Building a Personalized Online Course Recommender system with ML

João Coroa

24/05/2025

Outline

- Executive Summary
- Introduction and Background
- Exploratory Data Analysis
- Content-based RS using UL
- Collaborative-filtering based RS using SL
- Conclusion
- Appendix

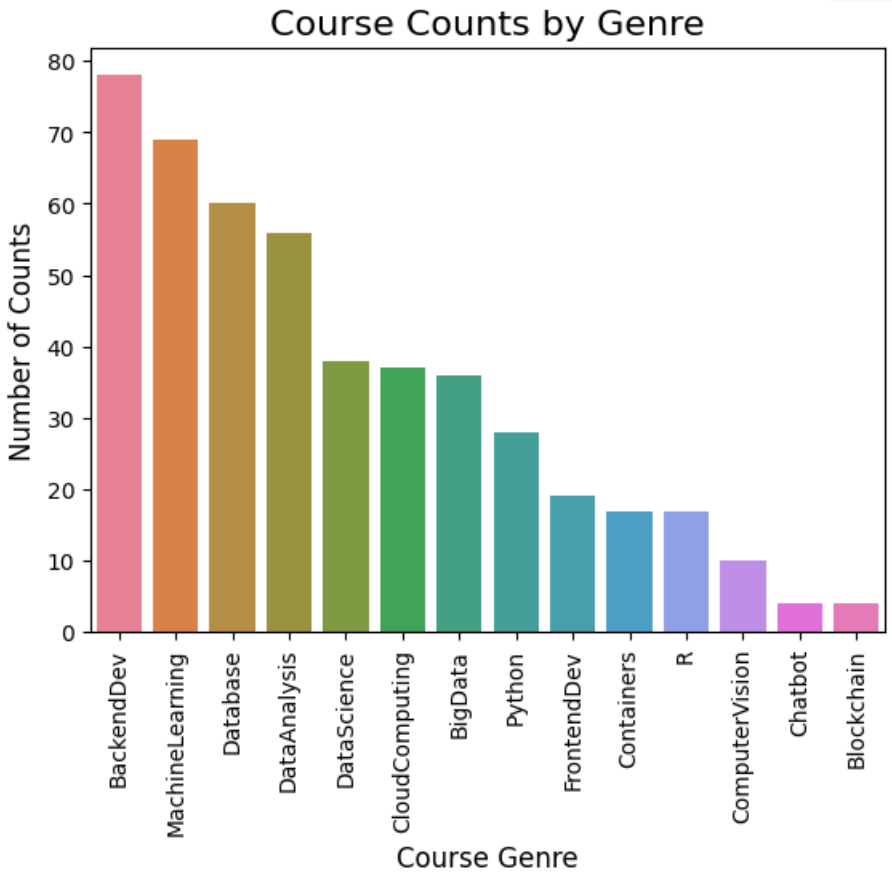
Executive Summary

- This report will recommend a specific course based on the preferences of a certain user.
- Exploratory Data Analysis reveal that most courses available are around machine learning and data analysis.
- Content-based systems recommend an avarega of 9 courses per new users, related with those topics.
- Filtering-based systems with NeuralNetworks can accuretely recommend a course.

Introduction & Background

- Big data is part of our daily life's. And because of that, deciding on what to watch and follow is increasingly more difficult.
- Machine Learning algorithms can tell which courses/movies/books we tend to like.
- The objective of this report is to recommend a specific course based on the preferences of a certain user (**Recommender Systems**)

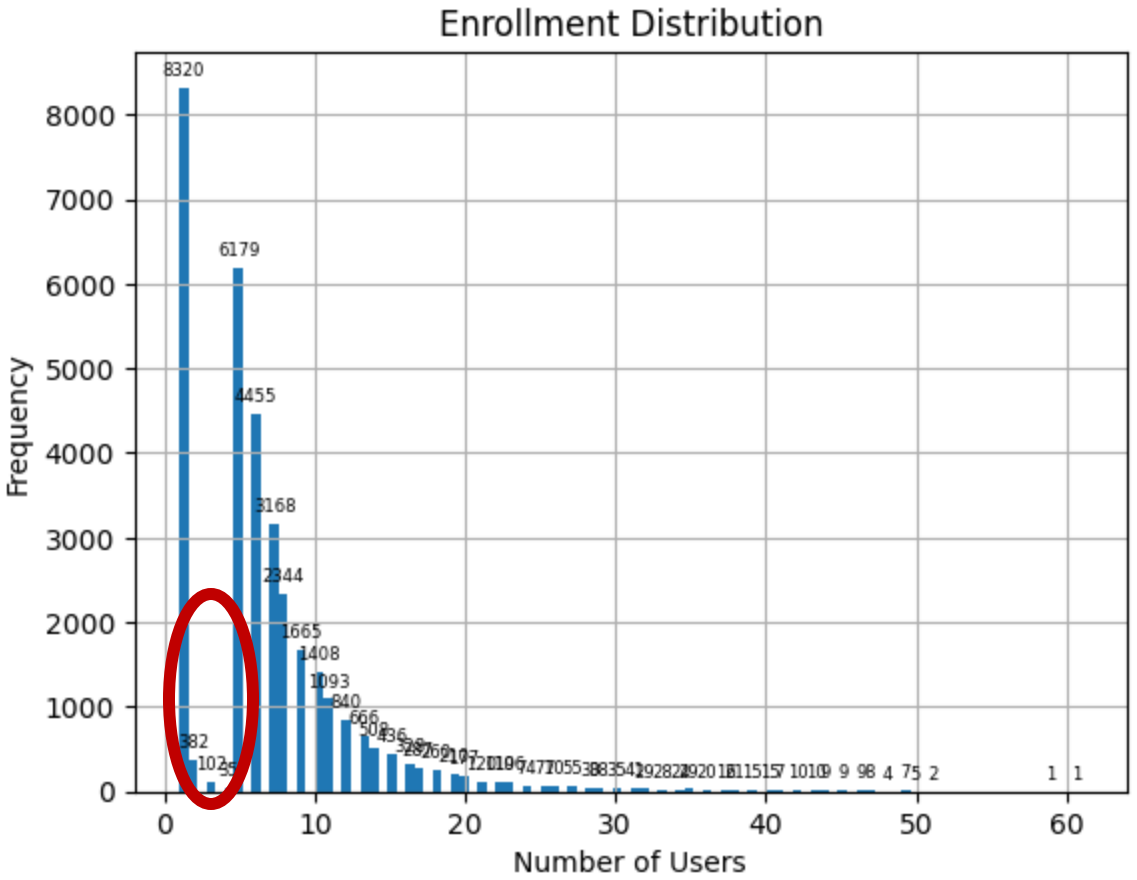
Exploratory Data Analysis



Course counts per genre

- The graph shows the number of available courses
- The data is organized in descended order and with a color palette *husl* for better visualization
- Most of the courses taught are around topics such as Backend-Developer, Machine Learning and Database
- There are few courses available on subjects such as Block-Chain, Chatbot and Computer Vision

Exploratory Data Analysis



Course enrollment distribution

- The graph shows the number of enrolments that the users rated
- The data is organized by user and the total size of ratings, plotted in a histogram with 100 bins
- Most students that are enrolled rate between 2 and 10 courses, with 2 courses being the most ratings they do
- Interestingly, there is a gap of ratings between 2 and 5 courses

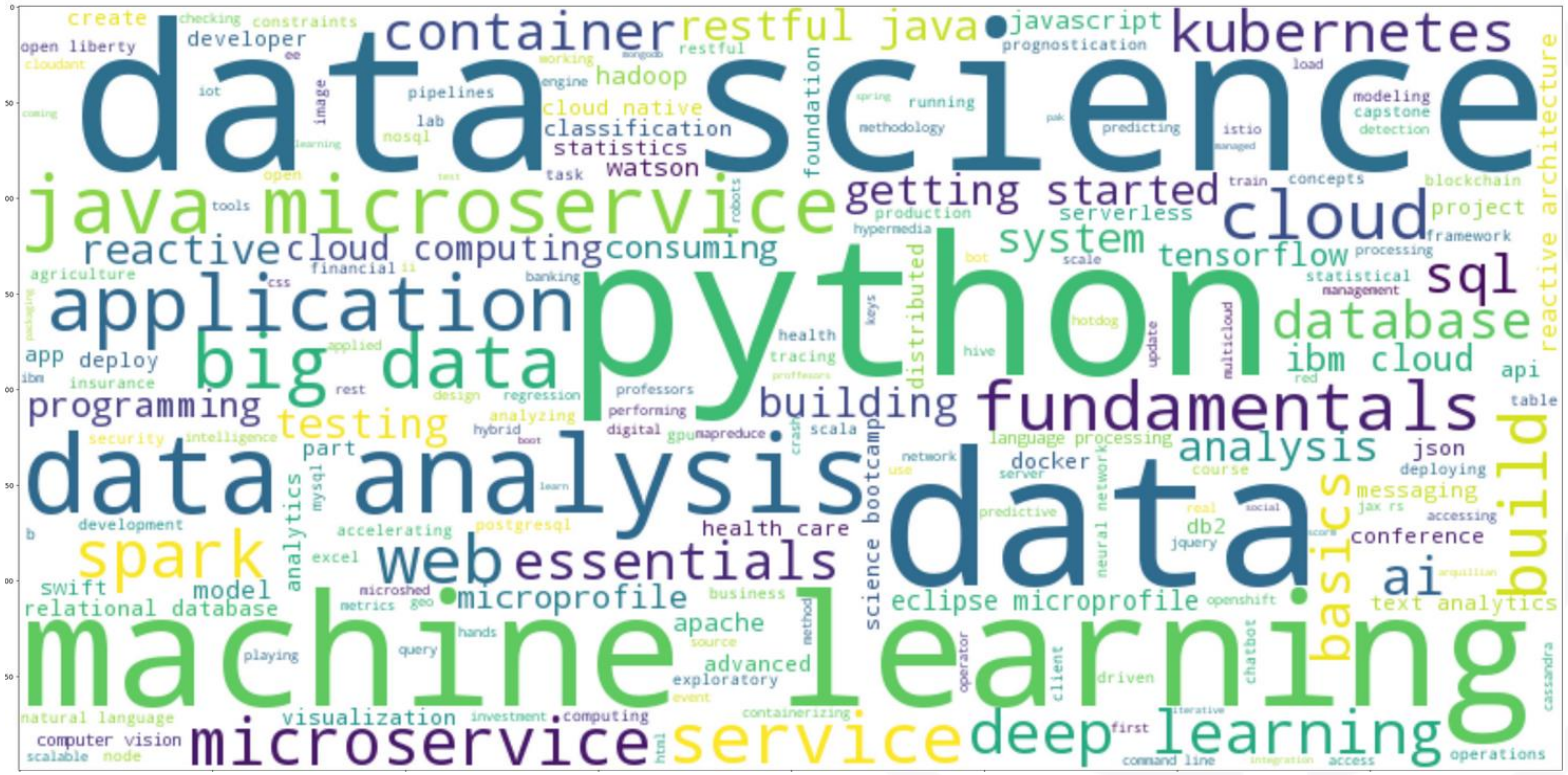
Exploratory Data Analysis

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

20 most popular courses

- The table shows the top 20 rated courses and their respective course_ID, that account for 63% of all ratings
- The data is organized by descending order
- Most positive ratings goes to courses related with data science and big data treatment
- Less ratings are given to databases, sqls and data privacy

Exploratory Data Analysis

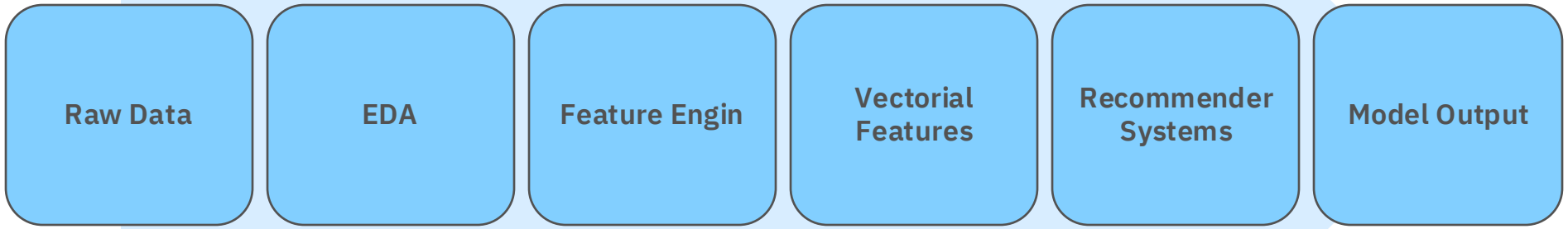


Word cloud of course titles

- Most courses are centered around data science, python and machine learning

Content-based recommender system

1. Flow-chart USING user profile and course genres



For this approach, **after data treatment (EDA)**, user profile vectors (user's given ratings) and course genres vectors are considered in **Feature Engineering** (similarity of content – **TRESHOLD SCORE**) to use as input to the recommender systems that will give us an output (a recommendation)

Content-based recommender system

1. Evaluation results

	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

1500424 rows × 3 columns

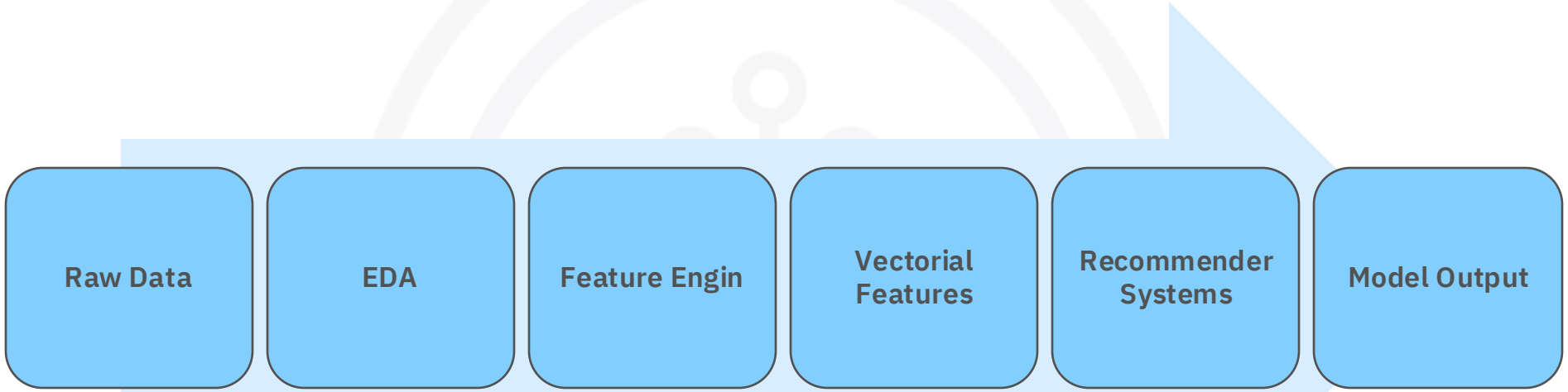
Threshold score = **10** (minimum result between dot product of user vector • ratings vector)

	COURSE_ID	count	TITLE
0	TA0106EN	17390	text analytics at scale
1	excourse22	15656	introduction to data science in python
2	excourse21	15656	applied machine learning in python
3	GPXX0IBEN	15644	data science in insurance basic statistical a...
4	ML0122EN	15603	accelerating deep learning with gpu
5	excourse06	15062	sql for data science capstone project
6	excourse04	15062	sql for data science
7	GPXX0TY1EN	14689	performing database operations in the cloudant...
8	excourse73	14464	analyzing big data with sql
9	excourse72	14464	foundations for big data analysis with sql

- Grouping by users and applying a len() and mean() function, the average new courses recommender per user is: **9.18**
- The 10 most frequent recommendations across all users

Content-based recommender system

2. Flow-chart USING course similarity



For this approach, **after data treatment (EDA)**, course genres vectors are compared in **Feature Engineering** (similarity of content – **COSINE, EUCLIDEAN DISTANCE, ETC.**) to use as input to the recommender systems that will give us an output (a recommendation)

Content-based recommender system

2. Evaluation results

```
res_dict = {}
users, courses, sim_scores = gener
res_dict['USER'] = users
res_dict['COURSE_ID'] = courses
res_dict['SCORE'] = sim_scores
res_df = pd.DataFrame(res_dict, co
res_df.head()
```

	USER	COURSE_ID	SCORE
0	17	TMP0101EN	0.889499
1	17	TA0105EN	0.659829
2	21	excourse67	0.708214
3	21	excourse72	0.652535
4	21	excourse74	0.650071

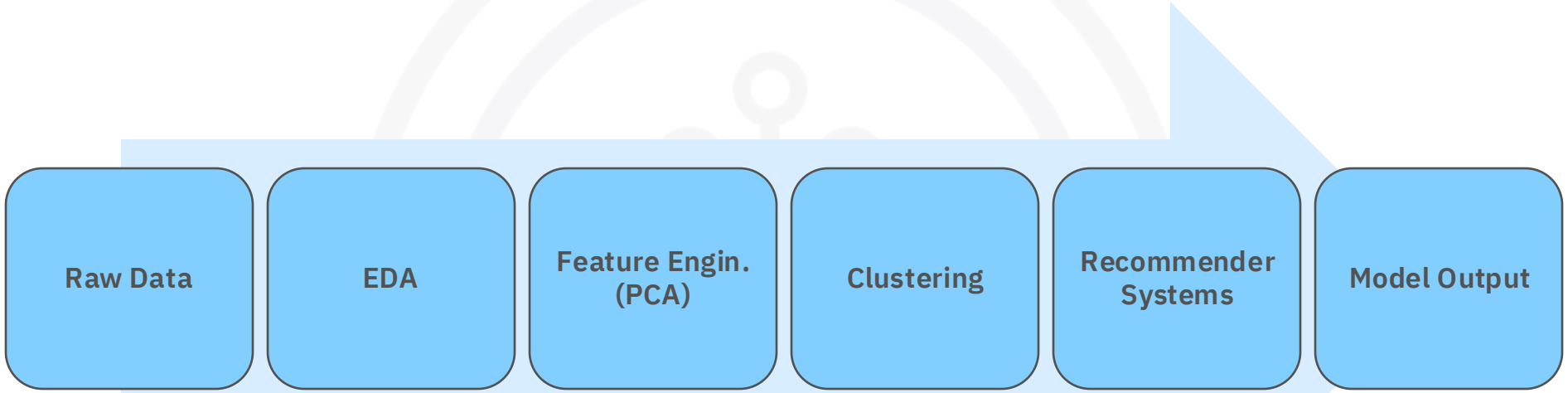
Threshold score = **65%** (to be recommended, at least 65% of similarity between courses)

	COURSE_ID	count	TITLE
0	CB0101EN	1429	build your own chatbots
1	excourse63	1413	a crash course in data science
2	DS0110EN	1356	data science with open data
3	ML0120ENv3	979	deep learning with tensorflow
4	TA0105	968	text analytics 101
5	ML0120EN	899	deep learning with tensorflow
6	ML0120ENv2	873	deep learning with tensorflow
7	ML0122ENv3	647	accelerating deep learning with gpus
8	CC0103EN	517	ibm cloud essentials v3
9	DS0132EN	441	data ai jumpstart your journey

- Grouping by users and applying a len() and mean() function, the average new courses recommender per user is: **8.78**
- The 10 most frequent recommendations across all users

Content-based recommender system

3. Flow-chart USING clustering

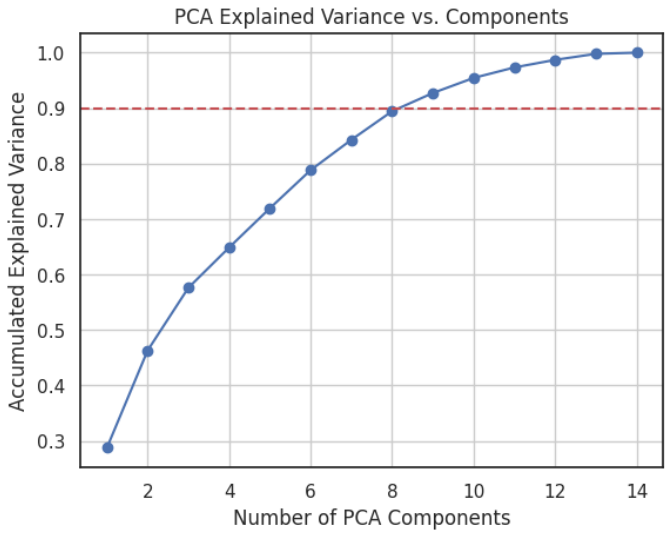


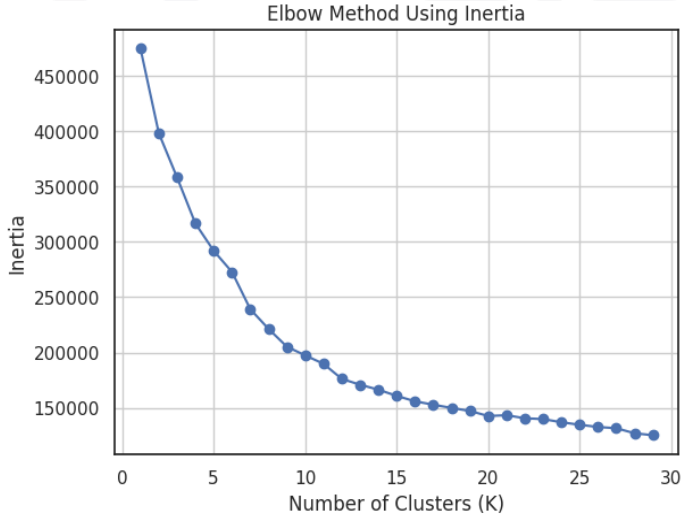
For this approach, **after data treatment (EDA) and feature engineering (PCA)**, users are aggregated together with KMeans to use as input to the recommender systems that will give us an output (a recommendation)

Content-based recommender system

3. Evaluation results

n_components = 9 || n_clusters = 20





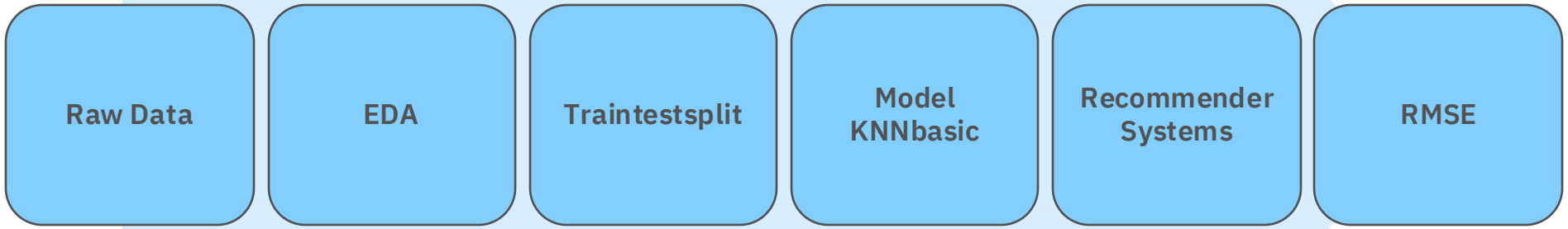
- Grouping by users and applying a len() and mean() function, the average new courses recommender per user is: **8.14**

	recommended_course	count	COURSE_ID	TITLE
0	DW0101EN	32632	DW0101EN	introduction to machine learning with sound
1	DB0151EN	32326	DB0151EN	nosql and dbaas 101
2	ML0151EN	31704	ML0151EN	machine learning with r
3	WA0101EN	31635	WA0101EN	watson analytics 101
4	ML0120ENv2	31320	ML0120ENv2	deep learning with tensorflow
5	SC0101EN	31162	SC0101EN	scala 101
6	TA0105EN	31102	TA0105EN	text analytics 101
7	TA0105	31094	TA0105	text analytics 101
8	CC0103EN	31013	CC0103EN	ibm cloud essentials v3
9	BD0131EN	31010	BD0131EN	moving data into hadoop

The 10 most frequent recommendations across all user

Collaborative-filtering recommender system

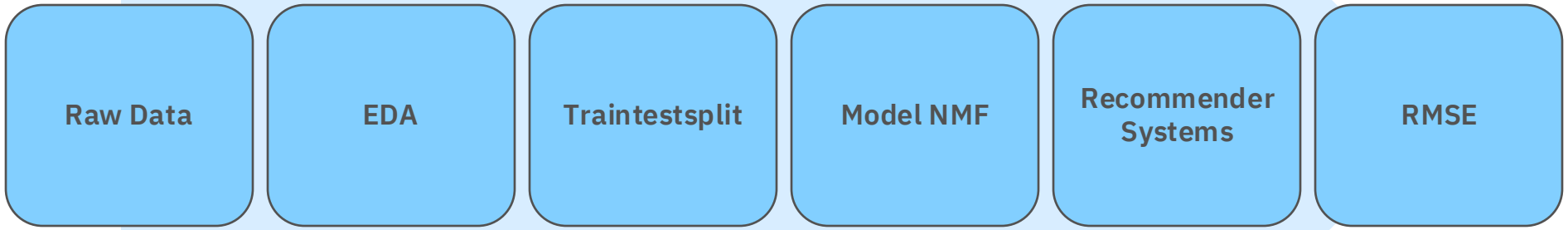
1. Flow-chart KNN-based



For this approach, **after data treatment (EDA) and train-test-split,** we decide **similarity options** (cosine/pearson, user_based/item_based) and model with KNNbasics and the train set. Then we predict using the testset to use as output to the recommender systems. We then calculate the RMSE to acertain if it was a good prediction or not.

Collaborative-filtering recommender system

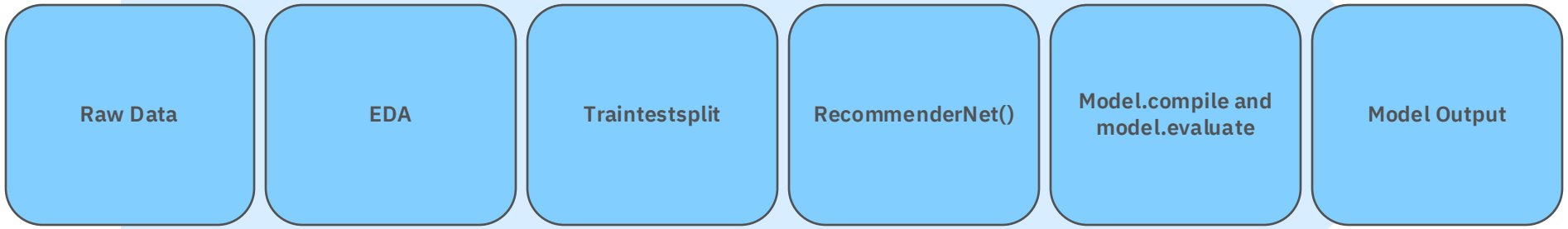
2. Flow-chart NMF-based



For this approach, **after data treatment (EDA) and train-test-split,** we decide **the arguments** (init_low=0.5, init_high = 5.0, n_factors=32) and model with NMF and the train set. Then we predict using the testset to use as output to the recommender systems. We then calculate the RMSE to acertain if it was a good prediction or not.

Collaborative-filtering recommender system

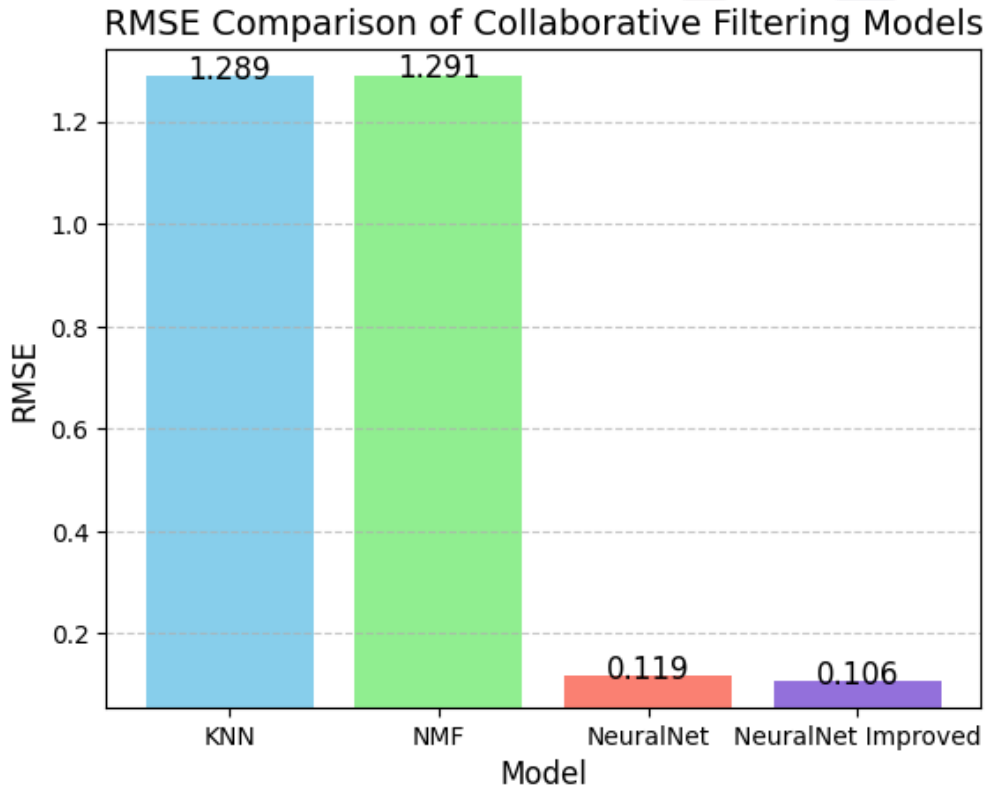
3. Flow-chart Neural Network Embedding



For this approach, **after data treatment (EDA) we encode the data.** We traintestsplit the data and decide **the arguments** (embedding size, dense layers and activation function) for compiling the model with RecommenderNet(). Then we fit the model using the trainset and we finally evaluate with the testset . We look at the RMSE to acertain if it was a good prediction or not.

Collaborative-filtering recommender system

1-3. Evaluation Results



- The NeuralNetm model outperform the traditional CF methods by a large margin
- The improved NeuralNet shows the **benefit of tuning the parameters**
- NeuralNet is capturing user-item interactions **much better**

CONCLUSION

- Content and collaborative-filtering are different ways to recommend a course
- Parameter tunning is very important for an accurate prediction and can easily misled a recomendation
- In content-based, the average number of courses recommender per user is 9. The 10 most popular courses have to do with data science, machine learning and data analysis.
- In filtering-based neural networks are much better in capturing user-item interactions, predicting a recommendation much better.

APPENDIX

- https://github.com/CoroaPT/Recommending_Systems.git