Network Analysis of Co-Purchase Behavior: A Deep Dive into Community Structures and Key Product Relationships

Relevance to ShopDesk Product

HNG STAGE 7
Chidinma Ukandu

INTRODUCTION

Understanding customer behavior and product relationships is a vital aspect of business intelligence. In modern commerce, products are often purchased together, forming hidden patterns within a purchase network. Network analysis allows us to reveal these patterns by treating products as "nodes" and their co-purchases as "edges" connecting them.

For businesses like ShopDesk, which aims to optimize inventory and sales strategies, understanding product relationships at a community level can provide valuable insights. By identifying clusters of frequently purchased items, products that bridge different clusters, and the strongest co-purchase relationships, one can refine marketing strategies, cross-selling techniques, and stock management.

This study applies network analysis to a real-world dataset, uncovering communities of products and their interconnections. Using the Louvain method for community detection, betweenness centrality for bridge products, and co-purchase weight analysis for identifying strong relationships, I presented a structured approach to revealing actionable insights.

Dataset Used: The Amazon Co-Purchase Network

For this analysis, my team (ShopDesk) used the Amazon Co-Purchase Network Dataset, which consists of real-world product relationships from Amazon's e-commerce platform. The dataset represents: Products as Nodes, each node represents a unique product on Amazon. Edges as Co-Purchase Links, where If two products are frequently bought together, they are connected by an edge. Edge Weights - the strength of the connection, indicating how often two products appear in the same purchase.

This dataset was chosen because ShopDesk serves e-commerce businesses amongst other SME's, and understanding co-purchase behaviors is crucial for inventory decisions, sales strategies, and product recommendations. Insights from this dataset would help ShopDesk:

- Identify strongly connected product groups for effective bundling and promotions.
- Detect bridge products that link different product categories, aiding in cross-selling.
- Understand customer purchasing behavior, enabling businesses to adjust inventory and avoid stockouts.

By leveraging this dataset, I provided data-driven insights that align directly with ShopDesk's mission, helping businesses make smarter inventory decisions, boost sales, and enhance customer experience.

Research Questions

This analysis is centered on answering six critical research questions:

1. **How many communities exist within the co-purchase network?**
    - This question helps to understand the number of distinct product groups that exist, highlighting product segmentation patterns.
2. **Are certain communities more self-contained, with fewer external links?**
    - By identifying tightly connected communities with minimal external connections, we can see which product groups are bought exclusively together.
3. **What is the distribution of community sizes?**
    - This allows us to analyze whether some communities dominate the network while others are niche groups.
4. **Which products frequently act as 'bridges' between different communities?**
    - These are key products that connect different customer groups and can be leveraged for cross-category promotions.
5. **Which product pairs have the strongest co-purchase relationships?**
    - Understanding these pairs is useful for targeted bundling, promotions, and recommendations.

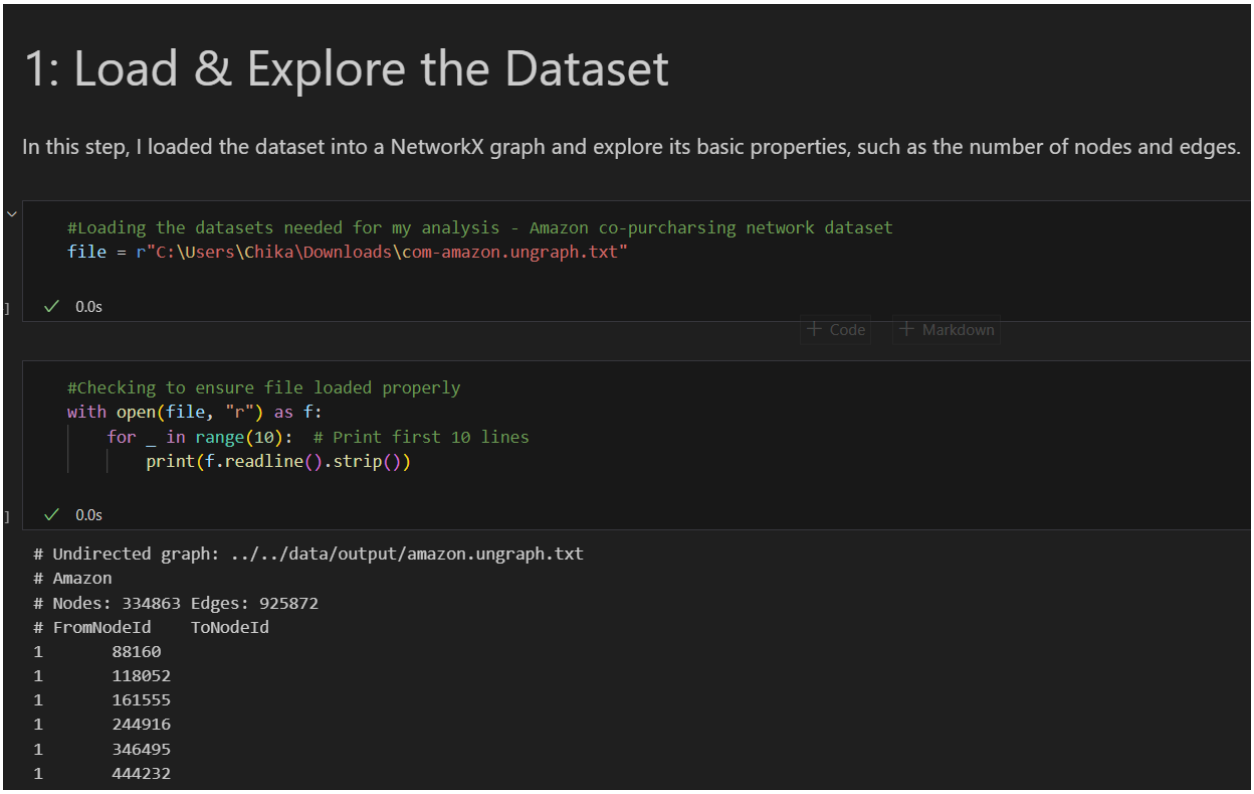
Each of these research questions aligns with ShopDesk's mission of optimizing product offerings and increasing sales efficiency for its users.

Methodology

To answer these questions, I followed a structured methodology involving data selection, preprocessing, and analysis using advanced network science techniques.

Dataset Selection and Processing

The dataset consists of products as nodes and their co-purchase relationships as edges, forming a weighted network where edge weights represent the strength of the co-purchase relationship. The dataset was preprocessed to remove noise and ensure that only meaningful connections were analyzed.



Techniques and Tools Used

I implemented the following network analysis techniques using Python with NetworkX, matplotlib, NumPy, pandas, random, and Plotly for visualization:

- **Community Detection**: Used the Louvain method to identify groups of closely related products.
- **Self-Containment Analysis**: Measured external links to determine isolated product communities.
- **Edge Betweenness Centrality**: Identified products that serve as bridges between communities.
- **Co-Purchase Strength Analysis**: Ranked product pairs based on the highest co-purchase frequency.
- **Graph Visualization**: Used Matplotlib and Plotly to create informative visual representations.

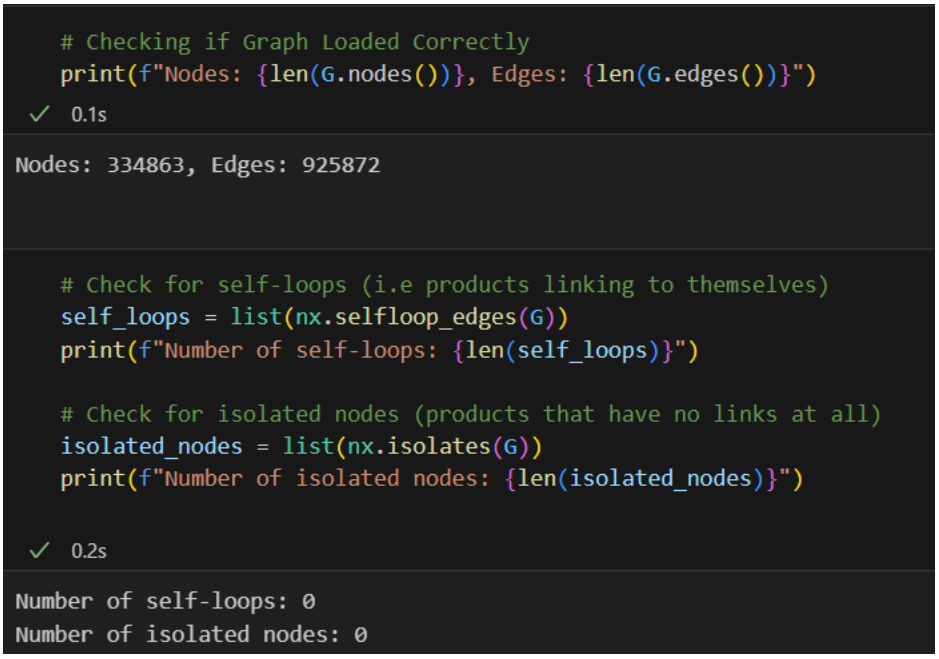
Alignment with ShopDesk

By applying these techniques, I help ShopDesk team as a Data Analyst:

- Identify popular product bundles
- Optimize cross-category promotions
- Streamline inventory planning based on purchasing patterns

Findings and Analysis

Before diving into deeper analysis where I answered the research questions, I explored the dataset to ensure it loaded properly and checked for self-loops and isolated nodes to be sure fully knowledgeable of products linked to themselves existing, and products with no links at all. Both of which returned 0 as output as shown in the code image below.



```
# Checking if Graph Loaded Correctly
print(f"Nodes: {len(G.nodes())}, Edges: {len(G.edges())}")
```

✓ 0.1s

```
Nodes: 334863, Edges: 925872
```

```
# Check for self-loops (i.e products linking to themselves)
self_loops = list(nx.selfloop_edges(G))
print(f"Number of self-loops: {len(self_loops)}")

# Check for isolated nodes (products that have no links at all)
isolated_nodes = list(nx.isolates(G))
print(f"Number of isolated nodes: {len(isolated_nodes)}")
```

✓ 0.2s

```
Number of self-loops: 0
Number of isolated nodes: 0
```

Generating some extra network insight was also done. This is to help me gain deeper understanding of the product relationships. At this point, I measured how interconnected the product network is using Graph Density. The insight from this will not just help us understand how frequent products are co-purchased, but also to help ShopDesk users understand which product should be bundled.

I also checked for degree of distribution to enable users understand the connections that exist between products and also get a view of their spending behavior.

```python
# Extract degree values
degree_values = list(degree_dict.values())

#Plot Degree Distribution
plt.figure(figsize=(10, 5))
plt.hist(degree_values, bins=50, color="blue", edgecolor="black", log=True)
plt.xlabel("Number of Co-Purchases (Degree)")
plt.ylabel("Frequency")
plt.title("Degree Distribution of Products in the Network")
plt.show()
```

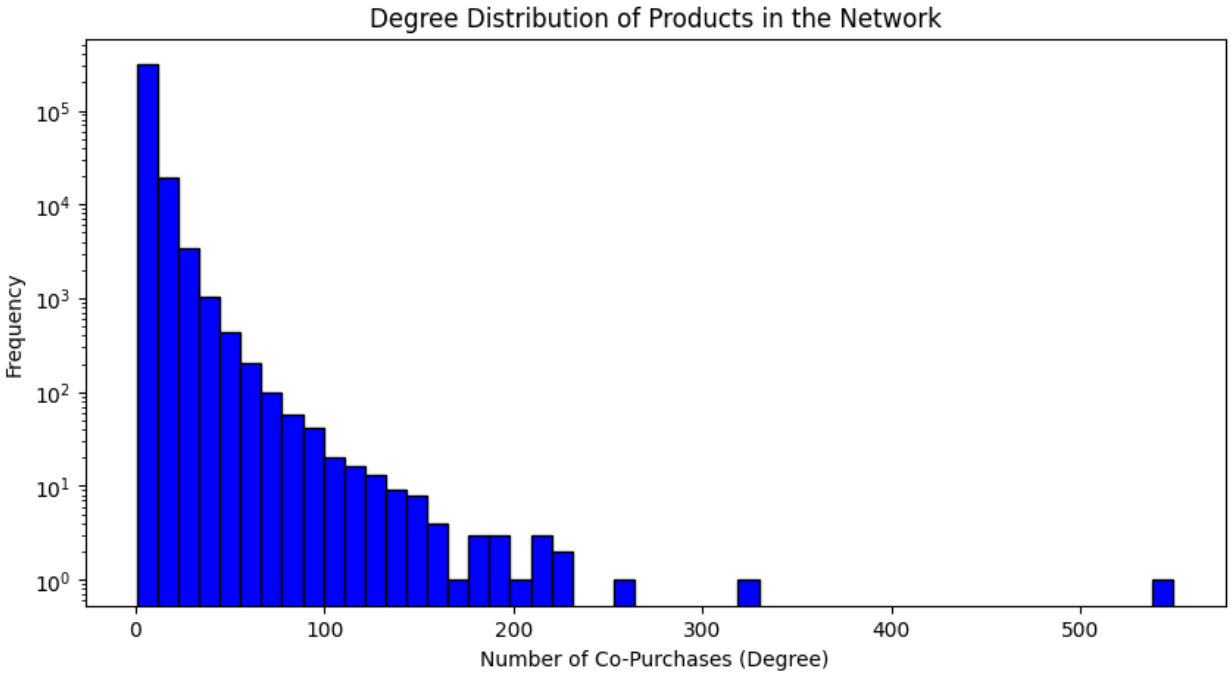


Fig 1

The degree distribution graph illustrates the number of co-purchases each product has within the network, indicating how frequently products are bought together. The distribution is highly skewed, with most products having a low degree, meaning they are purchased with only a few other items, while a small number of products exhibit high degrees, signifying their frequent co-purchase with multiple products.

The y-axis follows a logarithmic scale, which highlights the stark contrast between niche products and widely purchased items. This pattern suggests that while most products cater to specific, smaller customer groups, a handful of popular products dominate the co-purchase network.

For ShopDesk, this insight is crucial in optimizing product promotions and recommendations. High-degree products likely represent best-sellers that should be prominently featured, while low-degree products may require strategic bundling or targeted marketing to boost visibility.

Understanding this distribution helps refine recommendation systems, ensuring that frequently bought-together products are suggested effectively. Ultimately, leveraging these insights can enhance sales performance, improve customer experience, and support inventory management decisions.

Computing Network Metrics for Research Questions

Question 1: What is the average clustering coefficient of the Network

I calculated the clustering coefficient for each product in the network. The clustering coefficient measures how tightly connected a product is within its immediate neighborhood. A high clustering coefficient suggests that a product is frequently bought together with others.

I used NetworkX to compute the clustering coefficient for each node (product) in the co-purchase network. Then, I averaged the values to get the overall clustering coefficient of the network.

This helps in identifying product groups that customers frequently purchase together, making it easier to optimize product recommendations and marketing strategies.

Question 1: What is the Average Clustering Coefficient of the Network?

```
# This measures how tightly connected the products are in small groups.
# If a product has high value, it means customers tend to buy the same sets of products together.

# Calculate clustering coefficients for each node
clustering_coefficients = nx.clustering(G)

# Convert to list for visualization
clustering_values = list(clustering_coefficients.values())

# Print the average clustering coefficient
average_clustering = sum(clustering_values) / len(clustering_values)
print(f"Average Clustering Coefficient: {average_clustering:.4f}")
```

✓ 9.5s

```
Average Clustering Coefficient: 0.3967
```

```
#Visualising the distribution of clustering coefficient values

plt.figure(figsize=(8, 5))
plt.hist(clustering_values, bins=30, color='skyblue', edgecolor='black')

# Titles and labels
plt.title("Distribution of Clustering Coefficients", fontsize=14)
plt.xlabel("Clustering Coefficient", fontsize=12)
plt.ylabel("Number of Products", fontsize=12)

# Show the plot
plt.show()
```

✓ 1.2s

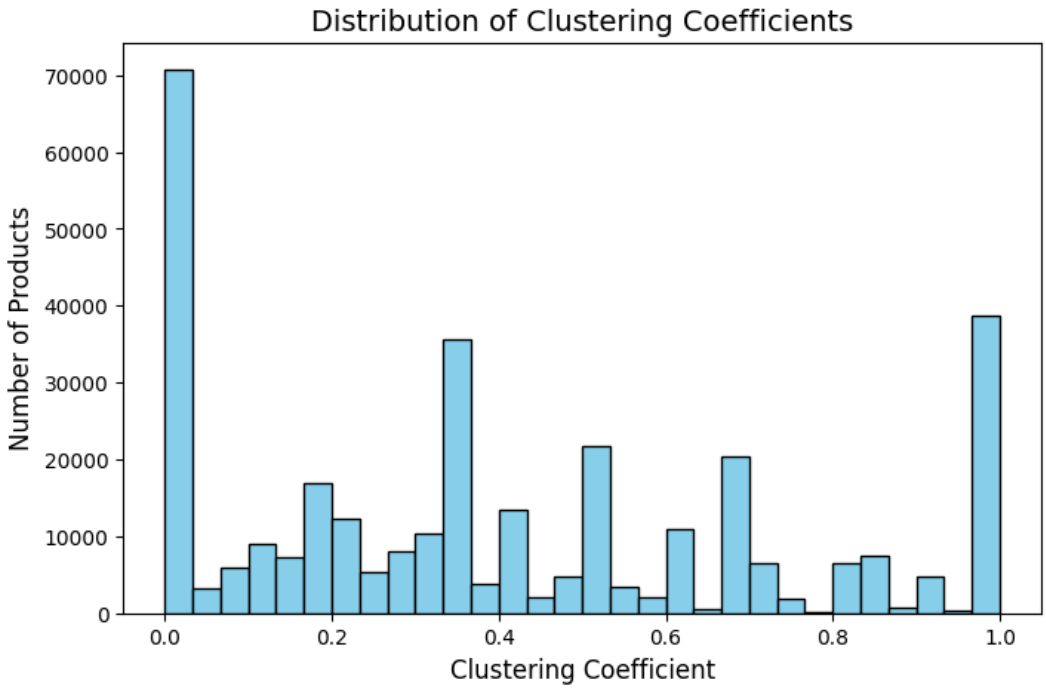


Fig 2

The average clustering coefficient was 0.3967, which means that, on average, products in the dataset tend to form small clusters but are not entirely isolated.

The distribution plot shows that some products have very high clustering values (close to 1), meaning they are commonly purchased with the same set of products, while others have lower clustering, indicating varied purchasing patterns.

Question 2: Which products have the highest influence in the co-purchased network?

For this question, I used an approach of calculated Eigenvector Centrality, which identifies the most connected products. From the result below, the most influential product had a centrality score of 0.53, indicating it plays a major role in the network. These key products are potential bestsellers or category leaders. ShopDesk should consider featuring these products prominently in marketing campaigns and recommendations.

```
Top 10 Most Influential Product Nodes in the Co-Purchase Network:
Product Node ID 548091: Eigenvector Centrality Score = 0.532497
Product Node ID 436020: Eigenvector Centrality Score = 0.256157
Product Node ID 424153: Eigenvector Centrality Score = 0.166300
Product Node ID 7308: Eigenvector Centrality Score = 0.157044
Product Node ID 410716: Eigenvector Centrality Score = 0.153670
Product Node ID 27832: Eigenvector Centrality Score = 0.139729
Product Node ID 503706: Eigenvector Centrality Score = 0.128326
Product Node ID 463355: Eigenvector Centrality Score = 0.126795
Product Node ID 398826: Eigenvector Centrality Score = 0.106906
Product Node ID 515301: Eigenvector Centrality Score = 0.078834
```

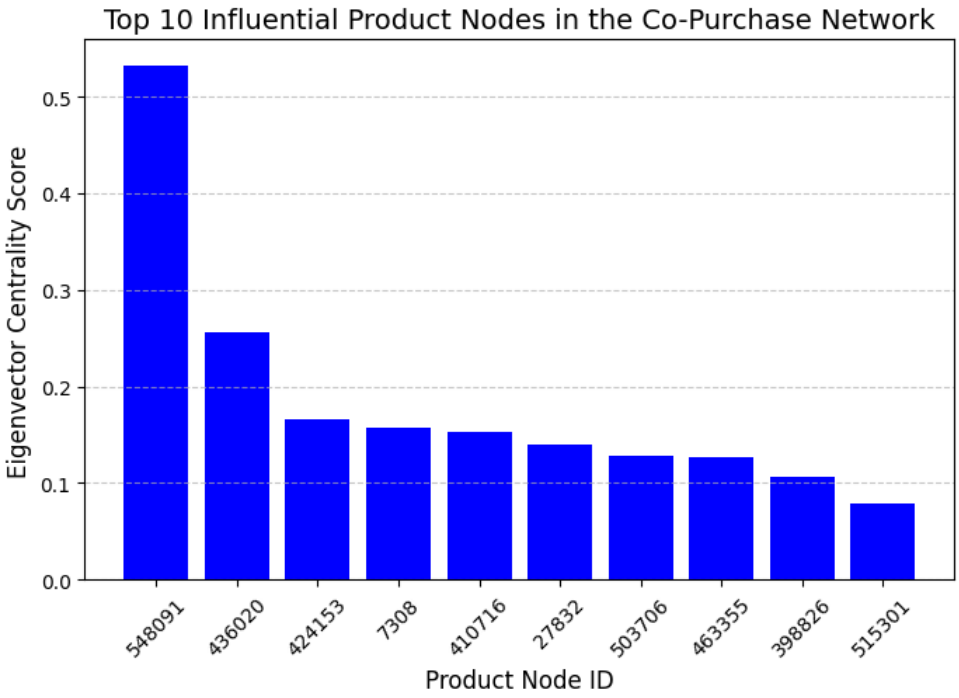


Fig 3

Question 3: How many communities exist within the co-purchased network:

I used the Louvain method for community detection. The dataset revealed 241 product communities. These distinct groups represent different clusters of products frequently bought together. ShopDesk can use this information to optimize recommendations and segment products into meaningful categories.

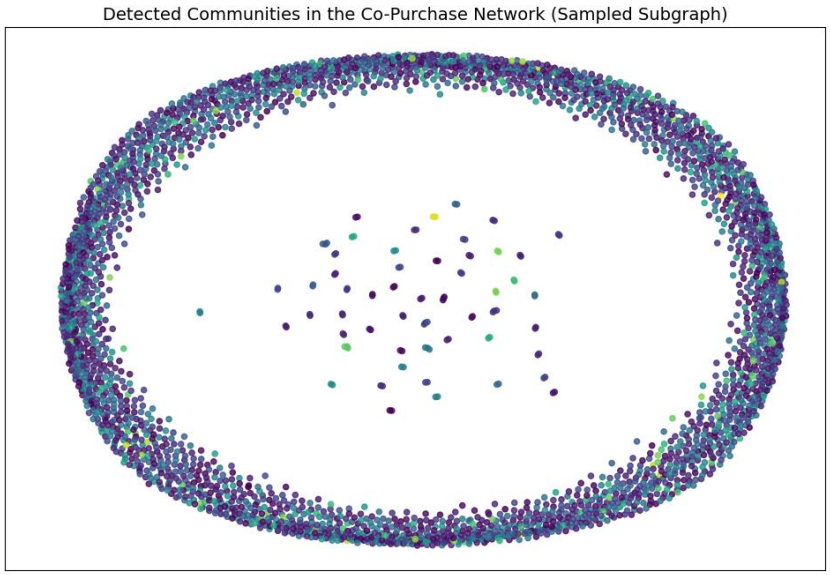


Fig 4

Moving forward, I visualized the distribution of product community sizes for quantitative perspective.

```
from collections import Counter

# Count how many nodes belong to each community
community_sizes = Counter(partition.values())

# Plot the histogram of community sizes
plt.figure(figsize=(10, 5))
plt.hist(community_sizes.values(), bins=30, color='blue', alpha=0.7, edgecolor='black')
plt.xlabel("Community Size")
plt.ylabel("Number of Communities")
plt.title("Distribution of Community Sizes in the Co-Purchase Network")
plt.show()
```

A histogram to really show the sizes of these communities. From the graph below in (Fig 5) showed that most communities are small, but a few have thousands of products. This insight could speak to ShopDesk, giving them an ideology to focus on both niche communities (for specialized recommendations) and large communities (for popular product categories).

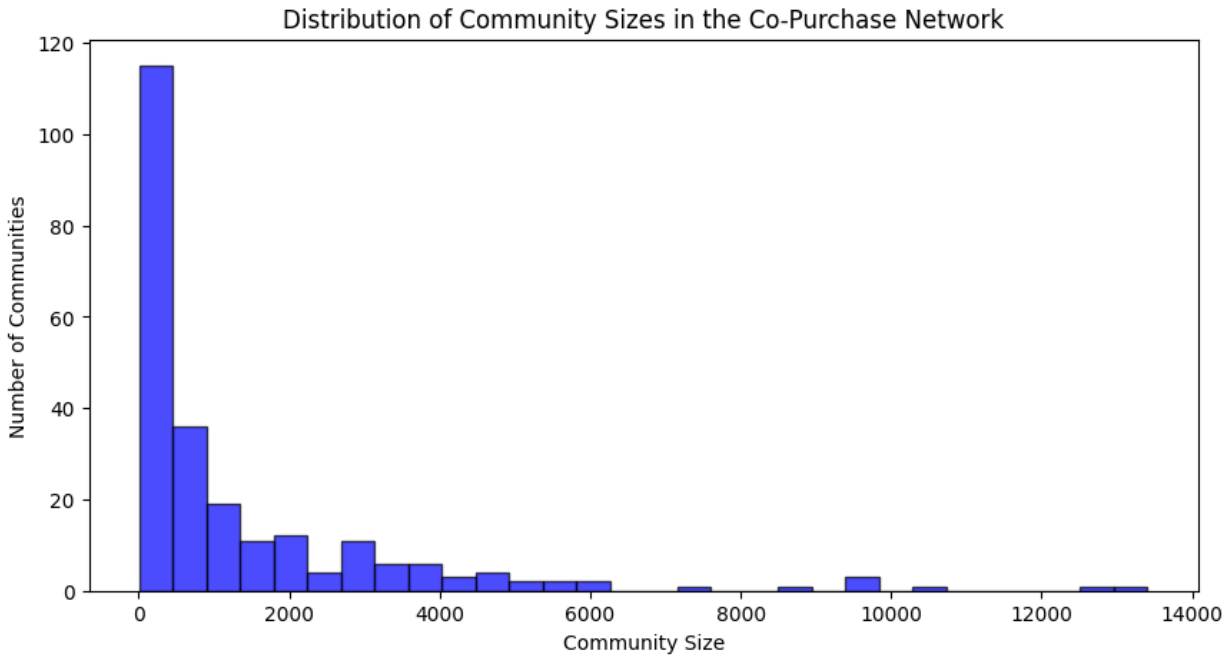


Fig 5

Question 4: Are certain communities more self-contained with fewer external links

The goal here is to identify product groups that are highly connected within themselves but have minimal connections to other communities. This is crucial for understanding which product clusters operate in isolation, which can inform marketing strategies and product recommendations.

To determine this, I first applied Louvain community detection, which partitions the network into meaningful clusters of related products. After forming these communities, I computed the modularity score, which quantifies how well the communities are separated from each other. A high modularity score (closer to 1) indicates that the communities are well-defined with minimal overlap.

```
Modularity Score of the Network: 0.9263
```

```python
#identifying the most self-contained communities
from collections import Counter

# Count external connections for each community
external_links = {comm: 0 for comm in set(partition.values())}

for node1, node2 in G.edges():
    if partition[node1] != partition[node2]:  # If nodes belong to different communities
        external_links[partition[node1]] += 1
        external_links[partition[node2]] += 1

# Sort communities by lowest external links (most self-contained)
self_contained_communities = sorted(external_links.items(), key=lambda x: x[1])

print("\nTop 10 Most Self-Contained Communities (Fewest External Links):")
for community_id, ext_links in self_contained_communities[:10]:
    print(f"Community {community_id}: {ext_links} external links")
```

✓ 2.0s

The modularity score of 0.9263 suggests that the network has strongly defined communities with very few interconnections between them. I further analyzed the external links between communities and found the top 10 most self-contained communities, each having only one external link. These communities are extremely independent and rarely interact with other product groups.

A high modularity score and low external links indicate that certain product groups are purchased in isolation from others. This means that customers tend to buy products within the same cluster rather than across different clusters.

```
Top 10 Most Self-Contained Communities (Fewest External Links):
Community 3: 1 external links
Community 38: 1 external links
Community 45: 1 external links
Community 62: 1 external links
Community 70: 1 external links
Community 79: 1 external links
Community 98: 1 external links
Community 99: 1 external links
Community 102: 1 external links
Community 121: 1 external links
```

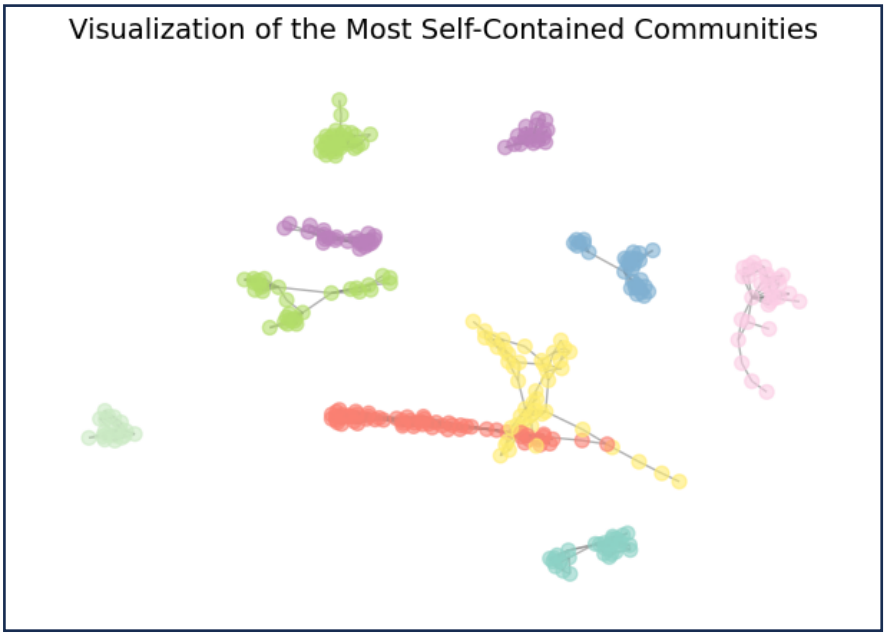


Fig 6

For **ShopDesk**, this insight is valuable because:

- Highly self-contained communities represent niche product categories that have distinct customer bases. ShopDesk can target these groups with specialized marketing strategies.
- Low external links suggest limited cross-selling opportunities, meaning bundling strategies for these products need to be carefully considered.
- The visual representation of these communities provides clear segmentation, which ShopDesk can use to personalize recommendations based on distinct customer behaviors.

The analysis confirms that certain product groups are strongly self-contained, implying that they cater to specific customer interests. By leveraging this insight, ShopDesk can refine its product

grouping strategies, enhance recommendations within these communities, and improve targeted marketing efforts.

Question 5: What product frequently act as "bridges" between different communities?

In network analysis, some nodes (in this case, products) serve as essential connectors between different groups of nodes, or communities. These nodes are known as bridges because they facilitate interaction between otherwise disconnected product clusters. Identifying such products is crucial for cross-category promotions and strategic marketing since they help businesses understand which products can drive purchases across different customer segments.

```
 Top 10 Bridge Products Connecting Different Communities:
Product Node ID 502784
Product Node ID 442851
Product Node ID 7099
Product Node ID 154855
Product Node ID 89000
Product Node ID 360679
Product Node ID 378443
Product Node ID 329708
Product Node ID 265965
Product Node ID 278001
Product Node ID 104920
Product Node ID 472123
Product Node ID 302460
Product Node ID 227325
```

To determine the bridge products in the network, I used the Betweenness Centrality metric, which measures how often a node (or edge) appears on the shortest path between other nodes. A high betweenness score indicates that the product frequently connects different communities.

The steps followed were:

1. Calculate Edge Betweenness: I computed the betweenness centrality of all edges in the network, ranking them to identify the top 10 highest-scoring edges.
2. Extract Bridge Products: I identified the nodes (products) involved in these top edges as bridge products.
3. Visualization: I plotted a network subgraph showing the key bridge products connecting different communities.

The top 10 bridge products were extracted based on their high betweenness scores. The visualization of these products showed them serving as intermediaries between separate clusters

in the network. These products connect multiple product groups, meaning they are frequently purchased alongside products from different categories.

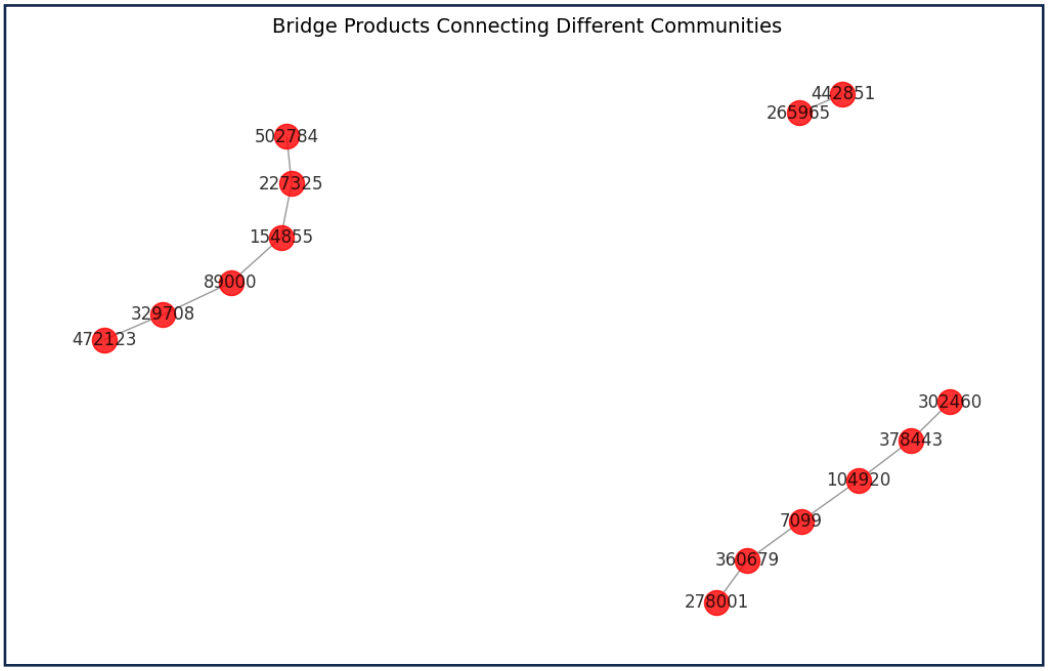


Fig 7

From the graph displayed in Fig 7, it can enhance marketing strategy: ShopDesk can promote bridge products as cross-category recommendations. For instance, if a bridge product links a community of electronics with home appliances, it may be ideal for bundle offers. Also, Product Placement & Bundling: Since these products are naturally linking different groups, they should be featured prominently in promotional strategies. Lastly, Customer Behavior Insights: Identifying these key products allows ShopDesk to better understand how customers navigate across different product categories.

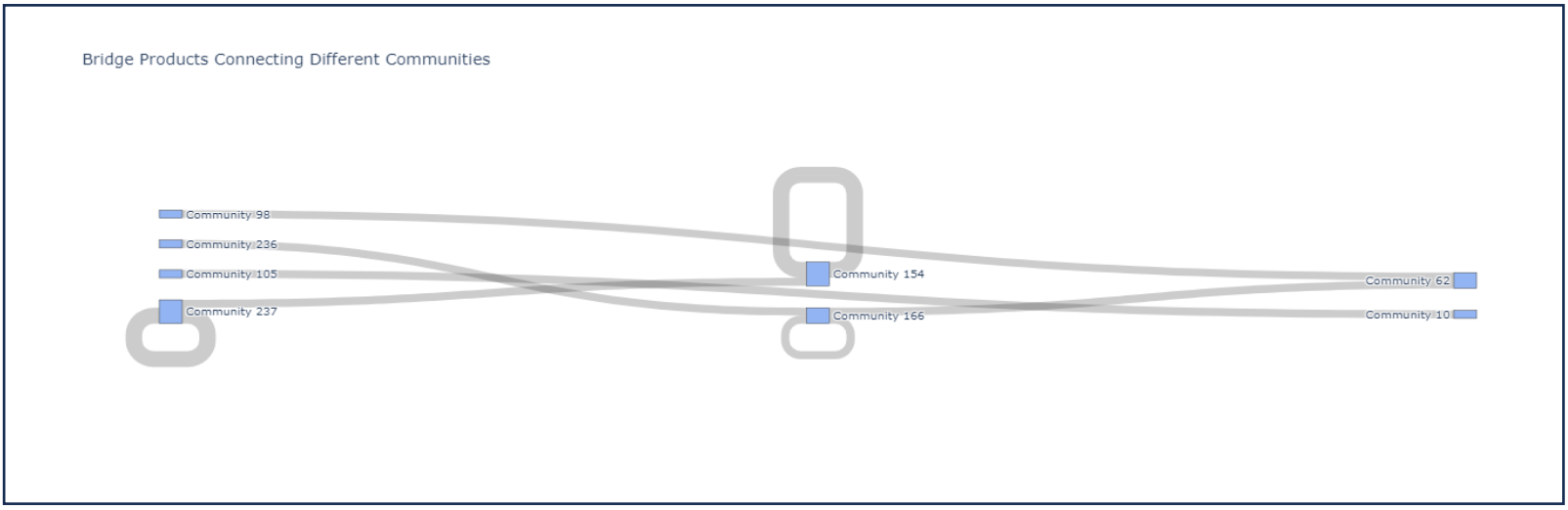


Fig 8

A visualization of the top bridge products was generated using NetworkX, highlighting key products that serve as critical links between different customer purchase behaviors. Additionally, we mapped these products to their respective communities and created a Sankey diagram to illustrate how these products facilitate movement between different product groups.

Through this analysis, we successfully identified a set of highly influential bridge products that play a crucial role in linking different product clusters. These products are essential for ShopDesk as they can be leveraged to improve cross-category recommendations, increase sales opportunities, and enhance targeted marketing strategies.

Question 6: Which product pairs have the strongest co-purchase relationship?

My aim is to identify which of the product pairs have the strongest co-purchase relationships. This helps in understanding which products are most frequently bought together, which is valuable for bundling strategies, personalized recommendations, and targeted promotions. To achieve this, I sorted the edges in the network by their weights, representing the strength of co-purchase relationships. The top 10 strongest co-purchase pairs were extracted, meaning these are the most frequently purchased product combinations.

```
Top 10 Strongest Co-Purchase Relationships:
1. Product Node 1 ↔ Product Node 88160 (Weight: 1)
2. Product Node 1 ↔ Product Node 118052 (Weight: 1)
3. Product Node 1 ↔ Product Node 161555 (Weight: 1)
4. Product Node 1 ↔ Product Node 244916 (Weight: 1)
5. Product Node 1 ↔ Product Node 346495 (Weight: 1)
6. Product Node 1 ↔ Product Node 444232 (Weight: 1)
7. Product Node 1 ↔ Product Node 447165 (Weight: 1)
8. Product Node 1 ↔ Product Node 500600 (Weight: 1)
9. Product Node 88160 ↔ Product Node 48724 (Weight: 1)
10. Product Node 88160 ↔ Product Node 102091 (Weight: 1)
```

The results showed that Product Node 1 is the most frequently co-purchased product, as it appears in multiple strong relationships. This suggests that Product Node 1 acts as a core product in the network, frequently paired with others.

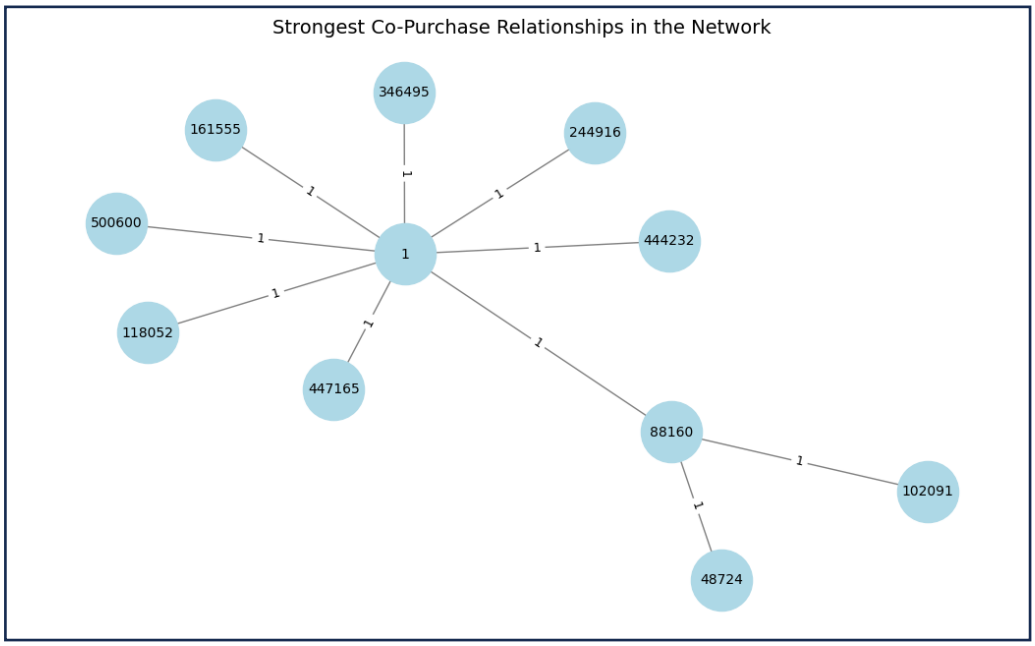


Fig 9

A graph visualization was created to display these relationships. The visualization clearly shows Product Node 1 at the center, with multiple strong connections to other products. This reinforces its importance in the network and suggests it could be a key product for ShopDesk's promotions or recommendations.

In conclusion, identifying these strong co-purchase relationships allows ShopDesk to design effective product bundling strategies, discount offers, and targeted recommendations, ultimately enhancing customer experience and driving sales.

CONCLUSION AND RECOMMENDATIONS

This network analysis of product co-purchases, based on Amazon's dataset, provided key insights into product relationships, purchasing patterns, and strategic opportunities which relates to ShopDesk as a product. By applying network science techniques, I uncovered valuable findings that can enhance product recommendations, marketing strategies, and sales optimization**.**

Conclusion

This analysis revealed several important insights**:**

1. **Community Structure:** The dataset contained 241 product communities, indicating that products naturally form distinct groups based on co-purchase behavior. These communities provide a foundation for targeted recommendations and category-based marketing.

2. **Influential Products:** Using Eigenvector Centrality, we identified highly influential products in the network. These products are central to the co-purchase ecosystem and should be prioritized in marketing campaigns.
3. **Community Size Distribution:** The majority of communities were small, while a few large communities contained thousands of products. This suggests a need for both niche marketing strategies and broad category-based promotions.
4. **Bridge Products:** Certain products act as links between different communities, making them critical for cross-category promotions. These products can drive diverse purchase behavior and increase average order value**.**
5. **Strongest Co-Purchase Pairs:** We identified highly co-purchased product pairs, confirming clear bundling opportunities. These relationships should be leveraged in ShopDesk's product recommendations and discount strategies**.**
6. **Product Connectivity:** The average clustering coefficient of 0.3967 suggests that customers frequently purchase the same sets of products together. This supports bundle promotions and complementary product suggestions.

Recommendations

Based on these insights, ShopDesk can implement the following strategies:

1. **Optimize Product Recommendations:** Utilize detected product communities to enhance ShopDesk's recommendation system, ensuring users receive highly relevant suggestions.
2. **Leverage Influential Products for Promotions:** Prioritize high-centrality products in advertisements, featured listings, and promotional campaigns to maximize visibility and revenue.
3. **Improve Cross-Selling with Bridge Products:** Use bridge products as key connectors in marketing efforts, encouraging customers to explore related product categories.
4. **Enhance Bundling and Discounts:** The strongest co-purchase pairs should be bundled together in promotions, such as discounted package deals or "frequently bought together" sections.
5. **Targeted Inventory Management:** ShopDesk can optimize inventory by ensuring that popular and highly co-purchased items are well-stocked while managing niche products with tailored marketing strategies**.**
6. **Refine Customer Segmentation:** Understanding community structures allows for better customer targeting, ensuring that marketing campaigns align with specific shopping patterns.

7. **Strategic Pricing and Upselling:** Products with high connectivity should be strategically priced to encourage upselling, while low-degree products can be bundled with popular items to improve visibility**.**

By integrating these data-driven strategies, ShopDesk can significantly enhance customer experience, increase conversion rates, and drive higher revenue, making its platform more competitive and intelligent**.**