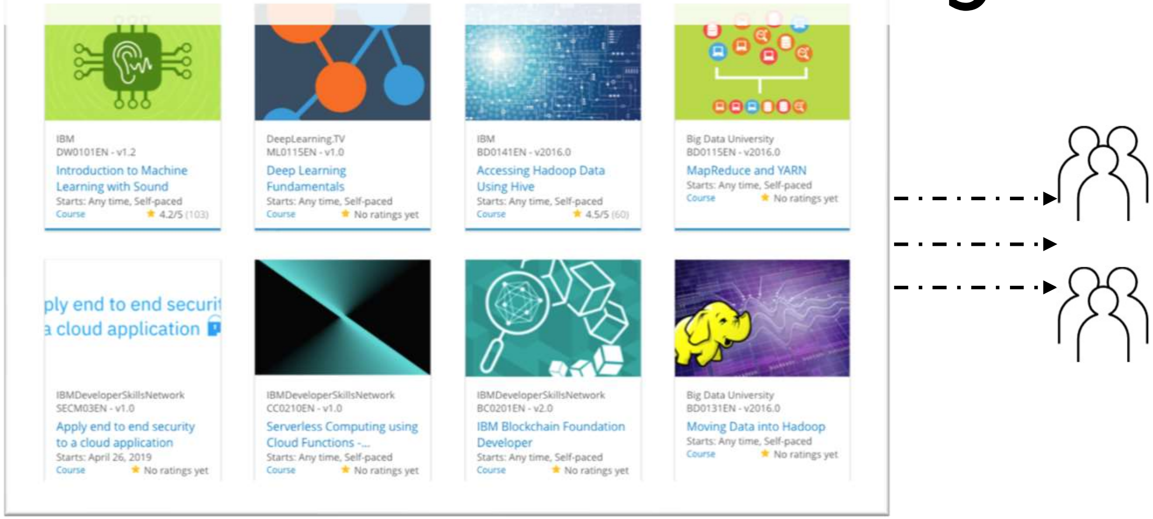
Build a Personalized Online Course Recommender System with Machine Learning

Muhammad Farooq
3rd November 2023



IBM
DW0101EN - v1.2
Introduction to Machine Learning with Sound
Starts: Any time, Self-paced
Course ★ 4.2/5 (103)

DeepLearning.TV
ML0115EN - v1.0
Deep Learning Fundamentals
Starts: Any time, Self-paced
Course ★ No ratings yet

IBM
BD0141EN - v2016.0
Accessing Hadoop Data Using Hive
Starts: Any time, Self-paced
Course ★ 4.5/5 (60)

Big Data University
BD0115EN - v2016.0
MapReduce and YARN
Starts: Any time, Self-paced
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
SECM03EN - v1.0
Apply end to end security to a cloud application
Starts: April 26, 2019
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
CC0210EN - v1.0
Serverless Computing using Cloud Functions -...
Starts: Any time, Self-paced
Course ★ No ratings yet

IBMDeveloperSkillsNetwork
BC0201EN - v2.0
IBM Blockchain Foundation Developer
Starts: Any time, Self-paced
Course ★ No ratings yet

Big Data University
BD0131EN - v2016.0
Moving Data into Hadoop
Starts: Any time, Self-paced
Course ★ No ratings yet

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

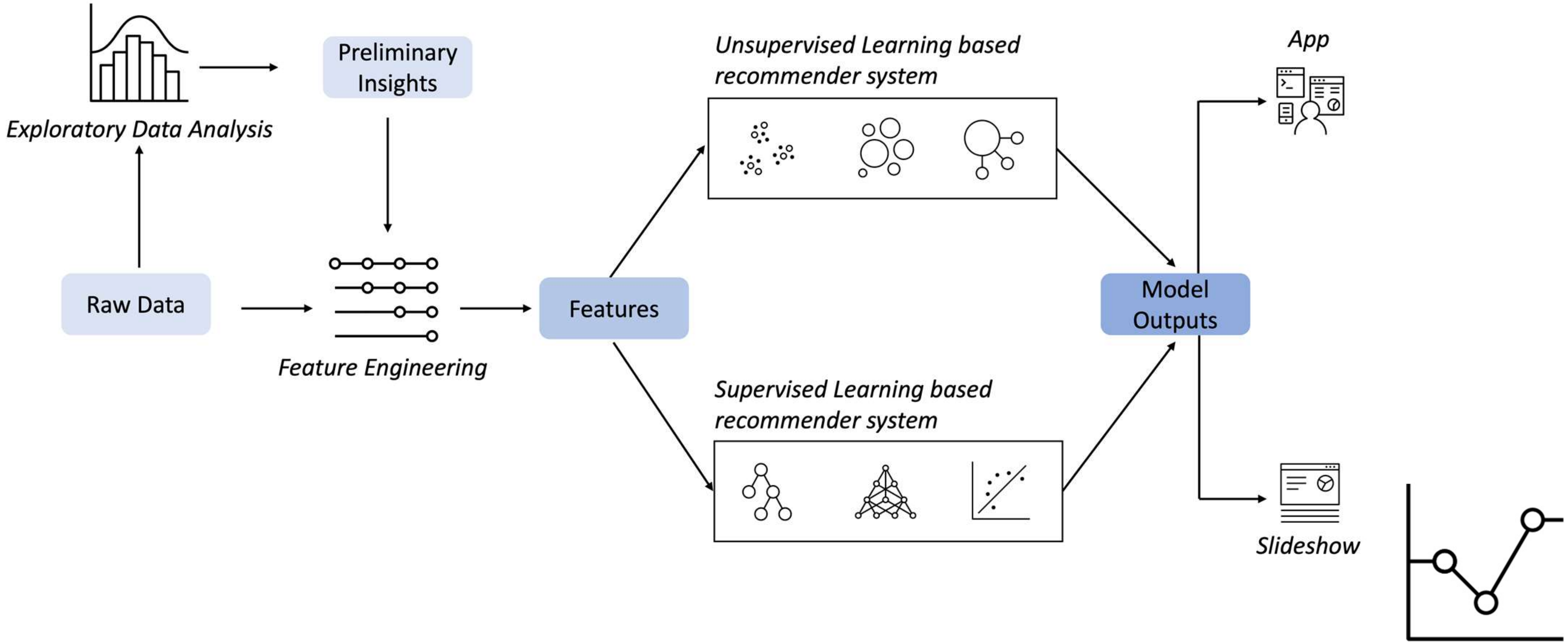
Problem statement and hypotheses

The main problem addressed in this project is the difficulty for the users to discover new courses that likely to their interests by creating a personalized learning track system more appropriate to the users' subject requirements to empower the desired skills. The main object and hypothesize is to build a user personalized recommendation system based on course content and students' learning history in discovering of new courses and to facilitate the learning pathways. By adopting content and collaborative-filtering systems will develop by comparing appropriate supervised and unsupervised machine learning models to find the best model.

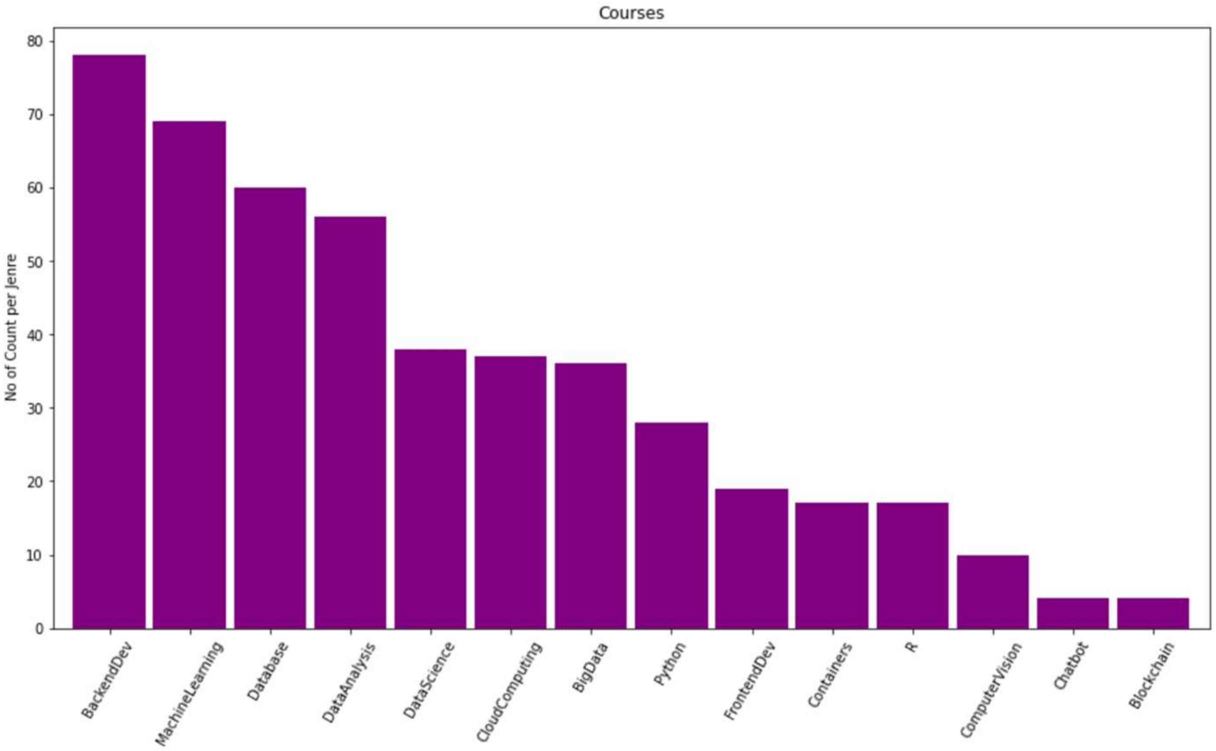
Project background and context

An unprecedented demand and growth an Online Learning Management System has challenged the Machine Learning engineer a startup of and AI based MOOC (Massive Open Online Course) with universal access like Udemy, Edx and Coursra. Building a personalized and effective online course recommender system with machine learning is a complex task, but it can greatly enhance user experience and engagement on course recommender platform. By ensuring an up-to-date with the latest advancements in recommendation algorithms and practices for ongoing improvements.

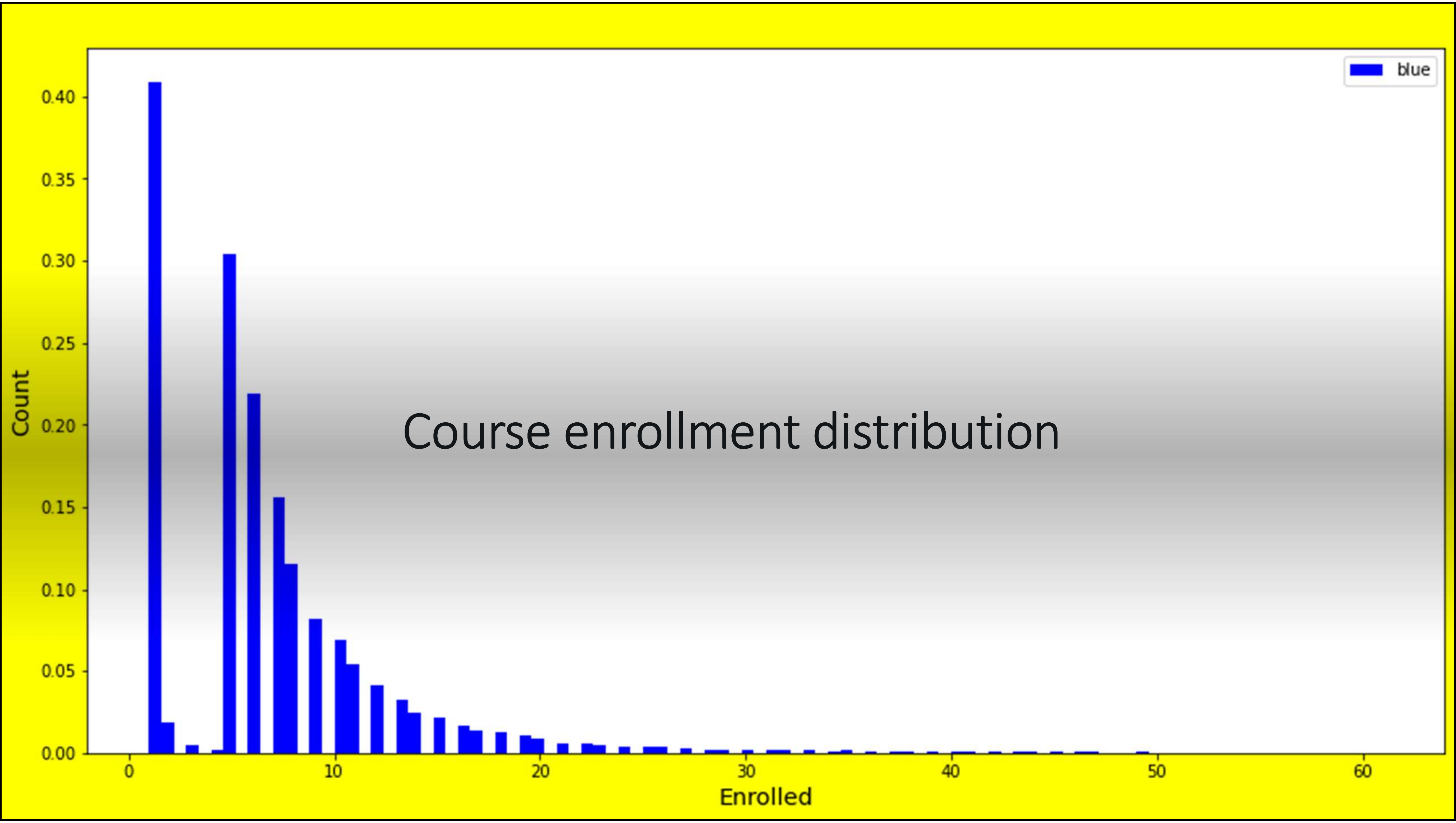
Exploratory Data Analysis


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

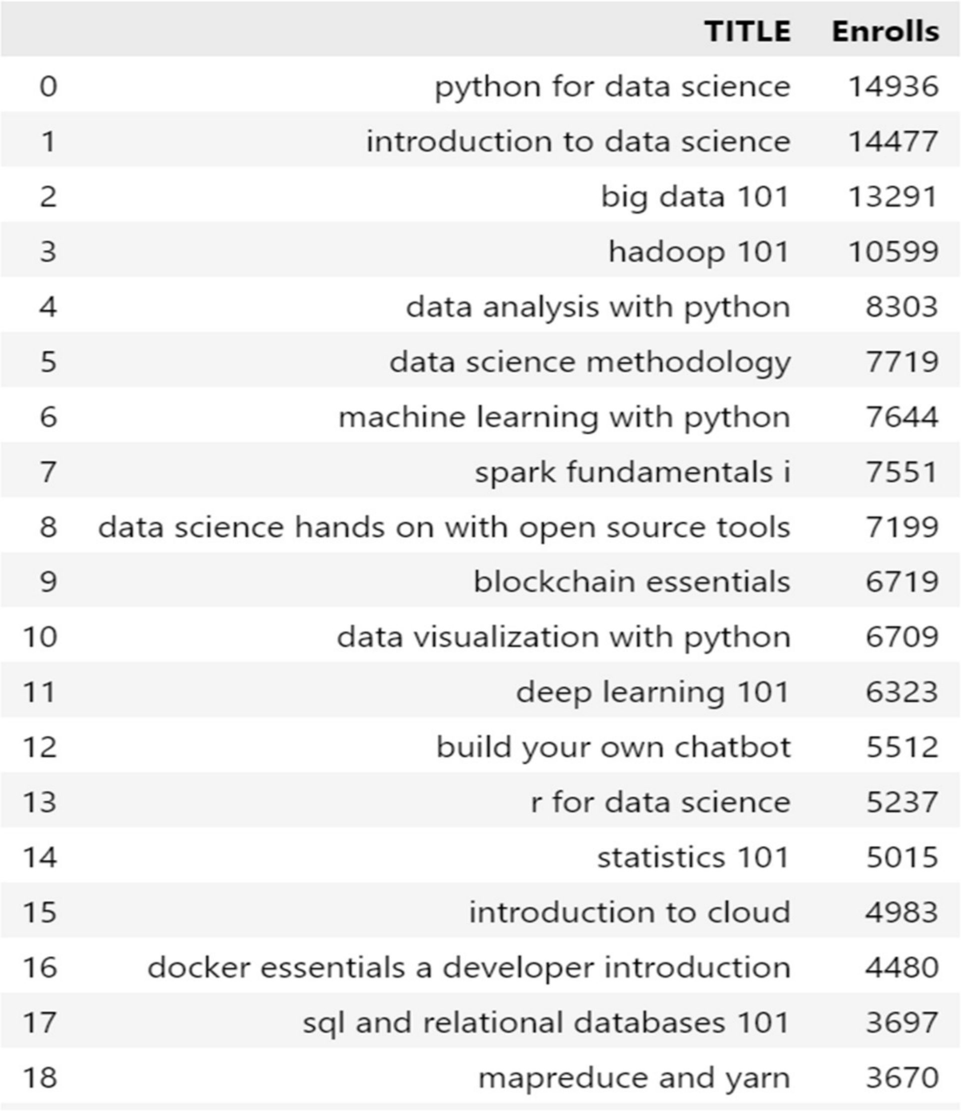
Course counts per genre



Course enrollment distribution

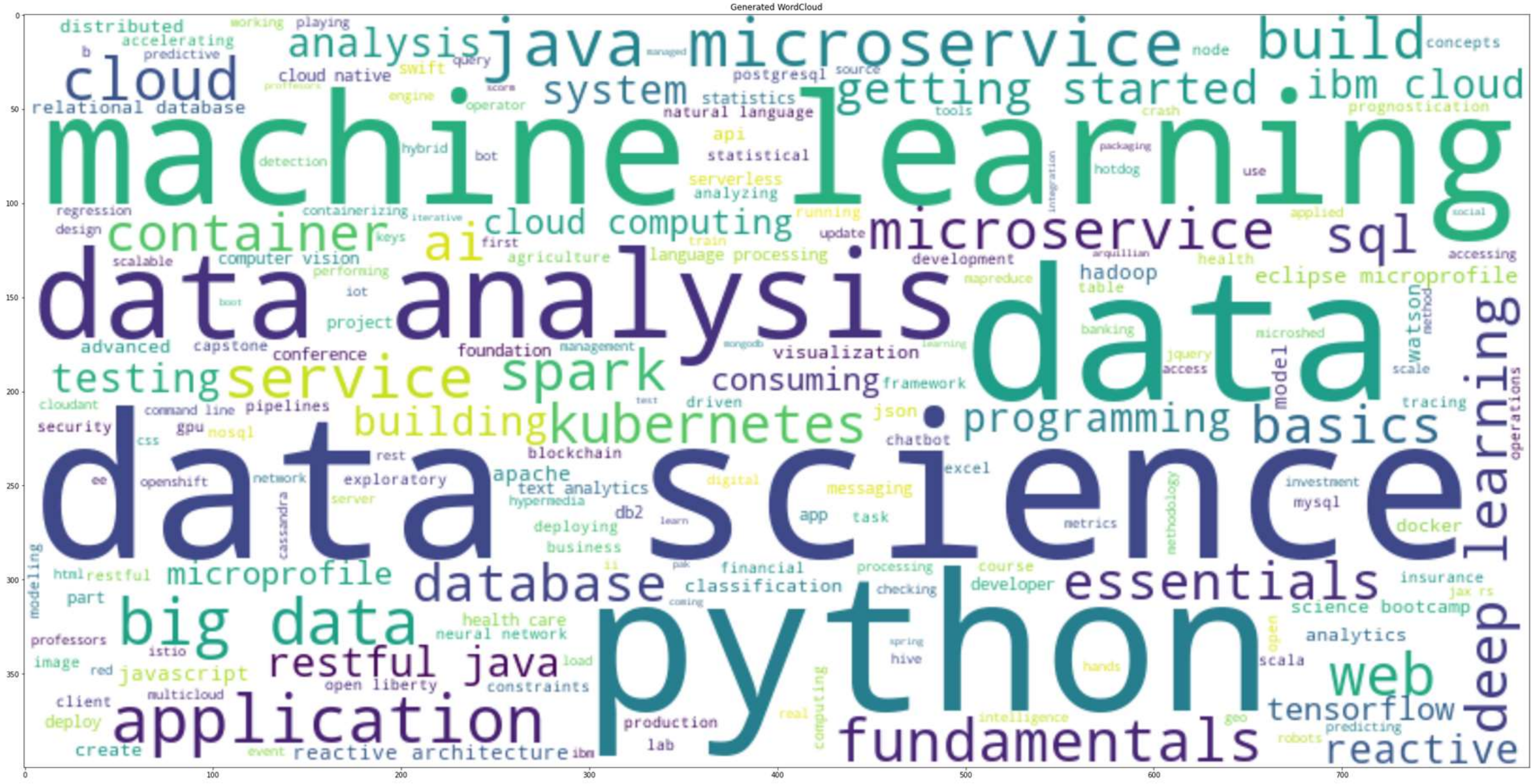


20 most popular courses

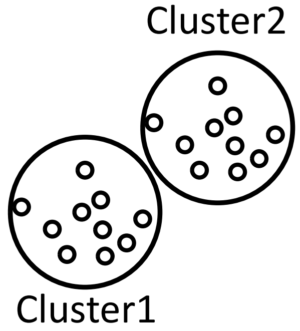


	TITLE	Enrolls
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199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

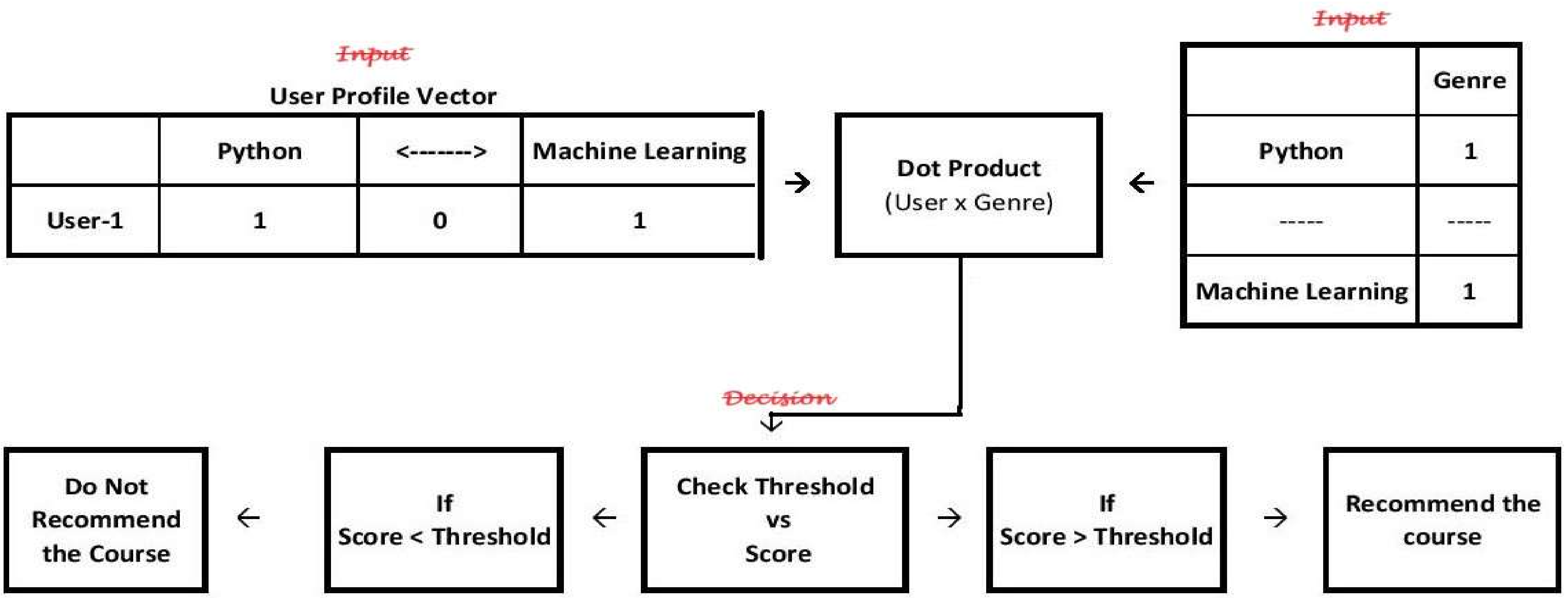
Word cloud of course titles



Content-based Recommender System using Unsupervised Learning



Flowchart of content-based recommender system using a user profile and course genres. For each dot product



~~Input~~

User Profile Vector

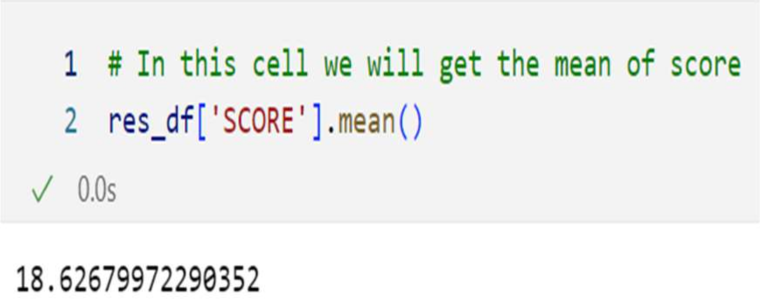
	Python	<------->	Machine Learning
User-1	1	0	1

~~Input~~

	Genre
Python	1
-----	-----
Machine Learning	1

Evaluation results of user profile-based recommender system (Threshold Score = 10)

On average 18 courses approximately have been recommended per user (in the test user dataset)

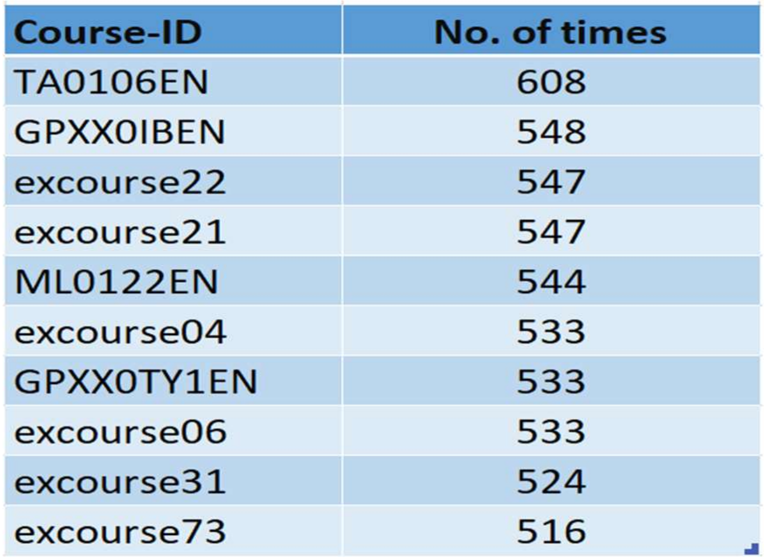


```
# In this cell we will get the mean of score
res_df['SCORE'].mean()
```

✓ 0.0s

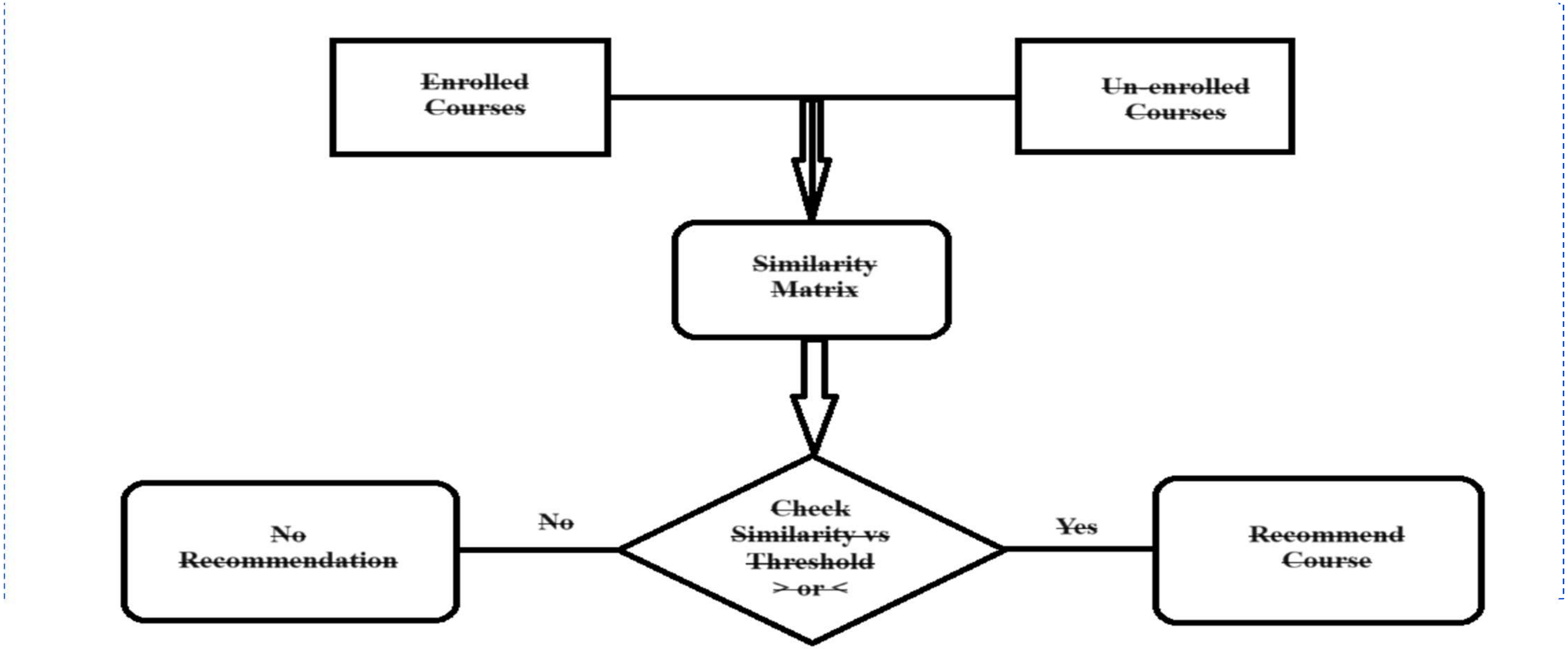
```
18.62679972290352
```

Top-10 common recommended courses



Course-ID	No. of times
TA0106EN	608
GPXX0IBEN	548
excourse22	547
excourse21	547
ML0122EN	544
excourse04	533
GPXX0TY1EN	533
excourse06	533
excourse31	524
excourse73	516

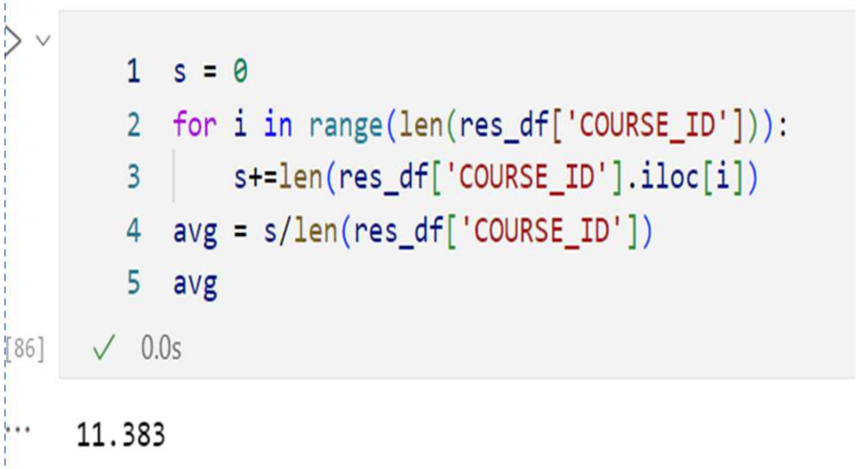
Flowchart of content-based recommender system using course similarity



Evaluation results of course similarity based recommender system

Hyper-parameter settings, such as a score or similarity threshold

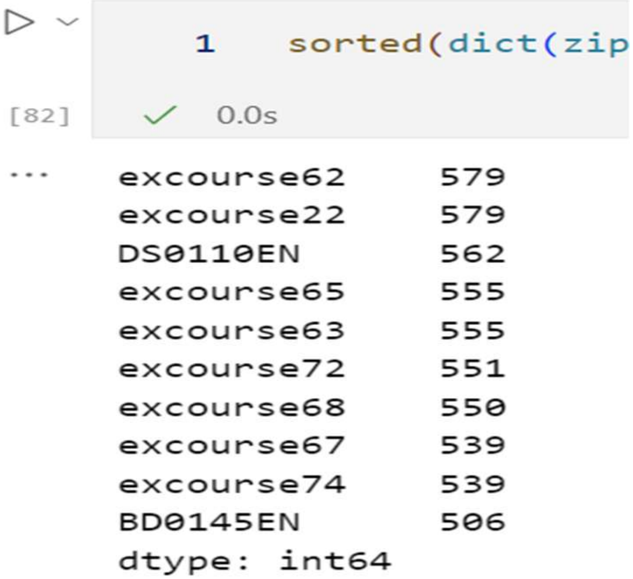
On average 11.383 approximately unseen courses have been recommended per user.



```
s = 0
for i in range(len(res_df['COURSE_ID'])):
    s+=len(res_df['COURSE_ID'].iloc[i])
avg = s/len(res_df['COURSE_ID'])
avg
```

```
11.383
```

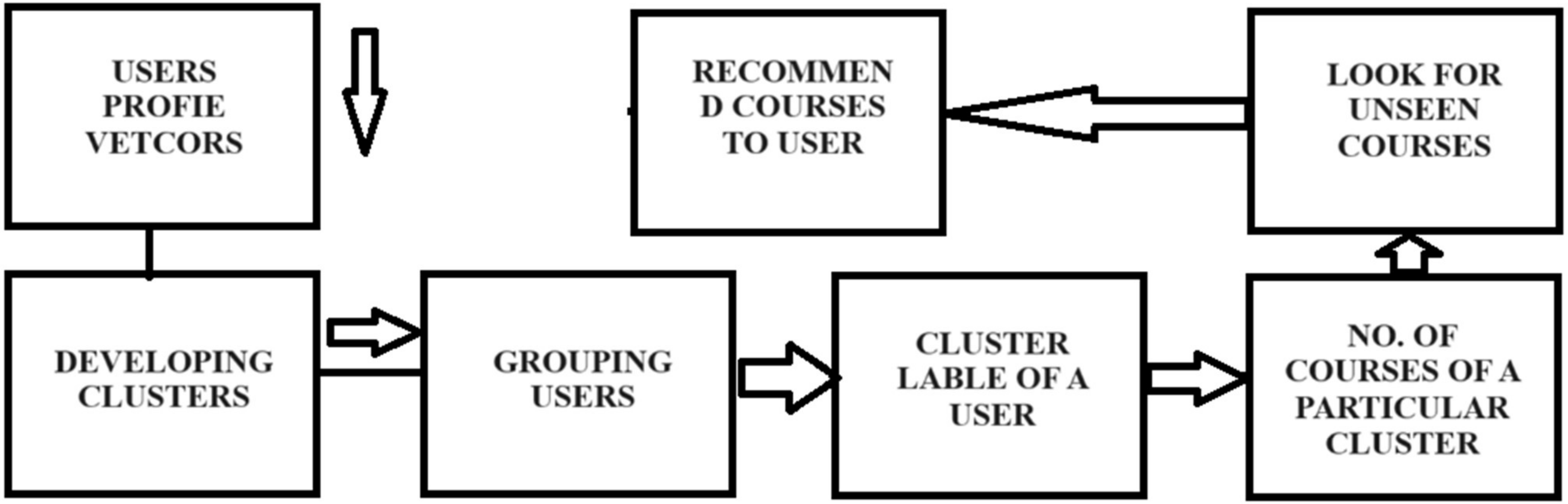
Top-10 most frequently recommended courses.



```
sorted(dict(zip
```

```
excourse62     579
excourse22     579
DS0110EN       562
excourse65     555
excourse63     555
excourse72     551
excourse68     550
excourse67     539
excourse74     539
BD0145EN       506
dtype: int64
```

Flowchart of clustering-based recommender system

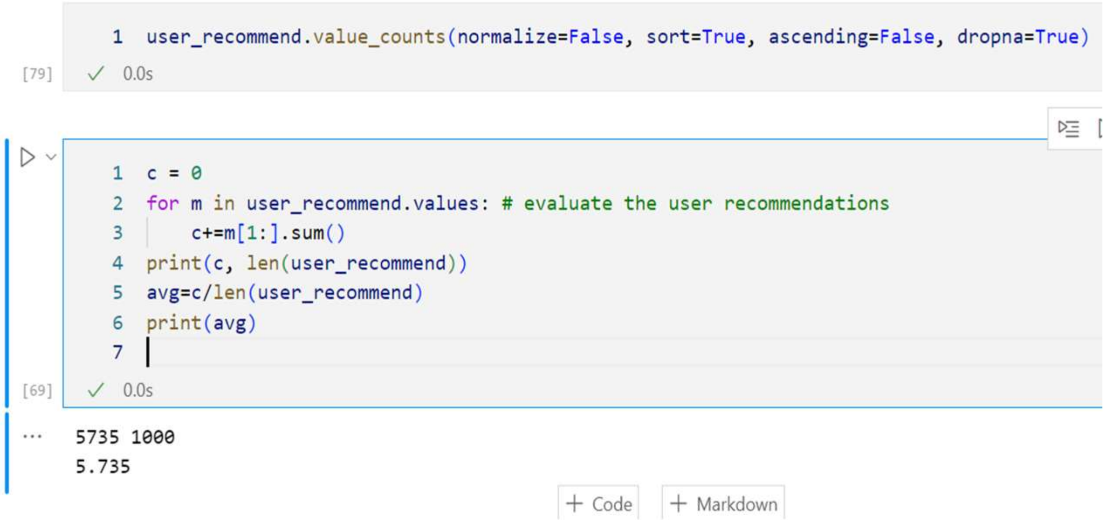


USERS GROUPSED IN CLUSTER 3 RECOMMENDED 02 COURSES 'PYTHON', 'MACHINE LEARNING'

Evaluation results of clustering-based recommender system

Threshold = 5.735 considering it to be 6 approximately

On average, how many new courses



```
user_recommend.value_counts(normalize=False, sort=True, ascending=False, dropna=True)
```

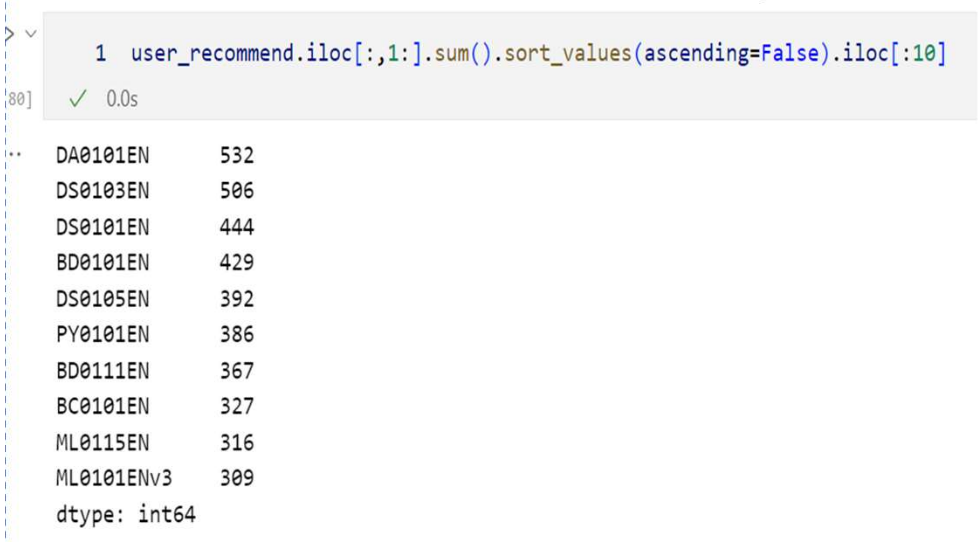
[79] ✓ 0.0s

```
c = 0
for m in user_recommend.values: # evaluate the user recommendations
    c+=m[1:].sum()
print(c, len(user_recommend))
avg=c/len(user_recommend)
print(avg)
```

[69] ✓ 0.0s

```
5735 1000
5.735
```

Top-10 commonly recommended courses

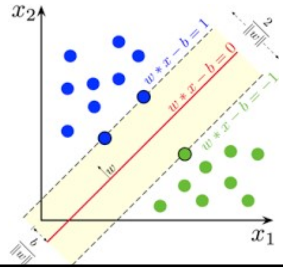


```
user_recommend.iloc[:,1:].sum().sort_values(ascending=False).iloc[:10]
```

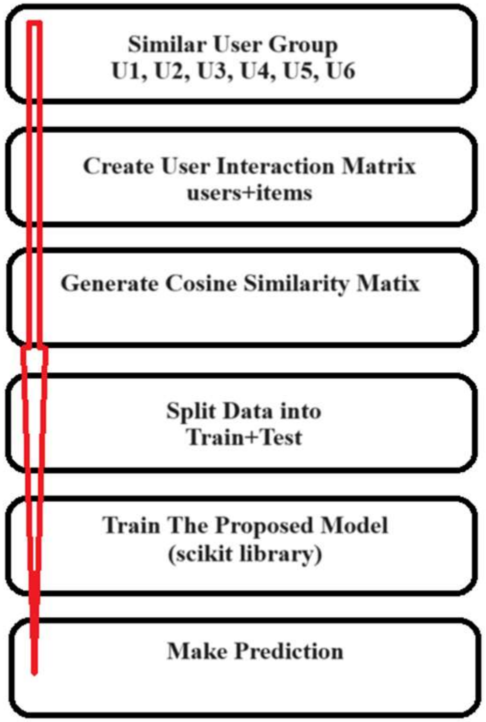
[80] ✓ 0.0s

```
DA0101EN      532
DS0103EN      506
DS0101EN      444
BD0101EN      429
DS0105EN      392
PY0101EN      386
BD0111EN      367
BC0101EN      327
ML0115EN      316
ML0101ENv3    309
dtype: int64
```

Collaborative-filtering Recommender System using Supervised Learning



Flowchart of KNN based recommender system



Similar User Group
U1, U2, U3, U4, U5, U6

Create User Interaction Matrix
users+items

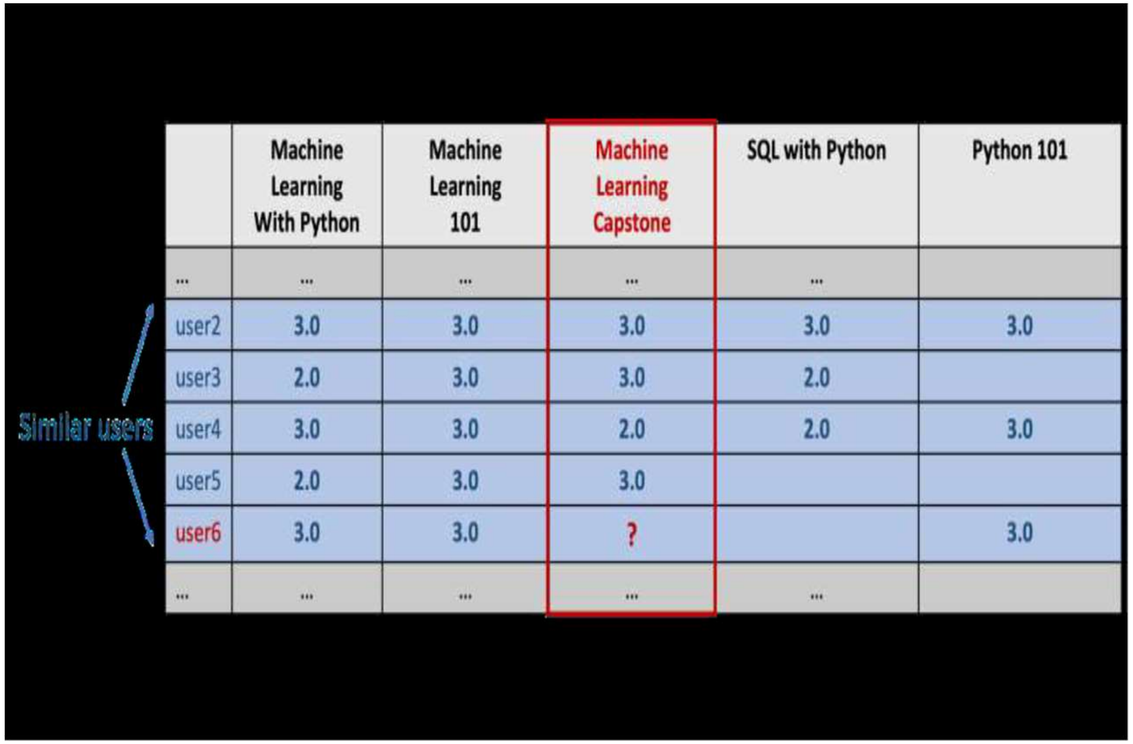
Generate Cosine Similarity Matix

Split Data into
Train+Test

Train The Proposed Model
(scikit library)

Make Prediction

KNN
High
Memory
Cost

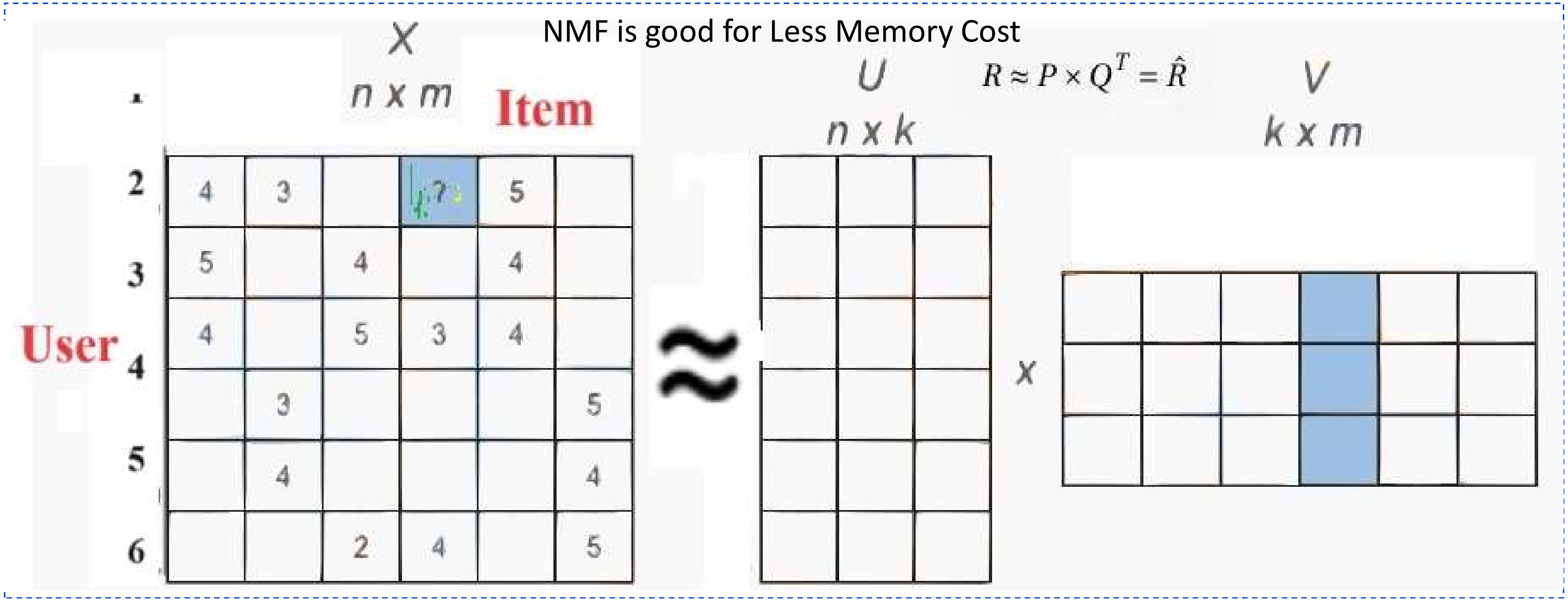


Similar users

	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
...	...	...	...	...	

MFOC1

Flowchart of Non-Negative Factor based recommender system



NMF is good for Less Memory Cost

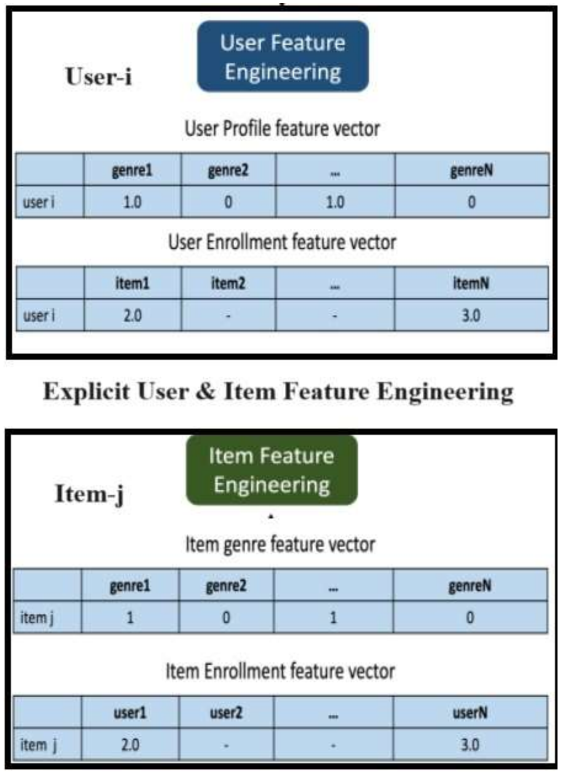
$R \approx P \times Q^T = \hat{R}$

4	3		?	5	
5		4		4	
4		5	3	4	
	3				5
	4				4
		2	4		5

Slide 18

MFOC1 Muhammad Farooq/Network/Service Operations Center/Technology/Islamabad, 03/11/2023

Flowchart of Neural Network Embedding based recommender system



User Profile feature vector

	genre1	genre2	...	genreN
user i	1.0	0	1.0	0

User Enrollment feature vector

	item1	item2	...	itemN
user i	2.0	-	-	3.0

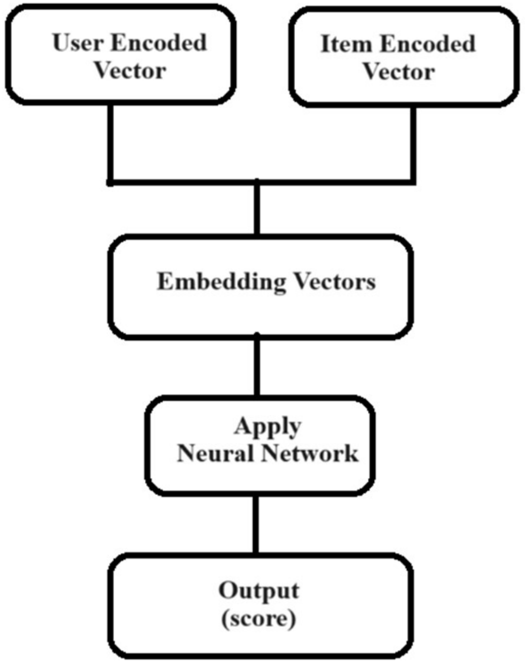
Explicit User & Item Feature Engineering

Item genre feature vector

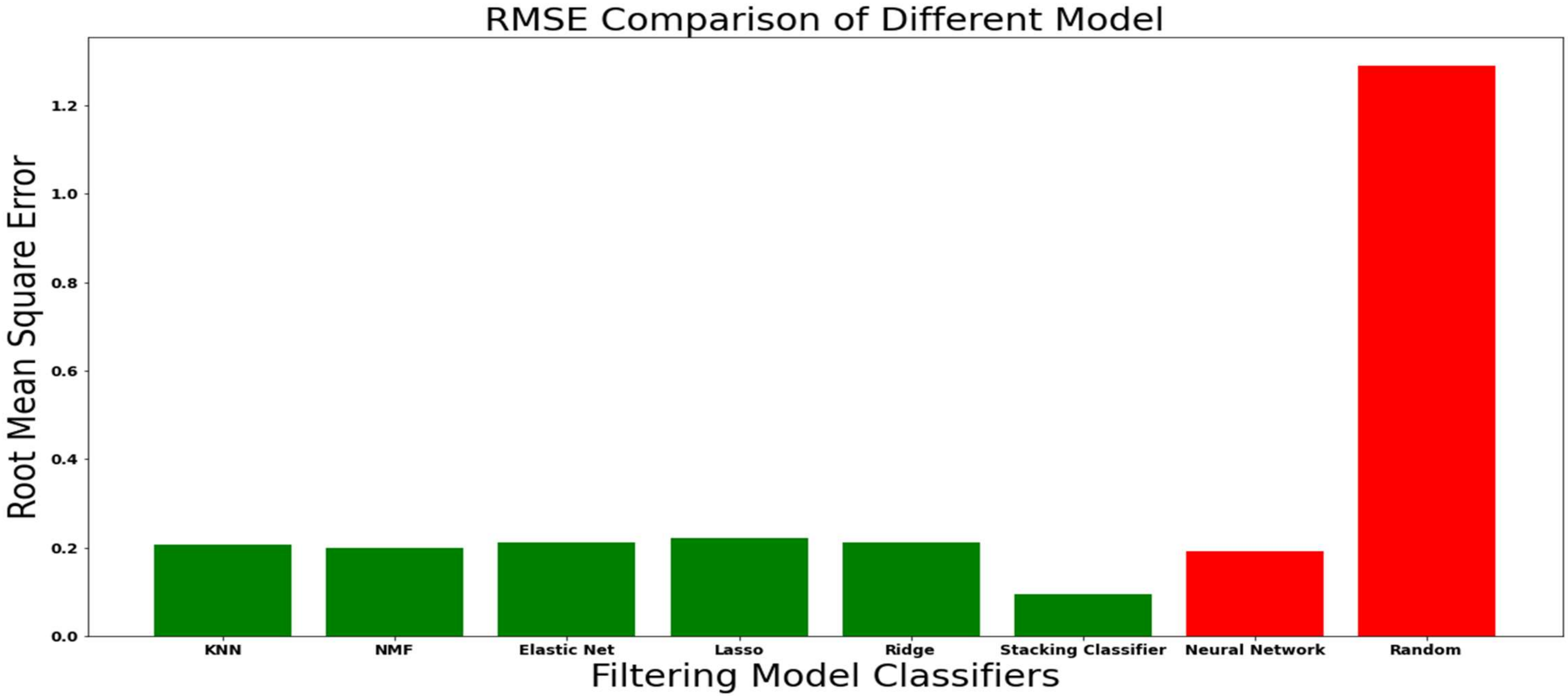
	genre1	genre2	...	genreN
item j	1	0	1	0

Item Enrollment feature vector

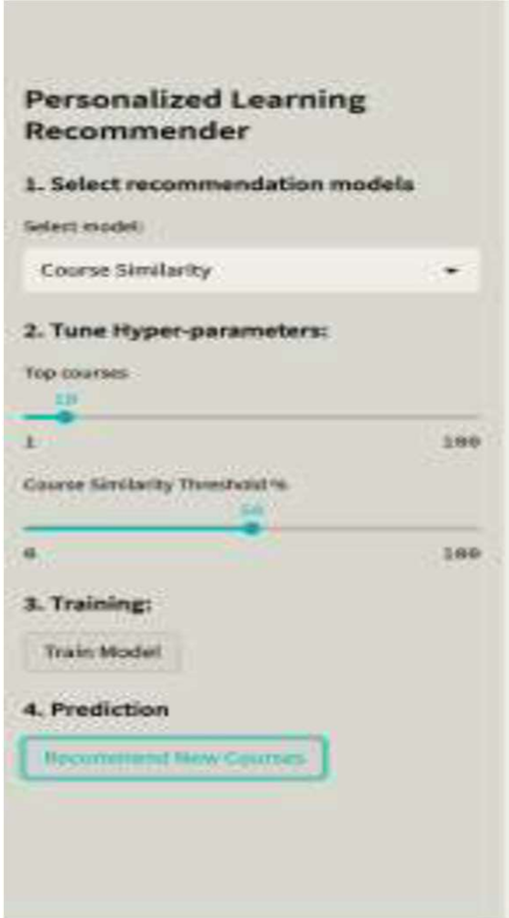
	user1	user2	...	userN
item j	2.0	-	-	3.0



Compare the performance of collaborative-filtering models



Optional: Build a course recommender system app with Streamlit App



Personalized Learning Recommender

1. Select recommendation models

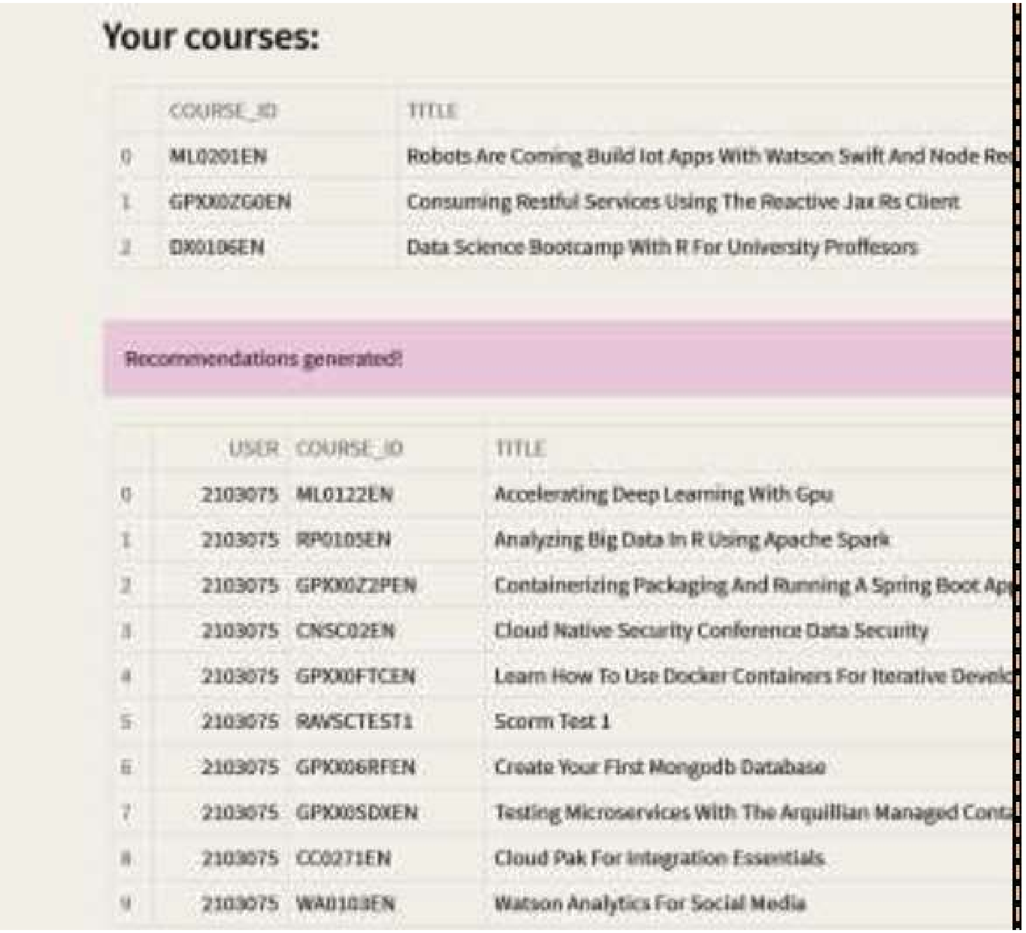
Select model: Course Similarity

2. Tune Hyper-parameters:

Top courses: 10

Course Similarity Threshold %: 50

3. Training:



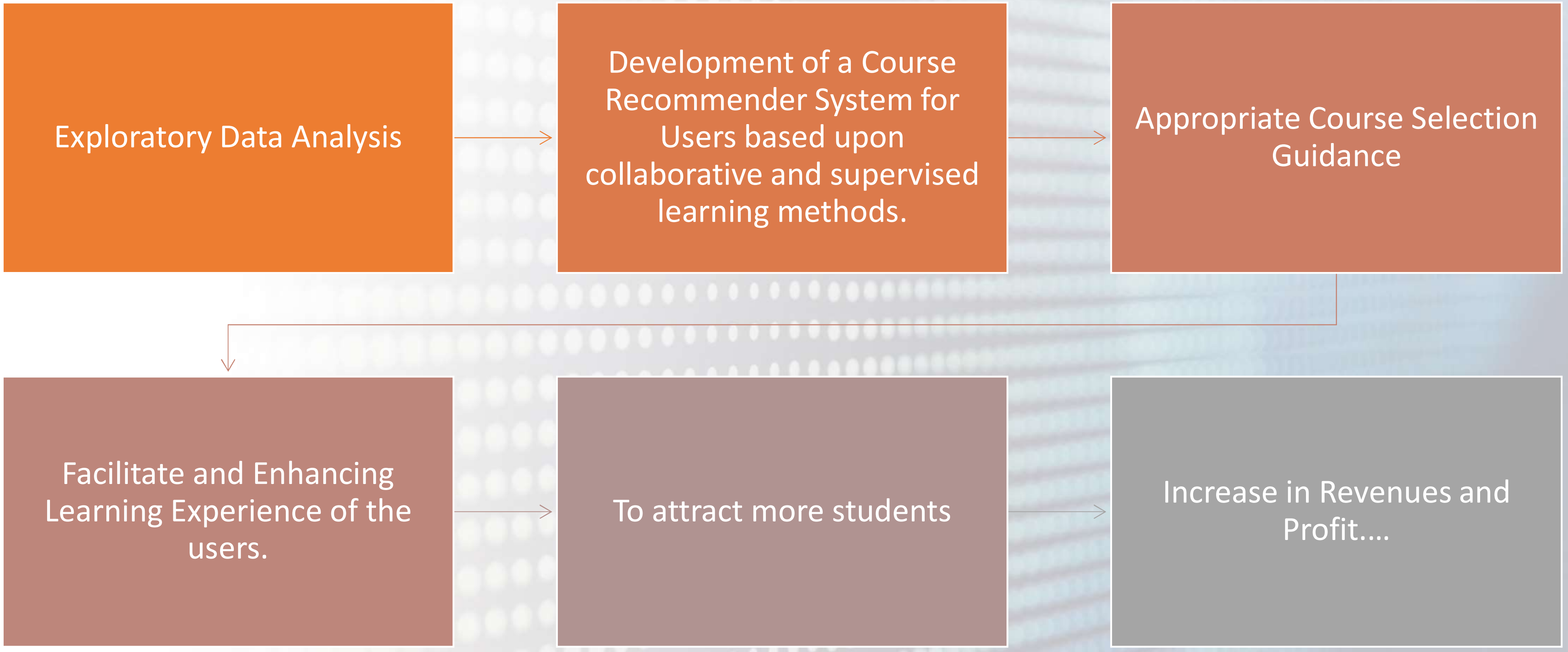
Your courses:

	COURSE_ID	TITLE
0	ML0201EN	Robots Are Coming Build Iot Apps With Watson Swift And Node Re
1	GPXX0ZG0EN	Consuming Restful Services Using The Reactive Jax Rs Client
2	DX0106EN	Data Science Bootcamp With R For University Proffesors

Recommendations generated!

	USER	COURSE_ID	TITLE
0	2103075	ML0122EN	Accelerating Deep Learning With Gpu
1	2103075	RP0105EN	Analyzing Big Data In R Using Apache Spark
2	2103075	GPXX0Z2PEN	Containerizing Packaging And Running A Spring Boot Ap
3	2103075	CNSC02EN	Cloud Native Security Conference Data Security
4	2103075	GPXX0FTCEN	Learn How To Use Docker Containers For Iterative Develo
5	2103075	RAVSCTEST1	Scorm Test 1
6	2103075	GPXX06RFEN	Create Your First Mongodb Database
7	2103075	GPXX0SDXEN	Testing Microservices With The Arquillian Managed Conta
8	2103075	CC0271EN	Cloud Pak For Integration Essentials
9	2103075	WA0103EN	Watson Analytics For Social Media

Conclusions



Exploratory Data Analysis

Development of a Course Recommender System for Users based upon collaborative and supervised learning methods.

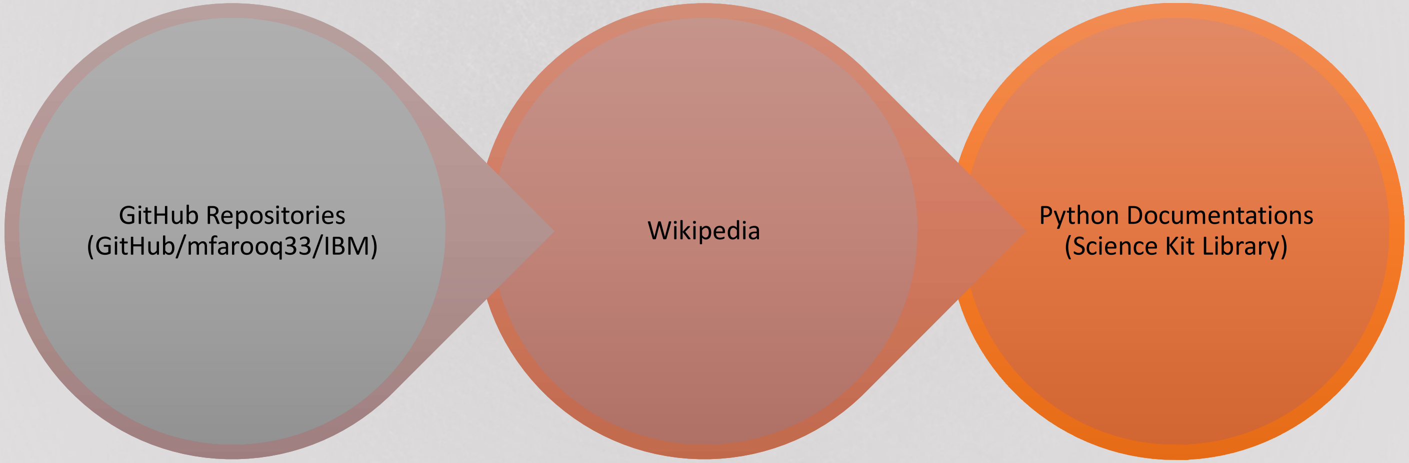
Appropriate Course Selection Guidance

Facilitate and Enhancing Learning Experience of the users.

To attract more students

Increase in Revenues and Profit….

Appendix



GitHub Repositories (GitHub/mfarooq33/IBM)

Wikipedia

Python Documentations (Science Kit Library)