Task 6: To design and perform visualization for Graphs and Networks

• Force based Layout

Tools: Tableau, Language :Python

Data set link : Stack Overflow Tag Network | Kaggle

Utilize a given dataset link it contains network links, source and target technical tags, and the link value between each pair. It also contains the nodes of the network, the name of each node, the group to which this node belongs and a node size based on the frequency of use of this technological beacon. How to visualize and analyze a network graph using the Python programming language:

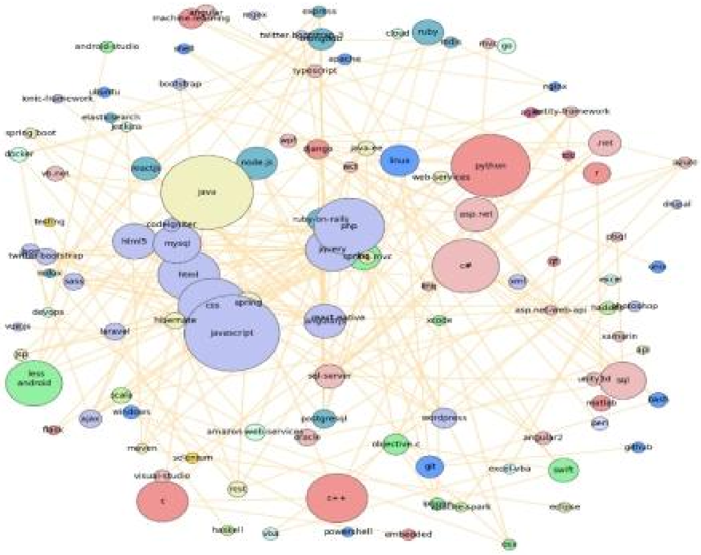
Algorithm:

- Import necessary libraries
- Import NetworkX as nx for creating and manipulating network graphs.
- Import Matplotlib.pyplot as plt for data visualization.
- Import Pandas as pd for data handling.
- Create an empty graph object G and set an attribute 'day' to "Stackoverflow."
- Read node and edge data from CSV files ('stack_network_nodes.csv' and 'stack_network_links.csv') into DataFramesdf_nodes and df_links, respectively.
- Add nodes to the graph G:
- Iterate through the rows in df_nodes.
- For each row, add a node to the graph with a 'name,' 'group,' and 'nodesize' attributes.
- Add weighted edges to the graph G
- Iterate through the rows in df_links.
- For each row, add a weighted edge from the 'source' to the 'target' with the 'value' as the weight.
- Set visualization parameters:
- Create a figure with a specified size.
- Define visualization options like edge color, width, node labels, and font weight.
- Determine colors for nodes based on their 'group' attribute.
- Adjust node sizes based on the 'nodesize' attribute.
- Draw the network graph:
- Use nx.draw to visualize the graph.
- Customize the node colors and sizes based on the calculated values.
- Define the layout of the nodes using spring layout with specified parameters.
- Set the edge color to a specific color.
- Show the graph using plt.show().
- End of the algorithm.

Code:

```
import networkx as nx
import matplotlib.pyplot as plt
import pandas as pd

G = nx.Graph(day="Stackoverflow")
df_nodes = pd.read_csv('stack_network_nodes.csv')
df_links = pd.read_csv('stack_network_links.csv')

for index, row in df_nodes.iterrows():
G.add_node(row['name'], group=row['group'], nodesize=row['nodesize'])
for index, row in df_links.iterrows():
G.add_weighted_edges_from([(row['source'], row['target'], row['value'])])

plt.figure(figsize=(15,15))
options = {
   'edge_color': '#FFDEA2',
   'width': 1.5,
   'with_labels': True,
   'font_weight': 'regular',
}
colors = [color_map[G.nodes[node]['group']] for node in G]
sizes = [G.nodes[node]['nodesize']*25 for node in G]

nx.draw(G, node_color=colors, node_size=sizes, pos=nx.spring_layout(G, k=1.5, iterations=15), **options)
ax = plt.gca()
ax.collections[0].set_edgecolor("#555555")
plt.show()
```

Output:



Result:
The force-based network graph was successfully visualized, showing connections between Stack Overflow tags. Nodes represent tags, edges show relationships, and node sizes reflect tag frequency. The layout clearly illustrates clusters of related technologies.