In [ ]:
```
# Supress Warnings

import warnings
warnings.filterwarnings('ignore')
```

In [ ]:
```
# Import the numpy and pandas packages

import numpy as np
import pandas as pd
```

Task 1: Reading and Inspection

- ## Subtask 1.1: Import and read

Import and read the movie database. Store it in a variable called `movies`.

In [ ]:
```
movies = # Write your code for importing the csv file here
movies
```

- ## Subtask 1.2: Inspect the dataframe

Inspect the dataframe's columns, shapes, variable types etc.

In [ ]:
```
# Write your code for inspection here
```

Task 2: Cleaning the Data

- ## Subtask 2.1: Inspect Null values

Find out the number of Null values in all the columns and rows. Also, find the percentage of Null values in each column. Round off the percentages upto two decimal places.

In [ ]:
```
# Write your code for column-wise null count here
```

In [ ]:
```
# Write your code for row-wise null count here
```

In [ ]:
```
# Write your code for column-wise null percentages here
```

- ## Subtask 2.2: Drop unecessary columns

For this assignment, you will mostly be analyzing the movies with respect to the ratings, gross collection, popularity of movies, etc. So many of the columns in this dataframe are not required. So it is advised to drop the following columns.

- color
- director_facebook_likes
- actor_1_facebook_likes
- actor_2_facebook_likes
- actor_3_facebook_likes
- actor_2_name
- cast_total_facebook_likes
- actor_3_name
- duration
- facenumber_in_poster
- content_rating
- country
- movie_imdb_link
- aspect_ratio
- plot_keywords

In [ ]:
```
# Write your code for dropping the columns here. It is advised to keep inspect
```

Subtask 2.3: Drop unecessary rows using columns with high Null percentages

Now, on inspection you might notice that some columns have large percentage (greater than 5%) of Null values. Drop all the rows which have Null values for such columns.

In [ ]:
```
# Write your code for dropping the rows here
```

Subtask 2.4: Drop unecessary rows

Some of the rows might have greater than five NaN values. Such rows aren't of much use for the analysis and hence, should be removed.

In [ ]:
```
# Write your code for dropping the rows here
```

Subtask 2.5: Fill NaN values

You might notice that the `language` column has some NaN values. Here, on inspection, you will see that it is safe to replace all the missing values with `'English'`.

In [ ]:
```
# Write your code for filling the NaN values in the 'language' column here
```

• Subtask 2.6: Check the number of retained rows

You might notice that two of the columns viz. `num_critic_for_reviews` and `actor_1_name` have small percentages of NaN values left. You can let these columns as it is for now. Check the number and percentage of the rows retained after completing all the tasks above.

In [ ]:
```
# Write your code for checking number of retained rows here
```

Checkpoint 1: You might have noticed that we still have around `77%` of the rows!

Task 3: Data Analysis

• Subtask 3.1: Change the unit of columns

Convert the unit of the `budget` and `gross` columns from `$` to `million $`.

In [ ]:
```
# Write your code for unit conversion here
```

• Subtask 3.2: Find the movies with highest profit

1. Create a new column called `profit` which contains the difference of the two columns: `gross` and `budget`.
2. Sort the dataframe using the `profit` column as reference.
3. Extract the top ten profiting movies in descending order and store them in a new dataframe - `top10`

In [ ]:
```
# Write your code for creating the profit column here
```

In [ ]:
```
# Write your code for sorting the dataframe here
```

In [ ]:
```
top10 = # Write your code to get the top 10 profiting movies here
```

• Subtask 3.3: Drop duplicate values

After you found out the top 10 profiting movies, you might have notice a duplicate value. So, it seems like the dataframe has duplicate values as well. Drop the duplicate values from the dataframe and repeat `Subtask 3.2`.

In [ ]:
```
# Write your code for dropping duplicate values here
```

In [ ]:
```
# Write code for repeating subtask 2 here
```

Checkpoint 2: You might spot two movies directed by `James Cameron` in the list.

• Subtask 3.4: Find IMDb Top 250

1. Create a new dataframe `IMDb_Top_250` and store the top 250 movies with the highest IMDb Rating (corresponding to the column: `imdb_score`). Also make sure that for all of these movies, the `num_voted_users` is greater than 25,000. Also add a `Rank` column containing the values 1 to 250 indicating the ranks of the corresponding films.
2. Extract all the movies in the `IMDb_Top_250` dataframe which are not in the English language and store them in a new dataframe named `Top_Foreign_Lang_Film`.

In [ ]:
```
# Write your code for extracting the top 250 movies as per the IMDb score here
# and name that dataframe as 'IMDb_Top_250'
```

In [ ]:
```
Top_Foreign_Lang_Film = # Write your code to extract top foreign language film
```

Checkpoint 3: Can you spot `Veer-Zaara` in the dataframe?

• Subtask 3.5: Find the best directors

1. Group the dataframe using the `director_name` column.
2. Find out the top 10 directors for whom the mean of `imdb_score` is the highest and store them in a new dataframe `top10director`.

In [ ]:
```
# Write your code for extracting the top 10 directors here
```

Checkpoint 4: No surprises that `Damien Chazelle` (director of Whiplash and La La Land) is in this list.

• Subtask 3.6: Find popular genres

You might have noticed the `genres` column in the dataframe with all the genres of the movies seperated by a pipe (`|`). Out of all the movie genres, the first two are most significant for any film.

1. Extract the first two genres from the `genres` column and store them in two new columns: `genre_1` and `genre_2`. Some of the movies might have only one genre. In such cases, extract the single genre into both

the columns, i.e. for such movies the `genre_2` will be the same as `genre_1`.
2. Group the dataframe using `genre_1` as the primary column and `genre_2` as the secondary column.
3. Find out the 5 most popular combo of genres by finding the mean of the gross values using the `gross` column and store them in a new dataframe named `PopGenre`.

In [ ]:
```
# Write your code for extracting the first two genres of each movie here
```

In [ ]:
```
movies_by_segment = # Write your code for grouping the dataframe here
```

In [ ]:
```
PopGenre = # Write your code for getting the 5 most popular combo of genres he
```

Checkpoint 5: Well, as it turns out. `Family + Sci-Fi` is the most popular combo of genres out there!

- ## Subtask 3.7: Find the critic-favorite and audience-favorite actors

1. Create three new dataframes namely, `Meryl_Streep`, `Leo_Caprio`, and `Brad_Pitt` which contain the movies in which the actors: 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' are the lead actors. Use only the `actor_1_name` column for extraction. Also, make sure that you use the names 'Meryl Streep', 'Leonardo DiCaprio', and 'Brad Pitt' for the said extraction.
2. Append the rows of all these dataframes and store them in a new dataframe named `Combined`.
3. Group the combined dataframe using the `actor_1_name` column.
4. Find the mean of the `num_critic_for_reviews` and `num_user_for_review` and identify the actors which have the highest mean.

In [ ]:
```
# Write your code for creating three new dataframes here

Meryl_Streep = # Include all movies in which Meryl_Streep is the lead
```

In [ ]:
```
Leo_Caprio = # Include all movies in which Leo_Caprio is the lead
```

In [ ]:
```
Brad_Pitt = # Include all movies in which Brad_Pitt is the lead
```

In [ ]:
```
# Write your code for combining the three dataframes here
```

In [ ]:
```
# Write your code for grouping the combined dataframe here
```

In [ ]:
```
# Write the code for finding the mean of critic reviews and audience reviews h
```

Checkpoint 6: `Leonardo` has aced both the lists!