Actors Tour of Data Science

Adrian Villarroel, Andres Montero, Ariel Alba, Elias Poroma

EPFL - 2019

1 Introduction

The idea of the project is to estimate and predict quantities related to the perception of the audience related to a movie with respect to the actors, i.e. the hiring cost of an actor. This information is crucial during the cast-selection period as it plays a significant role in the success of the movie. We believe that this is important given that few datasets provide data related to actors; most the datasets have information only about the movies. We propose to, first, do an estimation of this information by doing an aggregation on the movie data, then, train a linear regression model to estimate the quantities of interest. We also enhance the model by leveraging information from the social network graph of actors, for this, we are using the Louvain algorithm to find communities on the graph and then use this information as a feature on the model. We believe that the social relationship will have a major influence on the quantities of interest. To quantify this, we will compare the baseline with the information of the graph using Louvain and filters. As the number of actors is around 54000 we restrict our analysis only to the actors that have been protagonists in at least one movie. We also remove movies that have the quantities of interest (e.g. budget, revenue, popularity) equal to 0. Finally, we will work with around 3000 valid movies.

2 Data set

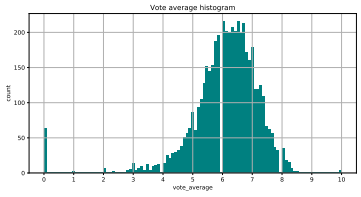
The IMDb dataset is relevant for this work, because it provides reliable information about the movies on which each actor has performed on, the people he/she has interacted with, the production companies he/she has worked for, etc. We can use this information to infer the affinity between actors by looking at how many of these elements they have in common. The dataset can be downloaded from kaggle: https://www.kaggle.com/tmdb/tmdb-movie-metadata.

2.1 Tools

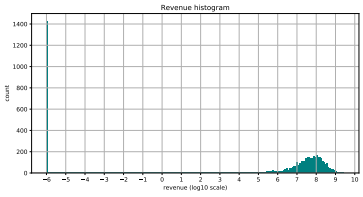
We are building a graph from the data and analyzing it with the following specialized tools: Python, Pandas, Seaborn, Gephi, Pygsp, Scikit-learn, Networkx, Matplotlib, Pythoun Louvain and others.

2.2 Data Pre-Processing

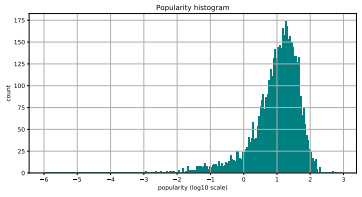
The dataset consists of two tables containing a variety of features from movies. To explore and obtain the statistics of the data we have the following figures:



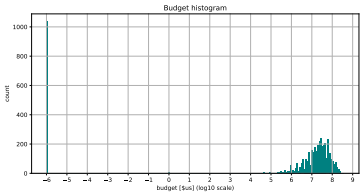
(a) Votes Histogram



(b) Revenue Histogram



(c) Popularity Histogram



(d) Budget Histogram

Figure 1: Histograms of the signals of interest

As we are interested in movies that have values in the quantities of interest we filter the movies that have these values equal to zero. Then, we convert the json information to lists, so that, we can do the aggregation for each actor that played at least one time as a protagonist in one of the previously filtered movies.

3 Exploration

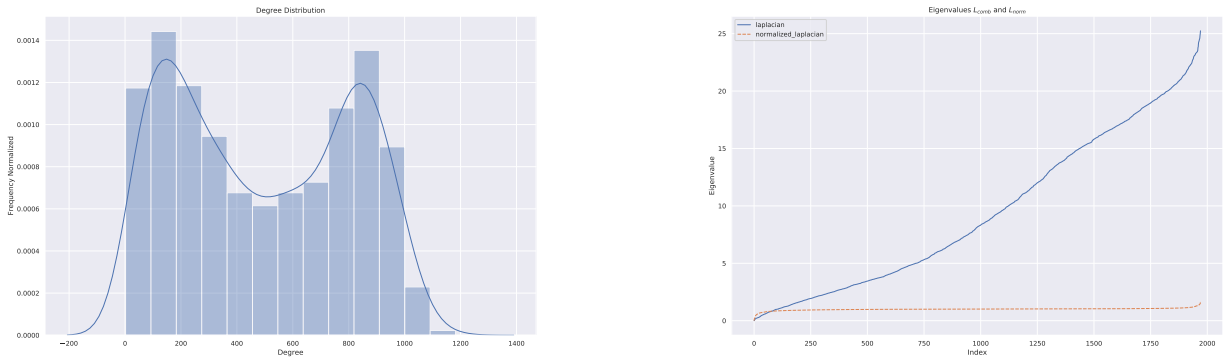
3.1 Graph Creation

As we wanted to create a graph of actors we had to process the dataframes in the following way:

- Transform the dataframe to actors instead of movies
- Aggregate features for each actor
- Define weight between two actors:
  $w_{ij} = \frac{0.3|movie_i \cap movie_j| + 0.3|cast_i \cap cast_j| + 0.2|crew_i \cap crew_j| + 0.1|genre_i \cap genre_j + 0.1|companies_i \cap companies_j}{0.3|movie_i \cup movie_j| + 0.3|cast_i \cup cast_j| + 0.2|crew_i \cup crew_j| + 0.1|genre_i \cup genre_j + 0.1|companies_i \cap companies_j}$

3.2 Graph Properties

- Connected Components: Given the weight definition presented before we have one connected component.
- Diameter: The diameter of the graph is 4, meaning that, any actor is 4 actors away of knowing any other actor.
- Degree Distribution: The degree distribution of the graph can be observed in figure 2.
- Type of Graph: The network is a small world, to get this assumption, a similar generated Erdos-Renyi network was created and network statistics like the clustering coefficient and the mean of the shortest path were calculated for both cases. The Mean shortest path is the same, however, the clustering coefficients are different. Small networks should have some spatial structure that is reflected in a bigger clustering coefficient.
- Properties of the nodes:
  - Average Degree: 492.75
  - Average clustering coefficient: 0.6016
  - Actors that have more dense connections: Eric Bogosian 0.600, Patrick Dempsey 0.583.
  - Actors that dont have dense connections: Karra Elejalde: 0.004, Ellar Coltrane: 0.003.
  - Hub Nodes: a total of 958 hubs
- Spectrum: both laplacian and normalized laplacian can be observed in figure 2.



(a) Degree Distribution

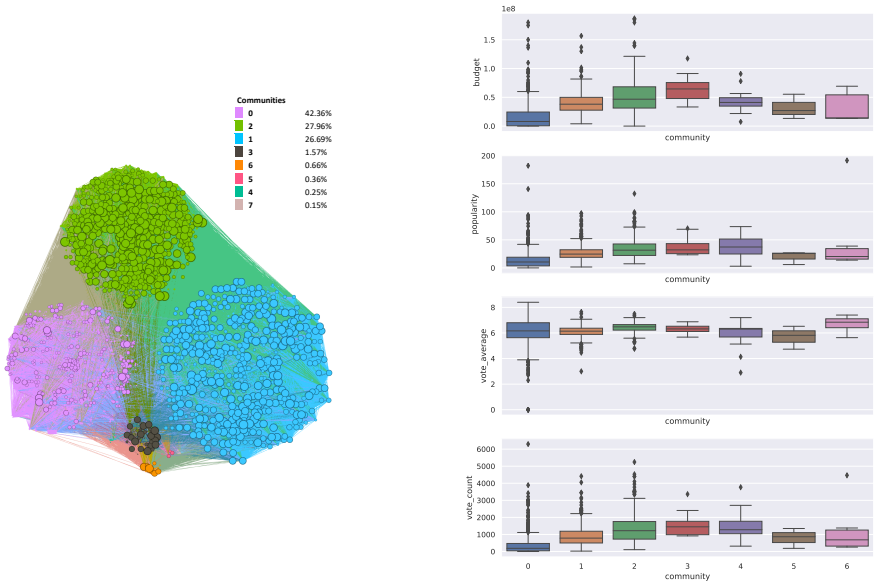
(b) Laplacian and Normalized Laplacian

Figure 2: Degree Distribution and Laplacian spectrum

4 Exploitation

4.1 Louvain

The idea of a community detection algorithms is based on a hierarchical approach, i.e. within communities, other sub-communities can be identified until they arrive at the individual nodes. Louvain is an agglomerative community detection based on the use of the modularity that allows assessing the quality of a community when clustering [1]. We used Louvain algorithm to have the communities of the graph as features, thus, we can have more information for the next task, train a machine learning model to do a regression so that we can estimate some signal values of each actor (revenue, popularity, etc). The communities and its statistics can be seen in figure 3.



(a) Louvain Communities graph

(b) Statistics Louvain

Figure 3: Louvain Communities

The most important actors for revenue and popularity according to the communities made by louvain can be observed in figures 4 and 5.

Figure 4: Most important Actor for Revenue

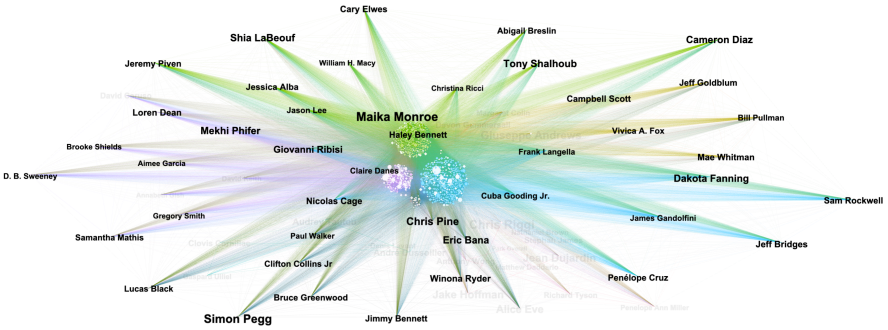
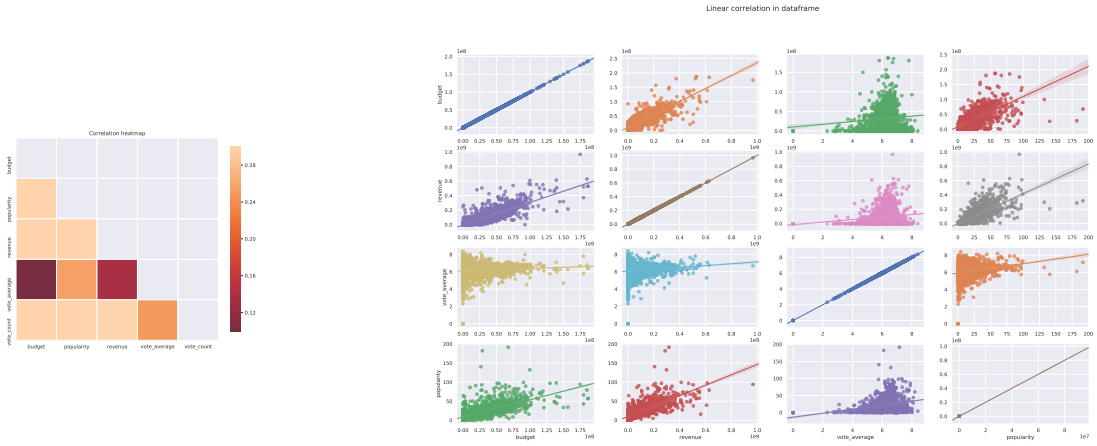


Figure 5: Most important Actor for Popularity

4.2 Linear Regression, Filters and Tikhonov Regularization

First, we analyzed the correlation of the features, this can be seen in figure 6.



(a) HeatMap

(b) Pearson

Figure 6: Correlation of Features

Then, a Linear Regression model is implemented on the graph, as a baseline we implemented a vanilla linear regression, then, linear regression with communities, i.e. adding information found when Louvain is applied, and finally filters are also used in the model. The filters created are: low pass, band pass, high pass and tikhonov. The filters are created using pygsp [2]. For example the filters for the signal of revenue can be seen in figure 7

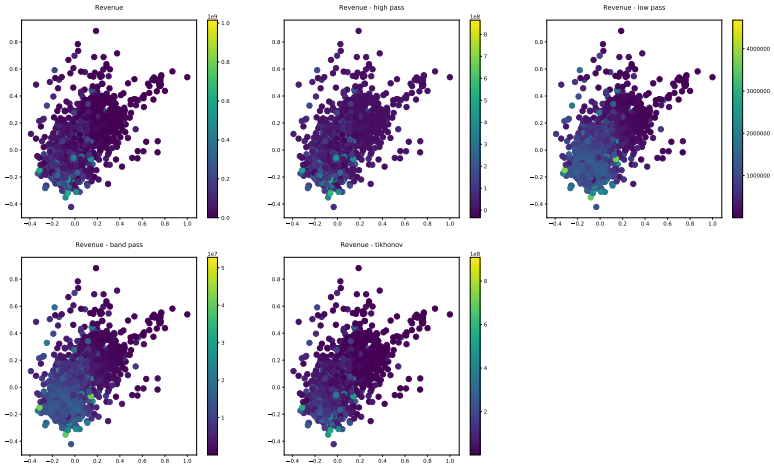


Figure 7: Filters of Revenue

5 Results

The overall results of the project is summarized in the table 1. We can see that the best results are obtained using a linear regression with the communities given by Louvain algorithm and the filter band pass, this is explained because filter bandbass is like an outlier removal, "filtering" the signals with low and high values.

NMAE	VLR	VLRC	LRHP	LRHPC	LRLP	LRL{C	LRBP	LRBPC	LRTK	LRTKC
Budget	0.0537	0.0489	0.0630	0.0583	0.0302	0.0295	0.0299	**0.0291**	0.0505	0.0465
Revenue	0.0447	0.0435	0.0360	0.0354	0.0206	0.0206	0.0199	**0.0199**	0.0283	0.0281
Popularity	0.0267	0.0265	0.0647	0.0637	0.0437	0.043	0.0432	**0.0425**	0.0607	0.0596
Vote Average	0.0834	0.0827	0.0334	0.0699	0.0334	0.033	0.0333	**0.0334**	0.0519	0.0519

Table 1: Linear Regression Results

6 Conclusion

The objective of creating a model able to reliably estimate the quantities of interest for the success of a movie based on the main actor featuring in the movie is achieved. We use the Louvain algorithm to create communities and use this information as features for the linear regression model, also develop filters for each quantity of interest to exploit the information of the graph. In the end, the best results are obtained with a linear regression model with the communities as features and a bandpass as a filter. We identified the most important actors for each quantity of interest and the overall most important actors for the success of a movie.

References

[1] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. 2008.

[2] EPFL LST2 LAboratory. Pygsp. https://pygsp.readthedocs.io/en/stable/, 2020.