Information Visualization

CHECKPOINT II: Data cleaning and processing

G44-A

1. Initial Dataset

Our data set was aggregated from two sources: TheMovieDB and an API that gives us scores from Rotten Tomatoes, IMDB and Metacritic. From TheMovieDB we requested information on each episode, and information on each guest star. From the API we requested information on each movie or tv show the actors worked on.

2. Selected/Derived Data

From our dataset we selected these features, for each of these dimensions: from the episode information, we kept the season number, episode number, date and the guest stars; from the guest actor information we kept their personal info (name, date of birth, place of birth, gender, death day), and all the acting credits they have; from the movie/TV information we kept the ID, the cast information, air-date, gender, name, country, and other optional data; and from the rotten tomatoes we kept the rating.

We selected the following derived measures:

- Guest stars in each movie/show;
- Common works by actors who were in SVU;
- Amount of acting credits in SVU only, by guest star;
- The average IMDB rating of each work the actor was in, before and after starring in SVU;
- The frequency of the roles they were cast in, before and after SVU;
- Guest stars by sex and age (grouping actors, showing all women, all people older than fifty at the time of their role, etc).

The bottom three measures are not yet calculated, but we already have the required data to do so. We will make these calculations before the next checkpoint.

These measures will show how Law and Order: Special Victims Unit (SVU) affects its guest stars' careers.

3. Data abstraction

Our datasets are all in JSON format (field type). All items and attributes are nominal, in String format, except the movie/show ratings, which each have a sequential scale, and the actor and movie/show IDs, which are ordinal, in Integer format (these are relative to the site of origin).

The names in the JSON files are self-explanatory.

4. Dataset processing

We used python scripts to treat all the data, and to create the derived measures.

From the episode data, we only kept season number, episode number, date and the guest stars.

The major problem we had with the JSON files was that we had objects within objects, and it wasn't possible to convert and use the data as a table. This occurred because each episode had a list of actors within it. To fix this problem, we flattened the data: we took the guest actors and created an entry for each actor, with the date, season and episode number for each of them.

From the guest actor data, we kept pretty much all their personal info and fetched their acting credits. This acting credits came with the movie/show info. We used a script to create a list of movie/show infos, by id. From this movie/show info data, we kept air-date, gender, name, country, and other optionals. Also, we also remove SVU mentions here.

Given each movie/tv, we needed to make another request for each *imdb_id*. We had problems because some don't have *imdb_id*, but we ignored them

We used the *imdb_id* to get the ratings from another database. Here, we got IMDB, Rotten Tomatoes (some) and Metacritic (some) ratings. This is how we crossed reference information.

About missing values, all the information that was important for calculating their career progression does not have gaps, and while there are some fields missing in the movie/show JSON, for example, it's for superfluous things, like budget or original language, and these will only be used for visualization purposes.

5. Mapping (Data sample / Questions)

Which guest stars participated in each movie/show? *guest_starts.json (list of participation of each actor)*

How many common works did actors who were in SVU had? *tv_movies.json* (list of movies/tv info, with all the actors in it)

How many acting credits each guest star had in SVU only? *moreThanOnce.json* (list of credits by actor id)