Felicia Glover

Lashanti Jones

LeAnne Sevier

ETL Project: Academy Award Winners Db

Goal: The group’s goal was to be able to build a database of movies and Academy Award winners, giving the ability of a user to query movie details and link to information on Academy Awards the movie may have been nominated for and possibly won.

Extract:

Source 1 - from Kaggle - The Movies Dataset - https://www.kaggle.com/rounakbanik/the-movies-dataset

Source 2 - from Kaggle - https://www.kaggle.com/theacademy/academy-awards

Both sources were extracted as csv files and began the task of reviewing the data provided in the files, identifying information of value and potential areas requiring transformation.

Transform: We utilized Pandas to read the csv files and create data frames to begin transforming the data into usable information. We were able to identify the ‘Film’ column from the Awards file and the ‘title’ column from the Metadata file as our matching ID fields. These would be used to join data from the two sources in our queries.

We began by identifying the columns we wanted to extract and normalizing the naming convention of these columns. The Awards file contained a column using a value of ‘1’ to identify winners and blanks otherwise. We had to convert this data to a string type, then we replaced the ‘1’ values with a ‘Yes’ and the blanks as ‘No’. This will help users to more easily determine award winners from the database.

As we began the task of loading the data frames into MySQL, it was discovered that the files were encoded in Latin1, rather than UTF-8, requiring us to correct the Pandas csv read function to account for this.

Load: Our group decided the best method for storing this information was as a relational database using MySQL. We opted to house the datasets as separate tables, allowing for addition to both more easily. Originally, we created the tables using SQL statements, with the plan to append data using SQLAlchemy. However, due to the encoding, it was determined that the best course was to delete the tables and generate them using SQLAlchemy and the data frames. Once loaded, we used MySQL to successfully query the data and test the ability to join using the Film name.