**ETL Report**

Mary Bellavia and Garrett DeBlois

02/08/2021

We pulled our data from two different Kaggle datasets, “IMDb movies extensive dataset” and “Netflix Movies and TV Shows” in the form of two csv files. We began by looking at the raw datasets and considering the final database and table schema. We chose to use the Postgres SQL relational database for our tables because the data lent itself to the relational data model. We used string or varchar for many of the fields. The avg\_votes field required a float datatype. We intentionally used the duration field from the IMDb dataset instead of the Netflix dataset so that we could use the integer datatype (the duration field in the Netflix dataset read “102 min”, “84 min”, etc). We named the final database ETL\_Database.

Once we roughly knew how we wanted our final tables to be structured, we loaded the datasets into a Jupyter Notebook and used Python’s Pandas to explore the dataset. There were a lot of unnecessary columns in both datasets. We reduced the Netflix dataset to only show movies and not TV shows. We then eliminated the excess columns from both datasets.

Once the data was filtered down and cleaned, we used an inner join to combine the datasets and, at the same time, reduce the IMDb dataset to only include Netflix movies. We chose to do an inner join on both the title and director fields of the two datasets to eliminate duplicate entries. Afterwards, we renamed the columns and then separated out the dataset. We made the movieid column the primary key for the main tables and the primary key and foreign key for the secondary table. Finally, we used SQLAlchemy to create the two tables and populate the Postgres database with information from the separate data frames.

Data analysts could use this dataset to delve deeper into the ratings of movies on Netflix. The rating of movies from different countries, different languages, or movie duration could be compared. Data analysts could also inspect duration in respect to language or country the movie was produced.