MusicMap

A strategic approach towards music communities

Final version

Friday, 22 April 2022

Prepared for:

ID boot camps

Content table

Contactos de IDBootcamps para cualquier cuestión acerca de este documento:

Daniel Roldán
IDBootcamps Student
ID Digital School Madrid

Ian Borrego Obrador
IDBootcamps Student
ID Digital School Madrid

1.

Story telling & Introduction

Story telling & Introduction

Music Connectivity

Many of us consider music as something essential in our lives, it motivates us, inspires creativity and fuels us towards anything we might be doing.

We can all agree that it activates a more human side of us.

In a changing world with emerging ideas and continuous specialization, the music industry isn't an exception, and casually we are very passionate about music and more specifically the chosen genre, Rap.

Could you imagine a way to relate the rap communities and create a network using Data Science? Well this Project is an approach for this idea.

Putting together everything we have learn in the Bootcamp and our passion for music we decided to create a set of tools that could be useful for the artists in the Rap genre for the spanish community.

Main Ideas

Exploration of a certain space

First of all in order to tackle this approach we decided to look into a specific space within the music industry, in this case spanish Rap Music. For the last years we have seen this specific part of the music industry grow exponentially as a rise of the urban/hip hop music influenced by the US.

One thing that characterizes this genre is the amount of collaborations in their songs and the lyricism used.

Graph community analysis

The project is separated in two parts, one of them is community analysis. This consists on the connections between the given Nodes (artists) and the Edges (collaborations) between each nodes which connects them.

Using the library NetworkX we can draw the network and separate it into communities for their further connectivity analysis.

Lyric analysis classification

The other part of the Project corresponds to the track lyrics of the chosen artists. Here we analyze the lexical richness of their tracks and using a non-supervised machine learning model called clustering, we are capable of creating different groups (also called 'clusters') and thereby be able to classify any new tracks that we include in this model.

Tool development & conclusions

Using the previous mentioned approaches we are able to draw a set of tools that the artists can use to boost their careers, backedup with real data and statistics.

2.

Project structure

Project Structure

Two way approach towards classification

Graph community analysis

Dataset

Using the Spotify API and Spotipy to gather the data and creating the dataset.

Graph creation & EDA

Using the library NetworkX we can calculate all the the data for the different artists and distribute them in communities

Tools & Conclusions

Drawing up conclusions for the chosen artist space and demonstrating the created tools

Lyric analysis classification

Dataset

Using the Spotify API, Spotipy & Spotiscience to gather the data and creating the dataset.

NLP & Dimension reduction

Chosing the adecuate words and data cleanup for the clusters. Dimension reduction to facilitate the Clustering.

Clustering & EDA

Non-supervised methodology such as kmeans & DBscan

Finally with the EDA gather up conclusions

3.

Graph community analysis

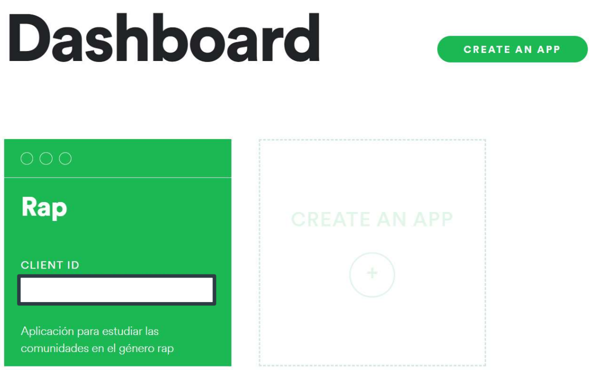
Graph community analysis

Dataset creation

In order to extract the desired information we decided to use the Spotify API which is free to use and granted us the information we wanted. In order to sign up to the API we need to have a Spotify account and suscribe in the following link:

Link: https://developer.spotify.com/dashboard/login

Once we validate our account we have to create an APP in which our purpose was academic:



Aplicación para estudiar las comunidades en el género rap

In order to connect to the API with your account you will need the following variables:

- Client ID
- Client Secret
- Redirect URL (recommended use: https://google.com/)

Dataset creation (cont.)

Besides the fact that the Spotify API is free, we decided to create this dataset with Spotify because Python has a library called Spotipy with a variety of functions that helped us greatly on the creation of the Dataset:

Link: https://spotipy.readthedocs.io/en/2.19.0/#

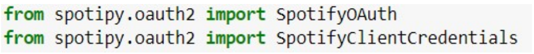
Client creation:

Note: This Project has been designed with Jupyter and Dataspell however the used coding language is Python.

In order to be able to call requests to the API from our Notebook we have to use the Client with the credentials attained from the Spotify API developer tool.

First of all we need to install spotipy using the command: `!pip install spotipy`

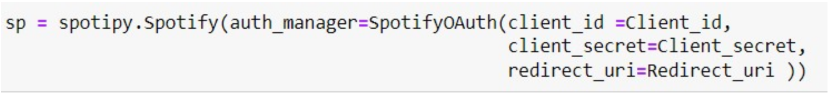
When the installation has been successfull we import the following in our Notebook:



```
from spotipy.oauth2 import SpotifyOAuth
from spotipy.oauth2 import SpotifyClientCredentials
```

To declare our client we can use either of those, the only difference is that with “SpotifyOAuth” you have to include the redirect URI. Another difference is that The Client Credentials flow is used in server-to-server authentication. Only endpoints that do not access user information can be accessed. The advantage here in comparison with requests to the Web API made without an access token, is that a higher rate limit is applied.

Here is the code to declare the Client variable:



```
sp = spotipy.Spotify(auth_manager=SpotifyOAuth(client_id =Client_id,
                                               client_secret=Client_secret,
                                               redirect_uri=Redirect_uri ))
```

Graph community analysis

Dataset creation (cont.)

Metrics:

Now that we can extract data using the Client and the API for the graph community analysis we wanted to create a network based on the collaborations amongst the artists. However in order to make this collaborations relevant to what is trending right now we decided to grab the collaborations from the their top tracks in the Spotify platform. In order to do this extraction Spotipy grants us three useful functions:

1

search(q, limit=10, offset=0, type='track', market=None)

- q: the search query (doc:https://developer.spotify.com/documentation/web-api/reference/search/)
- limit: the number of items to return (min = 1, default = 10, max = 50). The limit is applied
- type - the types of items to return. One or more of ‘artist’, ‘album’
- market - An ISO 3166-1 alpha-2 country code or the string

2

track(track_id, market=None)

- track_id - a spotify URI, URL or ID
- market - an ISO 3166-1 alpha-2 country code.

3

artist_top_tracks(artist_id, country='US’)

- artist_id - the artist ID, URI or URL
- country - limit the response to one particular country.

Dataset creation (cont.)

Other metrics that we decided to collect which Spotipy provided us are:

- Popularity
- Genre(s)
- Nº of Followers

Function:

4

artist(artist_id)

- artist_id - an artist ID, URI or URL
  - artist(artist_id)[‘popularity’]
  - artist(artist_id)[‘genres’]
  - artist(artist_id)[‘followers’]

With the use of these functions and an intensive research on how the API works we managed to create our own functions for the Dataset creation. We decided to create 2 datasets for this part of the Project:

- Collaboration Dataset (adjacency matrix & list)
- Metric Dataset (centrality measures, popularity, genres, followers & communities) – This Dataset is constructed further into the analysis and presented in the last part.

Graph community analysis

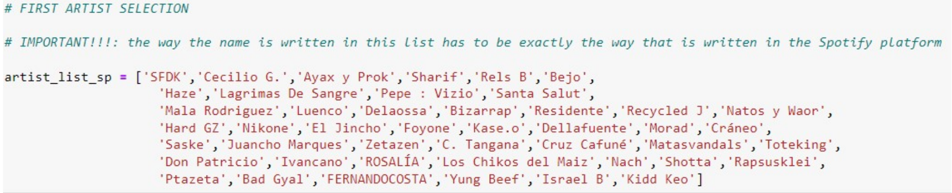
Dataset creation (cont.)

Collaboration dataset:

In order to create the first set of artists, we grab a representation of artists by choice from the community we want to analyse

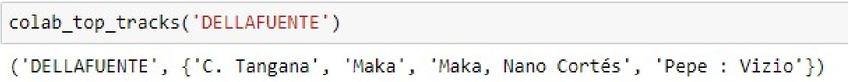
In this case we have chosen artists from the rap/urban music Spanish community. From this selection we use the API to search the collaborations which appear in their top tracks.

with the initial set and the name of the collaborators we grow again the initial list of artists as many times as we feel fit, until we have a community of which we consider representative for our analysis.



```
# FIRST ARTIST SELECTION

# IMPORTANT!!!: the way the name is written in this list has to be exactly the way that is written in the Spotify platform

artist_list_sp = ['SFDK','Cecilio G.','Ayax y Prok','Sharif','Rels B','Bejo',
                   'Haze','Lagrimas De Sangre','Pepe : Vizio','Santa Salut',
                   'Mala Rodriguez','Luenco','Delaossa','Bizarrap','Residente','Recycled J','Natos y Waor',
                   'Hard GZ','Nikone','El Jincho','Foyone','Kase.o','Dellafuente','Morad','Cráneo',
                   'Saske','Juancho Marques','Zetazen','C. Tangana','Cruz Cafuné','Matasvandals','Toteking',
                   'Don Patricio','Ivancano','ROSALÍA','Los Chikos del Maiz','Nach','Shotta','Rapsusklei',
                   'Ptazeta','Bad Gyal','FERNANDOCOSTA','Yung Beef','Israel B','Kidd Keo']
```

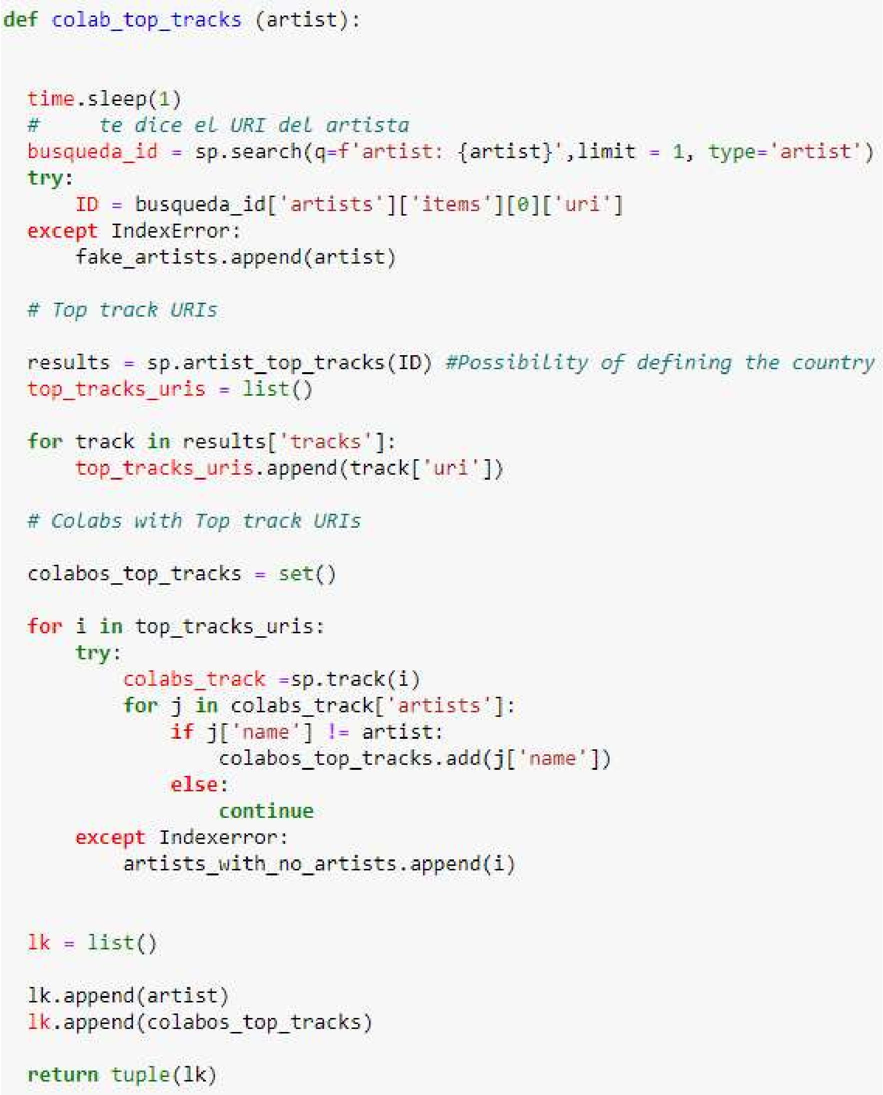
We now declare the function that will pick up all the collaborations from the top tracks of these artists. Depending on the name of the artist our function returns the following tuple:



```
colab_top_tracks('DELLAFUENTE')
```

```
('DELLAFUENTE', {'C. Tangana', 'Maka', 'Maka, Nano Cortés', 'Pepe : Vizio'})
```

Dataset creation (cont.)



```
def colab_top_tracks (artist):


  time.sleep(1)
  #     te dice el URI del artista
  busqueda_id = sp.search(q=f'artist: {artist}',limit = 1, type='artist')
  try:
      ID = busqueda_id['artists']['items'][0]['uri']
  except IndexError:
      fake_artists.append(artist)

  # Top track URIs

  results = sp.artist_top_tracks(ID) #Possibility of defining the country
  top_tracks_uris = list()

  for track in results['tracks']:
      top_tracks_uris.append(track['uri'])

  # Colabs with Top track URIs

  colabos_top_tracks = set()

  for i in top_tracks_uris:
      try:
          colabs_track =sp.track(i)
          for j in colabs_track['artists']:
              if j['name'] != artist:
                  colabos_top_tracks.add(j['name'])
              else:
                  continue
      except Indexerror:
          artists_with_no_artists.append(i)


  lk = list()

  lk.append(artist)
  lk.append(colabos_top_tracks)

  return tuple(lk)
```

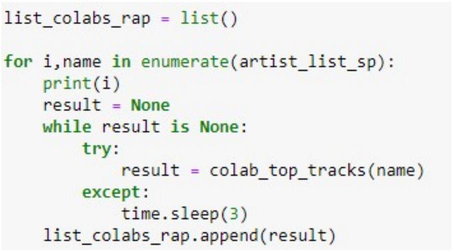
Graph community analysis

Dataset creation (cont.)

Collaboration dataset (cont.):

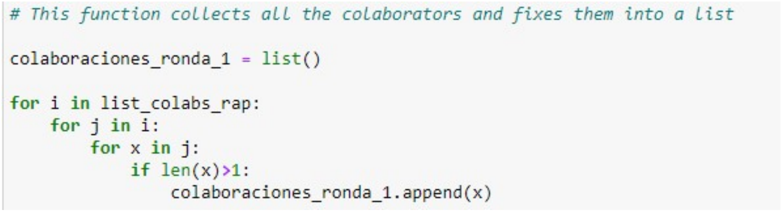
Now we run the initial list with our function however due to the Spotify API request limits we created a while loop in case an artist doesn't go through

Note: It is recommended to use more than one client in case the request volume is high as you might be temporarily banned. In our case we used 5 different accounts and sliced the requests



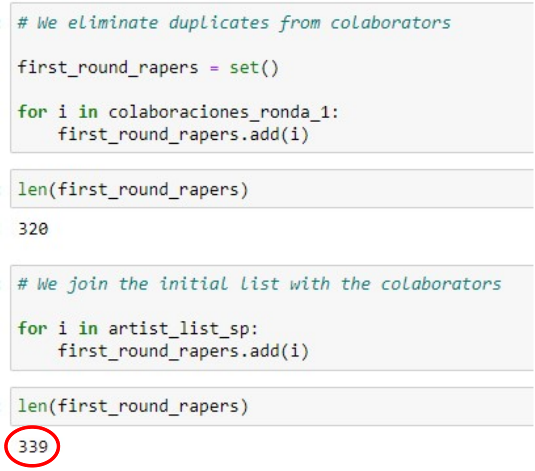
```
list_colabs_rap = list()

for i,name in enumerate(artist_list_sp):
    print(i)
    result = None
    while result is None:
        try:
            result = colab_top_tracks(name)
        except:
            time.sleep(3)
    list_colabs_rap.append(result)
```

Now we are going to re-do the initial list and include the collaborators thereby increasing the initial number of artists. This iteration can be done as many times as desired until you reach a representative group within the space you want to analyse. In our case we did one iteration as the hispanic rap community isn't that big.



```
# This function collects all the colaborators and fixes them into a list

colaboraciones_ronda_1 = list()

for i in list_colabs_rap:
    for j in i:
        for x in j:
            if len(x)>1:
                colaboraciones_ronda_1.append(x)
```

Dataset creation (cont.)



```
# We eliminate duplicates from colaborators

first_round_rapers = set()

for i in colaboraciones_ronda_1:
    first_round_rapers.add(i)
```

```
len(first_round_rapers)
```

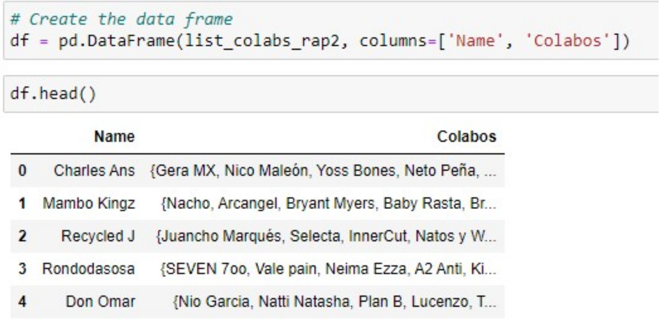
```
320
```

```
# We join the initial list with the colaborators

for i in artist_list_sp:
    first_round_rapers.add(i)
```

```
len(first_round_rapers)
```

```
339
```

We have increased the initial list from 45 to 339 artists in 1 iteration. Now we run the initial collaboration function with this new list and set up a DataFrame using Pandas:



```
# Create the data frame
df = pd.DataFrame(list_colabs_rap2, columns=['Name', 'Colabos'])
```

```
df.head()
```

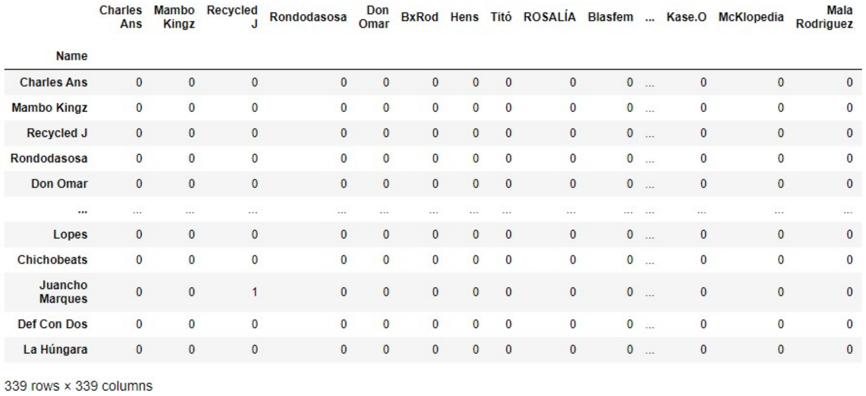
	Name	Colabos
0	Charles Ans	{Gera MX, Nico Maleón, Yoss Bones, Neto Peña, ...
1	Mambo Kingz	{Nacho, Arcangel, Bryant Myers, Baby Rasta, Br...
2	Recycled J	{Juancho Marqués, Selecta, InnerCut, Natos y W...
3	Rondodasosa	{SEVEN 7oo, Vale pain, Neima Ezza, A2 Anti, Ki...
4	Don Omar	{Nio Garcia, Natti Natasha, Plan B, Lucenzo, T...

Graph community analysis

Dataset creation (cont.)

Collaboration dataset (cont.):

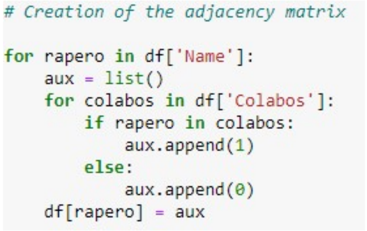
Once we have the Dataset we have to transform it into an adjacency matrix which has a quadratic form where the columns are a reflection of the index and each cell has a 1 if there is a collaboration and 0 if there isn't:



Name	Charles Ans	Mambo Kingz	Recycled J	Rondodasosa	Don Omar	BxRod	Hens	Tító	ROSALÍA	Blasfem	...	Kase.O	McKlopedia	Mala Rodriguez
Charles Ans	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Mambo Kingz	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Recycled J	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Rondodasosa	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Don Omar	0	0	0	0	0	0	0	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
Lopes	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Chichobeats	0	0	0	0	0	0	0	0	0	0	...	0	0	0
Juancho Marques	0	0	1	0	0	0	0	0	0	0	...	0	0	0
Def Con Dos	0	0	0	0	0	0	0	0	0	0	...	0	0	0
La Húngara	0	0	0	0	0	0	0	0	0	0	...	0	0	0

339 rows × 339 columns

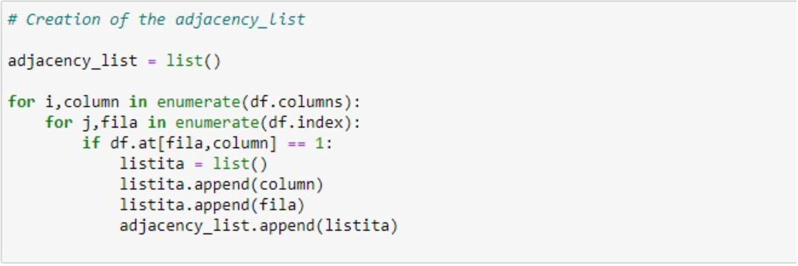
The code to transform the initial DataFrame to the adjacency matrix is the following:



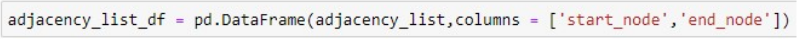
```
# Creation of the adjacency matrix

for rapero in df['Name']:
    aux = list()
    for colabos in df['Colabos']:
        if rapero in colabos:
            aux.append(1)
        else:
            aux.append(0)
    df[rapero] = aux
```

Following up, we have to convert this DataFrame into an adjacency list which is the required format that the library NetworkX and the software Gephi needs in order to analyse the data and attain the desired result. To convert it we use the following code:

Dataset creation (cont.)

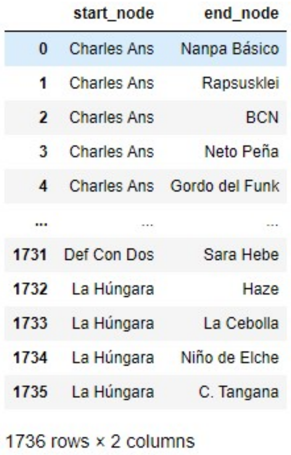


```
# Creation of the adjacency_list

adjacency_list = list()

for i,column in enumerate(df.columns):
    for j,fila in enumerate(df.index):
        if df.at[fila,column] == 1:
            listita = list()
            listita.append(column)
            listita.append(fila)
            adjacency_list.append(listita)
```



```
adjacency_list_df = pd.DataFrame(adjacency_list,columns = ['start_node','end_node'])
```

The DataFrame looks like this:



	start_node	end_node
0	Charles Ans	Nanpa Básico
1	Charles Ans	Rapsusklei
2	Charles Ans	BCN
3	Charles Ans	Neto Peña
4	Charles Ans	Gordo del Funk
...	...	...
1731	Def Con Dos	Sara Hebe
1732	La Húngara	Haze
1733	La Húngara	La Cebolla
1734	La Húngara	Niño de Elche
1735	La Húngara	C. Tangana

1736 rows × 2 columns

- Each node represents an artist.
- Each edge corresponds to a collaboration of 2 artists that belong to this network. In other words, two artists must collaborate in a song together in order for them to be connected in the particular network.

Graph community analysis

Graph creation & clean up

NetworkX & Gephi:

For this section we are using a Python library called NetworkX to attain the metrics and for the visualization we are using Gephi which is an independent software for this type of analysis.

- NetworkX: https://networkx.org/nx-guides/index.html
- Gephi: https://gephi.org/

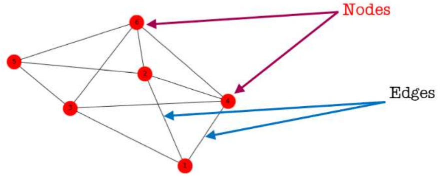
NetworkX is a repository which provides high-quality educational resources for learning about network analysis and graph theory. Examples include:

- Long-form narrative documentation, such as tutorials
- In-depth examinations of common graph and network algorithms and their implementations in NetworkX
- Demonstrations or domain-specific applications of NetworkX highlighting best-practices for network analysis.

Network analysis theory:

A network refers to a structure representing a group of objects/people and relationships between them. It is also known as a graph in mathematics. A network structure consists of nodes and edges. Here, nodes represent objects we are going to analyse while edges represent the relationships between those objects.

In our case, if we are studying a collaboration between artists, nodes are artists and edges are relationships which in this case represents if they have collaborated in a song.



Graph creation & clean up (cont.)

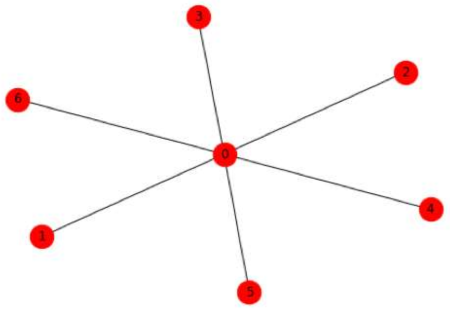
Network Analysis is useful in many living application tasks. It helps us in deep understanding the structure of a relationship in social networks, a structure or process of change in natural phenomenons, or even the analysis of biological systems of organisms.

Analyzing this network helps in:

- Identifying the most influent person/people in a group
- Defining characteristics of groups of users
- Prediction of suitable items for users
- Shortest routes between certain nodes

Who is the Important Person?

A crucial application of network analysis is identifying the important node in a network. This task is called Measuring Network Centrality. It can refer to the task of identifying the most influential member, or the representative of the group.



Graph community analysis

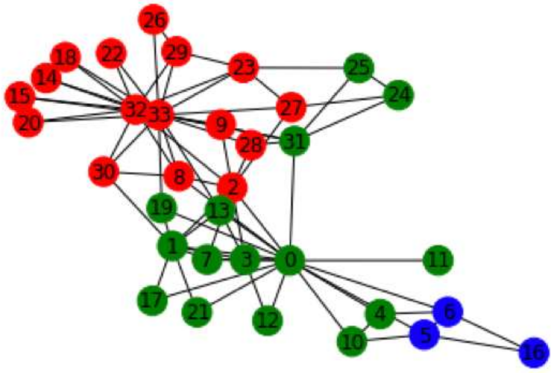
Graph creation & clean up (cont.)

Network analysis theory (cont.):

Communities

Another application of network analysis is the Community Detection task. This task purpose to divide a network into groups of nodes that are similar in any specific features.

Many researchers are working on algorithms to effectively solve community detection problems. Some well-known algorithms/methods in this task are Kernighan-Lin algorithms, Spectral Clustering, Label propagation, Modularity Optimization, etc.



What is else?

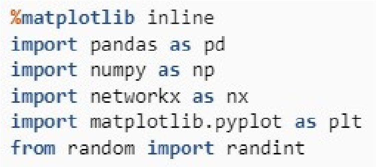
Besides these applications, network analysis also plays important role in time series analysis, natural language processing, telecommunication network analysis, etc. Recently, the technology of Machine Learning (Deep Learning) is also used in network analysis. In this case, research on Graph Embedding and Graph Neural Networks are interesting topics.

Source: https://towardsdatascience.com/network-analysis-d734cd7270f8

Graph creation & clean up (cont.)

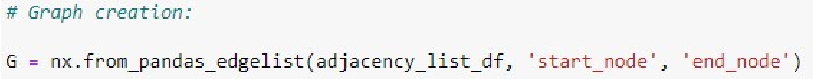
Now that we understand Network analysis we are going to start the data treatment process to extract the necessary metrics using NetworkX.

First of all the installation using: !pip install networkx and it's necessary imports:



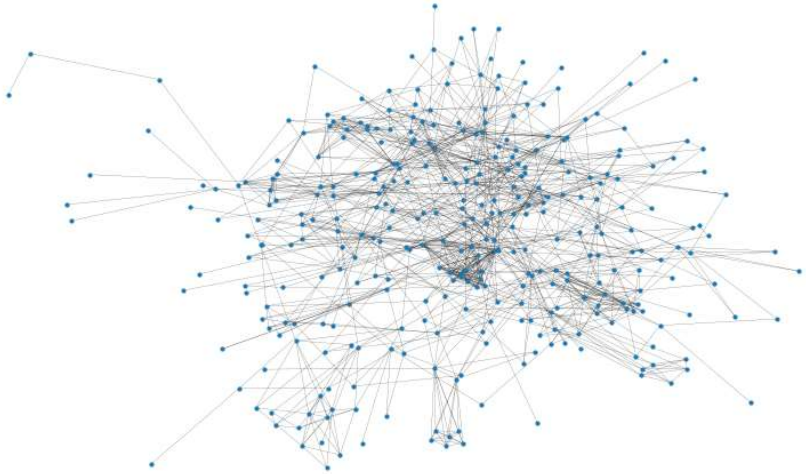
```
%matplotlib inline
import pandas as pd
import numpy as np
import networkx as nx
import matplotlib.pyplot as plt
from random import randint
```

Creation of the graph object using the adjacency list:



```
# Graph creation:

G = nx.from_pandas_edgelist(adjacency_list_df, 'start_node', 'end_node')
```

Visualizing the graph



Graph community analysis

Graph creation & clean up (cont.)

Besides these applications, network analysis also plays important role in time series analysis, natural language processing, telecommunication network analysis, etc. Recently, the technology of Machine Learning (Deep Learning) is also used in network analysis. In this case, research on Graph Embedding and Graph Neural Networks are interesting topics.

N°of nodes	339
N°of edges	1185
Avg edges per node	6.9

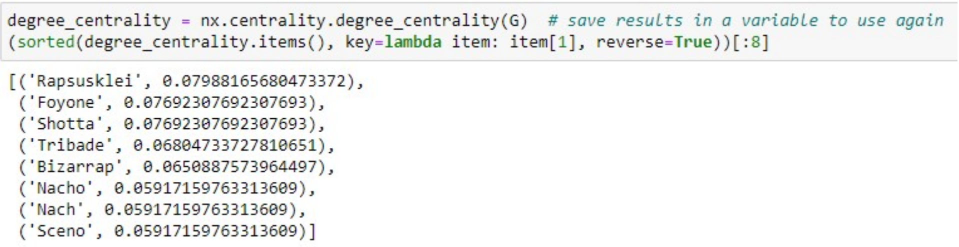
Centrality measures:

Degree Centrality

Degree centrality asigns an importance score based simply on the number of links held by each node. In this analysis, that means that the higher the degree centrality of a node is, the more edges are connected to the particular node and thus the more neighbor nodes (colaborations) this node has. In fact, the degree of centrality of a node is the fraction of nodes it is connected to. In other words, it is the percentage of the network that the particular node is connected to meaning colaborating with.

Starting, we find the nodes with the highest degree centralities. Specifically, the nodes with the 8 highest degree centralities are shown below together with the degree centrality:

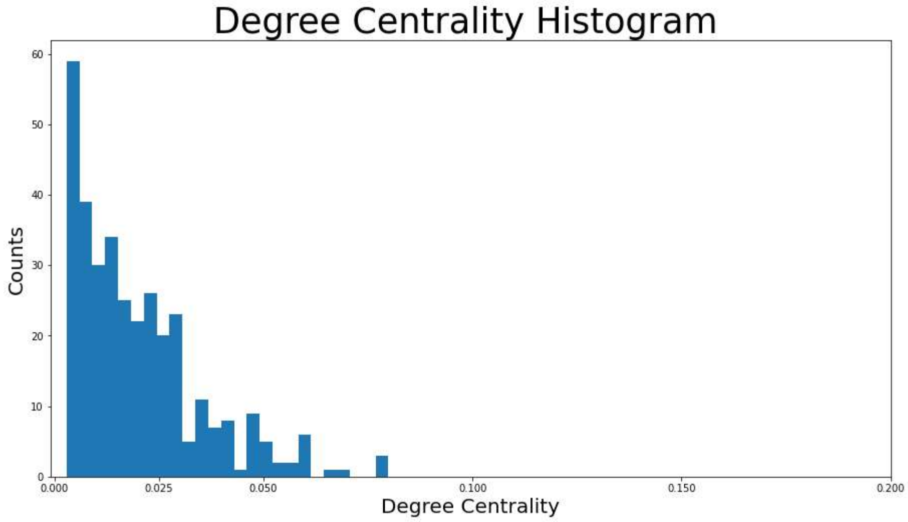
Graph creation & clean up (cont.)



```
degree_centrality = nx.centrality.degree_centrality(G)  # save results in a variable to use again
(sorted(degree_centrality.items(), key=lambda item: item[1], reverse=True))[:8]
```

```
[('Rapsusklei', 0.07988165680473372),
 ('Foyone', 0.07692307692307693),
 ('Shotta', 0.07692307692307693),
 ('Tribade', 0.0680473372781065l),
 ('Bizarrap', 0.0650887573964497),
 ('Nacho', 0.05917159763313609),
 ('Nach', 0.05917159763313609),
 ('Sceno', 0.05917159763313609)]
```

That means that node Rapsusklei has the highest degree centrality with 0.079, meaning that this artist has collaborations with around the 7.9% of the whole network. Similarly, nodes Foyone, Shotta, Tribade & Bizzarap also have very high degree centralities.



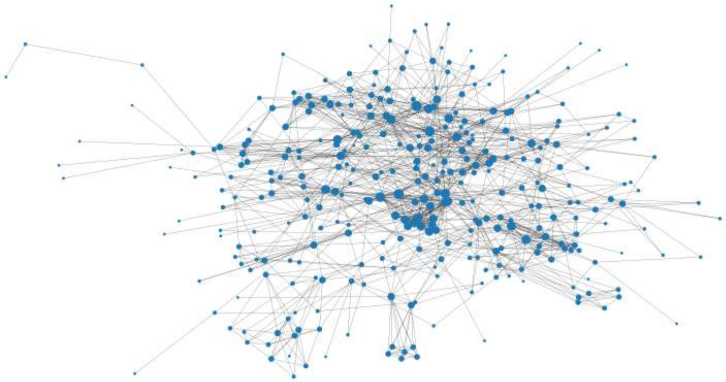
It is visible that the vast majority of artists have degree centralities of less than 0.03 (roughly). In fact the majority has less than 0.01. Actually, that makes sense because not all artists in this community which necessarily collaborate in their top tracks. Many are individual songs from a new album or they have been running a solo career. However it is noticeable that the most famous artists (higher views, followers, popularity... are those with most collaborators).

Graph community analysis

Graph creation & clean up (cont.)

Centrality measures (cont.):

Degree Centrality (cont.)

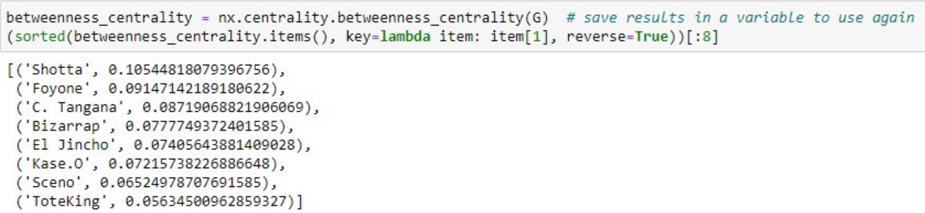


Betweenness Centrality

Betweenness centrality measures the number of times a node lies on the shortest path between other nodes, meaning it acts as a bridge. In detail, betweenness centrality of a node v is the percentage of all the shortest paths of any two nodes (apart from v), which pass through v. Specifically, in the rappers graph this measure is associated with the user's ability to collaborate with others. A user with a high betweenness centrality acts as a bridge to many users that are not collaborating and thus has the ability to influence them by conveying information or even connect them via the user's circle (which would reduce the user's betweenness centrality after).

Now, the nodes with the 8 highest betweenness centralities will be calculated and shown with their centrality values:

Graph creation & clean up (cont.)



```
betweenness_centrality = nx.centrality.betweenness_centrality(G)  # save results in a variable to use again
(sorted(betweenness_centrality.items(), key=lambda item: item[1], reverse=True))[:8]
```

```
[('Shotta', 0.10544818079396756),
 ('Foyone', 0.09147142189180622),
 ('C. Tangana', 0.08719068821906069),
 ('Bizarrap', 0.0777749372401585),
 ('El Jincho', 0.07405643881409028),
 ('Kase.O', 0.07215738226886648),
 ('Sceno', 0.06524978707691585),
 ('ToteKing', 0.05634500962859327)]
```

Looking at the results, the node Shotta has a betweenness centrality of 0.10, meaning it lies on 1/10 of the total shortest paths between other nodes. Also, combining the knowledge of the degree centrality.

Nodes Foyone, Shotta, Bizarrap & Sceno have both the highest degree and betweenness centralities and are **spotlight nodes**. That indicates that those nodes are both the most popular ones in this network and can also influence and spread information in the network.

Nodes El Jincho, ToteKing are not spotlight nodes, have some of the highest betweenness centralities and have not the highest degree centralities. That means that even though those nodes are not the most popular users in the network, they have the most influence in this network among artists of spotlight nodes when it comes to spreading information.

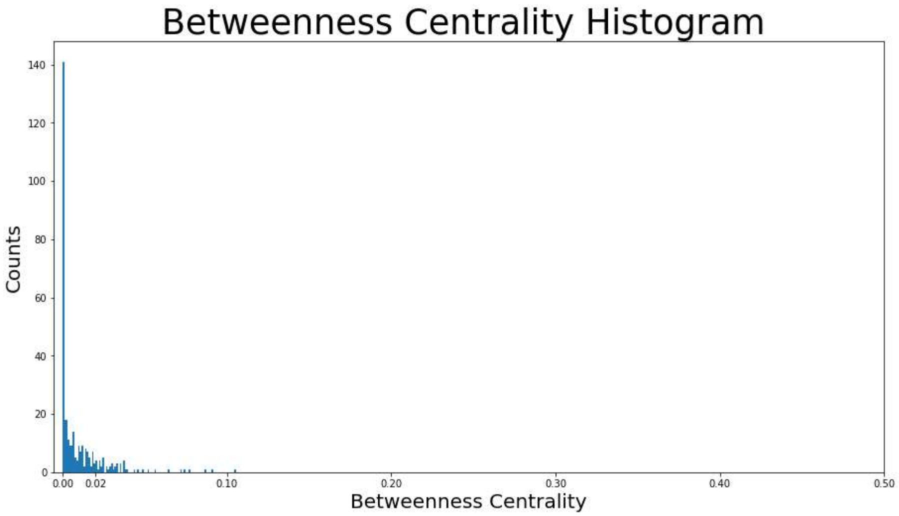
Node C. Tangana is a spotlight node as it has a very high betweenness centrality even though it doesn't have the highest degree centralities. In other words, this node does not have a very collaborative network. However, the user's whole list of collaborations is a part of the network and thus the user could connect different circles in this network by being the middleman.

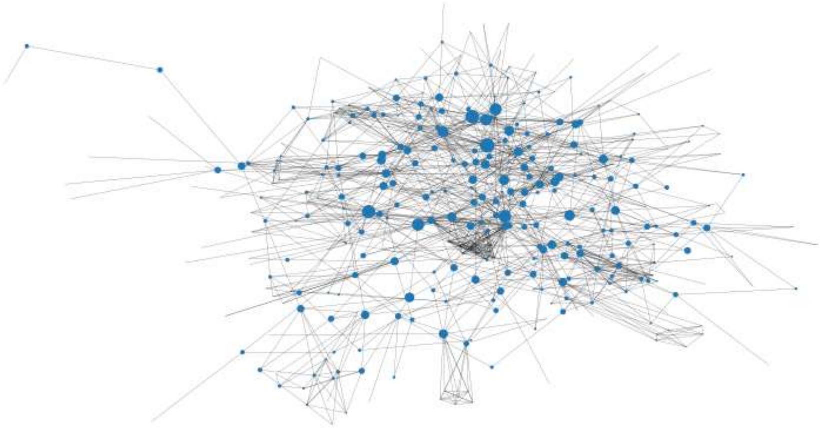
Graph community analysis

Graph creation & clean up (cont.)

Centrality measures (cont.):

Betweenness Centrality (cont.)





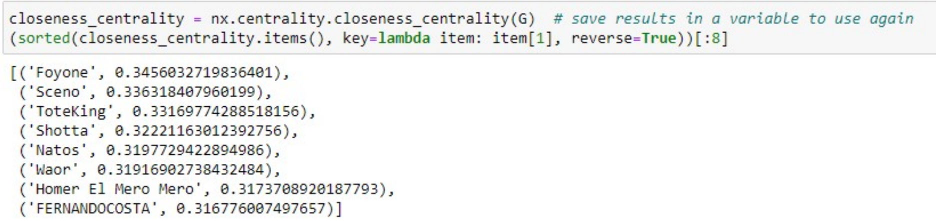
Graph creation & clean up (cont.)

Closeness Centrality

Closeness centrality scores each node based on their ‘closeness’ to all other nodes in the network. For a node v, its closeness centrality measures the average farness to all other nodes. In other words, the higher the closeness centrality of v, the closer it is located to the centre of the network.

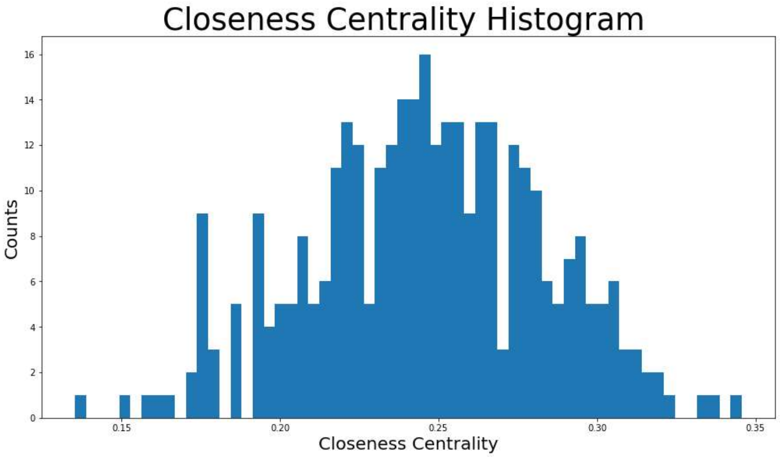
The closeness centrality measure is very important for marketing campaigns or beef between artists. Let’s examine the example of marketing campaigns. If a company wanted to start a marketing campaign with a community it should contact the artists with highest closeness centrality in order to spread it out within the network. However, if it was done in a user with very low closeness centrality then the spread of the marketing campaign won't reach as many people.

The nodes with the highest closeness centralities will be found now:



```
closeness_centrality = nx.centrality.closeness_centrality(G)  # save results in a variable to use again
(sorted(closeness_centrality.items(), key=lambda item: item[1], reverse=True))[:8]
```

```
[('Foyone', 0.3456032719836401),
 ('Sceno', 0.336318407960199),
 ('ToteKing', 0.33169774288518156),
 ('Shotta', 0.32221163012392756),
 ('Natos', 0.3197729422894986),
 ('Waor', 0.31916902738432484),
 ('Homer El Mero Mero', 0.3173708920187793),
 ('FERNANDOCOSTA', 0.316776007497657)]
```

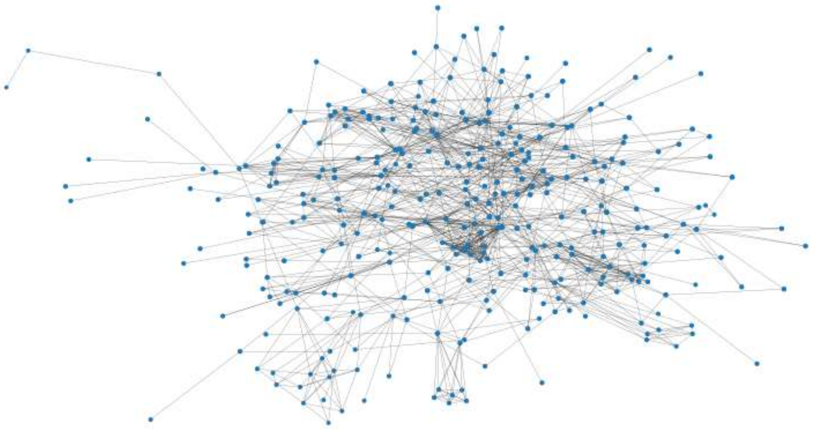


Graph community analysis

Graph creation & clean up (cont.)

Centrality measures (cont.):

Closeness Centrality (cont.)



Page Rank

Returns the PageRank of the nodes in the graph.

PageRank computes a ranking of the nodes in the graph G based on the structure of the incoming links. It was originally designed as an algorithm to rank web pages.

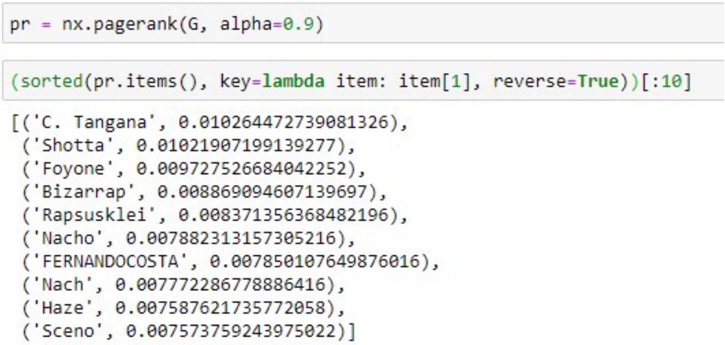
The PageRank algorithm was designed for directed graphs but this algorithm does not check if the input graph is directed and will execute on undirected graphs by converting each edge in the directed graph to two edges.

The PageRank was developed by the Google founders when they were thinking about how to measure the importance of webpages using the hyperlink network structure of the web. And the basic idea, is that PageRank will assign a score of importance to every single node. And the assumption that it makes, is that important nodes are those that have many in-links from important pages or important other nodes.

Graph creation & clean up (cont.)

The Page Rank can be also used on any type of network, for example, the web or social networks, but it really works better for networks that have directed edges. In fact, the important pages are those that have many in-links from more important pages.

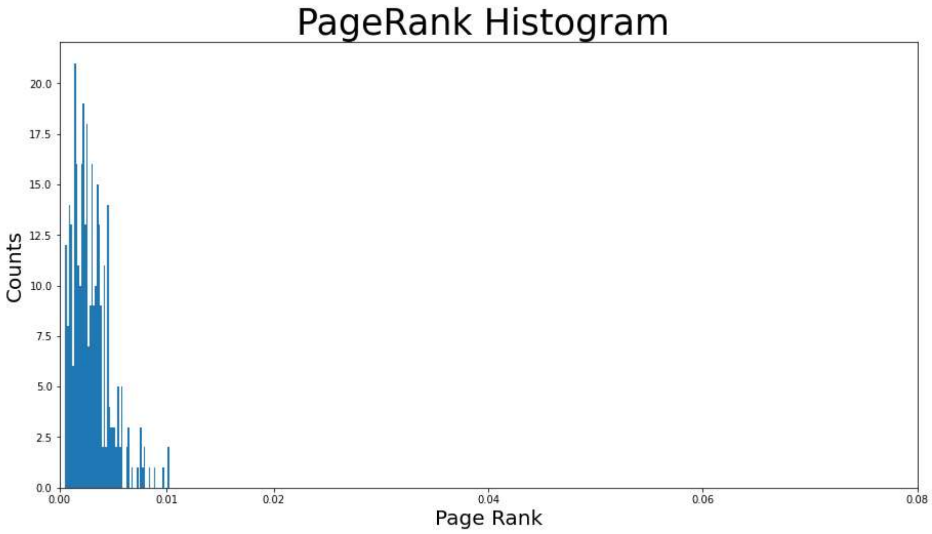
Source:https://www.andreaperlato.com/graphpost/page-rank-in-network-analysis/#:~:text=The%20page%20rank%20was%20developed,importance%20to%20every%20single%20node.



```
pr = nx.pagerank(G, alpha=0.9)
```

```
(sorted(pr.items(), key=lambda item: item[1], reverse=True))[:10]
```

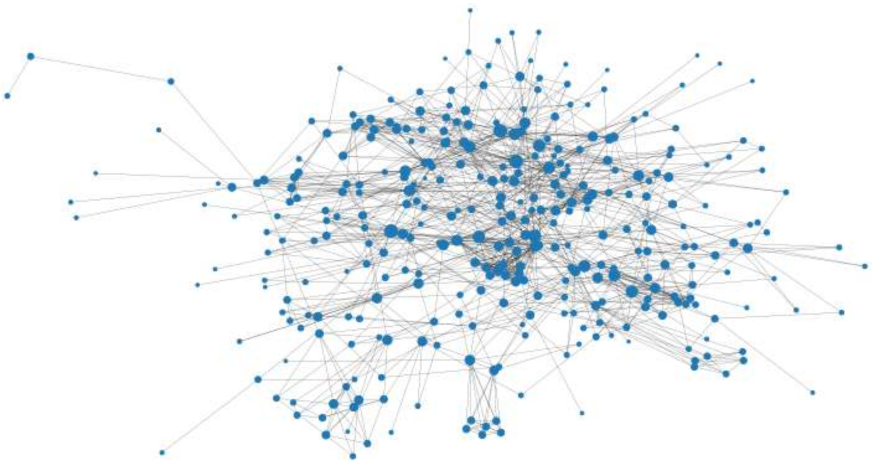
```
[('C. Tangana', 0.010264472739081326),
 ('Shotta', 0.01021907199139277),
 ('Foyone', 0.009727526684042252),
 ('Bizarrap', 0.008869094607139697),
 ('Rapsusklei', 0.008371356368482196),
 ('Nacho', 0.007882313157305216),
 ('FERNANDOCOSTA', 0.007850107649876016),
 ('Nach', 0.007772286778886416),
 ('Haze', 0.007587621735772058),
 ('Sceno', 0.007573759243975022)]
```



Graph community analysis

Graph creation & clean up (cont.)

Page Rank (cont.)



Network communities:

A community is a group of nodes, so that nodes inside the group are connected with many more edges than between groups. Two different algorithms will be used for communities detection in this network

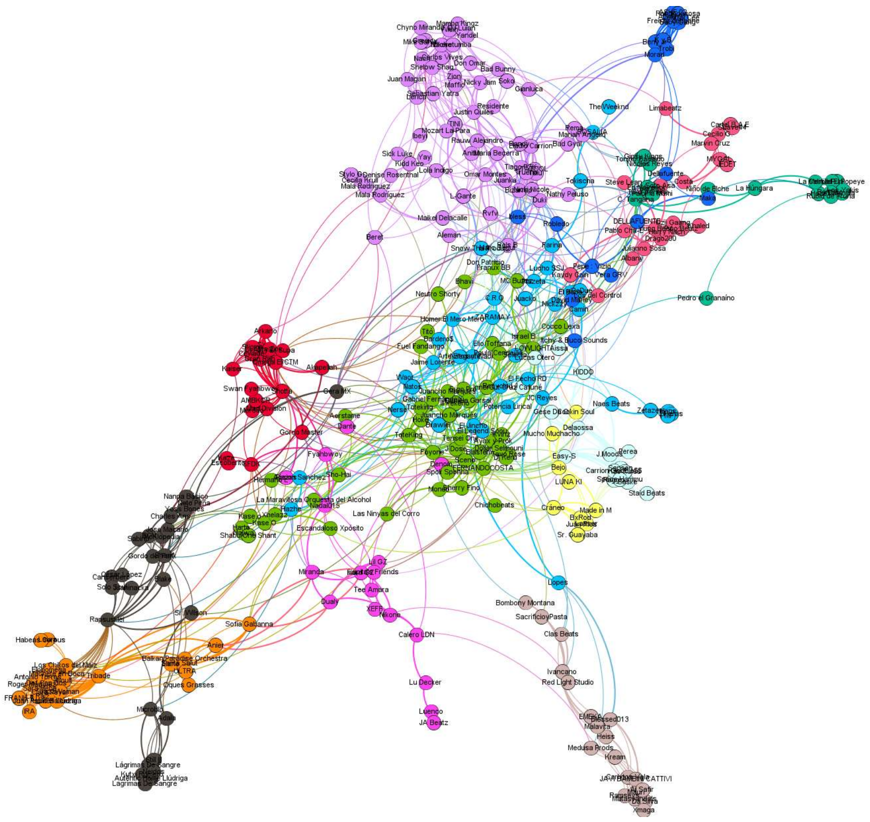
Firstly, a semi-synchronous label propagation method is used to detect the communities.

This function determines by itself the number of communities that will be detected. Now the communities will be iterated through and a colours list will be created to contain the same colour for nodes that belong to the same community. Also, the number of communities is printed:

Graph creation & clean up (cont.)

Network communities (cont.):

Using Gephi and the adjacency list we have used for NetworkX we create the network and separate it into communities (13):

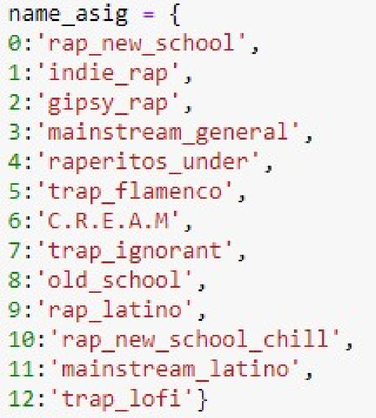


Graph community analysis

Graph creation & clean up (cont.)

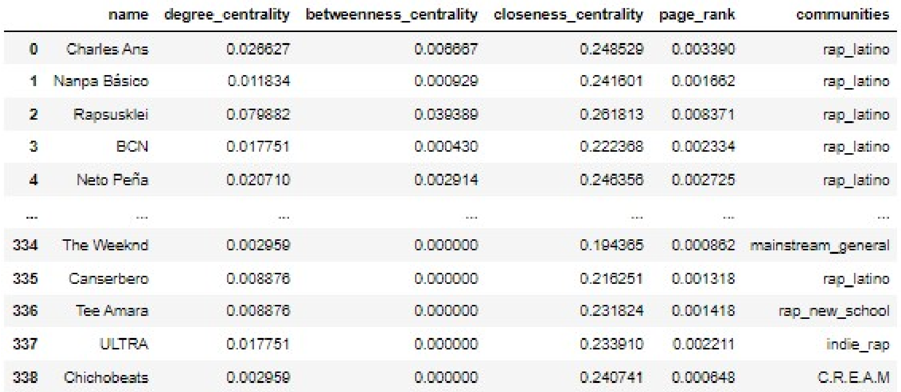
Network communities (cont.):

For each community we assigned a name which represents that group of artists:



```
name_asig = {
0:'rap_new_school',
1:'indie_rap',
2:'gipsy_rap',
3:'mainstream_general',
4:'raperitos_under',
5:'trap_flamenco',
6:'C.R.E.A.M',
7:'trap_ignorant',
8:'old_school',
9:'rap_latino',
10:'rap_new_school_chill',
11:'mainstream_latino',
12:'trap_lofi'}
```

Using the metrics we attained using NetworkX and the communities we got from Gephi we draw the last Dataset (still needs the popularity, followers & genres metrics which we will include):

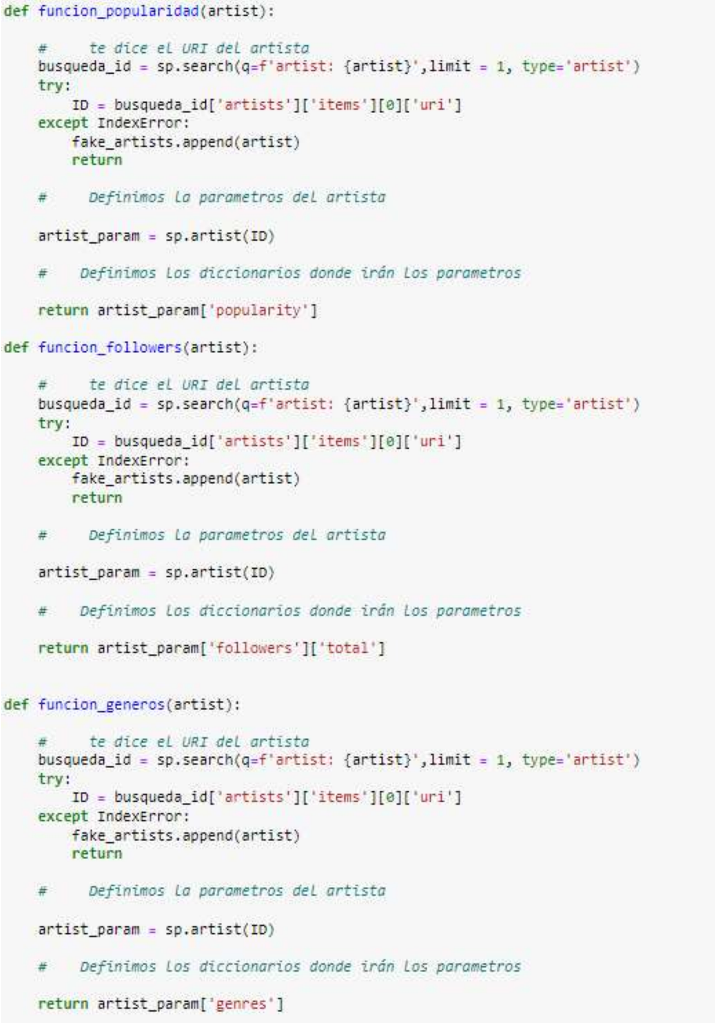


	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank	communities
0	Charles Ans	0.026627	0.006667	0.248529	0.003390	rap_latino
1	Nanpa Básico	0.011834	0.000929	0.241601	0.001662	rap_latino
2	Rapsusklei	0.079882	0.039389	0.261813	0.008371	rap_latino
3	BCN	0.017751	0.000430	0.222368	0.002334	rap_latino
4	Neto Peña	0.020710	0.002914	0.246356	0.002725	rap_latino
...	...	...	...	...	...	...
334	The Weeknd	0.002959	0.000000	0.194365	0.000862	mainstream_general
335	Canserbero	0.008876	0.000000	0.216251	0.001318	rap_latino
336	Tee Amara	0.008876	0.000000	0.231824	0.001418	rap_new_school
337	ULTRA	0.017751	0.000000	0.233910	0.002211	indie_rap
338	Chichobeats	0.002959	0.000000	0.240741	0.000648	C.R.E.A.M

Graph creation & clean up (cont.)

Network communities (cont.):

To this we include Popularity, Followers & Genres with the following functions we created:



```
def funcion_popularidad(artist):

    #    te dice el URI del artista
    busqueda_id = sp.search(q=f'artist: {artist}',limit = 1, type='artist')
    try:
        ID = busqueda_id['artists']['items'][0]['uri']
    except IndexError:
        fake_artists.append(artist)
        return

    #    Definimos la parametros del artista

    artist_param = sp.artist(ID)

    #   Definimos los diccionarios donde irán los parametros

    return artist_param['popularity']

def funcion_followers(artist):

    #    te dice el URI del artista
    busqueda_id = sp.search(q=f'artist: {artist}',limit = 1, type='artist')
    try:
        ID = busqueda_id['artists']['items'][0]['uri']
    except IndexError:
        fake_artists.append(artist)
        return

    #    Definimos la parametros del artista

    artist_param = sp.artist(ID)

    #   Definimos los diccionarios donde irán los parametros

    return artist_param['followers']['total']


def funcion_generos(artist):

    #    te dice el URI del artista
    busqueda_id = sp.search(q=f'artist: {artist}',limit = 1, type='artist')
    try:
        ID = busqueda_id['artists']['items'][0]['uri']
    except IndexError:
        fake_artists.append(artist)
        return

    #    Definimos la parametros del artista

    artist_param = sp.artist(ID)

    #   Definimos los diccionarios donde irán los parametros

    return artist_param['genres']
```

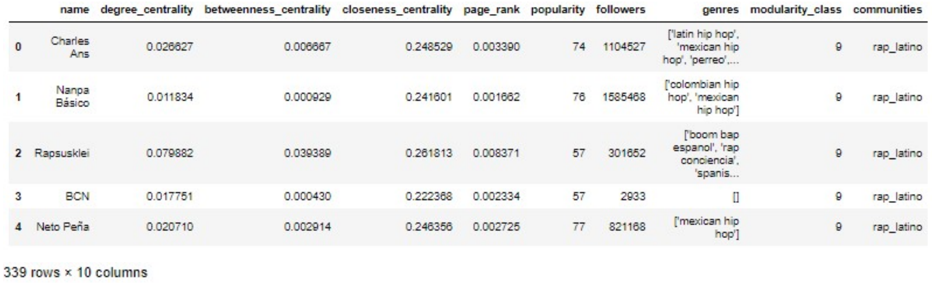
Graph community analysis

Graph creation & clean up (cont.)

Network communities (cont.):

Now using this DataFrame we will have all the desired metrics for a more complete analysis on the network.

Final metric dataset:



	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank	popularity	followers	genres	modularity_class	communities
0	Charles Ans	0.026627	0.006667	0.248529	0.003390	74	1104527	['latin hip hop', 'mexican hip hop', 'perreo',...	9	rap_latino
1	Nanpa Básico	0.011834	0.000929	0.241601	0.001662	76	1585468	['colombian hip hop', 'mexican hip hop']	9	rap_latino
2	Rapsusklei	0.079882	0.039389	0.261813	0.008371	57	301652	['boom bap espanol', 'rap conciencia', 'spanis...	9	rap_latino
3	BCN	0.017751	0.000430	0.222368	0.002334	57	2933	[]	9	rap_latino
4	Neto Peña	0.020710	0.002914	0.246356	0.002725	77	821168	['mexican hip hop']	9	rap_latino

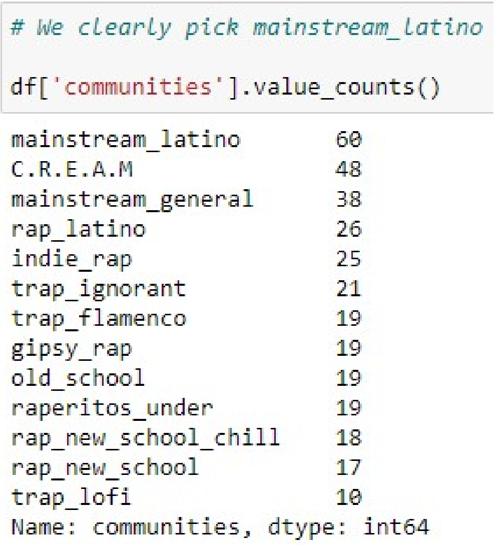
339 rows × 10 columns

Graph community analysis

Tools & conclusions

First Tool - General metrics for a certain community:

First tool corresponds to an analysis of the network. As we have already done an analysis the network as a whole, we are now going to analyse a representative community.



```
# We clearly pick mainstream_latino

df['communities'].value_counts()
```

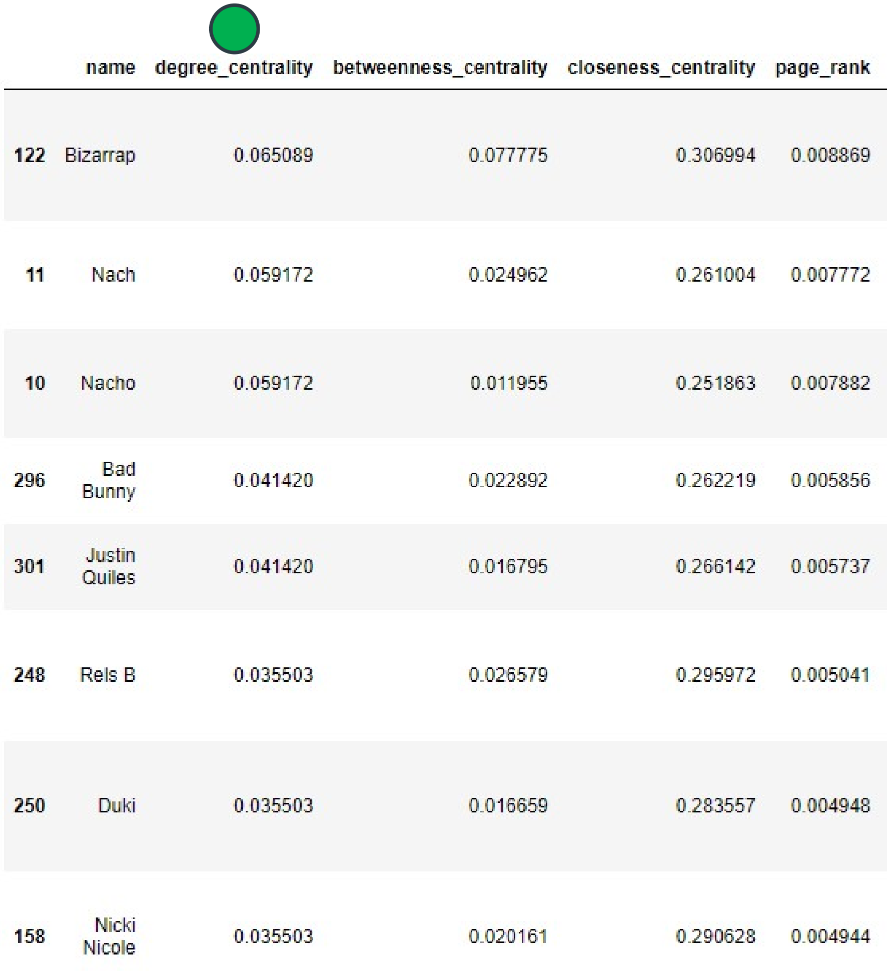
```
mainstream_latino       60
C.R.E.A.M               48
mainstream_general      38
rap_latino              26
indie_rap               25
trap_ignorant           21
trap_flamenco           19
gipsy_rap               19
old_school              19
raperitos_under         19
rap_new_school_chill    18
rap_new_school          17
trap_lofi               10
Name: communities, dtype: int64
```

Tools & conclusions(cont.)

First Tool - General metrics for a certain community (cont.):

Centrality metrics:

Degree_centrality top 8



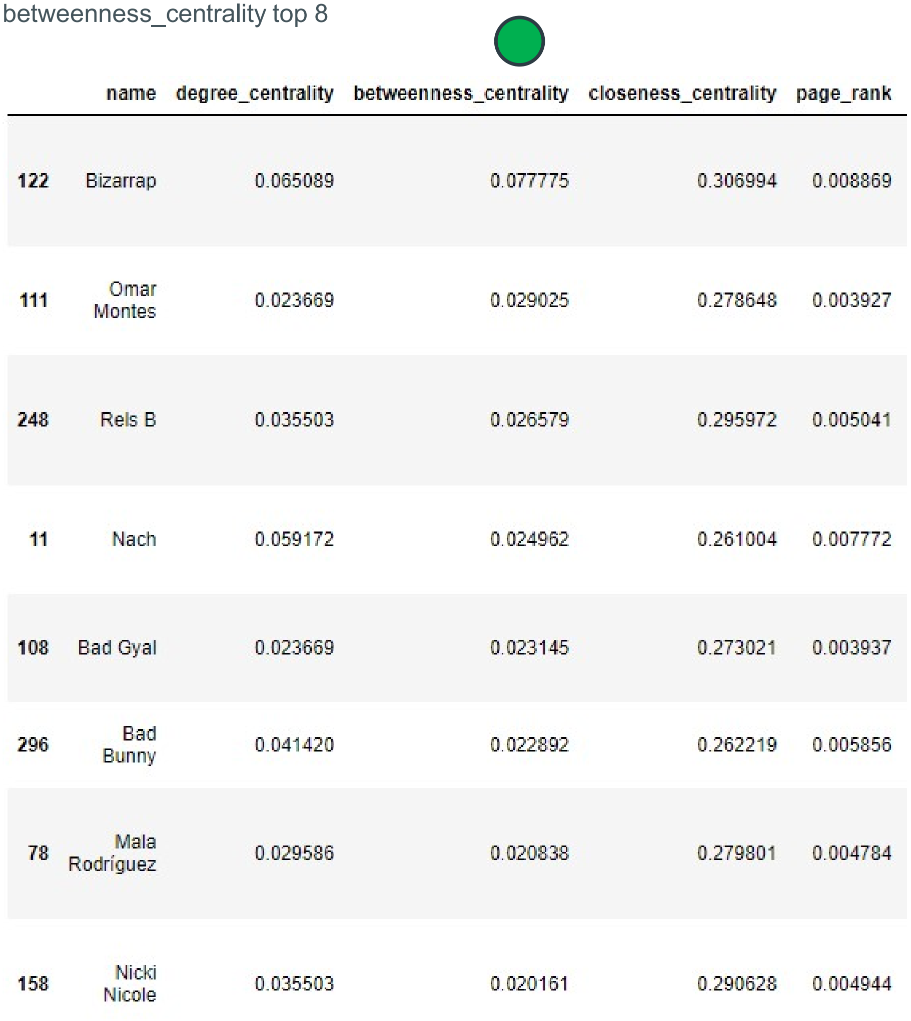
	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank
122	Bizarrap	0.065089	0.077775	0.306994	0.008869
11	Nach	0.059172	0.024962	0.261004	0.007772
10	Nacho	0.059172	0.011955	0.251863	0.007882
296	Bad Bunny	0.041420	0.022892	0.262219	0.005856
301	Justin Quiles	0.041420	0.016795	0.266142	0.005737
248	Rels B	0.035503	0.026579	0.295972	0.005041
250	Duki	0.035503	0.016659	0.283557	0.004948
158	Nicki Nicole	0.035503	0.020161	0.290628	0.004944

Graph community analysis

Tools & conclusions

First Tool - General metrics for a certain community (cont.):

betweenness_centrality top 8

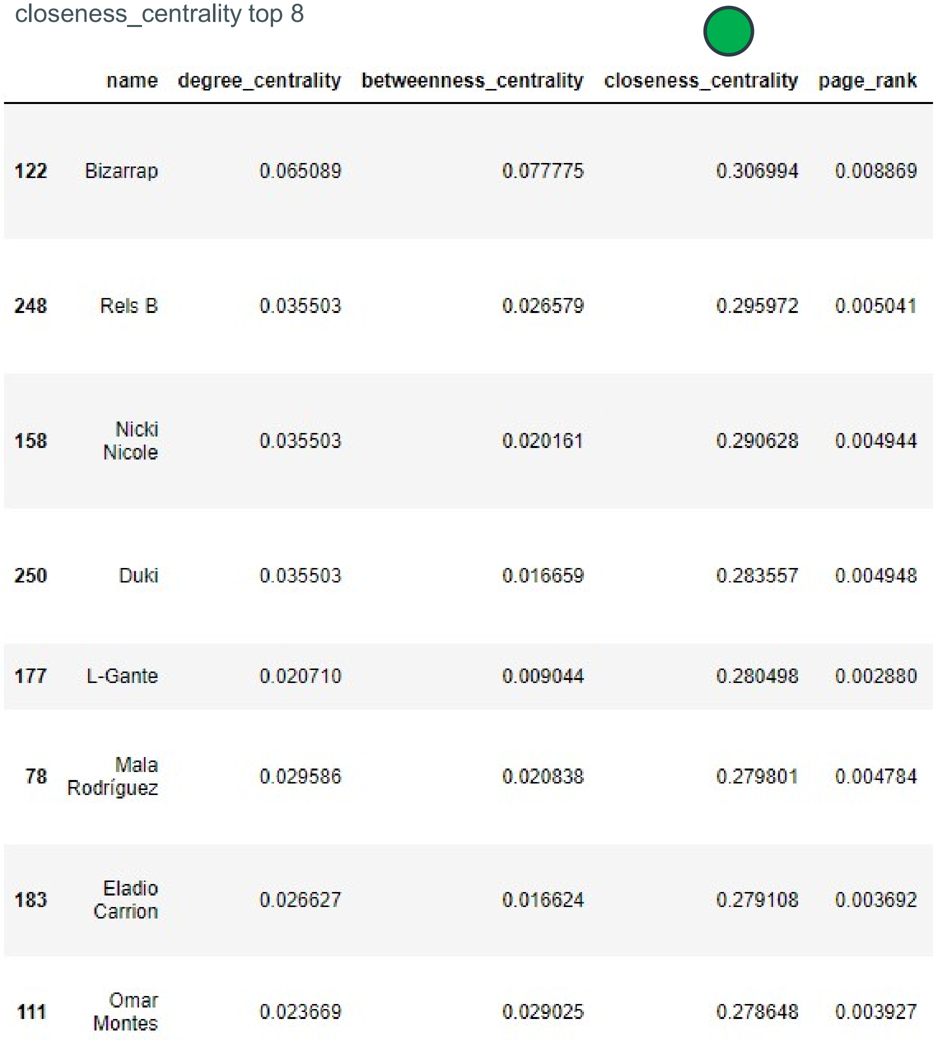


	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank
122	Bizarrap	0.065089	0.077775	0.306994	0.008869
111	Omar Montes	0.023669	0.029025	0.278648	0.003927
248	Rels B	0.035503	0.026579	0.295972	0.005041
11	Nach	0.059172	0.024962	0.261004	0.007772
108	Bad Gyal	0.023669	0.023145	0.273021	0.003937
296	Bad Bunny	0.041420	0.022892	0.262219	0.005856
78	Mala Rodríguez	0.029586	0.020838	0.279801	0.004784
158	Nicki Nicole	0.035503	0.020161	0.290628	0.004944

Tools & conclusions(cont.)

First Tool - General metrics for a certain community (cont.):

closeness_centrality top 8



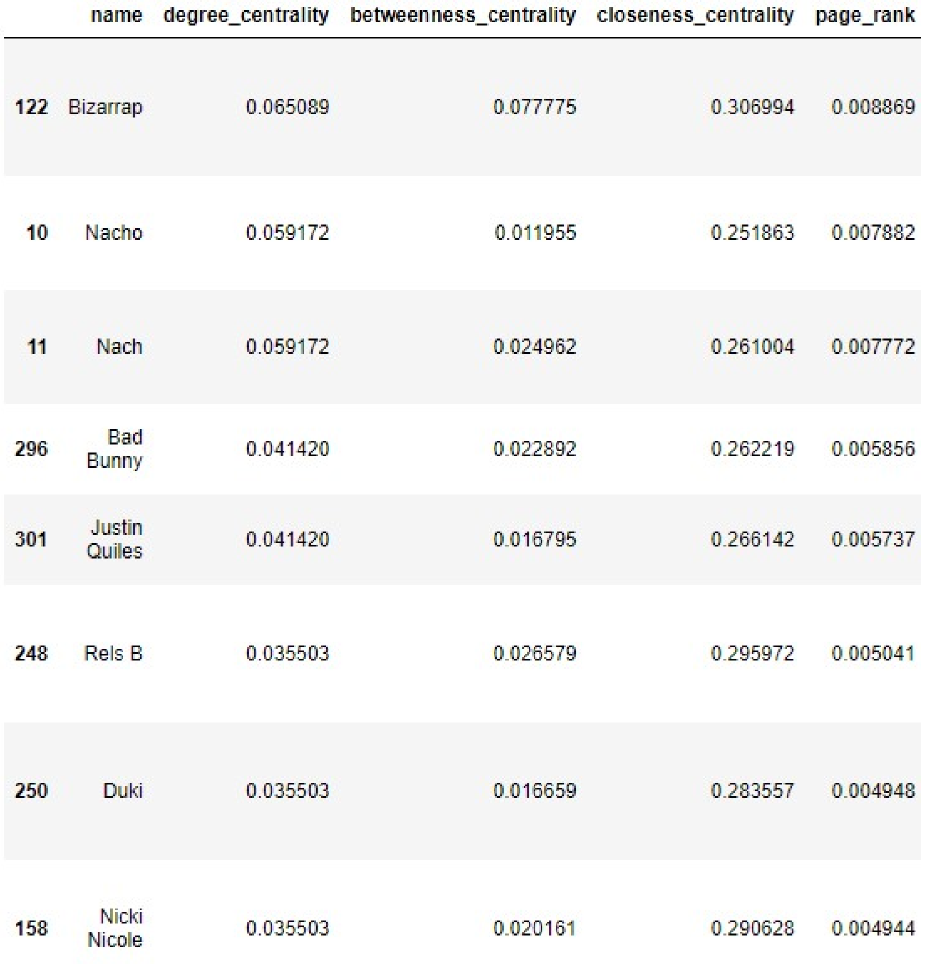
	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank
122	Bizarrap	0.065089	0.077775	0.306994	0.008869
248	Rels B	0.035503	0.026579	0.295972	0.005041
158	Nicki Nicole	0.035503	0.020161	0.290628	0.004944
250	Duki	0.035503	0.016659	0.283557	0.004948
177	L-Gante	0.020710	0.009044	0.280498	0.002880
78	Mala Rodríguez	0.029586	0.020838	0.279801	0.004784
183	Eladio Carrion	0.026627	0.016624	0.279108	0.003692
111	Omar Montes	0.023669	0.029025	0.278648	0.003927

Graph community analysis

Tools & conclusions

First Tool - General metrics for a certain community (cont.):

PageRank top 8



	name	degree_centrality	betweenness_centrality	closeness_centrality	page_rank
122	Bizarrap	0.065089	0.077775	0.306994	0.008869
10	Nacho	0.059172	0.011955	0.251863	0.007882
11	Nach	0.059172	0.024962	0.261004	0.007772
296	Bad Bunny	0.041420	0.022892	0.262219	0.005856
301	Justin Quiles	0.041420	0.016795	0.266142	0.005737
248	Rels B	0.035503	0.026579	0.295972	0.005041
250	Duki	0.035503	0.016659	0.283557	0.004948
158	Nicki Nicole	0.035503	0.020161	0.290628	0.004944

Tools & conclusions(cont.)

First Tool - General metrics for a certain community (cont.):

Conclusions:

We can observe clearly that the spotlight nodes are:

- Bizarrap
- Nacho
- Rels B
- Bad Bunny

This makes sense as Bizarrap is a DJ/Music producer and right now has contact with all of this artists, so collaborating with him has great repercussion. It has been seen when one of his las collaborations with the artist Residente received over 60 Million views in 3 days.

Regarding the other spotlight nodes, Nacho & Bad bunny are Reguetton artists with many collaborations with other nodes that are in most of our top 8 like Duki and are also nodes that have a high response towards the network.

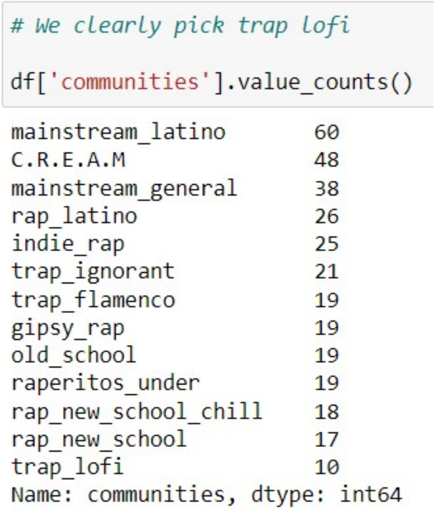
In this analysis however, Rels B is a spotlight node which caught my interest as they are Spanish artists and have moved over from Spain more towards the latino community, with their style and connections within that community. They are a great example and connection from those other artists that would want to move into the latino_mainstream community!

Graph community analysis

Tools & conclusions

Second Tool – Shortest route from one node to another:

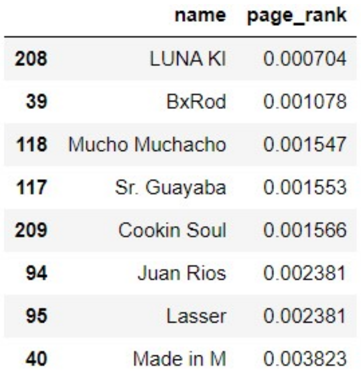
Following our example we will grab an artist from a low populated community and make a route to the latino_mainstream:



```
# We clearly pick trap lofi

df['communities'].value_counts()
```

```
mainstream_latino        60
C.R.E.A.M                48
mainstream_general       38
rap_latino               26
indie_rap                25
trap_ignorant            21
trap_flamenco            19
gipsy_rap                19
old_school               19
raperitos_under          19
rap_new_school_chill     18
rap_new_school           17
trap_lofi                10
Name: communities, dtype: int64
```

Now with the PageRank metric we can see who is at the bottom of this community and try to help this artist reach the latino_mainstream community.

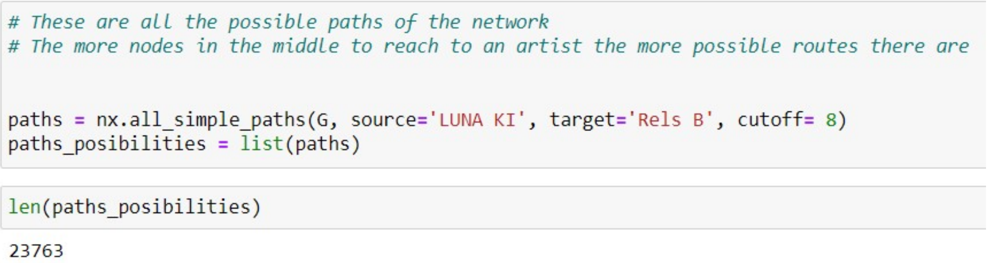


	name	page_rank
208	LUNA KI	0.000704
39	BxRod	0.001078
118	Mucho Muchacho	0.001547
117	Sr. Guayaba	0.001553
209	Cookin Soul	0.001566
94	Juan Rios	0.002381
95	Lasser	0.002381
40	Made in M	0.003823

Tools & conclusions(cont.)

Second Tool – Shortest route from one node to another (cont.):

We can observe that LUNA KI could be a potential artist that we could try to contruct a route to the latino_mainstream community. In this case we could find a shortest route towards Rels B.

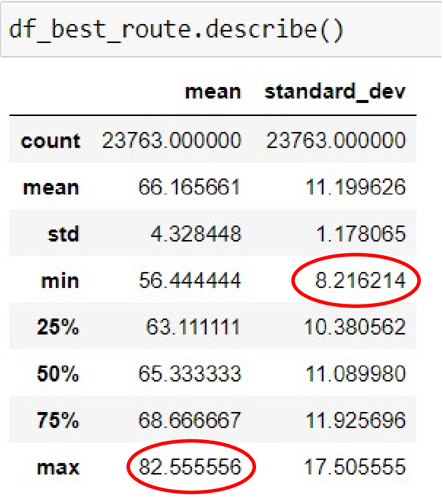


```
# These are all the possible paths of the network
# The more nodes in the middle to reach to an artist the more possible routes there are


paths = nx.all_simple_paths(G, source='LUNA KI', target='Rels B', cutoff= 8)
paths_posibilities = list(paths)
```

```
len(paths_posibilities)
```

```
23763
```

In order to reduce the number of paths we will work with the highest popularity mean and lowest std deviation for each route to find the most popular artists which are all at the same level. This will increase the likelihood of a greater impact with the artist that we are trying to reach.



```
df_best_route.describe()
```

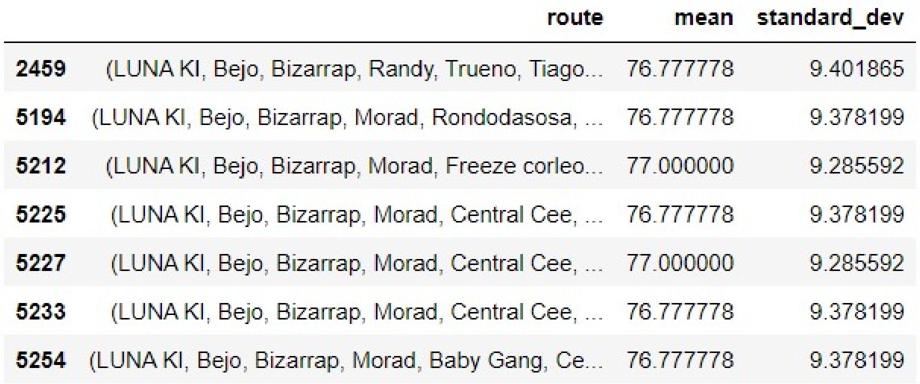
	mean	standard_dev
count	23763.000000	23763.000000
mean	66.165661	11.199626
std	4.328448	1.178065
min	56.444444	8.216214
25%	63.111111	10.380562
50%	65.333333	11.089980
75%	68.666667	11.925696
max	82.555556	17.505555

Graph community analysis

Tools & conclusions

Second Tool – Shortest route from one node to another:

These are the best routes to reach Rels B, 5212 & 5227 seems the best ones, however we believe that it would be up to the artist and her team to decide which route to take and their possibilities with the shown artists.



	route	mean	standard_dev
2459	(LUNA KI, Bejo, Bizarrap, Randy, Trueno, Tiago...	76.777778	9.401865
5194	(LUNA KI, Bejo, Bizarrap, Morad, Rondodasosa, ...	76.777778	9.378199
5212	(LUNA KI, Bejo, Bizarrap, Morad, Freeze corleo...	77.000000	9.285592
5225	(LUNA KI, Bejo, Bizarrap, Morad, Central Cee, ...	76.777778	9.378199
5227	(LUNA KI, Bejo, Bizarrap, Morad, Central Cee, ...	77.000000	9.285592
5233	(LUNA KI, Bejo, Bizarrap, Morad, Central Cee, ...	76.777778	9.378199
5254	(LUNA KI, Bejo, Bizarrap, Morad, Baby Gang, Ce...	76.777778	9.378199

4.

Lyric analysis classification

5.

Project enhancement & conclusion

Project enhancement

Possibilities

As we developed this model, we have identified many ways that it could be improved, new tools that we could create and plentiful of different analysis we could perform.

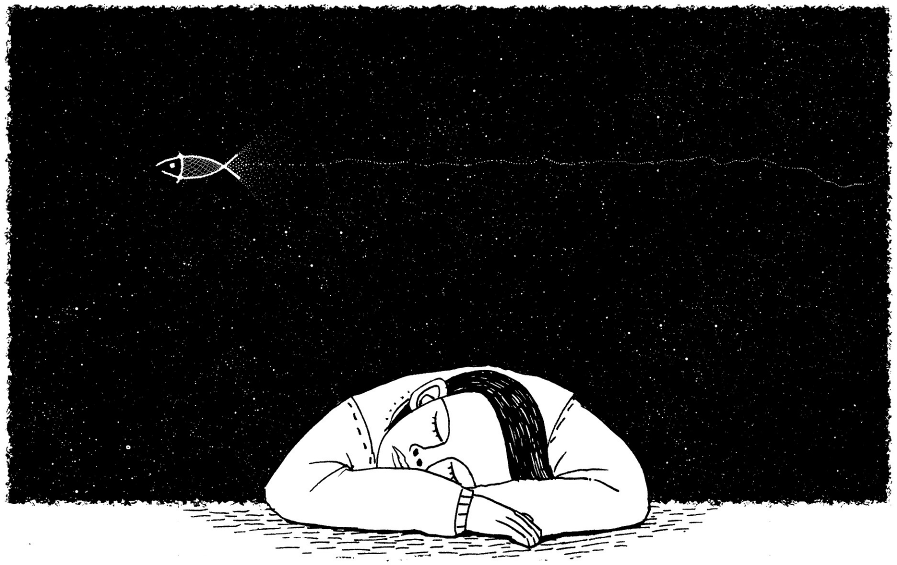
Community analysis

This idea would englobe grabbing a larger portion of the music industry and performing a similar analysis. However, this would be more focused towards a more globalized view of the music industry and thereby the use and conclusion would be for different purposes.

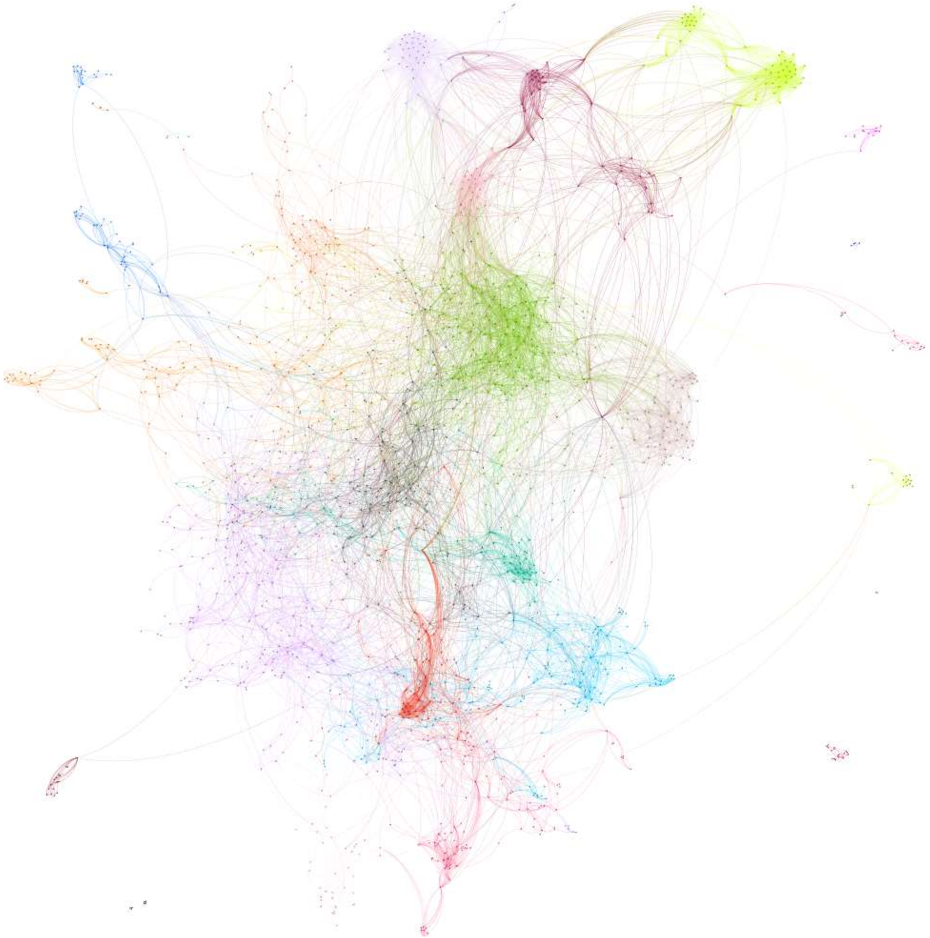
We have created a simulation of how the graph would look with 1,700+ artists and 13,000+ edges (collaborations).

This ended up in 33 different communities within the network that we would have to separately analyze and give a specific meaning.

But as mentioned previously, we could perform the analysis to any other music genre: Rock, American Hip Hop, Pop, Country, Electronic/house music etc… the options are endless, and the results always have room for enhancement.



Possibilities (cont.)



Project enhancement

Possibilities

NLP & Network analysis synergies

Another approach would be to do a classifier using NLP (supervised ML model) in order to better understand the styles and approaches for each artist. This would be a great synergy with the shortest route tool as within the given artist list in the route we can also track down which artists would be better to do collaborations with according to our style classifier.

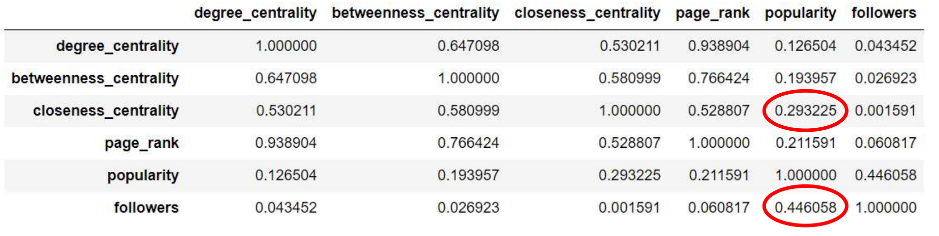
Spotlight node predictor

Applying the predictability tools that Machine Learning provides us in Data Science, we have also been investigating on the metrics which make an artist successful. Our approach would be to do a time series with all the metrics that we see make a difference in the career of an artist which make him/her push up a knot in the pyramid of the best artist in a certain point in time.

This is an ambitious idea however we believe that with our technical knowledge as Data Scientists and our knowledge about the music industry we could perform a thorough analysis and attain some important insights on this.

Metric correlation

To extend the possibilities within our ideas, we were curious to know if there was any correlation between the metrics provided by Spotiscience and the Spotify API about the artists, albums, tracks etc… and the metrics provided by the network analysis. Just to run an example we found some correlations after the addition of two other metrics: Popularity & Followers



	degree_centrality	betweenness_centrality	closeness_centrality	page_rank	popularity	followers
degree_centrality	1.000000	0.647098	0.530211	0.938904	0.126504	0.043452
betweenness_centrality	0.647098	1.000000	0.580999	0.766424	0.193957	0.026923
closeness_centrality	0.530211	0.580999	1.000000	0.528807	0.293225	0.001591
page_rank	0.938904	0.766424	0.528807	1.000000	0.211591	0.060817
popularity	0.126504	0.193957	0.293225	0.211591	1.000000	0.446058
followers	0.043452	0.026923	0.001591	0.060817	0.446058	1.000000

Possibilities (cont.)

Conclusions

Music is still one of our passions and this mixed up with the Data Science tools we have been able to produce a model that can be enhanced and used for the better.

Uncertainty is always an issue when it comes to decision making specially when the opportunity cost is extremely high, all we wanted to do is to reduce this uncertainty and help out the stakeholders within the music industry to continue evolving into a better form of itself and thereby fueling those who are passionate about this everyday, and be proud to put in a little help for the common good

Fuentes de Información y Procedimientos

Fuentes de Información

En el transcurso de nuestro análisis, hemos utilizado información financiera, incluyendo la información proporcionada por Cinfa y Orliman, así como de diversas fuentes públicas, financieras y de la industria. Nuestras conclusiones son dependientes de que dicha información sea completa y correcta en todos los aspectos materiales. Sin embargo, el alcance de nuestro trabajo no nos permite aceptar responsabilidad por la exactitud o integridad de la información proporcionada por el Cliente.

Como parte de nuestro programa de trabajo, hemos llevado a cabo reuniones con la Dirección de Cinfa (la "Dirección"), Roberto Otamendi, Esther Lacave, María Castiella y el equipo de Orliman que cubren las siguientes áreas:

- Dirección General y Estrategia (Enric Florensa) para obtener una visión general del negocio, entender la información proporcionada, los factores clave del negocio y las expectativas de cara al futuro;
- Dirección Financiera y Contabilidad (Bárbara González, Miquel Navarro) para entender la información contable y financiera reportada y las particularidades de la misma;
- Dirección Comercial (Ángel Garde) para entender la operativa comercial y el posicionamiento en el sector de la marca los productos Orliman y las relaciones con los clientes;
- Dirección Operativa (Salvador Figueres) para obtener un visión sobre las operaciones de la Compañía.

Fuentes de Información (cont.)

A continuación facilitamos la principal información y documentos recibidos en los que hemos basado nuestro trabajo:

- Share Purchase Agreement: *" 20170731 – Elevación público transmisión Prodigo.pdf, Purchase Price Extract.xlsx".*
- Planes de negocio: "*Modelo Orliman V5 (003).xlsx*";
- Cuentas Auditadas: *"Pródigo 2014 Consolidadas, Pródigo 2015 Consolidadas, Pródigo 2016 Consolidadas";*
- Balances por UGE y consolidado a fecha de valoración: *"072017 – Balance – Sabana LEGAL Orliman.pdf ";*
- Due Dilligence Financiera, Comercial y Operativa, Fiscal, Laboral y Legal: *"DD Financiera.pdf, DD Comercial y Operativa.pdf, DD Fiscal.pdf, DD Laboral.pdf, DD Legal.pdf";*
- Información sobre la marca Orliman: *"Informe de Marcas.pdf, BDO_Informe Valoración Marca 30 11 10.pdf";*
- Información sobre patentes;
- Presentación de la dirección con información sobre el mercado, competidores, entorno competitivo, proposición comercial, estrategia de marketing, I+D y modelo productivo: *" Manegement Presentation.pdf";*
- Información sobre el personal;
- Información de estudio de mercado: *"Presentación Estudio de Mercado Orliman en Farmacias.pptx, Presentación Estudio de Mercado Orliman en ortopedias España.pptx".*