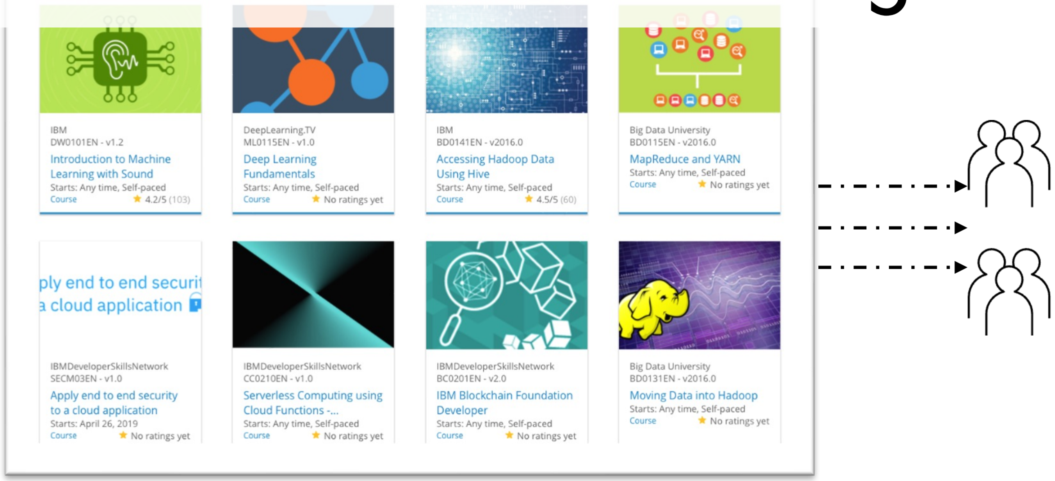
Build a Personalized Online Course Recommender System with Machine Learning

Faheemuddin Sayyed
18.12.2024



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

- **Project Overview:**
- The goal of this capstone project is to build a personalized course recommender system for an online learning platform. This system utilizes multiple machine learning techniques to deliver customized recommendations based on user preferences and course content.
- **Objectives:**

1. Understand user preferences and course attributes.
2. Implement different types of recommender systems:
   - Content-based filtering using user profiles and course genres.
   - Collaborative filtering (KNN, NMF, and neural networks).
   - Hybrid approaches for better recommendations.
3. Evaluate the performance of these models.

- **Why Personalized Recommendations Matter:**

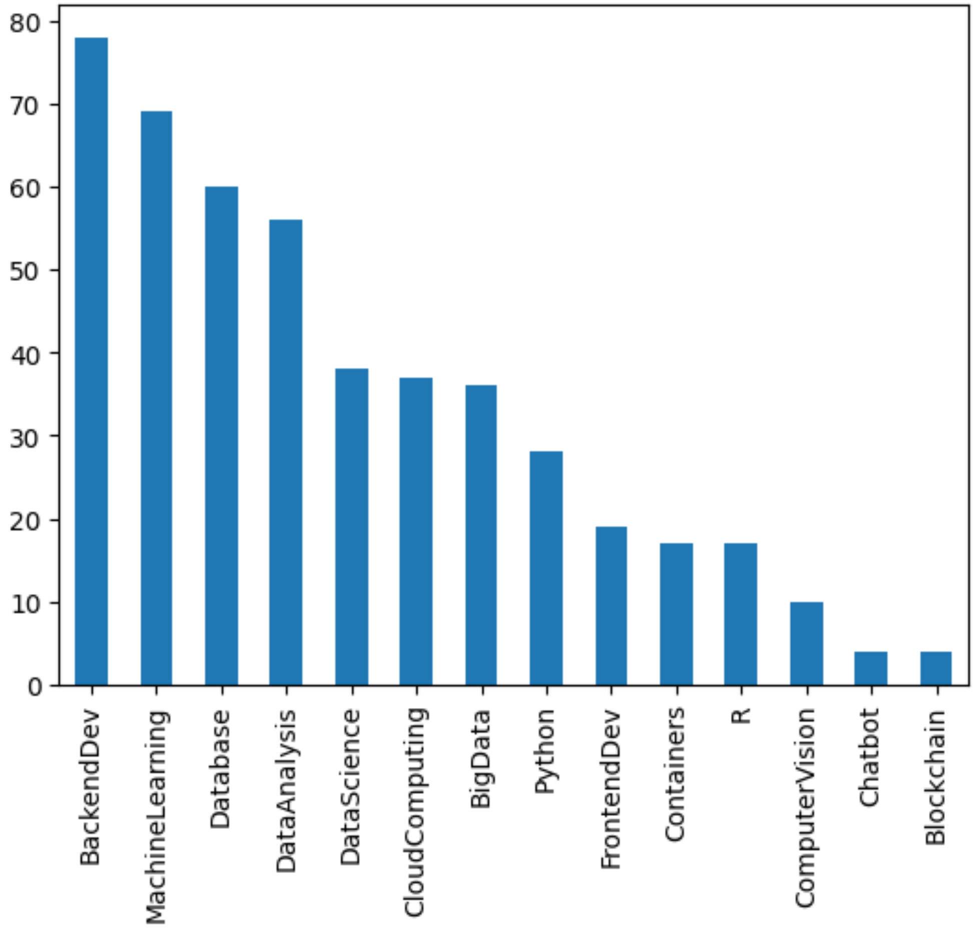
Personalized recommendations can:

- Improve user engagement and satisfaction.
- Help users discover relevant courses more efficiently.
- Increase course enrollments on the platform.

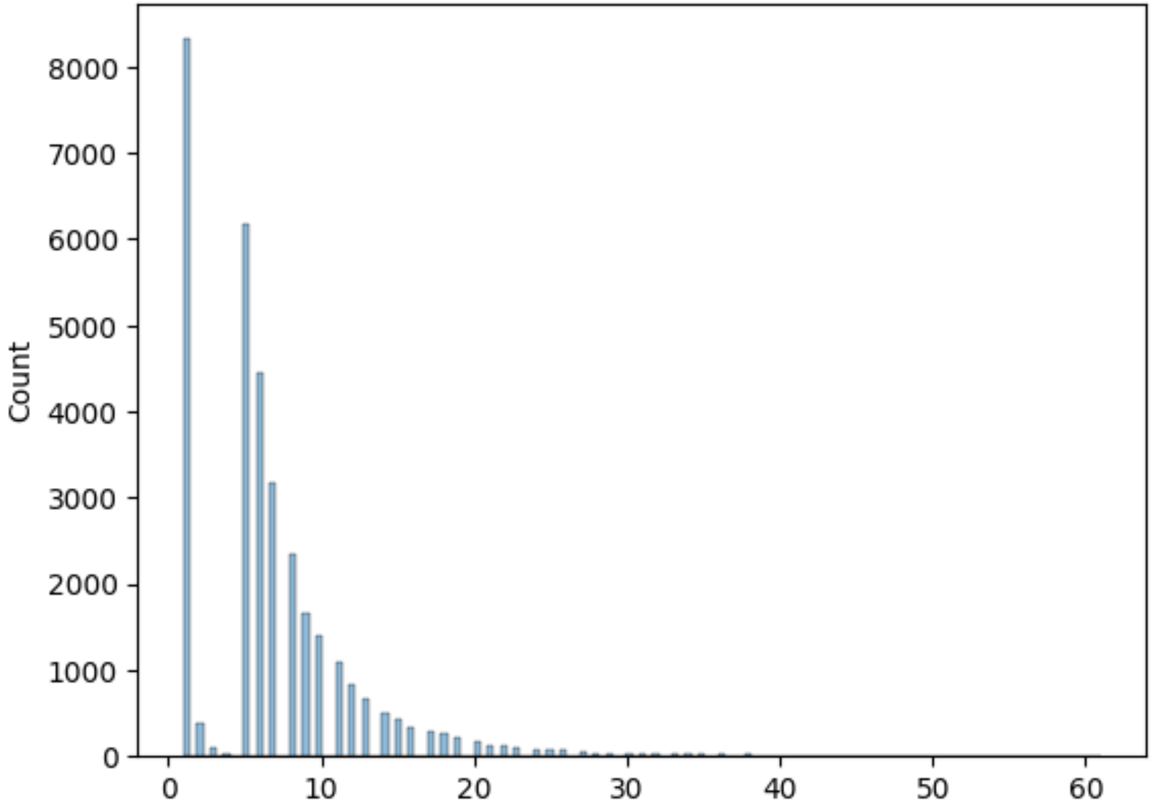
Exploratory Data Analysis

Course counts per genre

- The bar chart shows the distribution of courses across various genres. It highlights the most popular genres by their course counts, helping identify the platform's content focus.



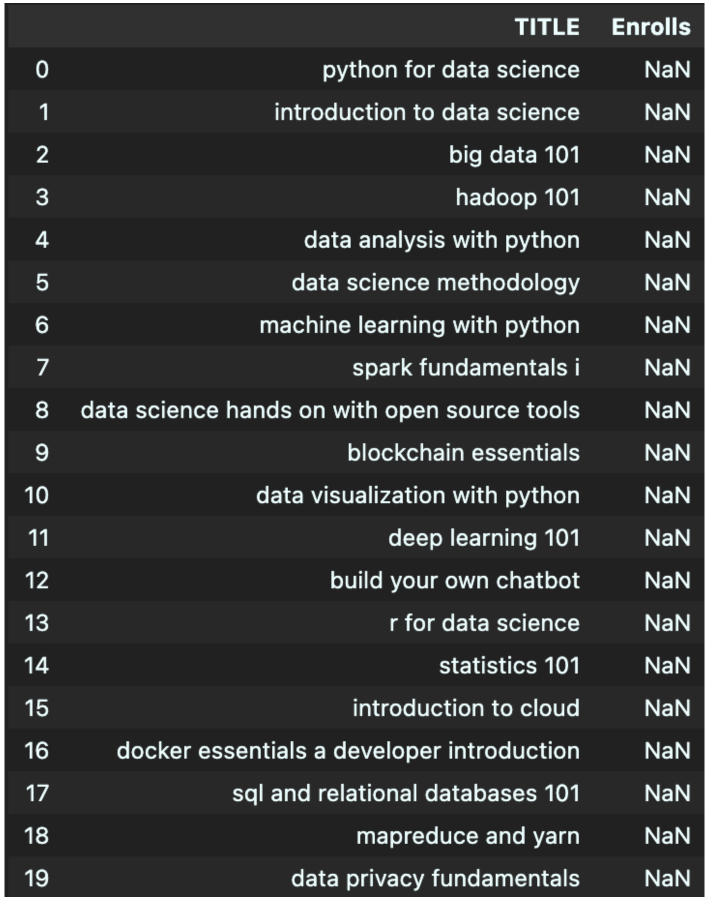
Course enrollment distribution



- The histogram illustrates user engagement, showing enrollment frequency across courses. It identifies trends like courses with the highest or lowest enrollments.

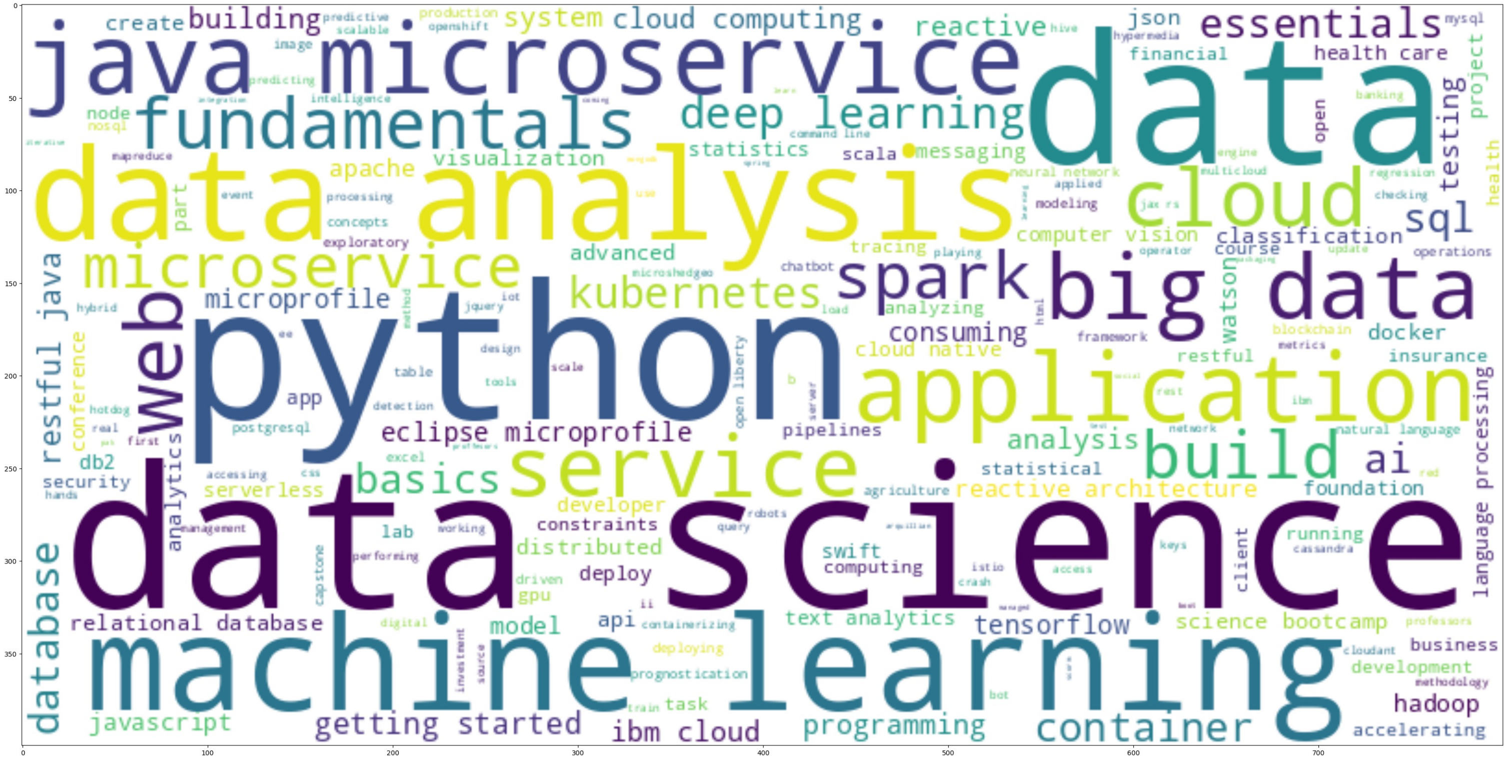
20 most popular courses

- This list reflects the platform's top-enrolled courses, giving insight into user preferences and trending topics.



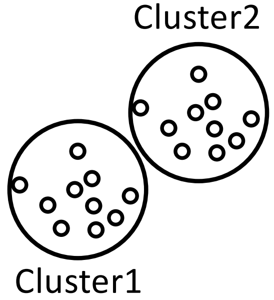
	TITLE	Enrolls
0	python for data science	NaN
1	introduction to data science	NaN
2	big data 101	NaN
3	hadoop 101	NaN
4	data analysis with python	NaN
5	data science methodology	NaN
6	machine learning with python	NaN
7	spark fundamentals i	NaN
8	data science hands on with open source tools	NaN
9	blockchain essentials	NaN
10	data visualization with python	NaN
11	deep learning 101	NaN
12	build your own chatbot	NaN
13	r for data science	NaN
14	statistics 101	NaN
15	introduction to cloud	NaN
16	docker essentials a developer introduction	NaN
17	sql and relational databases 101	NaN
18	mapreduce and yarn	NaN
19	data privacy fundamentals	NaN

Word cloud of course titles



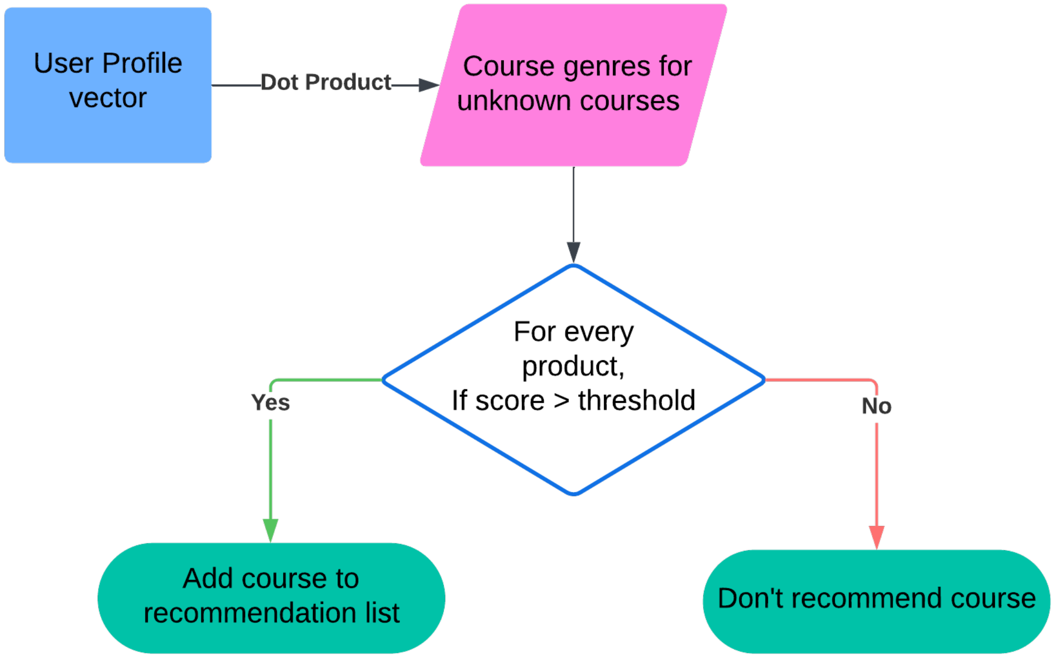
- The word cloud emphasizes the most common keywords in course titles, showcasing the primary themes and topics offered.

Content-based Recommender System using Unsupervised Learning



Flowchart of content-based recommender system using user profile and course genres

- Illustrates the process of constructing user vectors from their profiles and matching them with course genre vectors to recommend content.

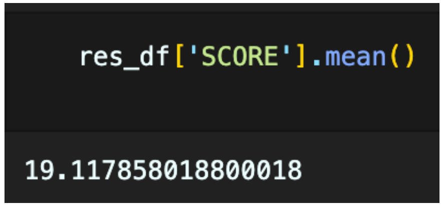

User Profile vector
Dot Product
Course genres for unknown courses
For every product, If score > threshold
Yes
No
Add course to recommendation list
Don't recommend course

Evaluation results of user profile-based recommender system

Hyperparameters tuned:

score_threshold = 10

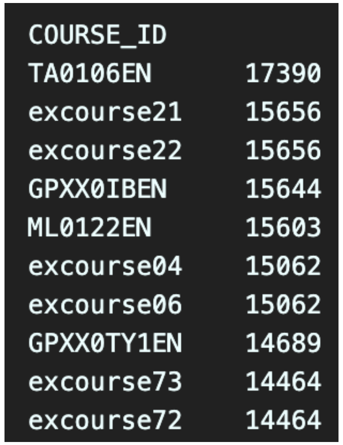
On average, how many new/unseen courses have been recommended per user (in the test user dataset)



```
res_df['SCORE'].mean()
```

```
19.117858018800018
```

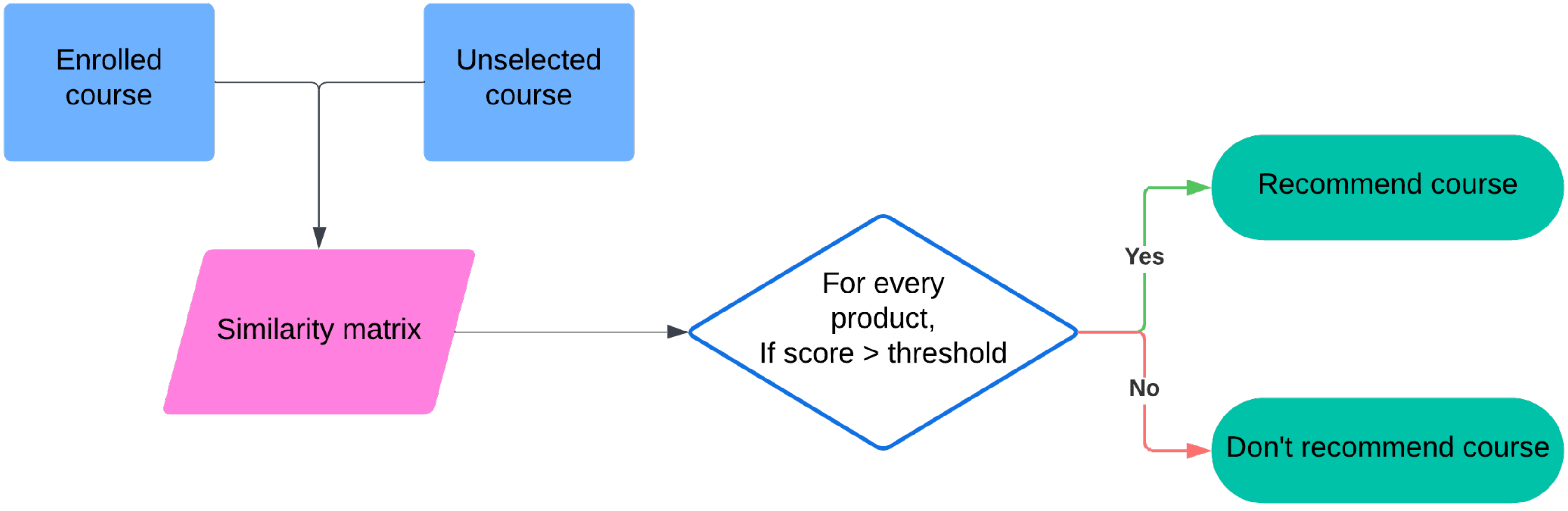
What are the most frequently recommended courses? Return the top-10 commonly recommended courses across all users



```
COURSE_ID
TA0106EN       17390
excourse21     15656
excourse22     15656
GPXX0IBEN      15644
ML0122EN       15603
excourse04     15062
excourse06     15062
GPXX0TY1EN     14689
excourse73     14464
excourse72     14464
```

Flowchart of content-based recommender system using course similarity

- Describes the methodology for calculating similarity between courses and generating recommendations for users based on their history.

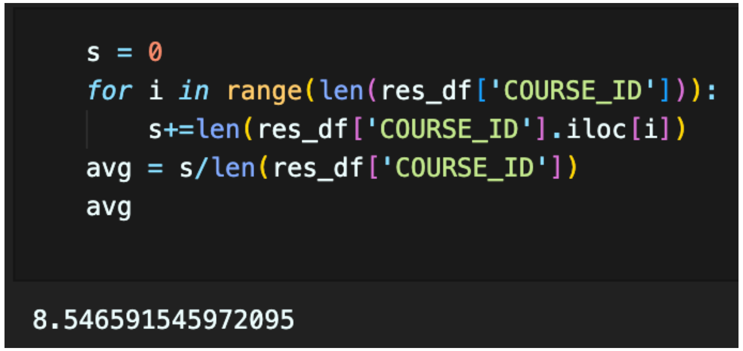


Evaluation results of course similarity based recommender system

Hyperparameters tuned:

threshold = 0.6

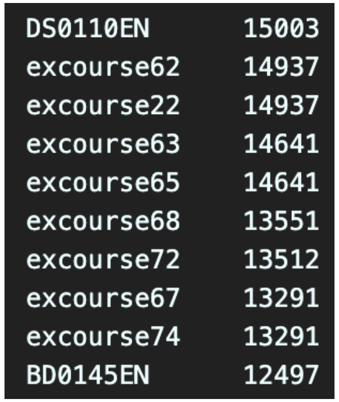
On average, how many new/unseen courses have been recommended per user (in the test user dataset)



```
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
```

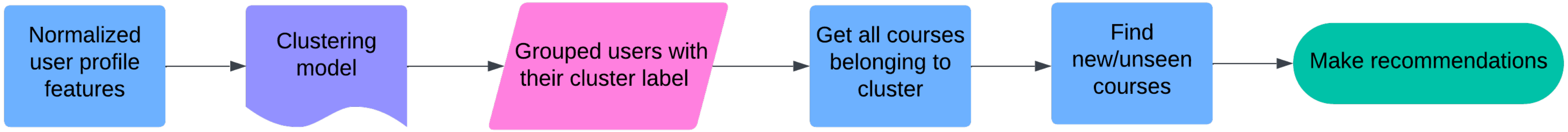
```
8.546591545972095
```

What are the most frequently recommended courses? Return the top-10 commonly recommended courses



```
DS0110EN       15003
excourse62     14937
excourse22     14937
excourse63     14641
excourse65     14641
excourse68     13551
excourse72     13512
excourse67     13291
excourse74     13291
BD0145EN       12497
```

Flowchart of clustering-based recommender system


Normalized user profile features
Clustering model
Grouped users with their cluster label
Get all courses belonging to cluster
Find new/unseen courses
Make recommendations

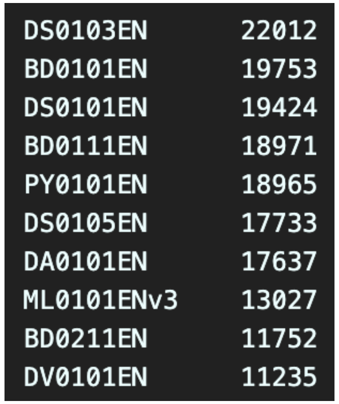
- Demonstrates clustering user profiles into segments to deliver group-based recommendations.

Evaluation results of clustering-based recommender system

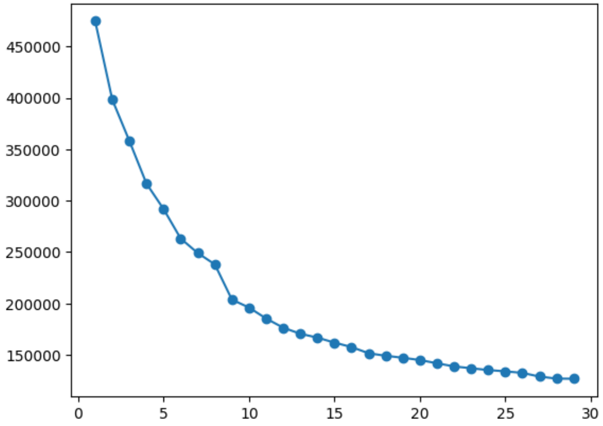
Hyperparameters tuned:

n_clusters = 20

What are the most frequently recommended courses? Return the top-10 commonly recommended courses

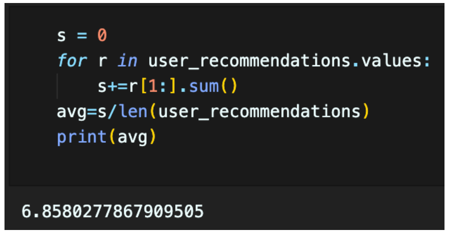


```
DS0103EN       22012
BD0101EN       19753
DS0101EN       19424
BD0111EN       18971
PY0101EN       18965
DS0105EN       17733
DA0101EN       17637
ML0101ENv3     13027
BD0211EN       11752
DV0101EN       11235
```



n_clusters vs sum of squared distances

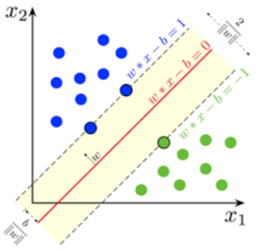
On average, how many new/unseen courses have been recommended per user (in the test user dataset)



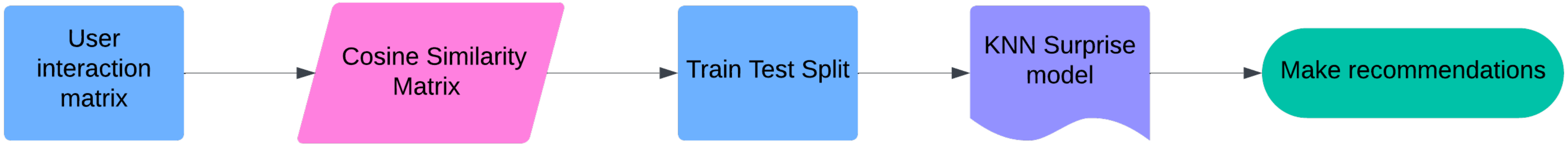
```
s = 0
for r in user_recommendations.values:
    s+=r[1:].sum()
avg=s/len(user_recommendations)
print(avg)
```

```
6.8580277867909505
```

Collaborative-filtering Recommender System using Supervised Learning

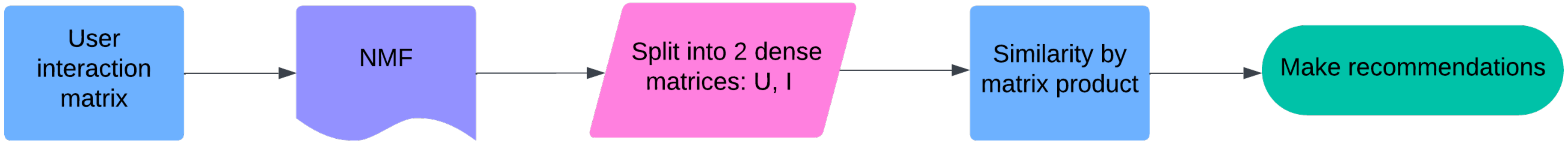


Flowchart of KNN based recommender system



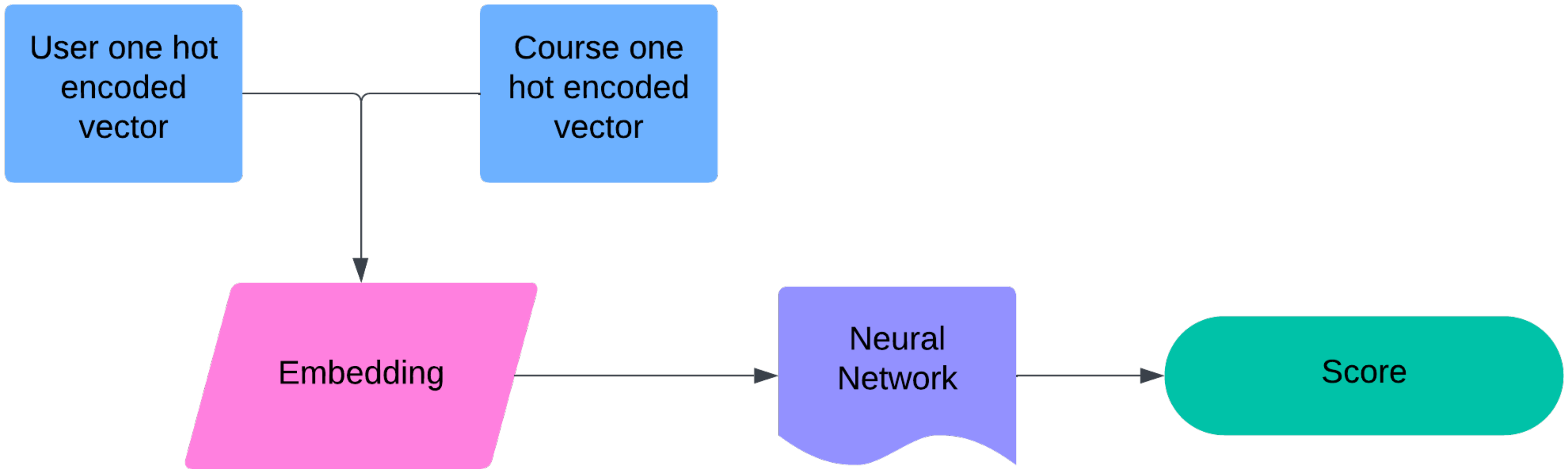
- Explains the implementation of KNN using historical user-course interactions to generate recommendations.

Flowchart of NMF based recommender system


User interaction matrix
NMF
Split into 2 dense matrices: U, I
Similarity by matrix product
Make recommendations

- Visualizes how NMF decomposes the user-course matrix to predict missing values and generate personalized suggestions.

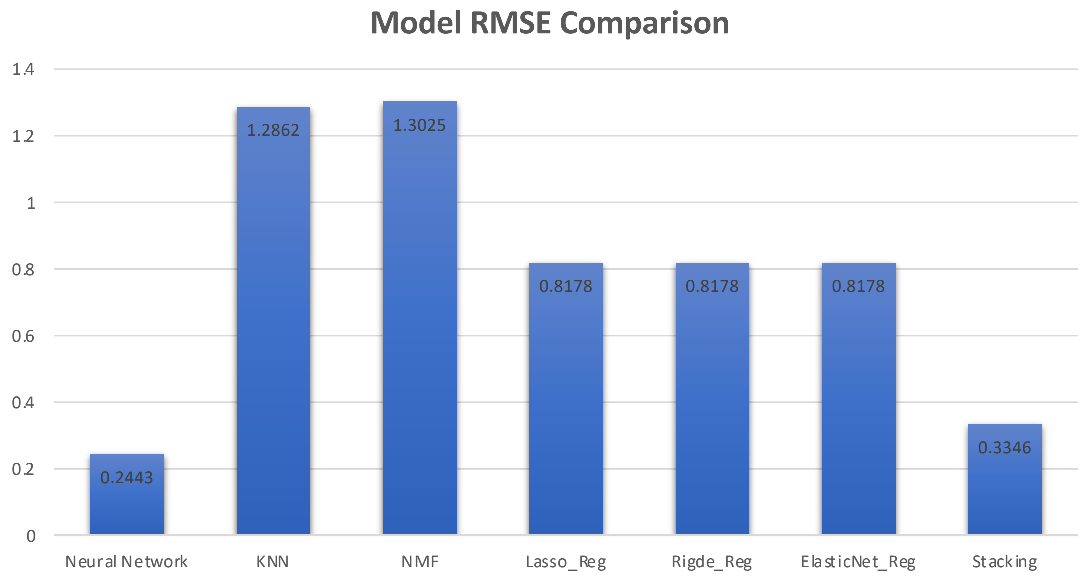
Flowchart of Neural Network Embedding based recommender system



- Outlines the process of embedding users and courses into a shared space for high-quality recommendations.

Compare the performance of collaborative-filtering models

- The bar chart compares the RMSE of various collaborative filtering models, with the neural network model achieving the best performance, followed by NMF and KNN.



Conclusions

- Summary:
  The project successfully demonstrated a personalized course recommendation system using multiple machine learning techniques, enhancing user satisfaction and engagement.
- Key Findings:
  - Content-based models efficiently utilized user preferences and course metadata.
  - Collaborative filtering methods, especially neural networks, outperformed in prediction accuracy.
  - Combining diverse approaches ensures robust recommendations.

Additional Insights

1. **Hybrid Recommendations:**
   Exploring combining content-based and collaborative filtering for improved accuracy and diversity in recommendations.

2. **Explainable Recommendations:**
   Implementing feature to show users why a course was recommended, increasing trust and engagement.

3. **Real-Time Recommendations:**
   Introducing models capable of updating recommendations dynamically based on user activity.

4. **Enhanced Evaluation Metrics:**
   Beyond RMSE, considering user engagement metrics like click-through rates and completion rates for holistic model assessment.

Appendix

- GitHub Repo Link:

https://github.com/Faheem219/IBM-Machine-Learning-Codes