Social Networks Analysis

MINIPROJECT

Group Members

P T Jayannthan – 205001049

Kishaanth S – 205001054

Koushik Viswanath S - 205001055

UCS 1722 – Social Network Analysis **MINIPROJECT**

Take any social network dataset and analyse the network using the following parameters.

a) Use any one of the modern SNA tools apart from what you learned in the course such as GEPHI, GraphViz, nodeXL, Netminer etc. (K3) (5.2.2)

b) List out the questions for which you are going to find the answers from the network. (K3) (2.1.1)

c) Identify the nodes, edges, edge direction and edge weight (K3)(1.4.1))

d) Compute the parameters such as degree centrality, closeness centrality and betweenness centrality to find out the influential node in the network, a node close to all nodes and a node with maximum connections (K3)(1.4.1)

e) Find out the network level measures such as Network size and Network Density(K3)(1.4.1)

f) Find out the path level measures such as characteristic path length, average geodesic distance of the network and diameter of the network(K3)(1.4.1)

g) Identify the connected components and bridges in the network (K3)(1.4.1)

h) Apply any Community detection algorithm to detect communities in the network (K3)(1.4.1)

i) Provide a neat visualization of the network using appropriate layouts, colors, links and labels (K3) (2.1.2)

Team Information:

A team of maximum 4 members should take one project. Each team should demonstrate the project. Each team member should contribute towards the project and should explain their part to get their marks. (9.3.1)

Report Details:

One project report per team should be submitted. The report should contain the following information: Objective, Dataset Description, Description about the tool, Visualization of the network wherever necessary with explanation, Analysis or inference about the values of network parameters and conclusion. (10.1.2)

Aim

The aim is to leverage a social network dataset and conduct a comprehensive analysis by employing various visualization techniques that showcase different metrics, layouts, statistics, and community detection.

Description of the problem

The project involves analysing data retrieved from the IMDB site, specifically focusing on the top 1000 films spanning from 2006 to 2016. Our objective is to model this data as a social network, incorporating actors, films, and genres. The data, provided in CSV format, includes key information such as:
This data, given in the form of a CSV file, have the following values:

- *Rank*- ranking of the film on the top list
- *Title*- the name of the movie
- *Genre*- genres to which the film belongs
- *Description*- description of the plot of the film
- *Director*- the director of the film
- *Actors*- the main cast of actors in the film
- *Year*- the year the film was made
- *Runtime (Minutes)*- movie duration in minutes
- *Rating*- movie rating on a scale from 1 to 10
- *Votes*- the number of user votes the film received
- *Revenue (Millions)*- film earnings expressed in millions of dollars
- *Metascore*- the average rating of the critics that the film received

Technologies used

- **Python:** Employed for data manipulation and processing using the 'pandas' library, as well as for network analysis through the 'networkx' library and visualization using 'matplotlib.'
- **Gephi :** Utilized as a valuable tool for visualizing and manipulating graph structures, facilitating a comprehensive analysis of the social network dataset.

Data preprocessing

Prior to initiating the analysis, it is imperative to refine the acquired data for enhanced suitability in subsequent examinations. In this process, upon reading and importing data from the CSV file into a Pandas DataFrame, the columns 'Genre' and 'Actors' undergo a transformation. The initially comma-separated strings representing genres and actors are converted into lists, facilitating a more structured representation.
Moreover, observations of missing values in the 'Revenue' and 'Metascore' columns have been made. However, these gaps are deemed inconsequential for the current analysis, as these specific columns will not be utilized in the subsequent examinations.

Representation of data with a graph

Three graph structures have been established:

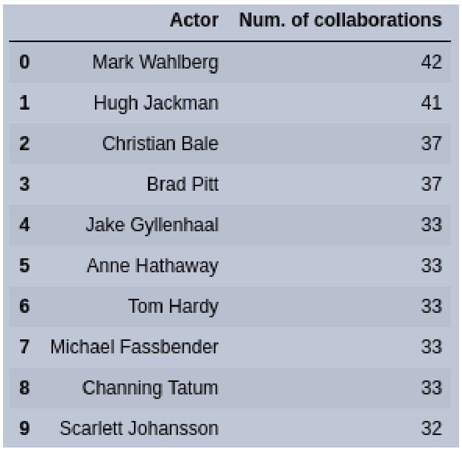
1. **Cast Graph:**
    - An undirected weighted graph representing actors as nodes.
    - Edges between actors signify how many times they have collaborated in movies.
2. **Genre Graph:**
    - An undirected weighted graph with genres as nodes.
    - Connections between genres indicate the frequency of movies falling under both genres.
3. **Movie Graph:**
    - A directed weighted graph where movies are nodes.
    - An edge from Movie A to Movie B exists if there are actors who participated in both movies, and Movie B was released after Movie A.

These structures are implemented using the **networkx** library. The data from the pandas DataFrame is transformed into the corresponding adjacency matrices for each graph type. The resulting graphs can be exported to the. graphml format, facilitating visualization and further analysis in tools like Gephi.

Analysis results

Actors with Most Collaborations:

Identifying actors who collaborated with the highest number of peers.



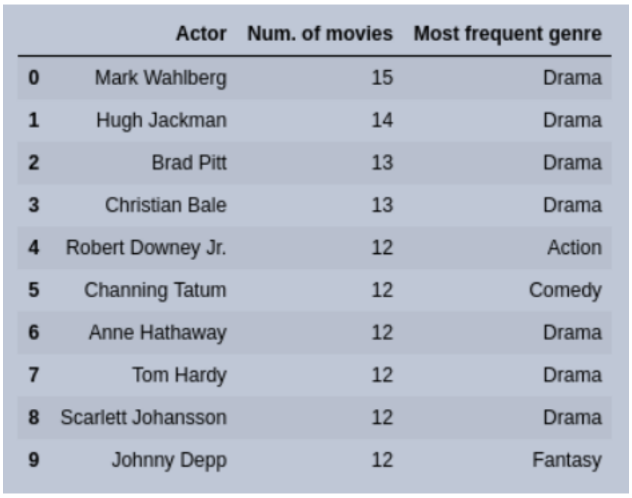
	Actor	Num. of collaborations
0	Mark Wahlberg	42
1	Hugh Jackman	41
2	Christian Bale	37
3	Brad Pitt	37
4	Jake Gyllenhaal	33
5	Anne Hathaway	33
6	Tom Hardy	33
7	Michael Fassbender	33
8	Channing Tatum	33
9	Scarlett Johansson	32

Average Number of Collaborations per Actor:

On average, actors have played alongside approximately 5.8 other actors.

Most Productive Actors and Preferred Genres:

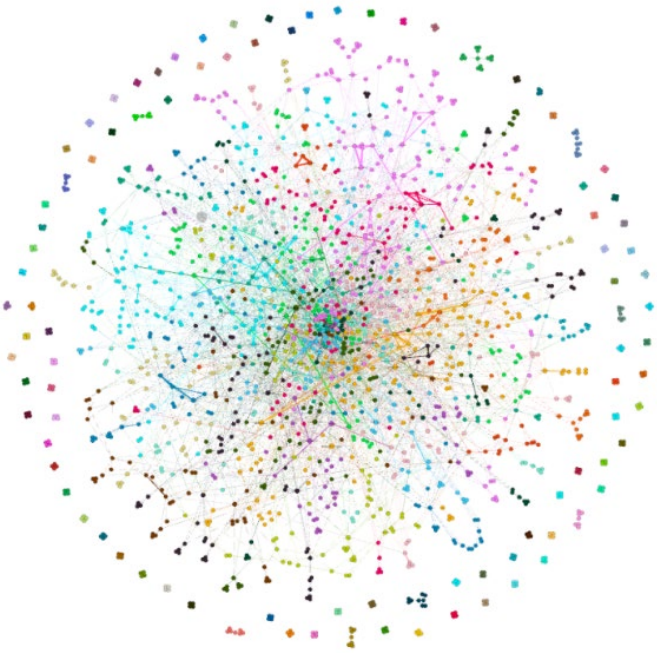
Highlighting the most prolific actresses and the genres in which they have predominantly worked.



	Actor	Num. of movies	Most frequent genre
0	Mark Wahlberg	15	Drama
1	Hugh Jackman	14	Drama
2	Brad Pitt	13	Drama
3	Christian Bale	13	Drama
4	Robert Downey Jr.	12	Action
5	Channing Tatum	12	Comedy
6	Anne Hathaway	12	Drama
7	Tom Hardy	12	Drama
8	Scarlett Johansson	12	Drama
9	Johnny Depp	12	Fantasy

Actor Communities:

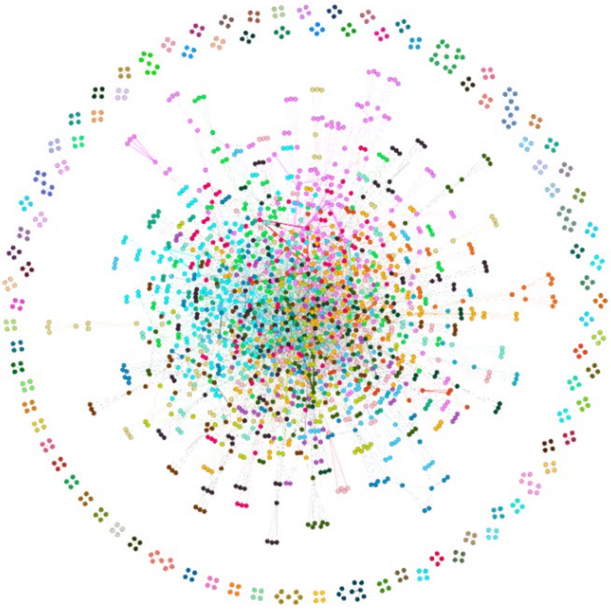
A visual representation of the actor graph reveals groupings based on modularity. Noteworthy are isolated actors in a clique, having acted in only one film.



Actor graph colored by modularity class

Genre-Based Actor Grouping:

Examining a graph illustrating actors grouped by the film genres they most frequently engage in. Notably, some actors on the graph's periphery, having acted in only one film, form distinct cliques.



Graph of the actors colored by the genre in which the largest number of their films belong

How dense is the network?

The density of the actors' network is 0.003.

To what extent is the network connected and centralized?

The network's connectivity is 1.54, indicating the average number of nodes that need removal for any two nodes to be disconnected. The centrality metrics, resembling a star-shaped graph, have relatively small values:

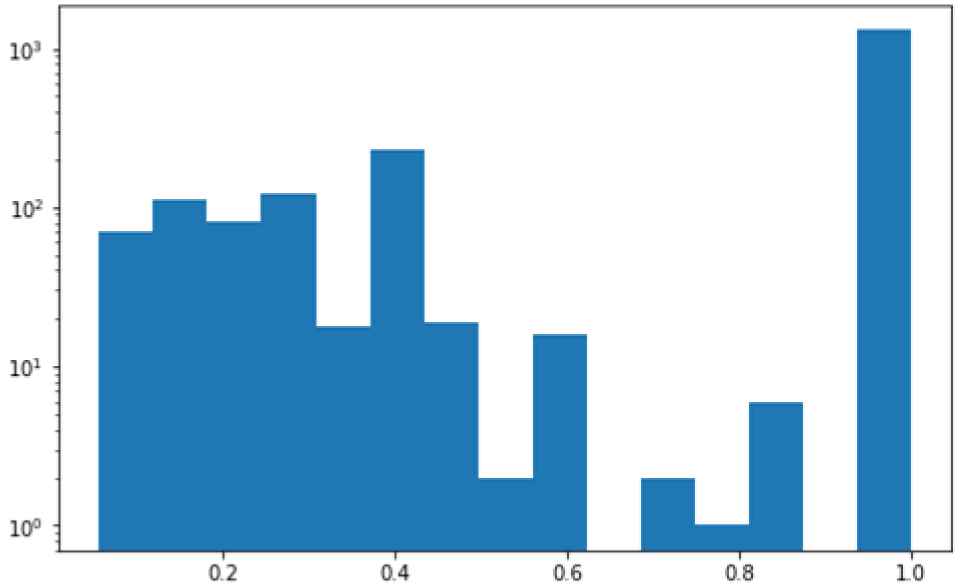
- Degree centrality: 1.83%
- Relational centrality: 2.54%
- Closeness centrality: 22.94%
- Centrality by eigenvector: 24.41%

These values suggest a balanced network with no significantly dominating nodes.

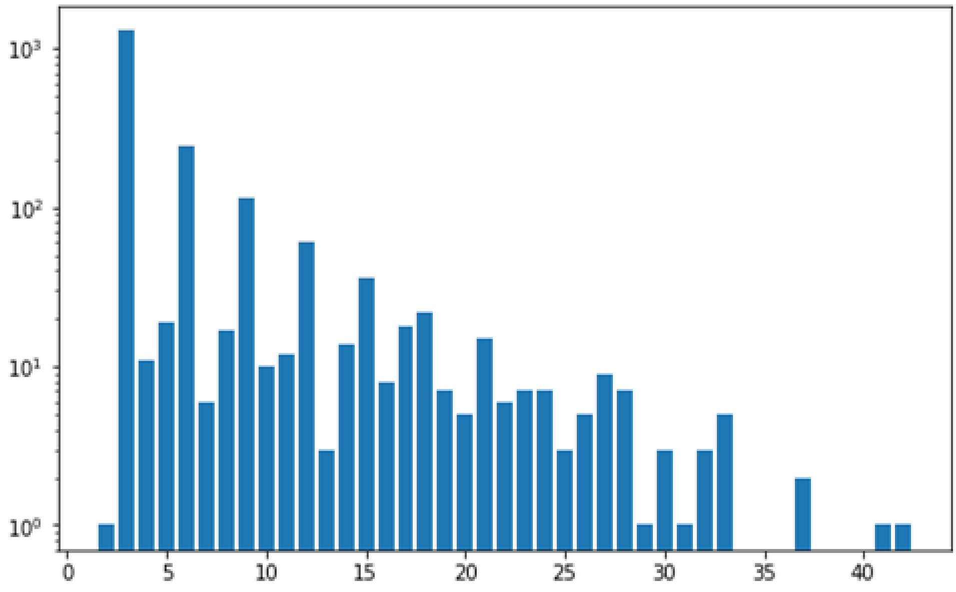
What are the average distances within the network and the diameter of the network?

The average distance between actors in the network is 4.28, and the diameter is 9. This implies that, on average, any two actors are connected by around 4 relationships, and the maximum number of connections needed to connect any two actors is 9.

Clustering Coefficient:
The network's average clustering coefficient is 0.76, indicating strong connections. Many nodes have a clustering coefficient of 1.0, suggesting the presence of cliques formed by actors who collaborated in only one movie.

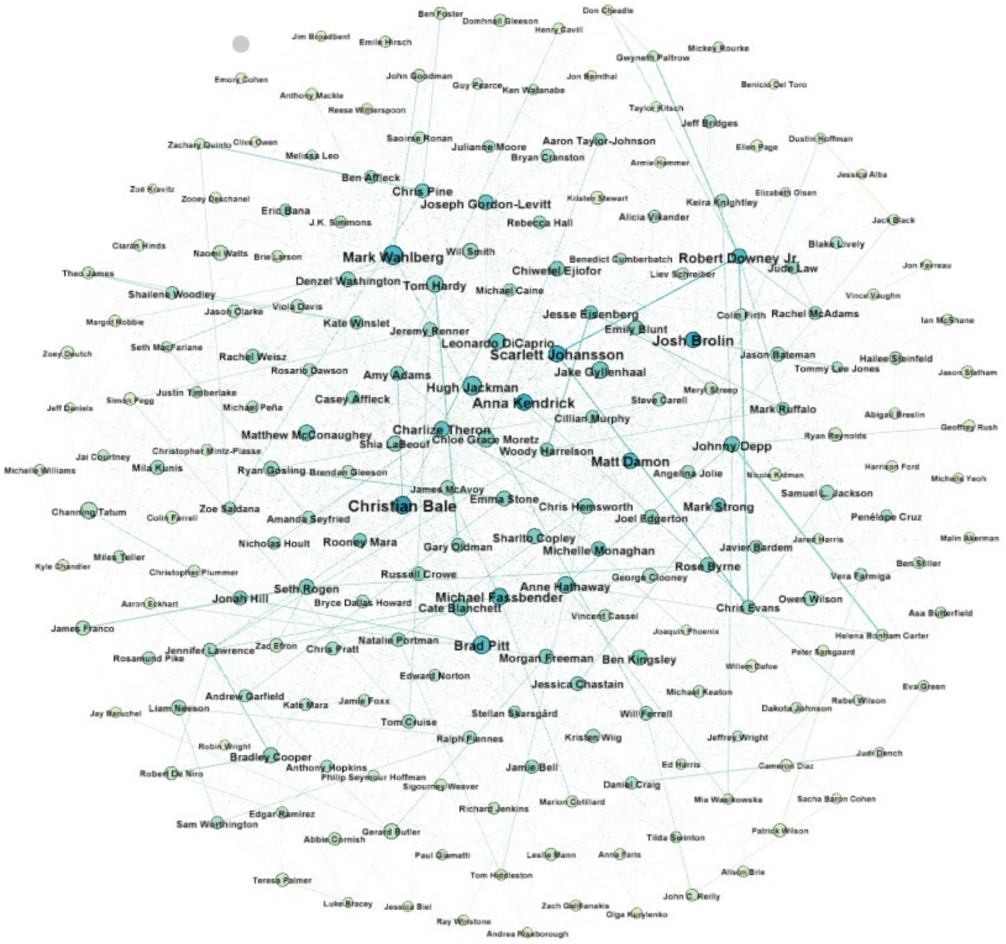


Node Degree Distribution:
The distribution of nodes follows a power-law, indicating a scale-free network. Numerous nodes have low degrees, while a few have extremely high degrees.

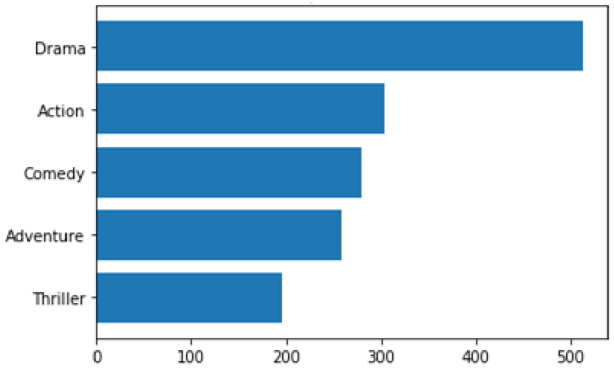


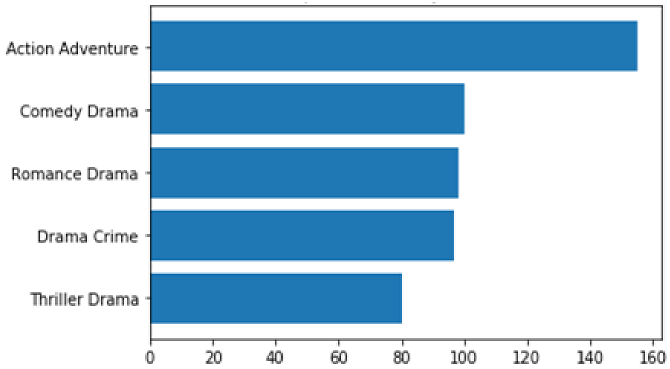
Small World Characteristics:
The network exhibits small-world characteristics with a high degree of clustering (0.76) and a small average distance between nodes (4.28).

Core Actors:
The filtered network, based on a k-core value of 6, highlights actors with a minimum of 6 connections. These actors form the well-connected core of the network.

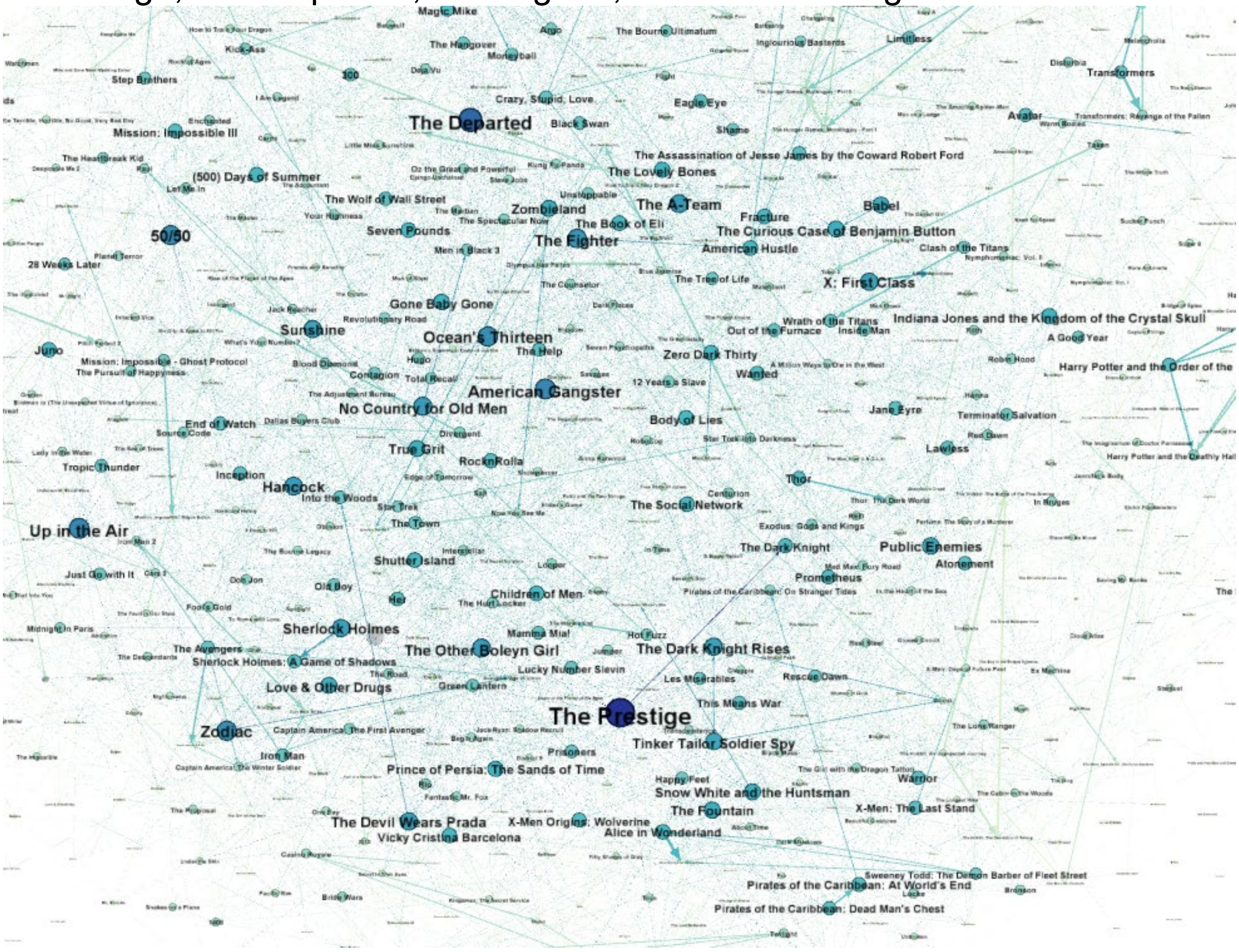


Popular Movie Genres:
Explore the popularity of movie genres and their combinations within the network.





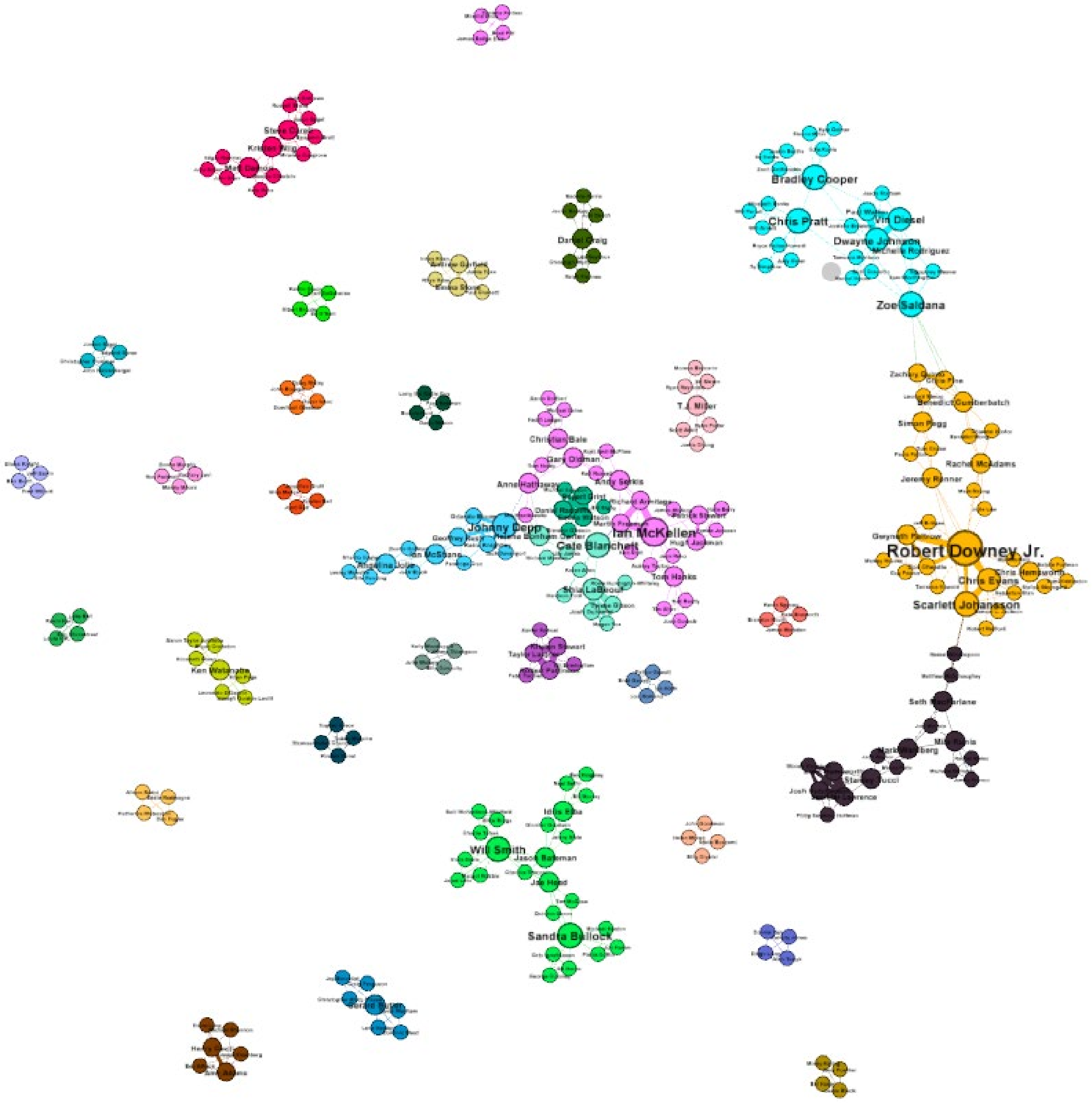
Film Influence:
Visualize films' influence on actors using a directed graph. Films like The Prestige, The Departed, The Fighter, and 50/50 emerge as influential.



Network Changes with Earnings Filter:
Filtering the network to the top 100 movies by earnings results in changes:

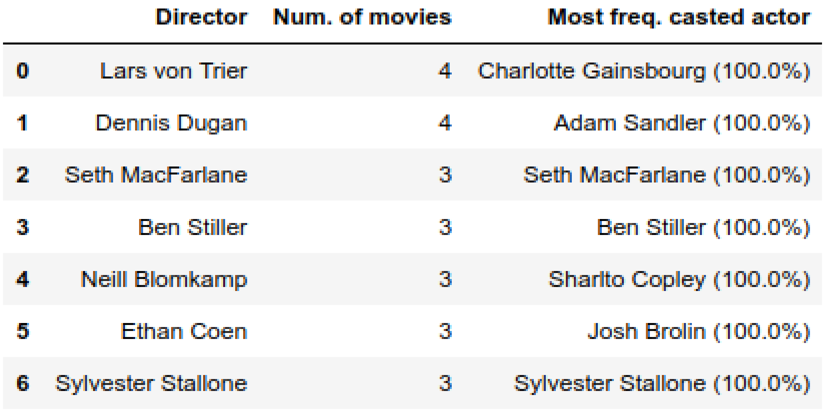
- the mean degree of the network nodes decreases to **3.79**

- the network density increased to **0.014**
- the diameter of the net increases to **12**
  (average distance remains similar as before with 4.15)
- average coef. of clustering increases to **0.86** and we notice much more pronounced clusters (noticeable clusters of actors from popular movie Franchises: Batman, Avengers, Harry Potter, Fast & Furious etc.)



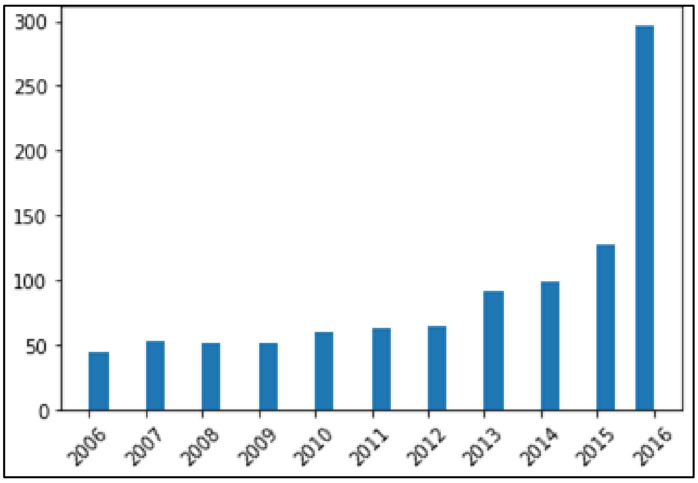
Which director directed the most films?
Ridley Scott directed the most films, with a total of 8.

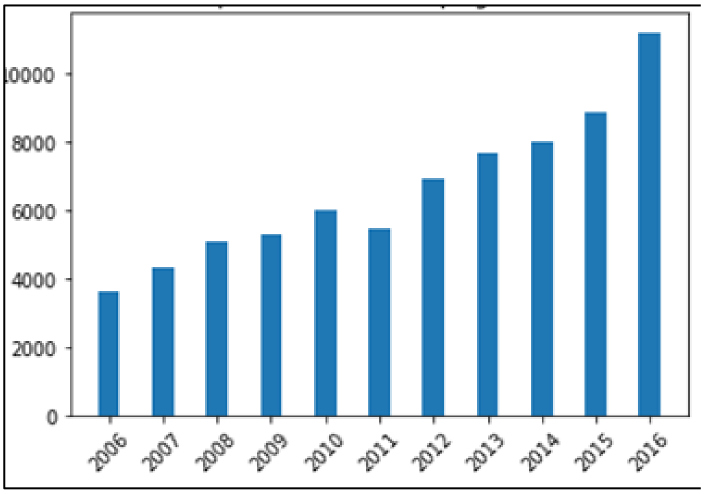
Director-Actor Collaboration:
Identify directors who frequently collaborate with actors. Three directors consistently act in all their films.



	Director	Num. of movies	Most freq. casted actor
0	Lars von Trier	4	Charlotte Gainsbourg (100.0%)
1	Dennis Dugan	4	Adam Sandler (100.0%)
2	Seth MacFarlane	3	Seth MacFarlane (100.0%)
3	Ben Stiller	3	Ben Stiller (100.0%)
4	Neill Blomkamp	3	Sharlto Copley (100.0%)
5	Ethan Coen	3	Josh Brolin (100.0%)
6	Sylvester Stallone	3	Sylvester Stallone (100.0%)

Peak Film Production Year:
Determine the year with the highest film production based on the dataset.


300
250
200
150
100
50
0
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016


10000
8000
6000
4000
2000
0
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016