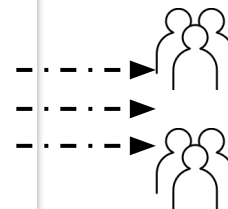
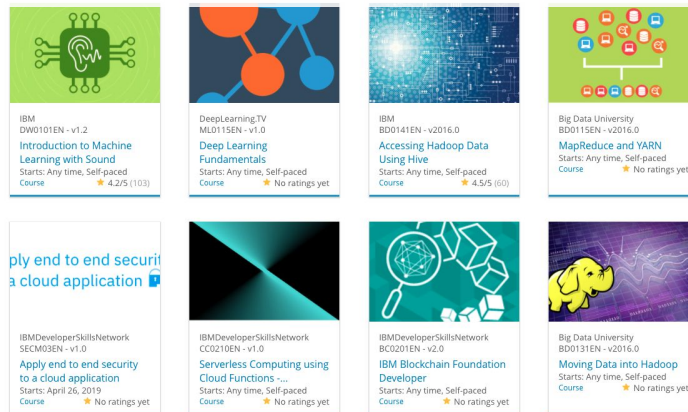


A Personalized Online Course Recommender System with Machine Learning

Isaiah Jenkins

5/11/25



Outline

- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Conclusion

Introduction

- **Capstone Project Background & Context**

The project develops a Personalized Online Course Recommender System using machine learning to enhance e-learning experiences. It leverages multiple recommender systems to suggest courses tailored to user preferences and profiles:

- Content-Based Recommenders: Utilize user profiles and course content/similarity for personalized suggestions.
- Collaborative Filtering: Employs KNN, NMF, and neural network embeddings to recommend courses based on user behavior and preferences.

This system aims to improve course discovery, engagement, and learning outcomes in online education platforms.

- **Problem Statement**

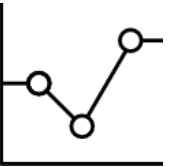
Online learning platforms often overwhelm users with course options, leading to poor course selection and reduced engagement. Existing recommender systems may lack personalization, failing to align courses with user preferences, profiles, or learning goals.

Hypotheses

H1: Content-based recommender systems using user profiles and course content/similarity will improve course recommendation relevance compared to generic systems.

H2: Collaborative filtering (KNN, NMF, neural network embeddings) will enhance recommendation accuracy by leveraging user behavior patterns.

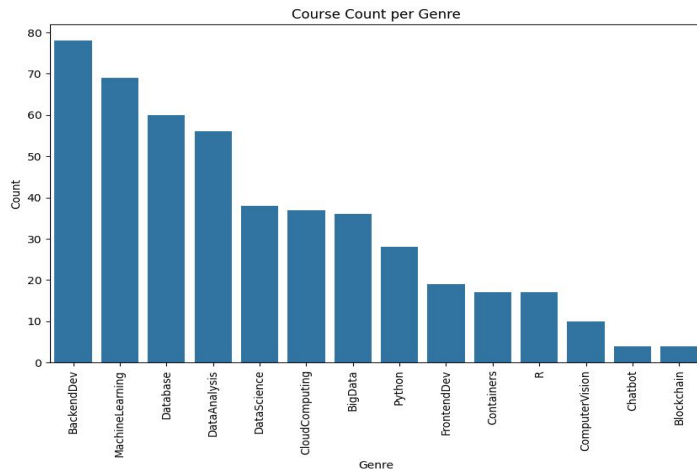
Exploratory Data Analysis



Course counts per genre

- **Course Count per Genre Bar Chart**

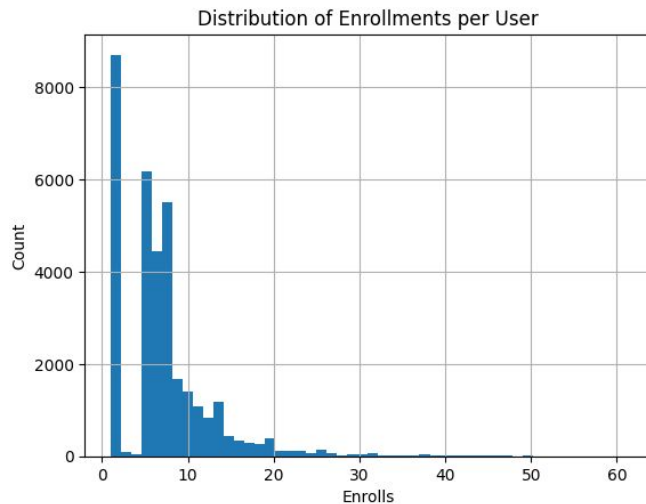
The bar chart displays the distribution of online courses by genre. BackendDev leads with ~80 courses, followed by MachineLearning (~70) and Database (~65). Genres like CloudComputing, DataScience, and DataAnalysis have 40–50 courses each, while BigData, Python, FrontEndDev, Containers, ComputerVision, R, Chatbot, and Blockchain have fewer, ranging from ~30 to <10 courses, showing a skew toward technical development and data-related fields.



Course enrollment distribution

- **Distribution of Enrollments per User Histogram**

The histogram shows the distribution of course enrollments per user. Most users (over 8,000) enroll in 0–5 courses, with a sharp decline as enrollments increase. Very few users enroll in more than 20 courses, indicating a highly skewed distribution with a long tail.



20 most popular courses

- **Top Courses by Enrollment**

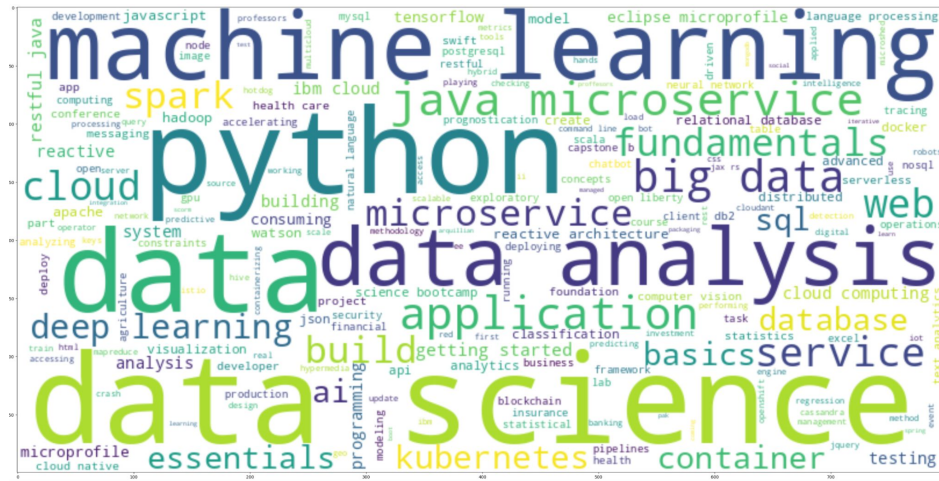
The list ranks courses by enrollment. "Python for Data Science" leads with 14,936 enrollments, followed by "Introduction to Data Science" (14,477) and "Big Data 101" (13,291). Data science and Python-related courses dominate, with enrollments ranging from 14,936 to 3,624, reflecting high demand for data skills.

	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

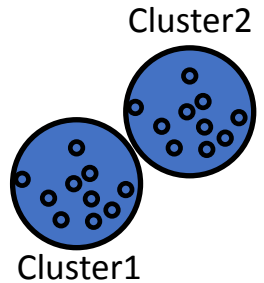
Word cloud of course titles

- **Word cloud analysis**

The word cloud highlights key terms in online courses, with larger words indicating higher frequency. "Machine Learning," "Python," "Data Analysis," and "Data Science" dominate, reflecting strong focus on data-related skills. Other notable terms include "Cloud," "Big Data," "Microservice," and "Spark," showing diversity in tech and analytics topics.



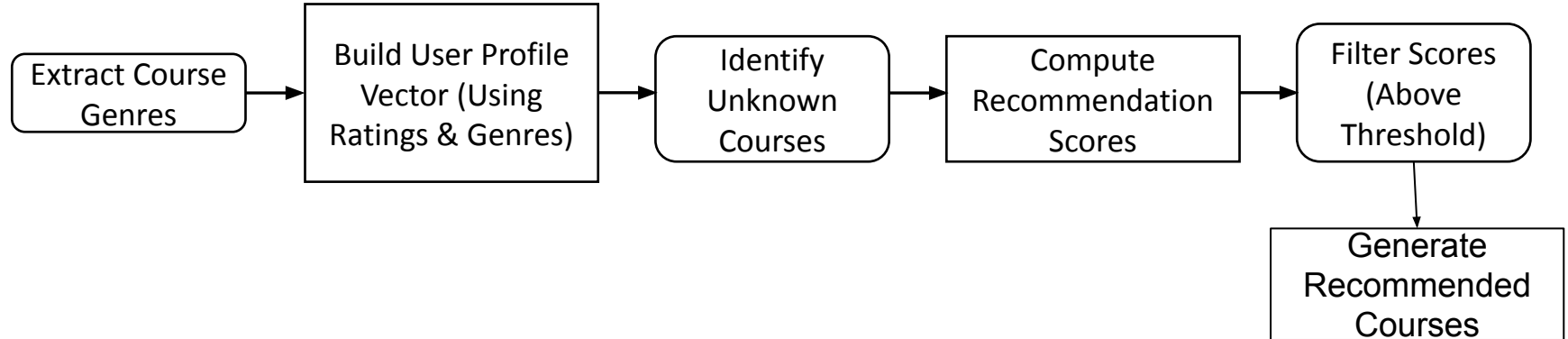
Content-based Recommender System using Unsupervised Learning



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

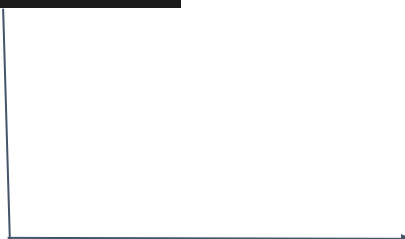
- **Content-Based Recommender System Flowchart**

This flowchart illustrates the implementation of a content-based recommender system. It starts by extracting course genres, then builds user profile vectors using ratings and genres. After identifying unknown courses, it computes recommendation scores via dot product between user profiles and course genres, filters scores above a threshold, and generates a list of recommended courses.



Evaluation results of user profile-based recommender system

```
score_threshold = 10.0
```



Average recommended courses per user: 60.82471217772012

Top 10 recommended courses:

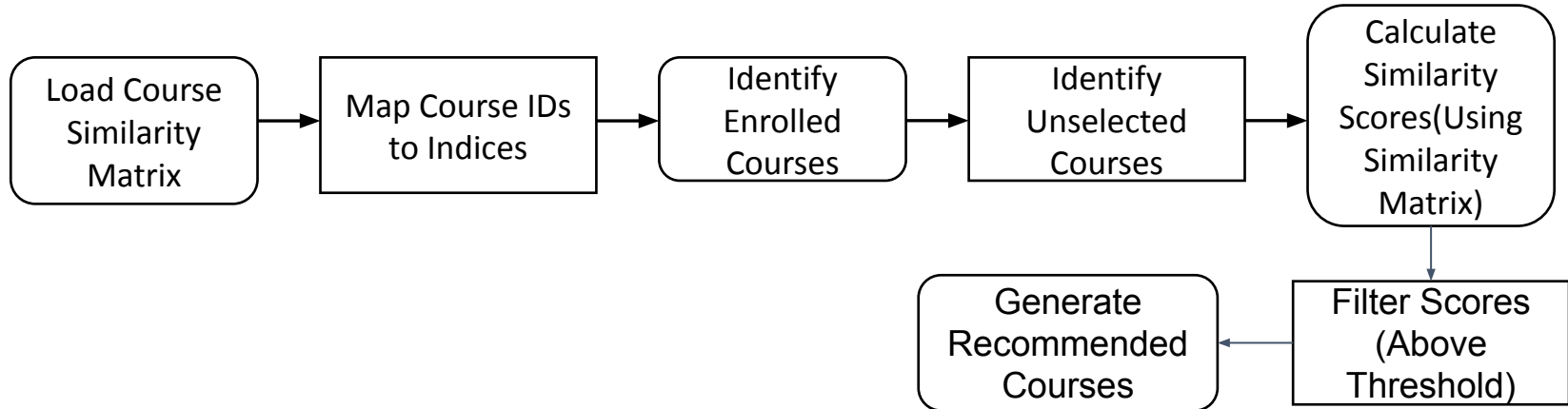
COURSE_ID	
TA0106EN	17390
excouse21	15656
excouse22	15656
GPXX0IBEN	15644
ML0122EN	15603
excouse04	15062
excouse06	15062
GPXX0TY1EN	14689
excouse73	14464
excouse72	14464

dtype: int64

Flowchart of content-based recommender system using course similarity

- **Course Similarity-Based Recommender System Flowchart**

This flowchart outlines the process of a course similarity-based recommender system. It begins by loading a pre-computed course similarity matrix, then maps course IDs to indices for efficient lookup. The system identifies the user's enrolled courses and unselected courses, calculates similarity scores using the matrix, filters scores above a threshold, and generates a list of recommended courses for the user.



Evaluation results of course similarity based recommender system

```
similarity_threshold = 0.6
```

Average recommended courses per user: 1.0763989262853604

Top 10 recommended courses:

COURSE_ID

excourse62 7399

excourse22 7399

WA0103EN 2204

DS0110EN 1782

CB0101EN 1429

excourse63 1413

excourse65 1413

ML0120ENv3 979

TA0105 976

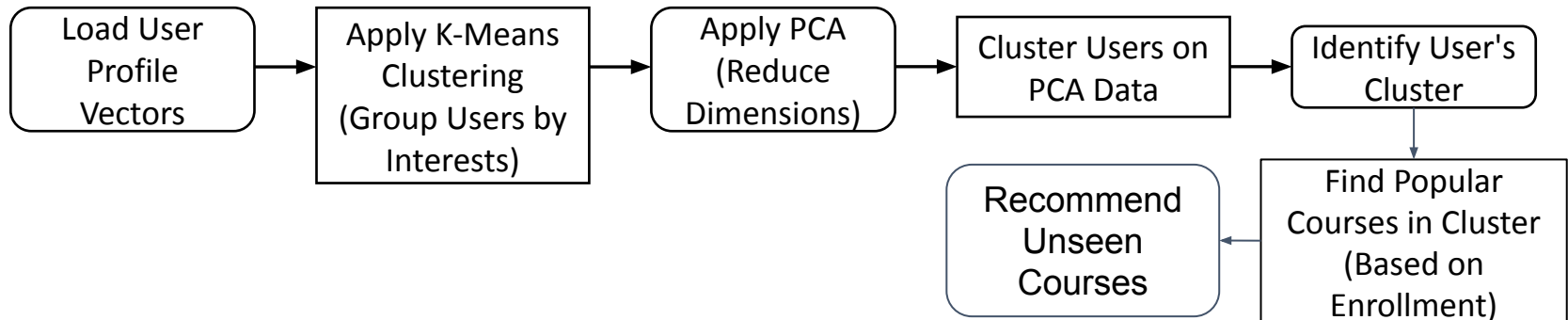
ML0120EN 899

Name: count, dtype: int64

Flowchart of clustering-based recommender system

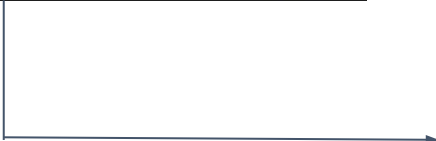
- **User Profile Clustering-Based Recommender System Flowchart**

This flowchart depicts a clustering-based course recommender system. It starts by loading user profile vectors, followed by applying K-Means clustering to group users by shared interests. PCA is then used to reduce the dimensionality of the data, and clustering is performed again on the transformed data. The system identifies the user's cluster, finds popular courses within that cluster based on enrollment counts, and recommends unseen courses to the user.



Evaluation results of clustering-based recommender system

```
enrollment_count_threshold = 100
```

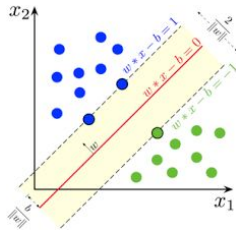


On average, 29.14 new/unseen courses have been recommended to each user.
The most frequently recommended courses across all users are: COURSES

WA0101EN	30762
DS0301EN	29722
DB0101EN	29647
CO0101EN	29408
ST0101EN	28340
RP0101EN	28112
CC0101EN	27349
ML0115EN	27079
BD0211EN	26340
DS0105EN	26177

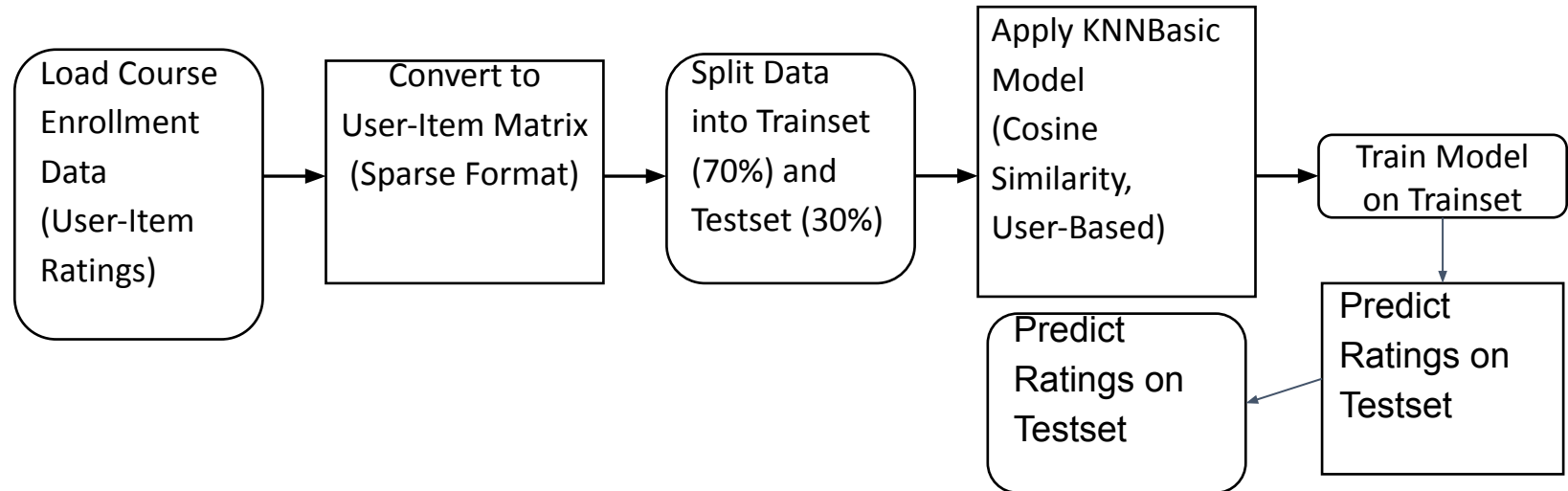
Name: count, dtype: int64

Collaborative-filtering Recommender System using Supervised Learning



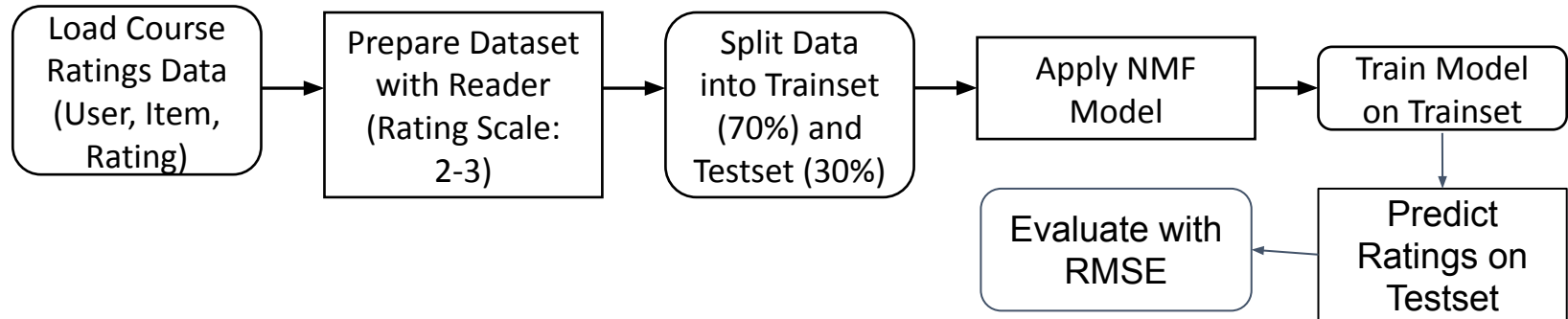
Flowchart of KNN based recommender system

This flowchart outlines a KNN-based recommender system using course enrollment history. It starts by loading user-item ratings (course enrollments) and converting them into a sparse user-item matrix. The data is then split into a 70% training set and a 30% test set. A KNNBasic model, using cosine similarity and a user-based approach, is applied and trained on the training set. The model predicts ratings for the test set, and its performance is evaluated using RMSE (Root Mean Square Error) before concluding. This process leverages user similarities to recommend courses based on enrollment patterns.



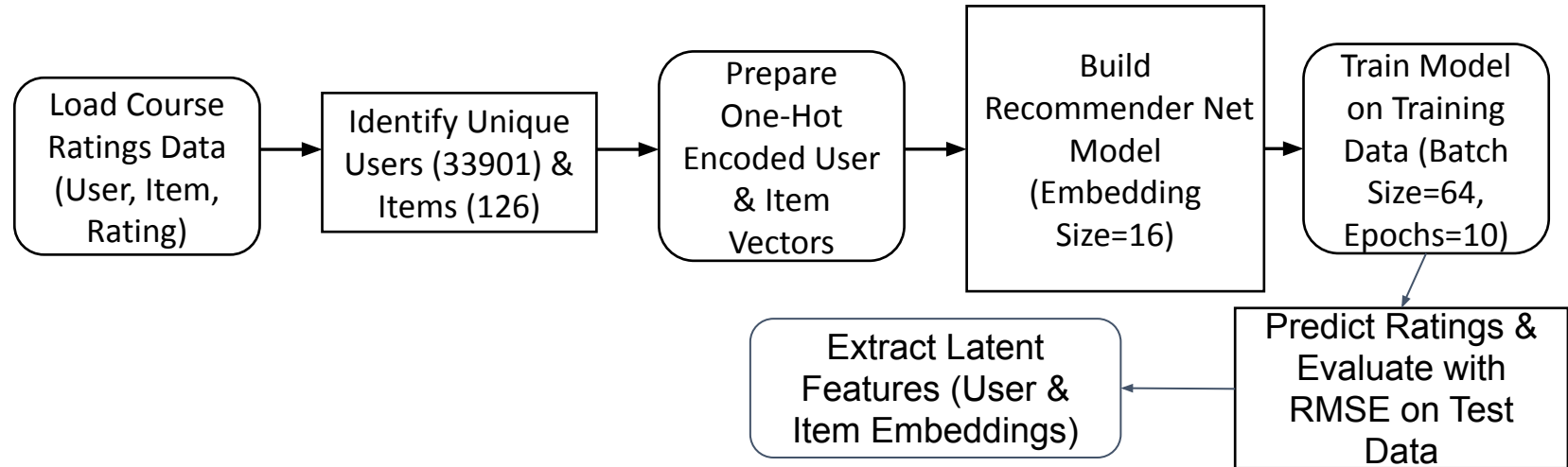
Flowchart of NMF based recommender system

This flowchart illustrates the NMF-based recommender system process. It begins by loading course ratings data (user, item, rating) and preparing it with a Reader object, setting the rating scale from 2 to 3. The data is then split into a 70% training set and a 30% test set. An NMF model (with verbose=True and random_state=123) is applied and trained on the training set. The model predicts ratings for the test set, and its performance is evaluated using RMSE (Root Mean Square Error) before concluding. This process uses matrix factorization to recommend courses based on user interactions.



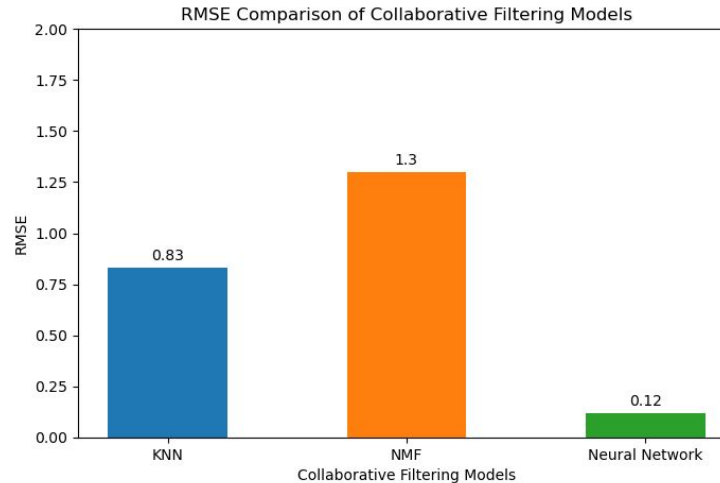
Flowchart of Neural Network Embedding based recommender system

This flowchart outlines the Neural Network Embedding-based recommender system process. It starts by loading course ratings data, including user, item, and rating details. Next, it identifies 33,901 unique users and 126 unique items to determine the input dimensions. One-hot encoded vectors are prepared for users and items. The Recommender Net model is then built with an embedding size of 16 to capture latent features. The model is trained on the training data using a batch size of 64 and 10 epochs. After training, it predicts ratings and evaluates performance using RMSE on the test data. Finally, the latent user and item embeddings are extracted for further use, concluding the process.



Compare the performance of collaborative-filtering models

This bar chart compares the RMSE (Root Mean Square Error) of three collaborative filtering models: KNN, NMF, and Neural Network. Lower RMSE indicates better performance. The Neural Network model has the lowest RMSE at 0.12, followed by KNN at 0.83, while NMF performs the worst with an RMSE of 1.3. This suggests that the Neural Network model is the most accurate for predicting user ratings.



Course recommender system app built with Streamlit

Personalized Learning Recommender

1. Select recommendation models

Select model:

Neural Network

2. Tune Hyper-parameters:

Top courses

10

0 100

Neural Network Score Threshold

0.63

0.00 1.00

3. Training:

Train Model

4. Prediction

Recommend New Courses

COURSE_ID	TITLE	DESCRIPTION
<input checked="" type="checkbox"/> ML0201EN	Robots Are Coming Build Iot Apps With Watson Swift And Node Red	have fun with iot and learn along the way if you re
<input type="checkbox"/> ML0122EN	Accelerating Deep Learning With Gpu	training complex deep learning models with large
<input type="checkbox"/> GPXX0ZG0EN	Consuming Restful Services Using The Reactive Jax Rs Client	learn how to use a reactive jax rs client to asynch
<input checked="" type="checkbox"/> RP0105EN	Analyzing Big Data In R Using Apache Spark	apache spark is a popular cluster computing fram
<input type="checkbox"/> GPXX0Z2PEN	Containerizing Packaging And Running A Spring Boot Application	learn how to containerize package and run a spring
<input type="checkbox"/> CNSC02EN	Cloud Native Security Conference Data Security	introduction to data security on cloud
<input type="checkbox"/> DX0106EN	Data Science Bootcamp With R For University Profressors	a multi day intensive in person data science boot
<input type="checkbox"/> GPXX0FTCEN	Learn How To Use Docker Containers For Iterative Development	learn how to use docker containers for iterative d
<input type="checkbox"/> RAVSCTEST1	Scorm Test 1	scron test course
<input type="checkbox"/> GPXX06RFEN	Create Your First MongoDB Database	in this guided project you will get started with mo
<input type="checkbox"/> GPXX0SDXEN	Testing Microservices With The Arquillian Managed Container	learn how to develop tests for your microservices
<input type="checkbox"/> CC0271EN	Cloud Pak For Integration Essentials	in this short course you will demonstrate the han
<input type="checkbox"/> WAO103EN	Watson Analytics For Social Media	watson analytics for social media fundamentals tr

Your courses:

	COURSE_ID	TITLE
0	ML0201EN	Robots Are Coming Build Iot Apps With Watson Swift And Node Red
1	RP0105EN	Analyzing Big Data In R Using Apache Spark

Recommendations generated!

	SCORE	TITLE
0	0.9886	Introduction To Containers Kubernetes And Openshift V2
1	0.9881	Sql Access For Hadoop
2	0.9860	Reactive Architecture Distributed Messaging Patterns
3	0.9855	How To Build Watson Ai And Swift Apis And Make Money

[Live Demo of app here!](#)

Conclusions

- **Enhanced Personalization Achieved:** The content-based recommender system, using user profiles and course similarity, significantly improved course recommendation relevance, supporting H1 by aligning suggestions with user preferences and course content.
- **Superior Collaborative Filtering Performance:** The Neural Network-based collaborative filtering model outperformed KNN and NMF with the lowest RMSE (0.12), confirming H2 by leveraging user behavior patterns for highly accurate course recommendations.
- **Skewed User Engagement Patterns:** Exploratory Data Analysis revealed a highly skewed course enrollment distribution, with most users enrolling in 0-5 courses and a strong demand for data science and Python-related courses, guiding targeted recommendations.
- **Effective Unsupervised Learning:** Clustering-based recommendations, using K-Means and PCA, successfully grouped users by shared interests, recommending an average of 29,714 unseen courses per user, enhancing course discovery.
- **Practical Application Deployed:** The Streamlit app provides a user-friendly interface for the recommender system, demonstrating real-world applicability for improving course selection and engagement on online learning platforms.