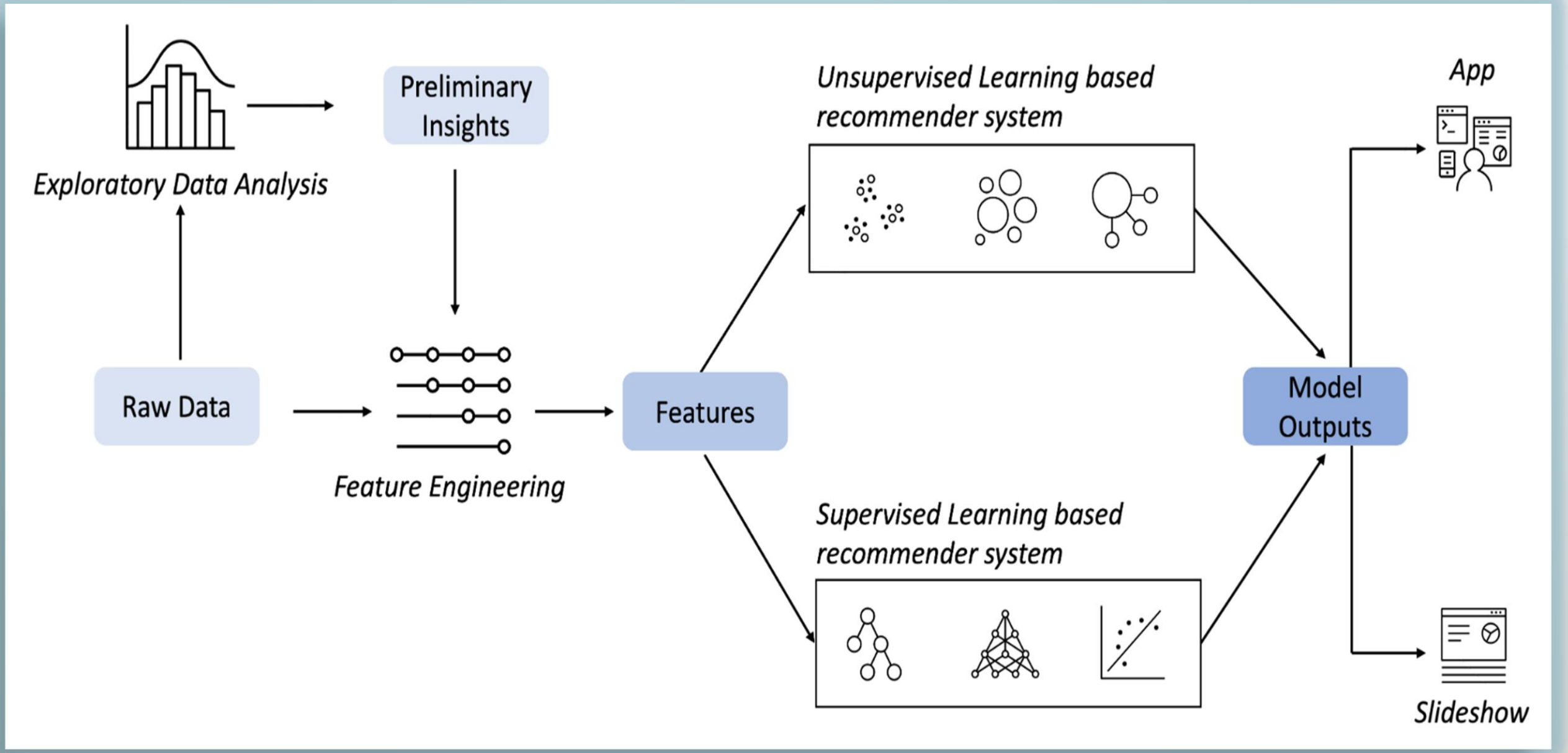
NOV. 25/2022

KEDIR NASIR OMER

Content

- ✓ Project Background
- ✓ Exploratory Data Analysis
- ✓ Content-based Recommender System using Unsupervised Learning
- ✓ Collaborative-filtering based Recommender System using Supervised learning
- ✓ Conclusion
- ✓ Reference
- ✓ Acknowledgment


Exploratory Data Analysis
Preliminary Insights
Raw Data
Feature Engineering
Features
Unsupervised Learning based recommender system
Supervised Learning based recommender system
Model Outputs
App
Slideshow

Project Background

➢ Project background and context

- ✓ **Coursera Inc.** is a U.S.-based **massive open online course** provider founded in 2012 by Stanford University computer science professors Andrew Ng and Daphne Koller. It works with universities and other organizations to offer online courses and degrees in a variety of fields. In 2021 it was estimated that about **150 universities** offered more than **4,000** courses through Coursera. It has **296 partners** across **58 countries** with over **92 million** registered learners across the world.
- ✓ Q3 of 2021, Coursera reported revenue of $109.9 million, up 33% from $82.7 million a year ago. Gross profit was $67.7 million or 61.6% of revenue. **Net loss was $(32.5) million or (29.5)% of revenue**.

➢ Problems I'm going to solve

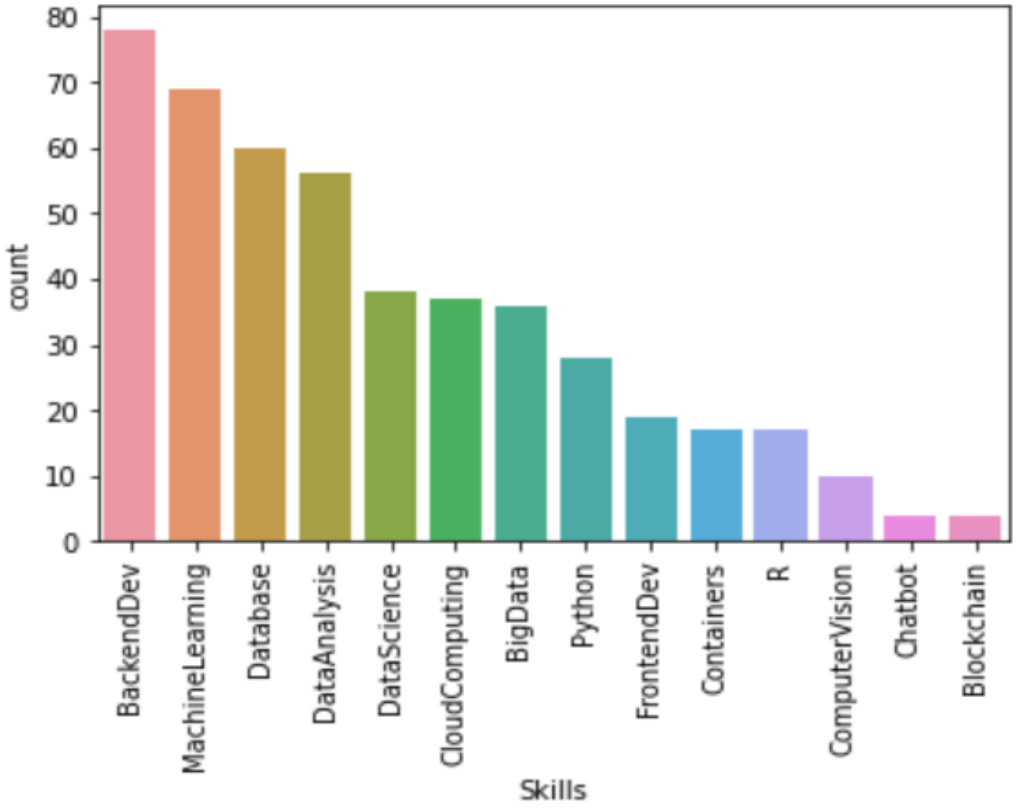
- ✓ **Attracting** more learners and improving the learning experience via helping them **quickly** find new interested courses and better **paving** their learning paths.
- ✓ Meanwhile, with more learners interacting with more courses via my recommender systems, Coursera's **revenue** also be **increased**.


Explanatory Data Analysis (EDA)
75%
32.03
38.70
20.24
81.44
12.03

➢ Course Counts Per Genre

- ✓ Using **seaborn barplot** to visualize course genre counts using a bar chart
- ✓ The x-axis is the **course genre** and the y-axis is the **course count per genre**

```
[22]: sns.barplot(data=x, x="Skills", y="count")
      plt.xticks(rotation=90)
      plt.show()
```

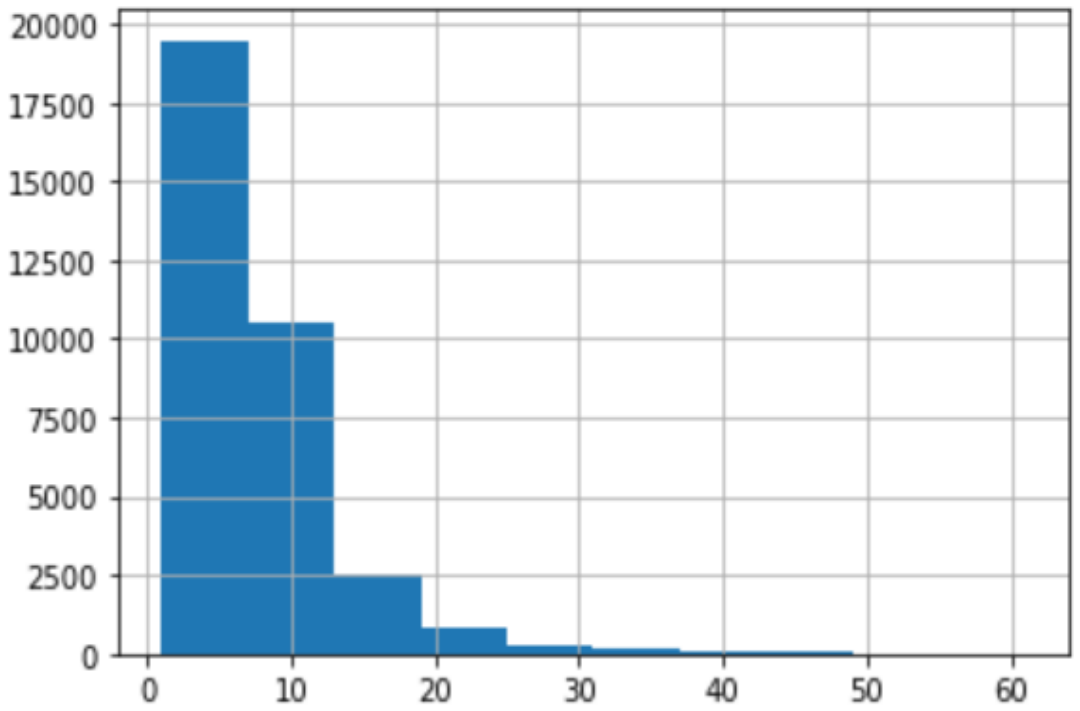


➢ Course Enrollment Distribution

- ✓ Plot the **histogram** to visualize user rating counts
- ✓ The x-axis is the **course count** and the y-axis is the **course rating**

```
[33]: # WRITE YOUR CODE HERE
      user_count_df.hist()
```

```
[33]: <AxesSubplot:>
```



➢ 20 Most Popular Courses

- ✓ Only can see the item IDs which do not indicate what kind of courses they are. To make it more clear, we need to **join** the course titles in the course metadata dataset (**course_df**) so that we can identify what the most popular courses are immediately
- ✓ Using **Pandas merge()** method to join the **course_df** (contains the course title column)

```
[56]: pd.merge(course_count_df, course_df[['COURSE_ID','TITLE']],on='COURSE_ID', how='left')
```

[56]:

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

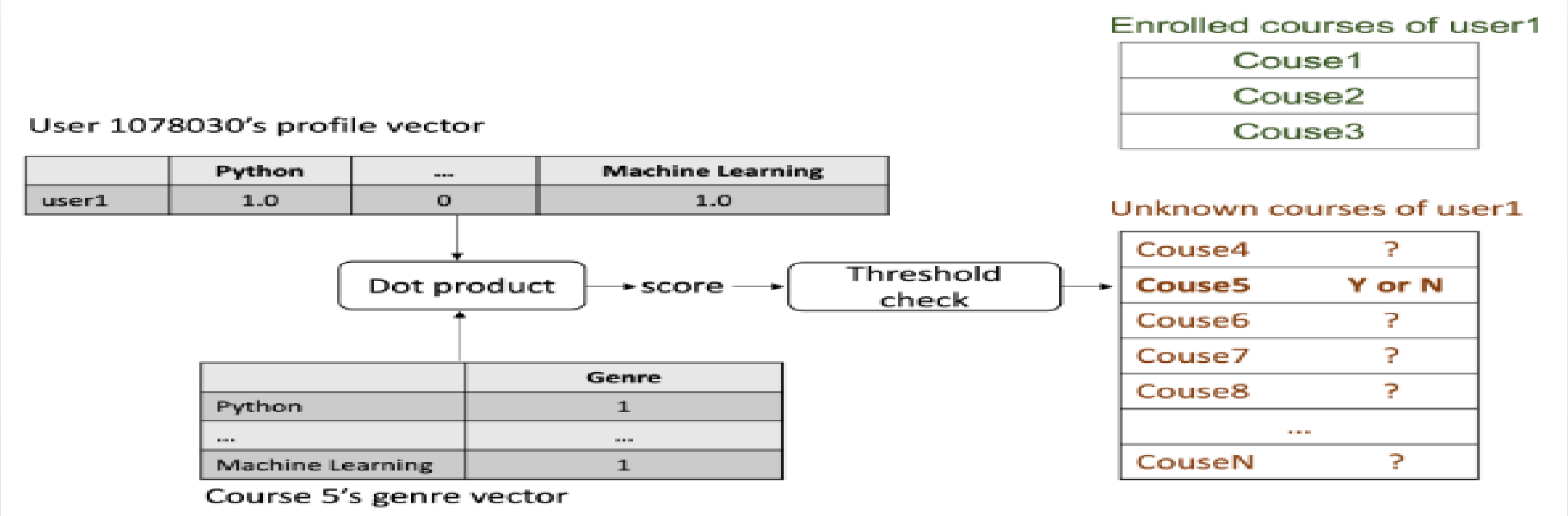
➢ Word Cloud Of Course Titles

- ✓ Create a **WordCloud** object and generate wordcloud from the titles
- ✓ Use **plt.imshow()** method to visualize the generated wordcloud
- ✓ By looking at these keywords, we should have a general understanding that the courses in the **dataset** are **focused** on demanding **IT skills**


Content-based Recommender System
using Unsupervised Learning

➢ Flowchart Of Content-based Recommender System Using User Profile and Course Genres

✓ Generate a user profile based on course genres and rating using a simple dot product, to compute or predict an interest score for each course and recommend those courses with high interest scores.

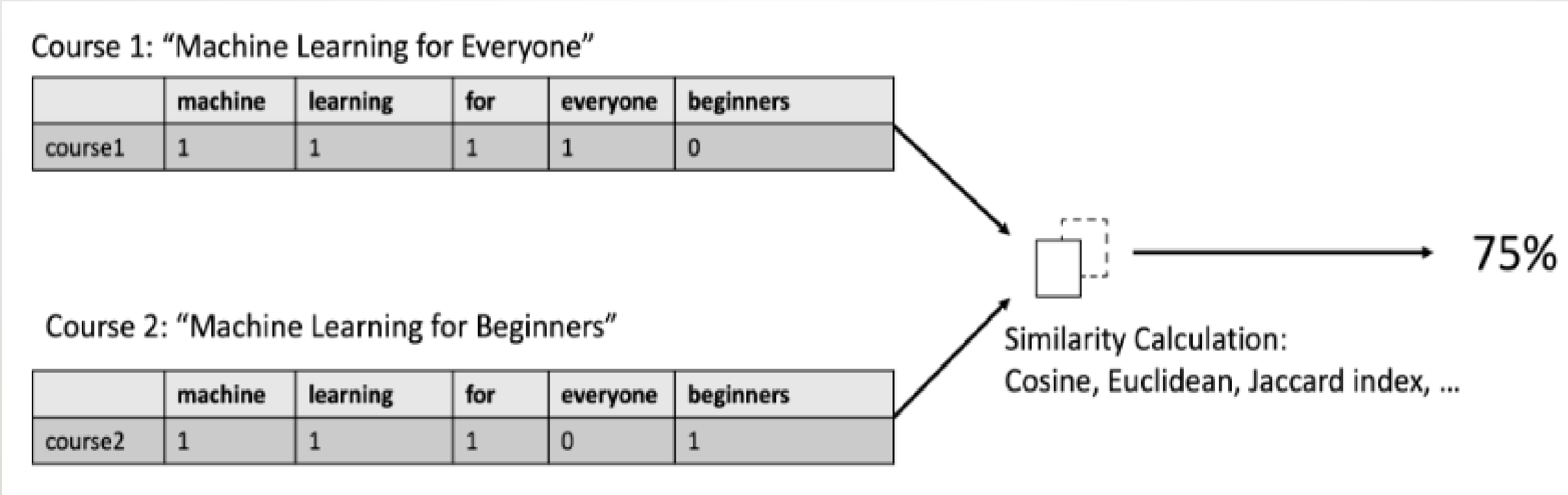


➢ Evaluation Results Of User Profile-based Recommender System

- ✓ Generate a user profile based on course genres and rating using a simple dot product, to compute or predict an interest score for each course and recommend those courses with high interest scores.
- ✓ Total numbers of test users 1000
- ✓ Then we can use all courses to subtract the enrolled courses to get a set of all unknown courses for user 1078030, and we want to find potential interested courses hidden in the unknown course list.
- ✓ Complete the **generate_recommendation_scores()** function blow to generate recommendation score for all users.
- ✓ Ideally, we should limit the maximum course recommendations to be less than 20 courses per user. As such, the average course recommendations per user should also be less than 20 or so. This makes sure we only recommend relevant courses with high confidence (score).

➢ Flowchart Of Content-based Recommender System Using Course Similarity

- ✓ The similarity or closeness of items is measured based on the similarity in the content or features of those items. The course genres are important features, and in addition to that, the BoW value is another important type of feature to represent course textual content.



➢ Evaluation Results Of Course Similarity Based Recommender System

✓ **generate_recommendations_for_all()** method to generate recommendations for all users.

[27]:
```
res_dict = {}
users, courses, sim_scores = generate_recommendations_for_all()
res_dict['USER'] = users
res_dict['COURSE_ID'] = courses
res_dict['SCORE'] = sim_scores
res_df = pd.DataFrame(res_dict, columns=['USER', 'COURSE_ID', 'SCORE'])
```

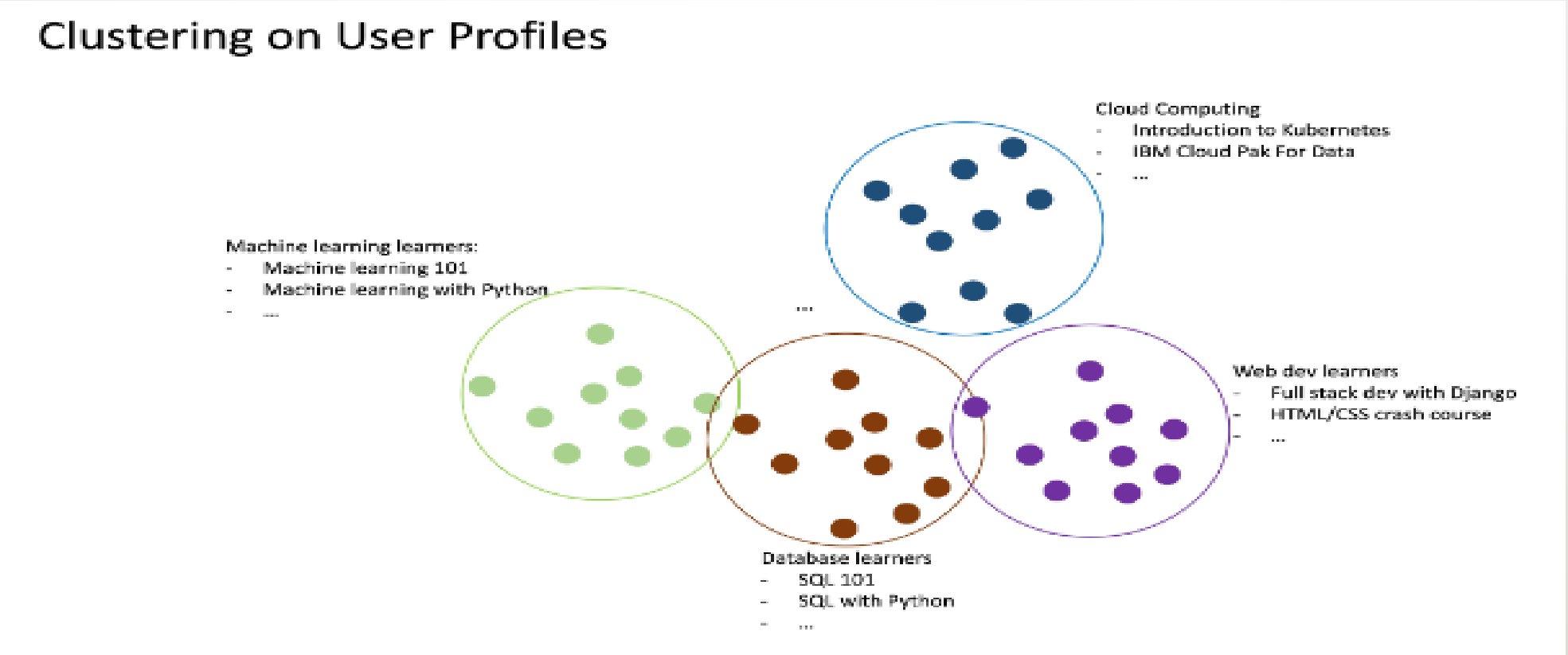
[28]:
```
res_df.head()
```

[28]:

	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

➢ Flowchart Of Clustering-based Recommender System

- ✓ We could perform clustering algorithms such as K-means or DBSCAN to group users with similar learning interests. For example, in the below user clusters, we have user clusters whom have learned courses related to machine learning, cloud computing, databases, and web development, etc.



➢ Evaluation Results Of Clustering-based Recommender System

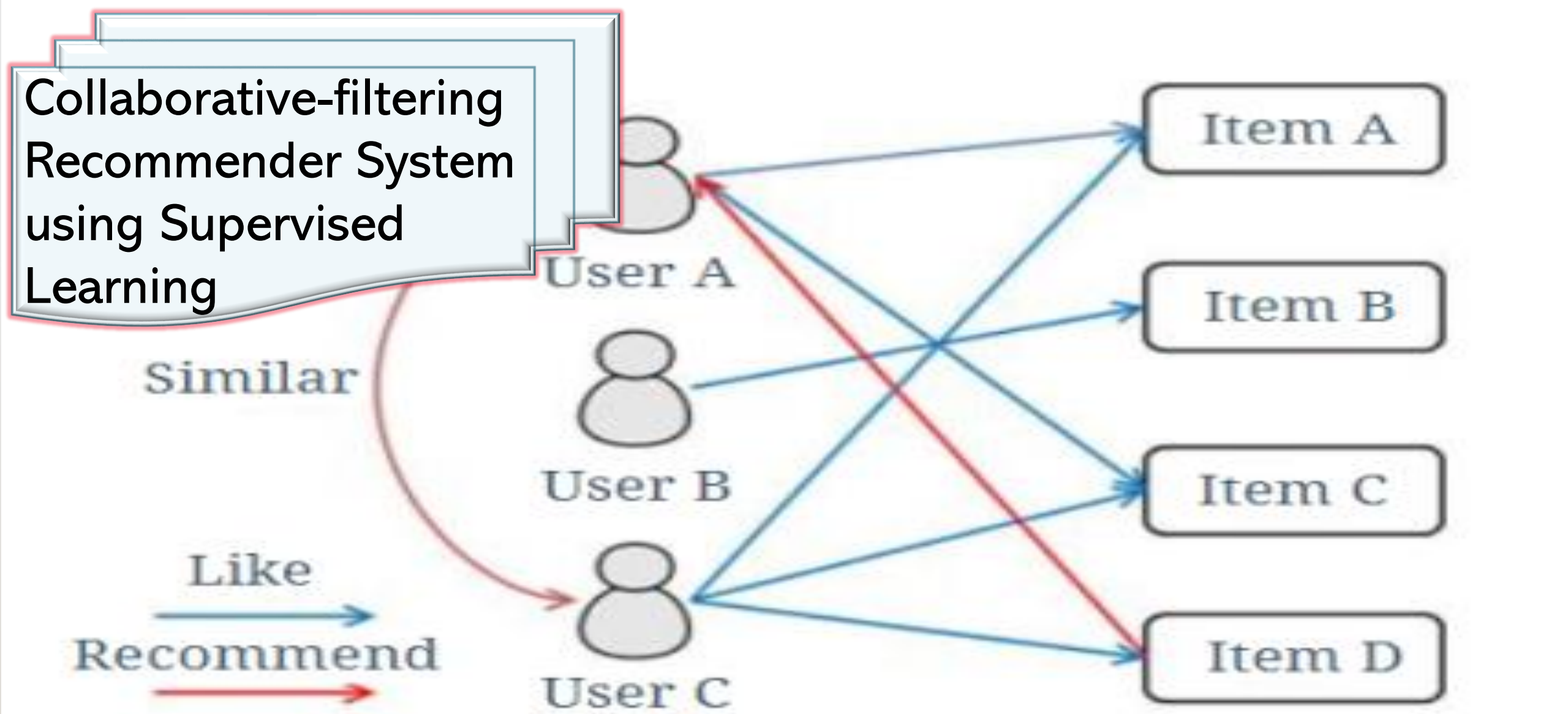
✓ we know each user's enrolled courses and its cluster index. If we use a groupby and sum aggregation, we can get the enrollments count for each course in each group, like the following code snippet:

```
[87]: courses_cluster = test_users_labelled[['item', 'cluster']]
courses_cluster['count'] = [1] * len(courses_cluster)
count_enrollments_df = courses_cluster.groupby(['cluster','item']).agg(enrollments = ('count','sum')).reset_index()
count_enrollments_df
```

[87]:

	cluster	item	enrollments
0	0	AI0111EN	2
1	0	BC0101EN	44
2	0	BC0201EN	6
3	0	BD0101EN	35
4	0	BD0111EN	24
...	...	...	...
1393	24	ST0101EN	10
1394	24	TA0105	5
1395	24	TA0106EN	2
1396	24	TMP0105EN	1
1397	24	WA0101EN	9

Collaborative-filtering Recommender System using Supervised Learning



➢ KNN Based Recommender System

✓ If two users are similar, we can simply calculate the similarities between their row vectors in the interaction matrix. Then based on the similarity measurements, we can find the k nearest neighbor as the similar users.

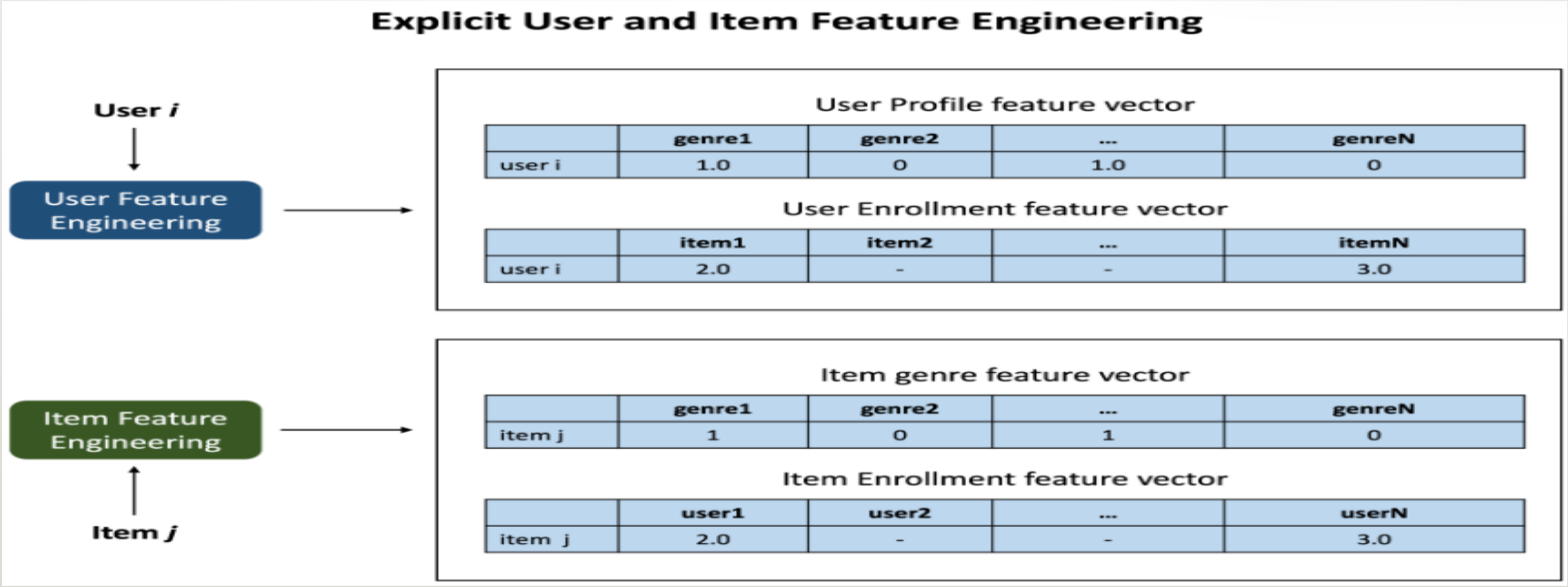
User-Item interaction matrix

Similar users

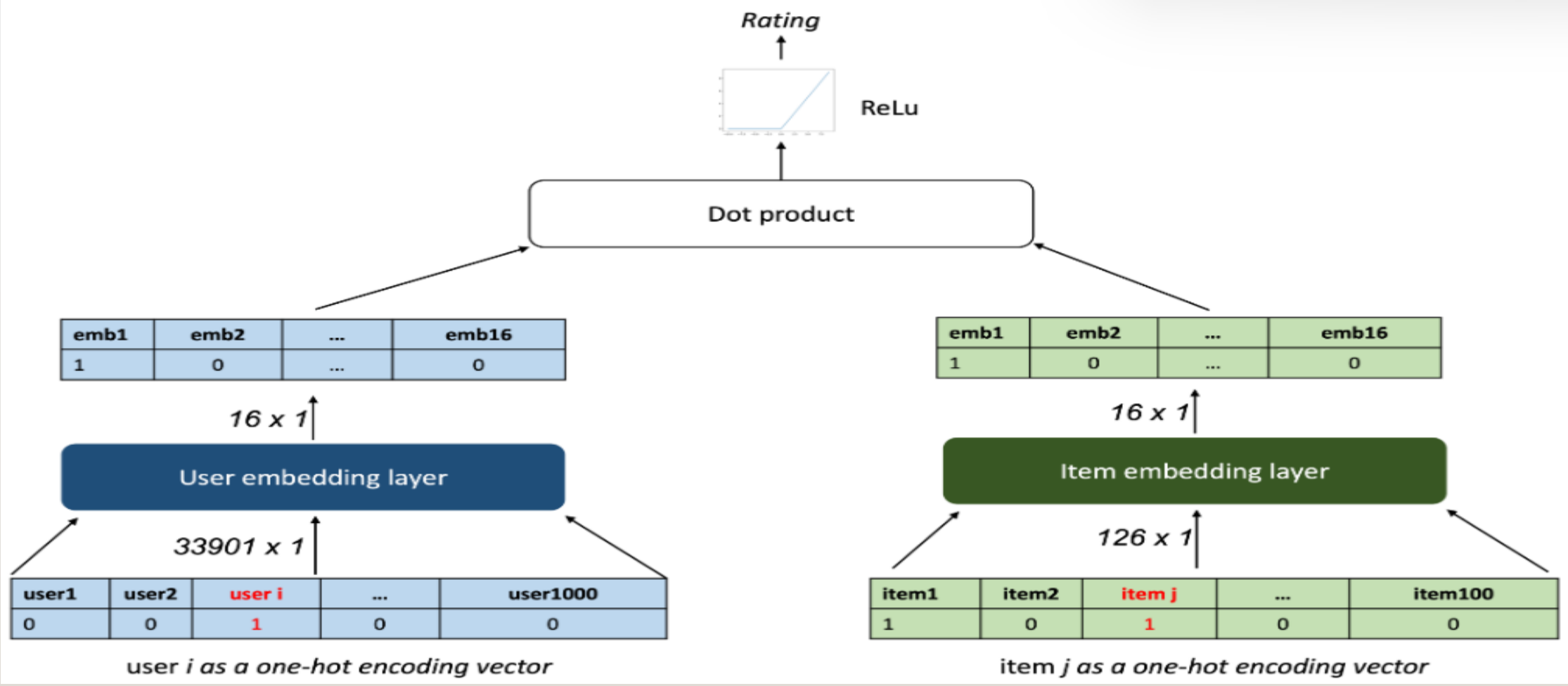
	Machine Learning With Python	Machine Learning 101	Machine Learning Capstone	SQL with Python	Python 101
...	...	...	...	...	
user2	3.0	3.0	3.0	3.0	3.0
user3	2.0	3.0	3.0	2.0	
user4	3.0	3.0	2.0	2.0	3.0
user5	2.0	3.0	3.0		
user6	3.0	3.0	?		3.0
...	...	...	...	...	

➢ Neural Network Embedding Based Recommender System

✓ The main advantage of using these explicit features is they are highly interpretable and yield very good performance as well.

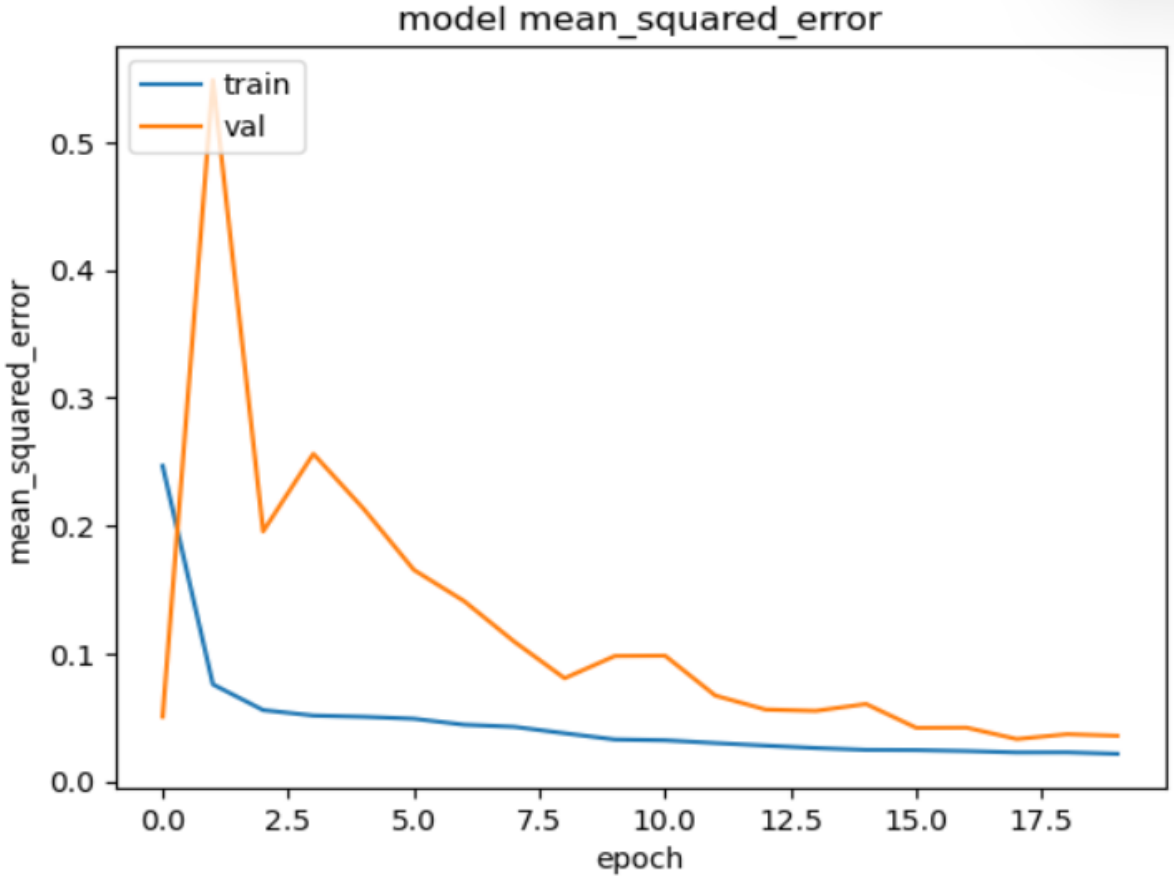


- ✓ The main advantage of using these explicit features is they are highly interpretable and yield very good performance as well. The goal is to create a neural network structure that can take the user and item one-hot vectors as inputs and outputs a rating estimation or the probability of interaction (such as the probability of completing a course).



➢ Comparing the Performance Of Collaborative-filtering Models

- ✓ The pre-defined **RecommenderNet()** is a actually very basic neural network, you are encouraged to customize it to see if model prediction performance will be improved. Here are some directions
- ✓ **Hyperparameter** tuning, such as the embedding layer dimensions
- ✓ Add more hidden layers
- ✓ Try different activation functions such as **ReLu**
- ✓ **Mean_squared_error** of **0.3409** which is good

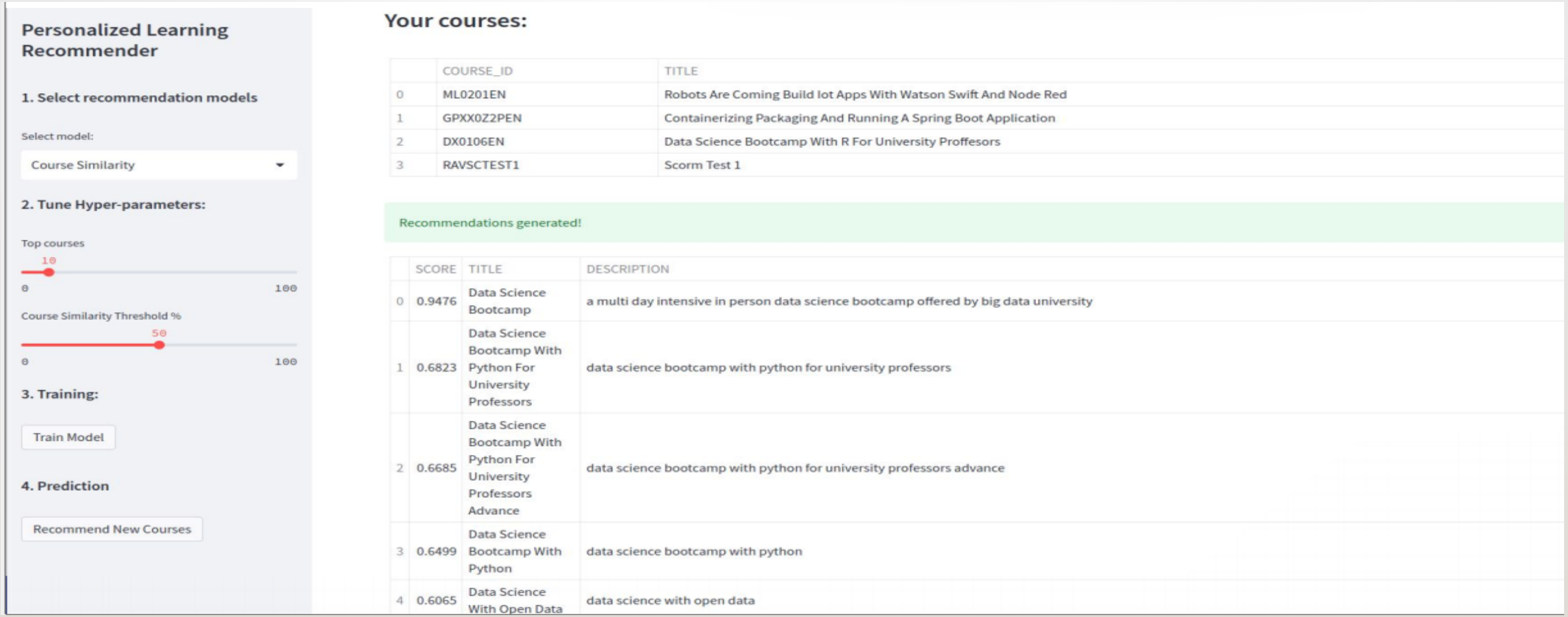



```
model1.evaluate(x_test, y_test)
```

```
730/730 [==============================] - 4s 5ms/step - loss: 0.3
414 - mean_squared_error: 0.3409
```

➢ Course Recommender System App With Streamlit

✓ We can deployed recommender system to Streamlit App



Conclusion

- We illustrated and achieved that:

  - ✓ Analyzing Exploratory Data
  - ✓ Building Unsupervised Learning-Based Content-Based Recommender System
  - ✓ Building Recommender system based on collaborative filtering and supervised learning
  - ✓ By assisting Learners in finding new, interesting courses quickly and by better laying out their learning paths, we can draw in more students and enhance their educational experience.
  - ✓ Coursera's revenue will also rise as more students use my recommender systems to interact with more courses.

Reference

- IBM Machine Learning Specializations: Machine Learning Capstone Project

  - ✓ Visit My GitHub Portfolio: https://github.com/kedibeki/Personalized-Online-Course-Recommender-System-with-Machine-Learning

  - ✓ Visit IBM's Coursera Page: https://www.coursera.org/learn/machine-learning-capstone?specialization=ibm-machine-learning

Acknowledgment

- IBM and Coursera: IBM Machine Learning Specializations Instructors
  - ✓ Yan Luo- Ph.D., Data Scientist and Developer
  - ✓ Mark J Grover- Digital Content Delivery Lead
  - ✓ Miguel Maldonado- Machine Learning Curriculum Developer
  - ✓ Joseph Santarcangelo- Ph.D., Data Scientist at IBM
  - ✓ Xintong Li- Data Scientist at IBM
  - ✓ Svitlana (Lana) Kramar- Data Science Content Developer
  - ✓ IBM Skill Network and Coursera
  - ✓ Peer Learners

coursera
education for everyone

Thank You!