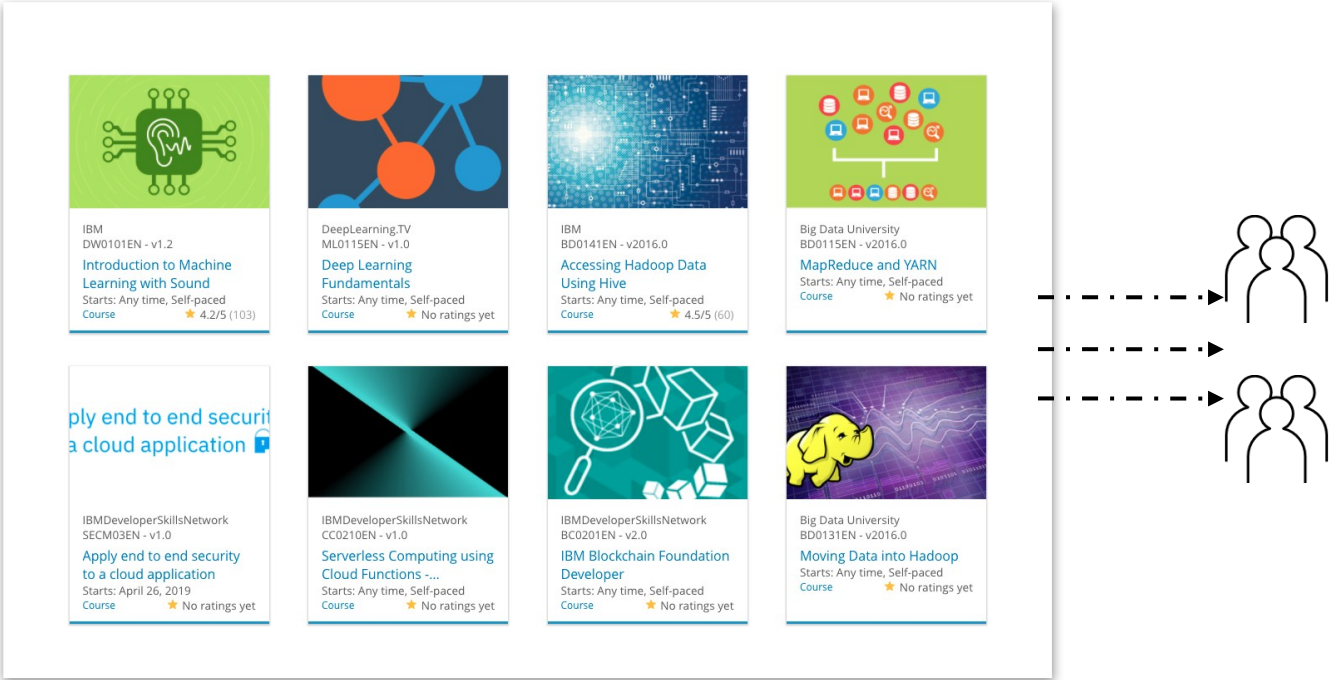
Build a Personalized Online Course Recommender System with Machine Learning

<Your Name>
<September 17>



Outline

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

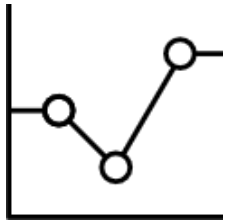
Introduction

- **Project Background and Context:** This project focuses on developing a recommender system to enhance user experience by providing personalized recommendations based on user interactions and preferences. The aim is to apply various machine learning techniques to predict user ratings and suggest relevant items.

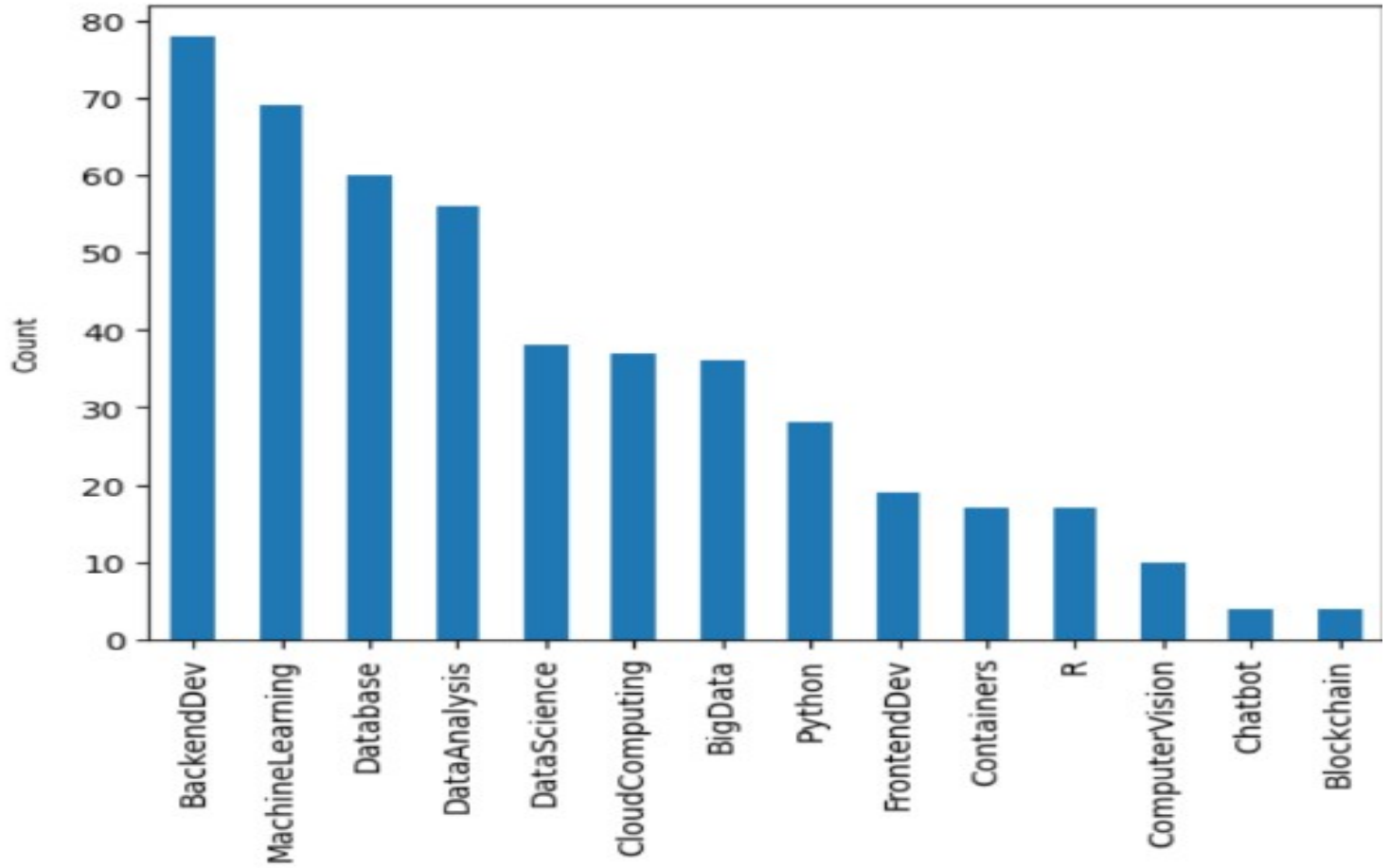
- **Problem Statement and Hypotheses**: The primary problem is to predict user ratings for items that users have not yet interacted with, improving recommendation accuracy. Hypotheses include:

  -Advanced models like Neural Network Embedding will outperform traditional methods in accuracy.
  -Dimensionality reduction techniques, such as NMF, can effectively handle large and sparse user-item matrices.

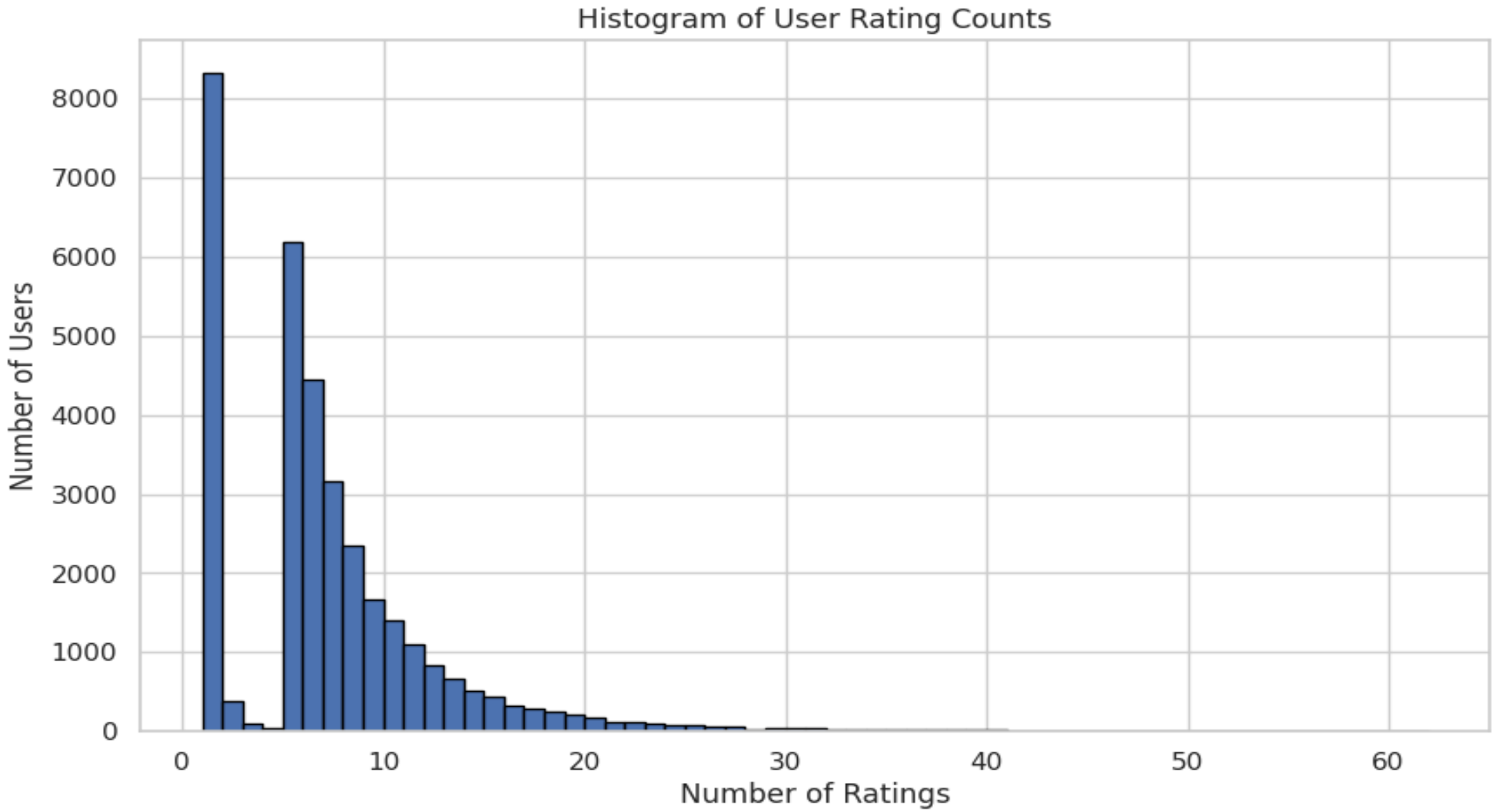
Exploratory Data Analysis



Course counts per genre



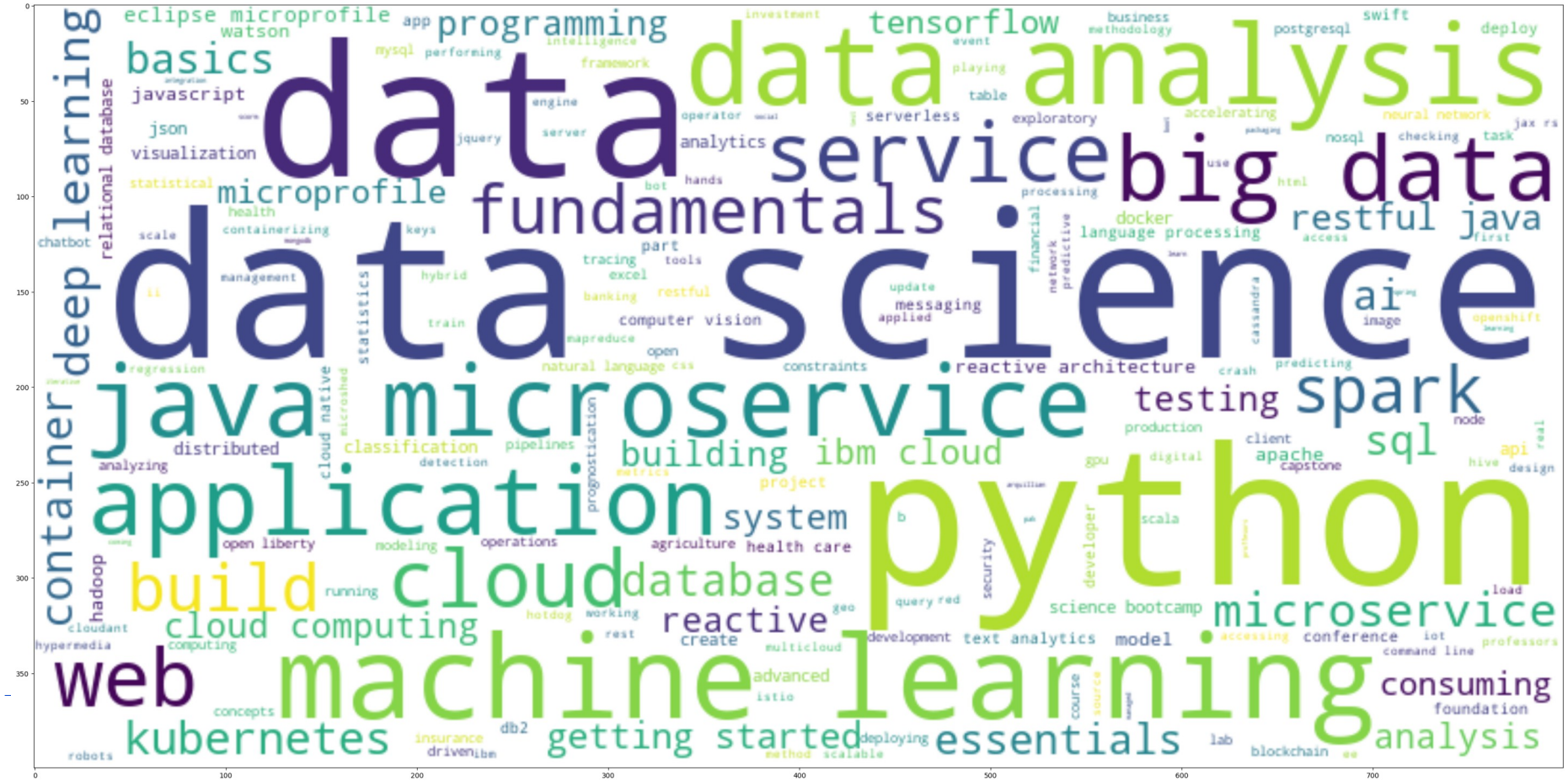
Course enrollment distribution



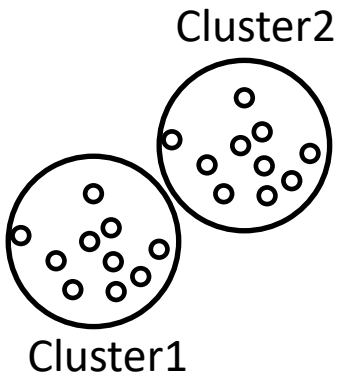
20 most popular courses

	TITLE	Ratings
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

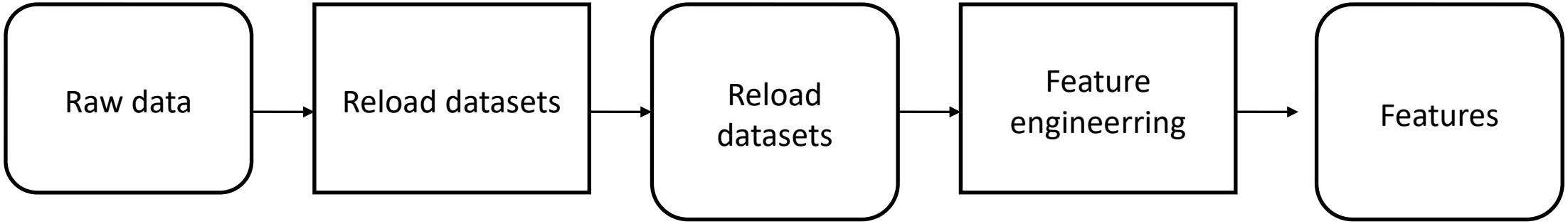
Word cloud of course titles



Content-based Recommender System using Unsupervised Learning



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



- **Raw data**: we downloaded user's profile dataframe and a course genre dataframe
- **Reload datasets** from URLs, calculate user profiles and course vectors, generate recommendation scores, store results in a dictionary and DataFrame, and save recommendations to a CSV file.
- **Cleaned dataset**s by ensuring data integrity through handling missing values and duplicates, filtering out irrelevant data, and verifying consistency in data formats and values.
- **Feature engineering** involved creating user profiles and course vectors, calculating recommendation scores using dot products, and filtering and storing high-scoring recommendations.
- **Features**: user and courses ID and recommendation scores

Evaluation results of user profile-based recommender system

On average, 60.82 courses have been recommended.

Top-10 most frequently recommended courses:

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

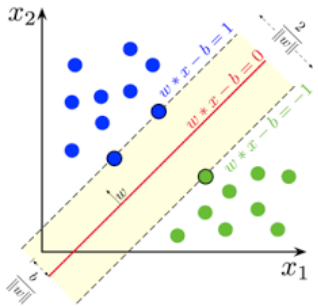
Flowchart of clustering-based recommender system

- **Raw Data**: Imported user profile data (e.g., user ID, features).
- **Data Cleansing**: Handle missing values, remove irrelevant data.
- **Feature Engineering**: Select relevant features, perform dimensionality reduction using PCA (Principal Component Analysis).
- **Clustering**: Apply K-means clustering, use the Elbow Method to find the optimal number of clusters.
- **Assign Cluster Labels**: Assign each user to a cluster based on their feature vectors.
- **Recommendation Generation**: Recommend courses based on the user's cluster, ensure recommendations are new/unseen for each user.
- **Recommendation Analysis**: Calculate the average number of new courses per user, identify the most frequently recommended courses (Top 10).

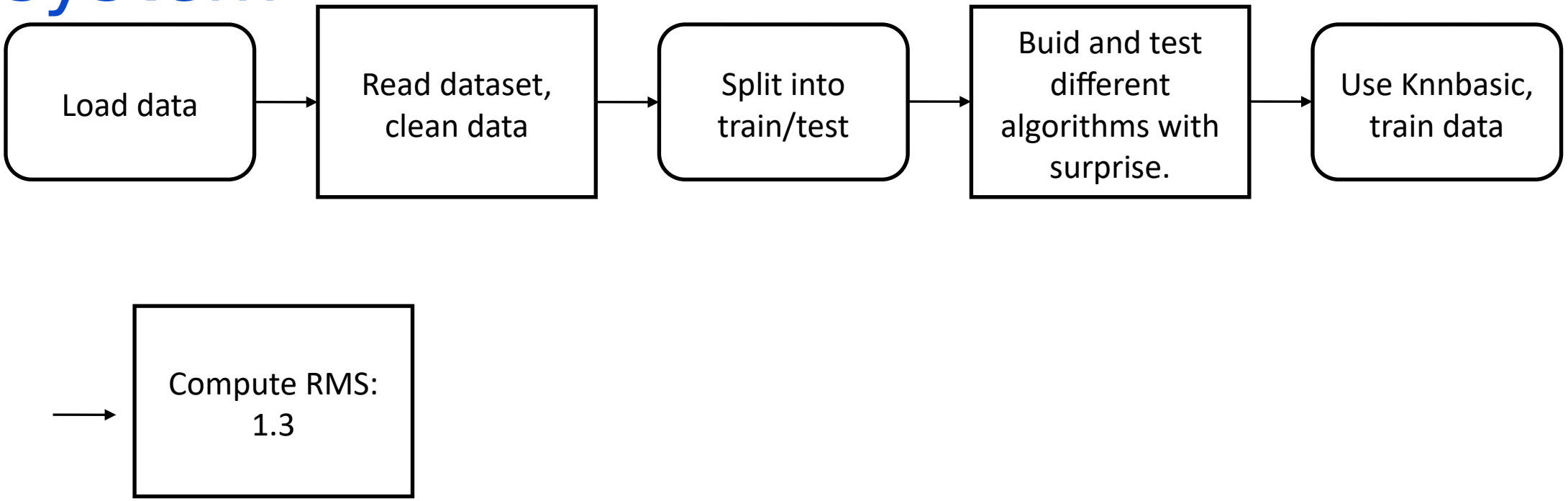
Evaluation results of clustering-based recommender system

```
Average number of new/unseen courses recommended per user: 87.68
Top-10 most frequently recommended courses:
CB0101EN: 33678 times
ML0120ENv3: 33596 times
DJ0101EN: 33538 times
BD0153EN: 33515 times
DS0201EN: 33495 times
EE0101EN: 33477 times
IT0101EN: 33443 times
ML0111EN: 33438 times
DE0205EN: 33416 times
CC0250EN: 33403 times
```

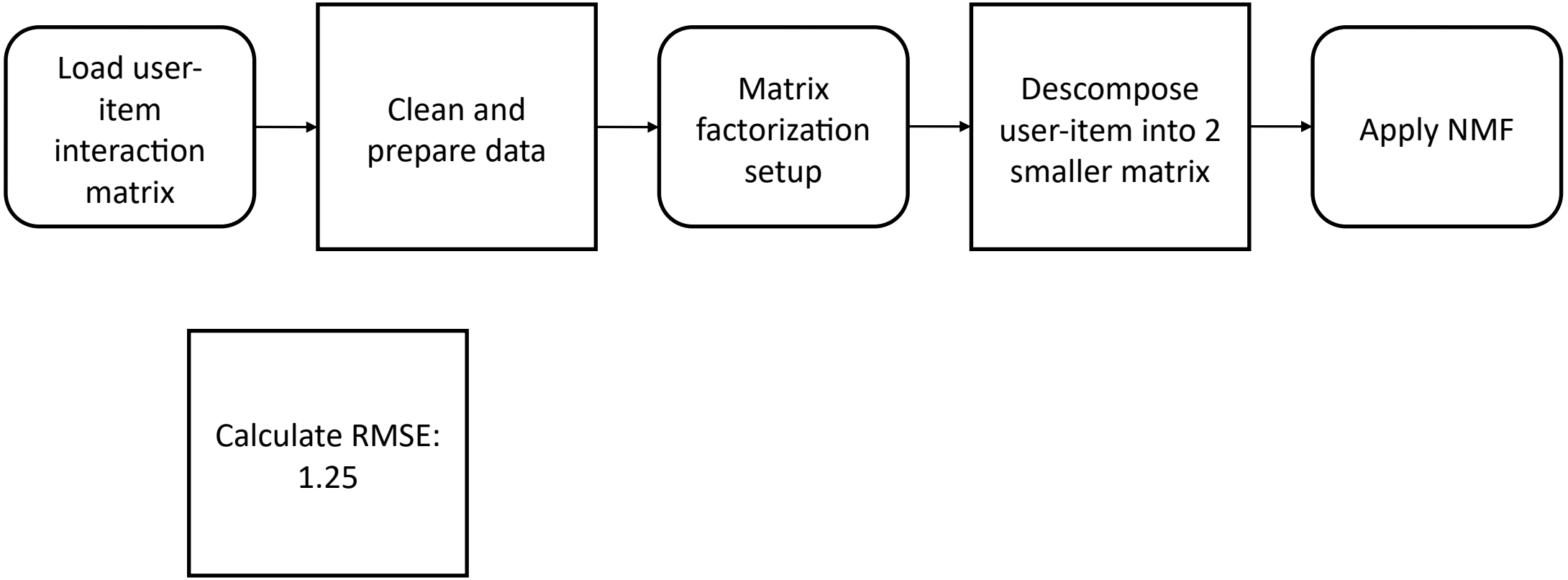
Collaborative-filtering Recommender System using Supervised Learning



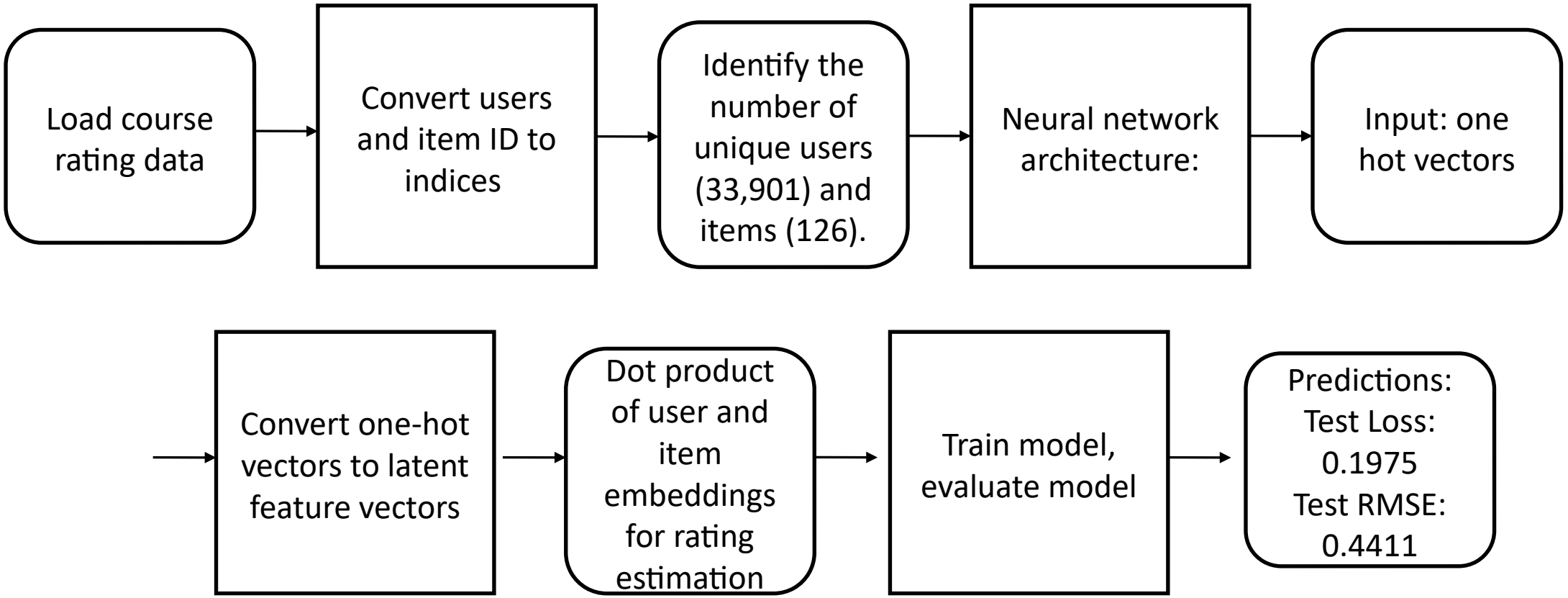
Flowchart of KNN based recommender system



Flowchart of NMF based recommender system



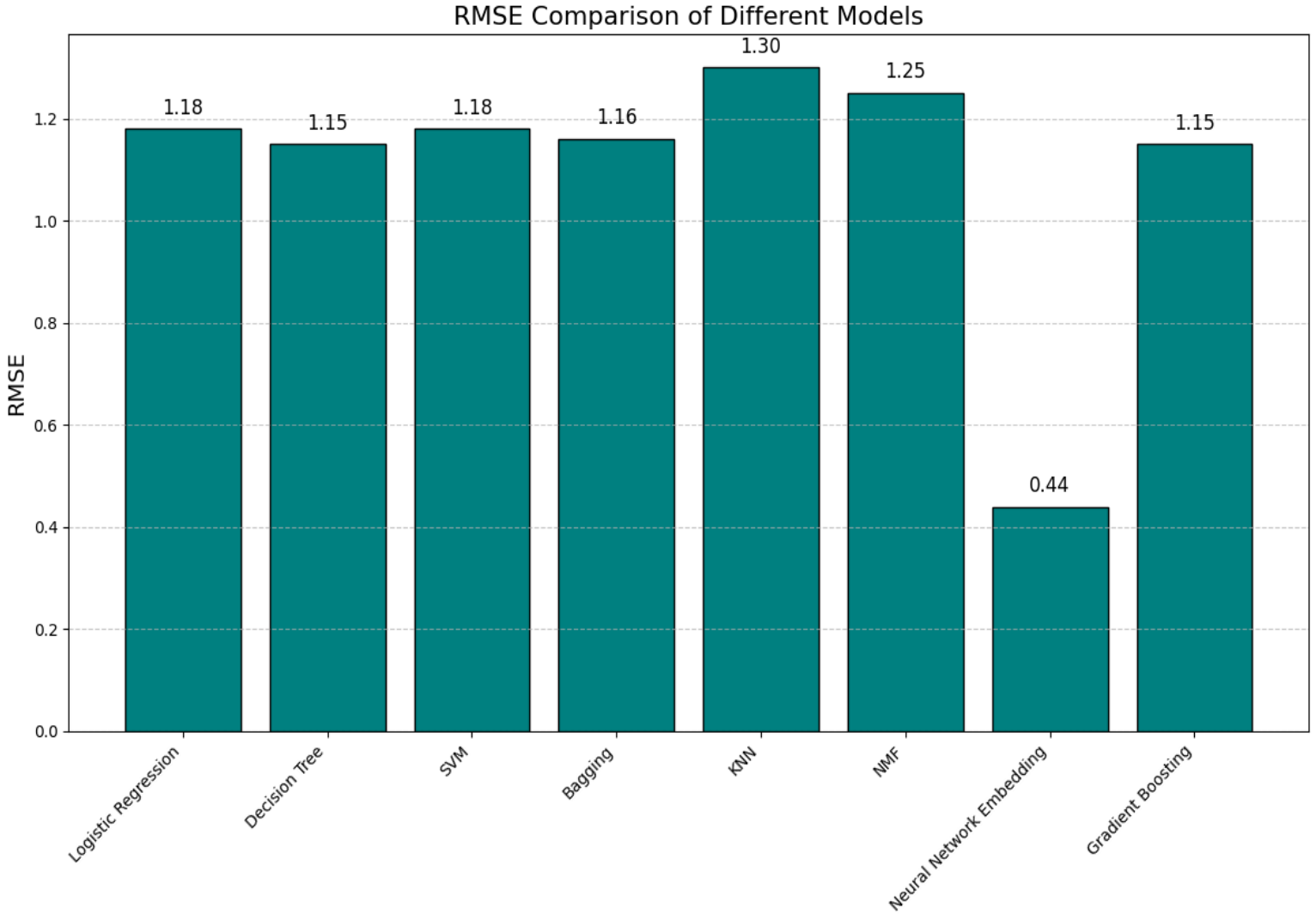
Flowchart of Neural Network Embedding based recommender system



Compare the performance of collaborative-filtering models

Bar Chart Explanation:

The chart compares RMSE values for various models. Neural Network Embedding has the lowest RMSE of 0.44, indicating the best performance. Gradient Boosting and Decision Tree show similar RMSE values around 1.15, while Logistic Regression and SVM are slightly higher at 1.18. KNN and NMF have the highest RMSE values, 1.30 and 1.25 respectively, suggesting less accuracy.



Conclusions

1) **Model Performance**: Neural Network Embedding demonstrated the best performance with the lowest RMSE, indicating superior accuracy in predicting user ratings compared to other models.
2) **Algorithm Comparison:** Traditional models like Logistic Regression and Decision Trees showed reasonable performance but were outperformed by advanced techniques like Gradient Boosting and Neural Network Embedding.
3) **Dimensionality Reduction:** Techniques such as NMF effectively managed large and sparse matrices, though they did not outperform Neural Network Embedding in prediction accuracy.

Appendix

https://github.com/inakilopezjuan/IBM-machine-Learning-capstone/upload