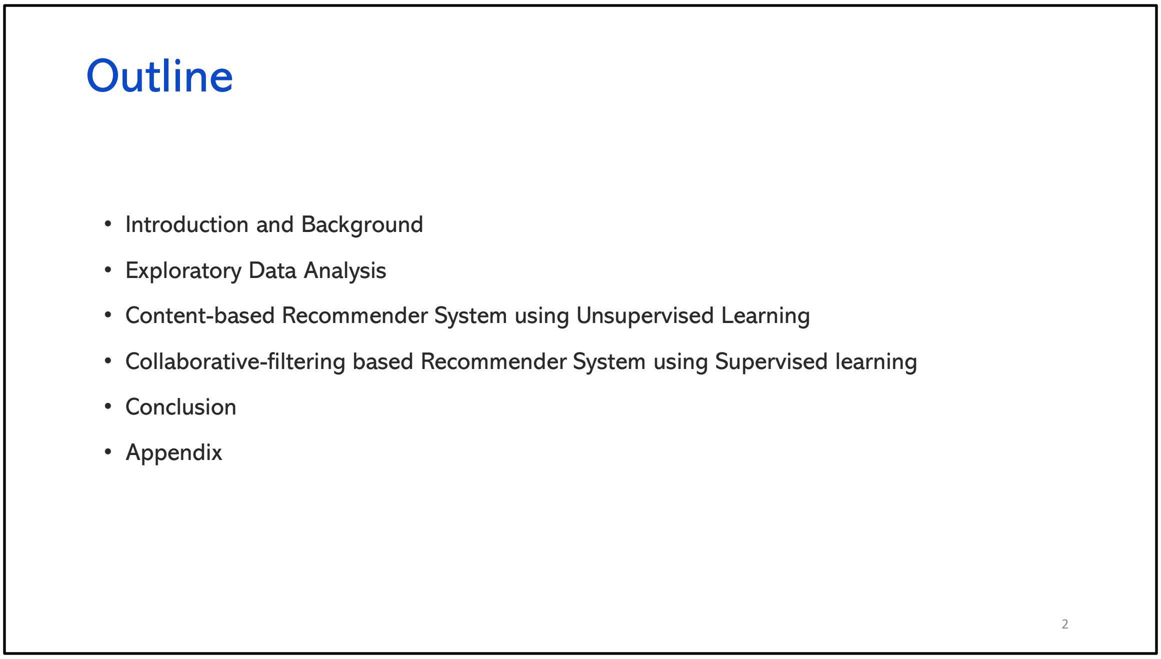
Build a Personalized Online Course Recommender System with Machine Learning

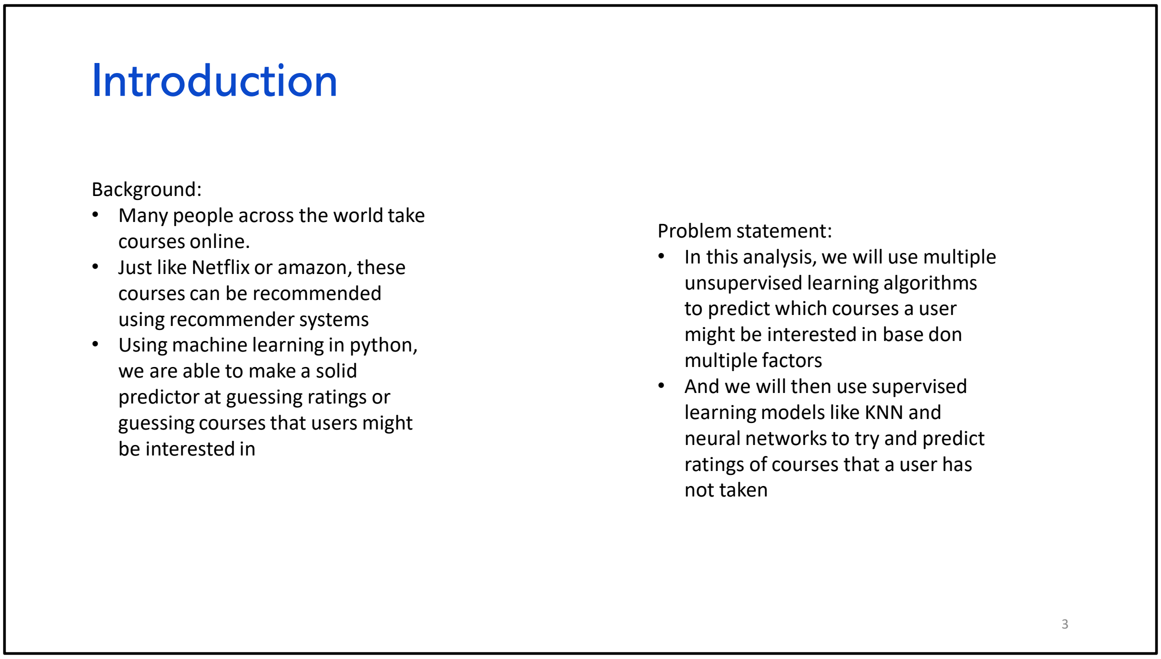
IBM
DW0101EN - v1.2
Introduction to Machine Learning with Sound
Starts: Any time, Self-paced
Course ★ 4.2/5
DeepLearning.TV
ML0115EN - v1.0
Deep Learning Fundamentals
Starts: Any time, Self-paced
Course ★ No ratings yet
IBM
BD0141EN - v2016.0
Accessing Hadoop Data Using Hive
Starts: Any time, Self-paced
Course ★ 4.5/5
Big Data University
BD0115EN - v2016.0
MapReduce and YARN
Starts: Any time, Self-paced
Course ★ No ratings yet
ply end to end securit
a cloud application
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

Shreyas
Dandavate
11/30/2024



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

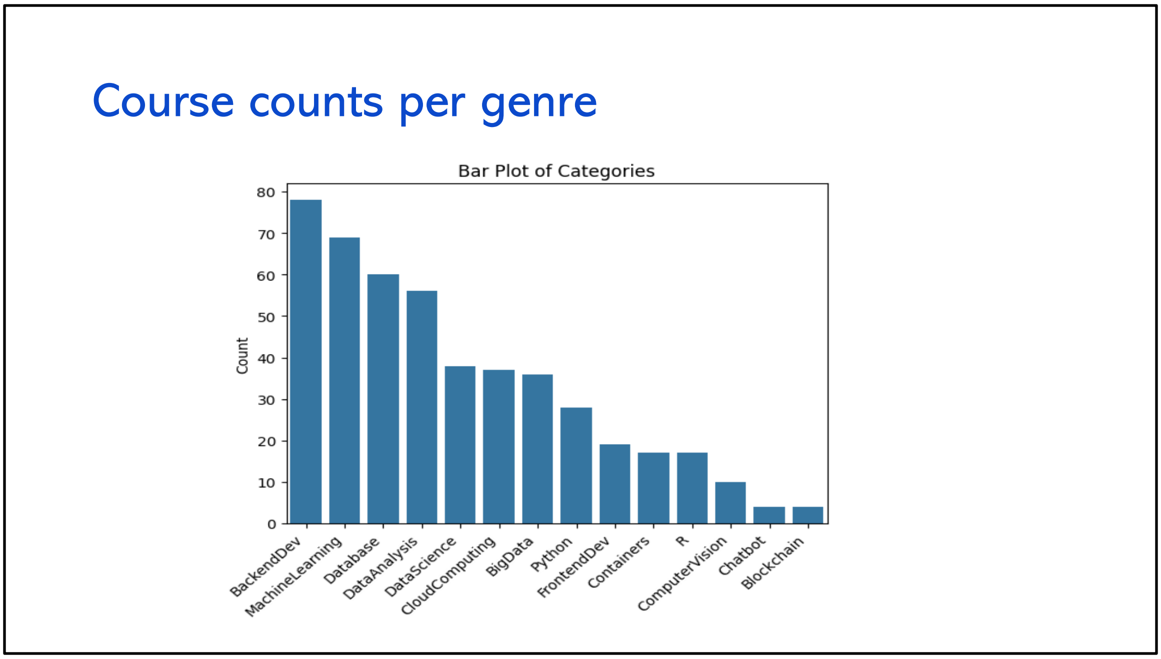
Background:

- Many people across the world take courses online.
- Just like Netflix or amazon, these courses can be recommended using recommender systems
- Using machine learning in python, we are able to make a solid predictor at guessing ratings or guessing courses that users might be interested in

Problem statement:

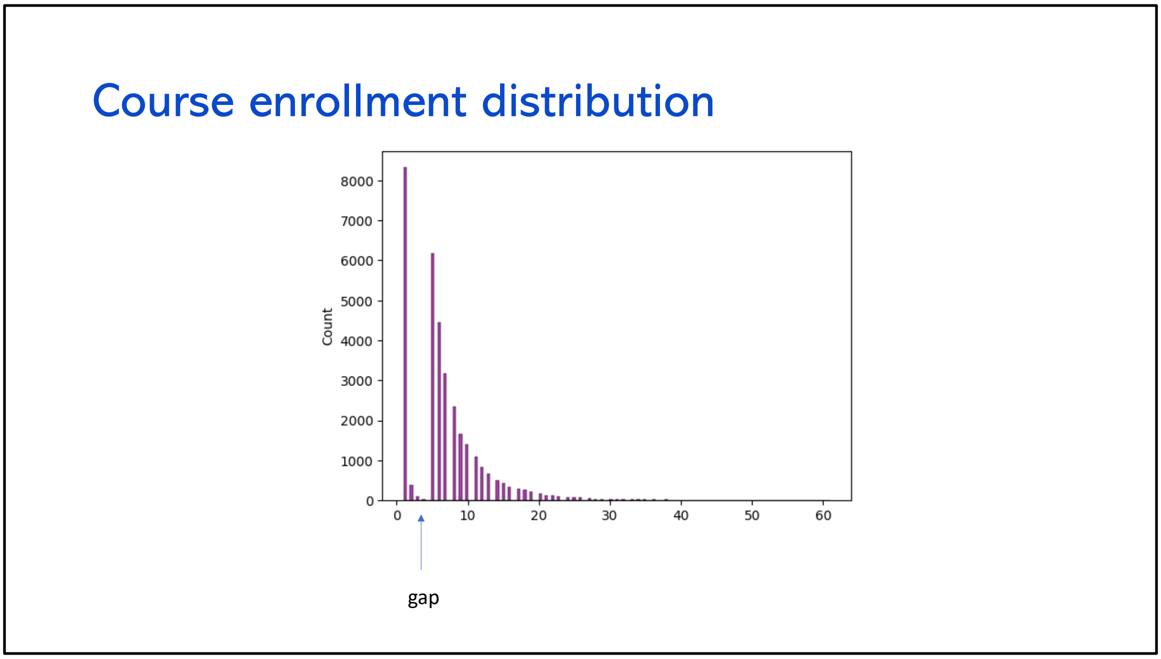
- In this analysis, we will use multiple unsupervised learning algorithms to predict which courses a user might be interested in base don multiple factors
- And we will then use supervised learning models like KNN and neural networks to try and predict ratings of courses that a user has not taken

Exploratory Data Analysis



Course counts per genre

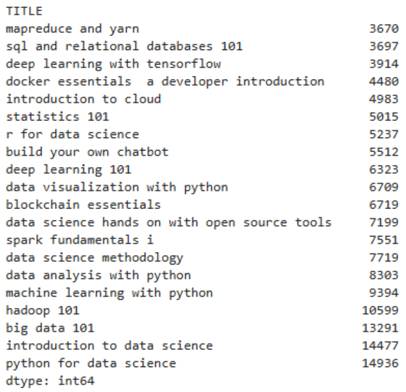
From this, we see that Back end Developing is number 1 with almost 80 course names its in, followed my machine learning and database. On the other end, chatbot and blockchain each have only 4 occurrences.



Course enrollment distribution

There is an evident left skew here with a big gap at around 2-4. This means that 1 enrollment was very common, but 2-4 were not, but then 5 and onward were back to having a fair amount of people. Almost all people either take 1 course, or many, it seems.

20 most popular courses



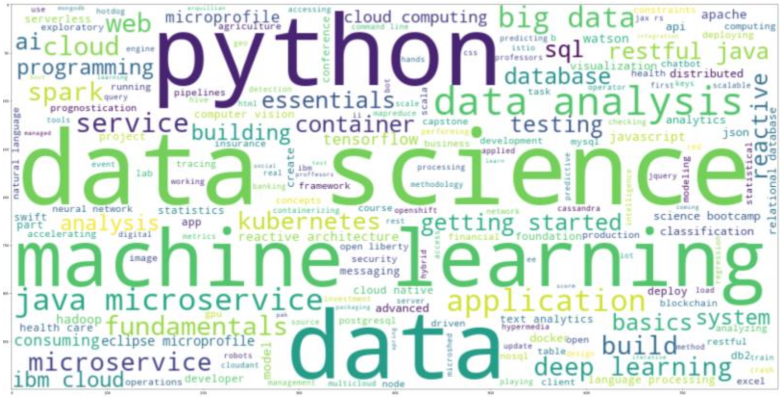
```
TITLE
mapreduce and yarn                                  3670
sql and relational databases 101                    3697
deep learning with tensorflow                       3914
docker essentials  a developer introduction         4480
introduction to cloud                               4983
statistics 101                                      5015
r for data science                                  5237
build your own chatbot                              5512
deep learning 101                                   6323
data visualization with python                      6709
blockchain essentials                               6719
data science hands on with open source tools        7199
spark fundamentals i                                7551
data science methodology                            7719
data analysis with python                           8303
machine learning with python                        9394
hadoop 101                                         10599
big data 101                                       13291
introduction to data science                       14477
python for data science                            14936
dtype: int64
```

20th MOST ENROLLED

MOST ENROLLED

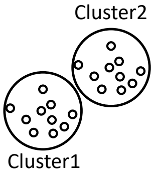
Here, we see python for data science is the most taken course. This would make sense, because it sounds like it is a very introductory course to data analysis, a very popular field.

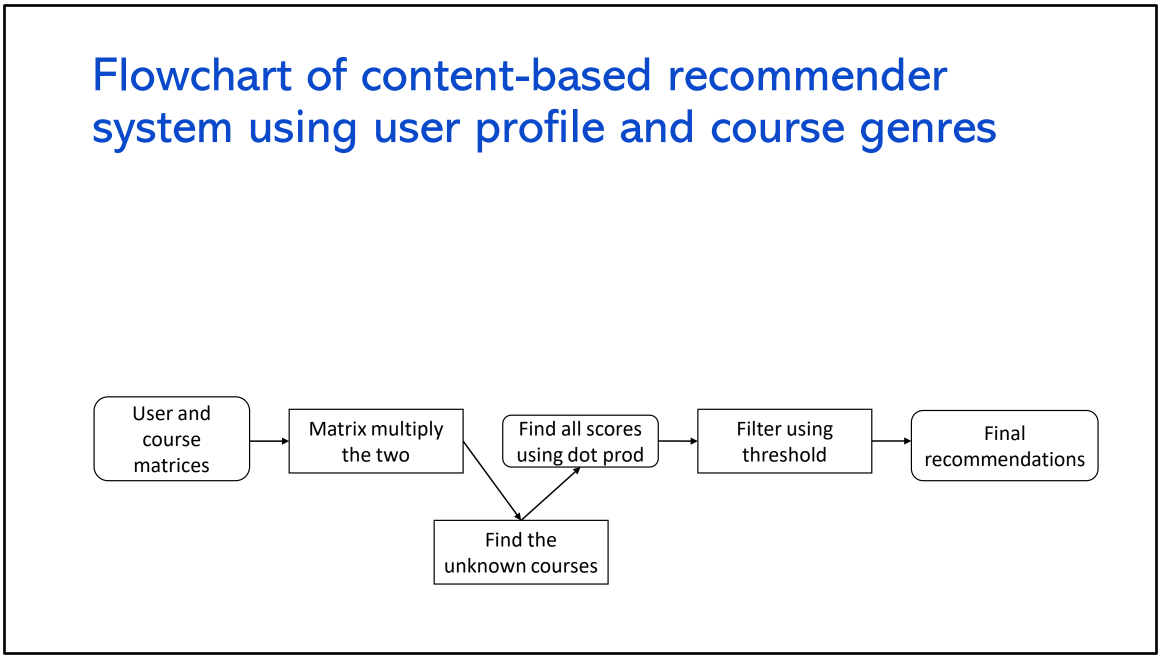
Word cloud of course titles



This is the world cloud of most common phrases in all the course list. I was kind of confused for this part because I did not have to code anything. The code for this word cloud was already given, so I am just pasting the image here.

Content-based Recommender System using Unsupervised Learning





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

User and course matrices

Matrix multiply the two

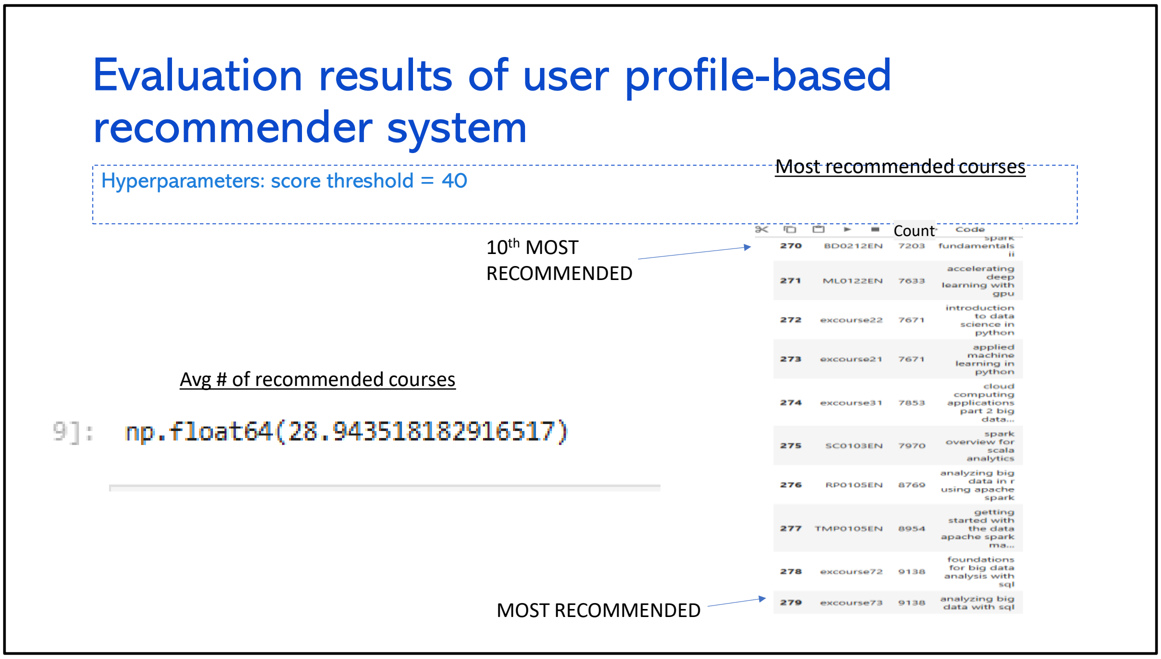
Find the unknown courses

Find all scores using dot prod

Filter using threshold

Final recommendations

After we multiply the matrices for user and the course, we need to create a matrix for each user’s unknown courses. This will allow us to recommend courses only that the user ha snot taken. After we create this matrix, we need to find all scores for each user’s unknown courses from the matrix. We find this score by dotting the unknown course matrix’s vector for the user with the test user's vector. Once we have a score for each unknown course, we filter them out with threshold and recommend all courses with a score above the threshold.



Evaluation results of user profile-based recommender system

Hyperparameters: score threshold = 40

Most recommended courses

10th MOST RECOMMENDED

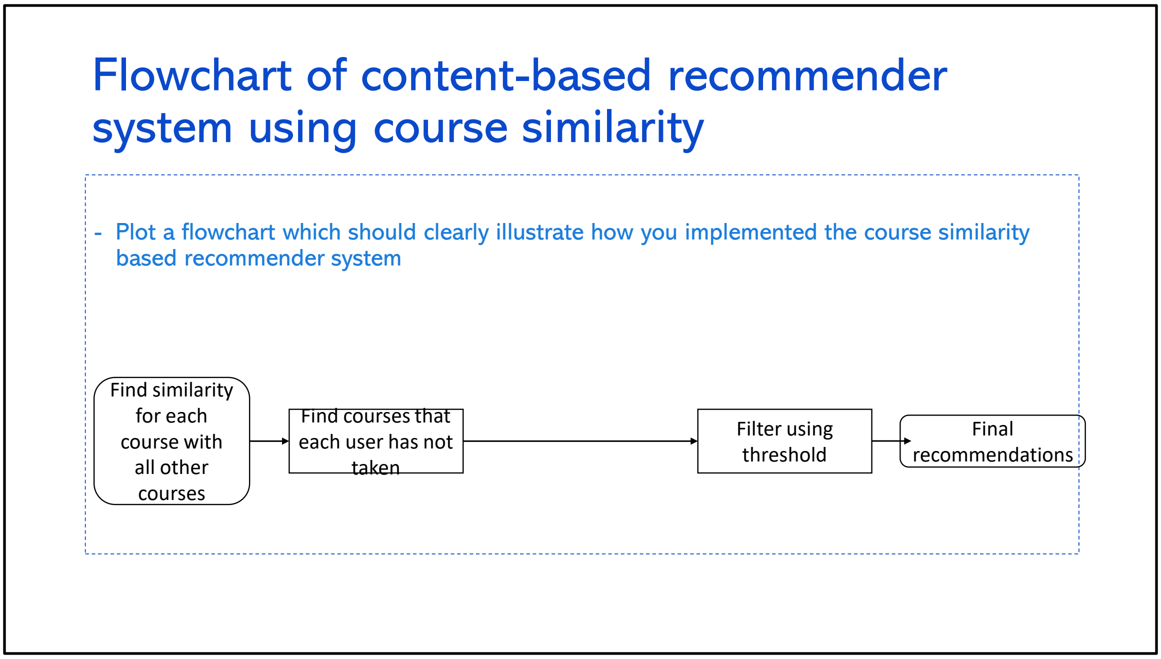
MOST RECOMMENDED

		Count	Code
270	BD0212EN	7203	spark fundamentals ii
271	ML0122EN	7633	accelerating deep learning with gpu
272	excourse22	7671	introduction to data science in python
273	excourse21	7671	applied machine learning in python
274	excourse31	7853	cloud computing applications part 2 big data...
275	SC0103EN	7970	spark overview for scala analytics
276	RP0105EN	8769	analyzing big data in r using apache spark
277	TMP0105EN	8954	getting started with the data apache spark ma...
278	excourse72	9138	foundations for big data analysis with sql
279	excourse73	9138	analyzing big data with sql

Avg # of recommended courses

```
9]:  np.float64(28.943518182916517)
```

1. Average number of unseen courses: 28.94. This is quite a lot. This is good because it means we are recommending new stuff
2. Top 10 courses: At the number 1 spot (at the bottom), we see analyzing big data with SQL with 9138 reccomendations. In 10th place, we see spark fundamentals ii with 7203 recommendations



Flowchart of content-based recommender system using course similarity

- Plot a flowchart which should clearly illustrate how you implemented the course similarity based recommender system

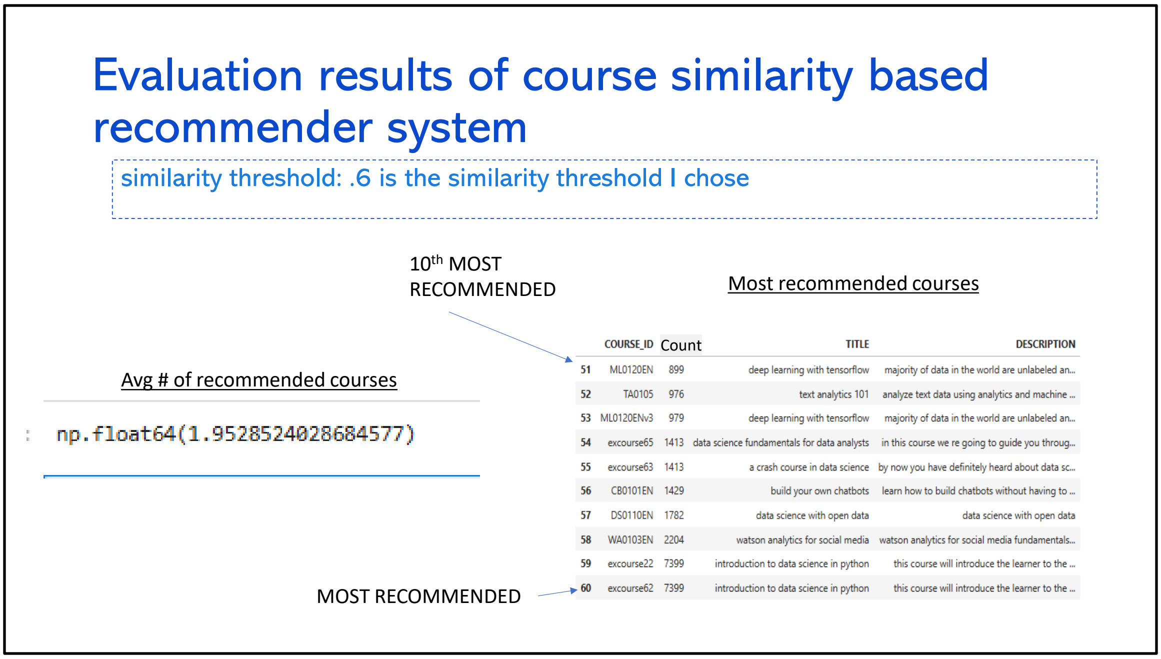
Find similarity for each course with all other courses

Find courses that each user has not taken

Filter using threshold

Final recommendations

Firstly we need to establish each course’s similarity with al other courses. This value ranges from 0 to 1. Similar to the recommender using content and course genres, we again need to find the unknown courses. This is much simpler than the previous recommender, because we already have scores, which is just the similarity value between courses. We will filter these courses by this similarity measure and recommend all those that are above the threshold.



Evaluation results of course similarity based recommender system

similarity threshold: .6 is the similarity threshold I chose

10th MOST RECOMMENDED

Most recommended courses

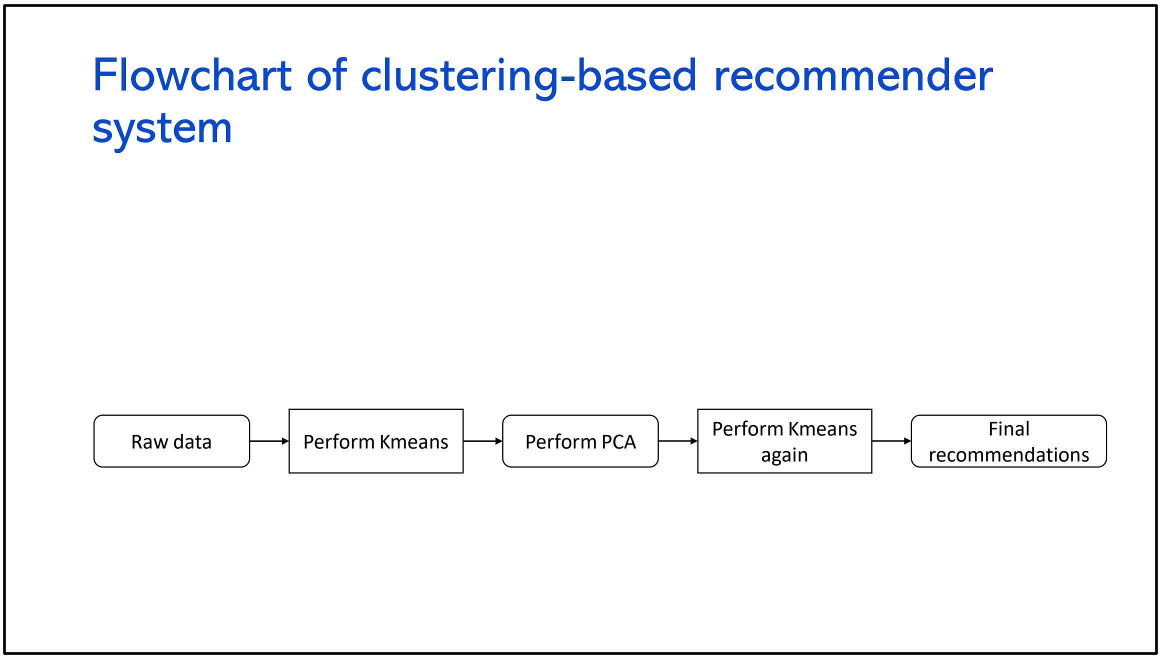
Avg # of recommended courses

```
: np.float64(1.9528524028684577)
```

	COURSE_ID	Count	TITLE	DESCRIPTION
51	ML0120EN	899	deep learning with tensorflow	majority of data in the world are unlabeled an...
52	TA0105	976	text analytics 101	analyze text data using analytics and machine ...
53	ML0120ENv3	979	deep learning with tensorflow	majority of data in the world are unlabeled an...
54	excourse65	1413	data science fundamentals for data analysts	in this course we re going to guide you throug...
55	excourse63	1413	a crash course in data science	by now you have definitely heard about data sc...
56	CB0101EN	1429	build your own chatbots	learn how to build chatbots without having to ...
57	DS0110EN	1782	data science with open data	data science with open data
58	WA0103EN	2204	watson analytics for social media	watson analytics for social media fundamentals...
59	excourse22	7399	introduction to data science in python	this course will introduce the learner to the ...
60	excourse62	7399	introduction to data science in python	this course will introduce the learner to the ...

MOST RECOMMENDED

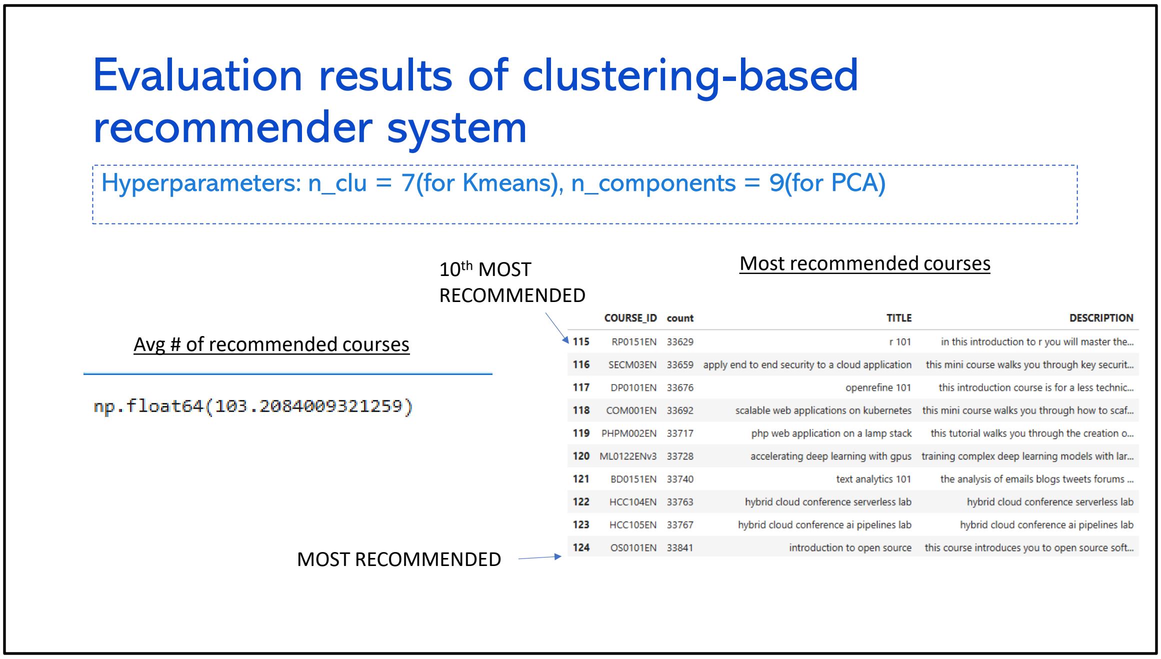
1. For the average number of new courses recommended per person, we see that it is just under two. This means that our algorithm and our threshold is really strict on the similarity we are allowing between recommendations. If we wanted more average reccomendations, we could decrease the threshold to solve this.
2. At number 1, we see introduction to data science in python with 7399 recommendations. AT 10th place is deep learning with TensorFlow at 899 recommendations



Flowchart of clustering-based recommender system

Raw data → Perform Kmeans → Perform PCA → Perform Kmeans again → Final recommendations

First, we use the Kmeans clustering algorithm on the user profile feature vectors in order to generate clusters for each user vector. Then we apply PCA on this in order to reduce the dimensionality. Then, Kmeans again. Finally, we have our recommendations.



Evaluation results of clustering-based recommender system

Hyperparameters: n_clu = 7(for Kmeans), n_components = 9(for PCA)

Avg # of recommended courses

```
np.float64(103.2084009321259)
```

10th MOST RECOMMENDED

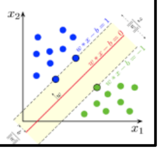
MOST RECOMMENDED

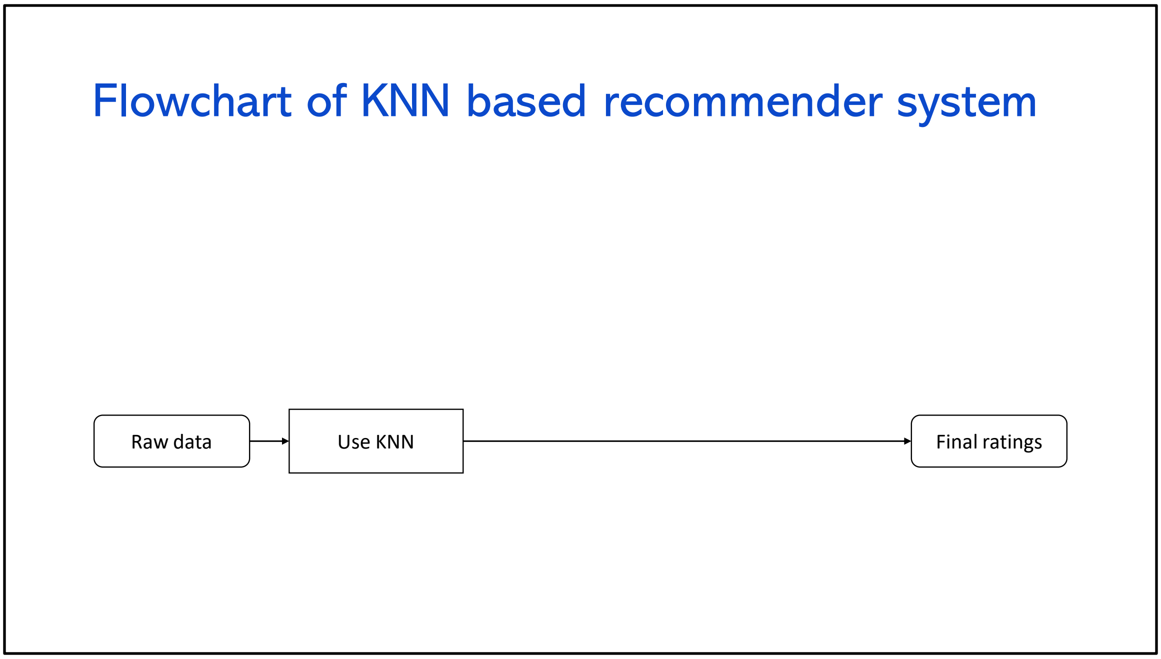
Most recommended courses

	COURSE_ID	count	TITLE	DESCRIPTION
115	RP0151EN	33629	r 101	in this introduction to r you will master the...
116	SECM03EN	33659	apply end to end security to a cloud application	this mini course walks you through key securit...
117	DP0101EN	33676	openrefine 101	this introduction course is for a less technic...
118	COM001EN	33692	scalable web applications on kubernetes	this mini course walks you through how to scaf...
119	PHPM002EN	33717	php web application on a lamp stack	this tutorial walks you through the creation o...
120	ML0122ENv3	33728	accelerating deep learning with gpus	training complex deep learning models with lar...
121	BD0151EN	33740	text analytics 101	the analysis of emails blogs tweets forums ...
122	HCC104EN	33763	hybrid cloud conference serverless lab	hybrid cloud conference serverless lab
123	HCC105EN	33767	hybrid cloud conference ai pipelines lab	hybrid cloud conference ai pipelines lab
124	OS0101EN	33841	introduction to open source	this course introduces you to open source soft...

1. Holy cow that’s a lot of courses recommended per user. This could be because I only used 7 clusters, meaning there are a ton of courses within each cluster. If we wanted less reccomendations, we could increase the amount of clusters
2. At number 1, we see introduction to open source with 33,841 recommendations. This is kind of weird because this is getting recommended to almost everybody. Im pretty sure I did the code right though, so maybe its just really similar to a lot of courses.

Collaborative-filtering Recommender System using Supervised Learning





Flowchart of KNN based recommender system

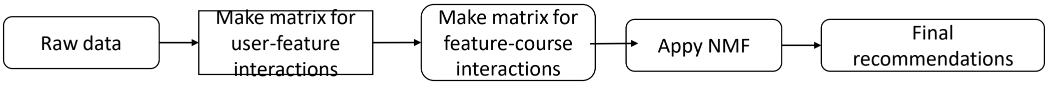
Raw data

Use KNN

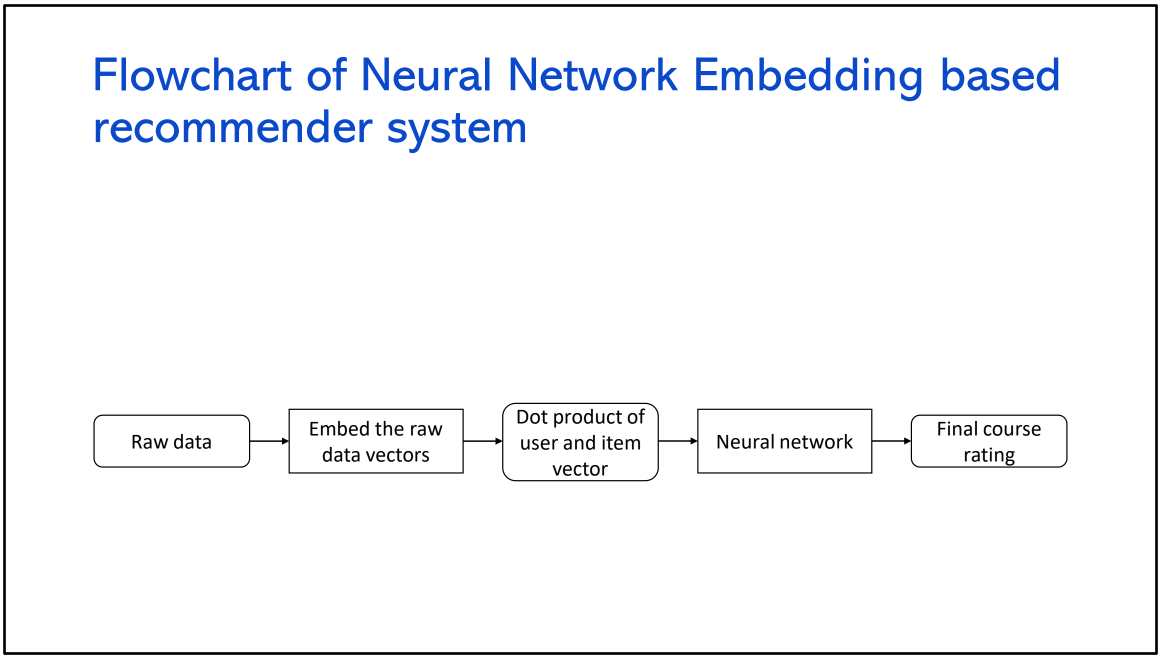
Final ratings

With KNN, there isn’t really much to do on our part. We could convert the data into a sparse matrix, but we don’t have to. The KNN library just takes care of everything for us.

Flowchart of NMF based recommender system



Raw data → Make matrix for user-feature interactions → Make matrix for feature-course interactions → Appy NMF → Final recommendations



Flowchart of Neural Network Embedding based recommender system

Raw data

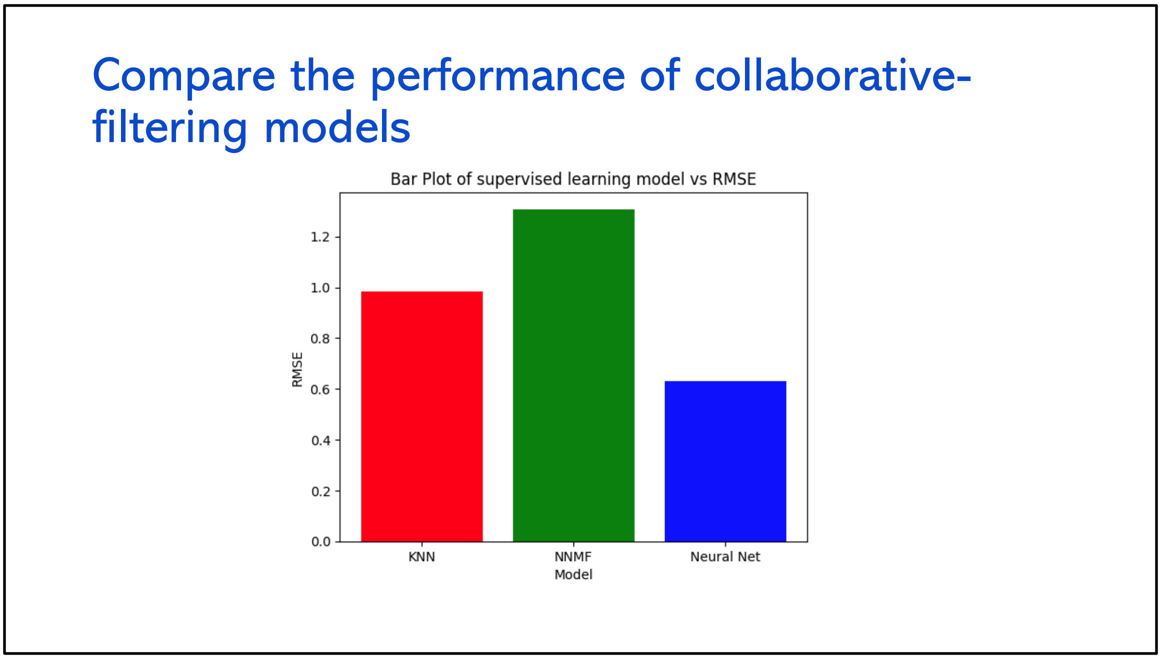
Embed the raw data vectors

Dot product of user and item vector

Neural network

Final course rating

With our raw data, we first need to embed the vectors into the embedded vectors, which will make the neural network compute more accurately. Once we have 2 embedded vectors, one for the user and one for the item, we then dot product them and feed this vector into the activation function for the neural network. I’m pretty sure that Relu was the function that was set in the notebook. Then, e fit and evaluate our model on our test data, and we have a matrix of course ratings for each user.

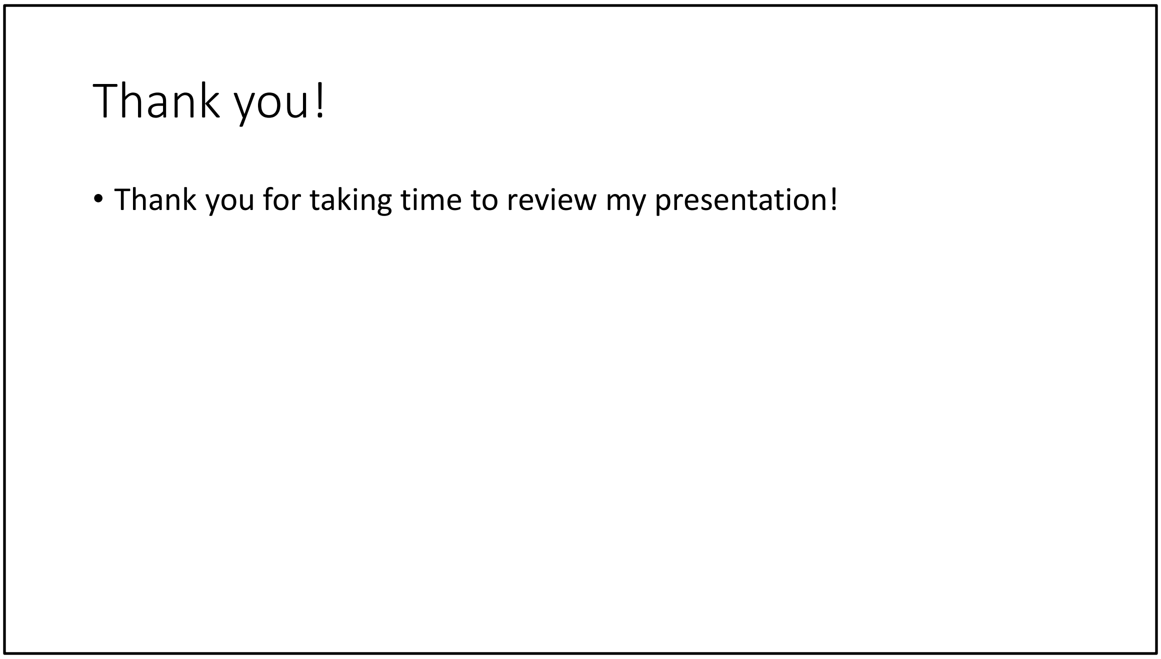


Compare the performance of collaborative-filtering models

As one might expect, the neural network has the lowest RMSE out of all of them. However, I would like to point out that the KNN model I used had the test size as .9, because anything less would cause my computer to crash. So, if I had more memory, maybe the KNN would do better. But for the resources I have, the neural network was the best by quite a bit.

Conclusions

- For predicting user ratings of an unknown course, the neural network has the lowest RMSE by a fair amount.
- This would lead me to recommend the neural network model over the NMF or KNN model to businesses.
- For course recommendations, I noticed that there were quite a range of values in recommended courses. We saw just under 2 all the way to over 100.
- This leads me to recommend the user profile-based recommender system, because its average courses recommended is about 28, which is not too little but not too big
- This means that a customer will not be overwhelmed by a heap of possible courses, but also exposes them to enough where they might find something interesting.



Thank you!

- Thank you for taking time to review my presentation!

Thanks! Have a nice day!