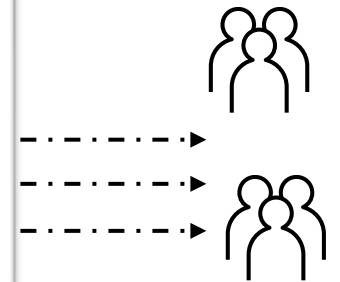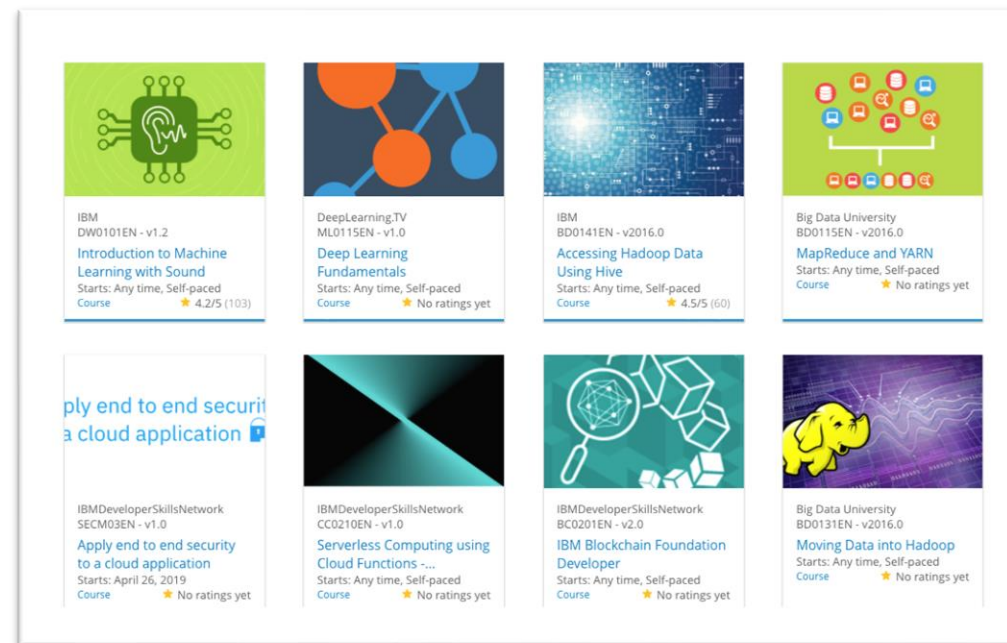# Build a Personalized Online Course Recommender System with Machine Learning

Mehnaz Khan
October 4, 2022

Outline

- Introduction and Background

- Exploratory Data Analysis

- Content-based Recommender System using Unsupervised Learning

- Collaborative-filtering Recommender System using Supervised Learning

- Conclusion

- Appendix

Introduction

Project Background: Building a recommender model for courses by utilizing and evaluating different recommender systems

The following techniques are implemented:

- Content-based Recommender System using Unsupervised Learning

  - Content-based recommender system using user profile and course genres

  - Content-based recommender system using course similarity

  - Content-based recommender system using user profile clustering

- Collaborative-filtering Recommender System using Supervised Learning

  - KNN-based collaborative filtering

  - NMF-based collaborative filtering

  - Neural network embedding based collaborative filtering

  - Collaborative filtering algorithms evaluation

Exploratory Data Analysis
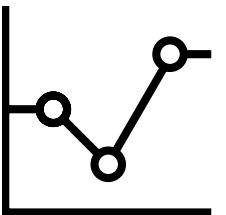
Course counts per genre
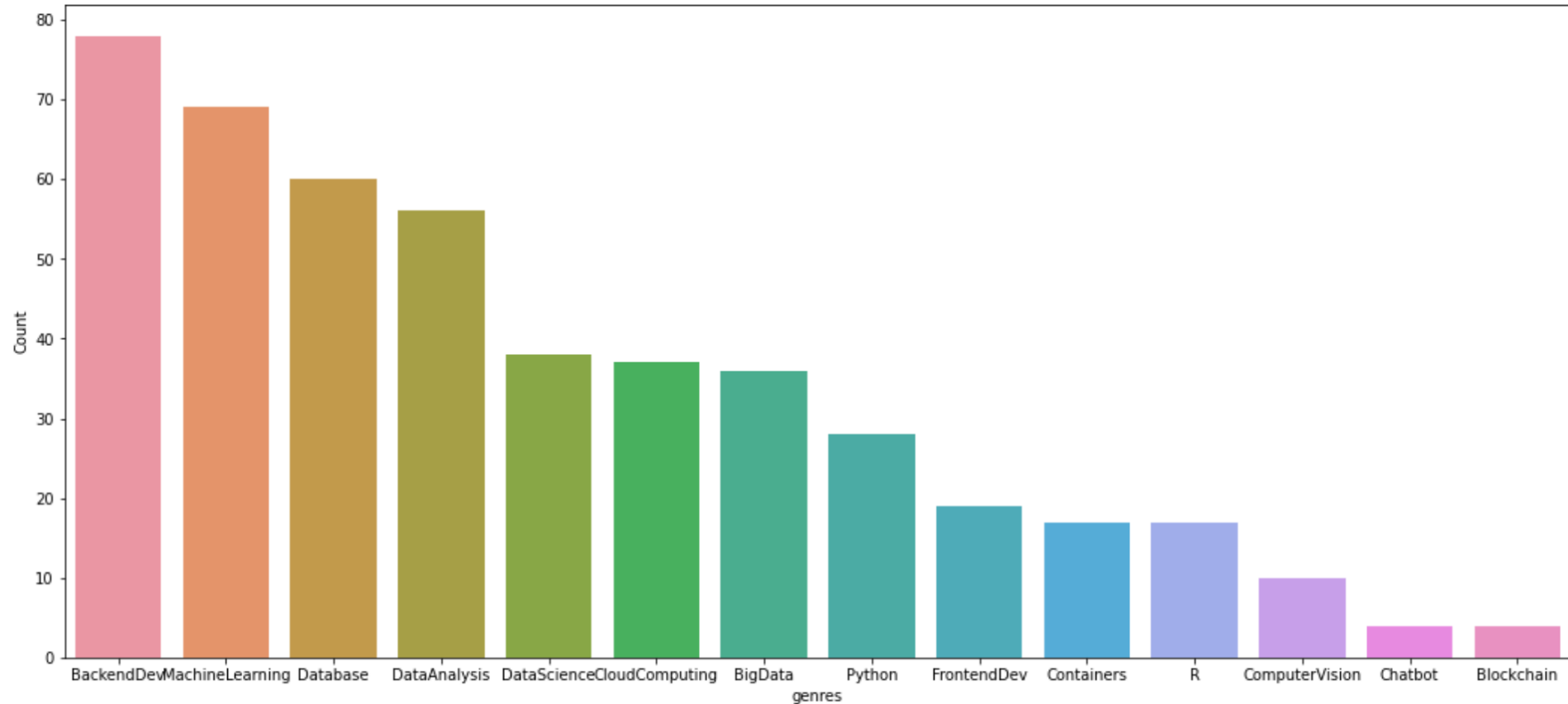
- It is important to get the idea on how many courses in each genre to choose the right courses.
- The bar chart shows the total counts in each genre.

Course enrollment distribution

- Course enrollment distribution provides the idea on how many courses (such as just 1 item or 10, 20 etc.) have been rated by the users.

- The following histogram shows the enrollment distribution of the courses.
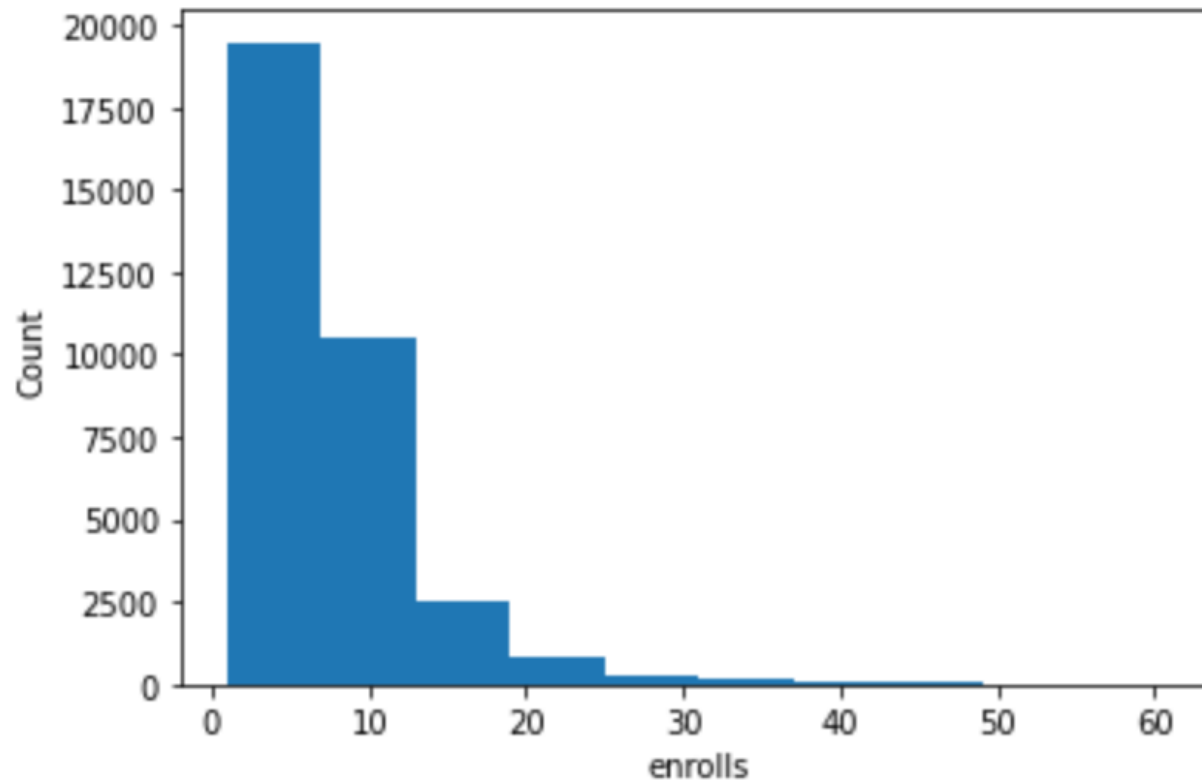
20 most popular courses

- Need to find total enrollment in each course from the user rating dataset by grouping by item and sorting them
- Need to find the title of each course corresponding course ID from the course dataset

	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

Word cloud of course titles

Word cloud provides the more intuitive summary of what kind of courses we have in the dataset.
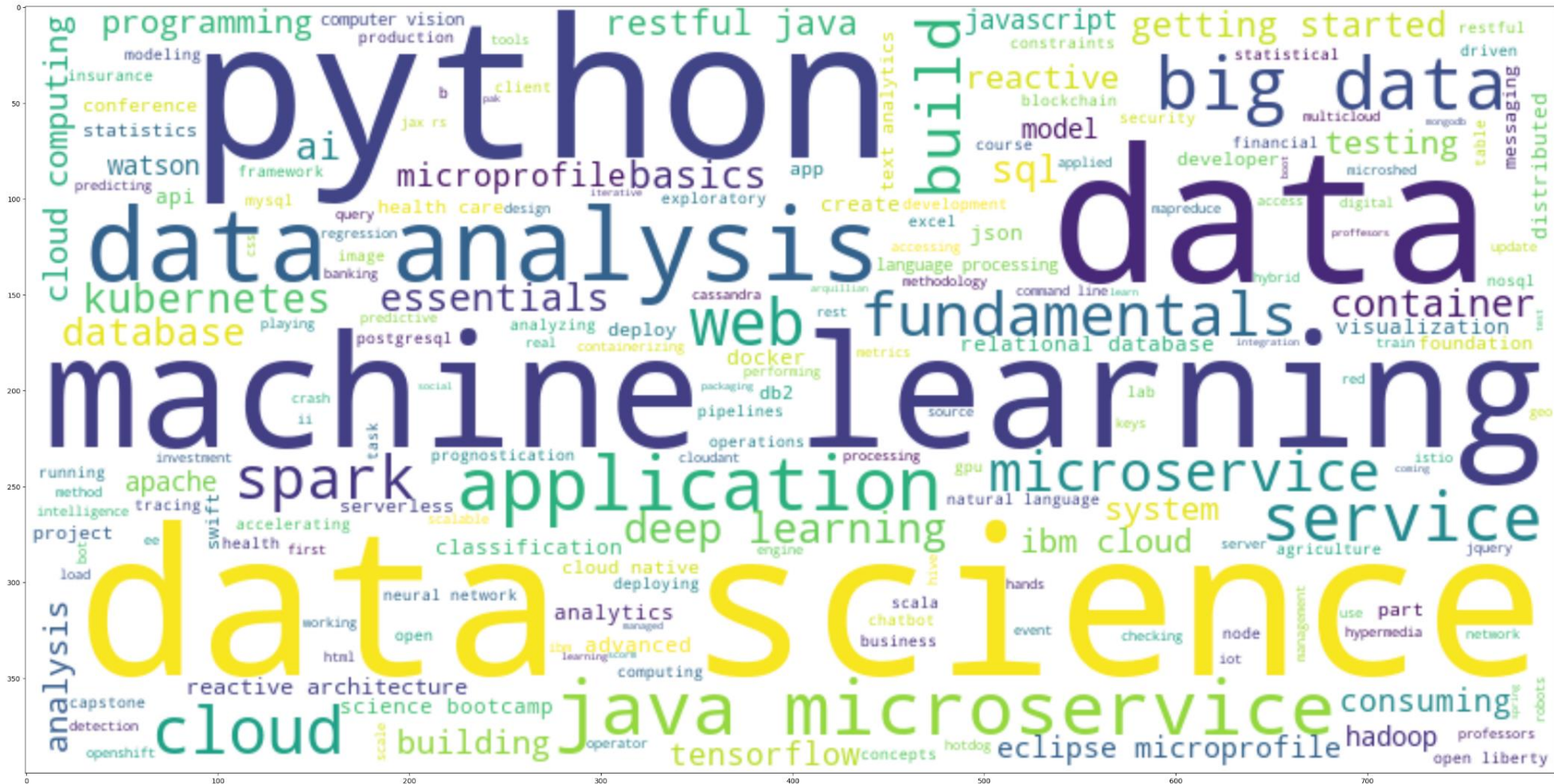
Content-based Recommender System using Unsupervised Learning
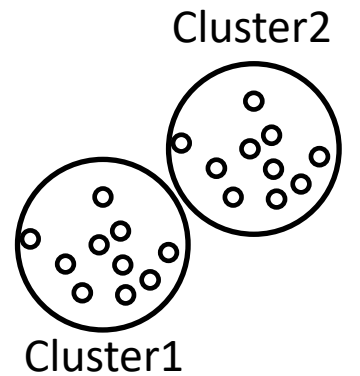
Cluster2

Cluster1

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

```
┌──────────┐     ┌──────────────┐     ┌──────────┐     ┌──────────────┐     ┌──────────┐
│ Raw data │ ──> │Data processing│ ──> │ Cleaned  │ ──> │   Feature    │ ──> │ Features │
│          │     │              │     │ datasets │     │ engineering  │     │          │
└──────────┘     └──────────────┘     └──────────┘     └──────────────┘     └──────────┘
                                                                                  │
                                                                                  v
┌──────────────┐     ┌──────────────┐     ┌──────────────────┐     ┌──────────────┐
│ Finding most │     │   Finding    │     │ Creating test user│     │Choosing Score│
│ recommended  │ <── │recommendation│ <── │vector and unknown │ <── │  Threshold   │
│   courses    │     │    score     │     │  course matrix    │     │              │
└──────────────┘     └──────────────┘     └──────────────────┘     └──────────────┘
```
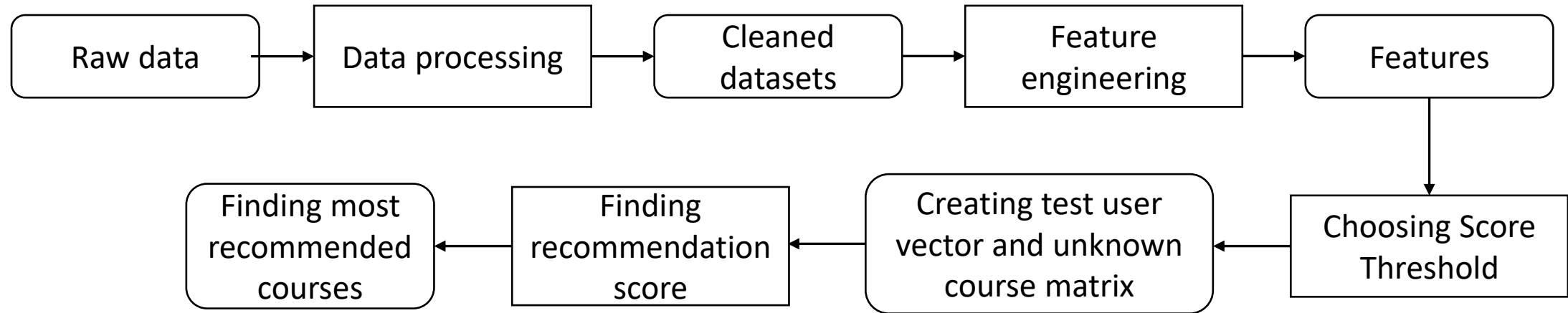
The flowchart illustrates the process of the implementation of the content-based recommender system using user profile vectors and course genre vectors where we need to choose appropriate score threshold and find out the recommendation score through course matrix and user vector to choose most recommended courses.

Evaluation results of user profile-based recommender system

Score Threshold is chosen as 10, that means it will filter out all score lower than 10.

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

```
recom_total.describe()

count    864.000000
mean      61.818287
std       46.592144
min        1.000000
25%       30.000000
50%       51.500000
75%       81.250000
max      267.000000
dtype: float64
```

The top-10 commonly recommended courses across all users

	COURSE_ID	TITLE
0	RP0105EN	analyzing big data in r using apache spark
1	excourse73	analyzing big data with sql
2	excourse72	foundations for big data analysis with sql
3	TMP0105EN	getting started with the data apache spark ma...
4	SC0103EN	spark overview for scala analytics
5	RP0105EN	analyzing big data in r using apache spark
6	excourse73	analyzing big data with sql
7	excourse72	foundations for big data analysis with sql
8	TMP0105EN	getting started with the data apache spark ma...
9	TMP0105EN	getting started with the data apache spark ma...

Flowchart of content-based recommender system using course similarity

The following flowchart illustrates how we can implement the course similarity based recommender system through similarity matrix considering similarity score threshold.
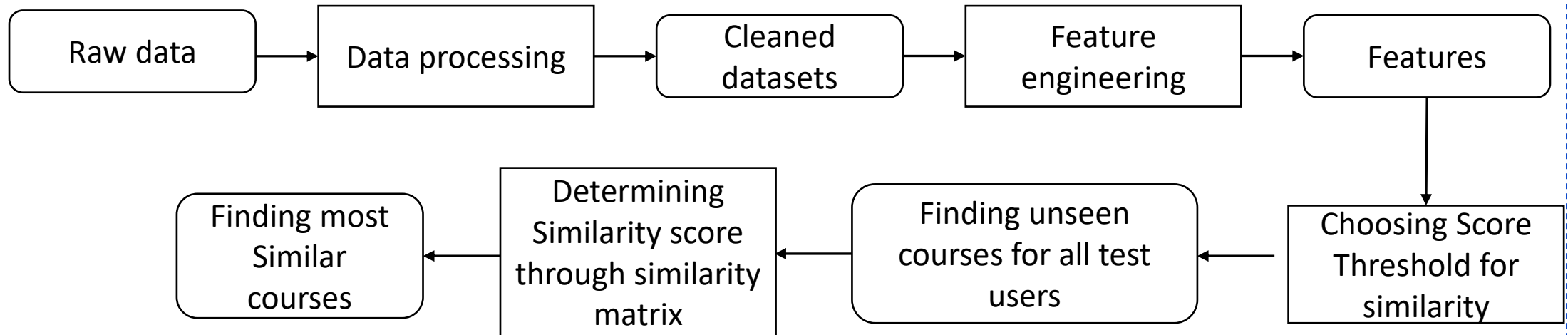
Evaluation results of course similarity based recommender system

Similarity threshold is chosen as 0.6 to filter out all lower similar courses.

On average, 7 new/unseen courses have been recommended per user (in the test user dataset)

The top-10 commonly recommended courses

	COURSE_ID	TITLE
0	ML0120ENv3	deep learning with tensorflow
1	ML0120ENv2	deep learning with tensorflow
2	ML0120EN	deep learning with tensorflow
3	ML0122ENv1	accelerating deep learning with gpu
4	ML0120ENv3	deep learning with tensorflow
5	ML0120ENv2	deep learning with tensorflow
6	ML0120EN	deep learning with tensorflow
7	CB0103EN	build your own chatbot
8	CB0101EN	build your own chatbots
9	TMP0101EN	text analysis

Flowchart of clustering-based recommender system

The flowchart illustrates how you performed user profile clustering based recommender system

```
┌──────────┐     ┌──────────────┐     ┌──────────┐     ┌──────────────┐     ┌──────────┐
│ Raw data │ ──> │ Data         │ ──> │ Cleaned  │ ──> │ Feature      │ ──> │ Features │
│          │     │ processing   │     │ datasets │     │ engineering  │     │          │
└──────────┘     └──────────────┘     └──────────┘     └──────────────┘     └──────────┘
                                                                                  │
                                                                                  v
┌──────────────────┐   ┌──────────────┐   ┌──────────────────┐   ┌──────────────────┐
│ Finding most     │   │ Determining  │   │ Finding the      │   │ Determining the  │
│ enrolled courses │<──│ unseen       │<──│ courses from the │<──│ cluster from     │
│ from the same    │   │ courses for  │   │ same cluster of  │   │ Kmeans algorithm │
│ cluster for      │   │ the all users│   │ the user         │   │                  │
│ recommendations  │   │              │   │                  │   │                  │
└──────────────────┘   └──────────────┘   └──────────────────┘   └──────────────────┘
```

Evaluation results of clustering-based recommender system
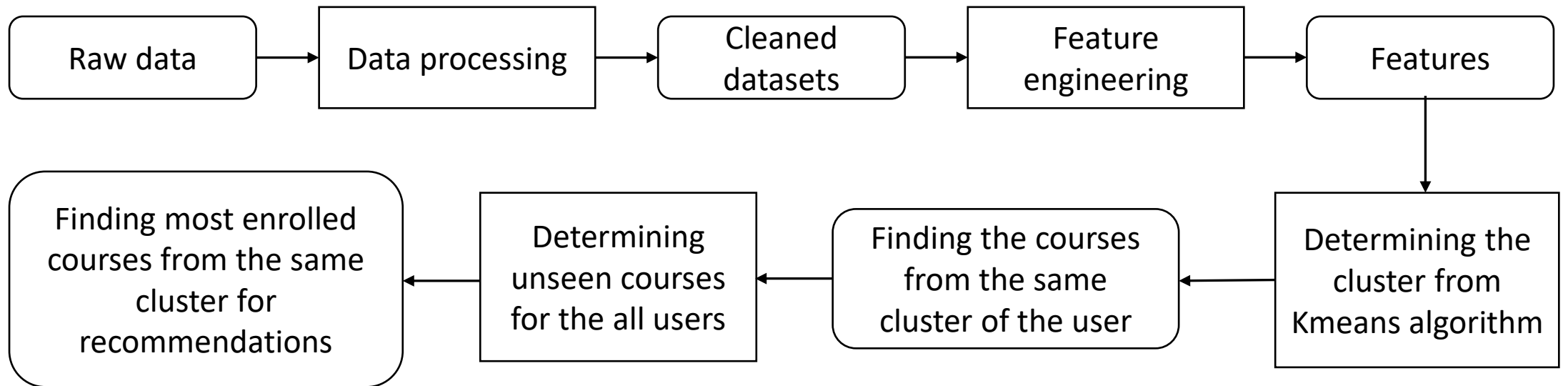
On average, 59 new/unseen courses have been recommended per user (in the test user dataset)
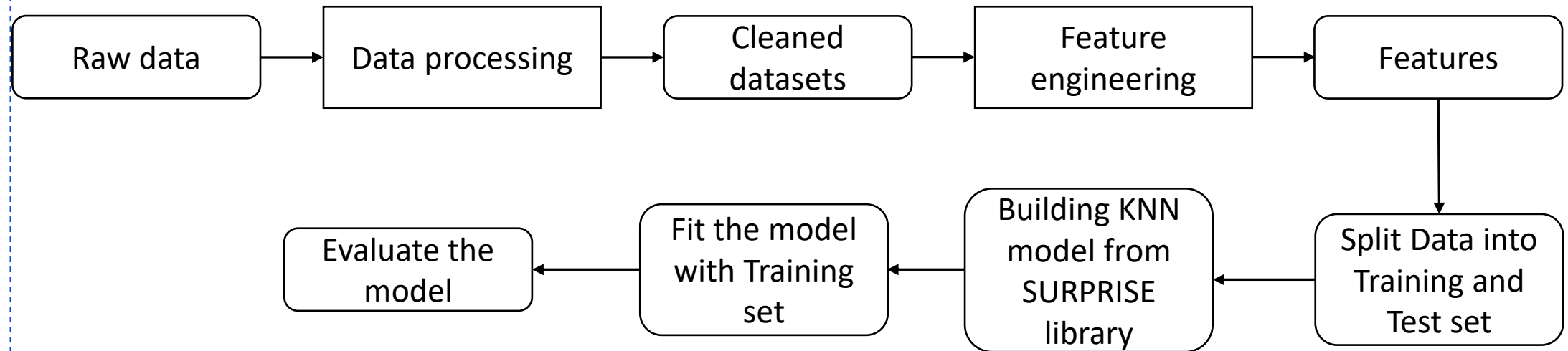
The top-10 commonly recommended courses

COURSE_ID	TITLE
DS0201EN	end to end data science on cloudpak for data
ML0109EN	machine learning dimensionality reduction
BD0151EN	text analytics 101
BD0223EN	exploring spark s graphx
LB0103ENv1	reactive architecture domain driven design
WA0103EN	watson analytics for social media
LB0101ENv1	reactive architecture introduction to reactiv...
CC0210EN	serverless computing using cloud functions d...
DV0151EN	data visualization with r
CO0401EN	beyond the basics istio and ibm cloud kuberne...

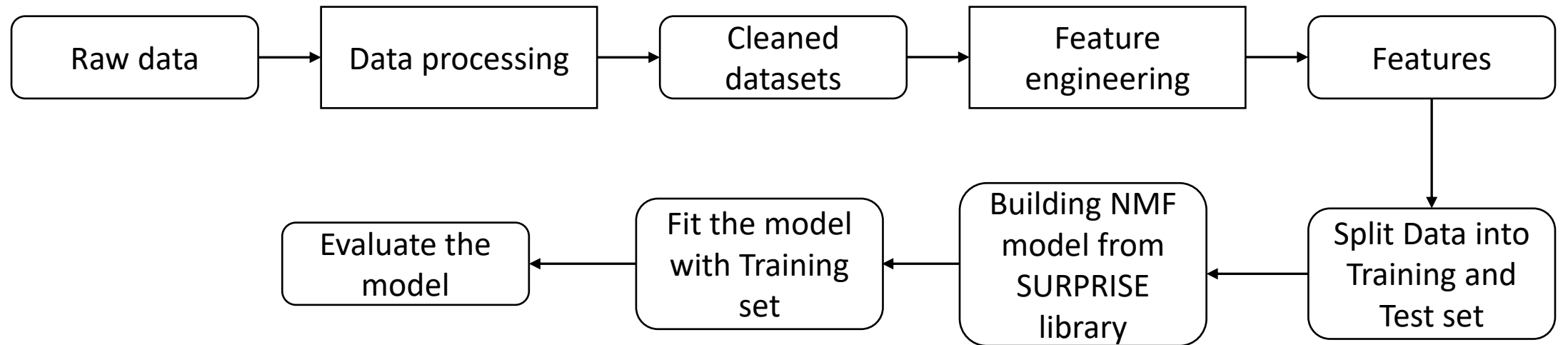# Collaborative-filtering Recommender System using Supervised Learning
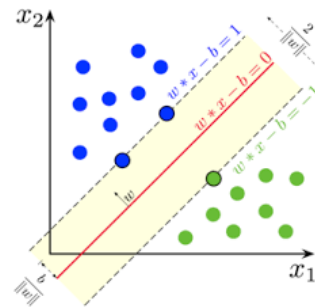
Flowchart of KNN based recommender system

The flowchart illustrates how we performed KNN based recommender system using course enrollments history through building KNN based model and evaluating model through Root Means Square Error (RMSE).

Raw data → Data processing → Cleaned datasets → Feature engineering → Features

Split Data into Training and Test set ← Features

Building KNN model from SURPRISE library ← Split Data into Training and Test set

Fit the model with Training set ← Building KNN model from SURPRISE library

Evaluate the model ← Fit the model with Training set

Flowchart of NMF based recommender system

The flowchart illustrates how we performed NMF based recommender system through building NMF model and evaluating model through Root Means Square Error (RMSE).

```
┌──────────┐   ┌──────────────┐   ┌──────────┐   ┌──────────────┐   ┌──────────┐
│ Raw data │──▶│Data processing│──▶│ Cleaned  │──▶│   Feature    │──▶│ Features │
│          │   │              │   │ datasets │   │ engineering  │   │          │
└──────────┘   └──────────────┘   └──────────┘   └──────────────┘   └────┬─────┘
                                                                          │
                                                                          ▼
┌──────────┐   ┌──────────────┐   ┌──────────────┐              ┌──────────────┐
│Evaluate  │◀──│ Fit the model│◀──│ Building NMF │◀─────────────│Split Data into│
│the model │   │ with Training│   │  model from  │              │ Training and │
│          │   │     set      │   │   SURPRISE   │              │   Test set   │
└──────────┘   └──────────────┘   │   library    │              └──────────────┘
                                  └──────────────┘
```

Flowchart of Neural Network Embedding based recommender system
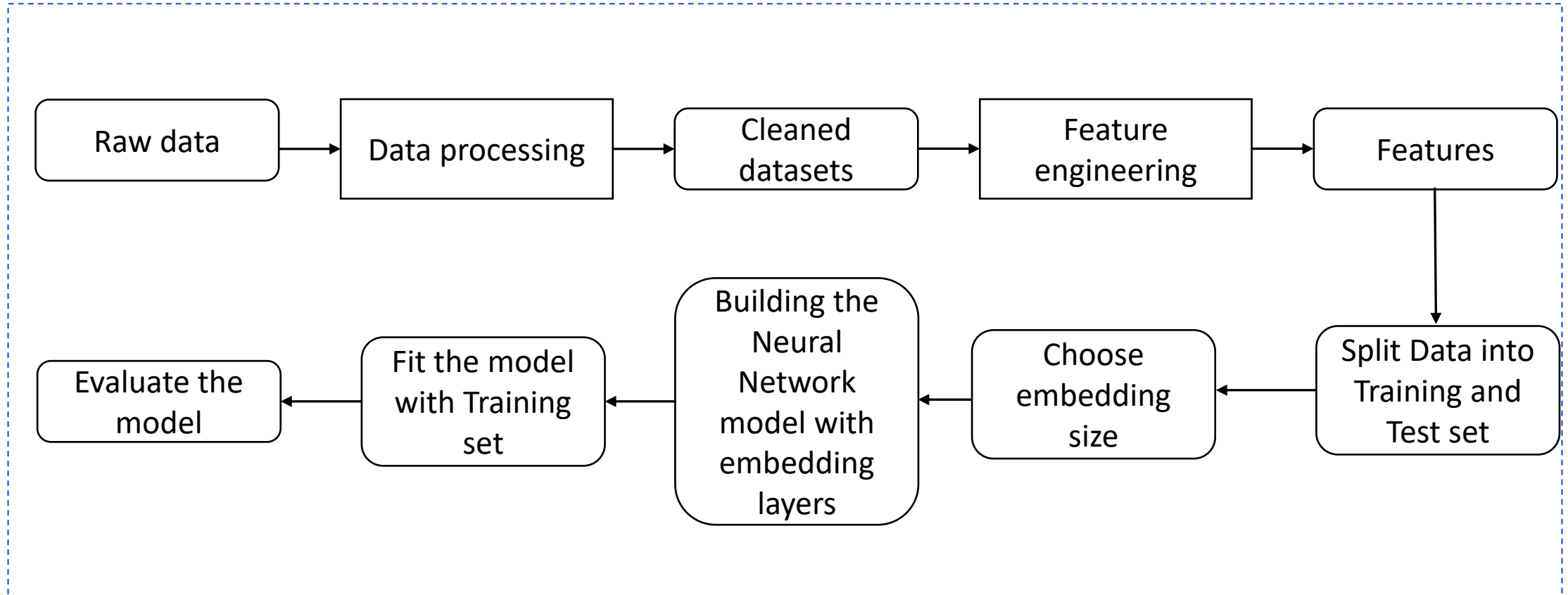
Comparing the performance of collaborative-filtering models

The bar chart shows the performance metric (such as RMSE) of different collaborative-filtering models we have built so far.
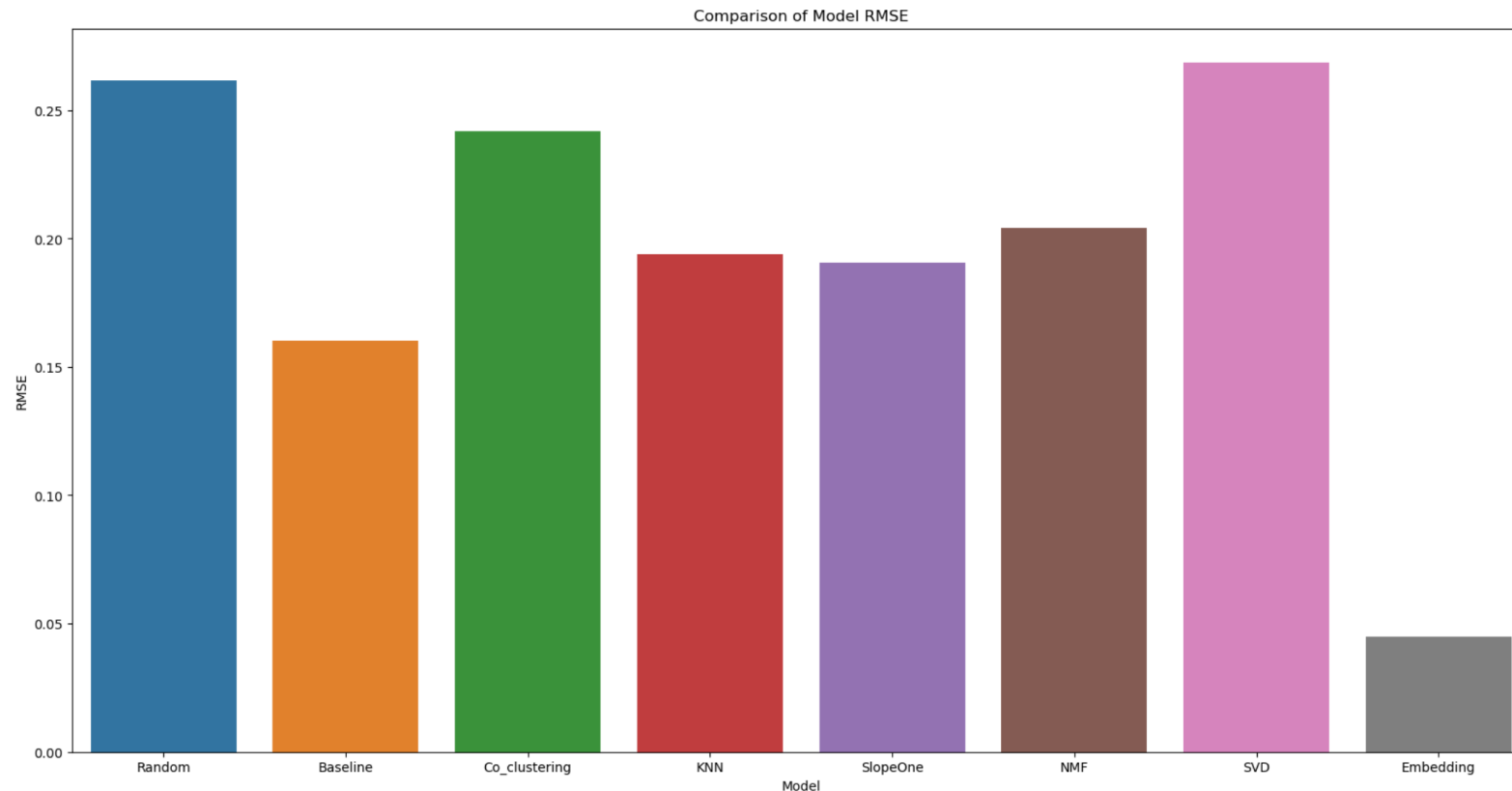
Conclusions

- The project is a good combination of different recommender systems with EDA and data visualization

- Building a content-based recommender system focusing on user profile and course similarity provides a better analysis for recommendations of the courses

- Evaluating different collaborative filtering algorithms helps to choose the right model for the personalized recommendations for the users

Appendix

GitHub link of this project:

https://github.com/MAK20202021/Course_Recommendations_ML_Capstone