In [1]:
```
import numpy as np
import pandas as pd

import seaborn as sns
import matplotlib.pyplot as plt

import requests
from bs4 import BeautifulSoup
import time

from selenium import webdriver
from selenium.webdriver.chrome.options import Options
```

Top 200 Movies based on IMDB and No. of Reviews

In [2]:
```
options = Options()
options.add_argument("--headless")
options.add_argument("user-agent=Mozilla/5.0")

driver = webdriver.Chrome(options=options)
driver.get('https://www.imdb.com/chart/top/')
time.sleep(5)

soup = BeautifulSoup(driver.page_source, 'html.parser')
driver.quit()

movies = []
movie_blocks = soup.find_all("li", class_="ipc-metadata-list-summary-item")

for item in movie_blocks:
    try:
        # title
        title_tag = item.find("h3")
        title = title_tag.text.strip() if title_tag else "N/A"

        # num of reviews
        vote_tag = item.select_one("span.ipc-rating-star--voteCount")
        votes_raw = vote_tag.text.strip() if vote_tag else "0"
        votes_raw = votes_raw.replace('\xa0', '').replace('(', '').replace(')', '')

        multiplier = 1
        if 'K' in votes_raw:
            multiplier = 1_000
            votes_raw = votes_raw.replace('K', '')
```

```python
        elif 'M' in votes_raw:
            multiplier = 1_000_000
            votes_raw = votes_raw.replace('M', '')

        votes = int(float(votes_raw) * multiplier)

        movies.append((title, votes))

    except Exception as e:
        print(f"Error parsing row: {e}")


df = pd.DataFrame(movies, columns=["Title", "NumReviews"])

top_df = df

print(top_df.head())
print(f"Total movies saved: {len(top_df)}")
```

```
                        Title  NumReviews
0  1. The Shawshank Redemption     3000000
1              2. The Godfather     2100000
2            3. The Dark Knight     3000000
3      4. The Godfather Part II     1400000
4               5. 12 Angry Men      920000
Total movies saved: 250
```

In [3]:
```python
print(top_df.head(200).to_string())
```

	Title	NumReviews
0	1. The Shawshank Redemption	3000000
1	2. The Godfather	2100000
2	3. The Dark Knight	3000000
3	4. The Godfather Part II	1400000
4	5. 12 Angry Men	920000
5	6. The Lord of the Rings: The Return of the King	2100000
6	7. Schindler's List	1500000
7	8. Pulp Fiction	2300000
8	9. The Lord of the Rings: The Fellowship of the Ring	2100000
9	10. The Good, the Bad and the Ugly	849000
10	11. Forrest Gump	2400000
11	12. The Lord of the Rings: The Two Towers	1900000
12	13. Fight Club	2500000
13	14. Inception	2700000
14	15. Star Wars: Episode V - The Empire Strikes Back	1400000
15	16. The Matrix	2100000
16	17. Goodfellas	1300000
17	18. One Flew Over the Cuckoo's Nest	1100000
18	19. Interstellar	2300000
19	20. Se7en	1900000
20	21. It's a Wonderful Life	526000
21	22. The Silence of the Lambs	1600000
22	23. Seven Samurai	382000
23	24. Saving Private Ryan	1600000
24	25. City of God	834000
25	26. The Green Mile	1500000
26	27. Life Is Beautiful	775000
27	28. Terminator 2: Judgment Day	1200000
28	29. Star Wars: Episode IV - A New Hope	1500000
29	30. Back to the Future	1400000
30	31. Spirited Away	904000
31	32. The Pianist	964000
32	33. Gladiator	1700000
33	34. Parasite	1100000
34	35. Psycho	749000
35	36. The Lion King	1200000
36	37. Grave of the Fireflies	345000
37	38. The Departed	1500000
38	39. Whiplash	1100000
39	40. Harakiri	79000
40	41. American History X	1200000
41	42. The Prestige	1500000
42	43. Léon: The Professional	1300000
43	44. Spider-Man: Across the Spider-Verse	447000
44	45. Casablanca	629000
45	46. The Usual Suspects	1200000
46	47. The Intouchables	973000

47	48. Cinema Paradiso	300000
48	49. Modern Times	272000
49	50. Alien	1000000
50	51. Rear Window	544000
51	52. Once Upon a Time in the West	366000
52	53. Django Unchained	1800000
53	54. City Lights	206000
54	55. Apocalypse Now	739000
55	56. Dune: Part Two	620000
56	57. Memento	1400000
57	58. WALL·E	1300000
58	59. Raiders of the Lost Ark	1100000
59	60. The Lives of Others	426000
60	61. Avengers: Infinity War	1300000
61	62. Sunset Boulevard	248000
62	63. Spider-Man: Into the Spider-Verse	725000
63	64. Paths of Glory	225000
64	65. Witness for the Prosecution	149000
65	66. The Shining	1200000
66	67. The Great Dictator	248000
67	68. 12th Fail	145000
68	69. Aliens	806000
69	70. Inglourious Basterds	1700000
70	71. The Dark Knight Rises	1900000
71	72. The Chaos Class Failed the Class	25000
72	73. Coco	645000
73	74. Amadeus	447000
74	75. Toy Story	1100000
75	76. Avengers: Endgame	1300000
76	77. Good Will Hunting	1100000
77	78. Oldboy	675000
78	79. Dr. Strangelove or: How I Learned to Stop Worrying and Love the Bomb	536000
79	80. American Beauty	1200000
80	81. Das Boot	278000
81	82. Braveheart	1100000
82	83. Princess Mononoke	457000
83	84. Your Name.	353000
84	85. High and Low	59000
85	86. 3 Idiots	456000
86	87. Joker	1600000
87	88. Once Upon a Time in America	393000
88	89. Capernaum	115000
89	90. Singin' in the Rain	271000
90	91. Come and See	111000
91	92. Requiem for a Dream	935000
92	93. Toy Story 3	928000
93	94. Star Wars: Episode VI - Return of the Jedi	1200000
94	95. The Hunt	386000

95	96. Ikiru	96000
96	97. Eternal Sunshine of the Spotless Mind	1100000
97	98. 2001: A Space Odyssey	750000
98	99. Reservoir Dogs	1100000
99	100. The Apartment	208000
100	101. Incendies	225000
101	102. Lawrence of Arabia	328000
102	103. Scarface	960000
103	104. Double Indemnity	174000
104	105. North by Northwest	359000
105	106. Heat	752000
106	107. Citizen Kane	480000
107	108. Up	1200000
108	109. M	176000
109	110. Full Metal Jacket	819000
110	111. Vertigo	443000
111	112. Amélie	816000
112	113. A Separation	268000
113	114. A Clockwork Orange	909000
114	115. To Kill a Mockingbird	342000
115	116. Die Hard	987000
116	117. Like Stars on Earth	217000
117	118. The Sting	289000
118	119. Indiana Jones and the Last Crusade	836000
119	120. Oppenheimer	876000
120	121. Metropolis	193000
121	122. Snatch	943000
122	123. 1917	723000
123	124. L.A. Confidential	639000
124	125. Bicycle Thieves	184000
125	126. Downfall	387000
126	127. Dangal	224000
127	128. Taxi Driver	973000
128	129. Hamilton	127000
129	130. The Wolf of Wall Street	1700000
130	131. Batman Begins	1600000
131	132. Green Book	630000
132	133. For a Few Dollars More	287000
133	134. Judgment at Nuremberg	91000
134	135. The Truman Show	1300000
135	136. Some Like It Hot	294000
136	137. Shutter Island	1500000
137	138. The Kid	141000
138	139. The Father	211000
139	140. All About Eve	144000
140	141. There Will Be Blood	668000
141	142. Jurassic Park	1100000
142	143. Casino	589000

143	144. The Sixth Sense	1100000
144	145. Ran	143000
145	146. Top Gun: Maverick	767000
146	147. No Country for Old Men	1100000
147	148. The Thing	495000
148	149. Pan's Labyrinth	723000
149	150. Unforgiven	453000
150	151. A Beautiful Mind	1000000
151	152. Kill Bill: Vol. 1	1200000
152	153. The Treasure of the Sierra Madre	137000
153	154. Prisoners	866000
154	155. Yojimbo	137000
155	156. Finding Nemo	1200000
156	157. The Great Escape	267000
157	158. Howl's Moving Castle	480000
158	159. Monty Python and the Holy Grail	586000
159	160. The Elephant Man	272000
160	161. Dial M for Murder	197000
161	162. Gone with the Wind	345000
162	163. Rashomon	189000
163	164. Klaus	213000
164	165. Chinatown	364000
165	166. The Secret in Their Eyes	232000
166	167. Lock, Stock and Two Smoking Barrels	633000
167	168. V for Vendetta	1200000
168	169. Inside Out	862000
169	170. Three Billboards Outside Ebbing, Missouri	580000
170	171. The Wild Robot	158000
171	172. The Bridge on the River Kwai	241000
172	173. Trainspotting	747000
173	174. Catch Me If You Can	1200000
174	175. Raging Bull	394000
175	176. Fargo	753000
176	177. Warrior	513000
177	178. Harry Potter and the Deathly Hallows: Part 2	992000
178	179. Gran Torino	837000
179	180. Million Dollar Baby	744000
180	181. My Neighbor Totoro	405000
181	182. Mad Max: Fury Road	1200000
182	183. Spider-Man: No Way Home	944000
183	184. Ben-Hur	262000
184	185. Children of Heaven	85000
185	186. Barry Lyndon	192000
186	187. Dead Poets Society	587000
187	188. 12 Years a Slave	762000
188	189. Before Sunrise	357000
189	190. The Grand Budapest Hotel	923000
190	191. Blade Runner	851000

```
191                                         192. Hacksaw Ridge      633000
192                                              193. Gone Girl    1100000
193                                      194. Memories of Murder    239000
194                                          195. I'm Still Here    101000
195                              196. In the Name of the Father     195000
196                                           197. Ratatouille      880000
197                                          198. Monsters, Inc.   1000000
198                                          199. The Gold Rush     123000
199                                            200. Wild Tales      227000
```

Top Foreign Movies

In [4]:

```
driver = webdriver.Chrome(options=options)
driver.get('https://www.imdb.com/list/ls052393071/')
time.sleep(5)

soup = BeautifulSoup(driver.page_source, 'html.parser')
driver.quit()

movies = []
movie_blocks = soup.find_all("li", class_="ipc-metadata-list-summary-item")

for item in movie_blocks:
    try:
        title_tag = item.find("h3")
        title = title_tag.text.strip() if title_tag else "N/A"

        vote_tag = item.select_one("span.ipc-rating-star--voteCount")
        votes_raw = vote_tag.text.strip() if vote_tag else "0"
        votes_raw = votes_raw.replace('\xa0', '').replace('(', '').replace(')', '')

        multiplier = 1
        if 'K' in votes_raw:
            multiplier = 1_000
            votes_raw = votes_raw.replace('K', '')
        elif 'M' in votes_raw:
            multiplier = 1_000_000
            votes_raw = votes_raw.replace('M', '')

        votes = int(float(votes_raw) * multiplier)

        movies.append((title, votes))

    except Exception as e:
        print(f"Error parsing row: {e}")
```

```python
df = pd.DataFrame(movies, columns=["Title", "NumReviews"])

top_foreign_df = df

print(top_foreign_df.head())
print(f"Total movies saved: {len(top_foreign_df)}")
```

```
                      Title  NumReviews
0  1. The Lives of Others      426000
1        2. Noi the Albino        9700
2              3. Das Boot      278000
3      4. Pan's Labyrinth      723000
4                5. Oldboy      675000
Total movies saved: 250
```

In [5]:
```python
print(top_foreign_df.head(200).to_string())
```

	Title	NumReviews
0	1. The Lives of Others	426000
1	2. Noi the Albino	9700
2	3. Das Boot	278000
3	4. Pan's Labyrinth	723000
4	5. Oldboy	675000
5	6. Open Your Eyes	75000
6	7. Max Manus: Man of War	31000
7	8. Respiro	5000
8	9. Run Lola Run	213000
9	10. Diva	15000
10	11. Spring, Summer, Fall, Winter... and Spring	89000
11	12. The Beat That My Heart Skipped	21000
12	13. The Wave	118000
13	14. The Counterfeiters	48000
14	15. Purple Noon	22000
15	16. A Prophet	105000
16	17. The Twilight Samurai	26000
17	18. Wild Strawberries	120000
18	19. Yojimbo	137000
19	20. Three Colors: Red	114000
20	21. City of God	834000
21	22. Ip Man	238000
22	23. The Method	13000
23	24. Live Flesh	35000
24	25. Mondays in the Sun	16000
25	26. Amélie	816000
26	27. The Valet	11000
27	28. Le Dîner de Cons	46000
28	29. Incendies	225000
29	30. Rust and Bone	72000
30	31. I Hired a Contract Killer	8300
31	32. Le Samouraï	62000
32	33. Biutiful	96000
33	34. Ali: Fear Eats the Soul	25000
34	35. Le Havre	25000
35	36. My Father's Bike	955
36	37. The Motorcycle Diaries	107000
37	38. Life Is Beautiful	775000
38	39. Behind the Sun	7600
39	40. Lower City	4600
40	41. Good Bye Lenin!	156000
41	42. Balzac and the Little Chinese Seamstress	4600
42	43. Cinema Paradiso	300000
43	44. La Dolce Vita	82000
44	45. The Wages of Fear	70000
45	46. Nine Queens	59000
46	47. Micmacs	32000

47	48. Tell No One	59000
48	49. Swimming Pool	50000
49	50. Jules and Jim	46000
50	51. Ran	143000
51	52. Vratné lahve	5100
52	53. Kolya	17000
53	54. In the House	35000
54	55. The Promise	8700
55	56. The Double Life of Véronique	55000
56	57. Three Colors: Blue	115000
57	58. Paris, Je T'aime	75000
58	59. The Hidden Face	48000
59	60. Philanthropy	14000
60	61. Bullhead	25000
61	62. Headhunters	109000
62	63. Neighboring Sounds	9000
63	64. You, the Living	17000
64	65. Klown	12000
65	66. Kagemusha: The Shadow Warrior	40000
66	67. The Skin I Live In	173000
67	68. Kon-Tiki	53000
68	69. Sons of Norway	1800
69	70. The Blind Swordsman: Zatoichi	52000
70	71. All Together	3100
71	72. Too Beautiful for You	3200
72	73. The Piano Teacher	76000
73	74. Everything Is Illuminated	61000
74	75. The Seventh Seal	207000
75	76. Volver	110000
76	77. The Grocer's Son	2600
77	78. La Femme Nikita	79000
78	79. Breaking the Waves	74000
79	80. Monsieur Lazhar	22000
80	81. Fanny and Alexander	69000
81	82. Intacto	14000
82	83. La haine	210000
83	84. The 400 Blows	131000
84	85. Breathless	91000
85	86. Nobody Knows	33000
86	87. Love Crime	5700
87	88. The Hunt	386000
88	89. Women on the Verge of a Nervous Breakdown	49000
89	90. Raise the Red Lantern	37000
90	91. The Apartment	16000
91	92. Starbuck	17000
92	93. Central Station	48000
93	94. Au Revoir les Enfants	37000
94	95. A Separation	268000

95	96. Point Blank	15000
96	97. Babette's Feast	23000
97	98. Barbara	17000
98	99. La Vie En Rose	92000
99	100. The Giants	1900
100	101. The Return	49000
101	102. The Taste of Others	11000
102	103. 36th Precinct	19000
103	104. Pelle the Conqueror	12000
104	105. Fearless	80000
105	106. Maria Full of Grace	37000
106	107. Ben X	20000
107	108. The Class	37000
108	109. Camille Rewinds	3000
109	110. The Man Without a Past	28000
110	111. In a Better World	42000
111	112. The Tin Drum	27000
112	113. 8½	129000
113	114. Reality	5900
114	115. Broken Embraces	43000
115	116. Sex and Lucía	40000
116	117. Fermat's Room	23000
117	118. The Bothersome Man	19000
118	119. Talk to Her	120000
119	120. This Man Must Die	5100
120	121. Eat Drink Man Woman	24000
121	122. The Names of Love	8400
122	123. Mon meilleur ami	7000
123	124. Sister	7400
124	125. Like Father, Like Son	30000
125	126. El Topo	32000
126	127. The Past	52000
127	128. Blue Is the Warmest Colour	168000
128	129. The Eighth Day	10000
129	130. When the Cat Comes	2200
130	131. Patagonia	661
131	132. A Royal Affair	53000
132	133. The Celebration	97000
133	134. A Place in the World	2900
134	135. After the Wedding	38000
135	136. Read My Lips	17000
136	137. The Wild Child	9200
137	138. Murmur of the Heart	11000
138	139. Bread and Tulips	12000
139	140. Together	25000
140	141. The Heineken Kidnapping	6100
141	142. The Kid with a Bike	29000
142	143. The Rocket	3200

143	144. Reprise	16000
144	145. The Barber of Siberia	13000
145	146. A Hijacking	18000
146	147. The Secret in Their Eyes	232000
147	148. Le Doulos	13000
148	149. Footnote	6400
149	150. Violette	2600
150	151. Autumn Sonata	40000
151	152. He Loves Me, He Loves Me Not	20000
152	153. Lovers of the Arctic Circle	21000
153	154. Frida	98000
154	155. Antonia's Line	9700
155	156. Simon	8300
156	157. Black Book	82000
157	158. The Vanishing	50000
158	159. Character	12000
159	160. All About My Mother	106000
160	161. The Snow Walker	12000
161	162. Songs from the Second Floor	21000
162	163. With a Friend Like Harry...	12000
163	164. The Best of Youth	25000
164	165. Mother of George	1300
165	166. The Other Son	3800
166	167. What If...	8800
167	168. Two Rabbits	5600
168	169. The Return of Martin Guerre	5000
169	170. Rabat	2300
170	171. A Man and a Woman	13000
171	172. The Hidden Blade	7400
172	173. My American Uncle	7100
173	174. My Life as a Dog	23000
174	175. The Sea Inside	87000
175	176. The Spanish Apartment	43000
176	177. 3 Idiots	456000
177	178. Kahaani	68000
178	179. Farewell	7100
179	180. About Elly	59000
180	181. Harakiri	79000
181	182. The Deep	5900
182	183. The White Ribbon	80000
183	184. Leak	3000
184	185. Wadjda	22000
185	186. Elling	16000
186	187. The Death of Mr. Lazarescu	16000
187	188. Mesrine: Public Enemy No. 1	32000
188	189. Not One Less	9600
189	190. Dreams	31000
190	191. Welcome	7700

```
191                    192. Love Me If You Dare      73000
192                193. The Exterminating Angel      37000
193                         194. Small Change         6700
194                             195. Revanche        17000
195                196. My Sweet Little Village       5000
196                           197. Teddy Bear         6900
197                     198. The Professional         7600
198                    199. The White Balloon         8500
199                 200. My Father and My Son        97000
```

Top 10 Directors

In [6]:
```
driver = webdriver.Chrome(options=options)
driver.get('https://www.imdb.com/list/ls026411399/')
time.sleep(5)

soup = BeautifulSoup(driver.page_source, 'html.parser')
driver.quit()

directors =[]

blocks = soup.find_all("li", class_="ipc-metadata-list-summary-item")

for item in blocks[:10]:
    try:
        name_tag = item.find("h3")
        name = name_tag.text.strip() if name_tag else "N/A"
        directors.append(name)
    except Exception as e:
        print("Error parsing director:", e)

top_dir_df = pd.DataFrame(directors, columns=["Director"])
print(top_dir_df)
```

```
                 Director
0      1. Christopher Nolan
1       2. Steven Spielberg
2      3. Quentin Tarantino
3        4. Martin Scorsese
4           5. Ridley Scott
5          6. David Fincher
6        7. Robert Zemeckis
7        8. Stanley Kubrick
8         9. Clint Eastwood
9  10. Francis Ford Coppola
```

In [7]:
```
print("Top 10 Directors")
for name in directors:
    print(f"{name}")
```

```
Top 10 Directors
1. Christopher Nolan
2. Steven Spielberg
3. Quentin Tarantino
4. Martin Scorsese
5. Ridley Scott
6. David Fincher
7. Robert Zemeckis
8. Stanley Kubrick
9. Clint Eastwood
10. Francis Ford Coppola
```

Recommending Movies Based on Actors

How my Recommending System Works

- The model recommends movies based on shared lead actors, using the top 3 actors listed for each film from IMDb.
- When a user selects a movie, the system finds others that share the most actors and ranks them based on how many appear in both films.
- The similarity between movies is measured using **cosine similarity**, which compares the overlap of actors represented in a **binary matrix**.

In [8]:
```
from sklearn.metrics.pairwise import cosine_similarity
import re
import json
```

In [9]:
```
driver = webdriver.Chrome(options=options)
driver.get('https://www.imdb.com/chart/top/')
time.sleep(5)

soup = BeautifulSoup(driver.page_source, 'html.parser')
movie_blocks = soup.find_all("li", class_="ipc-metadata-list-summary-item")

movie_links = []
for item in movie_blocks:
    try:
        title_tag = item.find("h3")
```

```
        title = title_tag.text.strip() if title_tag else None
        link_tag = item.find("a", href=True)
        link = "https://www.imdb.com" + link_tag["href"].split("?")[0] if link_tag else None
        if title and link:
            movie_links.append((title, link))
    except:
        pass

driver.quit()

movie_data = []

for i, (title,link) in enumerate(movie_links[:20]):  # does first 20 movies due to time it takes to run
    try:
        driver = webdriver.Chrome(options=options)
        driver.get(link)
        time.sleep(3)

        soup = BeautifulSoup(driver.page_source, 'html.parser')
        driver.quit()

        script_tag = soup.find('script', type='application/ld+json')
        actors = []

        if script_tag:
            try:
                data = json.loads(script_tag.string)
                if 'actor' in data:
                    actors = [a['name'] for a in data['actor'][:3]]
            except Exception as e:
                print(f"JSON parse error for {title}: {e}")

        row = [title] + actors
        row += [None] * (4 - len(row))
        movie_data.append(row)

        print(f"{i+1}. {title}: {actors}")

    except Exception as e:
        print(f"Error with {title}: {e}")
        movie_data.append([title, None, None, None])

actor_df = pd.DataFrame(movie_data, columns=["Title", "Actor1", "Actor2", "Actor3"])
print(f"\nSaved {len(actor_df)} movies with top 3 actors.")
```

```
1. 1. The Shawshank Redemption: ['Tim Robbins', 'Morgan Freeman', 'Bob Gunton']
2. 2. The Godfather: ['Marlon Brando', 'Al Pacino', 'James Caan']
3. 3. The Dark Knight: ['Christian Bale', 'Heath Ledger', 'Aaron Eckhart']
4. 4. The Godfather Part II: ['Al Pacino', 'Robert De Niro', 'Robert Duvall']
5. 5. 12 Angry Men: ['Henry Fonda', 'Lee J. Cobb', 'Martin Balsam']
6. 6. The Lord of the Rings: The Return of the King: ['Elijah Wood', 'Viggo Mortensen', 'Ian McKellen']
7. 7. Schindler's List: ['Liam Neeson', 'Ralph Fiennes', 'Ben Kingsley']
8. 8. Pulp Fiction: ['John Travolta', 'Uma Thurman', 'Samuel L. Jackson']
9. 9. The Lord of the Rings: The Fellowship of the Ring: ['Elijah Wood', 'Ian McKellen', 'Orlando Bloom']
10. 10. The Good, the Bad and the Ugly: ['Clint Eastwood', 'Eli Wallach', 'Lee Van Cleef']
11. 11. Forrest Gump: ['Tom Hanks', 'Robin Wright', 'Gary Sinise']
12. 12. The Lord of the Rings: The Two Towers: ['Elijah Wood', 'Ian McKellen', 'Viggo Mortensen']
13. 13. Fight Club: ['Brad Pitt', 'Edward Norton', 'Meat Loaf']
14. 14. Inception: ['Leonardo DiCaprio', 'Joseph Gordon-Levitt', 'Elliot Page']
15. 15. Star Wars: Episode V - The Empire Strikes Back: ['Mark Hamill', 'Harrison Ford', 'Carrie Fisher']
16. 16. The Matrix: ['Keanu Reeves', 'Laurence Fishburne', 'Carrie-Anne Moss']
17. 17. Goodfellas: ['Robert De Niro', 'Ray Liotta', 'Joe Pesci']
18. 18. One Flew Over the Cuckoo's Nest: ['Jack Nicholson', 'Louise Fletcher', 'Michael Berryman']
19. 19. Interstellar: ['Matthew McConaughey', 'Anne Hathaway', 'Jessica Chastain']
20. 20. Se7en: ['Morgan Freeman', 'Brad Pitt', 'Kevin Spacey']

Saved 20 movies with top 3 actors.
```

In [10]:

```python
actor_df['Title'] = actor_df['Title'].apply(lambda x: re.sub(r'^\d+\.\s*', '', x))
actor_df.head()
```

Out[10]:

	Title	Actor1	Actor2	Actor3
0	The Shawshank Redemption	Tim Robbins	Morgan Freeman	Bob Gunton
1	The Godfather	Marlon Brando	Al Pacino	James Caan
2	The Dark Knight	Christian Bale	Heath Ledger	Aaron Eckhart
3	The Godfather Part II	Al Pacino	Robert De Niro	Robert Duvall
4	12 Angry Men	Henry Fonda	Lee J. Cobb	Martin Balsam

In [11]:

```python
actor_df = actor_df.fillna('')

actor_df['AllActors'] = actor_df[['Actor1', 'Actor2', 'Actor3']].values.tolist()

all_actors = sorted(set(actor for sublist in actor_df['AllActors'] for actor in sublist if actor))

actor_matrix = pd.DataFrame(0, index=actor_df['Title'], columns=all_actors)

for idx, row in actor_df.iterrows():
    for actor in row['AllActors']:
```

```python
        if actor:
            actor_matrix.at[row['Title'], actor] = 1

similarity_matrix = cosine_similarity(actor_matrix)
similarity_df = pd.DataFrame(similarity_matrix, index=actor_df['Title'], columns=actor_df['Title'])
```

In [12]:
```python
title_to_actors = dict(zip(actor_df['Title'], actor_df['AllActors']))

def recommend_movies_simple(input_title, top_n=5):
    if input_title not in title_to_actors:
        return f"Movie '{input_title}' not found."

    input_actors = set(title_to_actors[input_title])
    similarities = {}

    for title, actors in title_to_actors.items():
        if title == input_title:
            continue
        overlap = input_actors.intersection(set(actors))
        similarities[title] = len(overlap)

    sorted_recs = sorted(similarities.items(), key=lambda x: x[1], reverse=True)
    return sorted_recs[:top_n]

recommendations = recommend_movies_simple("The Lord of the Rings: The Two Towers")
for title, score in recommendations:
    print(f"{title} (shared: {score})")
```

```
The Lord of the Rings: The Return of the King (shared: 3)
The Lord of the Rings: The Fellowship of the Ring (shared: 2)
The Shawshank Redemption (shared: 0)
The Godfather (shared: 0)
The Dark Knight (shared: 0)
```

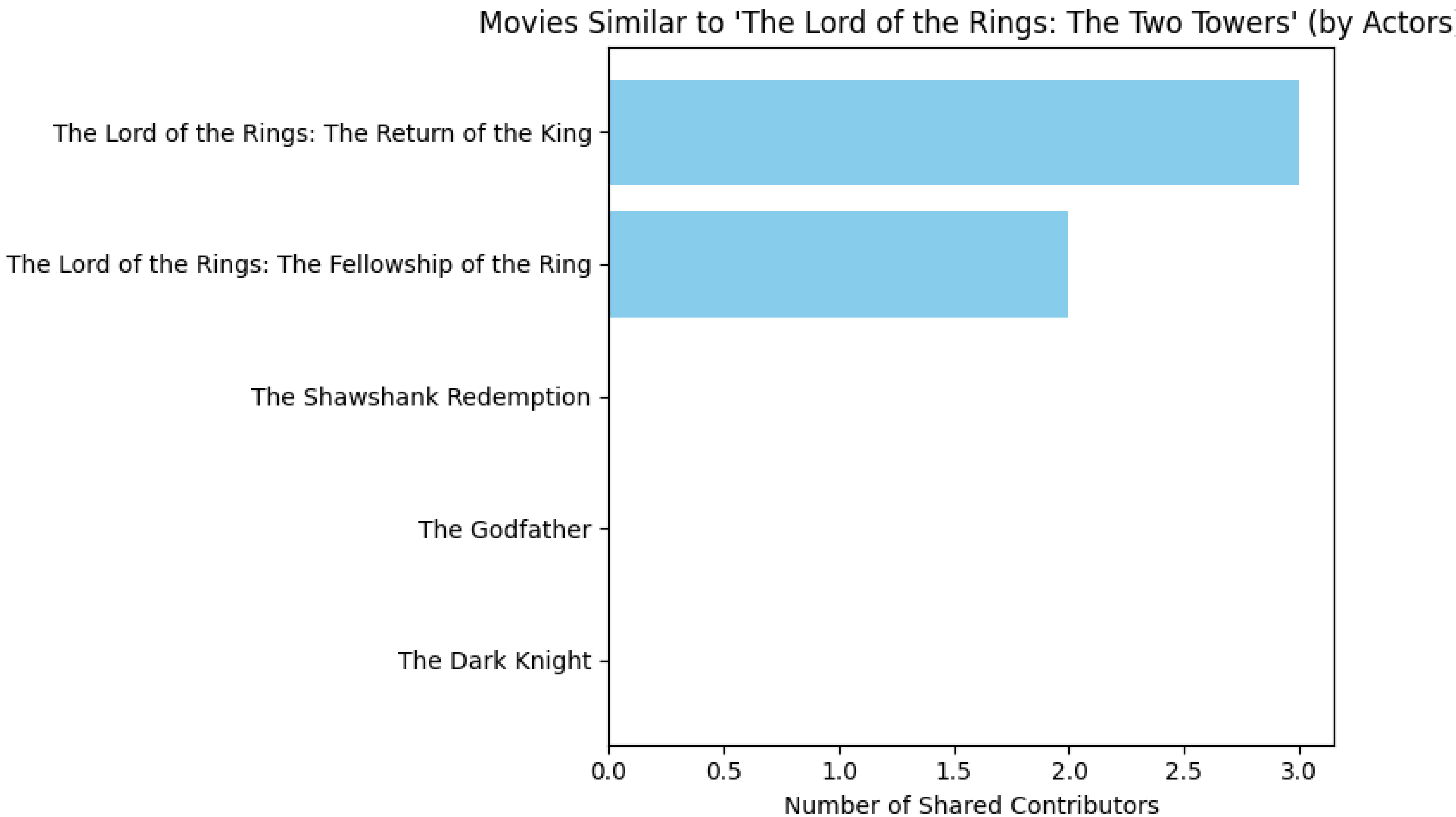
In [13]:
```python
def plot_recommendations(input_title):
    recs = recommend_movies_simple(input_title)
    if isinstance(recs, str):
        print(recs)
        return

    titles = [title for title, score in recs]
    scores = [score for title, score in recs]

    plt.figure(figsize=(8, 5))
    plt.barh(titles, scores, color='skyblue')
    plt.xlabel("Number of Shared Contributors")
    plt.title(f"Movies Similar to '{input_title}' (by Actors)")
    plt.gca().invert_yaxis()
```

```
    plt.tight_layout()
    plt.show()

plot_recommendations("The Lord of the Rings: The Two Towers")
```



In [ ]: