

Math Project Report

General Outline:

The problem statement is to find the leader from the given data of votes. To solve the problem I used a random walk of the matrix which contained the voting information and based on the frequency of visits concluded the leader.

Algorithm:

1. Collect the input data and store it in the form of a matrix (similar to what was discussed in the class). Each row of the matrix corresponds to one person in the group and the entries in the row correspond to people who have voted for that person.
(0- not voted; 1- voted)
2. Now choose any random person to initiate the process. (In our case it is done by the function: randomNode)
3. Initiate a counter list to all zeros, that will store the number of times a particular person is visited. (frequency of visits. In our case done by the 'count' list)
4. Increment the counter of the first selected person by 1.
5. Now based on the current person selected (the initial node) go to the column of the matrix that corresponds to that person.
6. This column will correspond to all the people our selected person has voted for. So search and choose from this column our next person to visit. The way I achieved this here is by storing the values of the column in a list. This list will have 0's and 1's. Our target is any index with the value 1. So I ran a while loop which will at random choose any number among the indices of the list, and check if the value there is 1. If it is 1 then update the counter of the new person in the count list and return that index. Otherwise, keep going. (This is achieved by the function: nextNeighbour)
7. In the above case to account for people who did not vote I included a condition where if the number of iterations in the while loop exceeded a limit and we still could not find anyone, then simply drop this person and once again choose another random person using the randomNode function.
8. I repeated this process of choosing a person, then visiting a person who that one has voted, updating the counter, and doing it again with the current person for 10000 steps.

9. Finally from the count list, I found the index with the maximum value. The person corresponding to this index is the leader. And I printed out the person's name and a bar graph showing the frequency of visits for the entire group.

Code:

```
import pandas as pd
import numpy as np
import random
import matplotlib.pyplot as plt

d = pd.read_excel("Leader2.xlsx").fillna(0)
a = d.to_numpy()[1:][:,2:] #extracting data from the excel sheet

#this is just for noting down the names of people
k = np.array(d['#'][1:])
n = len(k)
leader = {}
for i in range(n):
    leader[i] = k[i] # a dictionary that has names of people from our
class

n = len(a)
count = [0 for i in a] #this is the array that keeps count of how many
times we visit a person in the random walk

"""
idea: we are storing all the voting info as a matrix
we start at a random node(random person) then among the people he/she
voted we jump to one of them
then repeat the same process for 10000 times and everytime we keep count
of how many times we visited a particular person
result: the person(node) who was visited most is the leader -->
"""

def randomNode(a):
    """
    choose a random neighbor
```

```

    (mainly to select the first person)
    """
    node = random.randint(0,len(a)-1) #among the people in a choose a
random person
    return node

def nextNeighbour(a,node,count):
    """
    from the current node it will return the next node
    (from the current person we jump to one of the people he/she voted
for!)
    """
    n = len(a)
    list = a[:,node] #create a list of all people he voted for
                        #in this list 0 means he didnt vote to 1 means he
voted to
    for i in range(2000): #we took 2000 as random ...take any number of
high steps
                                #this step is useful when the person didnt vote
for anyone
        x = random.randint(0,n-1) #choose a random person in the class
        if list[x]==1:             #see if this person was voted by our
target
            count[x]+=1           #if true then increase the counter
            return x              # return the new person
        #if we cant find next neighbour in due time we teleport
    return(randomNode(a))

#initialize a node -->first person
node = randomNode(a)

#do the random walk! for 10000 steps in our case
for i in range(10000): # we did for 10000 steps(walks)
    node = nextNeighbour(a,node,count)

#find the leader-->find the maximum:
count = np.array(count)
count = count/np.linalg.norm(count)

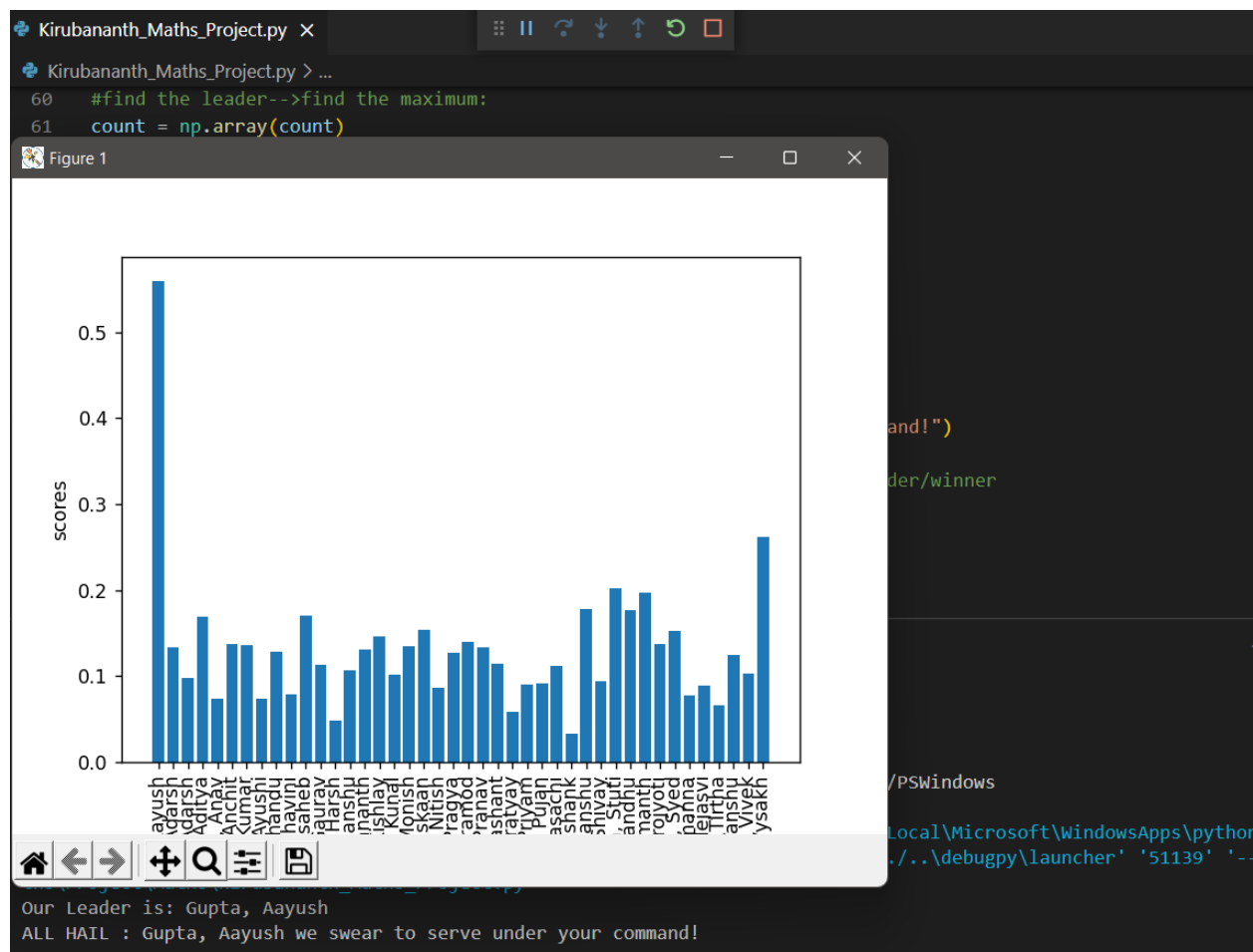
```

```
max_index = 0
max = 0
for i in range(len(count)):
    if max < count[i]:
        max = count[i]
        max_index = i

#print the leader!
print(f"Our Leader is: {leader[max_index]}")
print(f"ALL HAIL : {leader[max_index]} we swear to serve under your
command!")

#this is just to plot votes/count and the person with highest is the
leader/winner
node_list = list(leader.values())
plt.bar(node_list, count)
plt.xlabel("Candidates")
plt.ylabel("scores")
plt.xticks(node_list, rotation=90)
plt.show()
```

Output:



As we can see from the above figure the leader of the class is “Gupta, Ayush”.

References:

1. <https://www.geeksforgeeks.org/reading-excel-file-using-python/> - for learning to use pandas to read the given excel file.
2. <https://www.geeksforgeeks.org/pandas-dataframe-to-numpy-convert-dataframe-to-numpy-array/> - for learning to convert pandas data frame to NumPy array
3. <https://www.geeksforgeeks.org/bar-plot-in-matplotlib/> - for learning to plot bar graph using matplotlib
4. <https://stackabuse.com/rotate-axis-labels-in-matplotlib/> - for learning to rotate x-axis labels.

Readme:

Inside the math project folder, I have included the following:

1. Project report
2. .py file for code
3. Excel file from which the data is being read.