Course4U
Build a Personalized Online Course Recommender System with Machine Learning

Presented by: Wahyu Ardhitama

Last Updated: May 31st. 2024

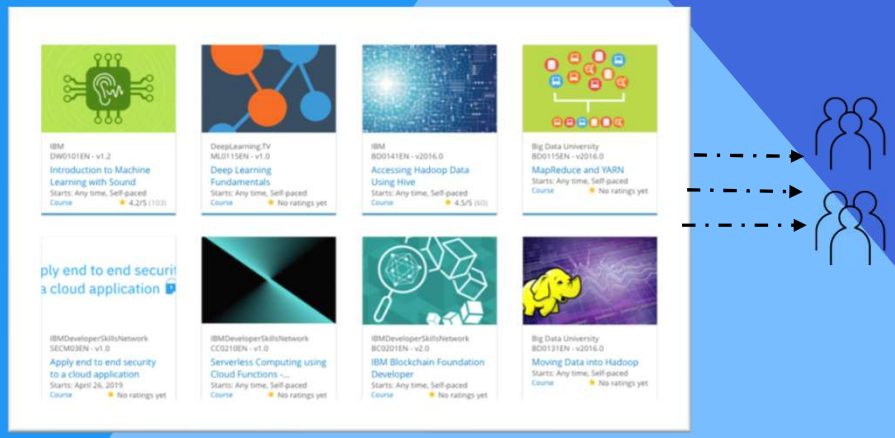


Table of Contents

Executive summary

➢ We use PACE framework

Data Analysis PACE Steps:

1. Plan/Prepare - import the relevant libraries and data

   Align project with business needs, requirements and contraints. Select an approriate machine learning model based on the problem and business context. KDD: Selection, Data Wrangling (Pre-processing and Transformation).

2. Analyze - Explarotary Data Analysis (EDA)

   Understanding data for accurate predictions, focus on the response variable (what the model predicts) and leverage exploratory data analysis to uncover patterns and address irregularities. KDD: Data Mining.

3. Construct - model

   Construct and evaluate model. KDD: Evaluation.

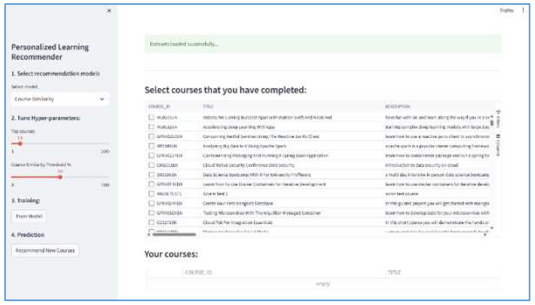
4. Execute - share

   Interpret model and share the story. KDD: Communicate to stakeholders.

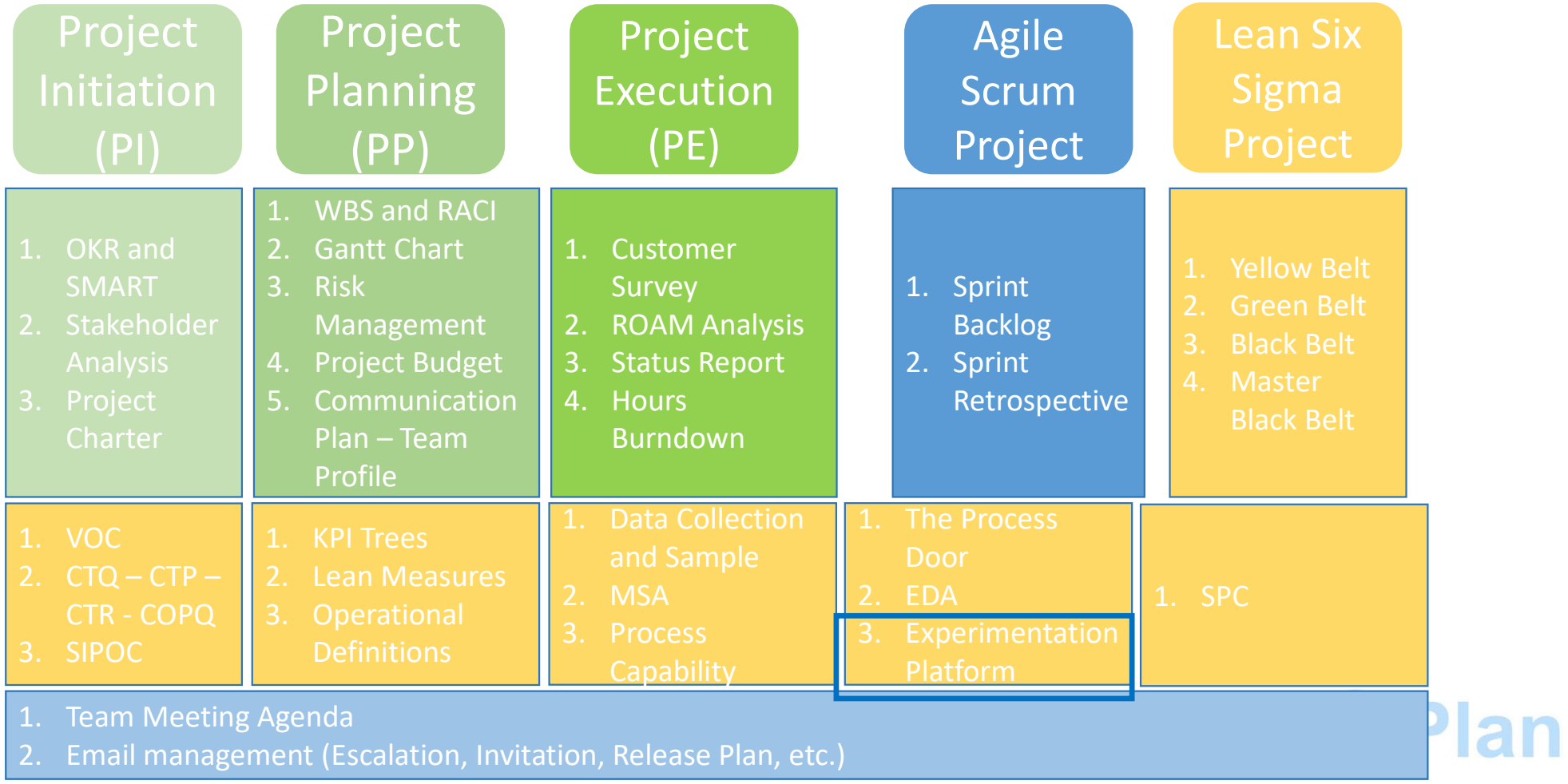
CO-V-FAST Principles: **C**lear/**C**lean/**C**ommunication/**C**ollaboration/**C**orrection, **O**bjective, **V**aluable, **F**ocus, **A**gile, **S**cientific and **T**ime-bound/Trustworthiness

➢ The recommender system apps

1. Recommender system with supervised and unsupervised learning

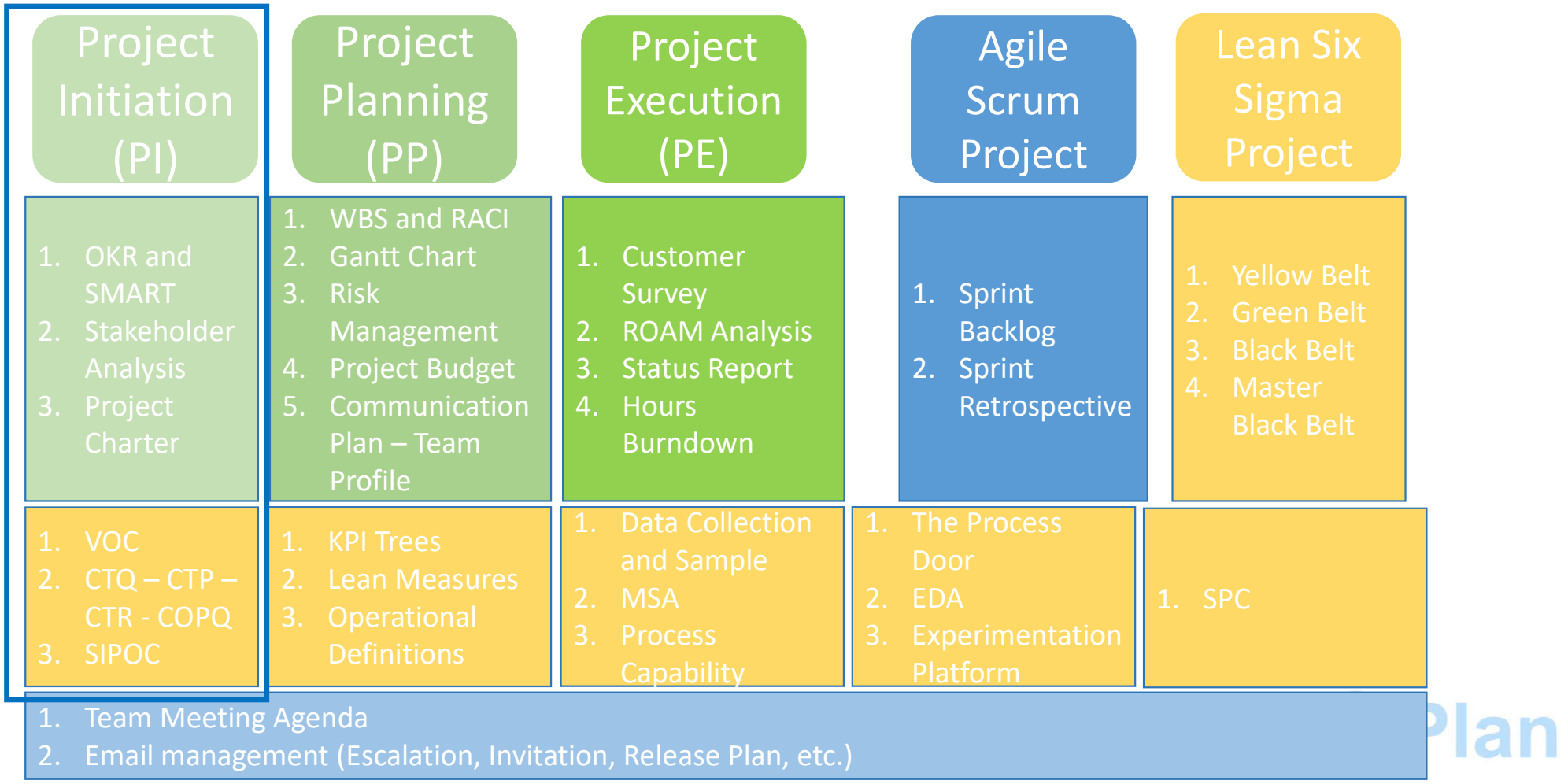


Project Management Flowchart



Project Initiation : Business case and Objective

Project Management Flowchart



Business Case

Course4U growing. having reached **~34.000 users** and over **233.000 enrollments** in **a year**.

Opportunity/Problem Statement:

- 25.000 users (**70%**) who have enrolled in **fewer than 10 courses**.
- Among them. **8.000** users have enrolled in only a single course**.**
- Only **less than 45%** of the total courses have been chosen by users.
- Encourage existing users to enroll in more than 10 courses.
- Acquiring new users.

Goals

Maximize user engagement. increase revenue streams. and solidify Course4U's position in the online education market. **257.500 enrollments next year.**

Campaign Objective:

Conversion/Enrollments

KPI:

Number of enrollments

(Tracked via online conversions and mobile - SDK)

Primary metric:

- Increase course **enrollments** by 10% by identifying and offering more engaging and relevant courses to **learners** . (courses enrolled in the list from 45% to > 50%)

Project Team

Plan

Project Team

Project Sponsor: Director of Customer Data

Project Lead: Head of Data Science and Machine Learning

Project Team: Director of Procurement, API Strategist, Data Warehousing Specialist, Data Governance Manager, Data Analyst, Director of IT, Project Manager, Product Manager, Marketing Promotion Manager, Financial Analyst, HR Recruitment and Training

Additional Stakeholders: Account Manager, Sales and Marketing Director, Investors

Data Science
1. Data collection
2. Data analysis
3. Modelling user profiles

Machine Learning Engineer
1. Algorithm development
2. Model training
3. Performance tuning

Data/Marketing Analyst
1. Feedback collection
2. Sentiment analysis
3. Insights generation

Data Engineer
1. ETL processes
2. Data pipeline maintenance
3. Data validation

Software Engineer
1. Notification system development
2. User preferences integration
3. Timeliness Assurance

Account/Content Manager
1. Content integration
2. Quality assurance
3. Course catalog management

UI/UX
1. Interface design
2. User testing
3. Responsive design

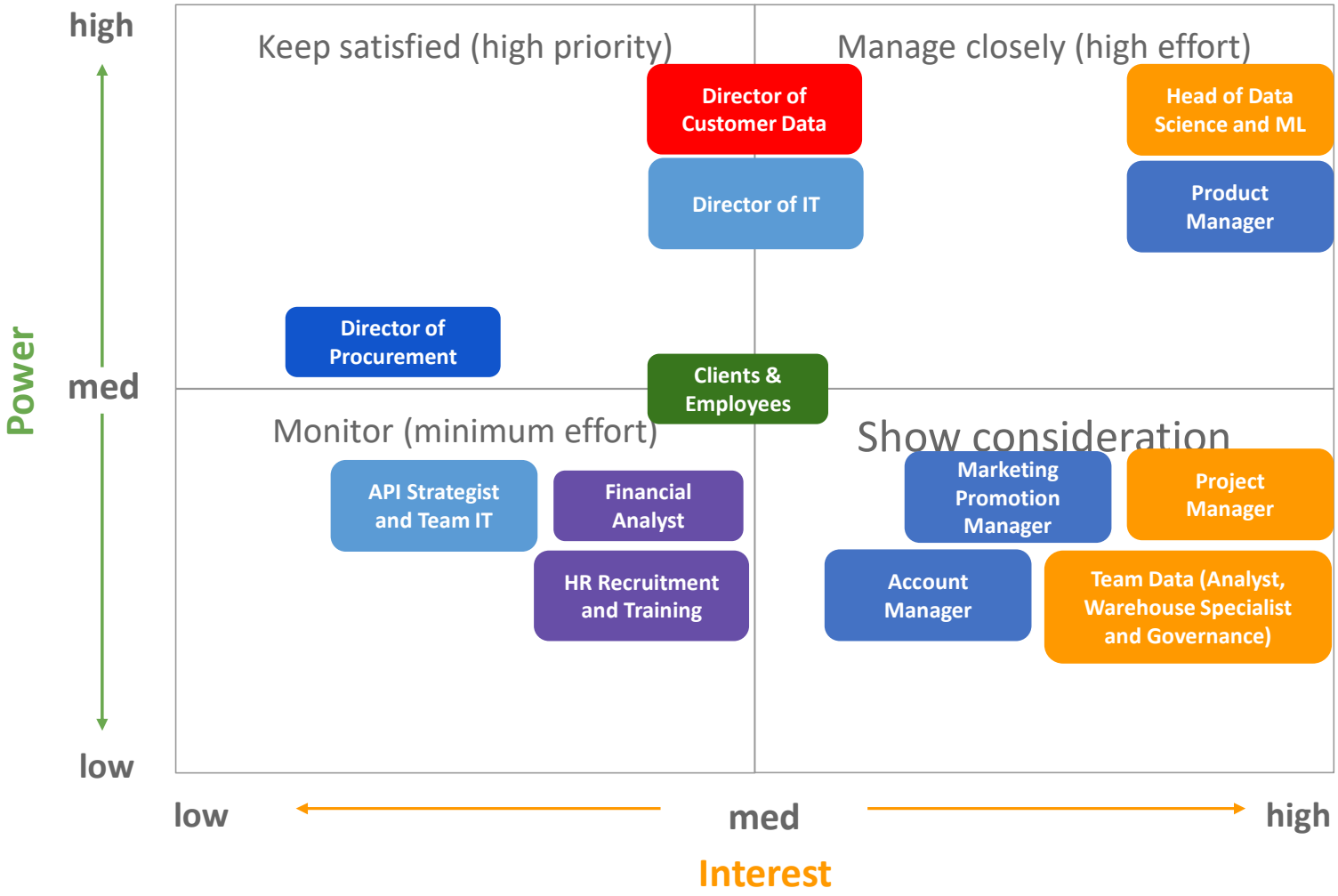
DevOps Engineer
1. Monitoring tools
2. Performance optimization
3. Incident response

Machine Learning Engineer
1. Parameter design
2. Testing and validation
3. Continous improvement

Product Manager
1. Feature development
2. User education
3. Customization option

Prioritizing stakeholders (power grid)

Plan



Analysis and Experimentation Team

Plan

Mission:

- Build a platform that is easy to integrate
- Foster a culture towards more data-driven decisions
- Accelerate innovation through trustworthy analysis and experimentation
- Empower the HiPPO (Highest Paid Person's Opinion) with valuable data

Team :

- Developers: Build the experimentation platform and the analysis tools
- Data Scientists – ML – Project (Program) Managers
- Admin

Analytics Objective

Explore and compare **various machine learning models** and **find one** with **the best performance** to improve learners' learning experience

C4U Recommender Systems :

- Quickly find new interested courses
- Better paving learning paths
- More learners interacting with more courses

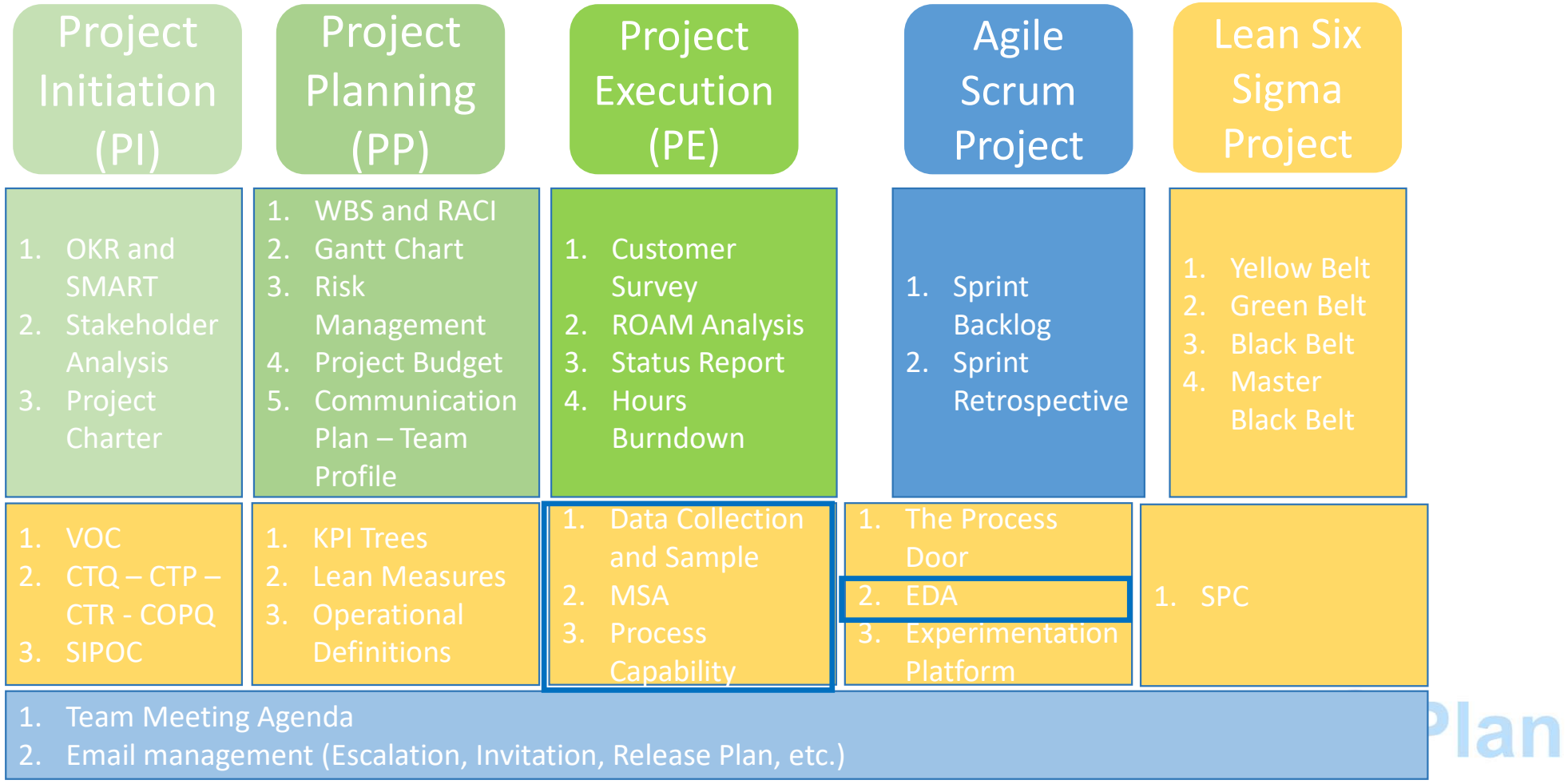
Hypothesis :

Recommender system delivers **more incremental value** of **enrollments** relative to **the current systems.**

Models and Findings

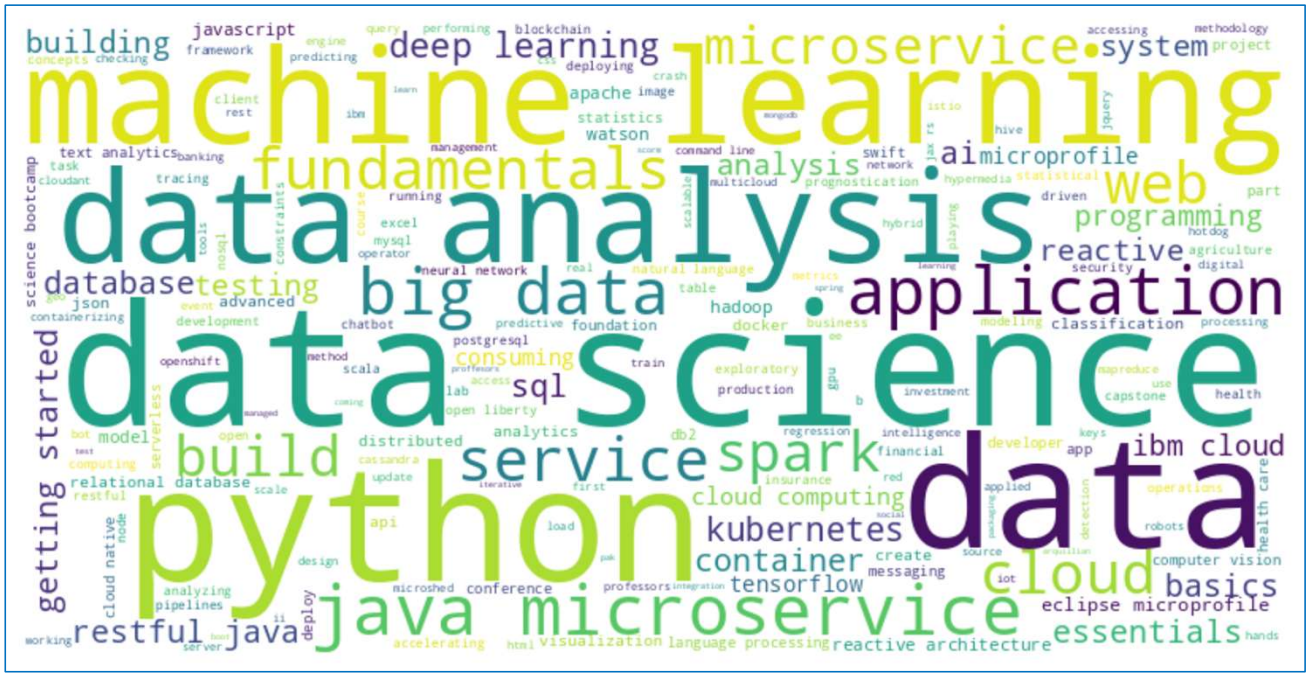
Exploratory Data Analysis

Project Management Flowchart



Keywords

In general. the courses are focused on **demanding IT skills**

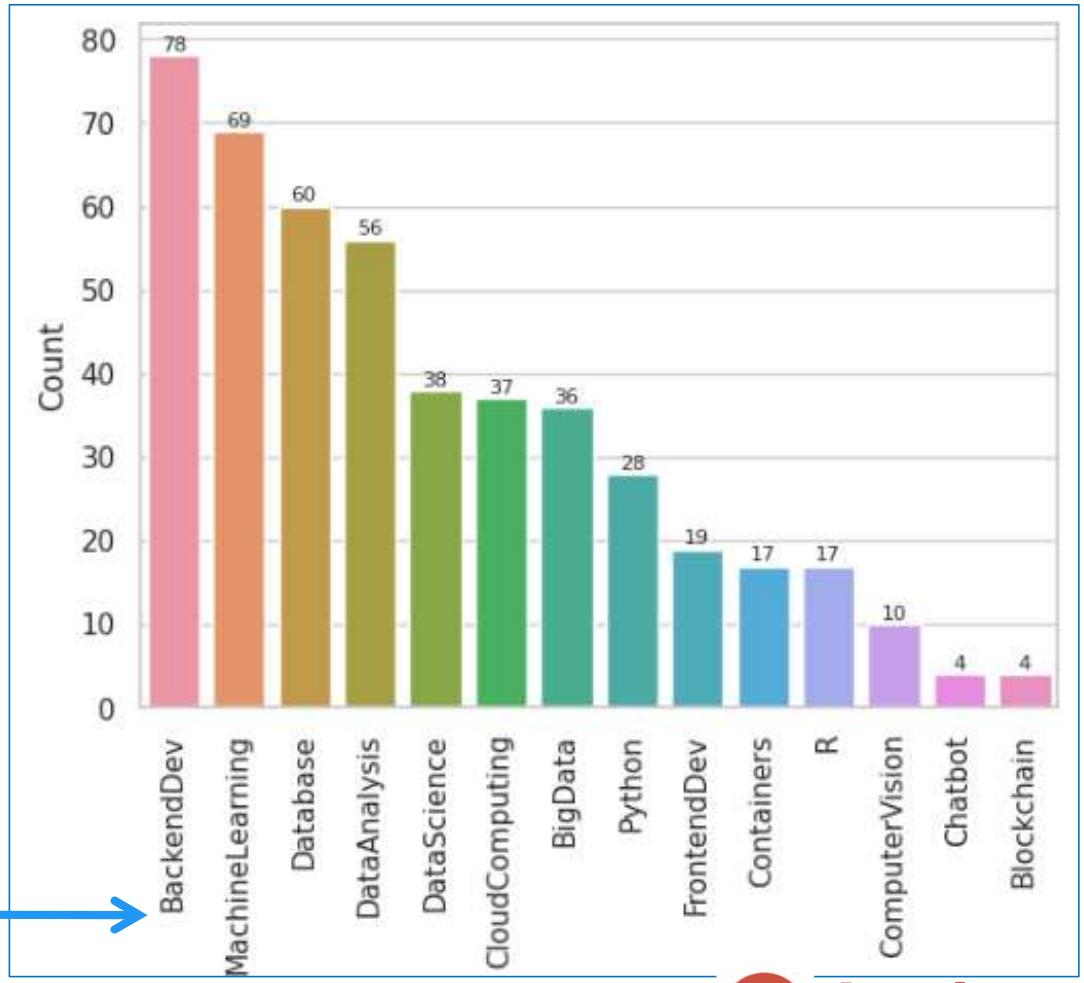


Plan

Course Genres

307 Total of courses offered

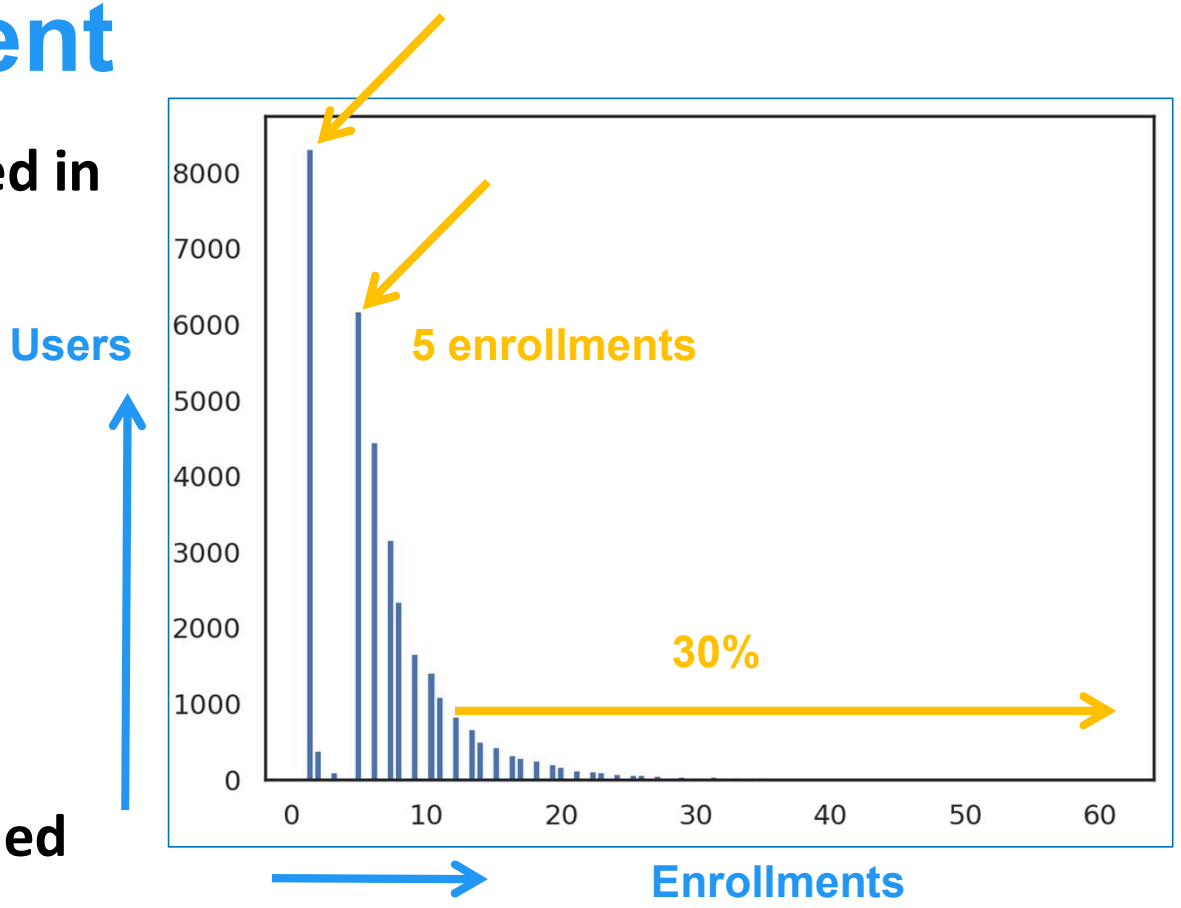
- Mostly related to **backend development**. **machine learning**. **database** and so on.
- Courses related to chatbot and blockchain are comparatively fewer.



Analyze

Course Enrollment

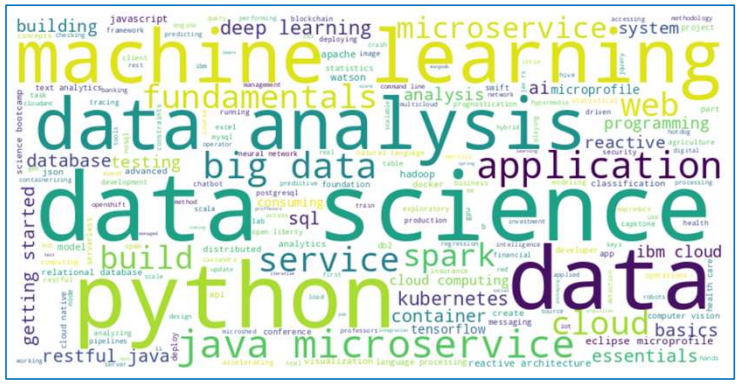
- **Over 8.000 users** have **enrolled in only one course.**
- The enrollment distribution is continuously declining. with **fewer users** as the number of **enrollments increases**.
- Only **30% of users have enrolled more than 10 courses.**



Analyze

Top 20 Courses

- **Just over 60%** of enrollment
- Related to data science. python. machine learning and so on as depicted on keywords
- Only **6.5 %** of total courses offered



	TITLE	Enrolls
0	python for data science	14936.0
1	introduction to data science	14477.0
2	big data 101	13291.0
3	hadoop 101	10599.0
4	data analysis with python	8303.0
5	data science methodology	7719.0
6	machine learning with python	7644.0
7	spark fundamentals i	7551.0
8	data science hands on with open source tools	7199.0
9	blockchain essentials	6719.0
10	data visualization with python	6709.0
11	deep learning 101	6323.0
12	build your own chatbot	5512.0
13	r for data science	5237.0
14	statistics 101	5015.0
15	introduction to cloud	4983.0
16	docker essentials a developer introduction	4480.0
17	sql and relational databases 101	3697.0
18	mapreduce and yarn	3670.0
19	data privacy fundamentals	3624.0

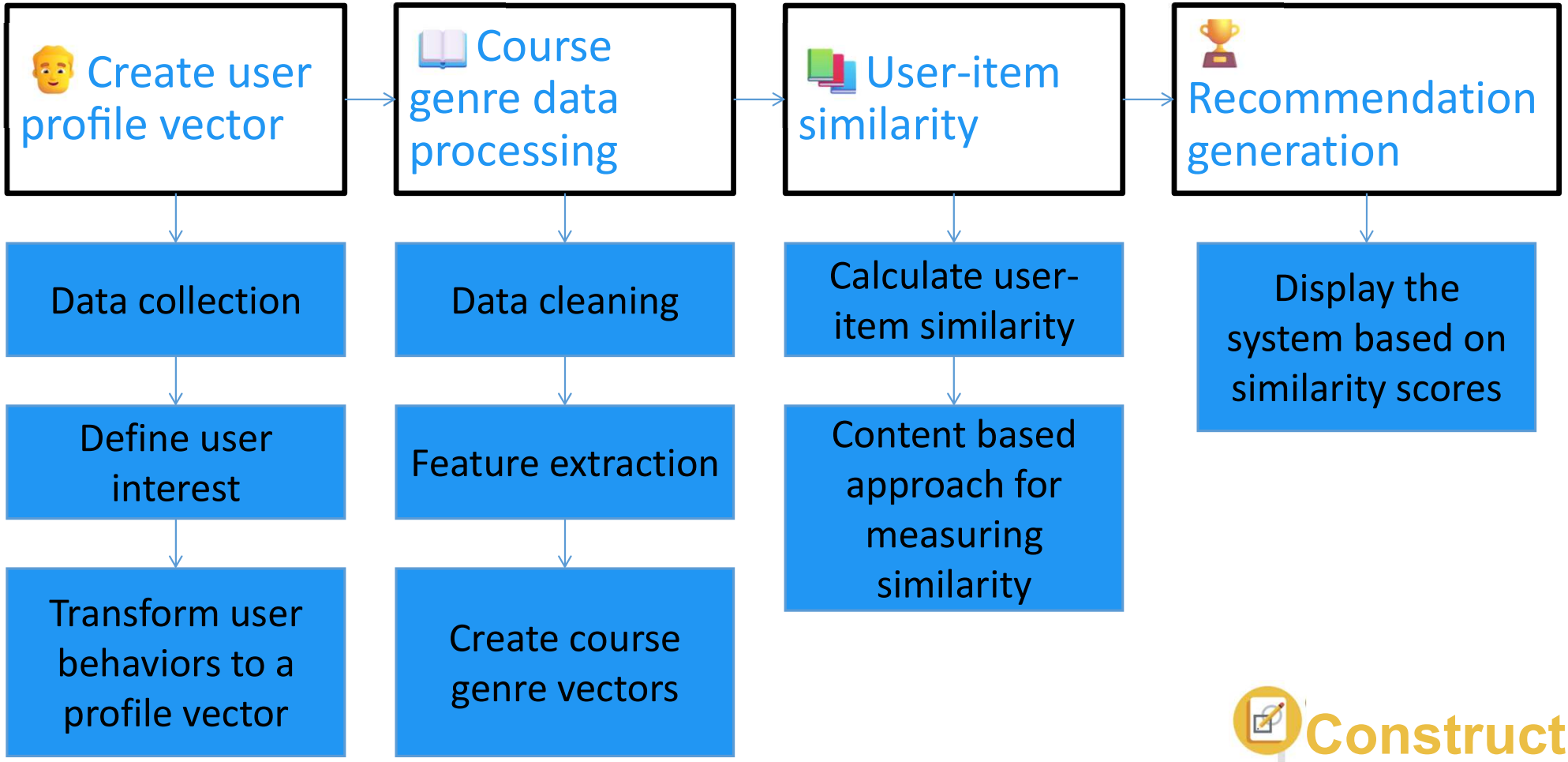
Analyze

Content-Based Recommender System using Unsupervised Learning

Models and Findings

Content-Based Recommender System using User Profile and Course Genres

User Profile Flowchart



Construct

Data

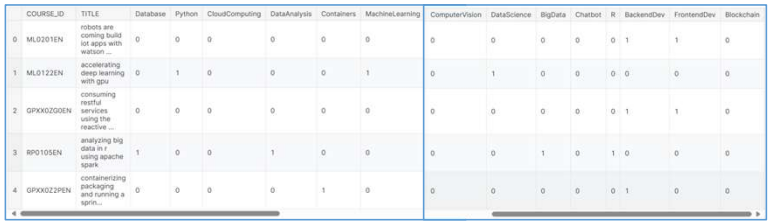
```
test_users_df.head()
```

	user	item	rating
0	1502801	RP0105EN	3.0
1	1609720	CNSC02EN	2.0
2	1347188	CO0301EN	3.0
3	755067	ML0103EN	3.0
4	538595	BD0115EN	3.0

test_users.df

	user	Database	Python	CloudComputing	DataAnalysis	Containers	MachineLearning	ComputerVision	DataScience	BigData	Chatbot	R	BackendDev	FrontendDev	Blockchain
0	2	52.0	14.0	6.0	43.0	3.0	33.0	0.0	29.0	41.0	2.0	18.0	34.0	9.0	6.0
1	4	40.0	2.0	4.0	28.0	0.0	14.0	0.0	20.0	24.0	0.0	6.0	6.0	0.0	2.0
2	5	24.0	8.0	18.0	24.0	0.0	30.0	0.0	22.0	14.0	2.0	14.0	26.0	4.0	6.0
3	7	2.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0
4	8	6.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	6.0	0.0	2.0	0.0	0.0	0.0

profile.df

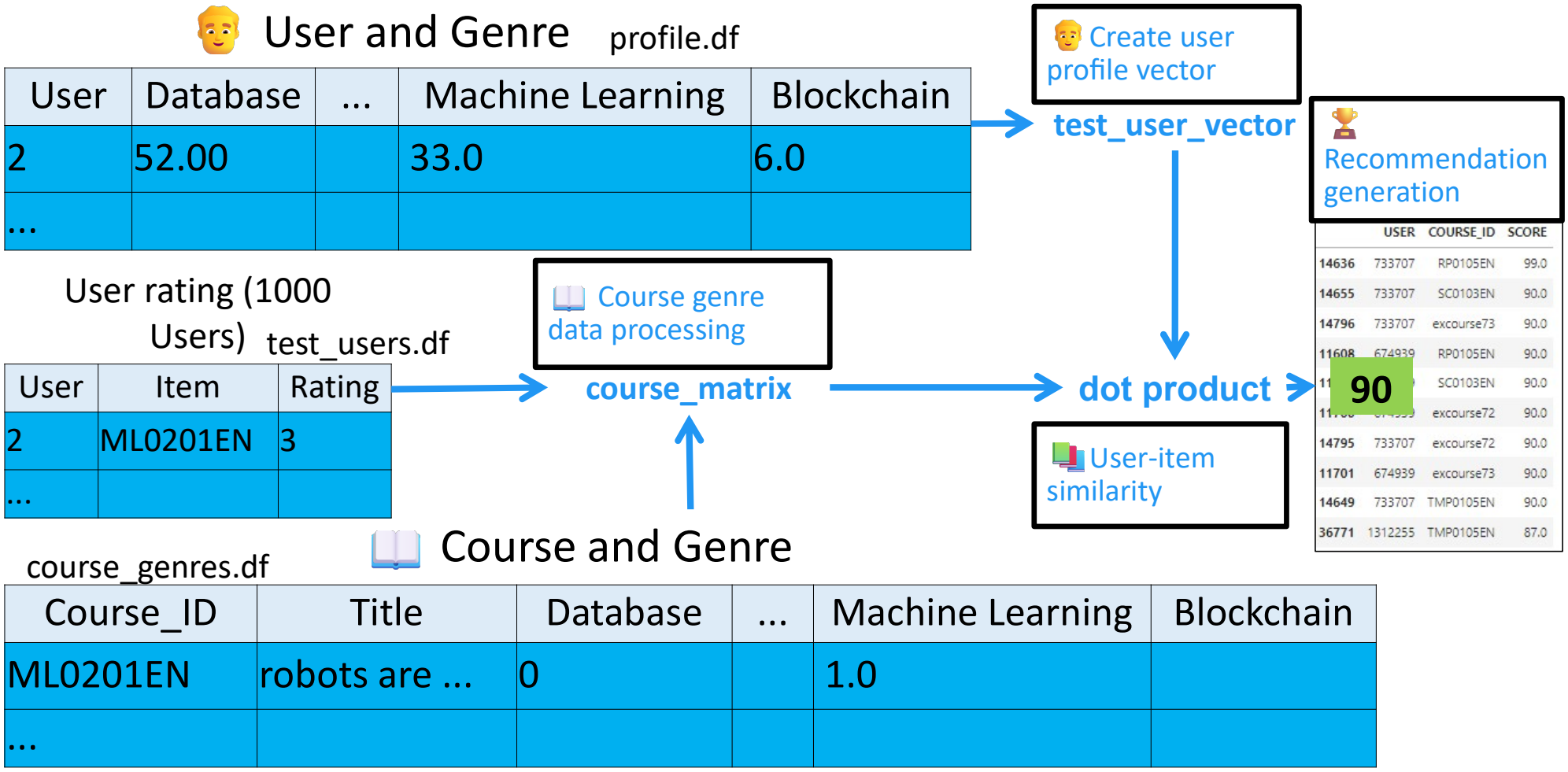


course_genres.df

course_df
(course_genre.csv)
307 courses
14 Genres

Construct

Recommendation Generation



Evaluation Results

score_threshold = 10

score_threshold = 10

	USER	COURSE_ID	SCORE
14636	733707	RP0105EN	99.0
14655	733707	SC0103EN	90.0
14796	733707	excourse73	90.0
11608	674939	RP0105EN	90.0
11618	674939	SC0103EN	90.0
...	...	...	...
3053	435051	excourse42	10.0
2680	418401	BD0131EN	10.0
3051	435051	excourse10	10.0
3050	435051	excourse05	10.0
2690	418401	GPXX0M6UEN	10.0

53411 rows × 3 columns

>5K recommendations for >850 users of 1.000 users(>85%)

Top 10 recomended all users courses

	USER	COURSE_ID	SCORE	TITLE
0	733707	RP0105EN	99.0	analyzing big data in r using apache spark
1	733707	SC0103EN	90.0	spark overview for scala analytics
2	733707	excourse73	90.0	analyzing big data with sql
3	674939	RP0105EN	90.0	analyzing big data in r using apache spark
4	674939	SC0103EN	90.0	spark overview for scala analytics
5	674939	excourse72	90.0	foundations for big data analysis with sql
6	733707	excourse72	90.0	foundations for big data analysis with sql
7	674939	excourse73	90.0	analyzing big data with sql
8	733707	TMP0105EN	90.0	getting started with the data apache spark ma...
9	1312255	TMP0105EN	87.0	getting started with the data apache spark ma...

Scores >= 99

Scores >= 87

3 users 5 courses
733…, 674…,131…

- Big data (data analysis) sql
- Foundation
- Apache spark

Construct

Recommendation based on user profile

User profile 1078030

	user	item	rating	COURSE_ID	TITLE
0	1078030	DA0101EN	3.0	DA0101EN	data analysis with python
1	1078030	ST0101EN	3.0	ST0101EN	statistics 101
2	1078030	ML0122ENv1	3.0	ML0122ENv1	accelerating deep learning with gpu
3	1078030	ML0120ENv2	3.0	ML0120ENv2	deep learning with tensorflow
4	1078030	DV0101EN	3.0	DV0101EN	data visualization with python
5	1078030	ML0115EN	3.0	ML0115EN	deep learning 101
6	1078030	ML0101ENv3	3.0	ML0101ENv3	machine learning with python
7	1078030	PY0101EN	3.0	PY0101EN	python for data science

Participate in 8 courses

- Data analysis
- Deep learning
- Python

Top 10 recomended score's for 1078030

	COURSE_ID	SCORE	TITLE
0	ML0122EN	30.0	accelerating deep learning with gpu
1	excourse21	30.0	applied machine learning in python
2	excourse22	30.0	introduction to data science in python
3	ML0101EN	30.0	machine learning with python
4	GPXX0IBEN	27.0	data science in insurance basic statistical a...
5	excourse49	24.0	applied machine learning in python
6	GPXX0D14EN	24.0	build a personal movie recommender with django
7	GPXX0YMEEN	24.0	launch an ai hotdog detector as a serverless p...
8	excourse54	21.0	exploratory data analysis for machine learning
9	excourse20	21.0	python and statistics for financial analysis

Score 30

Score 21

10 recommended courses

- Machine learning
- Deep learning
- Python
- Data analysis
- Data science

Construct

Recommendation based on user profile

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

10 recommendations
Highest score 30
Lowest score 21

10 recommended courses

- Machine learning
- Deep learning
- Python
- Data analysis
- Data science

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

172 recommendations
Highest score 99
Lowest score 12

10 recommended courses

- Big data (data analysis) sql
- Apache spark
- Foundation

Lowest score 69

User profile 674939

Participate in 15 courses

- Spark
- Hadoop
- Big data

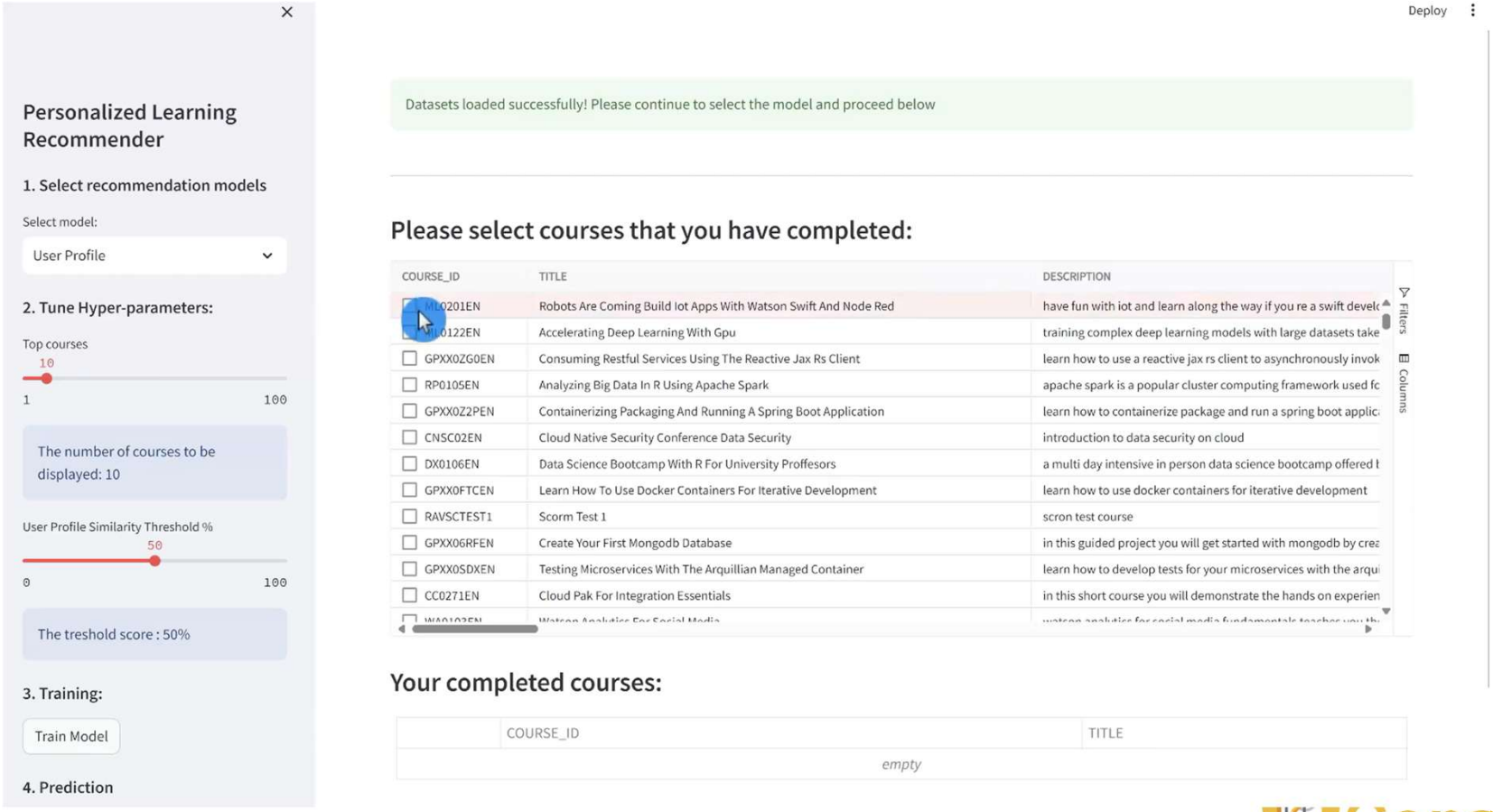
101 recommendations
Highest score 90
lowest score 12

10 recommended courses

- Big data (data analysis) sql
- Nosql
- Spark

Lowest score 78

Recommendation based on user profile



Construct

Rating 2 Users

score_threshold = 10

User 2057052 courses enrollment

	user	item	rating	COURSE_ID	TITLE
0	2057052	DS0132EN	2.0	DS0132EN	data ai jumpstart your journey
1	2057052	DS0101EN	3.0	DS0101EN	introduction to data science
2	2057052	ML0101ENv3	3.0	ML0101ENv3	machine learning with python
3	2057052	PY0101EN	3.0	PY0101EN	python for data science
4	2057052	DB0101EN	3.0	DB0101EN	sql and relational databases 101

Participate in 5 courses

Recomended courses for 2057052 & 1871627

USER	COURSE_ID	SCORE	TITLE

No recommendations; threshold 0; their highest scores are 2

User 1871627 courses enrollment

	user	item	rating	COURSE_ID	TITLE
0	1871627	CC0103EN	3.0	CC0103EN	ibm cloud essentials v3
1	1871627	ML0101ENv3	3.0	ML0101ENv3	machine learning with python
2	1871627	ML0103EN	3.0	ML0103EN	digital analytics regression
3	1871627	ST0101EN	3.0	ST0101EN	statistics 101
4	1871627	PY0101EN	3.0	PY0101EN	python for data science
5	1871627	DV0151EN	3.0	DV0151EN	data visualization with r
6	1871627	DS0101EN	3.0	DS0101EN	introduction to data science
7	1871627	DS0103EN	3.0	DS0103EN	data science methodology
8	1871627	CC0101EN	3.0	CC0101EN	introduction to cloud
9	1871627	ML0115EN	3.0	ML0115EN	deep learning 101
10	1871627	DB0101EN	3.0	DB0101EN	sql and relational databases 101
11	1871627	OS0101EN	3.0	OS0101EN	introduction to open source
12	1871627	CB0103EN	3.0	CB0103EN	build your own chatbot
13	1871627	DS0132EN	2.0	DS0132EN	data ai jumpstart your journey

Participate in 14 courses

Construct

Summary

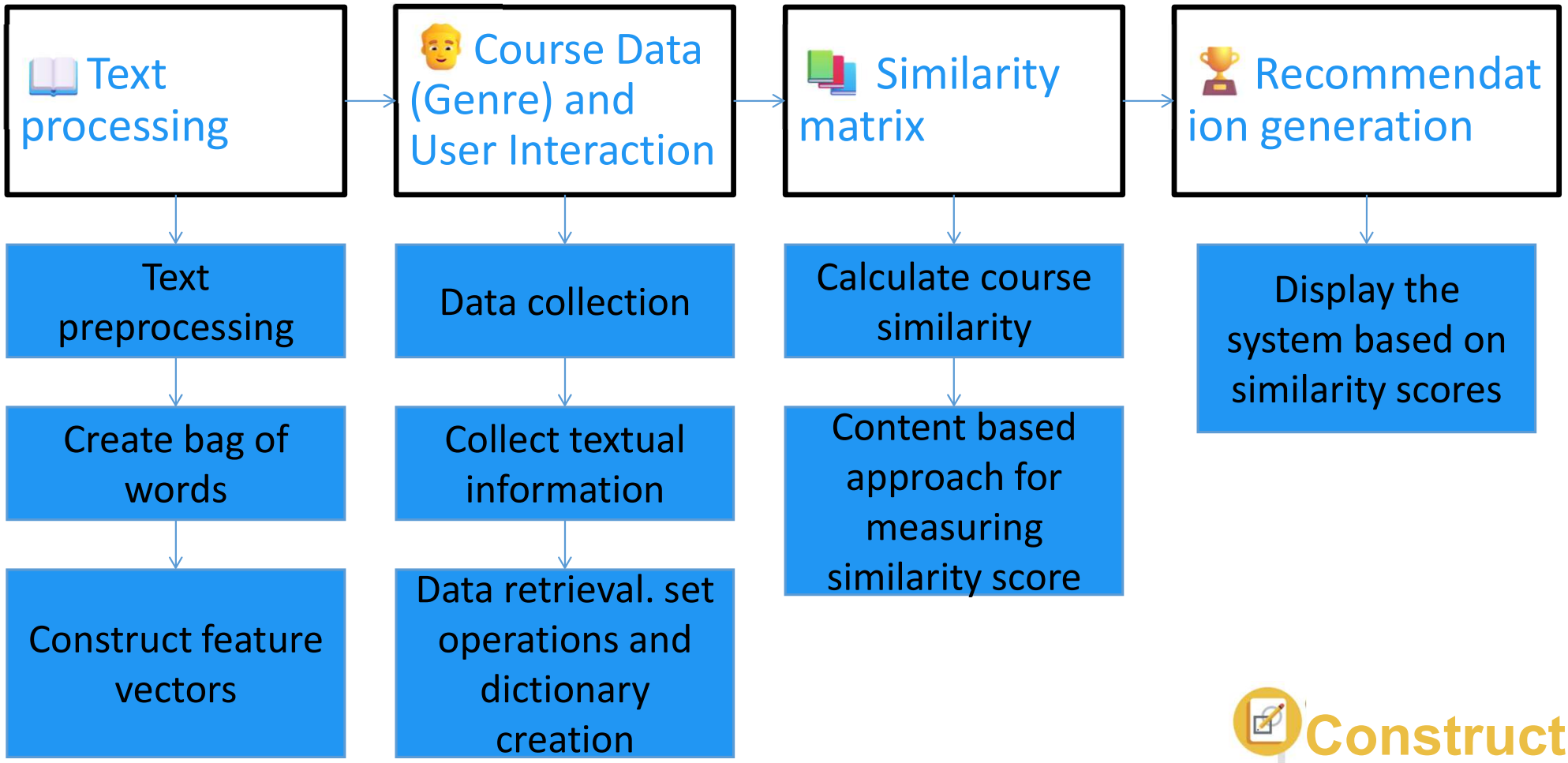
The idea

1. **Personalized suggestions based on a user's preferences and past interactions.**
2. **Has insights into a user's preferences and can recommend courses that align with their interests.**
3. **Can make recommendations even for new users with limited interaction history.**
4. **Based on explicit features (e.g.. genres) that users can understand.**
5. **Adjust the threshold for users who have courses with a rating of 2.**

Models and Findings

Content-Based Recommender System using Course Similarity

Flowchart



Construct

Files

sim_df.head()

Course Similarity

0	1	2	...	305	306
1.000000	0.088889	0.088475		0.039276	0.121113
...					

bow_df.head()

Bag of Words

doc_index	doc_id	token	bow
0	ML0201EN	ai	2
...			

course_df.head()

📖 Course 1

Course_ID	Title	Description
ML0151EN	machine learning ...	this machine learning...

Construct

Recommendation Generation

User and Course Data

Text processing

Recommendation generation

Course 1(index=200)

Course_ID	Title	Description
ML0151EN	machine learning ...	this machine learning...

Course 2(index=158)

Course_ID	Title	Description
ML0101ENv3	machine learning with ...	machine learning can be...

Similarity matrix

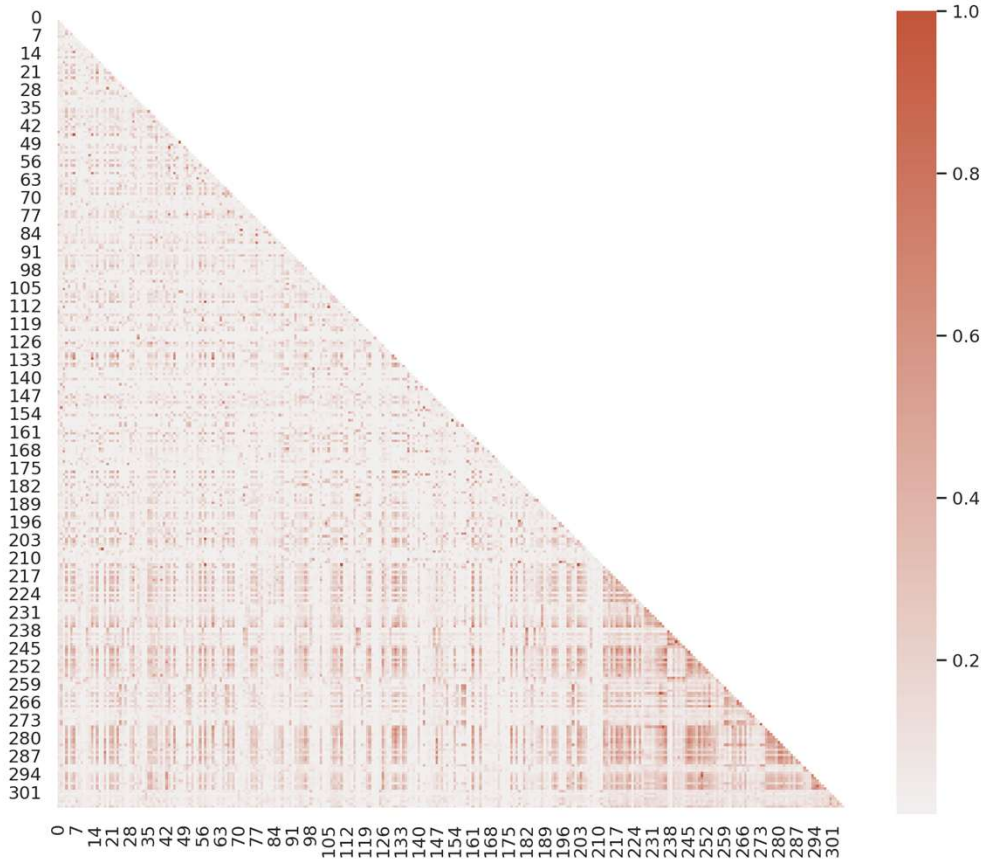
Similiraty calculation:
Cosine. Euclidean. Jaccard index...

	USER	COURSE_ID	SCORE
0	37465	ML0120EN	1.000000
1	37465	ML0120ENv3	1.000000
2	37465	excourse36	0.739704
3	37465	excourse23	0.739704
4	37465	DV0151EN	0.723536
...	...	...	...
15995	2087663	excourse62	0.647502
		excourse47	0.634755
		excourse60	0.615568
15998	2087663	excourse46	0.612054
15999	2087663	excourse09	0.608330

75%

Construct

Similarity matrix



Hot spots shown. Possible to build a recommender system based on course similarities.

Construct

Evaluation Results

Machine learning courses
ML0151EN & ML0101ENv3

score_threshold = 0.6

	USER	COURSE_ID	SCORE
0	37465	ML0120EN	1.000000
1	37465	ML0120ENv3	1.000000
2	37465	excourse36	0.739704
3	37465	excourse23	0.739704
4	37465	DV0151EN	0.723536
...	...	...	...
15995	2087663	excourse62	0.647502
15996	2087663	excourse47	0.634755
15997	2087663	excourse60	0.615568
15998	2087663	excourse46	0.612054
15999	2087663	excourse09	0.608330

16000 rows × 3 columns

16000 recommendations for 1000 users of 1.000 users(100%)

Top 10 recomended all users courses

	USER	COURSE_ID	SCORE	TITLE
0	37465	ML0120EN	1.000000	deep learning with tensorflow
1	37465	ML0120ENv3	1.000000	deep learning with tensorflow
2	37465	excourse36	0.739704	data analysis using python
3	37465	excourse23	0.739704	data analysis using python
4	37465	DV0151EN	0.723536	data visualization with r
5	37465	excourse32	0.722018	introduction to data analytics
6	37465	ML0122ENv3	0.707107	accelerating deep learning with gpus
7	37465	excourse38	0.681638	data analysis with python
8	37465	excourse33	0.664509	excel basics for data analysis
9	37465	ML0151EN	0.662622	machine learning with r

→ **Scores >= 100%**

→ **Scores >= 66%**

1 users **10 courses**
User 37465

- **Data analysis**
- **Deep learning**
- **Python**

Construct

Recommendation based on course similarity

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

16 recommendations
Highest score 100%
Lowest score 60%

10 recommended courses

- Data analysis
- Deep learning
- Python

Lowest score 66%

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

15 recommendations
Highest score 100%
Lowest score 60%

10 recommended courses

- Deep learning
- Data science
- Big data
- Machine learning

Lowest score 66%

User profile 674939

Participate in 15 courses

- Spark
- Hadoop
- Big data

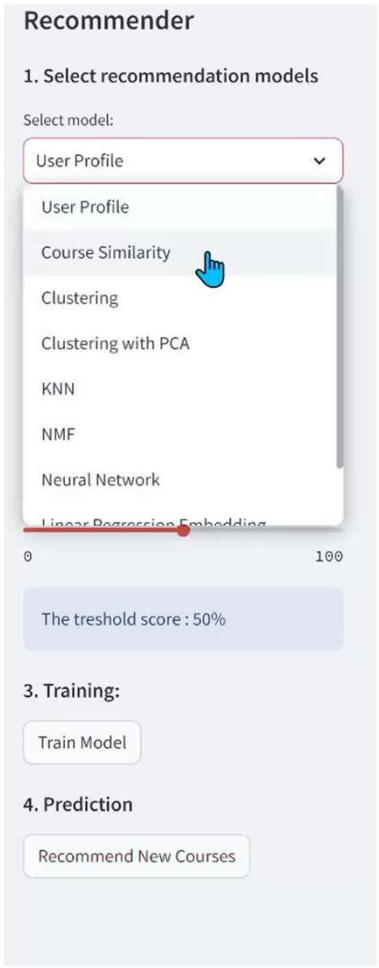
4 recommendations
Highest score 70%
Lowest score 61%

4 recommended courses

- Big data

Lowest score 61%

Recommendation based on course similarity



Deploy

1	ML0122EN	Accelerating Deep Learning With Gpu

After selection, please continue to tune your hyper-parameters then press "Train Model".

Recommendations generated! These are the 10 courses that we recommend for you using User Profile model.

	USER	COURSE_ID	TITLE	SCORE
0	2103296.000000	GPXX0D14EN	Build A Personal Movie Recommender With Django	100.00%
1	2103296.000000	ML0101EN	Machine Learning With Python	100.00%
2	2103296.000000	excourse21	Applied Machine Learning In Python	100.00%
3	2103296.000000	excourse22	Introduction To Data Science In Python	100.00%
4	2103296.000000	GPXX0ZG0EN	Consuming Restful Services Using The Reactive Jax Rs Client	66.67%
5	2103296.000000	DX0108EN	Data Science Bootcamp With Python For University Professors Advance	66.67%
6	2103296.000000	OS0101EN	Introduction To Open Source	66.67%
7	2103296.000000	PA0103EN	Predicting Customer Satisfaction	66.67%
8	2103296.000000	PA0107EN	Predicting Financial Performance Of A Company	66.67%
9	2103296.000000	GPXX0IBEN	Data Science In Insurance Basic Statistical Analysis	66.67%

truct

Evaluation Results

Machine learning courses
ML0151EN & ML0101ENv3

Top 10 recomended score's for 2057052 (rating 2)

	USER	COURSE_ID	SCORE	TITLE
8946	2057052	DS0110EN	0.732941	data science with open data
8947	2057052	excourse63	0.694563	a crash course in data science
8948	2057052	DAI101EN	0.668994	data ai essentials
8949	2057052	ML0151EN	0.662622	machine learning with r
8950	2057052	excourse22	0.647502	introduction to data science in python
8951	2057052	excourse62	0.647502	introduction to data science in python
8952	2057052	excourse65	0.638641	data science fundamentals for data analysts
8953	2057052	excourse47	0.634755	machine learning for all
8954	2057052	excourse46	0.612054	machine learning

Score 73%

Score 61%

9 recommendations
lowest score 61%

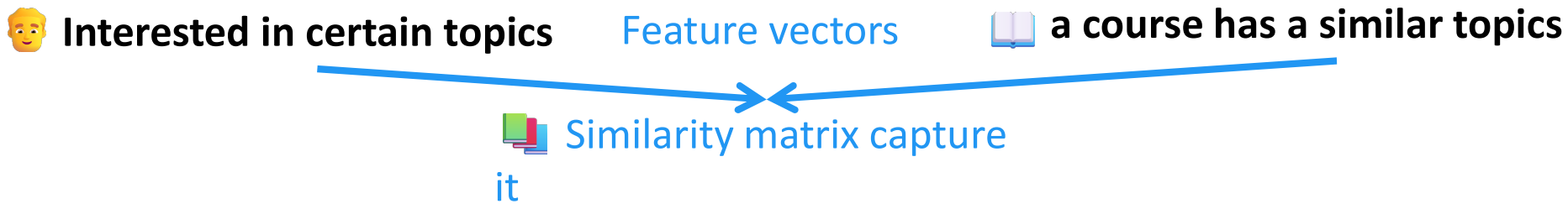
9 recommended courses

- Data science
- Data ai
- Machine learning

Construct

Summary

The idea

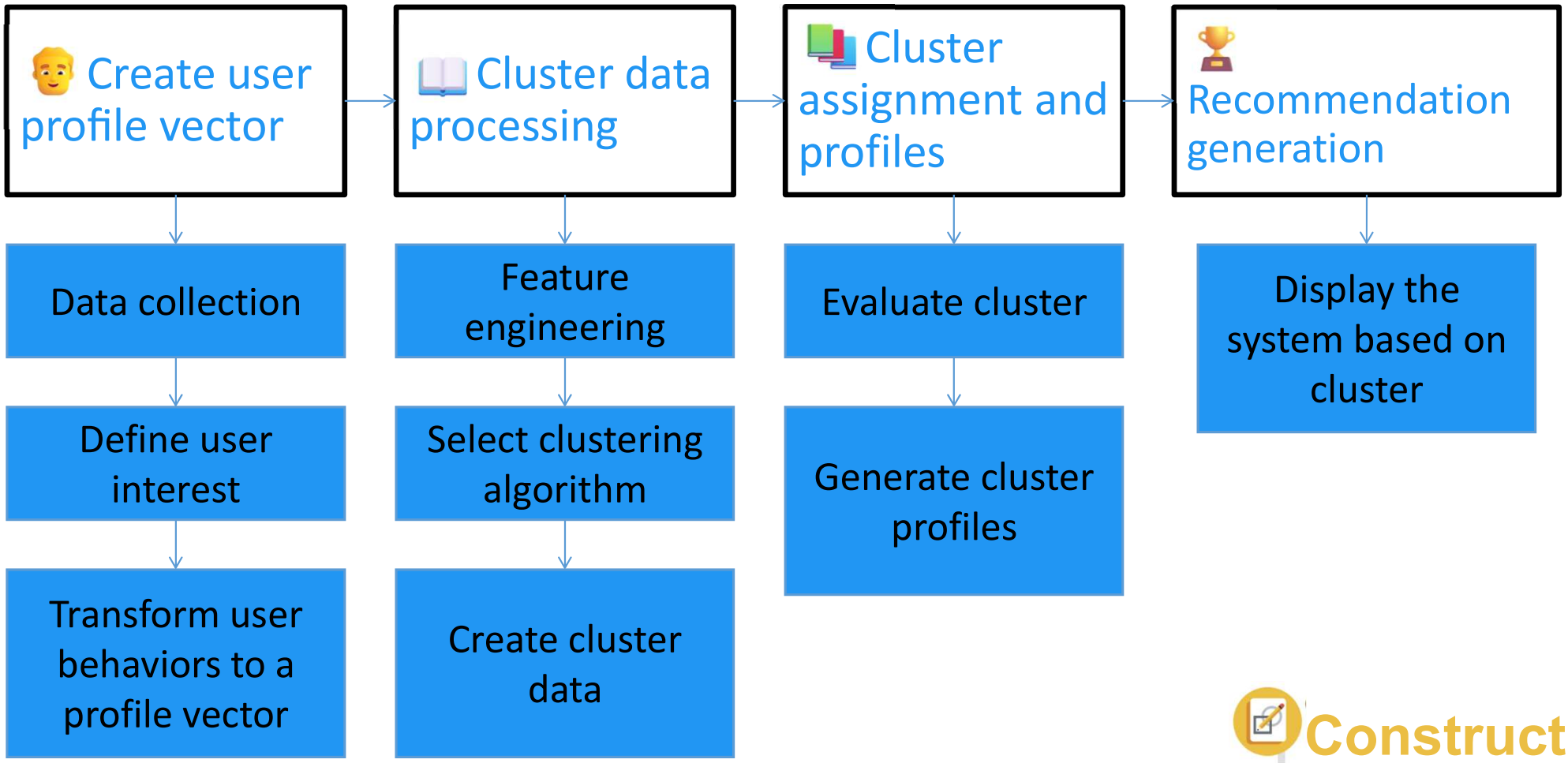


1. **Offer personalized suggestions based on the intrinsic characteristics of courses.**
2. **Based on explicit features (e.g.. genres) that users can understand.**
3. **Can make recommendations even for new users with limited interaction history.**
4. **Based on specific features of courses. allowing users to interpret and understand the reasons behind each recommendation.**

Models and Findings

Clustering-Based Recommender System

Flowchart



Construct

Recommendation Generation

Create user profile vector

Cluster data processing

Cluster assignment and profiles

Recommendation generation

User and Genre

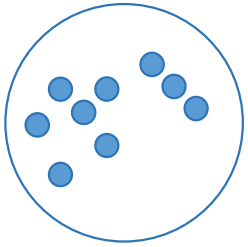
User	Python	Database	...	ML	Blockchain
1		52.00		33.0	6.0
...					

User profile standard scaler

User	Python	Database	...	ML	Blockchain
1	-0.3533	4.52998		2.3685	0.519419
...					

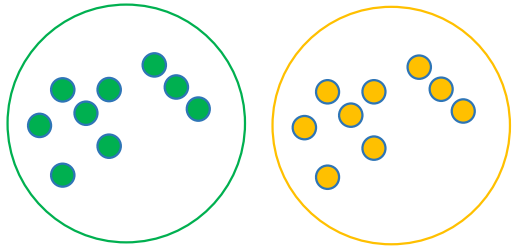
Machine learning (ML) learners:

- ML 101
- ML with python



Database learners:

- SQL 101
- SQL with python

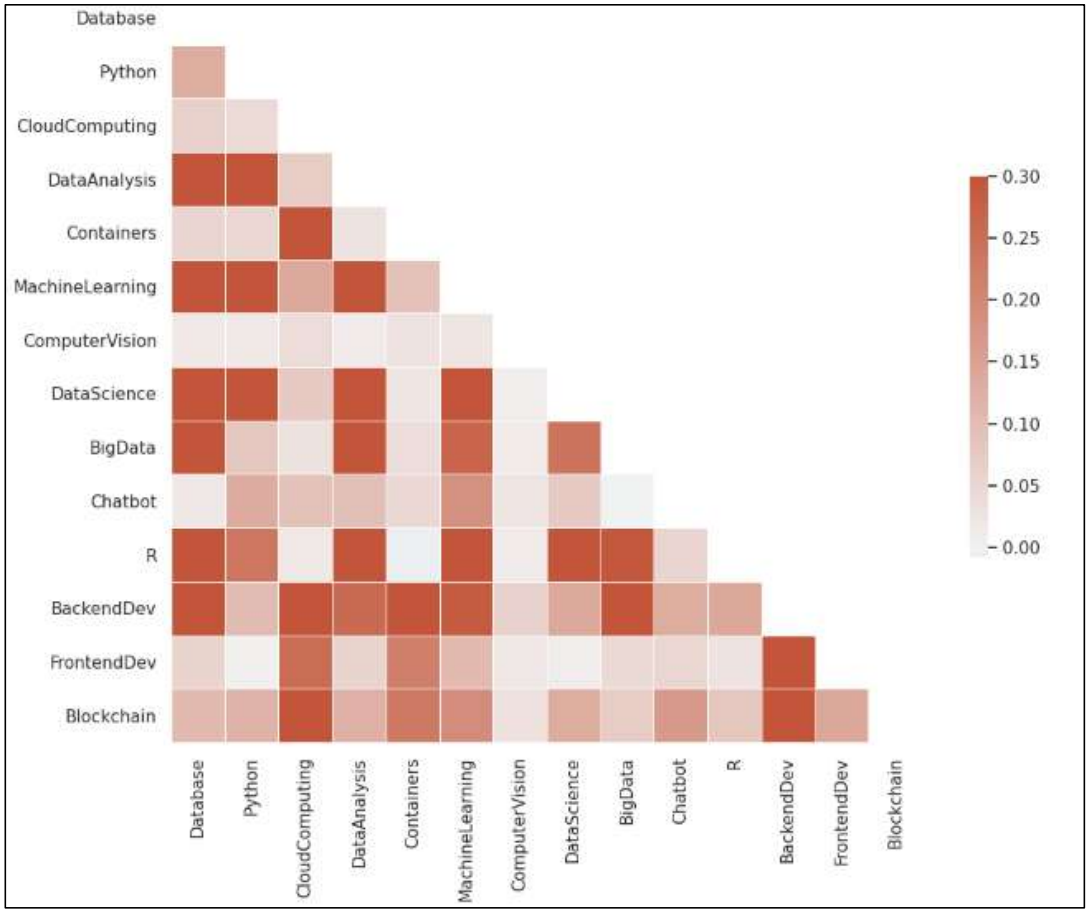


Python:

- Python 101
- Python for analysis

Construct

Covariance matrix of the user profile feature vectors with 14 features



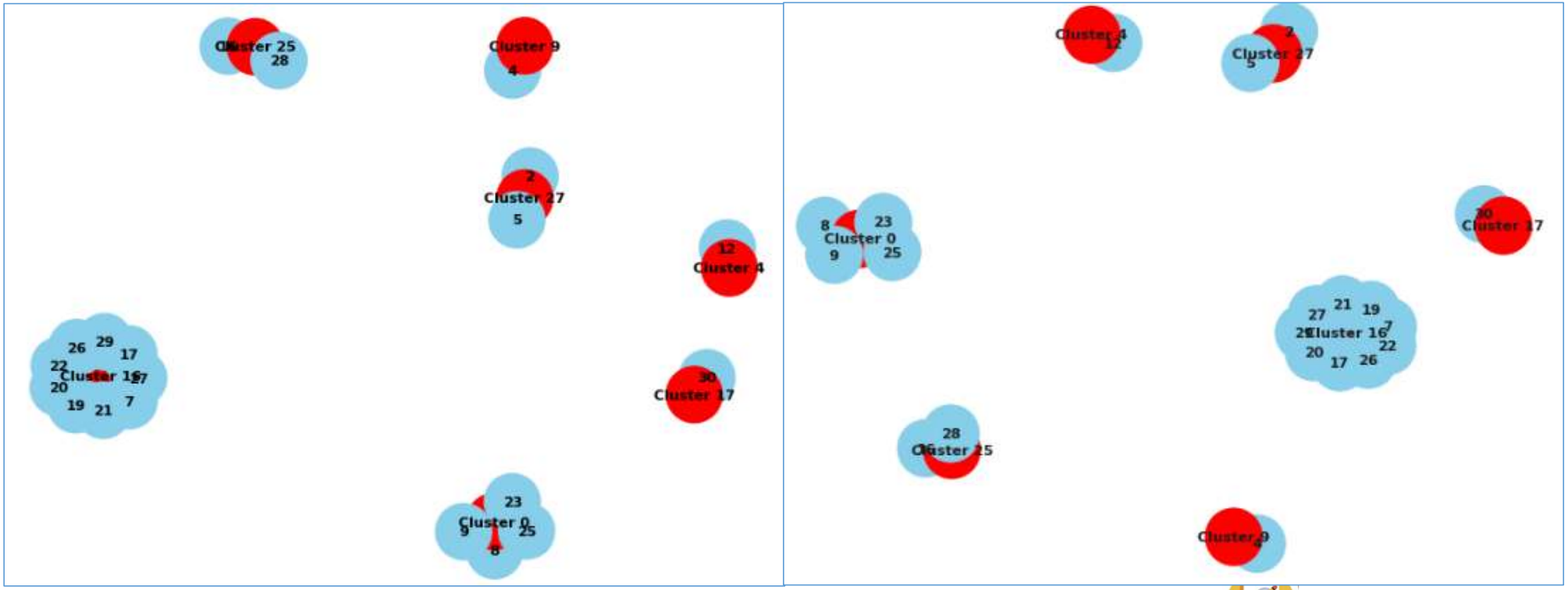
Hot spots shown. Possible to build a recommender system based on cluster.

Construct

Evaluation Results

Top 20 recomended course based on cluster user profiler feature vectors

PCA



Construct

Recommendation based on users and courses

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

User profile 674939

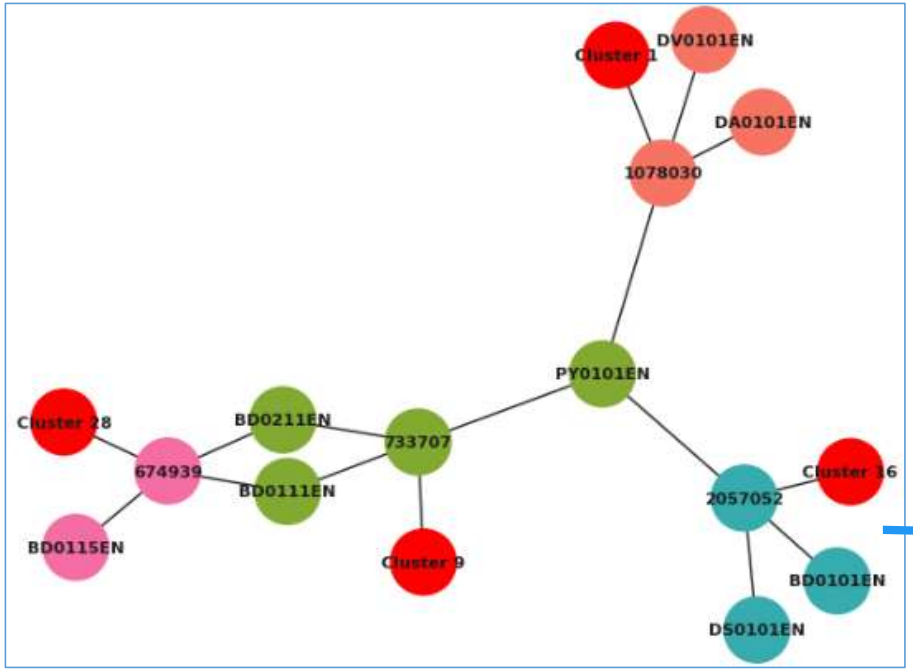
Participate in 15 courses

- Spark
- Hadoop
- Big data

Construct

Course recommendations based on the popular courses in the same cluster

	user	cluster	rec_1	rec_2	rec_3	title_1	title_2	title_3
102	1078030	1	PY0101EN	DA0101EN	DV0101EN	python for data science	data analysis with python	data visualization with python
151	674939	28	BD0111EN	BD0115EN	BD0211EN	hadoop 101	mapreduce and yarn	spark fundamentals i
221	2057052	16	DS0101EN	BD0101EN	PY0101EN	introduction to data science	big data 101	python for data science
298	733707	9	BD0111EN	PY0101EN	BD0211EN	hadoop 101	python for data science	spark fundamentals i



→ Rating 2

Construct

Recommendation based on cluster

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

3 recommended courses

- Python
- Data analysis
- Data science
- Data visualization

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

3 recommended courses

- Python
- Data science
- Hadoop
- Spark

User profile 674939

Participate in 15 courses

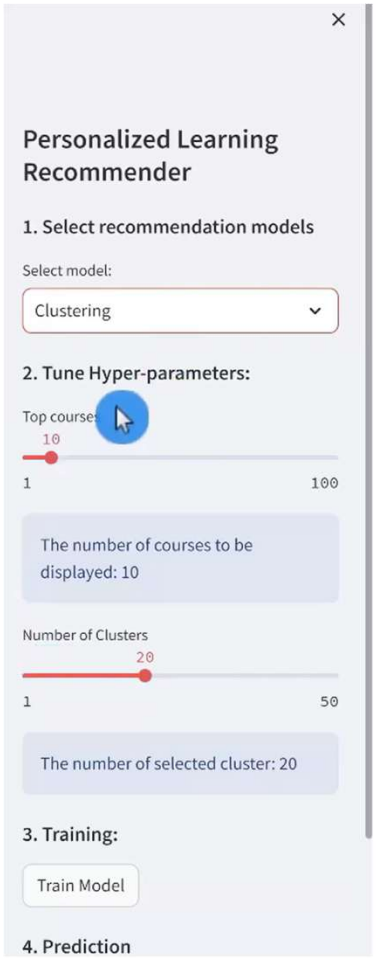
- Spark
- Hadoop
- Big data

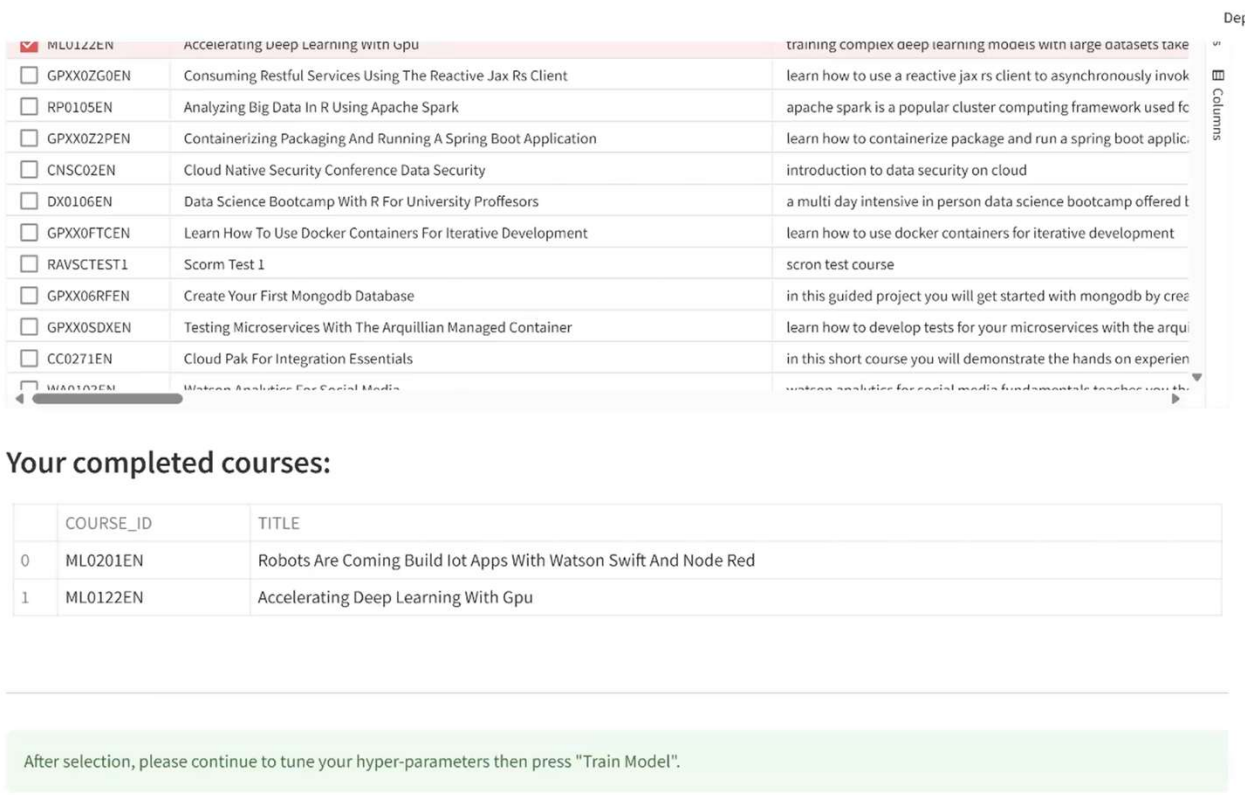
3 recommended courses

- Hadoop
- Maproduce
- Spark

Construct

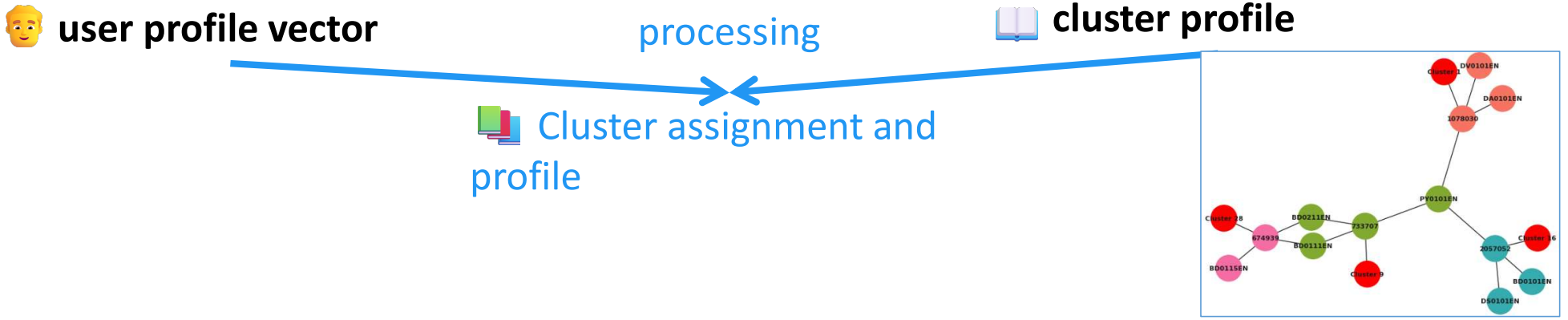
Recommendation based on cluster and PCA





Summary

The idea



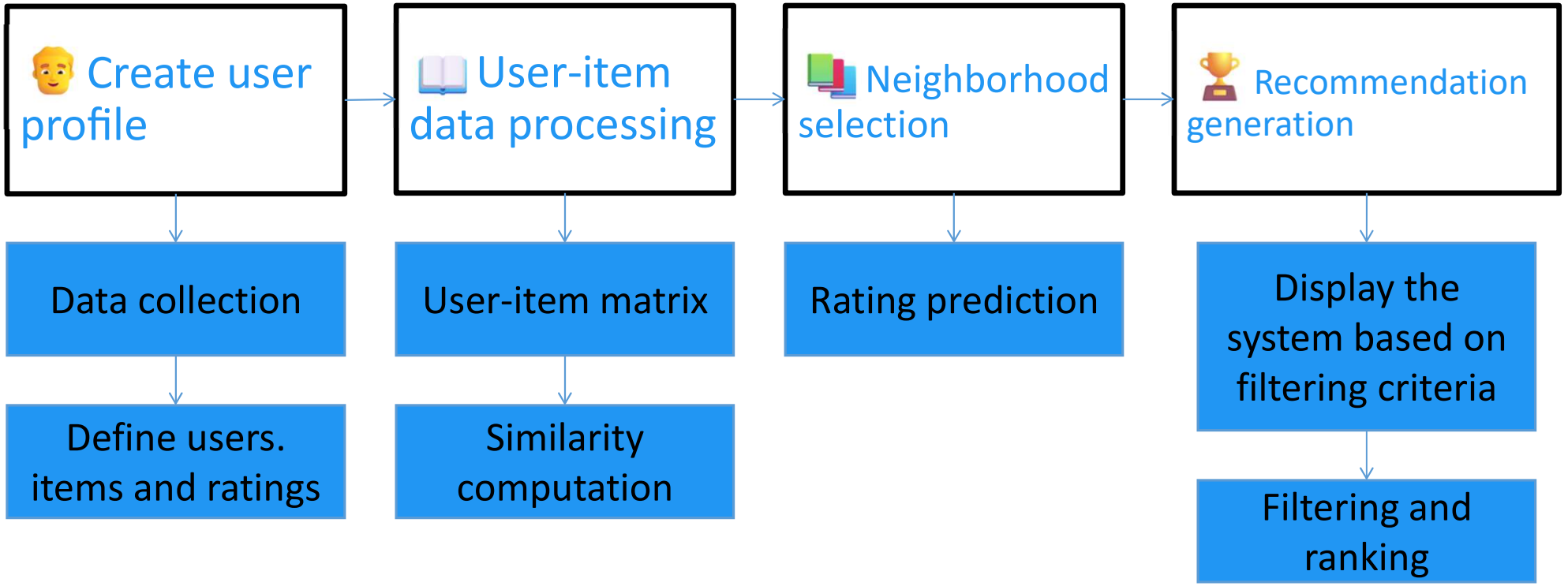
1. **Identify groups of users with similar preferences within the same cluster.**
2. **Users within the same cluster typically share common characteristics or preferences.**
3. **The system focuses on clusters. reducing the complexity from considering every user.**
4. **Recommendations remain relevant from changes.**
5. **PCA helps reduce dimensionality, highlighting key features that influence cluster formation, and improves model performance by simplifying the data while maintaining important patterns**

Collaborative-Filtering Recommender System using Supervised Learning

Models and Findings

KNN-Based Collaborative Filtering

Flowchart



Construct

Matrix

Collaborative filtering is probably the most commonly used recommendation algorithm. there are two main types of methods:

User-based collaborative filtering is based on the **user similarity or neighborhood**
Item-based collaborative filtering is based on **similarity among items**

User-item matrix

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

Predict the rating of the user user6 to item Machine Learning Capstone

Evaluation Results (display 15)

	User	Item	Predicted Rating	TITLE
0	1078030	ML0122ENv1	2.900	accelerating deep learning with gpu
1	1078030	DV0101EN	3.000	data visualization with python
2	733707	DS0101EN	3.000	introduction to data science
3	733707	ML0120EN	3.000	deep learning with tensorflow
4	733707	BD0101EN	3.000	big data 101
5	733707	BD0115EN	3.000	mapreduce and yarn
6	733707	ST0101EN	2.975	statistics 101
7	733707	DB0151EN	2.975	nosql and dbaas 101
8	733707	BD0212EN	3.000	spark fundamentals ii
9	733707	DV0151EN	3.000	data visualization with r
10	733707	ML0101EN	3.000	machine learning with python
11	733707	BD0135EN	3.000	developing distributed applications using zook...
12	674939	BD0141EN	3.000	accessing hadoop data using hive
13	674939	TMP0105EN	2.800	getting started with the data apache spark ma...
14	674939	BD0223EN	3.000	exploring spark s graphx
15	674939	BD0133EN	3.000	controlling hadoop jobs using oozie
16	674939	BD0115EN	3.000	mapreduce and yarn
17	674939	BD0145EN	3.000	sql access for hadoop

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

User profile 733707

Participate in 23 courses

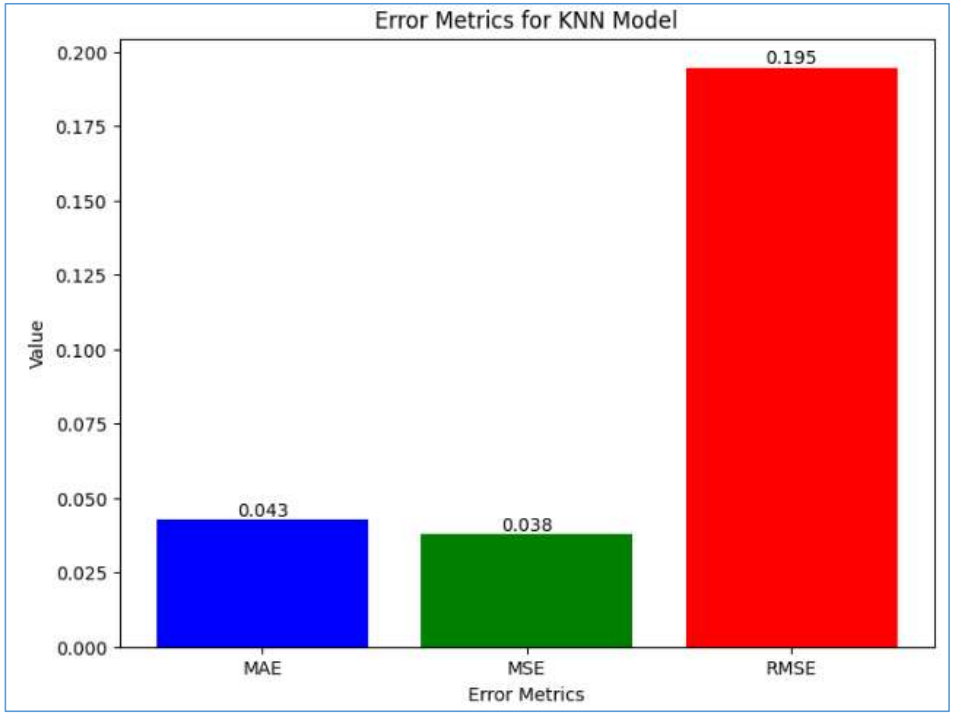
- Spark
- Sql
- Python

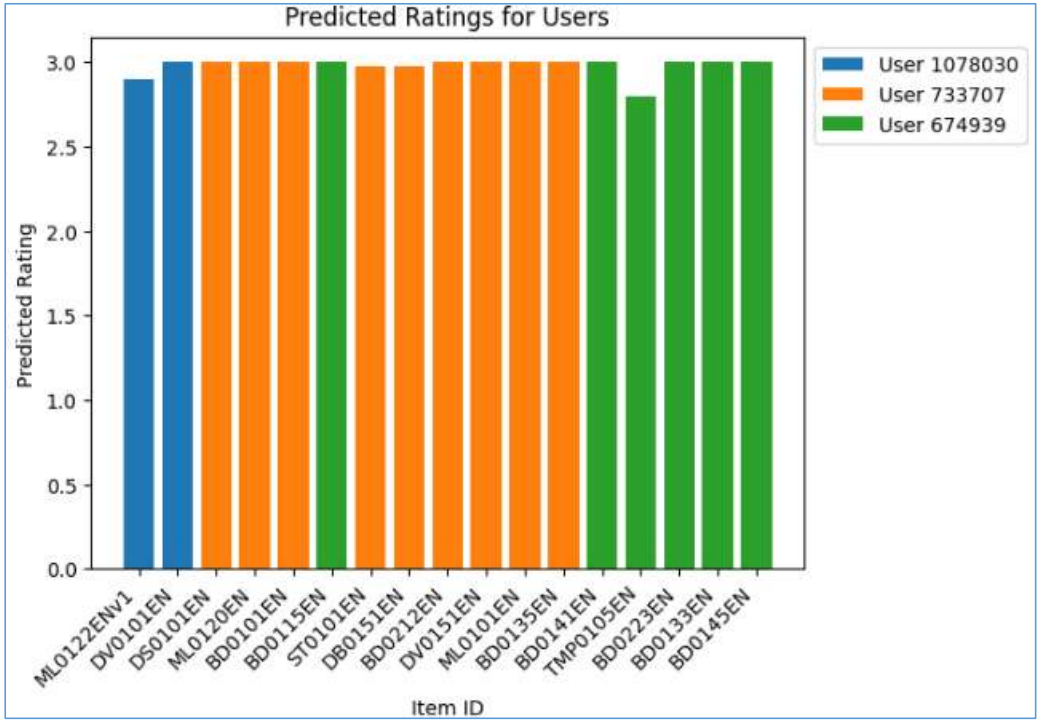
User profile 674939

Participate in 15 courses

- Spark
- Hadoop
- Big data

Evaluation Results





Construct

Recommendation based on neighborhood

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

- Python
- Data visualization

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

Recommended courses

- Data science
- Deep learning
- Big data
- Spark
- Data visualization
- Machine learning
- Statistics
- Nosql

User profile 674939

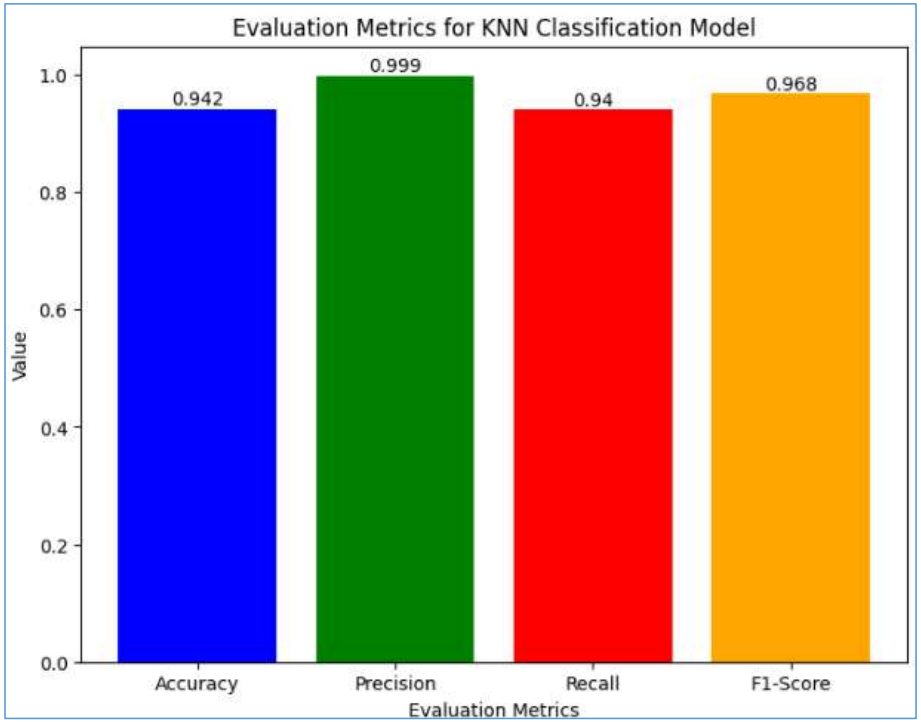
Participate in 15 courses

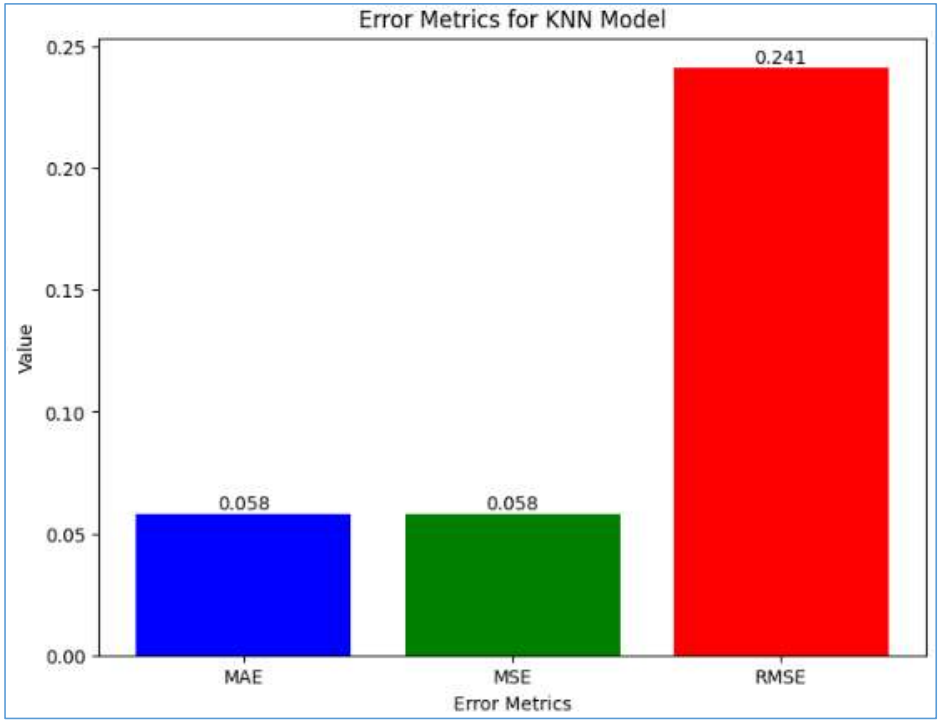
- Spark
- Hadoop
- Big data

- Spark
- Hadoop

Construct

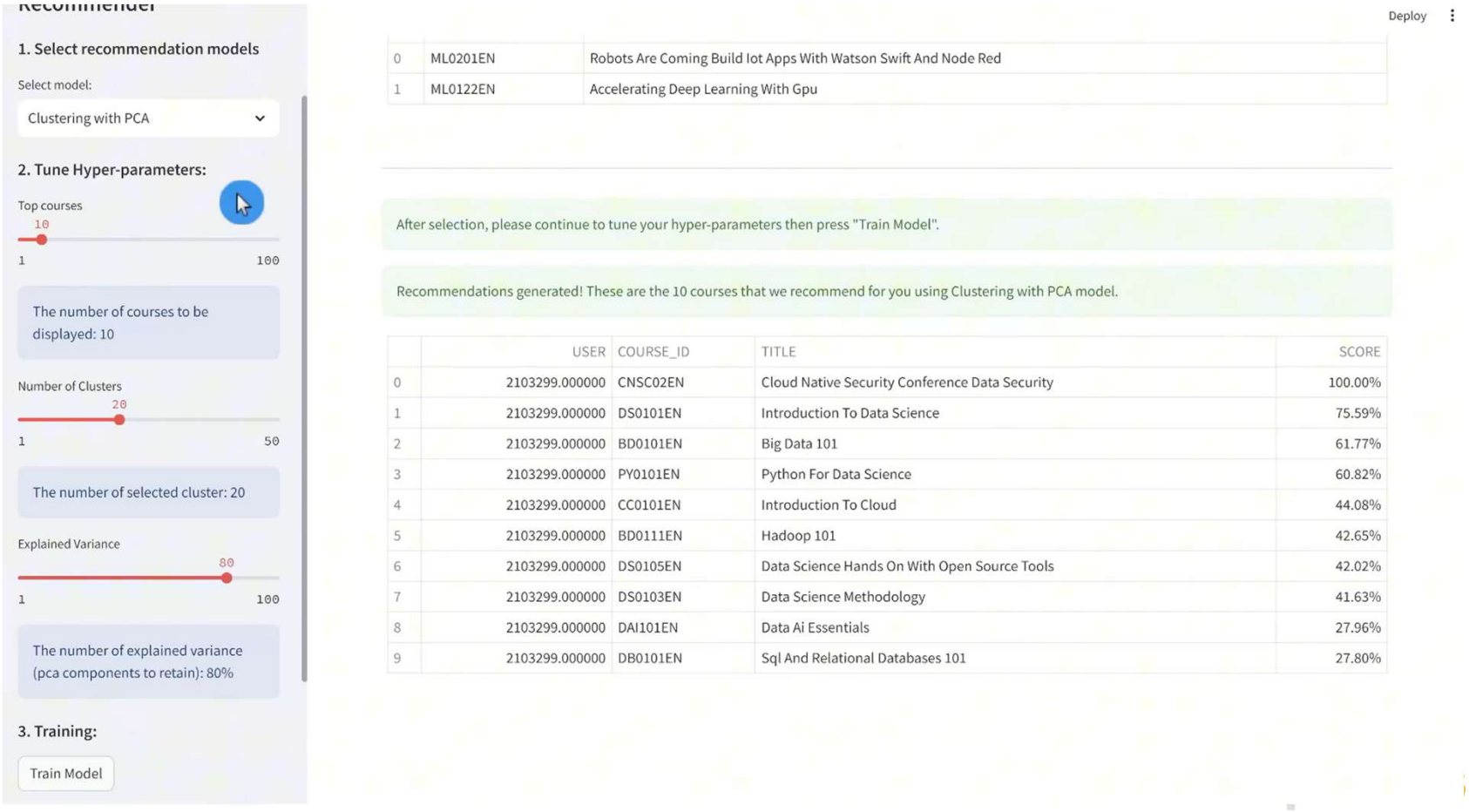
Evaluation Results (binary labels)





Construct

Recommendation based on KNN

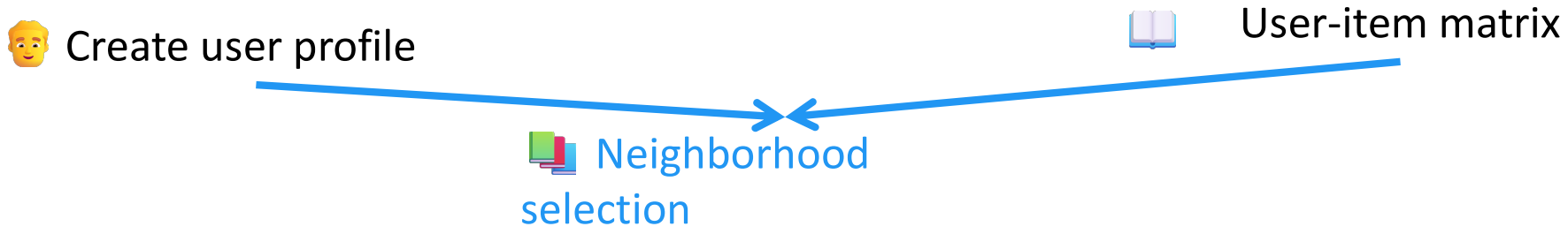


	USER	COURSE_ID	TITLE	SCORE
0	2103299.000000	CNSC02EN	Cloud Native Security Conference Data Security	100.00%
1	2103299.000000	DS0101EN	Introduction To Data Science	75.59%
2	2103299.000000	BD0101EN	Big Data 101	61.77%
3	2103299.000000	PY0101EN	Python For Data Science	60.82%
4	2103299.000000	CC0101EN	Introduction To Cloud	44.08%
5	2103299.000000	BD0111EN	Hadoop 101	42.65%
6	2103299.000000	DS0105EN	Data Science Hands On With Open Source Tools	42.02%
7	2103299.000000	DS0103EN	Data Science Methodology	41.63%
8	2103299.000000	DAI101EN	Data Ai Essentials	27.96%
9	2103299.000000	DB0101EN	Sql And Relational Databases 101	27.80%

truct

Summary

The idea

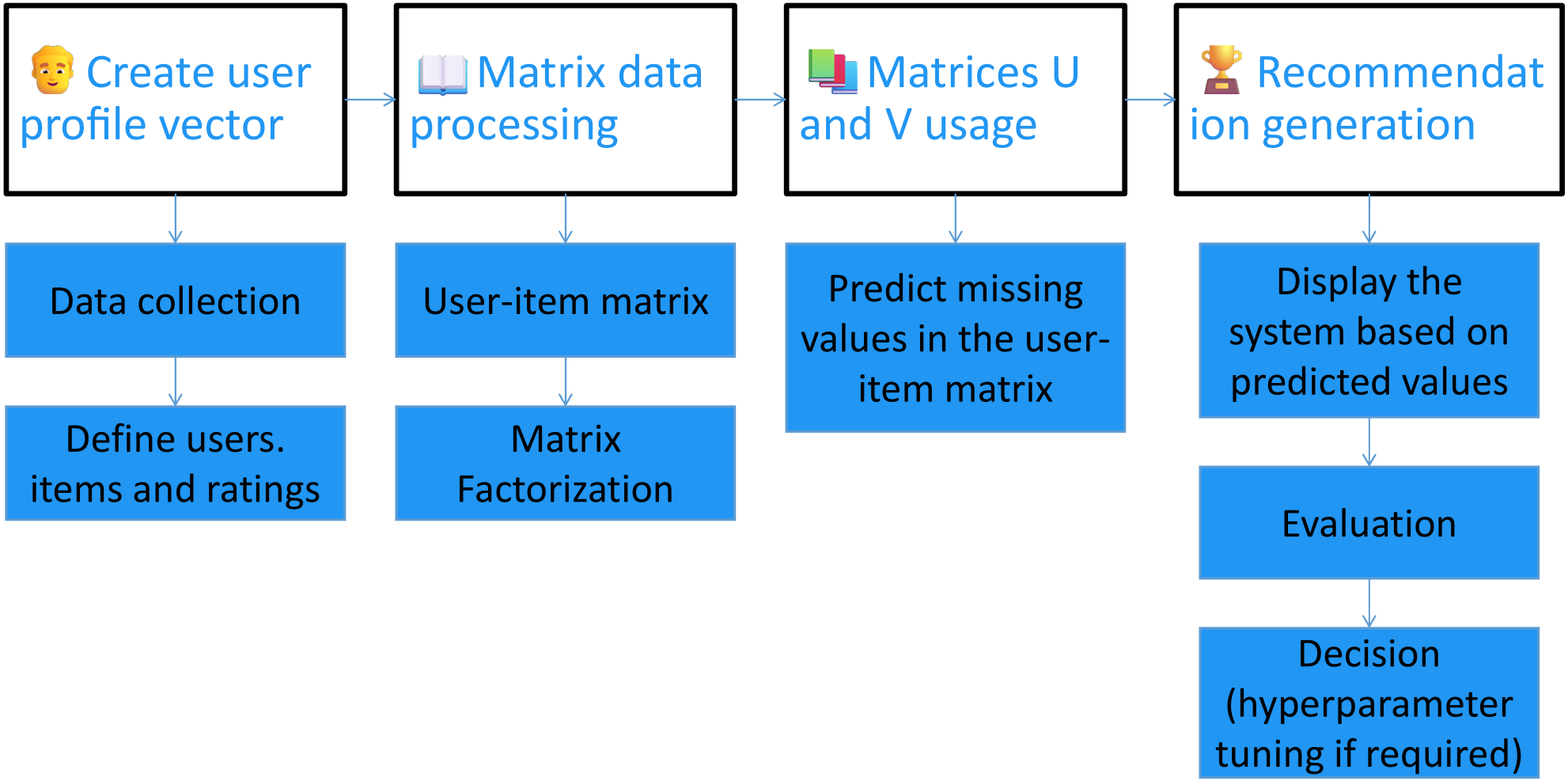


1. **Personalized recommendations by considering the preferences of similar users or items.**
2. **The similarity between users or items. using a straightforward nearest-neighbor approach.**
3. **New users or items based on the preferences of similar entities. Similar users or items are used to infer preferences for new entities.**
4. **Relies on the local neighborhood of users or items. and it can find meaningful connections.**

Models and Findings

NMF-Based Collaborative Filtering

Flowchart


Create user profile vector
Matrix data processing
Matrices U and V usage
Recommendation generation
Data collection
Define users. items and ratings
User-item matrix
Matrix Factorization
Predict missing values in the user-item matrix
Display the system based on predicted values
Evaluation
Decision (hyperparameter tuning if required)

Matrix

- **Non-negative matrix factorization (NMF**). decomposes a **big sparse matrix** into two **smaller** and **dense** matrices.
- **User features** and another represents the transformed **item features**.
- Non-negative matrix factorization can be one **solution to big matrix issues**.

Non-negative Matrix Factorization

User-item matrix: A 1000 x 100

	Item1	ML101	...	Item100
...	...	...	...	...
user2	3.0	3.0	3.0	3.0
user3	2.0	3.0	2.0	
user4	3.0	3.0	2.0	3.0
user5	2.0	3.0		
user6	3.0	3.0		3.0
...	...	...	...	...

≈

User matrix: U 1000 x 16

	Feature1	Feature2	...	Feature16
...	...	...	...	...
user2	....	....	....	....
user3	....	....	....	....
user4	....	....	....	....
user5	....	....	....	....
user6	....	....	....	....
...	...	...	...	...

X

Item matrix: I 16 x 100

	Item1	ML101	...	Item100
...	...	...	...	...
Feature1	....	....	....	....
Feature2	....	....	....	....
Feature3	....	....	....	....
....	....	....	....	....
Feature16	....	....	....	....
...	...	...	...	...

Evaluation Results

	User	Item	Predicted_Rating	TITLE
0	1078030	ML0122ENv1	2.798383	accelerating deep learning with gpu
1	1078030	DV0101EN	2.982449	data visualization with python
2	733707	DS0101EN	2.998862	introduction to data science
3	733707	ML0120EN	3.000000	deep learning with tensorflow
4	733707	BD0101EN	2.975825	big data 101
5	733707	BD0115EN	2.975243	mapreduce and yarn
6	733707	ST0101EN	2.963512	statistics 101
7	733707	DB0151EN	3.000000	nosql and dbaas 101
8	733707	BD0212EN	2.982911	spark fundamentals ii
9	733707	DV0151EN	2.896570	data visualization with r
10	733707	ML0101EN	2.898463	machine learning with python
11	733707	BD0135EN	2.979971	developing distributed applications using zook...
12	674939	BD0141EN	3.000000	accessing hadoop data using hive
13	674939	TMP0105EN	2.956095	getting started with the data apache spark ma...
14	674939	BD0223EN	2.965433	exploring spark s graphx
15	674939	BD0133EN	2.851486	controlling hadoop jobs using oozie
16	674939	BD0115EN	3.000000	mapreduce and yarn
17	674939	BD0145EN	3.000000	sql access for hadoop

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

User profile 733707

Participate in 23 courses

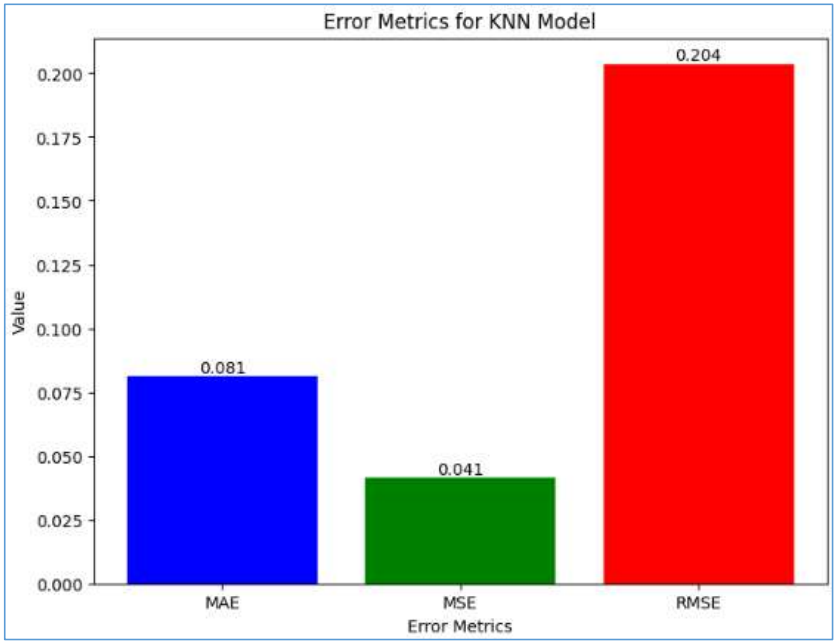
- Spark
- Sql
- Python

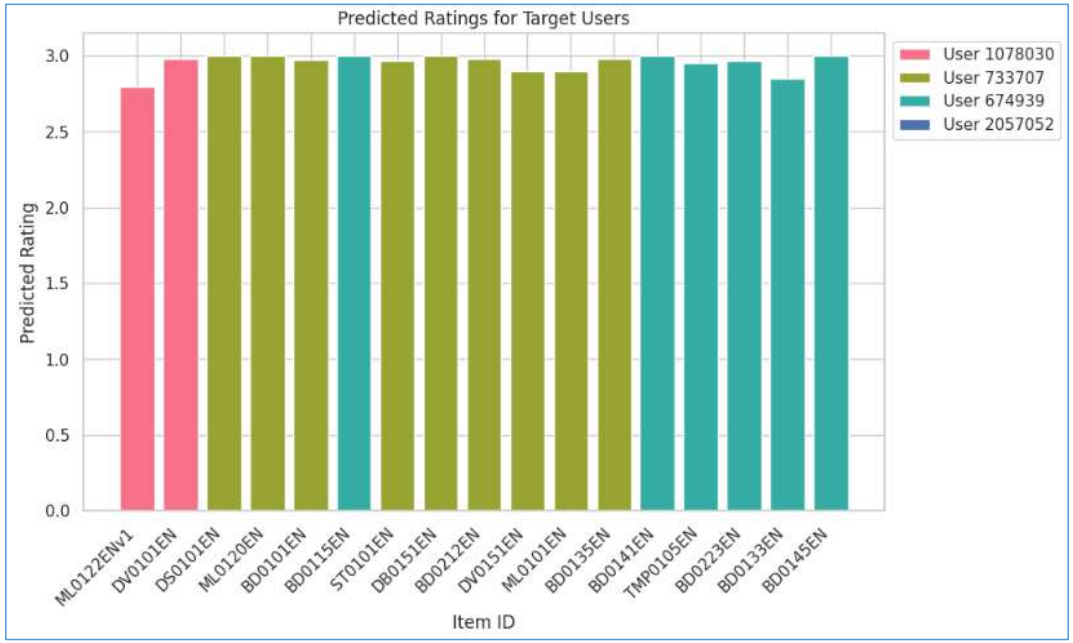
User profile 674939

Participate in 15 courses

- Spark
- Hadoop
- Big data

Evaluation Results





Construct

Recommendation based on NMF

User profile 1078030

Participate in 8 courses

- Data analysis
- Deep learning
- Python

Recommended courses

- Data visualization
- Deep learning

User profile 733707

Participate in 23 courses

- Spark
- Sql
- Python

Recommended courses

- Data science
- Deep learning
- Big data
- Spark
- Data visualization
- Machine learning
- Statistics
- Nosql

User profile 674939

Participate in 15 courses

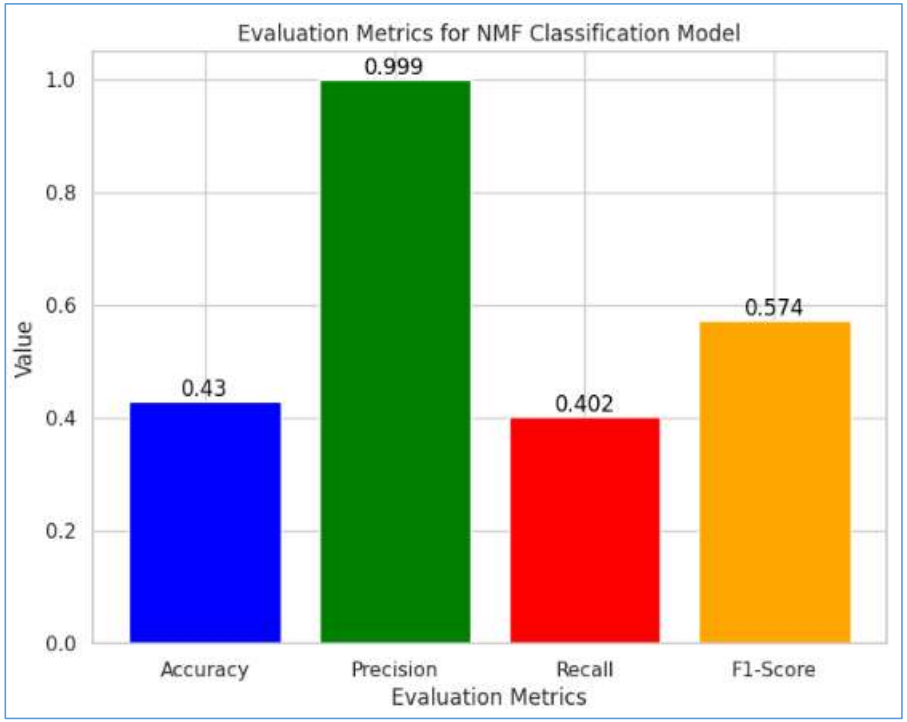
- Spark
- Hadoop
- Big data

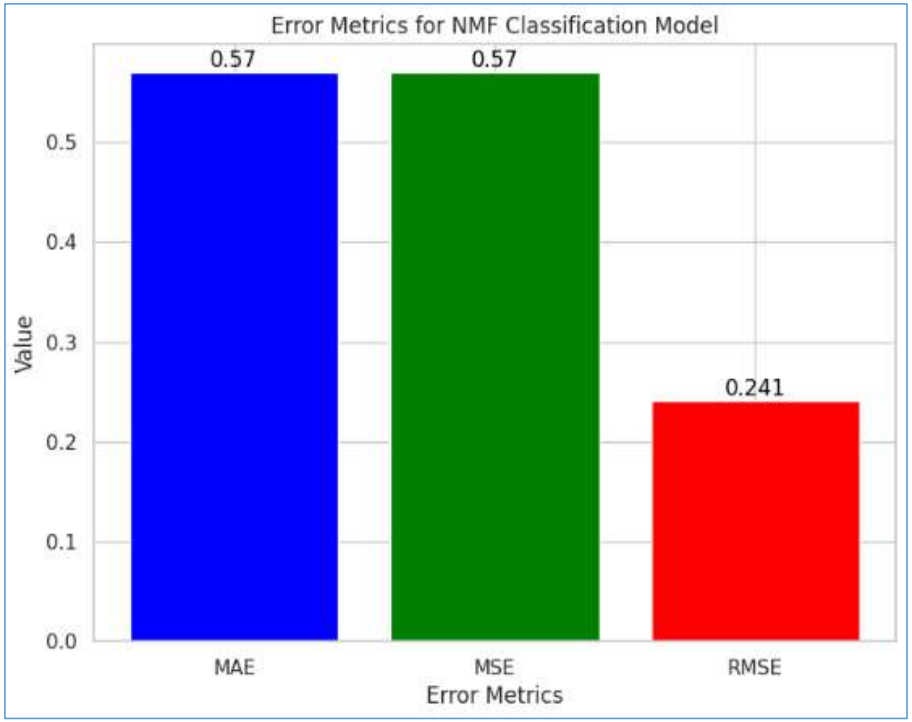
Recommended courses

- Spark
- Hadoop
- Apache
- Sql

Construct

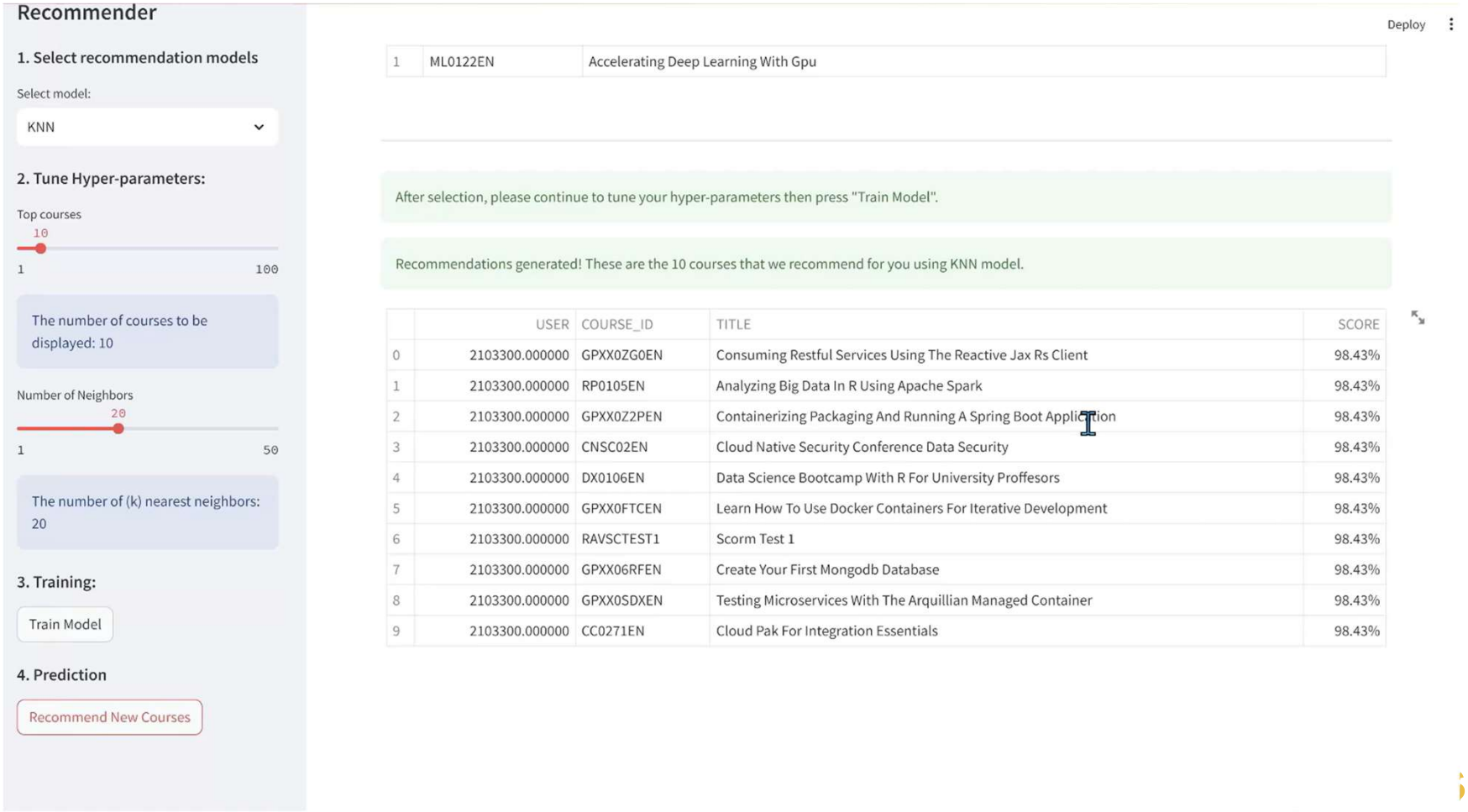
Evaluation Results (binary labels)





Construct

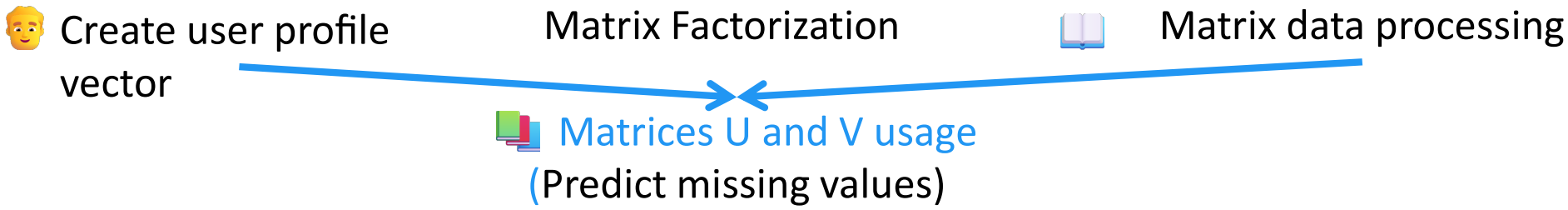
Recommendation based on NMF



truct

Summary

The idea

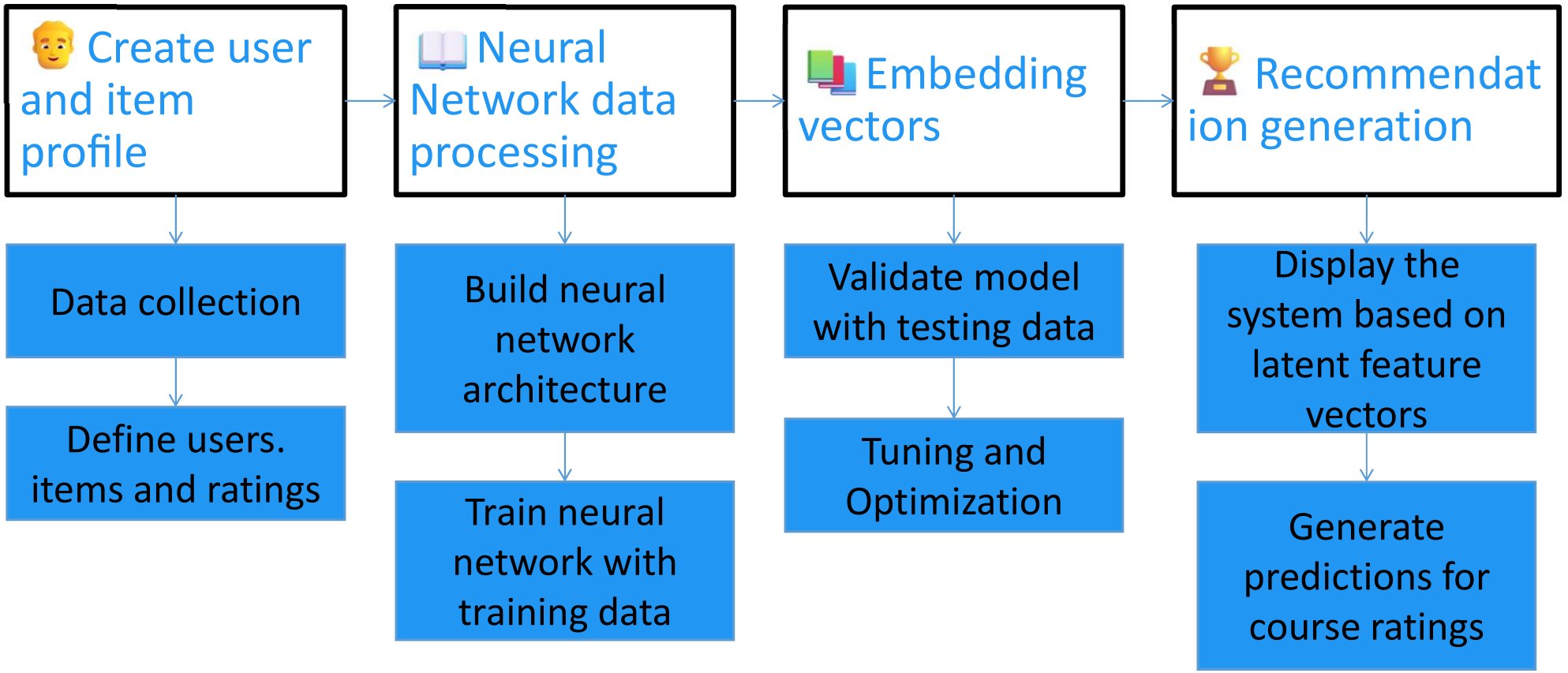


1. **Factorized matrices with non-negative values. providing an interpretable representation of users and items.**
2. **Relies on the underlying patterns and features present in the user-item interaction matrix.**
3. **New users or items based on the preferences of similar entities.Similar users or items are used to infer preferences for new entities.**
4. **Relies on the local neighborhood of users or items. and it can find meaningful connections.**

Models and Findings

Neural Network Embedding-Based Collaborative Filtering

Flowchart

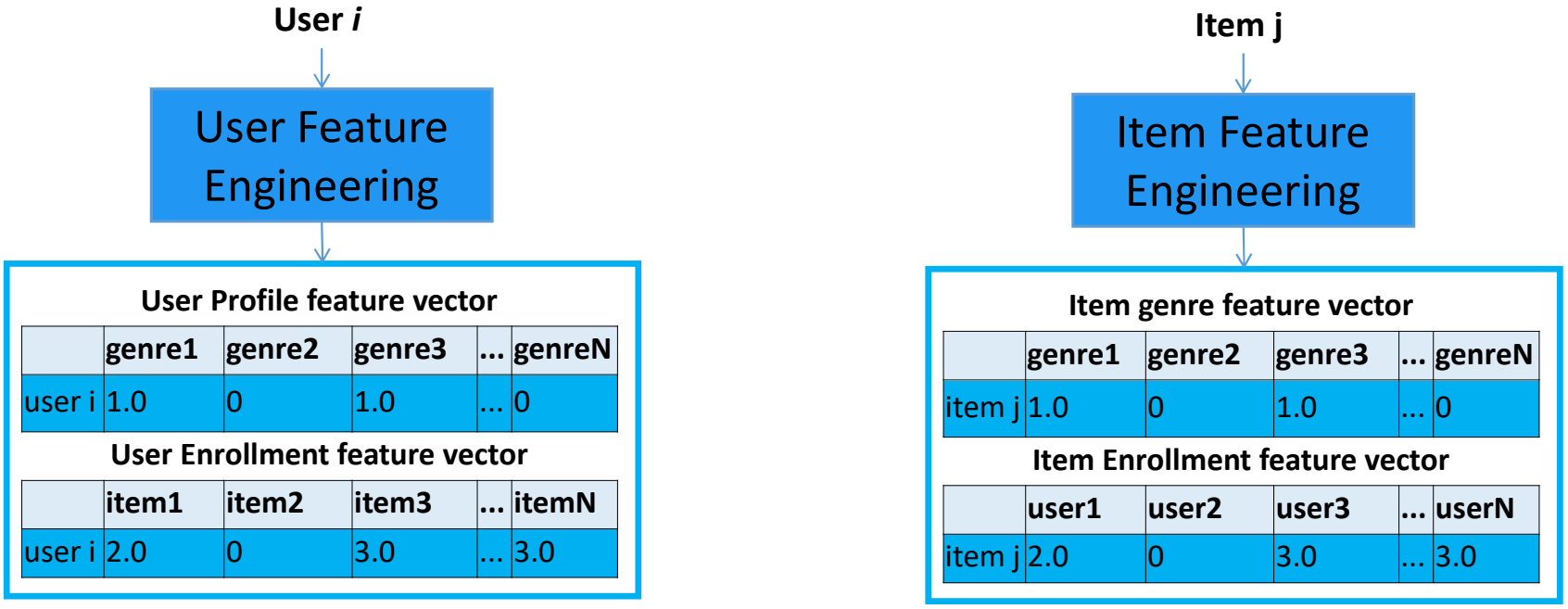


Construct

Matrix

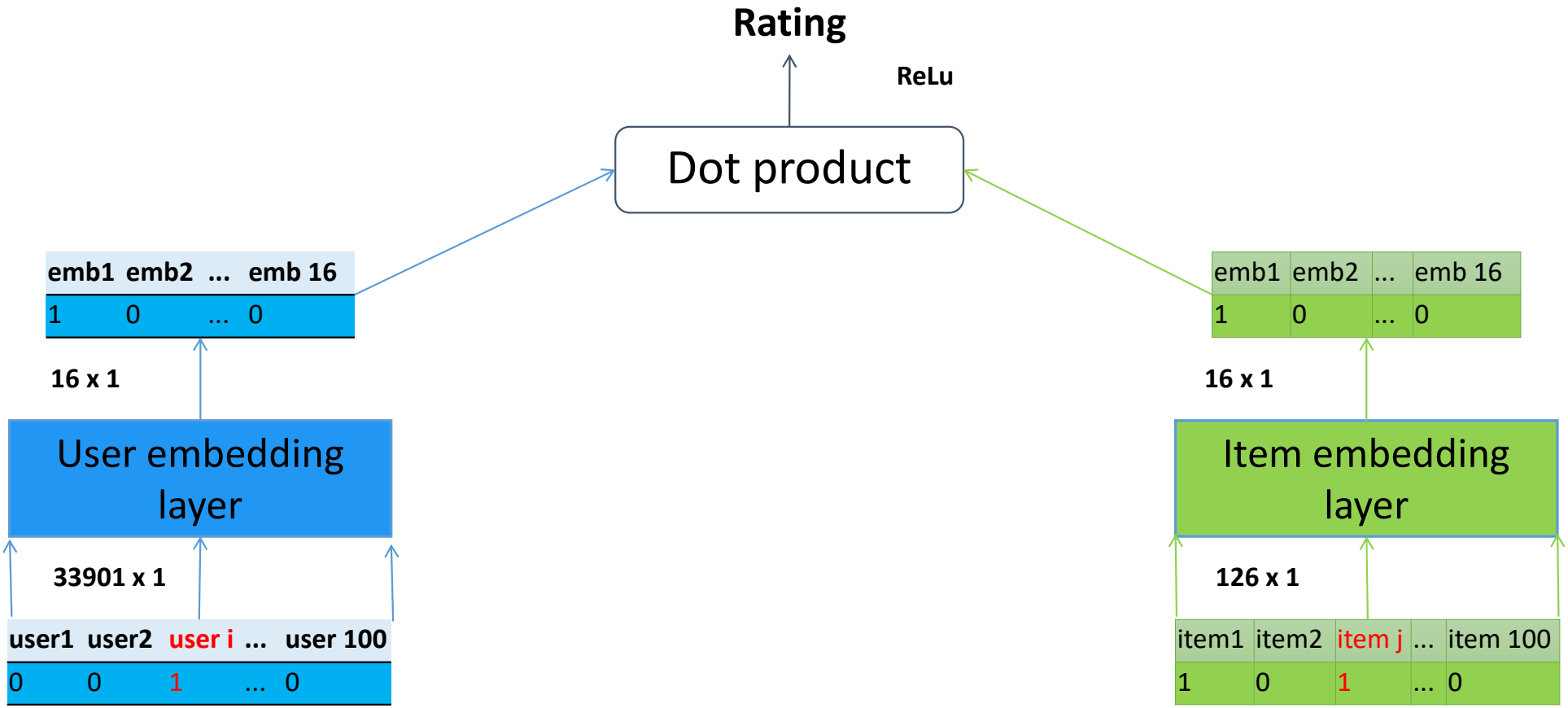
- Neural networks are very good at learning patterns from data and are widely used to extract latent features.
- Gradually captures and stores the features within its hidden layers as weight matrices and can be extracted to represent the original data.

Explicit User and Item Feature Engineering

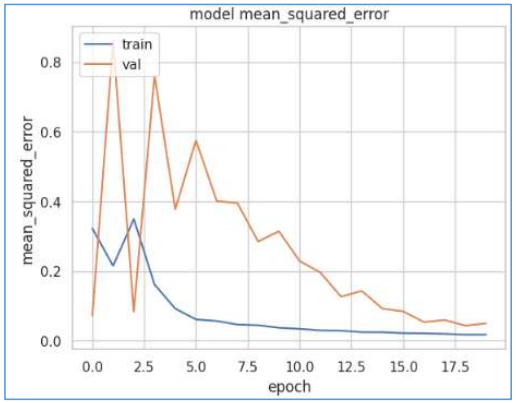


Latent feature vectors

Predict the **user-item interactions** while **simultaneously** extracting the **user** and **item embedding features**.

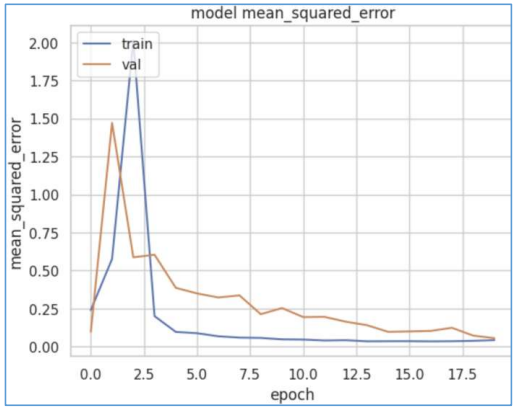


Evaluation Results (Improve performance)




```
Model: "recommender_net"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 user_embedding_layer (Embed  multiple                 542416
 ding)

 user_bias (Embedding)       multiple                  33901

 item_embedding_layer (Embed  multiple                 2016
 ding)

 item_bias (Embedding)       multiple                  126

=================================================================
Total params: 578,459
Trainable params: 578,459
Non-trainable params: 0
_________________________________________________________________
```

Before recommender_net




```
Model: "recommender_net_plus"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 user_embedding_layer (Embed  multiple                 1084832
 ding)

 user_bias (Embedding)       multiple                  33901

 item_embedding_layer (Embed  multiple                 4032
 ding)

 item_bias (Embedding)       multiple                  126

=================================================================
Total params: 1,122,891
Trainable params: 1,122,891
Non-trainable params: 0
_________________________________________________________________
```

After Improvement recommender_net_plus

Component	recommender_net	recommender_net_plus	Difference
User Embedding	56,416 params	1,641,022 params	29x larger
Item Embedding	2,016 params	4,052 params	2x larger
Total Params (Trainable Params)	~571K–591K	~1.12M–1.25M	~2x more complex
Validation Error Stability	Fluctuates	More Stable	
Generalization	Overfitting Risk	Improved Generalization	

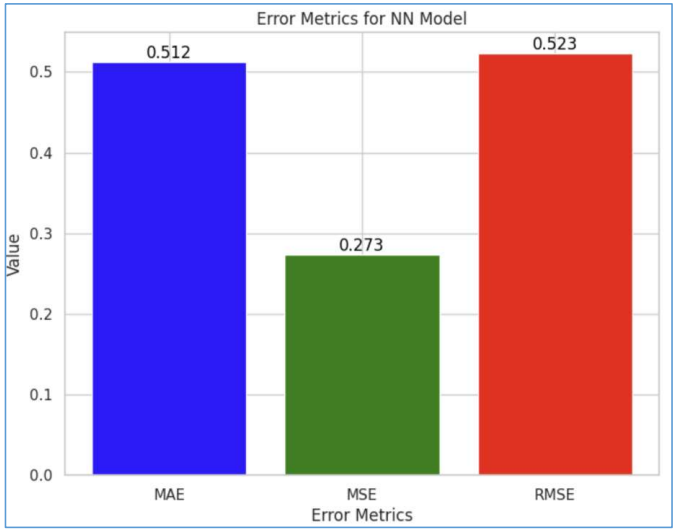
Evaluation Results (Improve performance)

Model	Initial MSE (Start)	Final MSE (End)	Epochs
recommender_net	~0.6	~0.0	17.5
recommender_net_plus	~1.75	~0.25	17.5

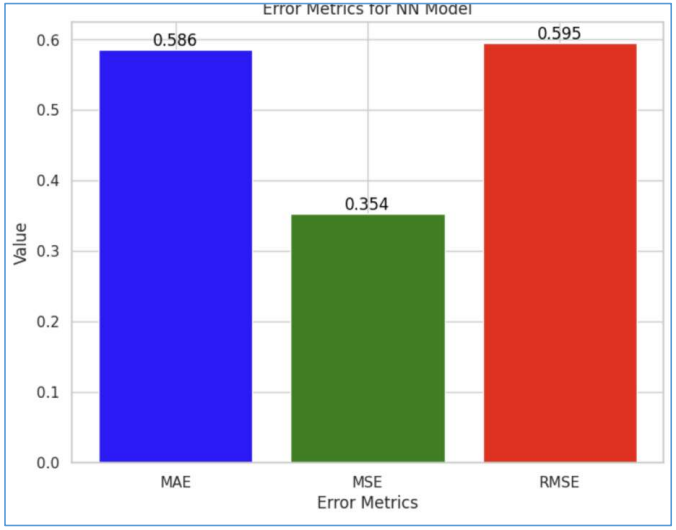
Key takeaways :

- **recommender_net**:
  - Achieves near-zero MSE quickly (by epoch 5), suggesting **overfitting**. The model memorizes training data but fails to generalize (validation MSE likely high, though not shown).
  - Example: Predicts training users' ratings perfectly but fails for new users.
- **recommender_net_plus**:
  - Starts with higher MSE but shows **steady improvement**, indicating better optimization and generalization.
  - Example: Predicts ratings for both training and new users more reliably, though not perfectly
- **What we did**:
  - Increased Embedding Dimensions.
  - Optimized Learning Rate and Training Strategy
  - Added Regularization & Dropout
  - More Training Data or Longer Training Time

Evaluation Results (3rd)



Before recommender_net



After Improvement recommender_net_plus

Key takeaways :

The improved model:

- **MAE (Mean Absolute Error):**
  - After the improvements, the MAE increased slightly to **0.586**.
- **MSE (Mean Squared Error):**
  - MSE increased to **0.354**, suggesting that while the error is slightly higher, the model is becoming more robust to outliers due to better handling of complex relationships between users and items.

Key takeaways :

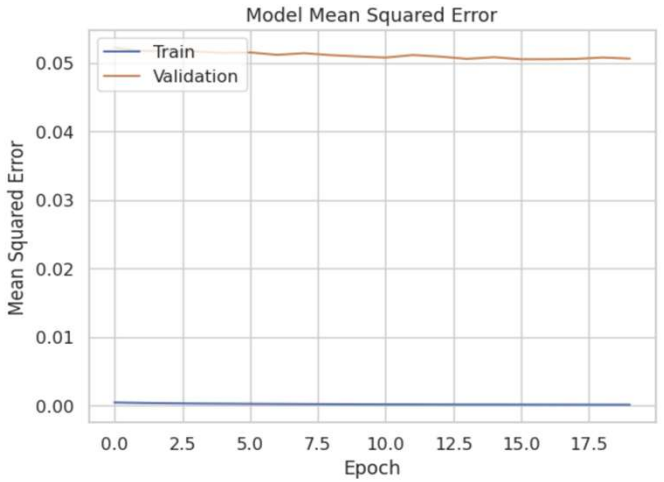
The improved model:

- **RMSE (Root Mean Squared Error):**
  - RMSE shows a slight increase to **0.595**, reflecting that while the error is somewhat higher, the model can handle unseen data better.

2nd Improvement

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 user_id (InputLayer)        [(None,)]                 0         []

 item_id (InputLayer)        [(None,)]                 0         []

 mlp_user_embedding (Embedding)  (None, 32)            1084832   ['user_id[0][0]']

 mlp_item_embedding (Embedding)  (None, 32)            4064      ['item_id[0][0]']

 flatten_2 (Flatten)         (None, 32)                0         ['mlp_user_embedding[0][0]']

 flatten_3 (Flatten)         (None, 32)                0         ['mlp_item_embedding[0][0]']

 concatenate (Concatenate)   (None, 64)                0         ['flatten_2[0][0]',
                                                                  'flatten_3[0][0]']

 mf_user_embedding (Embedding)  (None, 32)             1084832   ['user_id[0][0]']

 mf_item_embedding (Embedding)  (None, 32)             4064      ['item_id[0][0]']

 layer_0 (Dense)             (None, 16)                1040      ['concatenate[0][0]']

 flatten (Flatten)           (None, 32)                0         ['mf_user_embedding[0][0]']
```

```
Model: "model"
__________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
==================================================================================================
 flatten_1 (Flatten)         (None, 32)                0         ['mf_item_embedding[0][0]']

 layer_1 (Dense)             (None, 8)                 136       ['layer_0[0][0]']

 multiply (Multiply)         (None, 32)                0         ['flatten[0][0]',
                                                                  'flatten_1[0][0]']

 layer_2 (Dense)             (None, 4)                 36        ['layer_1[0][0]']

 concatenate_1 (Concatenate)  (None, 36)               0         ['multiply[0][0]',
                                                                  'layer_2[0][0]']

 interaction (Dense)         (None, 1)                 37        ['concatenate_1[0][0]']

==================================================================================================
Total params: 2,179,041
Trainable params: 2,179,041
Non-trainable params: 0
__________________________________________________________________________________________________
```



Feature	Recommender Net Plus	Model (from the image)
Architecture Complexity	Simple	Complex (Dual Pathway)
Training Speed	Faster	Slower
Number of Parameters	Lower (Fewer layers)	Higher (Over 2 million)
Computational Cost	Low	High
Interaction Modeling	Linear (Concatenation)	Nonlinear (Concatenation & Multiplication)
Risk of Overfitting	Lower	Higher
Generalization	Good	Dependent on regularization
Recommendation Accuracy	Moderate	Potentially higher
Best for	Small/Moderate datasets	Large/Complex datasets

Summary

The idea

1. **Capture complex. non-linear relationships in course-rating data. allowing for more accurate modeling of user preferences.**
2. **The learned embeddings can capture latent factors and provide insights into the underlying features influencing course ratings.**
3. **Handling implicit feedback. such as user interactions and engagement. which may not be explicitly rated.**

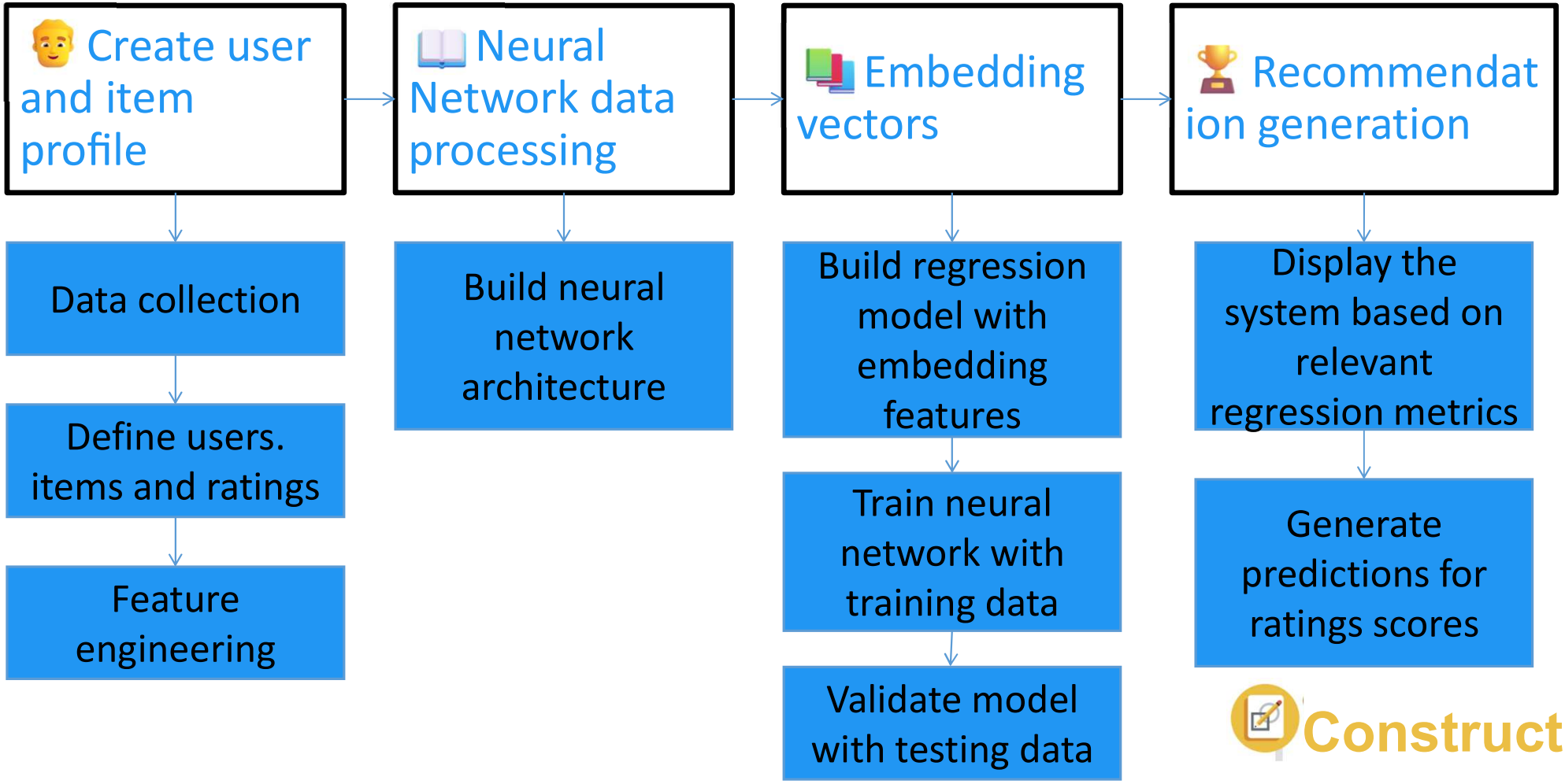
Models and Findings

Collaborative Filtering Algorithms Evaluation

Models and Findings

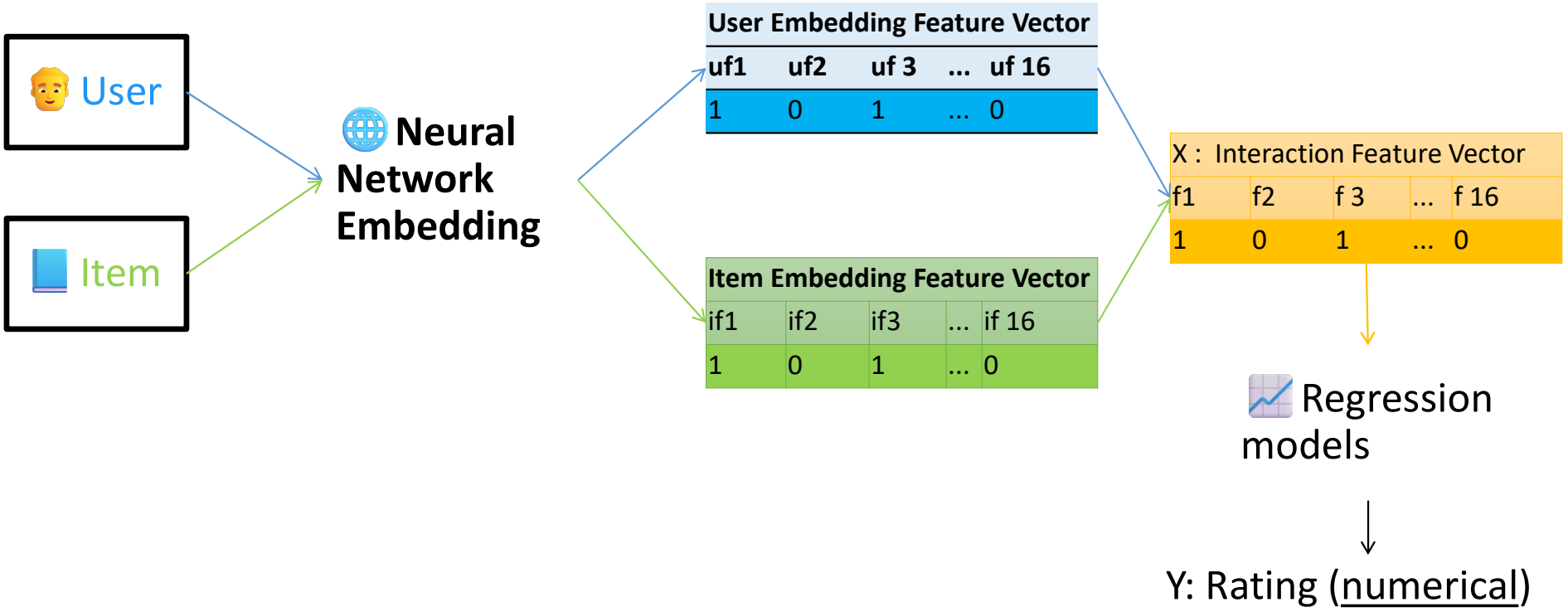
Regression-based Rating Score Prediction using Embedding Features

Flowchart Regression Based



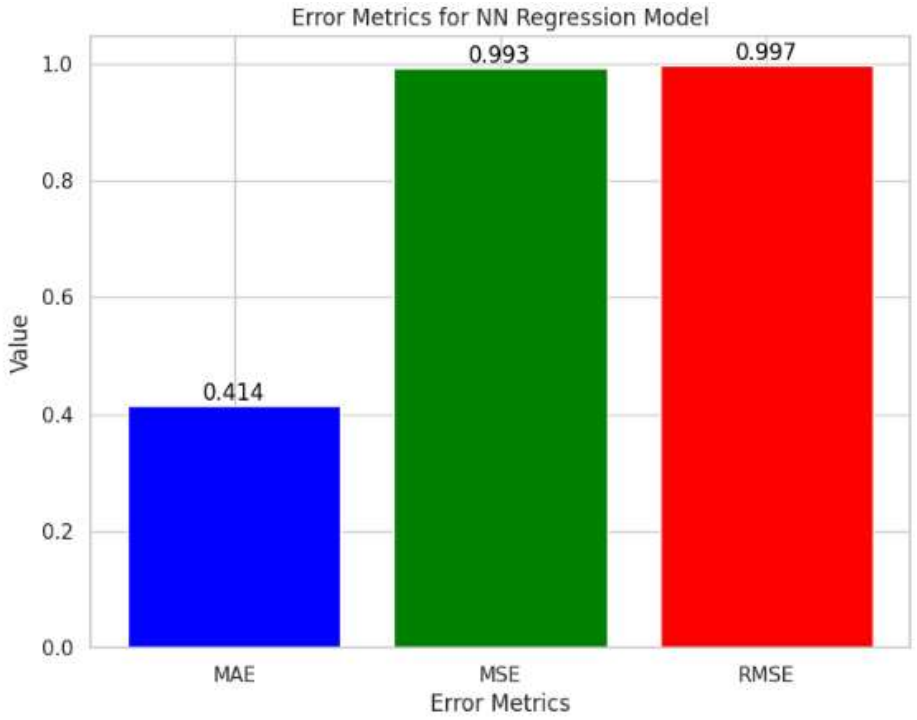
Neural Networks using Embedding Features

In the neural network. **extends** this by using **two embedding vectors as an input** into a **Neural Network** to **predict the rating**.



Evaluation Results (3rd)

```
MAE: 0.41428838083033687
MSE: 0.9932500760760065
RMSE: 0.9966193235513781
```



Construct

Summary

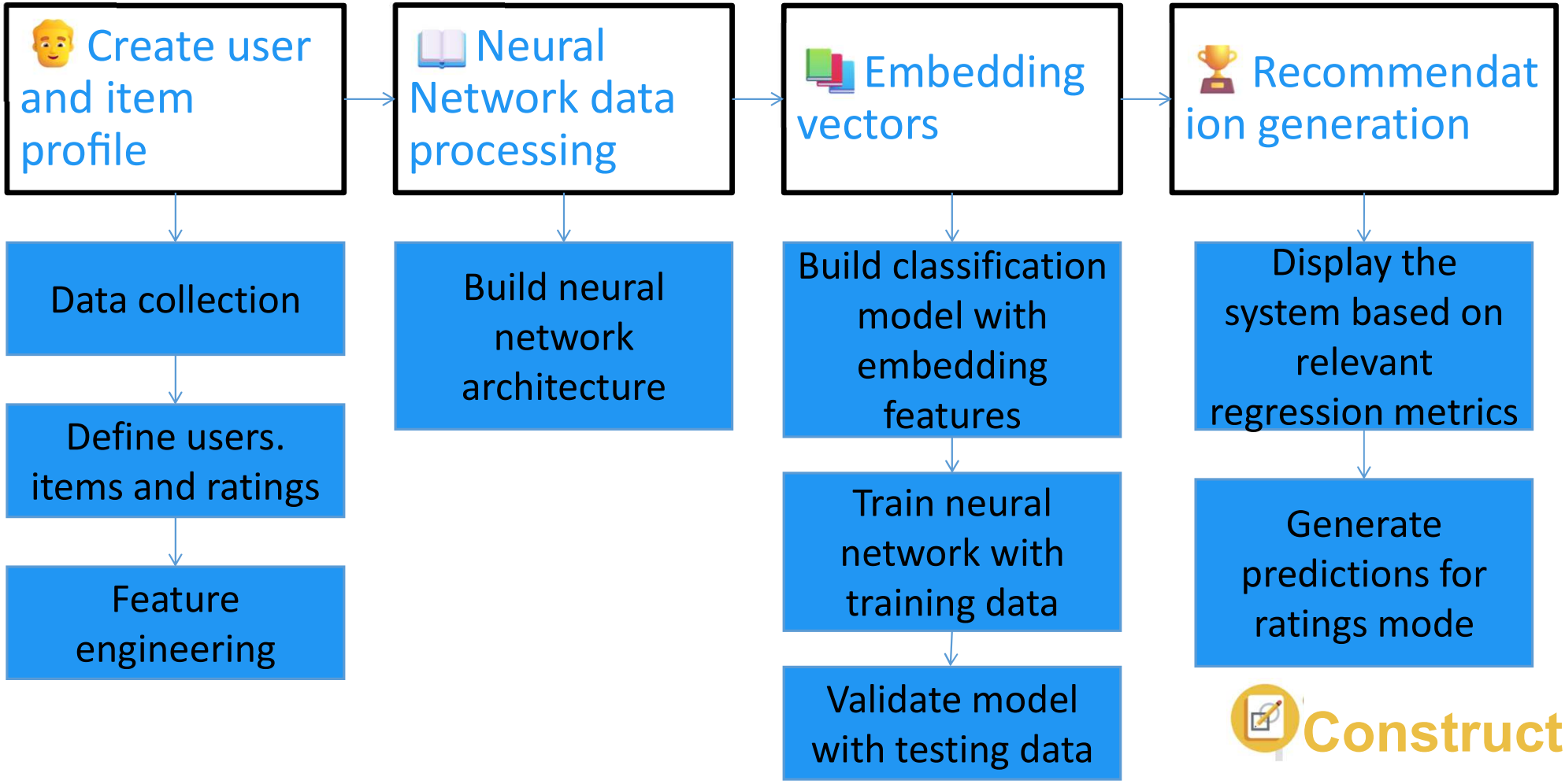
The idea

1. **Capture latent factors and relationships that contribute to the prediction of rating scores.**
2. **The embedding features implicitly learn latent factors without the need for explicit feature engineering. A more accurate representations of complex relationships in the data.**
3. **The learned embeddings enable the model to understand underlying patterns that contribute to rating scores.**
4. **Provide a dense representation that captures similarities between courses and users.**
5. **Users and courses with similar embeddings share common features. aiding interpretability.**

Models and Findings

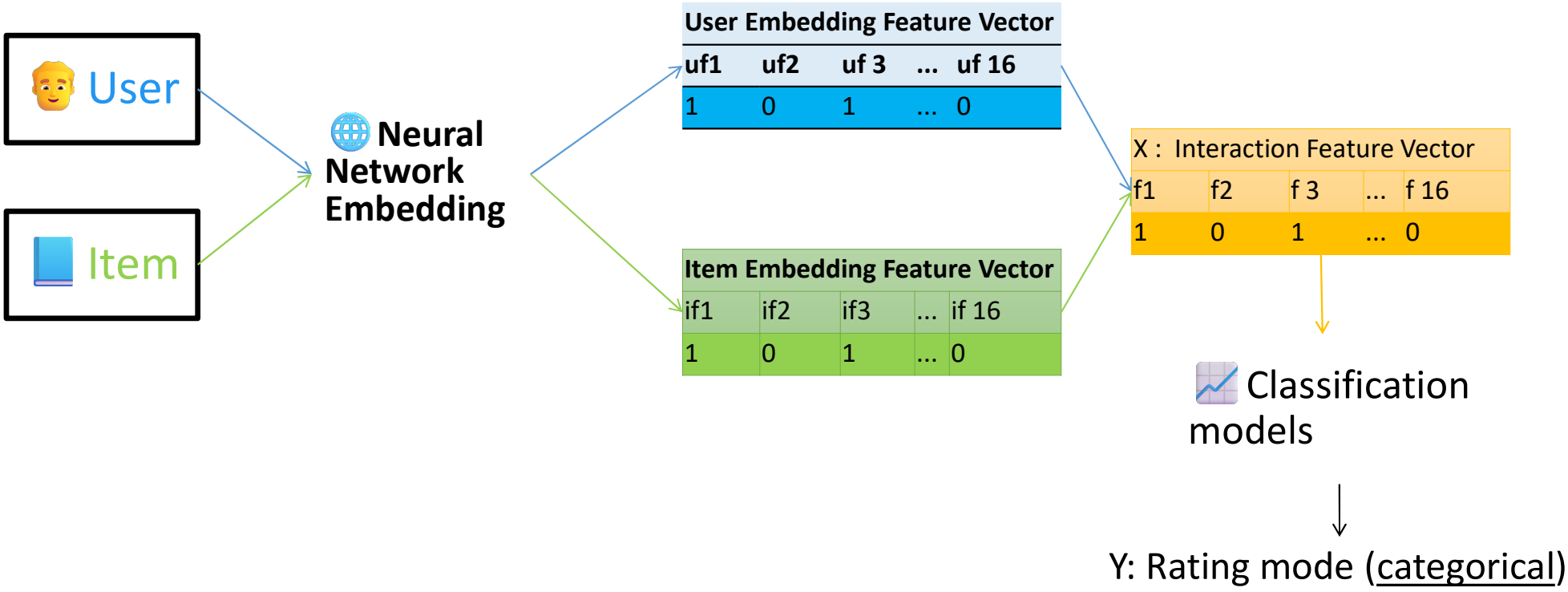
Classification-based Rating Score Prediction using Embedding Features

Flowchart Classification Based

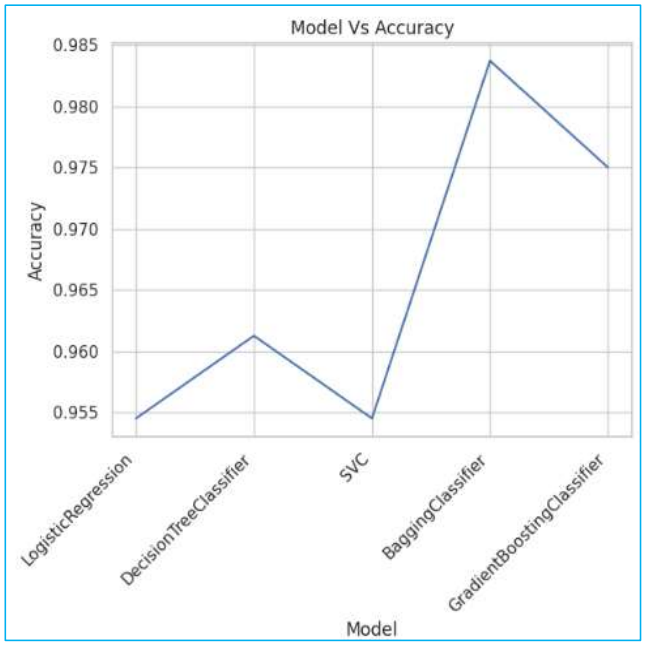


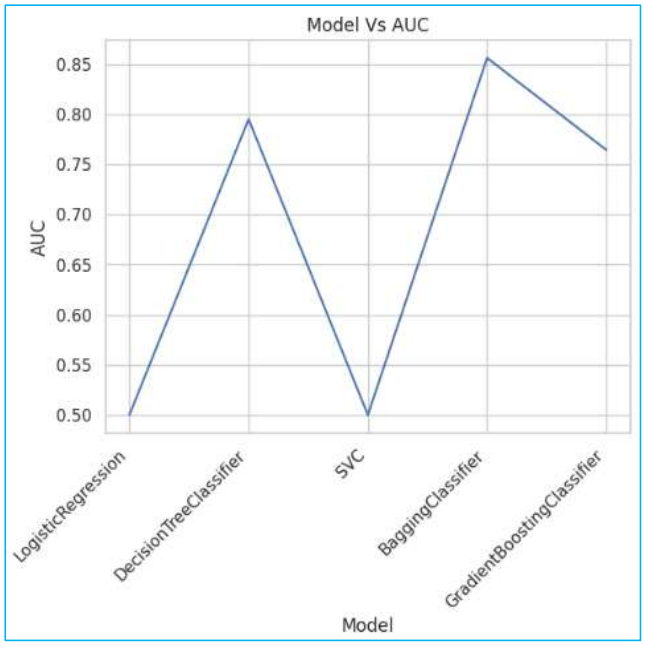
Neural Networks using Embedding Features

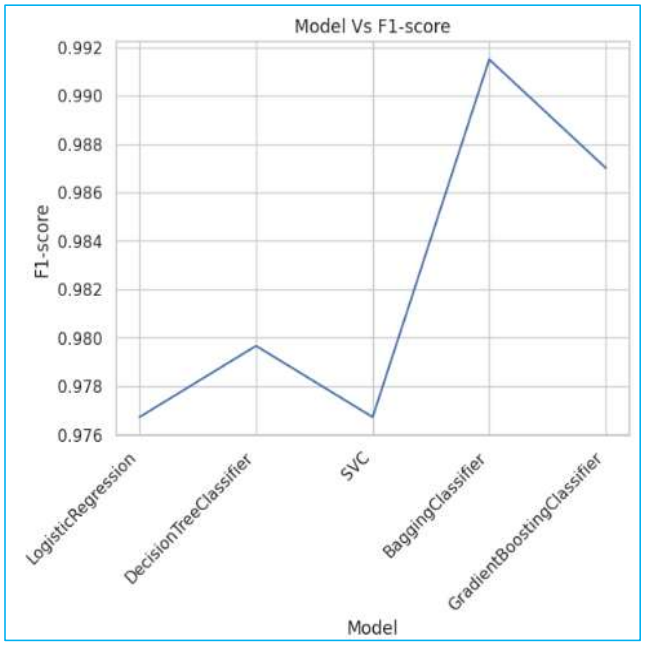
The prediction problem as **a classification problem** as rating only has **two categorical values (Aduit vs. Completion)**



Evaluation Results





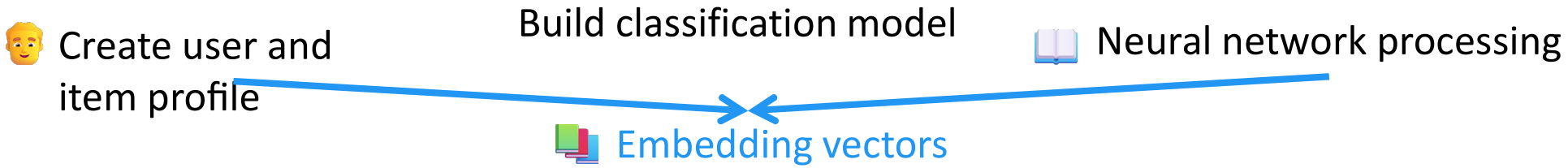


	Model	Accuracy	Precision	Recall	F1-Score	AUC
0	LogisticRegression	0.954503	0.954503	1.000000	0.976722	0.500000
1	DecisionTreeClassifier	0.961253	0.981454	0.977885	0.979666	0.795113
2	SVC	0.954503	0.954503	1.000000	0.976722	0.500000
3	BaggingClassifier	0.983713	0.986596	0.996475	0.991511	0.856221
4	GradientBoostingClassifier	0.974990	0.978129	0.996071	0.987018	0.764404

Construct

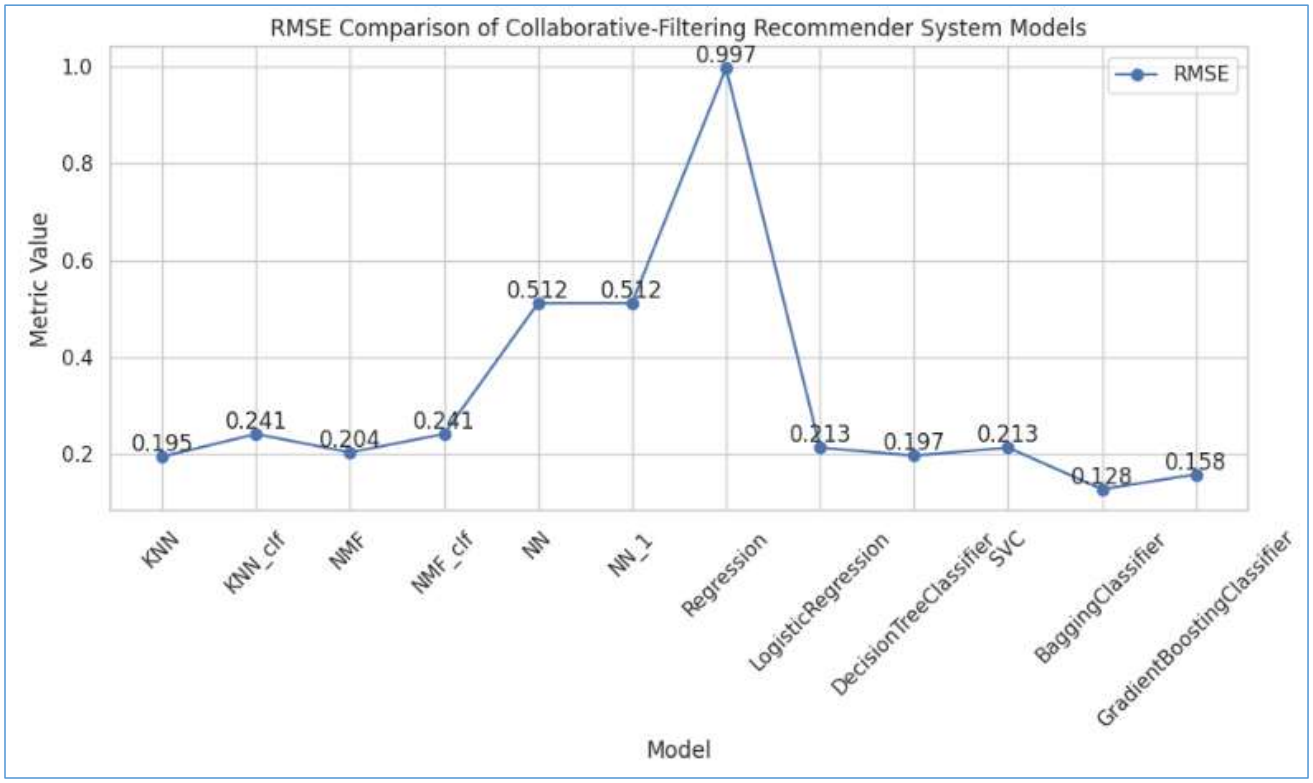
Summary

The idea



1. **Provide insights into which features contribute to a specific rating class.**
2. **Easier to understand the distinctions between various user preferences.**
3. **Classification is well-suited for scenarios where the ratings are discrete and categorical. such as a system where users provide ratings on a scale (e.g.. 1 to 5 stars).**

Summary



	Model	RMSE
0	KNN	0.194618
1	KNN_clf	0.241409
2	NMF	0.203553
3	NMF_clf	0.241409
4	NN	0.511761
5	NN_1	0.511761
6	Regression	0.996619
7	LogisticRegression	0.213301
8	DecisionTreeClassifier	0.196842
9	SVC	0.213301
10	BaggingClassifier	0.127622
11	GradientBoostingClassifier	0.158144

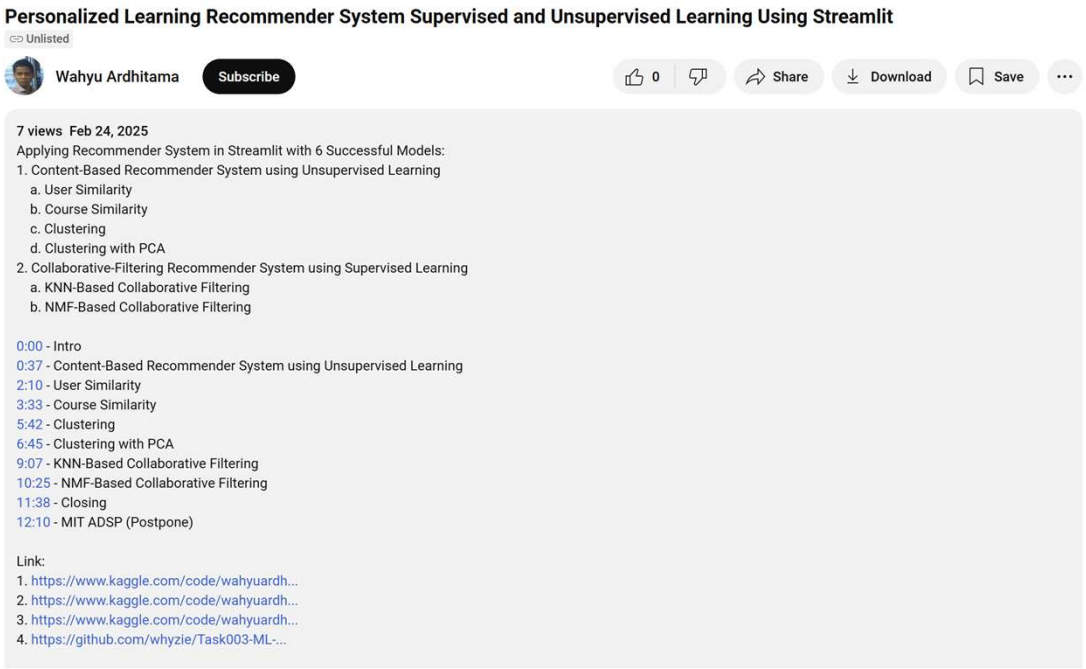
Construct

Streamlit

Personal Recommended System

Recommender System in Streamlit

https://youtu.be/7zvritN8VO0



Personalized Learning Recommender System Supervised and Unsupervised Learning Using Streamlit

Unlisted

Wahyu Ardhitama Subscribe

0 Share Download Save ...

7 views Feb 24, 2025

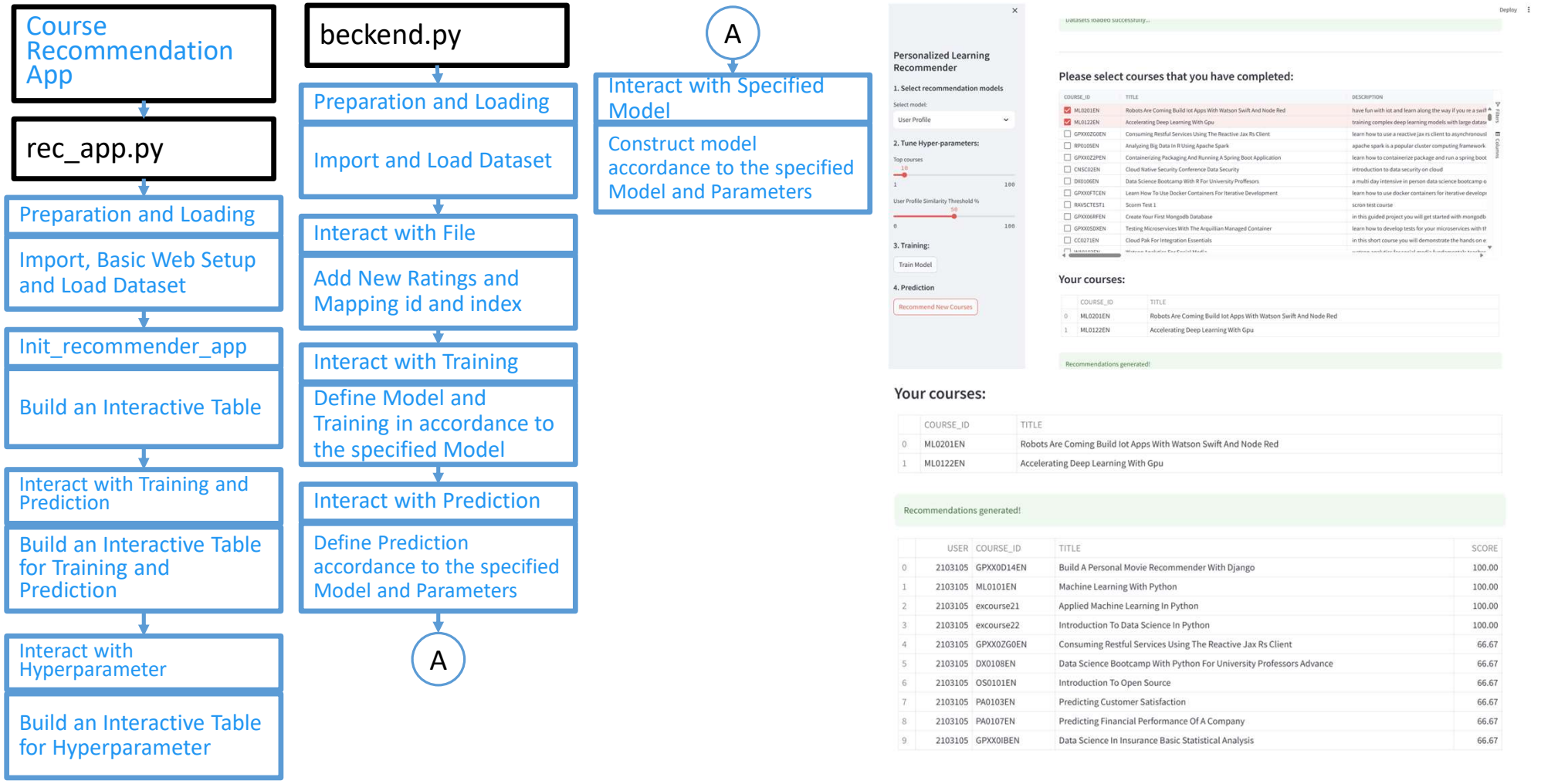
Applying Recommender System in Streamlit with 6 Successful Models:

1. Content-Based Recommender System using Unsupervised Learning
   a. User Similarity
   b. Course Similarity
   c. Clustering
   d. Clustering with PCA
2. Collaborative-Filtering Recommender System using Supervised Learning
   a. KNN-Based Collaborative Filtering
   b. NMF-Based Collaborative Filtering

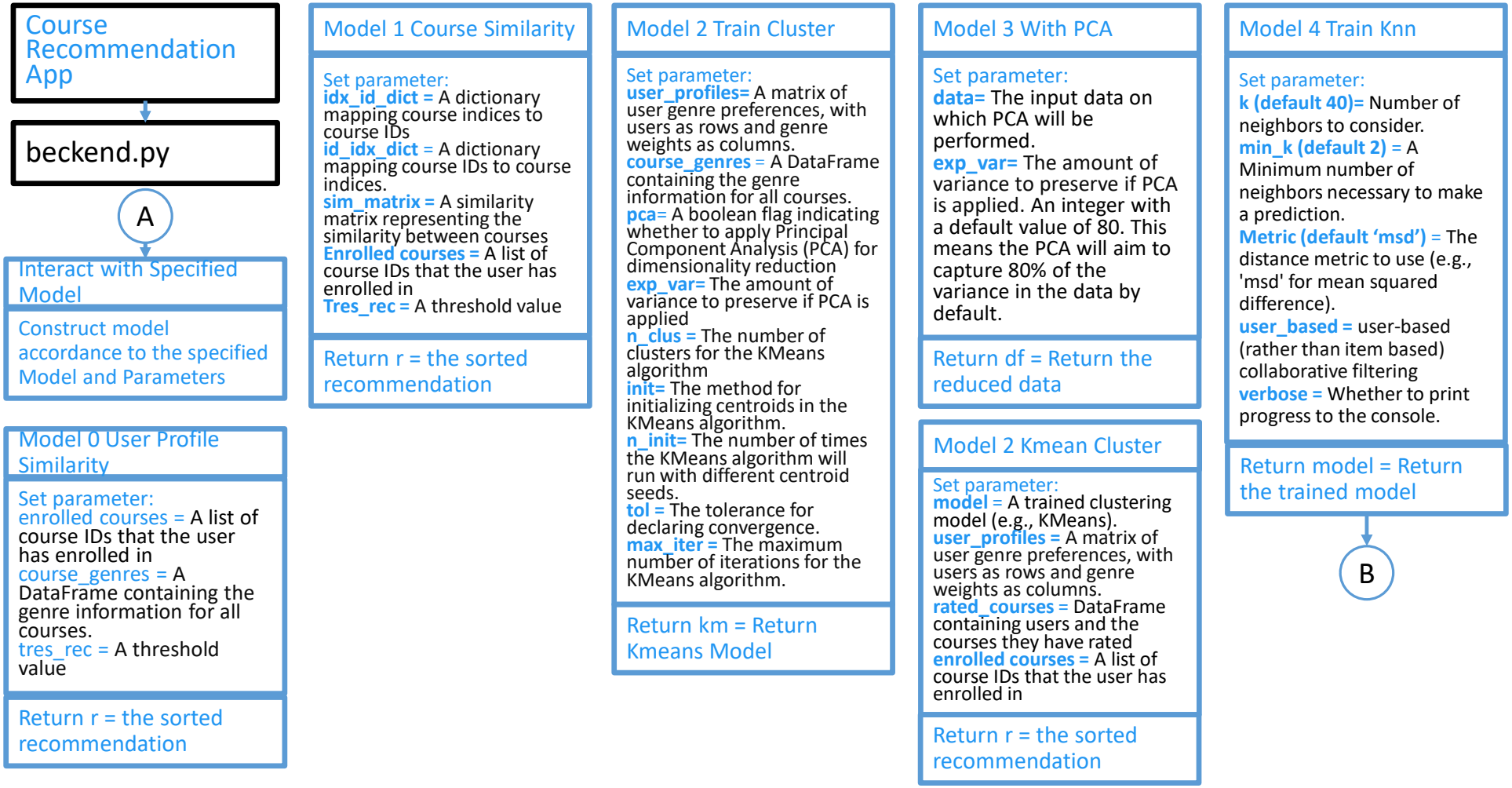
0:00 - Intro
0:37 - Content-Based Recommender System using Unsupervised Learning
2:10 - User Similarity
3:33 - Course Similarity
5:42 - Clustering
6:45 - Clustering with PCA
9:07 - KNN-Based Collaborative Filtering
10:25 - NMF-Based Collaborative Filtering
11:38 - Closing
12:10 - MIT ADSP (Postpone)

Link:
1. https://www.kaggle.com/code/wahyuardh...
2. https://www.kaggle.com/code/wahyuardh...
3. https://www.kaggle.com/code/wahyuardh...
4. https://github.com/whyzie/Task003-ML-...

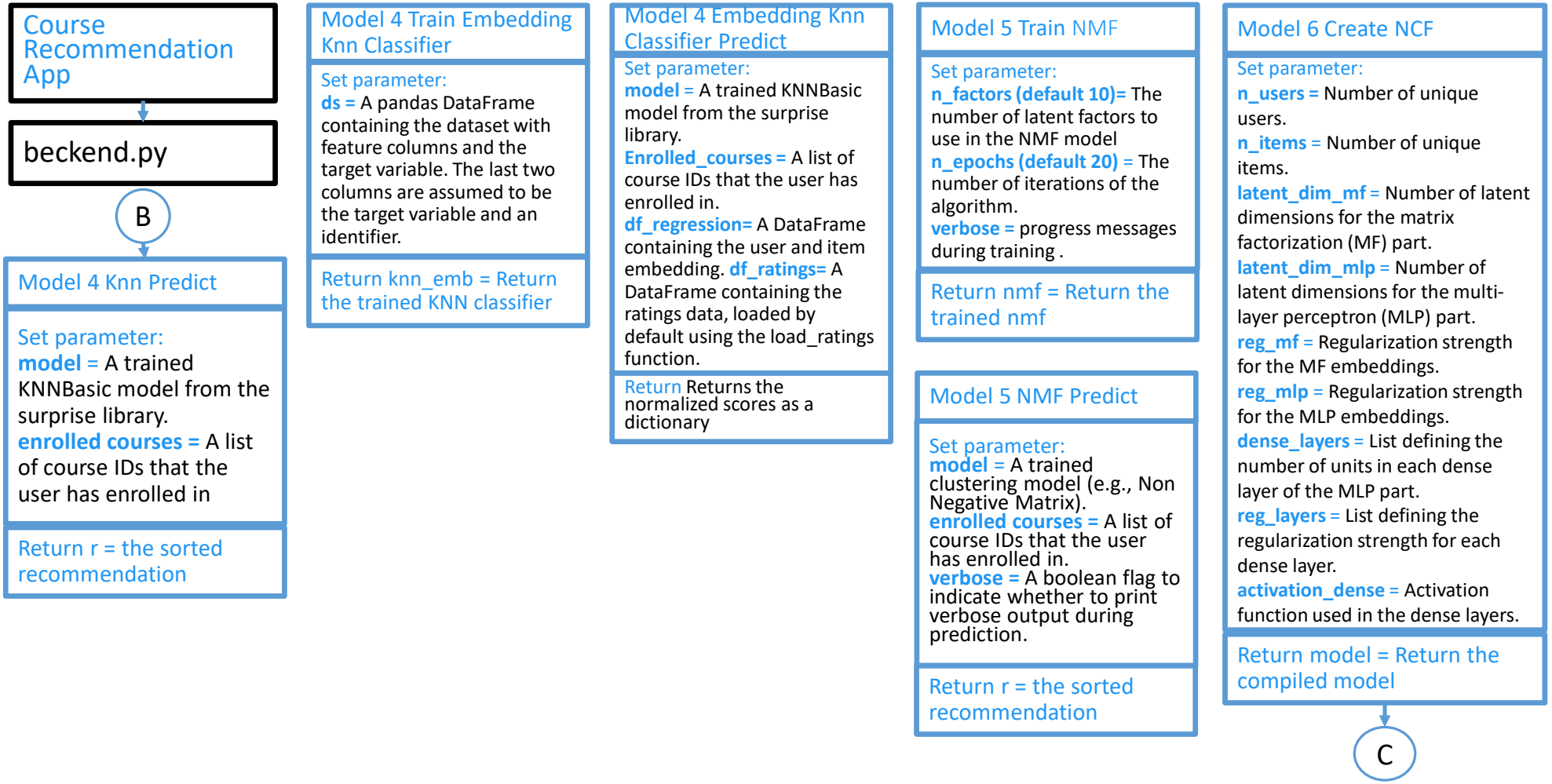
Recommender System Streamlit Flowchart



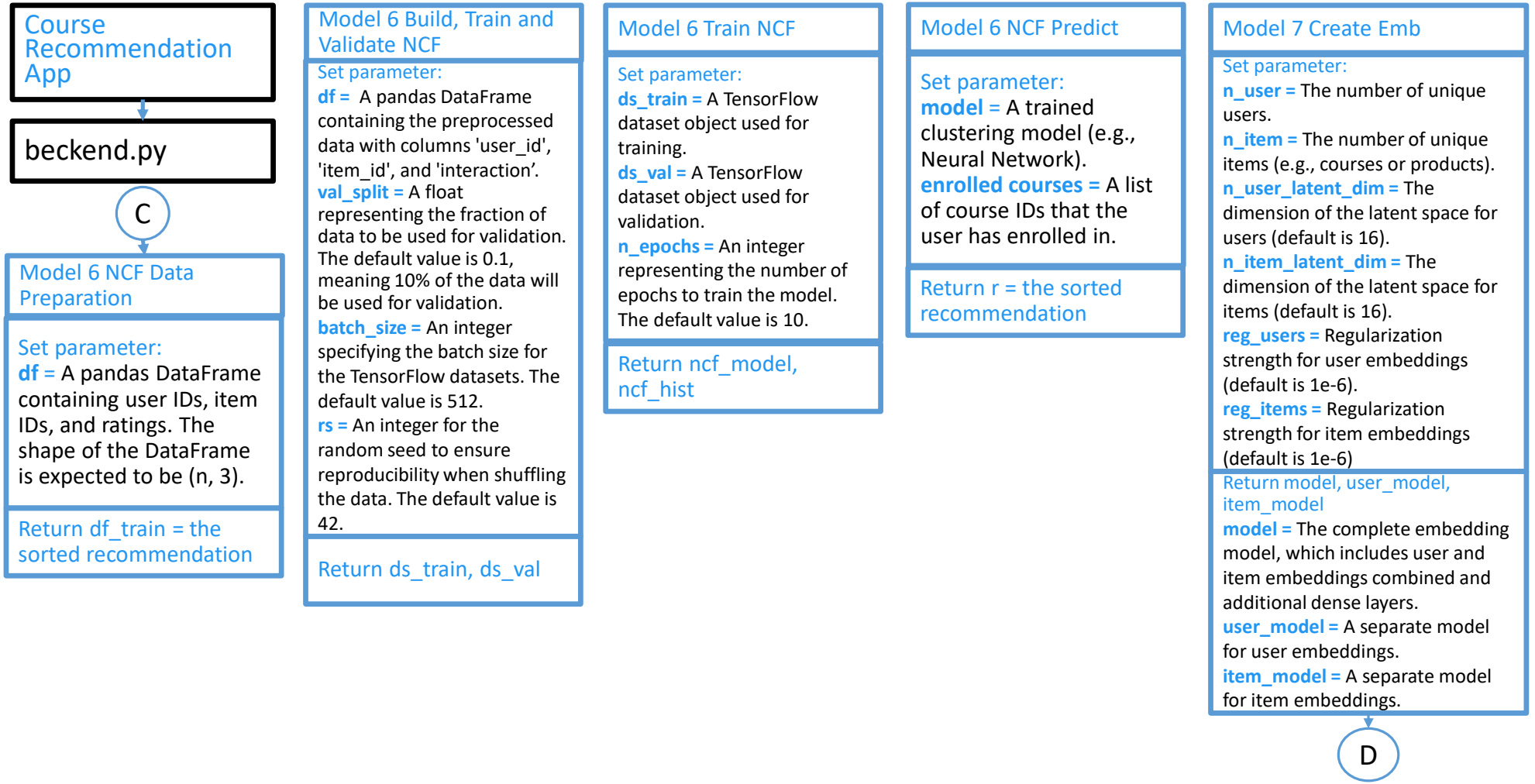
Recommender System Streamlit Flowchart



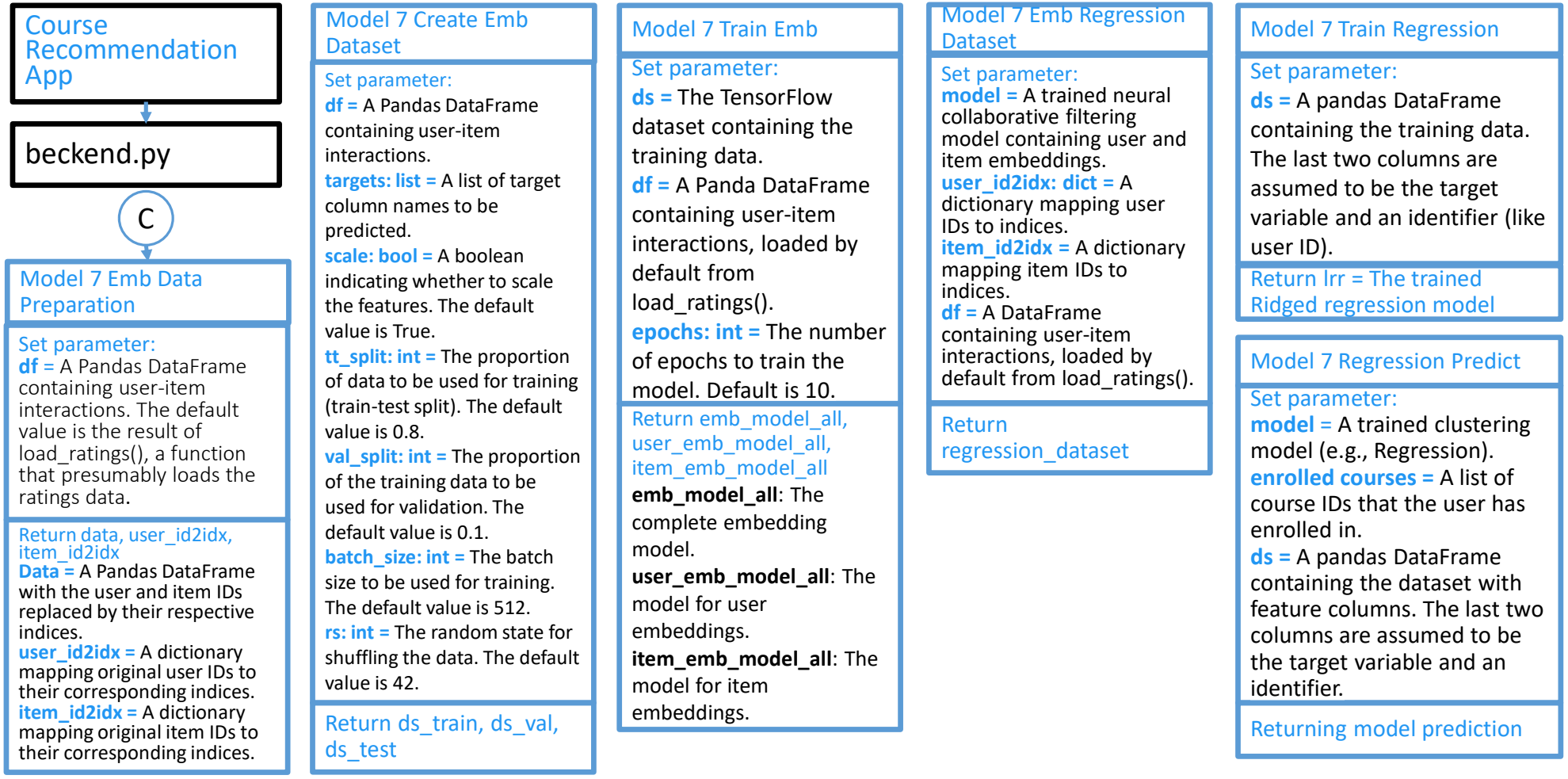
Recommender System Streamlit Flowchart



Recommender System Streamlit Flowchart



Recommender System Streamlit Flowchart



Architectural Design

Personal Recommended System

Architectural Design

App	Recommender System Data Processing and Model Training	Data Processing and Model Training	Data Processing and Model Training Cost	Components and Services EMR	EMR Configuration	EMR Cost	Storage	Storage Cost	Data Transfer	Data Transfer Cost	Monitoring & Logging	Monitoring & Logging Cost
Web Mobile	Apache Spark with Spark Mllib. Employ collaborative filtering or content-based filtering algorithms. Real-time or batch processing to provide recommendations. Integration with the web/app backend to serve recommendations to users.	Model Training and Inference: Using EC2 instances (p2.xlarge for GPU). Training: 10 hours/month. Cost: $0.90/hour * 10 = $9/month	$ 9.00	Apache Spark on Amazon EMR (Elastic MapReduce): Cluster Configuration: Master Node: Manages the Spark cluster. Worker Nodes: Perform data processing and model training. Instance Types: Choose instances based on processing requirements. Master Node: m5.large (2 vCPUs. 8 GB RAM) Worker Nodes: m5.xlarge (4 vCPUs. 16 GB RAM)	Cluster Configuration: Master Node: m5.large instance: $0.096 per hour Worker Nodes: m5.xlarge instances: $0.192 per hour Assume 4 worker nodes for sufficient processing power. EMR Costs: Master Node: $0.096/hour Monthly cost: $0.096 * 24 * 30 = $69.12 Worker Nodes: 4 * $0.192/hour Monthly cost: 4 * $0.192 * 24 * 30 = $552.96	$ 622.08	Storage for data and model artifacts. Assume 1 TB of storage. Cost: $0.023 per GB-month Monthly cost: 1024 GB * $0.023 = $23.55	$ 23.55	Assume 1 TB of data transfer per month. First 1 GB/month is free. Additional data transfer: $0.09 per GB Monthly cost: (1024 - 1) * $0.09 = $92.07	$ 92.07	AWS CloudWatch: Basic monitoring is free. Additional charges for detailed monitoring and logs. Assume $20/month for detailed monitoring and logs.	$ 20.00

Project Cost and Benefit Analysis

Personal Recommended System

Project Cost and Benefit Analysis

Metric	Budgeted	Actual (Example)	Variance	Notes
Initial Investment	$(87,833.40)	$(85,000.00)	+$2,833.40	Cost savings in setup
Year 1 Cash Flow	$13,899.60	$14,500.00	+$600.40	Higher early adoption
Year 2 Cash Flow	$20,157.90	$19,800.00	-$357.90	Minor delays in scaling
Year 3 Cash Flow	$27,042.03	$28,200.00	+$1,157.97	Improved customer retention
Year 4 Cash Flow	$34,614.57	$33,000.00	-$1,614.57	Market competition
Year 5 Cash Flow	$42,944.37	$45,000.00	+$2,055.63	Stronger-than-expected growth
Total Cash Flows	$138,658.47	$140,500.00	+$1,841.53	
NPV (10% Discount)	$12,086.39	$14,200.00	+$2,113.61	Higher actual cash flows
IRR	14.36%	15.10%	+0.74%	Outperformed expectations
LTV:CAC Ratio	70,071:1	75,000:1	4,929	Lower CAC achieved
ROI	**15.8%**	**18.2%**	**+2.4%**	Increased profitability

Recommendations:

- Reinvest Year 5 surplus ($2,055) into customer acquisition.
- Monitor Year 4 trends for competitive response strategies.
- Use ROI metrics to secure additional funding or resources.

Key takeaways:

- Actual **NPV (14,200) > *Budgeted* (12,086)**, validating project ROI.
- **Budgeted ROI (15.8%)** was exceeded by **actual ROI (18.2%)**, indicating higher-than-expected returns.
- **IRR (15.1%)** exceeds the 10% discount rate.
- Lower initial investment (85Kvs.85*Kvs*.87.8K) and **CAC reduction** drove LTV:CAC to **75,000:1**.
- Year 4 dip due to competition; mitigated by Year 5 surge.

Conclusion

Conclussion Part 1

- **Project Management**:
  - **Clear roadmap** and **defined milestones** ensured that project objectives were met on time and within scope.
  - **Resource allocation** and **risk management** enhanced decision-making, leading to optimized resource use.
- **Agile Scrum**:
  - **Iterative development** allowed for continuous improvements and faster delivery of working features.
  - **Customer feedback loops** were integrated, ensuring that user-centric features were prioritized, boosting system adoption.
- **Lean Six Sigma**:
  - **Process optimization** through Lean Six Sigma reduced waste and increased efficiency, particularly in data processing and model training phases.
  - **Quality improvements** minimized errors and bottlenecks, leading to smoother deployment and enhanced system performance.
- **Budgeted ROI (15.8%)** was exceeded by **actual ROI (18.2%)**, indicating higher-than-expected returns.

Conclussion Part 2

1. The BaggingClassifier has the lowest RMSE (0.127622). indicating better performance in predicting ratings or recommendations among the provided models.
2. From the provided list. models such as DecisionTreeClassifier (RMSE: 0.196842) and BaggingClassifier (RMSE: 0.127622) are typically less computationally expensive compared to neural network models like NN and NN_1 (RMSE: 0.534776).
3. If you're exploring the structure or patterns within the data without labeled examples. unsupervised learning is more appropriate. It can help in understanding the underlying structure of the data and finding hidden patterns.
4. Unsupervised learning algorithms like k-means clustering can be useful for segmenting data into distinct groups based on similarities.
5. Techniques like Principal Component Analysis (PCA) or t-distributed Stochastic Neighbor Embedding (t-SNE) are used for dimensionality reduction and visualization. which can be valuable for understanding high-dimensional data.
6. Unsupervised learning is often used for anomaly detection where the goal is to identify rare events or outliers in the data.
7. If you have a sufficient amount of labeled data. supervised learning models can be a good choice. For example. in classification or regression tasks where you have labeled examples of input-output pairs. supervised learning can be effective.
8. When the objective is well-defined and can be framed as predicting an outcome based on input features. supervised learning is suitable. For example. predicting customer churn. spam detection. sentiment analysis. etc.
9. Supervised learning models are evaluated based on metrics like accuracy. precision. recall. F1-score. etc.. which make it easier to assess model performance.
10. Sometimes. a combination of supervised and unsupervised learning techniques is used. known as semi-supervised learning. This can be beneficial when labeled data is limited but unlabeled data is abundant.

Points learn from Netflix’s Story (Credit)

Points from Netflix's Story

Business Case:

- In 2022, after a decade of remarkable growth, Netflix seemed to have reached a plateau.
  - **Competition** from new streaming services was intensifying, particularly in the US.
  - **Geopolitical conflicts** urged **the company's exit from certain regions**, where it had an expanding customer base.
  - **Rising inflation** was making **users more price-sensitive**, restricting Netflix's capacity to raise its subscription fees.

Problem Statements:

- **Netflix went from adding over 165,000 new customers daily** in the first quarter of 2020, to reaching a plateau in 2021, when **out of 60 million subscribers, approximately 30 million more people were using shared accounts**.
- In the first half of 2022, **Netflix experienced the worst six-month period in its history**, losing customers and prompting the company to **fire hundreds of people** and **scale back its programming**.
- This downturn led to a significant drop in its share price, **wiping out approximately $200 billion in market value**.

Points from Netflix's Story

Root Cause Analysis:

- Some users shared their accounts with partners or children they lived with, which was generally considered acceptable.
- Others shared with friends or relatives in different locations, a more problematic and common scenario.
- Additionally, there were instances of individuals sharing passwords with dozens of people, often reselling accounts to those unwilling or unable to pay through traditional means.

- This was Greg Peters approach, when the numbers **revealed untapped potential in how many subscribers would share their accounts - approximately 30 million more people were using shared accounts**..

Experimenting and Testing Possible Solutions:

- The company's management initiated two measures:
  - **Blocking Password Sharing:** aiming to curb the loss of potential revenue by ensuring that only paying subscribers could access Netflix's content.
  - **Introducing an Ad-Supported Version:** a new subscription tier designed to attract cost-sensitive customers who might be willing to endure advertisements in exchange for a lower subscription fee.
- Netflix developed a model to identify users who are traveling and differentiate them from those using someone else's password - **Blocking Password Sharing – The Experiment**

Points from Netflix's Story

- After identifying account sharers, **the next step was to decide how to make these users pay**.
  - On one side there was the belief that **Netflix should charge by residence**, similar to cable TV. This belief came from Reed Hastings, Netflix co-founder.
  - This would require users to pay per home and get another account for different locations.
  - The argument opposing this view was that **the residence model contradicted a core principle of Netflix: the ability to take the service anywhere**.
  - The alternative was **an individual user model**, allowing customers to access Netflix wherever they went, **with an additional fee for adding new users to their accounts**.

The Experimentation Roll Out:

- In 2022, Netflix introduced the **user model in Chile, Costa Rica, and Peru**, while **the residence model was deployed in five other Latin American countries**.
- This region, with its high incidence of password sharing, **served as an ideal testing ground due to common language (Spanish) and similar payment challenges**, as many residents lacked bank accounts.
- The results were clear-cut: **the subscriber-centric model was more successful.**

The Outcomes:

- Last year, **Netflix added 30 million new subscribers**, and in the first quarter of this year, **another 9.3 million.**
- According to Netflix, its **ad-supported plan now hosts 40 million monthly active users worldwide**, a significant **increase from 23 million in January**.
- This model not only increased the number of subscribers but also reduced churn rates (the rate at which customers unsubscribe) and minimized negative feedback on social media platforms.

Points from Netflix's Story

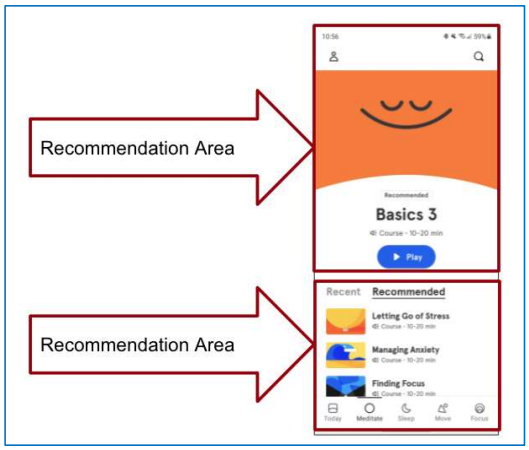
The subscriber-centric model's success can be attributed to several factors:

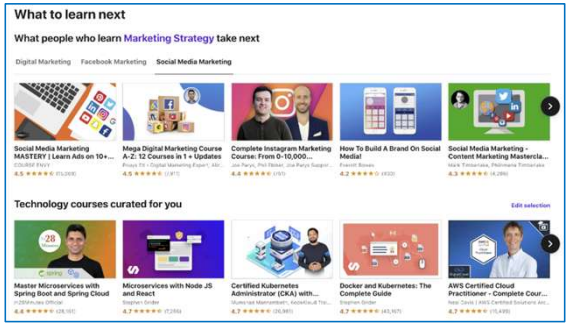
- **Flexibility**: users appreciated the ability to access Netflix from anywhere without being tied to a single household.
- **Cost-Effectiveness**: while there was an additional fee for adding new users, this was often seen as a better alternative than forcing each household member to have a separate subscription.
- **Reduced Friction**: the model minimised disruptions to the user experience, which could have led to dissatisfaction and cancellations.

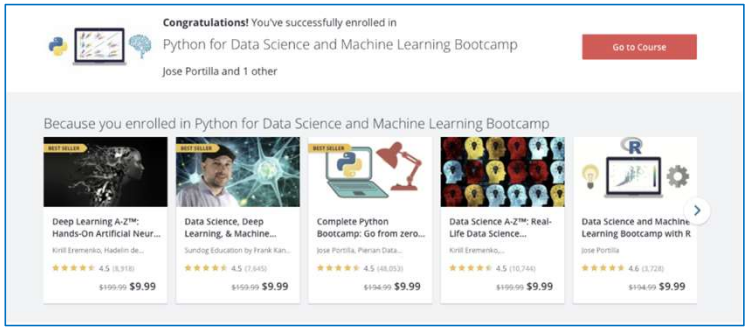
Project Related – AB Testing, Marketing Campaign and TV Ads and Process Capability
(Credit)

Recommender System

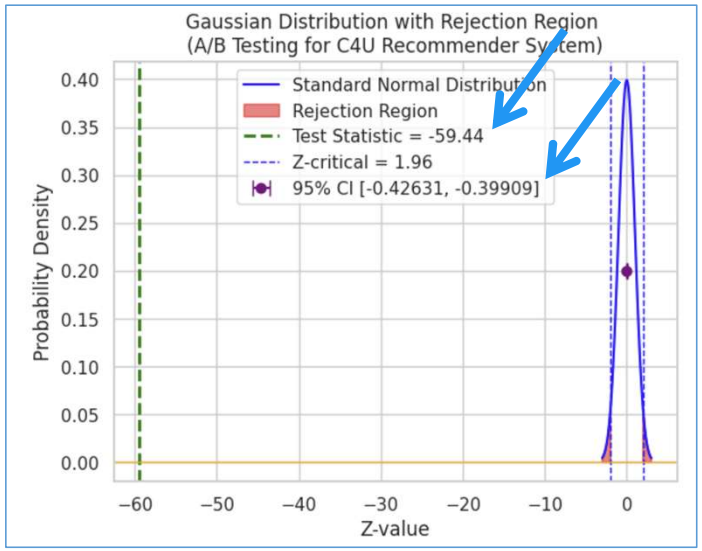
How many click from the control group and the experimental group?





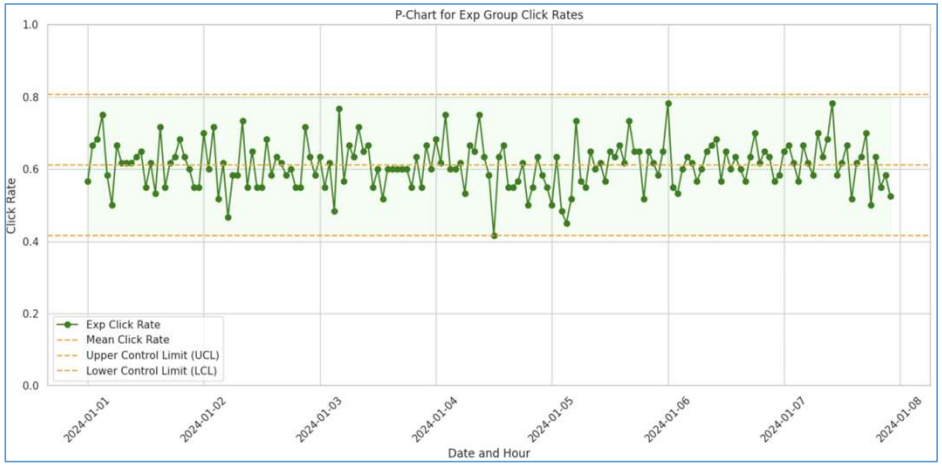


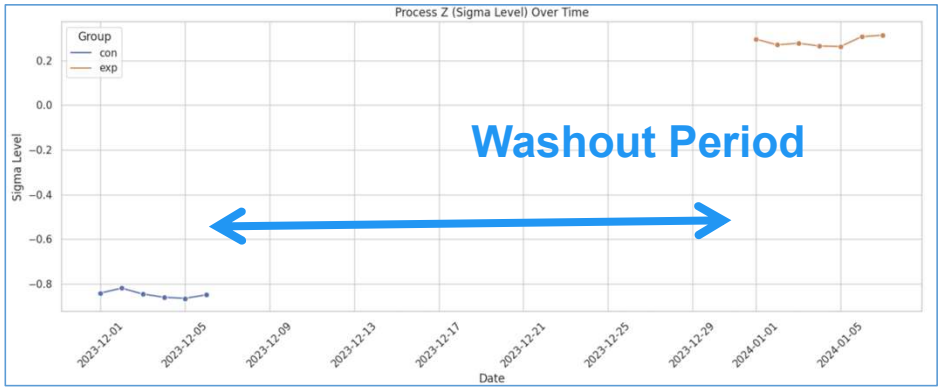
Recommender System



Probability

Z-Value





Con (Baseline)

Exp

Key takeaways : Will applying Control Chart and Process Capability add value to the Experimentation

Recommender System

How many converted with our ads vs product service announcement?

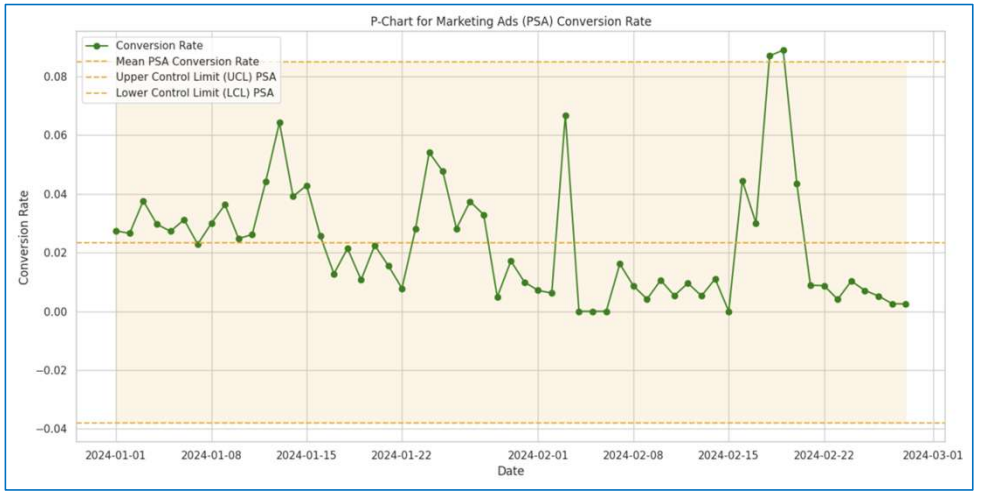
Ads

Products Service Announcement (PSA)

Plan

Ads and PSA

How many converted with our ads vs product service announcement?



Key takeaways : Should we compare to different Ads and determine which one is the most effective one?

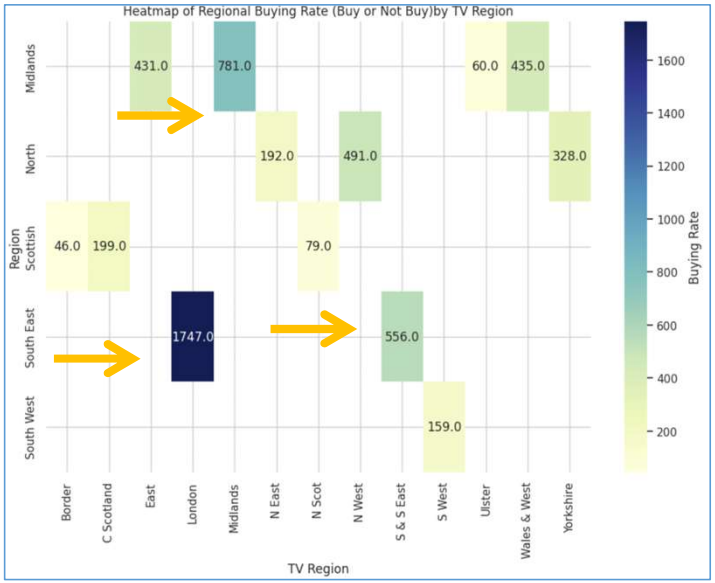
Plan

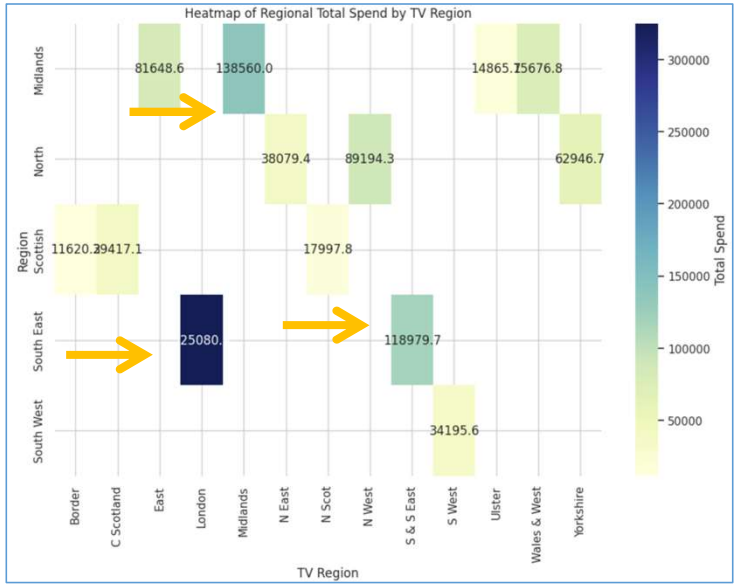
Recommender System

How many purchase with our tv campaign?

TV Campaign

Region and TV Region





- The highest buyers and total history spending are from London, South & South East, and Midlands.
- The middle range for both are from East, Wales & West, and Yorkshire.

Key takeaways : Is TV Ads still relevant? Which location is significant to our sales

Appendix

Appendix

Documents:

- https://www.kaggle.com/code/wahyuardhitama/task003-p001-ml-dl-rec-sys-course-20231025
- https://www.kaggle.com/code/wahyuardhitama/task003-p002-ml-dl-rec-sys-course-20231029
- https://www.kaggle.com/code/wahyuardhitama/task003-p003-ml-dl-rec-sys-course-20231101
- https://github.com/whyzie/Task003-ML-DL-Rec-Sys-Course-20231201