Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

ML-IV LAB 9

Insights using MapReduce

Bhuvi Ghosh
60009210191

Aim:

To analyze a real-world movie dataset using MapReduce to extract insights such as the number of ratings for each movie, the list of users who rated movies, and the maximum, minimum, and average ratings given by users.

Theory:

MapReduce is a distributed data processing model that allows for parallel computation across large datasets in a distributed cluster. It simplifies the process of working with massive amounts of data by breaking down tasks into two primary functions: **Map** and **Reduce**.

1. **Map Function:**
   - The map function processes each record in the input dataset and produces intermediate key-value pairs. For instance, when analyzing a movie dataset, the map function can emit each movie ID as a key and the corresponding rating as the value. ○ The map phase is completely parallelizable. All data can be mapped independently on different nodes of a cluster.
2. **Shuffle and Sort:**
   - After the map function, the output is grouped by key. This step involves shuffling the intermediate data so that all key-value pairs with the same key are sent to the same reducer. ○ Sorting is also applied to ensure that data passed to the reduce function is ordered by keys.
3. **Reduce Function:**
   - The reduce function aggregates the intermediate key-value pairs produced by the map function. It processes all values associated with a particular key and computes a result, such as summing ratings for each movie or calculating statistics (max, min, average) for a user's ratings.

Objectives:

- List all movies and the number of ratings each movie has received.
- List all Movie IDs that have been rated by at least one user.
- List all Users who have rated at least one movie.
- Compute the maximum, minimum, and average ratings given by each user.

Algorithm:

Task 1: Listing Movies and the Number of Ratings

This task involves counting how many times each movie has been rated.

Step-by-Step Process:

1. **Map Phase:**
    - Input: Each record from the dataset (UserID, MovieID, Rating, Timestamp).
    - Output: (MovieID, 1) for every movie rating.

Example:
- Input: (1, 101, 4.0, 964982703)
- Output: (101, 1)

2. **Shuffle and Sort:**
    - Group the intermediate key-value pairs by MovieID.

Example:
- Intermediate Data: (101, 1), (101, 1), (102, 1), (102, 1), (103, 1)
- Grouped Data: {101: [1, 1], 102: [1, 1], 103: [1]}

3. **Reduce Phase:**
    - Input: (MovieID, List of counts)
    - Output: (MovieID, Sum of counts), i.e., the number of ratings for each movie.

Example:
- Input: (101, [1, 1])
- Output: (101, 2) (Movie 101 received 2 ratings)

Task 2: Listing Movie IDs with at least one rating

This task identifies all the movies that have been rated by at least one user.

Step-by-Step Process:

1. **Map Phase:**
    - Input: Each record from the dataset.
    - Output: (MovieID, 1) for every rated movie.

Example:
- Input: (1, 101, 4.0, 964982703)

- Output: (101, 1)

2. **Shuffle and Sort:**
   - Group all the values by MovieID.

Example:
- Intermediate Data: (101, 1), (102, 1), (103, 1)
- Grouped Data: {101: [1], 102: [1], 103: [1]}

3. **Reduce Phase:**
   - Input: (MovieID, List of 1s)
   - Output: (MovieID) if the list has at least one element, meaning that the movie has been rated.

Example:
- Input: (101, [1])
- Output: 101

Task 3: Listing Users Who Rated Movies

This task involves identifying all the users who have rated at least one movie.

Step-by-Step Process:

1. **Map Phase:**
   - Input: Each record from the dataset.
   - Output: (UserID, 1) for every movie rating made by a user.

Example:
- Input: (1, 101, 4.0, 964982703)
- Output: (1, 1)

2. **Shuffle and Sort:**
   - Group all values by UserID.

Example:
- Intermediate Data: (1, 1), (2, 1), (3, 1)
- Grouped Data: {1: [1], 2: [1], 3: [1]}

3. **Reduce Phase:**
   - Input: (UserID, List of 1s)
   - Output: (UserID) if the user has at least one rating.

Example:
- Input: (1, [1])
- Output: 1

Task 4: Calculating Max, Min, and Average Ratings per User

This task requires computing the maximum, minimum, and average ratings given by each user.

Step-by-Step Process:

1. **Map Phase:**
   - Input: Each record from the dataset.
   - Output: (UserID, Rating).

Example: ○ Input: (1, 101, 4.0, 964982703)
○ Output: (1, 4.0)

2. **Shuffle and Sort:**
   - Group all ratings by UserID.

Example: ○ Intermediate Data: (1, 4.0), (1, 5.0), (2, 3.0)
○ Grouped Data: {1: [4.0, 5.0], 2: [3.0]}

3. **Reduce Phase:**
   - Input: (UserID, List of Ratings) ○ Output: (UserID, Max, Min, Average of Ratings).

Example:
   - Input: (1, [4.0, 5.0])
   - Output: (1, Max: 5.0, Min: 4.0, Avg: 4.5)

Conclusion:

This experiment demonstrates the effectiveness of the **MapReduce** framework in processing large datasets by breaking tasks into parallelizable components. By analyzing a movie dataset, we extracted meaningful insights, such as the number of ratings per movie, users who rated movies, and user-specific rating statistics. These results are obtained in a scalable and efficient manner, suitable for distributed computing environments.

Code:

```
import pandas as pd from
functools import reduce

# Load Netflix dataset from the downloaded folder
netflix_df = pd.read_csv('netflix_titles.csv')

# Print the attributes (i.e., columns) of the dataset
print("Attributes of the Netflix dataset:")
print(netflix_df.columns)

# 1. Movie Duration Distribution
# We'll clean up the 'duration' column to work only with movies
(excluding TV Shows)
movie_durations = netflix_df[netflix_df['type'] == 'Movie']
movie_durations['duration'] =
movie_durations['duration'].str.replace(' min', '').astype(float)

# Group movies into buckets based on duration
duration_bins = pd.cut(movie_durations['duration'], bins=[0, 90, 120,
float('inf')], labels=['<90 mins', '90-120 mins', '>120 mins'])
duration_distribution = duration_bins.value_counts()

# 2. Top 10 Countries by Content Count
# We'll clean up the 'country' column and split it by commas to count
each country separately
netflix_df['country'] = netflix_df['country'].fillna('Unknown')
country_list = netflix_df['country'].str.split(',',
expand=True).stack().str.strip().reset_index(drop=True)
top_10_countries = country_list.value_counts().head(10)

# 3. Trend of Content by Release Year
release_year_trend =
netflix_df.groupby('release_year').size().reset_index(name="count").so
rt_values(by='release_year', ascending=False)

# 4. Top 10 Directors by Number of Titles
top_10_directors =
netflix_df['director'].dropna().value_counts().head(10)

# 5. Most Frequent Genres
# Split 'listed_in' column by commas to count genres
genre_list = netflix_df['listed_in'].str.split(',',
expand=True).stack().str.strip().reset_index(drop=True)
top_genres = genre_list.value_counts().head(10)

# 6. Movies vs. TV Shows by Rating
rating_distribution = netflix_df.groupby(['type',
'rating']).size().unstack(fill_value=0)

# Display Results
```

```
print("\nMovie Duration Distribution:")
print(duration_distribution)

print("\nTop 10 Countries by Content Count:")
print(top_10_countries)

print("\nContent Trend by Release Year:")
print(release_year_trend.head(10))  # Display the last 10 years of
content trends

print("\nTop 10 Directors by Number of Titles:")
print(top_10_directors)

print("\nMost Frequent Genres:")
print(top_genres)

print("\nRating Distribution (Movies vs. TV Shows):")
print(rating_distribution)
```

```
Attributes of the Netflix dataset:
Index(['show_id', 'type', 'title', 'director', 'cast', 'country',
'date_added',
       'release_year', 'rating', 'duration', 'listed_in',
'description'],
dtype='object')
Movie           Duration
Distribution:
duration

90-120       mins
2996
<90          mins
1990
>120         mins
1142
Name:  count,  dtype:
int64
Top 10 Countries by Content
Count:
United       States
3690
India          1046
Unknown         831
United      Kingdom
806
Canada          445
France          393
Japan           318
Spain           232
```

```
South          Korea
231
Germany          226









Name:  count,  dtype:
int64
Content  Trend  by  Release
Year:
    release_year
count 73         2021
592
72               2020
953
```

```
71          2019
1030
70          2018
1147
69          2017
1032
68          2016
902
67          2015
560
66          2014
352
65          2013
288
64          2012
237
Top  10  Directors  by
Number of Titles:
director
Rajiv       Chilaka
19
Raúl  Campos,  Jan
Suter    18
Marcus        Raboy
16
Suhas         Kadav
16
Jay           Karas
14
Cathy Garcia-Molina
13
Martin     Scorsese
12
Youssef     Chahine
12
Jay         Chapman
12
Steven    Spielberg
11
Name:      count,
dtype: int64
Most   Frequent
Genres:
```

```
International  Movies
2752
Dramas
2427
Comedies
1674
International      TV
Shows      1351
Documentaries
869
Action   &   Adventure
859
TV             Dramas
763
Independent    Movies
756
Children   &   Family
Movies     641
Romantic       Movies
616
Name:      count,
dtype: int64
Rating Distribution (Movies vs. TV Shows): rating   66 min  74 min  84
min   G  NC-17  NR   PG  PG-13    R  TV-14 TV-G  \
type

Movie         1       1       1  41      3  75  287    490  797   1427
126
TV Show       0       0       0   0      0   5    0      0    2    733
94
rating   TV-MA  TV-PG  TV-Y  TV-Y7  TV-Y7-FV  UR


Movie     2062    540   131    139         5   3
TV Show   1145    323   176    195         1   0
```

```
<ipython-input-5-30f86cb12dc1>:14: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation:
https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#
returning-a-view-versus-a-copy
movie_durations['duration'] =
movie_durations['duration'].str.replace(' min', '').astype(float)

import pandas as pd
from collections import defaultdict

# Load Netflix dataset
netflix_df = pd.read_csv('netflix_titles.csv')

# Function to perform the Map operation
def map_function(data, key_column):
    mapped = []     for _, row in
data.iterrows():
        key = row[key_column]
        mapped.append((key, 1))  # Create key-value pairs (key, 1)
return mapped

# Function to perform the Shuffle operation def
shuffle_function(mapped_data):     shuffled =
defaultdict(list)     for key, value in mapped_data:
shuffled[key].append(value)  # Group values by key
return shuffled

# Function to perform the Reduce operation
def reduce_function(shuffled_data):
    reduced = {}     for key, values in shuffled_data.items():
reduced[key] = sum(values)  # Aggregate by summing values for each
key     return reduced

# Task 1: MapReduce to count Movies vs. TV Shows
mapped_type = map_function(netflix_df, "type")
shuffled_type = shuffle_function(mapped_type)
reduced_type = reduce_function(shuffled_type)
print("\nMovies vs TV Shows Count:")
print(reduced_type)

# Task 2: MapReduce to count content produced by each country
# We need to handle cases where there are multiple countries in the
'country' column. def
map_function_country(data):
```

```
    mapped = []     for _, row in
data.iterrows():         if
pd.isna(row['country']):
continue  # Skip NaN values
        countries = row['country'].split(',')  # Split by commas for
multiple countries         for country in countries:
mapped.append((country.strip(), 1))  # Create key-value pairs
(country, 1)     return mapped

mapped_country = map_function_country(netflix_df)
shuffled_country = shuffle_function(mapped_country)
reduced_country = reduce_function(shuffled_country)
print("\nCountry Content Count:")
print(reduced_country)

# Task 3: MapReduce to count content by rating
mapped_rating = map_function(netflix_df, "rating")
shuffled_rating = shuffle_function(mapped_rating)
reduced_rating = reduce_function(shuffled_rating)
print("\nContent Rating Count:")
print(reduced_rating)


Movies vs TV Shows Count:
{'Movie': 6131, 'TV Show': 2676}

Country Content Count:
{'United States': 3690, 'South Africa': 62, 'India': 1046, 'Ghana': 5,
'Burkina Faso': 1, 'United Kingdom': 806, 'Germany': 226, 'Ethiopia':
1, 'Czech Republic': 22, 'Mexico': 169, 'Turkey': 113, 'Australia':
160, 'France': 393, 'Finland': 11, 'China': 162, 'Canada': 445,
'Japan': 318, 'Nigeria': 103, 'Spain': 232, 'Belgium': 90, 'South
Korea': 231, 'Singapore': 41, 'Italy': 100, 'Romania': 14,
'Argentina': 91, 'Venezuela': 4, 'Hong Kong': 105, 'Russia': 27, '':
7, 'Ireland': 46, 'Nepal': 2, 'New Zealand': 33, 'Brazil': 97,
'Greece': 11, 'Jordan': 9, 'Colombia': 52, 'Switzerland': 19,
'Israel': 30, 'Taiwan': 89, 'Bulgaria': 10, 'Algeria': 3, 'Poland':
41, 'Saudi Arabia': 13, 'Thailand': 70, 'Indonesia': 90, 'Egypt': 117,
'Denmark': 48, 'Kuwait': 8, 'Netherlands': 50, 'Malaysia': 26,
'Vietnam': 7, 'Hungary': 11, 'Sweden': 42, 'Lebanon': 31, 'Syria': 3,
'Philippines': 83, 'Iceland': 11, 'United Arab Emirates': 37,
'Norway': 30, 'Qatar': 10, 'Mauritius': 2, 'Austria': 12, 'Cameroon':
1, 'Palestine': 1, 'Uruguay': 14, 'Kenya': 6, 'Chile': 29,
'Luxembourg': 12, 'Cambodia': 6, 'Bangladesh': 4, 'Portugal': 6,
'Cayman Islands': 2, 'Senegal': 3, 'Serbia': 7, 'Malta': 3, 'Namibia':
2, 'Angola': 1, 'Peru': 10, 'Mozambique': 1, 'Belarus': 1, 'Zimbabwe':
3, 'Puerto Rico': 1, 'Pakistan': 24, 'Cyprus': 1, 'Guatemala': 2,
'Iraq': 2, 'Malawi': 1, 'Paraguay': 1, 'Croatia': 4, 'Iran': 4, 'West
Germany': 5, 'Albania': 1, 'Georgia': 2, 'Soviet Union': 3, 'Morocco':
6, 'Slovakia': 1, 'Ukraine': 3, 'Bermuda': 1, 'Ecuador': 1, 'Armenia':
```

```
1, 'Mongolia': 1, 'Bahamas': 1, 'Sri Lanka': 1, 'Latvia': 1,
'Liechtenstein': 1, 'Cuba': 1, 'Nicaragua': 1, 'Slovenia': 3,
'Dominican Republic': 1, 'Samoa': 1, 'Azerbaijan': 1, 'Botswana': 1,
'Vatican City': 1, 'Jamaica': 1, 'Kazakhstan': 1, 'Lithuania': 1,
'Afghanistan': 1, 'Somalia': 1, 'Sudan': 1, 'Panama': 1, 'Uganda': 1,
'East Germany': 1, 'Montenegro': 1}

Content Rating Count:
{'PG-13': 490, 'TV-MA': 3207, 'PG': 287, 'TV-14': 2160, 'TV-PG': 863,
'TV-Y': 307, 'TV-Y7': 334, 'R': 799, 'TV-G': 220, 'G': 41, 'NC-17': 3,
'74 min': 1, '84 min': 1, '66 min': 1, 'NR': 80, nan: 4, 'TV-Y7-FV':
6, 'UR': 3}

# Task 4: MapReduce to count content by release year
mapped_year = map_function(netflix_df, "release_year")
shuffled_year = shuffle_function(mapped_year)
reduced_year = reduce_function(shuffled_year)

print("\nContent Count by Release Year:")
print(reduced_year)


Content Count by Release Year:
{2020: 953, 2021: 592, 1993: 28, 2018: 1147, 1996: 24, 1998: 36, 1997:
38, 2010: 194, 2013: 288, 2017: 1032, 1975: 7, 1978: 7, 1983: 11,
1987: 8, 2012: 237, 2001: 45, 2014: 352, 2002: 51, 2003: 61, 2004: 64,
2011: 185, 2008: 136, 2009: 152, 2007: 88, 2005: 80, 2006: 96, 1994:
22, 2015: 560, 2019: 1030, 2016: 902, 1982: 17, 1989: 16, 1990: 22,
1991: 17, 1999: 39, 1986: 13, 1992: 23, 1984: 12, 1980: 11, 1961: 1,
2000: 37, 1995: 25, 1985: 10, 1976: 9, 1959: 1, 1988: 18, 1981: 13,
1972: 5, 1964: 2, 1945: 4, 1954: 2, 1979: 11, 1958: 3, 1956: 2, 1963:
2, 1970: 2, 1973: 10, 1925: 1, 1974: 7, 1960: 4, 1966: 1, 1971: 5,
1962: 3, 1969: 2, 1977: 7, 1967: 5, 1968: 3, 1965: 2, 1946: 2, 1942:
2, 1955: 3, 1944: 3, 1947: 1, 1943: 3}

# Task 5: MapReduce to count most frequent directors
def map_function_director(data):
    mapped = []     for _, row in
data.iterrows():         if
pd.isna(row['director']):
continue  # Skip NaN values
        directors = row['director'].split(',')  # Split by commas for
multiple directors         for director in directors:
mapped.append((director.strip(), 1))  # Create key-value pairs
(director, 1)     return mapped
```

```
mapped_director = map_function_director(netflix_df)
shuffled_director = shuffle_function(mapped_director)
reduced_director = reduce_function(shuffled_director)

print("\nDirector Content Count:") # To display top 10 directors
top_directors = dict(sorted(reduced_director.items(), key=lambda item:
item[1], reverse=True)[:10]) print(top_directors)
```

```
Director Content Count:
{'Rajiv Chilaka': 22, 'Jan Suter': 21, 'Raúl Campos': 19, 'Suhas
Kadav': 16, 'Marcus Raboy': 16, 'Jay Karas': 15, 'Cathy Garcia-
Molina': 13, 'Youssef Chahine': 12, 'Martin Scorsese': 12, 'Jay
Chapman': 12}
```

```
# Task 6: MapReduce to count most frequent genres def
map_function_genres(data):     mapped = []     for _, row in
data.iterrows():         genres = row['listed_in'].split(',')  #
Split by commas for multiple genres         for genre in genres:
mapped.append((genre.strip(), 1))  # Create key-value pairs (genre,
1)     return mapped

mapped_genres = map_function_genres(netflix_df)
shuffled_genres = shuffle_function(mapped_genres)
reduced_genres = reduce_function(shuffled_genres)

print("\nGenre Count:")
# Display top 10 most frequent genres top_genres =
dict(sorted(reduced_genres.items(), key=lambda item: item[1],
reverse=True)[:10]) print(top_genres)
```

```
Genre Count:
{'International Movies': 2752, 'Dramas': 2427, 'Comedies': 1674,
'International TV Shows': 1351, 'Documentaries': 869, 'Action &
Adventure': 859, 'TV Dramas': 763, 'Independent Movies': 756,
'Children & Family Movies': 641, 'Romantic Movies': 616}
```

```
# Task 7: MapReduce to count Movies vs. TV Shows by release year
def map_function_type_year(data):
    mapped = []     for _, row in data.iterrows():         key =
(row['type'], row['release_year'])  # Tuple (type, release_year)
```

```
        mapped.append((key, 1))  # Create key-value pairs ((type,
year), 1)     return mapped

mapped_type_year = map_function_type_year(netflix_df)
shuffled_type_year = shuffle_function(mapped_type_year)
reduced_type_year = reduce_function(shuffled_type_year)

print("\nMovies vs TV Shows by Release Year:")
print(reduced_type_year)


Movies vs TV Shows by Release Year:
{('Movie', 2020): 517, ('TV Show', 2021): 315, ('Movie', 2021): 277,
('Movie', 1993): 24, ('TV Show', 2020): 436, ('TV Show', 2018): 380,
('Movie', 1996): 21, ('Movie', 1998): 32, ('Movie', 1997): 34,
('Movie', 2010): 154, ('Movie', 2013): 225, ('Movie', 2017): 767,
('Movie', 1975): 7, ('Movie', 1978): 7, ('Movie', 1983): 11, ('Movie',
1987): 8, ('Movie', 2012): 173, ('Movie', 2001): 40, ('TV Show',
2014): 88, ('Movie', 2002): 44, ('Movie', 2003): 51, ('Movie', 2004):
55, ('Movie', 2011): 145, ('Movie', 2008): 113, ('Movie', 2009): 118,
('Movie', 2007): 74, ('Movie', 2005): 67, ('Movie', 2006): 82, ('TV
Show', 1994): 2, ('TV Show', 2015): 162, ('Movie', 2018): 767, ('TV
Show', 2013): 63, ('TV Show', 2019): 397, ('TV Show', 2017): 265,
('Movie', 2019): 633, ('TV Show', 2016): 244, ('Movie', 1994): 20,
('Movie', 2015): 398, ('TV Show', 2012): 64, ('Movie', 1982): 17,
('Movie', 1989): 15, ('Movie', 2014): 264, ('Movie', 1990): 19,
('Movie', 1991): 16, ('Movie', 1999): 32, ('Movie', 2016): 658,
('Movie', 1986): 11, ('TV Show', 1992): 3, ('Movie', 1984): 12,
('Movie', 1980): 11, ('Movie', 1961): 1, ('Movie', 2000): 33, ('TV
Show', 2002): 7, ('Movie', 1995): 23, ('Movie', 1985): 9, ('TV Show',
2009): 34, ('TV Show', 2011): 40, ('TV Show', 2005): 13, ('TV Show',
2008): 23, ('Movie', 1992): 20, ('TV Show', 2010): 40, ('TV Show',
2007): 14, ('Movie', 1976): 9, ('Movie', 1959): 1, ('Movie', 1988):
16, ('TV Show', 2001): 5, ('Movie', 1981): 12, ('Movie', 1972): 4,
('TV Show', 2006): 14, ('TV Show', 1993): 4, ('TV Show', 1997): 4,
('TV Show', 2003): 10, ('Movie', 1964): 2, ('TV Show', 1945): 1, ('TV
Show', 1999): 7, ('Movie', 1954): 2, ('Movie', 1979): 10, ('TV Show',
1998): 4, ('TV Show', 2000): 4, ('TV Show', 2004): 9, ('Movie', 1958):
3, ('Movie', 1956): 2, ('Movie', 1963): 1, ('Movie', 1970): 2,
('Movie', 1973): 10, ('TV Show', 1986): 2, ('TV Show', 1995): 2, ('TV
Show', 1925): 1, ('TV Show', 1972): 1, ('TV Show', 1974): 1, ('Movie',
1960): 4, ('TV Show', 1988): 2, ('Movie', 1974): 6, ('Movie', 1966):
1, ('Movie', 1971): 5, ('Movie', 1962): 3, ('Movie', 1969): 2,
('Movie', 1977): 6, ('TV Show', 1991): 1, ('Movie', 1967): 4,
('Movie', 1968): 3, ('TV Show', 1977): 1, ('Movie', 1965): 2, ('TV
Show', 1979): 1, ('TV Show', 1990): 3, ('TV Show', 1996): 3, ('Movie',
1945): 3, ('Movie', 1946): 1, ('TV Show', 1981): 1, ('TV Show', 1946):
1, ('Movie', 1942): 2, ('Movie', 1955): 3, ('TV Show', 1985): 1, ('TV
```

```
Show', 1967): 1, ('Movie', 1944): 3, ('TV Show', 1989): 1, ('TV Show',
1963): 1, ('Movie', 1947): 1, ('Movie', 1943): 3}

# Task 8: MapReduce to calculate the average duration of Movies def
map_function_movie_duration(data):     mapped = []     for _, row in
data.iterrows():         if row['type'] == 'Movie' and
isinstance(row['duration'], str):             duration =
row['duration'].replace(' min', '')  # Extract duration in minutes
            mapped.append((row['type'], float(duration)))  # Create
key-value pairs (Movie, duration)     return mapped

def reduce_function_average(shuffled_data):
    reduced = {}     for key, values in shuffled_data.items():
reduced[key] = sum(values) / len(values)  # Calculate average for
each key     return reduced

mapped_movie_duration = map_function_movie_duration(netflix_df)
shuffled_movie_duration = shuffle_function(mapped_movie_duration)
average_movie_duration =
reduce_function_average(shuffled_movie_duration)

print("\nAverage Duration of Movies:")
print(average_movie_duration)


Average Duration of Movies:
{'Movie': 99.57718668407311}

# Task 9: Finding the country with the maximum content
# We will reuse the map_function_country from Task 2

mapped_country = map_function_country(netflix_df)
shuffled_country = shuffle_function(mapped_country)
reduced_country = reduce_function(shuffled_country)

# Find the country with the maximum count
max_country = max(reduced_country.items(), key=lambda item: item[1])

print("\nCountry with Maximum Content:")
print(max_country)
Country with Maximum Content:
('United States', 3690)
```

```
# Task 10: Finding the longest and shortest Movies def
map_function_movie_duration_minmax(data):     mapped = []     for
_, row in data.iterrows():         if row['type'] == 'Movie' and
isinstance(row['duration'], str):             try:
duration = row['duration'].replace(' min', '')  # Extract
duration in minutes
                mapped.append((row['title'], float(duration)))  #
Create key-value pairs (title, duration)                  except
ValueError:
                # Skip if conversion to float fails
continue     return mapped

mapped_movie_duration_minmax =
map_function_movie_duration_minmax(netflix_df)

# Find longest and shortest movies longest_movie =
max(mapped_movie_duration_minmax, key=lambda item:
item[1]) shortest_movie = min(mapped_movie_duration_minmax,
key=lambda item: item[1])

print("\nLongest Movie:")
print(longest_movie)

print("\nShortest Movie:")
print(shortest_movie)
Longest Movie:

('Black Mirror: Bandersnatch', 312.0)
Shortest Movie:
('Silent', 3.0)
# Task: MapReduce to categorize genres by rating
def map_function_genre_rating(data):
    mapped = []     for _, row in data.iterrows():         if
pd.notna(row['listed_in']) and pd.notna(row['rating']):  # Check for
valid genre and rating
            genres = row['listed_in'].split(',')  # Split genres by
comma             for genre in genres:
mapped.append(((genre.strip(), row['rating']), 1))  #
Create key-value pairs ((genre, rating), 1)
return mapped
```

```
def shuffle_function(mapped_data):     shuffled =
defaultdict(list)     for key, value in mapped_data:
shuffled[key].append(value)  # Group values by key (genre, rating)
    return shuffled

def reduce_function(shuffled_data):     reduced = {}     for key,
values in shuffled_data.items():         reduced[key] =
sum(values)  # Sum values for each (genre, rating) combination
return reduced

# Perform MapReduce for genres by rating
mapped_genre_rating = map_function_genre_rating(netflix_df)
shuffled_genre_rating = shuffle_function(mapped_genre_rating)
reduced_genre_rating = reduce_function(shuffled_genre_rating)

# Sort results by count in descending order
sorted_genre_rating = dict(sorted(reduced_genre_rating.items(),
key=lambda item: item[1], reverse=True))

# Display Results: Number of content by genre and rating, sorted by
count (highest to lowest)
print("\nNumber of content by Genre and Rating (sorted by count):")
for (genre, rating), count in sorted_genre_rating.items():
    print(f"Genre: {genre}, Rating: {rating}, Count: {count}")
```

```
Number of content by Genre and Rating (sorted by count):
Genre: International Movies, Rating: TV-MA, Count: 1130
Genre: International Movies, Rating: TV-14, Count: 1065
Genre: Dramas, Rating: TV-MA, Count: 830
Genre: International TV Shows, Rating: TV-MA, Count: 714
Genre: Dramas, Rating: TV-14, Count: 693
Genre: International TV Shows, Rating: TV-14, Count: 472
Genre: Comedies, Rating: TV-14, Count: 465
Genre: TV Dramas, Rating: TV-MA, Count: 434
Genre: Comedies, Rating: TV-MA, Count: 431
Genre: Dramas, Rating: R, Count: 375
Genre: Crime TV Shows, Rating: TV-MA, Count: 350
Genre: Independent Movies, Rating: TV-MA, Count: 344
Genre: Documentaries, Rating: TV-MA, Count: 321
Genre: International Movies, Rating: TV-PG, Count: 294
Genre: Stand-Up Comedy, Rating: TV-MA, Count: 291
Genre: TV Comedies, Rating: TV-MA, Count: 269
Genre: TV Dramas, Rating: TV-14, Count: 269
Genre: Thrillers, Rating: TV-MA, Count: 240
```

```
Genre: Documentaries, Rating: TV-14, Count: 227
Genre: Action & Adventure, Rating: R, Count: 220
Genre: Action & Adventure, Rating: TV-14, Count: 213
Genre: Romantic Movies, Rating: TV-14, Count: 208
Genre: Action & Adventure, Rating: TV-MA, Count: 201
Genre: Dramas, Rating: TV-PG, Count: 200
Genre: Children & Family Movies, Rating: PG, Count: 195
Genre: Independent Movies, Rating: R, Count: 193
Genre: Dramas, Rating: PG-13, Count: 192
Genre: Romantic TV Shows, Rating: TV-14, Count: 190
Genre: Kids' TV, Rating: TV-Y7, Count: 189
Genre: Comedies, Rating: R, Count: 180
Genre: Docuseries, Rating: TV-MA, Count: 179
Genre: Kids' TV, Rating: TV-Y, Count: 176
Genre: Comedies, Rating: PG-13, Count: 168
Genre: Documentaries, Rating: TV-PG, Count: 167
Genre: Horror Movies, Rating: TV-MA, Count: 159
Genre: Comedies, Rating: TV-PG, Count: 153
Genre: Thrillers, Rating: R, Count: 153
Genre: Romantic Movies, Rating: TV-MA, Count: 151
Genre: Action & Adventure, Rating: PG-13, Count: 148
Genre: Comedies, Rating: PG, Count: 148
Genre: TV Comedies, Rating: TV-14, Count: 140
Genre: International TV Shows, Rating: TV-PG, Count: 134
Genre: Romantic TV Shows, Rating: TV-MA, Count: 132
Genre: Spanish-Language TV Shows, Rating: TV-MA, Count: 130
Genre: Children & Family Movies, Rating: TV-Y7, Count: 129
Genre: Thrillers, Rating: TV-14, Count: 119
Genre: Independent Movies, Rating: TV-14, Count: 119
Genre: Music & Musicals, Rating: TV-MA, Count: 117
Genre: Children & Family Movies, Rating: TV-Y, Count: 113
Genre: Music & Musicals, Rating: TV-14, Count: 112
Genre: Crime TV Shows, Rating: TV-14, Count: 111
Genre: British TV Shows, Rating: TV-MA, Count: 108
Genre: International Movies, Rating: R, Count: 101
Genre: Docuseries, Rating: TV-PG, Count: 99
Genre: Horror Movies, Rating: R, Count: 97
Genre: Docuseries, Rating: TV-14, Count: 92
Genre: TV Action & Adventure, Rating: TV-MA, Count: 90
Genre: Romantic Movies, Rating: PG-13, Count: 87
Genre: Sci-Fi & Fantasy, Rating: PG-13, Count: 86
Genre: Children & Family Movies, Rating: TV-PG, Count: 85
Genre: Reality TV, Rating: TV-MA, Count: 83
Genre: Reality TV, Rating: TV-14, Count: 78
Genre: Sports Movies, Rating: TV-MA, Count: 73
Genre: Romantic Movies, Rating: TV-PG, Count: 72
```

```
Genre: Reality TV, Rating: TV-PG, Count: 71
Genre: Anime Series, Rating: TV-14, Count: 71
Genre: Dramas, Rating: PG, Count: 69
Genre: TV Mysteries, Rating: TV-MA, Count: 65
Genre: Korean TV Shows, Rating: TV-14, Count: 64
Genre: TV Comedies, Rating: TV-PG, Count: 60
Genre: Anime Series, Rating: TV-MA, Count: 59
Genre: LGBTQ Movies, Rating: TV-MA, Count: 59
Genre: TV Horror, Rating: TV-MA, Count: 57
Genre: British TV Shows, Rating: TV-PG, Count: 55
Genre: Thrillers, Rating: PG-13, Count: 54
Genre: TV Comedies, Rating: TV-Y7, Count: 54
Genre: International Movies, Rating: PG-13, Count: 54
Genre: Romantic Movies, Rating: R, Count: 54
Genre: International Movies, Rating: TV-G, Count: 54
Genre: Music & Musicals, Rating: TV-PG, Count: 52
Genre: Children & Family Movies, Rating: TV-G, Count: 51
Genre: Sci-Fi & Fantasy, Rating: TV-MA, Count: 50
Genre: TV Dramas, Rating: TV-PG, Count: 49
Genre: Korean TV Shows, Rating: TV-MA, Count: 48
Genre: Comedies, Rating: TV-Y7, Count: 47
Genre: Documentaries, Rating: TV-G, Count: 47
Genre: British TV Shows, Rating: TV-14, Count: 46
Genre: Sci-Fi & Fantasy, Rating: R, Count: 46
Genre: TV Action & Adventure, Rating: TV-14, Count: 44
Genre: Horror Movies, Rating: TV-14, Count: 44
Genre: Science & Nature TV, Rating: TV-PG, Count: 44
Genre: Romantic TV Shows, Rating: TV-PG, Count: 44
Genre: Kids' TV, Rating: TV-G, Count: 43
Genre: Horror Movies, Rating: PG-13, Count: 42
Genre: Kids' TV, Rating: TV-PG, Count: 41
Genre: TV Sci-Fi & Fantasy, Rating: TV-14, Count: 41
Genre: TV Thrillers, Rating: TV-MA, Count: 39
Genre: Sports Movies, Rating: TV-14, Count: 39
Genre: Stand-Up Comedy & Talk Shows, Rating: TV-MA, Count: 38
Genre: Cult Movies, Rating: R, Count: 37
Genre: Independent Movies, Rating: TV-PG, Count: 35
Genre: Independent Movies, Rating: PG-13, Count: 35
Genre: International Movies, Rating: NR, Count: 34
Genre: Documentaries, Rating: PG-13, Count: 33
Genre: Action & Adventure, Rating: TV-PG, Count: 33
Genre: Children & Family Movies, Rating: G, Count: 33
Genre: Action & Adventure, Rating: PG, Count: 32
Genre: Classic Movies, Rating: R, Count: 32
Genre: Teen TV Shows, Rating: TV-14, Count: 32
Genre: TV Mysteries, Rating: TV-14, Count: 31
```

```
Genre: Sci-Fi & Fantasy, Rating: TV-14, Count: 29
Genre: Comedies, Rating: TV-G, Count: 29
Genre: Stand-Up Comedy, Rating: TV-14, Count: 29
Genre: TV Comedies, Rating: TV-G, Count: 28
Genre: TV Comedies, Rating: TV-Y, Count: 28
Genre: Documentaries, Rating: R, Count: 27
Genre: Dramas, Rating: TV-G, Count: 27
Genre: Dramas, Rating: NR, Count: 27
Genre: Anime Series, Rating: TV-Y7, Count: 26
Genre: Sports Movies, Rating: PG-13, Count: 26
Genre: Sports Movies, Rating: TV-PG, Count: 26
Genre: Spanish-Language TV Shows, Rating: TV-14, Count: 26
Genre: Docuseries, Rating: TV-G, Count: 24
Genre: International TV Shows, Rating: TV-G, Count: 24
Genre: Music & Musicals, Rating: R, Count: 24
Genre: Anime Features, Rating: TV-14, Count: 23
Genre: Romantic Movies, Rating: PG, Count: 23
Genre: Classic Movies, Rating: TV-14, Count: 23
Genre: Reality TV, Rating: TV-G, Count: 22
Genre: Sports Movies, Rating: PG, Count: 22
Genre: Documentaries, Rating: NR, Count: 22
Genre: Science & Nature TV, Rating: TV-14, Count: 21
Genre: Sci-Fi & Fantasy, Rating: PG, Count: 21
Genre: TV Sci-Fi & Fantasy, Rating: TV-MA, Count: 21
Genre: Comedies, Rating: TV-Y, Count: 21
Genre: Documentaries, Rating: PG, Count: 20
Genre: British TV Shows, Rating: TV-Y, Count: 20
Genre: Teen TV Shows, Rating: TV-MA, Count: 20
Genre: Faith & Spirituality, Rating: TV-PG, Count: 20
Genre: Anime Features, Rating: TV-PG, Count: 19
Genre: Music & Musicals, Rating: TV-G, Count: 19
Genre: Children & Family Movies, Rating: TV-14, Count: 18
Genre: Faith & Spirituality, Rating: TV-14, Count: 18
Genre: Classic Movies, Rating: PG, Count: 17
Genre: Music & Musicals, Rating: PG-13, Count: 17
Genre: TV Action & Adventure, Rating: TV-PG, Count: 17
Genre: Anime Series, Rating: TV-PG, Count: 17
Genre: Romantic Movies, Rating: TV-G, Count: 17
Genre: British TV Shows, Rating: TV-G, Count: 17
Genre: Sports Movies, Rating: R, Count: 17
Genre: TV Horror, Rating: TV-14, Count: 17
Genre: Movies, Rating: TV-Y, Count: 17
Genre: Independent Movies, Rating: NR, Count: 17
Genre: Anime Features, Rating: TV-MA, Count: 15
Genre: Science & Nature TV, Rating: TV-MA, Count: 15
Genre: Teen TV Shows, Rating: TV-PG, Count: 15
```

Genre: Korean TV Shows, Rating: TV-Y7, Count: 14
Genre: TV Action & Adventure, Rating: TV-Y7, Count: 14
Genre: Korean TV Shows, Rating: TV-PG, Count: 14
Genre: Comedies, Rating: NR, Count: 14
Genre: Cult Movies, Rating: PG-13, Count: 13
Genre: Music & Musicals, Rating: PG, Count: 13
Genre: Movies, Rating: TV-MA, Count: 13
Genre: LGBTQ Movies, Rating: R, Count: 13
Genre: International Movies, Rating: PG, Count: 13
Genre: Faith & Spirituality, Rating: PG, Count: 13
Genre: Science & Nature TV, Rating: TV-G, Count: 12
Genre: Spanish-Language TV Shows, Rating: TV-PG, Count: 12
Genre: LGBTQ Movies, Rating: TV-PG, Count: 11
Genre: Classic Movies, Rating: TV-MA, Count: 11
Genre: TV Thrillers, Rating: TV-14, Count: 11
Genre: Children & Family Movies, Rating: PG-13, Count: 11
Genre: Comedies, Rating: G, Count: 11
Genre: Classic Movies, Rating: TV-PG, Count: 11
Genre: Classic Movies, Rating: PG-13, Count: 10
Genre: Stand-Up Comedy & Talk Shows, Rating: TV-14, Count: 10
Genre: Classic & Cult TV, Rating: TV-14, Count: 10
Genre: TV Sci-Fi & Fantasy, Rating: TV-Y7, Count: 9
Genre: TV Sci-Fi & Fantasy, Rating: TV-PG, Count: 9
Genre: Korean TV Shows, Rating: TV-Y, Count: 9
Genre: TV Dramas, Rating: TV-G, Count: 9
Genre: Faith & Spirituality, Rating: PG-13, Count: 9
Genre: Music & Musicals, Rating: TV-Y7, Count: 9
Genre: Classic & Cult TV, Rating: TV-MA, Count: 9
Genre: Movies, Rating: TV-Y7, Count: 9
Genre: Action & Adventure, Rating: NR, Count: 9
Genre: TV Shows, Rating: TV-14, Count: 8
Genre: LGBTQ Movies, Rating: PG-13, Count: 8
Genre: LGBTQ Movies, Rating: TV-14, Count: 8
Genre: Classic Movies, Rating: G, Count: 8
Genre: Stand-Up Comedy, Rating: R, Count: 7
Genre: Cult Movies, Rating: TV-14, Count: 7
Genre: Sports Movies, Rating: TV-G, Count: 7
Genre: Stand-Up Comedy, Rating: TV-PG, Count: 7
Genre: Movies, Rating: TV-PG, Count: 7
Genre: Horror Movies, Rating: NR, Count: 7
Genre: Dramas, Rating: G, Count: 6
Genre: Music & Musicals, Rating: G, Count: 6
Genre: Independent Movies, Rating: PG, Count: 6
Genre: Classic & Cult TV, Rating: TV-PG, Count: 6
Genre: Music & Musicals, Rating: TV-Y, Count: 5
Genre: TV Shows, Rating: TV-MA, Count: 5

Genre: Cult Movies, Rating: TV-MA, Count: 5
Genre: Stand-Up Comedy & Talk Shows, Rating: TV-PG, Count: 5
Genre: Thrillers, Rating: TV-PG, Count: 5
Genre: Anime Features, Rating: PG, Count: 5
Genre: Sci-Fi & Fantasy, Rating: TV-PG, Count: 5
Genre: British TV Shows, Rating: TV-Y7, Count: 5
Genre: Documentaries, Rating: G, Count: 5
Genre: Stand-Up Comedy, Rating: NR, Count: 5
Genre: Children & Family Movies, Rating: TV-Y7-FV, Count: 5
Genre: Horror Movies, Rating: PG, Count: 4
Genre: Cult Movies, Rating: PG, Count: 4
Genre: Crime TV Shows, Rating: TV-PG, Count: 4
Genre: Anime Features, Rating: TV-Y7, Count: 4
Genre: Horror Movies, Rating: TV-PG, Count: 4
Genre: Anime Features, Rating: PG-13, Count: 4
Genre: Thrillers, Rating: NR, Count: 4
Genre: Comedies, Rating: TV-Y7-FV, Count: 4
Genre: Sports Movies, Rating: NR, Count: 4
Genre: Movies, Rating: TV-14, Count: 3
Genre: Stand-Up Comedy, Rating: TV-G, Count: 3
Genre: Movies, Rating: TV-G, Count: 3
Genre: Independent Movies, Rating: TV-G, Count: 3
Genre: Dramas, Rating: TV-Y, Count: 3
Genre: Faith & Spirituality, Rating: TV-MA, Count: 3
Genre: Sports Movies, Rating: TV-Y, Count: 3
Genre: International TV Shows, Rating: TV-Y7, Count: 3
Genre: Spanish-Language TV Shows, Rating: TV-Y, Count: 3
Genre: Crime TV Shows, Rating: TV-Y7, Count: 3
Genre: Classic Movies, Rating: NR, Count: 3
Genre: Cult Movies, Rating: NR, Count: 3
Genre: Thrillers, Rating: PG, Count: 2
Genre: Teen TV Shows, Rating: TV-G, Count: 2
Genre: Sports Movies, Rating: TV-Y7, Count: 2
Genre: Romantic TV Shows, Rating: TV-G, Count: 2
Genre: TV Thrillers, Rating: TV-Y7, Count: 2
Genre: Anime Series, Rating: TV-Y, Count: 2
Genre: Korean TV Shows, Rating: TV-G, Count: 2
Genre: Sci-Fi & Fantasy, Rating: G, Count: 2
Genre: TV Thrillers, Rating: TV-PG, Count: 2
Genre: Sci-Fi & Fantasy, Rating: TV-Y7, Count: 2
Genre: Sci-Fi & Fantasy, Rating: TV-Y, Count: 2
Genre: Stand-Up Comedy & Talk Shows, Rating: TV-G, Count: 2
Genre: Comedies, Rating: NC-17, Count: 2
Genre: International Movies, Rating: NC-17, Count: 2
Genre: Cult Movies, Rating: TV-PG, Count: 2
Genre: LGBTQ Movies, Rating: NR, Count: 2

Genre: Independent Movies, Rating: NC-17, Count: 2
Genre: TV Shows, Rating: TV-G, Count: 2
Genre: International TV Shows, Rating: NR, Count: 2
Genre: Romantic TV Shows, Rating: NR, Count: 2
Genre: Romantic Movies, Rating: NR, Count: 2
Genre: Dramas, Rating: UR, Count: 2
Genre: International Movies, Rating: UR, Count: 2
Genre: Romantic Movies, Rating: UR, Count: 2
Genre: Independent Movies, Rating: TV-Y7, Count: 2
Genre: British TV Shows, Rating: NR, Count: 2
Genre: Anime Features, Rating: TV-G, Count: 1
Genre: TV Mysteries, Rating: TV-G, Count: 1
Genre: Faith & Spirituality, Rating: R, Count: 1
Genre: Children & Family Movies, Rating: TV-MA, Count: 1
Genre: Stand-Up Comedy, Rating: PG-13, Count: 1

```
Genre: TV Sci-Fi & Fantasy, Rating: TV-G, Count: 1
Genre: Classic Movies, Rating: TV-G, Count: 1
Genre: Classic & Cult TV, Rating: TV-Y, Count: 1
Genre: Spanish-Language TV Shows, Rating: TV-G, Count: 1
Genre: Classic & Cult TV, Rating: TV-Y7, Count: 1
Genre: Movies, Rating: R, Count: 1
Genre: Spanish-Language TV Shows, Rating: TV-Y7, Count: 1
Genre: LGBTQ Movies, Rating: TV-Y7, Count: 1
Genre: TV Sci-Fi & Fantasy, Rating: TV-Y, Count: 1
Genre: TV Action & Adventure, Rating: TV-G, Count: 1
Genre: TV Shows, Rating: R, Count: 1
Genre: TV Thrillers, Rating: TV-Y, Count: 1
Genre: TV Mysteries, Rating: TV-PG, Count: 1
Genre: TV Horror, Rating: TV-PG, Count: 1
Genre: Movies, Rating: 74 min, Count: 1
Genre: Movies, Rating: 84 min, Count: 1
Genre: Movies, Rating: 66 min, Count: 1
Genre: TV Thrillers, Rating: TV-G, Count: 1
Genre: Dramas, Rating: NC-17, Count: 1
Genre: Faith & Spirituality, Rating: TV-G, Count: 1
Genre: Spanish-Language TV Shows, Rating: NR, Count: 1
Genre: Crime TV Shows, Rating: TV-G, Count: 1
Genre: Action & Adventure, Rating: G, Count: 1
Genre: International Movies, Rating: TV-Y, Count: 1
Genre: Reality TV, Rating: TV-Y7, Count: 1
Genre: Dramas, Rating: TV-Y7, Count: 1
Genre: International Movies, Rating: TV-Y7, Count: 1
Genre: Action & Adventure, Rating: TV-Y7-FV, Count: 1
Genre: International TV Shows, Rating: R, Count: 1
Genre: TV Dramas, Rating: R, Count: 1
Genre: TV Thrillers, Rating: R, Count: 1
Genre: TV Dramas, Rating: NR, Count: 1
Genre: International Movies, Rating: G, Count: 1
Genre: Kids' TV, Rating: TV-Y7-FV, Count: 1
Genre: TV Action & Adventure, Rating: TV-Y7-FV, Count: 1
Genre: TV Sci-Fi & Fantasy, Rating: TV-Y7-FV, Count: 1
Genre: TV Action & Adventure, Rating: NR, Count: 1
Genre: TV Comedies, Rating: NR, Count: 1
Genre: TV Sci-Fi & Fantasy, Rating: NR, Count: 1
Genre: Docuseries, Rating: NR, Count: 1
Genre: Stand-Up Comedy & Talk Shows, Rating: NR, Count: 1
Genre: Classic & Cult TV, Rating: TV-G, Count: 1
Genre: Crime TV Shows, Rating: NR, Count: 1
Genre: Music & Musicals, Rating: NR, Count: 1
Genre: Action & Adventure, Rating: UR, Count: 1
Genre: Comedies, Rating: UR, Count: 1
```