Popular Erasmus destinations

Alessandro Lotta, Youssef Ben Khalifa

January 29, 2023

Contents

1 Dataset creation and pre-processing **2**

2 Graph creation **2**

3 Analysis performed **2**

4 Analysis results and comparisons **4**

5 Further analysis and improvements **4**

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 cleaning

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. From the resulting dataset we can start to filter out the type of entries we need for each of our analysis: for example if we want to perform our analysis only on the students that are currently studying in a university, we can filter out all the entries

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. through the usage of this library we were able to create the graphs we needed to perform our analysis.

Through the project m

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, which is a Python library for the creation, manipulation and study of the structure, dynamics, and functions of complex networks. The implementation of the algorithm can be found in the appendix.

Closeness centrality

The Closeness centrality of a node v is mathematically defined as:

$$C(v) = \frac{n - 1}{\sum_{u \in V} d(v, u)} \tag{3.2}$$

where V is the set of nodes in the graph and $d(v, u)$ is the length of the shortest path between nodes v and u. The Closeness centrality of a node v is yet another measure through which we can obtain useful information from our graphs. In particular, computing the Closeness centrality on the graph which is defined using

- as nodes the set of countries and/or universities;
- as edges the amount the students sent from the country/university to another country/university;

we can determine the "well" the students from that specific country/university are distributed over Europe.

Closeness centrality implementation

Once again, we used the **NetworkX** library to implement the Closeness centrality algorithm. The implementation can be found in the appendix.

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".

Countries Analysis

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

Country	**PageRank**
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2
Row 3, Column 1	Row 3, Column 2

Table 1: PageRank results

Country	**Authorities**
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2
Row 3, Column 1	Row 3, Column 2

Table 2: PageRank results

Country	**Hubs**
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2
Row 3, Column 1	Row 3, Column 2

Table 3: PageRank results

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 maps were created by only considering Europe and excluding the country Russia

Universities Analysis

5 Further analysis and improvements

1. Use of the **Random graphs** method to verify if the features we extracted actually give out interesting information;

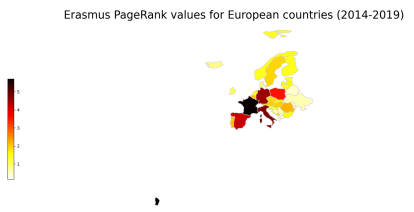
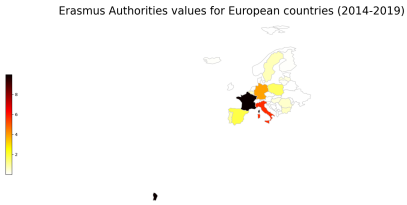
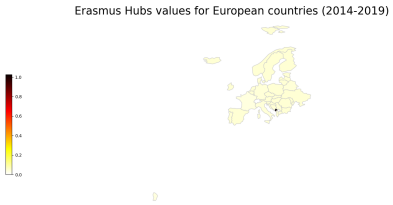


Figure 1: PageRank barplot



(a) Authorities barplot



(b) Hubs results

Figure 2: Overall barplot

2.