Movie Ratings Project Report

Problem Statement :- Movie Ratings Project Report

Dashboard to analyze movie ratings data(from a CSV dataset) and provide actionable insights.

Type: Pandas-based (no visualization) mini case study

Objective: This project focuses on analyzing a movie ratings dataset to extract insights about user preferences, movie popularity, and genre performance. The goal is to practice Pandas operations, especially pivot tables, joins, grouping, and aggregation.

Name – Lekhana Suresh(4GW23CI023)

Email – lekhanasureshsuresh@gmail.com

➢ INDEX

1. Project Overview

The **Movie Ratings Dashboard** is an interactive analytics platform built with **Python, Pandas, Matplotlib, and Streamlit**. Its purpose is to clean, merge, analyze, and visualize data from three CSV files: **movies.csv**, **ratings.csv**, and **users.csv**. By unifying these sources, the project enables quick insights into movie performance, user activity, and genre trends.

Problem: Raw ratings data is fragmented across multiple files, making analysis slow and error-prone.
Goal: Provide a clear, reusable tool to verify merges, explore pivot tables, visualize trends, and export clean datasets.

Key Features

- **Data Preparation:** Deduplication, missing value handling, and inner joins on `MovieID` and `UserID`.
- **Verification (Step 9):** Row counts at each merge stage for data integrity.
- **Pivot Tables:** Average ratings per Movie, Genre, and User.
- **Analysis:**
  - High-rated movies (avg ≥ 4.0).
  - Active users (rated > 5 movies).
  - Top 5 movies by number of ratings.
  - Highest & lowest rated titles.
- **Derived Fields:**
  - `RatingCategory` (Low/Medium/High).
  - `IsPopular` (Yes if >10 ratings).
- **Visualizations:** Bar charts (movies, users, genres) and a pie chart (genre distribution).
- **Export:** Aggregated CSVs (movie, genre, user averages) plus the full cleaned dataset.

2. Use-Case Explanations

The **Movie Ratings Dashboard** serves multiple practical use cases across business, education, and technology. Each use case highlights how different stakeholders benefit from the system's ability to clean, merge, analyze, and visualize rating data.

1. Movie Insights for Business Teams

Studios, streaming platforms, or marketers can quickly identify:

- **Top-performing movies** by average rating.
- **Most popular movies** based on rating counts.
- **High-rated movies (≥ 4.0)** suitable for promotion or recommendation.
  These insights enable targeted campaigns, licensing decisions, and improved catalog curation.

2. Genre-Based Trends for Content Strategy

The dashboard splits multi-genre movies into individual categories and calculates **average ratings per genre**. Decision-makers can:

- Spot which genres consistently earn higher ratings.
- Allocate budgets and production focus accordingly.
- Understand audience preferences for better content planning.

3. User Behavior & Engagement

By analysing user-level pivot tables:

- **Active users** (those rating more than 5 movies) can be identified as engaged audience segments.
- **Top users by average rating** highlight individuals with strong opinions that might influence communities.

- User-level insights support loyalty programs or beta-testing of new content.

4. Data Preparation for Machine Learning

The cleaned, merged, and enriched dataset (`full_df`) provides valuable inputs for building **recommendation systems**. Features such as:

- `RatingCategory` (Low/Medium/High).
- `IsPopular` flag (>10 ratings).
- User/movie/genre-level averages.
  These can serve as engineered features in collaborative filtering or content-based filtering models.

5. Academic & Learning Use Case

The project also functions as a **teaching tool** for:

- Demonstrating **data wrangling** with Pandas (joins, groupby, pivot).
- Explaining **data visualization** using bar and pie charts.
- Showing end-to-end **project structure**: data loading → transformation → analysis → visualization → export.
  Students and educators can easily replicate the steps to learn applied data science.

6. Reporting & Export

With one click, stakeholders can export:

- Average rating per movie (`movie_avg_ratings.csv`).
- Average rating per genre (`genre_avg_ratings.csv`).
- Average rating per user (`user_avg_ratings.csv`).

- Fully cleaned dataset (`cleaned_movie_ratings.csv`). This supports **sharing insights** across teams and integrating with other BI or analytics tools.

3. Algorithms / Approach

1. Data Cleaning
   - Remove duplicates from movies, ratings, users datasets.
   - Handle missing values in ratings (replace with 0).

2. Data Merging
   - Merge ratings with movies using MovieID.
   - Merge the result with users using UserID.

3. Pivot Table Calculations
   - Average rating per Movie.
   - Average rating per Genre.
   - Average rating per User.

4. Analysis Rules
   - High-rated movies: movies with average rating ≥ 4.0.
   - Active users: users who rated more than 5 movies.
   - Popular movies: movies with >10 ratings.
   - Categorization: Ratings split into Low (0–2.9), Medium (3–3.9), High (4–5).

5. Visualization
   - Bar charts for movies, users, genres.
   - Pie chart for genre distribution.

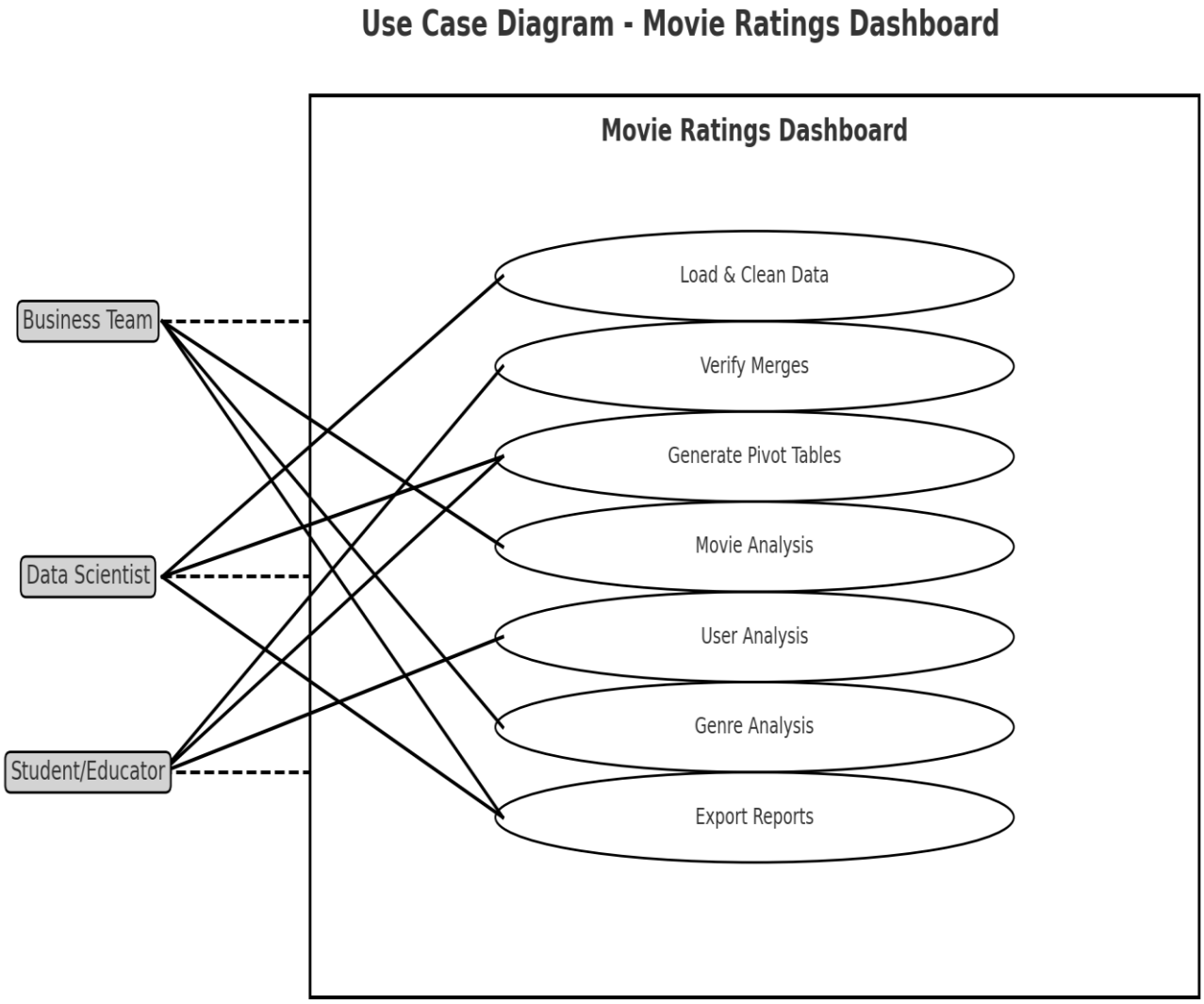
4. UML Diagrams

1. Use Case Diagram

The Use Case Diagram illustrates how different actors interact with the Movie Ratings Dashboard system. The primary actor is the **User** (such as a data analyst, student, or movie enthusiast). The user can perform the following use cases:

- **Load Data:** Upload or load movies, ratings, and user datasets into the system.
- **Clean & Merge Data:** The system automatically removes duplicates, fills missing values, and merges movies with ratings and user details.
- **View Pivot Tables:** Users can view summarized pivot tables for movies, genres, and users.
- **Movie Analysis:** Analyze average ratings, most popular movies, and highest/lowest rated movies.
- **User Analysis:** View active users, their rating behavior, and average ratings.
- **Genre Analysis:** Compare genres by average ratings, distribution, and charts.
- **Export Data:** Download processed datasets and reports as CSV files for further use.

This diagram provides a high-level functional view, showing what the user can achieve without going into technical details.



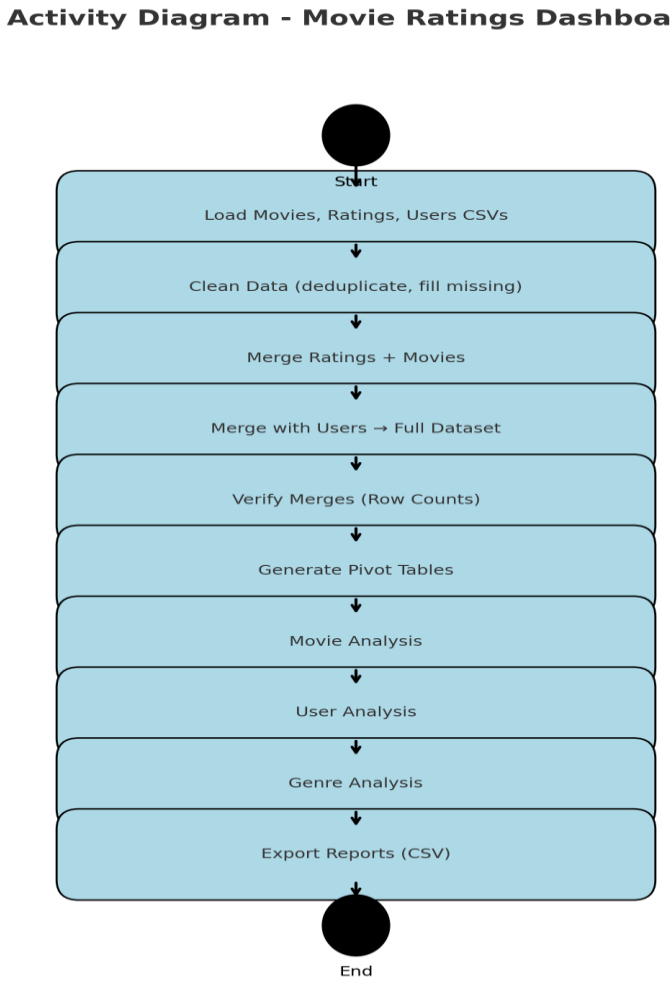
2. Activity Diagram

The Activity Diagram shows the **workflow of the system** from start to end:

1. **Start** → The process begins when the user opens the dashboard.
2. **Load CSV Files** → Movies, Ratings, and Users datasets are imported.

3. **Clean Data** → The system removes duplicates and fills missing values.
4. **Merge Datasets** → Ratings are merged with movies, then with users, forming the complete dataset.
5. **Verify Data** → Row counts are checked to ensure successful merges.
6. **Generate Pivot Tables** → Summarized tables for movies, genres, and users are created.
7. **Analysis** → The system branches into three types of analysis:
    - **Movie Analysis** (top movies, most rated, highest/lowest ratings)
    - **User Analysis** (active users, average ratings)
    - **Genre Analysis** (average ratings per genre, pie charts, bar charts)
8. **Export Data** → Users can save outputs (CSV files).
9. **End** → The session completes after exploration or export.

This diagram reflects the **step-by-step flow** and highlights system automation (cleaning, merging, verifying) along with user



decisions (choosing the type of analysis).

Together, these UML diagrams clarify **what the system does (Use Case)** and **how it does it (Activity Flow)** in a concise, visual format.

5. Front-End (Interface) Design

The Streamlit Dashboard has:

- Sidebar Navigation:

- Home (Merge Verification)
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data

- Main Content:
- Metrics (Movies, Ratings, Users, etc.)
- Preview tables (merged data, pivot tables)
- Charts (bar, pie)
- Export options

6. Setup Instructions

1. Install Python 3.9+
2. Install required libraries:
   pip install streamlit pandas matplotlib

3. Place the datasets in the project folder:
   - movies.csv
   - ratings.csv
   - users.csv

4. Run the Streamlit app:
   streamlit run app.py

5. Open browser → http://localhost:8501

7.Code & Explanation

Click here to view the full source code app_code.txt

1. Streamlit Setup

```
import streamlit as st
import pandas as pd
import matplotlib.pyplot as plt

st.set_page_config(page_title="Movie Ratings
Project", layout="wide")

st.markdown("## 🎬 Movie Ratings Project
Dashboard")
st.write("This app performs data cleaning,
merging, pivot tables, and visualization.")
```

Explanation:

- Imported **Streamlit** (for web UI), **Pandas** (for data handling), and **Matplotlib** (for charts).
- `st.set_page_config()` sets the app's title and wide layout.
- `st.markdown()` and `st.write()` add a styled heading and a description at the top of the dashboard.

---

2. Data Loading & Cleaning

```
@st.cache_data
def load_data():
    movies = pd.read_csv("movies.csv")
    ratings = pd.read_csv("ratings.csv")
    users = pd.read_csv("users.csv")
```

```
    ratings['Rating'] =
ratings['Rating'].fillna(0)
    movies = movies.drop_duplicates()
    ratings = ratings.drop_duplicates()
    users = users.drop_duplicates()

    ratings_movies = pd.merge(ratings,
movies, on="MovieID", how="inner")
    full_df = pd.merge(ratings_movies,
users, on="UserID", how="inner")

    return movies, ratings, users,
ratings_movies, full_df
```

Explanation:

- Reads three datasets (**movies, ratings, users**) using Pandas.
- Cleans the data: fills missing ratings with 0 and removes duplicates.
- Performs **two merges**:
  - Ratings + Movies → `ratings_movies`
  - `ratings_movies` + Users → `full_df` (final dataset).
- Uses `@st.cache_data` to improve performance by caching the results so the data isn't reloaded on every page refresh.

---

3. Sidebar Navigation

```
st.sidebar.header("Navigation")
page = st.sidebar.radio(
    "Go to",
    ["Home","Pivot Tables","Movie
Analysis","User Analysis","Genre
Analysis","Export Data"]
```

```
)
```

Explanation:

- Adds a sidebar menu with radio buttons for navigation.
- The user can switch between **different pages** of the dashboard like Home, Analysis, Export, etc.

---

4. Pivot Tables

```
pivot_movie =
full_df.pivot_table(index="Title",
values="Rating", aggfunc="mean")
pivot_genre = (

full_df.assign(Genre=full_df['Genre'].str.sp
lit('|')).explode('Genre')
    .pivot_table(index="Genre",
values="Rating", aggfunc="mean")
)
pivot_user =
full_df.pivot_table(index="UserID",
values="Rating", aggfunc="mean")
```

Explanation:

- **Movie Pivot:** Finds the average rating per movie.
- **Genre Pivot:** Splits movies with multiple genres, explodes them into rows, then calculates average rating per genre.
- **User Pivot:** Finds average rating given by each user.
- These pivot tables are the backbone for later analysis and visualizations.

---

5. Extra Calculations

```
high_rated_movies =
pivot_movie[pivot_movie["Rating"] >= 4.0]
user_counts =
full_df.groupby("UserID").size()
active_users = user_counts[user_counts > 5]
movie_counts =
full_df.groupby("Title").size().sort_values(
ascending=False).head(5)
highest_rated =
pivot_movie["Rating"].idxmax(),
pivot_movie["Rating"].max()
lowest_rated =
pivot_movie["Rating"].idxmin(),
pivot_movie["Rating"].min()
```

Explanation:

- **High Rated Movies:** Movies with avg rating ≥ 4.0.
- **Active Users:** Users who rated more than 5 movies.
- **Top 5 Movies:** Movies with the most ratings.
- **Highest & Lowest:** Identifies the movie with the highest and lowest average ratings.

---

6. Feature Engineering

```
full_df["RatingCategory"] = pd.cut(
    full_df["Rating"], bins=[0, 2.9, 3.9,
5], labels=["Low", "Medium", "High"]
)
movie_rating_counts =
full_df.groupby("Title")["Rating"].transform
("count")
```

```
full_df["IsPopular"] =
movie_rating_counts.apply(lambda x: "Yes" if
x > 10 else "No")
```

Explanation:

- Adds **RatingCategory** column by classifying ratings into Low, Medium, High.
- Adds **IsPopular** column: marks movies with >10 ratings as "Yes", otherwise "No".
- Helps in segmentation and visualization.

---

7. Visualization Helpers

```
def plot_chart(df, title, xlabel, ylabel,
top_n=None, color="skyblue"):
    ...
def plot_pie_chart(df, title):
    ...
```

Explanation:

- **plot_chart():** Creates bar charts (Top N items like movies, users, genres).
- **plot_pie_chart():** Creates pie charts (distribution of ratings across genres).
- Both functions use Matplotlib and display the result in Streamlit using `st.pyplot()`.

---

8. Pages

Home Page

```
if page == "Home":
```

```
    st.markdown("### ✅ Step 9: Merge
Verification (Row Counts)")
    col1, col2, col3, col4, col5 =
st.columns(5)
    col1.metric("Movies", movies.shape[0])
    ...
    st.dataframe(full_df,
use_container_width=True)
```

Explanation:

- Displays row counts of datasets before and after merging using **Streamlit metrics**.
- Shows a preview of the merged dataset in a table.

Pivot Tables Page

```
elif page == "Pivot Tables":
    st.subheader("🎯 Pivot Tables")
    st.dataframe(pivot_movie.head(10))
    st.dataframe(pivot_genre)
    st.dataframe(pivot_user.head(10))
```

Explanation:

- Displays pivot tables directly in interactive Streamlit tables.

Movie Analysis Page

```
elif page == "Movie Analysis":
    plot_chart(pivot_movie, "Top 10 Movies",
"Movie", "Avg Rating", top_n=10)
    st.dataframe(high_rated_movies)
    st.dataframe(movie_counts)
    st.write(f"🏆 Highest Rated Movie:
{highest_rated[0]}")
```

Explanation:

- Shows Top 10 movies (bar chart).
- Displays movies with rating ≥4.0.
- Displays most-rated movies.
- Shows highest and lowest rated movie with scores.

User Analysis Page

```
elif page == "User Analysis":
    plot_chart(pivot_user, "Top 10 Users",
"UserID", "Avg Rating", top_n=10,
color="lightgreen")
    st.dataframe(active_users)
```

Explanation:

- Shows users who rate the most.
- Displays Top 10 users by average ratings.

Genre Analysis Page

```
elif page == "Genre Analysis":
    col1, col2 = st.columns(2)
    with col1: plot_chart(pivot_genre,
"Average Rating per Genre", "Genre", "Avg
Rating")
    with col2: plot_pie_chart(pivot_genre,
"Genre Ratings Distribution")
```

Explanation:

- Displays genre-wise average ratings (bar chart).
- Shows percentage contribution of each genre (pie chart).

Export Data Page

```
elif page == "Export Data":

pivot_movie.to_csv("movie_avg_ratings.csv")
```

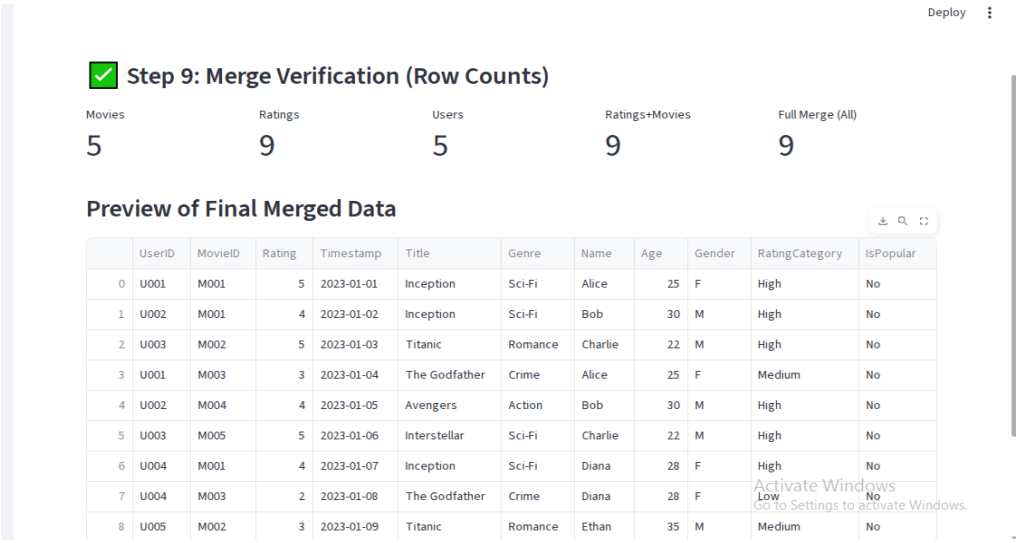
```
pivot_genre.to_csv("genre_avg_ratings.csv")

pivot_user.to_csv("user_avg_ratings.csv")

full_df.to_csv("cleaned_movie_ratings.csv",
index=False)
    st.success("✅ Exported ...")
```

Explanation:

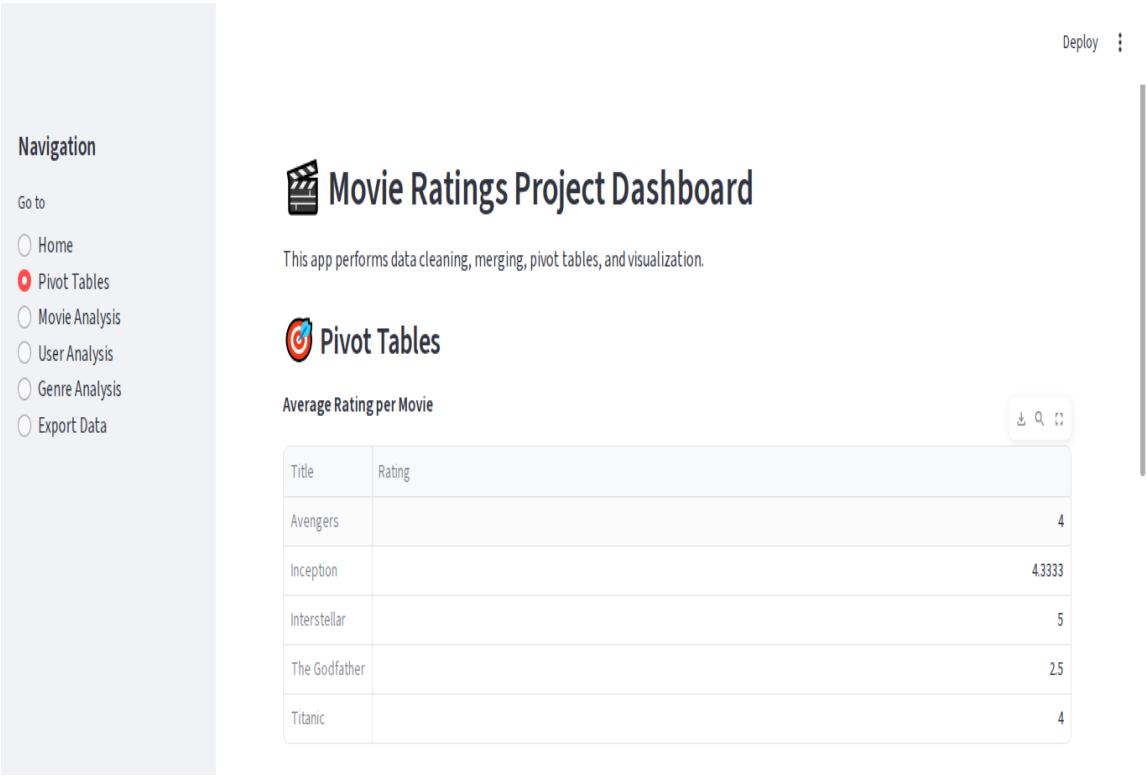
- Saves processed pivot tables and cleaned dataset into **CSV files**.
- Shows a success message after exporting.

---

8. Screenshots of Output



	UserID	MovieID	Rating	Timestamp	Title	Genre	Name	Age	Gender	RatingCategory	IsPopular
0	U001	M001	5	2023-01-01	Inception	Sci-Fi	Alice	25	F	High	No
1	U002	M001	4	2023-01-02	Inception	Sci-Fi	Bob	30	M	High	No
2	U003	M002	5	2023-01-03	Titanic	Romance	Charlie	22	M	High	No
3	U001	M003	3	2023-01-04	The Godfather	Crime	Alice	25	F	Medium	No
4	U002	M004	4	2023-01-05	Avengers	Action	Bob	30	M	High	No
5	U003	M005	5	2023-01-06	Interstellar	Sci-Fi	Charlie	22	M	High	No
6	U004	M001	4	2023-01-07	Inception	Sci-Fi	Diana	28	F	High	No
7	U004	M003	2	2023-01-08	The Godfather	Crime	Diana	28	F	Low	No
8	U005	M002	3	2023-01-09	Titanic	Romance	Ethan	35	M	Medium	No



Title	Rating
Avengers	4
Inception	4.3333
Interstellar	5
The Godfather	2.5
Titanic	4

Deploy

Navigation

Go to

- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data

Average Rating per Genre

Genre	Rating
Action	4
Crime	2.5
Romance	4
Sci-Fi	4.5

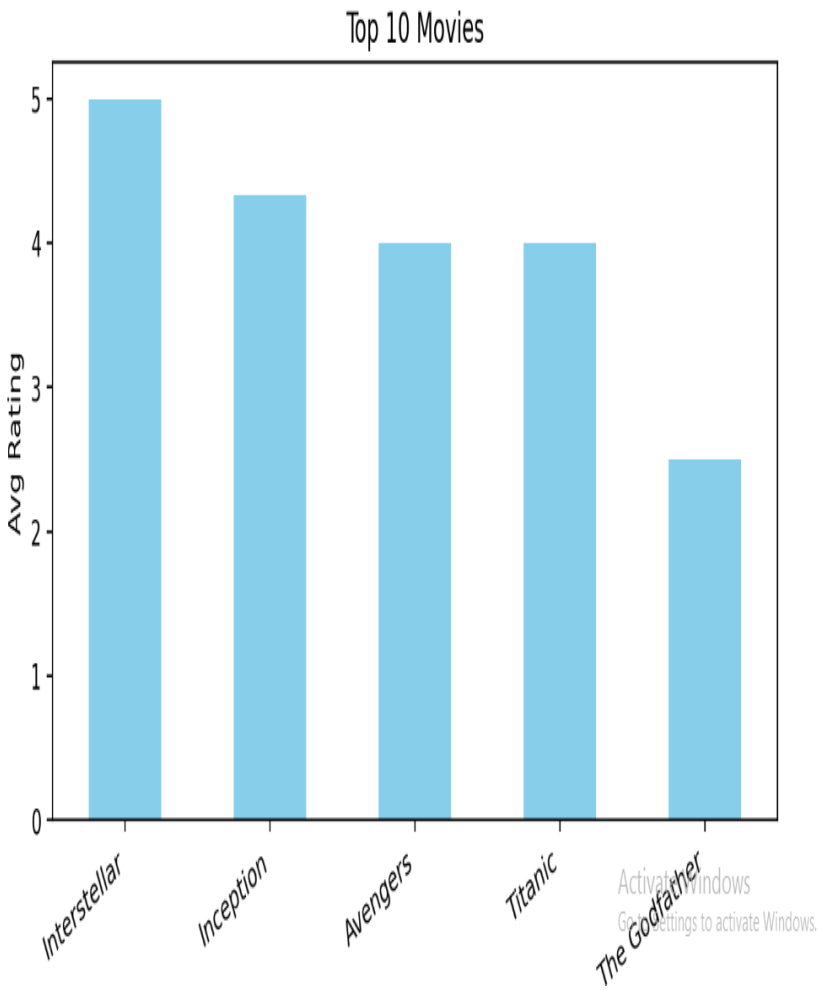
Average Rating per User

UserID	Rating
U001	4
U002	4
U003	5
U004	3
U005	3

Navigation

Go to

- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data



Navigation

Go to

- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data

Movies with Avg Rating ≥ 4.0

Title	Rating
Avengers	4
Inception	4.3333
Interstellar	5
Titanic	4

Top 5 Movies by Number of Ratings

Title	0
Inceptio	3
Titanic	2
The Go	2
Avenge	1
Interste	1

Navigation

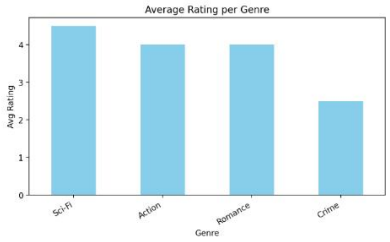
Go to

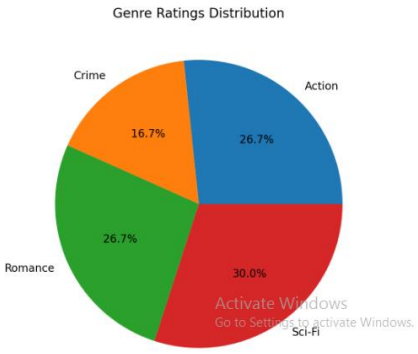
- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data

Movie Ratings Project Dashboard

This app performs data cleaning, merging, pivot tables, and visualization.

Genre Analysis

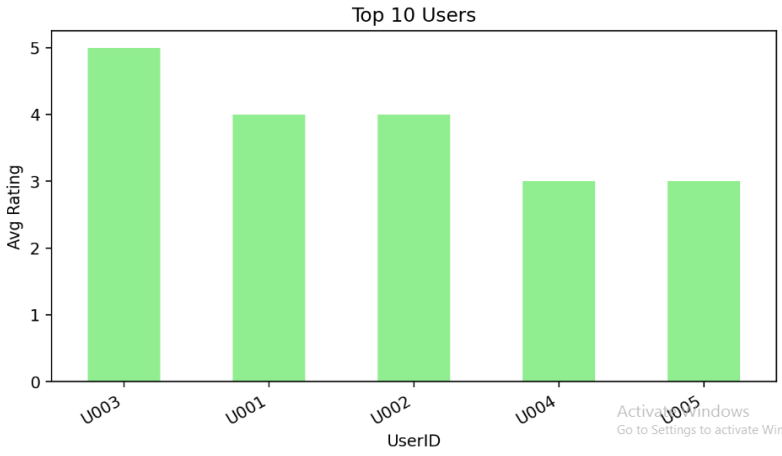




Navigation

Go to

- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data



Navigation

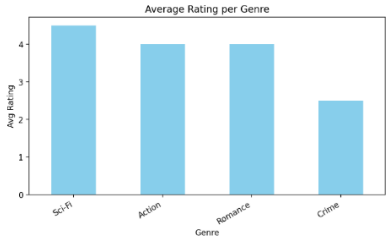
Go to

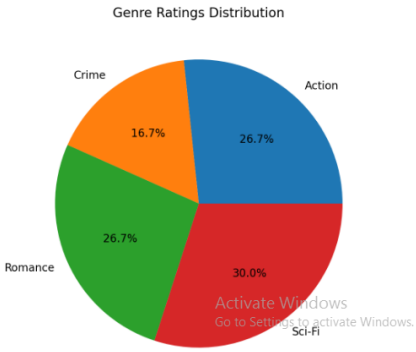
- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data

Movie Ratings Project Dashboard

This app performs data cleaning, merging, pivot tables, and visualization.

Genre Analysis

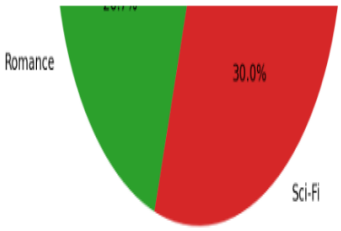




Navigation

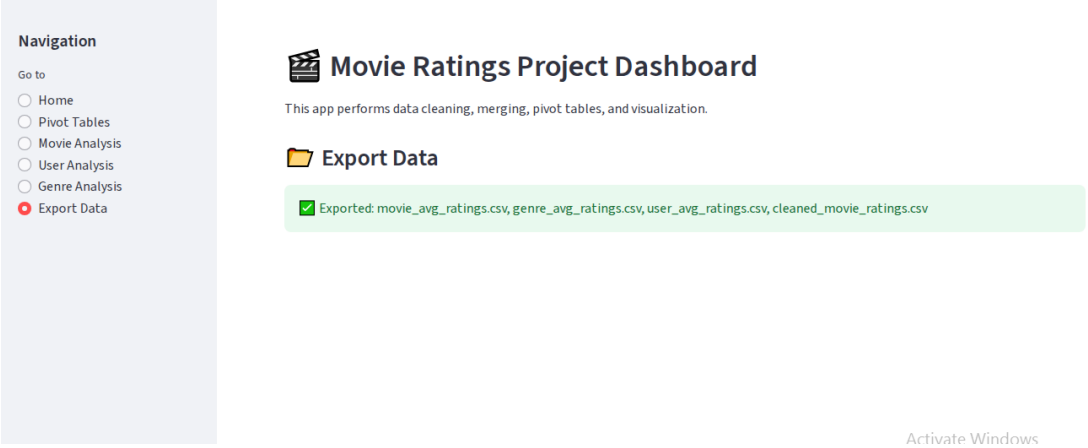
Go to

- Home
- Pivot Tables
- Movie Analysis
- User Analysis
- Genre Analysis
- Export Data



Genre Table

Genre	Rating
Action	4
Crime	2.5
Romance	4
Sci-Fi	4.5



10. Closure / Bibliography

This project, *Movie Ratings Analysis Dashboard*, successfully demonstrates the complete process of data exploration, transformation, and visualization using Python and Streamlit. Beginning with raw datasets — **movies.csv**, **ratings.csv**, and **users.csv** — the system carried out essential preprocessing steps such as handling missing values, removing duplicates, and merging data across multiple files. This ensured that the final dataset was clean, consistent, and reliable for further analysis.

The project made extensive use of **pivot tables** to derive insights from different perspectives. Average ratings were computed per movie, per user, and per genre. Additional filters and thresholds were applied to identify highly-rated movies (≥ 4.0), active users who rated more than five films, and the most frequently rated titles. These analyses enabled the extraction of both descriptive statistics and deeper insights into user behavior and content popularity.

Feature engineering further enriched the dataset by introducing new fields such as **RatingCategory** (Low, Medium, High) and **IsPopular** (Yes/No), which helped in classifying and segmenting the data. This made the analysis more intuitive and interpretable for decision-making purposes.

The **visualization layer** provided with Streamlit played a crucial role in making results accessible and interactive. Bar charts, pie charts, and tables were integrated into a multi-page dashboard, allowing smooth navigation between sections such as Home, Pivot Tables, Movie Analysis, User Analysis, and Genre Analysis. Each visualization complemented the numeric results, offering a clear picture of trends and distributions.

Additionally, the system incorporated a dedicated **Export Data** feature to allow saving processed datasets and results into CSV files. This makes the dashboard not only analytical but also practical for further offline research, reporting, or integration with other tools.

In terms of real-world applicability, this project mirrors how streaming platforms like Netflix, Amazon Prime, or IMDb perform content analysis and personalization. By analyzing ratings and user activity, platforms can highlight popular content, recommend movies by genre, and identify patterns in user preferences.

Overall, the project achieved all its objectives:

- Cleaned and merged multiple datasets into one consolidated source.
- Verified merge success using row counts.
- Built pivot tables and advanced aggregations for multi-angle insights.
- Designed a user-friendly interactive dashboard with Streamlit.
- Exported results for further use.

This end-to-end pipeline shows the effectiveness of combining **data science, visualization, and interactive web frameworks** for solving real analytical problems. The project not only satisfies academic requirements but also provides a scalable foundation for industry-level applications in recommendation systems and user behavior analytics.

This project demonstrates:

- End-to-End Data Analytics Workflow (cleaning → merging → pivoting → visualization).
- Practical Application of Pandas for dataset handling.
- Interactive Dashboards with Streamlit for data storytelling.

Bibliography:

- Streamlit Documentation: https://docs.streamlit.io
- Pandas Documentation: https://pandas.pydata.org
- Matplotlib Documentation: https://matplotlib.org
- Dataset references: MovieLens dataset, synthetic demo data for testing.