1. Load the Dataset

Import the Libraries

- **import pandas as pd** :

This imports the pandas library and assigns it the alias pd.

- **import networkx as nx** :

This imports the networkx library and assigns it the alias nx

In [1]:
```
import pandas as pd
import networkx as nx
```

Load the dataset

In [3]:
```
df = pd.read_csv('FashionDataset.csv')
df.head()
```

Out[3]:

	Unnamed: 0	BrandName	Deatils	Sizes	MRP	SellPrice	D
0	0	life	solid cotton blend collar neck womens a-line d...	Size:Large,Medium,Small,X-Large,X-Small	Rs\n1699	849	
1	1	only	polyester peter pan collar womens blouson dres...	Size:34,36,38,40	Rs\n3499	2449	
2	2	fratini	solid polyester blend wide neck womens regular...	Size:Large,X-Large,XX-Large	Rs\n1199	599	
3	3	zink london	stripes polyester sweetheart neck womens dress...	Size:Large,Medium,Small,X-Large	Rs\n2299	1379	
4	4	life	regular fit regular length denim womens jeans ...	Size:26,28,30,32,34,36	Rs\n1699	849	

Create the graph

In [13]:
```
G = nx.Graph()
```

Adding edges to the graph

This process builds a graph where nodes represent brands and categories, and edges represent the associations between them, allowing for further analysis or visualization.

In [16]:
```
for _, row in df.iterrows():
    if not pd.isna(row['BrandName']) and not pd.isna(row['Category']):
        G.add_node(row['BrandName'])
        G.add_node(row['Category'])
        G.add_edge(row['BrandName'], row['Category'])
```

Basis of Community Formation:

Brand and Category Relationships : The communities in the graph are formed based on **how closely related different BrandName and Category nodes are to each other**.

If a brand frequently appears in multiple categories, or if certain categories are closely linked to a specific set of brands, these nodes are likely to be grouped into the same community.

2. Apply the Louvain Algorithm

- The Louvain algorithm seeks to maximize modularity, a measure of the density of links inside communities compared to links between communities.

Import the Libraries

- **import community as community_louvain** :

Used for community detection in graphs.

- **import matplotlib.pyplot as plt** :

A popular Python library for creating static, interactive, and animated visualizations.

In [20]:
```
import community as community_louvain
import matplotlib.pyplot as plt
```

Compute the best partition using Louvain algorithm

- Computes the best partition of the graph G using the Louvain algorithm.
- The Louvain algorithm is a **greedy optimization method** that **attempts to maximize the modularity of the network**, which is a measure of the strength of division of a network into communities.

In [22]:
```
partition = community_louvain.best_partition(G)
```

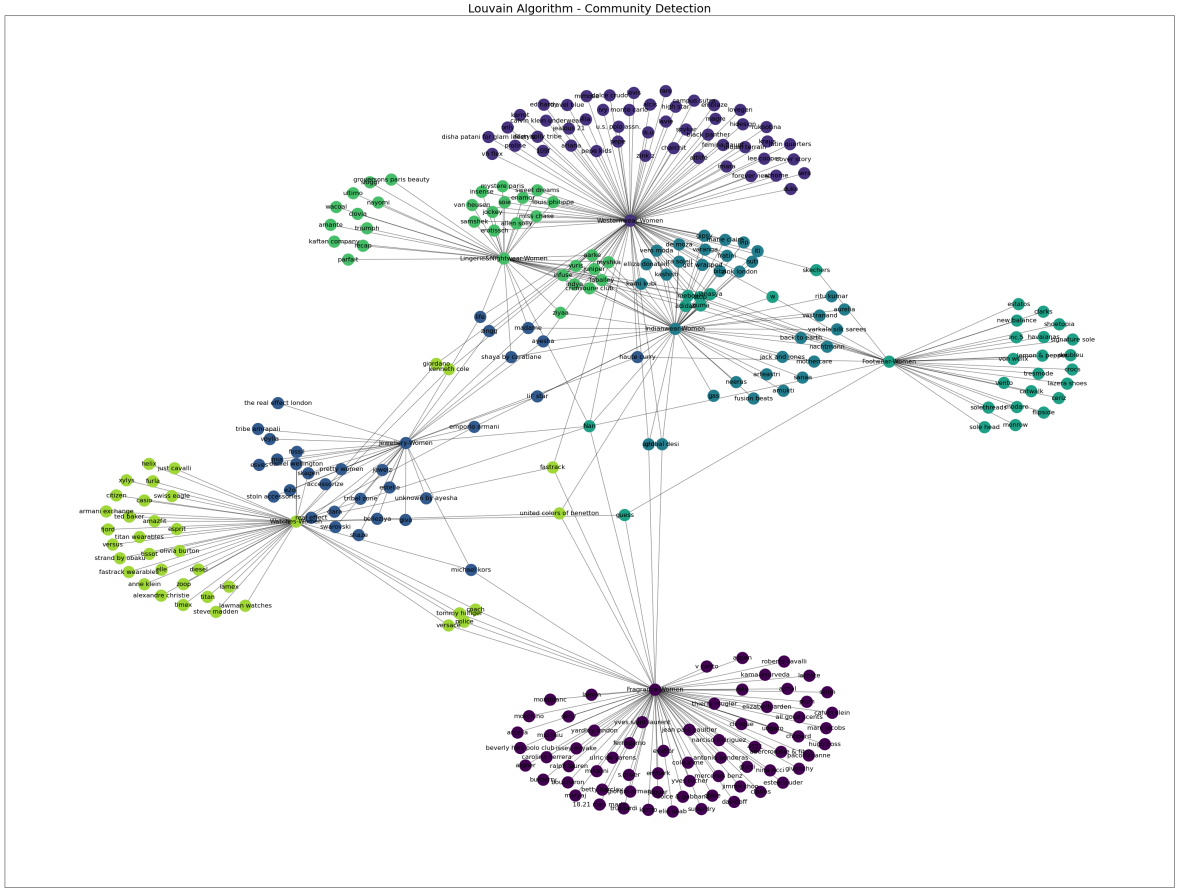
In [23]:
```
# Print the partition
print("Louvain Algorithm - Node communities:", partition)
```

```
Louvain Algorithm - Node communities: {'life': 2, 'Westernwear-Women': 1, 'only':
1, 'fratini': 3, 'zink london': 3, 'kraus': 1, 'rare': 1, 'van heusen': 5, 'Nan':
4, 'stop': 4, 'zink z': 1, 'cover story': 1, 'infuse': 5, 'allen solly': 5, 'altl
ife': 1, 'madame': 2, 'iti': 3, 'levis': 1, 'and': 3, 'faballey': 5, 'latin quart
ers': 1, 'sera': 1, 'adidas': 4, 'pepe': 1, 'indya': 5, 'juniper': 5, 'lovegen':
1, 'vero moda': 3, 'forever new': 1, 'magre': 1, 'insense': 5, 'calvin klein unde
rwear': 1, 'reebok': 4, 'marie claire': 3, 'enamor': 5, 'spykar': 1, 'get wrappe
d': 3, 'campus sutra': 1, 'dolce crudo': 1, 'emblaze': 1, 'global desi': 3, 'jeal
ous 21': 1, 'jockey': 5, 'crimsoune club': 5, 'haute curry': 2, 'proline': 1, 'pu
ma': 4, 'ivy': 1, 'vh flex': 1, 'myshka': 5, 'femina flaunt': 1, 'w': 4, '109f':
1, 'gipsy': 3, 'samshek': 5, 'is.u': 1, 'miss chase': 5, 'ira soleil': 3, 'fila':
1, 'kami kubi': 3, 'travel blue': 1, 'varanga': 3, 'de moza': 3, 'u.s. polo ass
n.': 1, 'vip': 3, 'janasya': 4, 'skechers': 4, 'lee cooper': 1, 'aarke': 5, 'kash
ish': 3, 'pepe kids': 1, 'kenneth cole': 6, 'zingg': 2, 'ayesha': 2, 'elliza dona
tein': 3, 'erotissch': 5, 'biba': 3, 'ed hardy': 1, 'black panther': 1, 'mystere
paris': 5, 'monte carlo': 1, 'louis philippe': 5, 'mimosa': 1, 'hidesign': 1, 'so
ie': 5, 'sweet dreams': 5, 'giordano': 6, 'karrot': 1, 'alcis': 1, 'imara': 1, 'c
harchit': 1, 'athome': 1, 'disha patani for glam lifestyle': 1, 'allen solly trib
e': 1, 'ariana': 1, 'lavie': 1, 'rukootina': 1, 'indian terrain': 1, 'yuris': 5,
'duke': 1, 'high star': 1, 'suti': 3, 'fastrack': 6, 'Indianwear-Women': 3, 'sana
a': 3, 'ziyaa': 5, 'mothercare': 3, 'jack and jones': 3, 'aurelia': 3, 'shaya by
caratlane': 2, 'neerus': 3, 'amukti': 3, 'vastranand': 3, 'varkala silk sarees':
3, 'fusion beats': 3, 'ritu kumar': 3, 'emporio armani': 2, 'gas': 3, 'arteastr
i': 3, 'back to earth': 3, 'united colors of benetton': 6, "lil' star": 2, 'nacht
mann': 3, 'Lingerie&Nightwear-Women': 5, 'clovia': 5, 'triumph': 5, 'amante': 5,
'nayomi': 5, 'kaftan company': 5, 'parfait': 5, 'wacoal': 5, 'groversons paris be
auty': 5, 'sloggi': 5, 'ultimo': 5, 'recap': 5, 'catwalk': 4, 'Footwear-Women':
4, 'inc.5': 4, 'crocs': 4, 'estatos': 4, 'tresmode': 4, 'lemon & pepper': 4, 'sho
etopia': 4, 'new balance': 4, 'solethreads': 4, 'ceriz': 4, 'modare': 4, 'clark
s': 4, 'doubleu': 4, 'guess': 4, 'monrow': 4, 'lazera shoes': 4, 'flipside': 4,
'von wellx': 4, 'signature sole': 4, 'havaianas': 4, 'vento': 4, 'sole head': 4,
'titan': 6, 'Watches-Women': 6, 'fastrack wearables': 6, 'armani exchange': 6, 'f
ossil': 2, 'michael kors': 2, 'casio': 6, 'timex': 6, 'anne klein': 6, 'tommy hil
figer': 6, 'daniel wellington': 2, 'titan wearables': 6, 'tissot': 6, 'swiss eagl
e': 6, 'amazfit': 6, 'coach': 6, 'skagen': 2, 'esprit': 6, 'helix': 6, 'just cava
lli': 6, 'steve madden': 6, 'ted baker': 6, 'lamex': 6, 'elle': 6, 'versus': 6,
'fjord': 6, 'lawman watches': 6, 'versace': 6, 'citizen': 6, 'olivia burton': 6,
'xylys': 6, 'strand by obaku': 6, 'diesel': 6, 'police': 6, 'furla': 6, 'zoop':
6, 'alexandre christie': 6, 'antonio banderas': 0, 'Fragrance-Women': 0, 'dolce &
gabbana': 0, 'paco rabanne': 0, 'carolina herrera': 0, 'skinn': 0, 'arcelia': 0,
'gucci': 0, 'burberry': 0, 'yves saint laurent': 0, 'jaguar': 0, 'jean paul gault
ier': 0, 'elie saab': 0, 'calvin klein': 0, 'hugo boss': 0, 'issey miyake': 0, 'a
jmal': 0, 'thierry mugler': 0, 'clinique': 0, 'davidoff': 0, 'mercedes benz': 0,
'nina ricci': 0, 'nike': 0, 'yardley london': 0, 'narciso rodriguez': 0, 'moschin
o': 0, 'giorgio armani': 0, 'montblanc': 0, 'embark': 0, 'yves rocher': 0, 'eliza
beth arden': 0, 'ferragamo': 0, 'lanvin': 0, 'clarins': 0, 'aspen': 0, 'jimmy cho
o': 0, 'maryaj': 0, 'abercrombie & fitch': 0, 'trussardi': 0, 'beverly hills polo
club': 0, 'miu miu': 0, 'evaflor': 0, 'colour me': 0, 'chopard': 0, 'betty barcla
y': 0, 'lacoste': 0, 'superdry': 0, 'kenzo': 0, 'estee lauder': 0, 's.oliver': 0,
'chloe': 0, 'ralph lauren': 0, 'marc jacobs': 0, '4711': 0, 'plum': 0, 'missoni':
0, 'boucheron': 0, 'ulric de varens': 0, 'givenchy': 0, 'v canto': 0, 'all good s
cents': 0, 'kama ayurveda': 0, 'roberto cavalli': 0, 'aigner': 0, 'dkny': 0, 'ung
aro': 0, '18.21 man made': 0, 'Jewellery-Women': 2, 'e2o': 2, 'pretty women': 2,
'belleziya': 2, 'estelle': 2, 'jewelz': 2, 'clara': 2, 'giva': 2, 'shaze': 2, 'sw
arovski': 2, 'mia': 2, 'tribal zone': 2, 'voylla': 2, 'unknown by ayesha': 2, 're
al effect': 2, 'stoln accessories': 2, 'tribe amrapali': 2, 'esves': 2, 'the real
effect london': 2, 'accessorize': 2}
```

Draw the graph

In [25]:
```
pos = nx.spring_layout(G)
cmap = plt.get_cmap('viridis')
unique_communities = set(partition.values())
colors = [cmap(i / len(unique_communities)) for i in range(len(unique_communitie
node_color = [colors[partition[node]] for node in G.nodes()]

plt.figure(figsize=(40, 30))
nx.draw_networkx_nodes(G, pos, node_color=node_color, node_size=500)
nx.draw_networkx_edges(G, pos, alpha=0.5)
nx.draw_networkx_labels(G, pos, labels={node: str(node) for node in G.nodes()},
plt.title("Louvain Algorithm - Community Detection", fontsize=22)
plt.show()
```



3. Implement the Girvan-Newman Algorithm

The Girvan-Newman algorithm works by **iteratively removing edges with the highest betweenness centrality** to separate the network into distinct communities.

Function to find the edge with the highest betweenness centrality

This function is designed to identify and return the edge in the graph g with the highest betweenness centrality, which is a key step in the Girvan-Newman algorithm.

In [33]:
```
def edge_to_remove(g):
    d1 = nx.edge_betweenness_centrality(g)
    sorted_list = sorted(d1.items(), key=lambda x: x[1], reverse=True)
    return sorted_list[0][0]
```

Girvan-Newman algorithm to remove edges and find communities

This function implements the **Girvan-Newman algorithm**, which is used for **detecting communities in a graph** by iteratively removing edges with the **highest betweenness centrality** until the graph breaks into **multiple connected components**.

In [36]:
```
def girvan(g):
    while len(list(nx.connected_components(g))) == 1:
        u, v = edge_to_remove(g)
        g.remove_edge(u, v)
    return list(nx.connected_components(g))
```

Apply the Girvan-Newman algorithm

In [39]:
```
components = girvan(G)
```

Extract and print the detected communities

Extracts the individual communities and prints them.

In [41]:
```
components_list = [list(component) for component in components]
print("Girvan-Newman Algorithm - Communities:", components_list)
```

```
Girvan-Newman Algorithm - Communities: [['fratini', 'kraus', 'jealous 21', 'the r
eal effect london', 'inc.5', 'jockey', 'w', 'ziyaa', 'ultimo', 'swarovski', 'iv
y', 'madame', 'doubleu', 'levis', 'estatos', 'clarks', 'dolce crudo', 'janasya',
'hidesign', 'van heusen', 'ariana', 'shaya by caratlane', 'biba', 'enamor', 'hava
ianas', 'forever new', 'nayomi', 'is.u', 'sweet dreams', 'lavie', 'travel blue',
'vastranand', 'Lingerie&Nightwear-Women', 'sloggi', 'jewelz', 'back to earth', 'a
rteastri', '109f', 'proline', 'indian terrain', 'amante', 'solethreads', 'giva',
'athome', 'myshka', 'erotissch', 'kaftan company', 'parfait', 'puma', 'suti', 'le
mon & pepper', 'modare', 'allen solly', 'yuris', 'duke', 'ritu kumar', 'reebok',
'tribal zone', 'insense', 'global desi', 'catwalk', 'clovia', 'magre', 'zink z',
'wacoal', 'lee cooper', 'haute curry', 'elliza donatein', 'zingg', 'samshek', 'im
ara', 'disha patani for glam lifestyle', "lil' star", 'signature sole', 'ceriz',
'triumph', 'faballey', 'cover story', 'real effect', 'jack and jones', 'aarke',
'unknown by ayesha', 'kashish', 'skechers', 'recap', 'miss chase', 'only', 'fusio
n beats', 'femina flaunt', 'lovegen', 'juniper', 'karrot', 'charchit', 'voylla',
'Indianwear-Women', 'u.s. polo assn.', 'e2o', 'stop', 'emporio armani', 'latin qu
arters', 'amukti', 'gas', 'iti', 'clara', 'rukootina', 'zink london', 'indya', 'g
et wrapped', 'calvin klein underwear', 'sanaa', 'tresmode', 'shoetopia', 'estell
e', 'black panther', 'de moza', 'ed hardy', 'Jewellery-Women', 'varanga', 'pepe k
ids', 'ira soleil', 'neerus', 'flipside', 'esves', 'sera', 'fila', 'crimsoune clu
b', 'vero moda', 'crocs', 'campus sutra', 'ayesha', 'monte carlo', 'sole head',
'rare', 'kami kubi', 'groversons paris beauty', 'alcis', 'allen solly tribe', 'an
d', 'von wellx', 'mimosa', 'soie', 'emblaze', 'aurelia', 'life', 'louis philipp
e', 'vip', 'mothercare', 'nachtmann', 'Footwear-Women', 'Nan', 'belleziya', 'monr
ow', 'mia', 'accessorize', 'gipsy', 'Westernwear-Women', 'marie claire', 'altlif
e', 'mystere paris', 'vh flex', 'high star', 'pretty women', 'vento', 'pepe', 'tr
ibe amrapali', 'infuse', 'spykar', 'lazera shoes', 'stoln accessories', 'varkala
silk sarees', 'new balance', 'shaze', 'adidas'], ['timex', 'calvin klein', 'arcel
ia', 'gucci', 'ferragamo', 'dolce & gabbana', 'burberry', 'fjord', 'united colors
of benetton', 'furla', 'marc jacobs', 'kenneth cole', 'swiss eagle', 'lamex', 'ja
guar', 'giordano', 'xylys', 'fossil', 'beverly hills polo club', 'skinn', 'titan
wearables', 'esprit', 'lanvin', 'anne klein', 'aigner', 'ted baker', 'jimmy cho
o', 'guess', 'steve madden', 'maryaj', 'just cavalli', 'chopard', 'hugo boss', 'p
aco rabanne', 'v canto', 'yardley london', 'police', 'daniel wellington', 'narcis
o rodriguez', 'ajmal', 'strand by obaku', 'elie saab', 'carolina herrera', 'montb
lanc', 'casio', 'fastrack', 'betty barclay', 'clarins', 'versace', 'tissot', 'tit
an', 'all good scents', 'ulric de varens', 'zoop', 'yves saint laurent', 'moschin
o', 'amazfit', 'superdry', 'kenzo', 'michael kors', 'Watches-Women', 'diesel', 'a
bercrombie & fitch', 'armani exchange', 'issey miyake', 'roberto cavalli', 'misso
ni', 'embark', 'yves rocher', 'elle', 'nike', '18.21 man made', 'dkny', 'colour m
e', 'estee lauder', 'boucheron', 'lawman watches', 'givenchy', 'antonio bandera
s', 'Fragrance-Women', 'elizabeth arden', 's.oliver', 'giorgio armani', 'coach',
'alexandre christie', 'aspen', 'davidoff', 'jean paul gaultier', 'fastrack wearab
les', 'miu miu', 'clinique', 'citizen', 'versus', 'trussardi', 'chloe', 'skagen',
'mercedes benz', 'thierry mugler', 'kama ayurveda', 'plum', 'lacoste', 'olivia bu
rton', '4711', 'helix', 'nina ricci', 'ralph lauren', 'ungaro', 'tommy hilfiger',
'evaflor']]
```

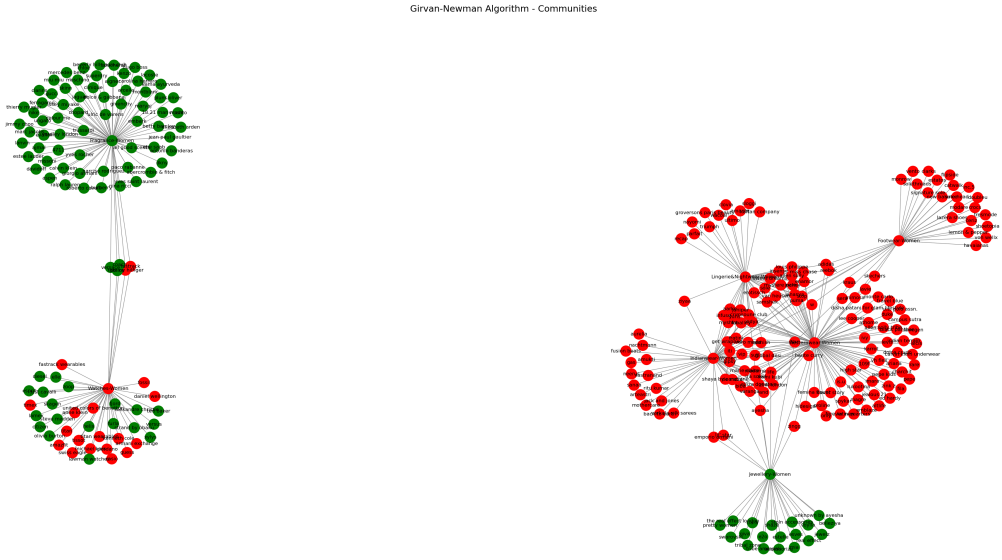
4. Visualize the Communities Found by Girvan-Newman Algorithm

In [43]:
```python
def draw_graph(g, components):
    color_map = []
    colors = ['red', 'green', 'blue', 'orange', 'purple', 'cyan']  # Add more co
    pos = nx.spring_layout(g, seed=42)  # Seed for reproducibility

    for idx, component in enumerate(components):
        for node in component:
            color_map.append(colors[idx % len(colors)])  # Cycle through colors
```

```
    plt.figure(figsize=(40, 20))
    nx.draw(g, pos, node_color=color_map, with_labels=True, node_size=700, font_
    plt.title("Girvan-Newman Algorithm - Communities", fontsize=22)
    plt.show()

# Visualize the graph after applying the Girvan-Newman algorithm
draw_graph(G, components_list)
```



In [44]:
```
# Calculate modularity for Louvain partition
modularity_louvain = community_louvain.modularity(partition, G)
print(f"Louvain Modularity: {modularity_louvain}")
```

```
Louvain Modularity: 0.5936441195323708
```

Insights

The modularity score ranges from -1 to 1. A higher value (closer to 1) indicates a **strong community structure**. In this case, a modularity score of approximately 0.594 suggests that the partitioning achieved by the Louvain algorithm has a fairly strong community structure where nodes within the same community are more densely connected than those across communities.