Practical work N°2

Social network

General remarks

1- This lab is an opportunity to discover networkx Python package to represent graphs, to calculate centrality metrics and then apply them on a case study.
2- A report is required to show obtained results with supporting comments.
3- Submission deadline: See deposit space on Moodle.

Part 1: Graph representation in Python

NetworkX is the most popular Python package for manipulating and analyzing graphs. Let's try to create the following simple graph:

1. Create an empty graph
2. Add a single vertex "Mike"
3. Add a bunch of vertices : "Amine", "Rémi", "Nick"
4. Add an edge between 'Mike" and "Amine"
5. Add an edge between "Amine" and "Rémi"
6. Add an edge between "Mike" and "Christophe"
7. Show your graph

Part 2: Calculating centrality metrics using Python

Let's consider a network with 10 vertices and edges distributed as following: **[(7,2), (2,3), (7,4), (4,5), (7,3), (7,5), (1,6),(1,7),(2,8),(2,9)]**

1. Create the associated graph by using networkx package.
2. Calculate its degree centrality
3. Calculate its betweenness centrality
4. Calculate its closeness centrality
5. Comment your result.

Part 3: Fraud analytics

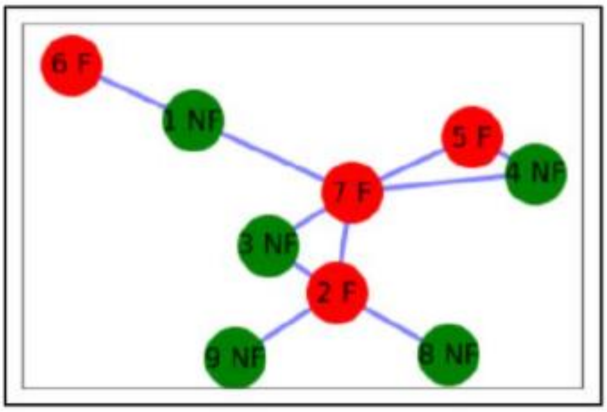
Let's look at how we can use SNA to detect fraud.

The word **homophily** has been coined to represent the effect human social network has on a person. Extending this concept, a homophilic network is a group of people who are likely to be associated with each other due to some common factor; for example, having the same origin or hobbies, being part of the same gang or the same university, or some combination of other factors.

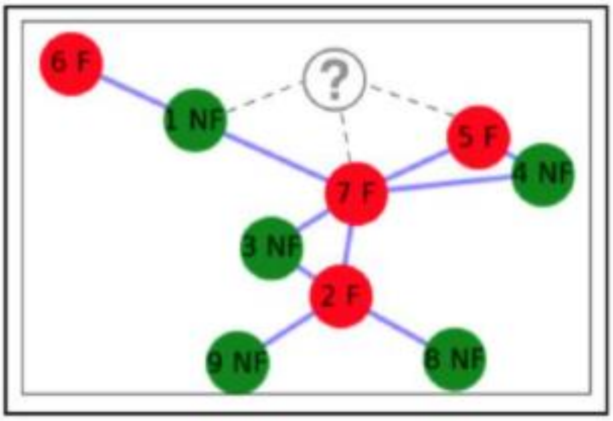
If we want to analyze fraud in a **homophilic network**, we can take advantage of the relationships between the person under investigation and other people in the network, whose risk of involvement in fraud has already been carefully calculated. Flagging a person due to their company is sometimes also called guilt by association.

In an effort to understand the process, let's first look at a simple case.

1. Create a network with nine vertices and eight edges distributed as following: [(7,2), (2,3), (7,4), (4,5), (7,3), (7,5),(1,6),(1,7),(2,8),(2,9)]
2. In this network, four of the vertices are known fraud cases and are classified as **fraud (F).** Five of the remaining people have no fraud-related history and are classified as **non-fraud (NF)**. Define nodes [1,4,3,8,9] as **NF** and [2,5,6,7] as **F**.
3. Add labels and color to your network so that you can show something like that:



Let's assume that we add another vertex, named ***q***, to the network, as shown in the following figure. We have no prior information about this person and whether this person is involved in fraud or not**. We want to classify this person as NF or F based on their links to the existing members of the social network:**



Two ways to do it:

8. Using a simple method that does not use centrality metrics and additional information about the type of fraud;
9. Using a watchtower methodology, which is an advanced technique that uses the centrality metrics of the existing nodes, as well as additional information about the type of fraud

Simple fraud analytics

The simple technique of fraud analytics is based on the assumption that in a network, the behaviour of a person is affected by the people they are connected to. In a network, two vertices are more likely to have similar behaviour if they are associated with each other.
Based on this assumption, we devise a simple technique. If we want to find the probability that a certain node, *a*, belongs to ***F***, the probability is represented by P(F/q) and is calculated as follows:

$$P(F|q) = \frac{1}{degree_q} \sum_{n_j \in Neighborhood_n | class(n_j)=F} w(n, n_j).$$

With $w(n, n_j)$ = the weight of the connection between n and n_j.

The node would be in **F** if ***P(F/q) > T*** (*T* is a given threshold).

1. Calculate the likelihood that node q being involved in fraud.

The watchtower fraud analytics methodology

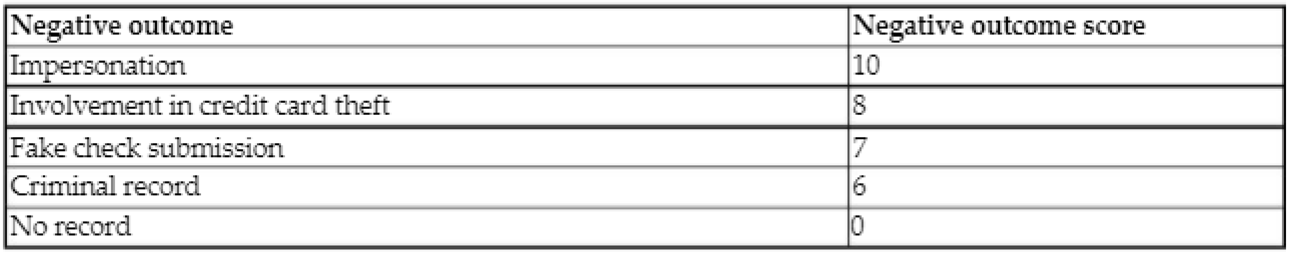
The previous, simple fraud analytics technique has the following two limitations:

- It does not evaluate the importance of each vertex in the social network. A connection to a hub that is involved in fraud may have different implications than a relationship with a remote, isolated person.
- When labeling someone as a known case of fraud in an existing network, we do not consider the severity of the crime.

The watchtower fraud analytics methodology addresses these two limitations employing the following concepts:

1. Scoring negative outcomes

If a person is known to be involved in fraud, we say that there is a negative outcome associated with this individual. The scores are based on an analysis of fraud cases and their impact from historical data.
From a score of 1 to 10, some negative outcomes can be rated as follows:



Negative outcome	Negative outcome score
Impersonation	10
Involvement in credit card theft	8
Fake check submission	7
Criminal record	6
No record	0

2. Degree of suspicion (DOS)

The degree of suspicion (DOS) quantifies our level of suspicion that a person may be involved in fraud. A DOS value of 0 means that this is a low-risk person and a DOS value of 9 means that this is a high-risk person.

Analysis of historical data shows that professional fraudsters have important positions in their social networks. To incorporate this:

- Calculate all of the four centrality metrics of each vertex in our network: degree of centrality, betweenness, closeness and eigenvector.
- Take the average of these vertices.
- If a person associated with a vertex is involved in fraud, illustrate this negative outcome by scoring the person using the pre-determined values shown in the preceding table.
- Multiply the average of the centrality metrics and the negative outcome score to get the value of the DOS.
- Finally, normalize the DOS by dividing it by the maximum value of the DOS in the network.

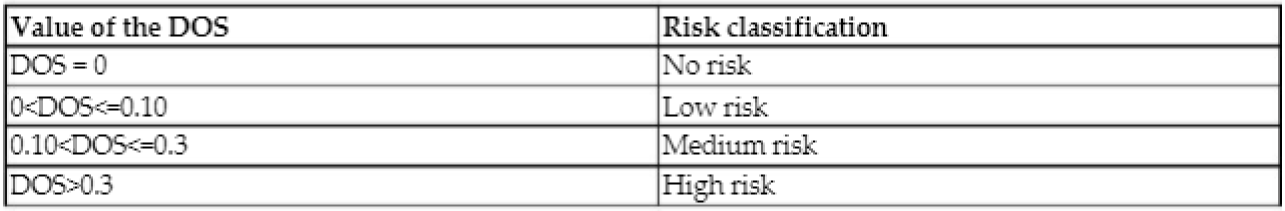
1- Calculate for each vertex, in a given network, its normalized DOS according to the process described above.

3-DOS of a new node

In order to calculate the DOS of the new node that has been added, the following formula is used:

$$DOS_k = \frac{1}{degree_k} \sum_{n_j \in Neighborhood_n} w(n, n_j) DOS_{normalized_j}$$

This will indicate the risk of fraud associated with this new node added to the system. It is possible to create different risk bins for the DOS, as follows:



Value of the DOS	Risk classification
DOS = 0	No risk
0<DOS<=0.10	Low risk
0.10<DOS<=0.3	Medium risk
DOS>0.3	High risk

1- Calculate the DOS of the new node q.