Master in Artificial Intelligence

Illinois Institute of technology

CS 579 - Online Social Network Analysis

Assigment 2

Anna Monso Rodriguez

A20653296

September 18, 2025

Contents

1 Chicago communites areas dataset

The first dataset contains information about 77 officially defined community areas in Chicago. Each area serves as a node in a network, with the edges representing shared physical boundaries between areas.

1.1 Data Cleaning

To construct a meaningful graph from this network, a precise criterion for adjency is required. In the analysis, two communites are considered adjacent only if they share a boundary of non-zero lenght. In four communities meet at a single point, diagonal connections are not counted as an edge. Only communities with a tangible, shared boundary are connected in the network. This assumption ensures that the resulting network reflects contiguous neighborhood relationships rather than incidental geometric intersections.

1.2 Labelled visualization of the network

In the visualization of the network we can clearly see the distributions of the communites.

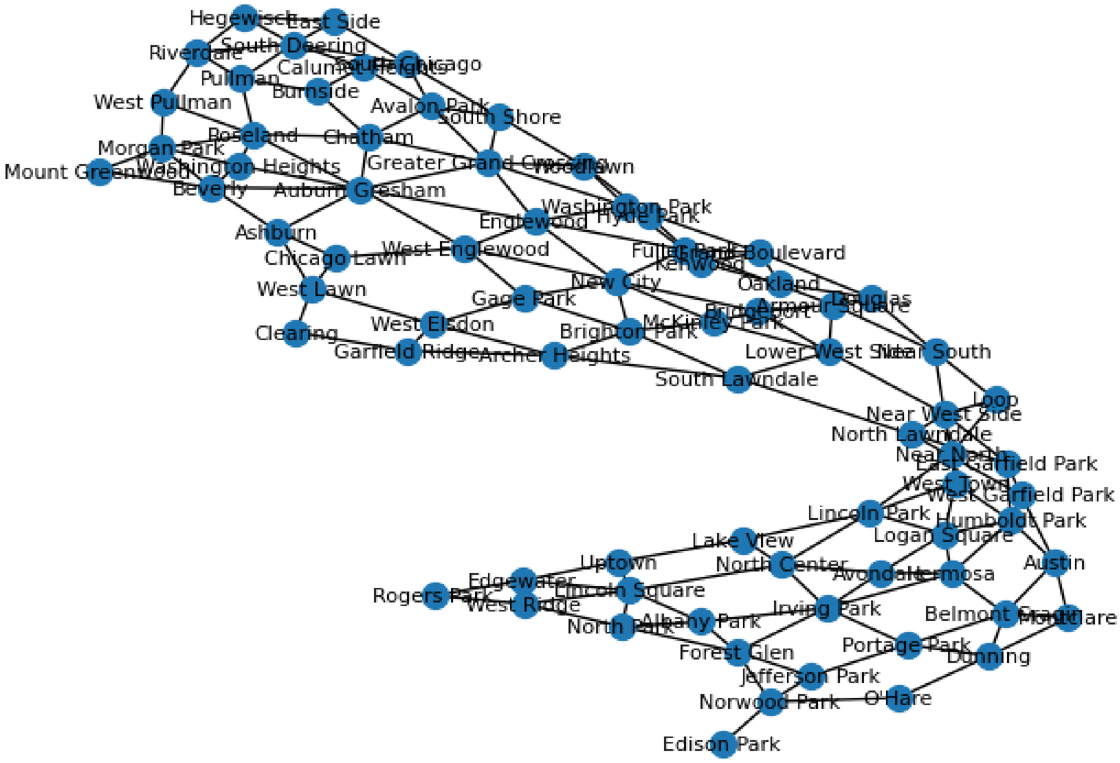

Figure 1: Labelled visualization of the Chicago communtites network

1.3 Plot of degree distribution

The degree distribution plot reveals that the majority of communities share boundaries with four other neighborhoods, indicating a typical pattern of adjency in the city's geography. Most of the nodes in the network cluster around this degree. However, there is a notable exception: Edison Park, which is uniquely situated with only one neighboring community, is reflected by a single node with degree one. On the other end, Auburn Gresham stands out as the community with the highest connectivity, adjoining eight different neighborhoods.

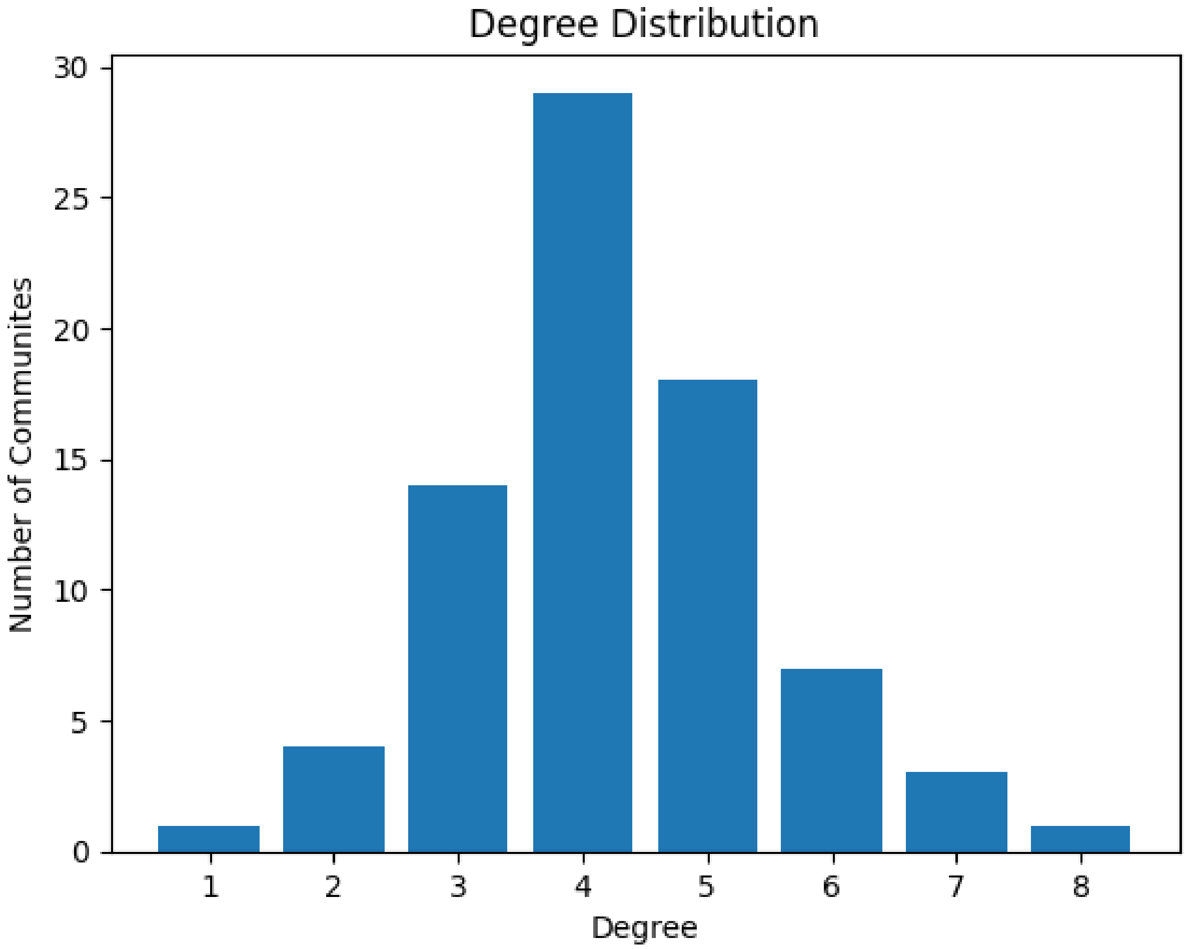

Figure 2: Plot of degree distribution of the network

2 Class participant dataset

In this dataset we are going to analyze the student of a class, having into account their email to identify them, the degree they are enroll to, the department this degree is in, their neighborhood, the computer languages they know, the languages that they know how to speak, their hobbies and their student clubs affiliations.

2.1 Data Cleaning

To accuretely represent this dataset as a graph, thorough data cleaning was essential. For fields containing multiple elements, the entries were standardized into vectors of strings to ensure consistency.

- **Neighborhood:** Each neighborhood name was capitalized. Entries containing a comma, indicating broader location information, were simplified to keep only the relevant neighborhood, city or country. For example, ”Seul, Korea” was normalized to ”Korea” and ”Bronzeville, Chicago” was reduced to ”Bronzeville”.
- **Computer languages:** Variations and common misspellings were consolidated into standarized groups. Examples: java_variations = [”java”, ”javascript”, ”javascipt”, ”js”] , c_variations = [”c”, ”c”]
- **Spoken languages:** Similar normalization was applied to spoken languges to merge variants and abbrevations such as: english_variations = [”english”, ”en”, ”eng”] or kannada_variations = [”kannada”, ”kanada”]
- **Degree programs:** Degree names were standarized by mapping common variations to a uniform descriptor. For instance: ”bachelor’s computer science”, ”bsc computer science”, and ”bachelors computer science” were unified as ”Bachelor’s in Computer Science”. Similarly, master’s and other degree variants were mapped to consistent titles.
- **Hobbies:** Given the large variety of hobby descriptions (200 unique entries), related hobbies were clustered into groups such as reading, movies, music, gaming, sports, traveling, anime, cooking, writing and photography. This grouping enabled a more tractable categorization of student interests.
- **Student clubs:** Cleaning involved removing extraneous characters like words in parentheses and ”@” symbols to standardize club names for easier analysis.

This cleaning process ensured uniformity and improved the quality of the graph representation and subsequent analyses.

2.2 Bipartite Graph

After cleaning and structuring the dataset, it is represented as a bipartite graph. A bipartite graph consists of two distinct sets of nodes where edges connect only nodes from different sets. In our case, one set contains the students, and the other set contains all the other entities mentioned earlier.

Due to the complexity and size of the dataset, three different visual representations were created to better illustrate the data distribution and relationships between students and these entites.

2.2.1 Representation 1

The visualization represents the bipartite graph with the students on one side and all other entity categories on the opposite side. It highlights the density of connections, showing how many students associate with each entity. This representation help us understand there meaning of a bipartite graph.

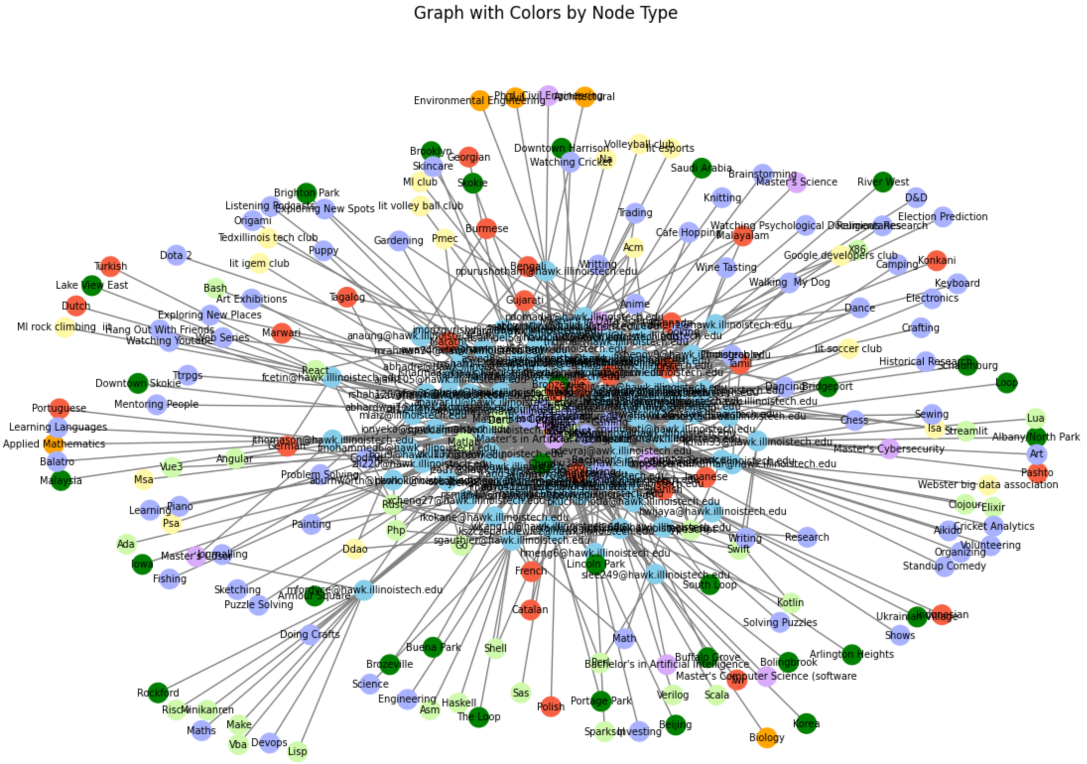

Figure 3: Graph with colors by Node Type

2.2.2 Representation 2

The second visualization applied color-coding to differentiate types of nodes, aiding in quick identification of the various entity categories and their connections to students. This makes it easier to spot clusters and dominant themes between students.

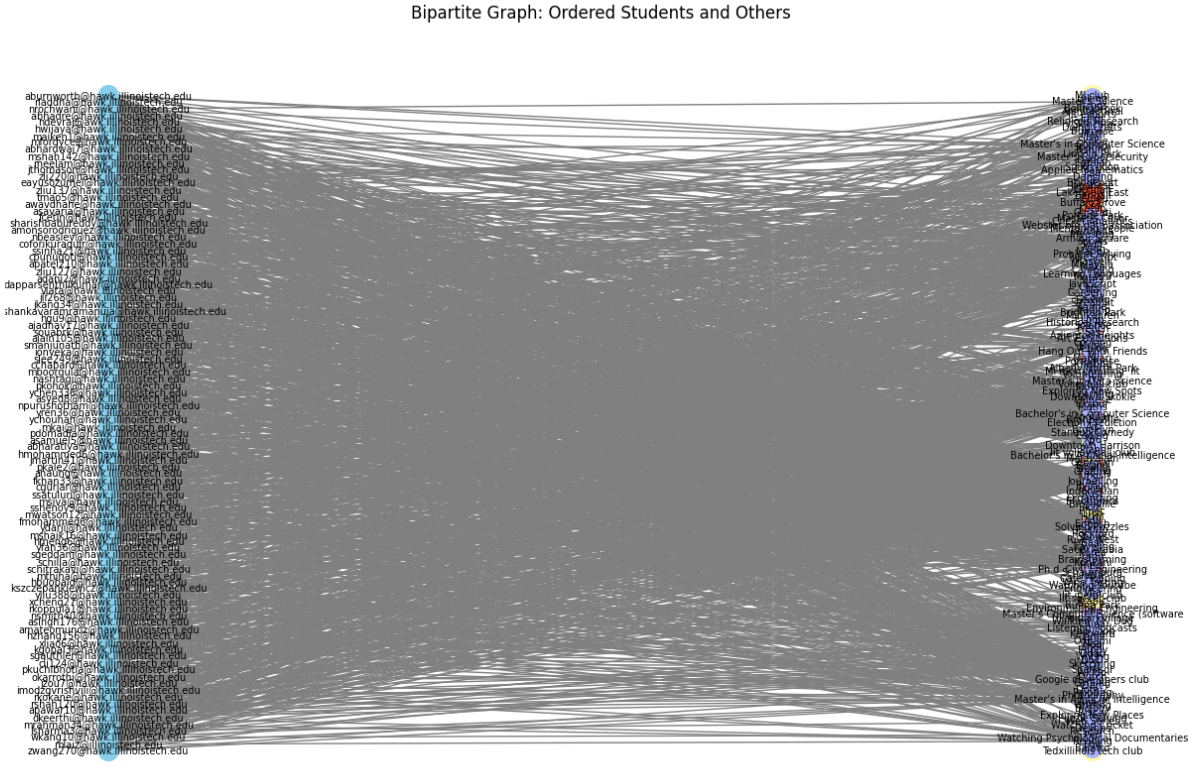
Bipartite Graph: Ordered Students and Others
aburnworth@hawk.illinoistech.edu
zwang270@hawk.illinoistech.edu
Master's in Computer Science
Religious Research
Applied Mathematics
Learning Languages
JavaScript
Historical Research
Hang Out with Friends
Bachelor's in Computer Science
Solving Puzzles
Google developers club
Master's in Artificial Intelligence
Watching Psychological Documentaries
Tedxillinoistech club

Figure 4: Graph separating students on the left and other entities in the right

2.2.3 Representation 3

The third graph provides a structured overview with an ordered layout, that separates students nodes and entity nodes explicitly while maintaining all connections. This help to visualize all nodes.

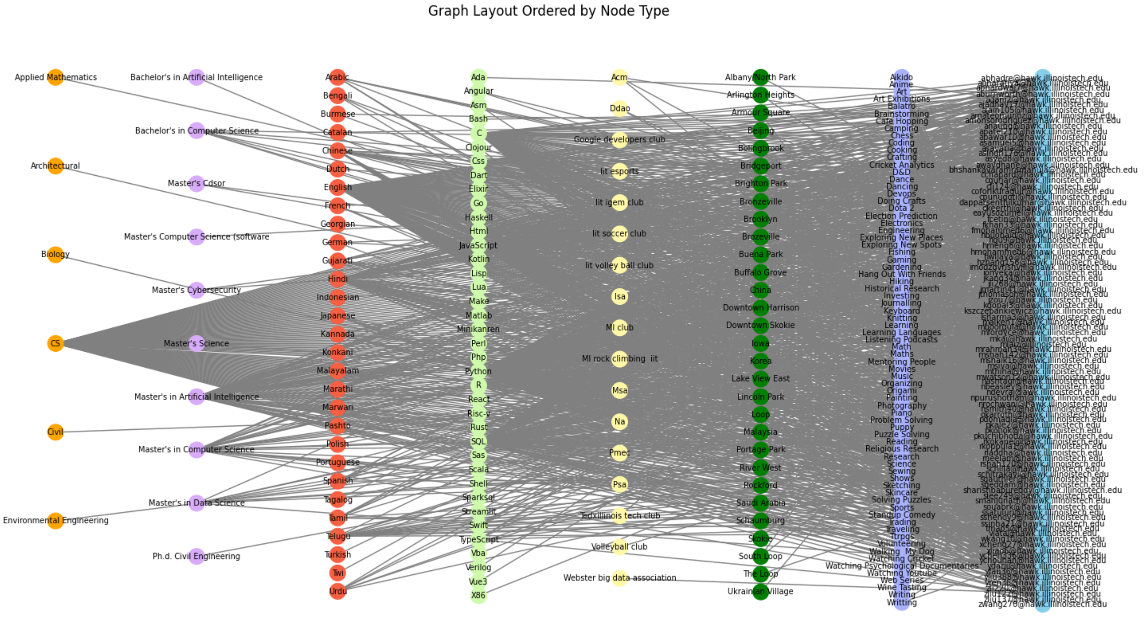

Figure 5: Graph with layout ordered by node type

2.3 Unimodal graph

Next, we analyze the network through a different perspective by constructing a unimodal graph. This graph is created as a projection of the bipartite graph, but only student nodes are considered. In this network, edges connect students who share at least one common entity.

2.3.1 Representation

This visualization highlights how students are interconnected based on shared attributes. Given that there are approximately 100 students. The high density of connections makes it challenging to interpret specific relationships or clusters by visual inspection.

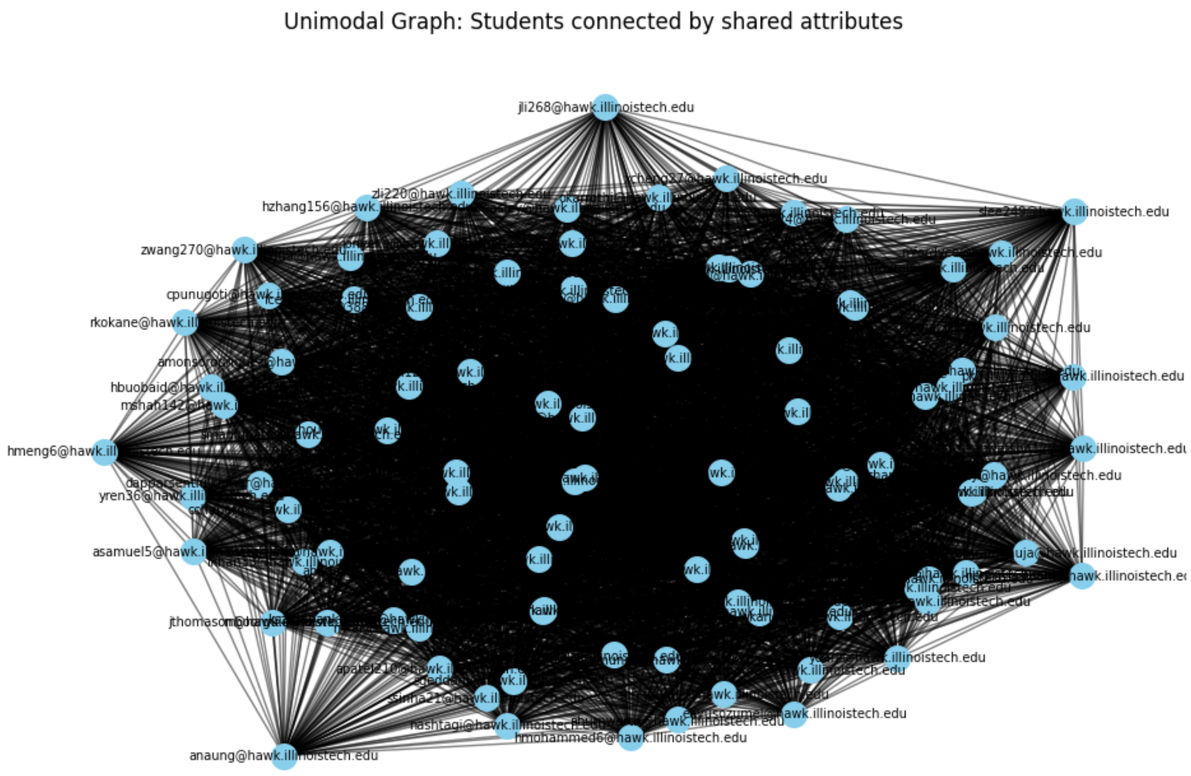
Unimodal Graph: Students connected by shared attributes
jli268@hawk.illinoistech.edu
hzhang156@hawk.
zwang270@hawk.i
cpunugoti@hawk.i
rkokane@hawk.ill
hbuobaid@hawk.il
mshah142@hawk.i
hmeng6@hawk.ill
yren36@hawk.illi
asamuel5@hawk.i
apatel210@hawk.i
ssinha21@hawk.il
nashtagi@hawk.illinoistech.edu
hmohammed6@hawk.illinoistech.edu
anaung@hawk.illinoistech.edu

Figure 6: Enter Caption

2.3.2 Description from bipartite graph to unimodal graph

To analyze relationships among students based on shared attributes, we project the original bipartite graph into a unimodal graph containing only student nodes. The process involves these steps:

- Collect all nodes classified as students.
- For each non-student attribute node, identify all student connected to it.
- For every attribute, fully connect all students sharing it in the unimodal graph.

2.3.3 Plot of degree distribution

Because all students share at least one universal attribute, such as speaking english, the graph is almost complete, leading to every student connecting to every other student. This is why the degree distribution plot features a single peak at degree 97, representing the maximum degree for 98 students.

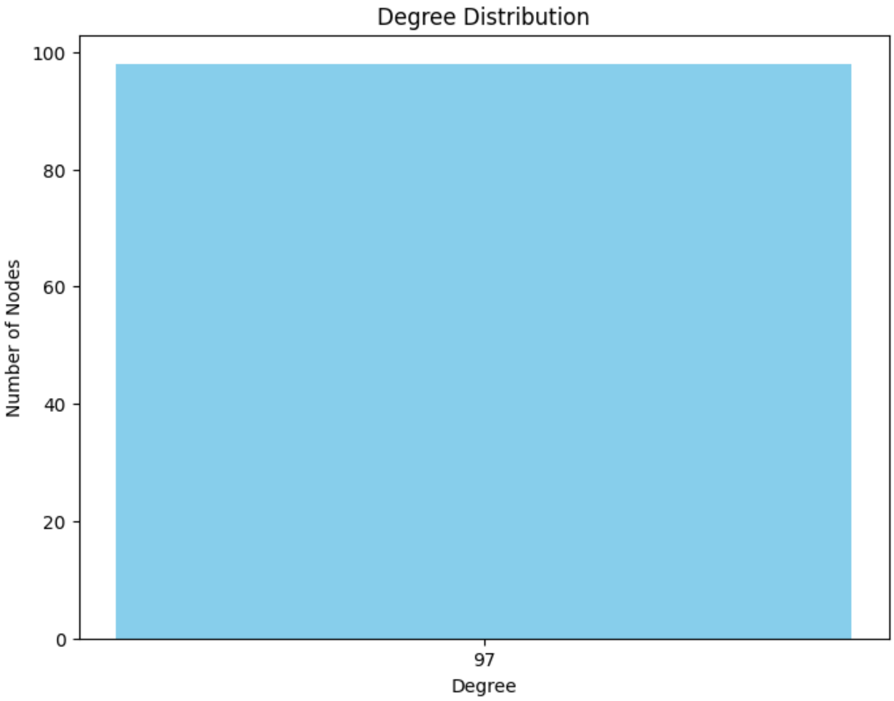

Figure 7: Enter Caption

2.4 Comparation between graphs

Both graphs reflect how entitites are interconnected through shared data but at different level. The bipartite graphs represent two distinct types of nodes, students and attributes (representated with different colors depending on the type) where edges connect students to these attributes. The degree distribution in bipartite graphs shows how many connections each node has within its own set. This kind of graph are useful to identify popular attributes or key connectors.

In contrast, the unimodal graph is a projection focusing on students. Here, edges represent shared attributes between students, effectively connecting students to each other if they have something in common. Unimodal projections allow us to explore direct relationships within one node set. Common analyses include community detection. centrality measures, and clustering. However if the dataset has a lot of different entities and these entities are really common, the graph loses a little bit its function because all nodes are going to be connected between them. All nodes are going to have N-1 degree.

The observations allign with expectations. The bipartite graph naturally separates types, making it easier to identify attribute popularity. The unimodal graph's density and degree peak come from share universal traits conneting most or all students , making the graph highly interconnecting. The complexity in the unimodal graph illustrates the need for additional analytic techniques like filtering by attribute types to gain more granular insights.

3 Assignment details

This assignment was completed using Python, and several libraries for data manipulation, network analysis, and graph visualization:

- Data Processing: Data cleaning, normalization, and structuring were accomplished using pandas for tabular operations and the collections library for efficient grouping and counting tasks.
- Graph construction and analysis: All graph related task were completed using the NetworkX library. I have use the official documentation and some tutorials for details and examples. NeuralNine Youtube video tutorial — NetworkX official documentation and tutorial
- Visualization: Graphs and degree distribution plots were generated using matplotlib. Matplotlib is a library commonly use in data science work.