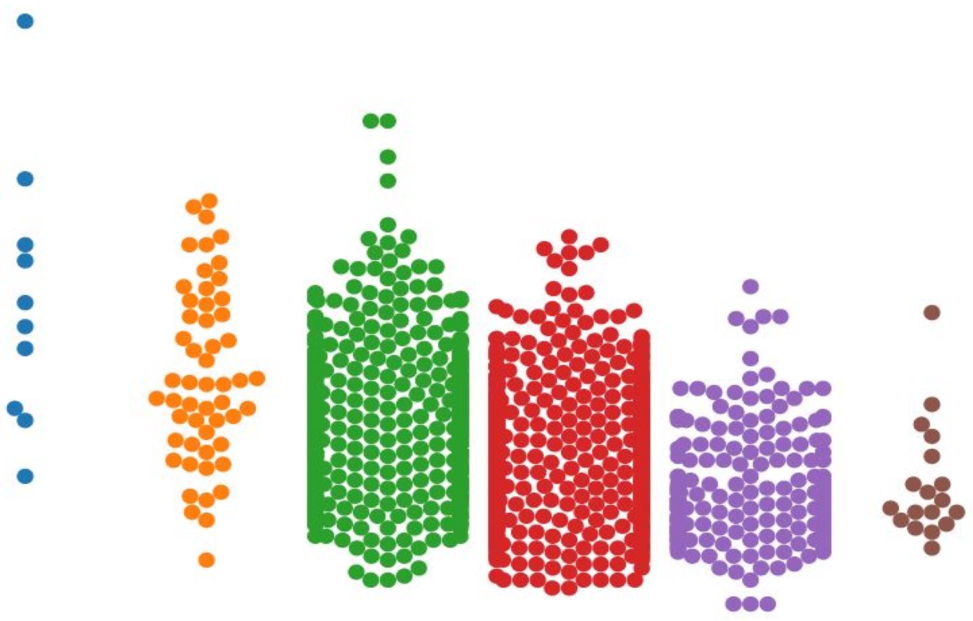


IBM Machine Learning Professional Certificate

Online Course Recommendation Model

Shankesh Raju MS

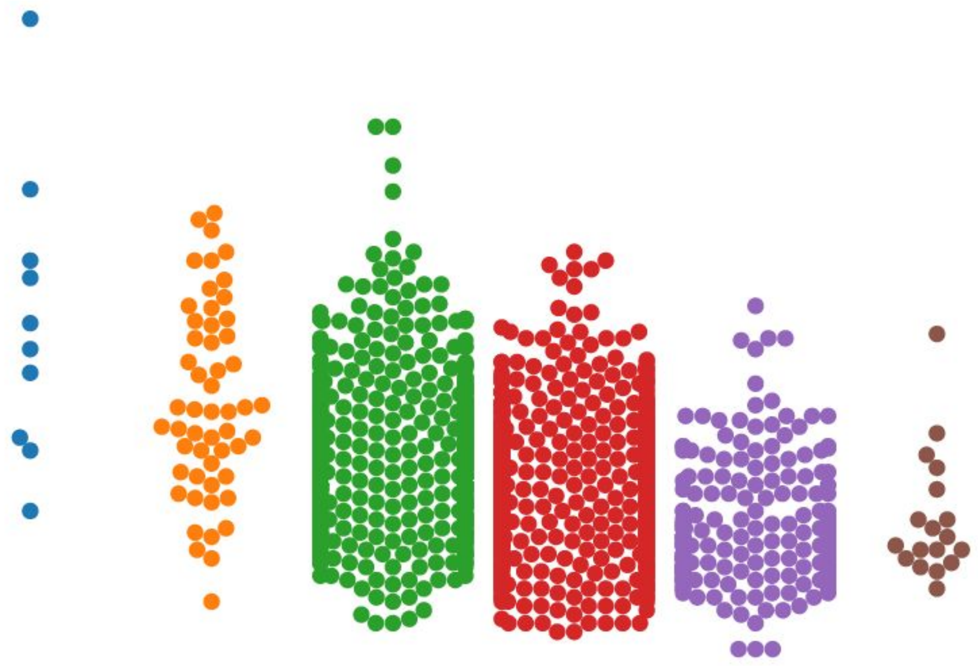
Outline

- Introduction
- Exploratory Data Analysis
- Word cloud
- Content - based Recommendation using user profile and genre
- Content - based Recommendation using clusters
- KNN - based collaborative filtering
- NMF - based collaborative filtering
- Neural Network embedding based on collaborative filtering
- Performance Comparison
- Conclusion

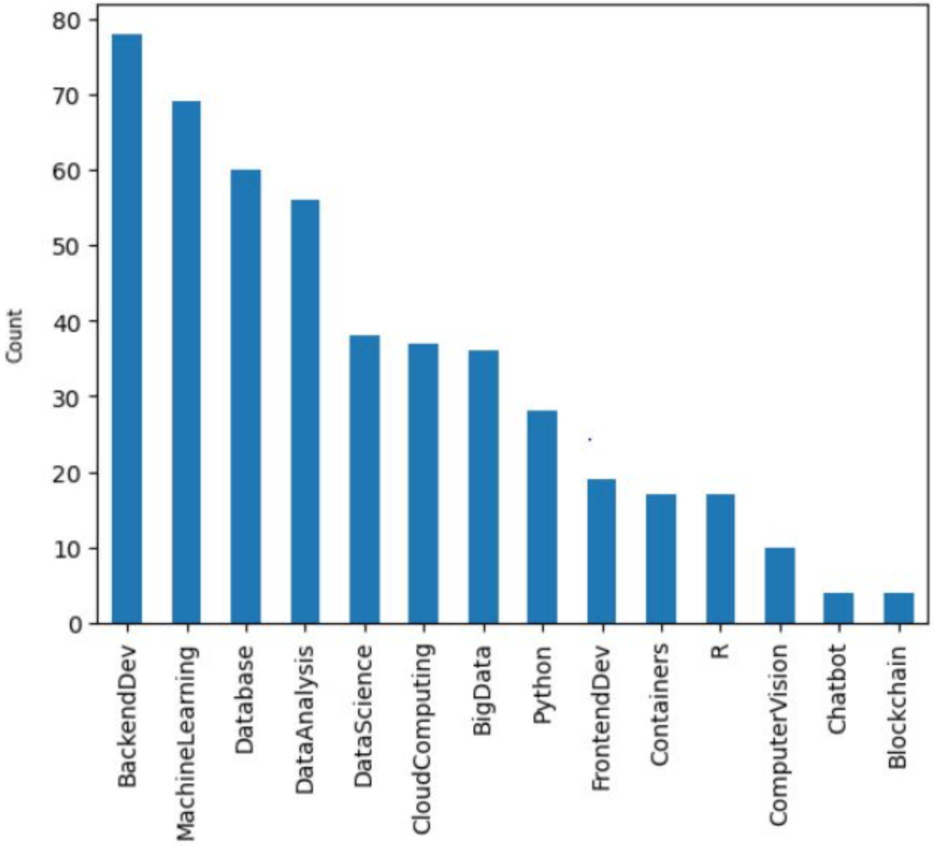
Introduction

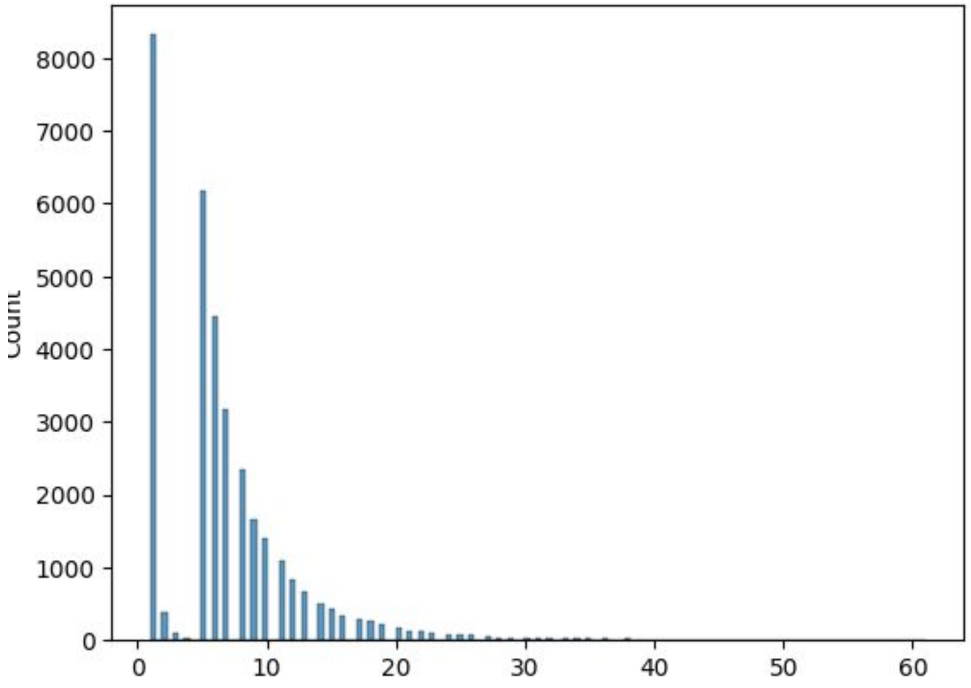
- **About**
  - Recommendation system for online courses
  - Model recommends new courses based on different criterias
  - Recommendations are generated based on user interest, course similarity in same group
- **Implementation**
  - using unsupervised learning
  - using supervised learning

Exploratory Data Analysis



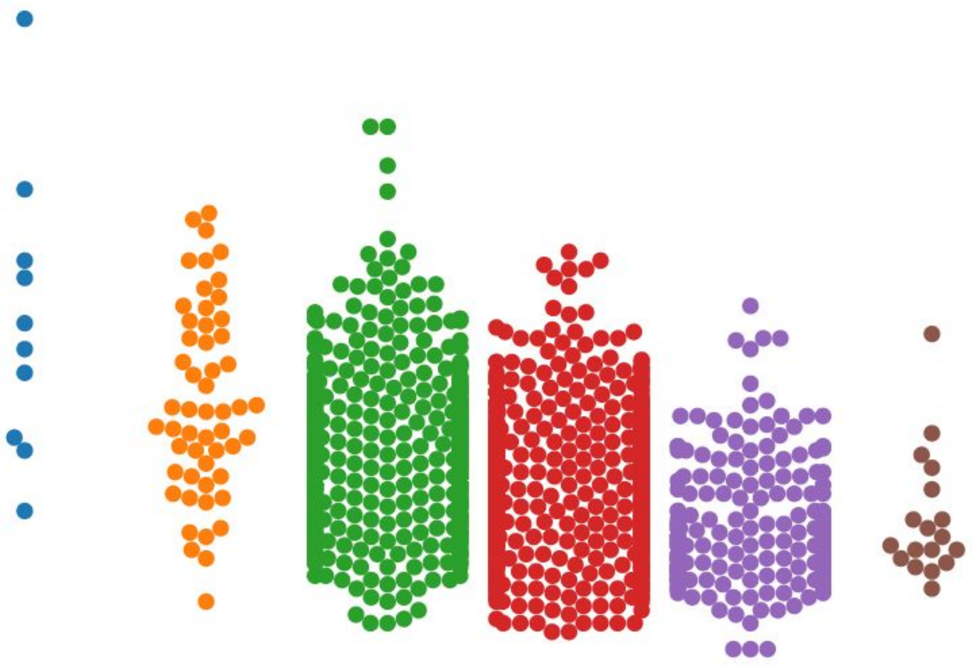
Course Enrollment and Genre Distribution



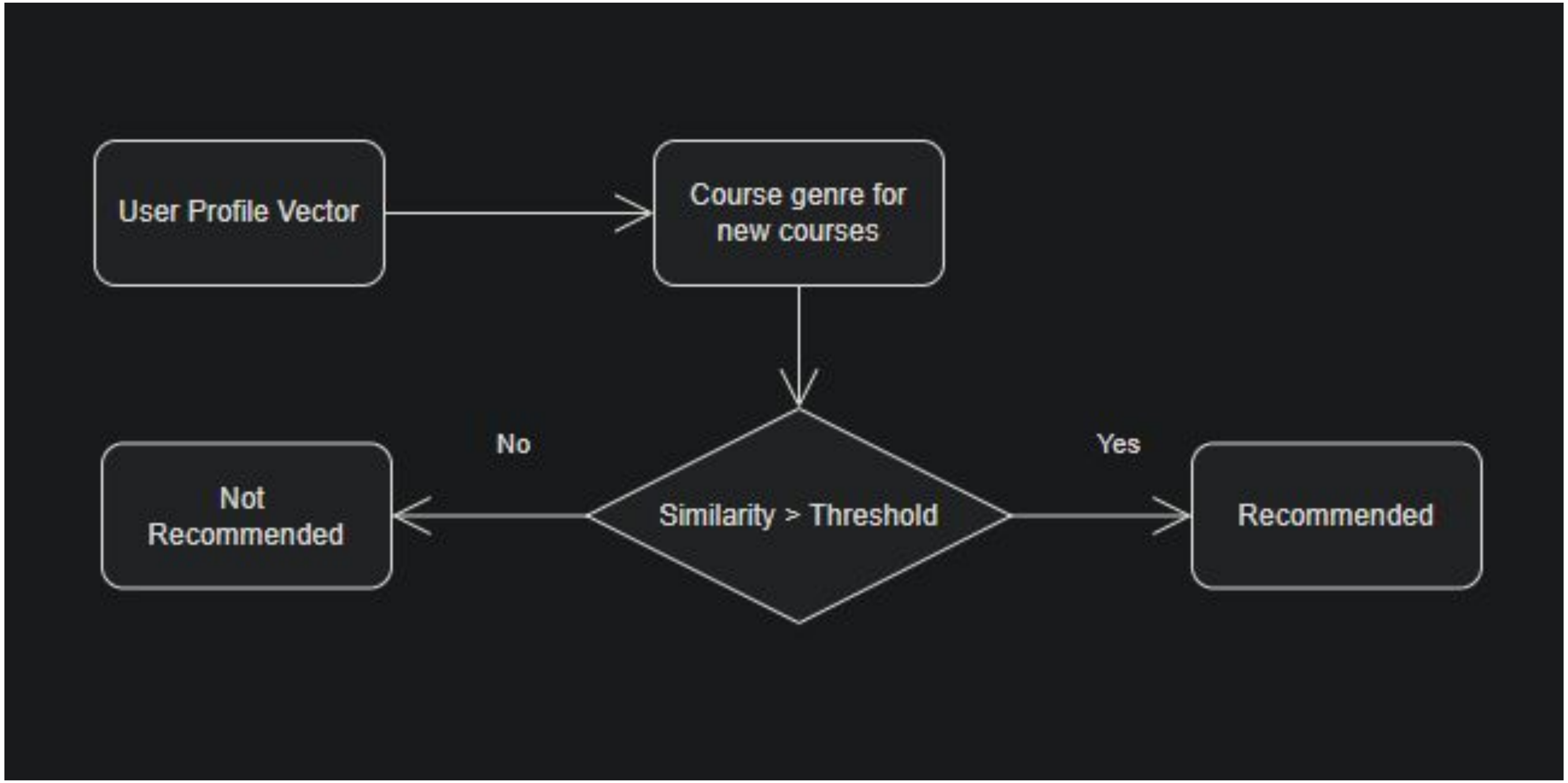


Word Cloud Titles

Content-based Recommender System
User Profile and Genre



Flowchart Implementation



Real-Time Implementation

- Creation of simple user rating Data Frame
- Generation of user profiles
- Generation of recommended scores
- Top 5 Recommendations

	user	Python	Database	MachineLearning
0	user0	3	0	3
1	user1	2	2	0

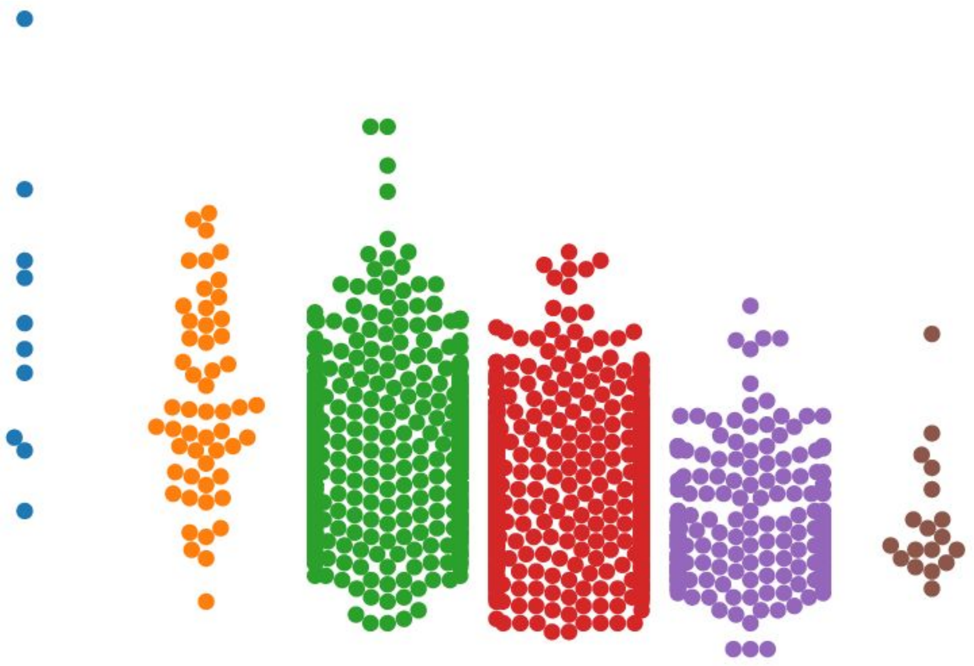
	User	Title	Rating
0	user0	Machine Learning with Python	3
1	user1	SQL with Python	2

Then, the average recommended courses per user is: $(2+2+3)/3 = 2.33$. The top-2 recommended courses are: `course3` : 2 times, and `course4` : 2 times.

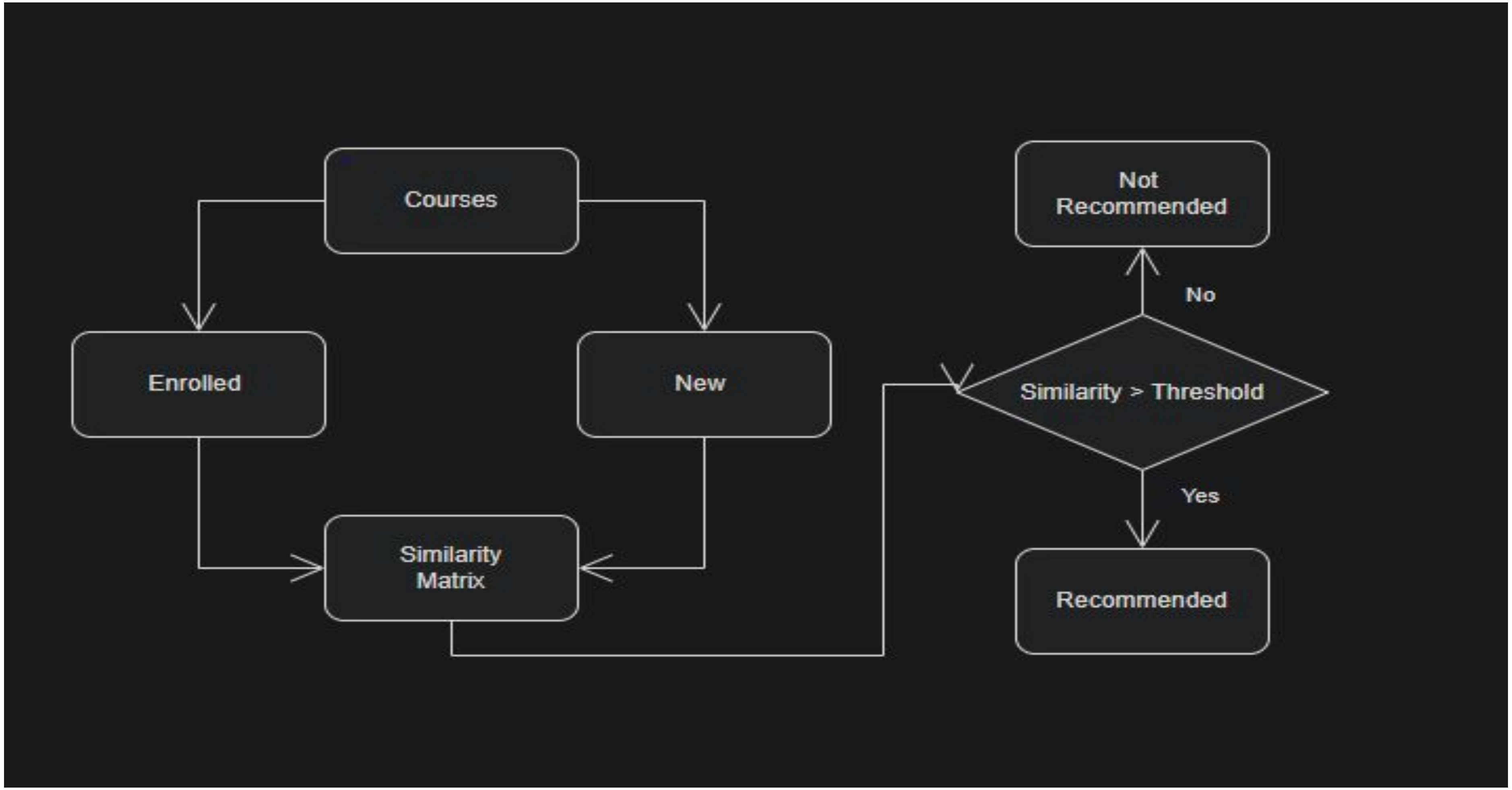
	USER	COURSE_ID	SCORE
0	37465	RP0105EN	27.0
1	37465	GPXX06RFEN	12.0
2	37465	CC0271EN	15.0
3	37465	BD0145EN	24.0
4	37465	DE0205EN	15.0

	Title	Python	Database	MachineLearning
0	Python 101	1	0	0
1	Database 101	0	1	0
2	Machine Learning with R	0	0	1

Content-based Recommender System
Course Similarity



Flowchart Implementation

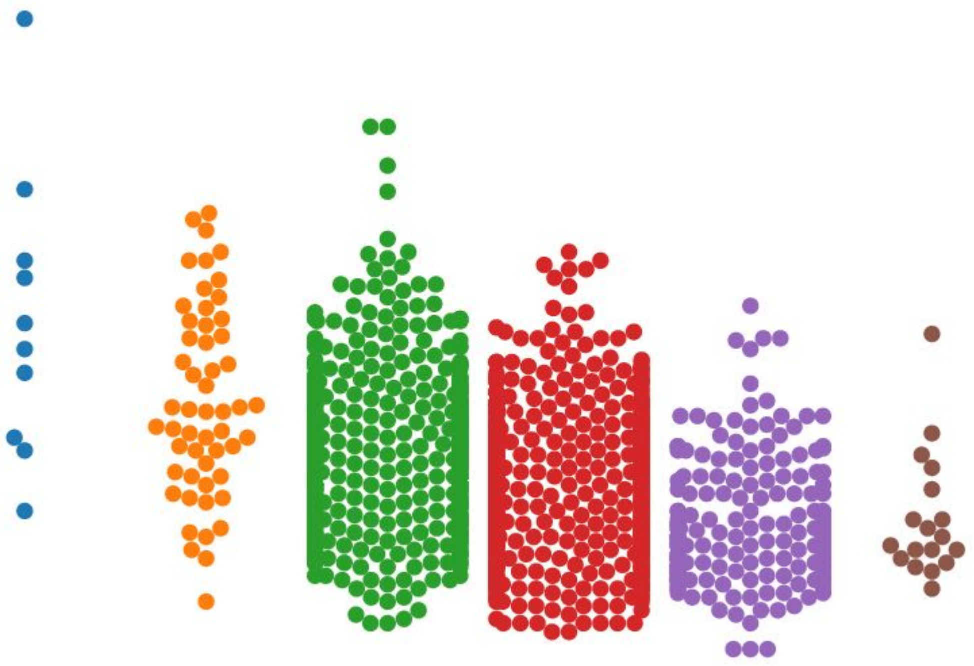


Evaluation

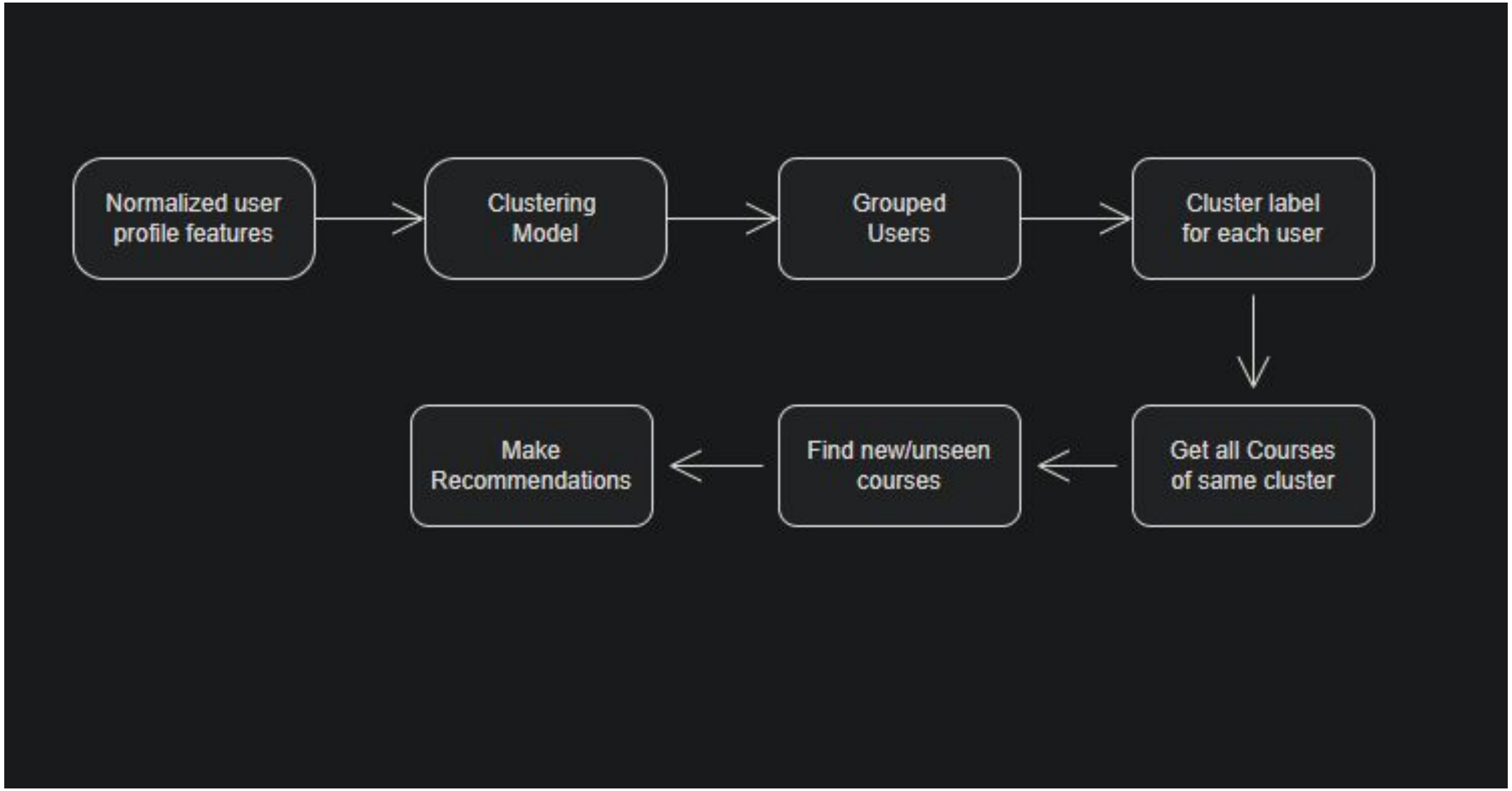
- Score Threshold = **0.73**
- On average, **12** new/unseen courses are recommended per user

1)'BD0101EN'
2)'DS0101EN'
3)'DS0110EN'
4)'excourse04'
5)'excourse23'
6)'excourse32'
7)'excourse33'
8)'excourse36'
9)'excourse63'
10)'excourse67'
11)'excourse68'
12)'excourse72'

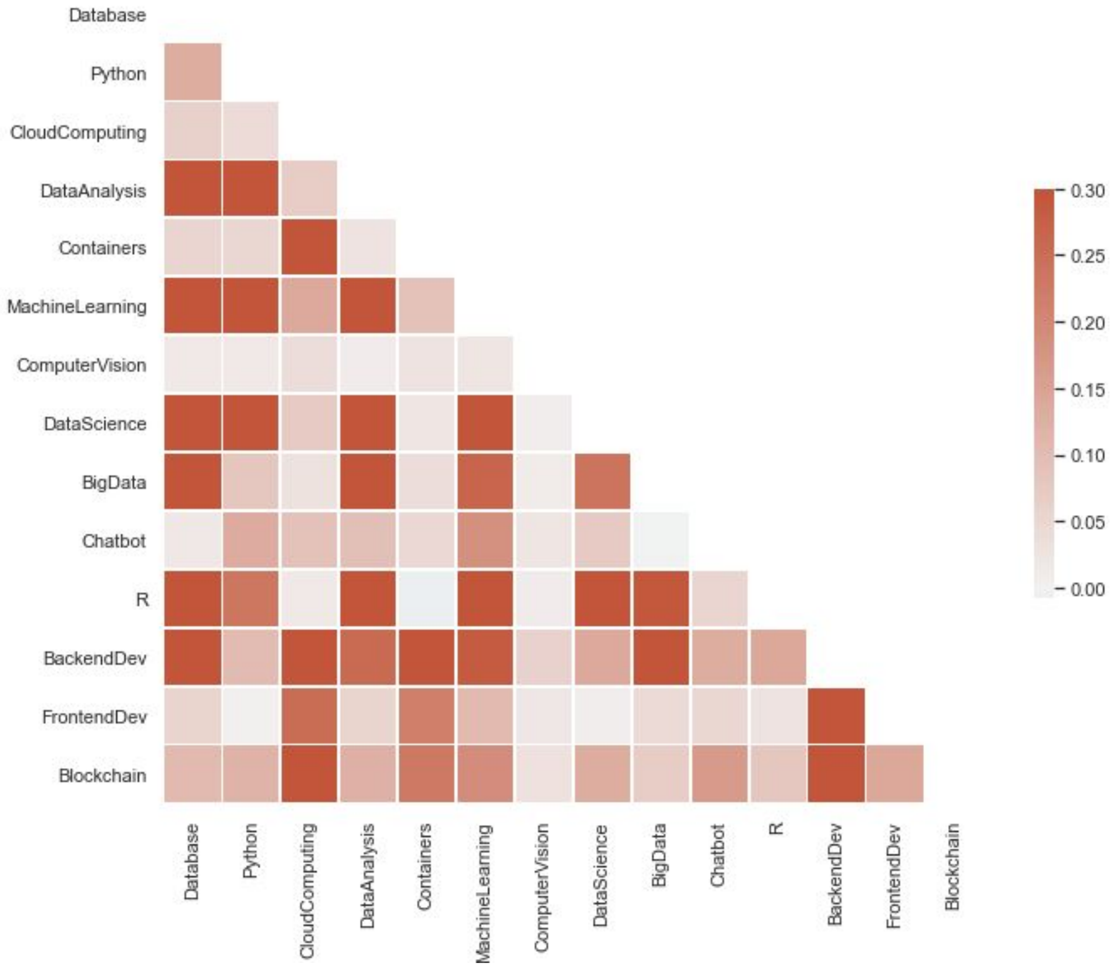
Content-based Recommender System
User Profile Clustering



Flowchart Implementation



Applying PCA on user profile to reduce dimensionality



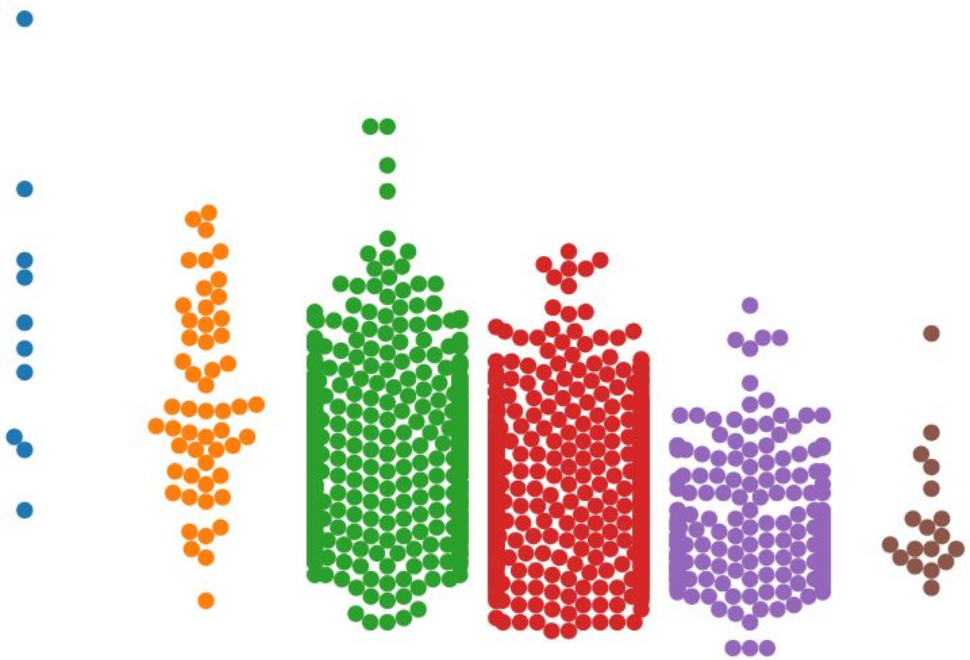
user	PC0	PC1	PC2	PC3	PC4
2	17.772494	0.200681	1.730609	2.567359	-3.825814
4	7.145199	-2.847481	2.358636	-0.576654	0.398803
5	11.363270	1.873619	-1.522077	1.076144	-1.711688
7	-1.834033	-0.277462	0.564905	0.053470	-0.064440
8	-1.049125	-0.684767	1.072765	0.006371	-0.005695
...	...	...	...	...	...
2102054	0.633824	0.108815	-0.388871	-0.122665	-0.098364
2102356	-2.095339	0.135058	0.244727	-0.088185	0.025081
2102680	0.625943	-0.547167	-1.692824	-0.630589	0.166632
2102983	-2.036832	-0.153534	0.162852	0.082651	-0.126419
2103039	-2.036832	-0.153534	0.162852	0.082651	-0.126419

Evaluation

- Average recommended courses per user is 2.33
- The top-2 recommended courses are
  - Course 3 - Twice
  - Course 4 - Twice

	user	item	cluster
0	1502801	RP0105EN	9
1	1502801	BD0131EN	9
2	1502801	BD0212EN	9
3	1502801	BD0115EN	9
4	1502801	BD0211EN	9
...	...	...	...
9397	630511	BD0121EN	0
9398	630511	SC0101EN	0
9399	630511	BD0111EN	0
9400	630511	BD0115EN	0
9401	630511	PY0101EN	0

KNN-Based Collaborative Filtering



Implementation

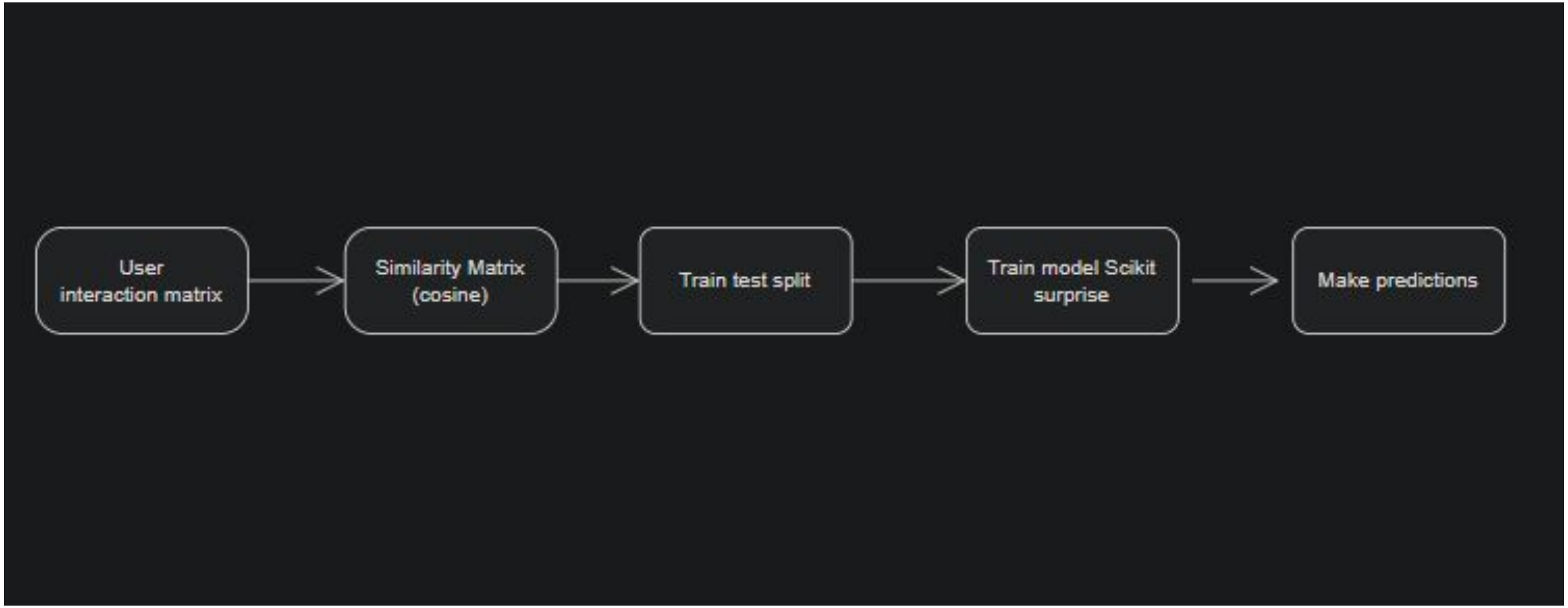
- Collaborative filtering is probably the most commonly used recommendation algorithm, there are two main types of methods
  - **User-based** collaborative filtering is based on the user similarity or neighborhood

$$\hat{r} * ui = \frac{\sum *v \in N_i^k(u) \text{similarity}(u, v) \cdot r\_vi}{\sum\_v \in N_i^k(u) \text{similarity}(u, v)}$$

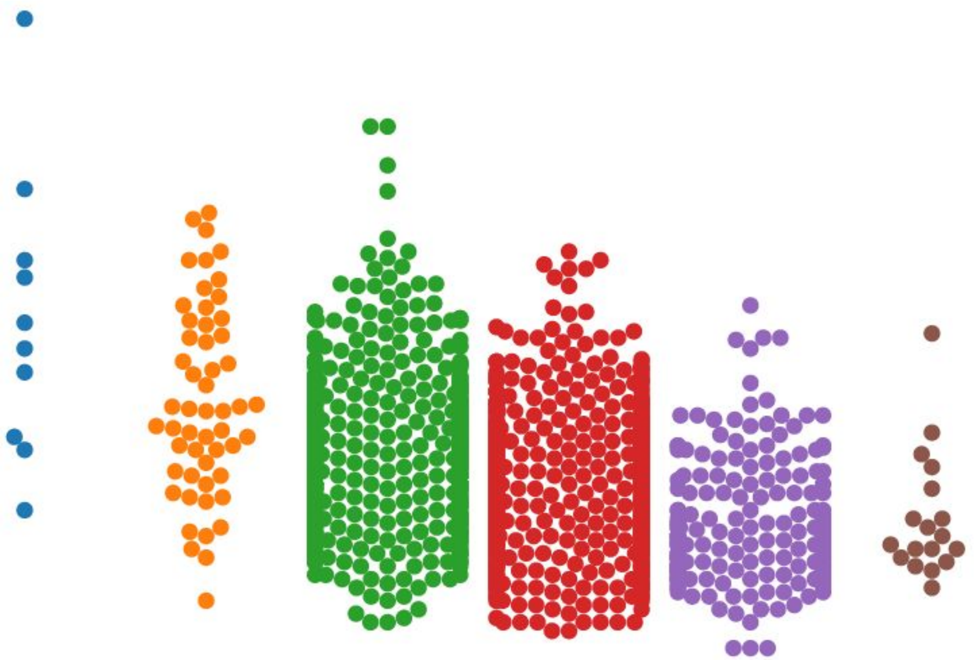
  - **Item-based** collaborative filtering is based on similarity among items

$$\hat{r} * ui = \frac{\sum *j \in N_u^k(i) \text{similarity}(i, j) \cdot r\_uj}{\sum\_j \in N_u^k(i) \text{similarity}(i, j)}$$

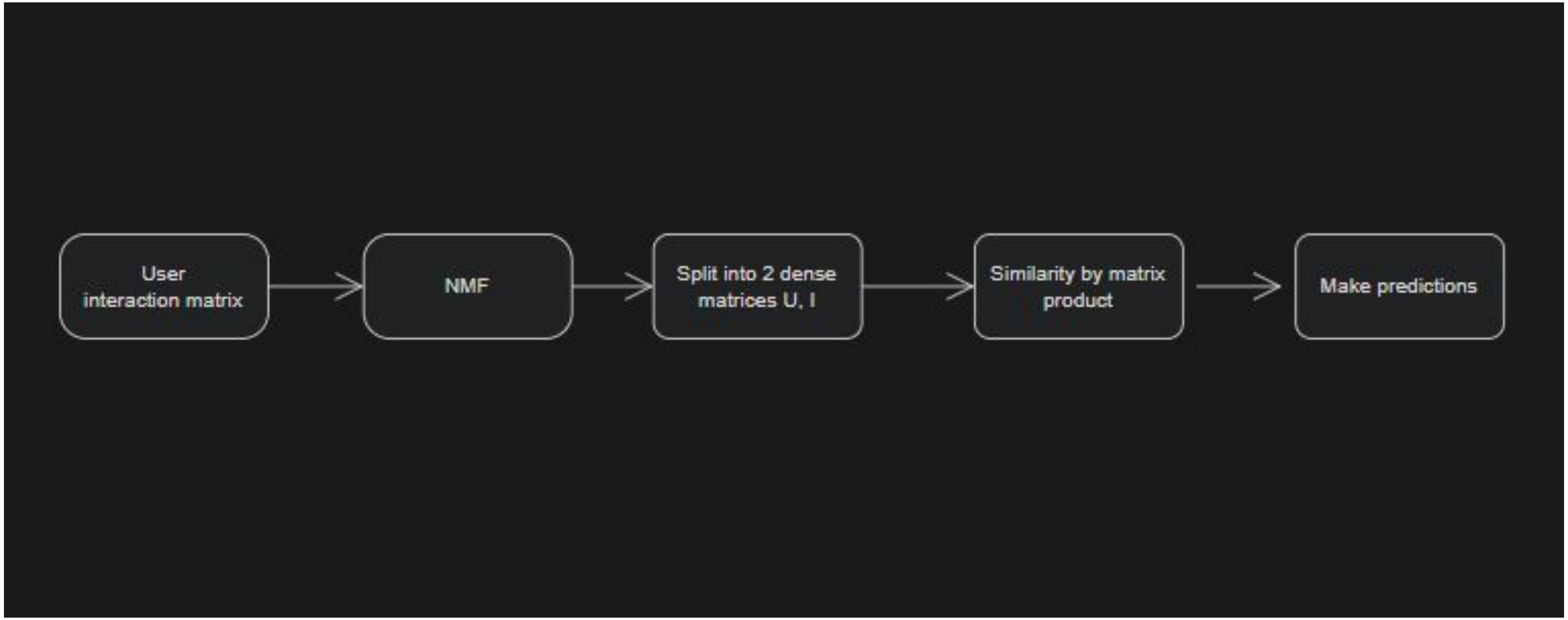
Flowchart Implementation



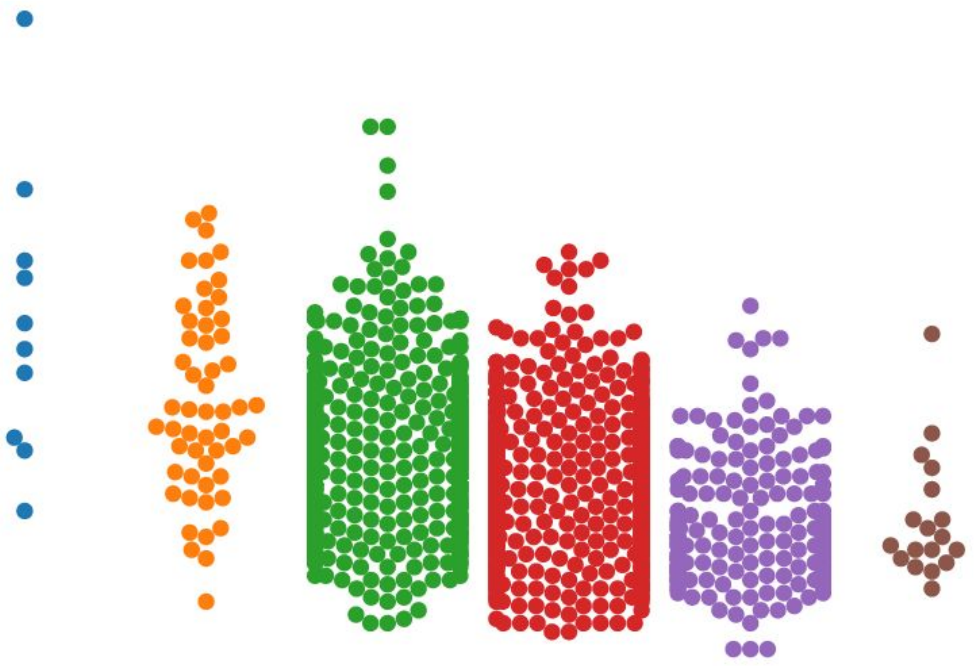
NMF-Based Collaborative Filtering



Flowchart Implementation



Neural Network Embedding - Based Collaborative Filtering



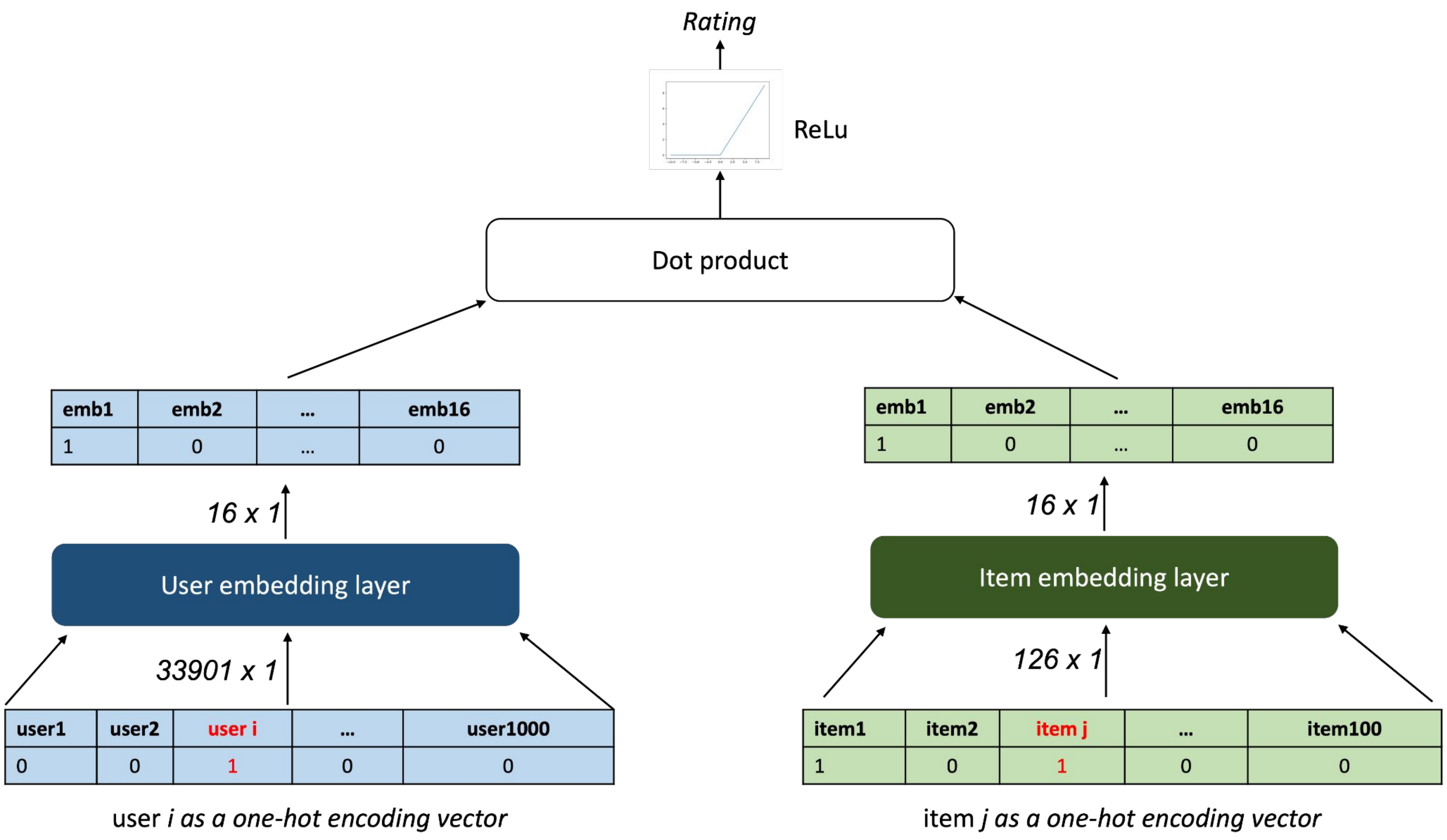
Implementation

- With explicit features vectors, we can perform machine learning tasks such as calculating the similarities among users or items, finding nearest neighbors, and using dot-product to estimate a rating value.

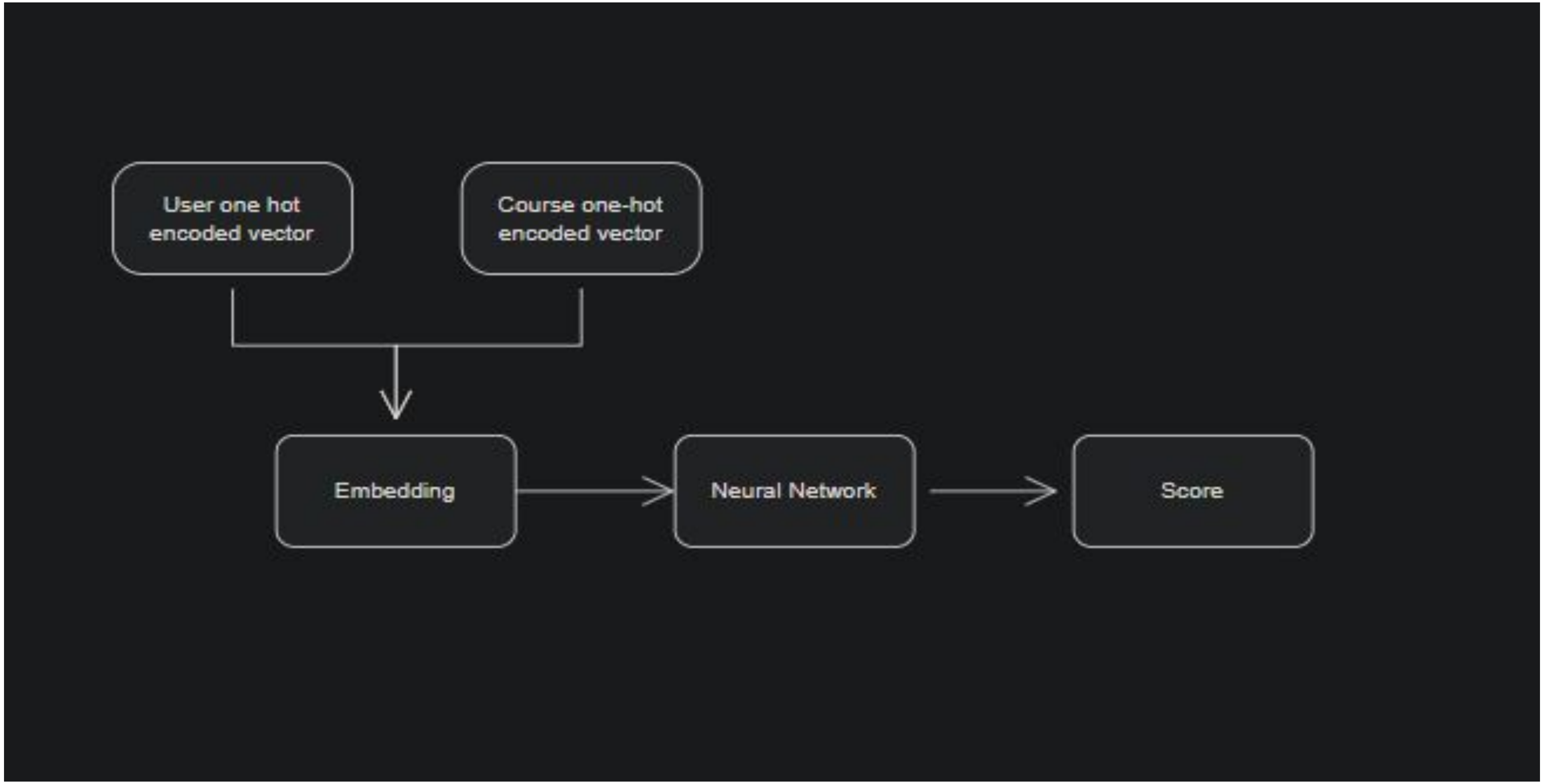
- The main advantage of using these explicit features is they are highly interpretable and yield very good performance as well. The main disadvantage is we need to spend quite some effort to build and store them

- Non-negative Matrix Factorization decomposes the user-item interaction matrix into user matrix and item matrix, which contain the latent features of users and items and you can simply dot-product them to get an estimated rating

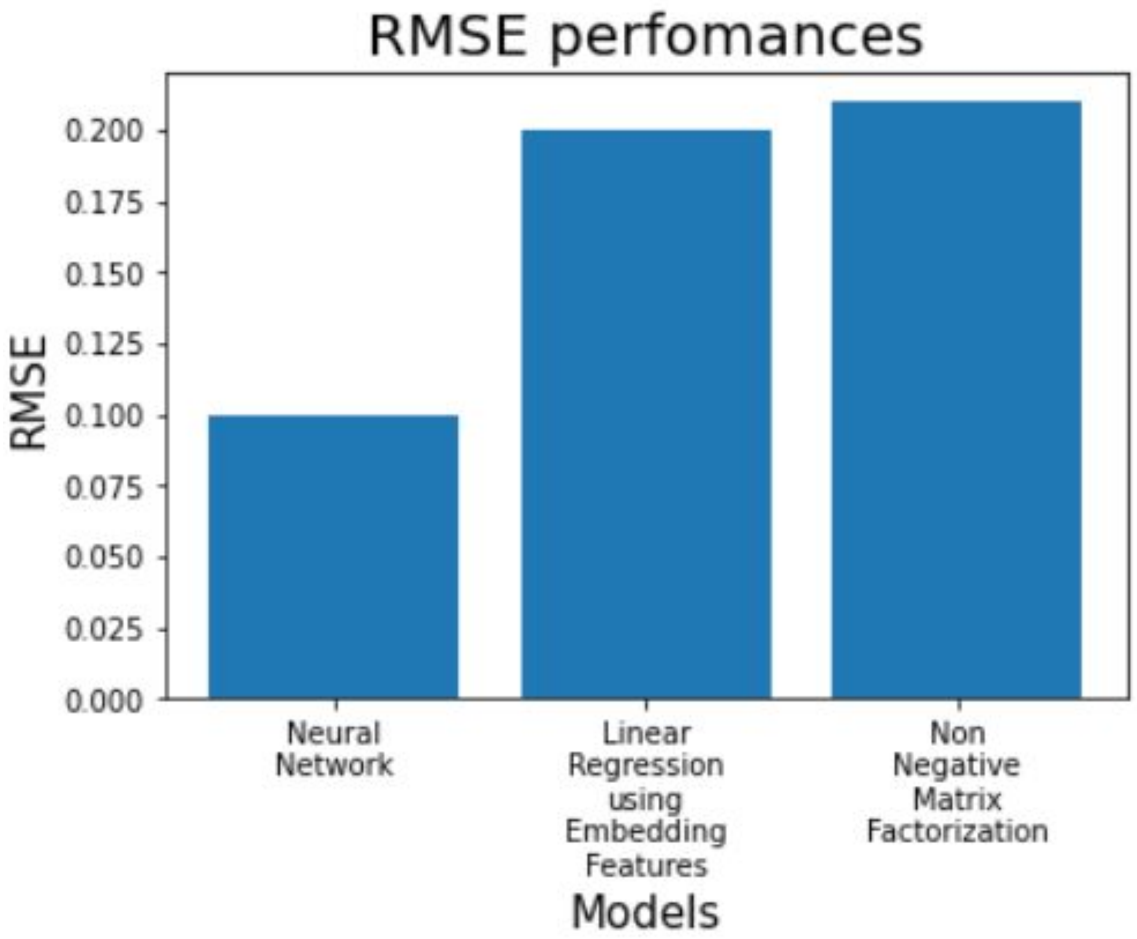
Rating Implementation



Flowchart Implementation



Algorithm Evaluation



Appendixes

- https://github.com/sabhashanki
- https://www.codechef.com/users/androshanki