

Análise de Redes

Slide – Atividade Prática – Unidade 2

Prof. Patrick Terrematte

Análise de Redes – Atividade Prática 1

- Execute o seguinte tutorial sobre Análise de Redes e Comunidades com dataset de Game of Thrones

https://github.com/terrematte/network_analysis/blob/main/notebooks/o6_Network_Analysis_on_GoT.ipynb

Análise de Redes – Atividade Prática 2

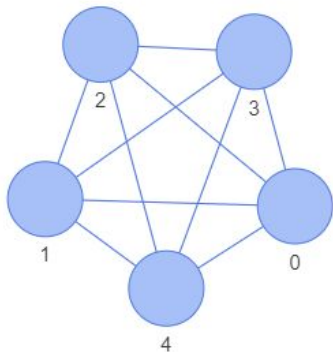
- Execute os seguintes tutoriais sobre o Pyvis em um único notebook do Colab:
- <https://pyvis.readthedocs.io/en/latest/tutorial.html>
- <https://towardsdatascience.com/making-network-graphs-interactive-with-python-and-pyvis-b754c22c270>

Using pyvis within Jupyter notebook

Pyvis supports [Jupyter](#) notebook embedding through the use of the `network.Network()` constructor. The network instance must be “prepped” during instantiation by supplying the `notebook=True` kwarg. Example:

```
In [1]: from pyvis import network as net  
import networkx as nx
```

```
In [2]: g=net.Network(notebook=True)  
nxg = nx.complete_graph(5)  
g.from_nx(nxg)  
g.show("example.html")
```



Note

while using notebook in chrome browser, to render the graph, pass additional kwarg 'cdn_resources' as 'remote' or 'inline'

pyvis library

<https://pyvis.readthedocs.io/en/latest/tutorial.html>

```
>>> from pyvis.network import Network
>>> net = Network()
```

```
>>> net.add_node(1, label="Node 1") # node id = 1 and label = Node 1
>>> net.add_node(2) # node id and label = 2
```

```
>>> nodes = ["a", "b", "c", "d"]
>>> net.add_nodes(nodes) # node ids and labels = ["a", "b", "c", "d"]
>>> net.add_nodes("hello") # node ids and labels = ["h", "e", "l", "o"]
```

```
>>> net.add_nodes(["a", "b", "c"])
>>> net.get_node("c")
>>> {'id': 'c', 'label': 'c', 'shape': 'dot'}
```

pyvis library

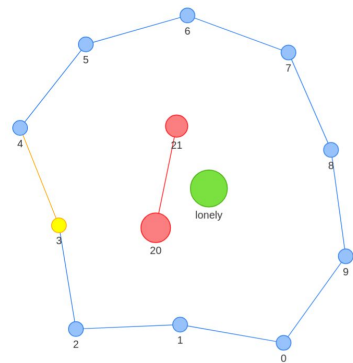
<https://pyvis.readthedocs.io/en/latest/tutorial.html>

```
>>> g = Network()  
>>> g.add_nodes([1,2,3], value=[10, 100, 400],  
                title=['I am node 1', 'node 2 here', 'and im node 3'],  
                x=[21.4, 54.2, 11.2],  
                y=[100.2, 23.54, 32.1],  
                label=['NODE 1', 'NODE 2', 'NODE 3'],  
                color=['#00ff1e', '#162347', '#dd4b39'])
```



Networkx integration

```
>>> from pyvis.network import Network
>>> import networkx as nx
>>> nx_graph = nx.cycle_graph(10)
>>> nx_graph.nodes[1]['title'] = 'Number 1'
>>> nx_graph.nodes[1]['group'] = 1
>>> nx_graph.nodes[3]['title'] = 'I belong to a different group!'
>>> nx_graph.nodes[3]['group'] = 10
>>> nx_graph.add_node(20, size=20, title='couple', group=2)
>>> nx_graph.add_node(21, size=15, title='couple', group=2)
>>> nx_graph.add_edge(20, 21, weight=5)
>>> nx_graph.add_node(25, size=25, label='lonely', title='lonely node', group=3)
>>> nt = Network('500px', '500px')
# populates the nodes and edges data structures
>>> nt.from_nx(nx_graph)
>>> nt.show('nx.html')
```



Example 1 : Visualizing a Game of Thrones character network

```
from pyvis.network import Network
import pandas as pd

got_net = Network(height="750px", width="100%", bgcolor="#222222", font_color="white")

# set the physics layout of the network
got_net.barnes_hut()
got_data = pd.read_csv("../notebooks/NetworkOfThrones.csv")

sources = got_data['Source']
targets = got_data['Target']
weights = got_data['Weight']

edge_data = zip(sources, targets, weights)

for e in edge_data:
    src = e[0]
    dst = e[1]
    w = e[2]

    got_net.add_node(src, src, title=src)
    got_net.add_node(dst, dst, title=dst)
    got_net.add_edge(src, dst, value=w)

neighbor_map = got_net.get_adj_list()

# add neighbor data to node hover data
for node in got_net.nodes:
    node["title"] += " Neighbors:<br>" + "<br>".join(neighbor_map[node["id"]])
    node["value"] = len(neighbor_map[node["id"]])

got_net.show("gameofthrones.html")
```

<https://raw.githubusercontent.com/melaniewalsh/sample-social-network-datasets/master/sample-datasets/game-of-thrones/got-edges.csv>

Example 2 : Visualization

```
import networkx as nx
from pyvis.network import Network
import random # Add this line to import the 'random' module

# Generate a random graph with 20 nodes and 40 edges
G = nx.gnm_random_graph(n=20, m=40)

# Assign random weights to edges
for edge in G.edges():
    G[edge[0]][edge[1]]['weight'] = round(random.uniform(1, 10), 2)

# Create a Pyvis network from the NetworkX graph
net = Network(notebook=True)

# Add nodes and edges to the Pyvis network
net.from_nx(G)

# Customize node and edge attributes
net.set_edge_smooth('dynamic')
net.show_buttons(filter_=['nodes', 'edges'])
net.force_atlas_2based(gravity=-50, central_gravity=0.05, spring_length=100,
spring_strength=0.15)

# Save the interactive visualization as an HTML file
net.show('random_graph.html')
```

Example 3 : Customizing the Pyvis Interactive Network Graphs

<https://www.askpython.com/python/examples/customizing-pyvis-interactive-network-graphs>

Análise de Redes – Atividade Prática 3

- Execute os seguintes tutoriais sobre o Streamlit:
- https://github.com/napoles-uach/streamlit_network/
- https://github.com/napoles-uach/streamlit_network/blob/main/pyvis_sample.ipynb

Alternativamente:

<https://towardsdatascience.com/building-interactive-network-graphs-using-pyvis-5b8e6e25cf64>

Análise de Redes – Atividade Prática 4

- A partir de um dataset de sua escolha, elabore um notebook contendo um fluxo de análise exploratória, contendo os principais conceitos de análise de redes vistos até o momento a serem integrados no Streamlit com Pyvis:

- Matriz de adjacência.
- Diâmetro e periferia da rede.
- Histograma de distribuição empírica de grau.
- Coeficiente de clustering local para nós escolhidos.
- Coeficiente de clustering global.
- Componentes Conectados Fortemente
- Componentes Conectados Fracamente.
- Degree centrality.
- Closeness centrality.
- Betweenness centrality.
- Eigenvector centrality.
- Assortatividade geral da rede.