Content-Based Recommendation Engine

The objective of a RecSys is to recommend relevant items for users, based on their preference. Preference and relevance are subjective, and they are generally inferred by items users have consumed previously.

In other words, these algorithms try to recommend items that are similar to those that a user liked in the past (or is examining in the present). In particular, various candidate items are compared with items previously rated by the user and the best-matching items are recommended.

A content-based recommendation engine will generate preferences by combining the profiles of items in a user's history and measuring the distance between a user's profile and the items your organization would like to suggest.

We see different flavors of recommenders, deployed across different verticals.

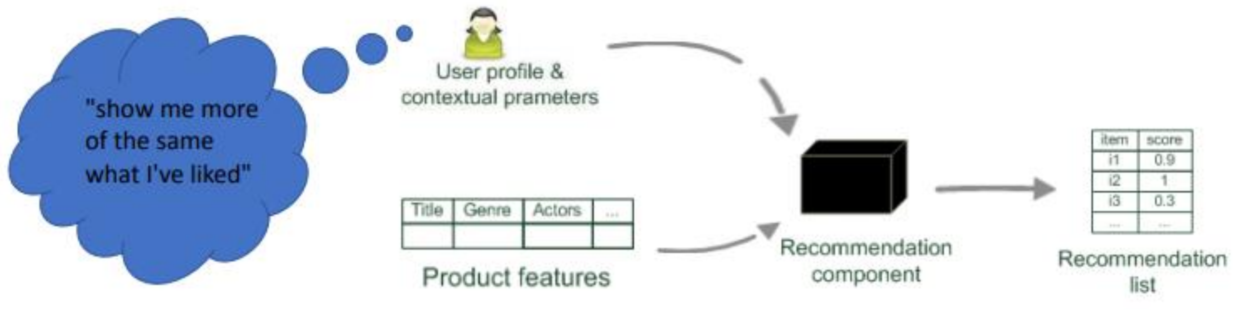
- Amazon
- Netflix
- Facebook
- Movie Lens

The Requirement

- ✓ some information about the available items such as the genre ("content")
- ✓ some sort of user profile describing what the user likes ("Preferences")

The Task:

- ✓ learn user preferences
- ✓ locate/recommend items that are "similar" to the user preferences


"show me more of the same what I've liked"
User profile & contextual prameters
Title | Genre | Actors | ...
Product features
Recommendation component
item | score
i1 | 0.9
i2 | 1
i3 | 0.3
Recommendation list

Tapmad Data that need to be stored for Recommendation Engine:
We require following features in new table in Database.

User Profile Table:

In User Profile Table Gender and Date of birth fields need to be updated.

We need to add occupation feature in user profile table and Occupation data should be stored like dropdown manner(Professional, Student and other).

User behavior:

A new table name (user stream table) will be created in database for Machine Learning.

User stream Table

User Stream Id	-- That is like a serial number of the table
User Id	-- We require User id to know user's favourite
Session Id	-- A session ID is a unique number that a Web site's server assigns a specific user for the duration of that user's visit (session)
startedAt	-- When user started watching stream, time should be mentioned
endedAt	-- When user ended watching stream, time should be mentioned
IP address	-- It will help out to trace user location
Channel Id	-- We need channel id to know which user is watching channels
VoD Id	-- VOD ID is required for end to end service that locates the source
Platform	-- User landed from which platform whether web, android
Country code	-- We required country code to know user's specific country

Video on Demand Table:

We need to update Genre field in Video on Demand table. Genre data should be like (Action, Adventure, Animation, Comedy, Crime, Fantasy, Horror, War etc.)

Video on Demand category Table:

We also need to update Genre field in Video on Demand category table in this table Genre data should be like all categories names

“**Similarity**” is computed from item attributes, e.g.,

- **User profiling**:
  - User ID
  - Gender
  - Age
  - Occupation
- **Similarity of movies by**:
  - Movie Id
  - Title
  - Actor
  - Director
  - Genre
  - Year of release
- **Rating that contains ratings of movie by users:**
  - User Id
  - Movie Id
  - Rating
  - Timestamp

Explicit feedback is intentionally provided by users in form of clicking the “like”/” dislike” buttons, rating an item/movie by number of stars, etc.

Release year of movie:

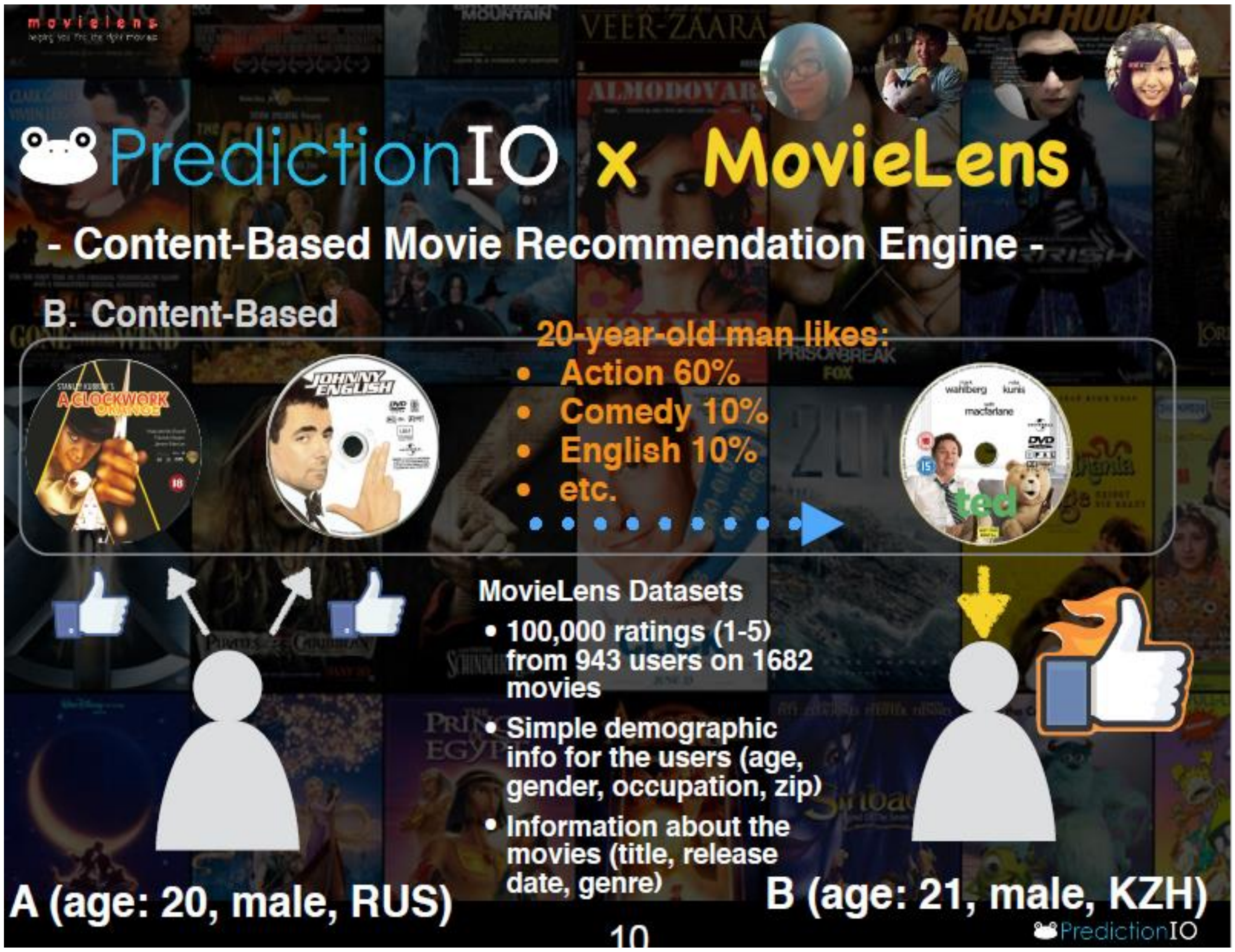
We require release date of movies.so that we should have information that which movie was release in which year? We can easily extract this information from the movies

Most popular movie genres year by year

We might expect that some events in history might have influenced the movie creators to produce specific genres. First, we will check what genres are the most popular in general. Which genres receive the highest ratings? How does this change over time?

Movie Genre

Determine the temporal trends in the genres/tagging activity of the movies released Looking at how each movie genre is tagged by users is a great way to see if a movie genre can be described using just a few words. We’ll explore a selection of movie genres and see if anything interesting pops out.



PredictionIO x MovieLens

- Content-Based Movie Recommendation Engine -

B. Content-Based

20-year-old man likes:

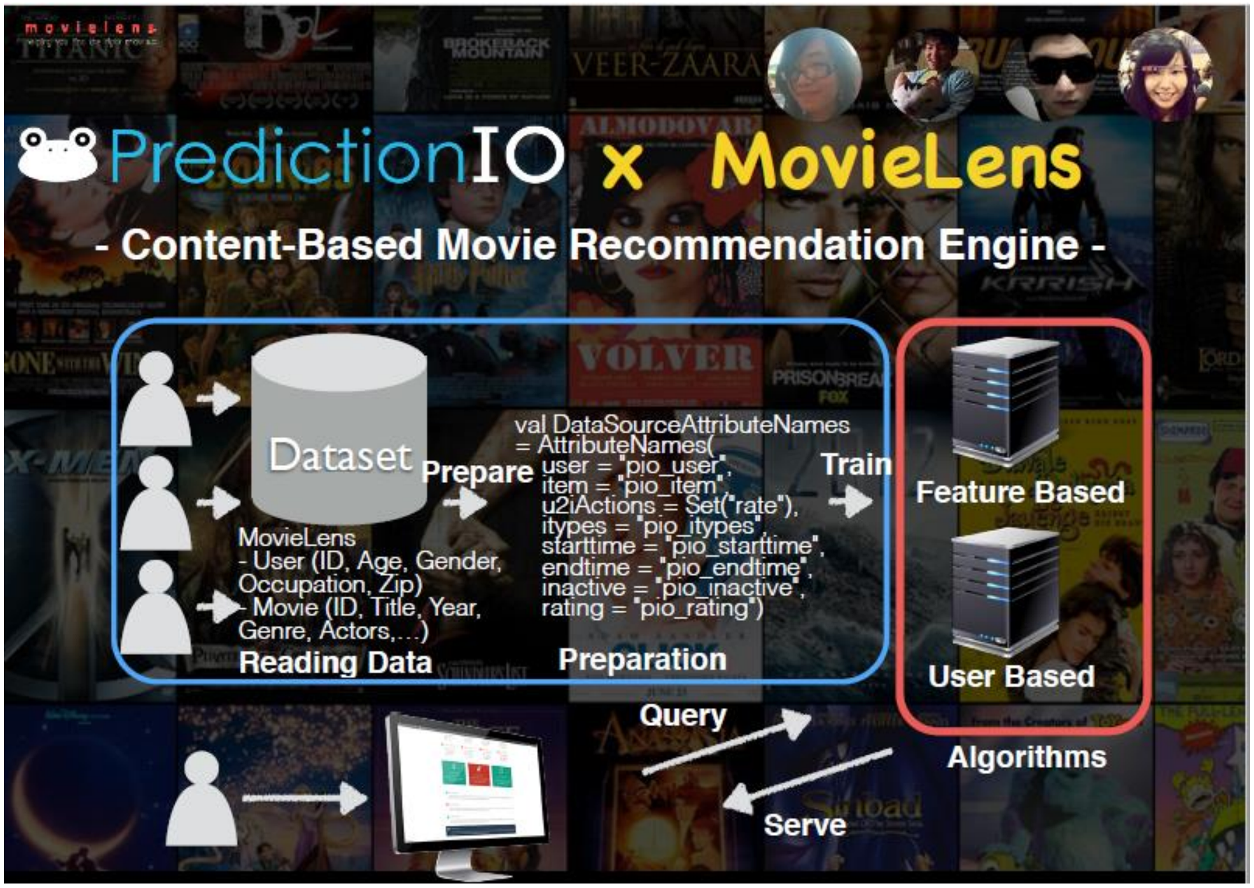
- Action 60%
- Comedy 10%
- English 10%
- etc.

MovieLens Datasets

- 100,000 ratings (1-5) from 943 users on 1682 movies
- Simple demographic info for the users (age, gender, occupation, zip)
- Information about the movies (title, release date, genre)

A (age: 20, male, RUS)

B (age: 21, male, KZH)



PredictionIO x MovieLens

- Content-Based Movie Recommendation Engine -

Dataset

MovieLens
- User (ID, Age, Gender, Occupation, Zip)
- Movie (ID, Title, Year, Genre, Actors,…)

Reading Data

Prepare

```
val DataSourceAttributeNames
= AttributeNames(
  user = "pio_user",
  item = "pio_item",
  u2iActions = Set("rate"),
  itypes = "pio_itypes",
  starttime = "pio_starttime",
  endtime = "pio_endtime",
  inactive = "pio_inactive",
  rating = "pio_rating")
```

Preparation

Train

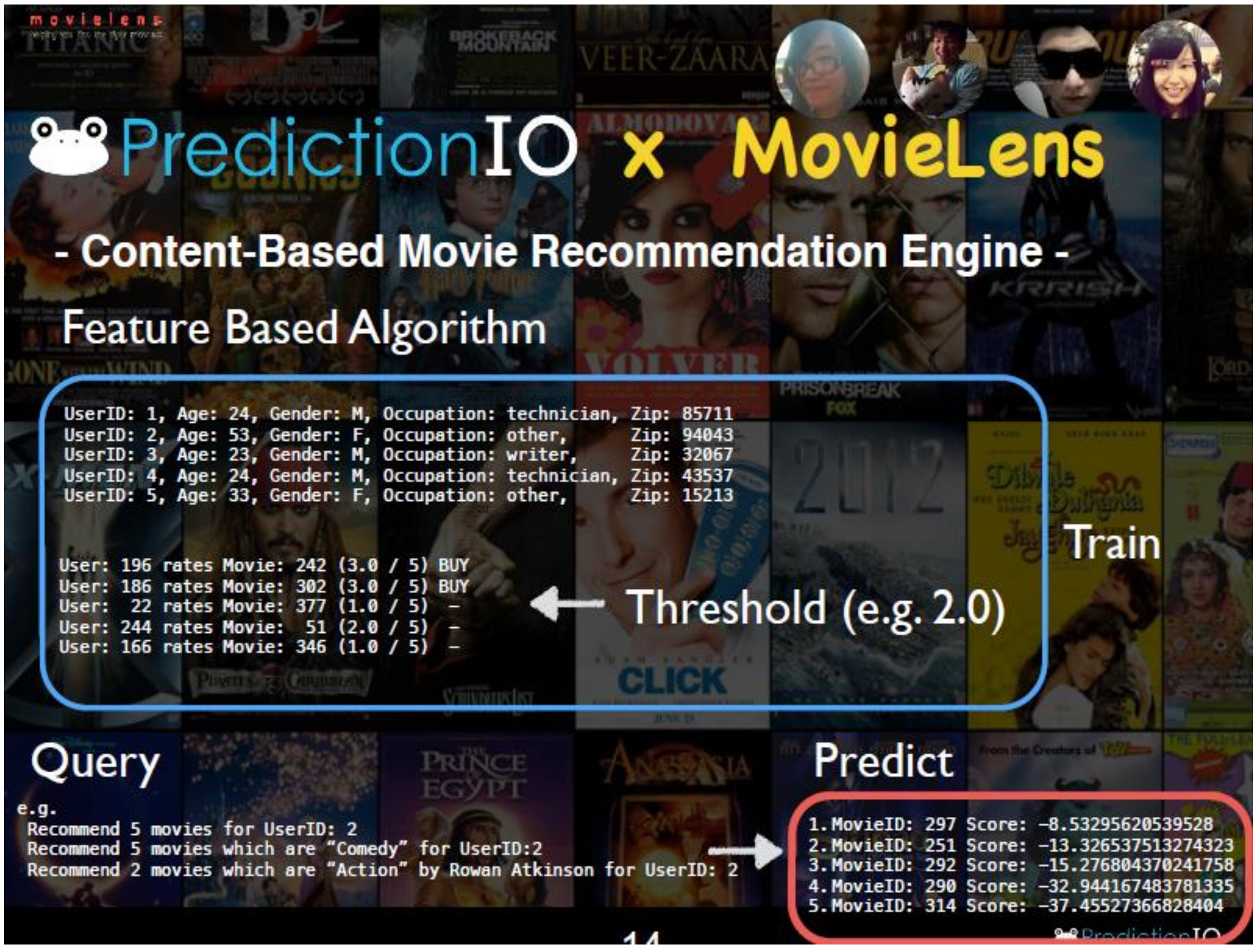
Feature Based

User Based

Algorithms

Query

Serve



PredictionIO x MovieLens

- Content-Based Movie Recommendation Engine -

Feature Based Algorithm

```
UserID: 1, Age: 24, Gender: M, Occupation: technician, Zip: 85711
UserID: 2, Age: 53, Gender: F, Occupation: other,      Zip: 94043
UserID: 3, Age: 23, Gender: M, Occupation: writer,     Zip: 32067
UserID: 4, Age: 24, Gender: M, Occupation: technician, Zip: 43537
UserID: 5, Age: 33, Gender: F, Occupation: other,      Zip: 15213

User: 196 rates Movie: 242 (3.0 / 5) BUY
User: 186 rates Movie: 302 (3.0 / 5) BUY
User:  22 rates Movie: 377 (1.0 / 5)  -
User: 244 rates Movie:  51 (2.0 / 5)  -
User: 166 rates Movie: 346 (1.0 / 5)  -
```

Threshold (e.g. 2.0)

Train

Query

```
e.g.
 Recommend 5 movies for UserID: 2
 Recommend 5 movies which are "Comedy" for UserID:2
 Recommend 2 movies which are "Action" by Rowan Atkinson for UserID: 2
```

Predict

```
1.MovieID: 297 Score: -8.53295620539528
2.MovieID: 251 Score: -13.326537513274323
3.MovieID: 292 Score: -15.276804370241758
4.MovieID: 290 Score: -32.944167483781335
5.MovieID: 314 Score: -37.45527366828404
```