ETL Project Report

Team: Christy Wang, Louis Razuki, Matthew Belevski

Project Aim

Compile a database of Movies & TV shows that are currently on popular streaming platforms, including:

- Netflix
- Disney+
- Hulu

Part 1: Extract

Our starting point was a CSV for both Movies and TV Shows found on Kaggle, providing a list of titles for us to use in the Open Movie Database (OMDb) API. We elected to remove Amazon Prime from the data source as it dramatically reduced the dataset.

With Matt focused on Movies and Louis on TV Shows, we used the API to find the following data:

Column	Type
ID (Title + Year)	Primary Key
Title	String
Year	Integer
Director / Writer	String
Genre	String
IMDB_Rating	Decimal
IMDB_Votes	String
RottenTomatoes	Decimal
MetaCritic	Decimal
Country	String
Awards	String

Data sources:

What is it	Type	Link
IMDB Api	API	http://www.omdbapi.com/
Movies on Netflix, Disney+, Hulu and Prime	Dataset	https://www.kaggle.com/ruchi798/movies-on-netflix-prime-video-hulu-and-disney
TV Shows on Netflix, Disney+, Hulu and Prime	Dataset	https://www.kaggle.com/ruchi798/tv-shows-on-netflix-prime-video-hulu-and-disney

Part 2: Transform

We started this process by considering possible use cases for the dataset. Some example questions we proposed were:

- Which streaming platform has the most content?
- Which streaming platform has the highest rated content?

We determined that the most effective route would be to create one major table for Movies and one for TV Shows, that housed all the relevant data. We recognized that this would involve a joining of the original dataset and API data, so we decided to use a combination of movie and year as a primary key (i.e Title: Avengers, Year: 2012 = Key: Avengers2012). We then split the work up as such to prepare it for the join.

API Data

Movies (notes):

- Storing each json object retrieved from searching the OMDB API into a list and then converting it to a dataframe.
- Dropping row nulls.
- Splitting the list of dictionaries in the “Rating” column which was formatted this way due to being a nested json object. After splitting this column, only keeping the Rotten Tomatoes score and the MetaCritic score.
- Removing the % from the Rotten Tomatoes score and the /100 from the MetaCritic score and converting these columns into integers.
- Removing the commas from the IMDB votes and converting it to an integer.

TV (notes):

- The year extracted from the API was a range (i.e 2006 - 2012), so we took the first four values from each to get the release year.
- The Rotten Tomatoes score required the % to be stripped to allow statistical analysis.
- The IMDB votes had commas, which needed to be stripped and converted to an integer to allow for analysis.
- A try / except was required as many of the titles were not found by the API.

Base Data

TV & Movies (notes):

- *Data cleaning including dropping any duplicate values/rows, removing '+' from Disney plus for SQL purpose.*
- *Retaining the age column and data stream platform columns for each TV shows and movies table.*
- *Merging both cleaned datasets & API datasets together based on the primary key for each TV shows & movies.*
- *Export the final datasets into the 'load' folder.*

Part 3: Load

Using a SQL server hosted on Google Cloud Platform, we created a database in PGAdmin to house the data. We ran a query that created separate tables for Movie and TV Shows and used the 'Load CSV' function in PGAdmin to bring in our data.