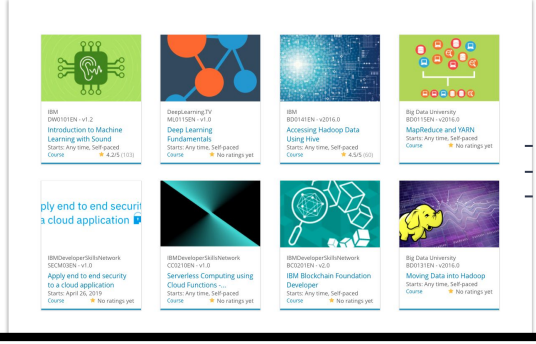
Build a Personalized Online Course Recommender System with Machine Learning

Liam Webster
7/21/2022



Outline

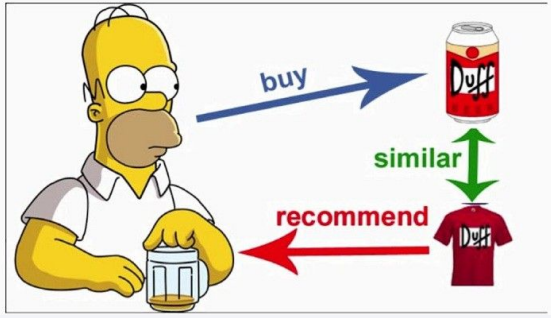
- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion
- Appendix

Introduction

- In this project, implementations of recommender systems for educational courses are analyzed. Specifically the project deals with recommending Computer Science related courses.

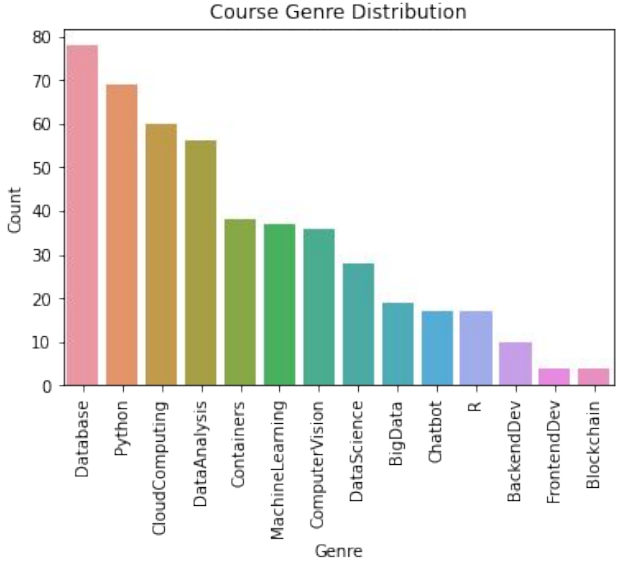
  “Man I really enjoyed that course! What course should I take now?”

- This is the question that will be answered through the development of our recommender system. If sample course data is thoroughly analyzed through EDA and models are built then students will have a resource which recommends tailored interesting courses.



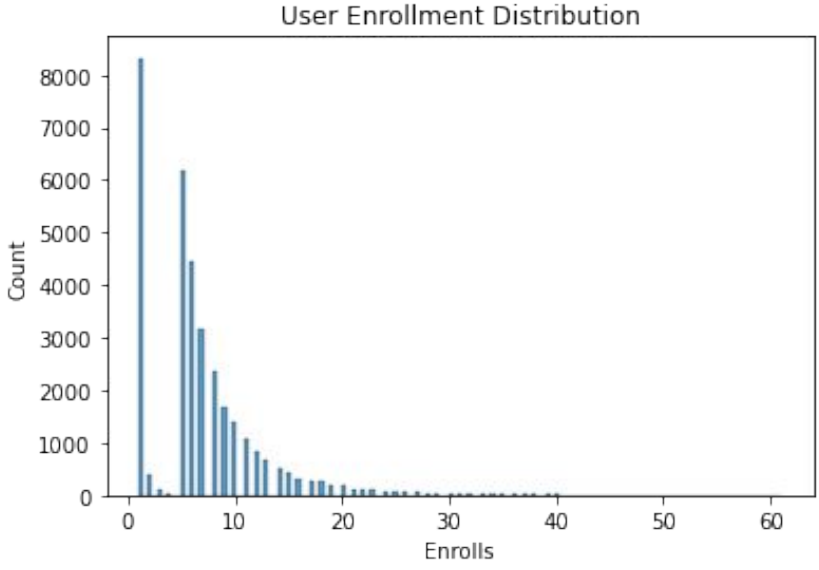
Exploratory Data Analysis

Course counts per genre



This bar chart shows the breakdown of course genre distribution. From this chart we can see that the “Database” genre has the most courses while the “blockchain” genre has the least courses.

Course enrollment distribution



This User Enrollment histogram displays the distribution of number of enrolled courses per student. It is clear that a large number of students enroll in just one course. Then there is an up shoot in the distribution at around 5 courses that exponentially decreases.

20 most popular courses

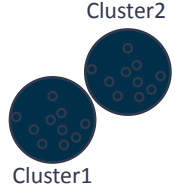
	TITLE	ENROLLS
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199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

This dataframe clearly displays the top enrolled courses aka the most popular courses. With “python for data science” being the most popular course.

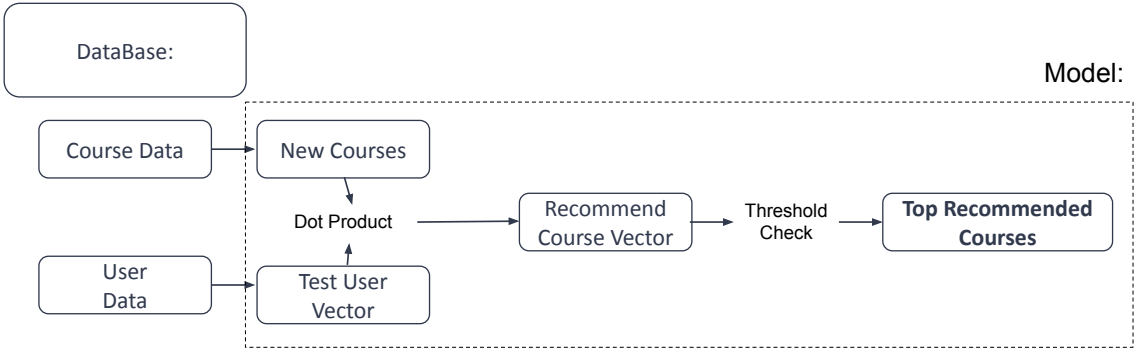
Word cloud of course titles

This word cloud easily and briefly describes the the most frequently used keywords in our course titles and descriptions. With “data”, “python”, and “machine learning” being of the most frequently used words.

Content-based Recommender System using Unsupervised Learning



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



We start with a course dataframe of which is composed of course genre vectors and a user dataframe of which is composed of user vectors. To recommend courses to a user, a dot product is taken between the respective user vector and untaken course matrix. Of which produces a course recommendation vector. Courses that meet the recommendation threshold are outputted.

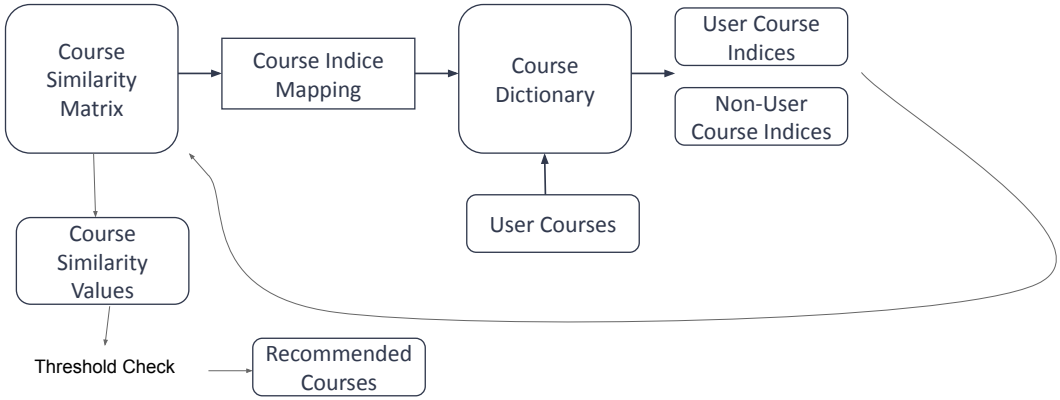
Evaluation results of user profile-based recommender system

Frequently Recommended Courses:

Course ID:	Course Title:	Total Recommendations:
TA0106EN	text analytics at scale	608
GPXX0IBEN	data science in insurance basic statistical a...	548
excourse22	introduction to data science in python	547
excourse21	applied machine learning in python	547
ML0122EN	accelerating deep learning with gpu	544
excourse06	sql for data science capstone project	533
excourse04	sql for data science	533
GPXX0TY1EN	performing database operations in the cloudant...	533
excourse31	cloud computing applications part 2 big data…	524
excourse73	analyzing big data with sql	516

On average about 62 courses were recommended to each user with a recommendation threshold of 10.

Flowchart of content-based recommender system using course similarity



First the course similarity matrix–the heart of this model–was imported. Then a course dictionary was created which mapped each course to its respective indices of the course similarity matrix. From there a users current courses where used to find similar courses via iterating through the similarity matrix.

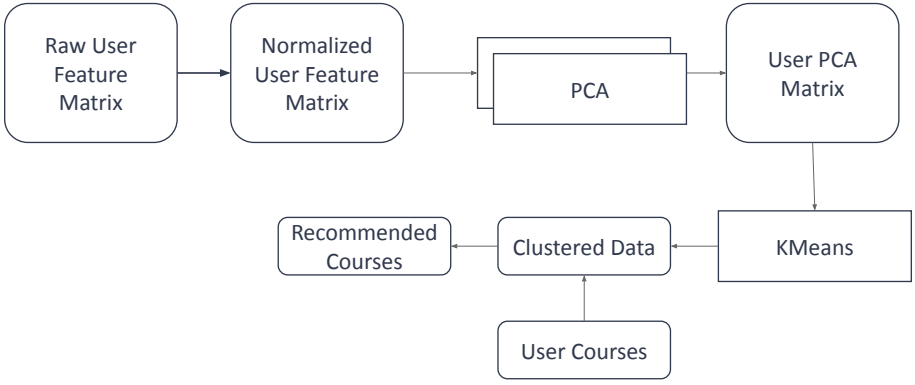
Evaluation results of course similarity based recommender system

Frequently Recommended Courses:

Course ID:	Course Title:	Total Recommendations:
excourse62	introduction to data science in python	579
excourse22	introduction to data science in python	579
DS0110EN	data science with open data	562
excourse65	data science fundamentals for data analysts	555
excourse63	a crash course in data science	555
excourse72	foundations for big data analysis with sql	551
excourse68	big data modeling and management systems	550
excourse74	fundamentals of big data	539
excourse67	introduction to big data	539
BD0145EN	sql access for hadoop	506

On average about 12 courses were recommended to each user with a recommendation threshold of 0.6.

Flowchart of clustering-based recommender system



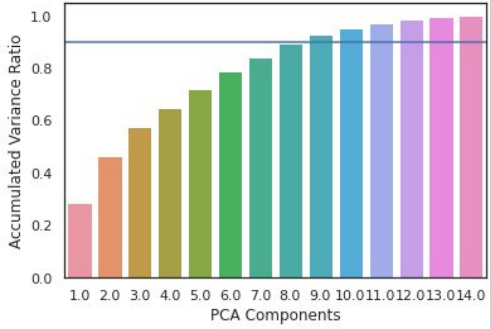
First the course feature matrix is normalized. Next PCA was performed on the matrix reducing the dimensionality to 16. A basic KMeans algorithm was performed on the matrix. From this courses with similar features were clustered together. Users were recommended via their current course clustering.

Evaluation results of clustering-based recommender system

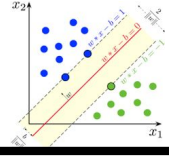
Frequently Recommended Courses:

Course ID:	Course Title:	Total Recommendations:
DS0101EN	introduction to data science	147
BD0101EN	big data 101	97
ML0115EN	deep learning 101	64
BD0111EN	hadoop 101	54
PY0101EN	python for data science	50
DV0101EN	data visualization with python	42
ML0101ENv3	big data modeling and management systems	41
BD0115EN	machine learning with python	38
CO0101EN	docker essentials a developer introduction	32
DA0101EN	data analysis with python	28

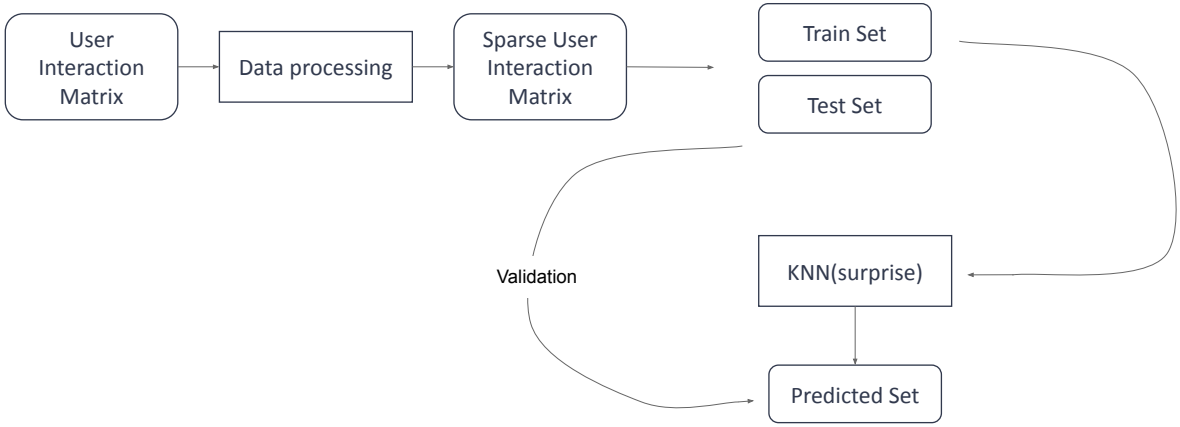
The final model consisted of 20 clusters. On average about 2 courses were recommended to each user.



Collaborative-filtering Recommender System using Supervised Learning

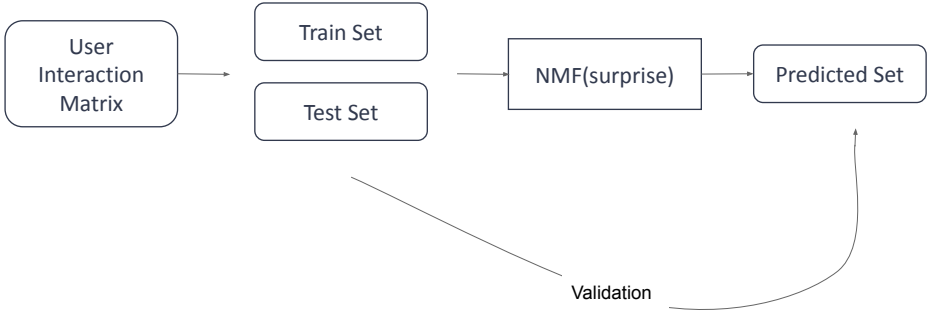


Flowchart of KNN based recommender system



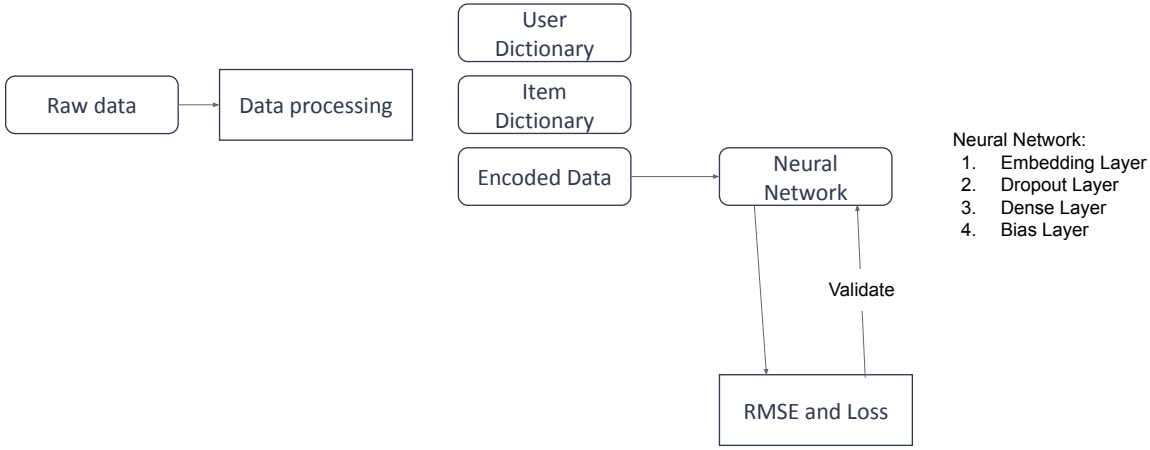
After importing the User-Interaction matrix it is pivoted to create a sparse User-Interaction matrix. The matrix is then split into train and test sets. With these data sets our KNN model is fit and validated all though the Surprise library.

Flowchart of NMF based recommender system



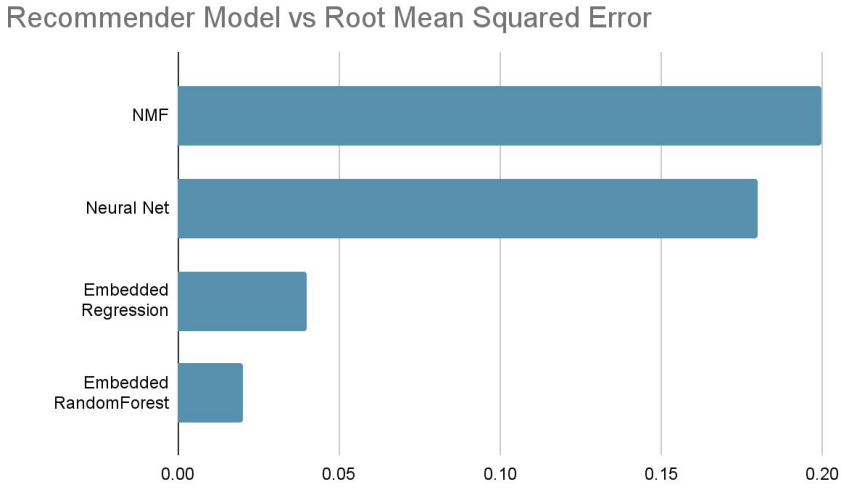
After importing the User Interaction matrix, it was split into a 75:25 train test split. The train set was then fit on the NMF model which decomposes the matrix into two latent user feature and item feature matrices. Which then are used to calculate prediction on the test set which was then validated in accordance with the ground truth.

Flowchart of Neural Network Embedding based recommender system



After importing the raw data it was put through some major data processing. The raw data was encoded and indice dictionaries were created along with. This encoded data was fed into the engineered neural network. Which iterated through the data learning the latent space similar to NMF. The neural network was validated using RMSE and tweaked until an optimal result was acquired.

Compare the performance of collaborative-filtering models



This is a plot of the best resulting supervised models and their respective RMSE values. As displayed in the plot both embedded models did exceptionally well.

Conclusions

- Supervised vs Unsupervised Learning: while the unsupervised models built were very intuitive and interpretable the supervised embedded models built were very accurate.
- NMF and a Neural Net only require a user interaction matrix.
- The clustering models are very memory intensive and will only get more intensive as further data is added.

Appendix

- GitHub: https://github.com/liamwebsterreal/IBM_Course_Projects