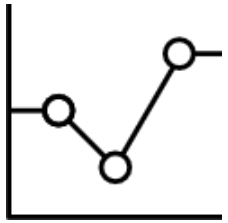
I was not able to export my notes.
If notes are necessary to rate the presentation you can find it here:

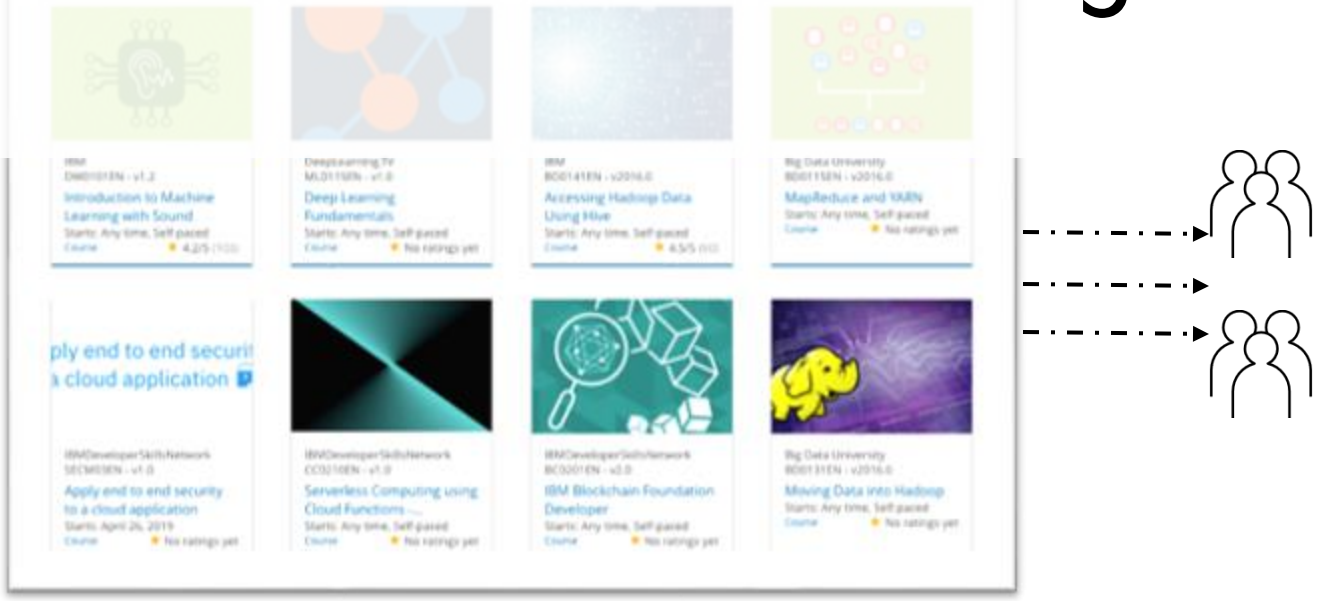
https://1drv.ms/p/s!AghPRykdUVbRgQm-OOemp3d6CDn4?e=fa8GSb

(klick the notes button on the bottom right)



Build a Personalized Online Course Recommender System with Machine Learning

Maximilian Fleck
16.01.2024



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

Massive Open Online Courses (MOOCs) startup recommender system project to improve learners' learning experience and company's revenue

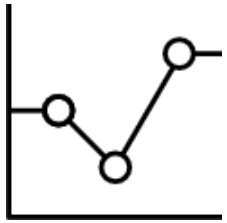
HYPOTHESIS:

- recommender systems improve learners learning experience
- this should lead to more enrolled and finished courses
- users spend more time on our platform
- companies revenue raises

TASK DERIVED FROM HYPOTHESIS:

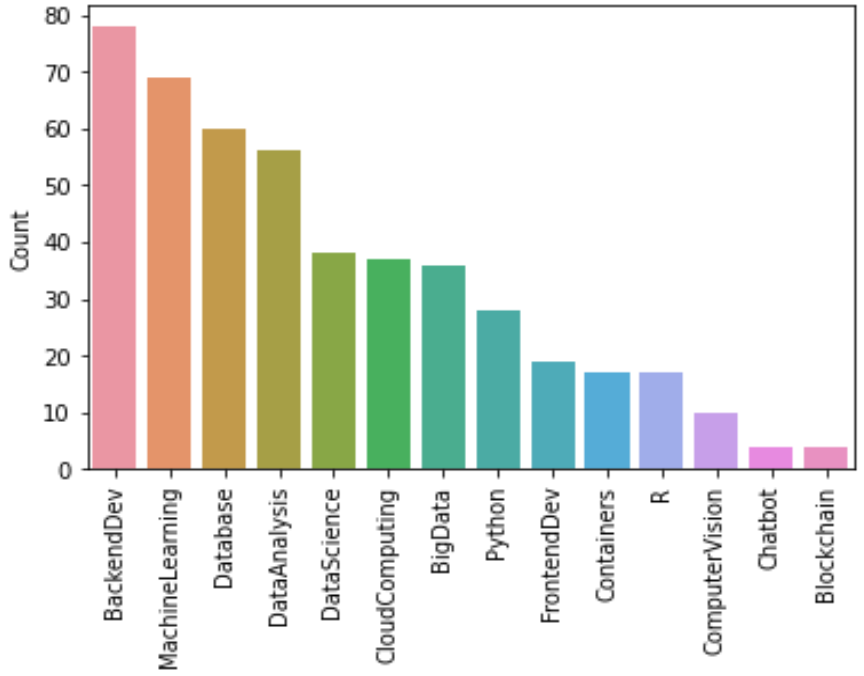
we need to find the best performing recommender system in off-line evaluations

Exploratory Data Analysis



Course counts per genre

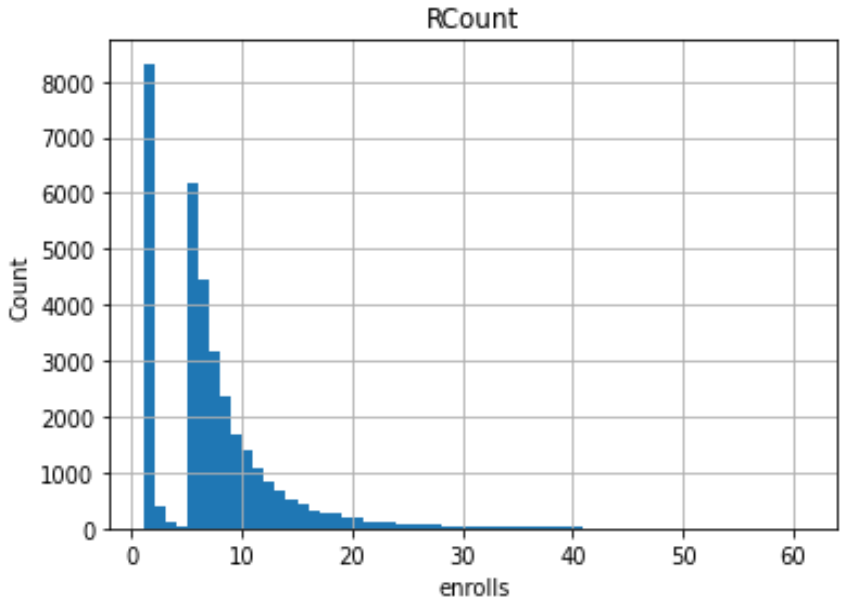
- the barchart below shows course genre counts
- BackendDev is the most common course genre followed by MachineLearning



Course enrollment distribution

- the enrollment distribution shows how many users enrolled in just 1 course or how many enrolled 10 courses, etc.
- the minimum enrolls is 1 whereas the maximum enrolls are 61
- many users enrolled only one course
- most users enrolled between 5 and 10 courses

→ we want users to take more courses
→ we can use recommender systems to improve learners' learning experience
→ this leads to more enrolled courses
→ and improves company's revenue

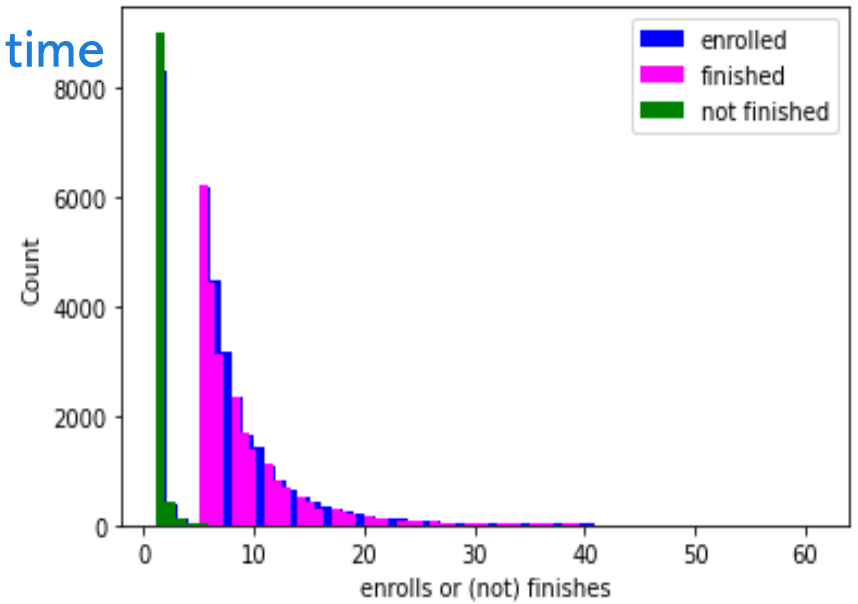


Course enrollment distribution

- this enrollment distribution shows how many users enrolled, finished or did not finish courses
- some of the enrolled and not finished courses might be finished in the future

→ most users only enroll and not finish one course at a time

→ users that finish courses finish more than one



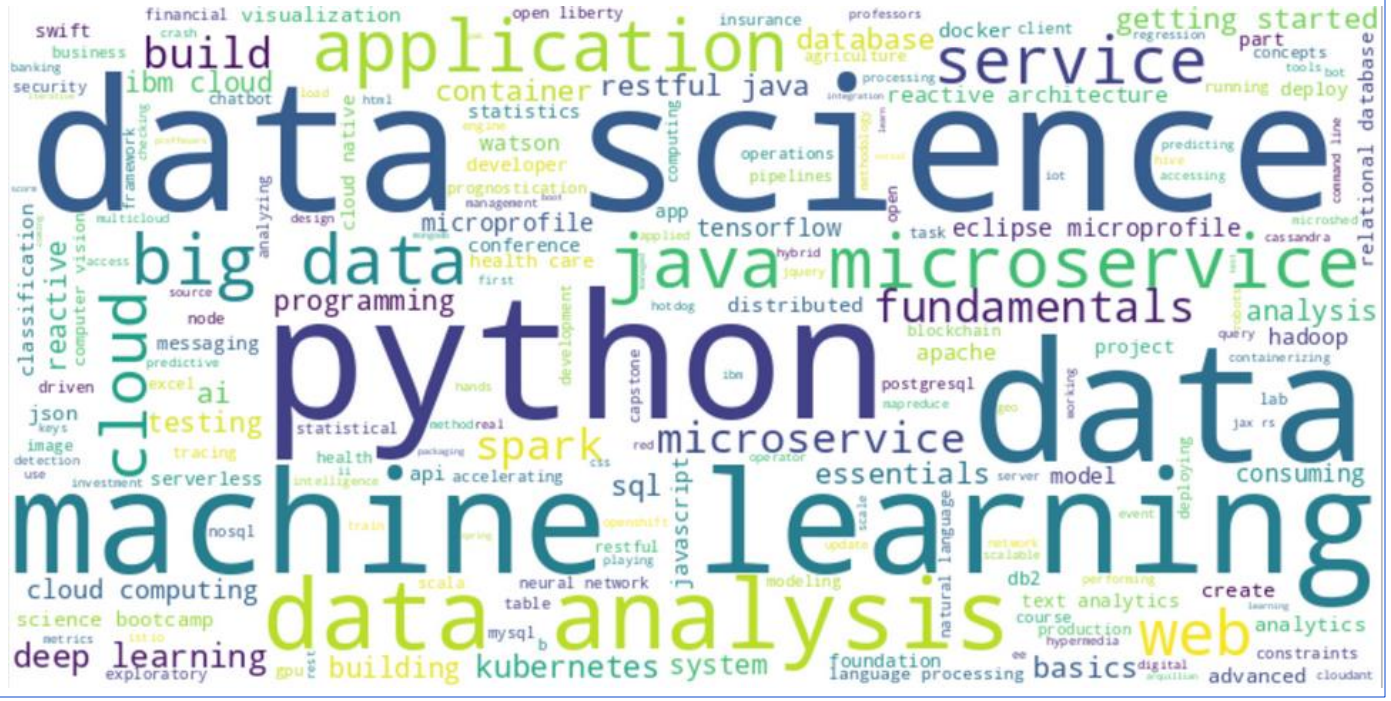
20 most popular courses

- 20 most popular courses are shown here
- high interest in python
- key words like “introduction”, “101”, “fundamentals”, etc. indicate that most courses are introductionary courses

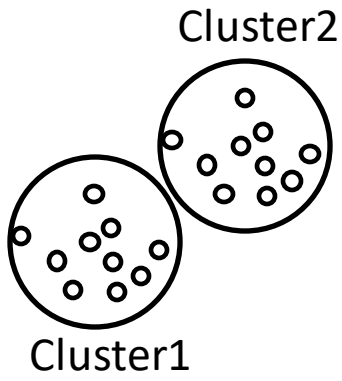
	TITLE	Enrolls
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199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

Word cloud of course titles

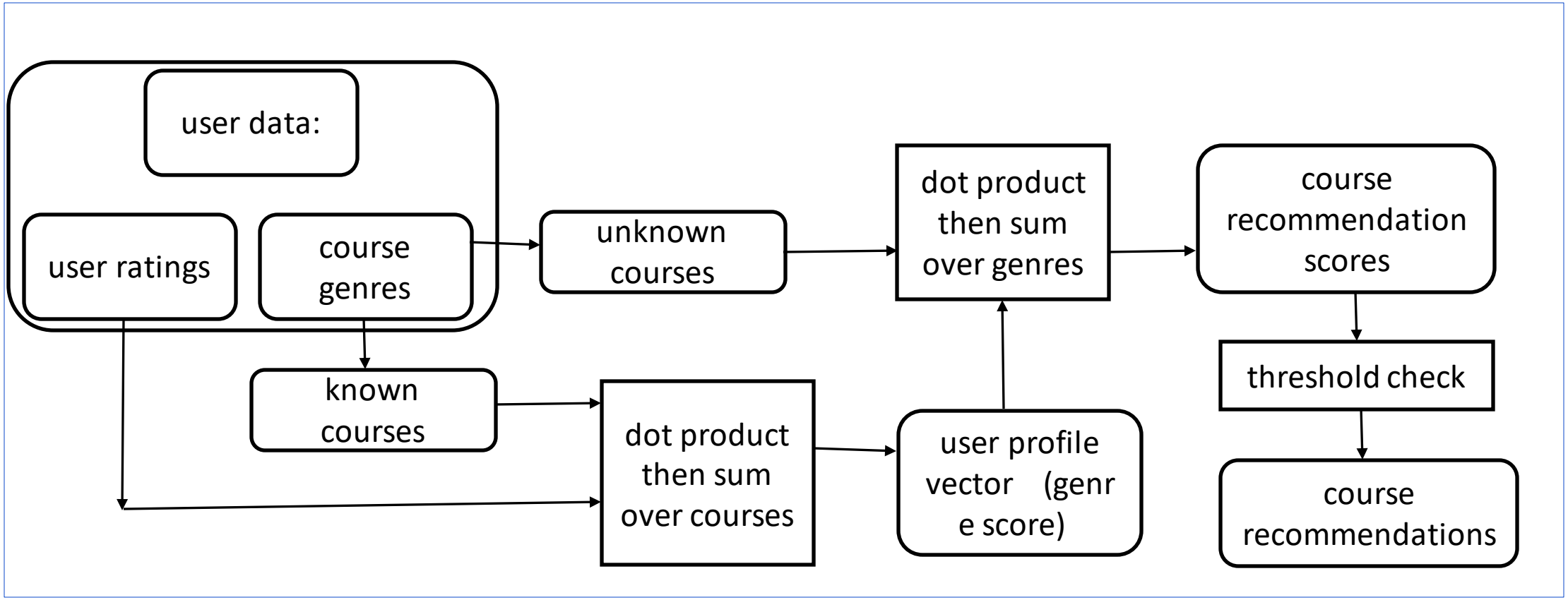
the word cloud provides a quick visualization of the popular learning topics across all the courses:



Content-based Recommender System using Unsupervised Learning



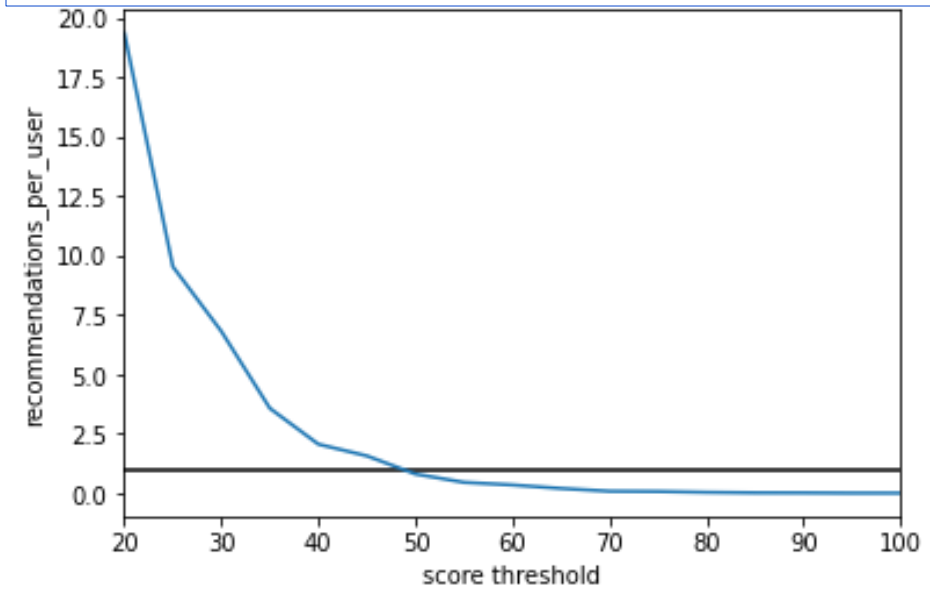
Flowchart of content-based recommender system using user profile and course genres



Evaluation results of user profile-based recommender system

recommendation scores are based on user ratings and course similarities are between 10 and 99 (all similarity scores below 10 are not saved in the initial scan)

The number of recommended courses per user depends on the score threshold



40 is a good choice with around one recommendation per user on average

:

COURSE_ID	no	TITLE
excourse72	93	foundations for big data analysis with sql
excourse73	92	analyzing big data with sql
TMP0105EN	92	getting started with the data apache spark ma...
RP0105EN	76	analyzing big data in r using apache spark
SC0103EN	74	spark overview for scala analytics
BD0212EN	59	spark fundamentals ii
excourse31	41	cloud computing applications part 2 big data...
excourse05	25	\r\ndistributed computing with spark sql
GPXX0IBEN	25	data science in insurance basic statistical a...
excourse21	23	applied machine learning in python

Suggestions are not in line with the most popular courses

Evaluation results of user profile-based recommender system

Courses with genres Database, DataScience and BigData are very heavily weighted but also quite generic. Most courses might fall into these genres. This explains the proposals on the previous slide.

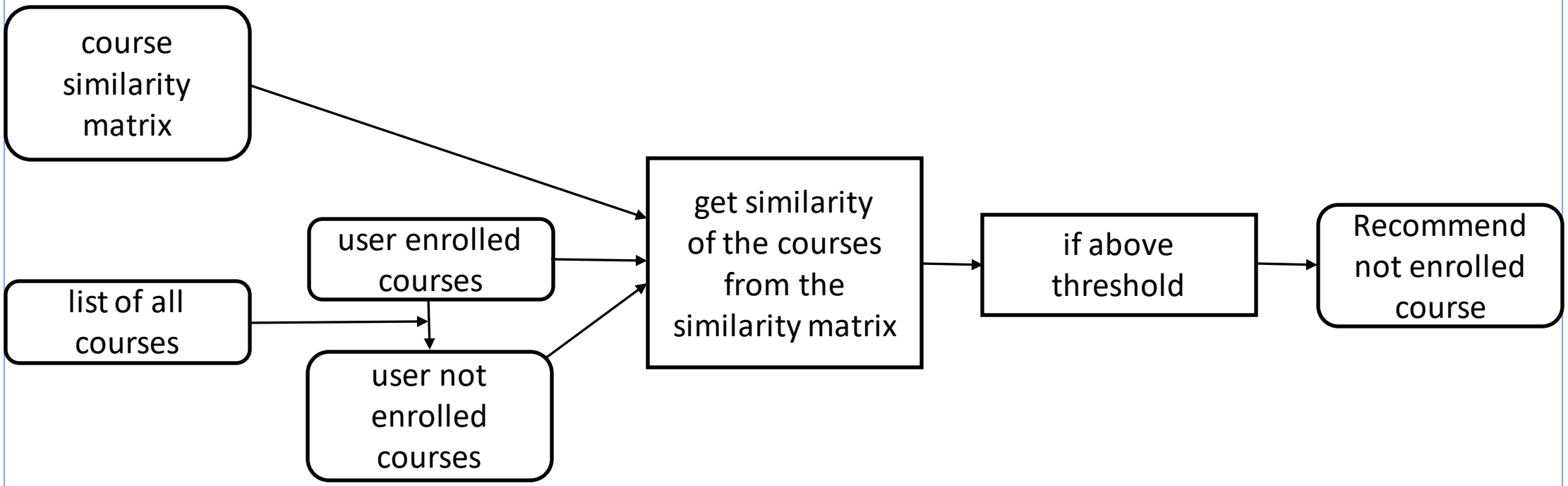
	user	Database	Python	CloudComputing	DataAnalysis	Containers	MachineLearning	ComputerVision	DataScience	BigData
count	3.390100e+04	33901.000000	33901.000000	33901.000000	33901.000000	33901.000000	33901.000000	33901.000000	33901.000000	33901.000000
mean	1.064064e+06	5.518569	3.493791	2.307100	3.624701	0.998938	3.048022	0.001770	5.087343	4.750450
std	4.972578e+05	7.611941	4.227254	3.841858	4.760135	2.351764	4.624004	0.072846	5.230697	7.216228
min	2.000000e+00	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	6.813480e+05	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	1.043907e+06	3.000000	3.000000	0.000000	3.000000	0.000000	0.000000	0.000000	3.000000	2.000000
75%	1.451159e+06	9.000000	6.000000	3.000000	6.000000	0.000000	3.000000	0.000000	9.000000	6.000000
max	2.103039e+06	63.000000	18.000000	38.000000	48.000000	15.000000	39.000000	3.000000	32.000000	54.000000

In general, courses that are assigned to many genres are favored.

Therefore specific recommendations are rather uncommon...

Flowchart of content-based recommender system using course similarity

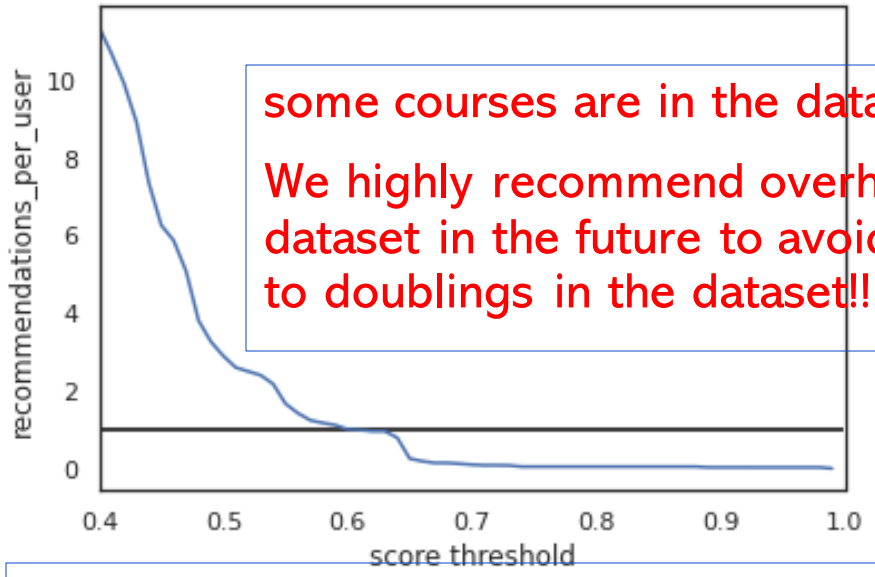
- the similarity matrix is given in this example, it can be rebuilt with other metrics or extended when new courses are available



Evaluation results of course similarity based recommender system

recommendation scores are based on user ratings and course similarities are between 0 and 1
(all similarity scores below 0.3 are not saved in the initial scan)

The number of recommended courses per user depends on the score threshold



some courses are in the dataset twice!!!

We highly recommend overhauling the dataset in the future to avoid errors due to doublings in the dataset!!!

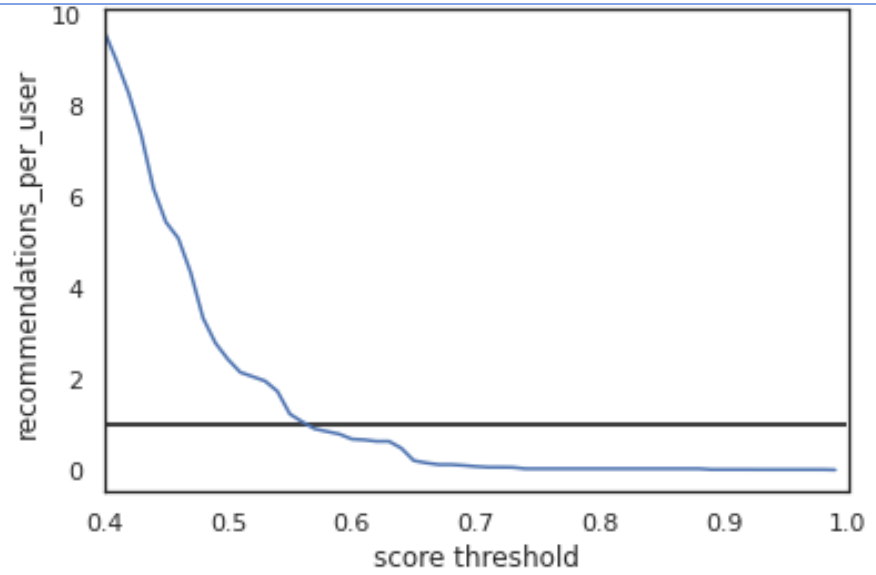
0.5 is a good choice with around one recommendation per user on average

COURSE_ID	no	TITLE
TMP107	269	data science bootcamp with python
excourse22	260	introduction to data science in python
excourse62	260	introduction to data science in python
excourse32	154	introduction to data analytics
DS0110EN	126	data science with open data
WA0103EN	101	watson analytics for social media
DA0151EN	94	data analysis using r 101
excourse86	76	the r programming environment
excourse82	76	getting started with data visualization in r
excourse81	76	data analysis with r programming

Evaluation results of course similarity based recommender system

We deleted courses appearing twice under the same title in the courses dataframe (not the users dataframe!!!). Avoiding double nominations and double recommendations changes the results slightly.

The number of recommended courses per user is shifted to the left as expected



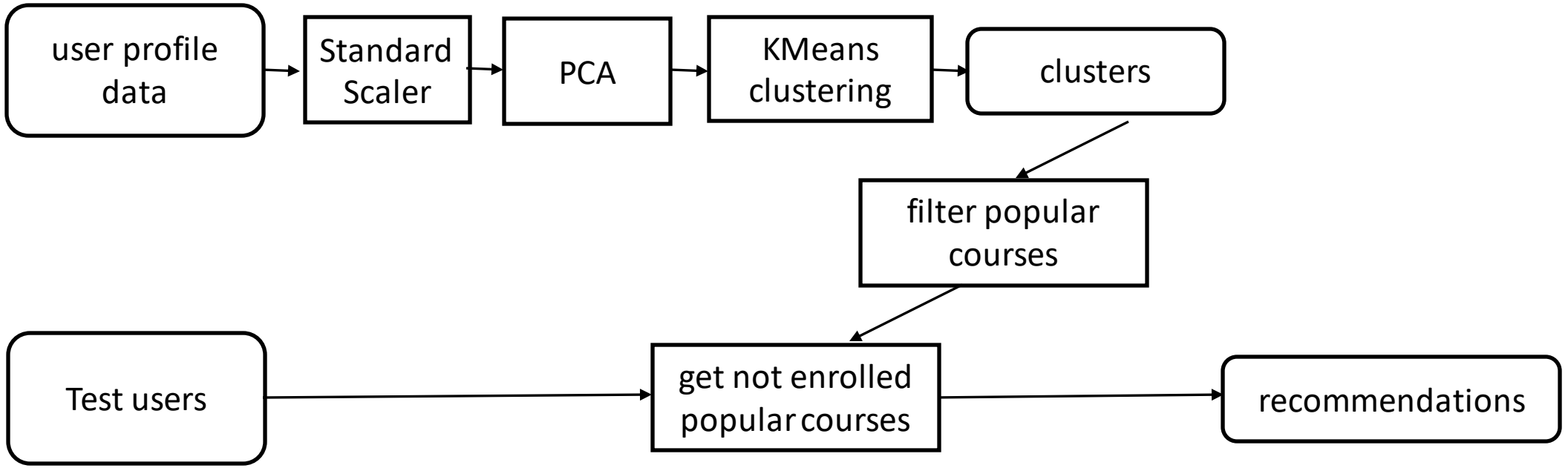
COURSE_ID	no	TITLE
TMP107	280	data science bootcamp with python
excourse22	260	introduction to data science in python
excourse32	154	introduction to data analytics
DS0110EN	126	data science with open data
WA0103EN	101	watson analytics for social media
DA0151EN	94	data analysis using r 101
excourse86	76	the r programming environment
excourse37	76	data analysis with r programming
excourse82	76	getting started with data visualization in r
excourse80	76	r programming

0.5 is a good choice with around one recommendation per user on average

Suggestions are not in line with the most popular courses

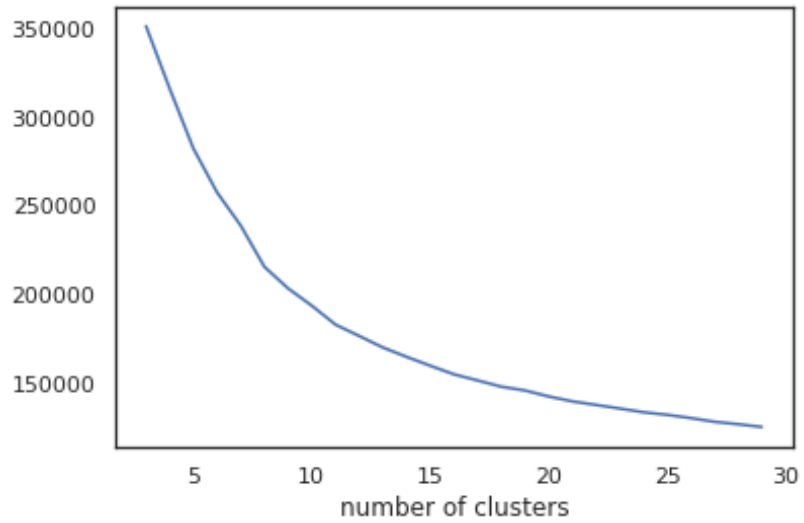
Flowchart of clustering-based recommender system

- A full user profile database and test users are given



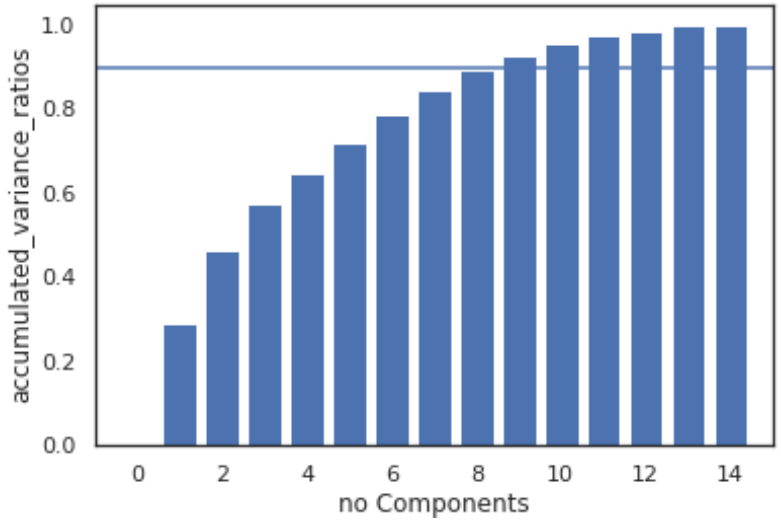
Number of Clusters

- We choose number of clusters with the elbow method
- Number of clusters : 20



Number of PCA Components

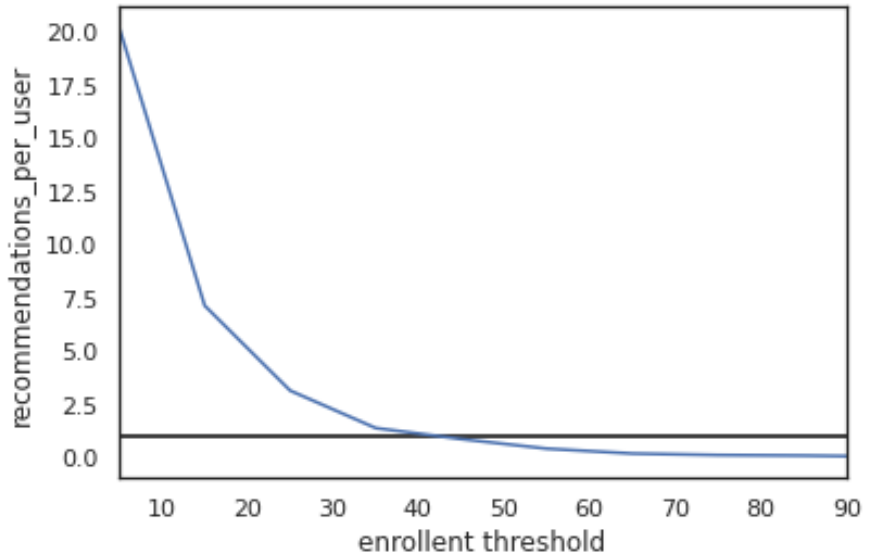
- We choose number of PCA components as the lowest number that explains more than 0.9 accumulated variance
- Number of PCA components: 9



Evaluation results of clustering-based recommender system

The enrollment threshold (how many enrollments a course needs within a cluster to be recommended). Number of clusters is 20. Number of PCA components is 9.

The number of recommended courses per user changes with enrollment threshold

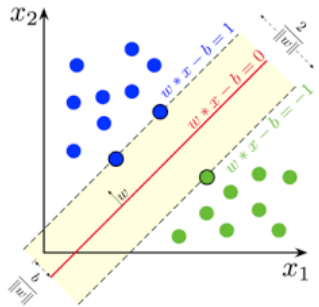


We choose an enrollment threshold of 30

	count	TITLE
DS0101EN	307.0	introduction to data science
PY0101EN	304.0	python for data science
DS0103EN	169.0	data science methodology
BD0101EN	160.0	big data 101
BD0111EN	153.0	hadoop 101
ML0101ENv3	146.0	machine learning with python
RP0101EN	146.0	r for data science
DS0105EN	120.0	data science hands on with open source tools
BD0131EN	74.0	moving data into hadoop
ML0115EN	73.0	deep learning 101

Suggestions are in line with the most popular courses indicating reasonable suggestions

Collaborative-filtering Recommender System using Supervised Learning



Comments on the dataset

- The initial dataset only comes with ratings between 2 (10976 samples) and 3 (222330 samples)
- This is not useful for a recommender system. We don't want to distinguish between enrolled and completed courses but between topics a user is interested in (rating 2 or 3) and those a user is not interested in (rating 0)
- We filled the not rated topics with zeros and gained a large dataset (4271526 samples) but working with this amount of data is not possible on a standard computer
- So we randomly choose 30000 zero rated samples and added them to the initial dataset to gain the (incomplete) dataset we will work with in the following study

Comments on the benchmarks

We choose two benchmarks:

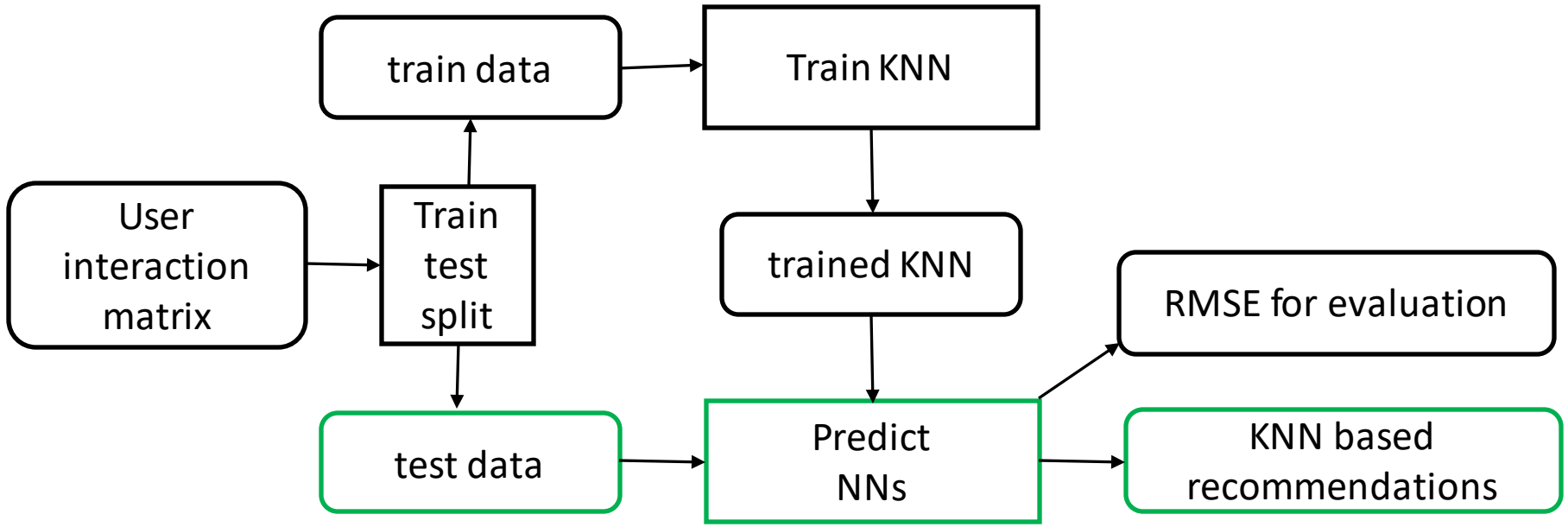
- Random samples of 0, 2 and 3 leading to an RMSE of 1.965
- All samples are either 0, 2 or 3 leading to an RMSE of 1.633 when all samples equal 2

Any other method should beat those !

Goal of the collaborative-filtering recommender system is to predict how a certain user would rate a certain item. They need to be trained before they can be used.

Flowchart of KNN based recommender system

- This flowchart shows both, the training and testing process but also that recommendations (marked by green frame) can be made based on KNNs
- Training basically means saving the training data. Hyperparameters are metric and number of neighbors used for predictions and will be discussed in the following



Comments on KNN based recommender system

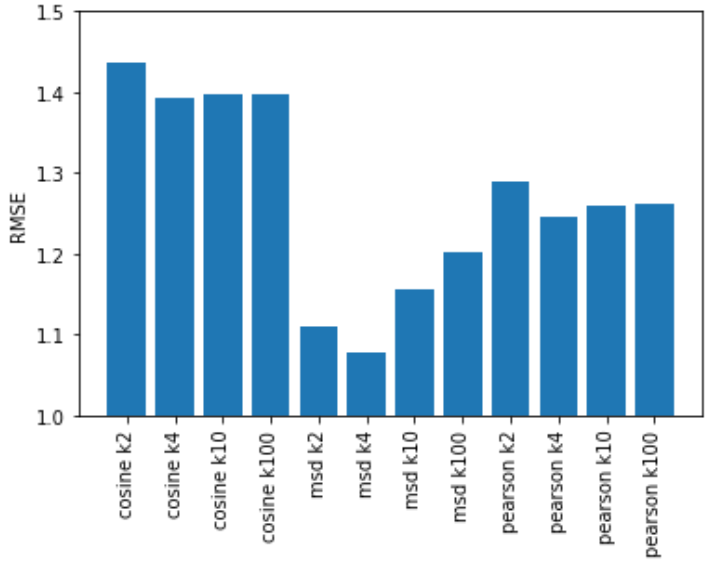
User based KNN:

- the dataset comes with around 30000 users and around 120 items
- The user based KNN is super-expensive because we compare the user of interest to every other user
- Consequently predictions are also expensive
- Hyperparameters don't play an important role
- Therefore it doesn’t perform very well: RMSE 1.454 (metric mse and k=4)

Comments on KNN based recommender system

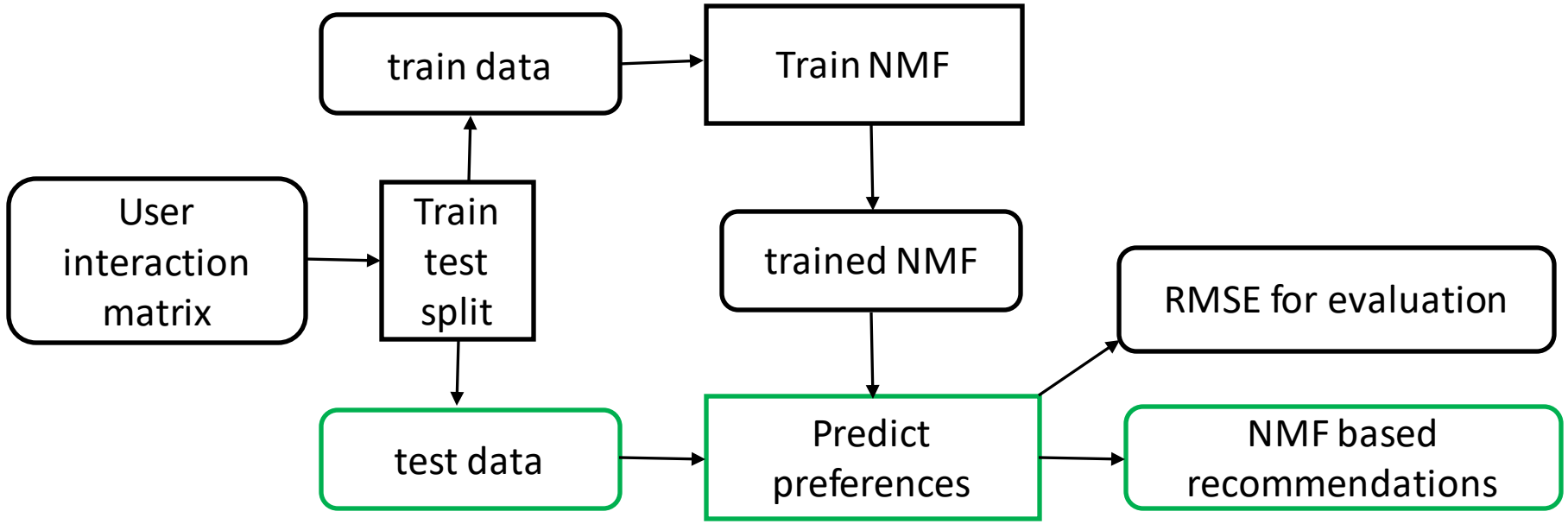
Item based KNN:

- the dataset comes with around 30000 users and around 120 items
- The item based KNN is fast in predicting
- Hyperparameters play an important role
- Optimal k = 4 with msd metric
- Therefore it doesn't perform very well: RMSE 1.078



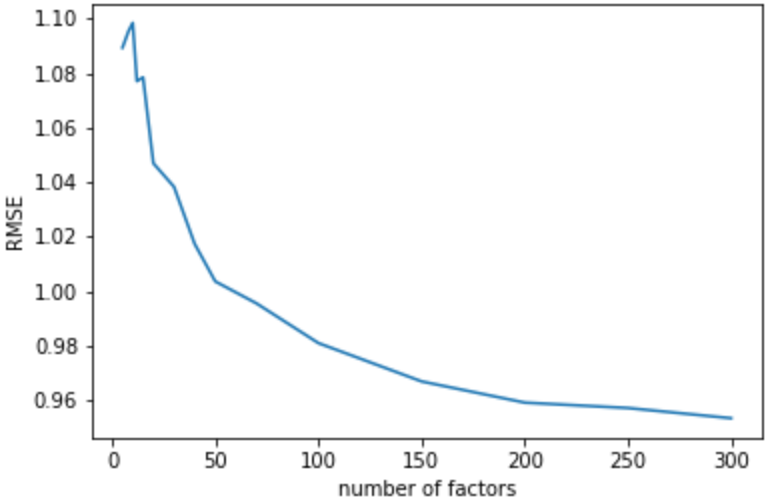
Flowchart of NMF based recommender system

- This flowchart shows both, the training and testing process but also that recommendations (marked by green frame) can be made based on NMFs
- Training means splitting user interaction matrix into user and item matrix with transformed latent features with number of factors hyperparameter (150) as dimension



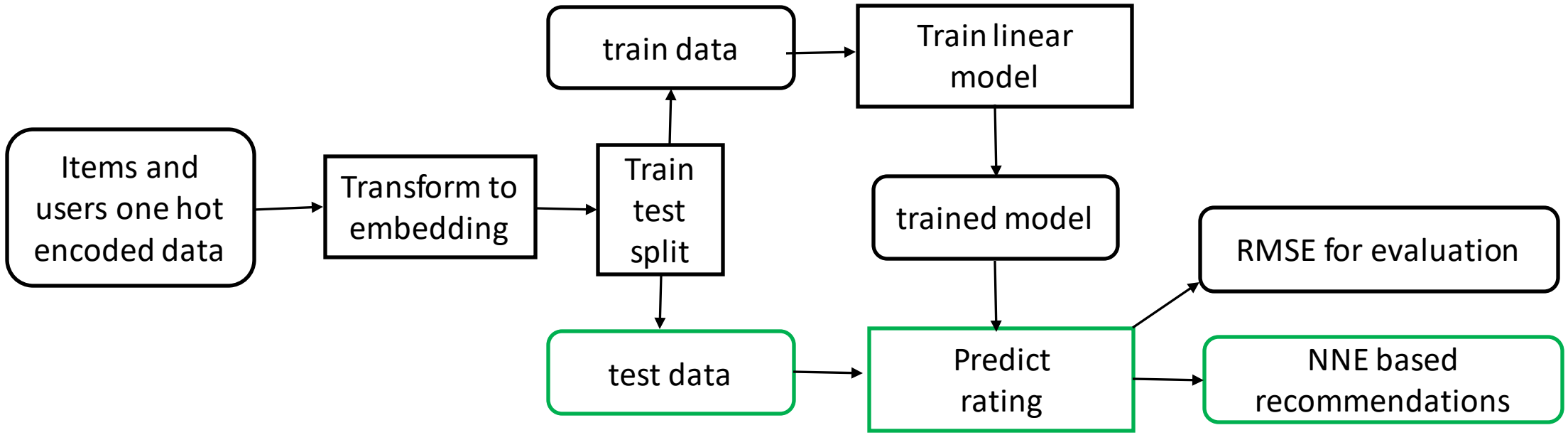
Comments on NMF based recommender system

- the number of factors hyperparameter needs to be tuned
- estimated ratings between 1 and 4 (i.e. +1) because the ratings need to be positive values for NMF
- with the elbow method on RMSE we end up with 150 factors and an RMSE of 0.966
- we use NMF to fill the gaps in the dataset. Therefore a number of factors higher than number of items is okay



Flowchart of Neural Network Embedding based recommender system

- This flowchart shows both, the training and testing process but also that recommendations (marked by green frame) can be made based on NN embedding regression

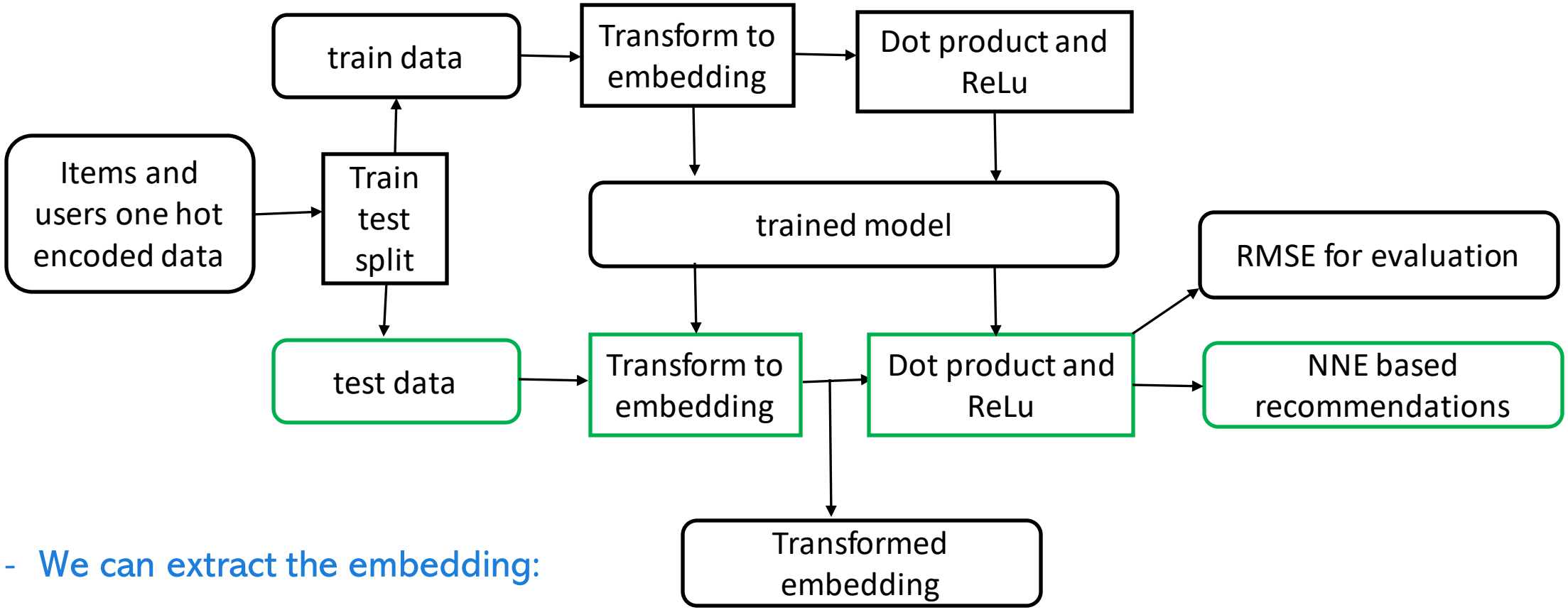


Comments on Neural Network Embedding based recommender system

- Ridge regression performs best with 1.466 RMSE (alpha=0.2)
- Lasso and ElasticNet are also tested but worse
- The results are not satisfying, maybe the embedding has too many dimensions (16)
- The regression model output y range is between 0.5 and 1.96, the training data y range between 0 and 3
- The model does not perform good and needs to be reevaluated
- The optimal embedding dimension of the full NN can be compared to the dimension of 16 of this embedding

Flowchart of ReLu Neural Network Embedding based recommender system

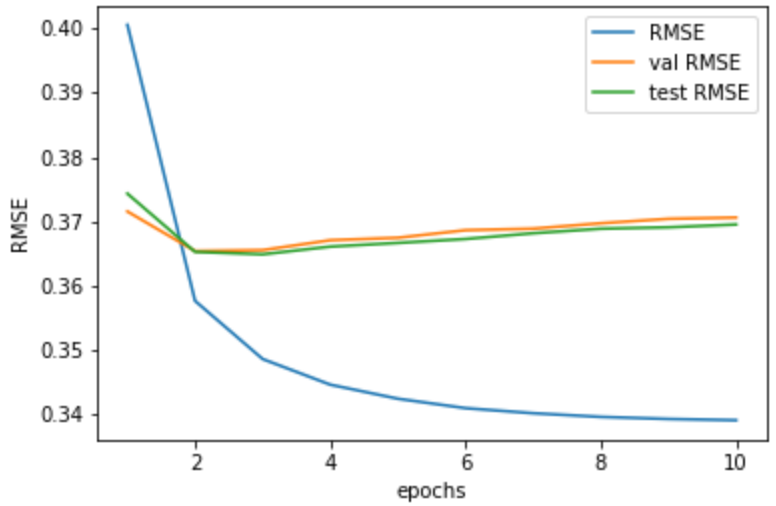
- This flowchart shows both, the training and testing process but also that recommendations (marked by green frame) can be made based on NN with relu

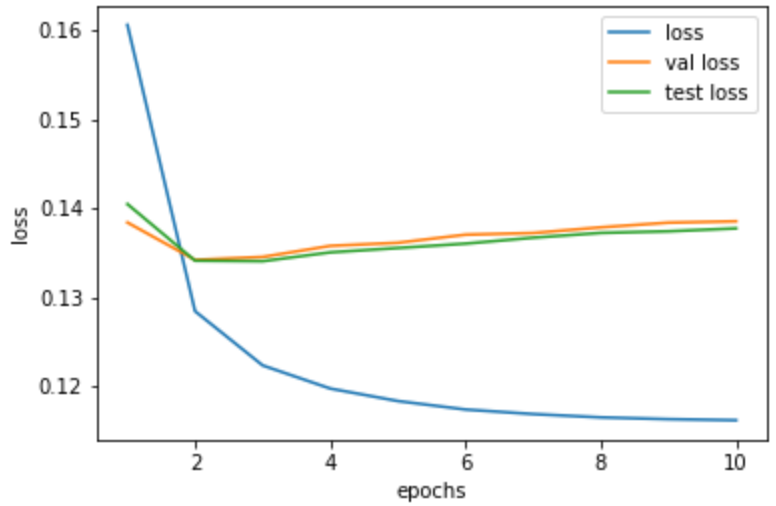


- We can extract the embedding:

Comments on relu Neural Network course rating prediction based recommender system

- different embedding sizes are tested. It turned out that rather small embedding sizes performed best (size 6 lead to a test RMSE of 0.365 as shown in plots below)
- predictions between 0 and 1.6 (training between 0 and 1) where only very few values are greater than 1 (the model probably extrapolates poorly)
- models with large embedding sizes (like 16) tend to overfit, "small" models perform well





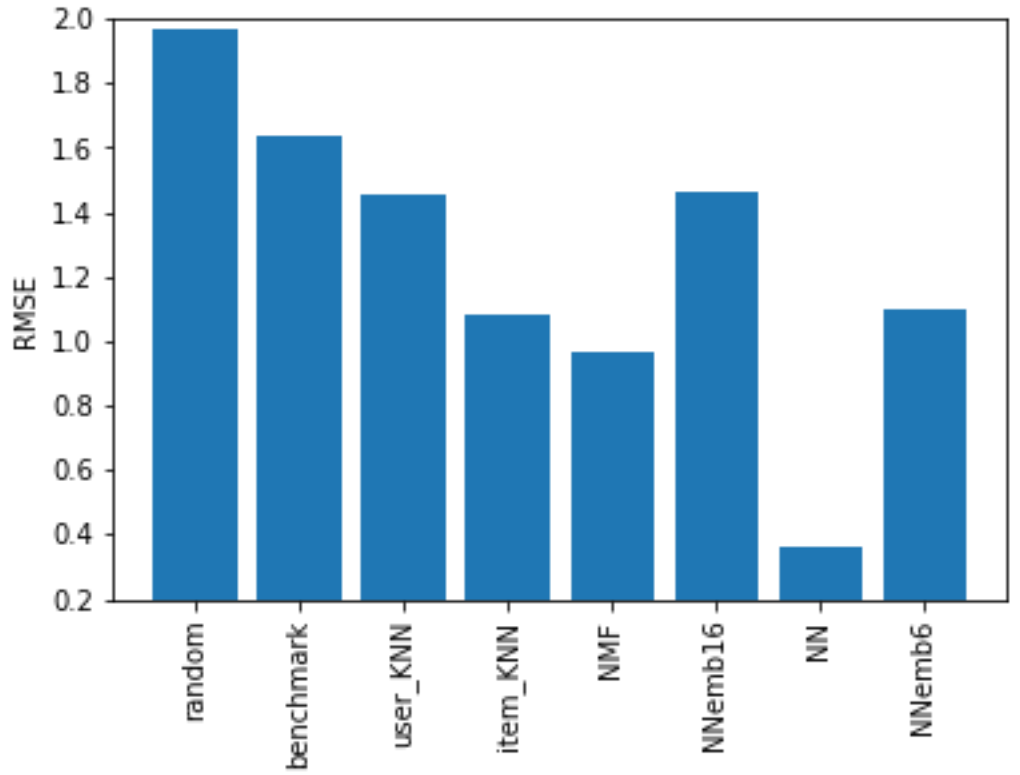
Comments on Neural Network with retrained Embedding based recommender system

We reevaluated the NN embedding approach with a 6-dimensional embedding extracted from the ReLu NN

- Ridge regression performs best with 1.1 RMSE (alpha=0.2)
- Lasso and ElasticNet are also tested but worse
- The results are not satisfying, maybe the embedding has too many dimensions (16)
- The regression model output y range is between -1.6 and 3.6, the training data y range between 0 and 3
- The model performs better but still not satisfying

Compare the performance of collaborative-filtering models

- The relu NN model performs clearly the best
- The NN embedded regularization model performs better with less dimensions
- NMF is the best-performing non-NN solution



Discussion on feedback metrics

The scoring distinguishes between enrolled (i.e. not finished yet) and completed courses and rates completed courses higher than enrolled ones

- There are reasons to rate enrolled courses higher than completed ones:
  - Enrolled courses might be courses that the user is working on and interested in right now. Maybe we should weight them higher than already completed courses
  - This might allow addressing advanced users which might currently be lost in the multitude of beginners
- Or to treat them equally:
  - This should be easier to handle for our models
  - The number of completed courses is much higher than number of enrolled ones
  - Enrolled courses might be completed courses in the future. We showed that most users only enroll one course at a time. Why treat them different?

Conclusions

- Content based recommender systems:
  - Kmeans clustering makes good suggestions
- Collaborative filtering recommender systems:
  - NMN is the best performing non-NN model
  - ReLu NN performs best and is RECOMMENDED
- A combined approach might be promising, for example using Full NN to predict ratings of users from a certain cluster. Then the strengths of both approaches can be combined.
- The feedback metrics should be reevaluated

Outlook

- Train NN model on the full dataset
- Reevaluate the dimensionality of the embedding to improve the regularization model
- Test different rating approaches
- check if advanced users get reasonable recommendations
- Investigate a combined approach

Appendix

You find the github repo here:

https://github.com/maxfleck/coursera_stuff

The repo contains every piece of code used to produce the results shown here

On the next slide you find my workplan. I hope I covered everything...

WORKPLAN:

Uploaded your completed presentation in PDF format (2 pts) (!!!!!!!!!!!!!)
Completed the required Introduction slide (4 pt) (DONE)
Completed the required Exploratory Data Analysis slides (8 pts) (DONE)
Completed the required content-based recommender system using user profile and course genres slides (6 pts) (DONE)
Completed the required content-based recommender system using course similarity slides (6 pts) (DONE)
Completed the required content-based recommender system using user profile clustering slides (6 pts) (DONE)
Completed the required KNN-based collaborative filtering slide (6 pts) (DONE)
Completed the required NMF-based collaborative filtering slide (6 pts) (DONE)
Completed the required neural network embedding based collaborative filtering slide (6 pts) (DONE)
Completed the required collaborative filtering algorithms evaluation slides (6 pts) (DONE)
Completed the required Conclusion slide (6 pts) (DONE)
Applied your creativity to improve the presentation beyond the template (4 pts) (DONE)
- Added a few slides to go into a few points in more detail
- Discussed the rating metrics
- Added the full NN model results
- Added benchmarks
- Commented on the dataset

Displayed any innovative insights (4 pts) (DONE)
- Flaws in the provided dataset
- Rating metrics discussion
- Combined approach