Code implementation

import networkx as nx

import matplotlib.pyplot as plt

import random

from collections import defaultdict

# Step 1: Simulate multi-platform data

platforms = ['Twitter', 'Reddit', 'Facebook']

users = [f'User{i}' for i in range(1, 11)]

emotions = ['Happy', 'Sad', 'Angry', 'Fear', 'Surprise']

# Simulate posts with emotions and platform

posts = []

for i in range(1, 21):

post = {

'post\_id': f'Post{i}',

'user': random.choice(users),

'platform': random.choice(platforms),

'emotion': random.choice(emotions),

'shared\_to': random.sample(users, random.randint(1, 4))

}

posts.append(post)

# Step 2: Build emotion contagion graph

G = nx.DiGraph()

for post in posts:

for target\_user in post['shared\_to']:

G.add\_edge(post['user'], target\_user, emotion=post['emotion'], platform=post['platform'])

# Step 3: Visualize the emotion contagion

plt.figure(figsize=(12,8))

pos = nx.spring\_layout(G, seed=42)

edge\_colors = [

'green' if G[u][v]['emotion'] == 'Happy' else

'red' if G[u][v]['emotion'] == 'Angry' else

'blue' if G[u][v]['emotion'] == 'Sad' else

'orange' if G[u][v]['emotion'] == 'Fear' else

'purple'

for u,v in G.edges()

]

nx.draw(G, pos, with\_labels=True, node\_color='lightblue', node\_size=2000, edge\_color=edge\_colors, width=2, arrows=True)

nx.draw\_networkx\_edge\_labels(G, pos, edge\_labels={(u,v): G[u][v]['emotion'] for u,v in G.edges()})

plt.title("Emotion Contagion Mapping Across Multi-Platform Cascades")

plt.show()

# Step 4: Analyze contagion

emotion\_count = defaultdict(int)

for u,v in G.edges():

emotion\_count[G[u][v]['emotion']] += 1

print("Emotion Spread Counts:")

for emotion, count in emotion\_count.items():

print(f"{emotion}: {count}")