How to run the project

Rowyda Askalani 888387
Nastaran Janialiabadi 922779
Sushil Kumar Chaurasiya 925329

In order to replicate the project and avoid running into any issues, it is required to follow the following steps:

1. Create a drive folder in which you upload the IMDB dataset CSV and the Language CSV for further use
2. Colab Notebook
    a. Setup
        i. Run the colab notebook
        ii. Mount the drive
        iii. Change the directory to your desired location in which you stored/will store the CSV files
    b. Extracting TMDB Data Using API
        i. run the language query
        ii. To extract the movie data you should:
            1. Uncomment the extraction cell by removing the ‘ “”” ‘ symbols at the beginning and ending of the cell
            2. Change the API_KEY to your own key
            3. Run the cell
    c. For the profiling it is possible to view the report either by running the notebook and observing the results directly or to download the reports from the shared drive and open them through a browser
    d. Run the rest of the notebook as you normally would
    e. Finally, in the last cell you obtain the refined CSV that will be used for MySQL, which is stored in the directory defined
3. MySQL
    a. The database was created using the following SQL commands:
    CREATE DATABASE IF NOT EXISTS movies_db;
    USE movies_db;
    b. The table was created with the following SQL commands:
    CREATE TABLE IF NOT EXISTS Movies (
    id INT AUTO_INCREMENT,
    Title VARCHAR(255),
    Genre VARCHAR(255),
    Description TEXT,
    Director VARCHAR(255),
    Actors TEXT,
    Year INT,
    Runtime_Minutes INT,
    Rating_imdb FLOAT,
    Votes_imdb INT,

```
    Revenue_Millions FLOAT,
    Metascore FLOAT,
    Rating_tmdb FLOAT,
    Votes_tmdb INT,
    original_title VARCHAR(255),
    popularity FLOAT,
    language VARCHAR(50),
    PRIMARY KEY (id)
);
```

c. The data importation was performed with the following SQL commands:

```
LOAD DATA INFILE 'C:/ProgramData/MySQL/MySQL Server
8.0/Uploads/Final_df02.csv'
INTO TABLE Movies
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n'
IGNORE 1 ROWS
(id, Title, Genre, Description, Director, Actors, Year, Runtime_Minutes,
Rating_imdb, Votes_imdb, Revenue_Millions, Metascore, Rating_tmdb,
Votes_tmdb, original_title, popularity, language);
```

d. To analyze the relationship between genres and other variables more effectively, we decided to decompose the composite Genre column into three distinct columns: “Genre_1”, “Genre_2”, and “Genre_3”. This is done through 2 Steps:

   i.
```
ALTER TABLE movies
ADD COLUMN Genre_1 VARCHAR(255) NULL,
ADD COLUMN Genre_2 VARCHAR(255) NULL,
ADD COLUMN Genre_3 VARCHAR(255) NULL;
```

   ii.
```
UPDATE movies
SET
   Genre_1 = SUBSTRING_INDEX(Genre, ',', 1),
   Genre_2 =
NULLIF(SUBSTRING_INDEX(SUBSTRING_INDEX(Genre, ',', 2), ',',
-1), Genre_1),
   Genre_3 = NULLIF(SUBSTRING_INDEX(Genre, ',', -1), Genre_2);
```

e. To Obtain the query result for the research question the following queries were written:

   i. Do Popularity and Metascore Correspond to the Ratings Received?

      1.
```
SELECT title, rating_imdb, rating_tmdb, popularity, metascore
FROM movies
ORDER BY popularity DESC
LIMIT 10;
```

   ii. Is There a Correlation Between Ratings, Popularity, and Revenue?

      1.
```
SELECT
   ROUND(CORR(rating_imdb, revenue_millions), 2) AS
correlation_rating_revenue,
   ROUND(CORR(popularity, revenue_millions), 2) AS
correlation_popularity_revenue,
```

ROUND(CORR(rating_imdb, popularity), 2) AS correlation_rating_popularity
FROM movies;

iii. Is There a Correlation Between Genres and Metascores?

1. SELECT genre_1 AS genre, AVG(metascore) AS avg_metascore
FROM movies
WHERE genre_1 IS NOT NULL
GROUP BY genre_1
ORDER BY avg_metascore DESC;