

Making Sense of Spotify's Genre Mosaic

A Network Science Perspective on Artists' Genres Through the Lens of Album Artwork

Alessio Cascione

a.cascione@studenti.unipi.it

Student ID: 582765

ABSTRACT

In this report, our goal is to comprehend and analyze a network of Spotify artists utilizing techniques from network science. The initial section of our research is dedicated to explaining the criteria underlying the extraction and graph-building process. This involves a thorough exploration of standard network science quantitative analysis methods, examining the network's characterization, and comparing it to synthetically constructed graphs. Moving forward, we delve deeper into the study of the network's properties. This includes a comparative analysis of different algorithms for solving a static community discovery task, as well as unsupervised and supervised link prediction tasks. In conclusion, we try to understand how album covers extracted from artists' networks are related to internal communities inside the graph and consider if the knowledge extracted is sufficient to predict the artist's main genre based on the associated album, given a simple machine learning model¹

KEYWORDS

Social Network Analysis

¹Project Repositories

Data Collection: https://github.com/sna-unipi/sna-2023-2023_cascione/tree/main/data_collection

Analytical Tasks: https://github.com/sna-unipi/sna-2023-2023_cascione/tree/main/network_analysis

Report: https://github.com/sna-unipi/sna-2023-2023_cascione/tree/main/report

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

SNA '23, 2022/23, University of Pisa, Italy

© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...\$0.00

<https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

ACM Reference Format:

Alessio Cascione. 2023. Making Sense of Spotify's Genre Mosaic: A Network Science Perspective on Artists' Genres Through the Lens of Album Artwork. In *Social Network Analysis '23*. ACM, New York, NY, USA, 14 pages. <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>

1 INTRODUCTION

In recent years, Spotify became one of the largest music streaming service providers, with over 551 million monthly active users, including 220 million paying subscribers, as of February 2022². One of the most interesting novelty the platform offers is a "Fans Also Like"-feature, used to relate artists in terms of similarity considering shared genres and audience³. We leverage Spotify API and network science techniques in order to extract a network of relevant related artists and perform quantitative and qualitative investigations on such network structure.

2 DATA COLLECTION

The network built for the analysis resulted from direct extraction of Spotify artists' information using Python Spotify library⁴, allowing us to retrieve for each artist up to 20 related musicians as suggested by Spotify⁵. We also took into account Kword.net's top 3000 most streamed artists⁶, in order to set an initial number of nodes from which the retrieval could start. For each artist, additional attributes were extracted regarding:

- Artist's main genre (and a set of all the genres associated to such artist).
- Artist's level of popularity, assigned by Spotify.
- Artist's number of followers.

²<https://newsroom.spotify.com/company-info/>

³<https://artists.spotify.com/blog/how-fans-also-like-works>

⁴More details in <https://github.com/spotify-dev/spotify>

⁵Criteria and algorithms leveraged by Spotify to establish the relationships are not public, so we will take their reliability for granted.

⁶<https://kword.net/spotify/listeners.html>, considering the list of July 2023.

Crawling Methodology and Assumptions

To minimize the extraction of irrelevant information, concerning nodes with very low popularity, we concentrated our retrieval efforts on artists with a popularity score equal to or higher than 20. Additionally, we opted to restrict the extraction of nodes once we reached a count of 63717 nodes, ensuring a more manageable network size. The graph was modeled as a simple undirected and unweighted graph, with each edge connecting two artists if Spotify designates them as a pair of related artists. As specified before, the extraction process starts from a graph that includes 3000 popular artists, building the network incrementally adding related artists to the previously found ones.

3 NETWORK CHARACTERIZATION

The following section will be focusing on basic characteristics of the extracted network, taking into account how measures on the graph relate to synthetically built undirected networks with the same number of nodes and approximately the same number of edges. We will compare the artists network (AN) to:

- Erdős-Rényi (ER) random graph, generated using $p = 0.000184$ as probability of link between two nodes.
- Barabási-Albert (BA) random graph with 6 as the number of edges to attach from a new node to existing nodes.
- Watts-Strogatz (WS) with $k = 12$ as nearest neighbors of a node to be joined with it and $p = 0.2$ as probability of rewiring each edge.
- Configuration Model (CM) fitting the degree distribution of the artists network.

Degree distribution analysis

Model	Number of nodes	Number of Links	Average Degree
AN	63717	373718	11.7306
ER	63717	372908	11.7051
BA	63717	382266	11.9989
WS	63717	382302	12.0000
CM	63717	373602	11.7269

Table 1: Links and degree comparison between artists networks and synthetic networks

Model	Highest Degree	Lowest Degree	Self Loops
AN	99	0	0
ER	28	1	0
BA	983	6	0
WS	20	7	0
CM	99	0	6

Table 2: Highest/Lowest degree comparison

Table 1 allows us to emphasize how the number of links is much less than the number of all the potential edges in the graph, implying a very low density value (0.00018). A first comparison between the artists graph and the synthetic one regards the similarity among them in terms of average degree. More relevant differences become evident when we compare their degree distribution: specifically, except for the CM model, which fits the AN data-distribution reasonably well since it was constructed with that very distribution as a reference, the BA model is the only one that closely matches the real distribution within a limited range of degree values. On the other hand, the remaining models exhibit noticeable differences in this regard: being poissonian distributions, they strongly differ from the real world network.

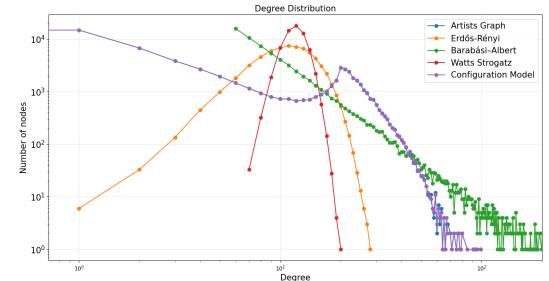


Figure 1: Degree distribution comparison

Analyzing the top nodes by degree in the artists network revealed a prevalence of classical performances, latin hip-hop and Indian film music as top genres - these being nodes that could be labeled as hubs for high degree thresholds. We can further analyze differences in terms of nodes highest/lowest degree with respect to synthetic graphs: relevant differences arise focusing on high degree nodes, as Table 2 suggests, with results consistent with the approaches used to build the synthetic networks.

Connected components analysis

We also perform an analysis regarding connected components in both real world network and synthetic ones: a connected component can be identified as a sub-graph whose nodes can be reached from one another by following the network's links. For ER, WS and BA networks a single giant component equal to the whole original graph can be isolated. CM network presents a total of 168 components, the biggest one having 63399 nodes, with the remaining components having a much smaller cardinality of 1, 2 or 3 total nodes. The same pattern is identified in the AN networks, having 21 components, the biggest one including 63695 nodes and all the others at most 2 nodes. Following [1, p. 59], the networks under analysis can be considered instances of connected regimes, where the average degree $\langle k \rangle$ is equal or very close

to $\ln(N)^7$, leading the vast majority or the totality of nodes to be encapsulated into a single giant component, still keeping the network relatively sparse.

Clustering coefficient and assortativity

The local clustering coefficient of a node can be considered as the proportion of the number of links between the nodes within its neighbourhood divided by the number of links that could possibly exist between them. When this value is close to one, it indicates how the node's neighbours are close to being a clique, or how complete the neighborhood of the node is. The average local clustering coefficient (Average CC), as the name suggests, provides insight regarding how complete the neighborhood of a node is, on average. All the synthetic models fail to correctly approximate the average clustering coefficients of the real world graph, which instead shows a much higher value, as depicted in Table 3: this highlights how nodes in the artists network tend to have neighbours which form cliques more frequently than their synthetic counterparts.

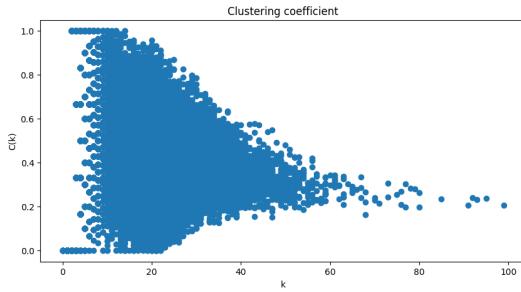


Figure 2: Clustering coefficient distribution

Model	Density	Average CC	#Triangles	Assortativity
AN	$18 \cdot 10^{-5}$	0.3962	3118005	0.1702
ER	$18 \cdot 10^{-5}$	0.0002	888	0.0016
BA	$18 \cdot 10^{-5}$	0.0018	19068	-0.0144
WS	$19 \cdot 10^{-5}$	0.0002	1469055	-0.0135
CM	$18 \cdot 10^{-5}$	0.0005	4839	-3.1908

Table 3: Density and clustering coefficient comparison between artists networks and synthetic networks

As mentioned before, all the graphs show very low density values; the artists network is the one having the highest degree assortativity score, measuring the similarity of connections in the graph with respect to the node degree. We also leverage three nodes' features in order to provide further assortativity scores, measuring the similarity of the connections with respect to specific attributes. For this aim, we

⁷In our particular case, $\ln(N) = 11.062$.

discretize the number of followers attribute into 250 bins using a quantile-strategy⁸. The highest result is given by using the main genre of an artist as reference attribute (0.3358), while lower results were obtained for popularity (0.0191) and the discretized number of followers (0.01162).

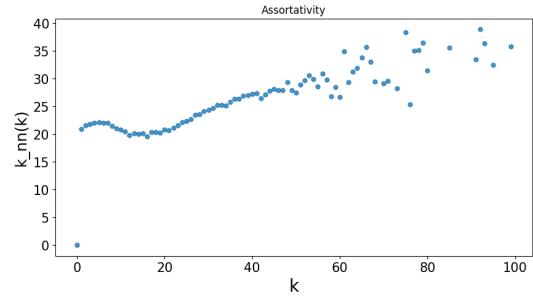


Figure 3: Nodes correlation through degree assortativity

Path analysis

Computing a large-size graph diameter and its average shortest path can be costly, as in our case. Therefore, we apply approximation methods in order to get an estimate of these real values computed on the respective biggest giant components. Following [8] and the 2-sweep algorithm implementation offered in Networkx library for undirected graphs, we pick the farthest node from a random node and return its eccentricity (i.e. length of a longest shortest path starting at that node) in order to obtain a lower bound on the diameter value. In order to compute an estimate for the average shortest path length, we leverage the following process: for each giant component, we randomly sample 200 pairs of nodes and compute their shortest path length, then we average the obtained values. We repeat the previous step 100 times, getting 100 average values coming from different samples of size 200. Their mean value represents our estimate for the average shortest path of the giant component under analysis. We report the obtained results⁹ in Table 4.

Model	2-sweep diameter	Appr. average shortest path
AN	23	8.7724 ± 0.1308
ER	8	4.7621 ± 0.0407
BA	6	3.9526 ± 0.0367
WS	8	5.7999 ± 0.0600
CM	12	4.5854 ± 0.0550

Table 4: Approximated diameter and average shortest path length

⁸All bins in each feature end up having the same number of points.

⁹For the average shortest path length, results are in the form of mean of the samples \pm standard deviation.

The artists network shows the highest average shortest path length, although presenting the highest standard deviation among the estimations made. This might be interpreted as the average number of jumps a user has to make to pass from one artist to a different one following the “Fans Also Like”-feature. In terms of diameter, the artists network shows again the highest value with respect to the other synthetic graphs, which present results in line with their topological inherent features.

Centrality analysis

We also try to identify the most relevant artists according to different centrality measures. We discuss results obtained with degree centrality (taking into account the number of neighbours of each node), pagerank centrality (computing a ranking of the nodes in the graph based on the structure of the incoming links), harmonic centrality (given a node u , its harmonic centrality is the sum of the reciprocal of the shortest path distances from all other nodes to u), eigenvector centrality (given a node with index i in the adjacency matrix A and the eigenvalue λ , its eigenvector centrality is the i -th element of the vector x defined by $Ax = \lambda x$), closeness centrality (given a node u , the reciprocal of the average shortest path distance to u over all $n - 1$ reachable nodes).

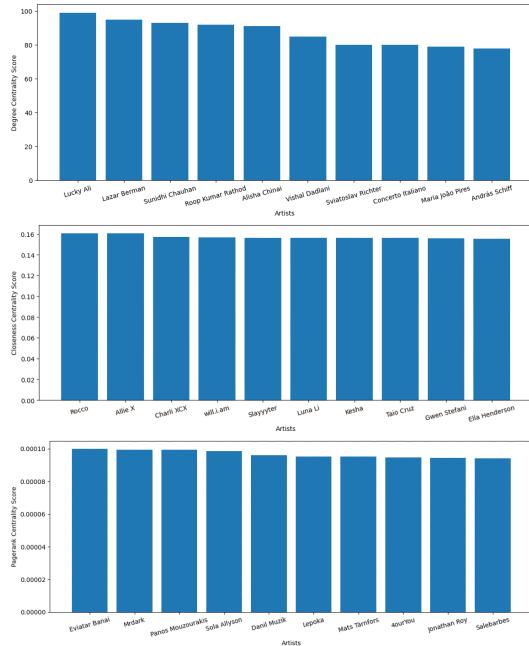


Figure 4: Top 10 nodes for degree, closeness and pagerank centrality

Focusing on the top results for each centrality measure, we mention how closeness and harmonic centrality measures match in terms of the top nodes identified, all of them

belonging to the bedroom r&b genre. Eigenvector and degree centrality measures match in terms of the top nodes, although in this case all of them belong to the Indian film music genre. Different results can be seen in terms of pagerank centrality, with most of the top nodes belonging to classic israeli pop genre.

4 TASK 1: STATIC COMMUNITY DETECTION

In this section we focus on a community detection task regarding the whole extracted artists network. For this purpose, we first take into account four different community detection algorithms, compare their performances in terms of internal and external evaluation metrics and then move to assess the nature of the clusters found by the best performing method among the ones under analysis.

Internal evaluation

We take into account the CDLIB¹⁰ implementation of Label Propagation, K-clique, Demon and Louvain community detection algorithms, performing for the last three a random search taking into account 3 instances of evaluation to find better parameters for the partitioning process, using Girvan-Newman modularity¹¹ for choosing the best parameters configuration. The best obtained parameters are shown in Table 5¹². Table 6 completes the internal evaluation summarizing the obtained results.

Algorithm	Parameters	#Communities	Modularity
Label Propagation	-	2027	0.7558
K-clique	$k = 7$	1422	0.2520
Demon	min_com_size=4, $\epsilon = 0.2$	460	0.5688
Louvain	res=0.6, rand=True	89	0.9345

Table 5: Communities size, modularity and coverage comparison between tested algorithms

Algorithm	Cut ratio	Avg. internal degree	Conductance
Label Propagation	$4.09 \cdot 10^{-5}$	5.5681	0.3486
K-clique	$2.02 \cdot 10^{-4}$	10.5952	0.5303
Demon	$6.68 \cdot 10^{-5}$	13.1771	0.2470
Louvain	$4.27 \cdot 10^{-6}$	8.4918	0.0244

Table 6: Cut ratio, average internal degree and conductance comparison between tested algorithms

Highest modularity scores (i.e the fraction of edges within communities minus the expected fraction of such edges) are obtained for Label Propagation and Louvain, which are also

¹⁰More information is offered in [11].

¹¹More details a in [4].

¹²For K-clique we tested with k from 2 to 10; for Demon, an ϵ from 0.1 to 0.9 with step=0.1 and a min. community size of 3,4,5; for Louvain, the randomize parameter with resolution from 0.1 to 0.9, with step=0.1.

capable of covering the whole graph in terms of nodes without overlapping clusters. Furthermore, Louvain resulted the partitioning method with the lowest cut ratio (a measure indicating the possible number of links leaving the community) and lowest conductance (i.e the ratio between relationships that point outside a community C and the total number of relationships of C): these properties, along with the one of having a 100% coverage of the graph's nodes and the relatively balanced average internal degree, make it the best partitioning method from the internal assessment process.

External evaluation

We compute the normalized mutual information between Label Propagation and Louvain results, having these two method a complete coverage over the graph and obtaining a score of 0.6941, indicating a partial overlapping between the two sets of communities identified. In order to analyze also community detection methods not covering the same set of nodes, we exploit the normalized F-1 score. Comparisons are shown in Table 7:

Algorithm	Label Propagation	K-clique	Demon	Louvain
Label Propagation	-	0.1018	0.0133	0.0030
K-clique	0.0779	-	0.0123	0.0014
Demon	0.0190	0.0075	-	0.0090
Louvain	0.0230	0.0088	0.0673	-

Table 7: Normalized F1 score of the algorithms partitions comparison

All the scores are relatively low: the highest is given by Label Propagation compared with K-clique algorithm but we can't generally claim a relevant similarities between NF-1 results. We can exploit the additional semantic information regarding music genres attached to the graph's nodes for a deeper analysis of the Louvain communities results. For each node, we have a list of genre the artist is associated with along with its main genre. Computing the average purity¹³ in terms of main genre gives us poor results ($24.56\% \pm 11.56\%$, if we remove from the analysis the 20 Louvain communities having a size smaller than 3), but this should not be immediately linked with poor clustering results in general: there is a total of 4728 different genres in the graph, rendering the attempt to make sense of the obtained clusters in terms of these fine-grained labels unreliable. In fact, switching to a more coarse-grained approach gives us more reasonable results: for each community, we evaluate for each genre how many times it occurs in one of the genre sets associated with

¹³Here with purity of a community with respect to an attribute we mean the number of instances the most frequent label of that attribute has in that community, taken in percentage.

each artist in the community, taking the value in percentage with respect to the community size. As expected, this approach leads to more reasonable results: even if most of the communities obtained tend to show a majority of not available genre as their most frequent genre, many results make sense regarding intuitive semantic similarity between the genres that tend to recurr the most in the community. Some interesting examples are shown in Table 8. Even if just approximately, we can claim that Louvain is also able to capture homogenous clusters in terms of coarse-grained information regarding genre.

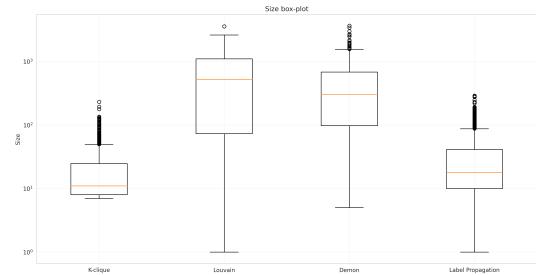


Figure 5: Algorithms comparison in terms of size

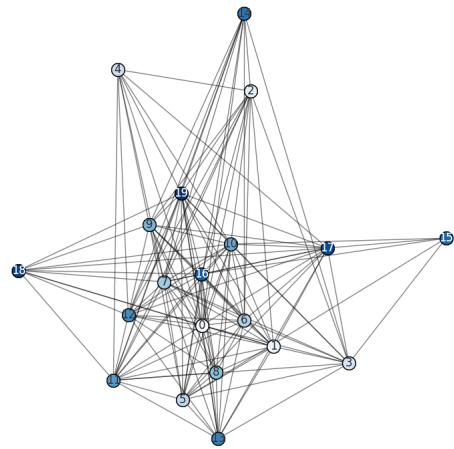


Figure 6: Visualization of top 20 Louvain communities relationships

Community label	Size	Top 5 genres in lists
59	143	('death metal', 59.44%), ('brutal death metal', 51.75%), ('deathgrind', 41.96%), ('technical death metal', 34.27%), ('grindcore', 30.77%)
38	665	('italian adult pop', 21.80%), ('italian pop', 18.16%), ('classic italian pop', 17.44%), ('italian hip hop', 15.49%), ('genre not available', 11.43%)
31	827	('k-pop', 27.90%), ('korean pop', 20.69%), ('k-rap', 15.96%), ('genre not available', 11.35%), ('k-indie', 10.40%)
54	261	('bebop', 31.42%), ('jazz', 26.82%), ('hard bop', 22.61%), ('background jazz', 22.22%), ('jazz saxophone', 17.62%)
3	2171	('genre not available', 18.19%), ('j-pop', 8.34%), ('j-rock', 8.20%), ('anime', 6.96%), ('j-rap', 5.02%)
0	3553	('genre not available', 10.86%), ('pop', 4.93%), ('r&b', 4.84%), ('alternative', 4.59%), ('hip hop', 3.86%)
29	900	('mpb', 18.11%), ('samba', 15.88%), ('bossa nova', 14.88%), ('pagode', 14.44%), ('velha guarda', 11.44%)
10	1827	('genre not available', 13.19%), ('chanson', 8.81%), ('french rock', 5.74%), ('nouvelle chanson francais', 4.81%), ('trip hop', 4.59%)

Table 8: Examples of most frequent genres in artist genre lists

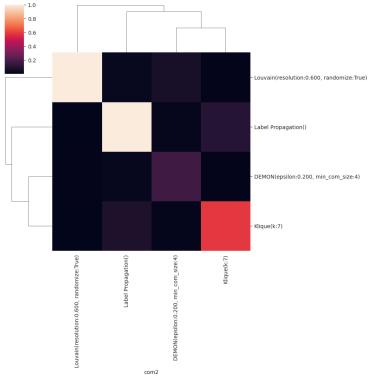


Figure 7: NF-1 comparison matrix between algorithms

5 TASK 2: LINK PREDICTION

The goal of this section is understanding to what extent and precision it is possible to predict potential connections between unlinked nodes inside the artists network. We tackle the problem from two distinct perspective: using unsupervised tools (relying on inherent topological structure of the network to solve the task without taking into account any ground-truth information to learn from) and supervised ones (exploiting "traditional" supervised machine learning methods, learning a model from a proper labeled data representation in order to test its performance on not already seen observations).

Unsupervised link prediction

We chose to test three unsupervised methods focusing on nodes' topology, these being Common Neighbors, Jaccard similarity, and Adamic Adar, and a link-based similarity approach, namely SimRank¹⁴. In Common Neighbors, the score for link prediction is computed by finding the number of common neighbors between two distinct nodes as follows, assuming that nodes sharing many common neighbors should also be linked themselves:

$$CN(u, v) = |N_u \cap N_v| \quad (1)$$

¹⁴A deeper explanation regarding the measures can be found in [7].

Jaccard similarity assumes that nodes sharing many neighbors over their merged common neighborhoods should be similar, hence connected:

$$Jaccard(u, v) = \frac{|N_u \cap N_v|}{|N_u \cup N_v|} \quad (2)$$

Adamic Adar takes into account nodes' degrees, giving higher impact to lower degree nodes and lower impact for hubs. More specifically, higher degree nodes could appear more frequently in the same neighborhood than nodes with less degree, hence the former should contribute less to the score than the latter:

$$AA(u, v) = \sum_{k \in N_u \cap N_v} \frac{1}{\log(|N_k|)} \quad (3)$$

SimRanks follows a different rationale, assigning a likelihood of observing a link between two nodes analysing the similarity of their respective neighbourhoods:

$$SR(u, v) = C \frac{\sum_{k \in N_u} \sum_{k' \in N_v} SR(k, k')}{|N_u| \cdot |N_v|} \quad (4)$$

where C is a decay factor in the range $[0, 1]$.

Due to the high computational cost of these methods¹⁵, we employ a sub-graph of the original network, including only nodes having a degree higher or equal to 40 and excluding from the obtained sub-graph nodes belonging to components having cardinalities lower than 5. This allows us to reduce the analysis over a graph comprised of 666 nodes and 2582 edges: 225 were used during testing phase and the remaining one for the training of the unsupervised methods. Figure 8 reveals the low performances concerning topology oriented methods on the analyzed sub-graph: Jaccard Similarity is slightly above the other two considered methods in terms of ROC-AUC, but still unsatisfactory, even if it shows the best trade-off between Precision and Recall scores focusing of PR-AUC. A relevant change in performances can be observed studying SimRank results, as shown in Figure 9: we

¹⁵The problem is mainly related to the necessity of computing every possible edge given a set of nodes: in fact, most methods we will deal with imply a not-optimal $O(|V|^2)$ cost, where V is the number of nodes in the network.

employed three different decay factor values, each of them showing roughly the same performances in terms of ROC-AUC and shared top predictions. Using $C = 0.3$ seems to lead to slightly better results in terms of Precision-Recall trade-off, hence we will focus on this method for further comparisons. We observe how each SimRank evaluation lead to optimal results once compared with a random link prediction classifier. Table 9 completes the analysis showing the top 250 shared predictions among the unsupervised methods, highlighting the similarities between Adamic Adar results and Common Neighbors and Jaccard with SimRank with $C = 0.3$.

Method	Common N.	AdamicAdar	Jaccard	SimRank
Common N.	250	205	59	0
AdamicAdar	205	250	93	4
Jaccard	59	93	250	127
SimRank	0	4	127	250

Table 9: Number of shared predictions among the top 250 scores across unsupervised methods

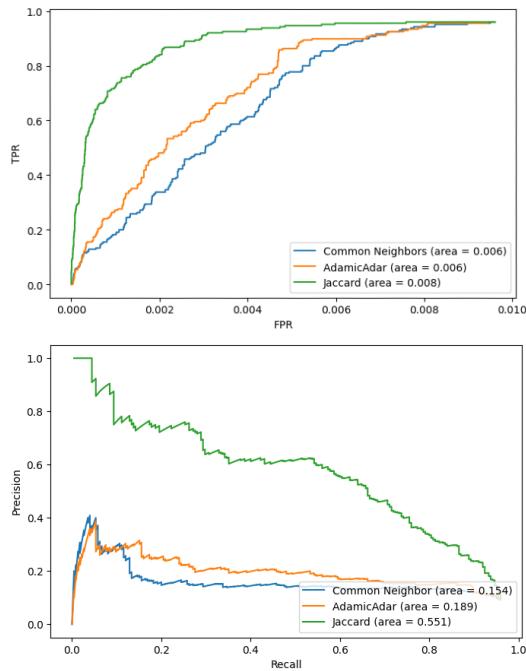


Figure 8: ROC curves and Precision-Recall curves for Common Neighbors, Jaccard and Adamic Adar methods

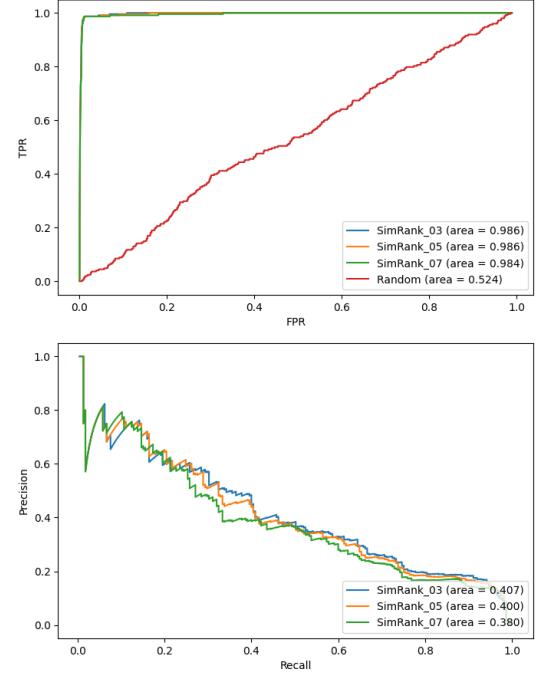


Figure 9: ROC curves and Precision-Recall curves for SimRank method

Supervised link prediction

We also explore solutions regarding supervised approaches, implementing traditional machine learning models for link prediction. We employ StellarGraph library¹⁶ in order to sample positive and negative links from the initial graph (where a positive label identifies the actual presence of that link, a negative label its absence), splitting the graph into a training set (50451 node pairs) and test set (74742 node pairs), setting 10% as percentage of links to be sampled from the starting graph. We encode the graphs obtained into Node2Vec embeddings¹⁷, using second-order random walks to generate sentences (i.e. list of node ids) from the training graph and learning an embedding vector for each word (i.e. node id). We build feature representations for links applying binary operators on the pair of embeddings associated to nodes forming that link, considering Hadamard, Average, L1 and L2 operators as described in [12]. We take into account three ML models¹⁸: K-Nearest Neighbors (K-NN)¹⁹, Decision Tree

¹⁶More information available at <https://github.com/stellargraph/stellargraph>, [2].

¹⁷More information available at <https://github.com/aditya-grover/node2vec>, [6].

¹⁸We consider the implementation offered in Scikit-learn library. More detailed information is offered in [9].

¹⁹Testing (3,5,7,9,11) as possible ks, giving uniform weights to the points or weighting them by their distance.

(DT)²⁰ and Logistic Regression (LR)²¹. For each solver, a grid-search using a stratified k -fold cross validation approach with $k = 5$ in order to identify the best parameters for each particular class of models has been performed, considering different feature spaces obtained from different binary operators. As scoring criterion considered for the extraction of the best model, accuracy has been chosen. Models reported in Table 1 are the ones with the highest mean score over the validation folds²² according to their respective gridsearch, for each feature space.

Feature Space	Logistic Regression		Decision Tree		K-Nearest Neighbors	
	Mean VL	TS	Mean VL	TS	Mean VL	TS
Hadamard	96.90% ± 0.13%	96.91%	88.58% ± 0.20%	88.67%	96.75% ± 0.11%	96.70%
Average	65.88% ± 0.40%	66.60%	83.49% ± 0.25%	84.52%	95.47% ± 0.24%	95.63%
L1	94.42% ± 0.15%	94.49%	86.60% ± 0.33%	87.01%	86.69% ± 0.18%	87.65%
L2	94.32% ± 0.17%	94.41%	86.55% ± 0.24%	87.05%	85.22% ± 0.35%	86.30%

Table 10: Supervised link prediction task: validation and assessment results for each model and binary operators

Hadamard operator provided the best results for every method considered: LR with lbfsgs solver, $C = 0.001$ and L2 penalty proved to be the best method in testing phase in terms of accuracy, followed by K-NN using $k = 3$ and weighting points by the inverse of their distance and at last by DT with Gini splitting criterion 2 as min_samples_leaf and 8 as min_samples_split. ROC-AUC values for each testing result is provided in Figure 10, emphasizing the goodness of each result. Even if LR resulted the best model in terms of Accuracy, a better trade off in terms of false positive and true positive rates is given by K-NN, which shows a slightly better AUC-score.

²⁰Testing Gini and Entropy as splitting criteria, (2,4,6,8,10) as min_samples_split (minimum number of samples required to split a node) and (1,2,3,5,8,10) as min_samples_leaf (minimum number of samples required to be a leaf node).

²¹Testing L2,L1 and no penalty, (1.0,0.0,0.001) as Cs and newton-cg, liblinear and lbfsgs solvers. In case of configurations with incompatible hyperparameter values, we simply skip them in the grid search process.

²²Results are in the form of accuracy ± standard deviation, for validation scores.

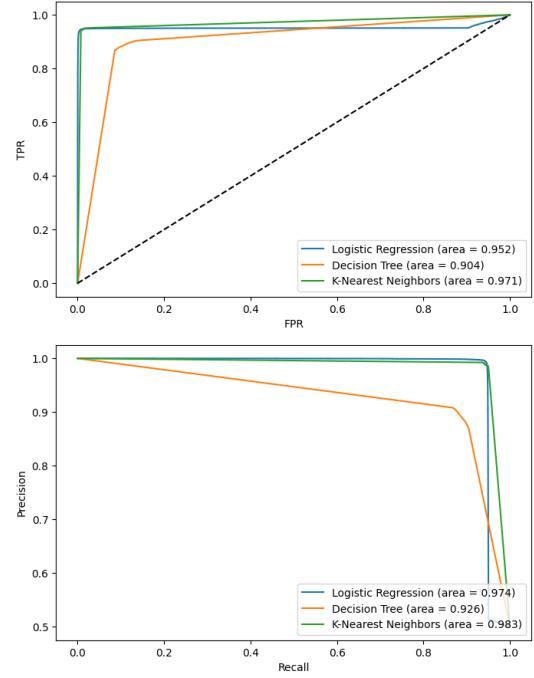


Figure 10: ROC curves and Precision-Recall curves for supervised methods

The results are completed by Table 11, reporting Precision, Recall and F1 scores regarding the presence of links for the test set in the Hadamard feature space, emphasizing the difference between the three methods.

Methods	Precision	Recall	F1
Logistic Regression	99.16%	94.61%	96.83%
Decision Tree	89.03%	88.20%	88.61%
K-Nearest Neighbors	98.62%	94.73%	96.64%

Table 11: Precision, Recall and F1 score of each solver on the Hadamard feature space for the test set

6 TASK 4: OPEN QUESTION

As an open research question, we tackle the issue of understanding if album covers released by each artist are coherently related to music genres the artist is associated with, following the identification made through Louvain community discovery algorithm. After having extracted for each artist the first accessible album cover image through Spotify API²³, we proceed with solving an image classification task, an image sentiment classification task and a topic modeling task for images. We then complete the section with a simple analysis of the performance of two convolutional neural

²³A total of 63415 over 63717 images were extracted: some albums lead to Spotify extraction problems and were simply discarded in the process.

network (CNN) architectures trying to solve the prediction task of associating to each album the genre of the artist who released that album.

Image and visual sentiment classification task

We use a pre-trained version of Vision Transformer model as illustrated in [3]²⁴ in order to assign to each album a label that should represent its visual content: the transformer is directly trained on image patches without the aid of convolutional layers and it is a sufficiently viable tools for our purpose. For the visual sentiment task, we use a modified pre-trained version of AlexNet for visual sentiment analysis as developed in [13]²⁵: the authors trained a visual sentiment classifier starting from a set of user-generated tweets containing both texts and images, leveraging on the sentiment polarity of the textual contents to train such model. We report here some quantitative results comparing the communities in terms of purity of genres, sentiment and image kind: the Spearman correlation coefficient and respective p-value checking the purity correlation between communities in terms of image content and genre, sentiment and genre and image content and sentiment are respectively -0.075 and 0.537, 0.077 and 0.525, 0.147 and 0.225. Hence no evident correlation is present in terms of purities, as shown visually with the scatter plots in Figure 11.

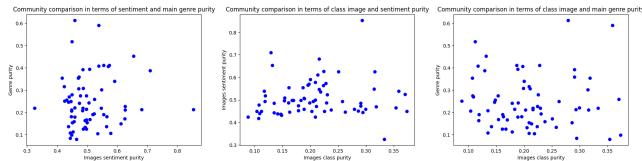


Figure 11: Scatter plots comparing purity values in terms of genre, sentiment and image content

To complete the analysis, we mention how the top 10 communities in terms of sentiment purity (having more than 57% of overall internal purity, with a max purity of 85%) comprise a majority of positive images each and can be identified in order as sertanejo, lagu jawa, mexican music, albanian pop, indian pop, funk, tamil pop, cumbia villera, israeli pop and reggaeton. We also directly check the top 5 communities in terms of image content purity (having more than 33% of internal purity, with a max purity of 37%), these respectively having as most frequent contents book jacket, dust cover, dust jacket, dust wrapper or comic book. Lower levels of purity for the image classification task are understandable,

²⁴Implementation available at https://huggingface.co/docs/transformers/model_doc/vit

²⁵Implementation available at <https://github.com/fabiocarrara/visual-sentiment-analysis>

given the higher number of labels and image variety inside the same community.

Topic modeling on images

For a complete understanding of the albums characteristics with respect to the found communities, we use Concept Python library²⁶ in order to perform topic modeling on images, finding in an unsupervised way clusters of images sharing the same semantic visual content. Using Concept, we leverage CLIP (Contrastive Language-Image Pre-training)²⁷ approach along with BERTopic²⁸ methods using transformers to create dense clusters of interpretable topics.

We perform a community-wise analysis, looking for prototypical combinations of images in the album cover of particularly homogeneous community in terms of music genres. The aim is understanding if intuitive semantic properties associated to music genres are coherently reflected and captured by the concept modeling procedure employed: for this reason we first analyze communities having both a sufficiently high cardinality to carry on the task and still keeping a somehow pure genre distribution at the same time²⁹. Some relevant examples are shown in Figure 12: each clusters of merged images captures the two or many main concepts identified by the approach³⁰. The first row focuses on the 54th community, showing how the album covers tend to represent either musicians playing classical jazz instruments or more abstract figures/natural environments. The second and third row focuses on the 31st community with the main concepts either capturing the frequency of albums displaying korean boy bands or idols and the other ones having more artistic representation or close-up photos of the artists. The fourth row displays the main concepts for brazilian popular music, focusing either on close-up images of artists or covers more abstract art representation. The last one is focused on the 14th community, showing an opposition between a more urban/street scenario concept and another one focusing on artists close-ups. On the other hand, in Figure 13 we focus on communities with less pure genre distribution but far more samples. The first eight concepts capture the pop/hip-hop community, the biggest community: reasonably, most concepts are characterized by close-up or full length photos of both male and female artists, with a sub-set of clusters focused on more artistic visual representations. The second

²⁶Implementation available at <https://github.com/MaartenGr/Concept>

²⁷More details in [10].

²⁸Details in [5].

²⁹The concept library did not provide meaningful results for very small communities, these being also the purest for the trivial reason of being rather small.

³⁰Information regarding community sizes and genres distributions used for the concept modeling examples are reported in Table 8

cluster of 7 images covers the 5th community, mainly including concepts regarding the j-pop and anime music: as one could intuitively expect, we have a concept characterized by Japanese ideograms and a different one characterized by cartoonish female figures along with other topics either focused on idol boy bands, female artists close-ups and artistic visuals. The last 6 concepts characterize the french chanson albums, focusing on male or female performers close-ups. In summary, a clear general tendency can be identified: most albums for each genre identified through Louvain community tend to include at least two separate topics that describe them: one more focused on the artist depiction or the group associated to that album and a distinct one more characterized by abstract or visually captivating representations.

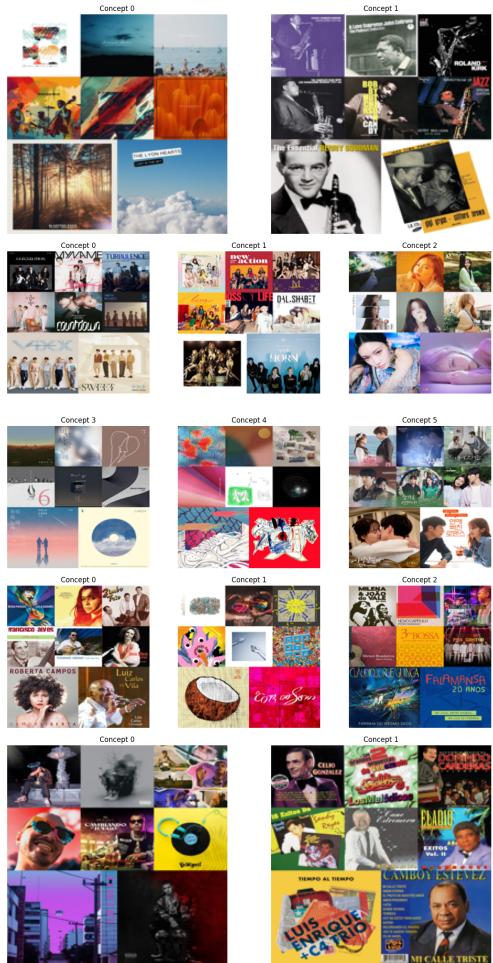


Figure 12: Concept modeling obtained from four relatively pure communities: the main genres are respectively bebop, k-pop, brazilian popular music, urbano latino



Figure 13: Concept modeling obtained from three communities of relevant size: the main genres are respectively pop/hip hop, j-pop/anime music and french chanson

Learning a CNN to predict music genres from album covers

The last task of our open question aims at investigating if a simple model can be trained on album covers in order to predict the genre associated with the artist's specific album. For this task, we first define a simpler multi-class problem: for each community, we identify the album associated to each artists belonging to such community and assign to each album a label. This label is either the most common genre in the artists' genres lists inside the community or a more generic *a priori* defined class, which captures coarser grained music genres. For instance, albums of artists belonging to German pop community, Turkish pop community and Israeli

pop are simply labeled as pop albums³¹ and an analogous reasoning is carried on for other genres. In total we obtain 23 distinct classes, as emphasized in the final test prediction heat-map in Figure 15: This allows us to obtain a data-set where samples are album images and we can treat as target instance the album/artist's genre to predict. Therefore, we extract a total of 63415³² images and keep the 30% of the samples as test set, then split the remaining part into 20% validation and 80% training set³³. We evaluate a convolutional neural network (CNN) with two distinct architectures: a *naïve* structure having 3 convolutional layers, each having a kernel size of 5, 0 padding and 1 stride, respectively of input-output size of (3,6), (6,16), (16,32) and each followed by a ReLU activation function and a max pooling operator of kernel 2 and stride 2. At the end of the convolutional process, a linear layer with 23 output values is added and a softmax function is applied on the linear output. We compare this structure with a refined architecture keeping the same *naïve* structure but adding a batch normalization operation after each convolutional layer and a final drop-out operation with 50% chance of zeroing the resulting output from the linear layer before applying a softmax operation. Each model is trained using a Cross entropy loss function with Adam optimizer, using a learning rate of 0.001 and a weight decay of 0.0001, both with a batch size of 128³⁴. The *naïve* model reached a performance peak of 33.99% accuracy on the validation set at the end of epoch 2, then started to badly overfit immediately after. On the other hand, the same regularized structure starts overfitting after epoch 6, with a peak of 33.96% accuracy. The performance are disappointing in both cases: it is evident that the models tested are very simple, but we could also speculate further. Perhaps the manually defined mapping between the Louvain communities and the 23 genres is ill-defined or data are not representative enough of the album-genres population. We could also consider the hypothesis, however not yet supported by enough tests and data, that there is no strong correlation between the graphical representation in an album and the genre of music associated with it, at least not strong enough to guarantee a simple classifier can discriminate between genres.

³¹The specific mapping for each Louvain community is specified in the provided notebooks.

³²Less then the whole graph because some artists do not have album images associated with them.

³³After the validation process, both the training and validation will be used to learn the model parameters for assessment on the test set.

³⁴We decided to avoid testing with many epochs (we tested 15 at most) and higher batch sizes mainly for lack of computational power and for the reason our goal is just getting a rough estimate of the complexity of such classification task in the context of simple CNNs.

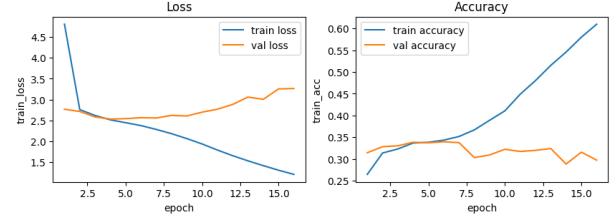


Figure 14: Training and validation loss/accuracy plot for regularized CNN architecture

Given these premises, we decided to train the regularized version of the CNN for 6 epochs on the entire defined training set and assess its result on the test set. We obtained a final 34.45% test accuracy and we depict in Figure 15 a heatmap relating true labels and predicted labels. We notice how, in terms of prediction tendency, the model is completely skewed towards the house music genres, especially in terms of classifying pop music and rap music. This at least allows us to identify the problem of genre classification in this particular image-dataset, emphasizing a negative result and leaving space for further research focusing on facing this issue.

7 DISCUSSION

Network collection, characterization and preliminary tasks

In this report, we focused on analyzing a Spotify artists network using network science quantitative tools.

- CM and BA are the two models which fits the AN degree distribution better than the other synthetic graphs tested. We notice how both AN and its synthetic counterparts are in a connected regime, with average degree very close to $\ln(N)$.
- The AN network tends to have a higher average local clustering coefficient when compared with its synthetic counterparts, but all graphs have very low density values. It also shows the highest estimated diameter and average shortest path compared to synthetic models.
- In terms of community discovery, Louvain proved to be the best method in terms of quantitative internal evaluation measures. The comparison of normalized F-1 scores did not indicate significant similarity with other community detection algorithms used for the analysis. We also observe how the communities found by Louvain relatively capture homogeneous genres when the analysis is made at a coarser-grain level.
- SimRank proved to be the best unsupervised approach for performing link prediction on the sub-sample of edges extracted from the main dataset. On the other

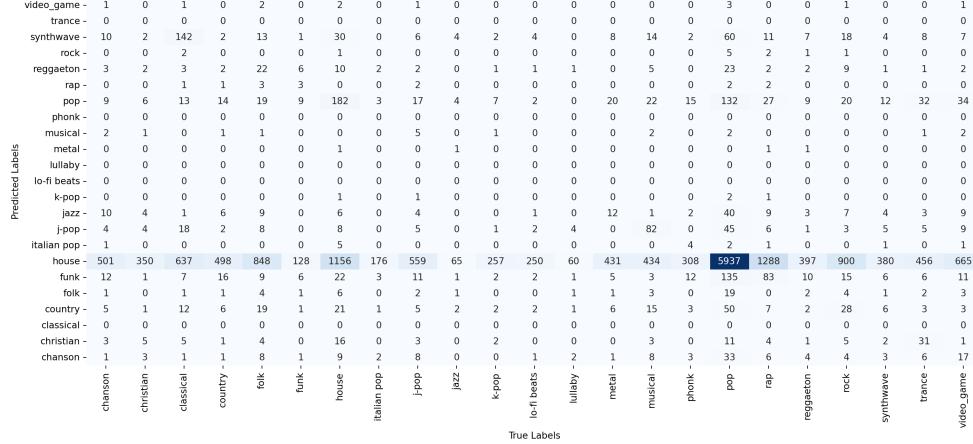


Figure 15: Heat-map representation for a comparison of predictions and true labels on album cover data-set

hand, Logistic Regression with the Hadamard operator achieved the best result in a supervised context.

Artists’ Genres Through the Lens of Album Artwork

We focused on understanding if any relevant information could be extracted about the artist network relating the communities identified through Louvain algorithm and the albums’ artwork:

- We notice how purest communities in terms of sentiment are mainly characterized by positive images, following the approach discussed in [13]. Less interpretable and reliable results are available in terms of image classification, and we also underline the weak correlation in terms of purity between image, genre and sentiment.
- Concept modeling allowed us to unveil a general common tendency among different Louvain genres communities: the fact that most communities’ albums can be described either in terms of artistic/visual representation or artists close-ups/full-length photos.
- A simple CNN architecture was trained in order to predict the Louvain music genre based on albums artwork: results were unsatisfactory. We speculate that either the labeling is ill-defined or there is no precise and strict correlation between albums’ artworks and music genres. Further research would allow us to better understand if these questions have a positive or negative answer

Future Work

Discussing future work in terms of Spotify network analysis, there lies potential in obtaining more insightful information by expanding the network size and extracting additional album images for individual artists.

This would contribute to enhancing the dataset available for the genre classification task. Moreover, incorporating textual information alongside images presents an avenue for improvement: considering the most pertinent lyrics from an artist’s songs could enhance the genre prediction task within a multimodal framework.

REFERENCES

- [1] Albert-László Barabási and Márton Pósfai. 2016. *Network science*. Cambridge University Press, Cambridge. <http://barabasi.com/networksciencebook/>
- [2] CSIRO’s Data61. 2018. StellarGraph Machine Learning Library. <https://github.com/stellargraph/stellargraph>.
- [3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. [arXiv:2010.11929](https://arxiv.org/abs/2010.11929)
- [4] M. Girvan and M. E. J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences* 99, 12 (2002), 7821–7826. <https://doi.org/10.1073/pnas.122653799> arXiv:<https://www.pnas.org/doi/pdf/10.1073/pnas.122653799>
- [5] Maarten Grootendorst. 2022. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794* (2022).
- [6] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable Feature Learning for Networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- [7] David Liben-Nowell and Jon Kleinberg. 2003. The Link Prediction Problem for Social Networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management* (New Orleans, LA, USA) (CIKM ’03). Association for Computing Machinery, New York, NY, USA, 556–559. <https://doi.org/10.1145/956863.956972>
- [8] Clémence Magnien, Matthieu Latapy, and Michel Habib. 2009. Fast Computation of Empirically Tight Bounds for the Diameter of Massive Graphs. *ACM J. Exp. Algorithms* 13, Article 10 (feb 2009), 9 pages. <https://doi.org/10.1145/1412228.1455266>

- [9] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.
- [10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. arXiv:2103.00020 [cs.CV]
- [11] Giulio Rossetti, Letizia Milli, and Rémy Cazabet. 2019. CDLIB: a python library to extract, compare and evaluate communities from complex networks. *Applied Network Science* 4 (Dec. 2019), 52. <https://doi.org/10.1007/s41109-019-0165-9>
- [12] Jörg Schlötterer, Martin Wehking, Fatemeh Salehi Rizi, and Michael Granitzer. 2019. Investigating Extensions to Random Walk Based Graph Embedding. In *2019 IEEE International Conference on Cognitive Computing (ICCC)*. 81–89. <https://doi.org/10.1109/ICCC.2019.00026>
- [13] Lucia Vadicamo, Fabio Carrara, Andrea Cimino, Stefano Cresci, Felice Dell'Orletta, Fabrizio Falchi, and Maurizio Tesconi. 2017. Cross-Media Learning for Image Sentiment Analysis in the Wild. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.