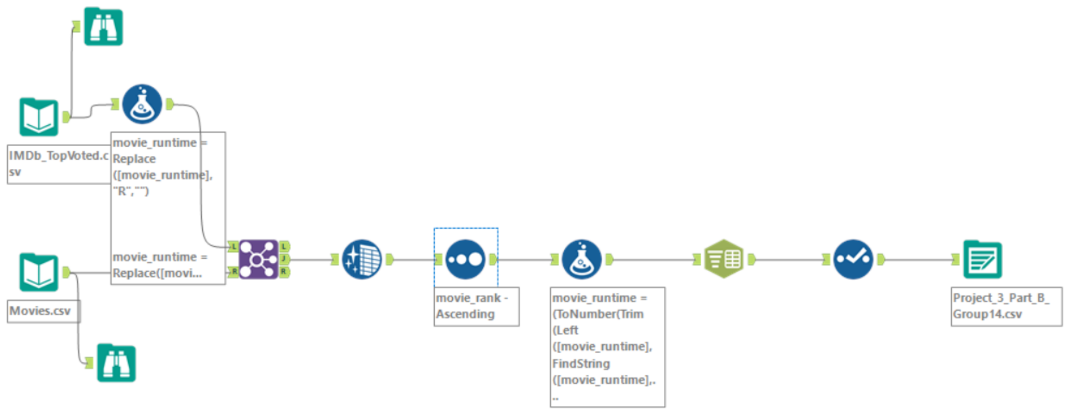
Data Wrangling Project 3 Part B Report

Screenshot of PART B Alteryx workflow


IMDb_TopVoted.csv
movie_runtime = Replace ([movie_runtime], "R","")
movie_runtime = Replace([movi...
Movies.csv
L
J
R
movie_rank - Ascending
movie_runtime = (ToNumber(Trim (Left ([movie_runtime], FindString ([movie_runtime],. ..
Project_3_Part_B_Group14.csv

Our approach to cleaning and transforming data involved a systematic method aimed at enhancing the quality and relevance of the "imdb.top-voted.csv" dataset. Initially, we eliminated leading and trailing whitespaces, tabs, line breaks, and duplicate whitespaces to establish a clean foundation for the dataset. To ensure the absence of trailing whitespace and line breaks in our client data, we implemented this step after merging the two datasets.

Subsequently, a touch of sophistication was added as the formula function seamlessly converted hours into minutes, ensuring consistency in the temporal features of the dataset. An intricate curation process was then employed to refine the dataset's narrative by delicately removing any undesirable terms in the movie runtime column.

Further refinement involved addressing unnecessary symbols such as "," in the "votes" column and "." in the movie title, utilizing formula functions to eliminate them. This meticulous process resulted in a refined dataset, ready for merging with the "movies.csv" dataset.

What data was used to enrich the client's data?

The dataset derived from "imdb.top-voted.csv" played a crucial role in enhancing and broadening the client's existing data repository. Careful consideration was given to the web extraction of this dataset using the jupyter notebook, which has list inclusive of movie_id, movie_rank, movie_title, movie_runtime, movie_year, movie_rating, and movie_votes. Each of these variables adds distinct informational elements, collectively enhancing the overall comprehensiveness and granularity of the client's data environment.

Describe the data cleaning and transformation that was implemented in Alteryx.

Importing the data sets

Initially, we imported two datasets, namely "imdb.top-voted.csv" (df2) – obtained through web scraping from the IMDB webpage using Jupyter notebook – and "movies.csv" (df1), provided via Canvas.

Data enhancing and cleansing with formula:

The data cleansing and transformation process for df2 began with the use of formula functions to enhance "IMDb.top-voted.csv." The runtime data in df2 contained attached ratings such as PG-13 and R. To address this, we utilized the formula function Replace([movie_runtime], "R", "") to remove these rating letters. Additionally, we removed '.' from the title and ',' from the votes simultaneously using the following functions:

1. ToNumber(REGEX_Replace([movie_votes], ",", "")): This was done to transform the datatype to int without losing dataset values, facilitating later conversion of the hour-minute format to minutes.
2. TrimLeft([movie_title], "."): This step removed unnecessary dots at the beginning of some titles, ensuring uniformity in the title column.

Data merge:

Subsequently, we employed the join function to seamlessly merge the two CSV files. Since there were two movie IDs during the join, we unticked the left movie ID, proceeding only with the right movie ID.

Data cleansing:

With both datasets merged, we applied data cleansing functions to remove leading and trailing white spaces, tabs, line spaces, and duplicate whitespaces in the columns from both df1 and df2.

Data Sorting:

Further, a sorting function was utilized to arrange the merged data in ascending order of rank, as instructed.

Data conversion to runtime minutes:

To convert runtime into minutes, we used the formula function:

```
(ToNumber(Trim(Left([movie_runtime], FindString([movie_runtime], 'h')))) * 60) +
ToNumber(Trim(Substring([movie_runtime], FindString([movie_runtime], 'h')+1,
FindString([movie_runtime], 'm')-FindString([movie_runtime], 'h')-1)))
```

Genre split:
A critical refinement involved using the "text to columns" filter to separate the genres column into distinct categories: genre1, genre2, and genre3. Some genre columns had empty rows, reflecting cases where not all movies had all three genres.

Select function:
Using the select function, we carefully arranged the columns in the desired sequence: movie_id, rank, title, originalTitle, description, year, votes, rating, runtimeMinutes, ratingCategory, and genres – in accordance with Canvas instructions. Additionally, the data types of the rank, year, and votes columns were changed to int. Convenient names were assigned, such as changing "right movie_id" to "movie_id." The select function was also used to exclude the original genre column.

Conclusion:
This thorough process ensures that the final dataset is not only coherent and well-organized but also tailored to meet specific requirements, providing a comprehensive view of the movie data. Finally, a csv file was exported.