Vibha Mandayam, Kiet Nguyen, Zhuldyz Ualikhankyzy

Professor Wu

CSDS 234

18 December 2022

Amazon Co-Purchasing Data Final Project Report

**Abstract**

A subset of Amazon purchasing data from 2006 was selected to carry out the co-purchasing analysis. The scope of the project was scaled down to products from the group “Book” and with sales rank above 5000. Using a set of schemata derived from the unstructured text data, the graph network of products were created to be further analyzed for patterns. Upon the processing of the data, the graph network of selected products was created. It was found that a minority of products have many connections with other products, making their subgraph networks densely connected. Book titles with related themes create dense clusters, as do books of the same series.

**Background and Motivation**

From being challenged by the initial skepticism of ordering goods from a far-away distributor, online shopping now has been integrated as an irreplaceable purchasing method of the consumer mass. Booming in popularity, the e-commerce market has become more competitive and lucrative than ever with a high estimate at roughly 905 billion U.S. dollars in 2022 (Statista). There are multiple advancements in technology, logistics, and manufacturing that contribute to the success of online shopping. But for any retailer who wants to get ahead in this race, the key is to be able to harness the powerful insights of the purchase data, which will ultimately appeal to more prospective customers. To that end, one of the biggest obstacles is to develop an effective schema that enables fast and accurate data retrieval. In addition, the ability to visualize and glean information from co-purchasing relationships has become increasingly important in order to comprehend the topological structure of purchases on Amazon. By understanding the manner in which products are often purchased together, online retailers can determine how best to optimize recommendation algorithms in order to increase profits.

Previous research has shown that such issues as finding and fixing data challenges and long system response time become particularly challenging when the object of investigation is a big data set (Cai et al.) Before being processed, it is important to find and correct errors encountered in data collection and entry or find a way to address the error in later stages of data processing. However, while working with big data, it is often difficult to find a testing method to identify the errors due to inefficiency of manual search for every possible error-causing case. Long response time of algorithms for big data is the issue that can be resolved in the presence of parallel computing systems, which is often not available for most small-scale research groups.

**Problem Formulation**

The main objective of this study was to develop a way to visualize the density of co-purchasing habits, particularly among the top 5000 most-sold books on Amazon in 2006. In order to do so, the Stanford University SNAP program’s Amazon metadata was utilized to create social networks depicting the topographical structure of the node co-purchasing relationships. Given SNAP’s Amazon metadata as unstructured text data, the process began with parsing said data into a usable format. The recorded metadata contained information such as the title of the product, its sales rank, a list of similar products that are often co-purchased with this product, a detailed categorization of the product, and information about the product’s reviews. Primarily focusing on the products’ sales ranks and lists of similar products, product insights were gained via graphical representations of the networks of similar products. The degree of each product node, meaning the amount of similar products it contains in combination with the number of appearances it makes as a similar product, was also analyzed to determine information about the node importance in the topological structure of co-purchasing in Amazon.

**Algorithm Description**

Firstly, the unstructured .txt file was parsed using Python row by row to be converted into a raw data frame with one attribute, where each row represents a line of the .txt file. To optimize the queries on the dataframe in the future, three schemas were designed to semantically separate the attributes as listed below:

Products<id, asin, title, group, salesrank>

Similarity <id, sim1, sim2, sim3, sim4, sim5>

Review <id, cus1, cus2, cus3, cus4, cus5>

*where id represents the product id, asin represents the Amazon Standard Identification Number, title is the name of the product, simn describes the ASIN of a similar product, and cusn describes a customer id.*

To corroborate the accuracy of the parsing algorithm, exploratory data analysis was carried out to count the occurrences of ID parameters. In total, there were 542684 ID entries observed in the parsed metadata; however, this number does not demonstrate the total occurrences of all other parameters since some of the IDs were of discontinued products. For all parsing algorithms, it was crucial to carry out sequential rather than parallel evaluation of row values due to the hierarchical relationship between the “ID” attribute that comes before all the other attributes of a single entry, such as “ASIN,” “Sales rank,” “Customer,” etc.

**Creating schemata**

*Products schema*

Firstly, “name” and “value” attributes were added to the initial schema, where “name” indicated the type of the attribute and “value” represents the value the attribute stored. An empty dataframe with id, asin, title, group, salesrank columns was created. The values of “name” columns were evaluated against id, asin, title, group, salesrank attributes in the given order. If the row being iterated contained id information, then the iterator of the newly created table was moved to the next row. If the value of the “name” column matched any of the columns in the schema, they were added to the corresponding columns of the row being iterated. There was one empty row in the head of the dataframe due to ID iteration specifics; however, dropping the first row and re-indexing the column was a simple way to handle this incongruence.

*Similarity schema*

Firstly, a data frame with id and similarity columns was created. Then, sim1, sim2, sim3, sim4, and sim5 columns were added to the dataframe. Every value of the“value” column of the data frame, where the “name” column is “similarity”, was split by comma into the five newly created columns. All rows which were not either id or similarity were dropped. All values of sim1, sim2, sin3, sim4, and sim5 columns of a specific row, where shifted to the preceding row with “name” equals to “ID.” Thus, we matched the id with similarities. Later, the rows which did not have the “name” column with the equal to “ID” were dropped.

*Review schema*

Firstly, only the rows with id, review (number of reviews), and customer were selected. Later, the process similar to the algorithm described for the similarity schema was carried out with the customer being the repeated element instead of similarity. However, this table was not used for the future analysis due to inconsistency spotted in declaration of reviews in similarities across the text file. Due to misplacement of flags like “review” and “customer” the final table did not demonstrate the true hierarchy between product ids and the customers those products were purchased by. Thus, due to mistaken data, the review schema was not used to gain product insight.

*Matching products with their similar products*

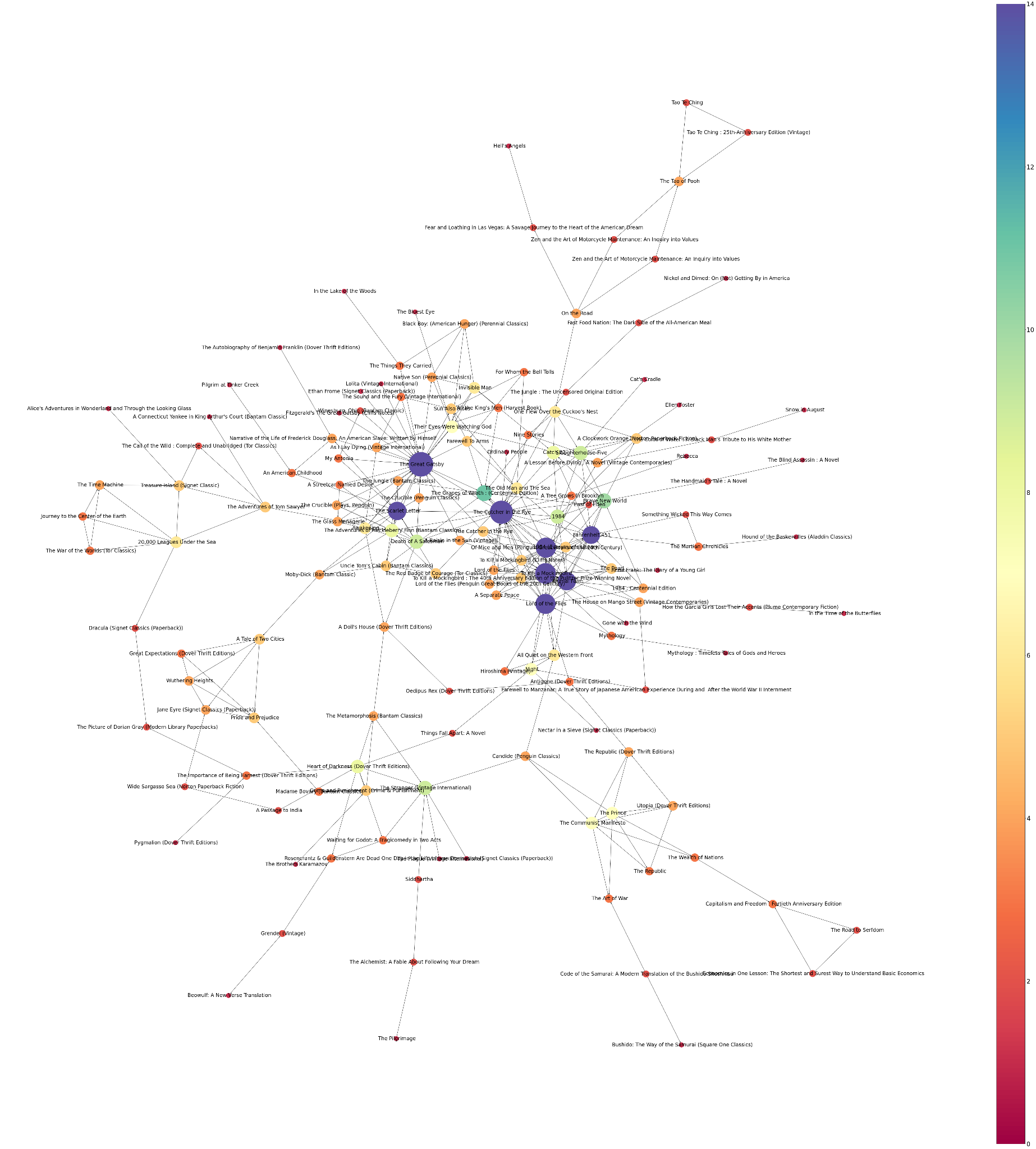
The similarity schema contained the ID of the product and ASIN numbers of products often co-purchased with the product in question. In order to replace the ASIN values with ID values, the similarity table was divided into five schemata, each containing the ID columns and one of the similarity columns. Before this step, however, the table was reduced to the scale of our investigation by selecting only the products from the group “Book” and with sales rank above 5000 from the Products schema. In total, there were 74554 unique ID values returned by this query, and they were used in the building graphs.

The original Products schema was utilized to create a new schema Product' that only contained the id and asin attributes. Product’ was then merged with all every new schemata created from the Similarity schema on the attribute asin (asin in Product, and similarity renamed into asin in each of the smaller Similarity schemata). Hence, every new schema contained the id of the product and pid (id in Product schema). The same process was repeated for all five columns of the Similarity schema. The resulting columns were later merged into one dataframe that is very similar to the Similarity schema, but now with id values for each similar product instead of asin values. The resulting schema with 6 attributes, where 5 belong to the same group (similar product), was reduced to a schema with 2 attributes. This caused consecutive repeated values in the rows for the id attribute but simplified the previous schema and made it easier to process as a graph.

**Converting schemata to graph network**

Utilizing the aforementioned dataframe containing repeated id values along with each of their similar products on consecutive rows, the in-degree and out-degree of each product could be calculated. The out-degree of the product was calculated by counting the number of similar products assigned to the product id. The in-degree of the product was calculated by counting how many times a certain product id was encountered as a similar product for other products. Both the in-degree and out-degree were aggregated using the group-by function in Python. A separate table containing each product id and its total degree (the combination of its in-degree and out-degree) was then created. This table was utilized to determine the products with the highest degree of co-purchasing.

Finally, a graphical representation of the co-purchasing behavior related to node 577 was created to demonstrate the topological structure of Amazon’s co-purchasing relationships. Node 577 was utilized due to its relatively complex structure that also fit within the computational constraints of the machines being used for the study. As seen in the figure below, node 577 of the title “One Flew Over the Cuckoo’s Nest” has a relatively low degree, as shown by its size and color, but the nodes it is connected to have several further connections, creating quite a complex network representing many classic novels.



*Figure 1: Graph network for node 577, a book titled “One Flew Over the Cuckoo’s Nest”*

**Experimental Study**

To analyze the node importance of the graph network pictured in Figure 1, a histogram was created to plot the distribution of the node degrees. As seen in Figure 2, very few nodes have high degrees, meaning that the majority of nodes are connected to only a select few other products. From this, it can be concluded that the products with the highest degrees of similarity are most integral towards creating proper product recommendations since they often appear to be purchased along with other products.

|  |  |
| --- | --- |
| *Figure 2: Degree histogram of nodes connected to node 557* | *Figure 3: Graph network for node 448, a book titled “Three to Get Deadly: A Stephanie Plum Novel”* |

Another graph network of interest was created using node 448; this network depicted in Figure 3 clearly depicts the purchasing relationship between books of the same series with the same author. Once again, the visualization of the relationship between these products can be highly insightful as to the purchasing habits of Amazon customers in that they tend to buy a series worth of books or books by the same author.

Network analysis visualization provided a fair understanding of how nodes within the network interact with one another. Furthermore, quantitative metrics opened the door for researchers to fundamentally differentiate networks, study their topologies, and ultimately transform the myriad of nodes and edges into meaningful insight for further implications of the dataset.

A good overall metric to begin with was the network density, which described the ratio of the edges in the network to all possible edges in the network. Network density provided a quick sense of how closely knit the network is. With the selected node and in its subgraph, the density of the network, which is on a scale of 0 to 1, was determined to be approximately 0.005. Thus, the network was not particularly dense.

**Centrality**

After gaining a fundamental understanding of the network, the next important step was to find which nodes were the most prominent. Within network analysis, a node could be denoted as important based on its measure of centrality. Although there exist many indicators of centrality, for the scope of this project, degree centrality, closeness, betweenness centrality, and eigenvector centrality were chosen as appropriate centrality measures.

Degree centrality is a measure of centrality in a network that is based on the number of connections a node has. It is calculated as the number of edges incident upon a node, i.e. the number of connections the node has to other nodes. Nodes with a high degree of centrality have many connections and are considered to be more central or influential in the network (Figure 4). Here, without network, the highest degree centrality was found in the book with id 199628 “The Great Gatsby” with degree 38, followed by id 98756 at a degree of 31 and id 502784 at a degree of 29.

|  |  |
| --- | --- |
| *Figure 4: Graph degree centrality of each node* | *Figure 5: Graph eigenvector centrality of each node* |

After identifying the hub of the subgraph, Eigenvector centrality (Figure 5) was determined. This centrality specifically focused on the combination of a node’s edges and the edges of that node’s neighbor. Thus, besides showing the connectedness of hubs, eigenvector centrality also showed which hub other nodes are connected to. With a scale from 0 to 1, Eigenvector centrality unveiled which nodes in the network could quickly get information to other nodes. Simply put in this context, books with higher eigenvector centrality were easily bought together with other books even if the book was not a hub. The eigenvector centrality graph appeared to have quite a few similarities with the degree centrality graph.

In contrast to the other two metrics, betweenness centrality did not take into account how many edges each given node or group of nodes had. The concept of "betweenness centrality" examined all shortest routes through a given node (see above). Consider that betweenness centrality would take longer to compute than other centrality metrics (although it would not be an issue in a dataset this size) since it must first determine every feasible shortest path in the network. Finding nodes that link two otherwise dissimilar regions of a network was very easy with betweenness centrality, which is also represented on a scale from 0 to 1.

Closeness centrality was calculated by taking the average length of the shortest path between nodes and all other nodes in the graph. It was important to note that closeness centrality was calculated using only the number of edges in the graph, and did not take into account the weights or strengths of those edges; in other words, a node with many weak connections may have had a similar closeness centrality to a node with fewer strong connections. For the subgraph from node “The Great Gatsby,” the closeness centrality was almost similar across all the nodes.

|  |  |
| --- | --- |
| *Figure 6: Graph betweenness centrality of each node* | *Figure 7: Graph closeness centrality of each node* |

**Prediction and metrics level of importance**

Although these quantitative metrics provide a deeper knowledge of the hub nodes and their influence within the network, these metrics are purely descriptive of the topological structure and less suggestive towards business decisions. In order to determine the level of influence of the metrics involved in this network analysis, a regression model was formulated to provide more business-centered suggestions. The data was prepped in the following manner in order to fit in the model: A data frame was prepared to store all the found metrics with the identifier of the books. Those included the centrality metrics found above along with the hub score and authority score as determined by the in-degree.

Given the dataset, a Poisson regression model was selected based on its versatility. Salesrank was then selected as the outcome of the predicting model. With higher sales rank, the book would be interpreted as more financially lucrative, hence prioritizing in finding how the network can help determine its success in claiming high rank.

|  |  |
| --- | --- |

*Figure 8: Poisson Regression Results*

Fitting the data into the regression model, the summary results containing critical indicators were obtained. First, interpretation of the model started with the value of the intercept.At 8.67, this displayed that without any co-purchased, lone purchased books still earned a salesrank. With the understanding that the closer the sales rank to 0, the higher its purchasing rank in the data, the most influential metric can be determined to be the centrality as it would decrease the numeric sales rank by the highest power among all the other quantitative metrics. On the other hand, authority score (in-degree) was found to be the most draw-back metric as it would increase the sales rank number, ultimately lowering the rank of the book. Within this context, books that are purchased as target products tended to have lower sales rank than products that are purchased as source products.

**Conclusion and Future Work**

The Amazon Co-purchasing data set contained critical information about customer purchasing behavior. Important steps were taken in preparing the data, including an extensive approach in constructing schemata and parsing to transform the initial unstructured text data to manageable and Python-friendly dataframes with the designated schemas. The project then mined the insight of co-purchasing traits by applying network and graph theory. Products that were purchased as co-product tend to have lower sales rank, which could be further implemented in a suggestion system for customers after they have bought a canon novel. A recommendation of more unique, yet to be discovered books would do better in enticing the overall purchase than recommending other high ranked books.

Though the findings applied for the selected subgraph from a chosen node, the same insight might not be applicable for other product types or a distinctly different set of data. To extend the barrier, it could be recommended to carry out more extensive research of larger, more representative datasets, which were unable to be carried out in this study due to limitations in computational strength. Ultimately, this project served as a foundational step into a well-grounded trend analysis not just within the Amazon co-purchasing data set but also other network-applicable data sets.

**Group Member Contributions**

Zhuldyz primarily worked through parsing the text file into the aforementioned schemata. Vibha worked through exploratory data analysis as well as graph network production. Kiet performed analysis on the graph network via centrality and regression. Finally, all group members collaborated and contributed to the final project and final presentation.

**Works Cited**

1. Statista. "Retail E-commerce Revenue in The United States from 2017 to 2027 (in Billion U.S. Dollars)." *Statista*, Statista Inc., 11 Nov 2022, <https://www.statista.com/statistics/272391/us-retail-e-commerce-sales-forecast/>
2. Leskovec, J., Adamic, L. A., & Huberman, B. A. (2007). The dynamics of viral marketing. *ACM Transactions on the Web*, *1*(1), 5. https://doi.org/10.1145/1232722.1232727
3. Social Networks. 2019. Github project. <https://github.com/haoyingyang/SN>.
4. Cai, L., & Zhu, Y. (2015). The challenges of data quality and data quality assessment in the Big Data Era. *Data Science Journal*, *14*, 2. https://doi.org/10.5334/dsj-2015-002