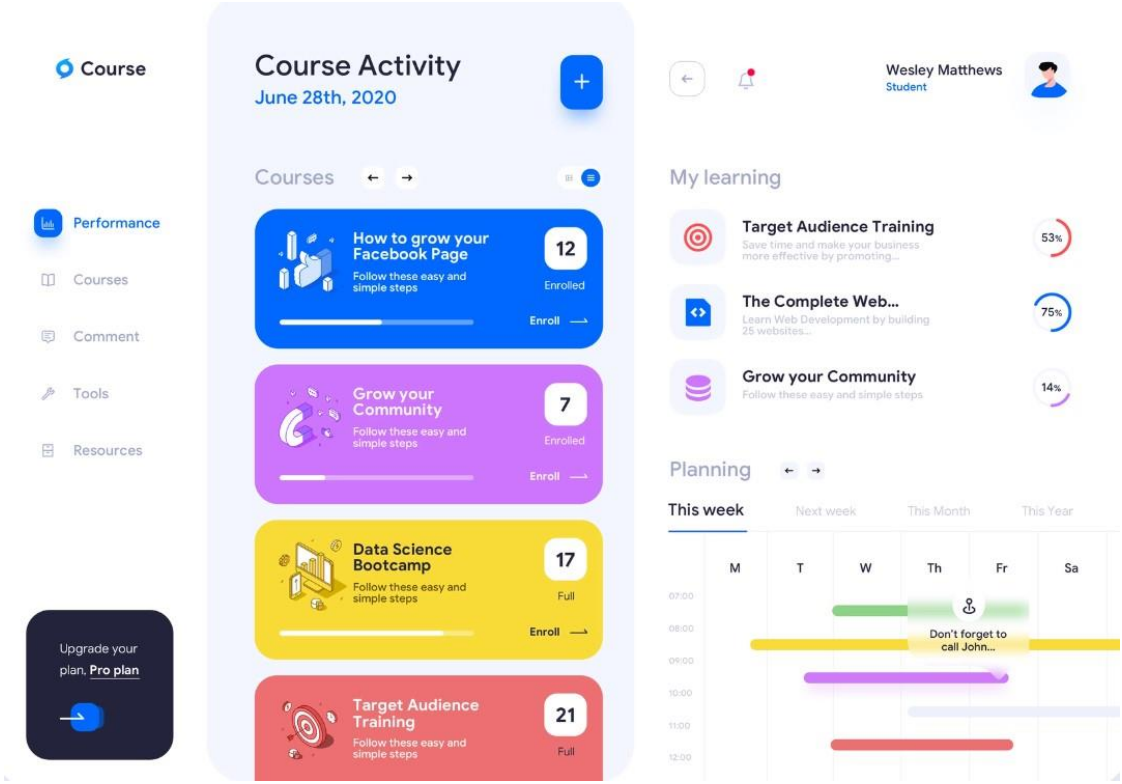
Building a Personalized Online Course Recommender System with Machine Learning

Syed Bokhari

25th January 2023



Outline

Introduction and Background

Recommender systems have become an increasingly ubiquitous part of the digital space. Online shopping sites, streaming services, browsing applications, news feeds and many other digital services and systems make use of recommender systems to provide digital users with additional options catering to their interests for a more engaging digital experience. As the digital world grows further, so too will the use of recommender systems..

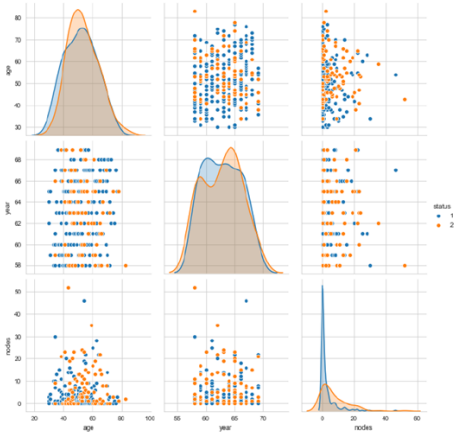
Recommender systems make use of machine learning algorithms along with user profile and history to generate recommendations. There are broadly two machine learning approaches employed by recommender systems: Content based which uses unsupervised learning and Collaborative Filtering based which uses supervised learning . This project will concern itself with the creation of a Course Recommender System making use of different algorithms from both of these approaches.

For a recommender system to generate recommendations, data about users such as user profile and history as well as the items they use is required. The project will assume that data exists about some online course hosting site with fictional user profiles as well as a list of online courses. It will make use of this data to investigate different machine learning algorithms from

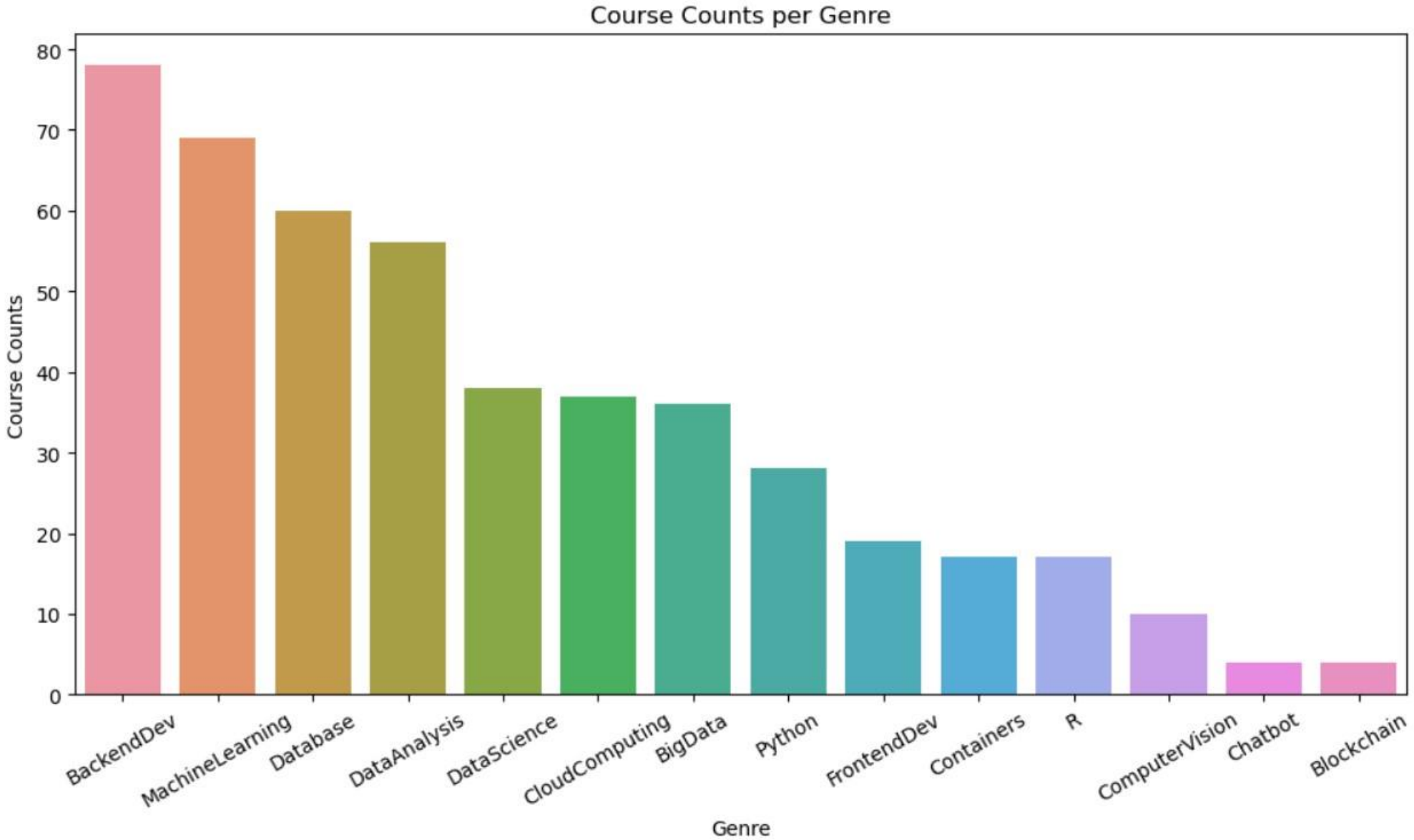
Introduction and Background

both the content based and collaborative filtering approaches. Models from both approaches will be investigated and trained to determine how course recommendations may be generated from the provided datasets. Subsequently, an application will be designed that integrates all these models into a coherent course recommender system which generates recommendations based on user input. After completion, the application will then be published online thereby creating a Personalized Online Course Recommender System with Machine Learning.

Exploratory Data Analysis



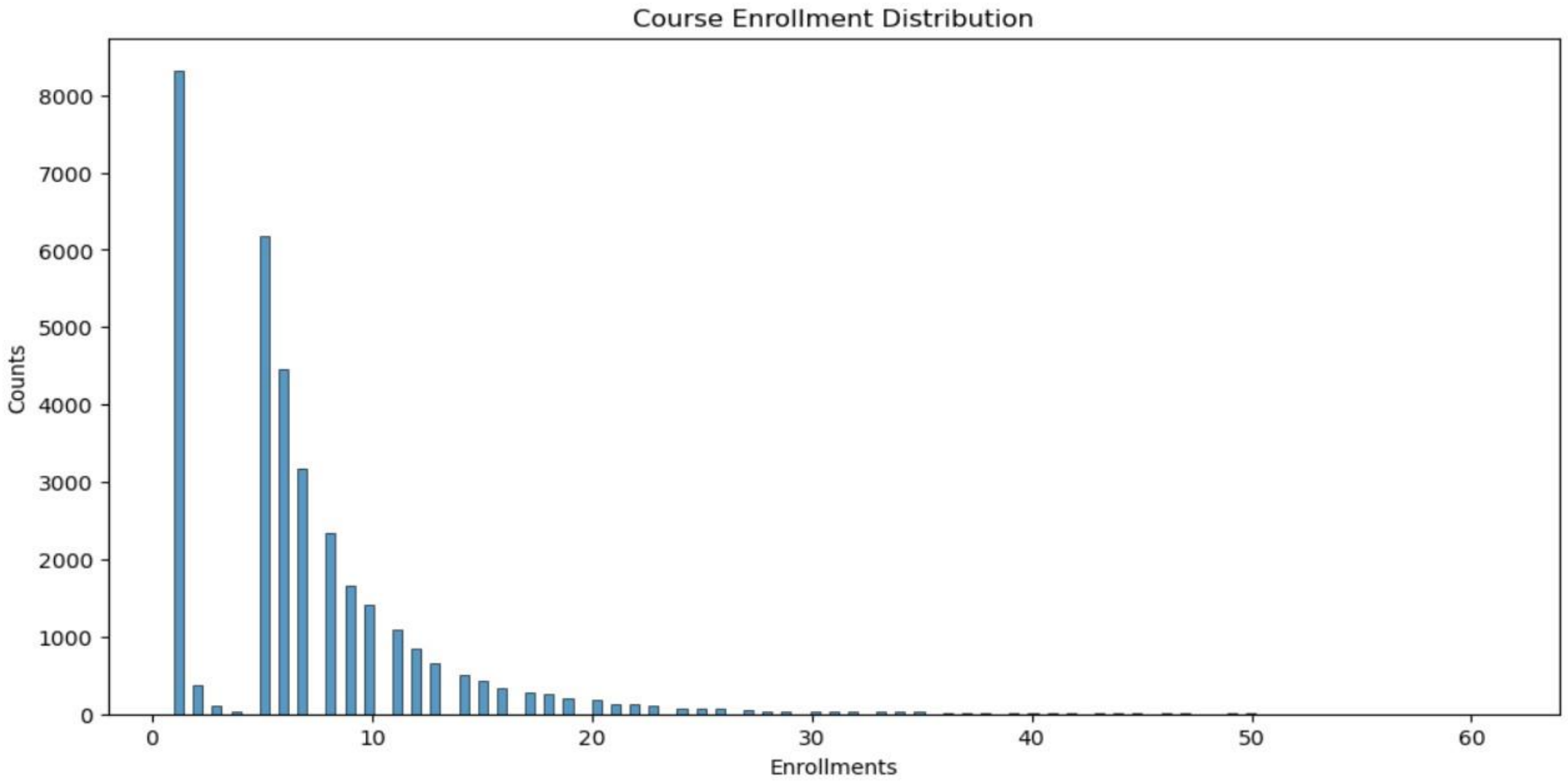
Course counts per genre



Course counts per genre

- A bar chart of the course counts was plotted.
- The bar chart provides a count of the courses for each of the genres specified from the dataset.
- From the bar chart, it can be observed that Backend Development is the genre with most courses in the dataset. It is followed by Machine Learning, then Databases, then Data Analysis and then Data Science.
- Blockchain is the genre with the least number of courses in the dataset. Chatbot is the genre with the second least number of courses.

Course enrollment distribution



Course enrollment distribution

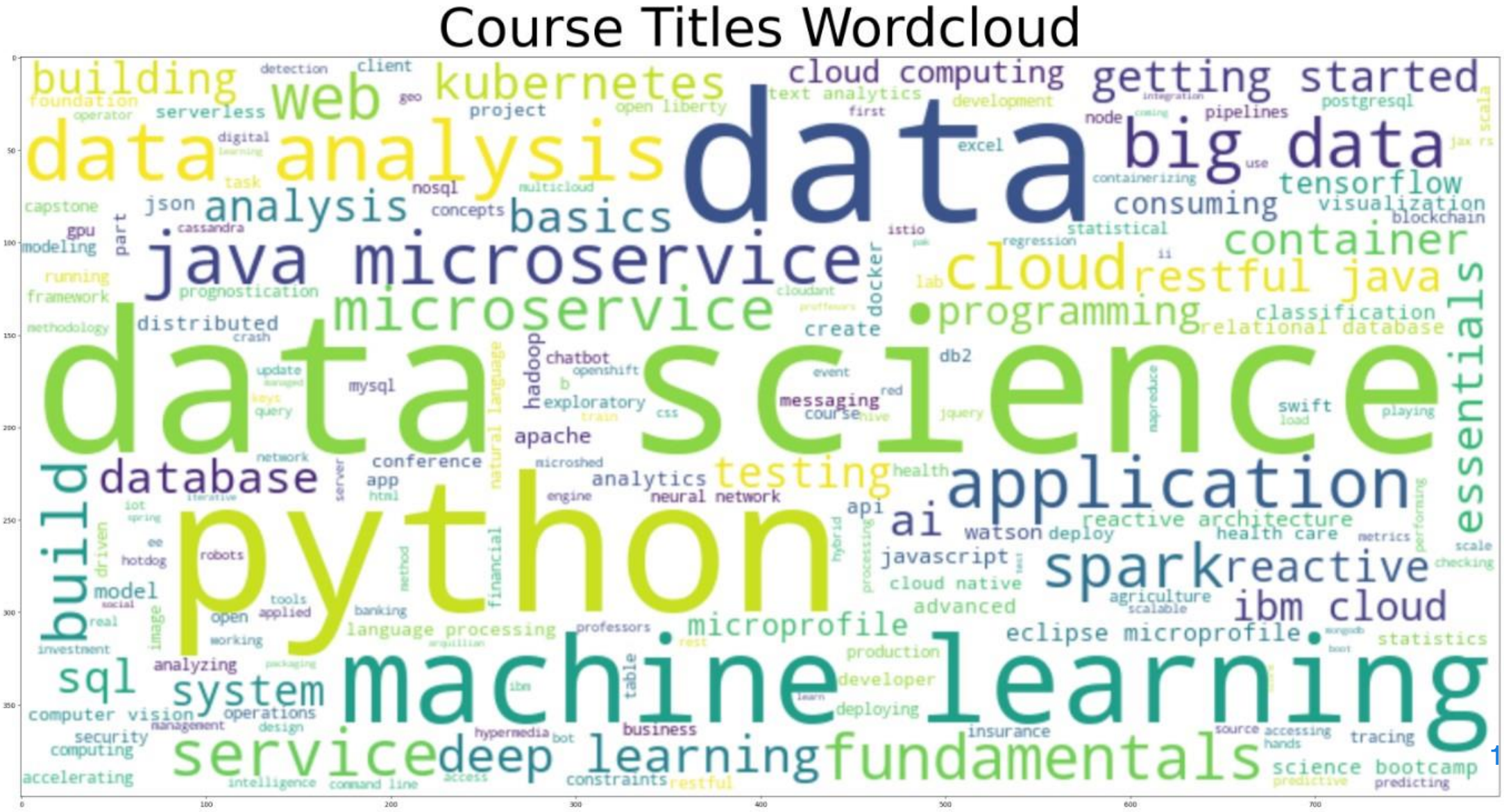
- A histogram showing the distribution of course enrollments was also plotted.
- The histogram shows the number of students enrolled in a specific number of courses e.g. x number are enrolled in 10 courses, y number are enrolled in 20 courses etc. Each is represented by a bar of the histogram.
- From the histogram, it is clear that the vast majority of students are enrolled in less than10 courses.
- It appears that most students are only enrolled in 1 course from the histogram.
- Interestingly, the histogram shows that the second most popular number of courses for enrollment is 5 courses. 50 courses is the highest number of courses a student has been enrolled in.

20 most popular courses

- A list of the top 20 most popular courses is shown. These courses are the ones with the highest student enrollments.
- Courses on using Python for various technology disciplines and Data Science seem to have the most enrollments.
- Big Data is also very popular with several courses on specific Big Data technologies (Hadoop, mapreduce and yarn etc.) also having high enrollments.
- Most popular courses are introductory courses introducing a field or subject. This indicates most users are interested in learning about new fields.
- These 20 courses account for ~63% of all enrolled courses.

	TITLE	Enrollments
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

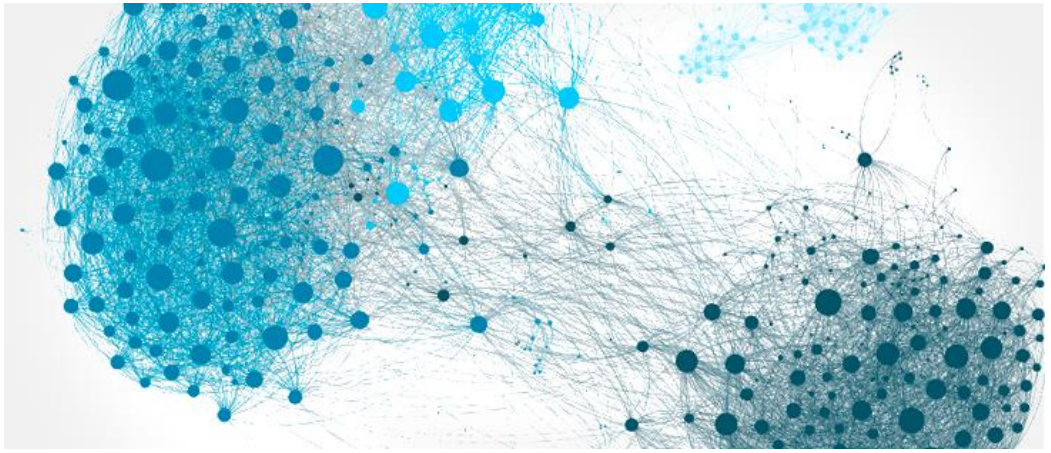
Word cloud of course titles



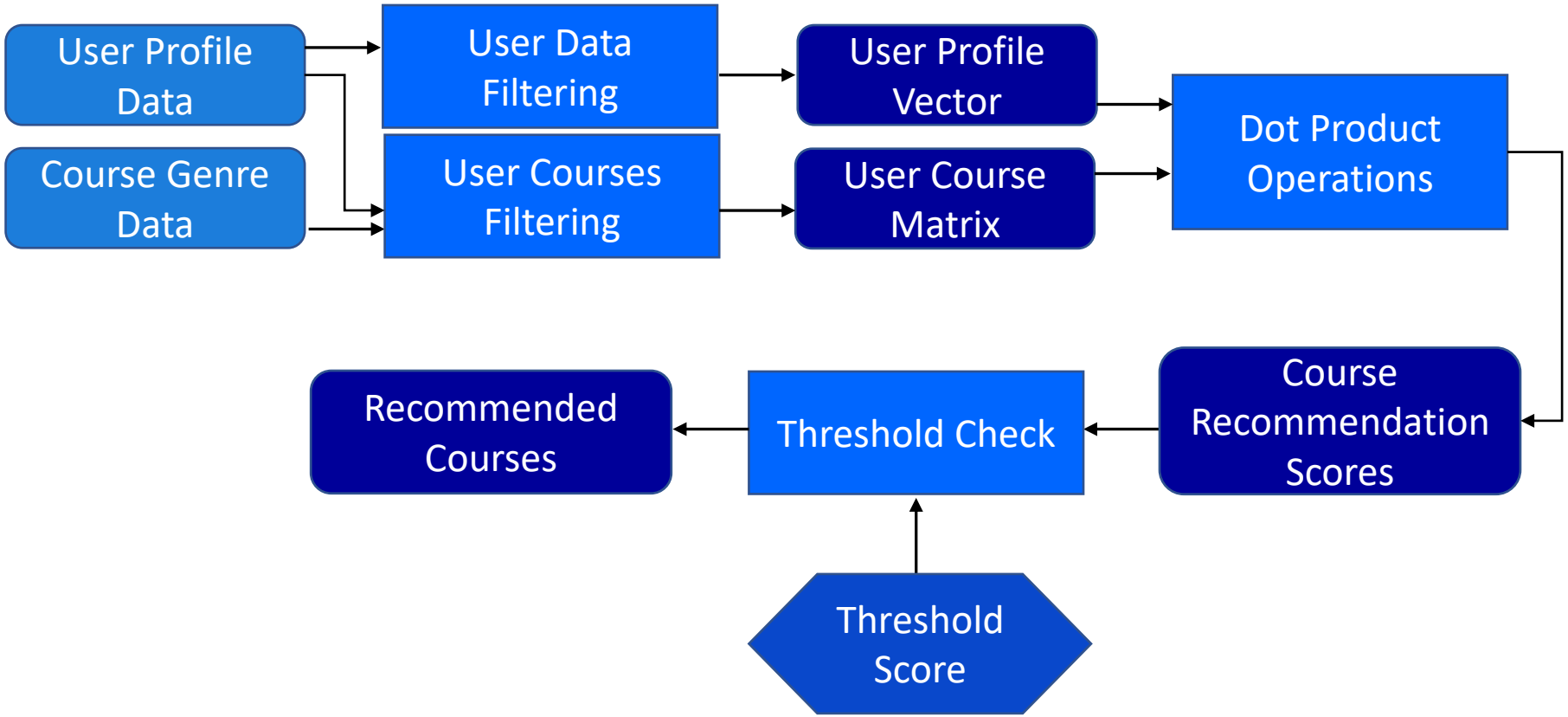
Word cloud of course titles

- A word cloud was generated from the titles of the courses. Some common stop words were removed from the titles so that only keywords would remain. The size of a word in the word cloud indicates how many times it was repeated.
- As expected, most of the words in the word cloud were IT, computing and technology related words.
- The word cloud shows that data, data science and python were the most oft repeated words in the course titles. Machine Learning was also repeated often.

Content-based Recommender System using Unsupervised Learning



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

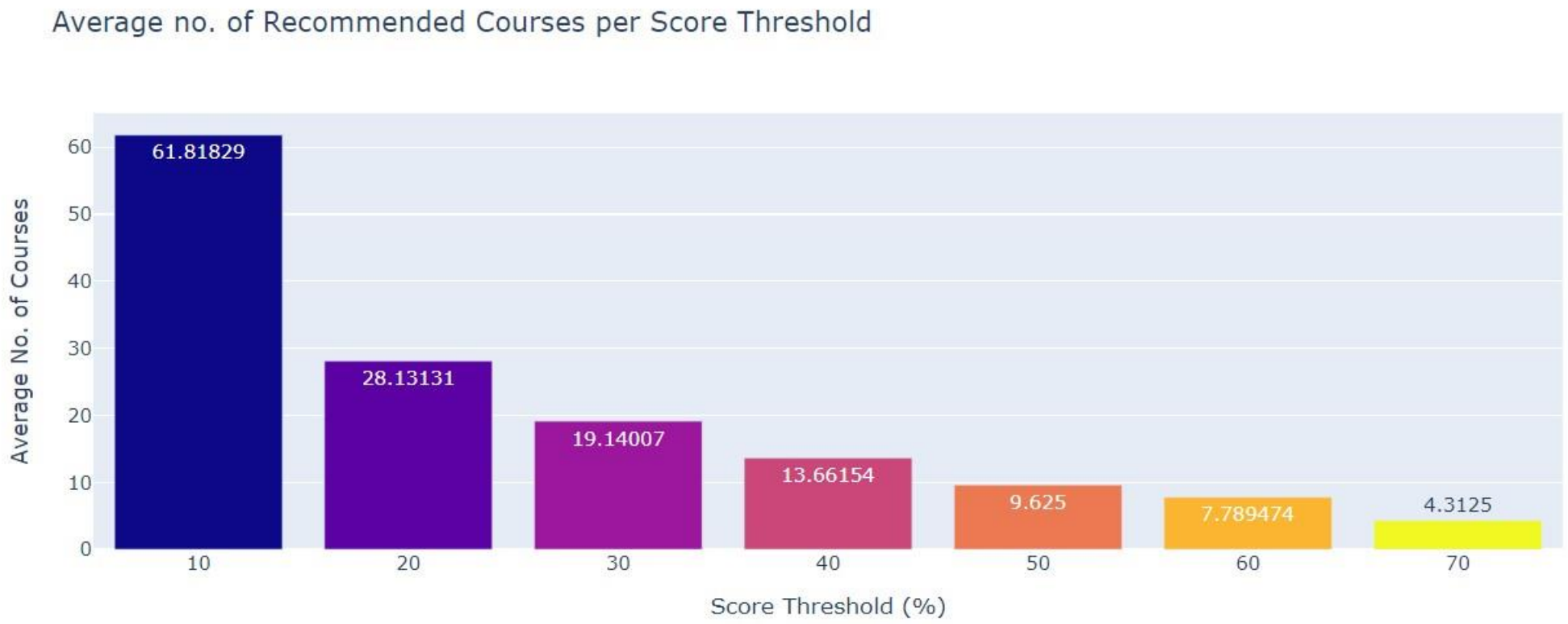


Flowchart of content-based recommender system using user profile and course genres

- Initially, user profile data with course genres a user has shown interest in generated from user course history is taken. In addition, a list of all courses and the genres each of those courses belongs to are also taken.
- Next, for each user, a user test vector with user interest in genres recorded in vector form is generated.
- Also, the courses the user enrolled for in the past are used to create a course matrix for the user. All previously enrolled courses are filtered out from the list of courses and only the one hot encoded genres of the remaining courses are used to generate this matrix.
- The dot product of the course matrix and user profile vector is taken and recommendation scores for the user are generated. Recommendation scores indicate user's likelihood of interest in the courses.
- The recommendation scores are subjected to a threshold check against a previously defined score threshold. Only courses passing the check are recommended to the user.

Evaluation results of user profile-based recommender system

- The average number of courses recommended per user with different similarity thresholds



Evaluation results of user profile-based recommender system

- Top 10 courses with similarity threshold 10%, 20%, 30% and 40%

	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	TA0106EN	608	excourse73	322	excourse72	197	excourse73	94
1	GPXX0IBEN	548	excourse72	322	excourse73	197	excourse72	94
2	excourse22	547	TMP0105EN	318	TMP0105EN	193	TMP0105EN	92
3	excourse21	547	RP0105EN	310	RP0105EN	178	SC0103EN	78
4	ML0122EN	544	SC0103EN	292	SC0103EN	173	RP0105EN	78
5	GPXX0TY1EN	533	excourse31	283	excourse31	156	excourse31	66
6	excourse04	533	excourse22	271	BD0212EN	147	GPXX0M6UEN	60
7	excourse06	533	excourse21	271	excourse71	132	GPXX097UEN	60
8	excourse31	524	BD0212EN	268	excourse42	132	excourse03	60
9	excourse73	516	ML0122EN	268	GPXX097UEN	132	excourse05	60

Evaluation results of user profile-based recommender system

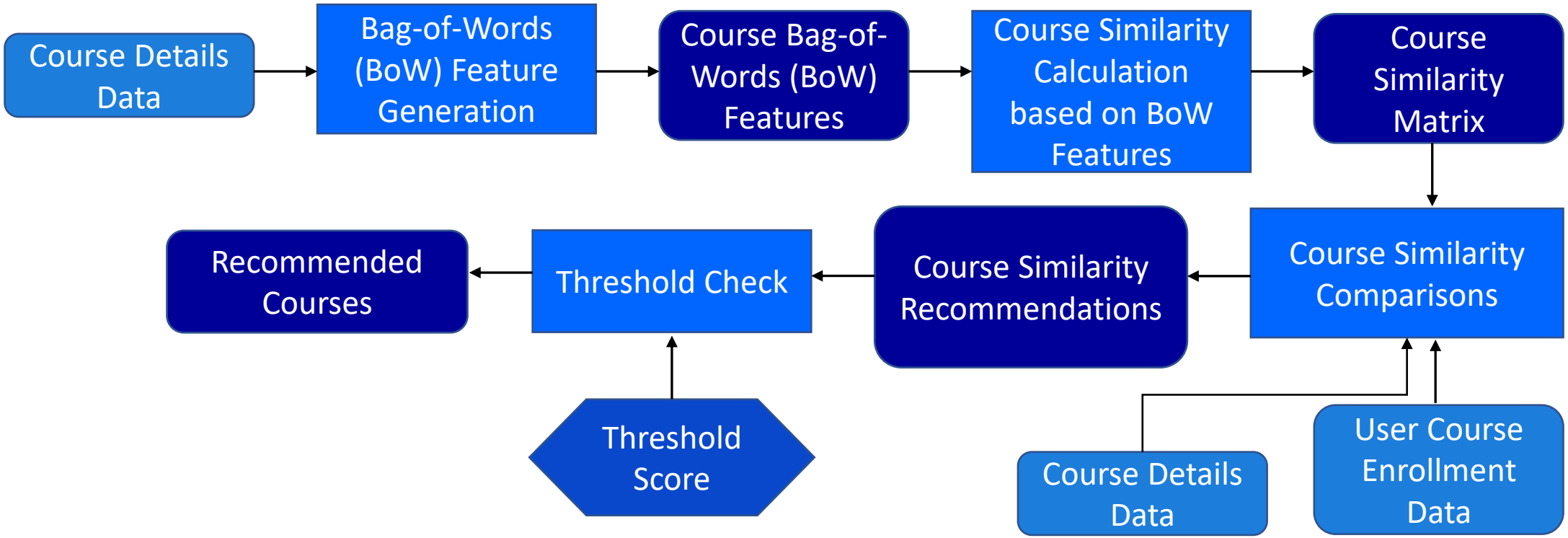
- Top 10 courses with similarity threshold 50%, 60% and 70%

	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	excourse72	50	excourse72	30	excourse72	14
1	excourse73	50	excourse73	30	excourse73	14
2	TMP0105EN	49	TMP0105EN	29	TMP0105EN	13
3	SC0103EN	37	SC0103EN	20	RP0105EN	7
4	excourse31	36	excourse31	18	SC0103EN	6
5	RP0105EN	33	RP0105EN	14	BD0212EN	3
6	excourse71	27	GPXX0M6UEN	12	excourse31	3
7	GPXX097UEN	27	GPXX097UEN	12	DB0151EN	1
8	excourse03	27	excourse03	12	GPXX0M6UEN	1
9	excourse05	27	excourse05	12	GPXX097UEN	1

Evaluation results of user profile-based recommender system

- From the bar graph on average courses recommended to users, it is clear that as similarity threshold is raised, number of courses recommended to users drops sharply. This is sensible as raising the threshold will filter out more courses that cannot meet the similarity threshold.
- From the top 10 courses recommended, it can also be observed that as the similarity threshold is raised, the number of times a course is recommended falls since fewer courses will meet the threshold's increased requirements and be recommended to students.
- It can be observed that for all similarity thresholds barring10%, the top 3 courses recommended to students remain the same.
- Several courses other than the top 3 also make repeat appearances in top 10 courses for all similarity thresholds except 10%. Some also appear for a few thresholds only but not all.
- 10% threshold seems to give the most spurious results while consistency of results for other thresholds indicates the recommendations thus generated are rather decent.

Flowchart of content-based recommender system using course similarity



Flowchart of content-based recommender system using course similarity

- First, data about various details of the courses is taken. Next, Bag-of-Words Features are generated from the course names. This yields features based on Bag-of-Words for future comparisons.
- Next, the similarities between all courses are calculated using the Bag-of-Words features. The similarity scores are stored in a Course Similarity Matrix. For ease of use, all courses are encoded as numbers prior to this. This makes similarity lookups from the matrix very easy.
- The Course Similarity Matrix is used, along with data on enrolled courses of the user and course details to create Course Similarity Recommendations. The similarity between an enrolled course and an unknown course is looked up from the similarity matrix and f there is no existing similarity score or a lower similarity score, the new score for the course is added.
- The Course Similarity Recommendations are subjected to a threshold check against a previously defined score threshold and only courses passing the check are recommended.

Evaluation results of course similarity based recommender system

- The average number of courses recommended per user with different similarity thresholds

Evaluation results of course similarity based recommender system

- Top 10 courses with similarity threshold 10%, 20%, 30% and 40%

	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	excourse43	1000	excourse31	965	excourse32	953	excourse68	921
1	excourse50	995	excourse38	961	excourse33	952	excourse23	915
2	excourse71	995	excourse69	961	excourse68	952	excourse36	915
3	excourse26	993	GPXX0ZMZEN	959	excourse36	952	excourse74	913
4	excourse31	992	excourse22	958	excourse23	952	excourse67	907
5	GPXX0NHZEN	992	excourse72	958	excourse67	952	excourse32	900
6	GPXX0LLEEN	990	excourse62	958	excourse63	950	excourse38	886
7	excourse54	990	excourse74	957	excourse04	941	excourse72	863
8	excourse40	990	excourse23	956	excourse69	940	excourse33	857
9	DP0101EN	989	excourse33	956	excourse09	940	excourse04	853

Evaluation results of course similarity based recommender system

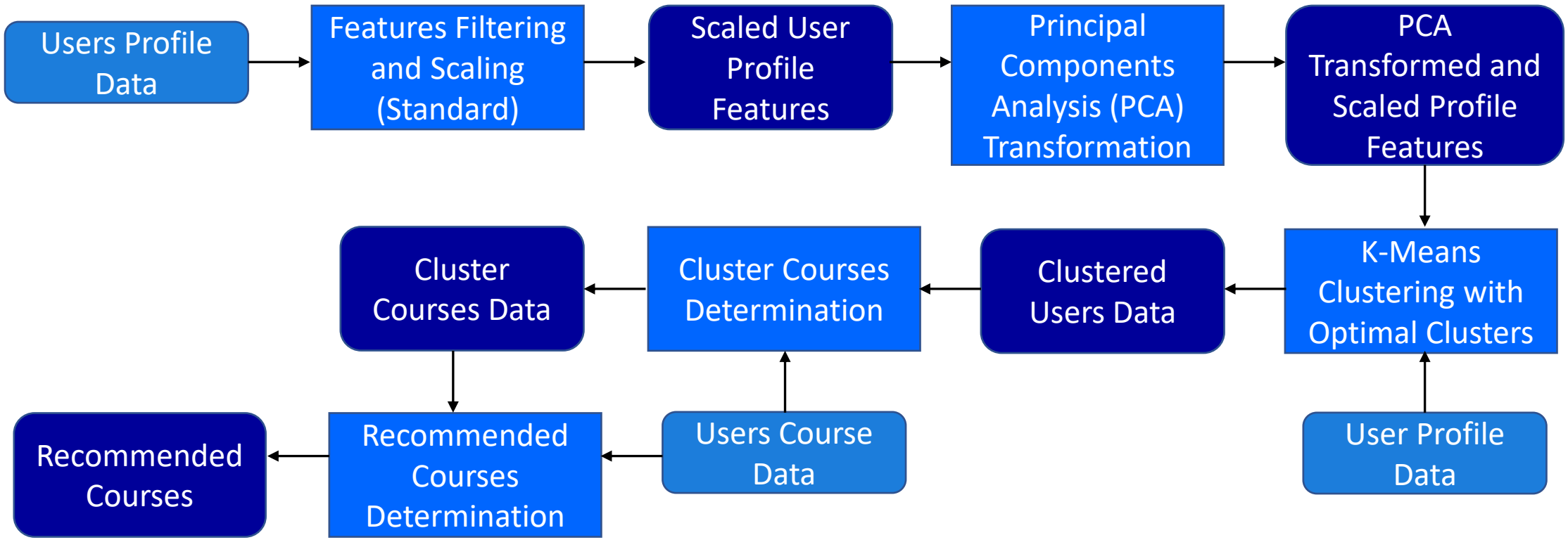
- Top 10 courses with similarity threshold 50%, 60% and 70%

	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	excourse68	887	excourse62	579	excourse67	539
1	excourse67	854	excourse22	579	DS0110EN	534
2	excourse32	828	DS0110EN	562	excourse32	329
3	excourse36	779	excourse63	555	excourse23	321
4	excourse23	779	excourse65	555	excourse36	321
5	TMP107	773	excourse72	551	ML0122ENv3	261
6	excourse74	728	excourse68	550	DV0151EN	225
7	DS0110EN	726	excourse74	539	excourse24	198
8	excourse09	715	excourse67	539	CB0101EN	181
9	excourse65	715	BD0145EN	506	ML0120ENv3	156

Evaluation results of course similarity based recommender system

- From the bar graph on average courses recommended to users, it is clear that as similarity threshold is raised, number of courses recommended to users drops sharply. Compared to user profile based system, far more courses on average are recommended at almost every threshold.
- From the top10 courses recommended, it can also be observed that as the similarity threshold is raised, the number of times a course is recommended falls slightly and then more sharply. At every threshold, top10 courses recommended are far higher than user profile based system.
- It is interesting to note that for the top10 courses, most courses recommended vary as thresholds are changed. There is no consistent pattern. This is in contrast to the user profile based system.

Flowchart of clustering-based recommender system

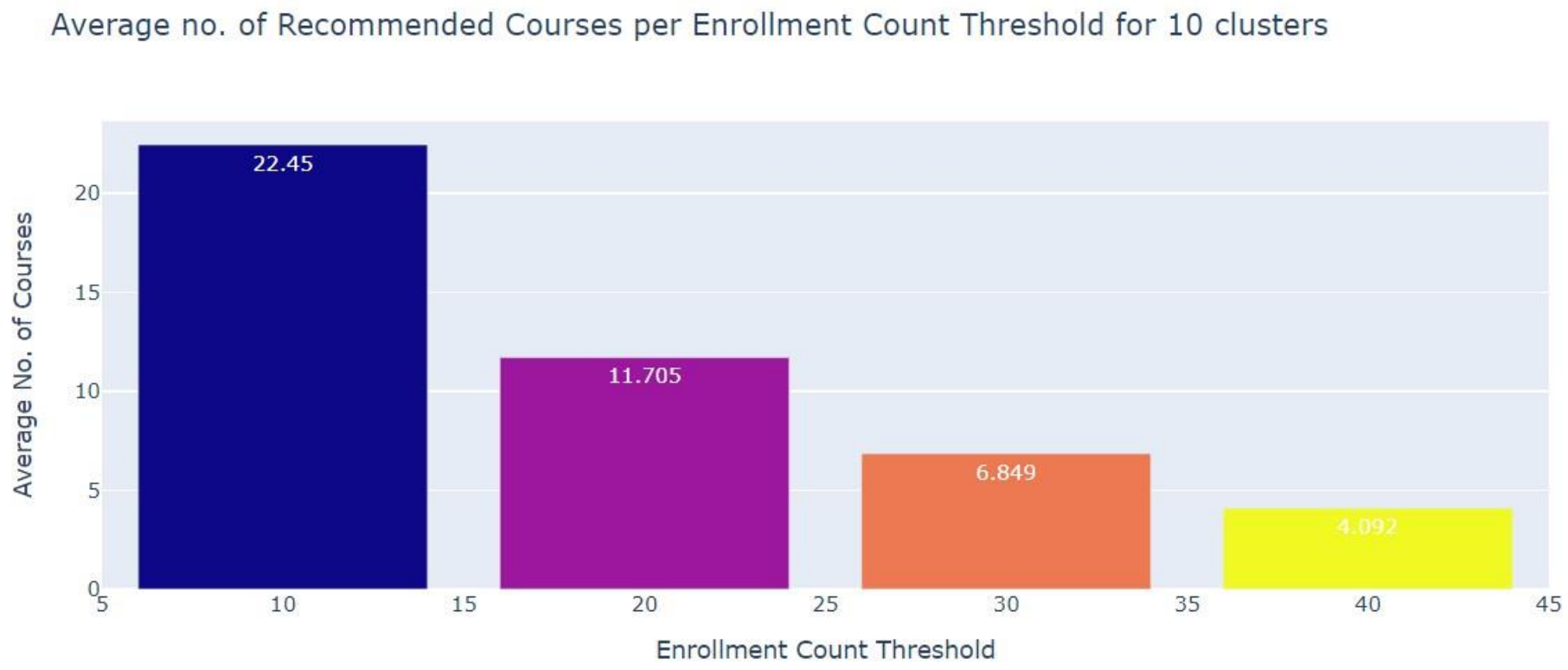


Flowchart of clustering-based recommender system

- User profiles for course genres are taken first. The features (genres) are subsequently taken and scaled for future use. Standard scaling method is used.
- Subsequently, the features are optionally transformed through Principal Components Analysis (PCA) to remove redundant features and reduce dimensionality.
- The transformed features, along with user profile data are clustered through the K-Means Algorithm. This results in all users being assigned a cluster. Optimal number of clusters may be found using elbow method or something similar.
- Users Cluster Data along with Users Course Data is used to determine popular courses in each cluster.
- Users Course Data and Cluster Courses Data is used to find popular courses in the cluster a user has not enrolled in. These courses are recommended to the user.

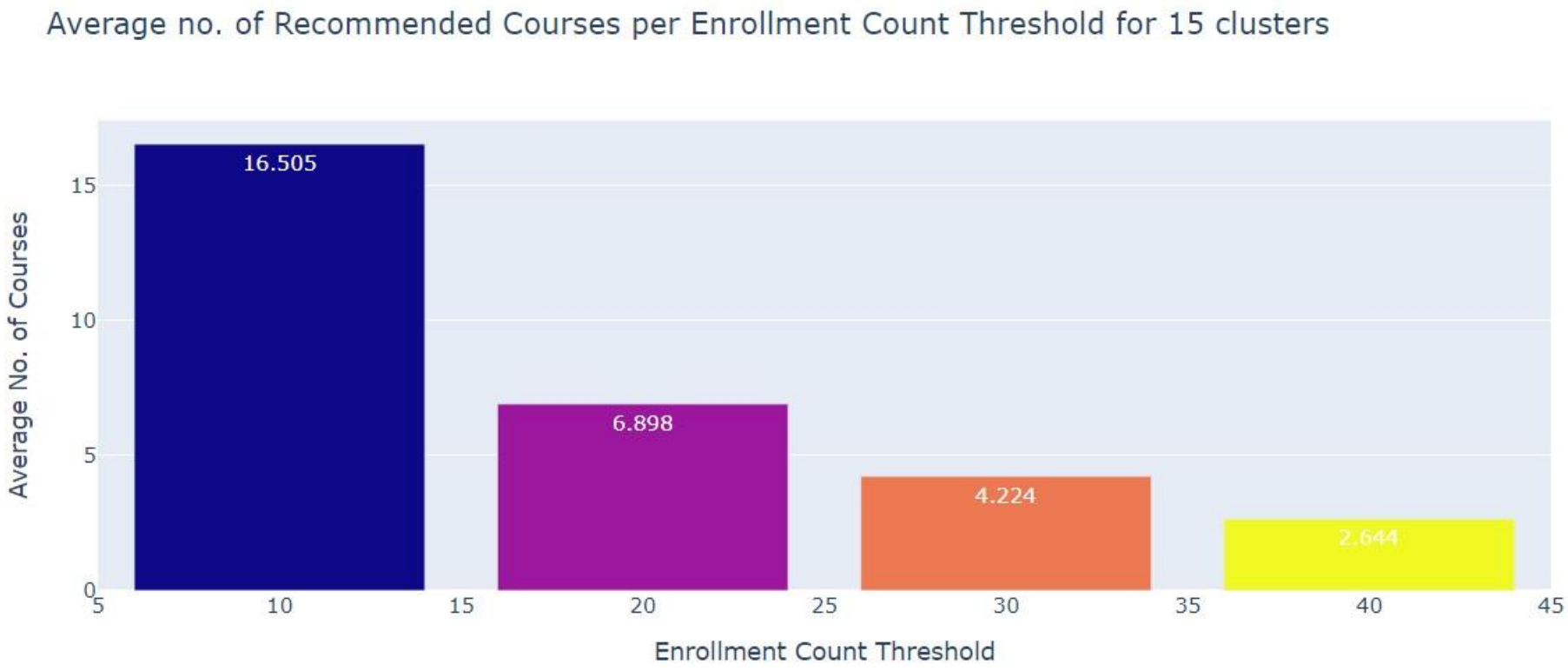
Evaluation results of clustering-based recommender system

- The average number of courses recommended per user with different enrollment count thresholds for 10 clusters



Evaluation results of clustering-based recommender system

- The average number of courses recommended per user with different enrollment count thresholds for 15 clusters



Evaluation results of clustering-based recommender system

- Top 10 courses with enrollment count threshold 10, 20, 30 and 40 for 10 clusters

	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	ST0101EN	778	DS0105EN	589	BD0111EN	468	DS0101EN	428
1	DB0101EN	720	DS0103EN	557	BD0101EN	442	PY0101EN	398
2	ML0115EN	710	CC0101EN	538	DS0101EN	428	BD0101EN	366
3	BD0211EN	674	RP0101EN	537	PY0101EN	398	BD0111EN	314
4	CL0101EN	670	CO0101EN	534	ML0115EN	375	ST0101EN	296
5	DA0101EN	658	ML0101ENv3	516	ST0101EN	342	DS0105EN	284
6	DS0103EN	658	ML0115EN	515	DA0101EN	327	DS0103EN	257
7	DS0301EN	614	BD0111EN	468	DS0301EN	318	ML0115EN	251
8	DS0105EN	596	BD0101EN	442	DV0101EN	314	DV0101EN	174
9	BD0111EN	561	ST0101EN	429	ML0101ENv3	299	BD0211EN	168

Evaluation results of clustering-based recommender system

- Top 10 courses with enrollment count threshold 10, 20, 30 and 40 for 15 clusters

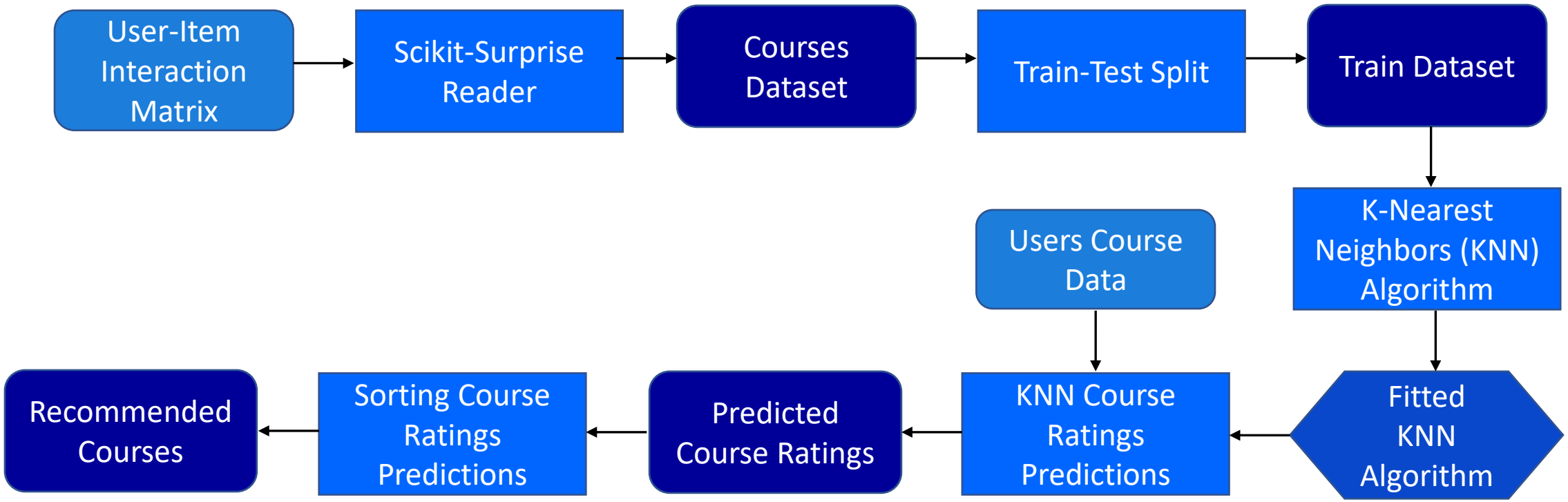
	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended	Course_ID	Times_Recommended
0	CC0101EN	653	BD0111EN	441	DS0101EN	385	BD0101EN	331
1	ST0101EN	635	BD0101EN	436	BD0101EN	349	DS0101EN	321
2	DS0105EN	603	DS0101EN	400	PY0101EN	346	PY0101EN	242
3	DA0101EN	592	DS0103EN	373	BD0111EN	304	DS0103EN	224
4	ML0115EN	590	PY0101EN	346	DS0105EN	259	DS0105EN	207
5	DB0101EN	581	DS0105EN	336	ST0101EN	235	ML0115EN	194
6	CL0101EN	569	BD0211EN	325	DS0103EN	233	ST0101EN	132
7	DS0103EN	569	ML0101ENv3	306	ML0101ENv3	211	BD0111EN	109
8	BD0111EN	562	ML0115EN	264	ML0115EN	206	BD0131EN	103
9	DS0301EN	554	DS0301EN	257	BD0211EN	154	BD0141EN	100

Evaluation results of clustering-based recommender system

- From the bar graph on average courses recommended to users, as enrollment count threshold is increased, average number of courses recommended per user fall dramatically. Furthermore, increasing the number of clusters reduces average number of courses recommended per user as well.
- From the top 10 most recommended courses, it can be seen that the number of recommendations for courses fall as enrollment count threshold is increased. Furthermore, it can be seen that the number of recommendations for courses fall as the number of clusters is raised as well.
- The top 10 most recommended courses also indicate that as enrollment count threshold is increased, the impact on less frequently recommended courses is greater than more frequently recommended courses. Furthermore, the same appears to be true as number of clusters is increased with less frequently recommended courses being impacted more.

Collaborative-filtering Recommender System using Supervised Learning

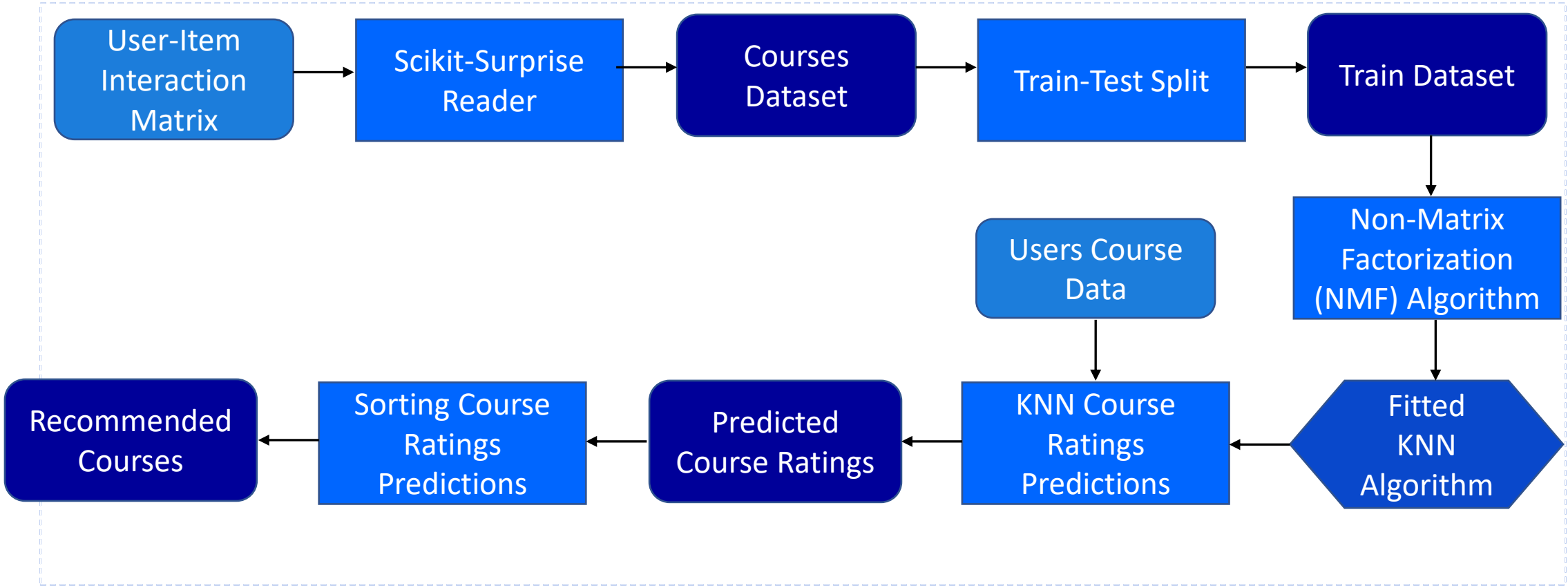
Flowchart of KNN based recommender system



Flowchart of KNN based recommender system

- First, a user-item interaction matrix is taken where user ratings for different items are given. This matrix is run through the Scikit-Surprise Library's Reader functionality to generate a Courses Dataset.
- The Courses Dataset is subjected to a train-test split. Although unnecessary, it is required as the next step involves creating a K-Nearest Neighbor (KNN) Algorithm object and fitting the training data to it. The algorithm only accepts the training dataset generated by the split.
- The fitted KNN Algorithm object is provided user-item matrix with users and possible courses they may be interested in for which ratings are to be predicted. It makes ratings estimation for users for different courses based on features learnt previously.
- The ratings estimations are sorted to produce recommended courses for users.

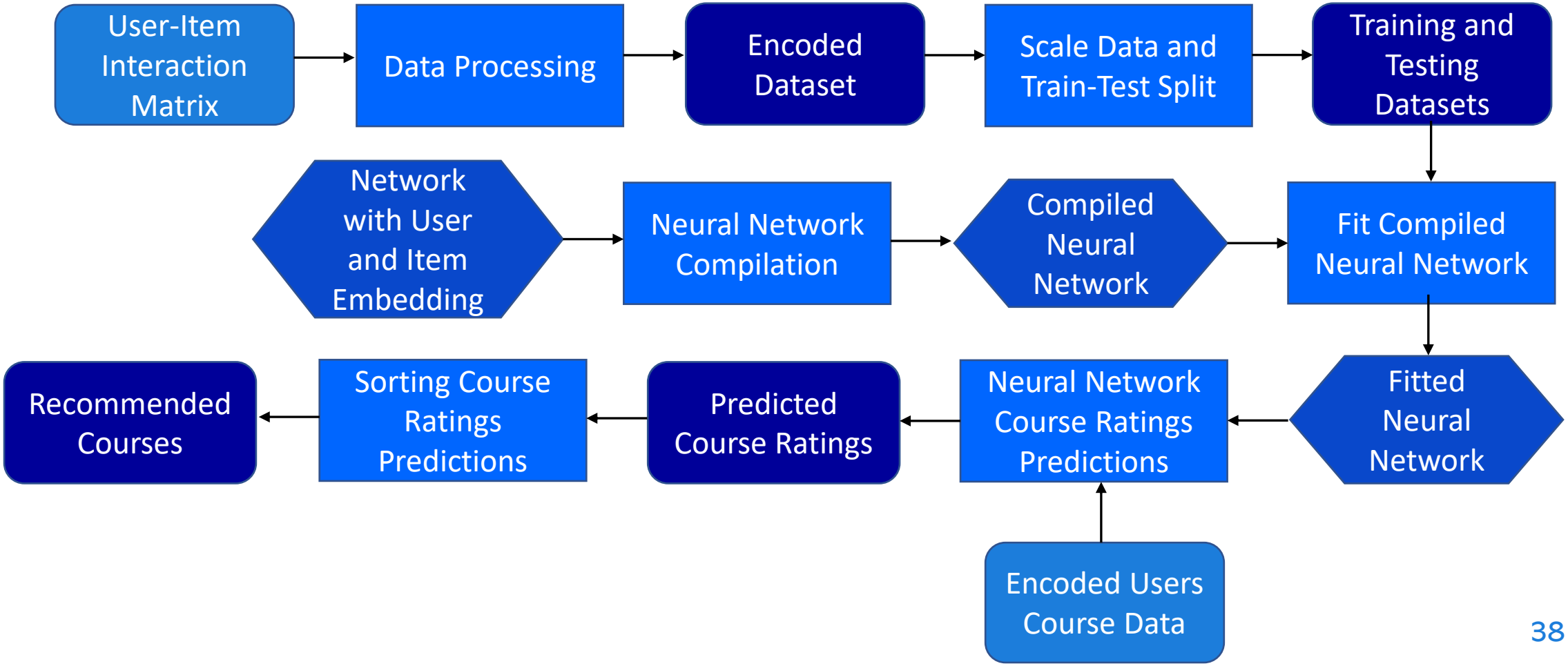
Flowchart of NMF based recommender system



Flowchart of NMF based recommender system

- First, a user-item interaction matrix is taken where user ratings for different items are given. This matrix is run through the Scikit-Surprise Library's Reader functionality to generate a Courses Dataset.
- The Courses Dataset is subjected to a train-test split. Although unnecessary, it is required as the next step involves creating a Non-Matrix Factorization (NMF) Algorithm object and fitting the training data to it. The algorithm only accepts the training dataset generated by the split.
- The fitted NMF Algorithm object is provided user-item matrix with users and possible courses they may be interested in for which ratings are to be predicted. It makes ratings estimation for users for different courses based on features learnt previously.
- The ratings estimations are sorted to produce recommended courses for users.

Flowchart of Neural Network Embedding based recommender system



Flowchart of Neural Network Embedding based recommender system

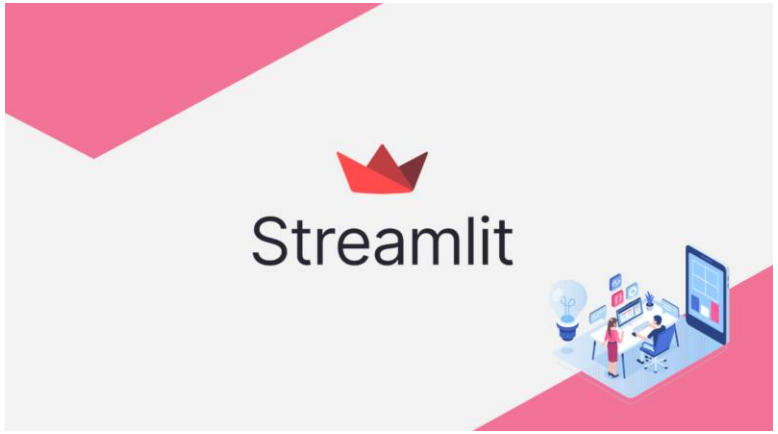
- First, a user-item interaction matrix is taken where user ratings for different items are given. The data is then processed to produce encoded versions of user and items dataset.
- The encoded dataset is min-max scaled and split into training and testing dataset.
- A neural network model with necessary embeddings and layers is defined. The model is compiled with the appropriate optimizer and loss metric defined. It is then fitted with the training dataset and (testing dataset for validation).
- The fitted network is then provided with data encompassing users and possible courses they may be interested in. The model predicts ratings for this data to determine which courses a user may be most interested in.
- The predicted ratings are sorted based on their scores and are used to make course recommendations for users.

Compare the performance of collaborative-filtering models

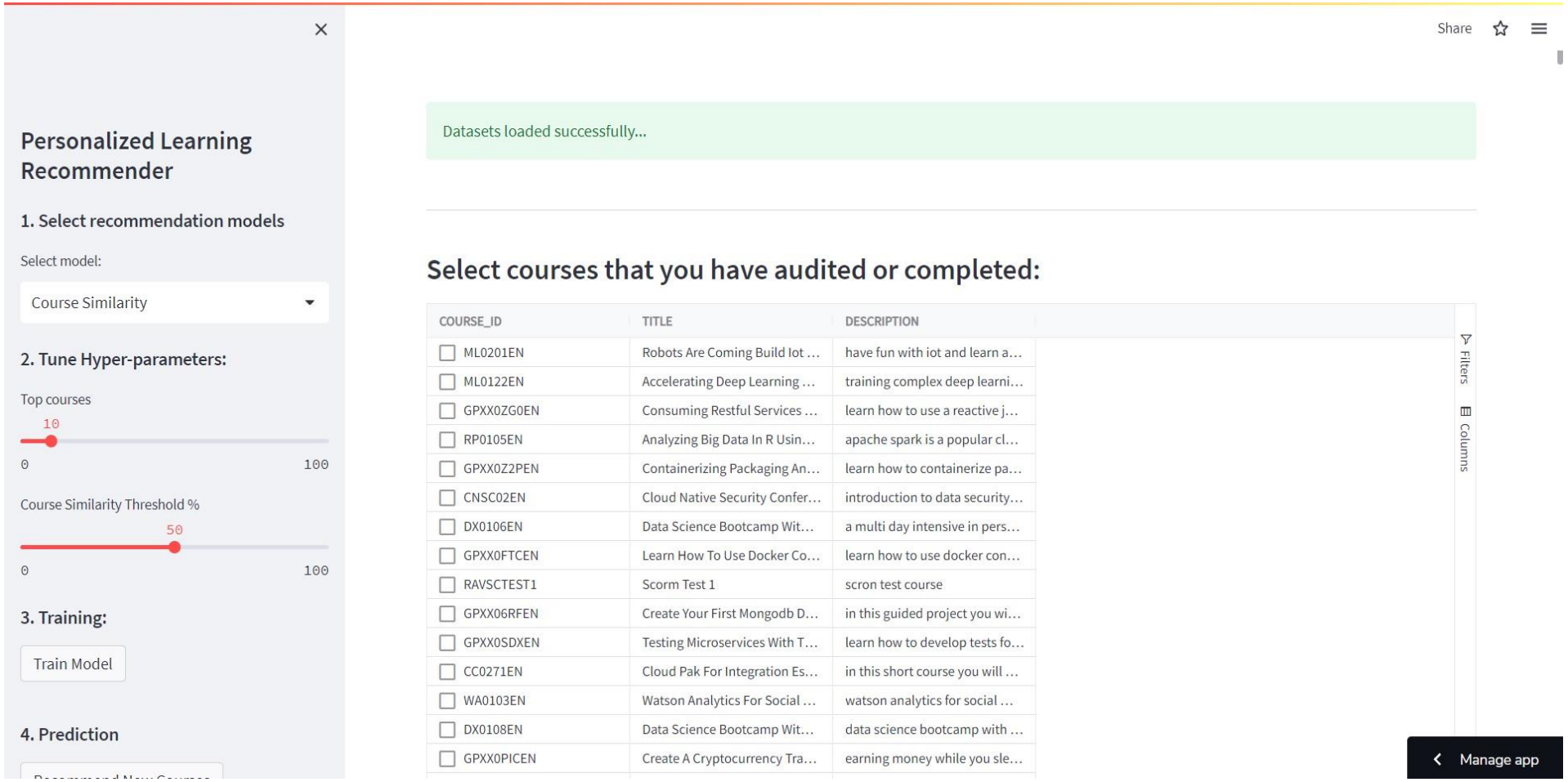
Compare the performance of collaborative-filtering models

- The bar chart comparing the performance of different collaborative filtering models is presented. NMF stands for Non Matrix Factorization and SVR for Support Vector Regression.
- Although several regression models were tried in addition to the ones presented in the chart, there was no discernible difference in performance. In the end, the 5 best models were presented.
- From the chart, it is clear that Random Forest Regression gave the least RMSE. However, this better result was obtained at the expense of additional training time for this model.
- Neural Network model gave the second best performance but it also required more time to train. However, the time required was significantly lesser than Random Forest Regression.
- Several classification models were also trained but were not included as the performance metric chosen (RMSE) could not be used to evaluate their performance.
- The KNN model was not added because it produced no discernible performance improvements but required a lot of memory to use.

Course Recommender System app with Streamlit

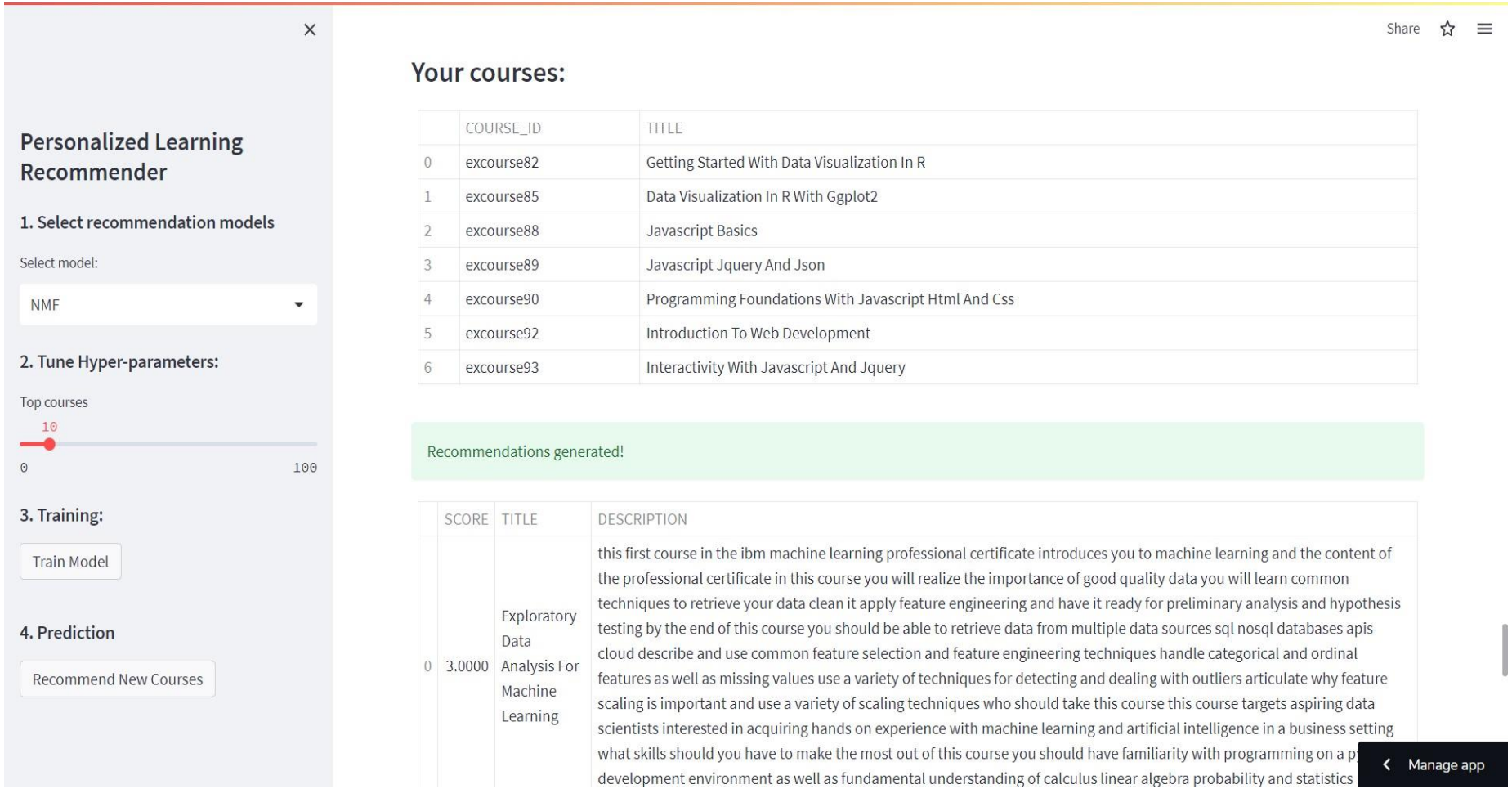


Course recommender system app with Streamlit



https://sheezer-course-recommender-app-recommender-app-6xy8bz.streamlit.app/

Course recommender system app with Streamlit



https://sheezer-course-recommender-app-recommender-app-6xy8bz.streamlit.app/

Course recommender system app with Streamlit

- As part of the project, a Course recommender system application was developed. The application made use of all the content similarity and collaborative filtering techniques studied previously. The app is deployed on Streamlit.
- Users of the application can select the courses they have audited or enrolled in, and choose a machine learning method to generate recommendations for courses to be taken based on similarity to the previously enrolled courses.
- The app has options for Course Similarity, User Profile, Clustering, Clustering with PCA, KNN, NMF, Neural Network, Regression with Embedding Features and Classification with Embedding Features. All these options work except KNN where the code is sound but the massive memory requirement causes crashes.
- Different options also have different hyperparameters that can be customized by the user according to their requirements.

Conclusions

- Most of the very popular courses seem to be introductory in nature indicating that most users are interested in learning about new subjects, not upskilling in existing subjects.
- Recommender systems using algorithms from the content based approach yielded varying results with different hyperparameter combinations.
- Recommender systems using algorithms from collaborative filtering approach yielded roundabout similar performance with RMSE metric (Root Mean Square Error) of ~ 0.2. The only exception to this was the Random Forest Regression. However, it took a prohibitively long time to train.
- All classification approaches yielded similar results.
- The online course recommender system was successfully created and is hosted at : https://sheezer-course-recommender-app-recommender-app-6xy8bz.streamlit.app/ . All options except KNN are functional. The KNN option does not work due to platform memory limitations.

Appendix

- Streamlit app url: https://sheezer-course-recommender-app-recommender-app-6xy8bz.streamlit.app/