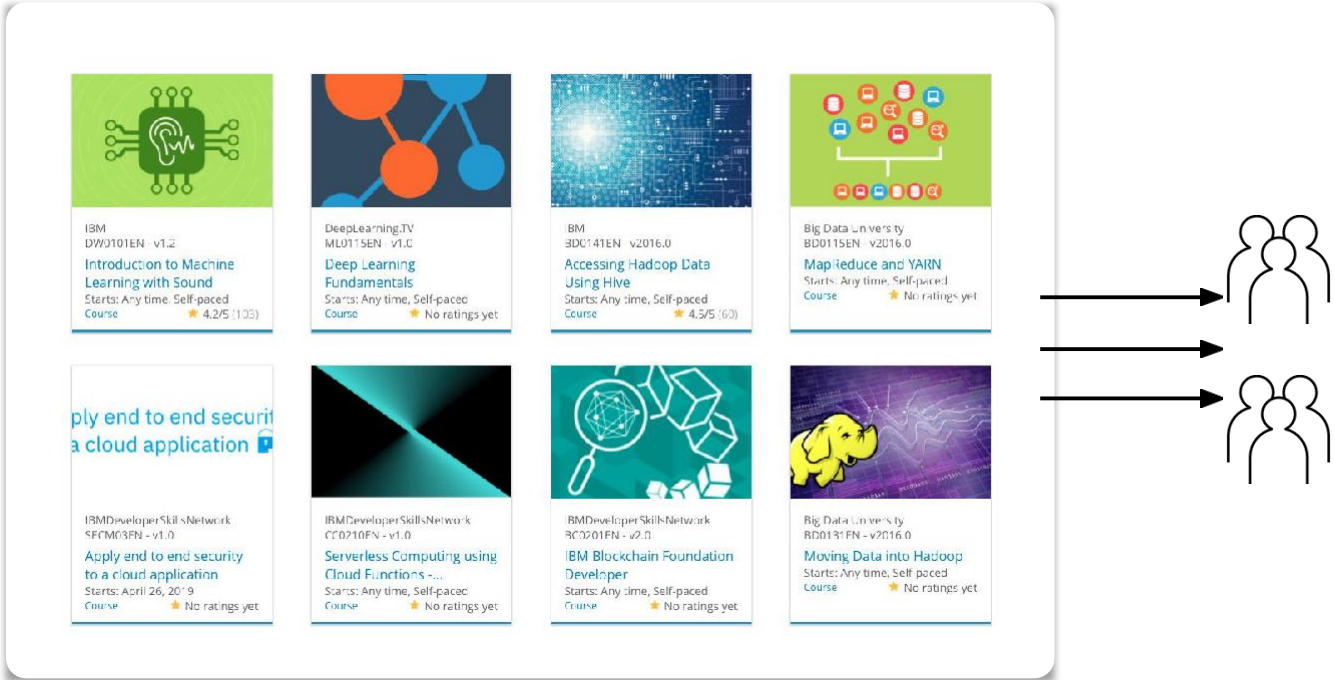
Build a Personalized Online Course Recommender System with Machine Learning

<Atheenthara Ram S>
<2022-08-17>



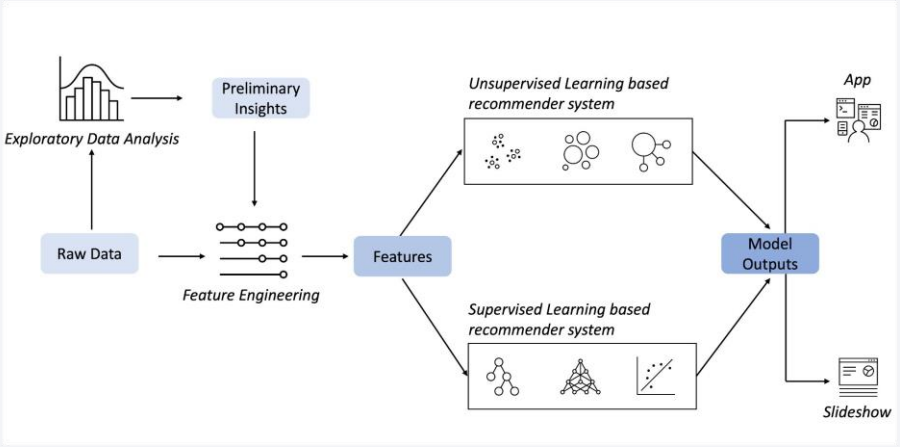
Outline

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

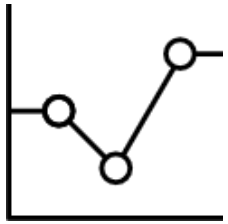
Introduction

- Project background and context
  - As a new machine learning engineer in a Massive Open Online Courses (MOOCs) startup called AI Training Room. In AI Training Room, learners across the world can learn leading technologies such as Machine Learning, AI, Data Science, Cloud, App development, etc. Your company grows rapidly and reaches millions of learners in a very short period.

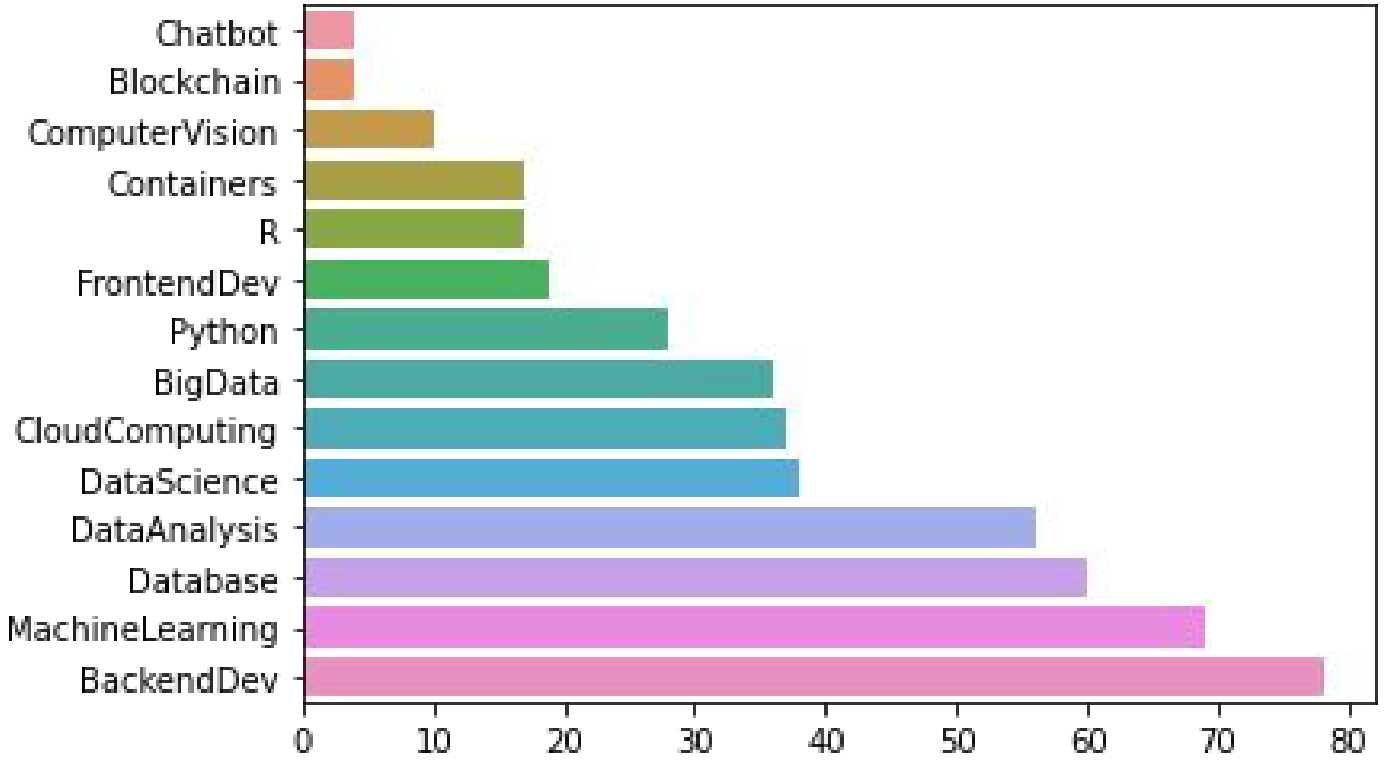
- Problem states and hypotheses
  - Collecting and understanding data
  - Performing exploratory data analysis on online course enrollments datasets
  - Extracting Bag of Words (BoW) features from course textual content
  - Calculating course similarity using BoW features
  - Building content-based recommender systems using various unsupervised learning algorithms, such as:
    - Distance/Similarity measurements, K-means, Principal Component Analysis (PCA), etc.
  - Building collaborative-filtering recommender systems using various supervised learning algorithms
    - K Nearest Neighbors, Non-negative Matrix Factorization (NMF), Neural Networks, Linear Regression, Logistic Regression, RandomForest, etc.
  - Creating an insightful and informative slideshow and presenting it to your peers



Exploratory Data Analysis



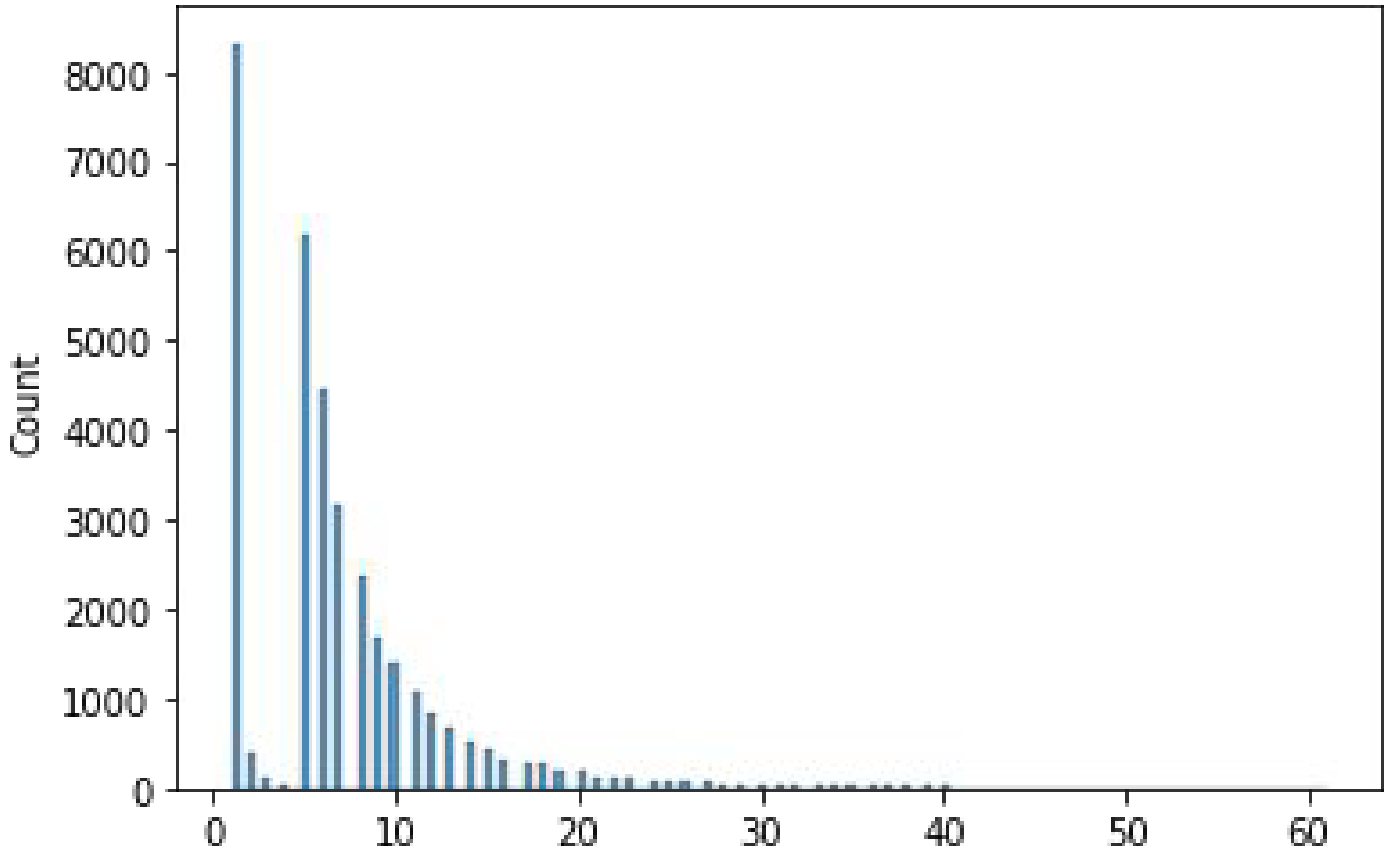
Course counts per genre



Chatbot	4
Blockchain	4
ComputerVision	10
Containers	17
R	17
FrontendDev	19
Python	28
BigData	36
CloudComputing	37
DataScience	38
DataAnalysis	56
Database	60
MachineLearning	69
BackendDev	78

- The most popular genre = BackendDev
- The least popular genre = Chatbot
- The most popular programming language = Python
- R is the second most popular language among all the courses

Course enrollment distribution



count	33901.000000
mean	6.881980
std	5.823548
min	1.000000
25%	2.000000
50%	6.000000
75%	9.000000
max	61.000000

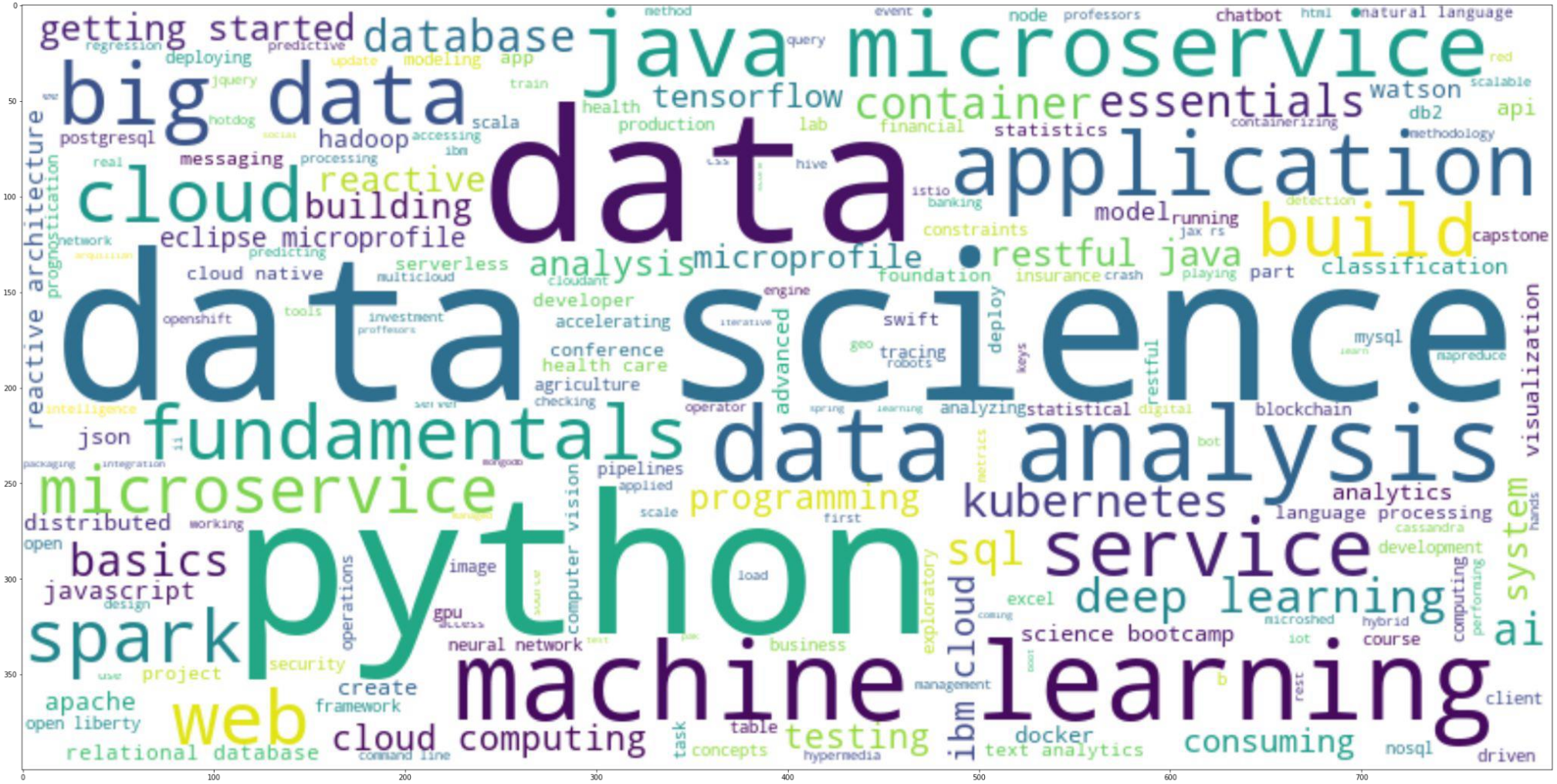
- there are 33901 users in total
- the mean number of rates per user = 6.88
- the max number of rates per user = 61

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

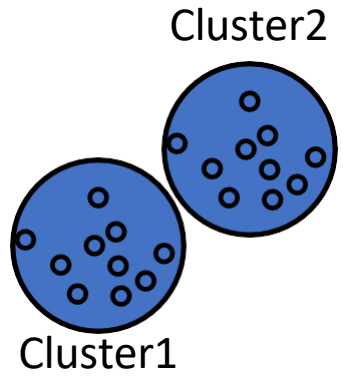
- most popular course = python for data science
- the number of most popular course enrolls = 14936
- most of the topics are related to data science & big data
- R language is less popular than Python (5237 enrolls)

Word cloud of course titles

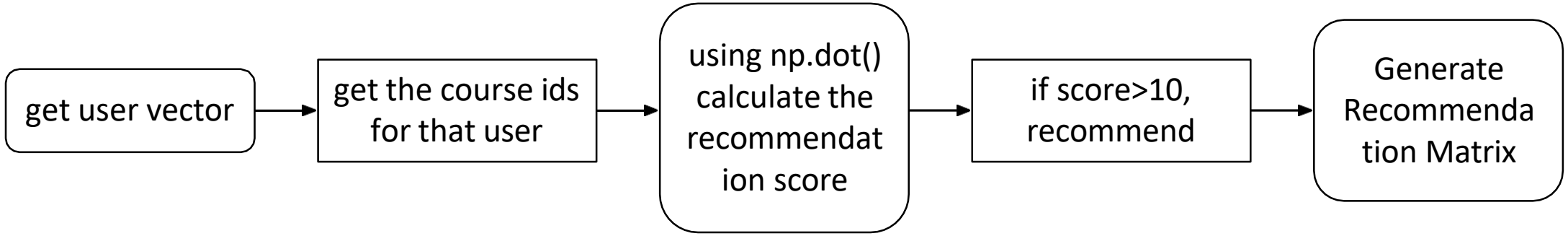


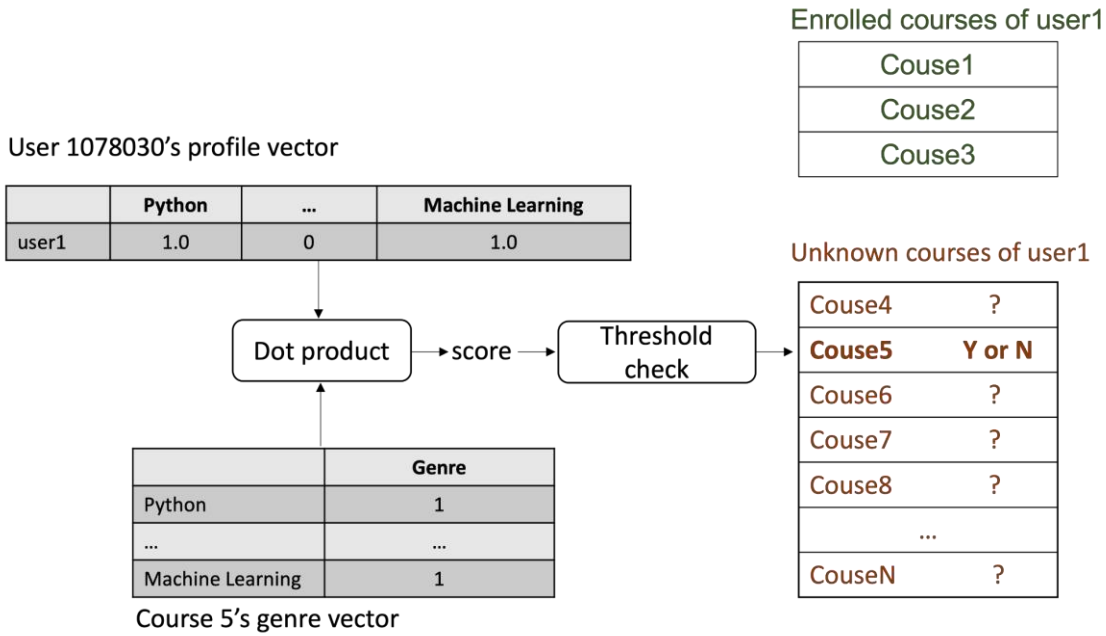
- the most frequent word = Data Science
- Python is also popular term in course title
- topic related to Data Science and Machine Learning are generally the popular courses
- traditional direction such as Java and Web technology are less popular compare to AI related courses

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

```
# The threshold can be fine-tuned to adjust the size of generated recommendations
score_threshold = 10.0
```

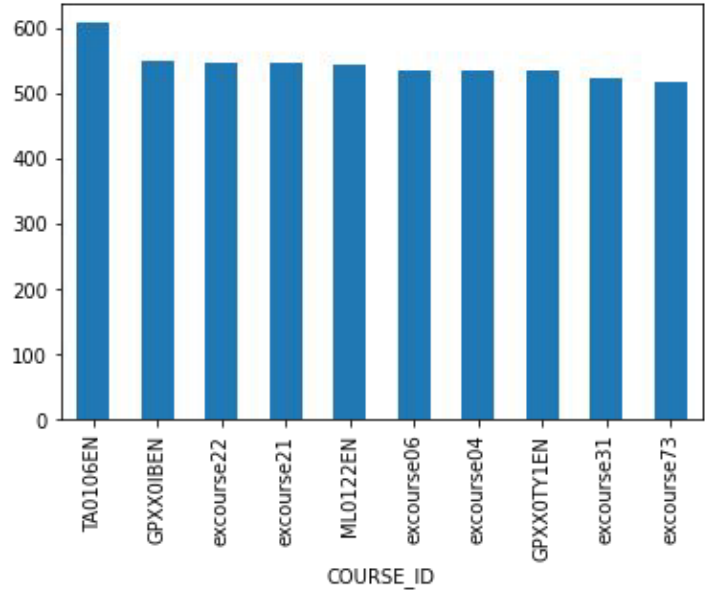
On average, how many new/unseen courses have been recommended per user (in the test user dataset)?

Answer= 61.82

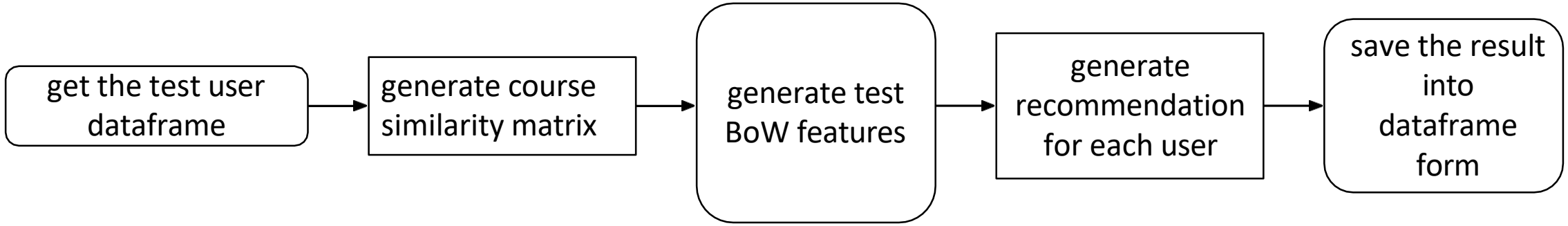
```
1 res_df.groupby(by='USER').size().mean()
```

```
61.81828703703704
```

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



Flowchart of content-based recommender system using course similarity

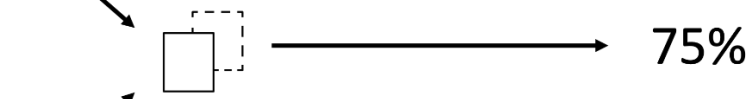


Course 1: “Machine Learning for Everyone”

	machine	learning	for	everyone	beginners
course1	1	1	1	1	0

Course 2: “Machine Learning for Beginners”

	machine	learning	for	everyone	beginners
course2	1	1	1	0	1



Similarity Calculation:
Cosine, Euclidean, Jaccard index, …

Evaluation results of course similarity based recommender system

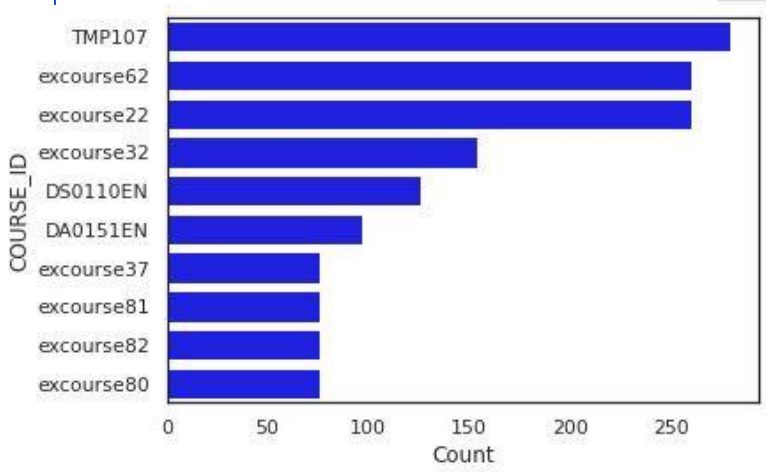
Your hyper-parameter settings, such as a score or similarity threshold

```
threshold = 0.5
```

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

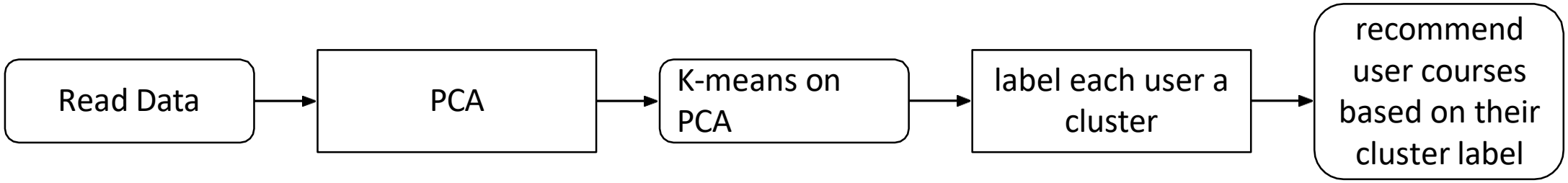
```
[4]  1 len(res_df['COURSE_ID'].sum())/len(res_df)

     2.392
```

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

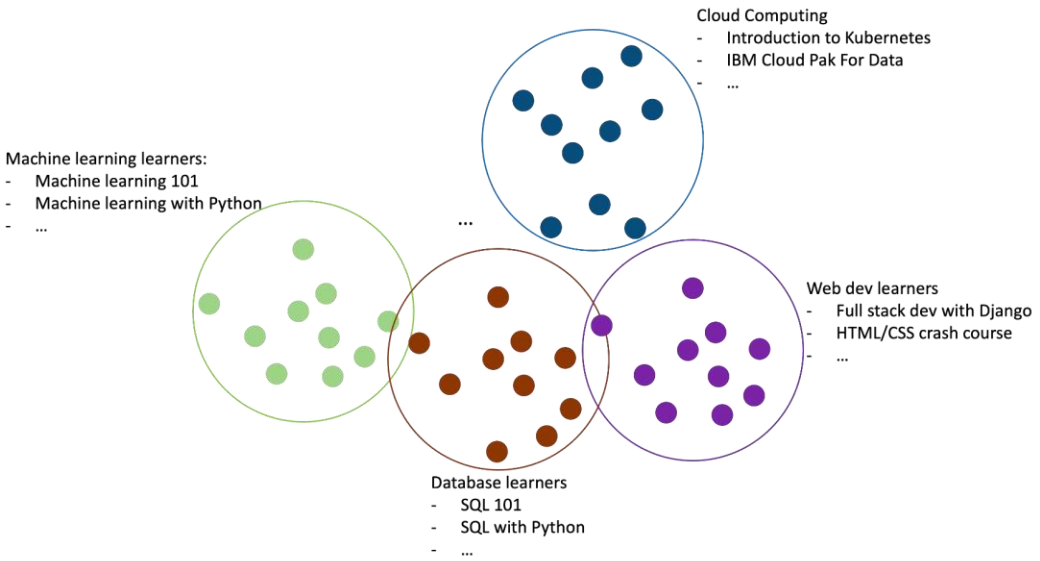


	COURSE_ID	Count
	TMP107	280
	excourse62	260
	excourse22	260
	excourse32	154
	DS0110EN	126
	DA0151EN	97
	excourse37	76
	excourse81	76
5	excourse82	76
3	excourse80	76

Flowchart of clustering-based recommender system



Clustering on User Profiles



Evaluation results of clustering-based recommender system

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

threshold num of courses = 10

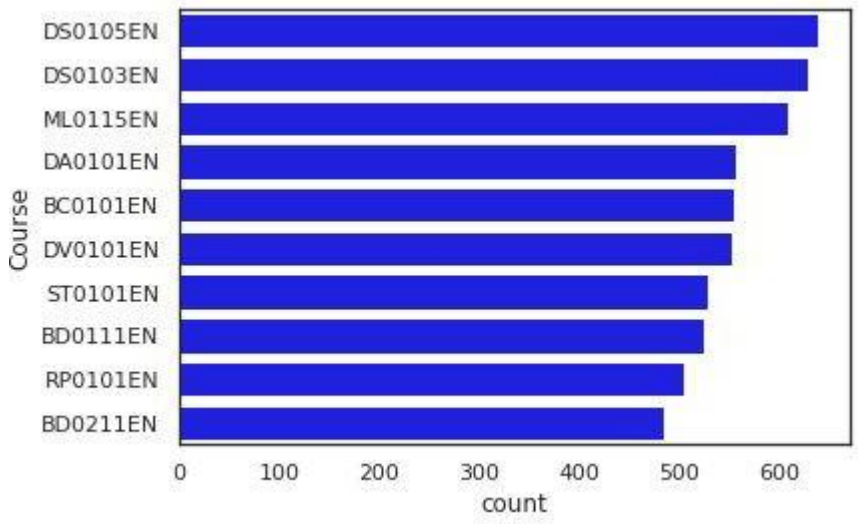
On average, how many new/unseen courses have been recommended per user (in the test user dataset)

```
1 total_recom_courses=len(df_recom['recom_courses'].sum())
2 recom_courses_per_user=total_recom_courses/len(df_recom)
3 print('recommend courses per user = ', recom_courses_per_user)
```

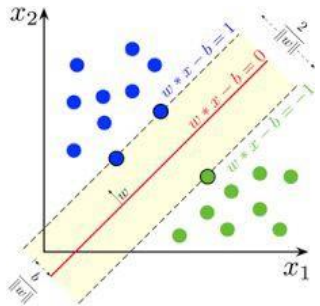
```
recommend courses per user =  13.702
```

ans = 13.7

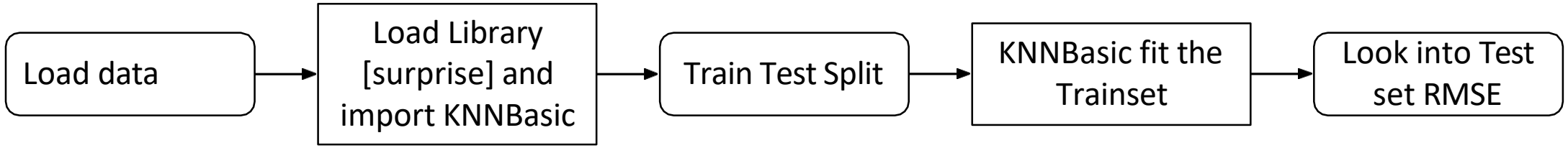
What are the most frequently recommended courses? Return the top-10 commonly recommended courses



Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system



User-based collaborative filtering:

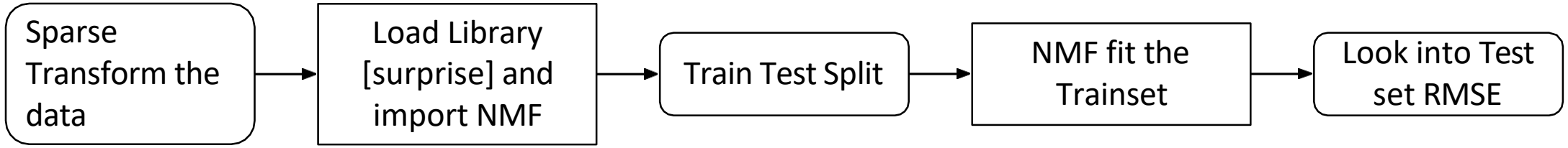
$$\hat{r} * ui = \frac{\sum * v \in N_i^k(u) \text{similarity}(u, v) \cdot r\_vi}{\sum \_v \in N_i^k(u) \text{similarity}(u, v)}$$

Item-based collaborative filtering:

$$\hat{r} * ui = \frac{\sum * j \in N_u^k(i) \text{similarity}(i, j) \cdot r\_uj}{\sum \_j \in N_u^k(i) \text{similarity}(i, j)}$$

Here $N_i^k(u)$ notates the nearest k neighbors of u.

Flowchart of NMF based recommender system



Non-negative Matrix Factorization

User-item interaction matrix: **A** 10000 x 100

	item1	...	item100
user1	...	...	
user2	3.0	3.0	3.0
user3	2.0	2.0	-
user4	3.0	2.0	3.0
user5	2.0	-	-
user6	3.0	-	3.0
...	...	...	

≈

User matrix: **U** 10000 x 16

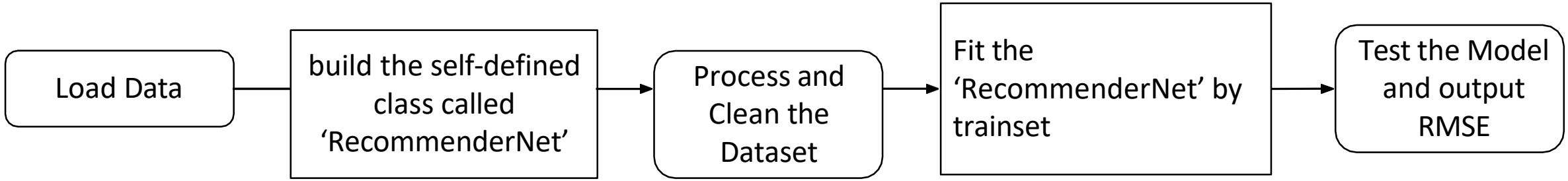
	feature1	...	feature16
user1	...	...	...
user2	...	...	...
user3	...	...	...
user4	...	...	...
...	...	...	...
...	...	...	...
user6	...	...	...

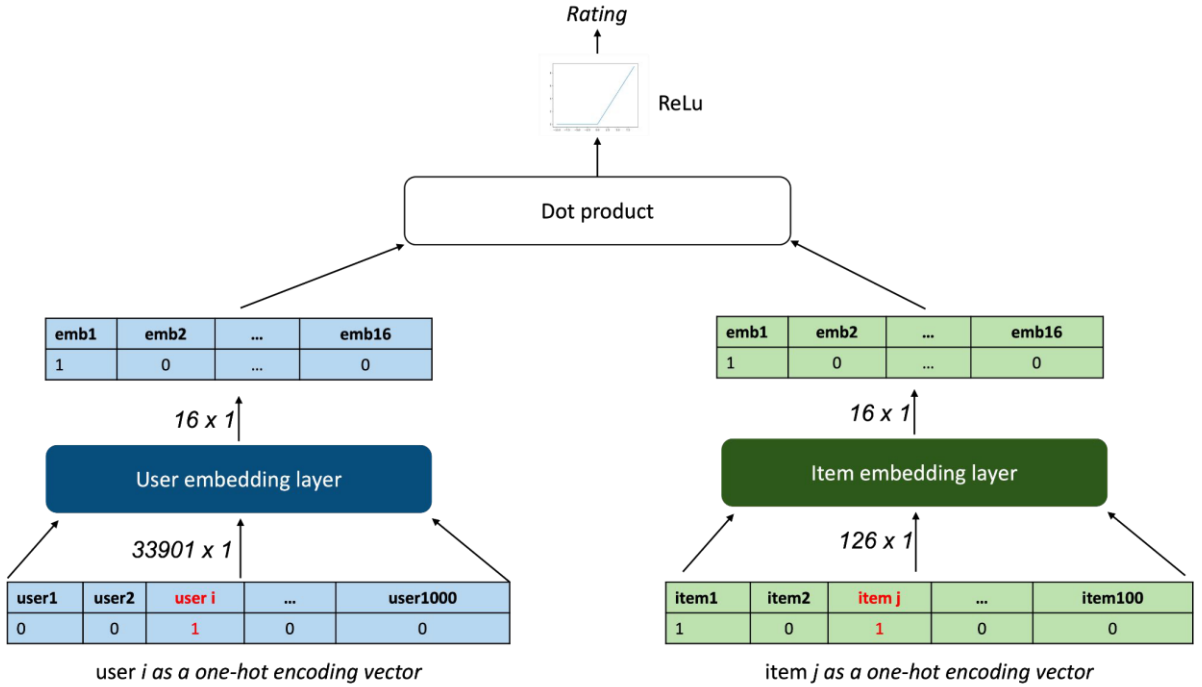
X

Item matrix: **I** 16 x 100

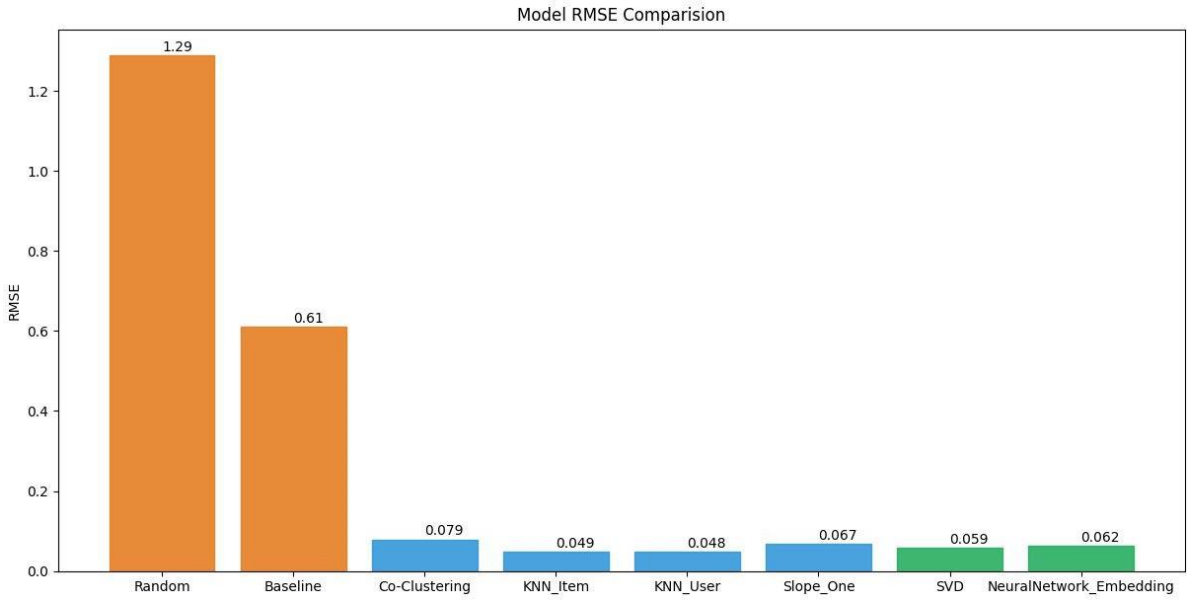
	item1	...	item100
feature1	...	...	...
feature2	...	...	...
...	...	...	...
feature16	...	...	...

Flowchart of Neural Network Embedding based recommender system





Compare the performance of collaborative-filtering models



- All Collaborative-filtering models significantly outperform the Baseline (RMSE)
- Generally KNN-based Recommend system have the best performance
  - KNN_User > KNN_Item
- SVD and NeuralNetwork have similar performance (0.059 and 0.062)
- In terms of Model Simplicity and Performance, i think KNN-based Model would be the optimal choice for the course recommender System

Conclusions

- In this Project, we explore different Modelling method for the Course recommender systems
  - Content_Based Recommender System (Unsupervised)
    - User Profile-Based
    - Course Similarity-Based
    - Clustering Based (PCA+K-means)
      - Clustering Based Modelling will generally have better performance than previous 2 options, but it is harder for model explanation because sometimes the large number of principal components will make it difficult to understand the meaning
  - Collaborative-filtering Recommender System (Supervised)
    - KNN based
    - NMF based
    - Neural Network Embedding based
      - Regression approach
      - Classification approach

- Generally, the Collaborative-filtering method is more advanced and accurate. It also provides RMSE as the metrics for Model Evaluation
  - Among all the models, the KNN-based Model have the Best Performance (RMSE=0.048)
  - Neural Network (Regression and Classification) and NMF Model have similar Accuracy (RMSE around 0.6)
  - However,NN model is computationally heavy. So the optimal final Model for the Course Recoomend System would be KNN based Model