Actors tour of data science "From popularity to rating"

Team 8 – Adrian Villarroel, Andres Montero, Ariel Alba, Elias Poroma.

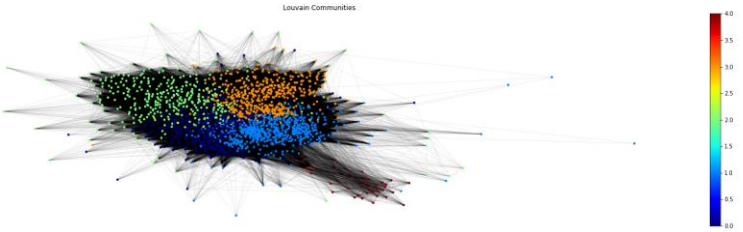
1. Story

The idea of the project is to find communities of actors, explain how they are related within their communities, find the most representative actors and estimate the signals of the actors with a machine learning model. The IMDb dataset is relevant for this work because it provides reliable information about the movies each actor has performed on, the people he/she has interacted with, the production companies he/she has worked for, etc.

2. Acquisition

The dataset can be downloaded from Kaggle: https://www.kaggle.com/tmdb/tmdb-movie-metadata. We are building a graph from the data and analyzing it with the following specialized tools: Pandas, Scikit-learn, Networkx, Matplotlib, Python Louvain.
The dataset consists of two tables containing features from movies.
For the creation of the graph we calculated the "affinity" (weights) between two actors by counting the number of common elements between them among the following categories: movie_id, cast, crew, genre and production companies. The diameter of the graph is 4, meaning that any actor is 4 steps away of knowing any other actor.

A visualization of the network:



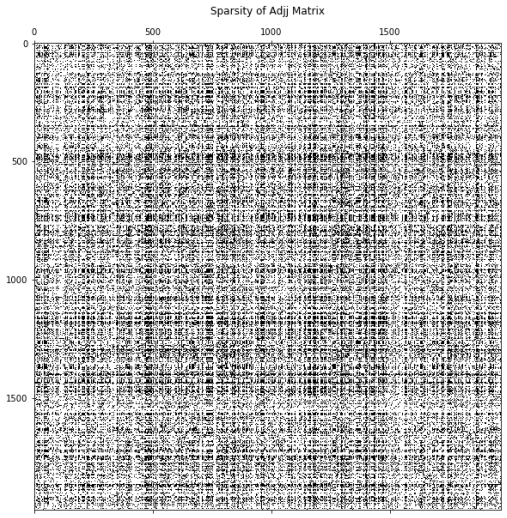
We used Louvain's algorithm to find the communities of the graph, this algorithm is a bottom up approach to find communities based on the modularity of the nodes. The idea is to use these values as a feature so that later we can train a Machine Learning model to do a regression. This model will be used to estimate the signal values of each actor like revenue, popularity, etc.

3. Next steps

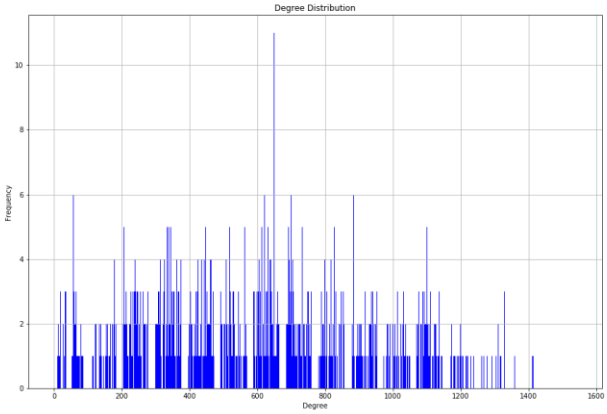
- Analyze the communities and try to find the relationships that explain them.
- Find most representative actors of each community.
- Create appropriate visualizations.
- Machine Learning model to do a regression with the communities' labels (louvain graph) as a feature so that we can estimate the signal values of the actors (revenue, popularity, etc.).

4. Plots

- Sparsity of the Adj Matrix:



- Degree Distribution



- Spectrum