**ETL-Project**

**Group 5 - Adrian, Jeremy, Laura**

Finding Data

* Use 2 sources of data
  + Data.world - Netflix Shows Dataset by Chase Willden:
    - <https://data.world/chasewillden/netflix-shows>
  + New York Times Movie Reviews API:
    - <https://developer.nytimes.com/docs/movie-reviews-api/1/overview>
  + Open Movie Database:
    - <http://www.omdbapi.com>

Data Cleanup & Analysis

* For our database we wanted to build a library of Netflix Shows and Movies. To build our database we choose Postgress SQL to build our tables and store the final data. We all felt we have a better understanding of Postgress and SQL and the visualization of the tables and columns are easier for us to understand for this short project.
* Our data sources included a CSV and 2 API calls to build this library.
* Using SQL we created tables and columns to store our data.
* Using Pandas we extracted data from the CSV and APIs, and transformed it by deleting duplicates and changing the column names so there would be fields to join on. Then the data was loaded into the database using Sqlalchemy

Final Report

To recreate our process this files should be run in the following order:

1. Create a Postgress Database called ‘Netflix\_db’
2. Then run the Netflix\_db.sql file to set up the schema (tables and columns) for the database
3. Netflix\_clean.ipynb file to clean and split the starting CSV file which should be stored in the ‘Resources’ folder
4. Netflix\_db\_setup.ipynb file to load the netflix shows and movies into the postgress database
5. NYT\_Reviews.ipynb for connect to the NYT Movie reviews and get review info and load to the postgress database
6. IMDB\_API.ipynb to connect to the OMBD and get additional info for the postgress DB

CSV from Data.World

* Extract
  + We started with a CSV of 1000 netflix shows in CSV format, this included data for both Shows and Movies
  + Using Pandas we read the csv into a Jupyternotebook
* Transform
  + On the field Rating, we noticed that there was a “TV” for TV shows so using Numpy’s Where function we created a new column to hold a new value for if that was a Show or Movie
  + Then using that new field we use Pandas’ .loc to split into two new dataframes, one of shows and one for movies
* Load
  + Using Pandas and SQL Alchemy, we loaded the data into two tables, one called Netflix\_shows and one called Netflix\_movies

API for NYT Reviews and Open Movie Database APIs

* Extract
  + Using sqlalchemy we connected to the postgress database for Neftlix\_movies
  + Using pandas we read the database into a dataframe
  + Then we connected to the New York times reviews and searched by the Movie Name
  + We stored those results in a dataframe
* Transform
  + From there we changed some of the column names so they could be easily loaded into a Postgres table and dropped a few columns we didn’t want to bring in
* Load
  + Using sql we created a table and columns for to match our pandas dataframe
  + Using pandas we used to\_sql to load the dataframe into our postgress database

Upload report to Github & Bootcamp

Github - ETL-project