UNIT-4

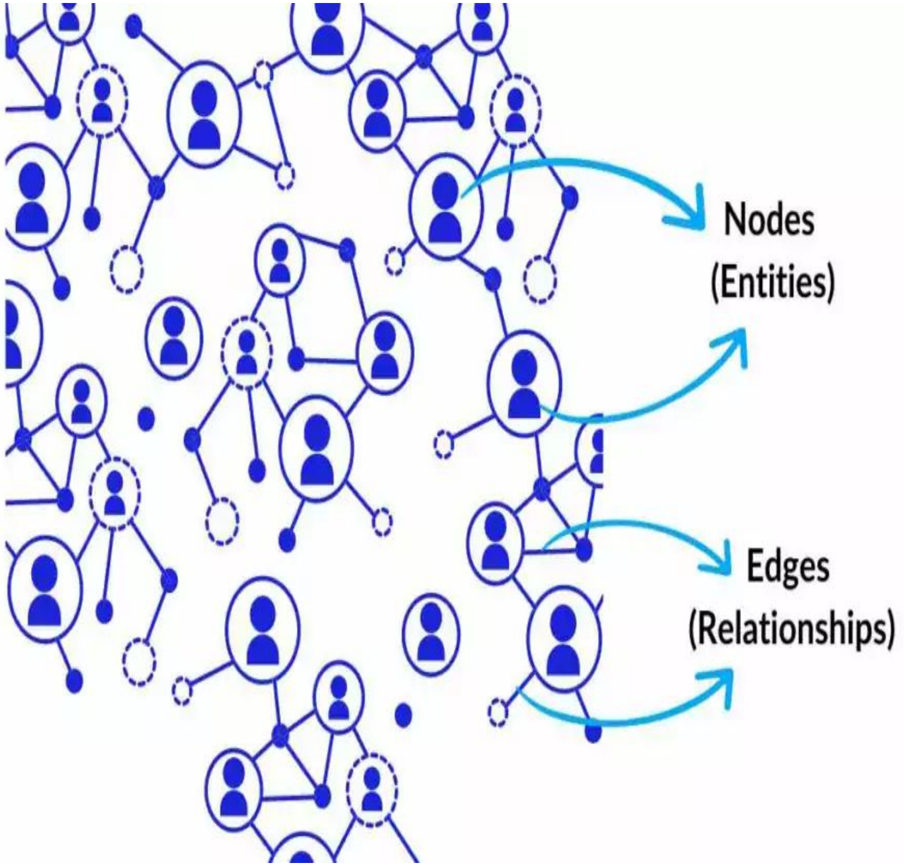
Facebook Friend Recommendation Using Graph Mining

Introduction

- Facebook connects billions of users globally.
- Friend recommendations improve user engagement.
- Graph mining helps analyze social interactions.

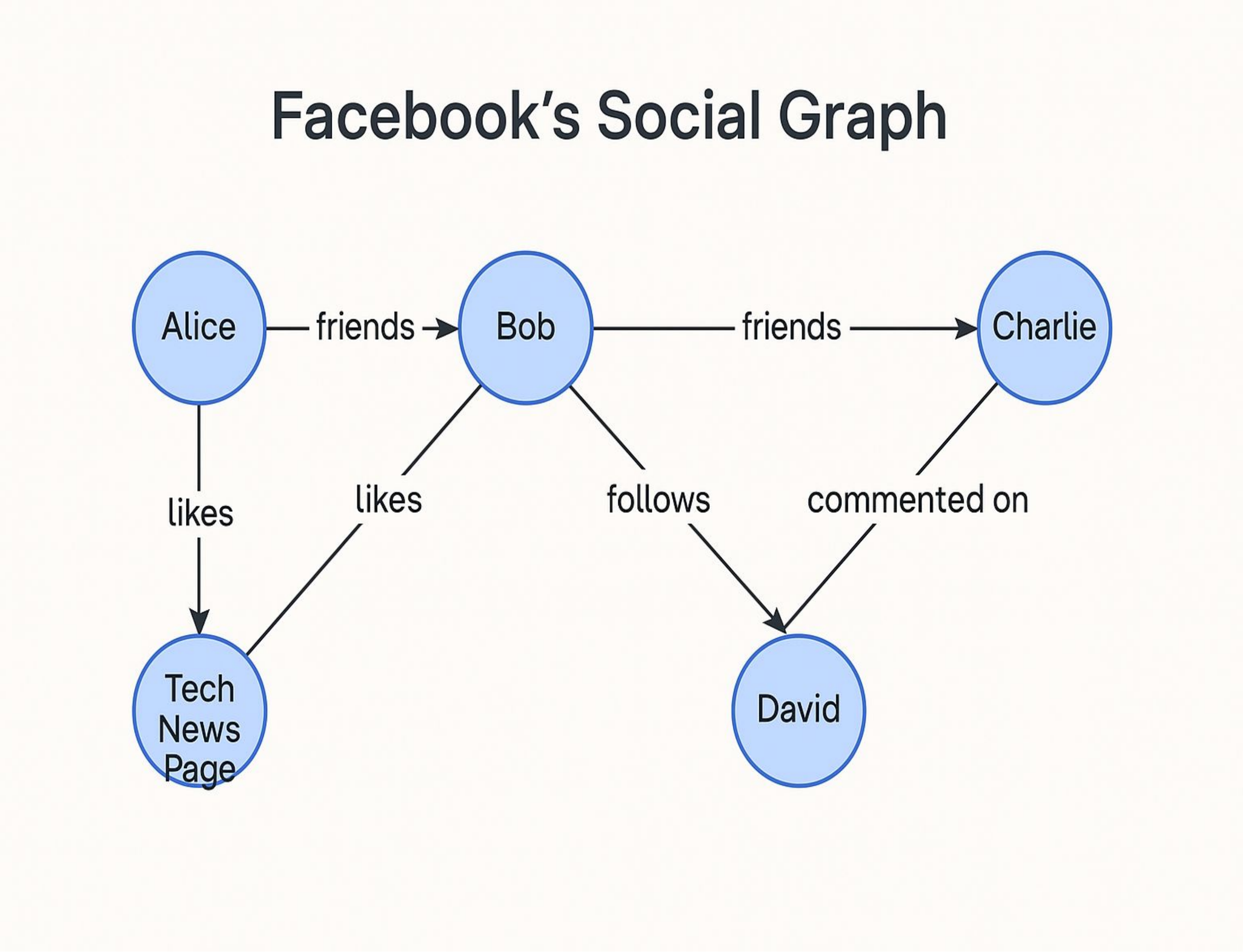
What is Graph Mining?

- Extracting patterns from graph data structures.
- Used for social networks, recommendation systems, and fraud detection.
- Key Graph Elements: **Nodes (Users), Edges (Connections)**.



Facebook's Social Graph

- Users represented as nodes.
- Friendships, interactions as edges.
- Additional attributes: Likes, Comments, Shares.



Data Format & Limitations

- Data columns (2 columns)
  - Source node
  - Destination node
- In total, we have 1862220(1.86 million) vertices/nodes and 9437520(9.43 million) edges/links in our directed graph.
- So, this is a purely graph-based link prediction problem.

- But, as the network grows, people are following new people. The network is very dynamic in real-world because today I may have discovered my old friend on Facebook and started following them. So as far as the problem is concerned, Facebook gave us a snapshot of the graph at one time. So, there are some constraints as we cannot understand the evolution of the graph.

Mapping to a Supervised Classification Problem

Let's map our data to a supervised classification problem.

- Let's say we have vertex Ui and Uj.
  If Ui is following Uj or there is a directed edge between Ui and Uj:
  → Then we will label it as "1".
  If Ui is not following Uj or there is no edge between Ui and Uj:
  → Then we will label it as "0".
  So, we are mapping this to a binary classification task with "0" implying the absence of an edge and "1" implying the presence of a directed edge.

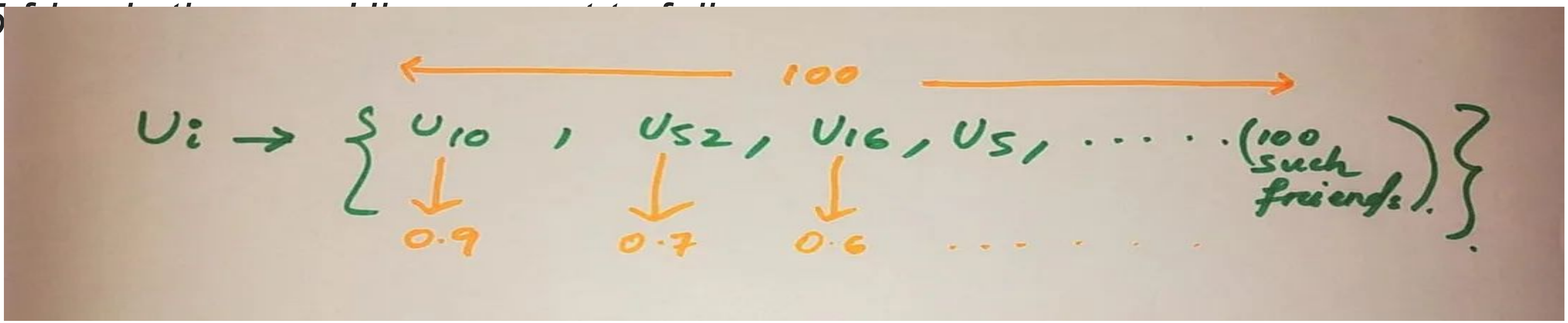
Performance Metric for Supervised Learning

- Both precision and recall are important, hence F1 score is a good choice here.
- We will also go for Confusion matrix.
- Another reasonable metric is Precision@topK.
- Let’s say our K = 10
- Let’s say Ui = {U1, U2, U3, ….,U10}, here these are the top 10 probable vertices or friends Ui may want to follow.
- Now, Precison@top10 means how many of them are actually correct ?
- As in most social networks you don’t get show all the users whom Ui may want to follow, as we have limited space. So, this metric is sensible.

Business Constraints & Metrics

- **No low-latency requirements**
  You can precompute the top 5 or 10 friends Ui should follow once in 2 days or weekly or every night and store it in a hash table-like structure and show it whenever Ui logs in. As we can precompute, so there's no strong latency requirement

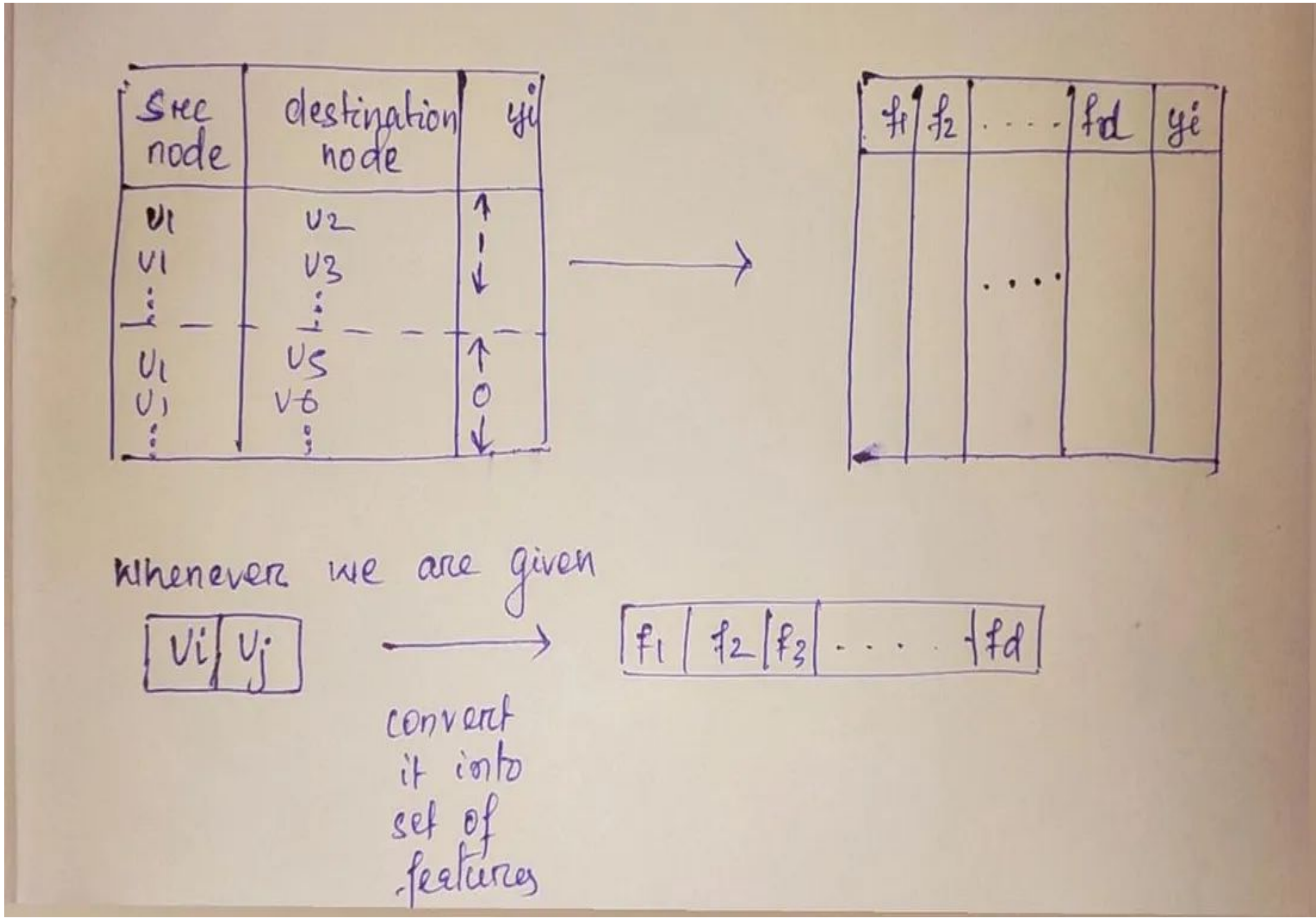
- **We will recommend the highest probability links to a user, so we need to predict the probabilities of the links which are useful.**
  I could have 100 such users which Ui could follow and I can have the probability values. I might have 5 slots or 10 slots, where I want to show the most probable top 5



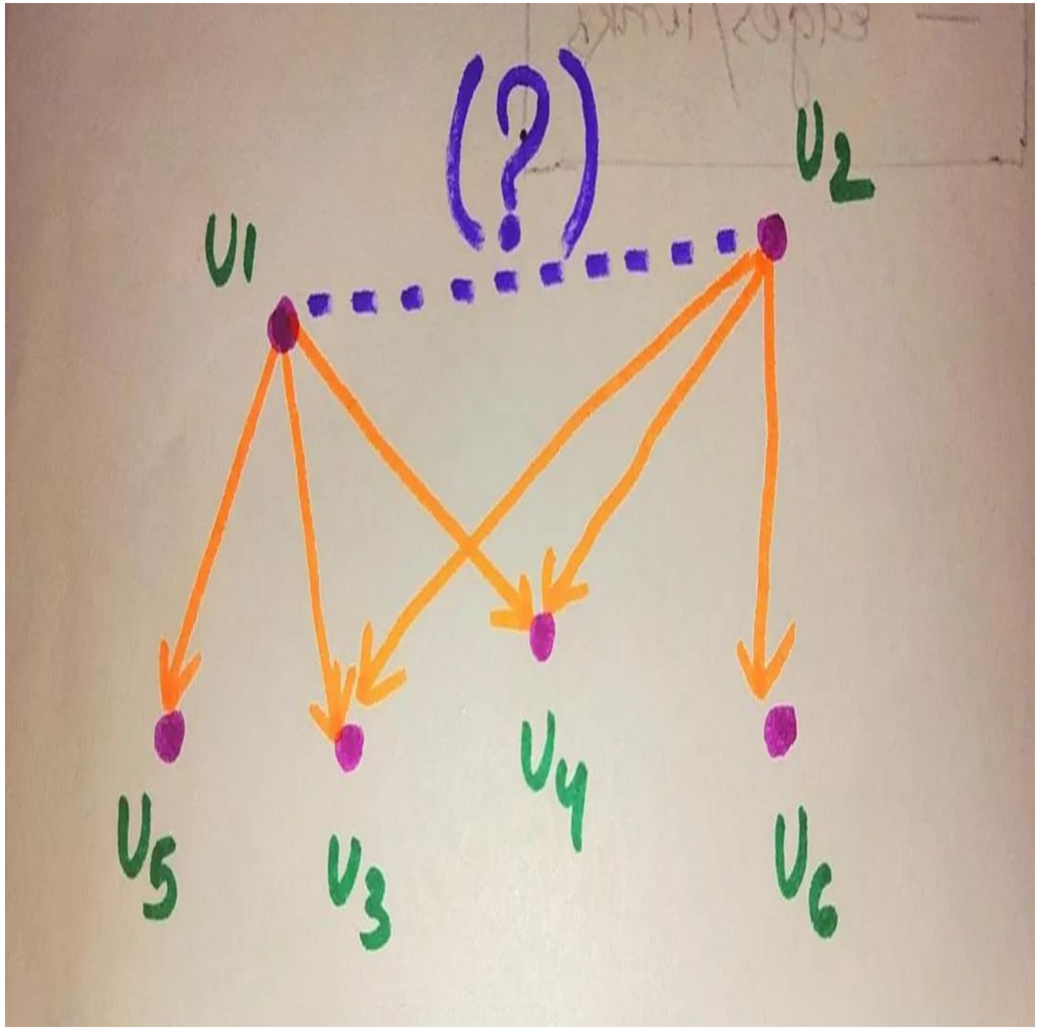
- Ideally, We want high Precision and high Recall when we are recommending Uj to Ui.

Graph : Featurization

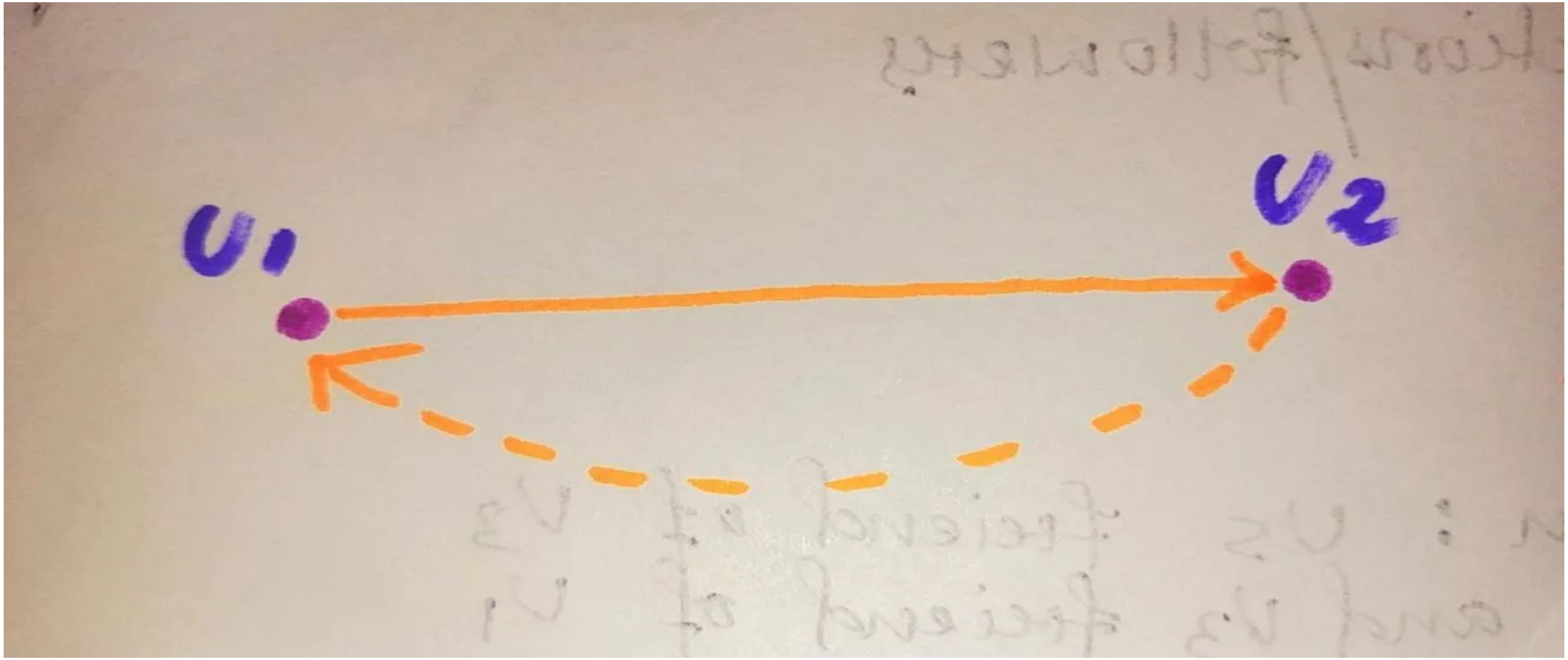
- Now, we will try to convert these pairs of vertices into some numerical, categorical or binary features to carry out machine learning, as we cannot carry out algorithms on the vertices set.
- For any graph-based machine learning problem, of course **featurization** is the most important part of the project.



- Now, how do we **featurize** our data ?
- → ***feature :***
  So, in the below figure, we are trying to predict that if the edge between U1 and U2 should be present or not.
  U1 follows → {U3,U4,U5}
  U2 follows → {U3,U4,U6}
  Here, U1 and U2 are having respective sets of nodes they follow. As they have so many common vertices or two sets are highly overlapped, there is a high chance that U1 and U2 have common interests.
  So, there is a high chance that U1 may want to follow U6 and U2 may want to follow U5.
  Similarly, there is a high chance that U1 could follow U2 and U2 could follow U1.



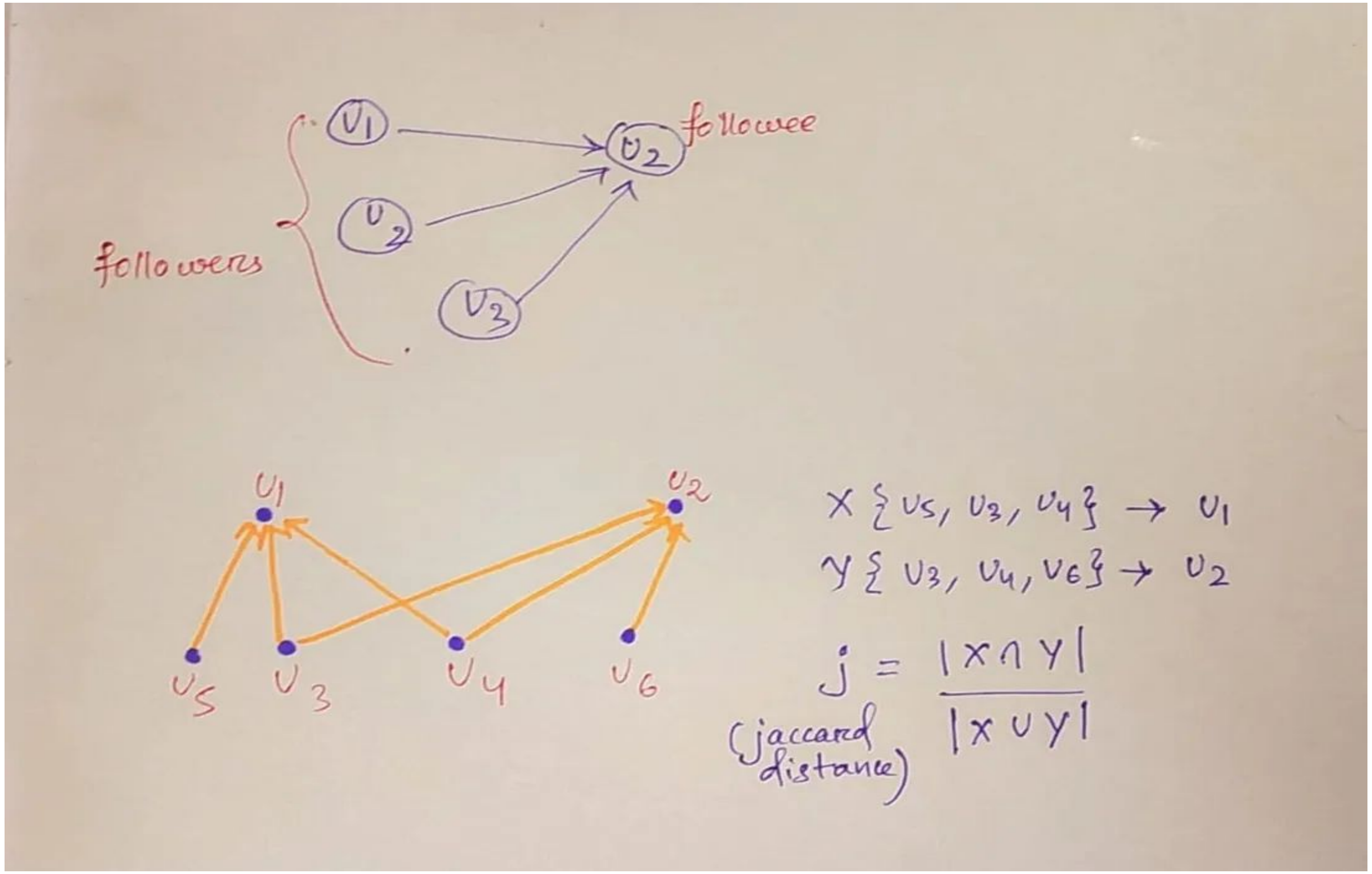
- The fact that U1 is following U2 signifies that there is a high chance that U2 will follow back U1.



- So, these are called graph features.

1) Similarity Measures: Jaccard Distance

- First of all, we will operate on sets of followers and followee.



Given any two sets, jaccard distance or jaccard similarity coefficient basically says:
It is a statistic used for gauging the similarity of sample sets, which is the size of X intersection Y divided by size of X union Y.

2) Similarity Measure : Cosine distance (Otsuka-Ochiai coefficient)

- X is the set of followers of U1
- Y is the set of followers of U2

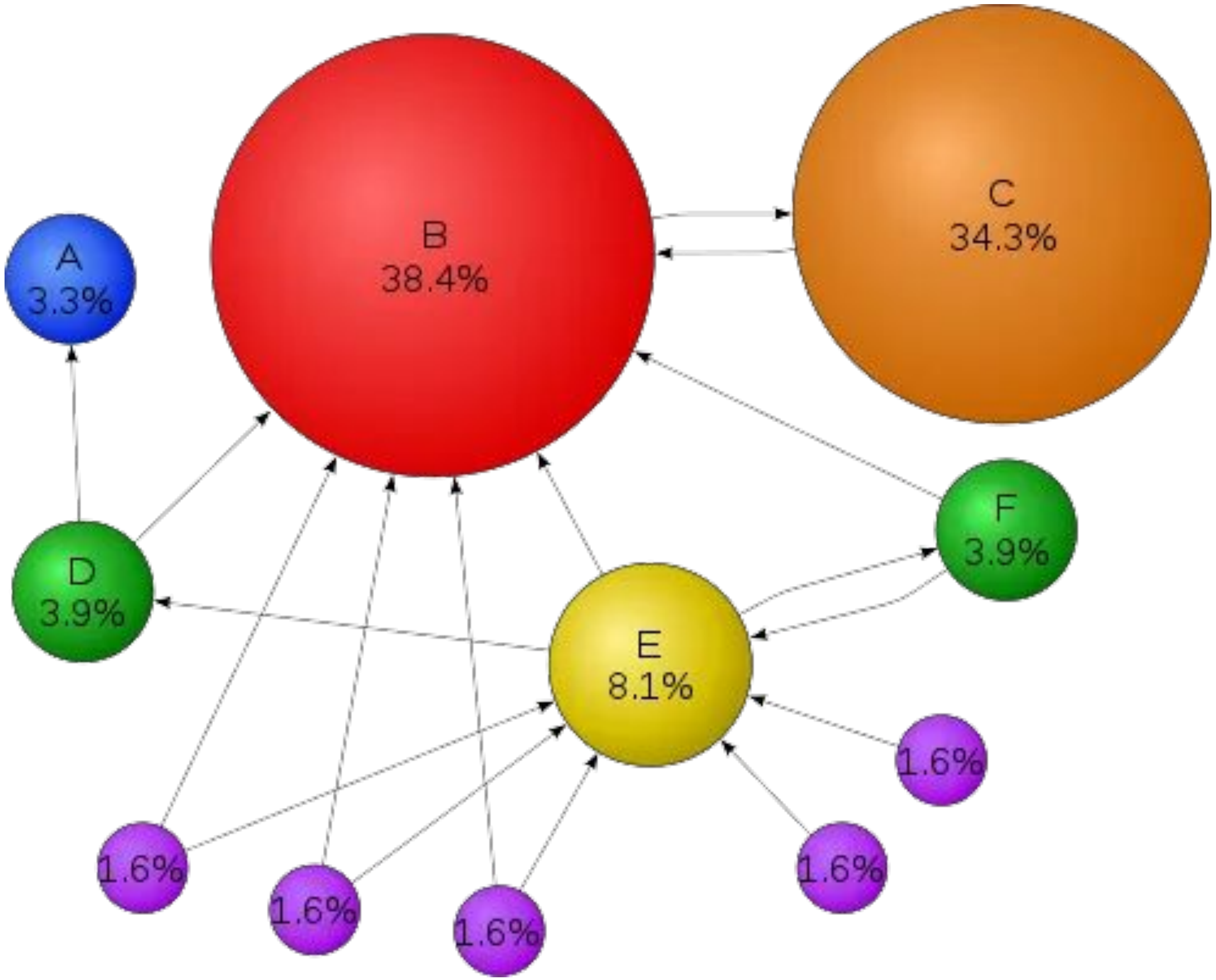
- $Cosine\ Distance = |X \cap Y| / |X| \cdot |Y|,$

  which is used when X and Y are vectors.

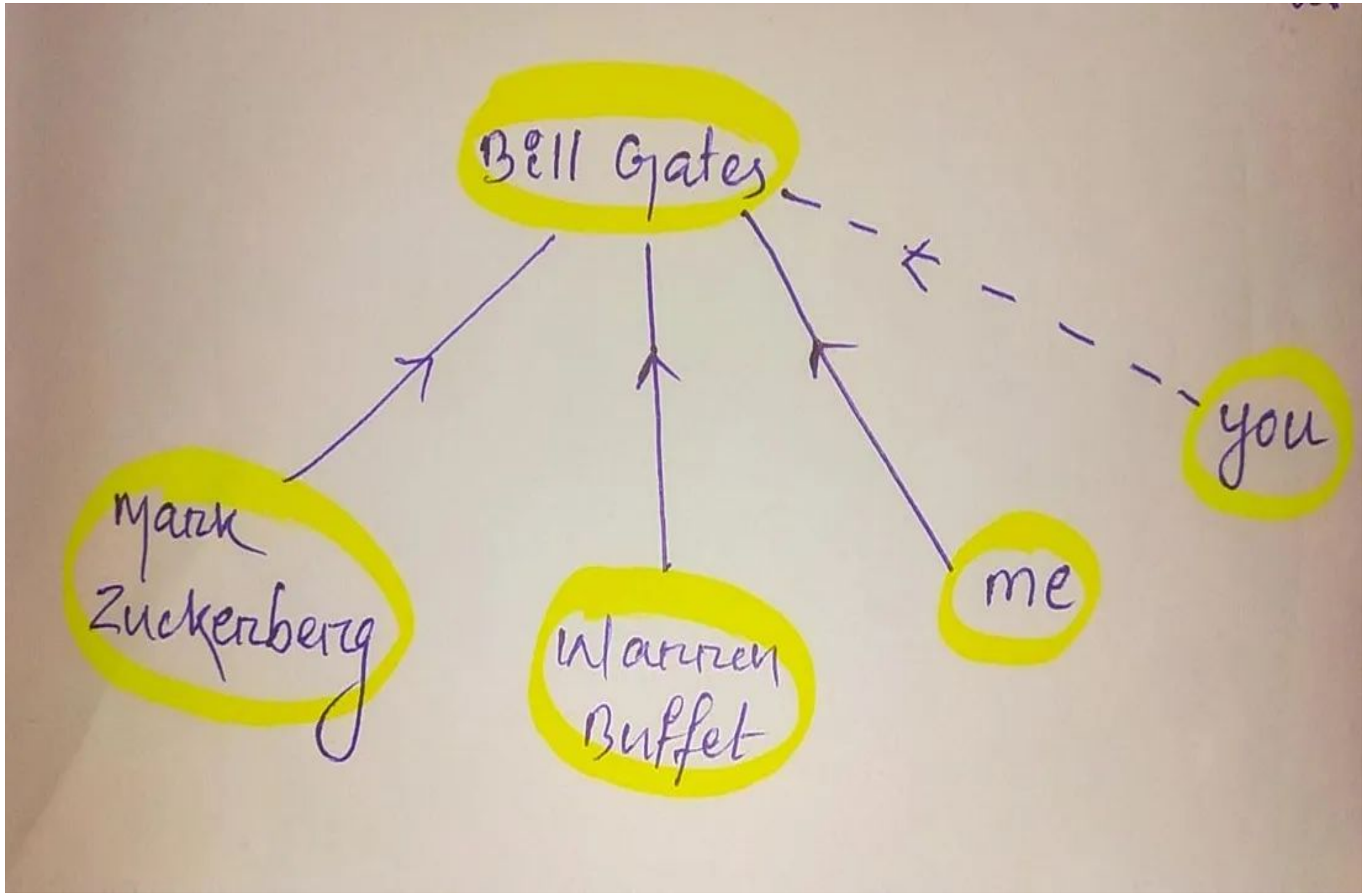
- So, Cosine distance (Otsuka-Ochiai Coefficient) will be high when there is more overlap between sets X and Y.

3) Page Rank

- It is a way of measuring the importance of website pages.
- PageRank works by **counting the number and quality of links to a page** to determine a rough estimate of how important the page is.
- If a lot of pages are having a destination as “B”, then “B” must be important.
  and
  If my page “B” is given as

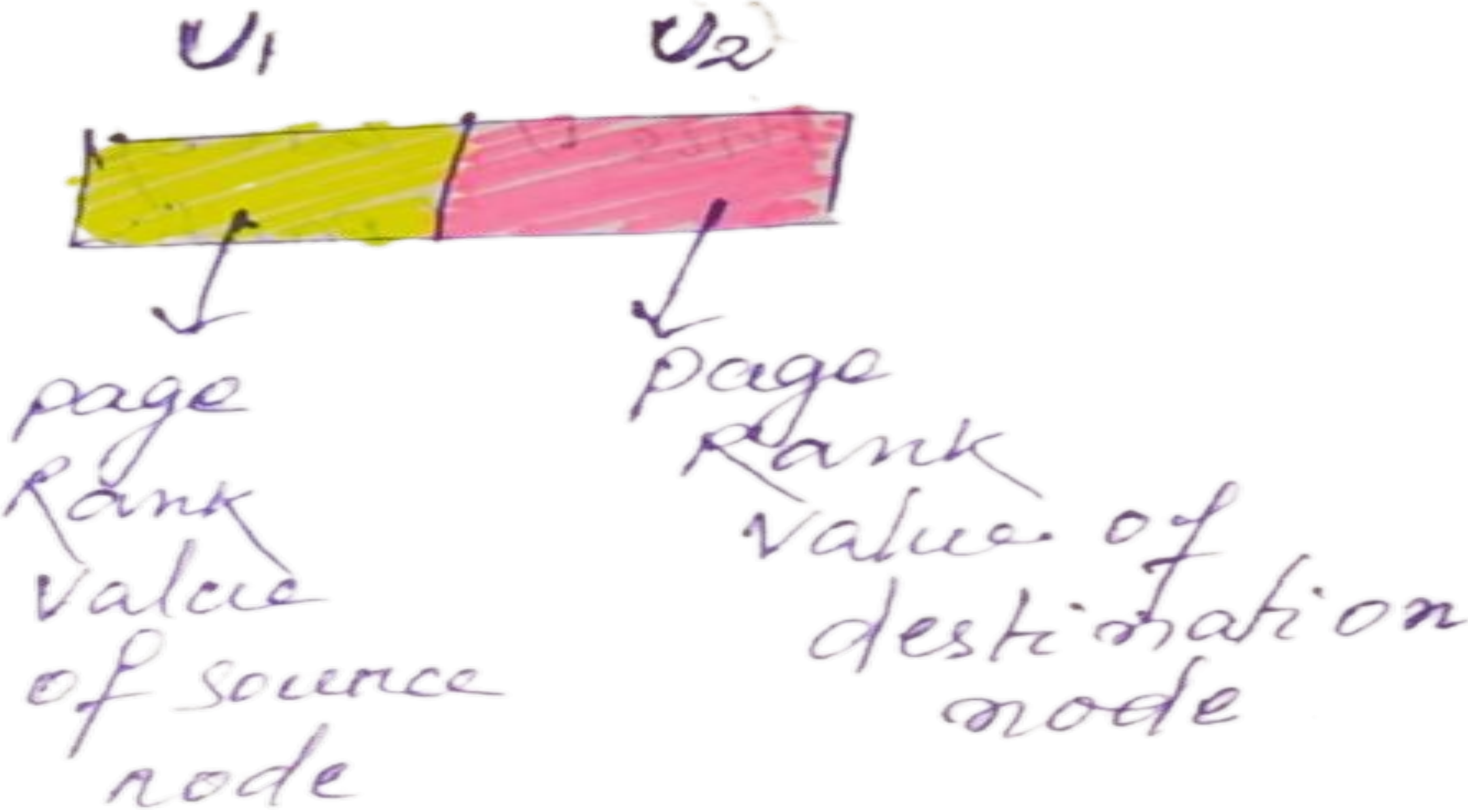


- If a user has high pagerank score, then it implies that other users and highly important users are linking to Ui.
- PageRank can tell us about relative importance.



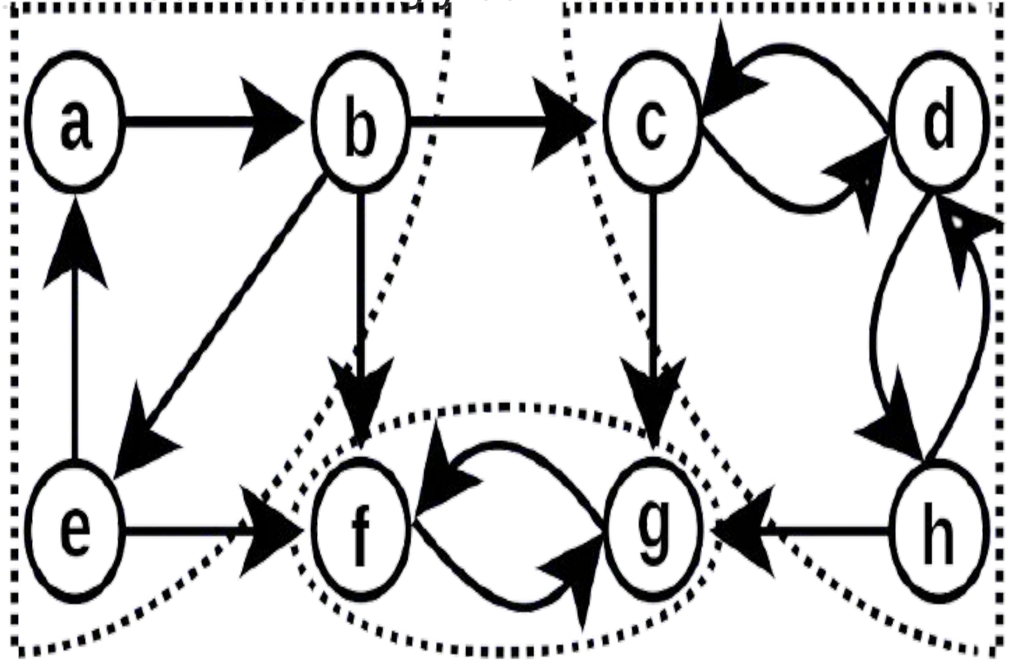
Bill Gates is a celebrity.
He is followed by some important people like Mark and Warren Buffet and also by some common people like me. So it is quite sure that, he is also an important person.
Now there is a significantly higher probability that Bill Gates will be followed by "you".

•The way we will use this for the both vertices is:



5) Connected Components

- **Strongly Connected Component**
  A graph is said to be **strongly connected** if every vertex is reachable from every other vertex. The **strongly connected components** of an arbitrary directed graph form a partition into subgraphs that are themselves strongly connected.



- **Weakly connected component**
  A weakly connected component is one in which all components are connected by some path, ignoring direction.