Popular Erasmus destinations

Alessandro Lotta, Youssef Ben Khalifa

January 30, 2023

Contents

1 Dataset creation and pre-processing 2

2 Graph creation 2

3 Analysis performed 3

4 Analysis results and comparisons 4

5 Further analysis and improvements 7

6 Team Effort 7

7 References 8

1 Dataset creation and pre-processing

To manage the large amount of data we had to work with, we decided to make use of a custom created class called **Dataset**, in which we define all the methods and objects we used.

The dataset class makes use of the **pandas** library, which is a Python library for data manipulation and analysis, with which we perform the import, the preprocessing and the manipulation of the datasets.

Dataset creation

As mentioned in the other reports, the dataset was extracted from a list of .csv files (the source can be found in the references section), from which we selected all the data going from 2014 to 2019. The dataset is then loaded into a single Pandas Dataframe object, from which we will then further filter out and clear the data we do not need.

Dataset preprocessing

The first thing we need to do is to clear out the entries that are either incomplete or not useful for our analysis. To do that, we start by applying a simple filter on the dataset to select only the columns we are interested in, to then remove all the rows that have at least one missing value. This is done using the *cleanDataframe()* method contained in the dataset class.

From the resulting dataset we can start to filter out the type of entries we need for each of our analysis using the *applyFilter()* method: for our analysis we kept the entries associated to only the students that were still going through either a bachelor or a master degree in the respective university.

2 Graph creation

The entire project is based on the usage of the python library **NetworkX** which is a Python library for the creation, manipulation and study of the structure, dynamics, and functions of complex networks.

Originally, we meant to use a dedicated class called *CustomGraph*, which can be found in the *graph.py* file, to manage the graph and its entries, but this method revealed itself to be quite time consumnig and computationally heavy. So we then opted to adopt a dedicated function found in the NetworkX library that allowed us to create the graph with all its nodes and edges by simply feeding the method a dataframe object in which we specify, for each edge, the starting and end node, along with the weight of the edge.
The *weight* of an edge in our graphs is determined by the total amount of students that participated in the Erasmus program in which the Sending and Receiving country/university are respectively

the starting node and ending node associated with that edge. The graph creation was achieved using the following code:

```python
import networkx as nx
import pandas as pd
edges = pd.DataFrame({"source" : ds.df["Sending Country Code"],
                      "target" : ds.df["Receiving Country Code"],
                      "weight" : ds.df["Participants"]
                         })
edges = edges.groupby([’source’, ’target’]).sum().reset_index()
CountryGraph = nx.from_pandas_edgelist(edges, "source", "target", "weight", nx.DiGraph())
```

Listing 1: Country Graph creation

The same process was done to create both the graph containing the countries as nodes and the one with the universities.

At the end we end up with two graphs onto which wee can perform our analysis, each with the following dimensions:

Dimensions	**CountryGraph**	**UniGraph**
# Nodes	152	76816
# Edges	3420	185126

3 Analysis performed

In this section we will go over all the analysis we have performed on the graph we generated from the data we had. As we said in the project proposal we will use two graphs $C = (V_c, E_c)$ and $U = (V_u, E_u)$: one having countries as nodes and the other one having universities as nodes, in which the edges and their weights represent the amount of students moving/received from one node to the other.

PageRank coefficient

The **PageRank** coefficient of a node v expresses the ”importance” of a node in the graph, this is done by considering the number of incoming edges and the PageRank coefficient of the nodes that are connected to it. Analytically, the PageRank coefficient of a node v is defined as:

$$Pr(v) = (1 - d) + d \sum_{u \in V} \frac{Pr(u)}{deg(u)} \tag{3.1}$$

where d is the **damping factor** given by the user, V is the set of nodes in the graph and $deg(u)$ is the degree of node u.

This particular feature is very useful when we compute it on the graph in which we define as the set of nodes and edges, respectively the countries/universities and as edges both the students that are sent and received. We can find a measure of how important a country or a university can be in terms of students flow.

PageRank coefficient implementation

We implemented the PageRank algorithm using the **NetworkX** library, using the following method:

```
    _ranks = nx.pagerank(CountryGraph, weight='weight')

```

HITS

The Hyperlink-Induced Topic Search (HITS) is a link analysis algorithm that rates nodes based on two scores, a hub score and an authority score. The authority score estimates the importance of the node within the network. The hub score estimates the value of its relationships to other nodes. The GDS implementation is based on the Authoritative Sources in a Hyperlinked Environment publication by Jon M. Kleinberg.

HITS implementation

NetworkX also provides the method for computing the HIST score for a given graph:

```
    (hubs, authorities) = nx.hits(CountryGraph)

```

4 Analysis results and comparisons

Each analysis was done on two different machines in order to try and extrapolate a measure of the efficiency of the algorithms we used. However, the comparisons between the different time results on the algorithm execution are not very reliable, since the machines used are affected by many other factors other that the hardware itself.

The main focus of the analysis is on how the algorithms implementations we adopted perform w.r.t. the size of the graph we created.

To analyze and compare the results we decided to concentrate only on the Students that participate on the Erasums program while being subscribed to a Master Degree, to do this we applied a filter on the ”Education Level” column selecting the value ”ISCED-7”.

Computation time results and comparisons

For the test we had at our disposal two different machines:

1. Asus ROG laptop with an AMD Ryzen 9 5900HS CPU @ 3.00-¿4.6 GHz, Nvidia RTX 3060 (Mobile) Dedicated GPU, 16GB of RAM;

2. HP Pavilion x360 Convertible with Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz, Intel(R) UHD Graphics 620;

the results are computed separately for both graphs:

Algorithm	Asus ROG	HP Pavilion
PageRank	0.018549	0.049995
Hits	0.114350	0.030997

Table 1: Results for CountryGraph

Algorithm	Asus ROG	HP Pavilion
PageRank	2.088115	2.556013
Hits	2.626953	1.399125

Table 2: Results for UniGraph

Considering the difference in dimensions between the two graphs, the algorithm manages to obtained good results on the UniGraph compared to the CountryGraph. Again, the reliability of the results cannot bee guaranteed as there are to many factors that can affect the performance of the machines.

Countries Results

The results obtained are presented by showing the top countries based on the values obtained from PageRank and HITS.

Country	PageRank
France	5.719243
Italy	4.760962
Germany	4.507946
Tunisia	4.419083
Spain	4.150749

Country	Authorities
France	9.997428
Italy	5.675233
Germany	3.920258
Spain	1.839538
Poland	1.600657

Country	Hubs
Tunisia	11.644793
Vietnam	3.165710
Algeria	2.065967
Egypt	1.954555
Montenegro	1.025746

For the country graphs we also decided to implement the code to generate the geographical heatmaps by using the library 'geopandas', so that we can obtain more representative results. The geographical maps were created by only considering Europe and excluding the country Russia. An example of these is the following map obtained with the PageRank results

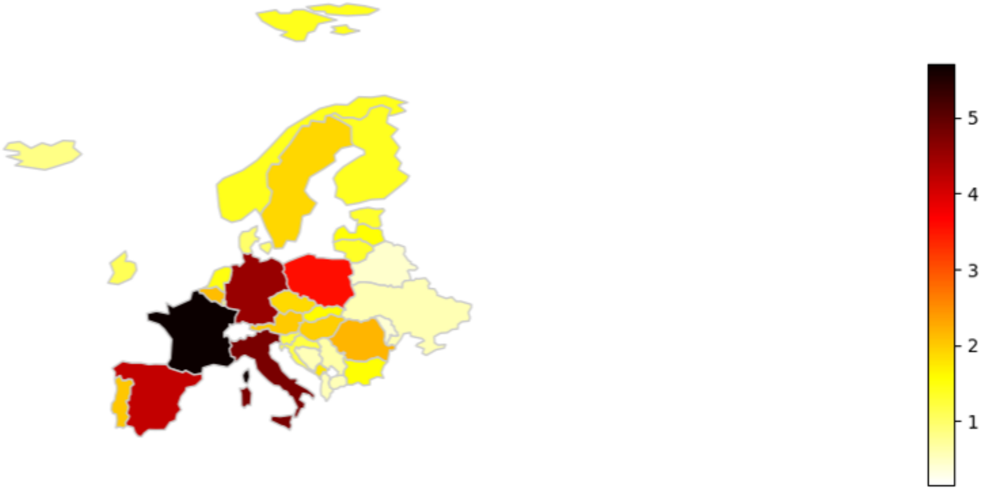

Figure 1: PageRank Heatmap

From the results we obtained it is possible to see that there are some differences between the two methods, more in detail in the top results for PageRank and Authorities values there are almost the same countries(France,Italy,Germany,Spain) instead the Hubs values are completely different.

Finally heare are the top 5 contries by centrality score:

Universities Results

In the case of the results obtained with the UniNodes graph, we can observe that the PageRank and Hits values have different rankings for top Universities, except for ”UNIVERSIDADE DE LISBOA” which is the most important for both PageRank and Authorities. From the table containing the Hubs scores it is clear that the most relevant Universities in Italy are also the ones that have students traveling to the universities that have high Authorities values. In particular ”UNIVERSITA DEGLI STUDI DI ROMA” and ”POLITECNICO DI MILANO” have also the top PageRank values.

Organization	**PageRank**
UNIVERSIDADE DE LISBOA	49.156703
Stichting ArtEZ	45.381767
UNIVERSITA DEGLI STUDI DI ROMA LA SAPIENZA	44.403827
Eesti Kunstiakadeemia	42.335885
POLITECNICO DI MILANO	41.404278

Organization	**Authorities**
UNIVERSIDADE DE LISBOA	8.715438e+01
NORGES TEKNISK-NATURVITENSKAPELIGE UNIVERSITET..	6.628848e+01
UNIVERSITAT POLITECNICA DE CATALUNYA	6.617643e+01
UNIVERSITAT POLITECNICA DE VALENCIA	6.022172e+01
KATHOLIEKE UNIVERSITEIT LEUVEN	5.544156e+01

Organization	**Hubs**
ALMA MATER STUDIORUM - UNIVERSITA DI BOLOGNA	9.454380e+01
UNIVERSITA DEGLI STUDI DI PADOVA	7.960611e+01
UNIVERSITA DEGLI STUDI DI ROMA LA SAPIENZA	7.632965e+01
POLITECNICO DI MILANO	7.479510e+01
UNIVERSIDADE DE LISBOA	6.127300e+01

Then as we did with the CountryGraph, we computed the centrality score for the universities graph:

5 Further analysis and improvements

1. Use of the **Random graphs** method to verify if the features we extracted actually give out interesting information;
2. Compute the same score on different types of graphs using different filters on the orinating dataset.
3. Try and obtain more recent data and repeat the analysis.

6 Team Effort

We are listed the contribution of each member of the team and the duration of each stage of the project:

- **Dataset Creation** : Youssef Ben Khalifa [2h], Alessandro Lotta [1h]
- **Dataset Preprocessing and Filtering** : Youssef Ben Khalifa [1.5h], Alessandro Lotta [1.5h]
- **Grah creation** : Youssef Ben Khalifa [3h], Alessandro Lotta [2h]
- **PageRank computation** : Youssef Ben Khalifa [1h], Alessandro Lotta [2h]
- **HITS computation** : Alessandro Lotta [1h]
- **Results visualization** : Youssef Ben Khalifa [1h], Alessandro Lotta [2.5h]

7 References

- The official portal for European data. *Erasmus mobility statistics 2014 - 2019.* `https://data.europa.eu/data/datasets/erasmus-mobility-statistics-2014-2019-v2?locale=en`

- NetworkX - Network Analysis in Python. *PageRank algorithm documentation.* `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html`

- NetworkX - Network Analysis in Python. *HITS algorithm documentation.* `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.hits_alg.hits.html`