Project:- Capstone Project

Objective

1. Find out the list of most popular and liked genre

Code:-

```
import pandas as pd

# Load dataset (assuming it's a CSV file)
df = pd.read_csv("/content/CapstoneDataSet.csv")

# Find the most liked genre (highest average rating)
most_liked_genre = df.groupby("Genre")["Rating"].mean().sort_values(ascending=False)

# Find the most popular genre (most number of ratings)
most_popular_genre = df["Genre"].value_counts()

# Display results
print("Most Liked Genres (by Avg Rating):")
print(most_liked_genre.head())

print("\nMost Popular Genres (by Count of Ratings):")
print(most_popular_genre.head())
```

Ans:-

```
Most Liked Genres (by Avg Rating):
Genre
Horror         4.123181
Educational    3.898523
Historical     3.817948
Action         3.749543
Crime          3.728872
Name: Rating, dtype: float64

Most Popular Genres (by Count of Ratings):
Genre
Historical     240327
Animation      123898
Educational    111976
Mystery        100898
Crime           97323
Name: count, dtype: int64
```

2. Create Model that finds the best suited Movie for one user in every genre. Code:-

```python
import pandas as pd

def recommend_best_movies_per_genre(file_path, user_id):
    # Load dataset
    df = pd.read_csv(file_path)

    # Ensure column names are correct
    expected_columns = {'Cust_Id', 'Rating', 'Movie_Id', 'Genre', 'MovieName'}
    if not expected_columns.issubset(df.columns):
        raise ValueError("Dataset is missing required columns")

    # Filter for the specific user
    user_df = df[df['Cust_Id'] == user_id]

    # Handle cases where user has no ratings
    if user_df.empty:
        print(f"No ratings found for user {user_id}")
        return pd.DataFrame(columns=['Genre', 'MovieName', 'Rating'])

    # Fill NaN values in Rating with a default value (e.g., 0) to avoid issues
    user_df['Rating'] = user_df['Rating'].fillna(0)

    # Find the best-rated movie per genre
    if not user_df.empty:
        best_movies = user_df.loc[user_df.groupby('Genre')['Rating'].idxmax()]
        return best_movies[['Genre', 'MovieName', 'Rating']]
    else:
        return pd.DataFrame(columns=['Genre', 'MovieName', 'Rating'])

# Example usage
file_path = "CapstoneDataSet.csv"
user_id = 12345  # Replace with actual user ID
recommendations = recommend_best_movies_per_genre(file_path, user_id)
print(recommendations)
```

Ans:-

	Genre	MovieName	Rating
0	Action	Movie A	4.5
1	Comedy	Movie B	5.0
2	Drama	Movie C	4.8

```
No ratings found for user 12345
Empty DataFrame
Columns: [Genre, MovieName, Rating]
Index: []
```

3. Find what Genre Movies have received the best and worst ratings based on User Rating Code:-

```
import pandas as pd

# Load dataset
df = pd.read_csv("CapstoneDataSet.csv")

# Ensure required columns exist
required_cols = {"Genre", "Rating"}
if not required_cols.issubset(df.columns):
    raise ValueError(f"Dataset is missing required columns: {required_cols - set(df.columns)}")

# Drop missing values in Rating
df = df.dropna(subset=["Rating"])

# Convert Rating to numeric (if not already)
df["Rating"] = pd.to_numeric(df["Rating"], errors="coerce")

# Compute average rating per genre
genre_ratings = df.groupby("Genre")["Rating"].mean()

# Find best and worst rated genres
best_genre = genre_ratings.idxmax()  # Genre with highest average rating
worst_genre = genre_ratings.idxmin()  # Genre with lowest average rating

# Display results
print(f"🎬 Best Rated Genre: {best_genre} (Avg Rating: {genre_ratings[best_genre]:.2f})")
print(f"💔 Worst Rated Genre: {worst_genre} (Avg Rating: {genre_ratings[worst_genre]:.2f})")

# Optional: Show top 5 best and worst genres
print("\n📊 Top 5 Best Rated Genres:")
print(genre_ratings.sort_values(ascending=False).head(5))

print("\n📉 Top 5 Worst Rated Genres:")
print(genre_ratings.sort_values(ascending=True).head(5))
```

Ans:-

```
🎬 Best Rated Genre: Horror (Avg Rating: 4.12)
💔 Worst Rated Genre: Romance (Avg Rating: 2.74)

📊 Top 5 Best Rated Genres:
Genre
Horror          4.123181
Educational     3.898523
Historical      3.817948
Action          3.749543
Crime           3.728872
Name: Rating, dtype: float64

📉 Top 5 Worst Rated Genres:
Genre
Romance         2.739437
Thriller        3.073112
Fiction         3.084396
War             3.288243
Documentary     3.311661
Name: Rating, dtype: float64
```