Amazon product co-purchasing network metadata

DATA VISUALIZATION PROJECT

Lorenzo Meloncelli
Martina Manno
Valerio Schettini

LUISS

Table of contents

LUISS

Dataset description

ID -Product Number

ASIN - Product Identification Number -

TITLE - Title of the product

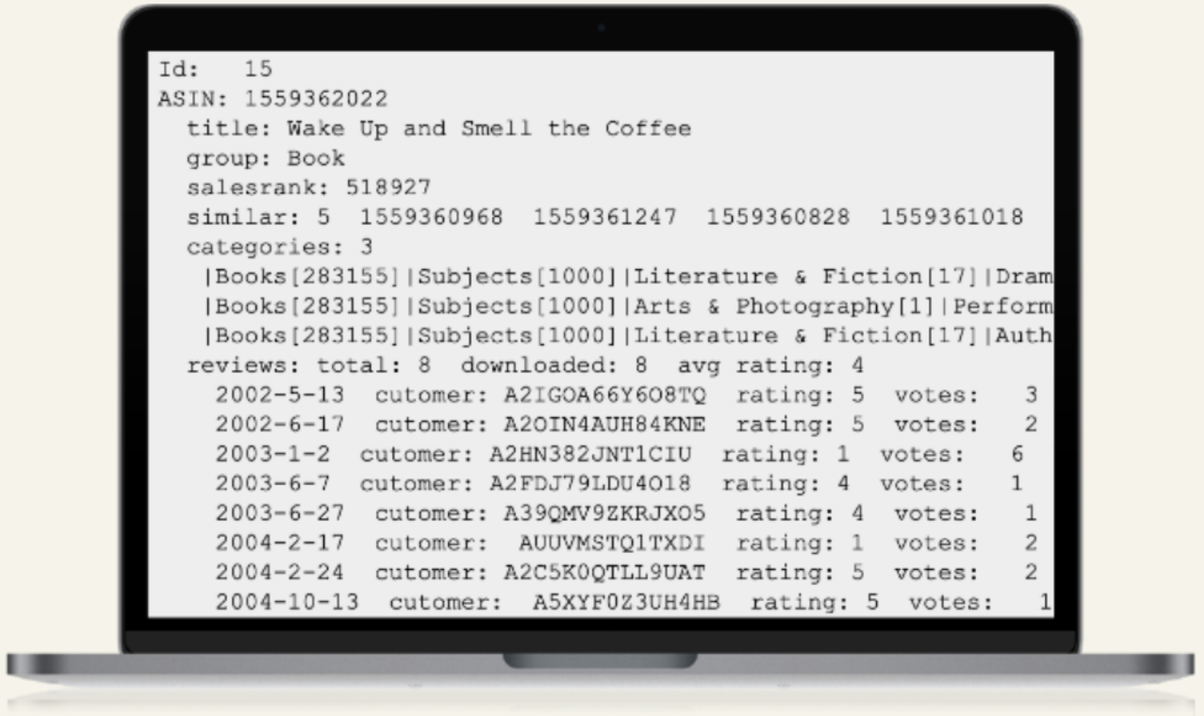
GROUP - Product Group

SALES RANK - How a product is selling in comparison to other products

SIMILAR - ASINs of co-purchased products

CATEGORIES - Product category

REVIEWS - Product review information



LUISS

Data Preprocessing

```
# Read the data from the Amazon file and fill the amazonProducts ne
(Id, ASIN, Title, Categories, Group, Copurchased, SalesRank, TotalReviews, AvgRating

for line in df:
    line = line.strip()
    if(line.startswith("Id")): # a product block started
        Id = line[3:].strip()
    elif(line.startswith("ASIN")):
        ASIN = line[5:].strip()
    elif(line.startswith("title")):
        Title = line[6:].strip()
        Title = ' '.join(Title.split())
    elif(line.startswith("group")):
        Group = line[6:].strip()
    elif(line.startswith("salesrank")):
        SalesRank = line[10:].strip()
    elif(line.startswith("similar")):
        ls = line.split()
        Copurchased = ' '.join([c for c in ls[2:]])
    elif(line.startswith("categories")):
        ls = line.split()
        Categories = ' '.join((df.readline()).lower() for i in range(int(ls[1].strip())
        Categories = re.compile('[%s]' % re.escape(string.digits+string.punctuation)).s
        Categories = ' '.join(set(Categories.split())-set(stopwords.words("english")))
        Categories = ' '.join(stem(word) for word in Categories.split())
    elif(line.startswith("reviews")):
        ls = line.split()
        TotalReviews = ls[2].strip()
        AvgRating = ls[7].strip() # a product block ended
    elif (line==""): # write out fields to amazonProducts dictionary
        try:
            MetaData = {}
            if (ASIN != ""):
                amazonProducts[ASIN] = MetaData
            MetaData['Id'] = Id
            MetaData['Title'] = Title
            MetaData['Categories'] = ' '.join(set(Categories.split()))
            MetaData['Group'] = Group
            MetaData['Copurchased'] = Copurchased
            MetaData['SalesRank'] = int(SalesRank)
            MetaData['TotalReviews'] = int(TotalReviews)
            MetaData['AvgRating'] = float(AvgRating)
            MetaData['DegreeCentrality'] = DegreeCentrality
            MetaData['ClusteringCoeff'] = ClusteringCoeff
        except NameError:
            continue
        (Id, ASIN, Title, Categories, Group, Copurchased, SalesRank, TotalReviews, AvgR
```

STEP 1: All the categories associated with the ASIN are concatenated, and then the are subject to Text Preprocessing

STEP 2: The copurchased ASINs in the “similar” field are filtered down to only those ASINs that have metadata associated with it.

LUISS

Data preprocessing

GROUP = BOOK

The platform's aim is to offer the largest number of various products while make it possible for customers to review them after the purchase. However, it started in 1994 as an online bookstore. For this reason, together with the need to reduce the number of possible nodes in the network, we decided to focus only on the book category of products. To do so, the next step would be to filter Amazon products dictionary down to only Group=Book and write it to amazonBooks dictionary.

LUISS

The Copurchase Graph

NODES

ASINs

EDGES

If two ASINs were co-purchased

EDGE WEIGHT

Based on category similarity

SIMILARITY

It can be calculated as the number of words that are common between categories of connected Nodes divided by the total number of words in both categories of connected nodes.

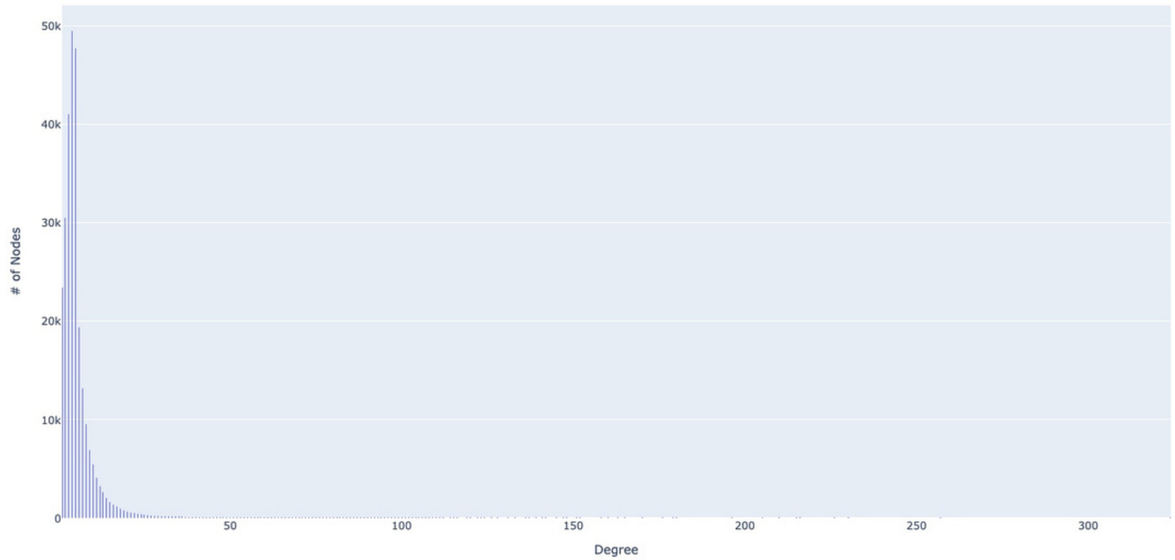
LUISS

Insights through plots

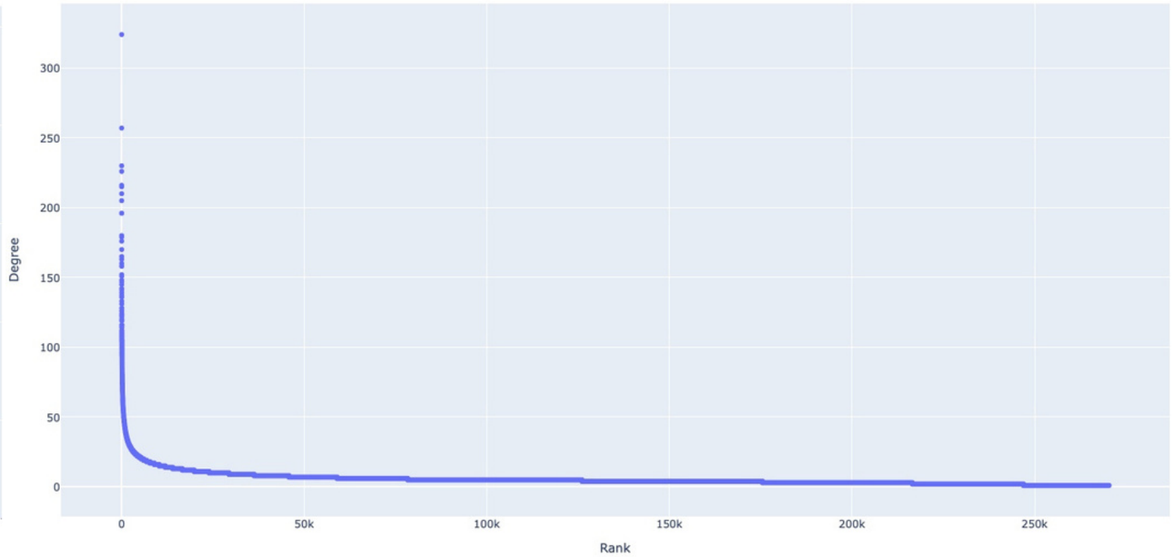
How similar is the book network to a social network distribution?

LUISS

Insights through plots



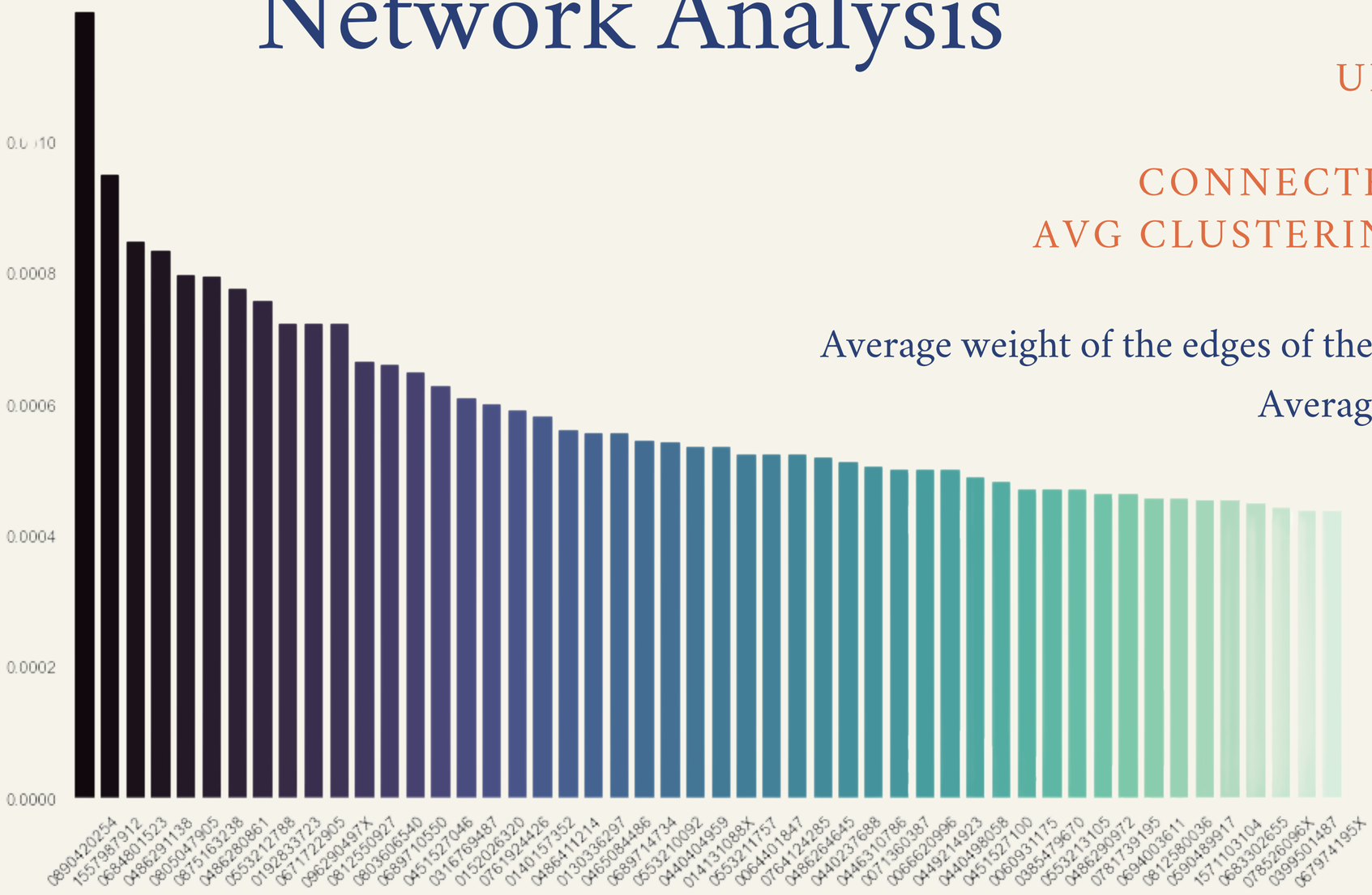
Degree histogram



Degree rank

LUISS

Network Analysis



UNCONNECTED GRAPH

AVERAGE DEGRE: 5.48

CONNECTED COMPONENT: 3840

AVG CLUSTERING COEFFICIENT: 0.40

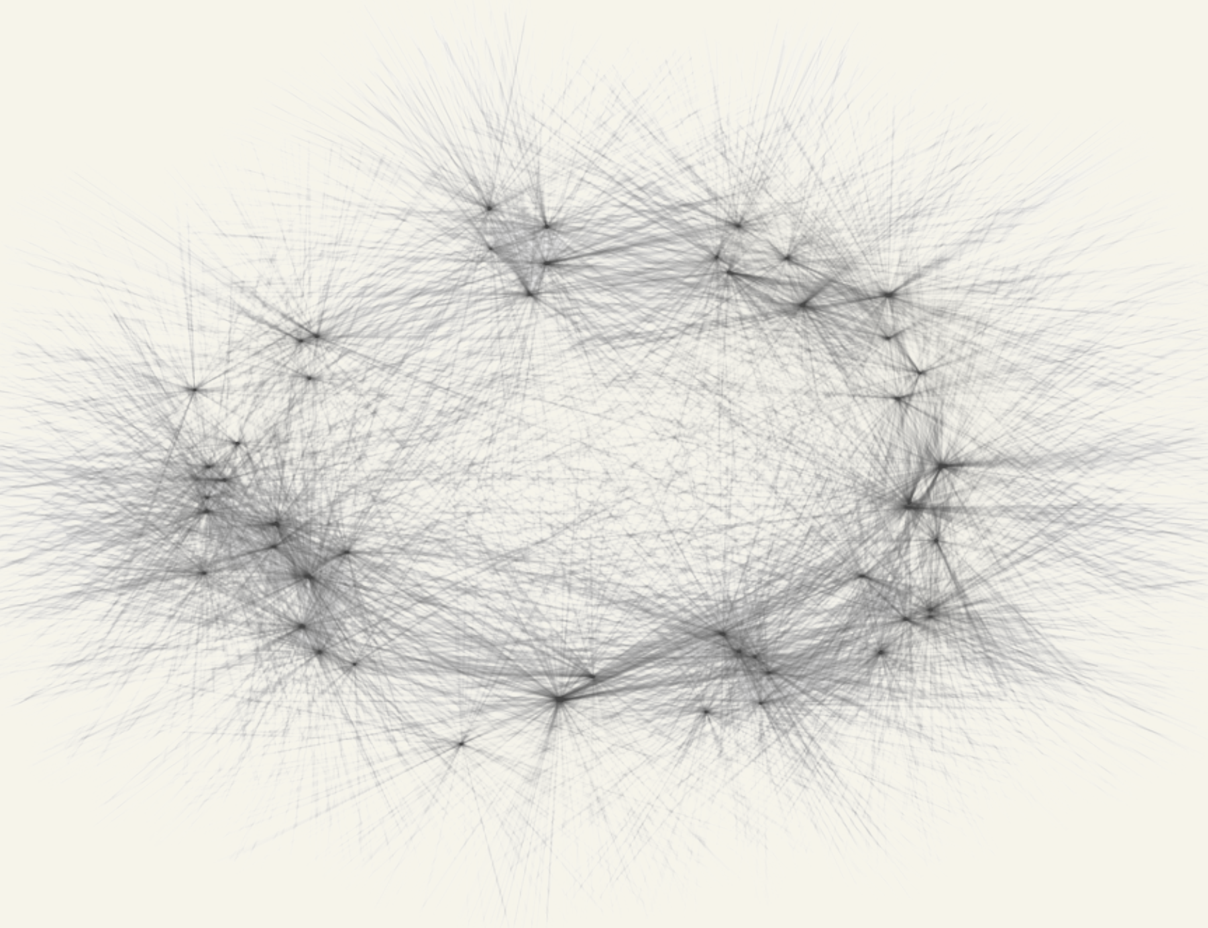
DENSITY: 2.03

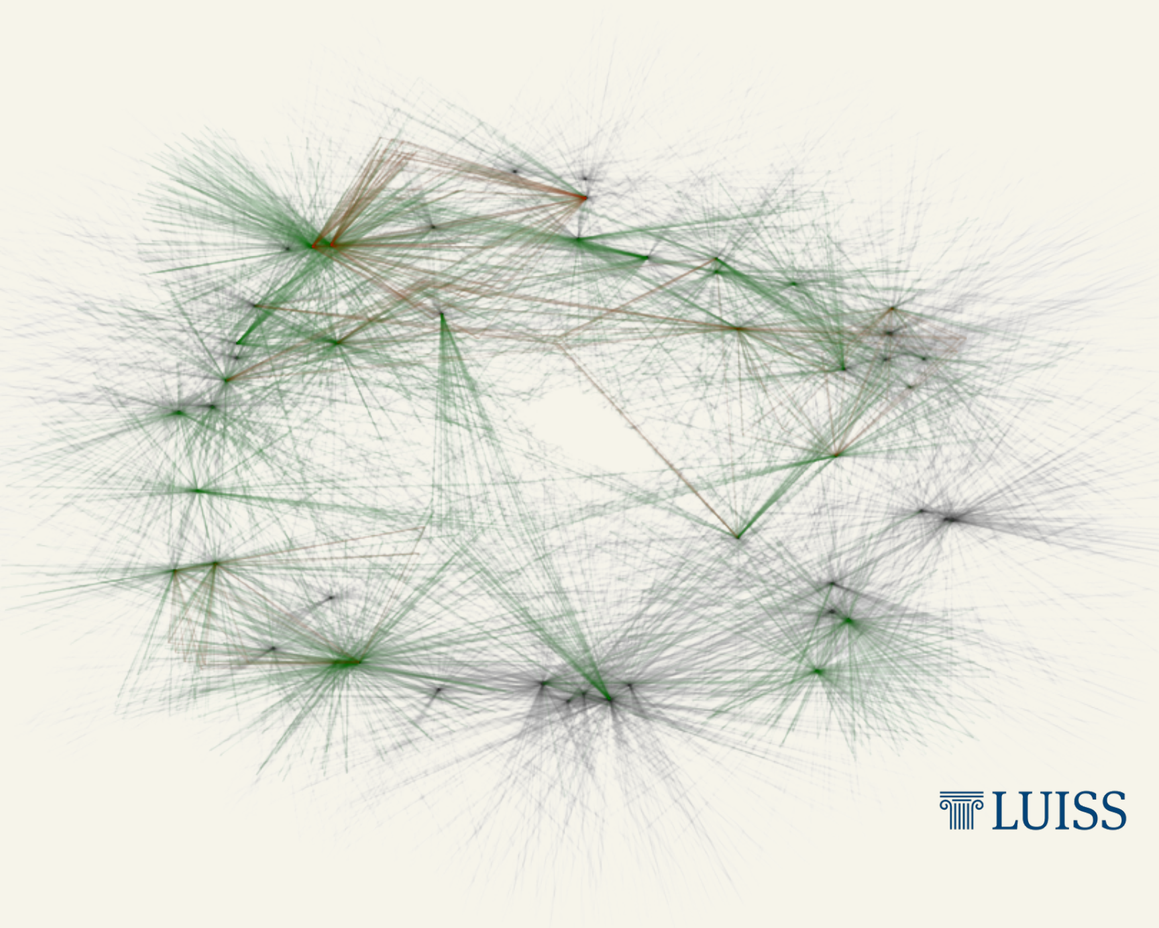
Average weight of the edges of the most "connected" nodes: 0.41

Average weight of all the edges: 0.51

LUISS

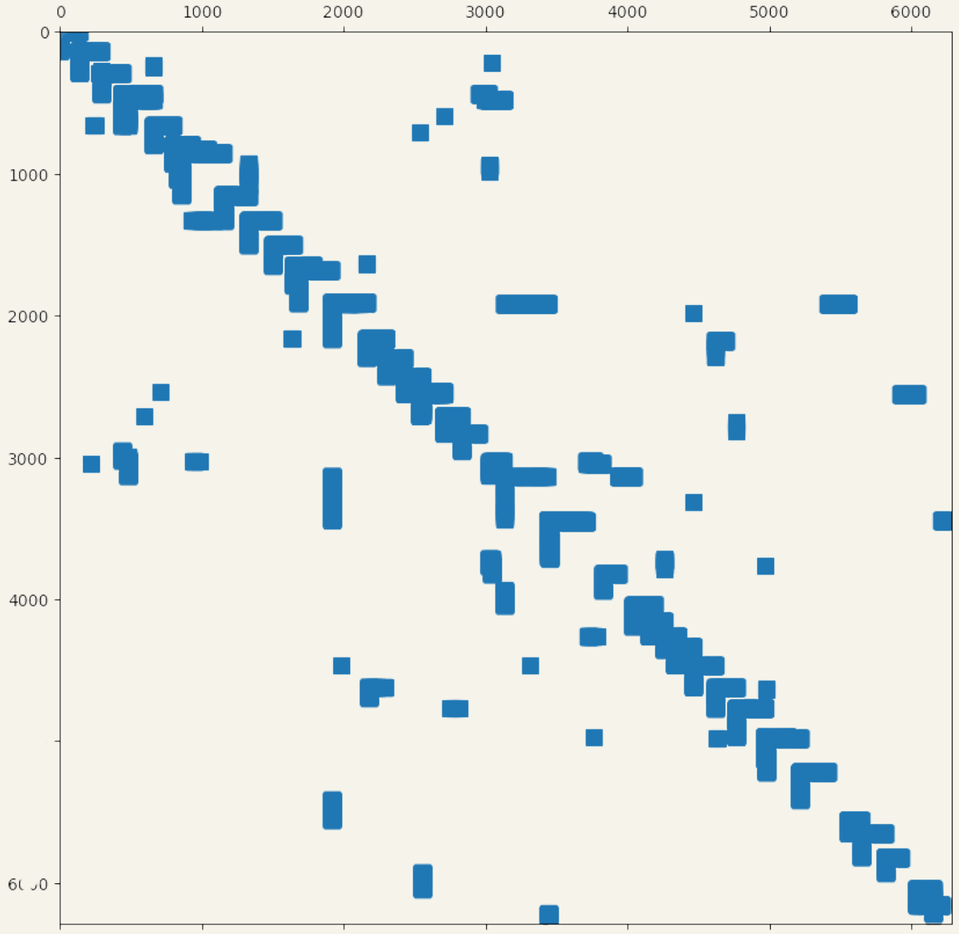
Network visualization - NETWORKX

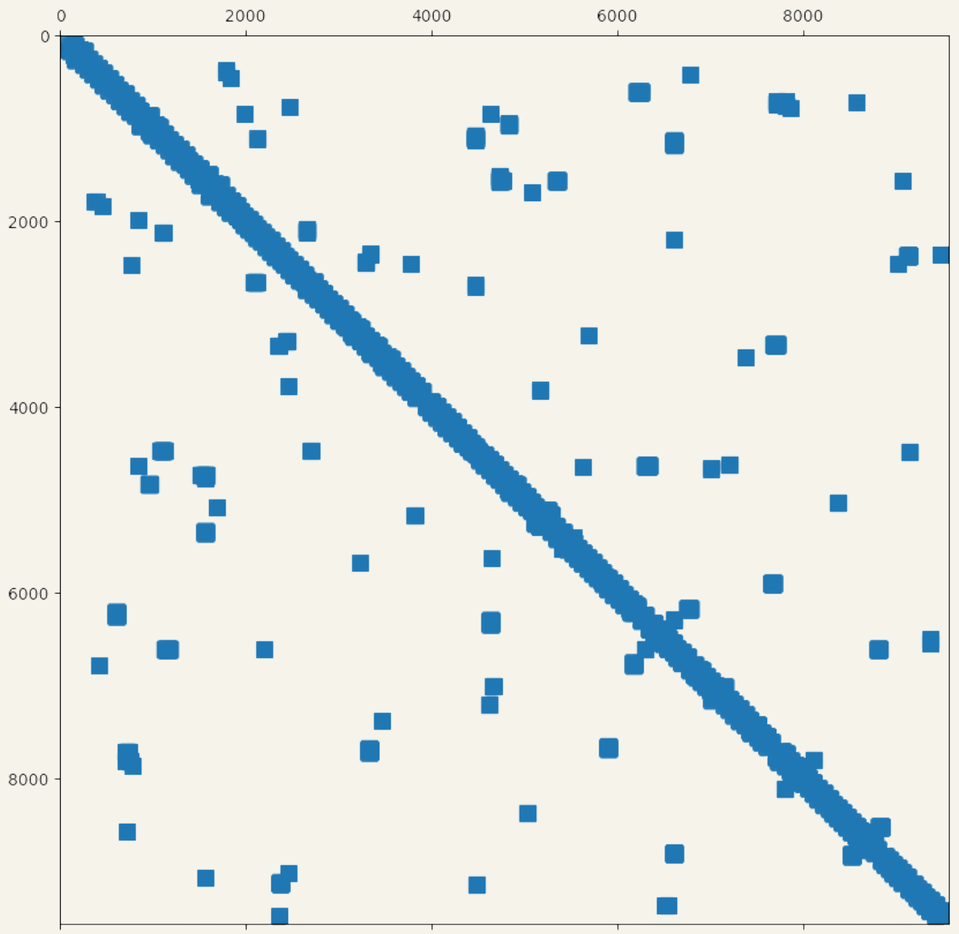




LUISS

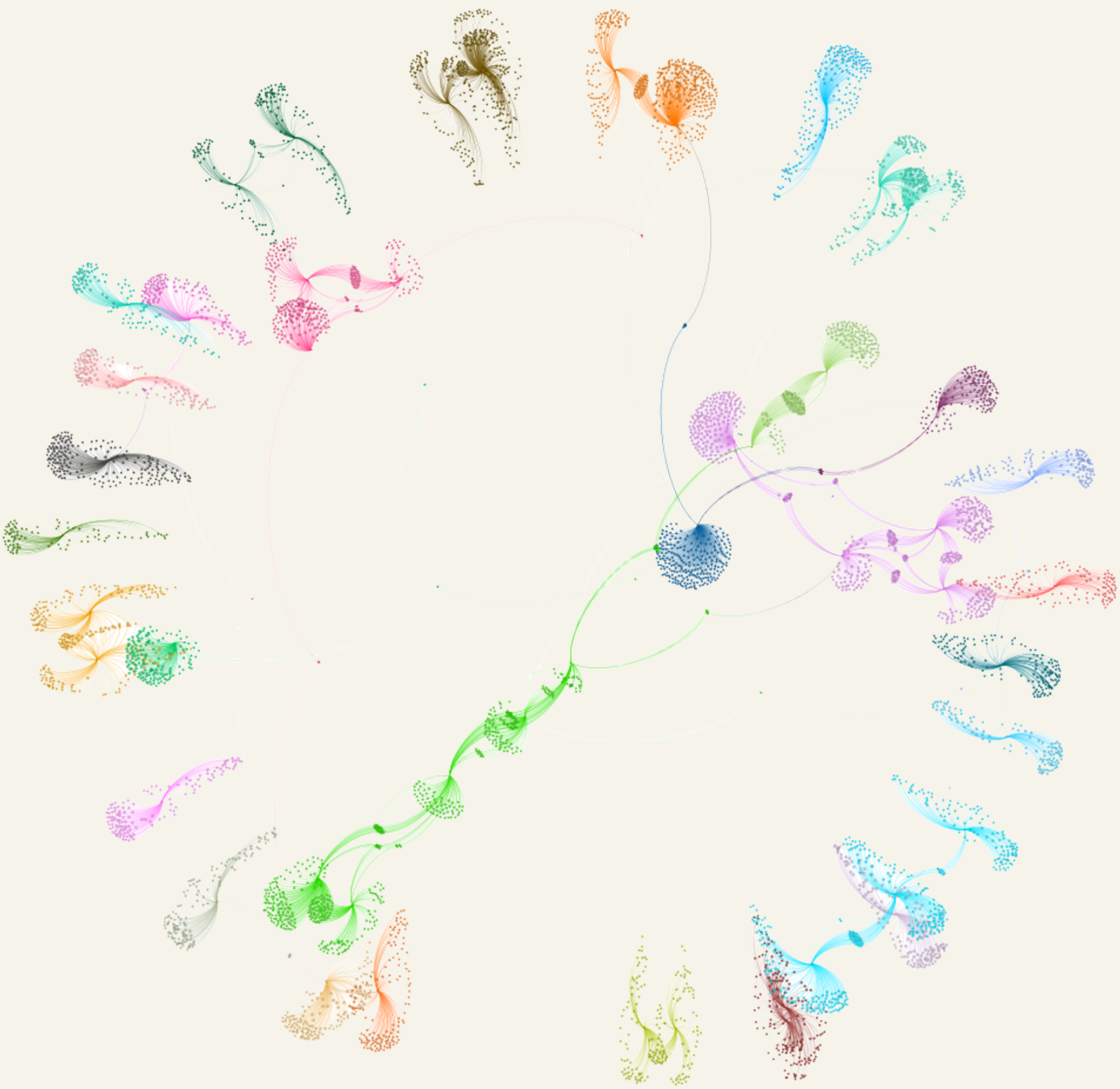
Network visualization - Adjacency Matrix





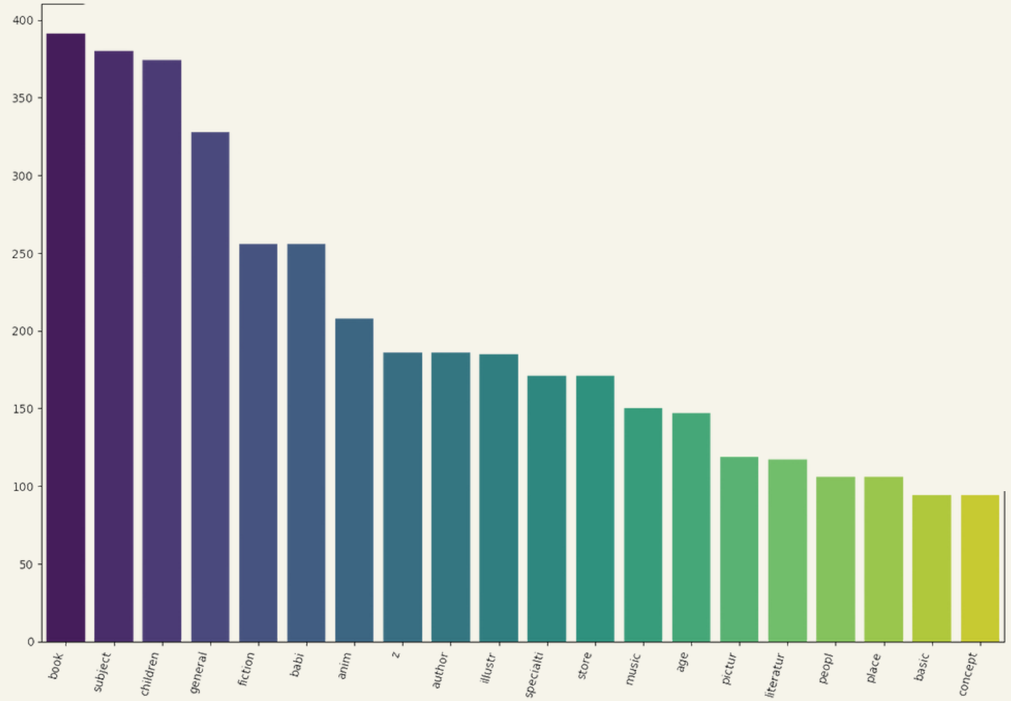
LUISS

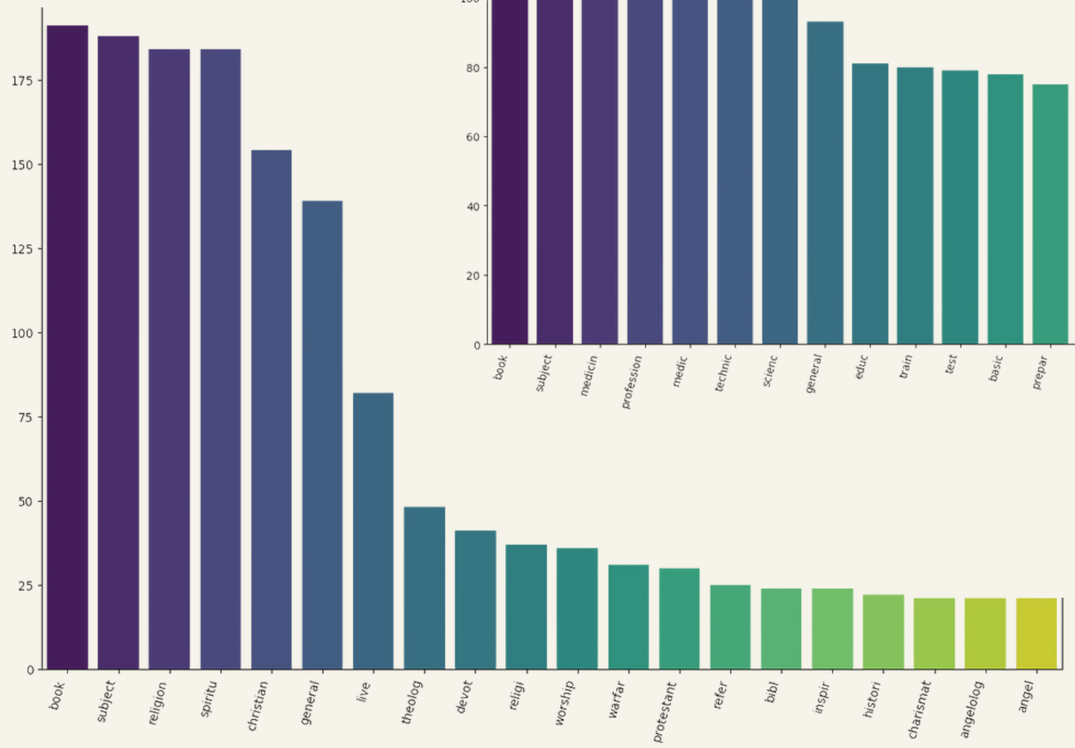
Network visualization - GEPHI

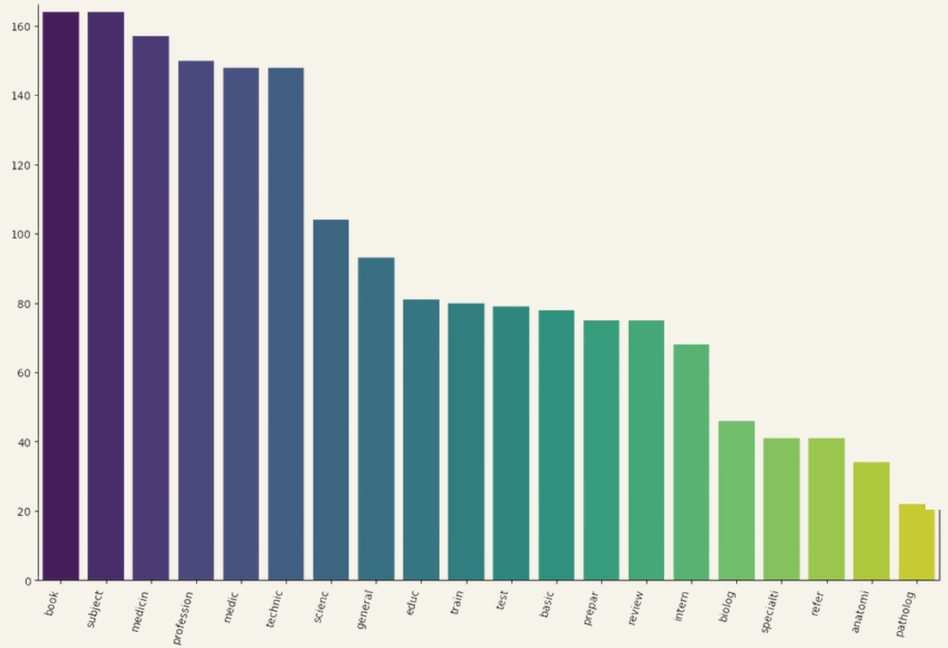


LUISS

Insights about communities 13 - 23 - 26

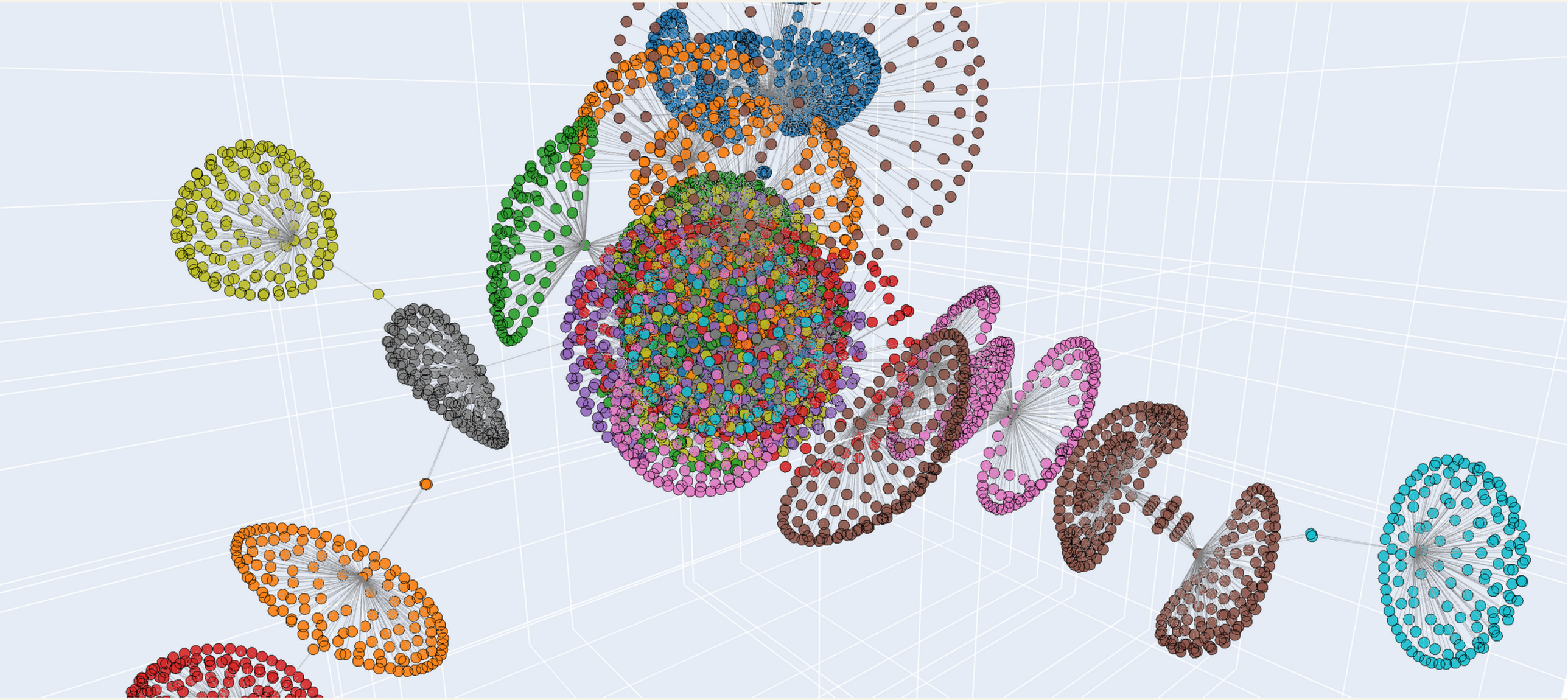






LUISS

Communities visualization - Network



LUISS

Recommendation System

How do we make other book
Recommendations to this customer, based on
the book copurchase data that we have?

Take all books that were ever copurchased with this book, and recommend all of them. However, the degree centrality of nodes in a product co-purchase network can typically be pretty large.

Examine the metadata associated with the book that the customer is looking to purchase

LUISS

Recommendation System

Networkx Eco Graph:
Returns induced subgraph of neighbors centered at node n within a given radius

Island Method:
Filter to the most similar books

Metrics:
Avg rating and total reviews

```
ASIN =  0812580036
Title =  Oliver Twist (Tor Classic)
SalesRank =  8750
TotalReviews =  105
AvgRating =  4.0
DegreeCentrality =  124
ClusteringCoeff =  0.55
```

```
Top 4 recommendations:
ASIN      Title   SalesRank       TotalReviews    Avgrating       DegreeCentrality
steringCoeff
0679420738', "A Tale of Two Cities (Everyman's Library (Cloth))", 36892, 347, 0.0,
0.61)
1582790795', 'Signature Classics - A Tale of Two Cities', 1185153, 347, 0.0, 5, 0.

0192545043', 'Tale of Two Cities (New Oxford Illustrated Dickens)', 671865, 347, 0
5, 0.61)
0140437304', 'A Tale of Two Cities (Penguin Classics)', 504413, 347, 0.0, 5, 0.61
```

How user behavior varies within user communities defined by a recommendation network?

LUISS

THANK YOU
FOR THE ATTENTION!

Lorenzo Meloncelli

Martina Manno

Valerio Schettini

LUISS