Build a Personalized Online Course Recommender System with Machine Learning

Can ŞENTAY
12.07.2023

Outline

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

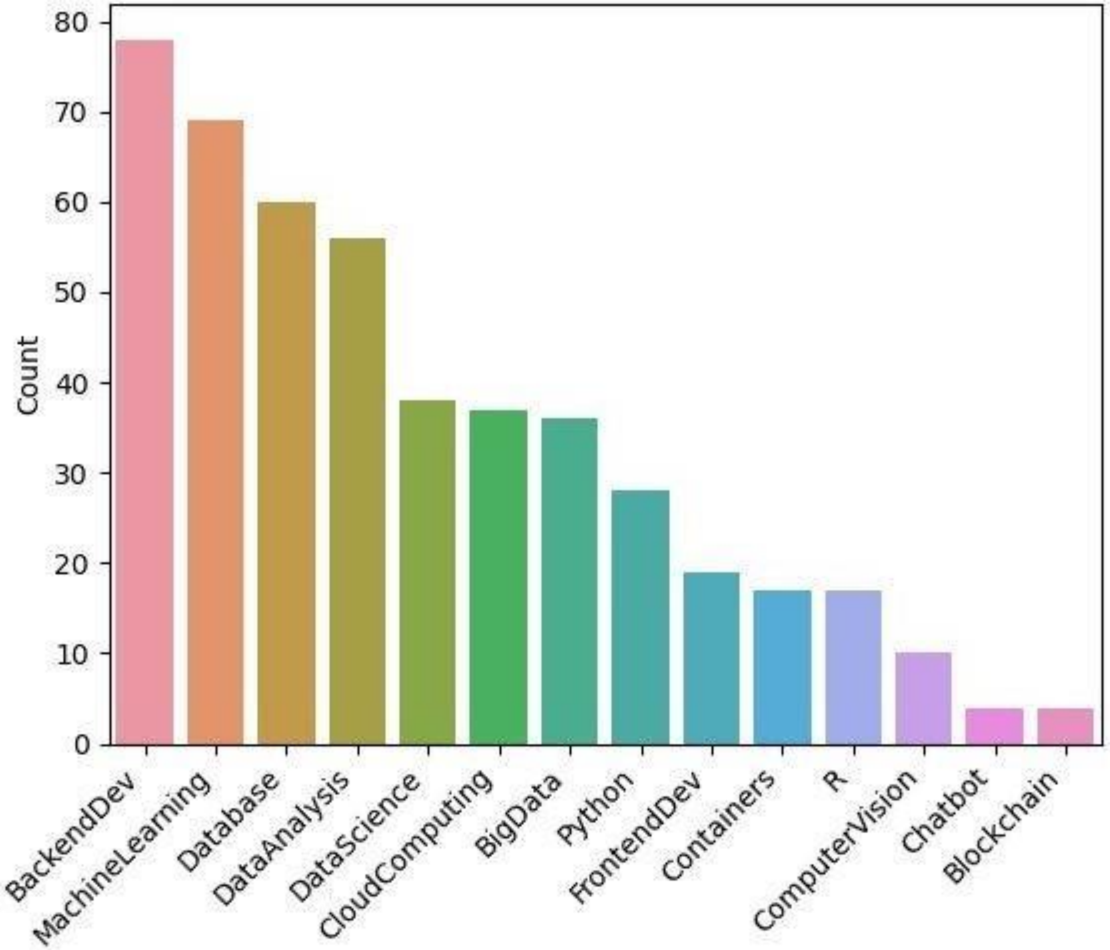
Introduction

- The AI training room company offers AI courses to millions of learners
- As the amount of courses available they have found it important to find the best next course for each consumer
- Creating a recommendation engine would then not only increase the quality of their service but also allow for upselling and increasing profits
- The project is currently in the proof-of-concept stage and multiple recommendation systems are being researched

Exploratory Data Analysis

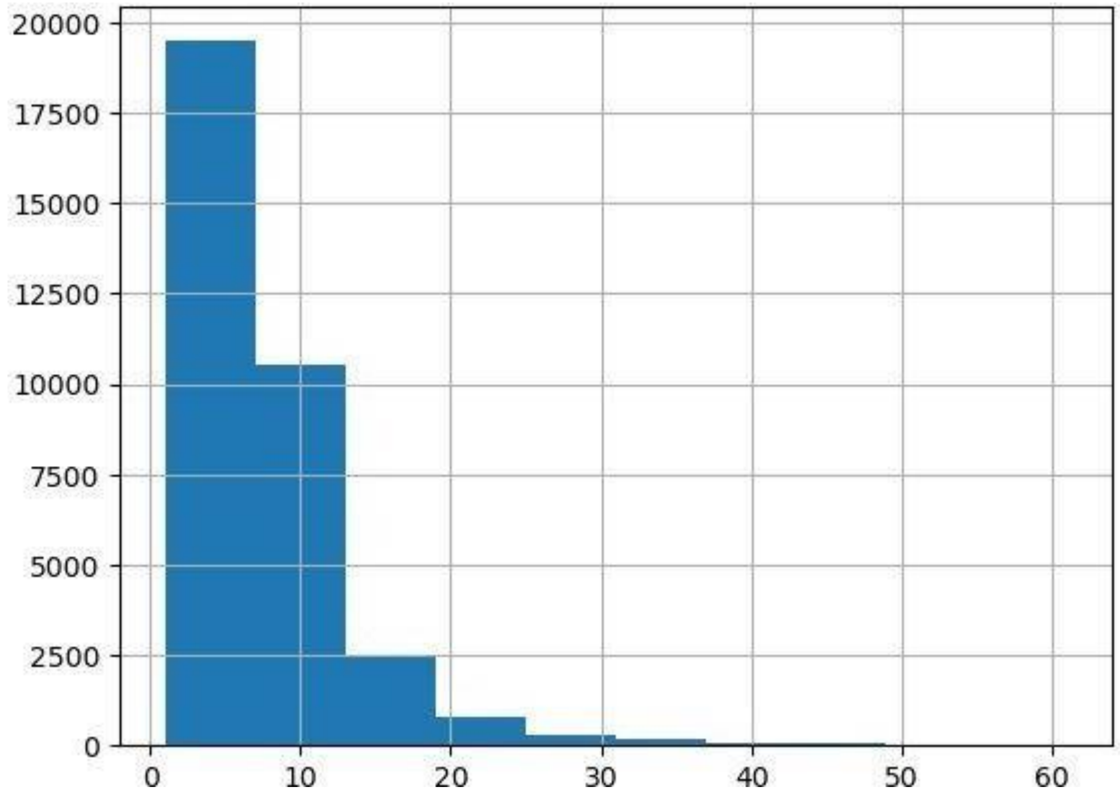
Course counts per genre

- The barchart to the right shows the count of courses per each topic. On the x-axis are the names of each genre while the count of courses in this genre are on the y-axis.
- Note that some courses can have multiple genres related to them (ie. a course in backend dev can feature information related to cloud computing)



Course enrollment distribution

-The histogram to the right shows the enrollment distributions.
- The x-axis shows the number of courses users are enrolled in while the y-axis shows the amount of users. It is noticeable that most users are enrolled in approximately 7 courses but there are those that take more. Some outliers take up to 60.



20 most popular courses

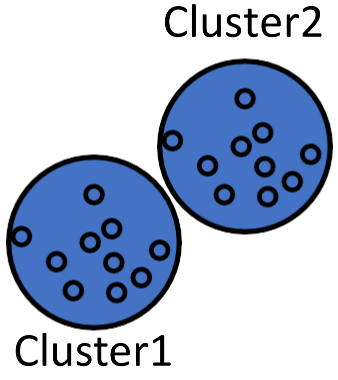
- The table to the right shows the top 20 courses as well as the amount of users enrolled

14936	python for data science
14477	introduction to data science
13291	big data 101
10599	hadoop 101
8303	data analysis with python
7719	data science methodology
7644	machine learning with python
7551	spark fundamentals i
7199	data science hands on with open source tools
6719	blockchain essentials
6709	data visualization with python
6323	deep learning 101
5512	build your own chatbot
5237	r for data science
5015	statistics 101
4983	introduction to cloud
4480	docker essentials a developer introduction
3697	sql and relational databases 101
3670	mapreduce and yarn
3624	data privacy fundamentals

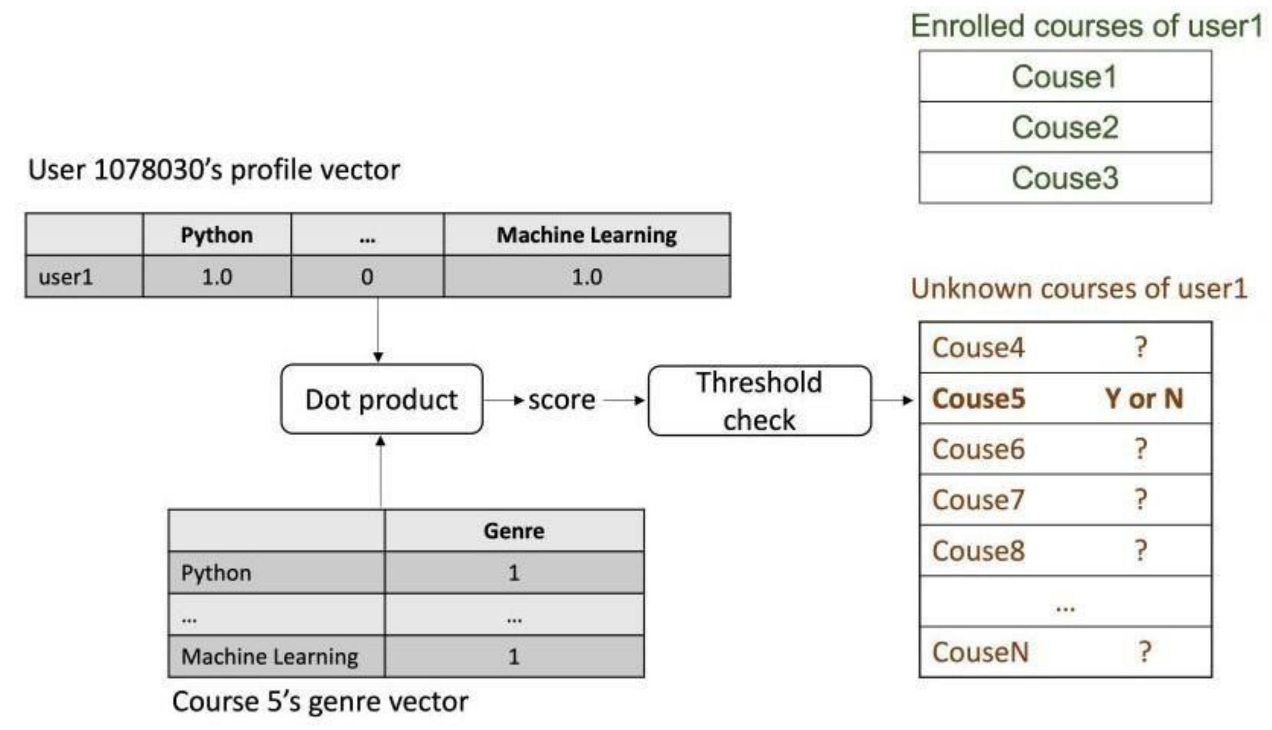
Word cloud of course titles

- The word cloud to the right shows a word cloud made from the names of the courses. The larger the font of the word, the more it appears in the corpus. It becomes clear that the company focuses on the use of python in data science, machine learning and and data in general.

Content-based Recommender System using Unsupervised Learning



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



Content-based recommender systems use the course content (ie genre, tags) that the user has already liked(or disliked) to find the best courses that the user has not yet completed but should be similar to the courses already taken.

The process is based on taking the dot product of the genres of each course possible (not taken) and the profile vector containing each of the genres the user liked.

In the context of this analysis we would presume that if an user that liked database courses with SQL and database management system, other courses based on database analysis might be interesting to the user.

The problems with this recommendation system is that it can not provide recommendations on courses that the share no genre that the user experienced.

Content-based recommender system using user profile and course genres

Find an example of an output of a content-based recommender system to the right.

```
users, courses, scores = generate_recommendation_scores()
res_dict['USER'] = users
res_dict['COURSE_ID'] = courses
res_dict['SCORE'] = scores
res_df = pd.DataFrame(res_dict, columns=['USER', 'COURSE_ID', 'SCORE'])
# Save the dataframe
#res_df.to_csv("profile_rs_results.csv", index=False)
res_df
```

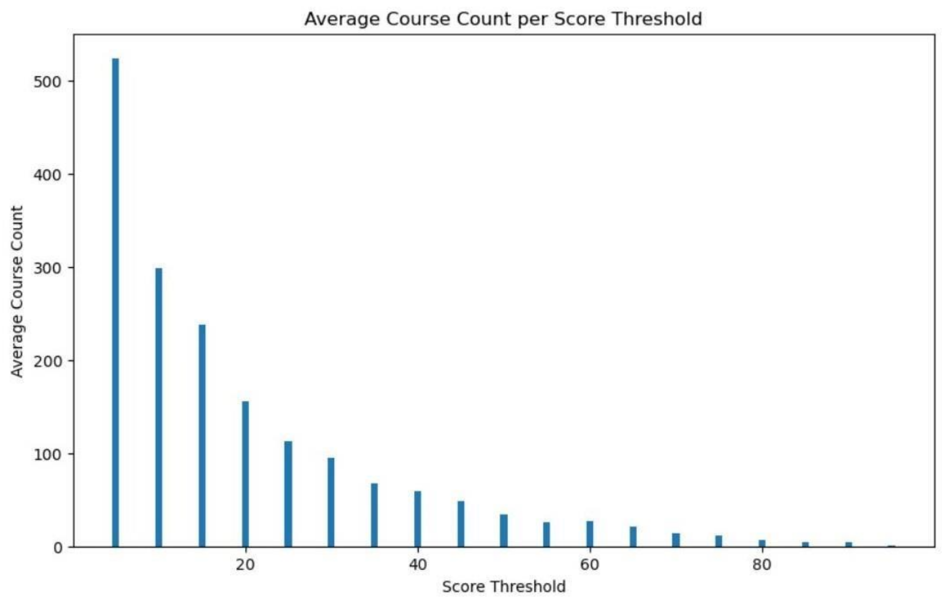
	USER	COURSE_ID	SCORE
0	37465	RP0105EN	27.0
1	37465	GPXX06RFEN	12.0
2	37465	CC0271EN	15.0
3	37465	BD0145EN	24.0
4	37465	DE0205EN	15.0
...	...	...	...
53406	2087663	excourse88	15.0
53407	2087663	excourse89	15.0
53408	2087663	excourse90	15.0
53409	2087663	excourse92	15.0
53410	2087663	excourse93	15.0

Evaluation results of user profile-based recommender system

Place your hyper-parameter settings, such as recommendation score or course similarity thresholds, etc.

Note: if you have tried multiple hyper-parameters, you may group and show all results in a grouped bar chart

A range from 0 to 101 (by 5) was made using the range function. This was used as the similarity threshold for the recommender system created. In the bar-plot below notice how the average number of recommended courses approaches zero.



Evaluation results of user profile-based recommender system

Place your hyper-parameter settings, such as recommendation score or course similarity thresholds, etc.

Note: if you have tried multiple hyper-parameters, you may group and show all results in a grouped bar chart

The most frequently recommended courses can be found in the table below:

```
introduction to data science in python                                        1011
accelerating deep learning with gpu                                            887
applied machine learning in python                                             845
data analysis using python                                                     632
text analytics at scale                                                        608
machine learning with python                                                   571
text analytics 101                                                             563
data science in insurance  basic statistical analysis                          548
using r with databases                                                         545
exploratory data analysis for machine learning                                 538
performing database operations in the cloudant dashboard                       533
sql for data science capstone project                                          533
sql for data science                                                           533
cloud computing applications  part 2  big data and applications in the cloud   524
analyzing big data with sql                                                    516
foundations for big data analysis with sql                                     516
getting started with the data  apache spark makers build                       512
analyzing big data in r using apache spark                                     501
spark overview for scala analytics                                             482
data science bootcamp with python for university professors  advance           464
Name: TITLE, dtype: int64
```

Flowchart of content-based recommender system using course similarity

- The chart below shows how a course similarity recommender system looks like:

- First a similarity measure is created for each of the observations (courses)
  - In the chart below it is the name of the course but usually other factors are used
- This similarity measure is then used in a calculation to find how similar the observations are
  - Note that this is usually given each course, meaning that course three would be analyzed based off of how similar it is to course one. This is repeated for each course afterwards

Course 1: “Machine Learning for Everyone”

	machine	learning	for	everyone	beginners
course1	1	1	1	1	0

Similarity Calculation:
Cosine, Euclidean, Jaccard index, ...

Course 2: “Machine Learning for Beginners”

	machine	learning	for	everyone	beginners
course2	1	1	1	0	1

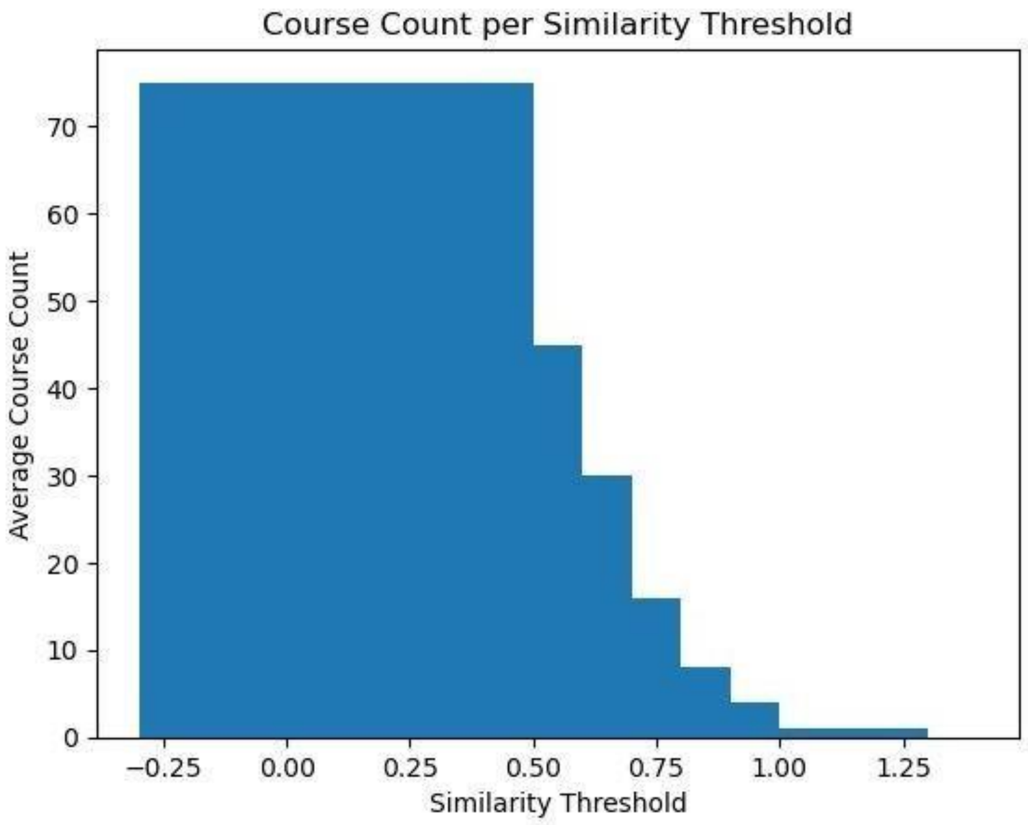
Evaluation results of course similarity based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

Note if you have tried multiple hyper-parameters, you may show your results in a grouped bar chart

Multiple similarity thresholds have been used. As in the previous recommender system, the larger the requirement, the less courses are recommended. For details find the table below:

	similarity_cutoff	count_courses
0	0.1	75.0
1	0.2	45.0
2	0.3	30.0
3	0.4	16.0
4	0.5	8.0
5	0.6	4.0
6	0.7	1.0
7	0.8	1.0
8	0.9	1.0
9	1.0	0.0



Evaluation results of course similarity based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

Note if you have tried multiple hyper-parameters, you may show your results in a grouped bar chart

Based off of the 0.5 similarity threshold, the courses to the right are the ones which would be suggested to the students which enjoyed “Machine Learning with python”.

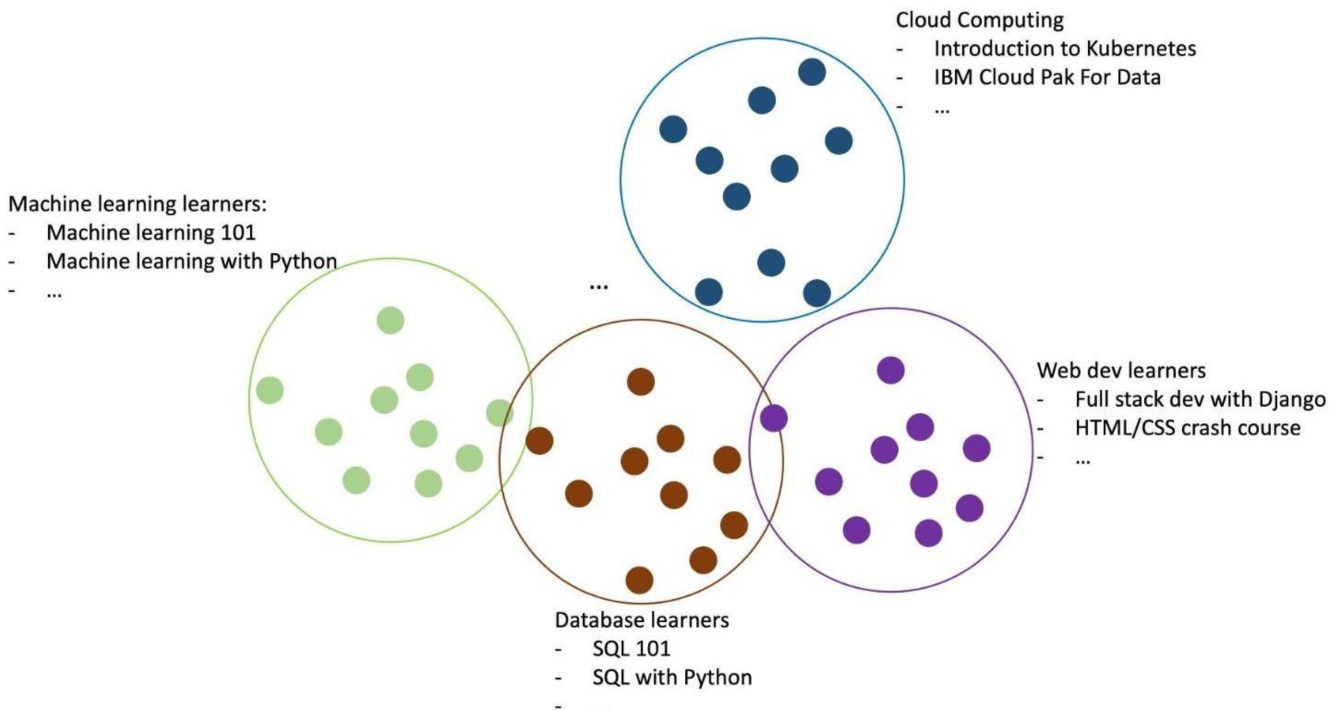
Out[68]:

	COURSE_ID	TITLE	DESCRIPTION
157	ML0109EN	machine learning dimensionality reduction	machine learning dimensionality reduction
158	ML0101ENv3	machine learning with python	machine learning can be an incredibly benefici...
200	ML0151EN	machine learning with r	this machine learning with r course dives into...
259	excourse46	machine learning	machine learning is the science of getting com...
260	excourse47	machine learning for all	machine learning often called artificial inte...
264	excourse51	introduction to machine learning in production	in the first course of machine learning engine...
273	excourse60	introduction to tensorflow for artificial inte...	if you are a software developer who wants to b...
282	excourse69	machine learning with big data	want to make sense of the volumes of data you ...

Click here for Hints

Clustering-based recommender system

- The methodology behind clustering-based recommender systems is to perform a clustering to obtain the segments. These are simply explained as customer segments (which can be later used for profiling).
- This algorithm is then ran on each new user to assign them a segment.
- Based off of the segment, the user receives recommendations.



Flowchart of clustering-based recommender system

20XX

20XX

20XX

20XX

Scaling

As the clustering algorithms are dependent on scaling, it is important to use a scaling methodology such as the minmaxscaler to make sure no feature is being made more important than the others

Dimensionality reduction

Clustering based algorithms (those that are not defined through neural nets) are susceptible to the curse of dimensionality. As such, the number of variables needs to be lowered

Cluster definition

As clustering is a unsupervised learning algorithm whose goal isto find the underlying structure of the dataset, there are many hyperparameters that can be used (ie number of clusters for k-means or the size with db-scan

Profiling/usage

In cases of recommender systems and given the dataset for this task, profiling can not beused. As such, the only item leftusing the algorithm for recommendation

Evaluation results of clustering-based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

Note if you have tried multiple hyper-parameters, you may show your results in a grouped bar chart

On average, the number of unseen courses recommended to the users was 64. The code used to show this is:

- course_count = recommendations_df.groupby("user")["course"].nunique()
- average_course_count = course_count.sum() / len(users_clusters_df)

Please note that this is with the cut off point of five. The issue arises when defining the limit based off of enrolment as the course that has the largest amount of enrollments within cluster 8, has only 5 enrollments. This was removed. Further analysis is needed to decide how to offer more courses to cluster 8.

Evaluation results of clustering-based recommender system

Your hyper-parameter settings, such as a score or similarity threshold

Note if you have tried multiple hyper-parameters, you may show your results in a grouped bar chart

This analysis was done per cluster as well as in total. The table to the right shows the output per cluster. The table below shows the top five courses recommended in total.

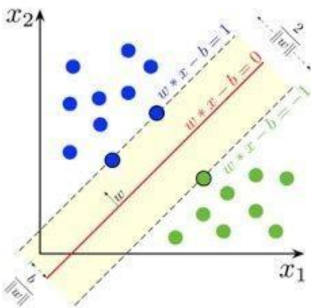
```
[77]: DS0101EN    23857
      BD0101EN    23490
      PY0101EN    21237
      BD0111EN    19819
      DS0103EN    19208
      Name: course, dtype: int64
```

```
top_courses_per_cluster2
```

	cluster	level_1	course
0	0	DS0101EN	4830
1	0	ML0115EN	4774
2	0	BD0101EN	4278
3	0	ML0101ENv3	4230
4	0	DS0103EN	4180
...	...	...	...
60	13	DS0105EN	4216
61	13	BD0101EN	4125
62	13	DS0103EN	4081
63	13	BD0111EN	4081
64	13	PY0101EN	4029

65 rows × 3 columns

Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system

The KNN algorithm was used from the surprise library. It is firstly defined using a variable. This variable is then fit with the training split. Predictions are made using the .test function fit with the testing data.

20XX

20XX

20XX

20XX

Dataset loading

The dataset was loaded using pandas read_csv

Train-test split

As the data was in the proper format, no cleaning was necessary. The data was loadeddivided via train-test split.

Cross Validation

The hyperparameters tuned werethe k (the number of nearest neighbours) and similarity metrics (pearson, and cosine simmilarity were used)

Evaluation

The RMSE of the model was .1.01284. The model with the best parameters had k of 30, pearson similarity, user-based

Flowchart of NMF based recommender system

The NMF algorithm was used from the surprise library. It is firstly defined using a variable. This variable is then fit with the training split. Predictions are made using the .test function fit with the testing data.

20XX

Dataset loading

The dataset was loaded using pandas read_csv

20XX

Train-test split

As the data was in the proper format, no cleaning was necessary. The data was loadeddivided via train-test split.

20XX

Evaluation

The RMSE of the NMF based recommender system was 0.2031

Flowchart of Neural Network Embedding based recommender system

20XX

Define user_embedding and item_embedding layers

These were preloaded using pandas

20XX

Encode layers

The layers were encoded usingthe dot product

20XX

Define algorithm

The algorithm used was RecommenderNet with the adam optimizer. The loss and optimization metrics were defined as rmse

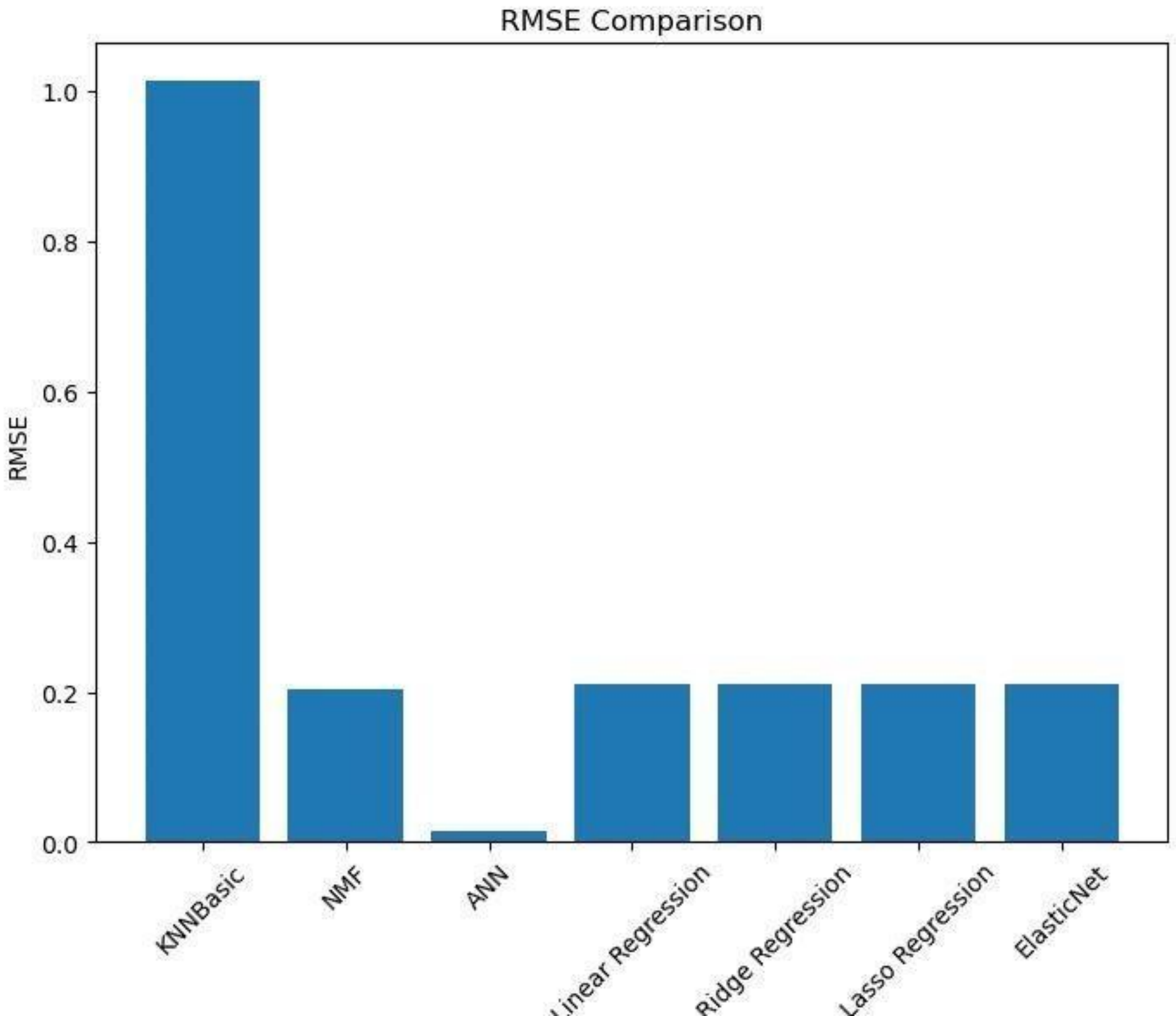
20XX

Evaluate

Training after 10 epocs resultedin an RMSE of 0.0143 on the testing set

Compare the performance of collaborative-filtering models

Observing the results of the collaborative-filtering models it becomes clear that the absolute winner is the artificial neural network. These models are indeed superior in their predictory power. Outside of KNNBasic performing the worst, the other algorithms had a similar performance. Other algorithms are not shown here asa classification algorithms can not use RMSE as a metric. Nevertheless, given F1 score, the ElasticNet performed the best.



Conclusions

- Given content based course recommender systems, the exact score threshold will define how much courses the user will be recommended. It is very common to try to keep that number low to make sure the user is not overwhelmed by the choices as this lowers the chance of upselling.
- Given content based course recommender systems, the number of suggested courses also depends on the cutoff point for the similarity measure. The results depend on the exact similarity metric being used.
- Clustering based recommender system could also be used to find out more about the segment itself. Sadly, further information on the users is not available and this is as such, outside the scope of the analysis
- For the purpose of collaborative filtering, the artificial neural networks perform best. Their training time and low explainability could however prove as a problem.

Appendix

- Github