Eurovision Network Analysis

Marco Aspromonte
marco.aspromonte@studio.unibo.it

Francesco Romito
francesco.romito2@studio.unibo.it

1 Abstract

The aim of this project is to develop a **framework** for the analysis of the Eurovision song contest. First we have defined several techical analysis about the structure of the network constructed on Eurovision's datasets by building some metrics on nodes and edges, then we focused on the interpretations of the results considering also affinities and differences between a temporal window from 2017 to 2021.

2 Context

We know that in general the **Social Network Analysis** is the discipline that allows the extrapolation of social information, tipically on graphs. **Graphs** are particular data structures that are really helpful to describe relationships between the participants (nodes) of the contest, in this case such relations are described with edges between the nodes of our network, which are numerical ones that can be used to extrapolate semantic correspondences. For example the Eurovision song contest is very interesting in terms of a geo-political study, because we could focus on musical or other kinds of influences of the countries that compete in the contest. The **Eurovision Song Contest** is an international songwriting competition organised annually by the European Broadcasting Union, featuring participants representing primarily European countries, but there are also some exceptions, for example the Israel. For our purposes we decided to analyze datasets regarding editions up to 2017, since starting from 2015 the voting system has been slightly changed, because in the new editions the final vote is decided for the 50% by the jury and for the other 50% by the televoting, while previously it was decided by only the first one. So in order to have a coherent set of datasets, we excluded all the previous editions into our study. Because of the Eurovision contest involves about 40 countries between voters and voted, there are some interesting cultural influences that we can take into account during our analysis. From a techical point of view, we built and analyze the graph using a python enviroment called **Jupyter Notebook** in which we have easily used some of the needed libraries.

3 Problem and Motivation

We are in a society where we are sometimes moved by prejudices towards foreign countries, this is an aspect that is inherited in different fields of our life, as in the music; this is not really a good thing because the music is a form of art understood as freedom of expression. So it could be interesting to see how the political and geographical aspects influence the results of an international contest as Eurovision that should be impartial and objective. The idea is to translate these aspects deploying a clustering algorithm taking into account first only the votes assigned from the countries, then looking for indipendent and exogeneous aspects as the geographical position or social dynamics; basing on them we would define and extrapolate different conclusions about countries' voting behaviours. The idea is that the model is generalizable , so except for the preprocessing section of the datasets, it could be useful to see how the same analysis changes during a temporal window of 4 editions (from 2017 to 2021) and check if there are similar or dissimilar trends.

4 Datasets

In order to deploy our analysis in a more generalized and complete way, we took into account different editions of the Eurovision contest, more editions mean more datasets, in particular the first one referring to **2021**, is apart from the others that belong to the same dataset and that we split into more DataFrames , each one referring to a single year. A recurrent problem dealing with different datasets is that they are dirth in different ways so they need different kind of cleaning processes; because of this reasons, we have preprocessed them forcely in two different ways in order to have a compatible structure between them, in this way we can extend the very same analysis for each split. In the two following subsections we describe such cleaning techinques in according to the structure of the datasets, the first for the last edition of 2021, the second for the 2017-2019 period.

4.1 2021 Edition Dataset

The first dataset is in turn divided in two parts, the jury votes into **'eurovision_juryvotes_2021.csv'** file, and the televotes into **'eurovision_televotes_2021.csv'**. When we talk about 'DataFrame', we refer to a given 'pandas' structure that is comfortable suite to use in terms of preprocessing and manipulation, for example it is simple to import a 'csv' format dataset with `read_csv` function. The preprocessing procedure is defined in the following function:

```
def preprocessing_dataset(data):
  data = data.drop(’index’, axis = 1).T
  data["Total"] = data.sum(axis=1)
  data["Total"][4] = 9999
  data = data.sort_values(by=[’Total’], ascending=False).T
  data = data.rename({’Country (Voters (vertical), Finalists (horizontal))’: ’ind’},
  data = data.set_index(’ind’)
  data.rename({9999: ’Total’}, axis=0, inplace=True)
  data = data.T
  data = data.reset_index()
  data = data.rename({’index’: ’Country’}, axis=’columns’)
  data = data.reset_index()
  data = data.rename({’index’: ’Rank’}, axis=’columns’)
  return data
```

Except for the basic functions of renaming and dropping, in order to have a handly dataset, in the ’csv’ we have the voting on the rows and the voted on the columns, so in order to extrapolate the ’Total’ column, in which we have the sum of all the votes that each country took, we consider the transpose of the original dataset and we make the sum of the column, which have the maximum value set to ’9999’. In this way we obtain the dataset in the following form :

ind	Rank	Country	Albania	Australia	Austria	Azerbaijan	Belgium	Bulgaria	Croatia	Cyprus	...	Russia	San Marino	Serbia	Slovenia	Spain	Sweden	Switzerland	Ukraine	United Kingdom	Total
0	0	Italy	14	13	14	10	10	22	22	18	...	20	22	20	22	10	13	18	24	3	524
1	1	France	16	11	14	6	12	17	13	15	...	6	22	22	8	24	12	18	15	17	499
2	2	Switzerland	24	15	15	4	16	3	13	2	...	6	7	2	12	17	10	0	15	9	432
3	3	Iceland	0	20	22	0	8	0	14	0	...	1	0	12	13	12	17	10	10	20	378
4	4	Ukraine	4	15	7	14	20	6	8	6	...	7	5	8	7	6	12	2	0	4	364

5 rows × 42 columns

Figure 1: Example of the DataFrame, shown with head() function

However it is not still a pretty **DataFrame**, so we consider a ’melted’ version of it, by calling the `melt` function that tranforms the dataset, ending up with a new column called ’points’, as we can check in the following figure:

	Rank	Country	Total	Source Country	points
0	0	Italy	524	Albania	14
1	1	France	499	Albania	16
2	2	Switzerland	432	Albania	24
3	3	Iceland	378	Albania	0
4	4	Ukraine	364	Albania	4

Figure 2: Example of the melted DataFrame, shown with head() function

Now, for each row, we have the points that a given ’Source Country’ (in which there are voting and voted countries) , gives to another one in ’Country’, (in which there are only voted). Moreover, for each voted country we have the Total number of votes that it received. Finally we drop the ’Rank’ column, since we are not dealing with it.

4.2 2017-2019 Edition Dataset

As already pointed out, for the seek of completeness of our research, we have also introduced a set of 3 additional DataFrames which are derived from a massive dataset called ‘eurovision_song_contest_1975_2019.xlsx’ in which there are all the datas regarding the Eurovision from 1975 to 2019.
This dataset is for sure more dirty with respect to the previous one, with a lot of NaN and missing values, so we need a more complex preprocessing phase, we have defined an indipendent notebook called **’Cleaning_Eurovision.ipynb’** indeed. Once imported with pandas, the dataset is in a form similar to the previous one after the melt function, so for each row basically we have the point assigned from a country to another but in this case is also specified:

- the column **Year**: in which year has been given that vote
- the column **(semi-) final**: in which phase of the competition (of course semi-final or final) these points has been given
- the column **Jury or Televoting**: if the vote has been given by the Jury or Televoting system

For our purposes we need only the final step of the competition, so we have done this by dropping all the rows which have a value different from ‘f’ in the ’(semi-) final’ field and we need also the information regarding only the years following the 2017, so we do that by the line `df = df.drop(df[df['Year'] < 2017].index)`.
Finally we can drop all the columns which do not interest us, so all the others except for ‘From country’, ‘Points’, ‘To country’ and ’Tot’, so the dataset will appear exactly as the previous melted version :

	From country	To country	Points	Tot
0	Albania	Armenia	7	79
1	Albania	Australia	4	173
2	Albania	Austria	0	93
3	Albania	Azerbaijan	2	120
4	Albania	Belarus	0	83

Figure 3: 2017 post-processed dataset example

4.3 Countries dataset

The **’countries.csv’** dataset contains some information about the countries of all the world; we focus our attention on ’name’, ’latitude’ and ’longitude’ columns, that are useful to define the geographical property. Fortunately it is really clean and small, so it does not need any preprocessing procedure, but only a conversion into pandas DataFrame. Here is a little view of the dataset:

country	latitude	longitude	name
AD	42.546245	1.601554	Andorra
AE	23.424076	53.847818	United Arab Emirates
AF	33.93911	67.709953	Afghanistan
AG	17.060816	-61.796428	Antigua and Barbuda
AI	18.220554	-63.068615	Anguilla

Figure 4: Countries dataset

5 Validity and Reliability

The **validity** of our model is strictly related to the kind of datas we have defined and managed. In order to explain how much consistent and valid is our model, we have deployed some metrics on datas that in general reflects the reality. The following examples are computed by considering the 2021 dataset, but it can be swapped with your chosen year.

- From the preprocessing procedure we dropped the NaN values, but it is not a so useful procedure because there were not so much missing values, so the built graph represents faithfully the connections between the countries.

- Removing an important node from the graph (for example the node that represents the winner, in this case, Italy): by this procedure we lose a lot of connections from different countries defining a lot of missing votes, however the validity for all the other nodes is still preserved because we kept all the information about the rest of the ranking

- Checkin the maximum number of votes given to a single nation from each voting nation:

  ```
  Albania            24
  Cyprus             24
  ...
  ```

  From the previous output we can see that the maximum is 24, that is coherent with the maximum allowed amount of points that can be assigned to a given country.

- How many times a nation got the maximum points:

  ```
  France         2
  Italy          1
  ...
  ```

  It is curious to see that the France is the unique country that recieved two times the maximum points, but it is coherent with its 2 place.

- How many times a nation got the minimum points:

  ```
  United Kingdom    38
  Germany           36
  ...
  ```

  As we can expect, the UK took the minimum of the points in general , this is coherent with its last place and with its zero collected points.

- How many times a nation gives the minimum points:

  ```
  Latvia             14
  Romania            13
  ...
  ```

  This result could suggest a sort of resentment by the countries that took the last positions in the previous edition.

We can at the end assert that the internal validity of our model could be generalized in an external context because it reflects faithfully the real world, however it is difficult to quantify the exogeneous variables that explain the nature of the connections that in our case is only described by a number that indicates the vote, therefore in that vote we could extrapolate extra-aspects if we do some research in geo-political relations between certain countries, as we do in the following 6.3.3 section.

6 Measures

For the aim of the study, we define two classes of measures applied to our graph: the Network measures and Nodes measures, so it is useful to describe them separately.

6.1 Network Measures

In order to describe the property of the entire network we have defined two kind of studies, the analysis of the **Cohesion** and a **Clustering Algorithm**.

- We measure the **cohesion** trough density function, which output is a value between 0 (no-edges) and 1 (fully-connected); if such value is closer to 0 , the graph is less cohesive of course. First we compute it for all the graph, but it can be more interesting if we divide the domain of our research in two groups of countries, one for the central-west countries, the other for the east ones. Such division is balanced in terms of the number of countries between east and west dominion.
- We choose to define a **hierarchical clustering algorithm** exposing the results in the form of an histogram. The clusters are defined basing on how much countries are similar with respect to the given votes. In fact, the values considered for the **clusterization** are stored in the column 'points' of the dataset. There are two main methods to build the clusters, **complete** or **ward** that can be changed in the parameter **method** of the **linkage** function. The difference between them is that in the first method the proximity between two clusters is the proximity between their two most distant objects, in the second the proximity between two clusters is the magnitude by which the summed square in their joint cluster will be greater than the combined summed square in these two clusters.

6.2 Node Measures

In order to study the relations between the nodes and their neighborhood and to analyze the influence ranking of the nodes on the graph, it is useful to call different centrality functions, also to check if they have coherent results:

- We used the **Eigenvector Centrality** in order to analyze the influence of the node and extract the most influent. To do this, we called the following networkx function:

  ```
  eigen_centrality = nx.eigenvector_centrality(G)
  ```

- The **Page Rank** is inspired to Eigenvector centrality, it extracts the same result in a different scale:

  ```
  page_rank = dict(nx.pagerank_numpy(G,weight = 'points'))
  ```

- The **Betweeness Centrality** is in general used to find the shortest path in a graph; however we are not dealing with this kind of problem (that is the reason why we do not post anything about it in the results section), but another aspect can be extrapolated: if a node has a low betweeness centrality, it means that it does not contain so much information about the graph, so we expect that the values with low centrality are the ones which are not so involved into the voting process and that have the last places in the rank:

  ```
  between = dict(nx.betweenness_centrality(G,weight='points'))
  ```

- The **Degree Centrality** is the centrality measure that consider the degree of the nodes; the degree centrality for a node is the fraction of nodes it is connected to, and it considers both in-coming and out-coming edges; however, since the countries have the same voting possibility, we expect that the real difference is about the in-coming edges; because of this, we expect that the nodes at the top of the rank have an higher degree centrality because they have more incoming edges.

```
degree_centrality = nx.degree_centrality(G)
```

  However, more important aspects about degree centrality will be explained in the results section.

- The **Hubs and Authorities Centrality** gives a more detailed information about the nodes, in which we extrapolate an hub and an authority score. A high hub actor points to many good authories and a high authority actor receives from many good hubs.

6.3 Results

6.3.1 Network Results

For this section, we decided to post the results computed on 2019 dataset for their expressiveness.

- For the study of the density we consider a sub density for each domain in which, as we said, we have the same number of countries. The density of each subgraph can tell us if we have split the graphs into two balanced subgraphs:

```
West Density:  0.24470
East Density:  0.2433
```

  From the previous result, we can assert that they are roughly perfectly balanced.

- The following histograms are the clusters extrapolated with the two different methods shown in the section 6.2. For each method we have to tune the treshold parameter in order to have handly results:

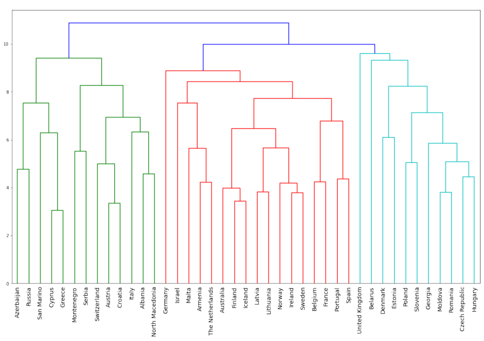


Figure 5: Histogram with Complete method and treshold = 9.7

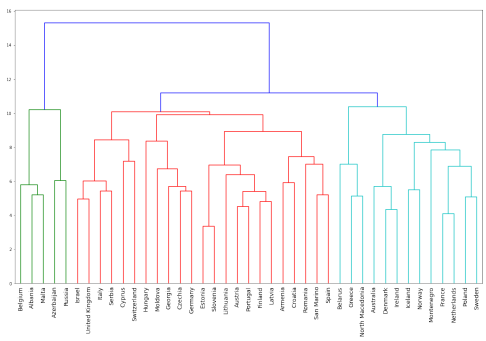


Figure 6: Histogram with Ward method and treshold = 11

Despite to the second method where we are not able to find significant relations from a geographical point of view, in the first we can notice that in the **green** cluster are grouped balcanic and mediterranean countries, in the **blue** we can group the north-east european countries, while in the **red** group there is the rest of the countries.

6.3.2 Nodes Results

The following results are extrapolated using the 2021 dataset. In particular, we have implemented a drawing function that builds a graph where the nodes are distributed in according to their geographical position (so basically, Italy is very near to San Marino for example). Here the implementation of the function, that takes as parameters the graph **G**, the geographical position **pos**, the utilized measure in **measures** and its name **measure_name**:

```
def draw(G, pos, measures, measure_name):

    nodes = nx.draw_networkx_nodes(G, pos, node_size=1250, cmap=plt.cm.plasma,
                                   node_color=list(measures.values()),
                                   nodelist=measures.keys())
    nodes.set_norm(mcolors.SymLogNorm(linthresh=0.01, linscale=1, base=10))
    labels = nx.draw_networkx_labels(G, pos)
    edges = nx.draw_networkx_edges(G, pos)

    plt.title(measure_name)
    plt.colorbar(nodes)
    plt.axis('off')
    plt.show()
```

The **pos** parameter can be swapped to Fruchterman positions, that is a Force-directed graph drawing algorithm.

- **Eigenvector Centrality and Page Rank**: Analyzing the plotted data of both tecniques we can observe that the nations with the more points recieved are the ones with the highest value indeed, so they are the ones with the best ranking. Both of the plots show the same kind of information, the only remarkable difference is that the magnitude of the values generated by the Page Rank are slightly lower then the ones generated by the Eigenvector centrality

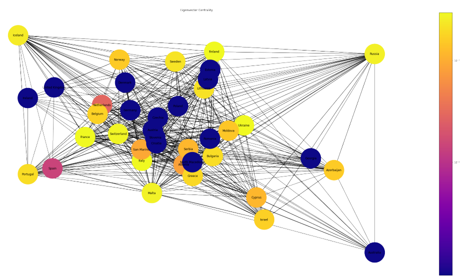


Figure 7: Eigenvector Plot Solution

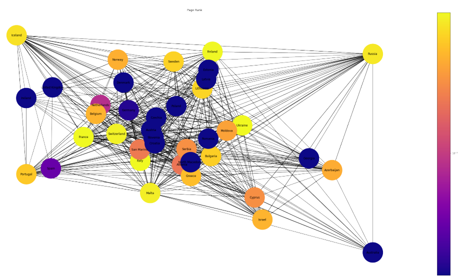


Figure 8: Page Rank Plot Solution

- **Degree Centrality**: In this kind of analisys we can retrive a lot of interesting information since we can measure how much the east side has voted the west and vice-versa. Regarding the entire graph, we can see that the amount of votes (in percentage) given from one side to the other (output degree) is perfectly balanced: both sides has given the same amount of points to the countries of the opposite side on average, remarking a fair behaviour by both factions, in detail:

```
From west to west:
50.14270556869522 %

From east to east:
48.92457538407479 %
```

  In order to have a general overview of the tournament, so considering also the non-challenging countries, we can do an analysis on the Input Degree. Here we can clearly see that the main disparity in the total amount of points is mainly given by the only-voting countries. We can see from the datas below that only the 30% of the total amount of recieved points is given by the same side, which means that there is a significant 20% of difference that depends on the only-voting countries, which means that in this edition they have been crucial in the final results.

```
From west to west:  31.285611871963688 %
From west to the rest of the partecipants:  68.71438812803632 %

Percentage of votes:
From east to east:  31.70816797034884 %
From east to the rest of the partecipants:  68.29183202965116  %
```

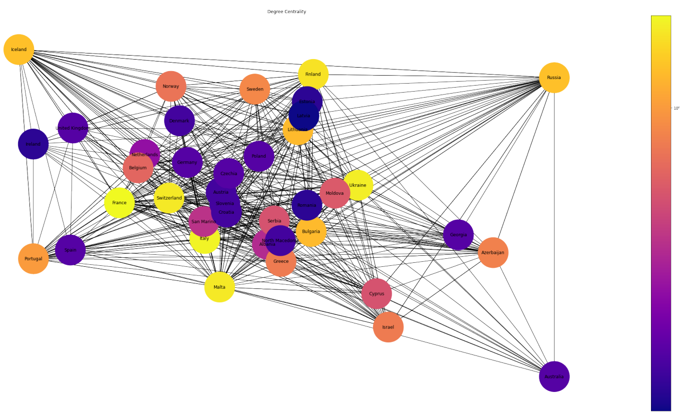
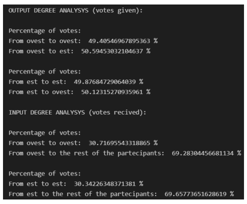


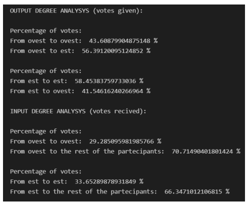
Figure 9: Degree Centrality Plot

Looking at the 2019,2018 and 2017 datasets we can see that this behaviour is common since the values are roughly the same also in the others editions as shown in the pictures below:



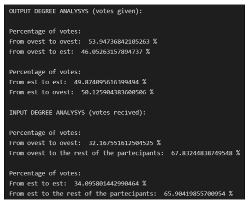
```
OUTPUT DEGREE ANALYSYS (votes given):

Percentage of votes:
From ovest to ovest:  49.40546967895363 %
From ovest to est:  50.59453032104637 %

Percentage of votes:
From est to est:  49.87684729064039 %
From est to ovest:  50.12315270935961 %

INPUT DEGREE ANALYSYS (votes recived):

Percentage of votes:
From ovest to ovest:  30.71695543318865 %
From ovest to the rest of the partecipants:  69.28304456681134 %

Percentage of votes:
From est to est:  30.34226348371381 %
From est to the rest of the partecipants:  69.65773651628619 %
```

Figure 10: Degree dataset 2017



```
OUTPUT DEGREE ANALYSYS (votes given):

Percentage of votes:
From ovest to ovest:  43.60879904875148 %
From ovest to est:  56.39120095124852 %

Percentage of votes:
From est to est:  58.45383759733036 %
From est to ovest:  41.54616240266964 %

INPUT DEGREE ANALYSYS (votes recived):

Percentage of votes:
From ovest to ovest:  29.285095981985766 %
From ovest to the rest of the partecipants:  70.71490401801424 %

Percentage of votes:
From est to est:  33.65289878931849 %
From est to the rest of the partecipants:  66.3471012106815 %
```

Figure 11: Degree dataset 2018



```
OUTPUT DEGREE ANALYSYS (votes given):

Percentage of votes:
From ovest to ovest:  53.94736842105263 %
From ovest to est:  46.05263157894737 %

Percentage of votes:
From est to est:  49.874095616399494 %
From est to ovest:  50.125904383600506 %

INPUT DEGREE ANALYSYS (votes recived):

Percentage of votes:
From ovest to ovest:  32.167551612504525 %
From ovest to the rest of the partecipants:  67.83244838749548 %

Percentage of votes:
From est to est:  34.095801442990464 %
From est to the rest of the partecipants:  65.90419855700954 %
```

Figure 12: Degree dataset 2019

- **Hubs and Authorities:** From the graphs below we can notice that when a nation receives a low amount of points, it receives also a low score in the Authority ranking. By the other side if we analyze the Hubs graph we can see that the value assigned to each country is affected mainly by the number of connections, i.e. the number of voted countries, so there is a significant baseline value, except for the Albania.

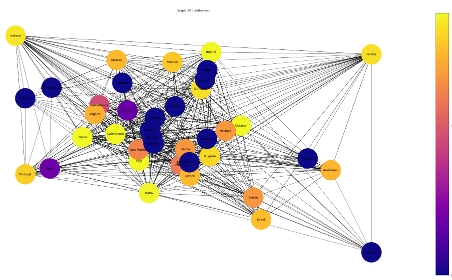


Figure 13: Authorities actors

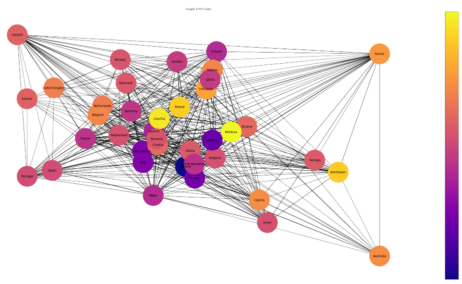


Figure 14: Hubs actors

6.3.3 Geo-political correlations

Some historical aspects can affect the results, so we did some research about interesting events that can explain some dynamics :

- In 2021 the main influencing aspect is for sure the pandemic, a lot of artists have been positive during the contest, and some of them have been forced to renounce to a live exibition, for example The Netherlands, with Duncan Laurence, was not be able to have the final performance in live; this could have negatively influenced its performance, considering that it takes 33 times the minimum points, and the 23th place in the final rank.

- In 2019 the contest took place in Israel, but this fact created some disappointments between Ireland, Iceland, Jerusalem and some others. Does this aspect influence the Israeli's results ? In order to answer to this question, we can check some details about Israel:
  - It gave 12 times the minimum points to the nations
  - It never took the maximum points
  - It took 18 times the minimum points
  - It was 17th in rank

- In 2018 edition during the performance of the UK , a man stopped the exibition stealing the microphone from the artist. This is an aspect that can explain why the UK took 29 times the minimum points, taking the 24th place in the rank.
- In 2017 the Portugal took a lot of approvals, taking 7 times the maximum points. During the analyzed years, it is the unique nation with such parameter greater than 2. This particular event could be explained also because during the festival , Salvador Sobral (the performer) launched a humanitarian message against the migration; maybe with this action he acquired more acceptance. On the other hand, the Ukraine registered 32 times of minimum points, with a very low points from Australia, maybe because during the Australian performance, a man showed his lower back to the public.

6.3.4 Critique

From a quantitative analysis we were able to detect some semantic aspects about the Eurovision contest, however they are based on suppositions, basing on some historical events.
The purpose of this analysis was to investigate not only about the relationship between the geographical position and the amount of points recieved by the neighbours nations, but also how exogenous events could affect the results. Regarding the first point from the Degree Centrality analisys we can derive that there is always a fair approach as we should expect in a competition like this.
Talking about the second one, in the section 6.3.3 we have noticed how much they are influent, instead.
As last consideration, some metrics were of course additional, since very similar to some others, such as Eigenvector Centrality and Page Rank, but this was done for the seek of completeness to check if the measure were done correctly and to see how the same kind of information is explained with different techniques.