Evolution of Course Management System

Author Names :- Shivani Rao, Konstantin Salomatin, Gungor Polatkan, Mahesh Joshi, Sneha Chaudhari,Vladislav Tcheprasov, Jeffrey Gee, Deepak Kumar.

Year of Publication:- 2019

Published in :- CIKM'19

Weblink/DOI:- https://dl.acm.org/doi/10.1145/3357384.3357817

Group-14

Yash Nirmal(201IT168)
Aditya Rama Hegde(201IT105)
Buddha Teja(201IT115)
L.Koushik(201IT131)
Vissampalli Balaji(201IT267)

Abstract

The evolution of course recommendation systems has been a topic of interest in recent years due to the increased demand for personalized learning experiences. This paper reviews the development of course recommendation systems from their early stages to their present state. The first systems were based on simple rule-based algorithms, while more recent systems have incorporated machine learning techniques such as collaborative filtering, content-based filtering, and hybrid filtering. The paper also discusses the challenges faced in the development of these systems, including the cold-start problem, scalability and sparsity. Finally, the paper highlights some of the emerging trends in course recommendation systems, such as the use of deep learning techniques, natural language processing, and the use of profile data.

The paper discusses the current state of course recommendation system on websites that sell courses, specifically related to LinkedIn Learning. The paper will discuss how improvements on the currently existing algorithm can be done and would be helpful for the users. From both the algorithmic and infrastructure points of view, the paper talks about the difficulties encountered and the solutions chosen. We will use different parameters and their relevance for recommendation. We will compare the similarity between the user's course taking and exploring behavior to improve the recommendation. We will also address the cold start problem for new user signing on the platform and try to solve it using the best fit approach.

Introduction

Online learning platforms like Linkedin Learning, Coursera and Udemy have thousands of courses and each year more are added to satidfy to the demand of students and professionals to learn various skills in a wide range of categories. There is a requirement to filter these thousands of courses into the ones the user would like to use. Additionally existing courses are recommended on the basis of ratings alone and not how each course will help users with different cognitive abilities.

This helps the platform to sell the right courses to users both existing and new, thereby increasing the audience. That is why course recommendation algorithms are an essential part of these educational platforms.

In this paper we are going to study the evolution of course recommendation algorithms used in these platforms over time and try to overcome the engineering and modeling challenges while building the course recommender system.A few of these challenges are as follows:-

i) One of the very first challenges we face is the classic 'cold start' problem which is quite common in these types of recommender systems. Here we don't have enough data on new users and new courses to train the model in its initial stages. Also, initially our system has limited content understanding capabilities to map the courses to entities like skill, etc. We will be using algorithms that work around these limitations in our paper.

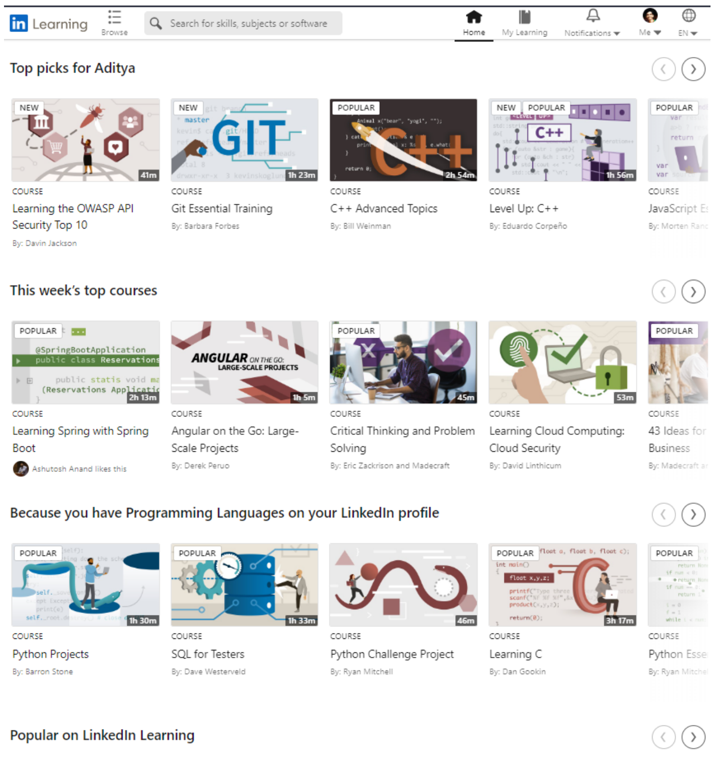
ii) After acquiring the initial member interaction data we will be able to train supervised models that will predict if the user will want to buy the given course. Despite having the necessary data to train predictive models, it will be a challenge to create an efficient and scalable system for modeling and analyzing member interaction data.

iii) After creating a supervised training platform and collecting large amounts of member engagement data we will focus on building more complex time efficient models that capture user-level interests and personalize the recommendations using mixed effects models with multiple coefficients.

Our model will give a list of recommended courses customized according to the user data.

Problem Statement

To design and develop an effective course recommendation system that can provide personalized and relevant course recommendations to learners on an online learning platform based on the different parameters.



Top picks for Aditya

NEW · 41m · COURSE · Learning the OWASP API Security Top 10 · By: Davin Jackson
NEW · 1h 23m · COURSE · Git Essential Training · By: Barbara Forbes
POPULAR · 2h 54m · COURSE · C++ Advanced Topics · By: Bill Weinman
NEW · POPULAR · 1h 56m · COURSE · Level Up: C++ · By: Eduardo Corpeño
POPULAR · COURSE · JavaScript E · By: Morten Ranc

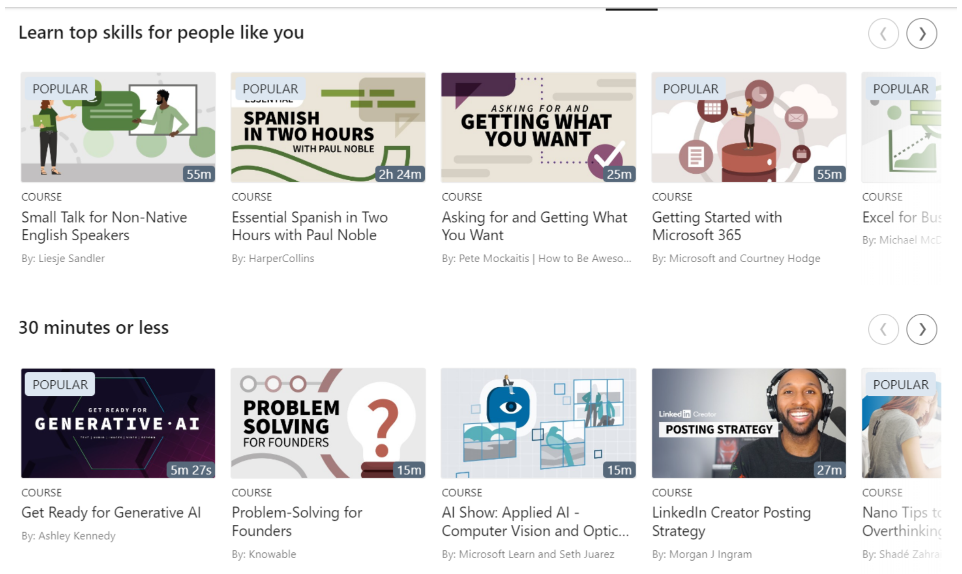
This week's top courses

POPULAR · 2h 13m · COURSE · Learning Spring with Spring Boot · Ashutosh Anand likes this
1h 5m · COURSE · Angular on the Go: Large-Scale Projects · By: Derek Peruo
POPULAR · 45m · COURSE · Critical Thinking and Problem Solving · By: Eric Zackrison and Madecraft
53m · COURSE · Learning Cloud Computing: Cloud Security · By: David Linthicum
POPULAR · COURSE · 43 Ideas for Business · By: Madecraft ar

Because you have Programming Languages on your LinkedIn profile

POPULAR · 1h 30m · COURSE · Python Projects · By: Barron Stone
POPULAR · 1h 33m · COURSE · SQL for Testers · By: Dave Westerveld
46m · COURSE · Python Challenge Project · By: Ryan Mitchell
POPULAR · 3h 17m · COURSE · Learning C · By: Dan Gookin
POPULAR · COURSE · Python Esse · By: Ryan Mitchel

Popular on LinkedIn Learning


Learn top skills for people like you

POPULAR
COURSE
Small Talk for Non-Native English Speakers
By: Liesje Sandler
55m

POPULAR
COURSE
Essential Spanish in Two Hours with Paul Noble
By: HarperCollins
2h 24m

COURSE
Asking for and Getting What You Want
By: Pete Mockaitis | How to Be Aweso...
25m

POPULAR
COURSE
Getting Started with Microsoft 365
By: Microsoft and Courtney Hodge
55m

POPULAR
COURSE
Excel for Bus
By: Michael McD

30 minutes or less

POPULAR
COURSE
Get Ready for Generative AI
By: Ashley Kennedy
5m 27s

COURSE
Problem-Solving for Founders
By: Knowable
15m

COURSE
AI Show: Applied AI - Computer Vision and Optic...
By: Microsoft Learn and Seth Juarez
15m

COURSE
LinkedIn Creator Posting Strategy
By: Morgan J Ingram
27m

POPULAR
COURSE
Nano Tips to Overthinking
By: Shadé Zahrai

LinkedIn Learning Interface

Dataset

The dataset will be a list of courses, skills these courses deal with, users interest, skills they want to acquire, skills they already have and skills that are required in the job they are applying for or are interested in.

Dataset Link :-
https://www.kaggle.com/datasets/khusheekapoor/coursera-courses-dataset-2021

The dataset also contains a course, their name, skills they help users to improve, their last updates date(showing they are still relevant or not), their reviews and rating, etc.

On the basis of overlap between these two major data , we can improve the course recommendation by many folds.

For new uses when they just come on the platform, no data is available for them, so we can go 2 ways from here

a. Recommend random courses and track on user’s behavior of what course they are opening and reading about
b. Ask the users for their interest and future plans through some kind of forms, and recommend courses based on that.

Methodology of the base paper

The methodology of the course recommendation project here includes mainly implementing two types of learning models.

Supervised and Unsupervised Modeling. For the new users and new learning platforms,there will be no previous data and history of the data available,so there we will be using the Unsupervised model to build our own model. For other experienced users and platforms we will be using Supervised Learning. At every phase our main focus is to predict whether a course C_i is suitable for the user U_j. Both the phases use the scoring of courses as a method of comparing and suggesting for the users. Also here we will be using Two modes of system,Offline and Online. All the ranking of the recommendations for each used is actually calculated in the offline mode and stored in online storage. Whenever a query is raised this will be fetched and top-scored courses will be recommended to the users.

Unsupervised Modeling for Course Recommendation:

Most of the Course Recommendation systems employed by most of the online Learning platforms depend upon behavioral data of the existing users in order to make recommendations more accurate and meaningful.That is when user activity and histories like browsing,viewing,clicking are available, it will provide implicit insights that can be used to train a supervised model or a collaborative filtering model. But for the new users or new platforms of learning such insights are unavailable and this creates a cold start problem.

Tagging the Courses with Skills

1) Via Supervised models

The model here learns to predict if a skill x is relevant to a course y. We here build a supervised binary classification model that will predict whether for a given (course, skill) pair of a course, the skill is relevant to the course or not. The labeled training data is derived by treating the human-tagged (course, skill) pairs as positive examples.The features used are binary, and indicate if a match exists between the skill and the course's textual components (description,title,categories, section names, video names etc.) based on the skills fetched by the skill tagger.

2) Via Semi-Supervised Learning

Here We will use only video transcripts of the course videos to derive various skills in the courses. Because they are more descriptive and Informative.We will store them as a key-value pair and then for every skill in the user's skills, we search for all the courses with those skills in it and add respective scores of it.

Supervised Modeling for Course Recommendation:

Mainly after the is a post launch of an app, the behavioral data that is generated(like the clicks by the user, bookmarks that are made, watch hours of the user, type of courses that are being chosen,etc.), along with the profile data like can be utilized in creating a scalable and efficient prediction algorithm for a user-based course recommendation system

Here the training data is stored offline(via date-versioned updation) and even the operations are performed offline and are connected with the interface using REST API, thus making it easier to add new features and learning algorithms for the features

As in the Unsupervised, the skills that are to be used in a course-to-skill specifications are collected based on the transcriptions of the course material descriptions and are not manually picked and are extracted directly from the course descriptions. And for the member-to-skill specifications, User’s profile data like the job interests,education,etc.. are used

But to gain a major advantage we use the behavioral data that is already available, We can make the best use of it by ;

Pre Processing

Out of all the labels, those that are best required are generated, like how correlated are the attributes like the user click details, watch hours, etc.. are analyzed(to eliminate the redundancy and complexity). Thus further employing feature engineering like the feature extraction, selection techniques for the better representation of the data.

Based on the strength of each label(like the number of clicks in the user click data,...) the score and ranks of each label is generated for a user (via the Map-Reduce approach) which are used in recommending the courses in a structured sequential manner

Training the Model

Though a Linear Regression on the learning of course-skill relationship(for a user) can be useful on effectively focusing just one particular skill it cannot keep up with the varying interests of the user(and might not address personalization in the course recommendation), thus we can use Generalized Linear Mixed(GLMix) effect models to address each attributes of course-skill and member-skill patterns

The predicted click probability in GLMix is expressed as sum of three components,global model, per-member model, and per-course model

g (E (click|m,c)) = (w^ᴛ) (xmc) + (α^ᴛ m)xc + (β^ᴛ c) xm

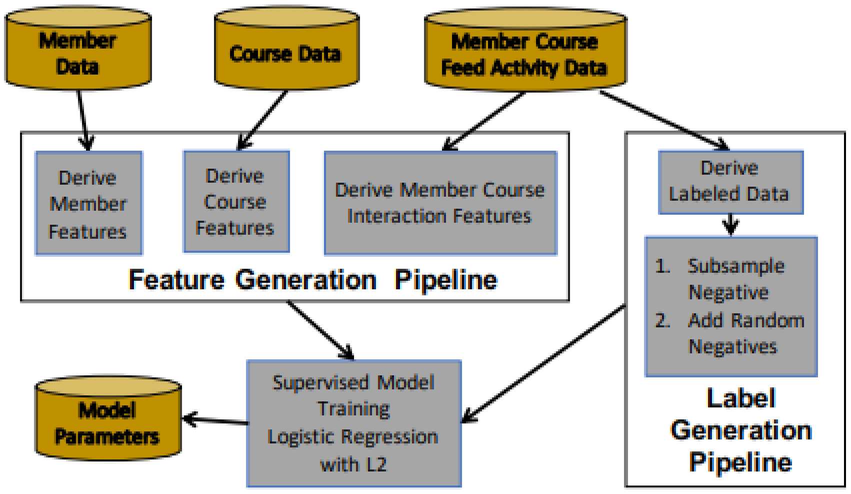
Where g - logit function with corresponding parameters

(x)mc - baseline for course,member and interaction features

2nd component represents per-member features like what did user click in the past his/her activity

the 3rd component represents the per-course feature like what type of courses did the user opt for..

The GLMix model has more parameters to tune and thus requires large data to use it and also becomes computationally harder than an LR(linear Regression) model.



Member Data
Course Data
Member Course Feed Activity Data
Derive Member Features
Derive Course Features
Derive Member Course Interaction Features
Feature Generation Pipeline
Derive Labeled Data
1. Subsample Negative
2. Add Random Negatives
Label Generation Pipeline
Model Parameters
Supervised Model Training Logistic Regression with L2

Figure 5: Supervised Training Pipeline

(Figure:from the research paper)

Website

The website is divided into 2 parts i) fronted and ii) Backend
i) Frontend
The front end of the website is made using a react framework called NextJs. Initially the frontend was going to be in VanillaJs but to increase the scalability and flexibility to add more features in the future works, we decided to go with NextJs. NextJs works in a way where we can make different components of a website separately and then combine them into a single place, which makes it highly flexible when making medium to big size applications. It also helps in increasing the code re-usability, which in turn makes the website load faster. For the styling purposes the website uses TailwindCSS, which is a popular open-source CSS framework that provides a comprehensive set of utility classes to style web applications. It is a highly flexible, developer friendly and performance based framework.It also makes it easy to make mobile responsive websites

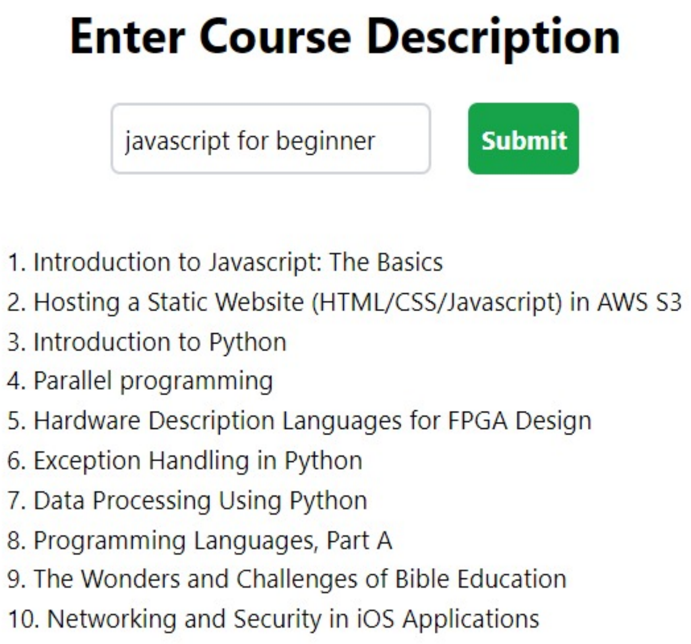
ii) Backend
The backend is made using the flask framework. It is a popular open-source web framework for building web applications using the Python programming language. We chose Flask because its a lightweight framework that allows us to build web applications faster and efficiently. Also working with machine learning models in flask with the help of python is very simple yet powerful.

Improvements and Implementation

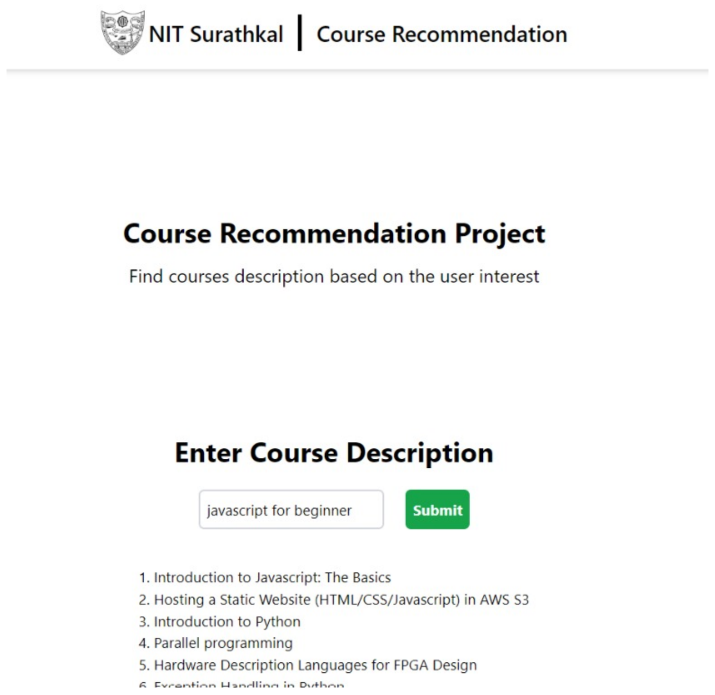
- The Improvements we have implemented for the Paper includes,

    1) Recommending Courses without User’s data (Unsupervised data - Only the course data):

        A website is designed using the flask in which when a user enters a course description (which can be anything from a single word to very large paragraphs) then the course decription’s score is checked against all the skills that are present in the dataset(based on the cosine similarity of the User entered course description and the course content) and then the top ten courses are displayed to the User as the recommendations.



1. Introduction to Javascript: The Basics
2. Hosting a Static Website (HTML/CSS/Javascript) in AWS S3
3. Introduction to Python
4. Parallel programming
5. Hardware Description Languages for FPGA Design
6. Exception Handling in Python
7. Data Processing Using Python
8. Programming Languages, Part A
9. The Wonders and Challenges of Bible Education
10. Networking and Security in iOS Applications


NIT Surathkal | Course Recommendation

Course Recommendation Project

Find courses description based on the user interest

Enter Course Description

javascript for beginner

Submit

1. Introduction to Javascript: The Basics
2. Hosting a Static Website (HTML/CSS/Javascript) in AWS S3
3. Introduction to Python
4. Parallel programming
5. Hardware Description Languages for FPGA Design
6. Exception Handling in Python

2) Recommend Courses based on Users Profile skills

User’s profile data like the skills he has can be learnt using a form-feed is collected which is a common practice that is observed, to give users more sophisticated course recommendations. In here the User skills are considered and these skills are used in calculating the scores(which indicates how strong a skill is related to a course, which is calculated by using the cosine similarity between the course description and user skills) of the courses and the first ten courses are recommended

3) Recommended Top Courses as per your Company

We have considered another feature which recommends courses based on the User’s organization/company using his profile details(like the people and Organizations he/she is following). Like the Skills related to the people of that particular organization or society are extracted and the courses(in the dataset) that are highly related to these skills are recommended to the User Beforehand as he might be looking for the same

4) Recommended Top Rated and most Viewed Courses

The courses which are top rated and top viewed are mostly very valuable and knowledgeable. Hence these are noted down in a dataset at regular intervals and then when requested by the user for the top view courses, these are recommended.

5) Recommended Courses based on difficulty Levels

According to the difficulty requirement of the user's job profiles or experiences, it is observed that beginners try to opt intermediate and advanced level courses to advance their careers and despite managers being highly experienced they prefer basic level courses to analyze how the basics of things works,hence their recommendations are also attached with the Difficulty Levels of the course.

Conclusion

Sophisticated Course recommendation techniques in which the User is recommended a course based on his Profile, Needs, History, Network, Difficulty Levels are implemented using the data mining techniques like score evaluations of the course content with repect to the skills of User,...

We have also implemented a User Interface with which the User is recommended a series of Courses based on the User’s actions, like for the different features like Course Recommendation via Cpurse Description, Obtained Skills,...

References

1) Base Paper :- https://dl.acm.org/doi/pdf/10.1145/3357384.3357817

2) https://www.analyticsvidhya.com/blog/2021/08/developing-a-course-recommender-system-using-python/