ETL PROJECT REPORT – MOVIE ANALYSIS

Objective:

* For this ETL project, I wanted to see the relationships between movie budget, movie ratings, and movie directors.

Extract the Data:

* I used data from two different sources: Kaggle and IMDB. I looked at a total of 4 different data tables which combined had over 23 million rows.
* Kaggle had CSV data about movie budget by movie ids (imdb\_id), along with other metadata about the movies.
* IMDB had 3 TSV files. The first TSV file had movie ratings by movie id (imdb\_id). The second had director\_id by movie\_id (imdb\_id). The third had director\_id and director name, along with other metadata about the directors.

Transform the Data:

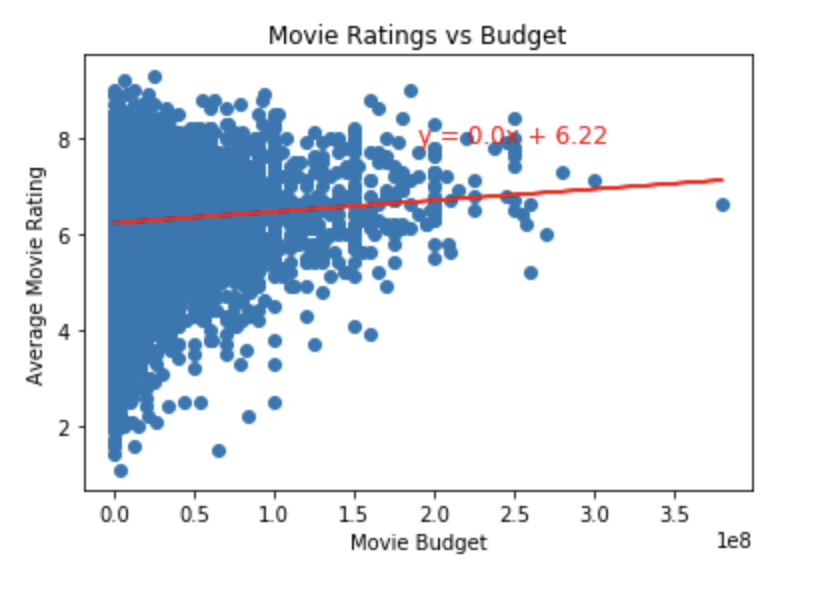
* Clean and Filter:
  + I cleaned the data by first dropping NaNs for numeric columns like budget and removing unneeded metadata columns about the movies and directors.
  + Then, I ensured that for each table there was only one row per imbd\_id because that was going to be the primary key connecting all the production data tables.
* Join and Aggregate:
  + I joined the (movie\_id, director\_id) table with the (director\_id, director\_name) table, to get the names of the directors for each movie.

Load the Data:

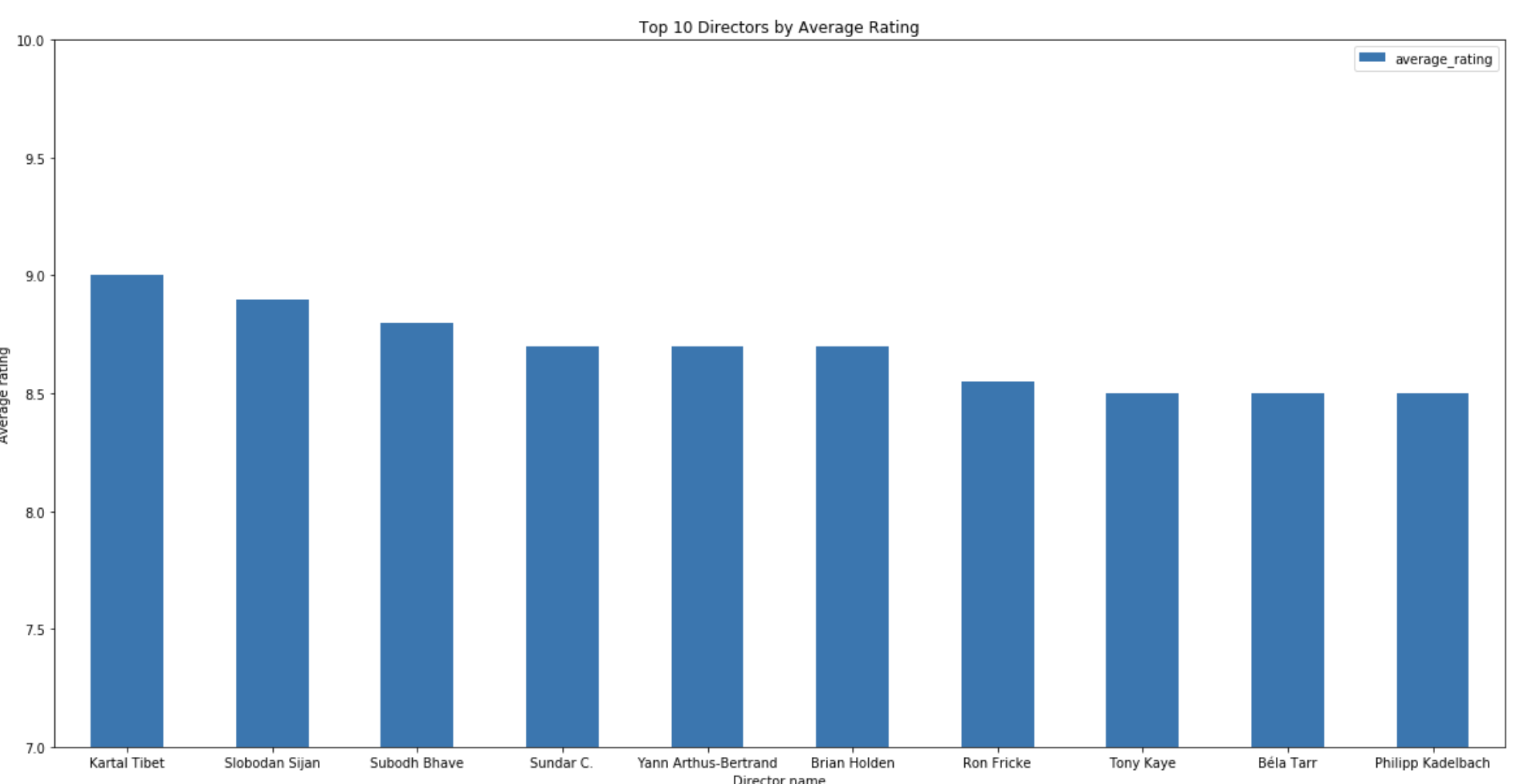
* I first created schemas for the 3 tables that were going to be loaded in the production database. The 3 tables were:
  + movie\_budget (imdb\_id, movie\_title, budget)
  + movie\_rating (imdb\_id, average\_rating)
  + movie\_director (imdb\_id, director\_id, director\_name).
* I decided to use a MySQL database, as opposed to MongoDB because all my data was extracted from CSVs and had the imdb\_id as a common column. Consequently, I made the imdb\_id the Primary Key for all 3 tables, so that I could join on it, to do analysis later.
* I used PGAdmin to create the database, and loaded the schemas to create the 3 tables. Then I used SQLAlchemy and Pandas to load the data into the created tables.

Analyze the Data:

* I joined the movie\_budget table with the movie\_rating table, on imdb\_id, so that I could explore the relationship between movie budget and movie ratings. I plotted the relationship on a scatter plot using matplotlib, and calculated the regression line. I found that low budget movies have variable ratings (high or low ratings), but high budget movies generally have higher ratings.



* I joined the movie\_director table with the movie\_rating table, on imdb\_id, so that I could see the highest rated directors in the dataset. I grouped by director and averaged their movie ratings. Then I sorted from high to low and plotted a bar graph with the 10 highest rated directors in the dataset.



Caveats:

* The Kaggle dataset only had movies that were released before July 2017, but the IMDB dataset did not specify the dates. So some movies with ratings (IMDB table) didn’t have a budget (Kaggle table) or vice versa.

Learnings:

* I learned that SQL schemas are not case sensitive. The original column was “averageRating”, but when I went to push the data to the database, it errored out. So then I changed the column title and schema to be “average\_rating”, which then solved the error.
* I learned that SQL primary keys have to be unique. Some tables had multiple rows per imdb\_id, which would cause the data load to fail. So I had to remove the duplicates, which were ~10 rows out of 8800.
* I learned that every row of a SQL column must adhere to the type in the schema. Some budget rows (which was an INT type column) had strings in them, due to some data corruption in the CSV. I had to drop all non INT rows, to successfully load to the database, which ended up being just 2 rows out of 8800.