DE LA SALLE UNIVERSITY

Dr. Andrew L. Tan
Data Science Institute

# Introduction to Graph Neural Networks

GRAPH BASED MACHINE LEARNING MODELS

**FRANCIS ZACCHARIE DE LEON**

BSMS Computer Science
ID121

**MS. JAZZIE JAO**

Adviser
College of Computer Studies

GRAPH NEURAL NETWORKS

# Contents


1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

7 PYTHON CODE

# Contents


1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

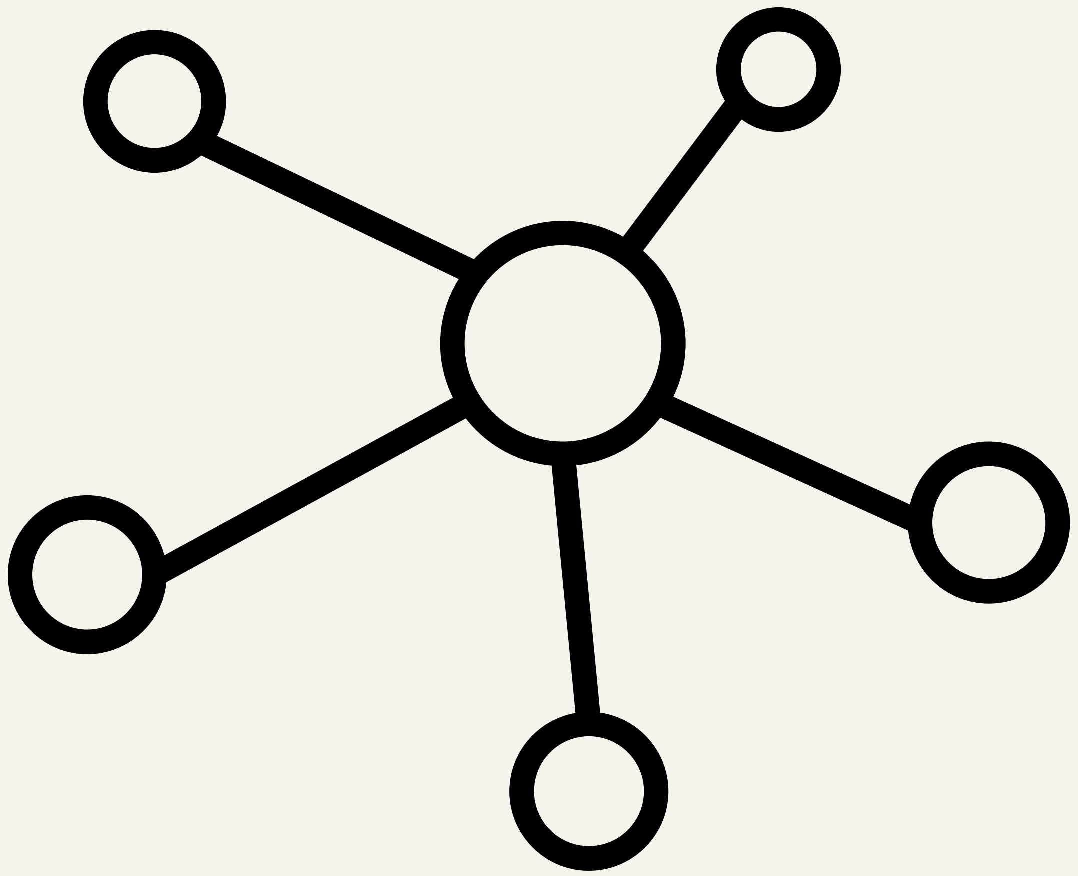
7 PYTHON CODE

# What are Graphs?

A **graph** has **nodes** and **edges**.

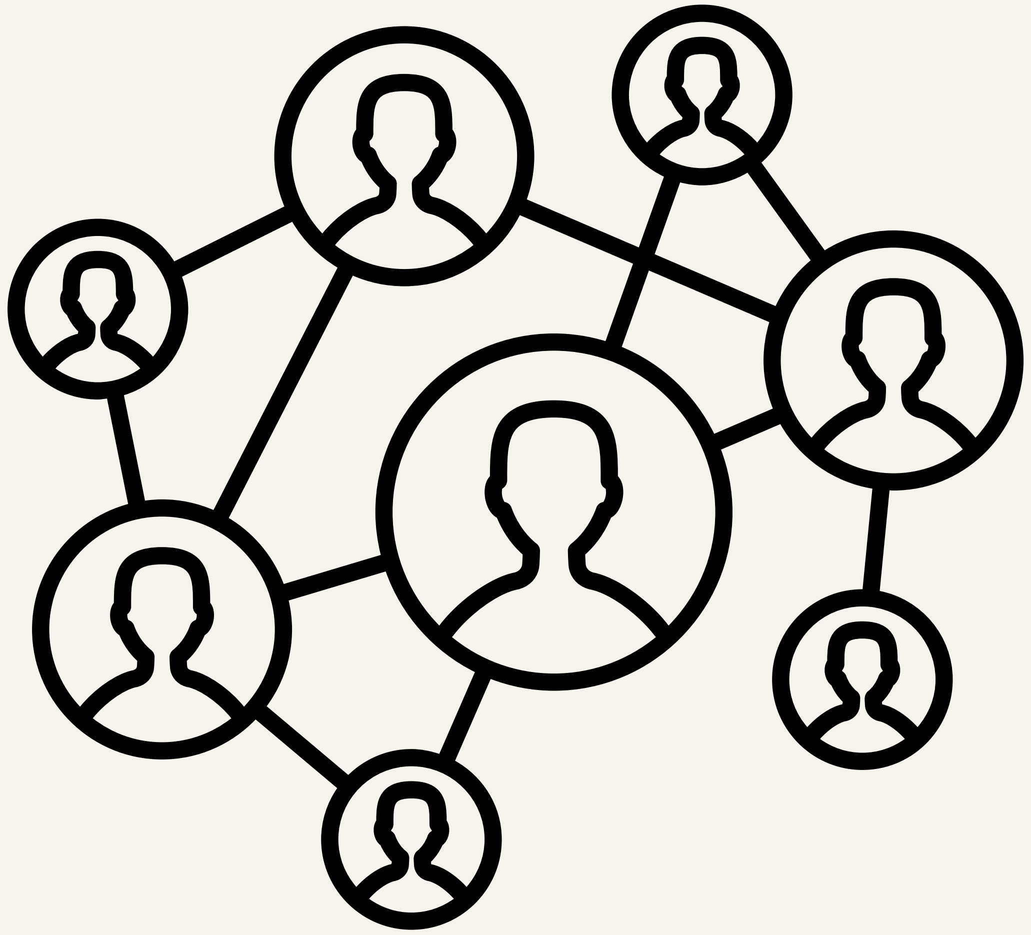
Represented as **G** = **(V, E)**, where:

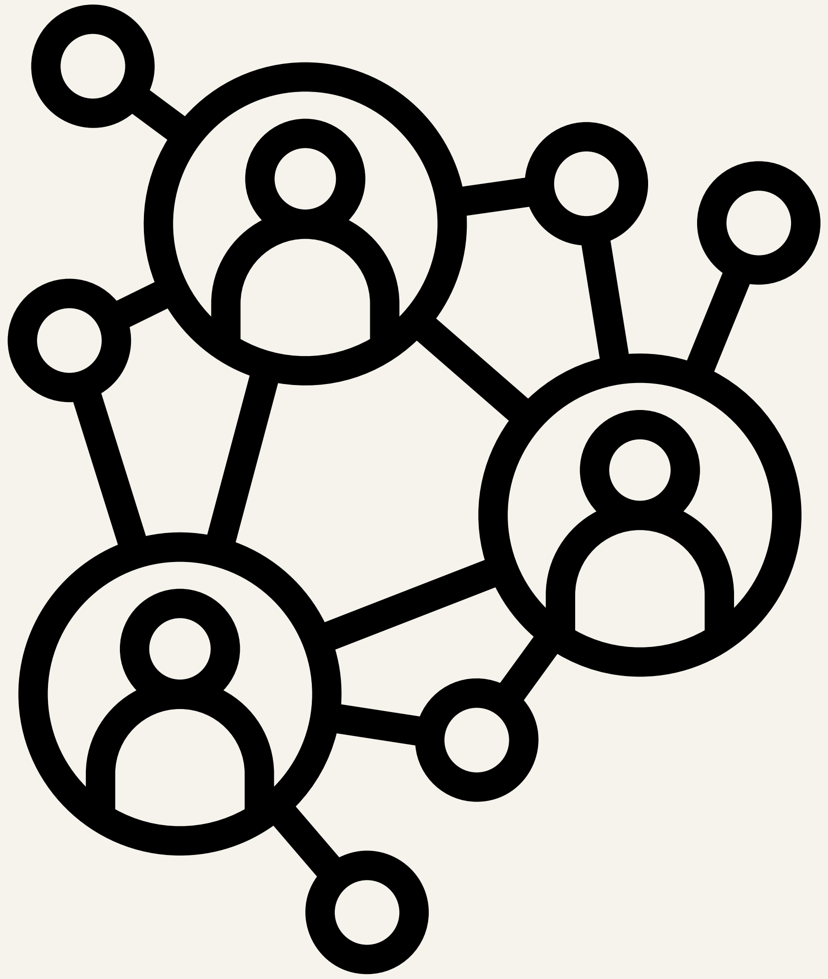
- **V** set of nodes (vertices).
- **E** set of edges (connections).



# Real World Example of Graphs

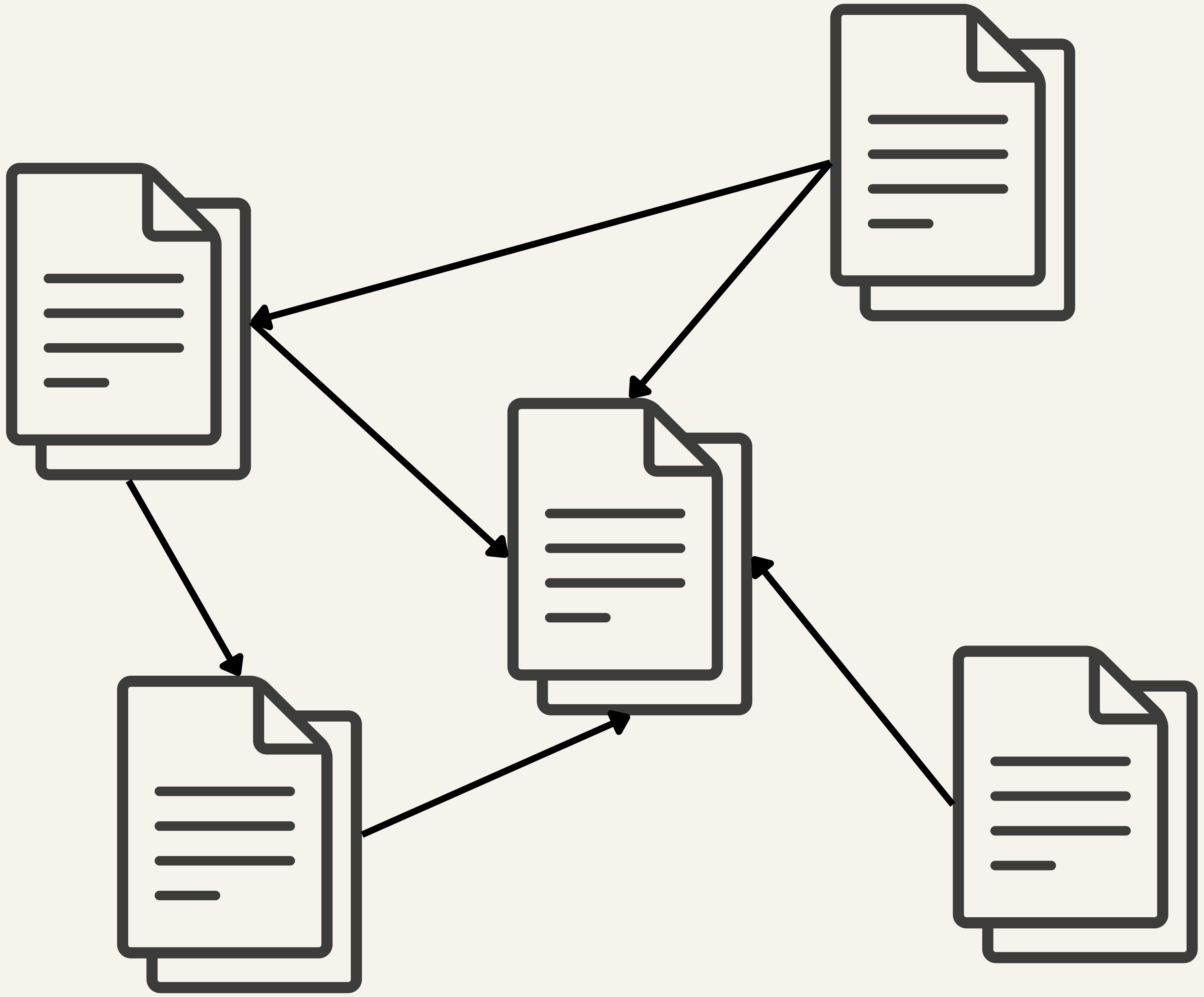
## Social Networks





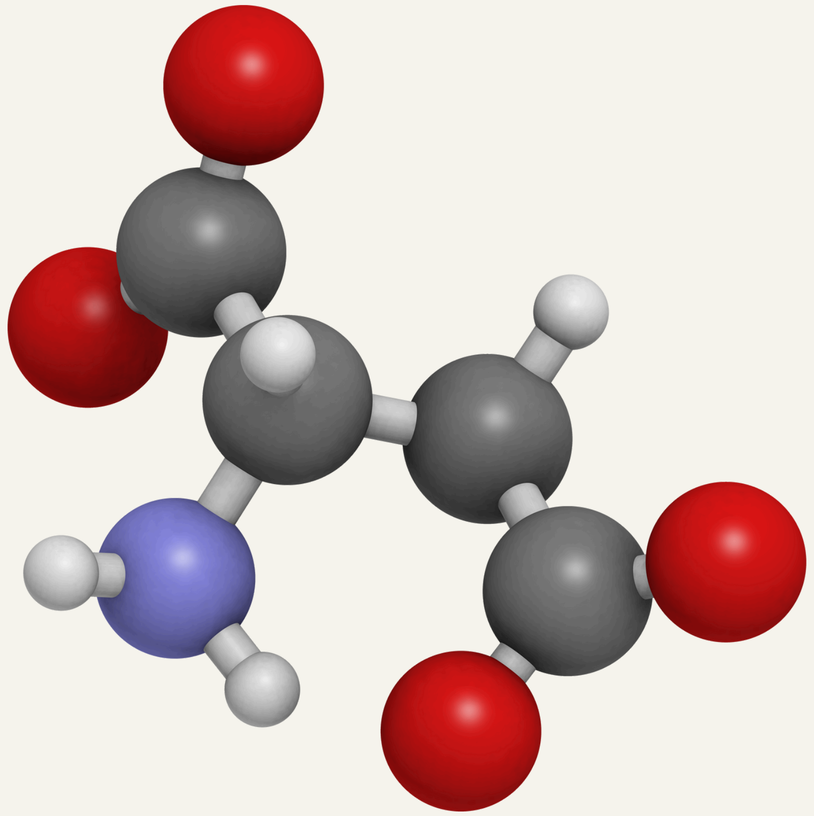
# Real World Example of Graphs

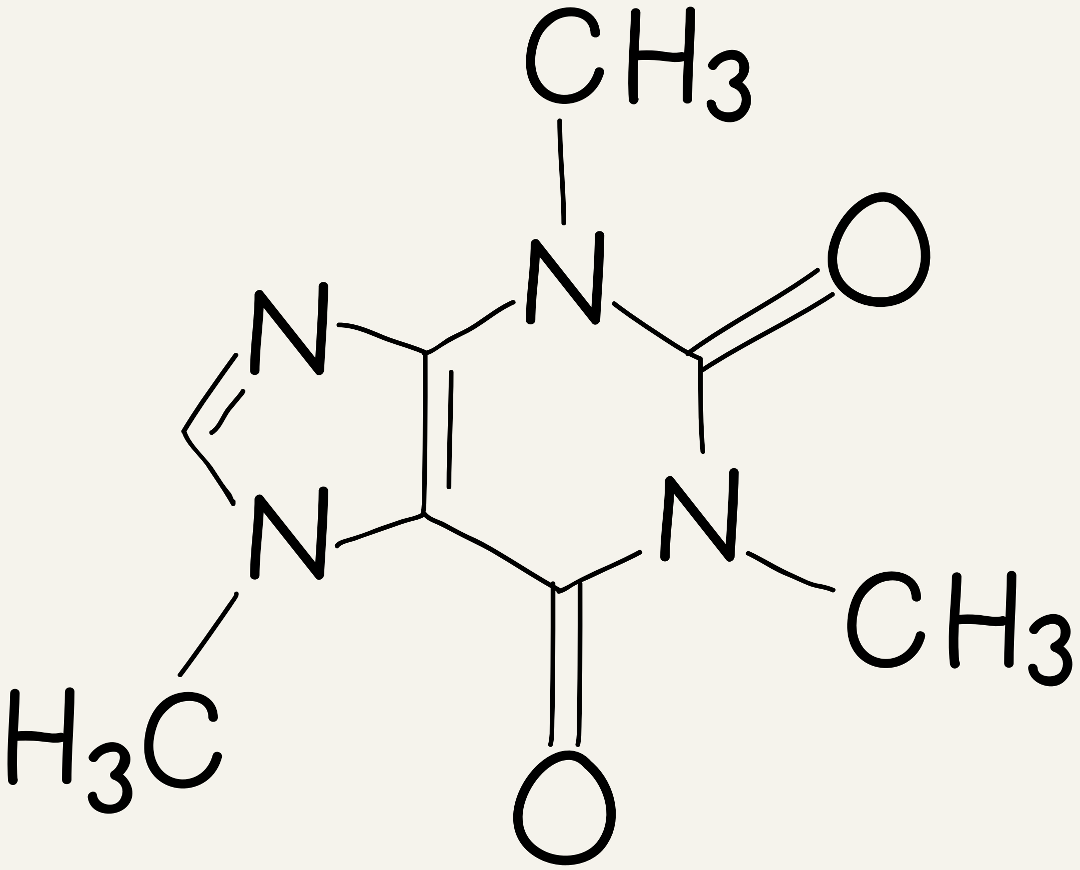
## Citation networks

# Real World Example of Graphs

## Molecular structures



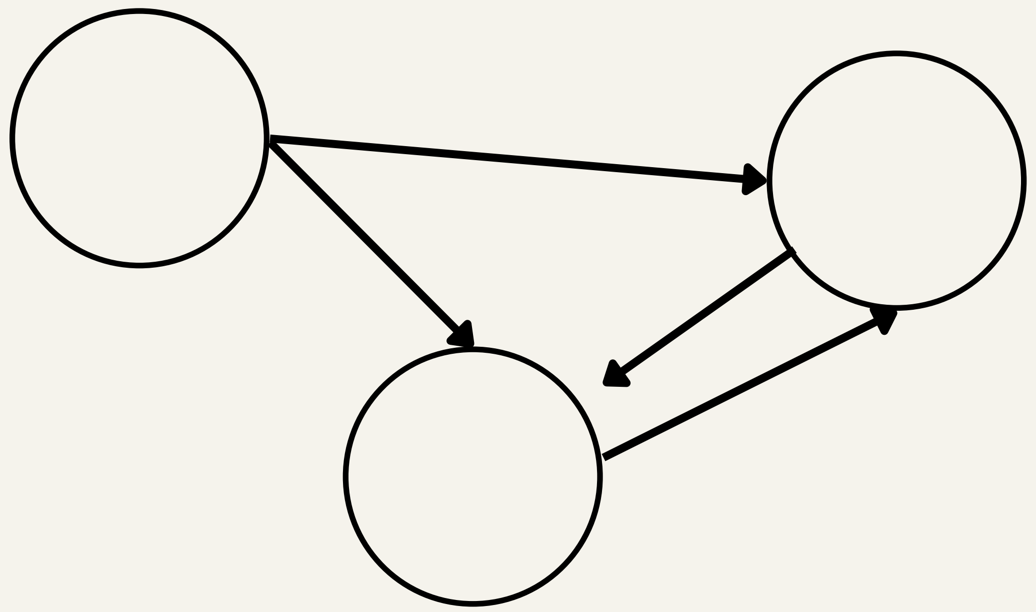


# Types of Graphs

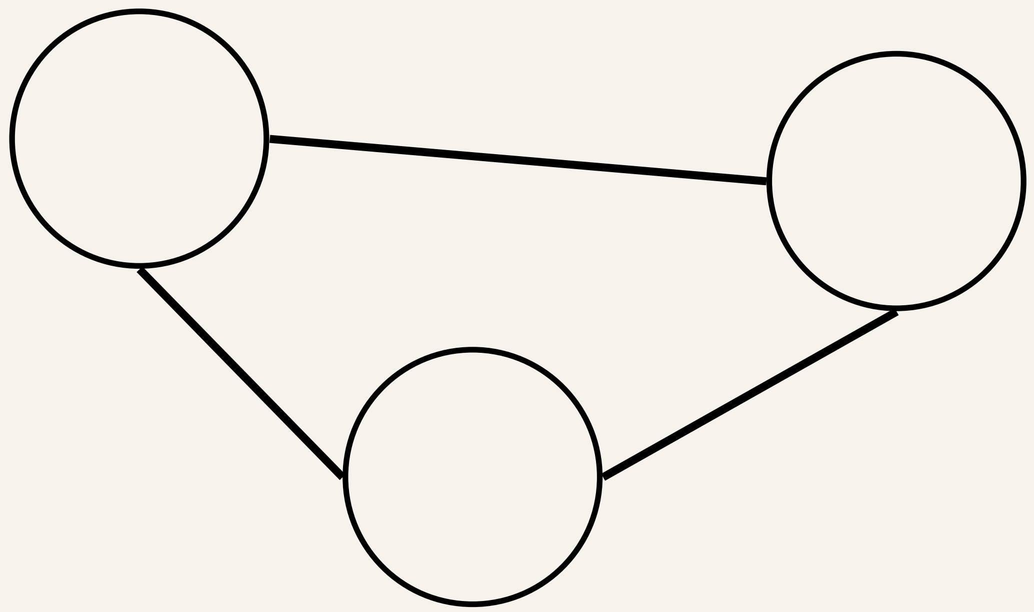
**Directed vs. Undirected Graphs**

- **Directed Graphs**: Edges have a direction (e.g., Twitter follows).
- **Undirected Graphs**: No direction in edges / 2 way (e.g., Facebook friendships).

Directed



Undirected

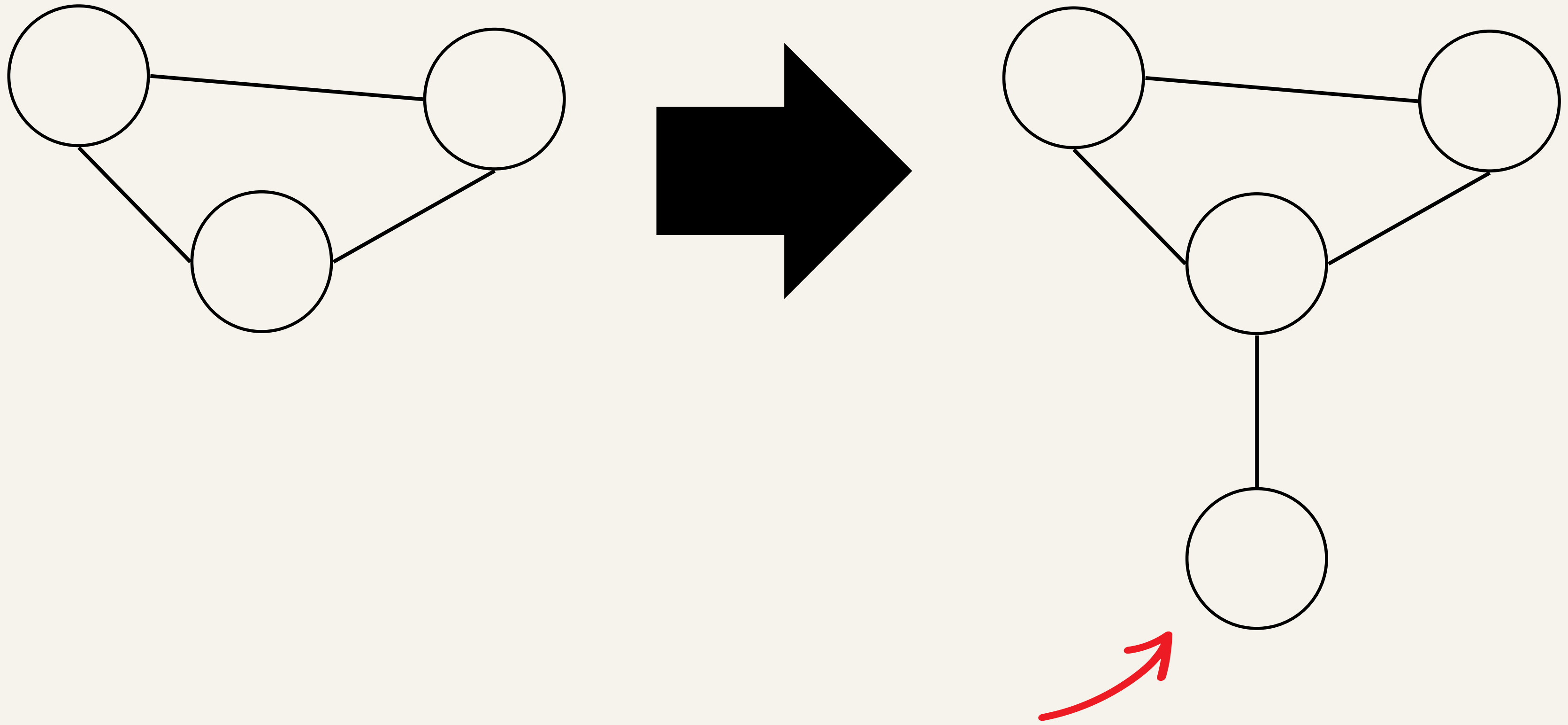


# Types of Graphs

**Static vs. Dynamic Graphs**

- **Static Graphs:** Fixed structure (e.g., molecular graphs).
- **Dynamic Graphs:** Evolve over time (e.g., social networks, traffic networks).

Dynamic
new node and edge



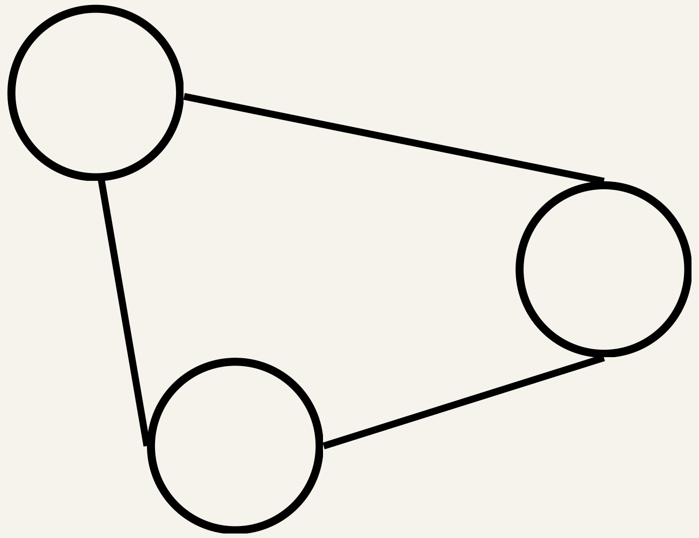
new node and edge!

# Types of Graphs

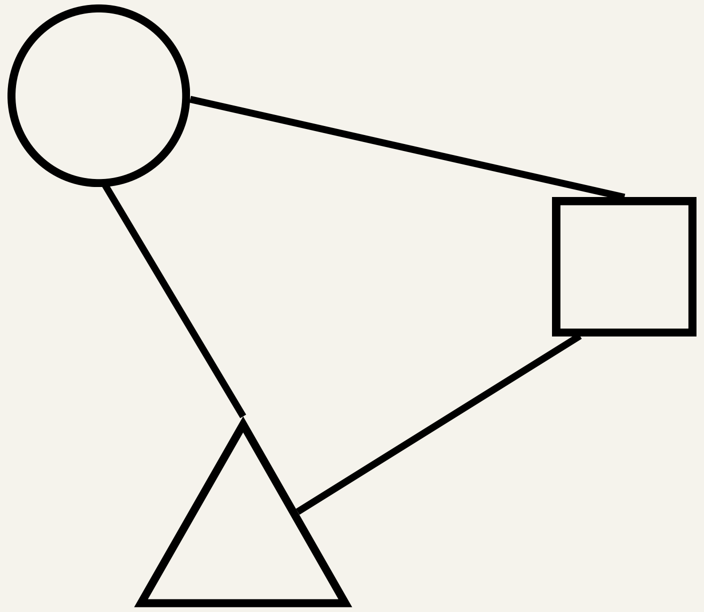
**Homogeneous vs. Heterogeneous**

- **Homogeneous Graphs:** Only one type of node and edge (e.g., social networks).
- **Heterogeneous Graphs:** Multiple node and edge types (e.g., knowledge graphs).

Homogenous graph



Heterogenous graph



# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

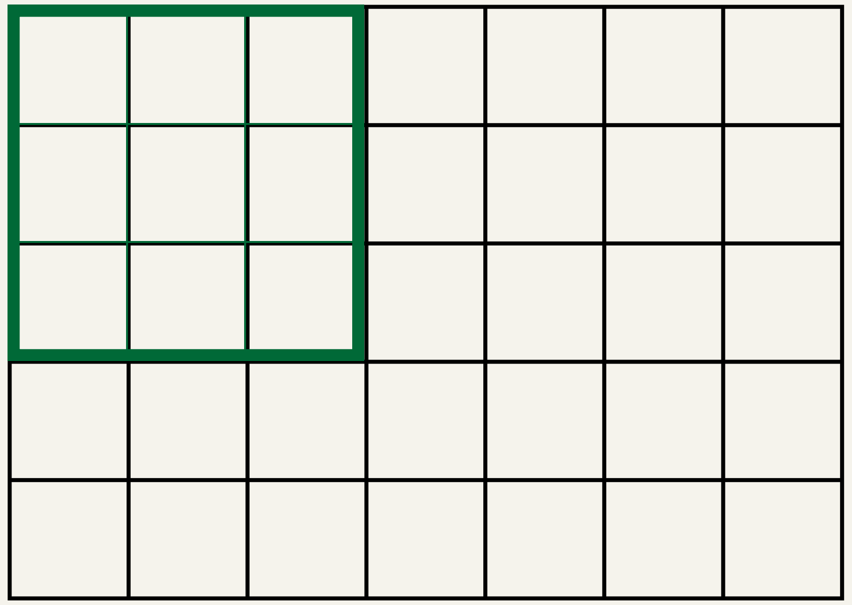
7 PYTHON CODE

GRAPH NEURAL
NETWORKS

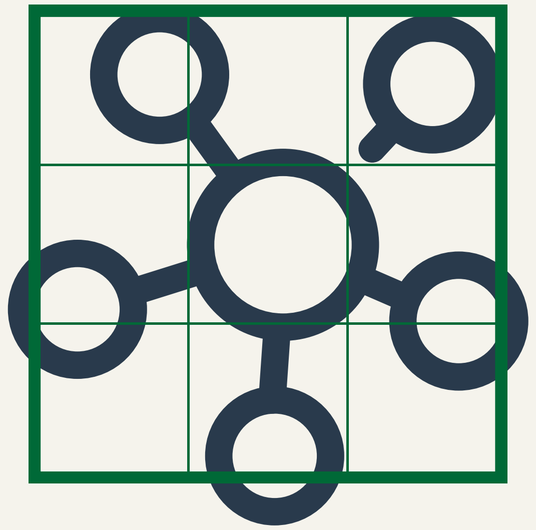
# Limitations of Traditional Neural Networks on Graph Data

Graphs have **irregular structures** with **varying numbers of neighbors** per node. This means that **we cannot** use standard convolutional operations.

CNN used for grid data



CNN cannot be used for graph data



# Limitations of Traditional Neural Networks on Graph Data

Recurrent Neural Networks may also struggle since the **neighbors** of a node **doesn't have an order** to it. Predictions by RNN relies of proper data ordering.

Neighbors have no order.
Which neighbor is considered first?

# Limitations of Traditional Neural Networks on Graph Data

Many machine learning algorithms rely on Euclidean distances, which do not apply well to graphs since edges can use different metrics.

Edges has no distance metric.

# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

7 PYTHON CODE

# Core Concepts of Graph Neural Networks (GNNs)

GNNs learn node representations from **graph-structured data.**

**Key Concept: Message Passing Mechanism**

- Nodes **aggregate information** from their neighbors.
- This process **updates node embeddings** iteratively.

**Three main types of Message Passing in GNNs:**

- **Convolutional GNNs** (e.g., GCN – Graph Convolutional Networks).
- **Attentional GNNs** (e.g., GAT – Graph Attention Networks).
- **General Message Passing GNNs**

# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

**4 MESSAGE PASSING**

5 PREVAILING ARCHITECTURES

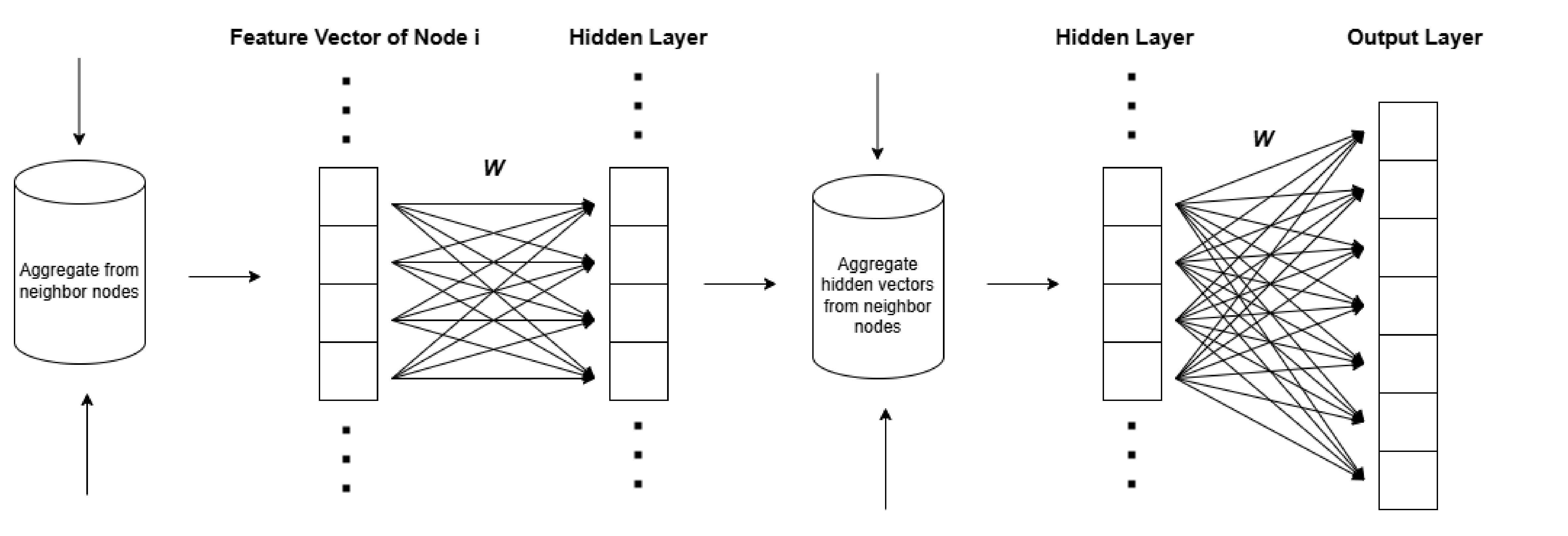
6 WHAT CAN IT DO?

7 PYTHON CODE

# Message Passing Mechanism in Graph Neural Networks

Steps of message passing:

- **Aggregate**: A node gathers information from its neighboring nodes.
- **Update**: The node updates its own representation based on the aggregated information.
- **Repeat**: This process continues for multiple layers, allowing information to spread across the graph.



# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

7 PYTHON CODE

# GNN Architectures

**Graph Convolutional Network**

- Uses graph convolution operations

**Graph Attention Network**

- Allows a node to focus on more relevant neighbors during aggregation.

**GraphSAGE (Sample and Aggregate)**

- Uses sampling to efficiently aggregate information from neighbors.

# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

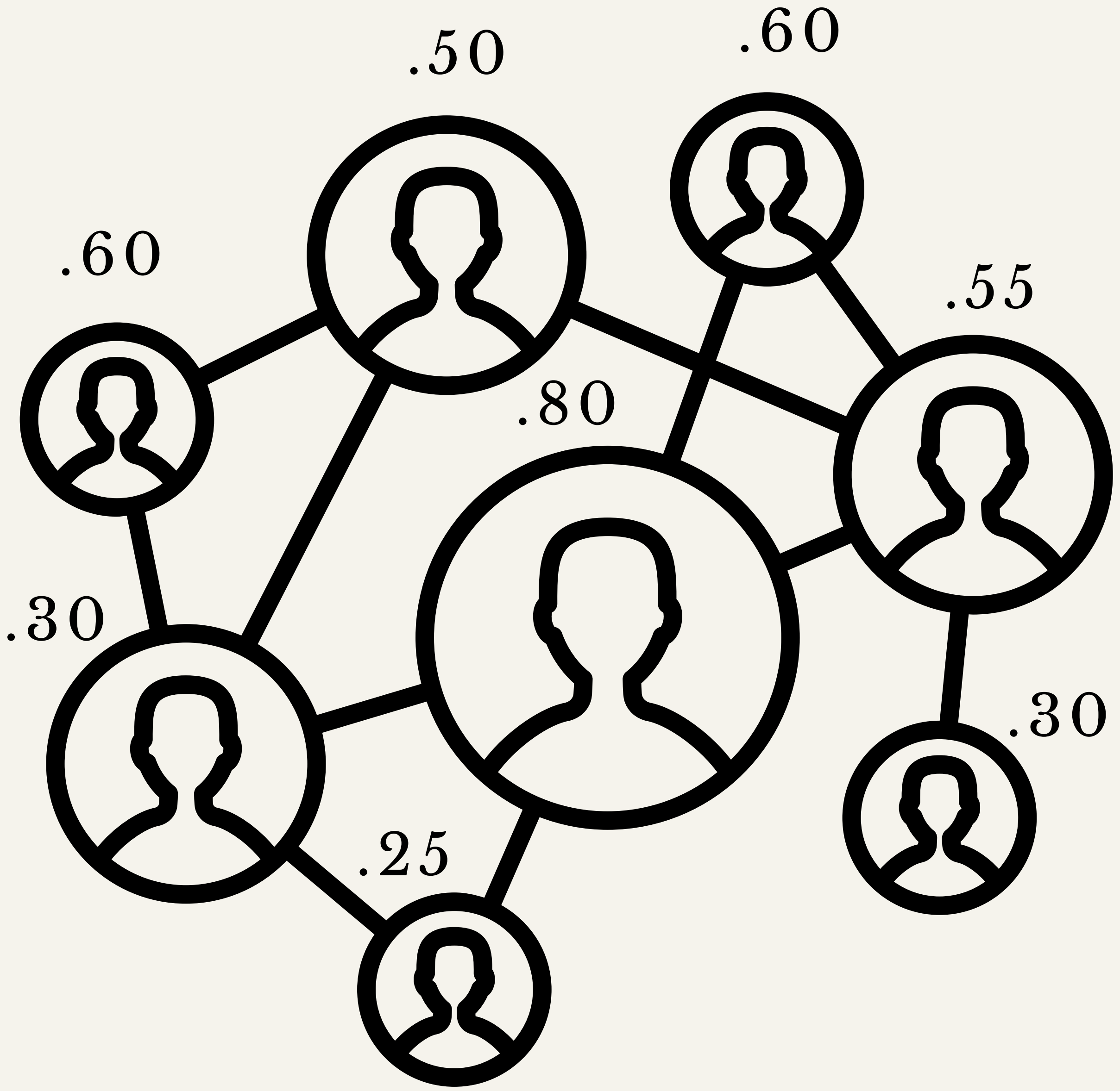
6 WHAT CAN IT DO?

7 PYTHON CODE

# Expected Outputs

**Node-Level Predictions** (e.g., classification, regression)

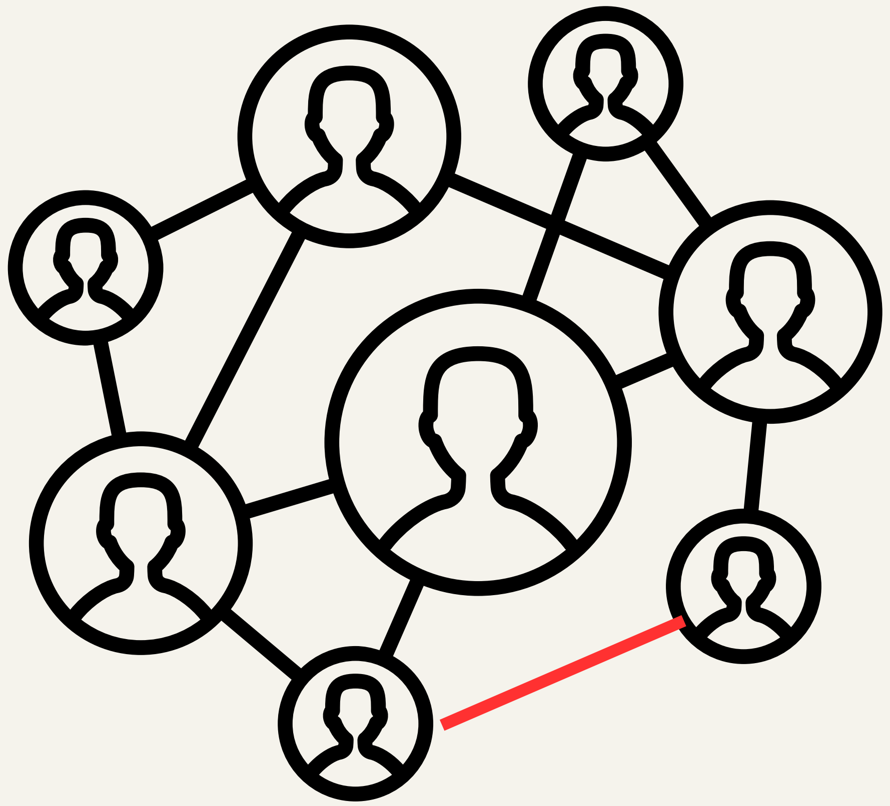
- Example: Predicting if a user is friendly.



# Expected Outputs

**Edge-Level Predictions** (e.g., link prediction, relationship strength estimation)

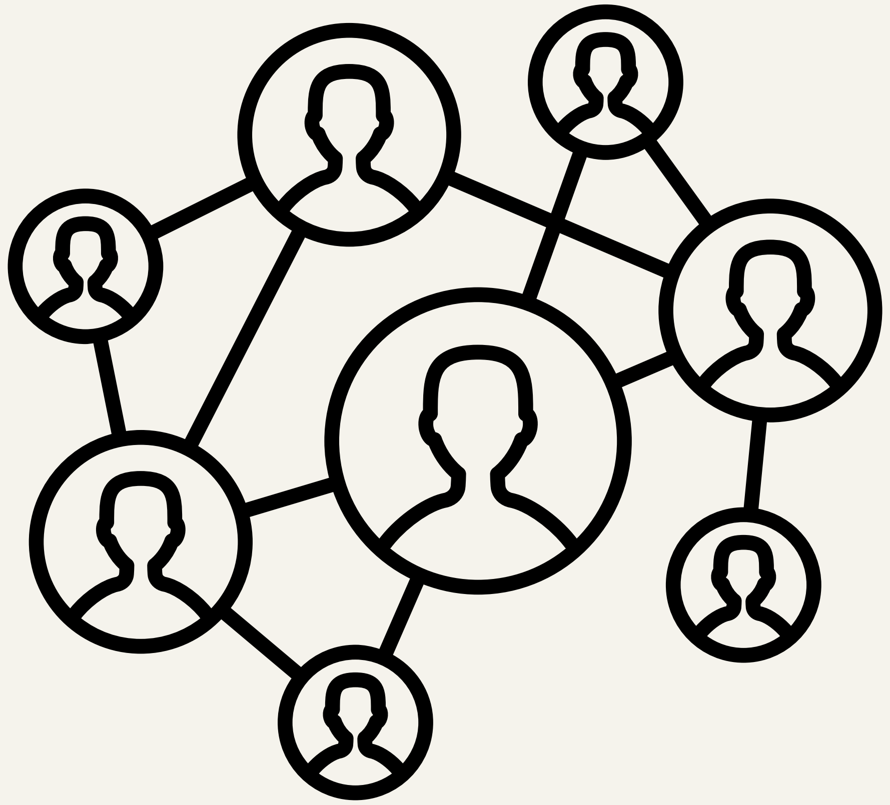
- Example: Predicting future friendships or interactions.



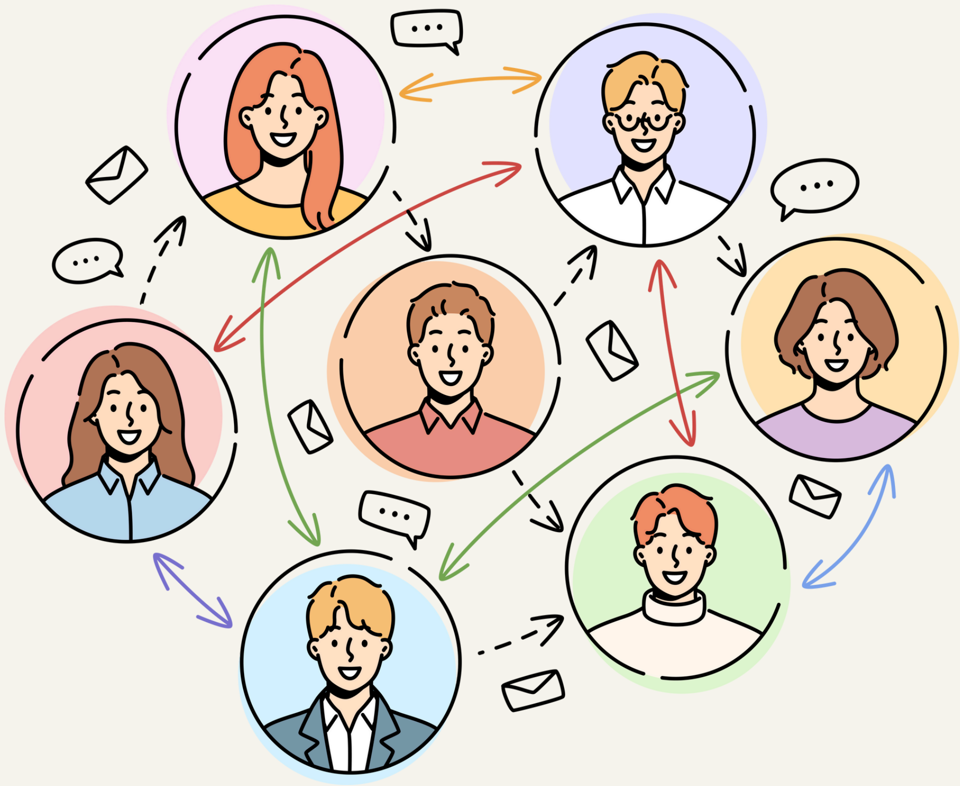
# Expected Outputs

**Graph-Level Predictions** (e.g., entire graph classification)

- Example: Predicting if a social network is conservative or liberal.



Conservative or liberal?



Conservative or liberal?

# Some Applications of GNN

- Social Network Analysis
- Recommendation Systems
- **Fraud Detection & Cybersecurity**

# Contents

1. GRAPHS

2 WHY DO WE NEED GNNS

3 CORE CONCEPT

4 MESSAGE PASSING

5 PREVAILING ARCHITECTURES

6 WHAT CAN IT DO?

7 PYTHON CODE

# Python Notebook Exercise

File > Save a Copy in Drive

bit.ly/3Dttvaa

DE LA SALLE UNIVERSITY

Dr. Andrew L. Tan
Data Science Institute

# The End

THANK YOU FOR LISTENING