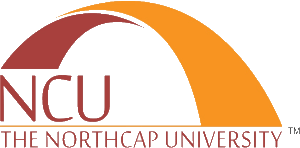
**TEXT AND WEB ANALYTICS ASSIGNMENT**

CASE STUDY

IMPLEMENT HOMOPHILY FOR SOCIAL MEDIA ANALYSIS

|  |  |
| --- | --- |
| Submitted By: | Diksha Madan |
| Class: | VIII-B |
| Roll Number: | 17CSU108 |
| Taught By: | Ms. Vaishali Kalra |



Department of Computer Science and Engineering

School of Engineering and Technology

The Northcap University, Gurugram- 122001, India

Session 2020-21

**DATA CAMP CASE STUDY 6**

Q1. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will calculate the chance homophily for an arbitrary characteristic. Homophily is the proportion of edges in the network whose constituent nodes share that characteristic. How much homophily do we expect by chance? If characteristics are distributed completely randomly, the probability that two nodes x and y share characteristic a is the probability both nodes have characteristic a, which is the frequency of a squared. The total probability that nodes x and y share their characteristic is therefore the sum of the frequency of each characteristic in the network. For example, in the dictionary favorite\_colors provided, the frequency of red and blue is 1/3 and 2/3 respectively, so the chance homophily is (1/3)^2+(2/3)^2 = 5/9.

**100 XP**

* Create a function that takes a dictionary chars with personal IDs as keys and characteristics as values, and returns a dictionary with characteristics as keys, and the frequency of their occurrence as values.
* Create a function chance\_homophily(chars) that takes a dictionary chars defined as above and computes the chance homophily for that characteristic.
* A sample of three peoples' favorite colors is given in favorite\_colors. Use your function to compute the chance homophily in this group, and store as color\_homophily.
* Print color\_homophily.

CODE:

from collections import Counter

def chance\_homophily(chars):

    """

    Computes the chance homophily of a characteristic,

    specified as a dictionary, chars.

    """

    #enter your code here

    chars\_counts\_dict = Counter(chars.values())

    chars\_counts = np.array(list(chars\_counts\_dict.values()))

    chars\_props  = chars\_counts / sum(chars\_counts)

    return sum(chars\_props\*\*2)

favorite\_colors = {

    "ankit":  "red",

    "xiaoyu": "blue",

    "mary":   "blue"

}

color\_homophily = chance\_homophily(favorite\_colors)

print(color\_homophily)

OUTPUT:

Graphical user interface, text

Description automatically generated

Q2. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In the remaining exercises, we will calculate and compare the actual homophily in these village to chance. In this exercise, we subset the data into individual villages and store them.

**Instructions**

**100 XP**

* individual\_characteristics.dta contains several characteristics for each individual in the dataset such as age, religion, and caste. Use the pandas library to read in and store these characteristics as a dataframe called df.
* Store separate datasets for individuals belonging to Villages 1 and 2 as df1 and df2, respectively.
  + Note that some attributes may be missing for some individuals. In this case study, we will ignore rows of data where some column information is missing.
* Use the head method to display the first few entries of df1.

CODE :

import pandas as pd

df  = pd.read\_stata(data\_filepath + "individual\_characteristics.dta")

df1 = df[df["village"]==1]# Enter code here!

df2 = df[df["village"]==2]# Enter code here!

# Enter code here!

df1.head()

OUTPUT:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

Q3. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we define a few dictionaries that enable us to look up the sex, caste, and religion of members of each village by personal ID. For Villages 1 and 2, their personal IDs are stored as pid.

**Instructions**

**100 XP**

* Define dictionaries with personal IDs as keys and a given covariate for that individual as values. Complete this for the sex, caste, and religion covariates, for Villages 1 and 2.
* For Village 1, store these dictionaries into variables named sex1, caste1, and religion1.
* For Village 2, store these dictionaries into variables named sex2, caste2, and religion2.

CODE:

sex1      = df1.set\_index("pid")["resp\_gend"].to\_dict()

caste1    = df1.set\_index("pid")["caste"].to\_dict()

religion1 = df1.set\_index("pid")["religion"].to\_dict()

sex2      = df2.set\_index("pid")["resp\_gend"].to\_dict()

caste2    = df2.set\_index("pid")["caste"].to\_dict()

religion2 = df2.set\_index("pid")["religion"].to\_dict()

Q4.Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will print the chance homophily of several characteristics of Villages 1 and 2. The function chance\_homophily is still defined from Exercise 1.

**Instructions**

**100 XP**

* sex1, caste1, religion1, sex2, caste2, and religion2 are already defined from previous exercises. Use chance\_homophily to compute the chance homophily for sex, caste, and religion In Villages 1 and 2. Is the chance homophily for any attribute very high for either village?

CODE :

print("Village 1 chance of same sex:", chance\_homophily(sex1))

# Enter your code here.

print("Village 1 chance of same caste:", chance\_homophily(caste1))

print("Village 1 chance of same religion:", chance\_homophily(religion1))

print("Village 2 chance of same sex:", chance\_homophily(sex2))

print("Village 2 chance of same caste:", chance\_homophily(caste2))

print("Village 2 chance of same religion:", chance\_homophily(religion2))

OUTPUT:

Graphical user interface, text, chat or text message

Description automatically generated

Q5. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will create a function that computes the observed homophily given a village and characteristic.

**Instructions**

**100 XP**

* Complete the function homophily(), which takes a network G, a dictionary of characteristics chars, and node IDs IDs. For each node pair, determine whether a tie exists between them, as well as whether they share a characteristic. The total count of these is num\_same\_ties and num\_ties respectively, and their ratio is the homophily of chars in G. Complete the function by choosing where to increment num\_same\_ties and num\_ties.

CODE:

def homophily(G, chars, IDs):

    num\_same\_ties, num\_ties = 0, 0

    for n1 in G.nodes():

        for n2 in G.nodes():

            if n1 > n2:   # do not double-count edges!

                if IDs[n1] in chars and IDs[n2] in chars:

                    if G.has\_edge(n1, n2):

                        num\_ties += 1# Should `num\_ties` be incremented?  What about `num\_same\_ties`?

                        if chars[IDs[n1]] == chars[IDs[n2]]:

                            num\_same\_ties += 1# Should `num\_ties` be incremented?  What about `num\_same\_ties`?

    return (num\_same\_ties / num\_ties)

Q6. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will obtain the personal IDs for Villages 1 and 2. These will be used in the next exercise to calculate homophily for these villages.

**Instructions**

**100 XP**

* In this dataset, each individual has a personal ID, or PID, stored in key\_vilno\_1.csv and key\_vilno\_2.csv for villages 1 and 2, respectively. data\_filepath contains the base URL to the datasets used in this exercise. Use pd.read\_csv to read in and store key\_vilno\_1.csv and key\_vilno\_2.csv as pid1 and pid2 respectively. The csv files have no headers, so make sure to include the parameter header = None.

CODE:

pid1 = pd.read\_csv(data\_filepath + "key\_vilno\_1.csv", dtype=int, header = None)

pid2 = pd.read\_csv(data\_filepath + "key\_vilno\_2.csv", dtype=int, header = None)

Q7. Network homophily occurs when nodes that share an edge share a characteristic more often than nodes that do not share an edge. In this case study, we will investigate homophily of several characteristics of individuals connected in social networks in rural India.

In this exercise, we will compute the homophily of several network characteristics for Villages 1 and 2, and compare this to chance homophily. The networks for these villages have been stored as networkx graph objects G1 and G2. homophily() and chance\_homophily() are pre-loaded from previous exercises.

CODE :

print("Village 1 observed proportion of same sex:", homophily(G1, sex1, pid1))

# Enter your code here!

print("Village 1 observed proportion of same caste:", homophily(G1, caste1, pid1))

print("Village 1 observed proportion of same religion:", homophily(G1, religion1, pid1))

print("Village 2 observed proportion of same sex:", homophily(G2, sex2, pid2))

print("Village 2 observed proportion of same caste:", homophily(G2, caste2, pid2))

print("Village 2 observed proportion of same religion:", homophily(G2, religion2, pid2))

print("Village 1 chance of same sex:", chance\_homophily(sex1))

print("Village 1 chance of same caste:", chance\_homophily(caste1))

print("Village 1 chance of same religion:", chance\_homophily(religion1))

print("Village 2 chance of same sex:", chance\_homophily(sex2))

print("Village 2 chance of same caste:", chance\_homophily(caste2))

print("Village 2 chance of same religion:", chance\_homophily(religion2))

OUTPUT:

Graphical user interface, text, chat or text message

Description automatically generated