

Coursera course Recommendation system



Santosh kumar velgapuri
YS79974
Prof : Chaojie Wang

Agenda



- **INTRODUCTION**
- **ABOUT DATA**
- **PROJECT WORKFLOW**
- **DATA PREPROCESSING**
- **MODELING AND RESULTS**
- **DEPLOYMENT**

INTRODUCTION



coursera

- Our Course Recommendation System delivers personalized course suggestions to users on platforms like Coursera, enhancing user experience and maximizing learning outcomes through targeted recommendations.
- By leveraging data analysis, machine learning and Natural language processing algorithms, we streamline course discovery, empowering users to engage in lifelong learning and skill development by introducing recommendation based on precedence.

ABOUT DATA



coursera

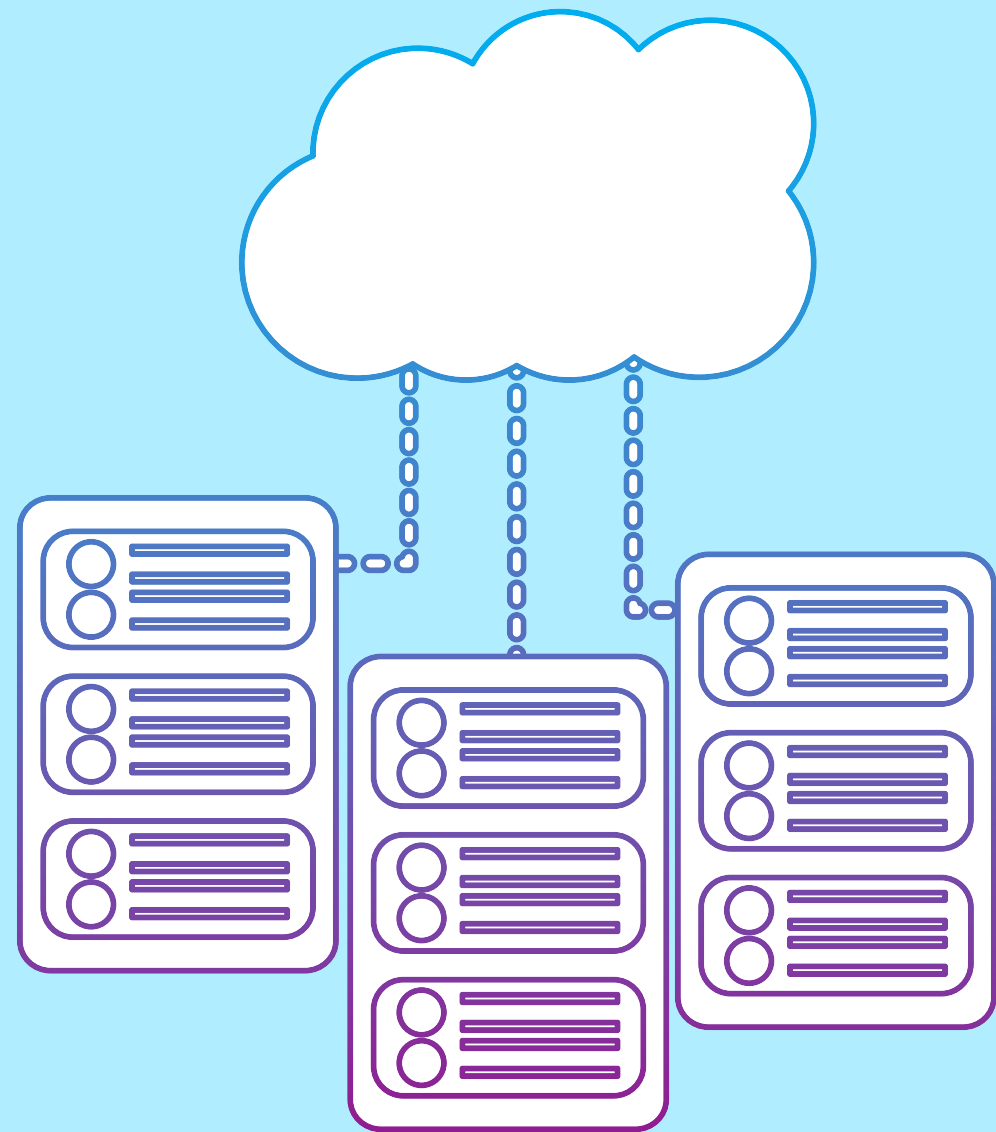
- **Data source:** Kaggle & Cousera
<https://www.kaggle.com/datasets/elvinrustam/coursera-dataset>
- **Data size:** 52.8 MB
- **Data shape:** 8370 rows x 13 columns
- **Columns Used:** Course Title, What will you learn, Skill gained, Keyword, level, Rating.
- **Data Types:** Object, Float64, Int64
- **Definition:** The above cloumns will be used to find the similarity tags which helps the user to find the similar courses for his field in a priority list.

PROJECT WORKFLOW



- The steps involved in building the recommendation system
 - Importing Libraries
 - Loading and Analyzing Data
 - Data Pre-Processing
 - Text Vectorization
 - Stemming Process
 - Similarity Measure
 - Recommendation Function
 - GUI Deployment

DATA PREPROCESSING



- **Importance of data preprocessing** : Essential for improving the accuracy and efficiency of the recommendation model by ensuring that the input data is clean, relevant, and standardized.
- **Steps involved in data preprocessing:**
 - Cleaning data
 - Handling missing values
 - Feature selection
 - Text preprocessing (Text vectorization, stemming, etc.)

MODELING

1. **Model Overview:** Uses a content-based filtering approach, combining course attributes into a single "tags" feature for recommendation.
2. **TF-IDF Vectorization:** Converts text data into numerical vectors based on term importance, balancing term frequency and document rarity.
3. **Cosine Similarity:** Measures similarity as the cosine of the angle between course vectors, where values closer to 1 indicate higher similarity.
4. **Recommendation Function:** Accepts a course title, finds its vector, computes similarity scores with all courses, and outputs the most similar courses.

RESULTS

- **Cosine Similarity Scores** for the recommendation function to measure how similar the recommended courses are to the query.

1	<code>recommend('Pixel Art for Video Games')</code>
Concept Art for Video Games	
Low Poly Art For Video Games	
Character Design for Video Games	
Art for Games Specialization	
World Design for Video Games	
Art and the Digital Transformation	

The image is a collage of screenshots from a Python GUI application titled 'Course Recommendation System'. The main window features a search bar containing the text 'data Science' and a button labeled 'Get Recommendations'. Below the button, the application displays two lists of recommendations: 'Top Recommendations' and 'Other Recommendations'. The 'Top Recommendations' list includes: 1. Unordered Data Structures, 2. Statistics for Genomic Data Science, 3. mercadeo digital Specialization, 4. Introduction to Relational Databases, 5. Health Data Science Foundation, 6. Data Science Ethics, and 7. Data Manipulation at Scale Systems and Algorithms. Other screenshots show an 'Input needed' dialog box with the message 'Please enter a course title to get recommendations.', a search for 'jbfuwg' resulting in a 'No match' error dialog box with the message 'No courses found with that title. Try another search.', and a 'coursera' logo.

- 

The background is a dark teal color with a complex, glowing network of white lines and dots. The lines vary in thickness and brightness, creating a sense of depth and connectivity. The dots are small and bright, acting as nodes in the network. The overall effect is a high-tech, digital aesthetic.

THANK YOU