ETL Project

Jennifer Cain & Micah Vandersteen

Technical Report

For our ETL Project, we created a non-relational database of film information, including any nominations and wins for the Oscars and the Golden Globes.

**Extract**

We used two datasets and an API. The datasets used are *The Oscar Award, 1927-2020* (Fontes, 2020b) and *Golden Globe Awards 1944-2020* (Fontes, 2020a).

We also used the Open Movie Database (OMDb) API (Fritz, n.d.) to request information related to the movies within the datasets. The API data was in JSON format.

**Transform**

We used Jupyter Notebook to transform the datasets.

***The Oscars.*** We loaded the Oscars CSV into a Pandas data frame. We removed all the rows where there was not a film listed. We then filtered for only the awards in a Directing category, which included “Directing”, “Directing (Comedy Picture)”, and “Directing (Dramatic Picture)”. Then we renamed some of the columns so the columns were ‘year’, ‘oscar\_year’, ‘category’, ‘nominee,’ and ‘winner’. We also dropped one column (‘ceremony’) that was not needed.

***The Golden Globes.*** We loaded the Golden Globes CSV into a Pandas data frame. We removed all the rows where there was not a film listed. We then filtered for only the awards in the “Best Director – Motion Picture” category. Then we dropped on column (‘ceremony’) that was not needed. We also renamed some of the columns so the columns were ‘year’, ‘golden\_globe\_year’, ‘category’, ‘nominee’, and ‘winner’.

***OMDb API.*** Before we made the API request, we created a list of all unique film titles from the Oscars data frame and the Golden Glove data frame. We then created an API request from OMDb using the list of unique film titles. We requested the following variables: ‘title’, ‘year’, ‘rating’, ‘box office amount’, ‘production company’, and ‘director’. And loaded them into a list.

**Load**

Within Jupyter Notebook and using Pymongo, we set up a connection to MongoDB and created an ‘award\_movies’ database. We decided to create a non-relational database because the data was semi-structured and if someone wanted to join the three collections together, the data would need to be non-relational. We created three collections, ‘oscars\_coll’, ‘golden\_globe\_coll’, and ‘omdb\_coll’. And loaded the collection with the corresponding Pandas data frame. For the ‘oscars\_coll’ and ‘golden\_globes\_coll’, we used an insert many command with transforming the data frame to a dictionary. We passed the ‘records’ argument in order to remove the index when creating the collection. For the ‘omdb\_coll’, we used an insert many command and loaded the list from the API call.

**Sources**

Fontes, R. (2020a). Golden Globe Awards, 1944-2020 (version 1) [Data file]. Available from<https://www.kaggle.com/unanimad/golden-globe-awards>.

Fontes, R. (2020b). The Oscar Award, 1927-2020 (version 7) [Data file]. Available from<https://www.kaggle.com/unanimad/the-oscar-award>.

Fritz, B. (n.d.). Open Movie Database (OMDb) API. Available from<http://www.omdbapi.com/>.