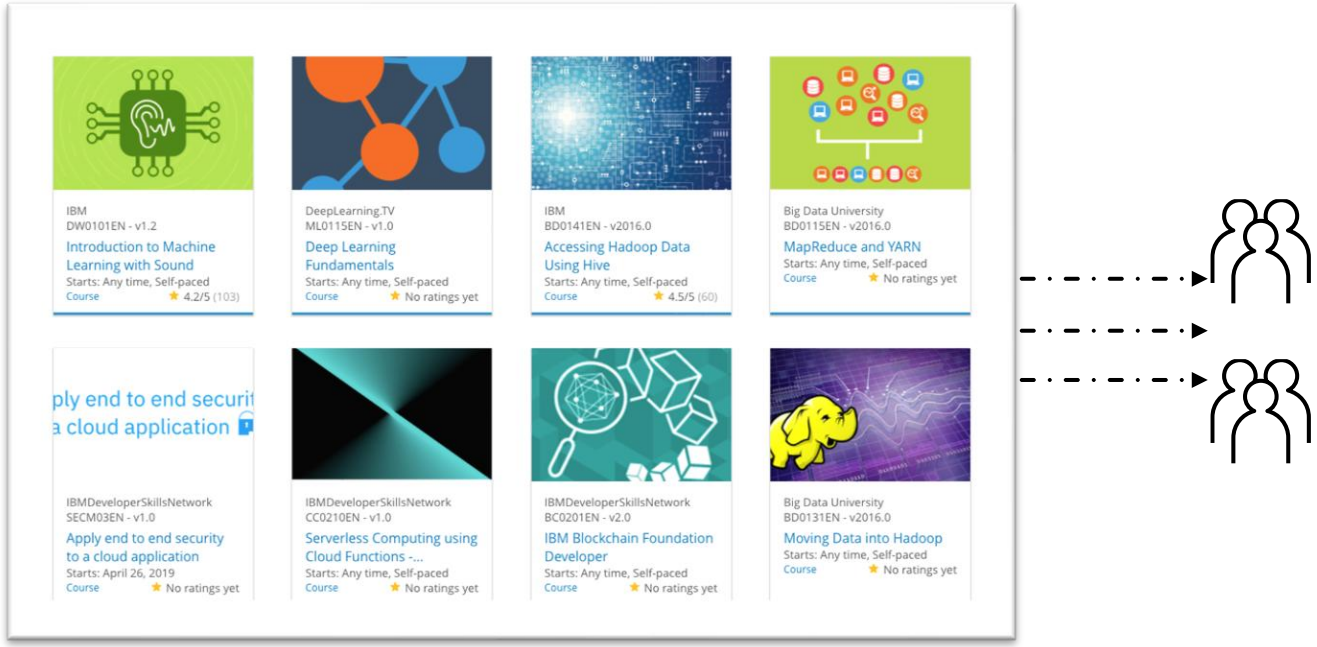
Build a Personalized Online Course Recommender System with Machine Learning



By David Satria Alamsyah

18 July 2025

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:

- Context: We are an ML team at AI Training Room, a rapidly growing MOOC platform with millions of global learners.
- Goal: To develop a course recommender system to enhance the learning experience. This project is currently in the Proof of Concept (PoC) phase, focusing on offline model evaluation.

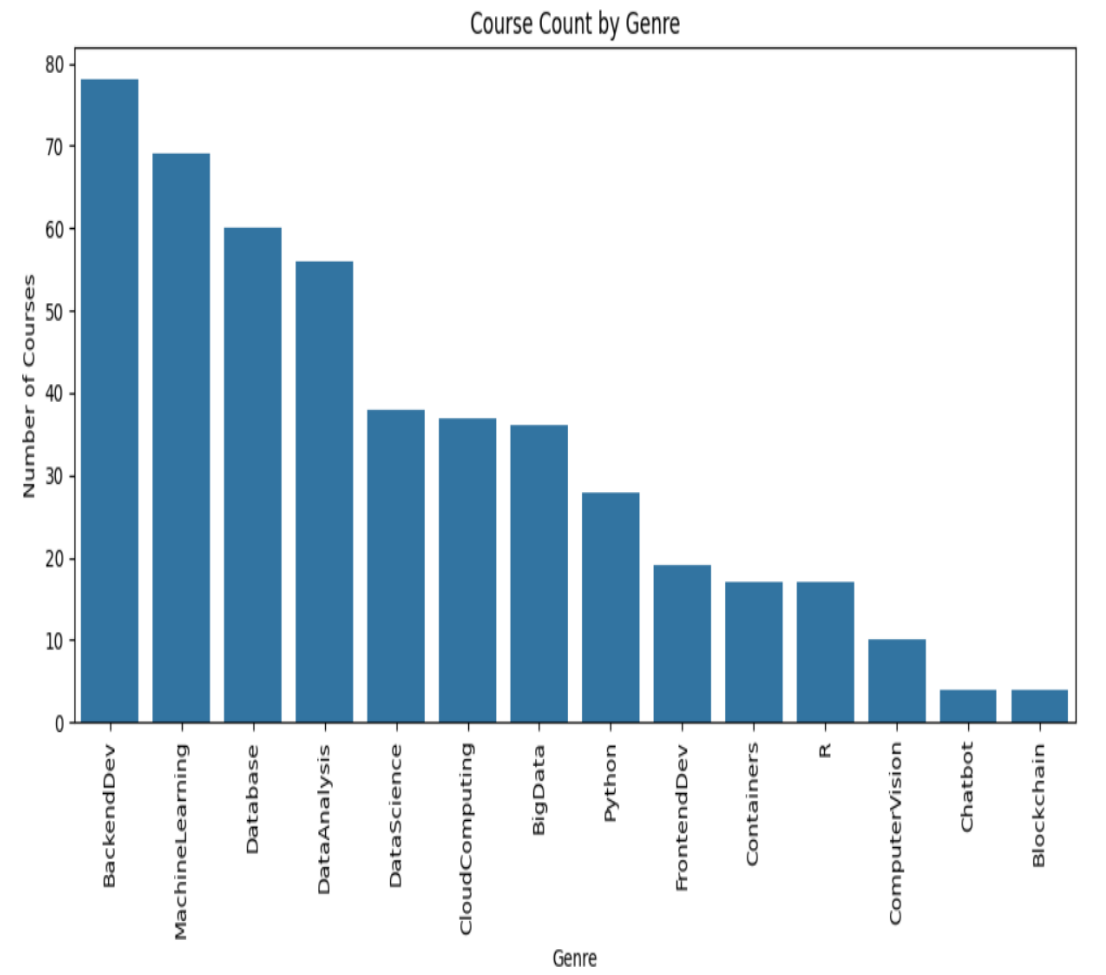
Problem Statement & Hypothesis:

- Problem: With a massive and growing catalog, learners may find it difficult to discover new, relevant courses, potentially leading to decreased engagement.
- Hypothesis: A personalized recommender system will improve course discovery and increase learner interaction, leading to better learning paths and potentially higher revenue.

Exploratory Data Analysis

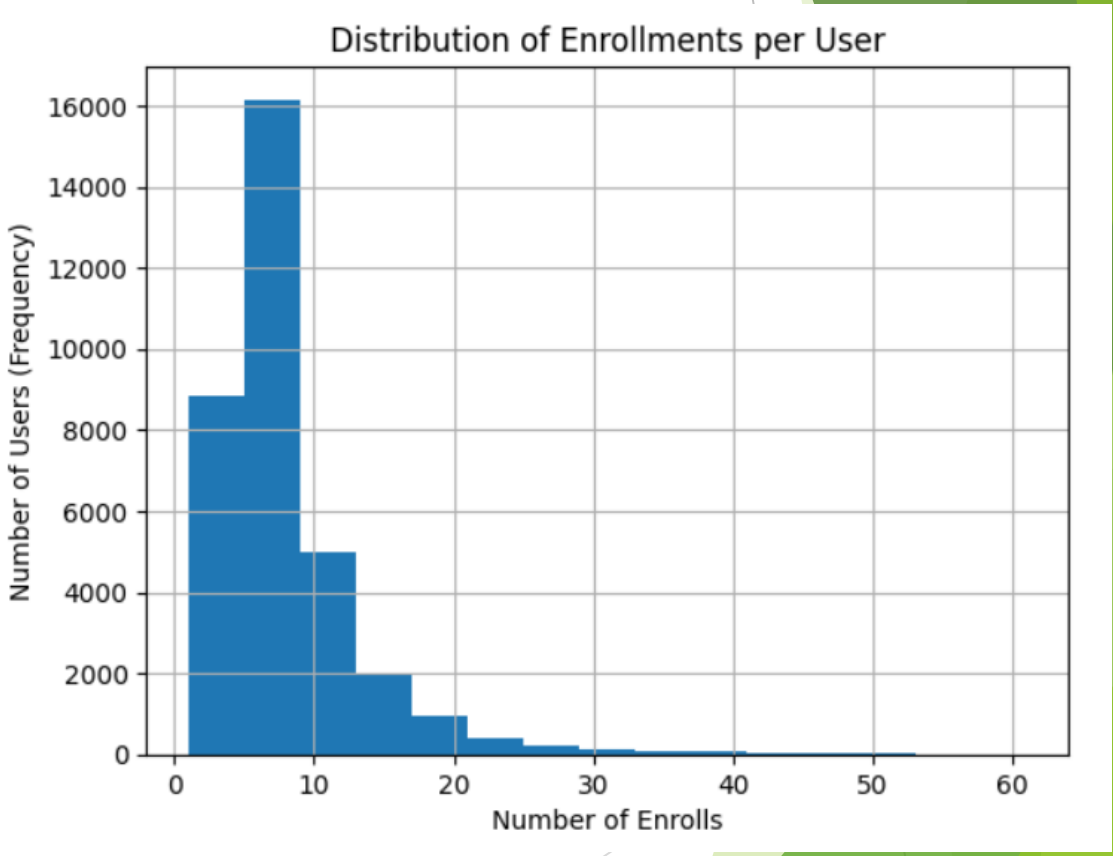
Course counts per genre

- Web Development has the highest number of courses.
- Machine Learning and Data Science are also dominant subjects.
- Subjects like Blockchain and Cloud have fewer, more specialized courses.



Course enrollment distribution

- The vast majority of users are enrolled in 1-10 courses.
- The number of users drops sharply as the number of enrolled courses increases.
- A very small number of "power users" enroll in a large number of courses (e.g., 20+).



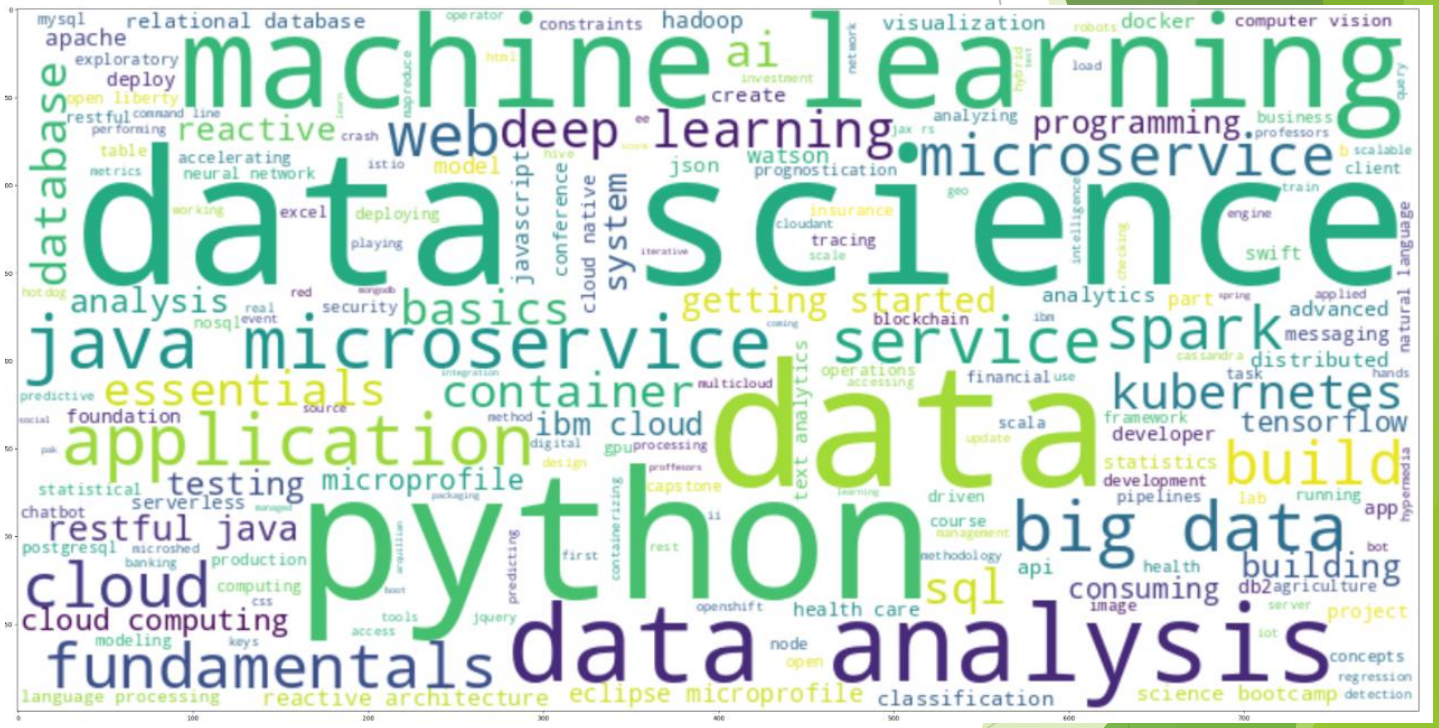
20 most popular courses

- Foundational courses like "Introduction to Data Science" and “Python for Data Science" are the clear leaders, each with over 10,000 enrollments.
- AI and Data Science subjects ("Machine Learning for Beginners", "Data Science with Python") form the next most popular group, showing strong interest in this domain.
- The list also includes many courses on specific, in-demand tools like AWS, React, Docker, and TensorFlow, indicating learners are seeking practical job skills.

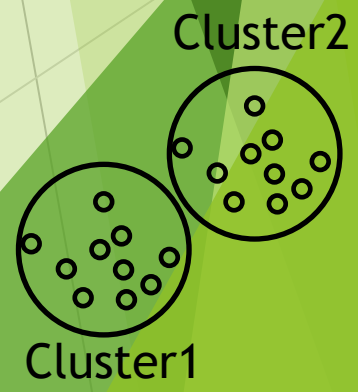
	TITLE	Enrolls
0	python for data science	14936
1	introduction to data science	14477
2	big data 101	13291
3	hadoop 101	10599
4	data analysis with python	8303
5	data science methodology	7719
6	machine learning with python	7644
7	spark fundamentals i	7551
8	data science hands on with open source tools	7199
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

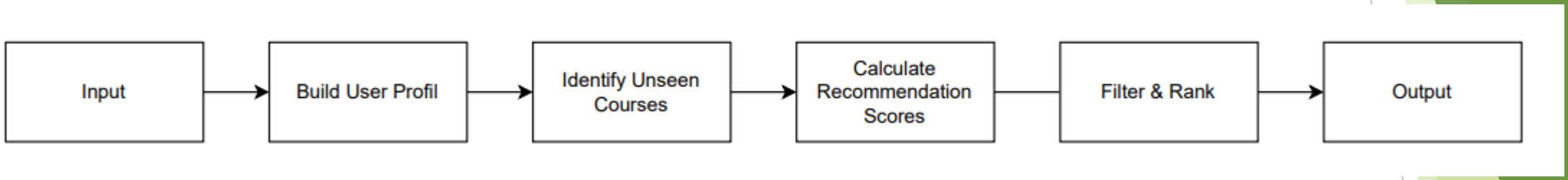
- The most prominent words are "Python", "Data", "Learning", "Web", and "Machine".
- The size of the words visually confirms the platform's strong focus on Data Science, Python, AI, and Web Development.
- Keywords like "Introduction" and "Beginners" are also frequent, aligning with the popularity of foundational courses.



Content-based Recommender System using Unsupervised Learning



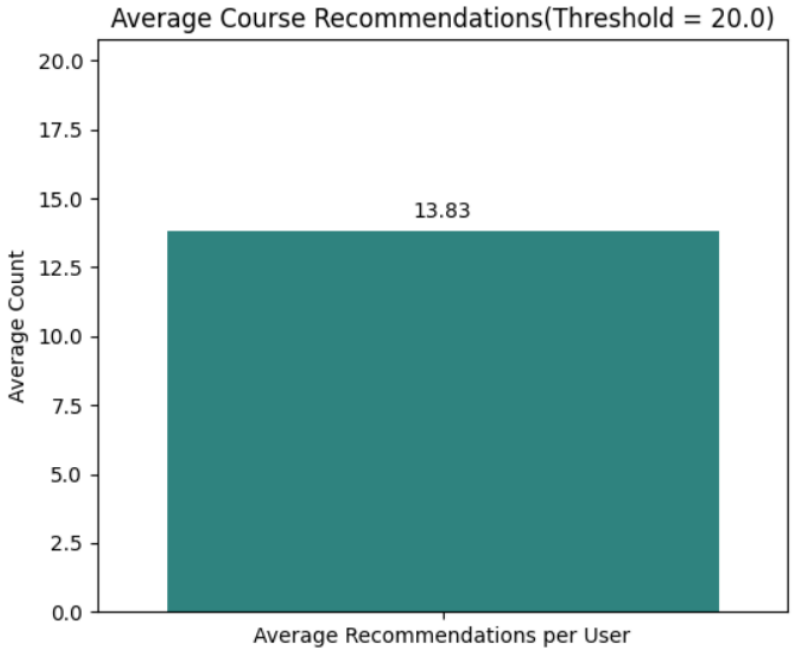
Flowchart of content-based recommender system using user profile and course genres

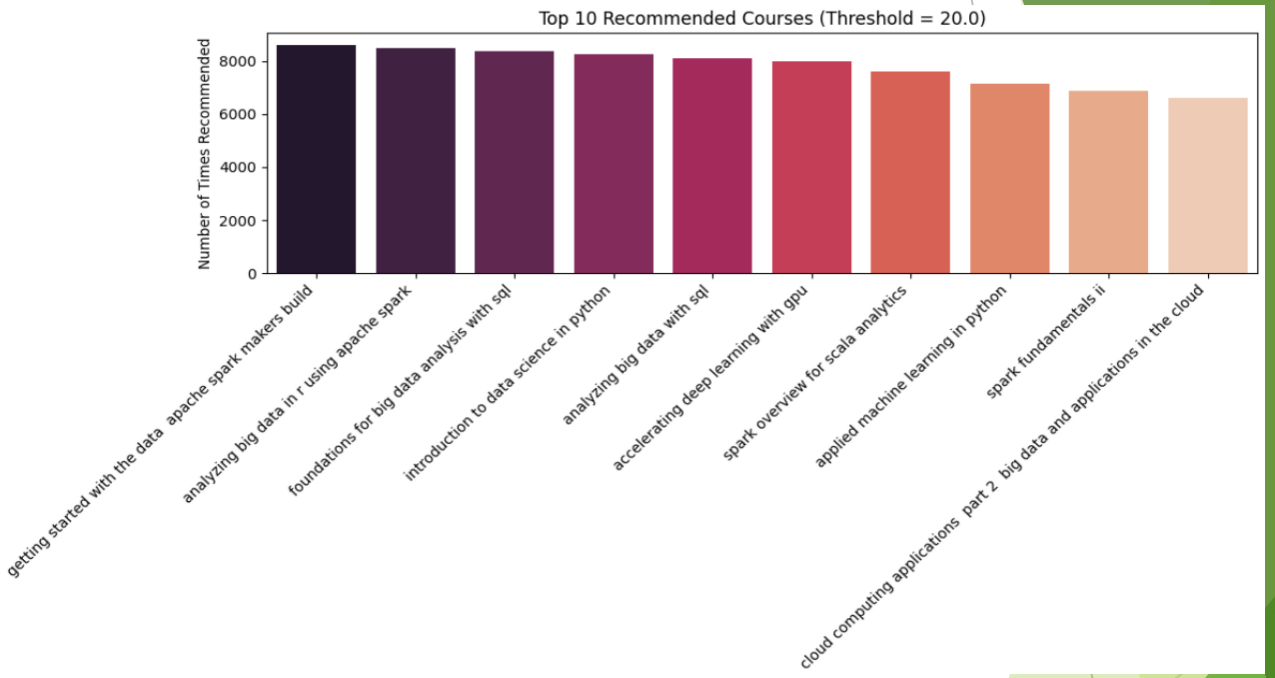


Hyper-parameters:

- A recommendation score threshold of 20.0 was set. Only courses with a score above this value are considered
- The maximum number of recommendations is capped at **20 courses** per user

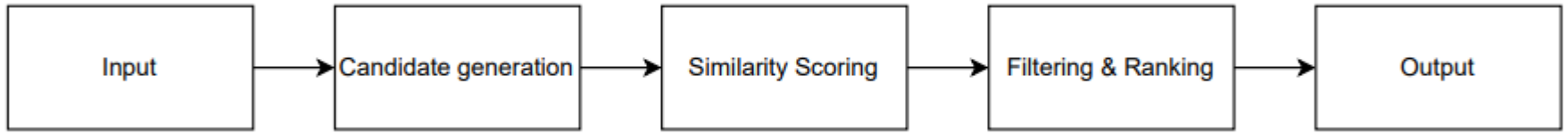
Evaluation results of user profile-based recommender system





- On average, **13.83 new/unseen courses** were recommended per user in the test dataset.
- The most frequently recommended courses are heavily focused on **Big Data (Spark, SQL)** and **Data Science**

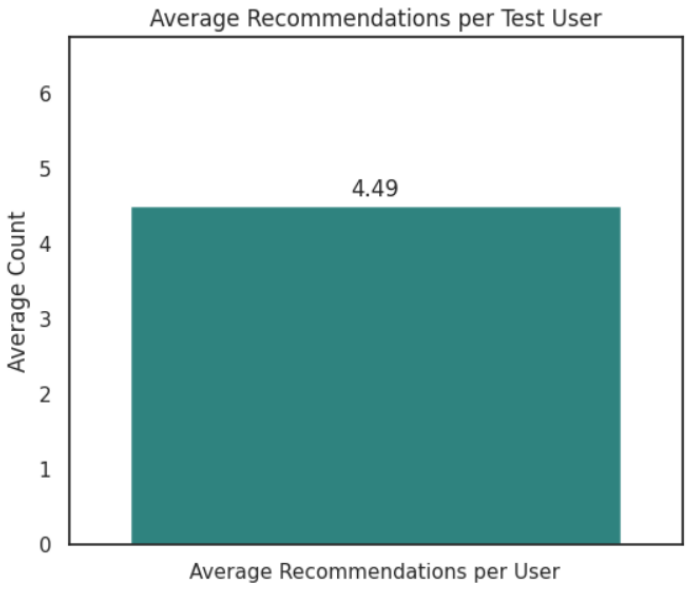
Flowchart of content-based recommender system using course similarity

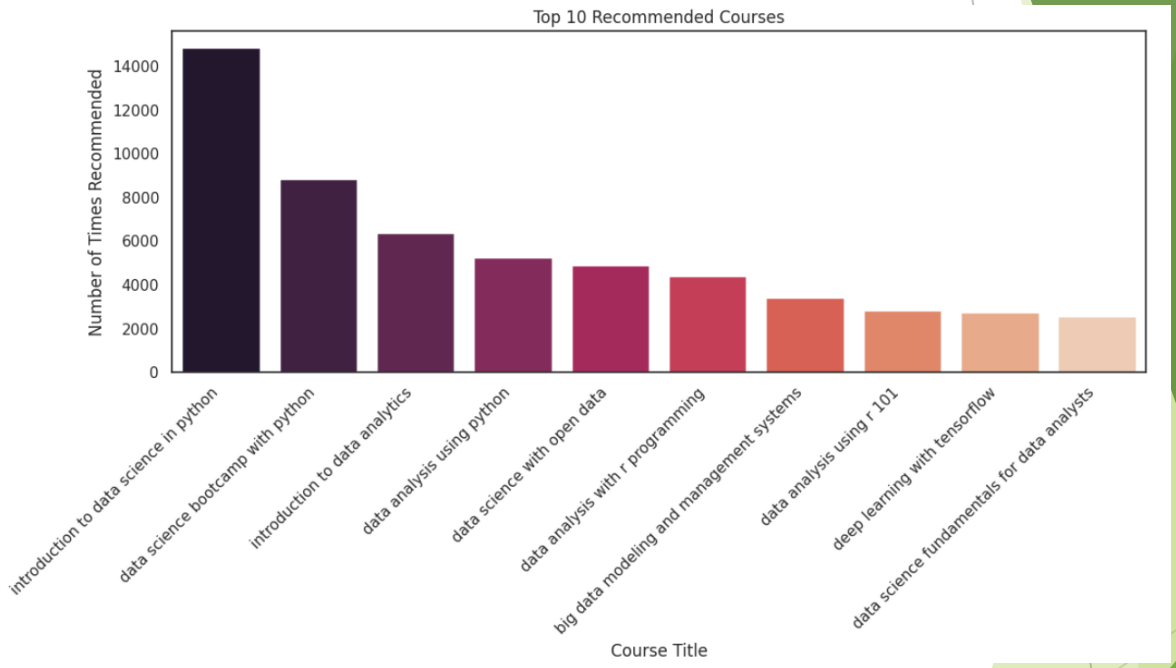


Hyper-parameters:

- **Similarity Score Threshold: 0.5**. A course is only considered for recommendation if its similarity score with an enrolled course is greater than this value.
- The maximum number of recommendations is capped at **20 courses** per user

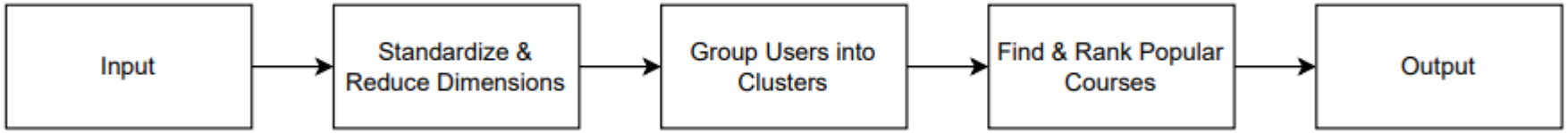
Evaluation results of course similarity based recommender system





- On average, each user in our test dataset receives **4.49 new course recommendations.**
- The single most recommended course is "introduction to data science in python", appearing in over 14,800 user recommendations.
- The top recommended courses are overwhelmingly focused on foundational skills in Data Science, with a heavy emphasis on the Python and R programming languages.

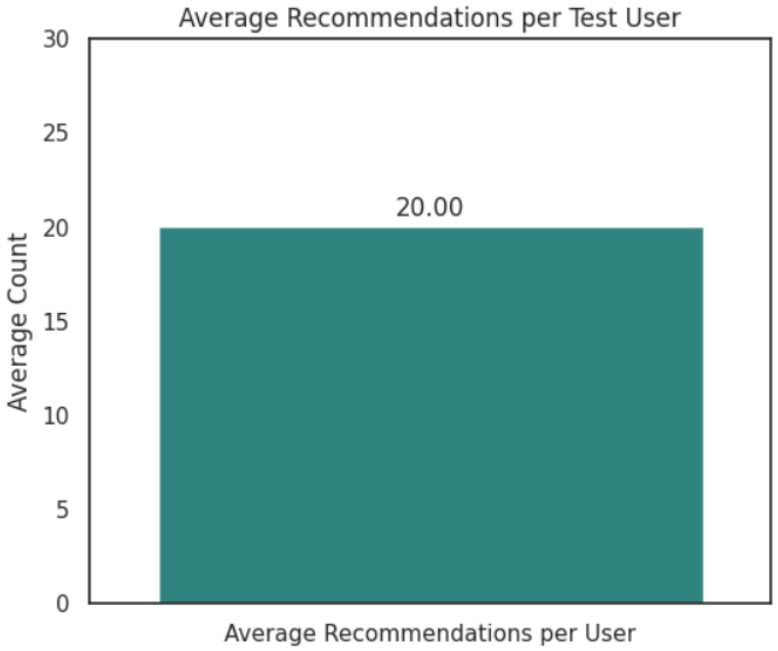
Flowchart of clustering-based recommender system

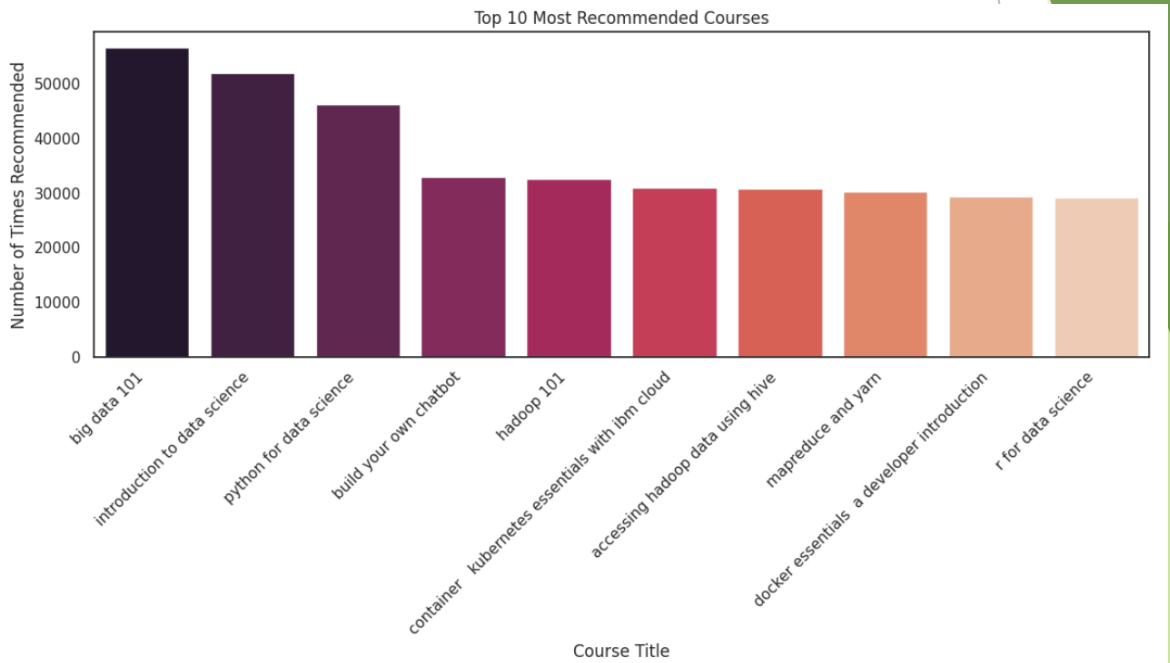


Hyper-parameters:

- **Number of Principal Components (PCA): 9**. This was determined as the minimum number of components required to explain at least 90% of the variance in the original user profile data.
- **Number of Clusters (K-Means): 9**. This was identified as the optimal number of user clusters using the elbow method.
- **Recommendation Cutoff:** The **Top 20** most popular courses within a user's cluster are selected as potential recommendations.

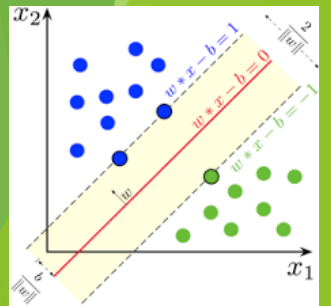
Evaluation results of clustering-based recommender system



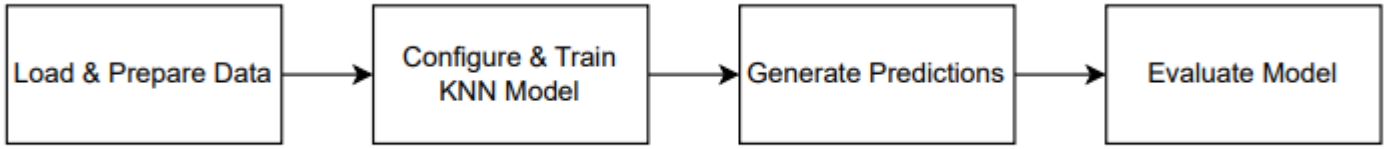


- On average, each user in the test dataset received **20.00 new course recommendations**
- The most frequently recommended courses are foundational topics in **Big Data and Data Science.**
- **"big data 101"** was the most common suggestion, recommended over 56,000 times, followed by **"introduction to data science"** and **"python for data science".**

Collaborative-filtering Recommender System using Supervised Learning



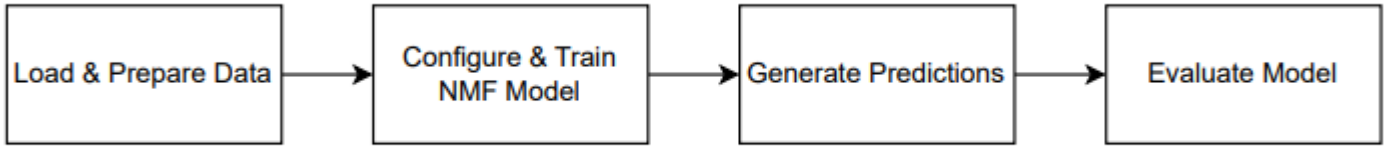
Flowchart of KNN based recommender system



Hyper-parameters:

- Algorithm: **KNNBasic** from the **surprise library**.
- **Collaborative Filtering Method:** The notebook implemented and tested two methods: **Item-based** and **User-based.**
- **Number of Neighbors (k): 40**
- **Minimum Neighbors (min_k): 1**
- **Similarity Metric: Cosine similarity**

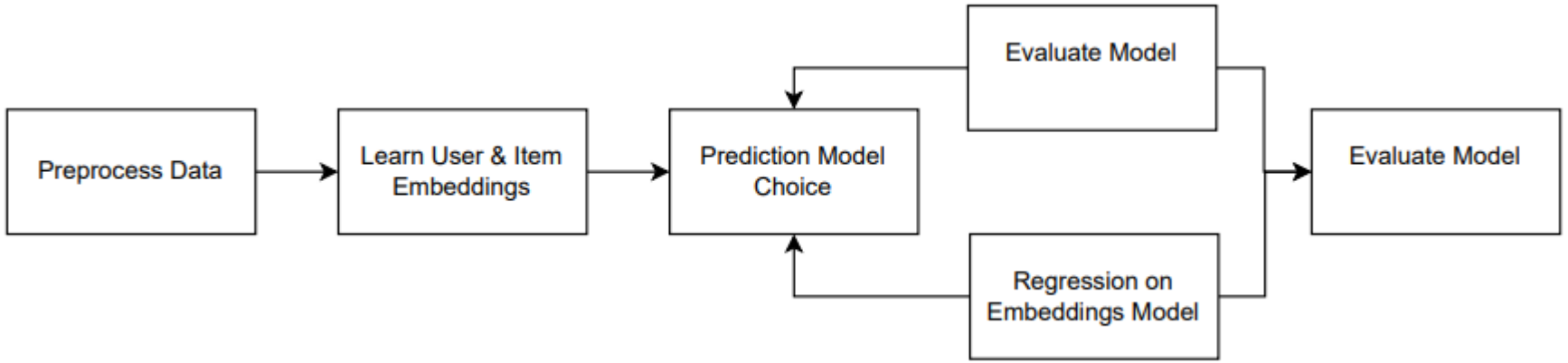
Flowchart of NMF based recommender system



Hyper-parameters:

- **Algorithm: NMF** from the surprise **library**.
- **Number of Latent Factors: 32**.
- **Initialization Range**: The initial random values for the user and item feature matrices are set between **0.5 and 5.0.**

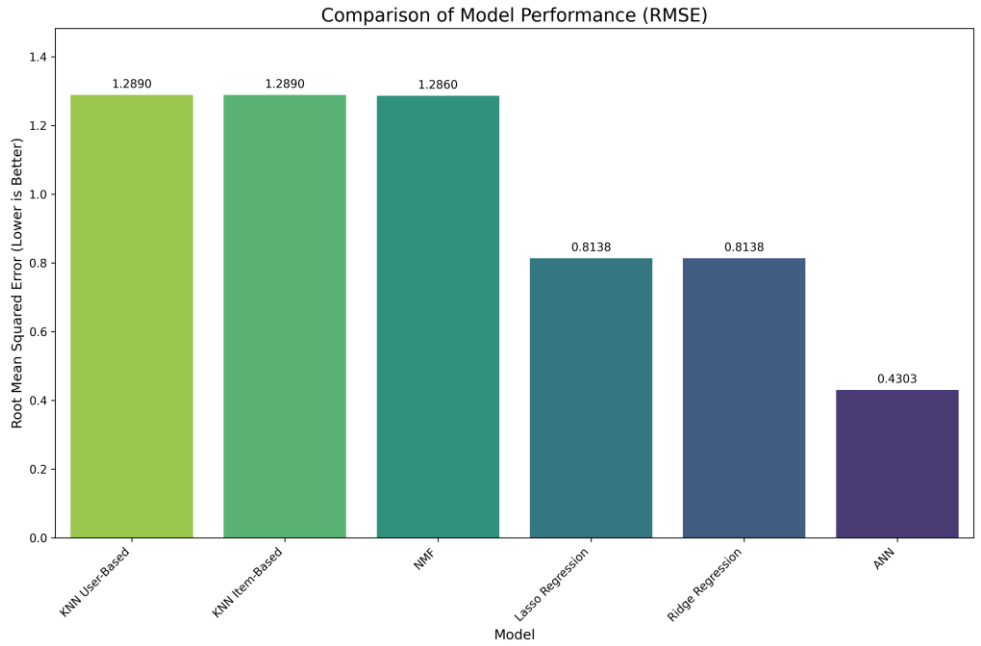
Flowchart of Neural Network Embedding based recommender system



Hyper-parameters:

- **Model A: ANN** (Deep Learning) Model
- **Embedding Size: 16** for both user and item latent feature vectors.
- **Neural Network Architecture**: Two Dense Layers with **128** units each.
- **ReLU activation** function for the dense layers.
- Training Parameters:
  - **Optimizer: adam.**
  - **Loss Function: mse** (Mean Squared Error).
  - **Batch Size: 64.**
  - **Epochs: 5.**
- **Input Features:** Concatenated user and item **embedding vectors (each of size 16),** extracted from a **pre-trained ANN model.**
- **Hyperparameter Tuning Method**: GridSearchCV with 5-fold cross-validation (**cv=5**).
- **Algorithm 1: Ridge Regression**
  - Parameter Grid for **alpha**: [0.001, 0.01, 0.1, 1, 10, 100]
- **Algorithm 2: Lasso Regression**
  - Parameter Grid for **alpha**: [0.001, 0.01, 0.1, 1, 10, 100]
  - **Max Iterations (max_iter):** Set to 10000

Compare the performance of collaborative-filtering models



- **Clear Performance Tiers:** The models can be grouped into **three distinct performance** tiers based on their Root Mean Square Error (RMSE).
- **Baseline Performance:** The traditional models (**User-based KNN, Item-based KNN, and NMF**) perform similarly, establishing a baseline with the highest error (**RMSE ≈ 1.29**).
- **Mid-Tier Improvement: Lasso and Ridge regression** models, which use learned embeddings as features, offer a significant improvement over the baseline, with a much lower **RMSE of approximately 0.81**.
- **Top Performer: The Artificial Neural Network (ANN)** model is the clear top performer, achieving the lowest error with an **RMSE of 0.43**, nearly halving the error of the regression models.

Key Insights

- **Multi-Model Approach:** The project successfully implemented and compared multiple recommender system models, including a content-based model and several collaborative filtering methods (K-Means Clustering, NMF, and ANN). This provides a comprehensive overview of different techniques for course recommendation.
- **Deep Learning Superiority:** For the task of predicting user ratings (collaborative filtering), the Artificial Neural Network (ANN) model significantly outperformed traditional methods. It achieved the lowest Root Mean Square Error (RMSE) of 0.43, compared to higher error rates for KNN and NMF models (RMSE ≈ 1.29).
- **Complementary Strengths:** The report highlights that content-based and collaborative filtering models have complementary strengths. The content-based model is effective at finding topically similar courses, while collaborative filtering excels at predicting user ratings based on the behavior of similar users

Conclusions

- The project successfully developed functional content-based and collaborative filtering recommender systems to help learners discover relevant courses.
- The content-based model proved effective for suggesting courses with similar textual content, while the ANN-based collaborative filtering model was the most accurate at predicting user ratings.
- The analysis concludes that a hybrid model, which combines both content-based and collaborative filtering techniques, would likely provide the most robust and comprehensive recommendation solution by leveraging the strengths of each approach.

Next Steps

- **Develop a Hybrid System:** The primary recommendation for future work is to build a hybrid recommender system that integrates both the content-based and the best-performing collaborative filtering models to generate more powerful and accurate suggestions.
- **Model Deployment:** A key next step is to deploy the finalized model as an interactive web application or a REST API, making the recommender system accessible to end-users on the learning platform.
- **Explore Advanced Models:** The report suggests investigating more advanced deep learning architectures, such as Transformer-based models, to potentially improve recommendation accuracy further by capturing more complex patterns and sequential user behaviors.

Appendix

- Code Program for EDA

  - https://colab.research.google.com/drive/1ygK_jR1uKGW7CM30pvjYNqW31qTDcGNK?usp=sharing

- Code Program for Content-Based

  - User profile: https://colab.research.google.com/drive/1CQwEkaRxxybvlTTkVLtP5oQ_PNWFEef8?usp=sharing

  - Course Similarity: https://colab.research.google.com/drive/1J5Fll-n7ZVGjrhwgQjlmr78auJ4qerad?usp=sharing

  - Clustering: https://colab.research.google.com/drive/1NpA43uWK1qyy5QCwSDhfchmrBvFV_acb?usp=sharing

- Code Program for Collaborative-Filtering

  - KNN: https://colab.research.google.com/drive/1j7kBBgylgTEXpH-zoe4dwQYXhdwwsh1s?usp=sharing

  - NMF: https://colab.research.google.com/drive/14yg8umTbJc4sSuErUHQy08in_xwwdphs?usp=sharing

  - ANN: https://colab.research.google.com/drive/1nz85nOAU-raiCWrs31--eG0v_BDx-GLT?usp=sharing

  - Regression : https://colab.research.google.com/drive/1Ne9YyizpNGmP8Ktv3Wkxd3VlftJ0lkRl?usp=sharing

- Pptx : if you want to see the slide notes for further explanation

  - https://docs.google.com/presentation/d/1A096KjlH5iA2tD66vvugt9ZLYkfbPln_/edit?usp=sharing&ouid=111107553465690610542&rtpof=true&sd=true