Hello, and welcome. In this video, we'll be covering Content-Based Recommender Systems. So let's get started. A Content-based recommendation system tries to recommend items to users based on their profile. The user's profile revolves around that user's preferences and tastes. It is shaped based on user ratings, including the number of times that user has clicked on different items or perhaps even liked those items. The recommendation process is based on the similarity between those items. Similarity or closeness of items is measured based on the similarity in the content of those items. When we say content, we're talking about things like the items category, tag, genre, and so on. For example, if we have four movies, and if the user likes or rates the first two items, and if Item 3 is similar to Item 1 in terms of their genre, the engine will also recommend Item 3 to the user. In essence, this is what content-based recommender system engines do. Now, let's dive into a content-based recommender system to see how it works. Let's assume we have a data set of only six movies. This data set shows movies that our user has watched and also the genre of each of the movies. For example, Batman versus Superman is in the Adventure, Super Hero genre and Guardians of the Galaxy is in the Comedy, Adventure, Super Hero and Science-fiction genres. Let's say the user has watched and rated three movies so far and she has given a rating of two out of 10 to the first movie, 10 out of 10 to the second movie and eight out of 10 to the third. The task of the recommender engine is to recommend one of the three candidate movies to this user, or in other, words we want to predict what the user's possible rating would be of the three candidate movies if she were to watch them. To achieve this, we have to build the user profile. First, we create a vector to show the user's ratings for the movies that she's already watched. We call it Input User Ratings. Then, we encode the movies through the one-hot encoding approach. Genre of movies are used here as a feature set. We use the first three movies to make this matrix, which represents the movie feature set matrix. If we multiply these two matrices we can get the weighted feature set for the movies. Let's take a look at the result. This matrix is also called the Weighted Genre matrix and represents the interests of the user for each genre based on the movies that she's watched. Now, given the Weighted Genre Matrix, we can shape the profile of our active user. Essentially, we can aggregate the weighted genres and then normalize them to find the user profile. It clearly indicates that she likes superhero movies more than other genres. We use this profile to figure out what movie is proper to recommend to this user. Recall that we also had three candidate movies for recommendation that haven't been watched by the user, we encode these movies as well. Now we're in the position where we have to figure out which of them is most suited to be recommended to the user. To do this, we simply multiply the User Profile matrix by the candidate Movie Matrix, which results in the Weighted Movies Matrix. It shows the weight of each genre with respect to the User Profile. Now, if we aggregate these weighted ratings, we get the active user's possible interest level in these three movies. In essence, it's our recommendation lists, which we can sort to rank the movies and recommend them to the user. For example, we can say that the Hitchhiker's Guide to the Galaxy has the highest score in our list, and it's proper to recommend to the user. Now, you can come back and fill the predicted ratings for the user. So, to recap what we've discussed so far, the recommendation in a content-based system is based on user's taste and the content or feature set items. Such a model is very efficient. However, in some cases, it doesn't work. For example, assume that we have a movie in the drama genre, which the user has never watch. So, this genre would not be in her profile. Therefore, shall only get recommendations related to genres that are already in her profile and the recommender engine may never recommend any movie within other genres. This problem can be solved by other types of recommender systems such as collaborative filtering. Thanks for watching.

# Content-based Course Recommender System using User Profile and Course Genres

Skills Network Labs (SN Labs) is a virtual lab environment used in this course. Upon clicking "Open Tool" in accordance with IBM Skills Network Privacy policy your Username and Email will be passed to Skills Network Labs and will only be used for communicating important information to enhance your learning experience.

To download the notebook, [click here.](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-ML321EN-SkillsNetwork/labs/module_3/lab_jupyter_content_user_profile.ipynb)