



Министерство науки и высшего образования Российской Федерации  
Калужский филиал  
федерального государственного бюджетного  
образовательного учреждения высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(КФ МГТУ им. Н.Э. Баумана)

**ФАКУЛЬТЕТ ИУК «Информатика и управление»**

**КАФЕДРА ИУК4 «Программное обеспечение ЭВМ, информационные технологии»**

## **ДОМАШНЯЯ РАБОТА №1**

### **«Анализ социальных сетей»**

**ДИСЦИПЛИНА: «Технологии анализа данных»**

Выполнил: студент гр. ИУК4-82Б \_\_\_\_\_ ( Карельский М.К. )  
(Подпись)

Проверил: \_\_\_\_\_ ( Ерохин И.И. )  
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2024

**Цель:** формирование практических навыков для анализа социальных сетей.

**Задачи:** ознакомиться с анализом социальных сетей посредством языка Python.

**Задание:**

Необходимо получить список друзей одной из своих социальных сетей. С помощью networkx визуализировать эти данные в виде графа, где узлами являются ваши друзья, а ребрами связь между ними (есть ли они в друзьях друг у друга). Посредством методов networkx построить график распределения степени узлов, а также посчитать 3 стандартные меры центральностей: betweenness, closeness, eigenvector.

В качестве результата работы необходимо построить граф и график и отобразить их в Jupyter Notebook. Также необходимо сохранить эти графики в png файле. Для каждого узла записать в txt файл значения центральностей в виде (node - betweenness - closeness – eigenvector).

**Листинг:**

```
import networkx as nx
import matplotlib.pyplot as plt
import csv

def load_friend_data(file_path):
    friend_data = {}
    with open(file_path, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            node = row[0]
            friends = row[1:]
            friend_data[node] = friends
    return friend_data

def save_centralities_to_file(centralities, file_path):
    with open(file_path, 'w') as file:
        for node, centrality_values in centralities.items():
            file.write(f"{node} - {centrality_values['betweenness']} - {centrality_values['closeness']} - {centrality_values['eigenvector']}\n")

friend_data = load_friend_data('friends.csv')

G = nx.Graph()
for node, friends in friend_data.items():
    for friend in friends:
        G.add_edge(node, friend)

plt.figure(figsize=(10, 8))
nx.draw(G, with_labels=True)
plt.savefig("friend_network_graph.png")
```

```

plt.close()

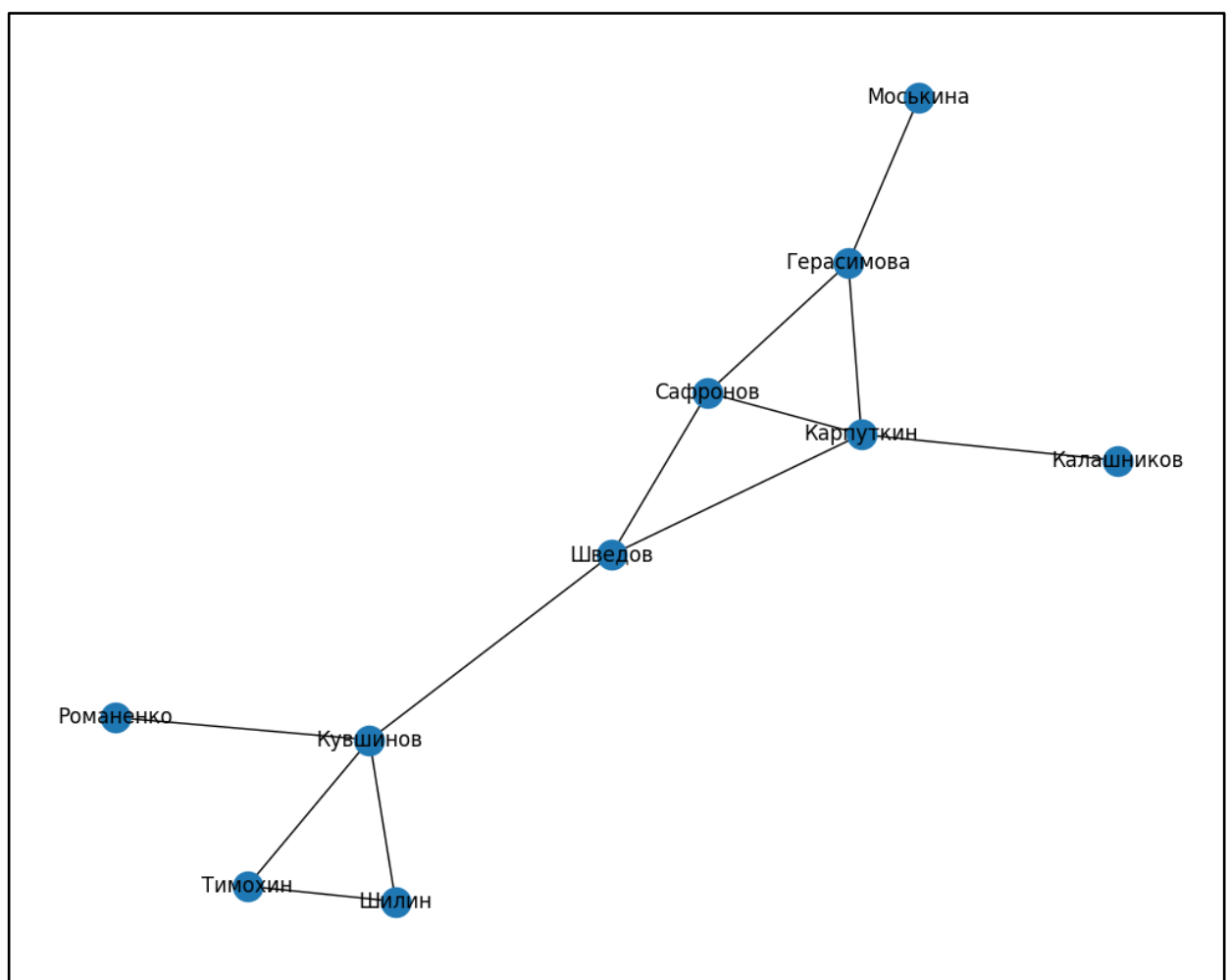
betweenness = nx.betweenness centrality(G)
closeness = nx.closeness centrality(G)
eigenvector = nx.eigenvector centrality(G)

centralities = {}
for node in G.nodes():
    centralities[node] = {
        'betweenness': betweenness[node],
        'closeness': closeness[node],
        'eigenvector': eigenvector[node]
    }
save_centralities_to_file(centralities, 'centralities.txt')

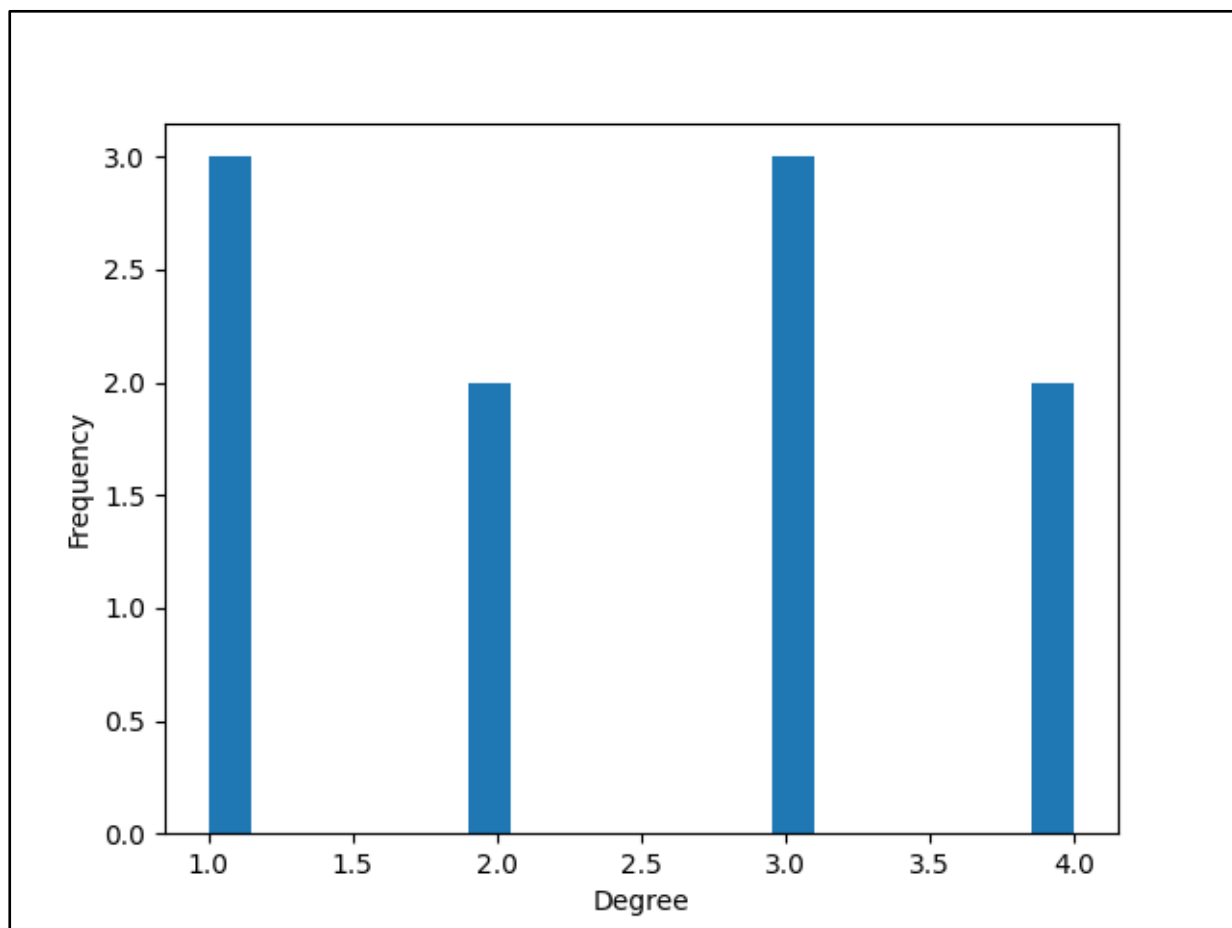
degrees = [G.degree(node) for node in G.nodes()]
plt.hist(degrees, bins=20)
plt.xlabel('Degree')
plt.ylabel('Frequency')
plt.savefig('degree_distribution.png')
plt.close()

```

## Результат:



**Рис. 1.** Граф связей друзей



**Рис. 2.** График распределения степени узлов

centralities.txt

1	Шведов	-	0.5555555555555556	-	0.5625	-	0.43807787136974746
2	Карпуткин	-	0.3611111111111111	-	0.5294117647058824	-	0.5044243668159861
3	Сафронов	-	0.1388888888888889	-	0.5	-	0.45929357718875374
4	Кувшинов	-	0.5555555555555556	-	0.5	-	0.29810752934249557
5	Калашников	-	0.0	-	0.36	-	0.17512449325254878
6	Герасимова	-	0.2222222222222222	-	0.4090909090909091	-	0.3804350241381108
7	Шилин	-	0.0	-	0.36	-	0.15853933799893455
8	Тимохин	-	0.0	-	0.36	-	0.15853933799893455
9	Романенко	-	0.0	-	0.34615384615384615	-	0.10349723111579238
10	Моськина	-	0.0	-	0.3	-	0.13207812732648183

**Рис. 3.** Меры центральностей

**Вывод:** в ходе выполнения лабораторной работы были получены практические навыки для анализа социальных сетей.