Build a Personalized Online Course Recommender System with Machine Learning

- Guillaume ROSPAPE

- 2024-04-25

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

Welcome to our capstone project! In this endeavor, we apply our machine learning knowledge and skills to address real-world industrial challenges. Our project is in collaboration with AI Training Room, a rapidly growing Massive Open Online Courses (MOOCs) startup that offers learning in leading technologies such as Machine Learning, AI, Data Science, Cloud, App development, and more.

Introduction

Project Scenario

We are working on a recommender system project. The primary objective of this project is to enhance the learners' experience by helping them quickly discover new courses of interest and providing a more structured learning path. Additionally, increased learner-course interaction via our recommender system may boost the company's revenue.

Introduction

Problem Statement

- Difficulty in helping learners quickly find new courses of interest.
- Inefficiency in providing personalized learning paths for learners.
- The need to improve learner-course interaction to increase company revenue.

Introduction

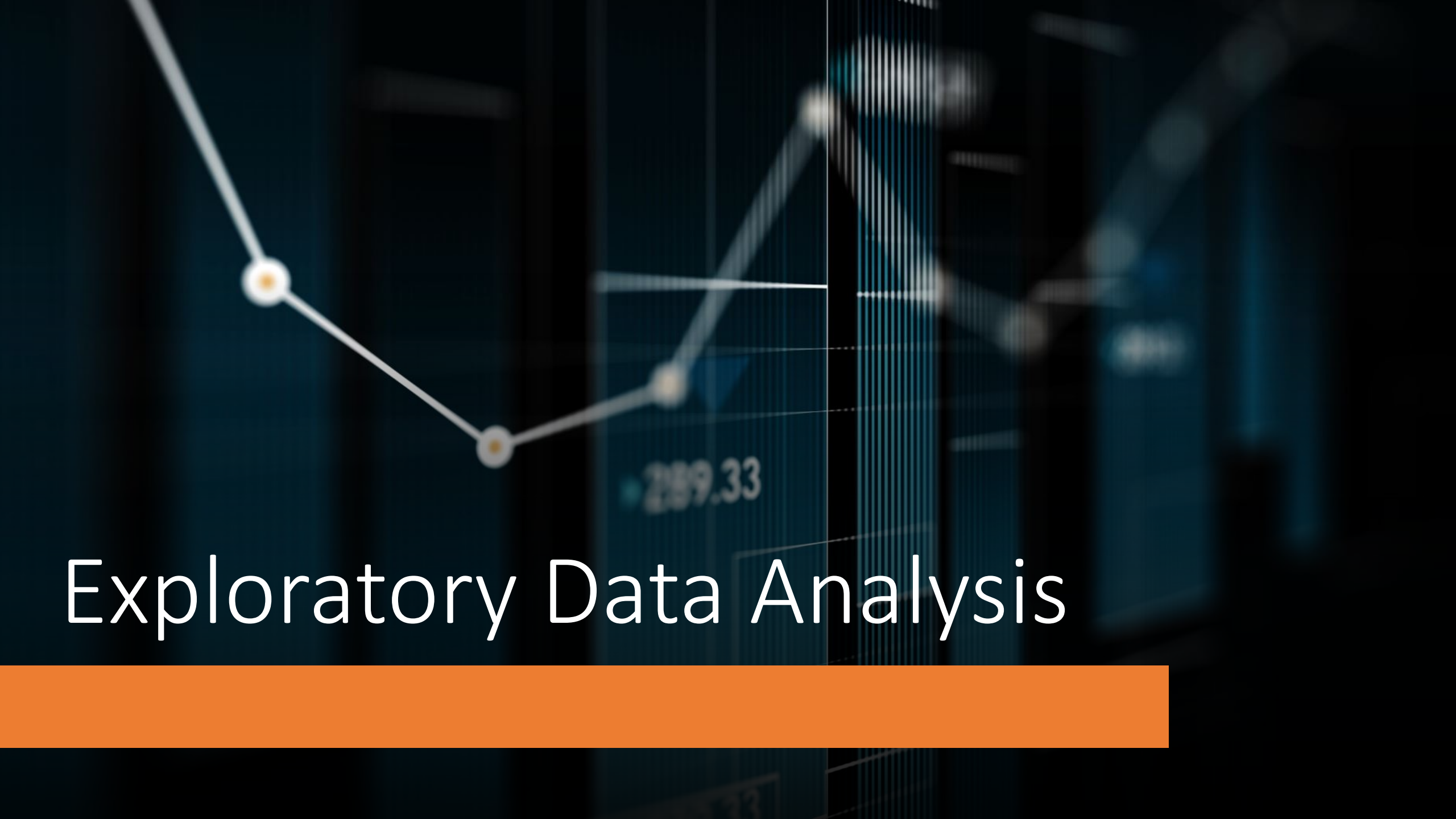
Our tasks in this project include:

- Collecting and understanding data
- Performing exploratory data analysis on online course enrollments datasets
- Extracting Bag of Words (BoW) features from course textual content
- Calculating course similarity using BoW features
- Building content-based recommender systems using various unsupervised learning algorithms
- Building collaborative-filtering recommender systems using various supervised learning algorithms

Introduction

Hypotheses

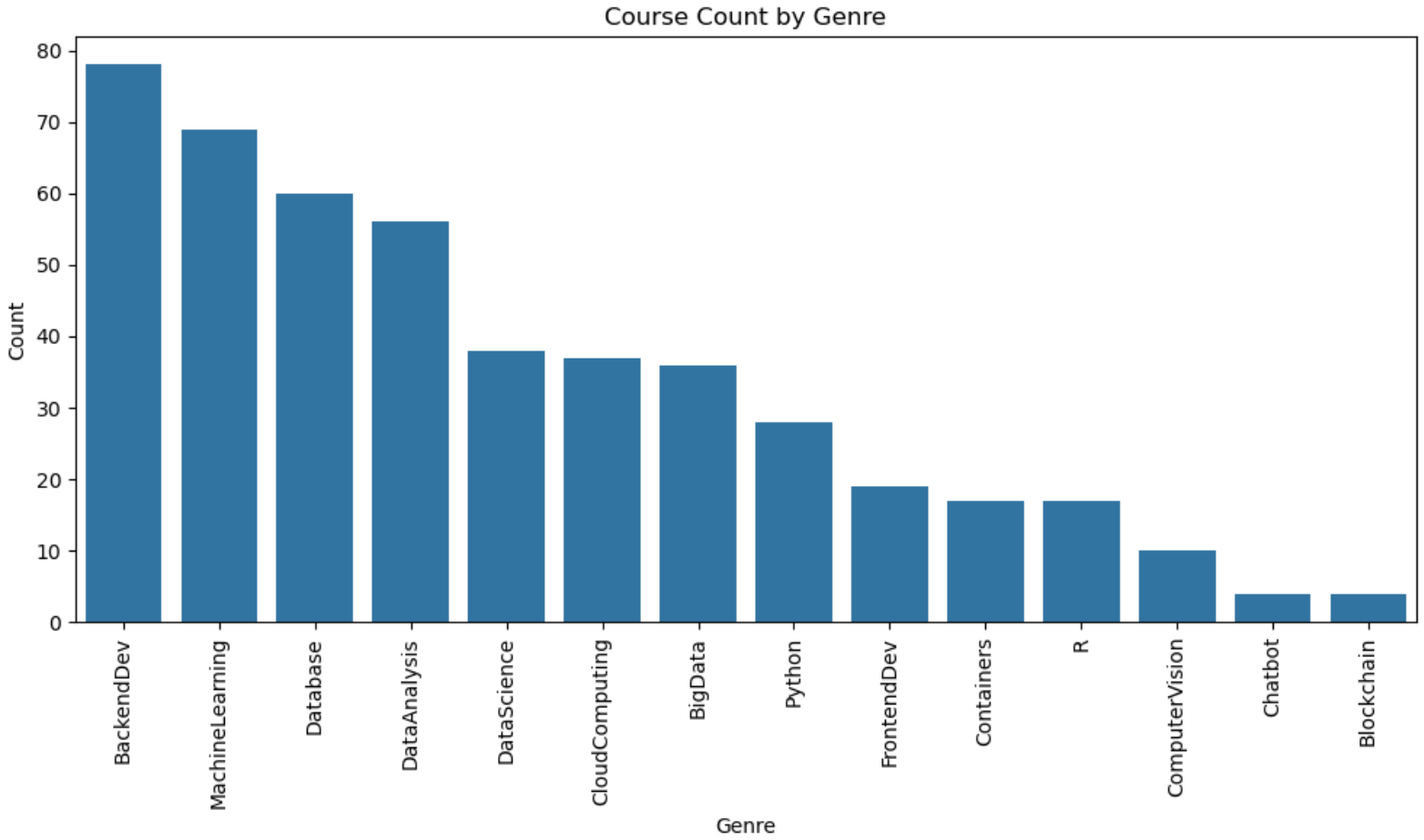
- Implementing a recommender system will improve learners' ability to find new courses of interest.
- Personalized learning paths will enhance the overall learning experience.
- Increased learner-course interaction facilitated by the recommender system will lead to increased company revenue.



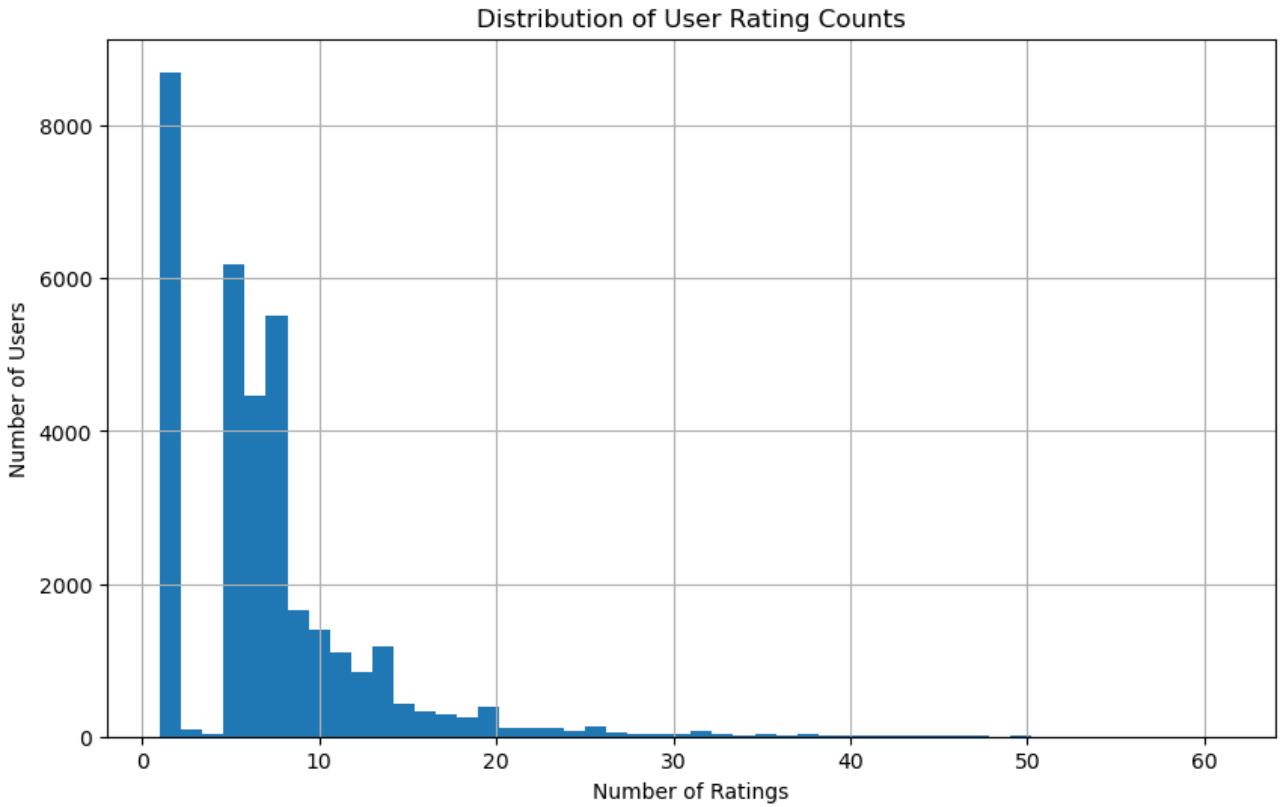
Exploratory Data Analysis

Course counts per genre

- The bar chart represents the count of courses available in different genres on the platform.
- The most popular genre is Backend Development with 78 courses, followed by Machine Learning with 69 courses, and Database with 60 courses. Data Analysis and Data Science also have a significant presence with 56 and 38 courses, respectively.
- Other genres like Cloud Computing, Big Data, Python, Frontend Development, Containers, and R have a moderate number of courses ranging from 17 to 37 courses each.
- Computer Vision has 10 courses, while Chatbot, Blockchain, and some other genres have fewer than 10 courses available.
- This bar chart helps to visualize the distribution of courses across different genres and identify the most popular and least popular genres on the AI Training Room platform.



Course enrollment distribution



- The histogram shows the distribution of user enrollments in courses.
- The x-axis displays the number of courses enrolled, while the y-axis shows the count of users. The tallest bars represent the most common number of courses enrolled by users, while shorter bars indicate fewer users enrolled in that many courses.
- This visualization helps identify patterns in user behavior and enrollment trends.

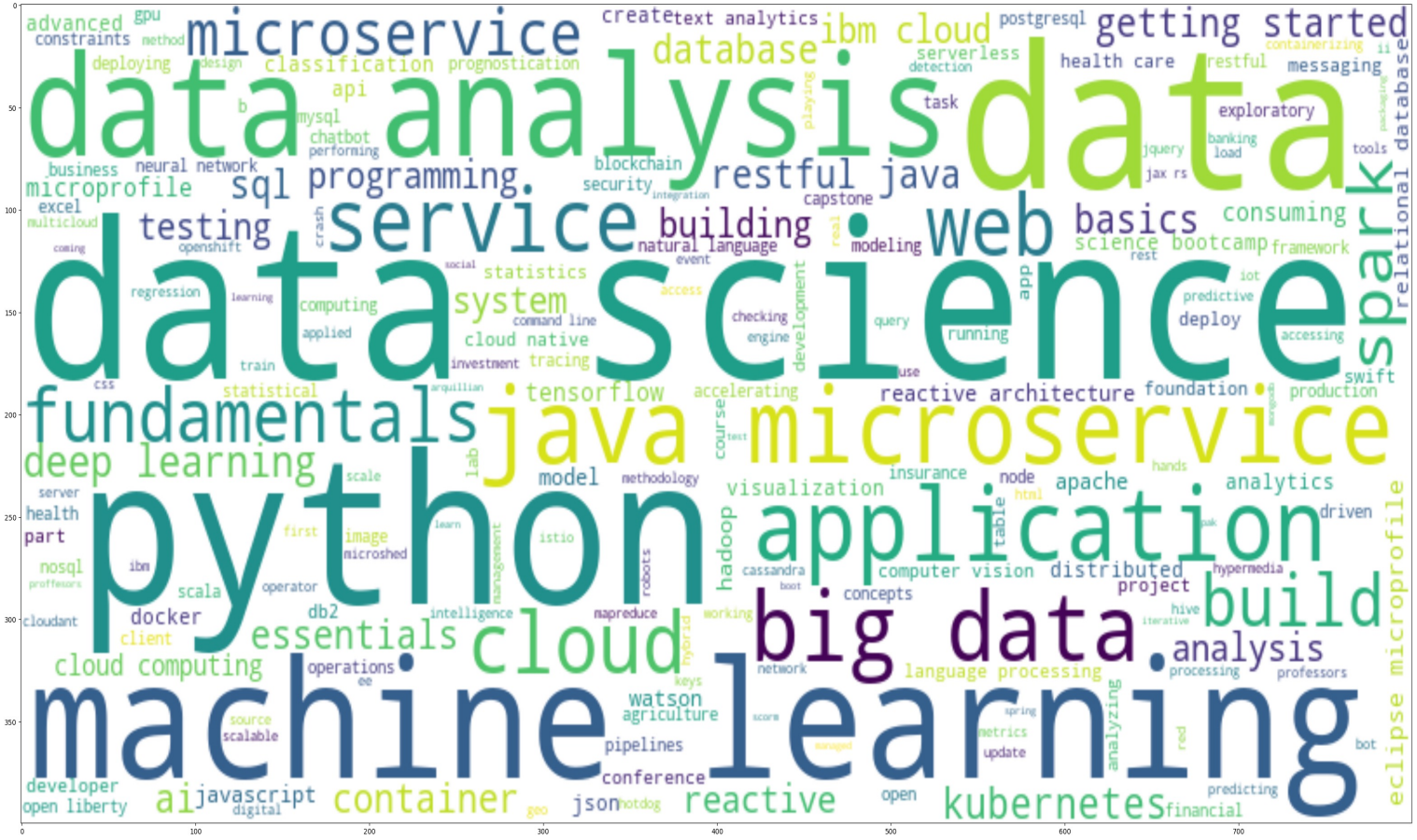
20 most popular courses

- This slide shows the 20 most popular courses on the platform, ranked by user ratings.
- Topics include data science, machine learning, cloud computing, and more. The top-rated course is "Python for Data Science" with over 14,000 ratings.
- Other popular courses include "Introduction to Data Science," "Big Data 101," and "Machine Learning with Python." The list provides insight into the most sought-after courses on the platform.

	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

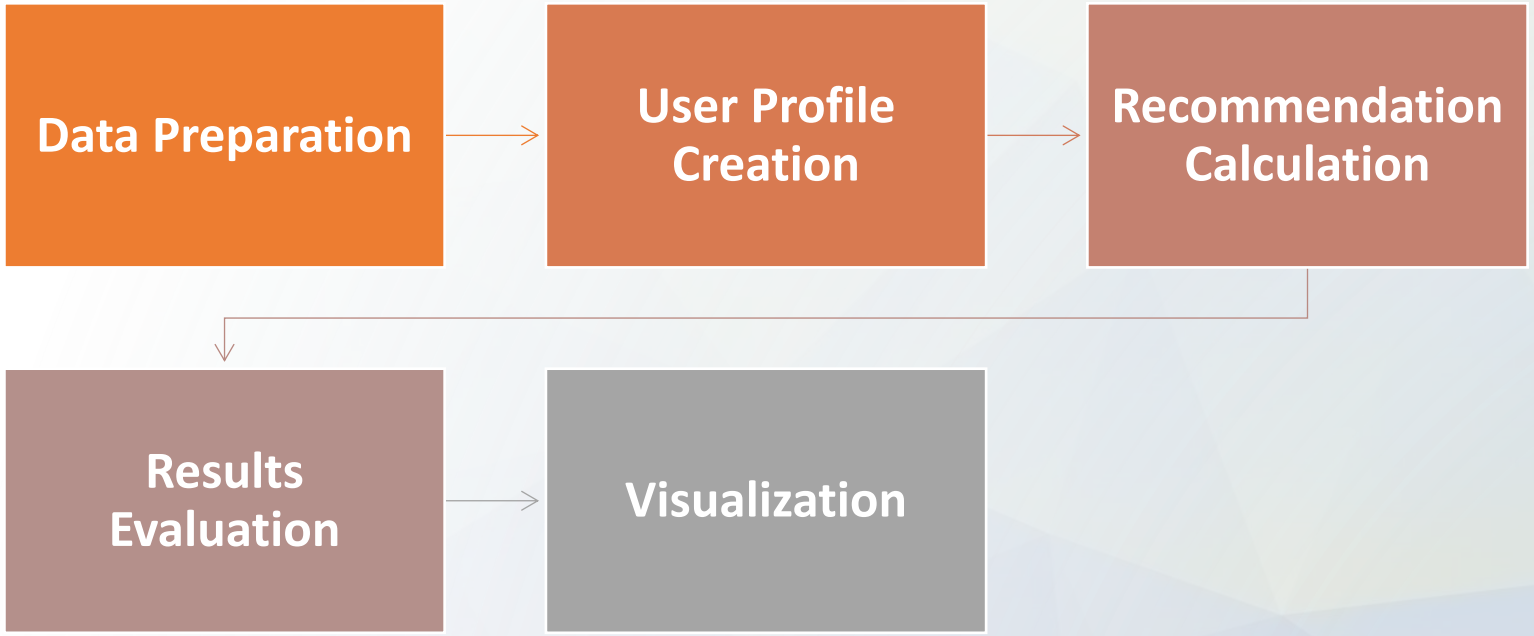
Word cloud of course titles

- The word cloud in the slide is a visual representation of the frequency of various terms of course titles on the platform.
- The larger the font size of a word, the more often it is presumably mentioned or emphasized in the related content.
- The word cloud effectively highlights the central themes and skills that are likely discussed in the presentation, making it a useful tool for a quick visual summary and for emphasizing the importance and interconnection of these concepts in the field.



Content-based Recommender System using Unsupervised Learning

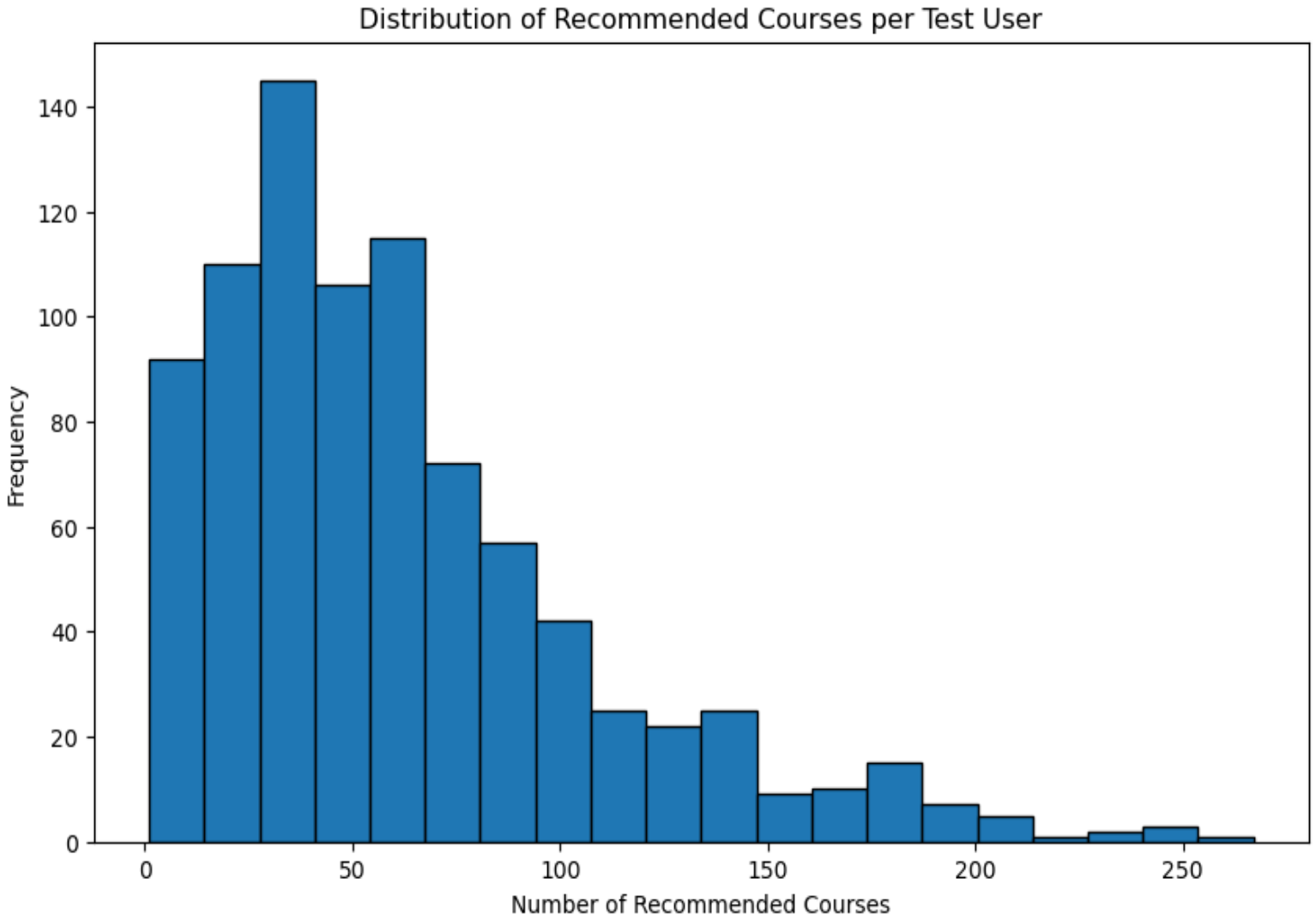
Flowchart of content-based recommender system using user profile and course genres



- **Data Preparation**: Courses and their associated genres are tabulated along with user ratings for these courses.
- **User Profile Creation**: User profiles are created by multiplying users' ratings for courses by the genre attributes of these courses, generating a profile vector that quantifies each user's preferences across different genres.
- **Recommendation Calculation**: Scores for recommending new courses are calculated using the dot product of the user profiles and the genre attributes of new courses, predicting user interest based on their profile.
- **Results Evaluation and Visualization**: The outcomes of the recommendations are saved and analyzed. Visualizations are created to display the distribution and frequency of recommended courses and to identify the most popular recommendations.

Evaluation results of user profile-based recommender system

- Random State (rs): We defined a random state rs = 123 at the beginning. This is used for reproducibility purposes, ensuring that any random operations in the code produce the same results each time the notebook is run with the same random state value.
- Recommendation Score Threshold: We defined a variable score_threshold = 10.0 as the threshold for filtering out courses with low recommendation scores.
- Only courses with recommendation scores greater than or equal to this threshold are considered for recommendation.

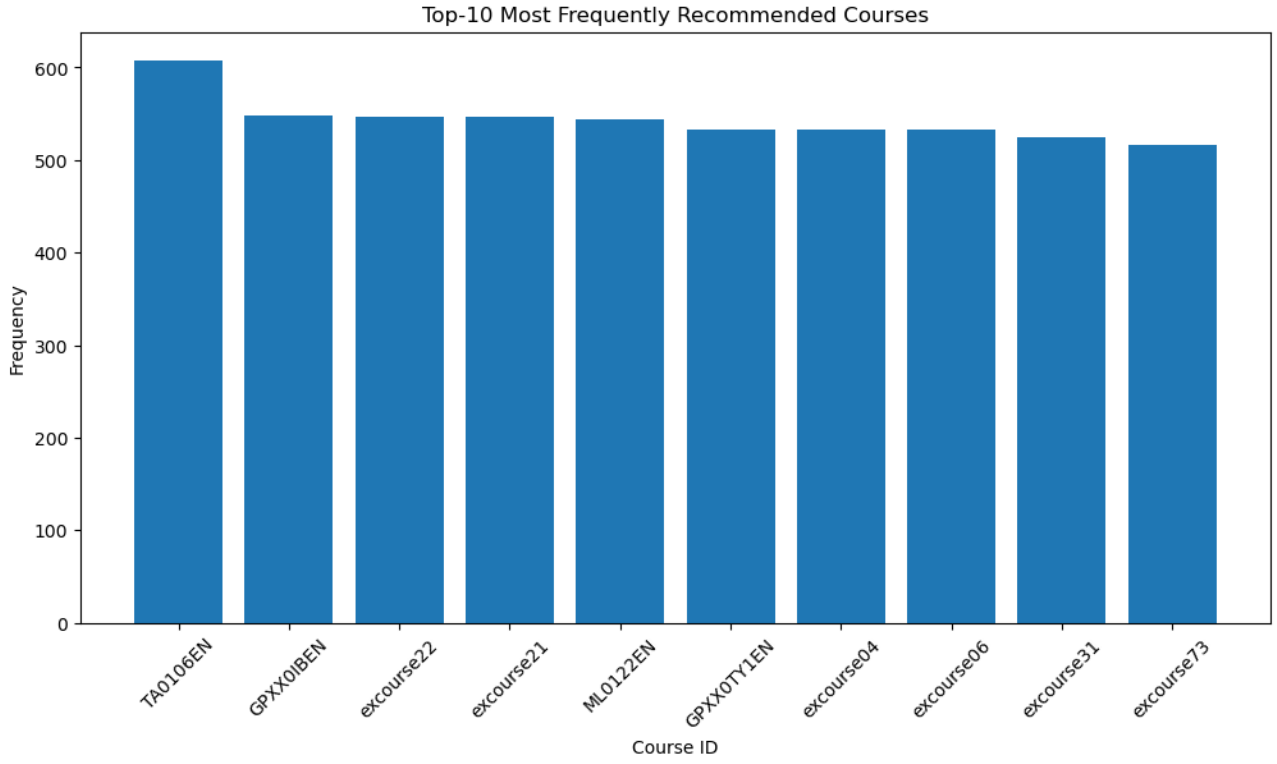


On average, 61.82 new courses have been recommended per test user.

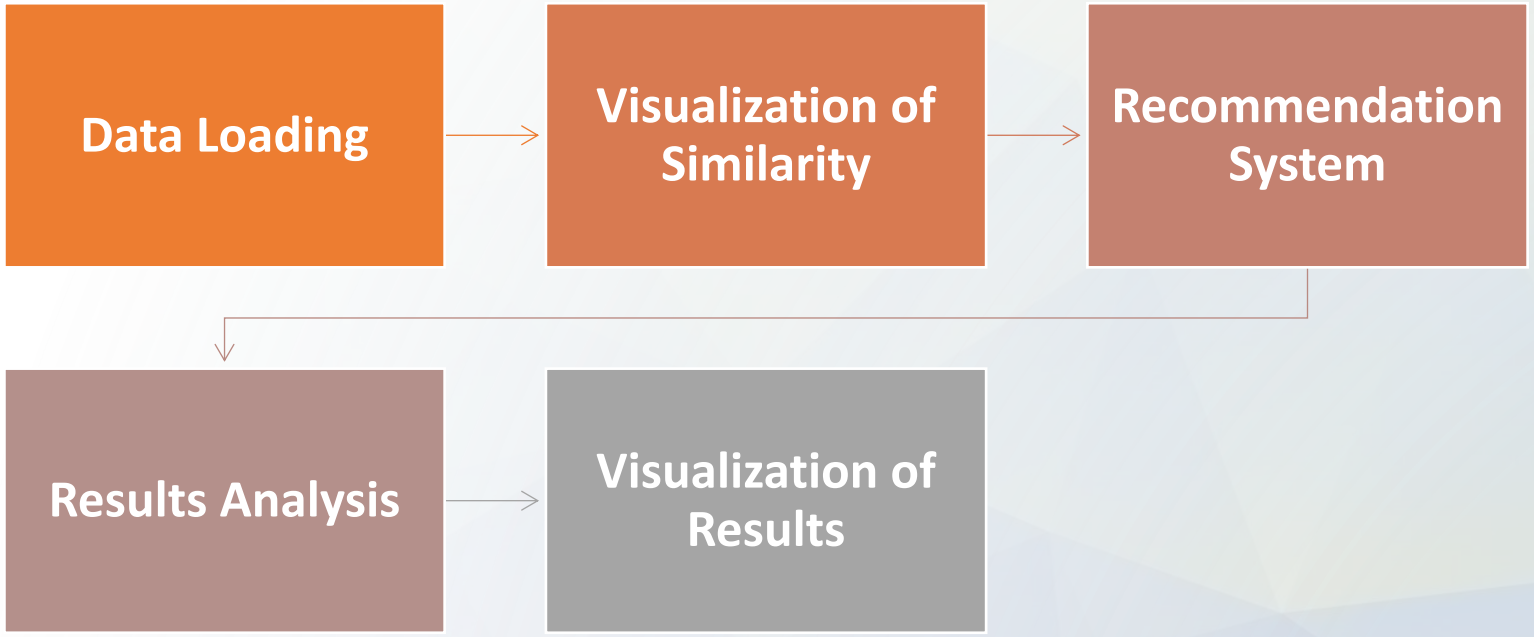
Evaluation results of user profile-based recommender system

Top-10 most frequently recommended courses:

	Course ID	Frequency
0	TA0106EN	608
1	GPXX0IBEN	548
2	excourse22	547
3	excourse21	547
4	ML0122EN	544
5	GPXX0TY1EN	533
6	excourse04	533
7	excourse06	533
8	excourse31	524
9	excourse73	516



Flowchart of content-based recommender system using course similarity



1.Data Loading: The system starts by loading course similarity data and course content from external sources into DataFrames.
2.Visualization of Similarity: A heatmap is generated to visually represent the similarity scores between courses, aiding in understanding the relationships based on content.
3.Recommendation System: Utilizing the similarity matrix, the system recommends new courses to users based on the courses they have previously interacted with, leveraging similarities to predict interests.
4.Results Analysis: The effectiveness of the recommendations is analyzed, providing insights such as the average number of courses recommended per user and identifying the most frequently recommended courses.
5.Visualization of Results: Results are visually presented through charts that show the distribution and frequency of recommended courses, including detailed displays of the top recommended courses.

Evaluation results of course similarity based recommender system

- Similarity threshold (threshold): The value is set to 0.6. This threshold is used to determine if a course is similar enough to be recommended. Courses with a similarity score greater than or equal to this threshold are considered as recommendations. threshold = 0.6
- Maximum number of recommended courses per user: 20
- Similarity matrix, user, course content, and BoW dataset

On average, 1.82 new/unseen courses have been recommended to each user.

Top-10 most frequently recommended courses:

	Course ID	Frequency
0	excourse62	257
1	excourse22	257
2	WA0103EN	101
3	TA0105	41
4	DS0110EN	38
5	excourse47	24
6	excourse46	24
7	excourse65	23
8	excourse63	23
9	ML0151EN	17

Flowchart of clustering-based recommender system

- **Data Preparation**: User profile data is loaded from a CSV file, and features are standardized to have zero mean and unit variance, making the data ready for clustering.
- **Clustering**: The K-means clustering algorithm is applied to group users into clusters based on similarities in their course interaction patterns. The optimal number of clusters is determined using the elbow method.
- **Analysis and Visualization**: Clusters are analyzed to understand common behaviors and preferences within each group. This step helps in tailoring recommendations based on observed cluster characteristics.
- **Recommendation Generation**: Courses that are popular within a user's cluster but not yet taken by the user are recommended, leveraging the collective preferences of the cluster.
- **Results Evaluation**: The system evaluates the effectiveness of the recommendations by calculating the average number of courses recommended per user and identifying the most frequently recommended courses.

Evaluation results of clustering-based recommender system

- **K-means Clustering**: Range of cluster numbers for determining the optimal number of clusters: 20. Optimal number of clusters identified from the elbow plot
- **PCA**: Range of candidate n_components arguments for determining the optimal number of components. Threshold for accumulated variance ratio to determine the optimal number of components: 0.9 (90%)
- **Course Recommendation**: Enrollment count threshold to determine popular courses within a cluster: 20

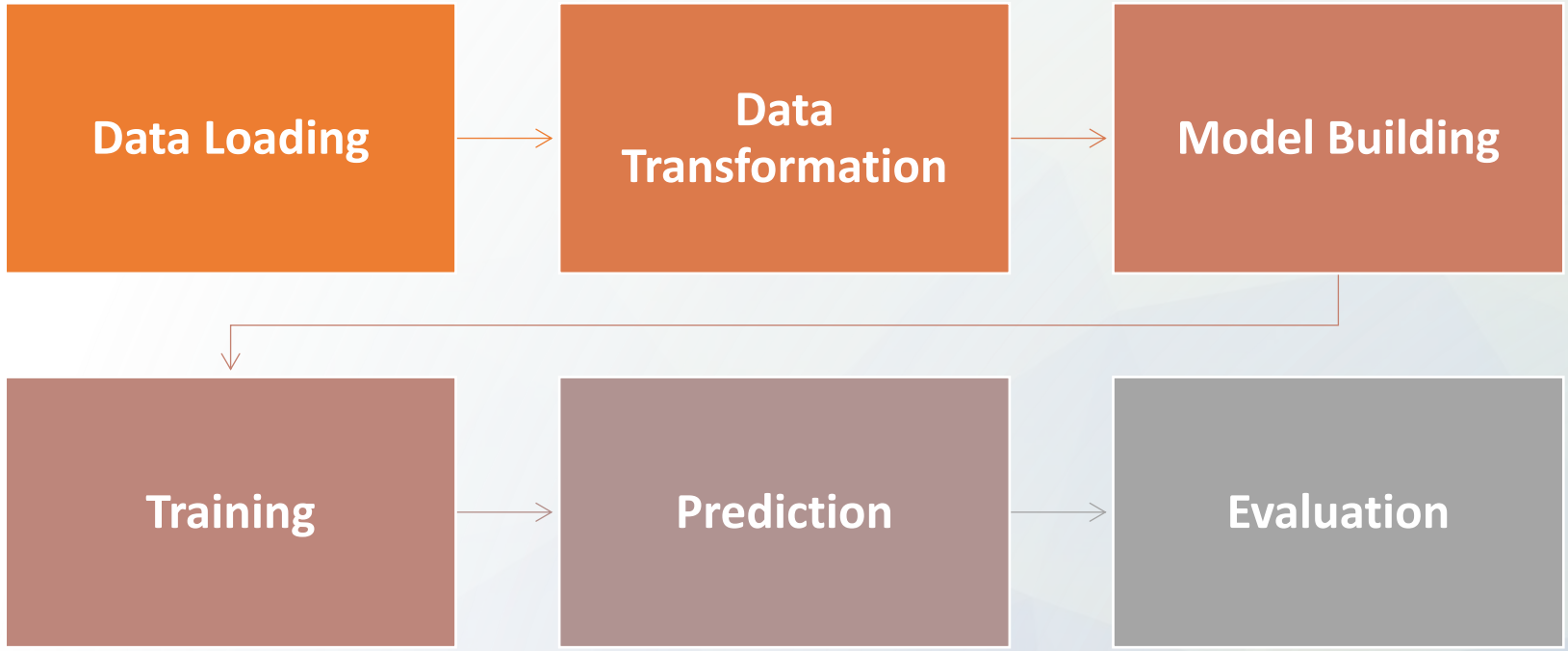
Average recommended courses per user: 3.959

Top-10 most frequently recommended courses:

	Course ID	Frequency
0	BD0101EN	341
1	DS0101EN	330
2	PY0101EN	287
3	ML0101ENv3	283
4	DS0105EN	237
5	BD0111EN	217
6	DS0103EN	213
7	CC0101EN	179
8	ST0101EN	161
9	DV0101EN	158

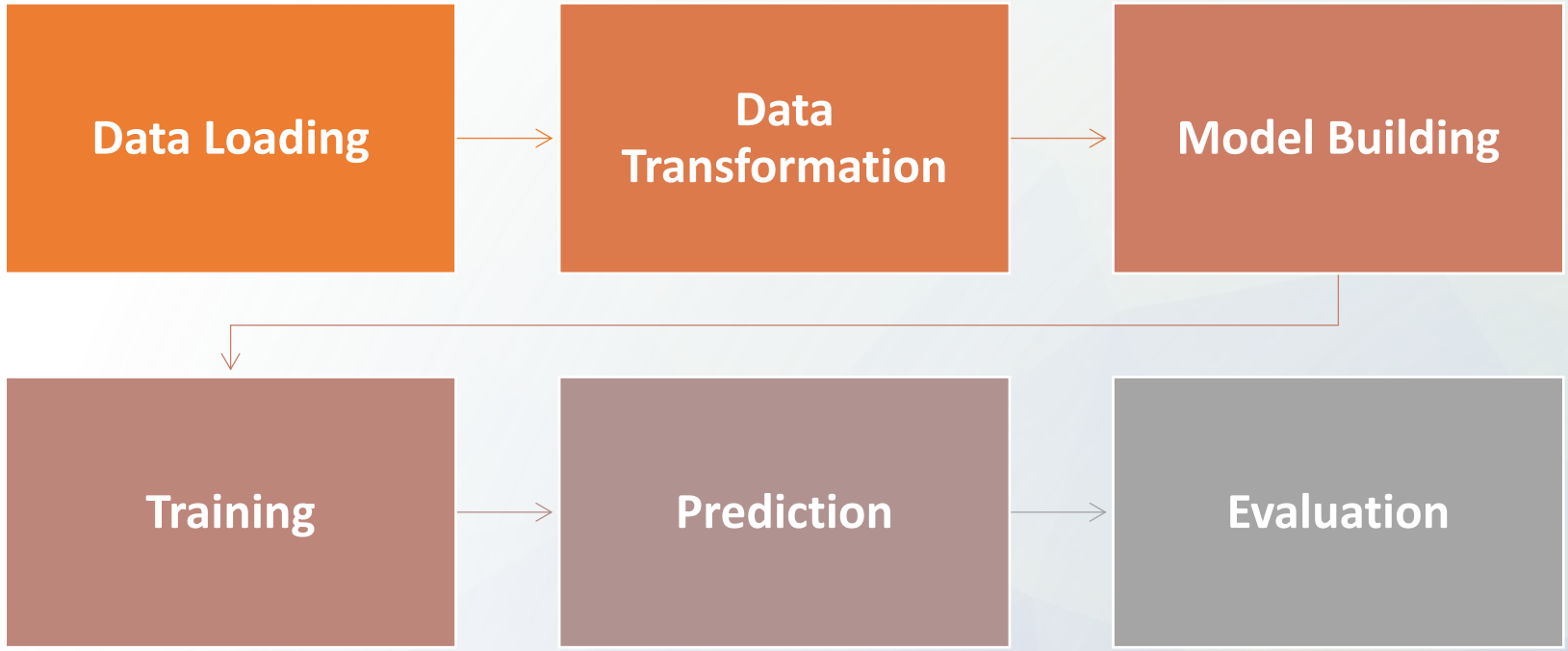
Collaborative-filtering Recommender System using Supervised Learning

Flowchart of KNN based recommender system



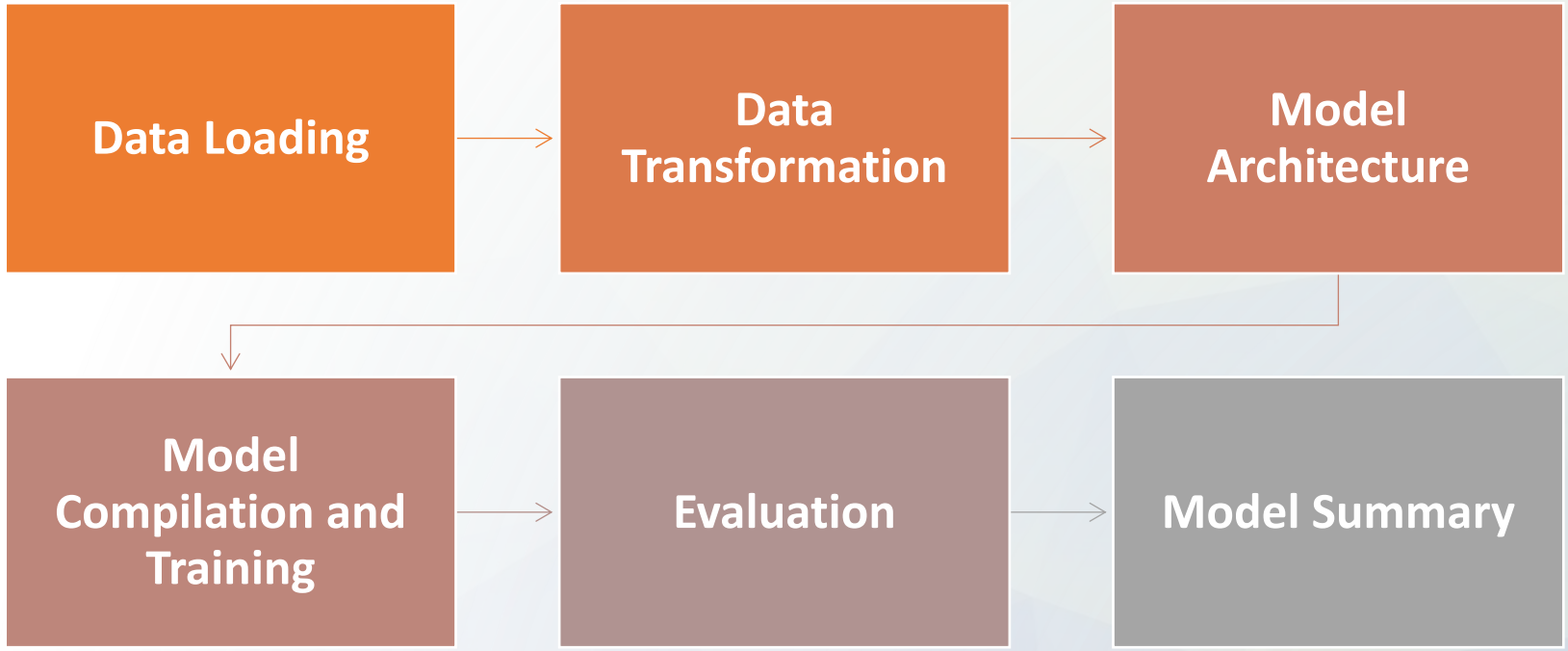
- **Data Loading**: The process begins by importing course enrollment data from a CSV file into a DataFrame.
- **Data Transformation**: The data is then transformed into a sparse matrix format, where users are rows, courses are columns, and the matrix values represent ratings.
- **Train-Test Split**: The dataset is divided into training and testing sets to facilitate model evaluation.
- **Model Building**: A KNN algorithm is employed for user-based collaborative filtering, leveraging cosine similarity to measure the closeness between users.
- **Prediction**: For each test user, the algorithm identifies the nearest neighbors (most similar users) who have rated the same courses and predicts ratings based on these neighbors' ratings, weighted by their similarity.
- **Evaluation**: Finally, the model's accuracy is assessed using the Root Mean Squared Error (RMSE), which quantifies the difference between the predicted and actual ratings.

Flowchart of KNN based recommender system



- **Data Loading**: The process begins by importing course enrollment data from a CSV file into a DataFrame.
- **Data Transformation**: The data is transformed into a sparse matrix format, where users are rows, courses are columns, and the matrix values represent ratings.
- **Model Building**: An NMF model is initialized with specific hyperparameters to factorize the matrix into latent user and item features.
- **Training**: The NMF model is trained on the training set to learn the relationships between users and items.
- **Prediction**: Using the trained model, predictions are made for the ratings in the test set.
- **Evaluation**: The model's accuracy is assessed using the Root Mean Squared Error (RMSE), which quantifies the difference between the predicted and actual ratings.

Flowchart of KNN based recommender system



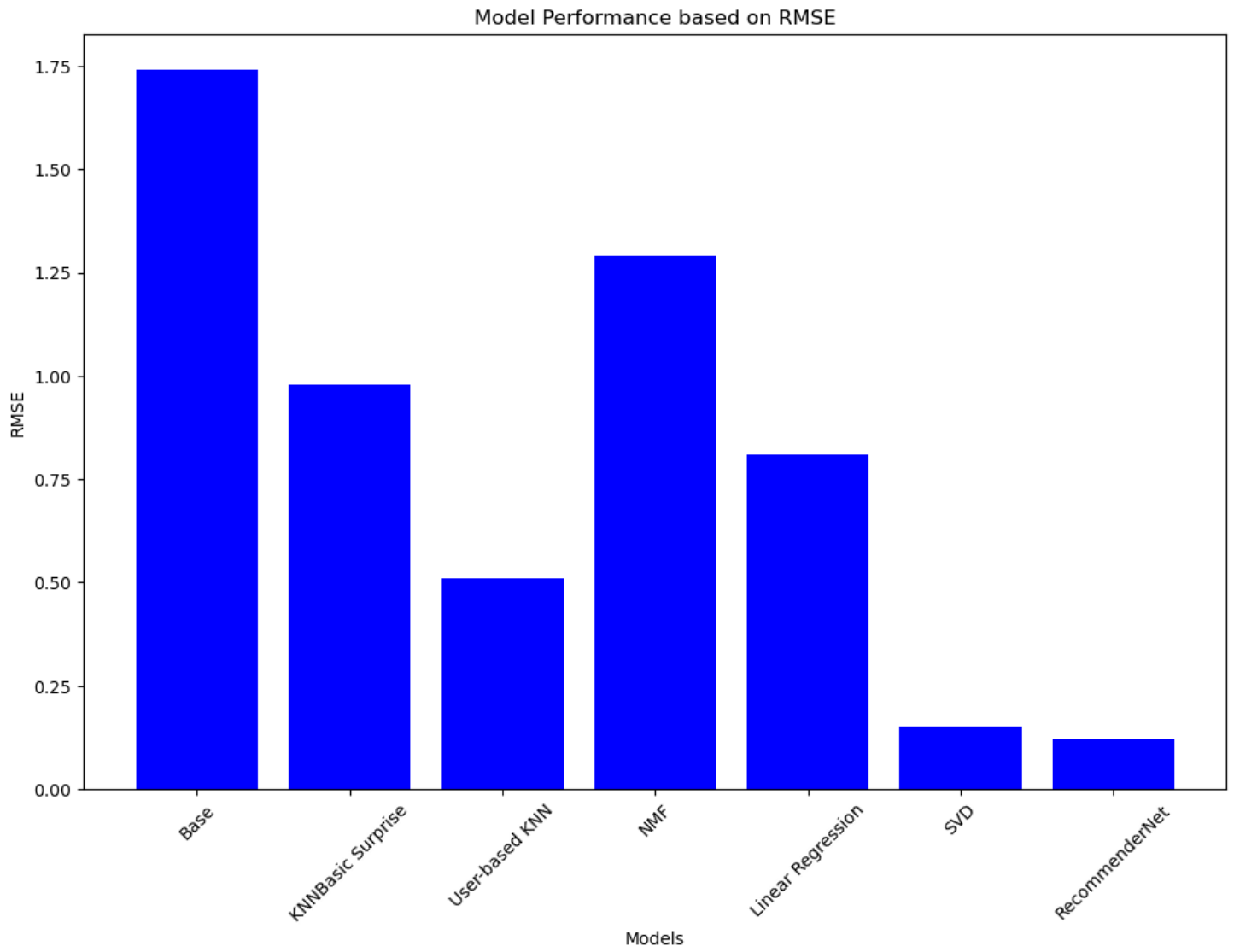
- **Data Loading**: The process begins by importing course enrollment data from a CSV file into a DataFrame.
- **Data Transformation**: The data is transformed into a sparse matrix format, where users are rows, courses are columns, and the matrix values represent ratings.
- **Model Architecture**: A custom RecommenderNet neural network model is defined, incorporating user and item embeddings, biases, and potential additional hidden layers for complexity.
- **Model Compilation and Training**: The model is compiled with a binary cross-entropy loss function and trained using the Adam optimizer, assessing performance with metrics such as accuracy, AUC, and RMSE during training and validation.
- **Evaluation**: The trained model is evaluated on a test set to calculate performance metrics like RMSE and AUC, aiding in understanding model effectiveness.
- **Model Summary**: A summary of the model is provided to review its architecture and parameter settings, ensuring a comprehensive understanding of the deployed neural network.

Compare the performance of collaborative-filtering models

Model Performance (RMSE)

- RecommenderNet: 0.12 (Best)
- SVD: 0.15
- User-based KNN: 0.51
- Linear Regression: 0.81
- KNNBasic Surprise: 0.98
- NMF: 1.29
- Base: 1.74 (Worst)

RecommenderNet and SVD perform best, while Base and NMF have the highest RMSE scores, indicating poorer performance.



Conclusion

- In this project, we built a customized online course recommender system using machine learning algorithms
- Our system enhances the learners' experience by helping them quickly discover new courses of interest and providing a more structured learning path
- We leveraged both content-based and collaborative-filtering methods
- Our results show that the recommender system improves the ability of learners to find new courses of interest and enhances the overall learning experience
- Our hypotheses were confirmed, and we believe that increased learner-course interaction facilitated by the recommender system will lead to increased company revenue

Harnessing Machine Learning for Personalized Online Course Recommendations

- Our innovative recommender system leverages both content-based and collaborative filtering methods to provide personalized course suggestions
- Content-based approach uses course genres and user profiles to identify relevant courses, while collaborative filtering analyzes enrollment patterns to recommend popular courses within user clusters
- Neural network model (RecommenderNet) and matrix factorization (SVD) achieved the best performance, significantly outperforming baseline approaches
- Hybrid approach combines the strengths of multiple algorithms to deliver highly relevant, tailored recommendations to each learner
- Empowers learners to efficiently discover courses of interest and provides structured learning pathways to enhance engagement and course completion rates

Advancing the AI Training Room Recommender System

Potential areas to explore further:

- Integrate additional data sources (e.g. course descriptions, learner demographics) to enable even more personalized and context-aware recommendations
- Experiment with deep learning architectures (e.g. RNNs, transformers) to capture more complex user-item interactions and sequential dependencies
- Develop a real-time recommender system that updates suggestions based on a learner's evolving interests and progress through courses
- Conduct A/B testing to quantify the impact of recommendations on key business metrics like course enrollments, completion rates, and revenue
- Expand the system to provide cross-platform recommendations (e.g. suggesting relevant blog articles, webinars to supplement course content)
- Productionize the recommender system and integrate it into the AI Training Room platform for real-world deployment and continuous improvement

Appendix

Firstly, I would like to express my deepest gratitude to the reviewers who have taken the time to provide valuable feedback and insights. Your expertise and suggestions have significantly contributed to the improvement of this project. Your efforts are greatly appreciated.

For those interested in exploring more of my work, I invite you to visit my GitHub profile at https://github.com/FitzChivhor. Here, you will find a collection of my projects, codes, and contributions to the coding community. I am always open to collaboration and feedback, so feel free to reach out.