Data Mining

Recommenders – Content-Based Filtering

Examples from "Data Science Concepts and Practice" 2018 MK : Vijay Kotu and Bala Deshpande

Terri Hoare – November 2023

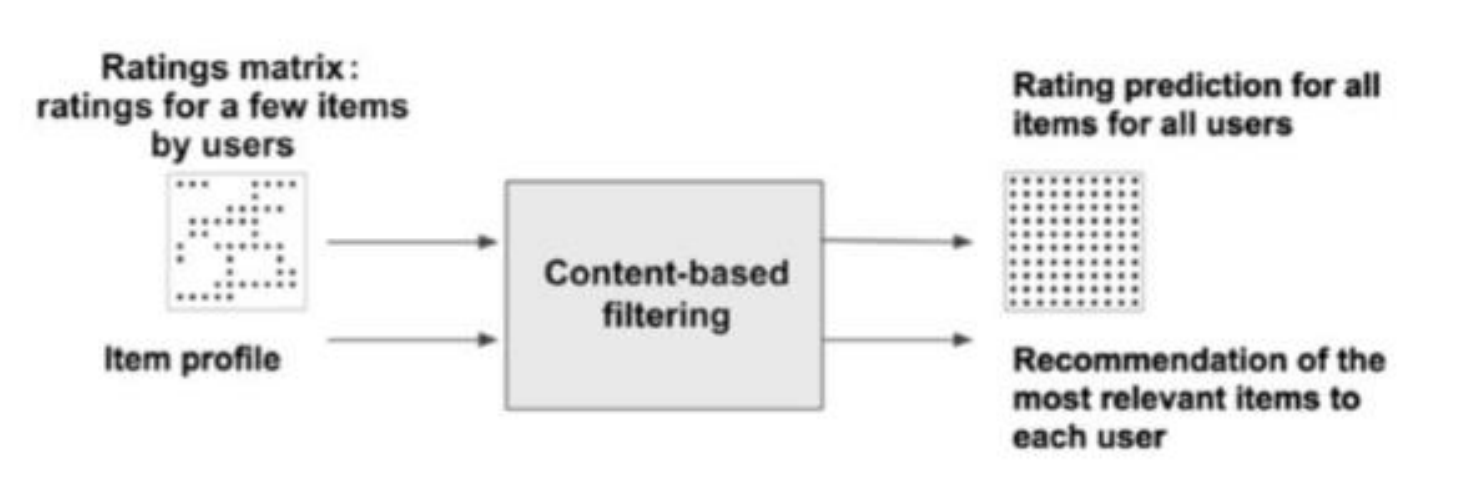
Recommendation Engines
Content-Based Filtering

The collaborative filtering method uses the past user-item interaction data as the only input for building the recommenders. Content or attribute-based recommenders use the explicit properties of an item (attributes) in addition to the past user-item interaction data as inputs for the recommenders.

They operate under the assumption that the items with similar properties have similar ratings at the user level. This assumption makes intuitive sense. If a user liked the movies Fargo, No Country for Old Men, and Burn After Reading, then the user would most likely prefer The Big Lebowski, which are all directed by the same people—the Coen brothers. Content-based recommendation engines keep suggesting an item to a user similar to the items rated highly by the same user. The user will most likely get recommendations about movies with the same cast or director as the user preferred in the past.

Recommendation Engines
Content-Based Filtering

The distinguishing feature of the content-based recommendation engine is that it sources attributes of an item, also known as building the item profile. The attribute data about the movies are readily available in public databases like IMDB4 where the cast, directors, genre, description, and the year of the title can be sourced. The item attributes can be derived from structured catalogues, tags, or unstructured data from the item description and images.



Recommendation Engines
Content-Based Filtering

Predicting ratings using a content-based recommendation method involves two steps:-

The first step is to build a good item profile. Each item in the catalogue can be represented as a vector of its profile attributes.

The second step is to extract the recommendations from the item profile and ratings matrix. There are two distinct methods used for extracting recommendations: a user profile-based approach and a supervised learning-based approach.

Recommendation Engines
Content-Based Filtering

The **user profile approach** computes the preference of users to the item attributes from the ratings matrix. The proximity of the users and the items against the item attribute space indicates the preference of the user to the items.

The **supervised learning approach** treats the user preference of attributes as a user level classification or regression problem with the ratings matrix serving as the label (target) and item attributes as the predictors (feature). If one uses a decision tree as the supervised learning technique, then **each user will have a personalized decision tree.** The nodes in the decision tree will be checking an item attribute to predict whether the user will prefer the item or not.

Content-Based Filtering
Building an Item Profile

An item profile is a set features or discrete characteristics about an item in the form of a matrix. Features, also called attributes, provide a description of an item. Each item can be considered as a vector against the set of attributes. In case of the books, the attributes may be the publisher, author, genre, subgenre, etc. In the case of movies, the attributes may be individual cast members, year, genre, director, producer, etc. A matrix can be built with columns as the universe of attributes for all the items where each row is a distinct item. The cells can be Boolean flags indicating if the item is associated with the attribute or not. Similar to the document vectors discussed in Text Mining, the number of columns or attributes will be large and the matrix will be sparse.

Table Item Profile

Movie	Tom Hanks	Helen Miren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action
Fargo				1				
Forrest Gump	1							
Queen		1						
Sleepless in Seattle	1						1	
Eye in the Sky		1						1

Content-Based Filtering
Building an Item Profile

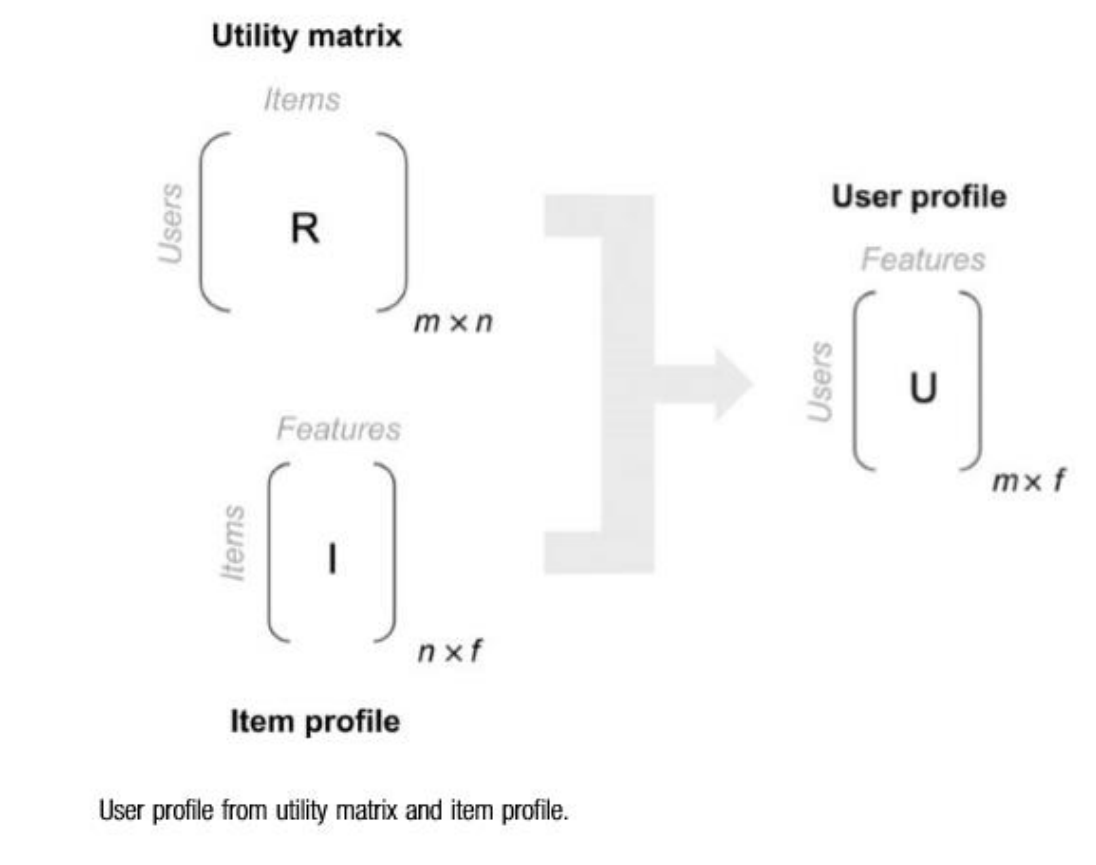
The **user profile approach** computes the preference of users to the item attributes from the ratings matrix. The proximity of the users and the items against the item attribute space indicates the preference of the user to the items.

The **supervised learning approach** treats the user preference of attributes as a user level classification or regression problem with the ratings matrix serving as the label (target) and item attributes as the predictors (feature). If one uses a decision tree as the supervised learning technique, then **each user will have a personalized decision tree.** The nodes in the decision tree will be checking an item attribute to predict whether the user will prefer the item or not.

Content-Based Filtering
Building an Item Profile – User Profile Computation

The **user profile approach** computes the user-item preference by building a user feature matrix in addition to the item feature matrix. The user feature matrix or the user profile maps the user preference to the same features used in the item feature matrix, thereby, measuring the strength of preference of the user to the features.

Proximity measures like centred cosine metric discussed in the neighbourhood based methods is used to measure the preference between a user and an item. The proximity measure between user and item is used to provide recommendations of the item to the user.

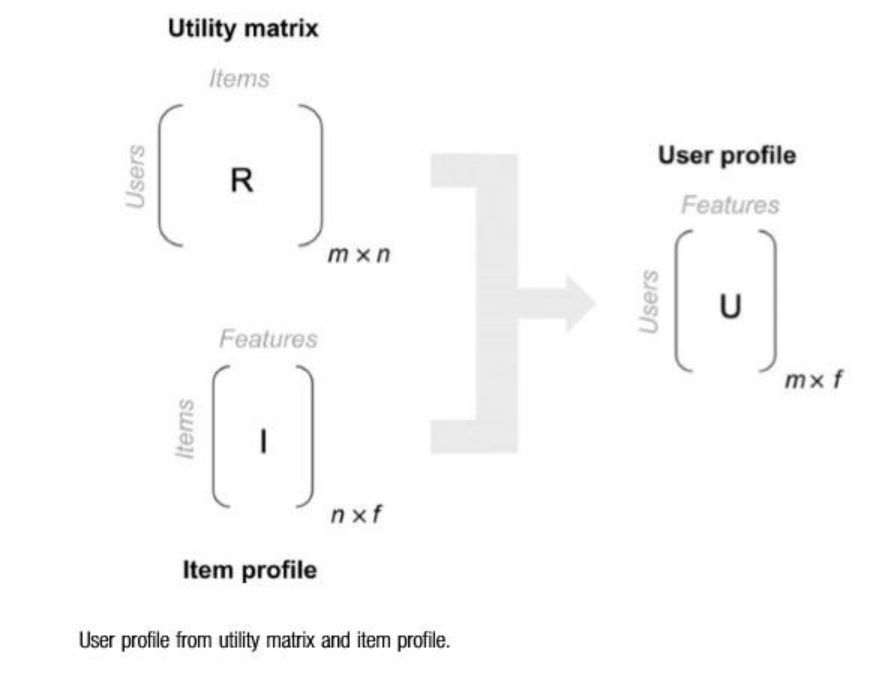


User profile from utility matrix and item profile.

Content-Based Filtering
Building an Item Profile – User Profile Computation

The matrix R is the ratings matrix with six users and five movies. This is a Boolean matrix with 1 indicating that the user likes the movie and the blank indicating no explicit preference for the movie. The matrix I is the item profile with f columns, starting with the cast, ... , director, movie genre. In practice, f will span thousands of columns, which is a superset of all the cast members, directors, and genres of all the movies in the catalog.

The user profile U can be derived in such a way than the value-shown user profile is percent of the time that the feature appears in the movies liked by the user.



User profile from utility matrix and item profile.

Content-Based Filtering
Building an Item Profile – User Profile Computation

For example, Olivia likes the movies Fargo, Queen, and Eye in the Sky. Two-thirds of all the movies liked by Olivia have Helen Mirren in the cast (Queen and Eye in the Sky). All the movies (Forrest Gump and Sleepless in Seattle) liked by Josephine have Tom Hanks in the cast.

Table Ratings Matrix R

	Fargo	Forrest Gump	Queen	Sleepless in Seattle	Eye in the Sky
Josephine		1		1	
Olivia	1		1		1
Amelia		1			
Zoe	1				
Alanna					
Kim		1			

Table Item Profile I

Movie	Tom Hanks	Helen Mirren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action
Fargo				1				
Forrest Gump	1							
Queen		1						
Sleepless in Seattle	1						1	
Eye in the Sky		1						1

The generic formula for the cells in the user profile U is the number of times the feature appears in the movies liked by the user divided by the number of movies liked by the user.

Table User Profile *U*

	Tom Hanks	Helen Mirren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action
Josephine	1						1/2	
Olivia		2/3						1/3
Amelia	1							
Zoe				1				
Alanna								
Kim	1							

Content-Based Filtering
Building an Item Profile – User Profile Computation

The user feature vector for Amelia is {1,0,...,0,0,...,0,0} and the item feature vector for Fargo is {0,0,...,1,0,...,0,0} and Forrest Gump is {1,0,...,0,0,...,0,0}. Out of these two item vectors, Forrest Gump is closer (in fact, perfect match in this example) to Amelia's user vector and, hence, it gets recommended.

Table Item Profile I

Movie	Tom Hanks	Helen Mirren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action
Fargo				1				
Forrest Gump	1							
Queen		1						
Sleepless in Seattle	1						1	
Eye in the Sky		1						1

Table User Profile *U*

	Tom Hanks	Helen Mirren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action
Josephine	1						1/2	
Olivia		2/3						1/3
Amelia	1							
Zoe				1				
Alanna								
Kim	1							

As more becomes known about the user's preferences (when the user "likes" a title), the user profile is updated. Consequently, the weights of the user feature matrix are updated and the latest user profile is used to compare against all the items in the item profile.

Content-Based Filtering
Building an Item Profile – User Profile Computation

Note that in this method, unlike the collaborative filtering approach, no information from other users is needed to make recommendations. This feature makes the content-based recommendation system a good fit to address the cold start problem, especially when a new item is added to the system. When a new movie title is added to the system, the item attributes are already known a priori. Therefore, the new items can be instantly recommended to the relevant users.

However, the content based method needs more consistent information about each item to make meaningful recommendations to the users. The content-based recommenders do not entirely address the cold start problem for new users. Some information is still needed on the new user's item preferences to make a recommendation for new users.

Content-Based Filtering
How To Implement

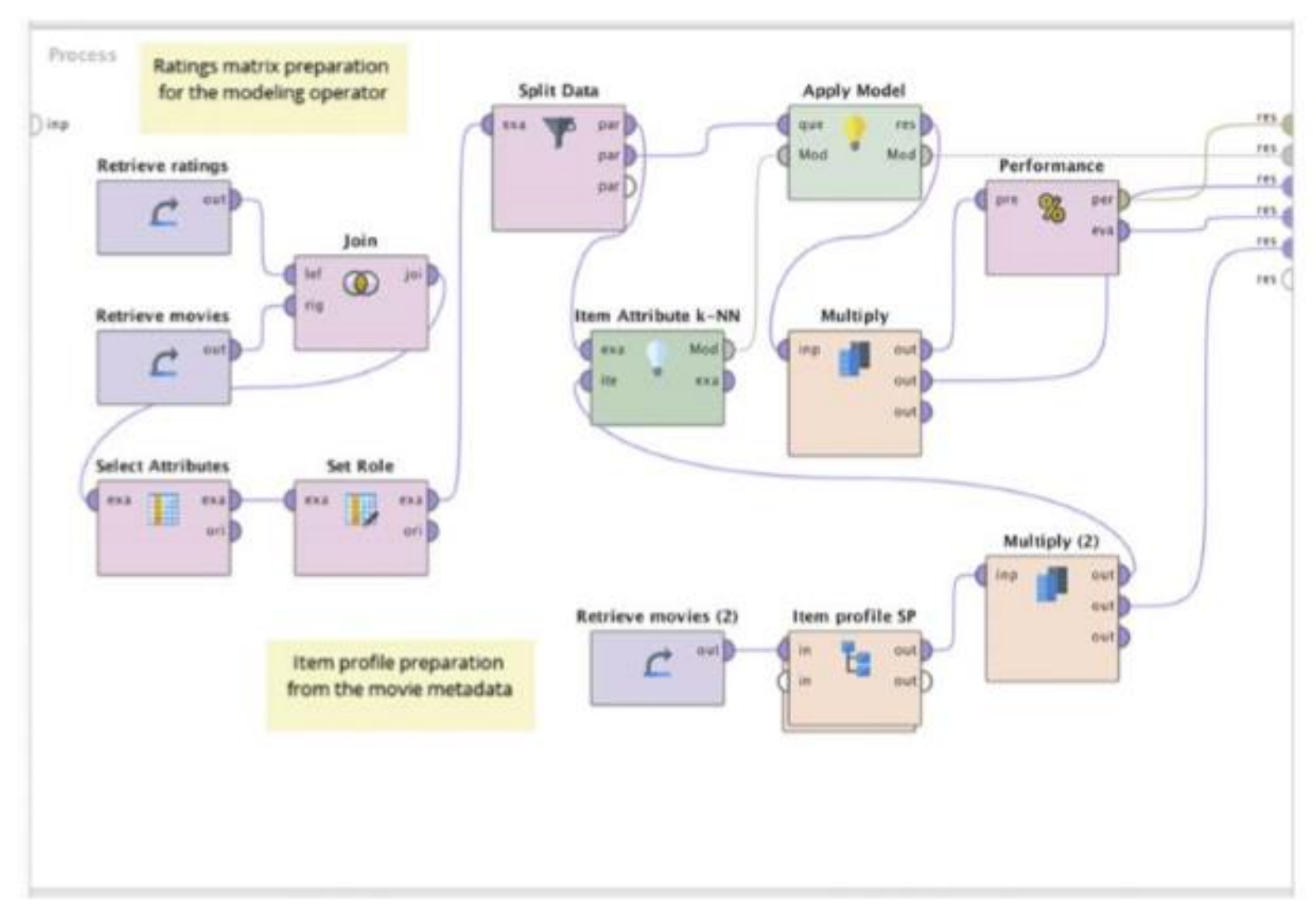
In addition to the standard ratings matrix, a content-based recommender needs the item profile. Sourcing item attribute dataset is one more additional data pre-processing step to be built in the data science tool for creating a content-based recommendation engine. In RapidMiner, implementing the content-based recommenders can be accomplished using the Recommender extension operators.

The same MovieLens ratings matrix dataset used earlier in collaborative filtering is used to implement content-based recommenders. There are two datasets provided by MovieLens. The first datafile contains a ratings matrix, with user ID, movie ID, and ratings.

The ratings matrix has 100,000 ratings given by 1000 users for 1700 titles. The movie datafile contains limited metadata about the movie ID: title and concatenated genres. This second dataset will serve as the item profile to build the content-based filtering.

Content-Based Filtering
How to Implement

11C Content based - attribute.rmp



Content-Based Filtering
How to Implement

11C Content based – attribute.rmp (Item Profile SP)

(A)

Row No.	movieId	title	genres
1	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	2	Jumanji (1995)	Adventure|Children|Fantasy
3	3	Grumpier Old Men (1995)	Comedy|Romance
4	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	5	Father of the Bride Part II (19...	Comedy
6	6	Heat (1995)	Action|Crime|Thriller
7	7	Sabrina (1995)	Comedy|Romance
8	8	Tom and Huck (1995)	Adventure|Children
9	9	Sudden Death (1995)	Action
10	10	GoldenEye (1995)	Action|Adventure|Thriller

(B)

movieId	(no genres ...	?	action	adventur	anim	children	comedi	crime	documentari
1	0	0	0	1	1	1	1	0	0
2	0	0	0	1	0	1	0	0	0
3	0	0	0	0	0	0	1	0	0
4	0	0	0	0	0	0	1	0	0
5	0	0	0	0	0	0	1	0	0
6	0	0	1	0	0	0	0	1	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	1	0	1	0	0	0
9	0	0	1	0	0	0	0	0	0
10	0	0	1	1	0	0	0	0	0

(C)

Row No.	movieId	Genre
1	1	4
2	1	5
3	1	6
4	1	7
5	1	11
6	2	4
7	2	6
8	2	11

Content-Based Filtering
Supervised Learning Models

A supervised learning model-based recommender approaches the problem of user-item preference prediction at the individual user level.

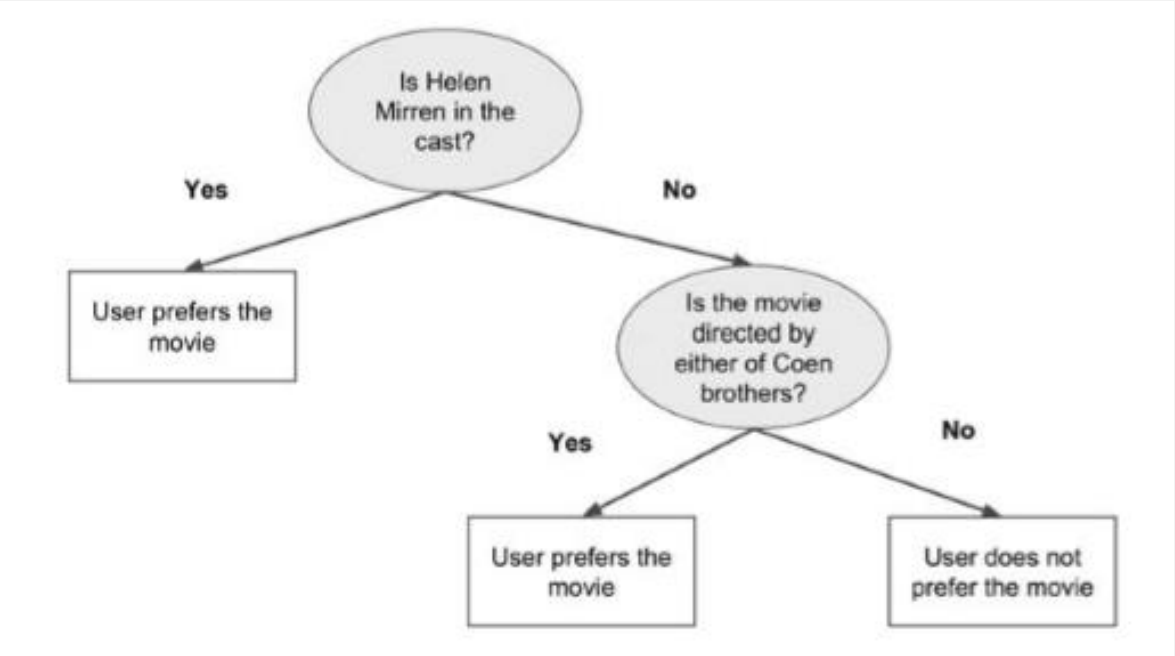
If a user has expressed interest in a few items and if those items have features, then the interest of the users to those features can be inferred. Consider the user-item ratings matrix and the item profile matrix. The item profile matrix can be customized just for one user, say Olivia, by introducing a new column in the item profile to indicate whether Olivia likes the movie. This yields the item profile matrix for one user (Olivia) shown. This matrix is strikingly similar to the training data used in the supervised models (Classification and Regression).

Table Item Profile With Class Label for One User

Movie	Tom Hanks	Helen Mirren	...	Joel Coen	Kathryn Bigelow	...	Romantic	Action	Class label for Olivia
Fargo				1					1
Forrest Gump	1								0
Queen		1							1
Sleepless in Seattle	1						1		0
Eye in the Sky		1						1	1

Content-Based Filtering
Supervised Learning Models

In the supervised learning model based approach, each user has a personalized decision tree. Suppose one has a straightforward preference for movies: they only like it if the movie has Helen Mirren in the cast or is directed by the Coen brothers. Their personalized decision tree would be like the one below.



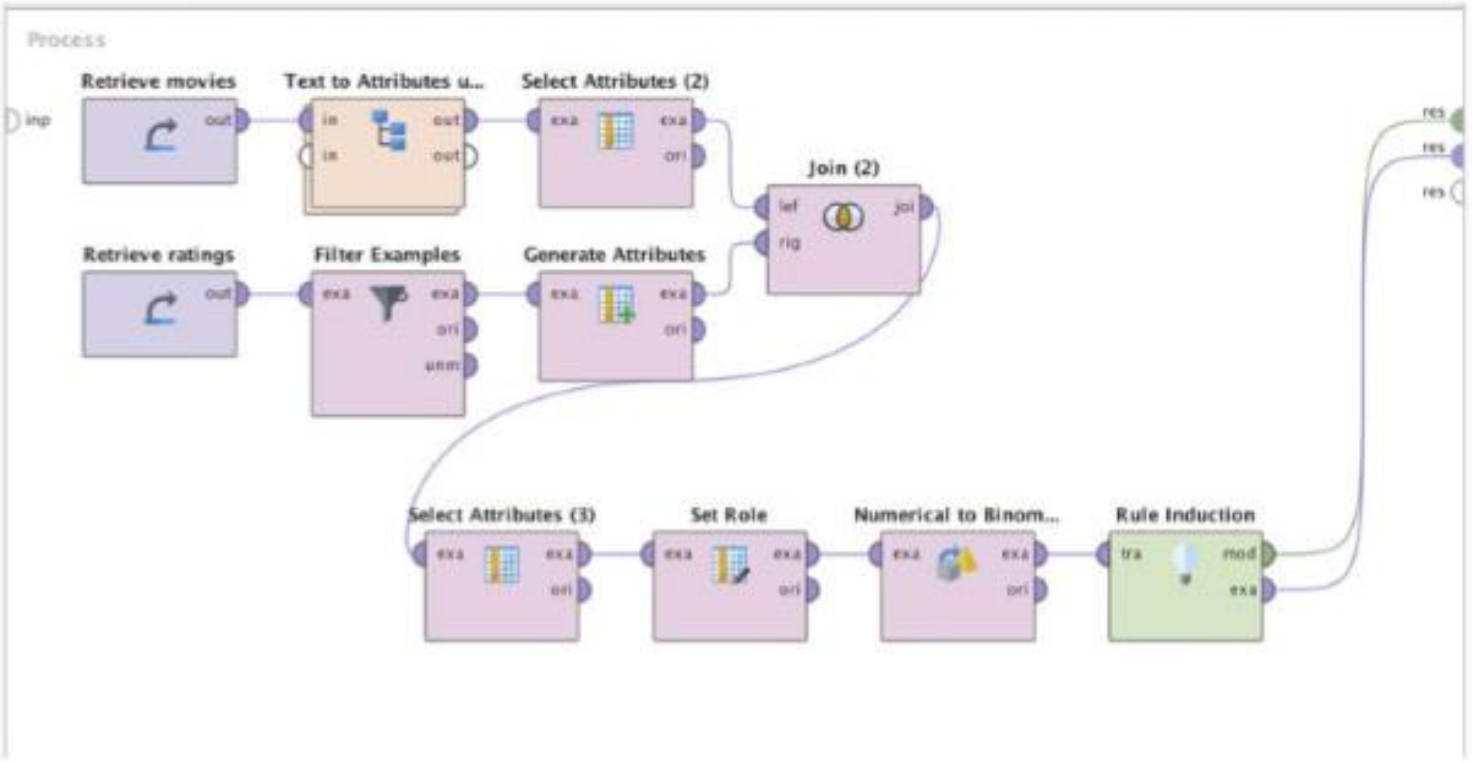
Supervised Learning Models
How To Implement

The supervised learning (classification) model approach for a content-based recommendation engine builds a classification model for each user. Hence, the model building is shown for one user and the process can be repeated in the loop for each user in the ratings matrix. The rule induction modelling technique is used in this recommender implementation. It can be replaced with various other classification or regression modelling techniques to suit the application and the data.

Dataset The datasets used for the process are from MovieLens database with two datasets. The first dataset (ratings dataset) contains ratings, user ID and movie ID attributes. The second dataset (item dataset) contains limited metadata about each movie—movie ID, movie title, and concatenated genre attributes. To create the supervised learning model, these two datasets have to be merged to one dataset that has labels and attributes for one user.

Supervised Learning Models
How to Implement

11D Content Classification – Tree



Classification process for one user in the system.

Recommendation Engines
Content-Based Filtering – Discussion

Content-based recommendation engines are better at explaining why the system is making the recommendation because it has generalized the features the user is interested in. For example, the system might recommend Apollo 13, Saving Private Ryan, and Captain Phillips because the user is interested in movies that have Tom Hanks in the cast.

Unlike the collaborative filtering method, data from other users is not needed for the content-based systems, however, additional data from the items are essential. This feature is significant because when a new user is introduced into the system, the recommendation engine does not suffer from the cold start problem. When a new item is introduced, say a new movie, the attributes of the movie are already known. Hence, the addition of a new item or a user is quite seamless for the recommenders.

Recommendation Engines
Content-Based Filtering - Discussion

The key datasets involved in the recommendation, that is, the item profile, user profile or classification models, and the recommendation list, can be pre-computed. Since the main objective in many cases is finding the top recommended items instead of filling the complete ratings matrix, decision trees can focus only on the attributes relevant to the user.

Content-based recommenders tend to address unique preferences for the user. For example: users interested in Scandinavian crime thrillers. There may not be enough users who like this subgenre for collaborative filtering to be effective because a sizable cohort of users is needed to prefer these unique genres and other items in the catalog.

Recommendation Engines
Content-Based Filtering – Discussion

Even though rating information from other users are not necessary, an exhaustive item profile is essential for obtaining the relevant recommendations from content-based systems. The features of the item are hard to get to begin with and some attributes like genre taxonomy are difficult to master. Blockbuster items, are by definition, watched by a wider audience, beyond the fanatics of a particular genre.

For example, one doesn't have to be a fan of science fiction to watch Avatar. Just because a user watched Avatar it doesn't mean that the recommender can be inundated with other science fiction movies.

Special handling is required so the system doesn't conflate the success of blockbusters with the user preference for specific attributes in the blockbusters.

Recommendation Engines
Content-Based Filtering - Discussion

Content-based recommenders are content specific. For example, it makes logical sense to recommend Moroccan cooking utensils if the user has shown an interest in books focused on North African cuisine, if the ecommerce platform offers both the categories.

Content-based systems will find this task to be difficult because the knowledge gained in a books category is hard to translate to a kitchen utensils category. Collaborative filtering is content agnostic.

Hybrid recommender systems combine two or more recommendation strategies in different ways to benefit from their complementary advantages.

Recommenders – Finding a user's preference for an item
Comparison of Algorithms

Algorithm	Description	Assumption	Input	Output	Pros	Cons	Use Case
Collaborative Filtering – neighborhood based	Find a cohort of users who provided similar ratings. Derive the outcome rating from the cohort users	Similar users or items have similar likes	Ratings matrix with user-item preferences.	Completed ratings matrix	The only input needed is the ratings matrix Domain agnostic	Cold start problem for new users and items Computation grows linearly with the number of items and users	eCommerce, music, new connection recommendations
Content-based filtering	Abstract the features of the item and build item profile. Use the item profile to evaluate the user preference for the attributes in the item profile	Recommend items similar to those the user liked in the past	User-item rating matrix and Item profile	Completed ratings matrix	Addresses cold start problem for new items Can provide explanations on why the recommendation is made	Requires item profile data set Recommenders are domain specific	Music recommendation from Pandora and CiteSeer's citation indexing
Content-based – Supervised learning models	A personalized classification or regression model for every single user in the system. Learn a classifier based on user likes or dislikes of an item and its relationship with item attributes	Every time a user prefers an item, it is a vote of preference for item attributes	User-item rating matrix and Item profile	Completed ratings matrix	Every user has a separate model and could be independently customized. Hyper personalization	Storage and computational time	eCommerce, content, and connection recommendations