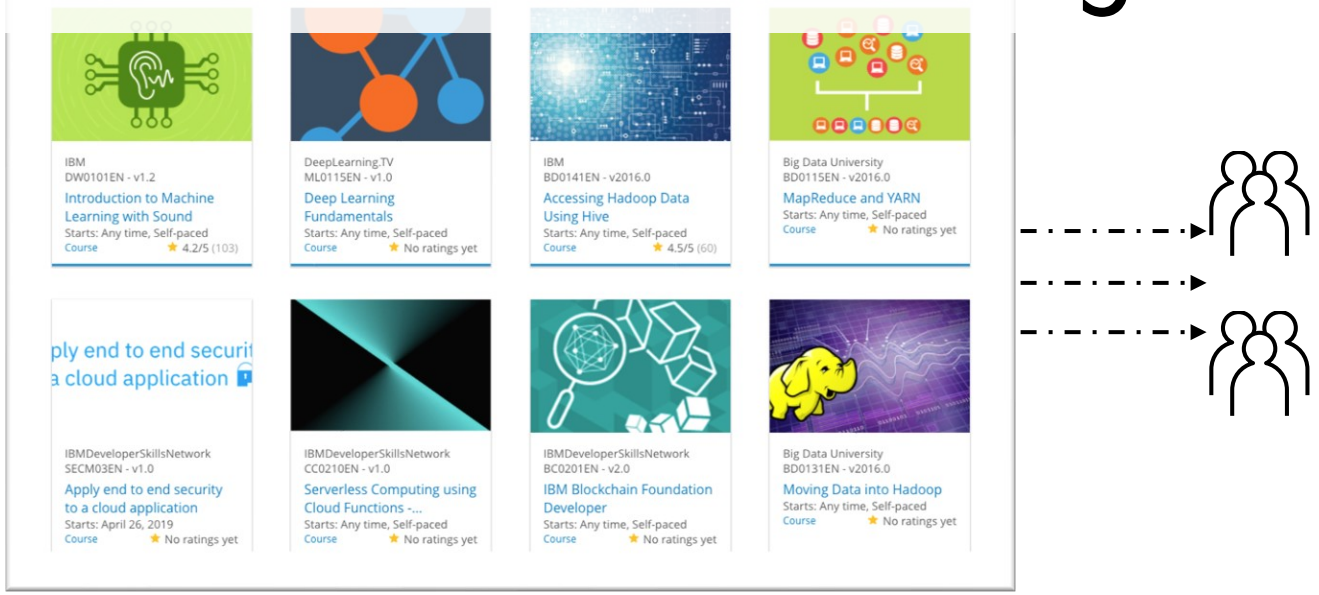
Build a Personalized Online Course Recommender System with Machine Learning

Kay Sun

September 28, 2022



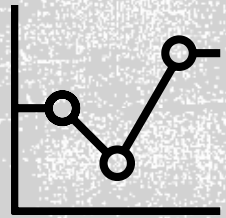
Outline

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

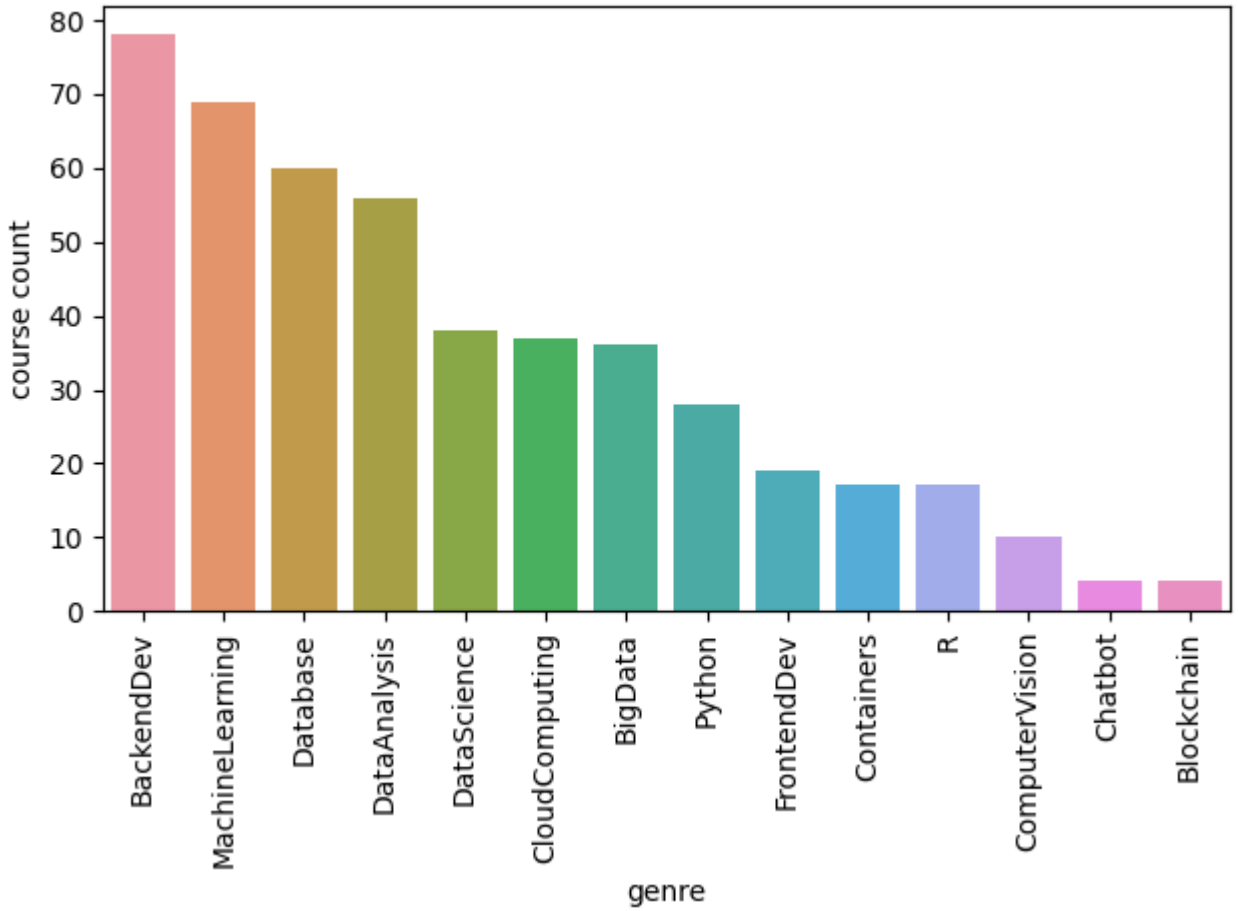
Introduction

- Goal of project is to build a personalized online courses recommender system for students based on different recommendation models available.
- Data available includes
  - Courses with one-hot encoding of their genre
  - User ratings of courses
  - User profile with one-hot encoding of their genre
- Models include
  - Content based using
    - User profiles and course genre
    - Course similarity
    - Clustering
  - Collaborative filtering using
    - KNN
    - NMF
    - NN
- Evaluate the performances of the different models in predicting courses for students.

EXPLORATORY DATA ANALYSIS



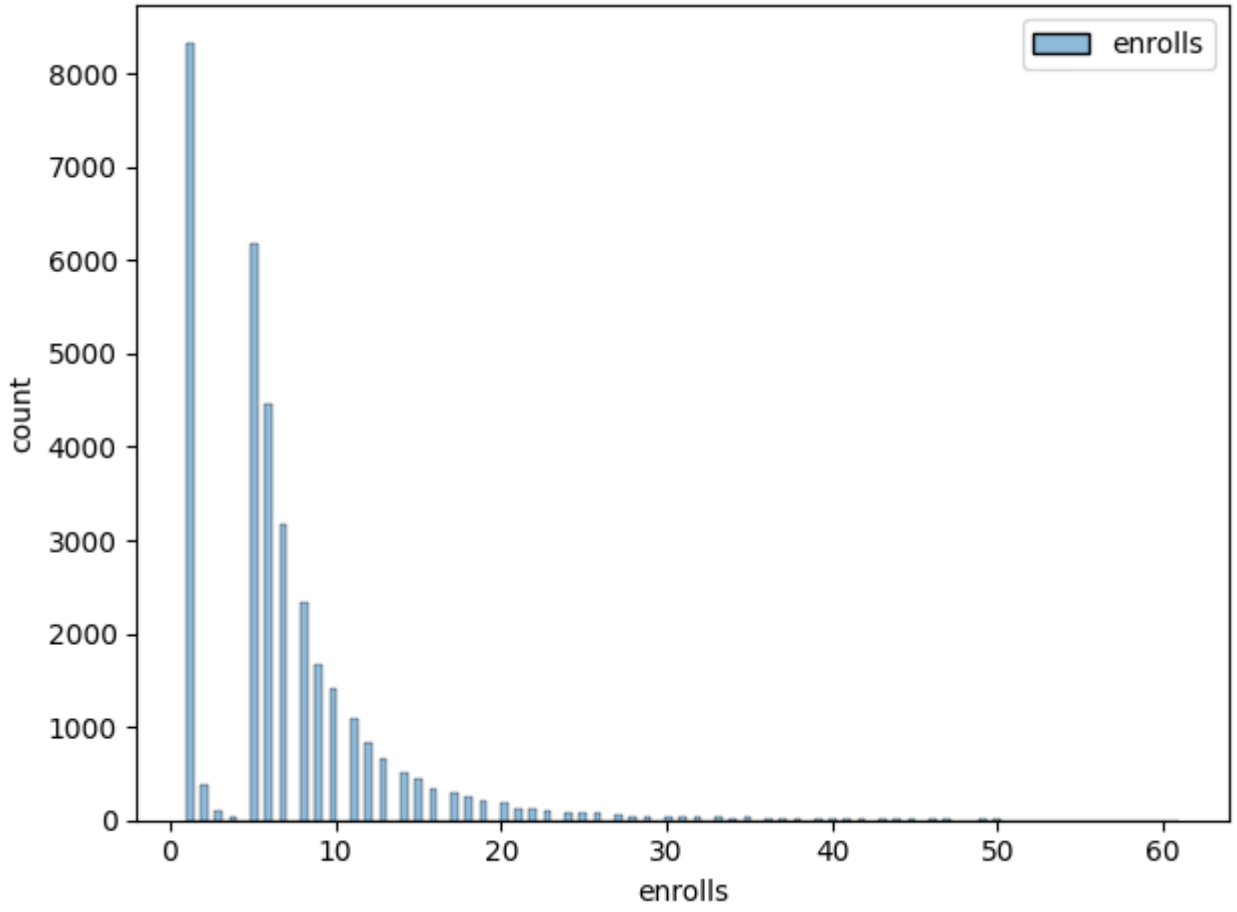
COURSE COUNTS PER GENRE



- Top 3 most popular course genres are Backend Dev, Machine Learning and Datase.
- 3 Least popular course genres are Computer Vision, Chatbot and Blockchain.

COURSE ENROLLMENT DISTRIBUTION

- Histogram of course enrollments.
- Most students just enroll to 1 course to try out the course content.
- Sharp drop for 2 to 4 enrolled courses.
- But after 5 enrolled courses, distribution resumes a normal distribution.
- Seems to indicate most students will try out a course. Those who like the courses offered will enroll in at least 5 more.
- Those who don't like the courses offered will mostly abandon. Very few will try out another 2 to 4 more.

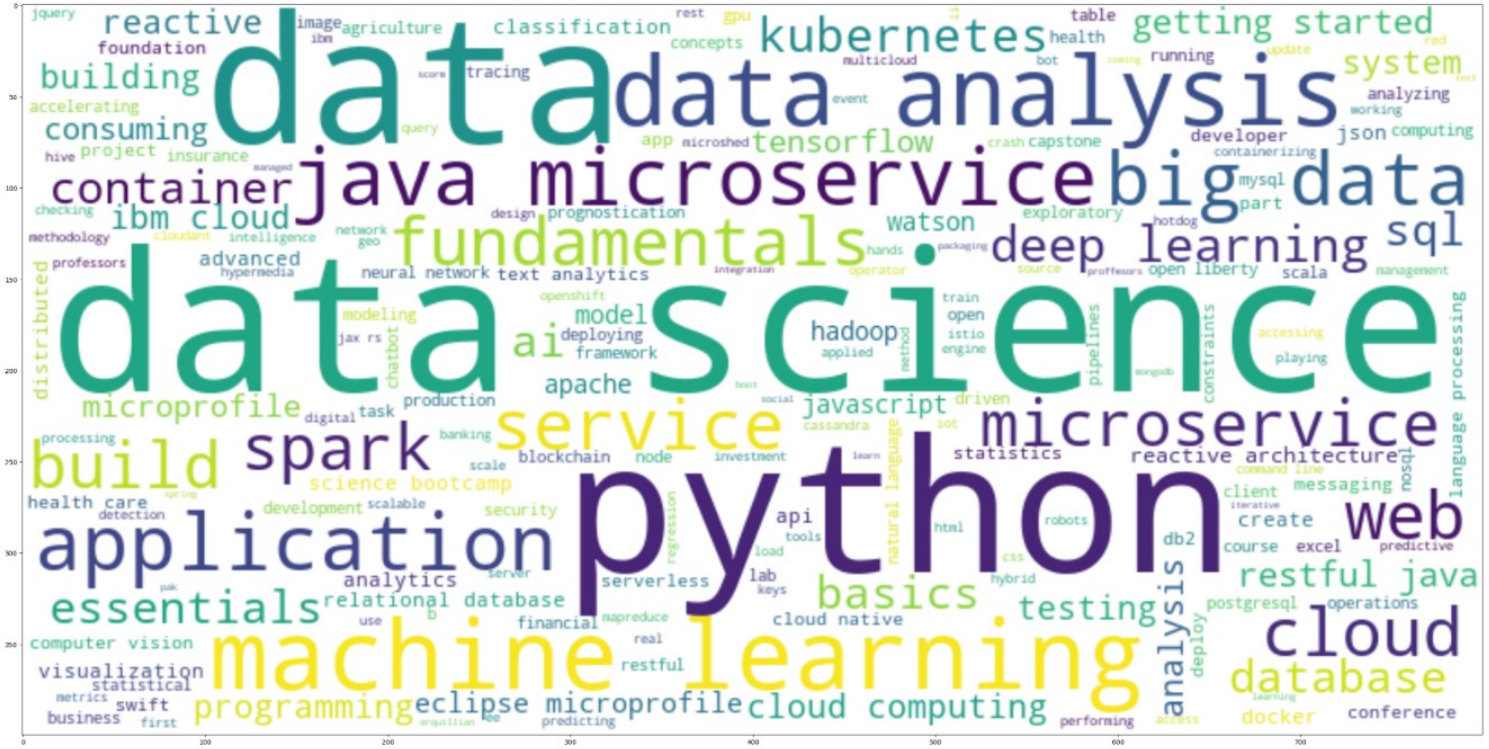


20 MOST POPULAR COURSES

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

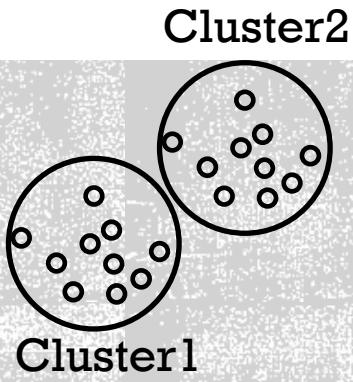
- Top 3 most popular courses are Python for Data Science, Introduction to Data Science and Big Data 101.
- Top 8 most popular courses are in field of data science.
- 3 Least popular courses are SQL and Relational Databases 101, MapReduce and Yarn, and Data Privacy Fundamentals.
- Enrollments for top 3 courses are at least 4 times that out the 3 least popular ones.

WORD CLOUD OF COURSE TITLES



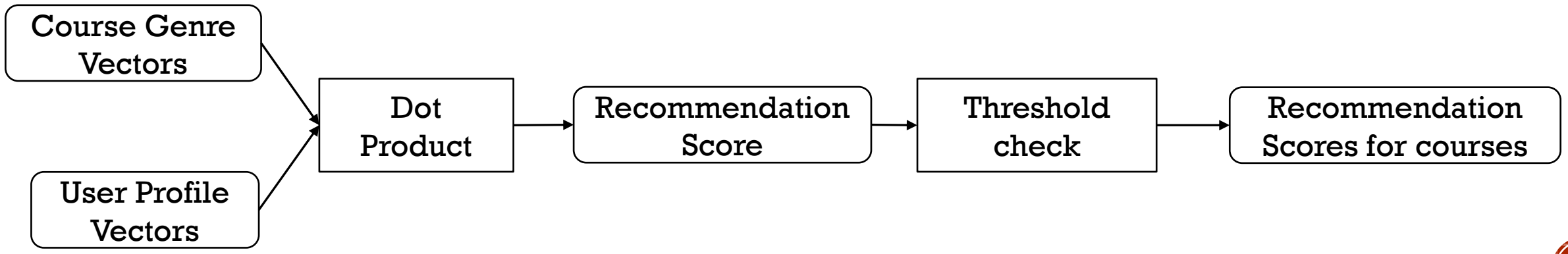
- Most popular courses based on course titles relates to python, data science, data analysis and machine learning.

CONTENT-BASED RECOMMENDER SYSTEM USING UNSUPERVISED LEARNING



FLOWCHART OF CONTENT-BASED RECOMMENDER SYSTEM USING USER PROFILE AND COURSE GENRES

- Course genre vectors – one hot encoding of genres for each course.
- User profile vectors – one hot encoding of user's ratings and interests for each genre.
- Dot Product of 2 vectors to produce a recommendation score for each user.
- Apply threshold on the score such that only those above threshold will be recommended. Threshold can be absolute value or relative based on the score range.
- Final output is the recommendation scores for courses for the user.



EVALUATION RESULTS OF USER PROFILE-BASED RECOMMENDER SYSTEM

Recommendation score is normalized by its maximum so its range is now from 0 to 1.

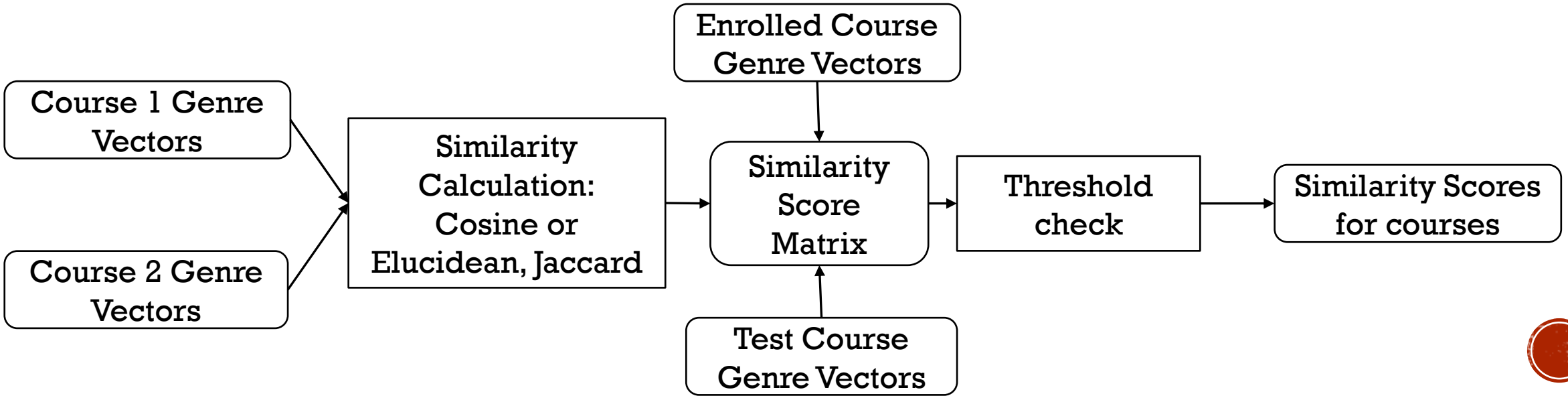
On average, how many new courses have been recommended per test user:
7.789473684210526 5

Top-10 commonly recommended courses across all users

	COURSE_ID	TITLE	COUNT
0	excourse72	foundations for big data analysis with sql	30
1	excourse73	analyzing big data with sql	30
2	TMP0105EN	getting started with the data apache spark ma...	29
3	SC0103EN	spark overview for scala analytics	20
4	excourse31	cloud computing applications part 2 big data...	18
5	RP0105EN	analyzing big data in r using apache spark	14
6	excourse05	\r\ndistributed computing with spark sql	12
7	excourse10	database architecture scale and nosql with e...	12
8	excourse42	big data analysis hive spark sql dataframes...	12
9	excourse71	big data essentials hdfs mapreduce and spark...	12

FLOWCHART OF CONTENT-BASED RECOMMENDER SYSTEM USING COURSE SIMILARITY

- Course genre vectors – one hot encoding of genres for each course.
- Similarity calculation of 2 vectors to produce a similarity score between 2 courses.
  - Similarity based on cosines, elucidean, Jaccard.
- Repeat similarity calculation between all pairs of courses to generate a matrix of similarity scores where the indices are the 2 course indices.
- For each enrolled and unselected course, extract similarity score from matrix and apply threshold such that only those above threshold will be recommended. Threshold can be absolute value or relative based on the score range.



EVALUATION RESULTS OF COURSE SIMILARITY BASED RECOMMENDER SYSTEM

Similarity based on cosine, which ranges from 0 to 1.

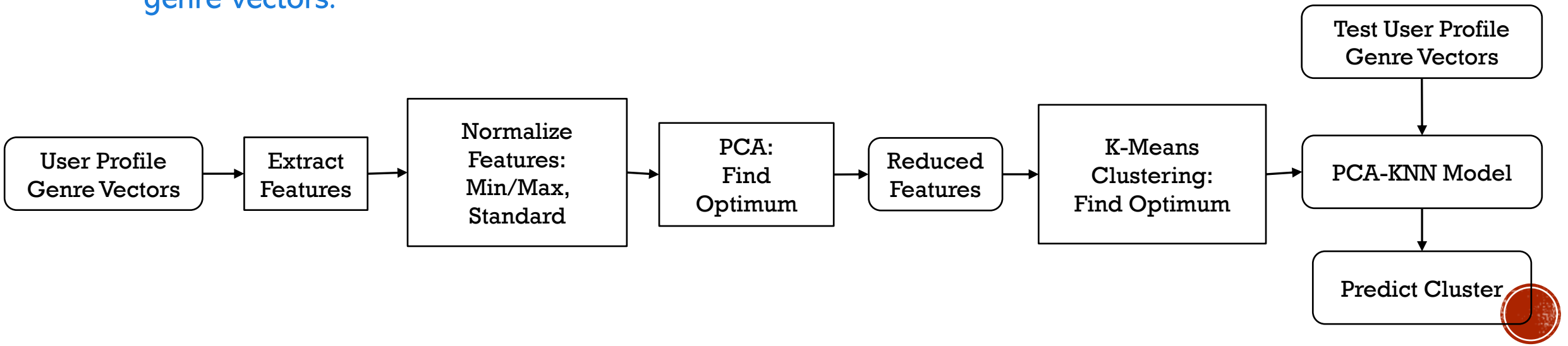
On average, how many new courses have been recommended per test user: 11.573753814852493

Top-10 commonly recommended courses

	COURSE_ID	TITLE	COUNT
0	excourse22	introduction to data science in python	579
1	excourse62	introduction to data science in python	579
2	DS0110EN	data science with open data	562
3	excourse63	a crash course in data science	555
4	excourse65	data science fundamentals for data analysts	555
5	excourse72	foundations for big data analysis with sql	551
6	excourse68	big data modeling and management systems	550
7	excourse67	introduction to big data	539
8	excourse74	fundamentals of big data	539
9	BD0145EN	sql access for hadoop	506

FLOWCHART OF CLUSTERING-BASED RECOMMENDER SYSTEM

- User profile genre vectors – one hot encoding of genres for each course.
- Extract genre or features out and normalize them by either min/max or standard scaler.
- Apply PCA dimension reduction on a range of components and find its optimum. Apply this optimum to get the reduced number of features.
- Apply reduced featured to K-Means clusters on a range of clusters to find its optimum. Apply this optimum to get the PCA-KNN model.
- Use PCA-KNN model to predict which cluster test users belong to according to their user profile genre vectors.



EVALUATION RESULTS OF CLUSTERING-BASED RECOMMENDER SYSTEM

Optimum number of PCA components = 9

Optimum number of K-Means clusters = 11

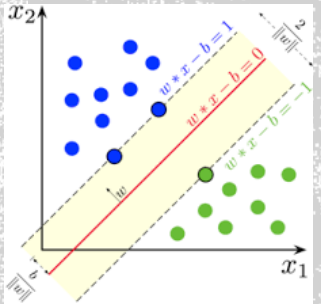
Recommend only courses with more than 100 enrollments.

On average, how many new courses have been recommended per test user: 1.249

Top-10 commonly recommended courses

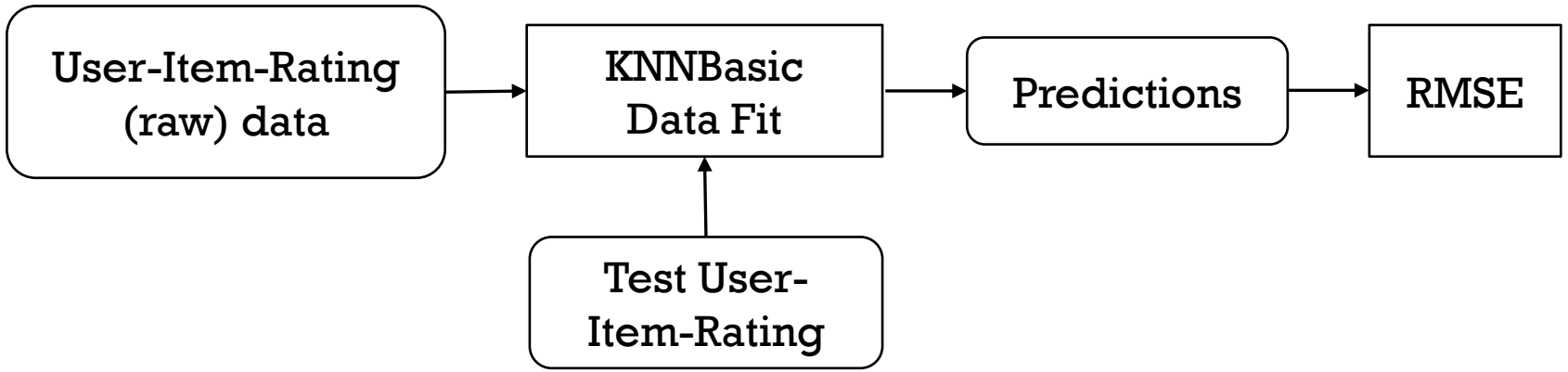
	COURSE_ID	TITLE	RECOMMENDS	COUNT
0	DS0101EN	introduction to data science	DS0101EN	151
1	BD0101EN	big data 101	BD0101EN	103
2	DV0101EN	data visualization with python	DV0101EN	103
3	ML0101ENv3	machine learning with python	ML0101ENv3	93
4	DA0101EN	data analysis with python	DA0101EN	75
5	BD0211EN	spark fundamentals i	BD0211EN	32
6	PY0101EN	python for data science	PY0101EN	18
7	BD0111EN	hadoop 101	BD0111EN	13
8	CB0103EN	build your own chatbot	CB0103EN	4

COLLABORATIVE-FILTERING RECOMMENDER SYSTEM USING SUPERVISED LEARNING



FLOWCHART OF KNN BASED RECOMMENDER SYSTEM

- Input data is user-item-rating interaction matrix.
- Raw data is split into train-test split by percentage, e.g., 25%.
- Surprise scikit learn KNN model is initiated using hyperparameters (similarity measurement = cosine, user_based=True, minimum cluster k = 1). Train dataset is fitted to model.
- Cosine similarity formula $$sim(u,u') = cos(\theta) = \frac{\mathbf{r}_u \cdot \mathbf{r}_{u'}}{\|\mathbf{r}_u\|\|\mathbf{r}_{u'}\|} = \sum_i \frac{r_{ui} r_{u'i}}{\sqrt{\sum_i r_{ui}^2}\sqrt{\sum_i r_{u'i}^2}}$$
- Normalized rating by total users ratings $$\hat{r}_{ui} = \frac{\sum_{u'} sim(u,u') r_{u'i}}{\sum_{u'} |sim(u,u')|}$$
- Predictions are then made on the trained model using the test dataset.
- Root mean square error between the predictions and test dataset where truth is known.



FLOWCHART OF NMF BASED RECOMMENDER SYSTEM

- Input data is user-item-rating interaction matrix.
- Raw data is split into train-test split by percentage, e.g., 30%.
- Surprise scikit learn NMF model is initiated using hyperparameters (n_factors, n_epochs). Train dataset is fitted to model.
- NMF: A = U x I

Lower number of features by reducing into lower dimensional space.

Non-negative Matrix Factorization

User-item interaction matrix: **A** 10000 x 100

	item1	...	item100
user1	...	...	
user2	3.0	3.0	3.0
user3	2.0	2.0	-
user4	3.0	2.0	3.0
user5	2.0	-	-
user6	3.0	-	3.0
...	...	...	

≈

User matrix: **U** 10000 x 16

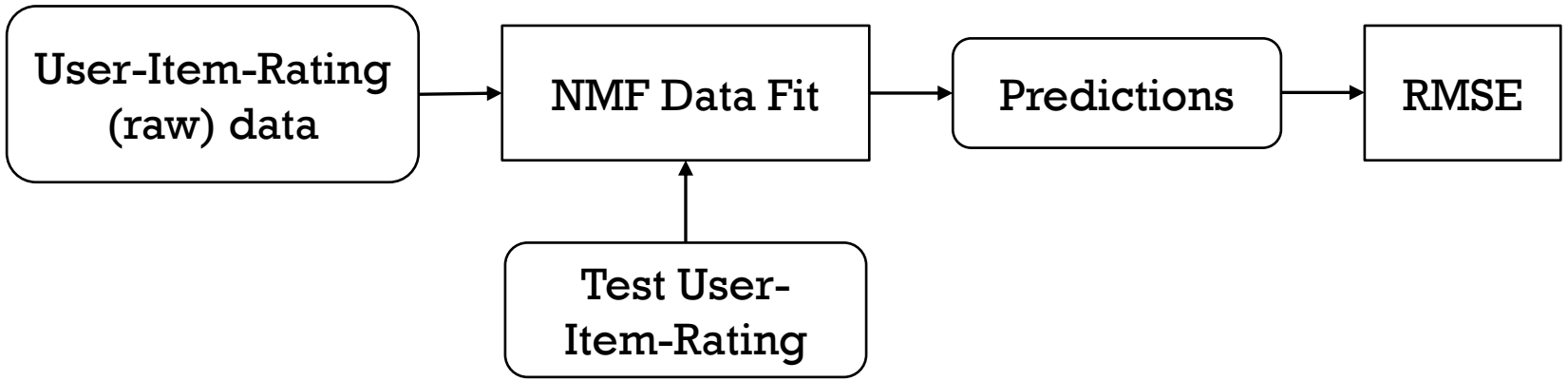
	feature1	...	feature16
user1	...	...	...
user2	...	...	...
user3	...	...	...
user4	...	...	...
...	...	...	...
...	...	...	...
user6	...	...	...

X

Item matrix: **I** 16 x 100

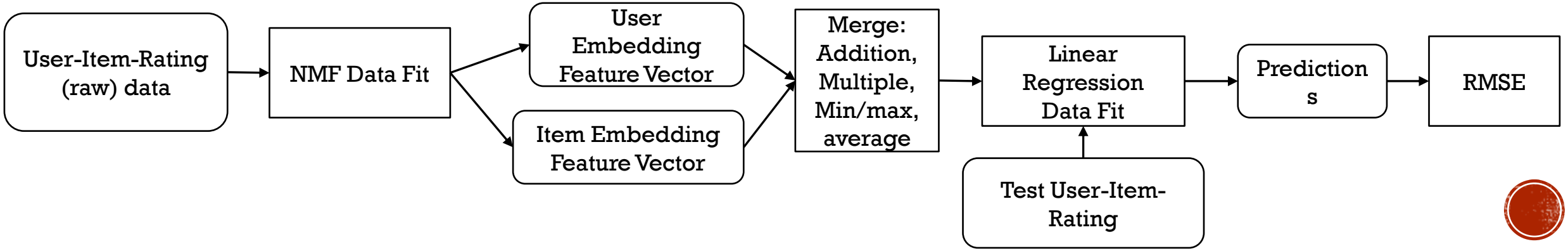
	item1	...	item100
feature1	...	...	...
feature2	...	...	...
...	...	...	...
feature16	...	...	...

- Predictions are then made on the trained model using the test dataset.
- Root mean square error between the predictions and test dataset where truth is known.



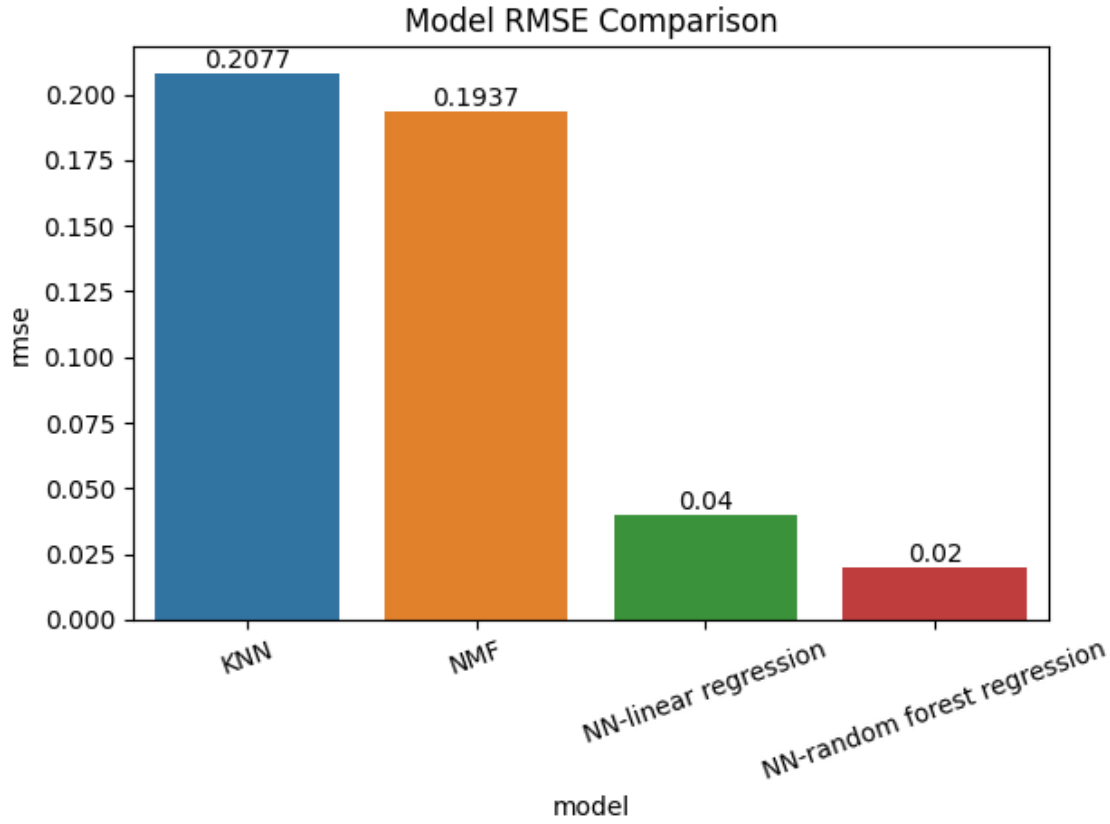
FLOWCHART OF NEURAL NETWORK EMBEDDING BASED RECOMMENDER SYSTEM

- Input data is user-item-rating interaction matrix.
- Surprise scikit learn NMF model is initiated using hyperparameters (n_factors, n_epochs). Train dataset is fitted to model to get the User Embedding and Item Embedding Feature Vectors.
- 2 Vectors are then merged either by addition, multiplication, min/max or average.
- Merged dataset is then split into training and test dataset.
- Linear Regression, or other regression ML model, is fitted with the training dataset.
- Predictions of ratings are then made on the trained model using the test dataset.
- Root mean square error between the predictions and test dataset where truth is known.



COMPARE THE PERFORMANCE OF COLLABORATIVE-FILTERING MODELS

- KNN and NMF models result in similar RMSE values.
- Much lower RMSE is with NN-linear regression. The combination of dimensionality reduction and linear regression produced lower RMSE than NMF alone.
- Even lower RMSE is with NN-random forest regression.
- Further optimization of hyperparameters given more computation resources may result in even lower RMSE.



OPTIONAL: BUILD A COURSE RECOMMENDER SYSTEM APP WITH STREAMLIT

Streamlit app screenshot1

Streamlit app screenshot2

A published Streamlit App URL for a live demo

Conclusions

- Most popular courses based on course titles relates to python, data science, data analysis and machine learning.
- Content based recommendation models based on user profiles and course genre, course similarity and clustering mostly recommended data science related courses, which could be simply due to the high popular of these courses and number of course with genres related to data science.
- Stricter constraints, like higher recommendation score thresholds and higher minimum number of enrolled, results in fewer courses being recommended. Adjusting these variables will change the number of recommended courses.
- Of the different collaborative filtering models evaluated, NN-random forest regression performed best in predicting ratings for courses by users.
- Further ML regression models and optimization of hyperparameters given more computation resources may result in even lower RMSE.

Appendix

- course_genre_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/course_genre.csv
- ratings_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/ratings.csv
- profile_genre_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/user_profile.csv
- test_users_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/rs_content_test.csv
- sim_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/sim.csv
- user_emb_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/user_embeddings.csv
- item_emb_url = https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/datasets/course_embeddings.csv