Machine Learning Capstone Project
Recommender System for Online Courses

Charles D. Milligan, Ph.D.

April 6, 2023

Outline

Introduction and Background

Exploratory Data Analysis

Content-based Recommender System using Unsupervised Learning

Collaborative-filtering based Recommender System using Supervised learning

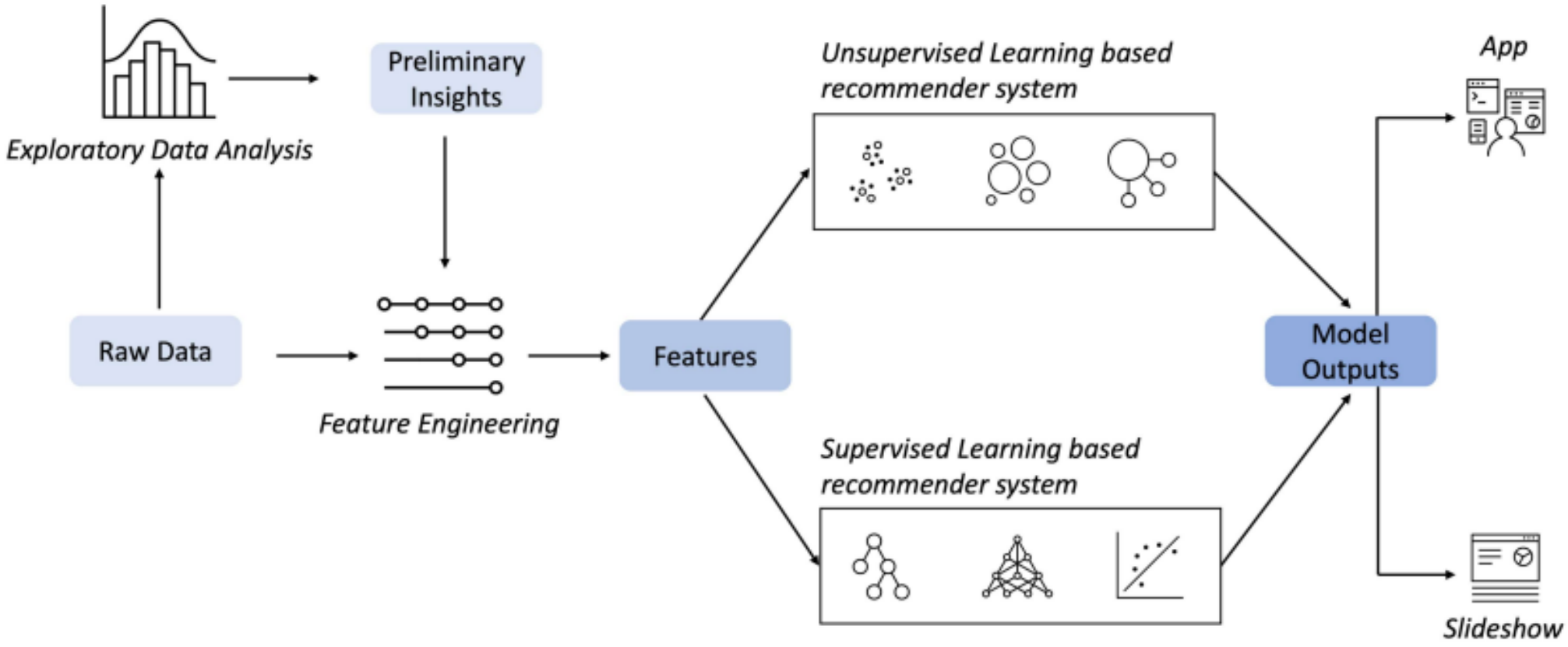
Conclusion

Appendix

Introduction

- The goal of this project is to develop a recommender system that will help students find new courses aligned with their interests and learning goals.
- The enhanced user experience will result in increased enrollments, new users and increased revenue for the company.
- **Hypothesis**: we can use data from user's previous enrollments and data on course characteristics to identify new courses that would be of interest to the user.

Technical Approach



Data Sources: Course Genres

course_genre.csv
307 courses
16 features

```
COURSE_ID          object
TITLE              object
Database            int64
Python              int64
CloudComputing      int64
DataAnalysis        int64
Containers          int64
MachineLearning     int64
ComputerVision      int64
DataScience         int64
BigData             int64
Chatbot             int64
R                   int64
BackendDev          int64
FrontendDev         int64
Blockchain          int64
dtype: object
```

	COURSE_ID	TITLE	Database	Python	CloudComputing	DataAnalysis	Containers	MachineLearning	ComputerVision	DataScie
0	ML0201EN	robots are coming build iot apps with watson ...	0	0	0	0	0	0	0	
1	ML0122EN	accelerating deep learning with gpu	0	1	0	0	0	1	0	
2	GPXX0ZG0EN	consuming restful services using the reactive ...	0	0	0	0	0	0	0	
3	RP0105EN	analyzing big data in r using apache spark	1	0	0	1	0	0	0	

Data Sources: Course Ratings

ratings.csv

233306 enrollments

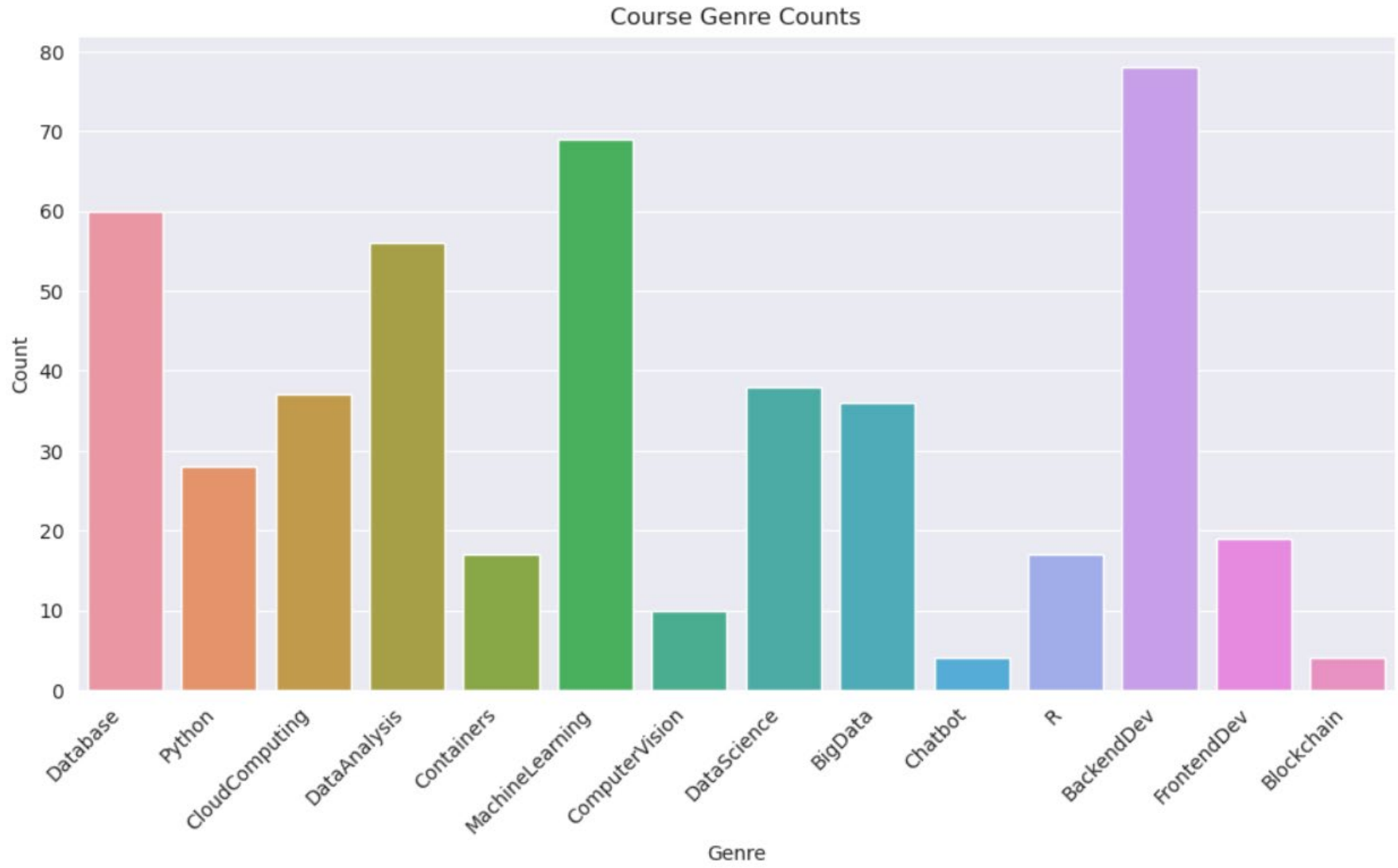
33901 users

- 0.0 – No interaction
- 1.0 – Browsed Course
- 2.0 – Audited Course
- 3.0 – Completed Course

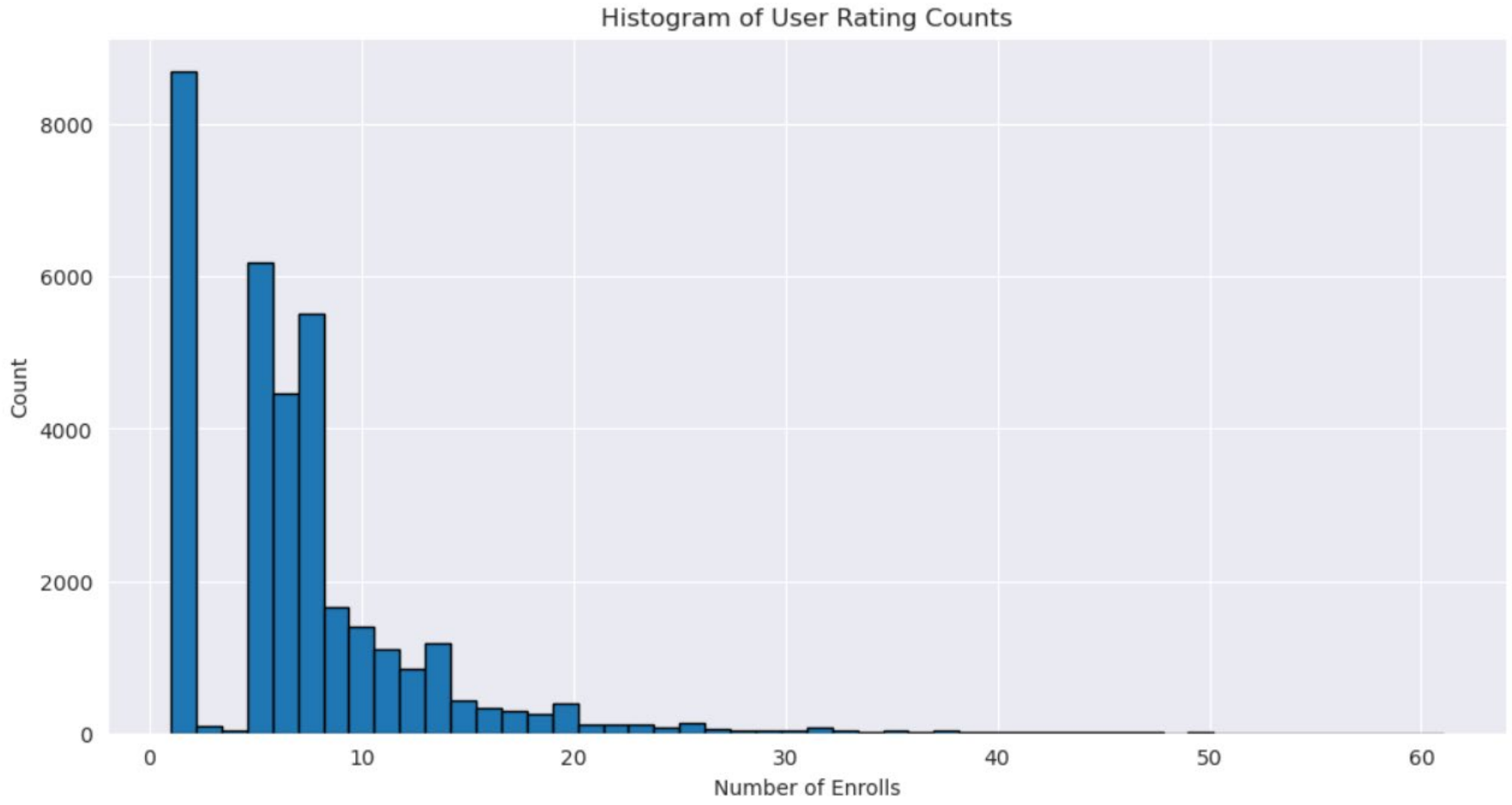
	user	item	rating
0	1889878	CC0101EN	3.0
1	1342067	CL0101EN	3.0
2	1990814	ML0120ENv3	3.0
3	380098	BD0211EN	3.0
4	779563	DS0101EN	3.0

Exploratory Data Analysis

Course counts per genre



Course enrollment distribution

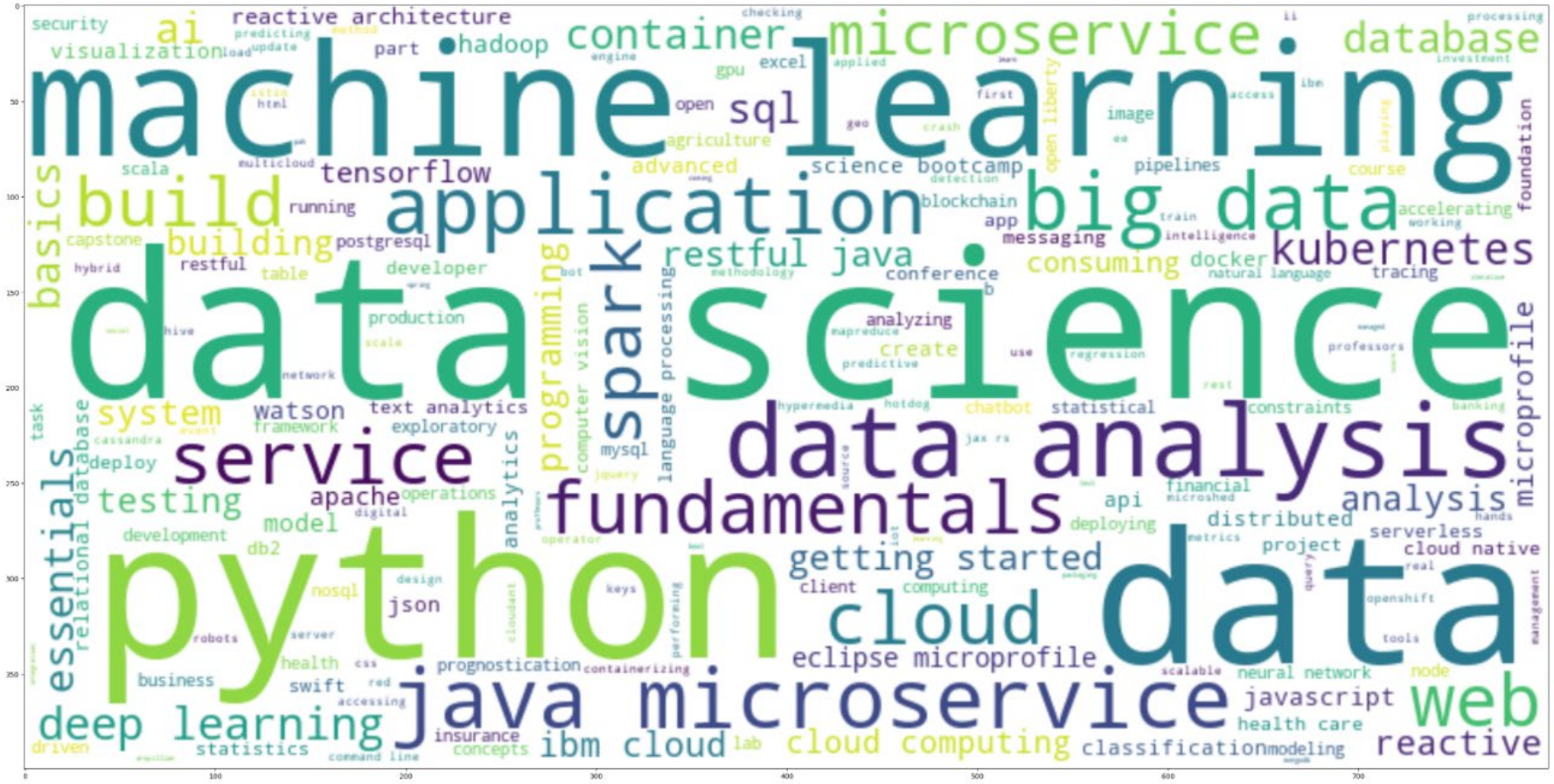


20 Most Popular Courses

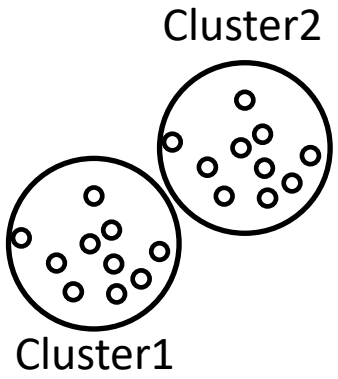
Top 20 Courses by Enrollment:

item	
PY0101EN	14936
DS0101EN	14477
BD0101EN	13291
BD0111EN	10599
DA0101EN	8303
DS0103EN	7719
ML0101ENv3	7644
BD0211EN	7551
DS0105EN	7199
BC0101EN	6719
DV0101EN	6709
ML0115EN	6323
CB0103EN	5512
RP0101EN	5237
ST0101EN	5015
CC0101EN	4983
CO0101EN	4480
DB0101EN	3697
BD0115EN	3670
DS0301EN	3624

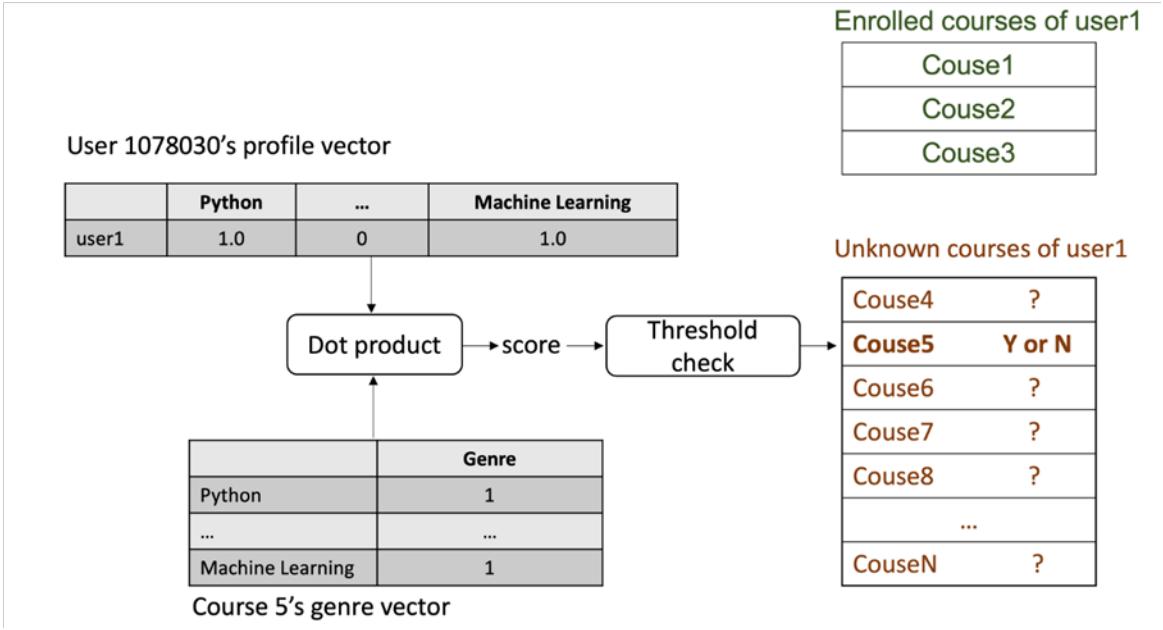
Word Cloud of **Course** Titles



Content-based Recommender System using Unsupervised Learning



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



Evaluation Results of User Profile-Based Recommender System

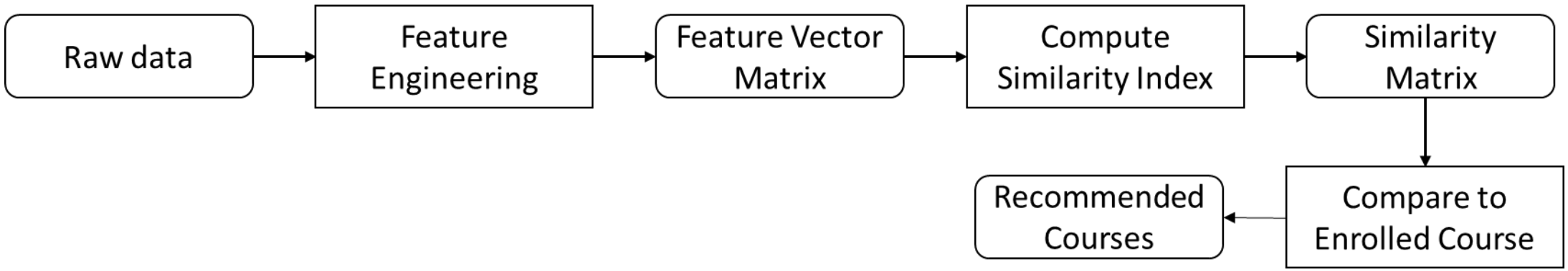
5.7 Courses/User

	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

Top 10 Recommended
BD0101EN
BD0211EN
DS0101EN
PY0101EN
BC0101EN
CB0103EN
LB0101ENv1
CC0101EN
RP0101EN
CO0101EN

Score Threshold = 10.0

Flowchart of Content-based Recommender System Using Course Similarity

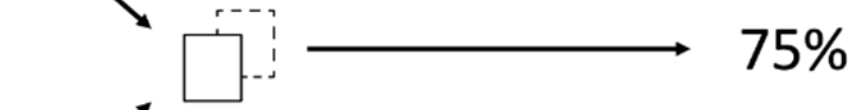


Course 1: "Machine Learning for Everyone"

	machine	learning	for	everyone	beginners
course1	1	1	1	1	0

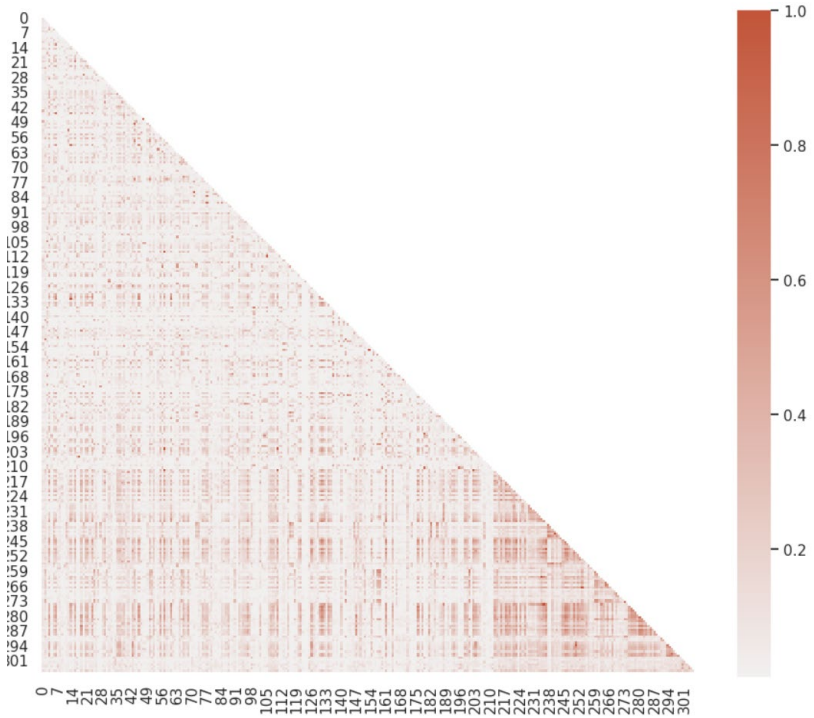
Course 2: "Machine Learning for Beginners"

	machine	learning	for	everyone	beginners
course2	1	1	1	0	1



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

Content-Based Recommender System Using Course Similarity



	0	1	2	3	4	
0	1.000000	0.088889	0.088475	0.065556	0.048810	0.1
1	0.088889	1.000000	0.055202	0.057264	0.012182	0.0
2	0.088475	0.055202	1.000000	0.026463	0.039406	0.0
3	0.065556	0.057264	0.026463	1.000000	0.000000	0.2
4	0.048810	0.012182	0.039406	0.000000	1.000000	0.0
...	...	...	...	...	...	

Course Similarity Matrix and Heatmap

Evaluation Results of Course Similarity Based Recommender System

5000 Recommendations (5/User)

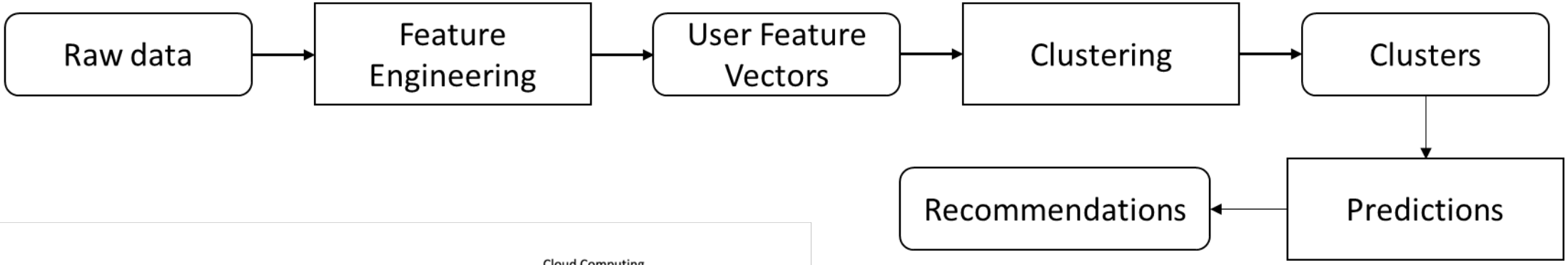
	USER	COURSE_ID	SCORE
0	37465	excourse67	0.708214
1	37465	excourse72	0.652535
2	37465	excourse74	0.650071
3	37465	BD0145EN	0.623544
4	37465	excourse68	0.616759
...	...	...	...
4995	2087663	excourse67	0.708214
4996	2087663	excourse72	0.652535
4997	2087663	excourse74	0.650071
4998	2087663	BD0145EN	0.623544
4999	2087663	excourse68	0.616759

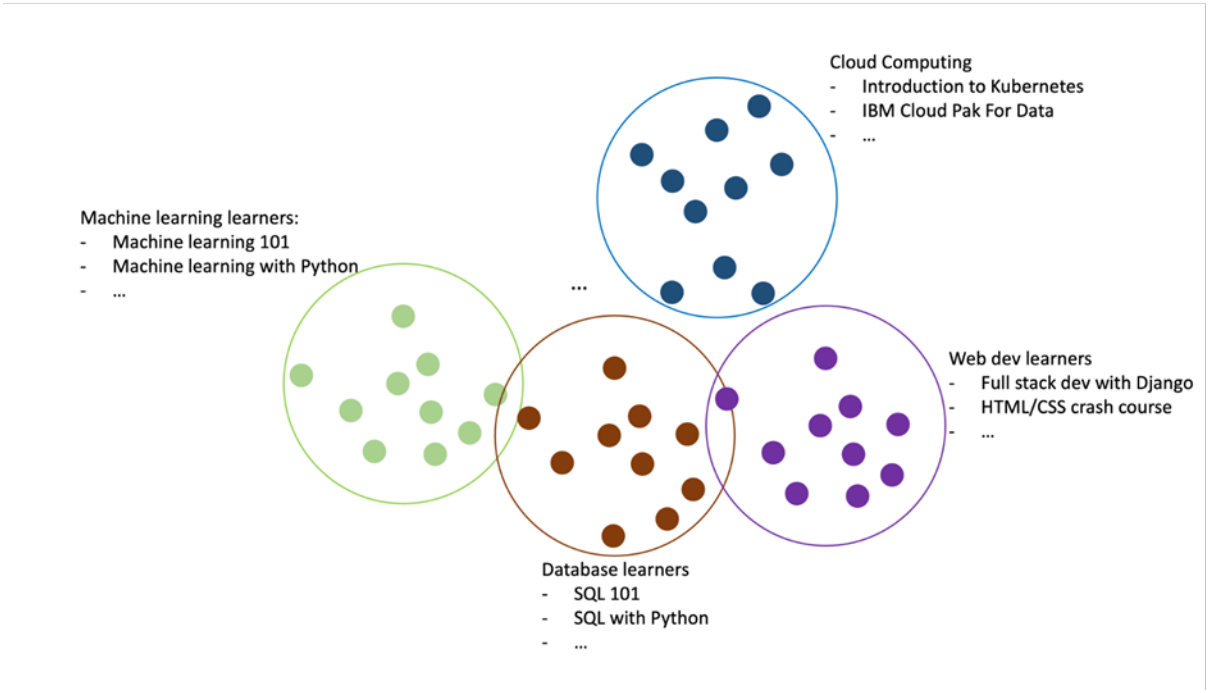
Most Frequently Recommended Courses

```
excourse67    1000
excourse72    1000
excourse74    1000
BD0145EN      1000
excourse68    1000
Name: COURSE_ID, dtype: int64
```

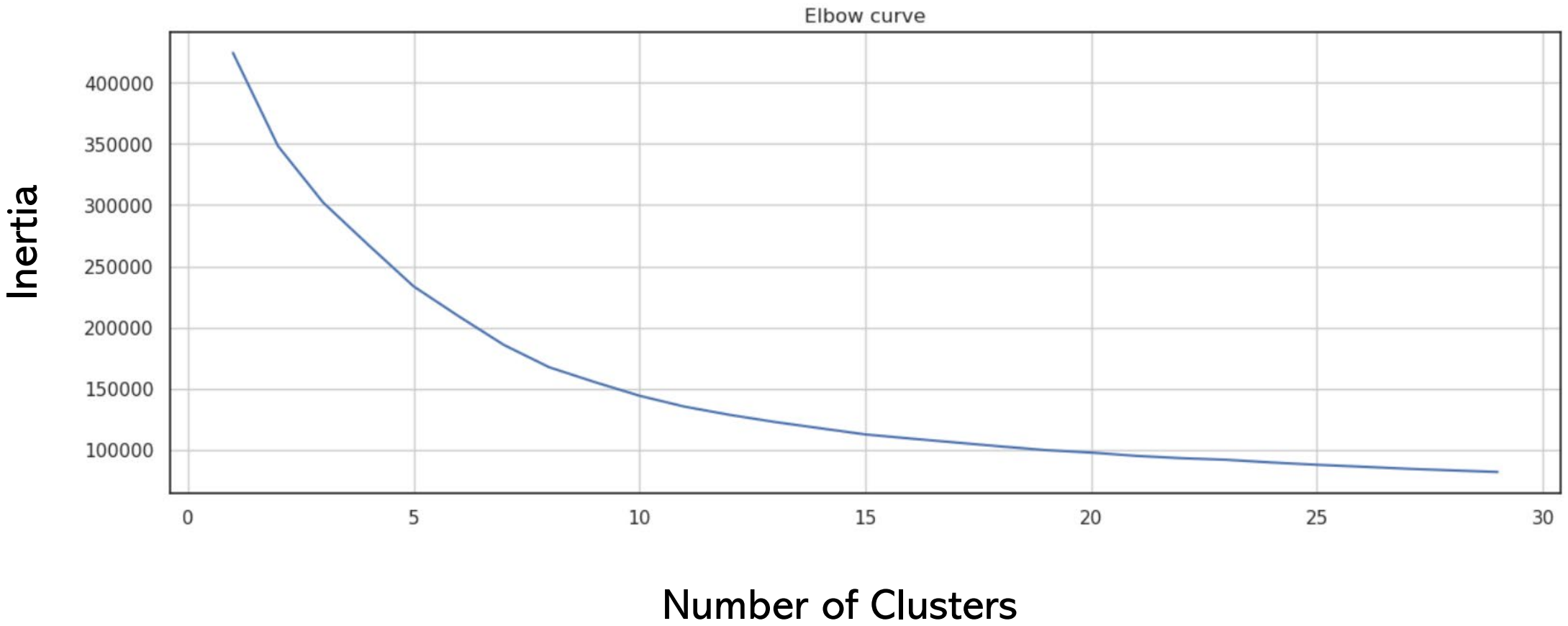
Course Similarity Score Threshold = 0.60

Flowchart of Clustering-Based Recommender System

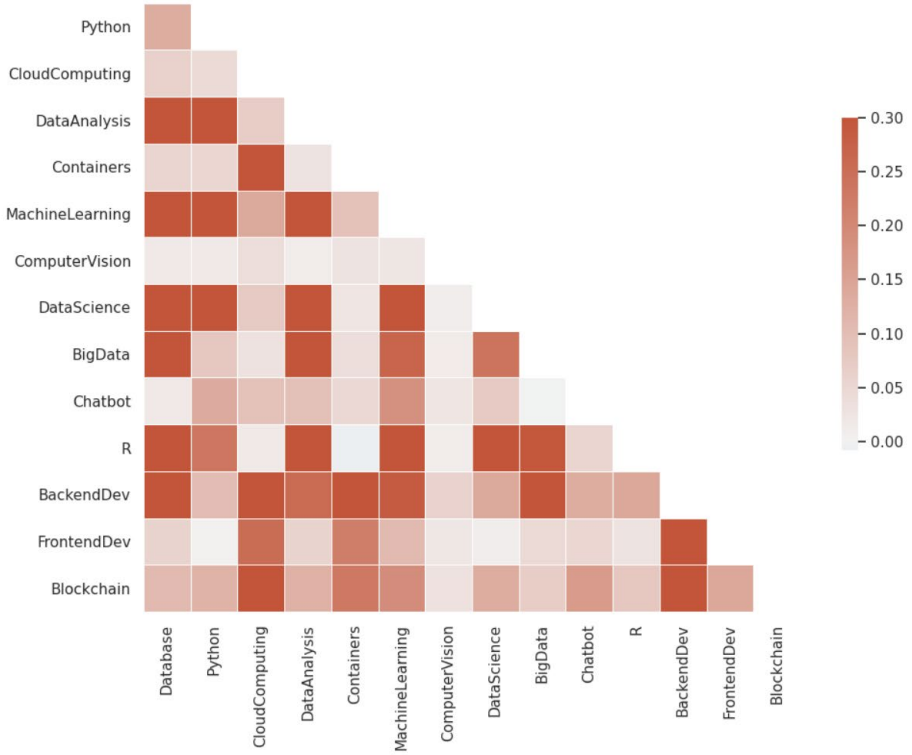


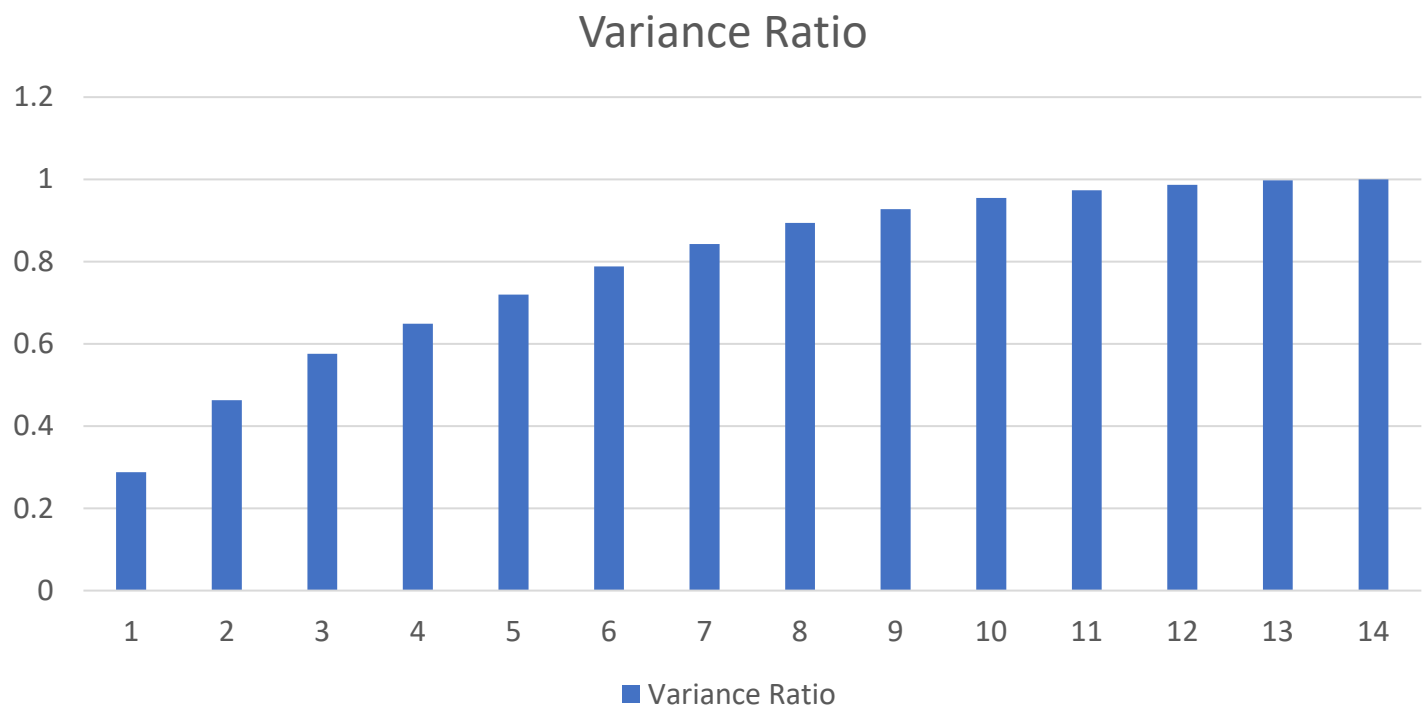


Clustering-Based Recommender System



Covariance Matrix and Principal Components





Evaluation Results of Clustering-Based Recommender System

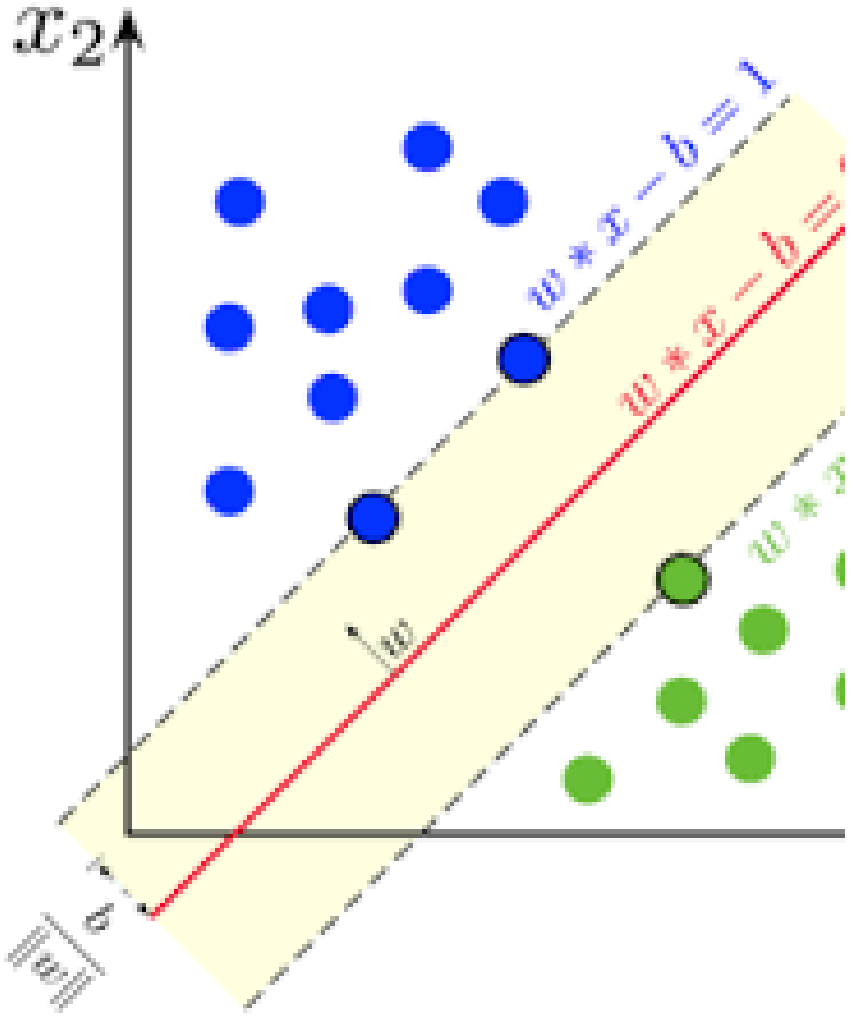
hyper-parameter settings; k=25, Number of Principal Components = 9

3 courses recommended per user

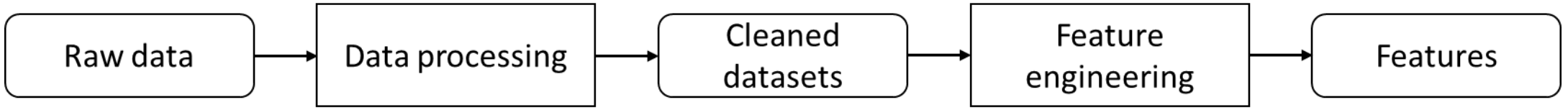
```
user in cluster 0 will be sugessted 3 courses as ['BD0101EN' 'BD0111EN' 'BD0211EN']
user in cluster 1 will be sugessted 3 courses as ['PY0101EN' 'DS0101EN' 'DA0101EN']
user in cluster 2 will be sugessted 3 courses as ['BC0101EN' 'BD0101EN' 'DS0101EN']
user in cluster 3 will be sugessted 3 courses as ['CO0201EN' 'CO0101EN' 'CO0301EN']
user in cluster 4 will be sugessted 3 courses as ['DS0101EN' 'RP0101EN' 'DS0103EN']
user in cluster 5 will be sugessted 3 courses as ['CB0103EN' 'CO0101EN' 'BC0101EN']
user in cluster 6 will be sugessted 3 courses as ['LB0101ENv1' 'LB0105ENv1' 'LB0103ENv1']
user in cluster 7 will be sugessted 3 courses as ['BD0211EN' 'BD0101EN' 'DS0101EN']
user in cluster 8 will be sugessted 3 courses as []
user in cluster 9 will be sugessted 3 courses as ['PY0101EN' 'DA0101EN' 'DV0101EN']
user in cluster 10 will be sugessted 3 courses as ['BD0111EN' 'BD0211EN' 'BD0115EN']
user in cluster 11 will be sugessted 3 courses as ['PY0101EN' 'DS0101EN' 'ML0101ENv3']
user in cluster 12 will be sugessted 3 courses as ['DS0101EN' 'BD0101EN' 'PY0101EN']
user in cluster 13 will be sugessted 3 courses as ['BC0101EN' 'BC0201EN' 'PY0101EN']
user in cluster 14 will be sugessted 3 courses as ['BD0111EN' 'BD0101EN' 'BD0211EN']
user in cluster 15 will be sugessted 3 courses as ['CB0103EN' 'DS0101EN' 'BD0101EN']
user in cluster 16 will be sugessted 3 courses as ['BD0111EN' 'PY0101EN' 'BD0211EN']
user in cluster 17 will be sugessted 3 courses as ['RP0101EN' 'DS0101EN' 'DS0103EN']
user in cluster 18 will be sugessted 3 courses as ['PY0101EN' 'CB0103EN' 'DS0101EN']
user in cluster 19 will be sugessted 3 courses as ['CB0103EN' 'BC0101EN' 'PY0101EN']
user in cluster 20 will be sugessted 3 courses as ['CB0103EN' 'PY0101EN' 'DA0101EN']
user in cluster 21 will be sugessted 3 courses as ['BD0111EN' 'BD0101EN' 'DS0103EN']
```

Top 10 Recommended
BD0101EN
BD0211EN
DS0101EN
PY0101EN
BC0101EN
CB0103EN
LB0101ENv1
CC0101EN
RP0101EN
CO0101EN

Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system

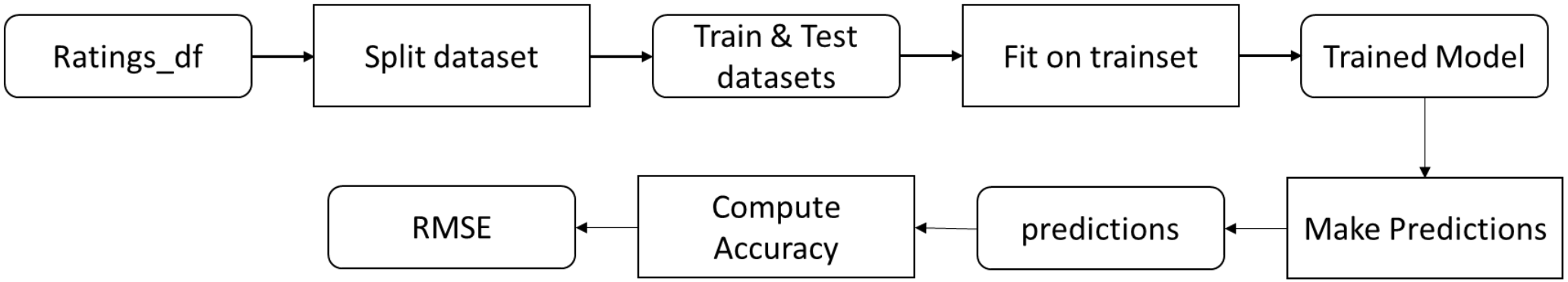


User-Item interaction matrix

Similar users

	Machine Learning With Python	Machine Learning 101	Machine Learning Capstone	SQL with Python	Python 101
...	...	...	...	...	
user2	3.0	3.0	3.0	3.0	3.0
user3	2.0	3.0	3.0	2.0	
user4	3.0	3.0	2.0	2.0	3.0
user5	2.0	3.0	3.0		
user6	3.0	3.0	?		3.0
...	...	...	...	...	

Flowchart of NMF Based Recommender System



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

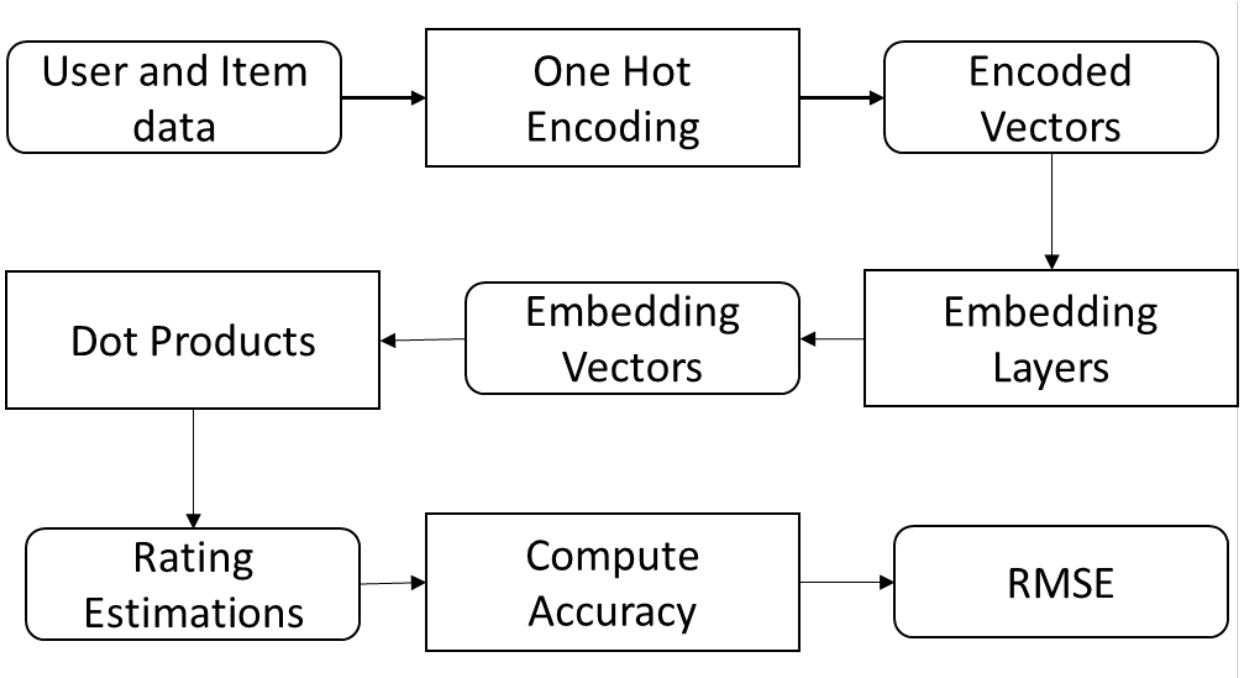
	feature1	...	feature16
user1	...	...	...
user2	...	...	...
user3	...	...	...
user4	...	...	...
...	...	...	...
...	...	...	...
user6	...	...	...

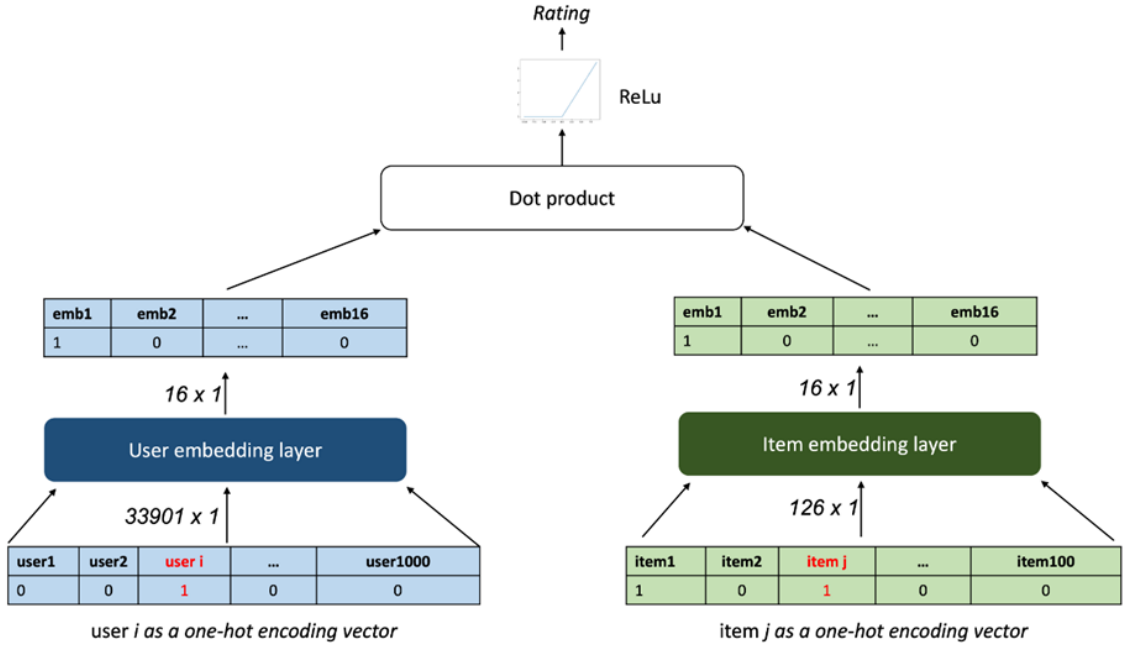
X

Item matrix: **I** 16 x 100

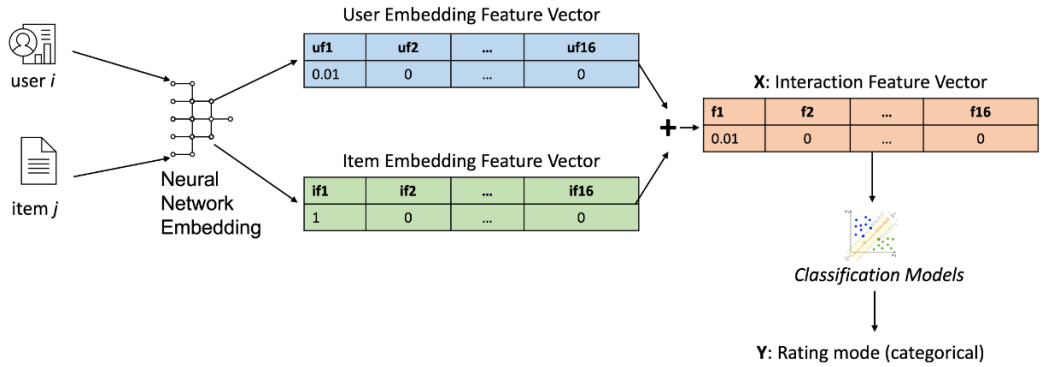
	item1	...	item100
feature1	...	...	...
feature2	...	...	...
...	...	...	...
feature16	...	...	...

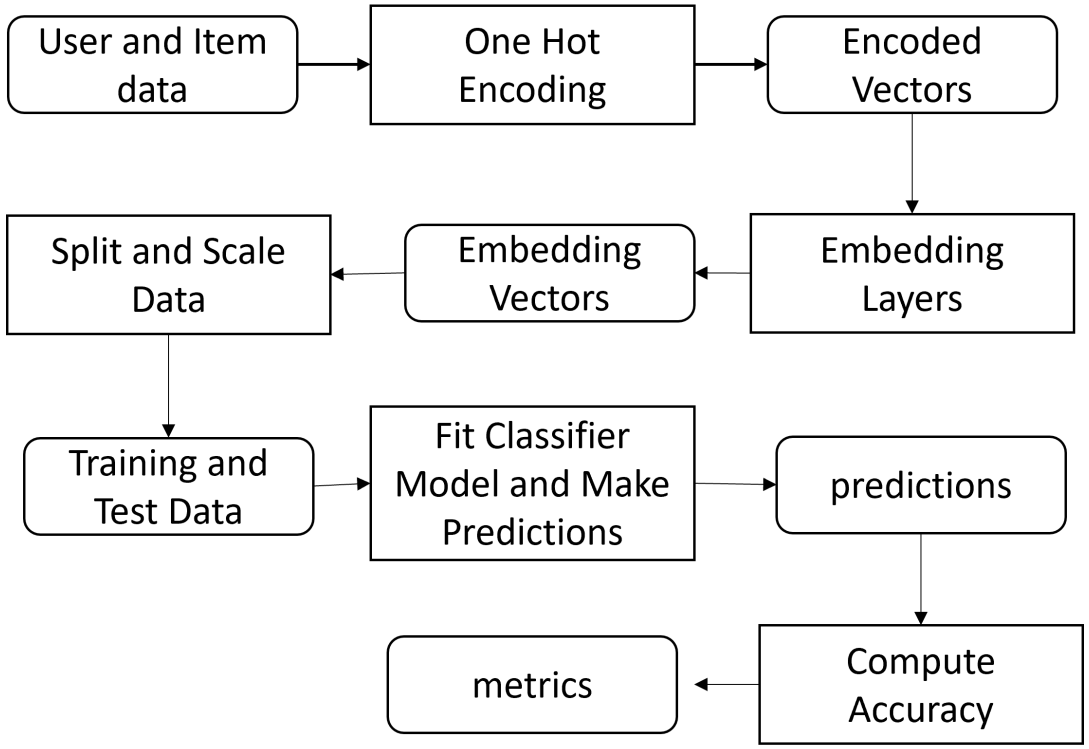
Flowchart of Neural Network Embedding Based Recommender System





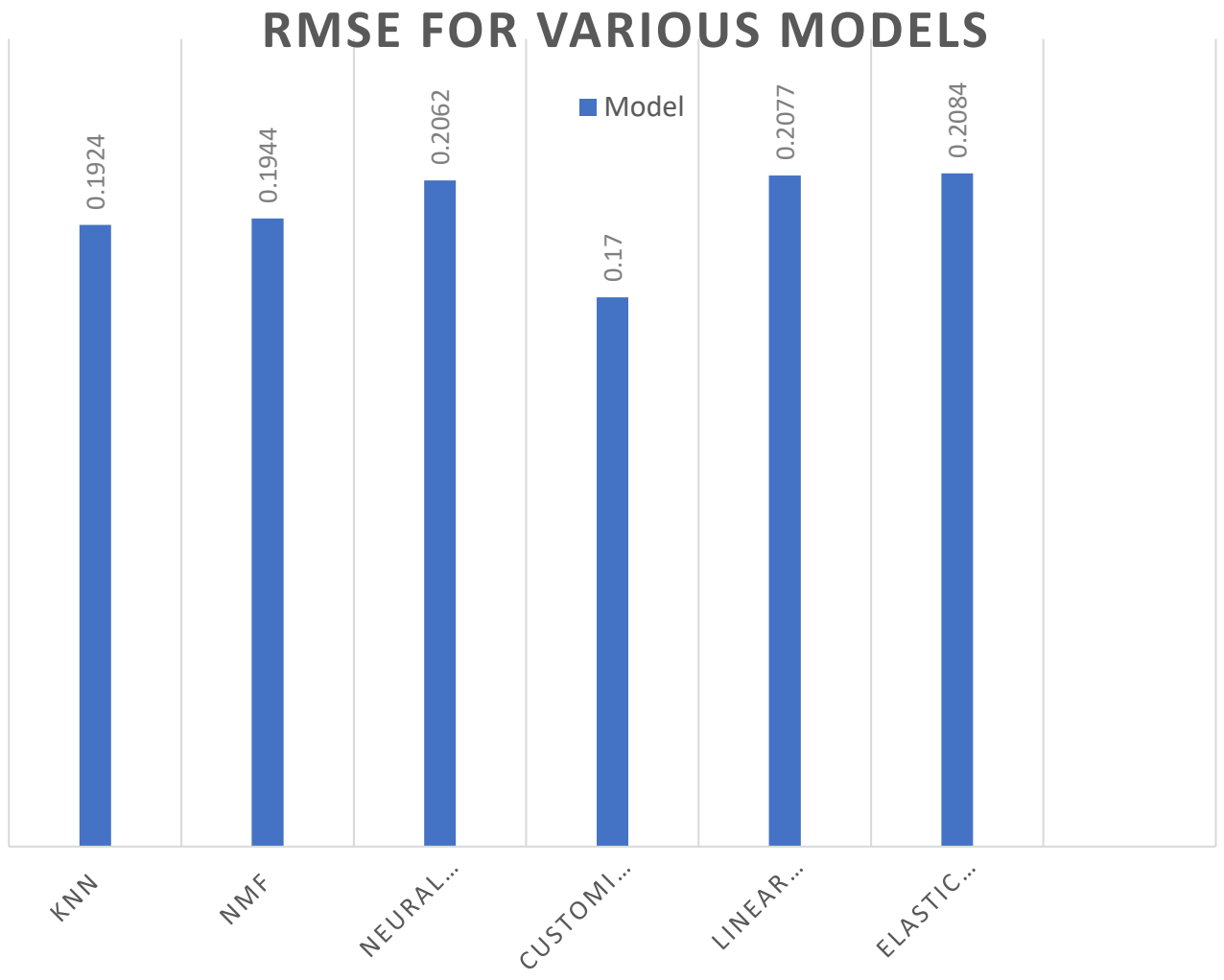
Flowchart of Classification-based Recommender Using Embedding Features





Comparison Collaborative-Filtering Model Performance

Model	RMSE
KNN	0.1924
NMF	0.1944
Neural Network	0.2062
Customized Neural Network	***0.1700***
Linear Regression w/Embedding	0.2077
ElasticNet w/Embedding	0.2084



Conclusions

- The data were ingested and analyzed using standard exploratory data analysis techniques
- Two classes (supervised and unsupervised) of machine learning methods were used to create recommender systems
- The unsupervised models used similarity metrics (Euclidean, Gaussian, Jaccard, ..) to identify similarities between new courses and previously enrolled courses. Similarity scores were computed and 3-6 courses were recommended for each user
- RMSE scores were computed for a variety of supervised learning-based recommenders. All produced RMSE around 0.20, but the approach that used a customized neural network produced RMSE=0.17
- The unsupervised approaches required more upfront feature engineering than did the supervised models
- The unsupervised approaches tended to be more explainable than the unsupervised methods

Future Work

- All of the models were based on item or user feature details, so they would recommend a course just because it was similar to others or to courses the user had already enrolled.
- This does not allow for consideration of progression where for example a user completed a beginner python course, and should not be recommended additional beginner courses, but rather the user should be recommended intermediate or advanced python courses.
- Hence, the models would be improved by removing all enrolled and similar courses to the enrolled from the candidate course dataset.

Appendix

- chazzd24/MachineLearningCapstoneProject: This repository contains the Jupyter Notebooks Used to Complete the IBM Machine Learning Capstone Project (github.com)
- Linkedin: (32) Charles Milligan | LinkedIn