|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **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)

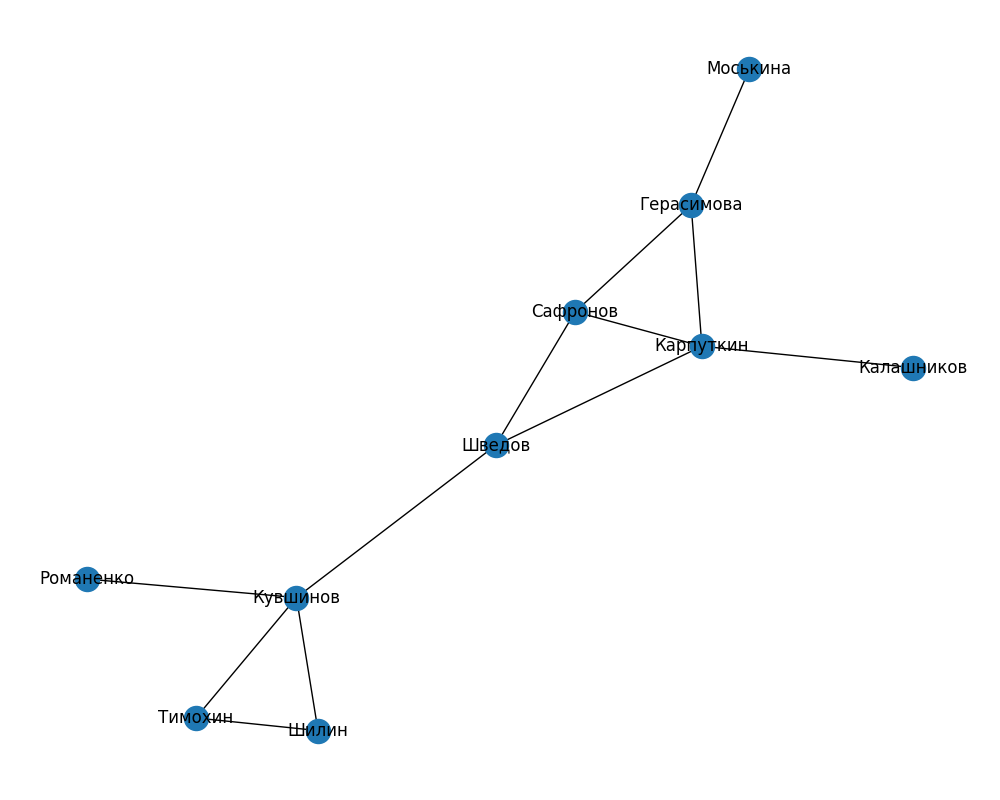
plt.xlabel('Degree')

plt.ylabel('Frequency')

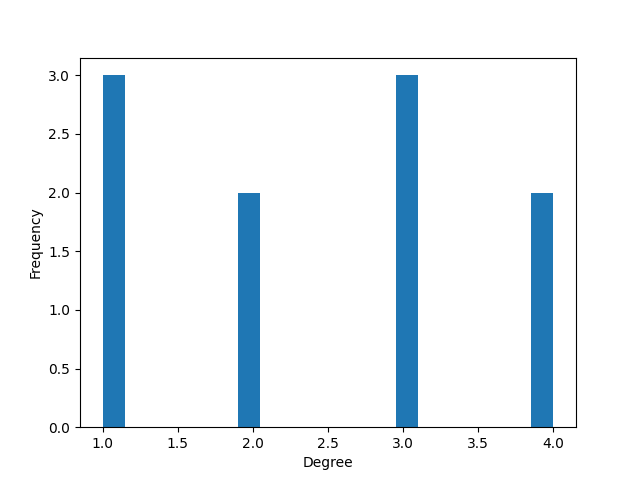
plt.savefig('degree\_distribution.png')

plt.close()

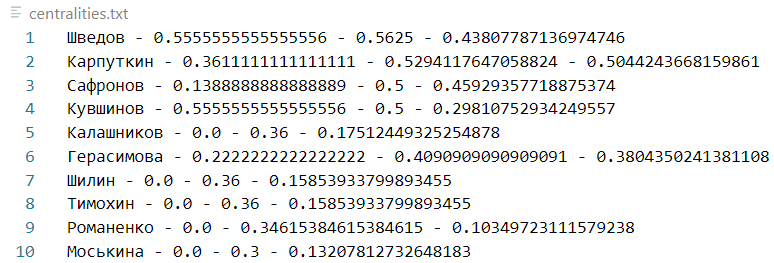
**Результат:**



**Рис. 1.** Граф связей друзей



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



**Рис. 3.** Меры центральностей

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