FINAL REPORT - RECOMMENDATION SYSTEM

Machine Learning Capston - IBM Professional Certificate

OUTLINE

- Introduction
- Exploratory Data Analysis
- Content-based recommendation using user profile and course genres
- Content-based recommendation system using course similarity
- Content-based recommendation system using user profile clustering
- KNN based collaborative filtering
- NMF based collaborative filtering
- Neural network embedding based collaborative filtering
- Collaborative filtering algorithms evaluation
- Conclusion
- Innovative Insights

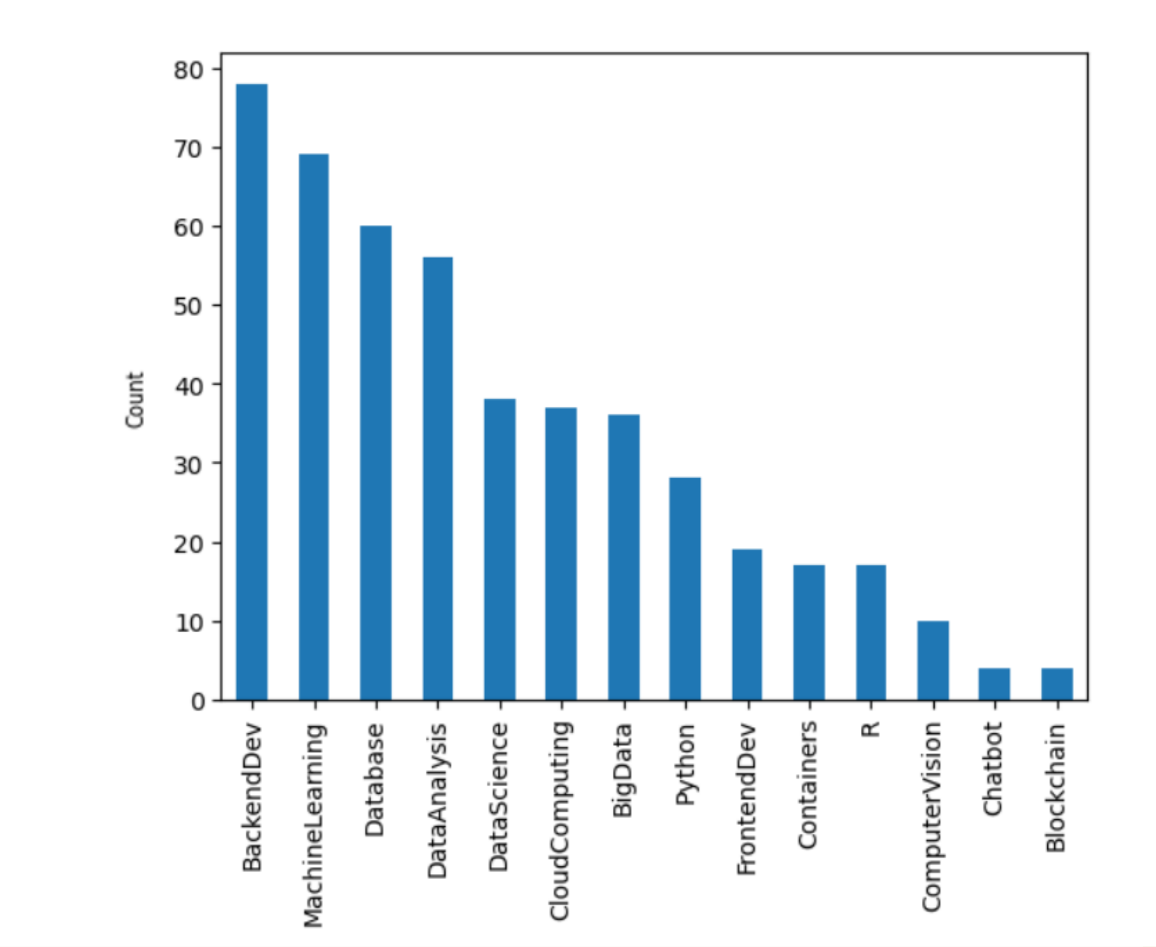
Introduction

This project is currently at the Proof of Concept (PoC) phase so the main focus at this moment is to explore and compare various machine learning models and find one with the best performance in off-line evaluations.

A course recommendation system will help in:

- Finding better courses
- Finding courses that well suits each person’s interests
- We aim to find the best courses to recommend to users based on their interests, their friend’s interests, and the courses they are enrolled in.

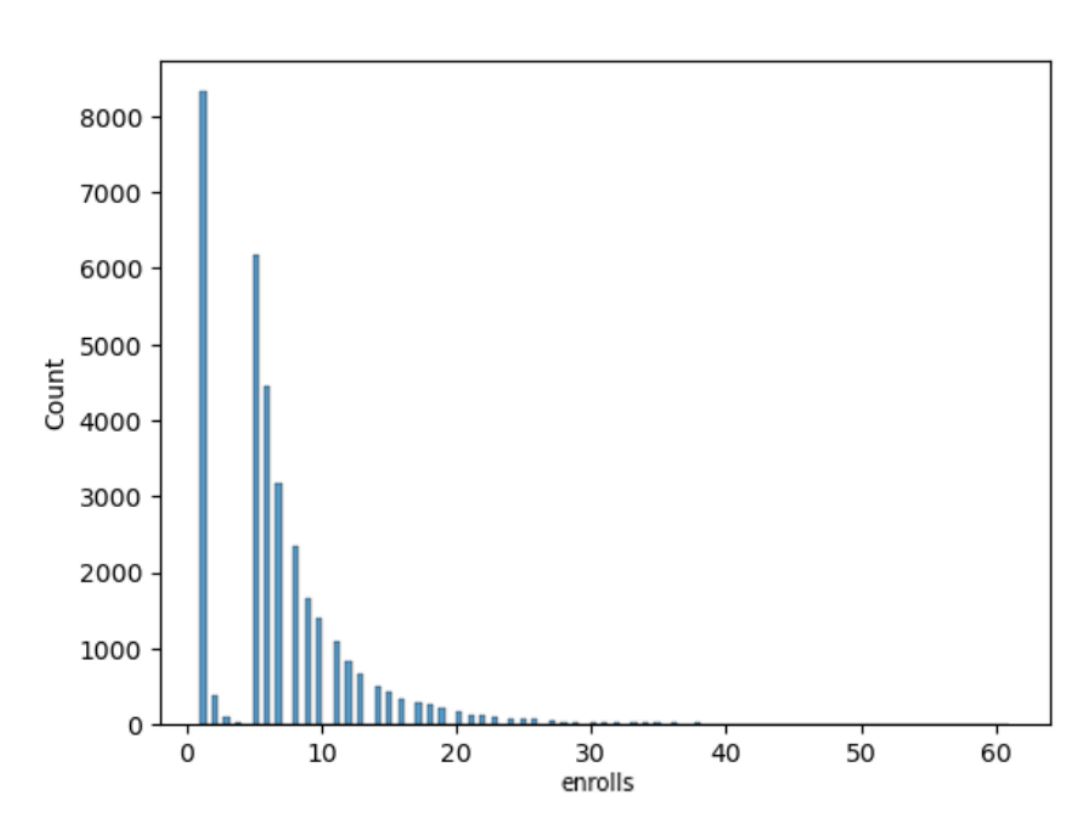
Exploratory Data Analysis



Course Counts Per Genre

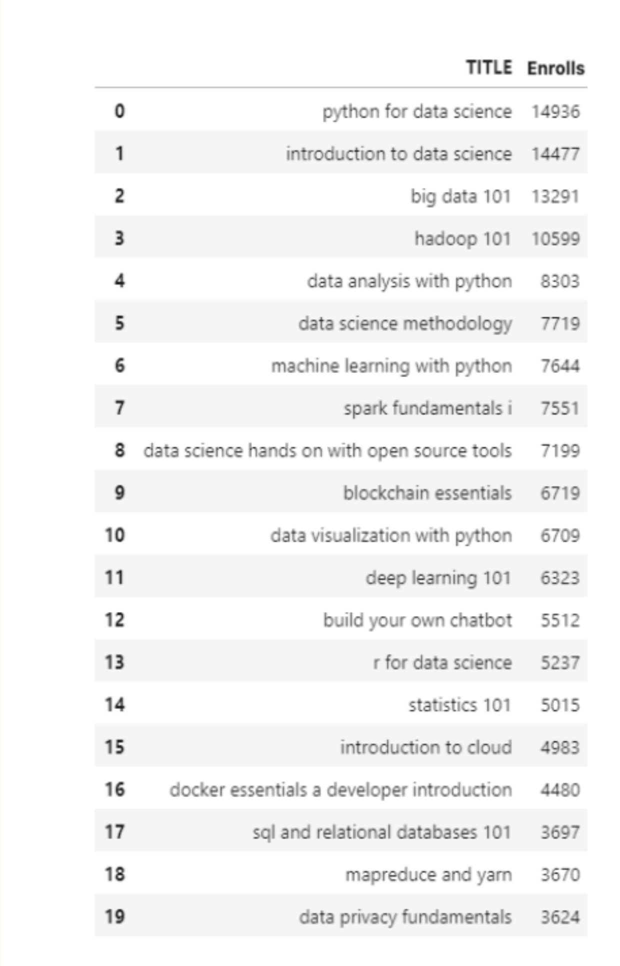
Exploratory Data Analysis

Course Enrollment Distribution

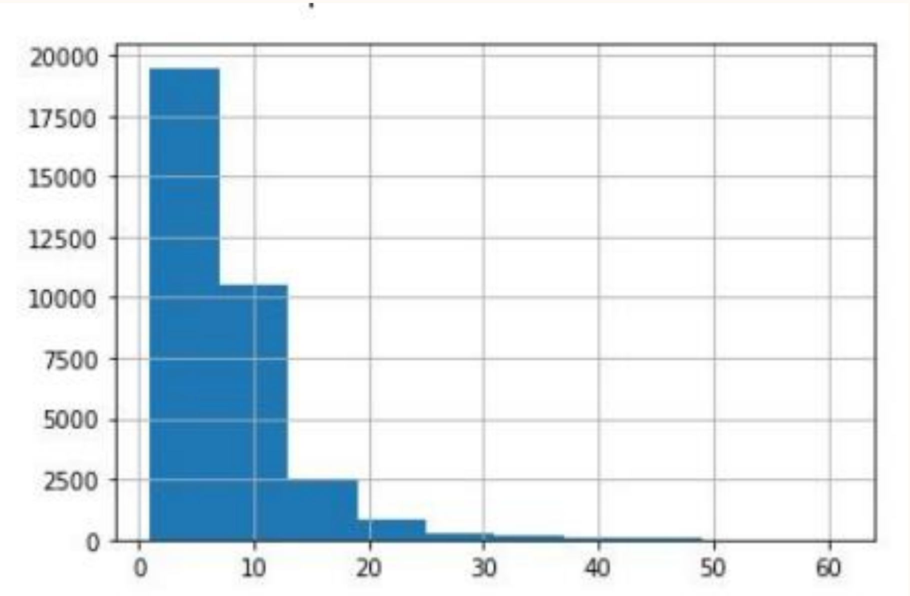


Exploratory Data Analysis

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
19	data privacy fundamentals	3624

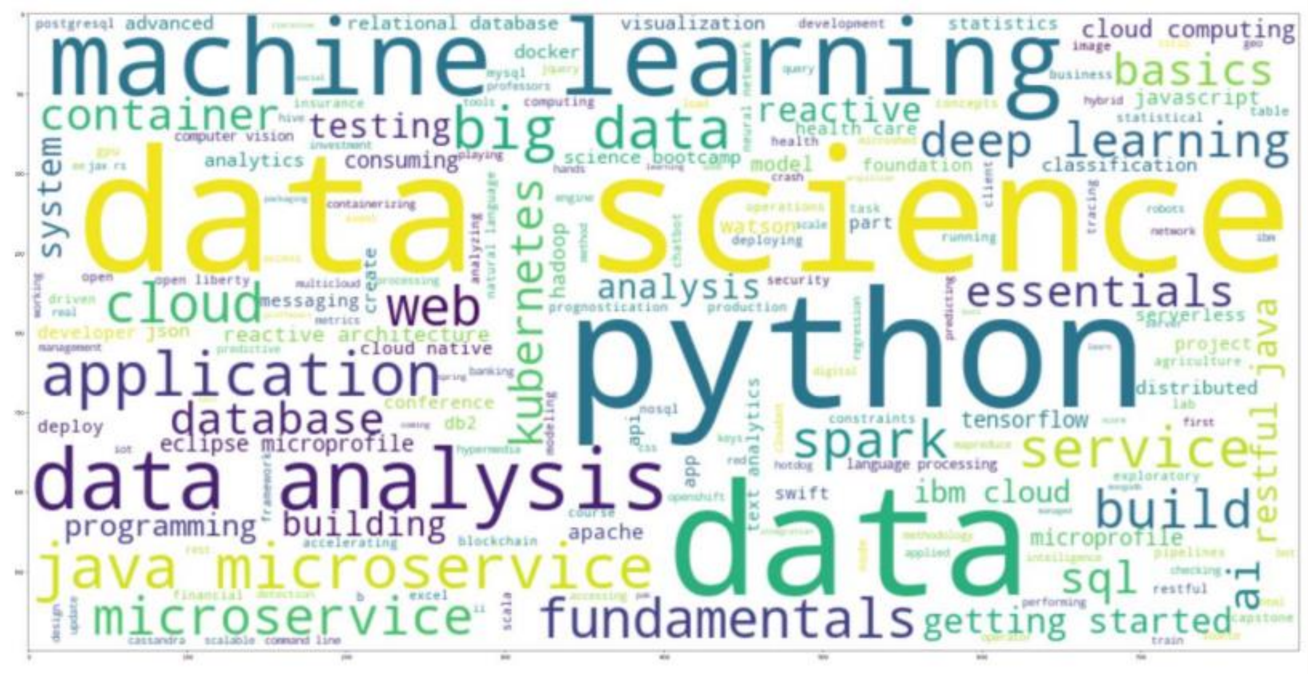


The histogram of user rating counts

Exploratory Data Analysis

The 5 most common words used in the Title:

1. Data
2. Data Science
3. Python
4. Machine Learning
5. Data Analysis



Word cloud of Course Titles

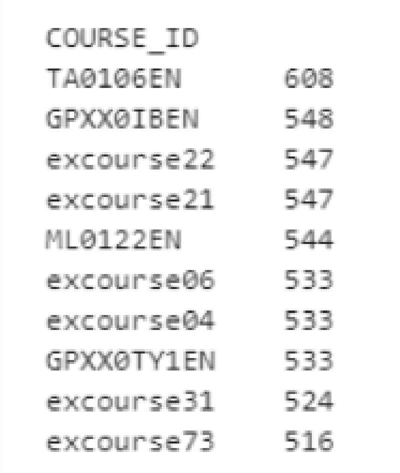
Content-based recommendation using user profile and course genres

with K= 10 (Score_threshold)

1. On average, how many new/unseen courses have been recommended per user (in the test user dataset) **18.82**
2. What are the most frequently recommended courses?

Return

the top10 commonly recommended courses across all users



```
COURSE_ID
TA0106EN        608
GPXX0IBEN       548
excourse22      547
excourse21      547
ML0122EN        544
excourse06      533
excourse04      533
GPXX0TY1EN      533
excourse31      524
excourse73      516
```

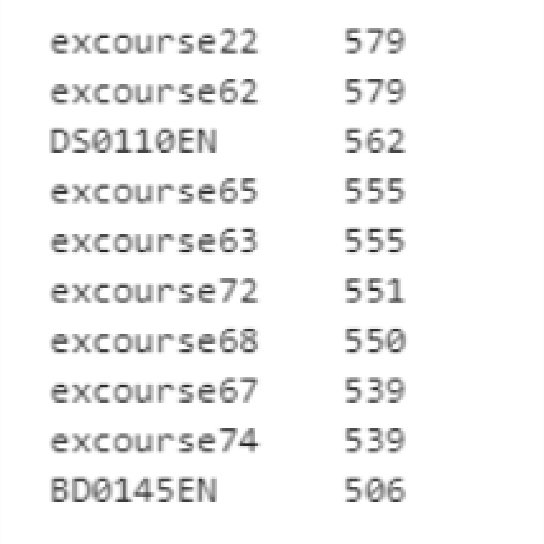
Course similarity based recommender system

with Threshold = 0.6

1. On average, how many new/unseen courses have been recommended per user (in the test user dataset) **11.37**
2. What are the most frequently recommended courses?

Return

the top10 commonly recommended courses across all users



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

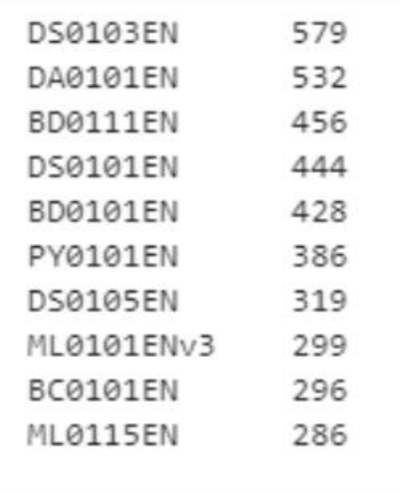
Clustering-based recommender system

with Number of clusters = 20

1. On average, how many new/unseen courses have been recommended per user (in the test user dataset) ***5.73***
2. What are the most frequently recommended courses?

Return

the top10 commonly recommended courses across all users



```
DS0103EN       579
DA0101EN       532
BD0111EN       456
DS0101EN       444
BD0101EN       428
PY0101EN       386
DS0105EN       319
ML0101ENv3     299
BC0101EN       296
ML0115EN       286
```

KNN based recommender system

Method to determine degree of similarity between two users

We use the Surprise Library to handle dataset and fit the data

Cosine Similarity Matrix :

RMSE 19%

Cosine_sim(u,v) $\dfrac{\sum_{i\,\epsilon I_{uv}} r_{ui}*r_{vi}}{\sqrt{\sum_{i\,\epsilon I_{uv}} r_{ui}^{2}}*\sqrt{\sum_{i\,\epsilon I_{uv}} r_{vi}^{2}}}$

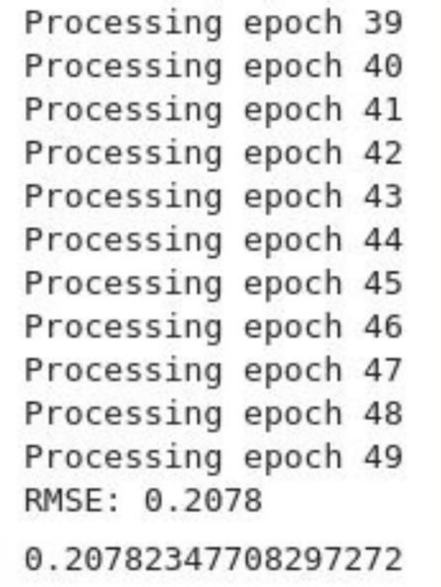
For items i,j: $\dfrac{\sum_{u\epsilon U_{ij}} r_{ui}*r_{uj}}{\sqrt{\sum_{u\epsilon U_{ij}} r_{ui}^{2}}*\sqrt{\sum_{u\,\epsilon U_{ij}} r_{uj}^{2}}}$

NMF based recommender system

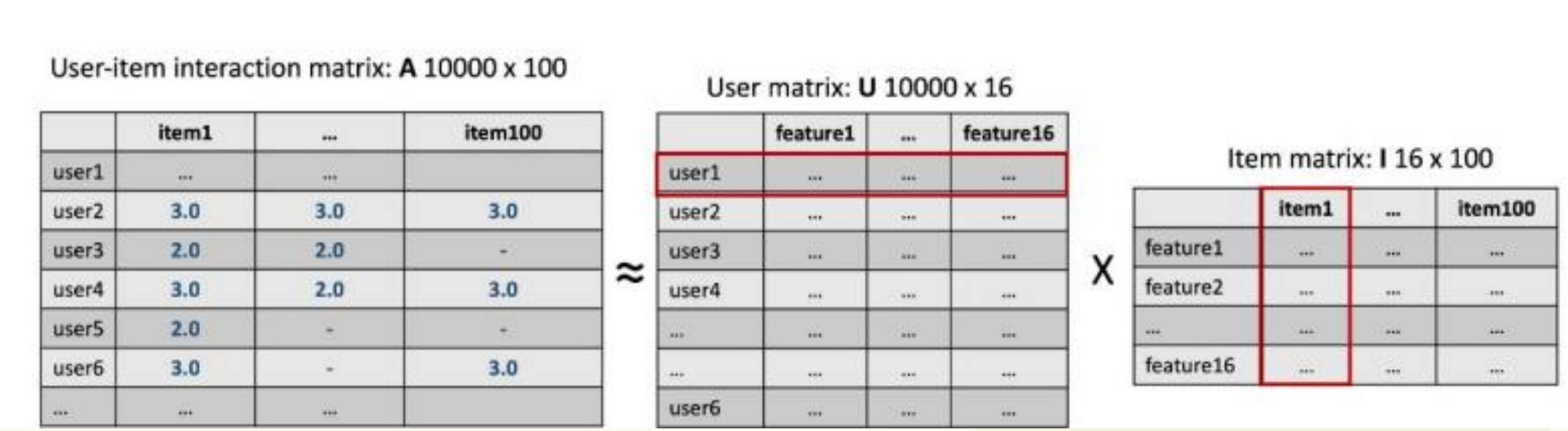
A dimensionality reduction algorithm called Non-negative matrix factorization (NMF), which decomposes a big sparse matrix into two smaller and dense matrices.

1. Use surprise library to decompose full matrix to two smaller and denser ones: user matrix and item matrix
2. Dot product each row in user matrix with each column in item matrix
3. Make prediction by test data, use RMSE metric to evaluate model performance

RMSE 20%



```
Processing epoch 39
Processing epoch 40
Processing epoch 41
Processing epoch 42
Processing epoch 43
Processing epoch 44
Processing epoch 45
Processing epoch 46
Processing epoch 47
Processing epoch 48
Processing epoch 49
RMSE: 0.2078
0.20782347708297272
```



User-item interaction matrix: **A** 10000 x 100

	item1	...	item100
user1	...	...	
user2	3.0	3.0	3.0
user3	2.0	2.0	-
user4	3.0	2.0	3.0
user5	2.0	-	-
user6	3.0	-	3.0
...	...	...	

≈

User matrix: **U** 10000 x 16

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

X

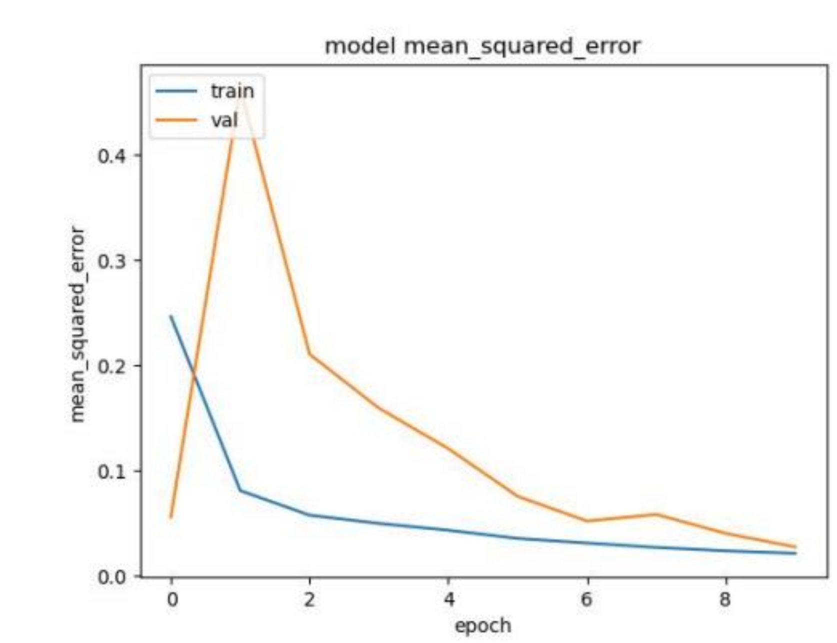
Item matrix: **I** 16 x 100

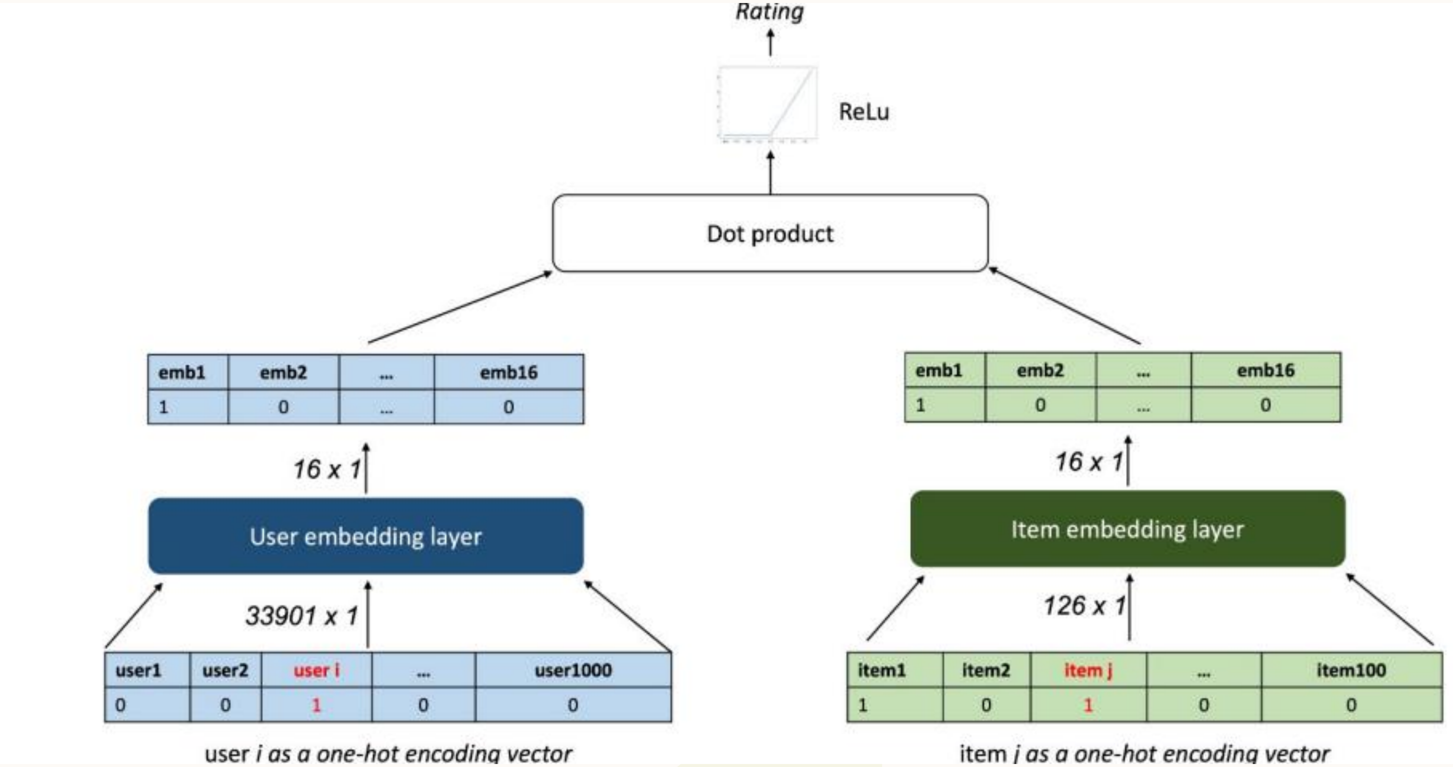
	item1	...	item100
feature1	...	...	...
feature2	...	...	...
...	...	...	...
feature16	...	...	...

Neural Network Embedding based recommender system

Model:
Optimizer: Adam
Loss: Mean Square Error

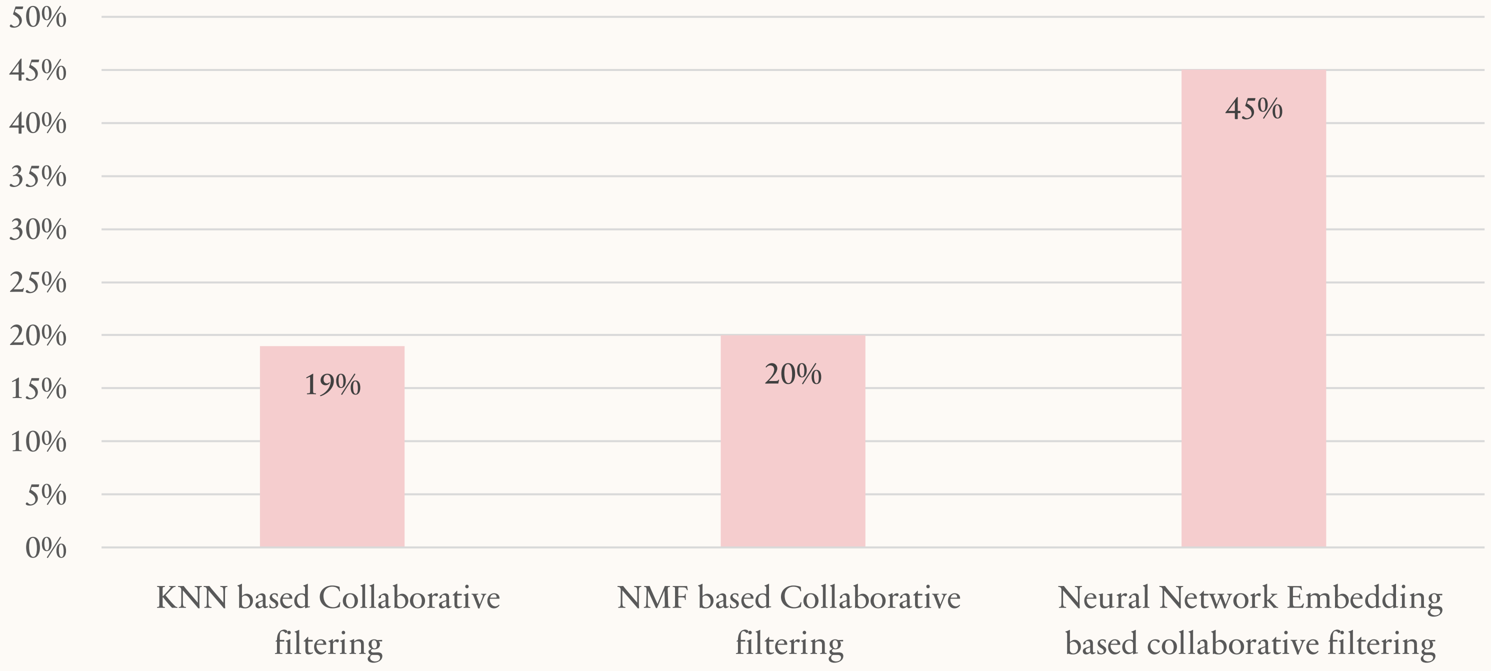
- Metric: Mean Square Error
- Epoch 12
- Batch size: 520





MSE : 25%
RSE: 45%

COMPARE THE PERFORMANCE OF COLLABORATIVE FILTERING MODELS



CONCLUSIONS

Neural Network model has the best accuracy .A model that is prone to overfitting so it needs more data to be sure of it reliability .

INNOVATIVE INSIGHTS

This project shows how a end-to-end machine learning pipeline work.

Although it passes all requirement , there are several enhancements that can be applied for better accuracy to avoid overfitting .

THANK YOU