**AMAZON PERSONALIZE**

*Visual Demonstration for Field Teams*

<https://www.personalisevideorecs.info/recommend/> 30th April 2019

Introduction

This was initially built to showcase the Amazon Personalize service, and has been demoed at the following two shows:

* **NRF Big Show** – New York *(January 2019, 3 days)*
* **ShopTalk** - Las Vegas *(March 2019, 3 days)*

The web application is designed to show each of the major features of Amazon Personalize, so whilst it isn’t a polished or professional-looking application it does manage to show what the service is capable of.

**NOTE:** The demo application does not yet support Event Streaming, as the time for any stream to have an impact is 60 seconds or more. This **will be fixed**, but is of lower priority at this time

The model built was the MovieLens dataset, as discussed [here](https://docs.aws.amazon.com/personalize/latest/dg/getting-started.html), but with a larger data-set and where I’d spent some time getting direct links to the movie posters on IMDB – I’ve got a some minor de-duplication work to do, and one or two features need updating or completing. However, as well as the model for the reviews, there is RDS Postgres holding all of the data that the web-app needs – e.g. links to the movie poster URLs, the review metrics, etc. The web-app server is currently in us-east-1 with the following architecture:

**TBD:** Architecture diagram – Cloudfront, WAF, ALB, EC2, RDS

Initial Screen



The application has been built on the basis of 100,000 reviews, from 1-star to 5-stars. These are real reviews, so the dataset is not small. The application currently contains 6 different models, each with different variations of the input file that Personalize wants: user-item interaction data, item data and user data.

The reviews themselves were dated between 20th September 1997 and 23rd April 1998. These have all been “brought forward” in time to be between 29th August 2018 and 1st April 2019; this is because the Similar Items algorithm will, by default, apply lesser and lesser influence to reviews as they get older, which is configurable during training via hyperparameters. Hence, in order to ensure that the default parameters work, the reviews dates were all made current.

The models used are as follows:

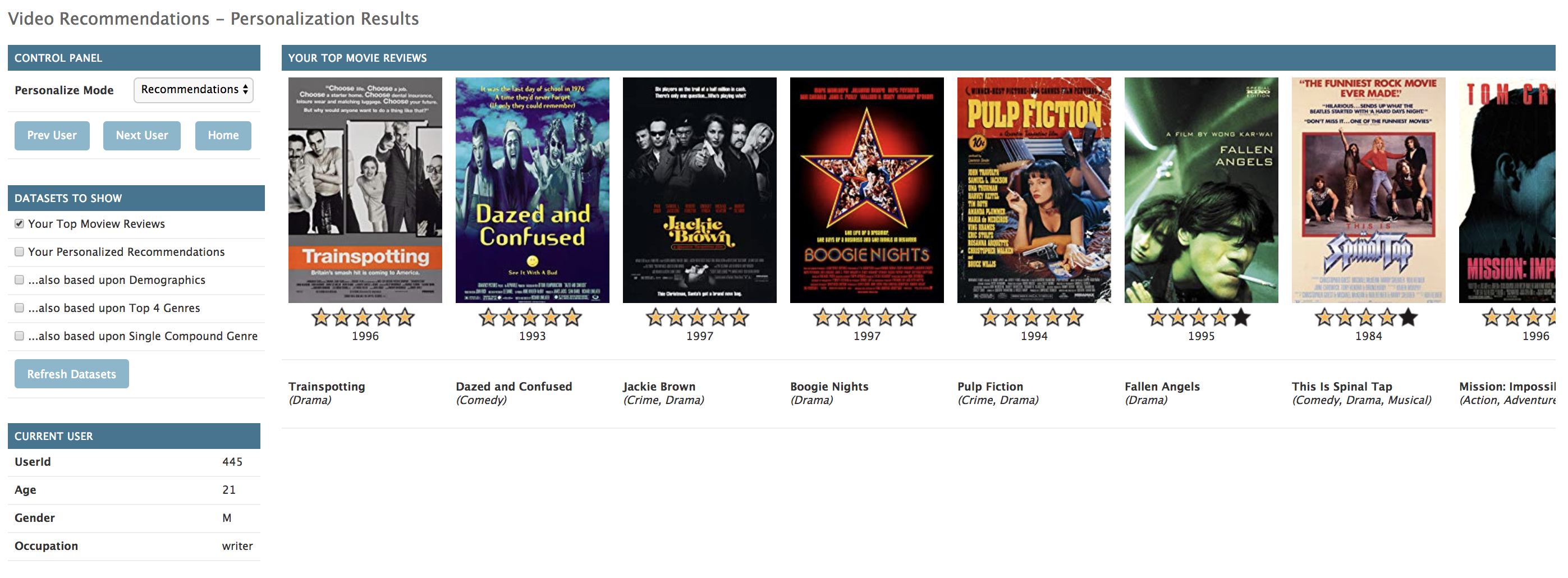
1. Recommend with Reviews – we only take into account the user-item interactions
2. Include Demographics – as #1, but with the user information on age, gender and occupation included
3. Include Compound Genre – user-item interactions, movie year and all movie genres compounded into a single string; e.g. “Action, Comedy”
4. Include Top 4 Genres – as #3, but with a movie’s top-4 genres split into 4 distinct features
5. Personalized Rankings – just user-item interactions, but with a recipe designed to re-rank lists of items into an order specific to a user
6. Similar Items – just user-item interactions, but with a recipe to return items similar to a given one; essentially, this is collaborative filtering, and you can optionally have these re-ranked for the current user using the *Personalized Rankings* algorithm or left in their natural order

In some use-case, such as high-end fashion or jewellery, the inclusion of the user demographic data can make the model better. However, for movie and book reviews it is not a good indicator – this was known about prior to building this model, and the metrics inherent in this model back this up.

To start viewing the reviews of a random user, hit the *Select Random User* button.

Mode: Recommendations

The application has multiple mode, but initially it is in *Recommendations* mode, and in this example has randomly selected user #445 (which you’ll see reflected in the URL):



This consists of the following panels:

* Control Panel – switch the operation mode of the application
* Datasets to Show – allows selection of mode-specific datasets to show on the right
* Current User – demographics of the currently-selected user
* User Review Summary (not shown) – count and distribution of reviews from this user

The *Control Panel* lets you move backwards and forwards through the user database, and the *Home* buttons takes you back to the initial screen with the model metrics. Under *Datasets To Show* we have check-boxes for the user’s reviews (initially shown with the user’s star rating) and the first 4 models that are designed to recommend items to a user from the whole dataset. Note, you can scroll the movie lists left and right to show up to 25 entries.

By selecting different check-boxes and hitting the *Refresh Datasets* button the screen will refresh and show those datasets – for instance, selecting *Your Personalized Recommendations* and refreshing will show the following:



This shows the top-25 recommendations for this user in order, starting from the highest ranked recommendation. Each time you change user there is a single API call to Amazon Personalize for every non-Review dataset you’ve asked for. I would then normally show the following dataset variations:

* Add Demographics – it might not be obvious, but these definitely look slightly worse
* Remove Reviews and Demographic, add Top-4 Genres – shows fairly different list of movies
* Remove Recommendations, add Compound Genre – slightly different again, feels slightly better, and according to the metrics on the Initial Screen these should be better

As an example on the Demographics, select User 888 via the URL, and enable the top 3 checkboxes. The user’s top reviews look like comedies, romances and dramas. Personalize recommends this:

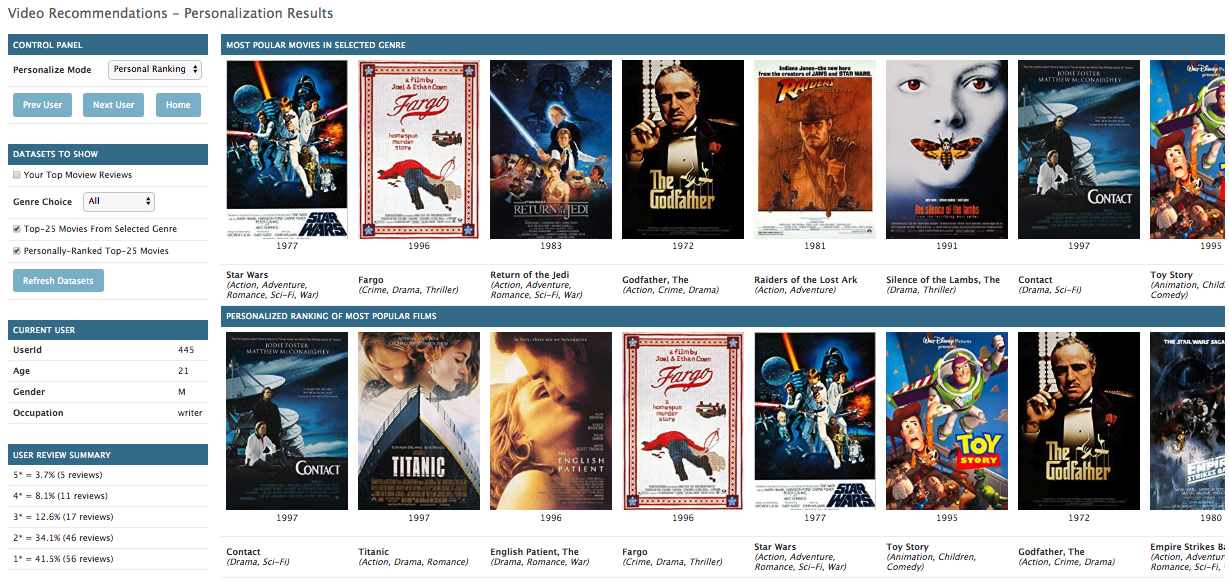
* Standard model – Twelve Monkeys, Independence Day, Sense and Sensibility, Jerry Maguire
* With demographics – Fargo, Braveheart, Scream, Speed, Die Hard 2

The user is a 41-year-old male scientist – the demographic response may well be pretty good, but based upon their actual preferences it is pretty poor.

Of course, the only real way to tell if a model is better is in live A/B testing – you need to know if one model gets more clicks than another model, as that’s really all we’re trying to achieve.

Mode: Personal Rankings

Sticking with user #444, in the *Personalize Mode* dropdown select *Personal Ranking*. De-select the *Reviews* check-box, enable the other two and refresh, and you should see the following screen:

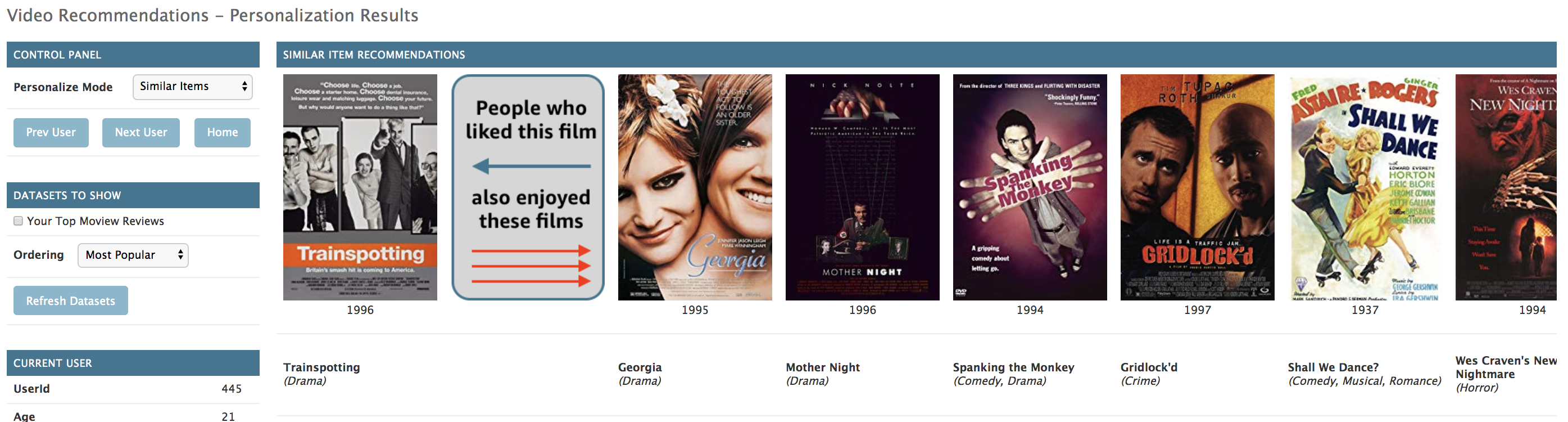


This shows the most popular movies in the dataset at the top, and underneath is that same list re-ordered for user #444. Given the age of the dataset this top-25 list is probably not surprising, but really this could be any list. If you change the *Genre Choice* dropdown to any of the film genres that we have then the screen will refresh and show that – e.g. the top-25 Crime films re-ordered for that user.

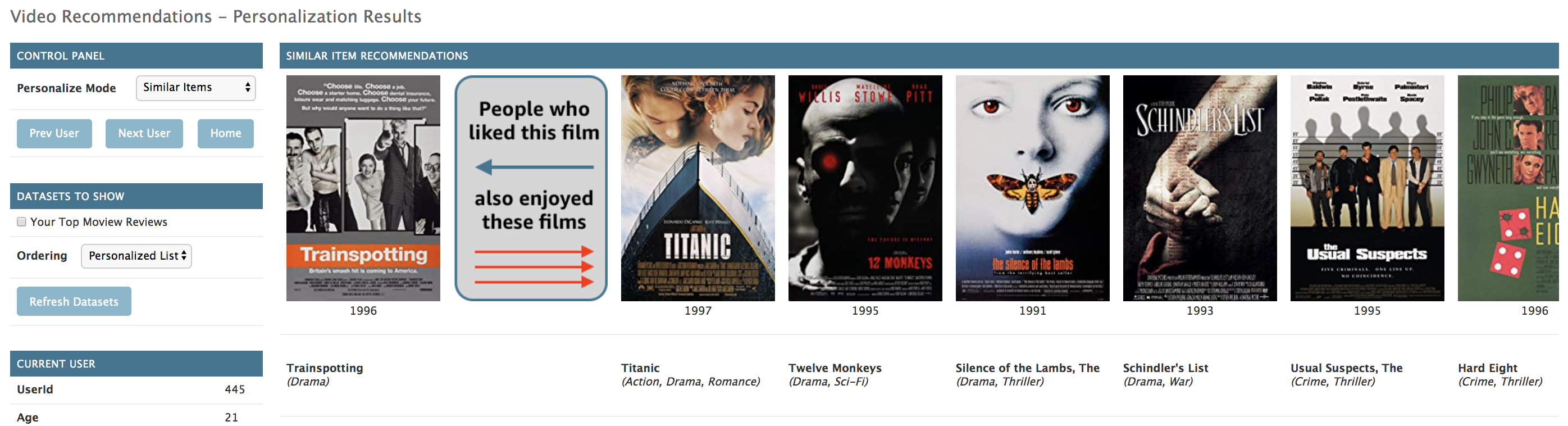
This can be used to do personally rank anything: the top-25 online offers, the most popular holiday destinations, most popular toys for a 9-year-old child, etc. But this example shows that simply showing a user the top-3 in a category can be pretty poor on the personalization front, as for this user none of the category top-3 are in the user’s top-3.

Mode: Similar Items

Jump back to user #445, an in the *Personalize Mode* dropdown select *Similar Items*, which will show a new list starting with that user’s most highly-rated movie, followed by up to 23 more movies that people that like this user’s top movie would also like. The API for this will return the 23 in the most popular order.



There is also an *Ordering* drop-down control, which allows you to choose to order the results either in the most popular order, as shown previously, or turned into a more personal order by pushing this most popular results list through the personal re-ranking algorithm – for our user this results in a rather different, but more personal and most-likely more accurate, result.



Each time you click on one of the moves to the right of the *People Who Liked This Film* icon the first movies to the left of the list will be replaced by the one you clicked on, and the similar items search is re-run for that movie. You can therefore click through and through on a long list of movies just to exercise your curiosity – note, if there is insufficient recent user-interactions for a movie, which is always possible with this data-set, then the recipe returns a list of the most popular items as its recommendation.

Planned Updates

There is just one major update – as users click on movies in the *Similar Items* screen, each one is sent as a clickstream-event to Amazon Personalize, which will slowly update a seventh model. This will be a clone of the initial *Recommendations* model, but it will be allowed to slowly evolve over time; this will be hard to demo live, but if run frequently against the same small list of users then it will be possible to show that the model does change based upon clickstream events.

Polite Ask

This is a LIVE project that is being regularly developed – hence, it may sometimes be temporarily broken as I work on adding new features. I’d appreciate it if you contacted me in advance at [andkane@amazon.co.uk](mailto:andkane@amazon.co.uk) if you are doing an important demo so that I can ensure it is working.

Also, whilst the Amazon Personalize service is in Preview it can – and has – suddenly become broken due to an unannounced change in the underlying service model. If you see this then please let me know, as I may not have spotted it yet and you may be the first.