```
33.30058144682861
```

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

	course	counts
32	SC0101EN	31162
38	WA0101EN	31041
15	CL0101EN	30308
64	DS0301EN	29679
43	DB0101EN	29599
41	CO0101EN	29408
55	CC0101EN	28906
35	ST0101EN	28298
25	ML0103EN	28133
31	RP0101EN	28069

Collaborative-filtering Recommender System using Supervised Learning



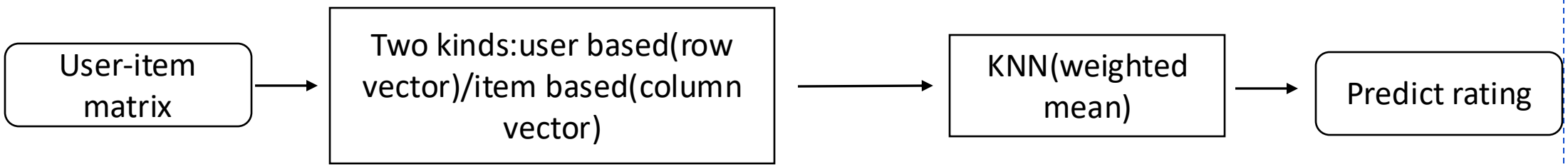
Collaborative_filtering:

Collaborative filtering is based on the users' history data. You don't have to know the features or content of certain user or item. What you need is the activity of the users,such as rating, clicking rate, viewing times.

We usually use user-item matrix during the process

Flowchart of KNN based recommender system

- Plot a flowchart which should clearly illustrate how you performed KNN based recommender system using course enrollments history
- Briefly explain the flowchart in the slide note



Steps:

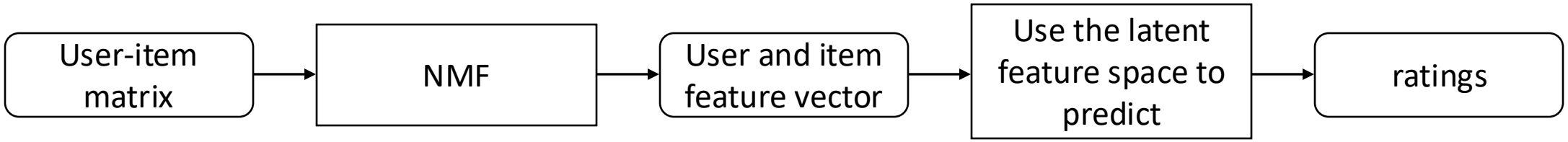
- I use user-item matrix ,set our target rating as y other features as X.
- I use KNNRegressor or the KNNBasic in surprise library(a library specialized for recommendation system) to predict the unknown rating on the certain user of certain item.
- Based on the predicted ratings, we could decide if recommendation is needed or not.

Flowchart of NMF based recommender system

- Plot a flowchart which should clearly illustrate how you performed NMF based recommender system

- Briefly explain the flowchart in the slide note

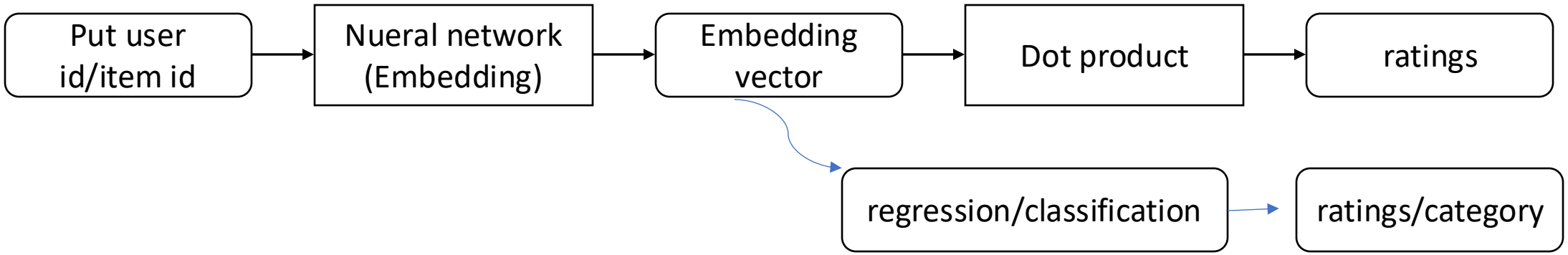
NMF will decompose the user-item matrix to separate user and item matrix with lower dimension. With using the latent space learned from training, we can predict the rating of users.



Flowchart of Neural Network Embedding based recommender system

- Plot a flowchart which should clearly illustrate how you performed Neural Network Embedding based recommender system

- Briefly explain the flowchart in the slide note



Conclusions

- Content based:

The recommendation based on content or features between users and items through compare similarity

Methods:clustering

- Collaborative- fitering:

Based on history data, rather than the content or features. It will use user-item matrix to record the activity of users.

Methods: KNN, NMF, neuaral network or neural network+regression/classification

Appendix

- Python surprise library(recommendation system)
- Python sklearn library(machine learning)
- Tensorflow,keras(nueral network)
- Basic python pandas, matplotlib