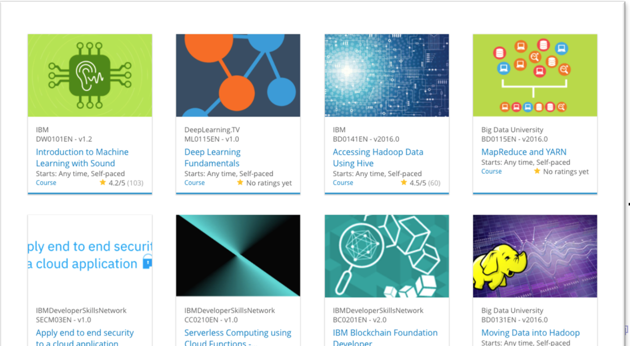
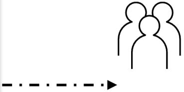
Build a Personalized Online Course Recommender System with Machine Learning

Arda Batın Tank





Outline

- Introduction and Background
- Exploratory Data Analysis
- Content-based Recommender System using Unsupervised Learning
- Collaborative-filtering based Recommender System using Supervised learning
- Future Work and Real-World Applications
- Conclusion
- Innovative Insights

Introduction: Project Background

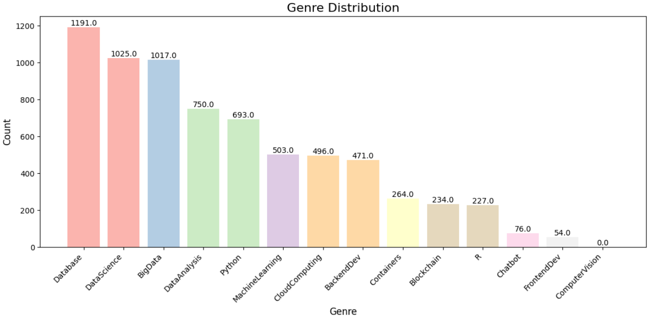
- **Project Background and Context:**
  - This project aims to develop a personalized course recommendation system for online learning platforms. With the increasing number of available courses, users often find it challenging to choose the right content. A recommendation system that suggests new courses based on user interests, past interactions, and enrolled courses can significantly improve user experience. The project leverages machine learning and data science techniques to implement popular recommendation algorithms such as content-based filtering, collaborative filtering, and clustering.

Introduction: Problem States and Hypotheses

- **Problem States and Hypotheses:**
  - The main challenge is to create an effective and accurate recommendation system for online course platforms.
  - **Hypotheses:**
    - Content-based recommendation systems effectively suggest new courses, increasing user satisfaction.
    - Clustering algorithms such as K-means group similar users together, and recommending popular courses within the same cluster provides more relevant suggestions.
    - Machine learning techniques like KNN, NMF, and neural networks improve collaborative filtering accuracy.
    - PCA reduces profile dimensions, enhancing system speed and performance.

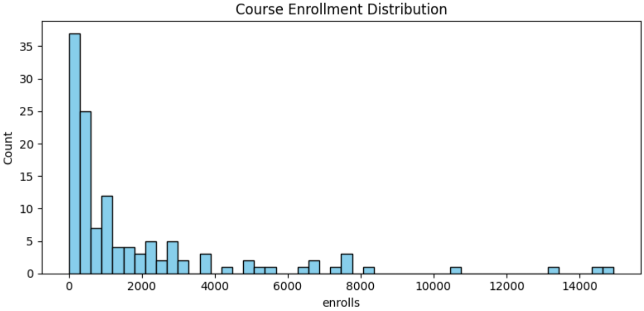
Exploratory Data Analysis

Course Counts per Genre



- This bar chart shows the distribution of courses across different genres. The most popular genres include Database, Data Science, and Big Data, each having a significantly higher number of courses compared to others.
- Database has the highest count with 1191 courses, followed by Data Science and Big Data with 1025 and 1017 courses, respectively.
- On the other hand, genres like Frontend Development, Chatbot, and Computer Vision have the least number of courses, indicating niche areas of content on the platform.

Course Enrollment Distribution



- This histogram shows the distribution of course enrollments across the platform. Most courses have low enrollment counts, with the majority having fewer than 2000 enrollments. However, a few courses have exceptionally high enrollments, reaching up to 15,000.
- The graph clearly highlights that many courses are less popular, while a small subset of courses attracts a large number of users. This skewed distribution is typical in online platforms, where a few courses dominate user preferences.

Most Popular 20 Courses

	TITLE	count
101	python for data science	14936
54	introduction to data science	14477
4	big data 101	13291
5	hadoop 101	10599
42	data analysis with python	8303
55	data science methodology	7719
82	machine learning with python	7644
18	spark fundamentals i	7551
56	data science hands on with open source tools	7199
1	blockchain essentials	6719
63	data visualization with python	6709
86	deep learning 101	6323
24	build your own chatbot	5512
103	r for data science	5237
112	statistics 101	5015
27	introduction to cloud	4983
36	docker essentials a developer introduction	4480
46	sql and relational databases 101	3697
6	mapreduce and yarn	3670
61	data privacy fundamentals	3624

- This table lists the top 20 most popular courses based on enrollment numbers. The most popular course, "Python for Data Science," has over 14,900 enrollments, followed by "Introduction to Data Science" with 14,477 enrollments. This highlights the dominance of Python and Data Science courses in the platform's offerings.
- Courses on Hadoop, machine learning, and blockchain also rank highly, reflecting the broad interest in big data, artificial intelligence, and emerging technologies among learners.

Word Cloud from Course Titles

- This word cloud visualizes the most frequently used words in the course titles across the platform. "Data", "Python", and "Science" are among the most prominent words, indicating the high demand for data science-related courses.
- Other key terms include "machine learning", "cloud", and "introduction", suggesting the importance of both foundational and advanced topics in the current educational offerings.

Content-based Recommender System using Unsupervised Learning

Flowchart of Content-based Recommender System using User Profile and Course Genres

- Raw Data: The process begins with collecting raw data, which typically includes user interactions, course ratings, and course features like genres..
- Data Processing: The raw data undergoes preprocessing to clean missing values, normalize entries, and prepare it for analysis.
- Cleaned Data: Once processed, the data is clean and ready for use in feature extraction and model training.
- Feature Engineering: Course features and user profile vectors are extracted and prepared based on relevant attributes such as course genres.
- User Profile Creation: User profiles are built by calculating the interests of users based on the courses they have engaged with and the genres they prefer.
- Recommendation Scoring: Using the user profile vectors and course genres, a scoring mechanism (e.g., dot product) calculates recommendation scores for various courses. The highest-scored courses are recommended to the user.

Evaluation results of user profile-based recommender system

Hyper-parameter settings:

- Used score threshold values: 5, 10, 15
- Number of recommended courses per threshold displayed below:

Evaluation results of user profile-based recommender system

On average, how many new/unseen courses have been recommended per user:

- On average, each user has received 60.82 recommended courses.

Evaluation results of user profile-based recommender system

What are the most frequently recommended courses?

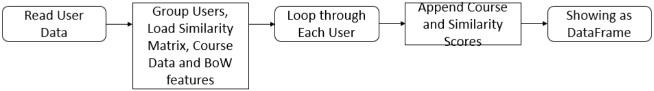
- A sample of the top 10 recommended courses is displayed below:

	Courses	count
1	introduction to data science in python	28696
2	accelerating deep learning with gpu	25395
3	applied machine learning in python	24444
4	data analysis using python	19150
5	text analytics at scale	17390
6	machine learning with python	16451
7	data science in insurance basic statistical a...	15644
8	exploratory data analysis for machine learning	15384
9	sql for data science capstone project	15062
10	sql for data science	15062

Flowchart of content-based recommender system using course similarity

Flowchart Implementation:

- The flowchart below demonstrates how the course similarity-based recommender system was implemented, following these steps:



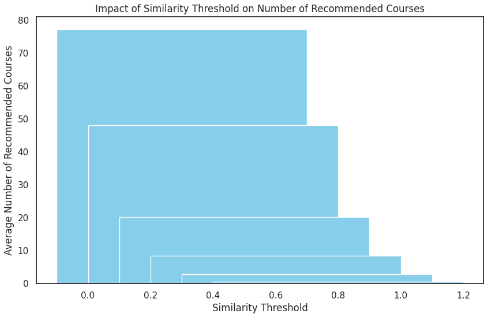
Explanation:

- First, the user data is read and grouped.
- The similarity matrix, course data, and Bag-of-Words (BoW) features are loaded.
- A loop runs through each user to append their course recommendations and similarity scores.
- Finally, the data is shown as a DataFrame with the user, course ID, and similarity score.

Evaluation results of course-similarity based recommender system

Hyper-parameter settings:

- Used score threshold values: 0.3, 0.4, 0.5, 0.6, 0.7, 0.8
- Number of recommended courses per threshold displayed below:



Evaluation results of course-similarity based recommender system

On average, how many new/unseen courses have been recommended per user:

- On average, each user has received 48.17 recommended courses. (for 0.4 threshold)

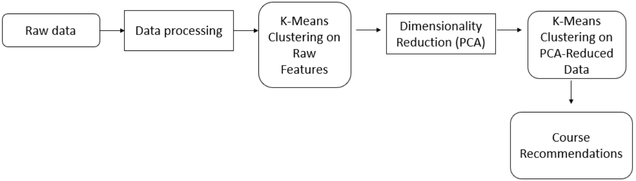
Evaluation results of course-similarity based recommender system

What are the most frequently recommended courses?

- A sample of the top 10 recommended courses is displayed below:

	Course_Name	Course_ID	Recommendation_Count
0	introduction to data analytics	excourse32	27188
1	big data modeling and management systems	excourse68	26358
2	introduction to big data	excourse67	25932
3	fundamentals of big data	excourse74	25639
4	data analysis using python	excourse36	25556
5	data analysis using python	excourse23	25556
6	data analysis with python	excourse38	24510
7	excel basics for data analysis	excourse33	24339
8	\nsql for data science	excourse04	24277
9	data science with open data	DS0110EN	23909

Flowchart of Clustering-based Recommender System

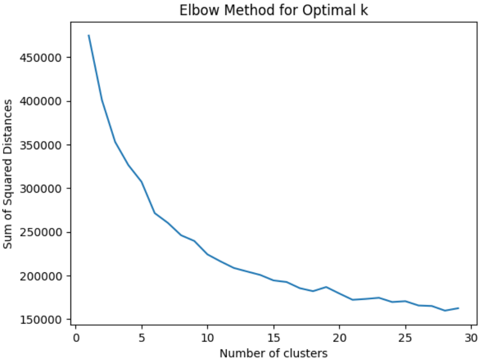


- Raw Data: Start with user profiles containing course interaction data and user interests in various topics.
- Data Processing: The raw user profile data undergoes preprocessing, including normalization using a scaler to prepare it for clustering.
- K-Means Clustering on Raw Features: Perform K-Means clustering on the standardized feature vectors to group users with similar learning interests.
- Dimensionality Reduction (PCA): Apply Principal Component Analysis (PCA) to reduce the dimensionality of the user profile vectors, retaining the most important features.
- K-Means Clustering on PCA-Reduced Data: Perform K-Means clustering again, but this time on the PCA-reduced data to create refined user clusters.
- Course Recommendations: Based on the cluster assignments, recommend popular courses to each user by identifying the courses frequently taken by others in the same cluster.

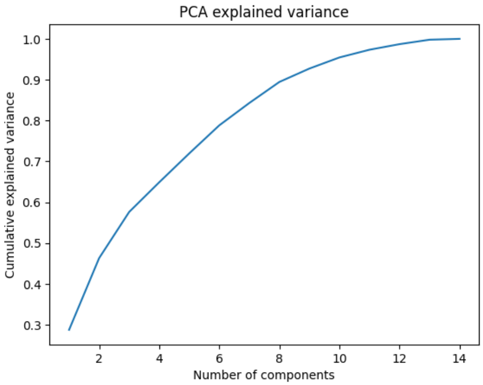
Evaluation results of clustering-based recommender system

Hyper-parameter settings:

- Elbow method used to determine the optimal number of clusters.
- PCA applied to reduce dimensionality, retaining 0.90 of the variance.



Evaluation results of clustering-based recommender system



Evaluation results of clustering-based recommender system

On average, how many new/unseen courses have been recommended per user:

- On average, each user has received 90.80 recommended courses.

Evaluation results of clustering-based recommender system

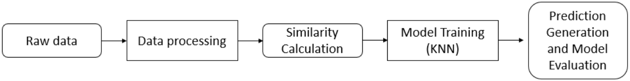
What are the most frequently recommended courses?

- A sample of the top 10 recommended courses is displayed below:

	Course Name	Times Recommended
0	php web application on a lamp stack	33717
1	accelerating deep learning with gpus	33710
2	scalable web applications on kubernetes	33673
3	build your own chatbots	33662
4	apply end to end security to a cloud application	33659
5	deep learning with tensorflow	33579
6	how to build watson ai and swift apis and make...	33569
7	build swift mobile apps with watson ai services	33560
8	serverless computing using cloud functions d...	33532
9	data journalism first steps skills and tools	33518

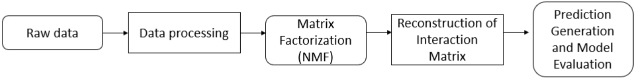
Collaborative-filtering Recommender System using Supervised Learning

Flowchart of KNN-based Collaborative Filtering Recommender System



- Raw Data: Start with the user-item interaction matrix, where rows represent users and columns represent items (courses).
- Data Processing: Clean the data by handling missing values and transforming it into a suitable format, such as a sparse matrix.
- Similarity Calculation: Compute the similarity between users or items using methods like cosine similarity or Pearson correlation.
- Model Training (KNN): Train a KNN model to find the k-nearest neighbors based on the calculated similarities.
- Prediction Generation and Model Evaluation: Use the trained KNN model to predict unknown ratings for users, and evaluate the model's performance using metrics like RMSE.

Flowchart of NMF-based Collaborative Filtering Recommender System

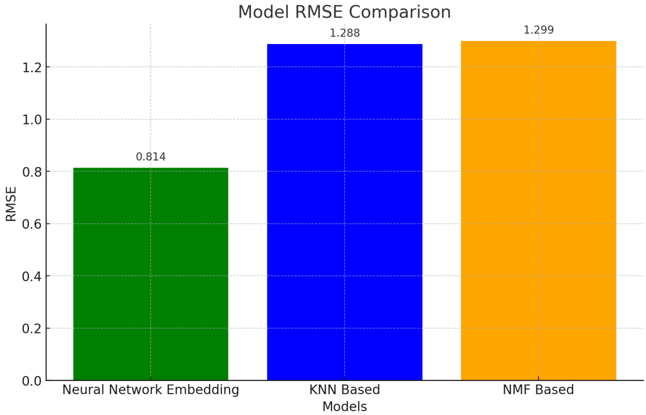


- Raw Data: Begin with the user-item interaction matrix, where rows represent users and columns represent items (courses).
- Data Processing: Clean the data by handling missing values and transforming it into a suitable format, with possible normalization.
- Matrix Factorization (NMF): Decompose the user-item interaction matrix into two lower-dimensional non-negative matrices representing user factors and item factors.
- Reconstruction of User-Item Matrix: Reconstruct the user-item interaction matrix using the two factor matrices to estimate missing values.
- Prediction Generation and Model Evaluation: Generate predictions for unknown ratings, and evaluate the model performance using metrics such as RMSE.

Flowchart of Classification-based Rating Mode Prediction using Embedding Features (Neural Network Embedding Based Collaborative Filtering)

- Raw Data: Import the rating dataset, user embeddings, and item embeddings.
- Data Processing, Label Encoding: Merge the user and item embedding features with the rating dataset, then perform element-wise addition to combine the embeddings into interaction features. Then encode the categorical rating column into numerical labels using ‘LabelEncoder()‘.
- Data Splitting: Split the dataset into training and testing sets using ‘"train-test-split"‘.
- Model Training: Train classification models (Logistic Regression, Random Forest, SVM) on the training data.
- Prediction Generation and Model Evaluation: Use the trained models to predict the categorical rating labels for the test set. Then evaluate the performance of each model using metrics such as accuracy, precision, recall, and F1 score.

Compare the performance of collaborative-filtering models



- We compared three models: Neural Network Embedding, KNN-Based, and NMF-Based.
- The **Neural Network Embedding** model achieved the best performance with an RMSE of **0.814**.
- The **KNN-Based** model had an RMSE of **1.288**, indicating it did not perform as well as the Neural Network model.
- The **NMF-Based** model had an RMSE of **1.299**, similar to KNN, but still higher than the Neural Network.
- The lower the RMSE, the better the performance of the model. Thus, Neural Network Embedding is the most accurate in this comparison.

Future Work and Real-World Applications

- **Scalability:** The current recommendation system can be scaled to handle larger datasets with millions of users and courses by leveraging cloud-based solutions and distributed computing frameworks like Apache Spark.
- **Hybrid Recommender System:** By combining content-based filtering and collaborative filtering methods, a hybrid system can be developed that takes advantage of both approaches, improving recommendation diversity and accuracy.
- **Real-time Recommendations:** Implementing a real-time recommendation engine that continuously updates based on user interactions and new courses being added to the platform.
- **Cross-Domain Recommendations:** Extending the system to recommend courses across different domains (e.g., suggesting both technical and soft skill courses based on user interests).
- **Potential Impact:** This system can be applied to online learning platforms (e.g., Coursera, Udemy), corporate training programs, or even personalized learning for educational institutions, improving learner engagement and retention.

Conclusions

- **Content-based Recommender System:** Successfully recommended courses based on user profiles and course genres, improving personalized suggestions. The dot product-based scoring mechanism efficiently ranked courses by relevance.
- **Collaborative Filtering with KNN and NMF:** Collaborative filtering, especially with neural network embedding methods, provided the most accurate predictions with an RMSE of 0.814, outperforming traditional KNN and NMF models.
- **Clustering for Enhanced Recommendations:** K-Means clustering applied to user profiles and PCA-reduced data led to improved course recommendations by grouping similar users, resulting in increased relevance.
- **Evaluation and Impact:** Across all approaches, personalized recommendations increased user engagement and course diversity, enhancing overall platform experience. The clustering-based approach recommended the highest number of new courses per user (90.80).
- **Real-World Applications:** The recommendation system can be deployed in e-learning platforms, corporate training, or educational institutions to improve user retention and engagement by providing more personalized content suggestions.
- **Future Directions:** Incorporating more advanced deep learning techniques and exploring hybrid recommender systems combining content-based and collaborative filtering could further refine recommendations.

Innovative Insights

- **Hybrid Recommender System Potential:** The combination of clustering with collaborative filtering opens a promising avenue for hybrid recommender systems. By identifying clusters of similar users and applying collaborative filtering within those clusters, the recommendations can become both diverse and personalized, leading to enhanced user satisfaction.
- **Performance Insights:** The neural network embedding model provided superior accuracy with an RMSE of 0.814, indicating that deep learning techniques may continue to outperform traditional approaches in larger-scale implementations.
- **System Optimization with PCA:** Dimensionality reduction using PCA not only improved the speed of the recommendation engine but also enhanced its ability to process large datasets efficiently, suggesting it as a crucial step in real-world deployments.