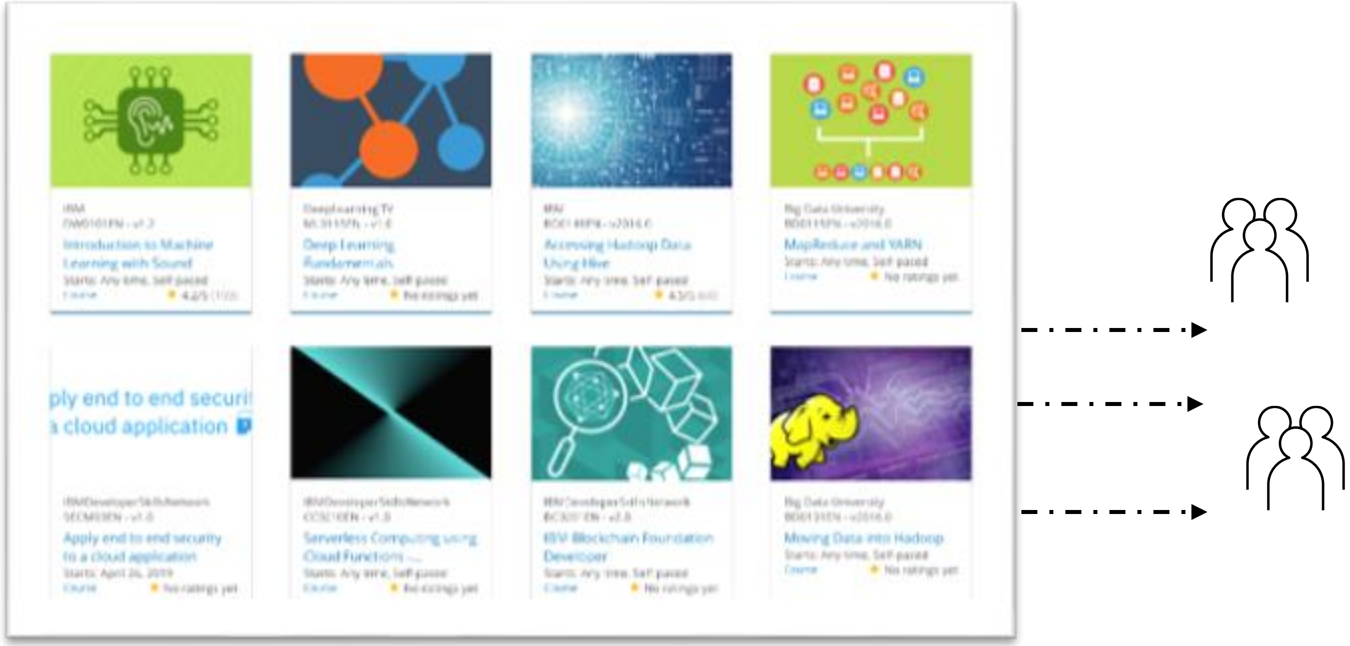
Build a Personalized Online Course Recommender System with Machine Learning

Sihan Wang
May 27th , 2025



IBM
[illegible]
Introduction to Machine Learning with Sound
Starts: Any time, Self paced
Course ★ 4.2/5 [illegible]

Deeplearning.TV
[illegible]
Deep Learning Fundamentals
Starts: Any time, Self paced
Course ★ No ratings yet

IBM
[illegible]
Accessing Hadoop Data Using Hive
Starts: Any time, Self paced
Course ★ 4.3/5 [illegible]

Big Data University
[illegible]
MapReduce and YARN
Starts: Any time, Self paced
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
SECM03EN - v1.0
Apply end to end security to a cloud application
Starts: April 26, 2019
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
CC0210EN - v1.0
Serverless Computing using Cloud Functions -...
Starts: Any time, Self paced
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
BC0201EN - v2.0
IBM Blockchain Foundation Developer
Starts: Any time, Self paced
Course ★ No ratings yet

Big Data University
BD0131EN - v2016.0
Moving Data into Hadoop
Starts: Any time, Self paced
Course ★ No ratings yet

Outline

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

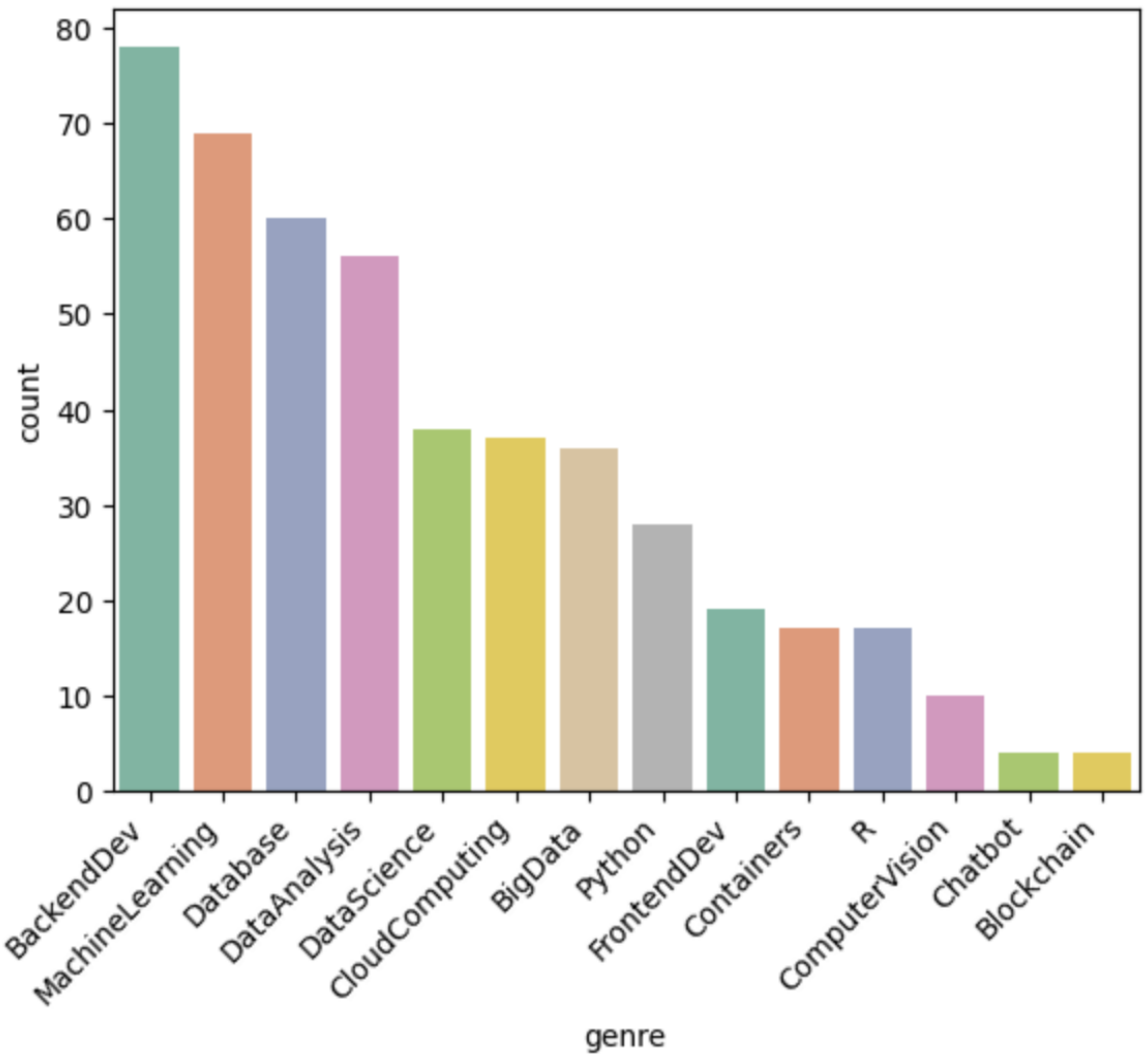
Introduction

- Project background and context
- E-learning platform in recent years has provide great educational opportunity to users. However, the abundance of available courses often overwhelm users for course selection. To increase user engagement and promote courses, course recommendation system is inevitable.
- Problem states and hypotheses
- 1. Which recommendation techniques yield the most accurate predictions for future enrollments?
- 2. Which factors/feature is significant for recommendation system.
- The hypotheses including users’ course rating, course popularity, course genre etc.

Exploratory Data Analysis

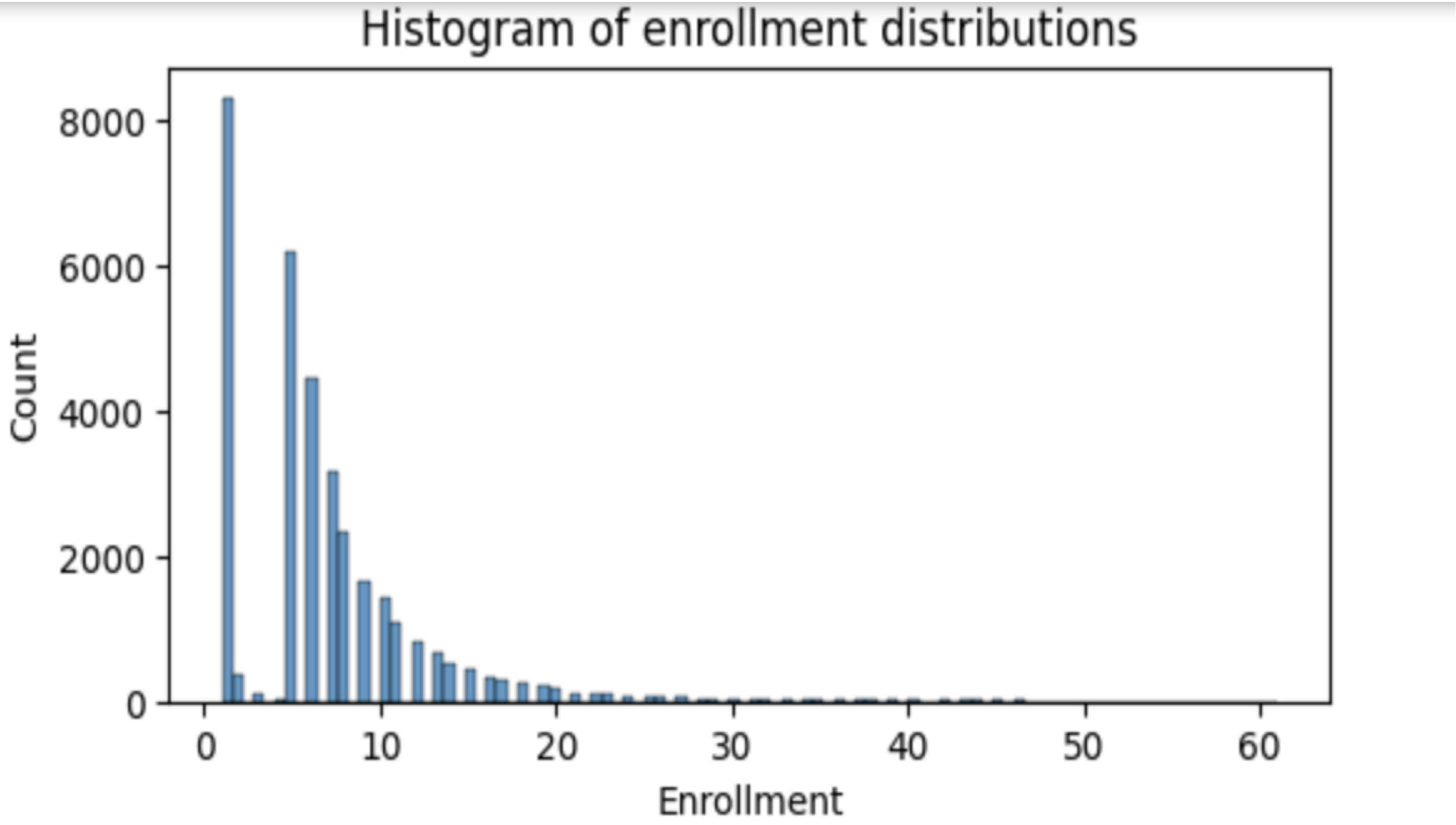
Course counts per genre

- Backend development, machine learning, databases, data analysis and data science rank the top 5 in count.
- Blockchain is the with the least count.



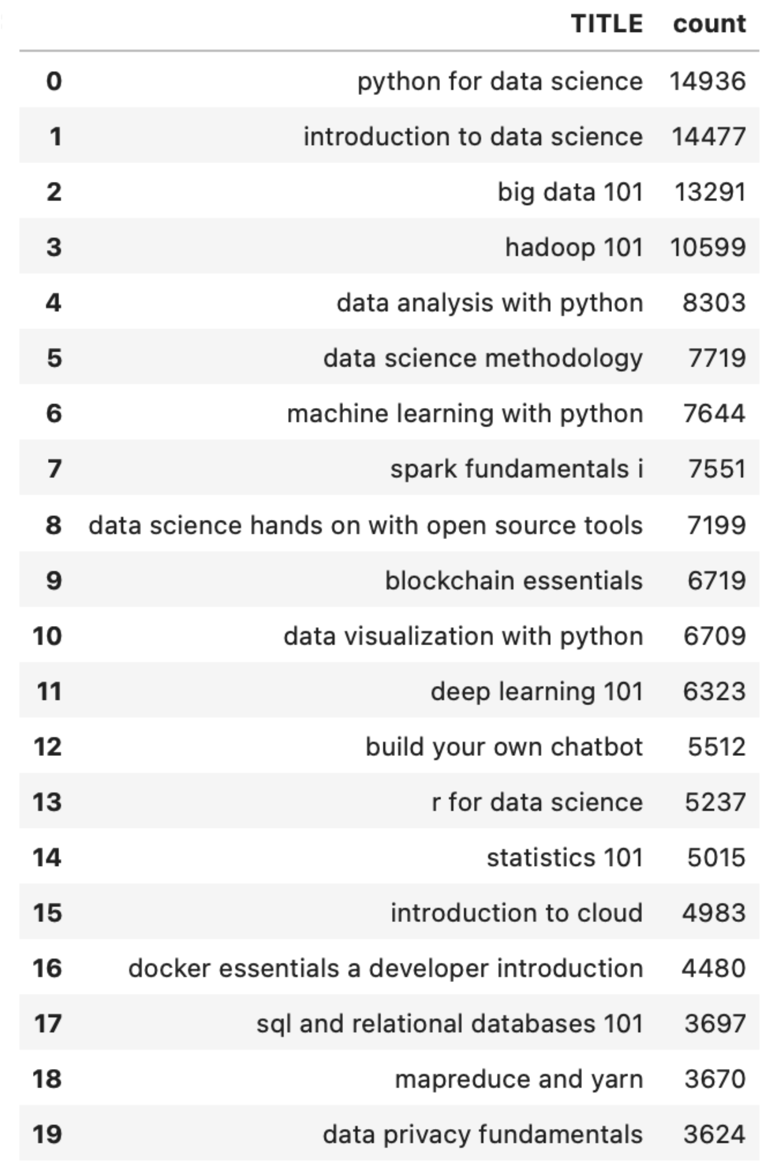
Course enrollment distribution

- The distribution of enrollment is right skew. The majority enrollment is less than 20.
- No users enroll above 60 courses.



20 most popular courses

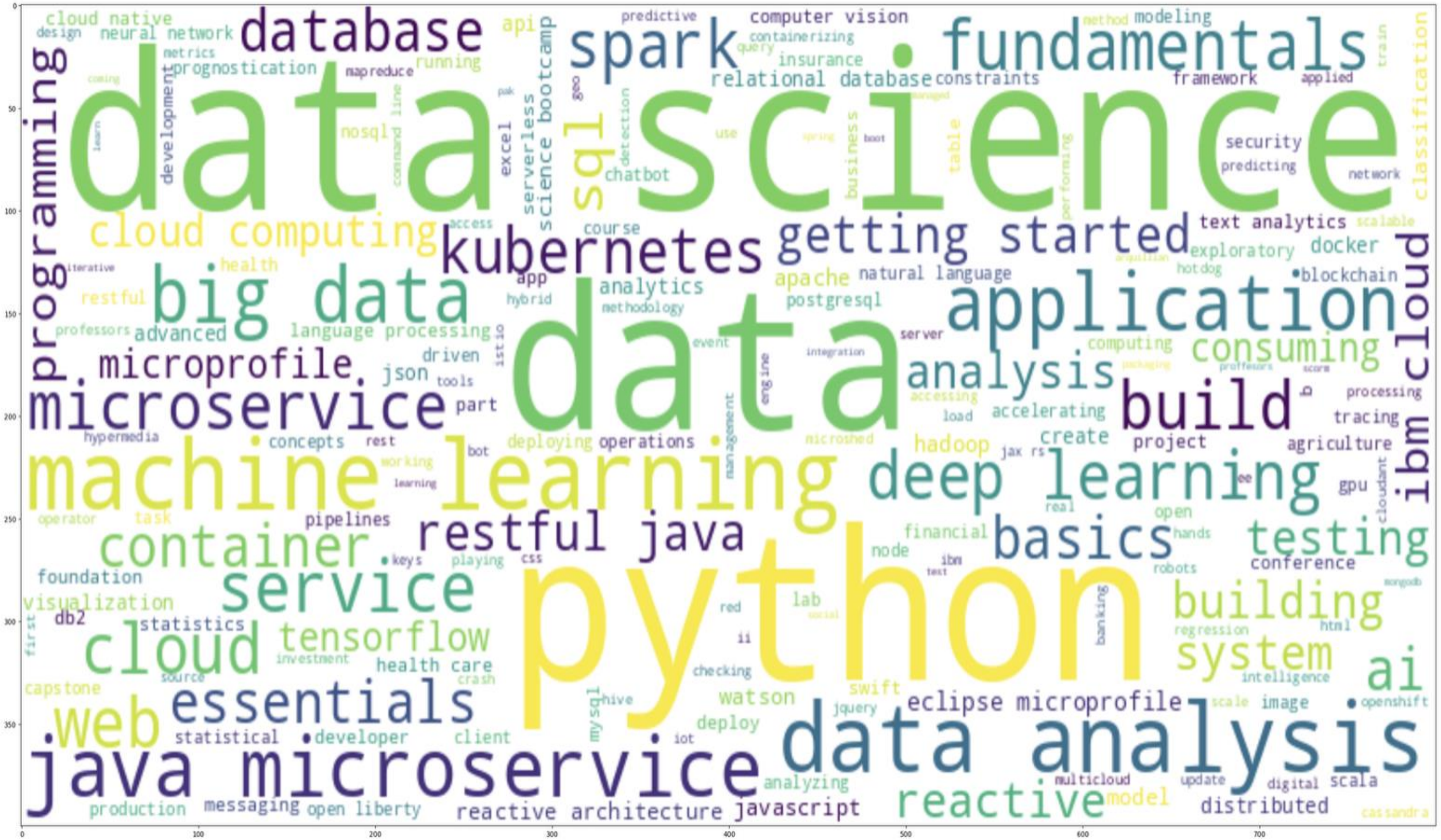
- Python for data science is the most popular course with enrollment 14936. The second and third popular course are introduction to data science and big data 101 with enrollment also exceed 10,000.
- Data science courses dominate the top 20 list.



	TITLE	count
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

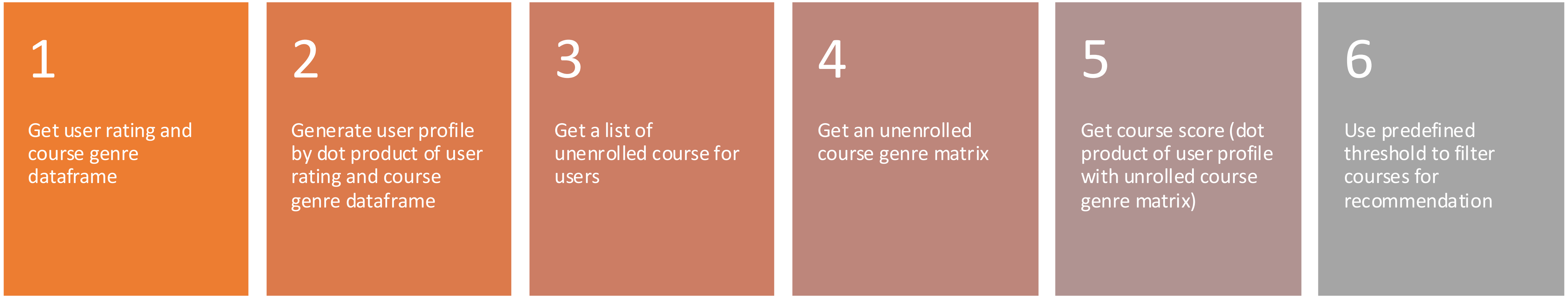
Word cloud of course titles

- Based on word cloud, words like ‘data science’, ‘data’, ‘python’, ‘big data’, ‘machine learning’ are dominant topics.



Content-based Recommender System using Unsupervised Learning

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



1

Get user rating and course genre dataframe

2

Generate user profile by dot product of user rating and course genre dataframe

3

Get a list of unenrolled course for users

4

Get an unenrolled course genre matrix

5

Get course score (dot product of user profile with unrolled course genre matrix)

6

Use predefined threshold to filter courses for recommendation

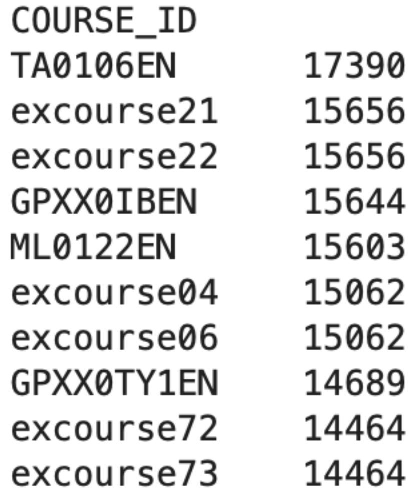
Evaluation results of user profile-based recommender system

Recommendation score = 10

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

Round down to 68 courses.

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
excourse72     14464
excourse73     14464
```

Flowchart of content-based recommender system using course similarity

1

Calculate the similarity between two courses using Bag of Words (BoW) features.

2

Get user enrolled courses and unenrolled courses

3

For each enrolled course, find similarity score with each unenrolled course, set threshold for similarity.

4

Recommend courses with above similarity threshold.

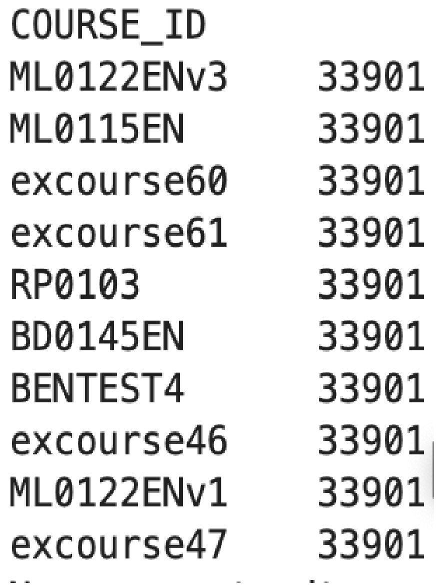
Evaluation results of course similarity based recommender system

similarity threshold = 0.4

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

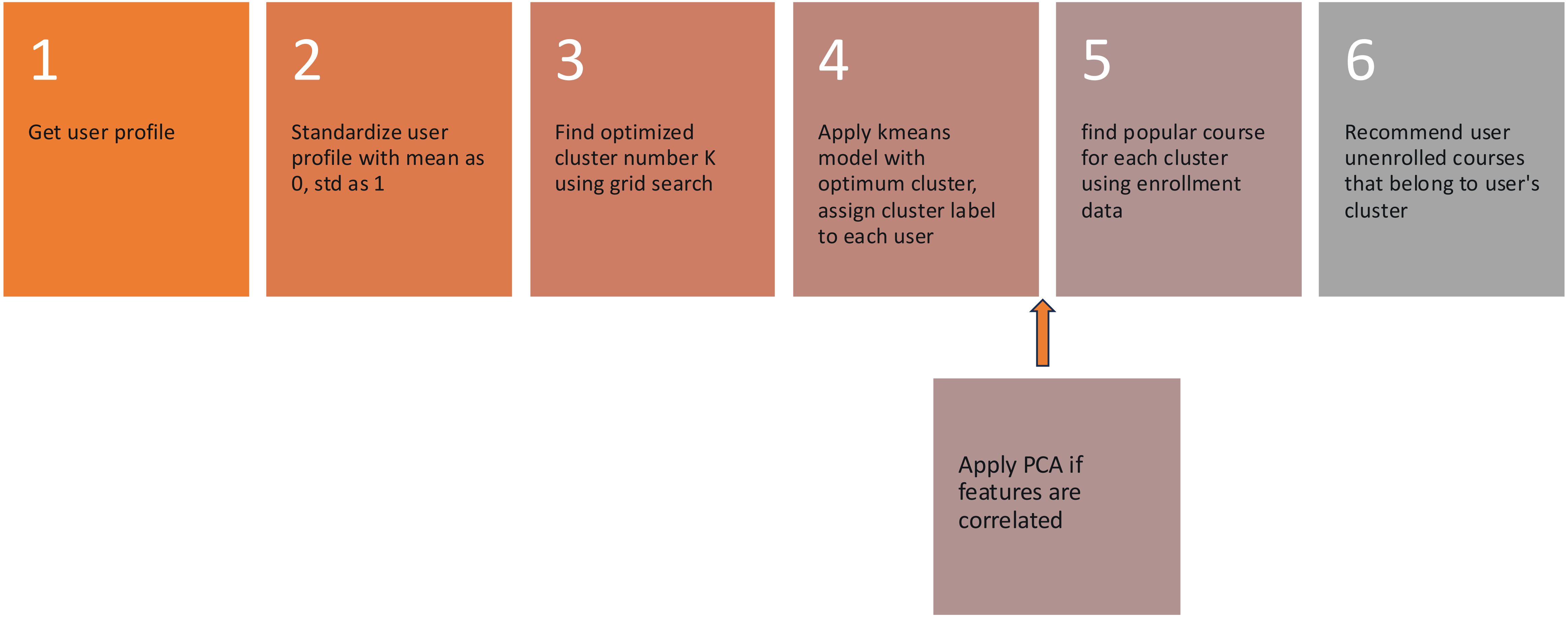
On average, 1 course has been recommended per user.

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



```
COURSE_ID
ML0122ENv3     33901
ML0115EN       33901
excourse60     33901
excourse61     33901
RP0103         33901
BD0145EN       33901
BENTEST4       33901
excourse46     33901
ML0122ENv1     33901
excourse47     33901
```

Flowchart of clustering-based recommender system



1

Get user profile

2

Standardize user profile with mean as 0, std as 1

3

Find optimized cluster number K using grid search

4

Apply kmeans model with optimum cluster, assign cluster label to each user

5

find popular course for each cluster using enrollment data

6

Recommend user unenrolled courses that belong to user's cluster

Apply PCA if features are correlated

Evaluation results of clustering-based recommender system

Optimum cluster = 20

PCA component number = 14

Enrollment threshold = 100

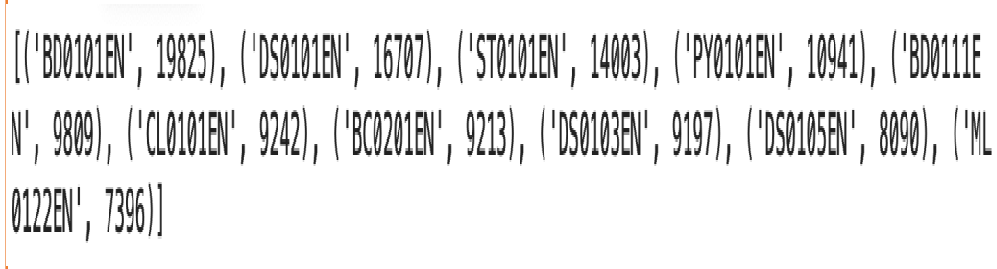
On average, how many new/unseen courses have been recommended per user (in the test user dataset)

Round down to 5 courses.

(5.8 as output)

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

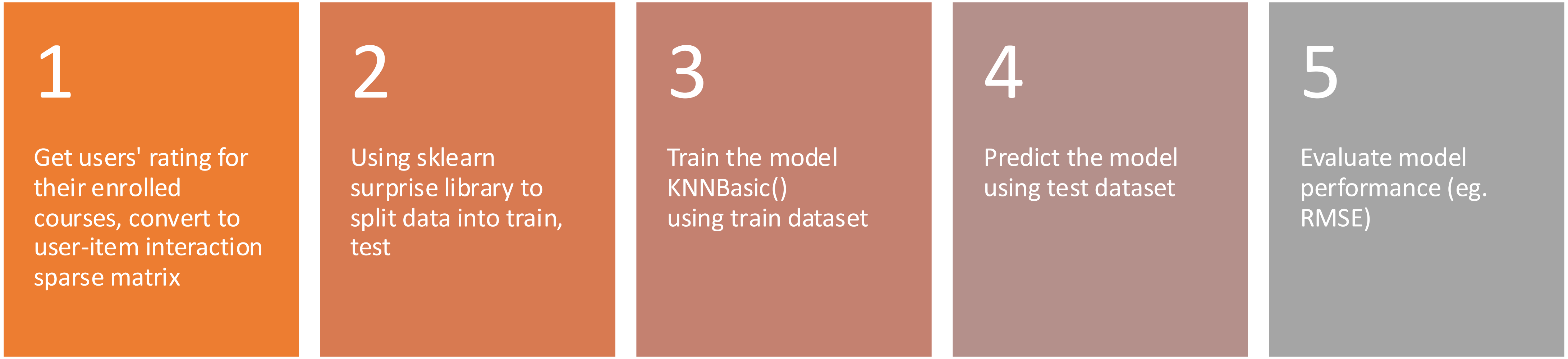
Below shows top ten recommended course and total times of recommendation



```
[('BD0101EN', 19825), ('DS0101EN', 16707), ('ST0101EN', 14003), ('PY0101EN', 10941), ('BD0111E
N', 9809), ('CL0101EN', 9242), ('BC0201EN', 9213), ('DS0103EN', 9197), ('DS0105EN', 8090), ('ML
0122EN', 7396)]
```

Collaborative-filtering Recommender System using Supervised Learning

Flowchart of KNN based recommender system



1

Get users' rating for their enrolled courses, convert to user-item interaction sparse matrix

2

Using sklearn surprise library to split data into train, test

3

Train the model KNNBasic() using train dataset

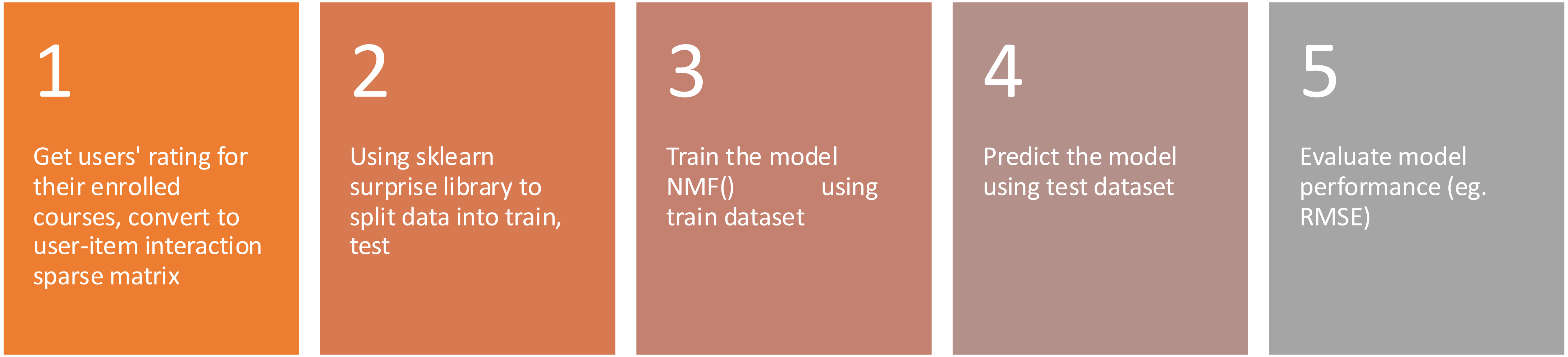
4

Predict the model using test dataset

5

Evaluate model performance (eg. RMSE)

Flowchart of NMF based recommender system



1

Get users' rating for their enrolled courses, convert to user-item interaction sparse matrix

2

Using sklearn surprise library to split data into train, test

3

Train the model NMF() using train dataset

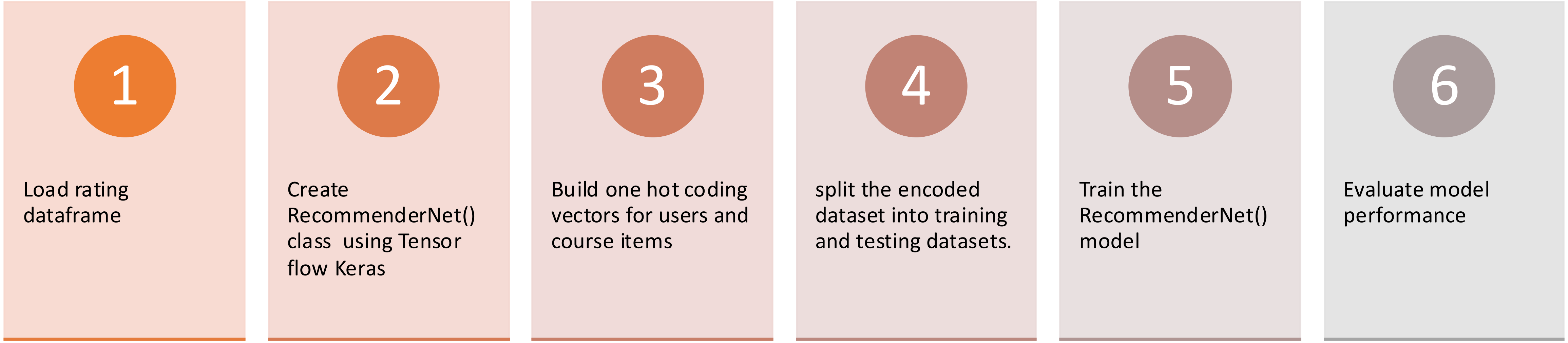
4

Predict the model using test dataset

5

Evaluate model performance (eg. RMSE)

Flowchart of Neural Network Embedding based recommender system



1 Load rating dataframe

2 Create RecommenderNet() class using Tensor flow Keras

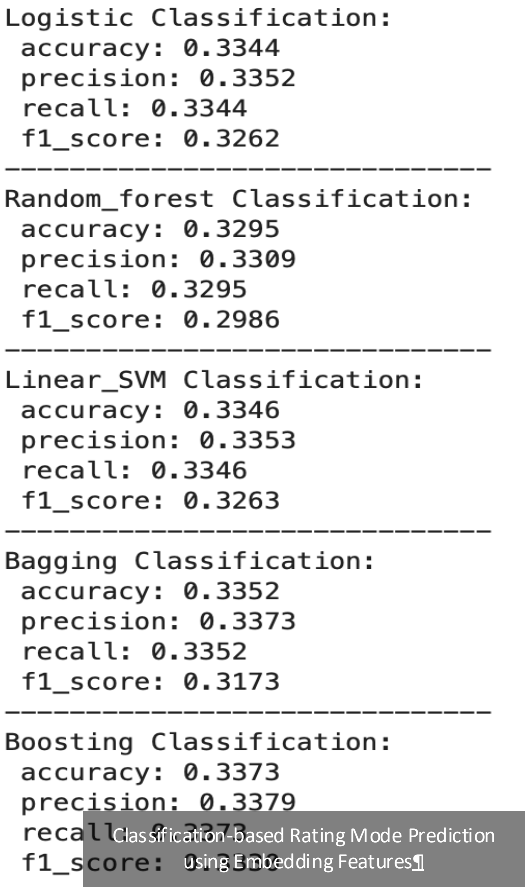
3 Build one hot coding vectors for users and course items

4 split the encoded dataset into training and testing datasets.

5 Train the RecommenderNet() model

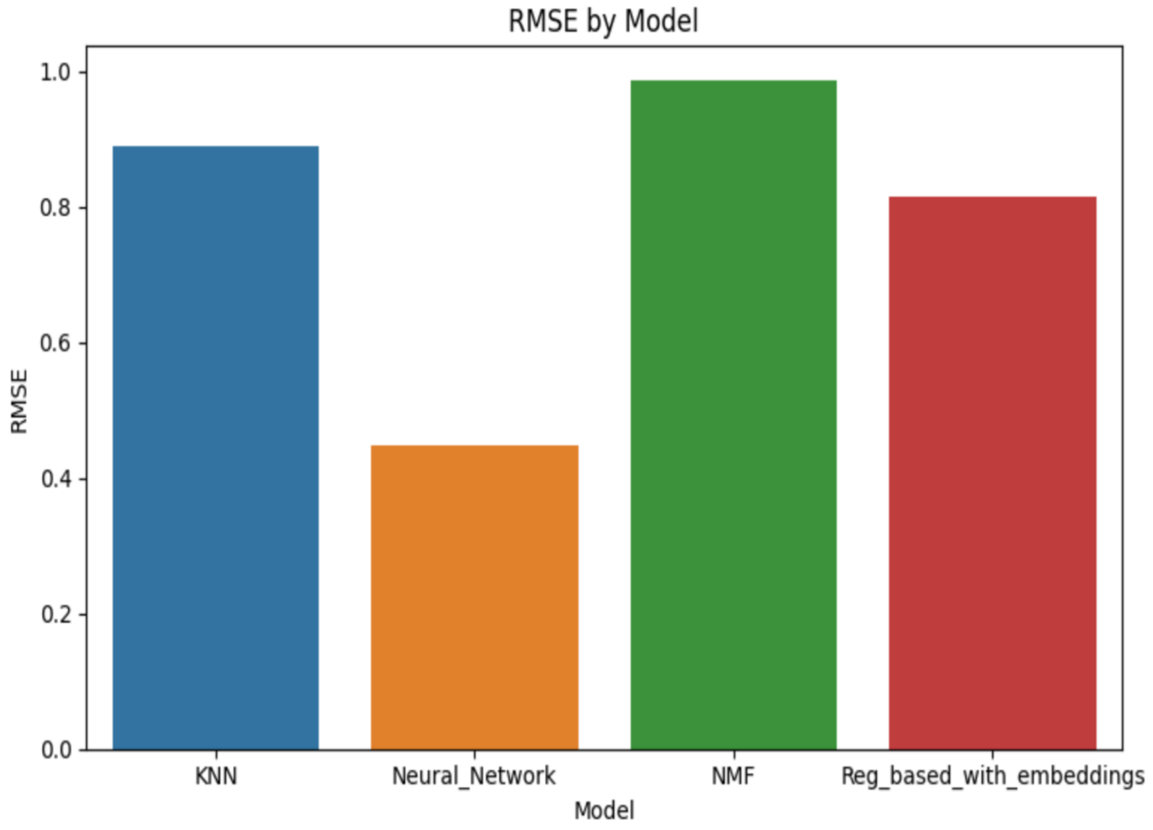
6 Evaluate model performance

Compare the performance of collaborative-filtering models



```
Logistic Classification:
 accuracy: 0.3344
 precision: 0.3352
 recall: 0.3344
 f1_score: 0.3262
------------------------------
Random_forest Classification:
 accuracy: 0.3295
 precision: 0.3309
 recall: 0.3295
 f1_score: 0.2986
------------------------------
Linear_SVM Classification:
 accuracy: 0.3346
 precision: 0.3353
 recall: 0.3346
 f1_score: 0.3263
------------------------------
Bagging Classification:
 accuracy: 0.3352
 precision: 0.3373
 recall: 0.3352
 f1_score: 0.3173
------------------------------
Boosting Classification:
 accuracy: 0.3373
 precision: 0.3379
 recall: 0.3373
 f1_score: [illegible]
```

Classification-based Rating Mode Prediction using Embedding Features¶



Conclusions

- In this project, several approaches to enhance online course enrollment recommendations were explored. Multiple algorithms including:
- **Content-Based Filtering:** recommends courses based on similarities in course content and unsupervised Kmeans model to group similar users.
- **Collaborative Filtering:** use user-course interaction data to identify popular courses and suggest courses favored by similar users.
- **Course rating prediction:** neural network, regression-based and classification-based using embedding features.
- Evaluation metrics such as RMSE, precision, recall, F1-score, were used to assess the performance of each model.
- Neural network didn’t outperform KNN, NMF may due to simpler models fits data better and avoid overfitting, or simpler model handle cold start better.
- All classification-based predictions has low accuracy, with bagging and boosting are slightly higher than others.