Self-Practice Exercise – MovieLens Database & EDA

In this exercise, you will practice the full data workflow using ONLY the tools demonstrated in the lecture notebook: NumPy, Pandas, MySQL (via SQLAlchemy), Matplotlib/Seaborn, and Automated EDA packages. You will download a public dataset (MovieLens), save it into a local MySQL database, retrieve it with Pandas, and perform both manual and automatic exploratory data analysis (EDA).

Dataset

We will use the MovieLens dataset (public ZIP archive).

Download: http://files.grouplens.org/datasets/movielens/ml-latest-small.zip

Contained files:
- ratings.csv → columns: userId, movieId, rating, timestamp
- movies.csv → columns: movieId, title, genres

Step 1 – Download & Extract the ZIP

Download and preview the dataset using Pandas (directly extract from the ZIP file):

```python
import pandas as pd
import requests
from zipfile import ZipFile
from io import BytesIO

url = 'http://files.grouplens.org/datasets/movielens/ml-latest-small.zip'
r = requests.get(url)
z = ZipFile(BytesIO(r.content))

ratings = pd.read_csv(z.open('ml-latest-small/ratings.csv'))
movies  = pd.read_csv(z.open('ml-latest-small/movies.csv'))

ratings.head()
movies.head()
```

Step 2 – Save to MySQL

1. Create a new database:

```sql
CREATE DATABASE movielens;
USE movielens;
```

2. Create tables:

```sql
CREATE TABLE ratings (
    userId INT,
    movieId INT,
    rating FLOAT,
    timestamp BIGINT
);

CREATE TABLE movies (
    movieId INT,
    title VARCHAR(255),
    genres VARCHAR(255)
);
```

3. Insert using Pandas + SQLAlchemy:

```python
from sqlalchemy import create_engine

engine =
create_engine('mysql+mysqlconnector://user:password@localhost:3306/movielens')
ratings.to_sql('ratings', engine, if_exists='replace', index=False)
movies.to_sql('movies', engine, if_exists='replace', index=False)
```

Step 3 – Query Back into Pandas

Retrieve the data from MySQL into a DataFrame:

```python
ratings = pd.read_sql('SELECT * FROM ratings', engine)
movies  = pd.read_sql('SELECT * FROM movies', engine)
```

Step 4 – Manual EDA

Perform EDA using Pandas, Matplotlib, and Seaborn:
- Inspect structure with `info()` and `describe()`
- Join ratings with movies
- Top 10 movies by average rating (with minimum number of ratings)
- Histogram of ratings distribution
- Average rating per genre (expand genre column if needed)
- Scatterplot of rating count vs average rating

Step 5 – Automatic EDA

Repeat the analysis using an **automatic EDA package** shown in the notebook (e.g., pandas_profiling or sweetviz).
- Generate a profile report for the ratings and movies data
- Compare automated insights with your manual EDA findings

Tasks

1. Download and extract the ZIP, then inspect the first 5 rows of ratings and movies.
2. Create the movielens database and load the data into MySQL.
3. Query back into Pandas and confirm row counts match.
4. Perform MANUAL EDA:
   - Top 10 highest-rated movies (with at least 20 ratings)
   - Average rating by genre
   - Histogram of ratings
   - Scatterplot of rating count vs average rating
5. Perform AUTOMATIC EDA:
   - Generate an automated EDA report
   - Compare automated vs manual findings