# Academic Collaboration Prediction using Research Networks

# Author: Divee Gayathri Naga Lakshmi

import pandas as pd

import networkx as nx

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import accuracy\_score, classification\_report

# Step 1: Load or simulate collaboration data

# Each row represents a pair of authors and whether they collaborated (1 = yes, 0 = no)

data = {

'author\_1': ['A', 'A', 'B', 'C', 'D', 'E', 'F', 'G'],

'author\_2': ['B', 'C', 'C', 'D', 'E', 'F', 'G', 'H'],

'collaboration': [1, 1, 0, 1, 0, 1, 0, 0]

}

df = pd.DataFrame(data)

print("Initial Data:")

print(df)

# Step 2: Construct the research collaboration network

G = nx.Graph()

for i, row in df.iterrows():

if row['collaboration'] == 1:

G.add\_edge(row['author\_1'], row['author\_2'])

print("\nNodes in the graph:", G.nodes())

print("Edges in the graph:", G.edges())

# Step 3: Feature extraction

# Compute graph-based metrics for prediction

def extract\_features(a1, a2):

common\_neighbors = len(list(nx.common\_neighbors(G, a1, a2))) if a1 in G and a2 in G else 0

jaccard\_coeff = list(nx.jaccard\_coefficient(G, [(a1, a2)]))[0][2] if a1 in G and a2 in G else 0

adamic\_adar = list(nx.adamic\_adar\_index(G, [(a1, a2)]))[0][2] if a1 in G and a2 in G else 0

pref\_attach = list(nx.preferential\_attachment(G, [(a1, a2)]))[0][2] if a1 in G and a2 in G else 0

return [common\_neighbors, jaccard\_coeff, adamic\_adar, pref\_attach]

features = []

for i, row in df.iterrows():

feat = extract\_features(row['author\_1'], row['author\_2'])

features.append(feat)

feature\_df = pd.DataFrame(features, columns=['CN', 'JC', 'AA', 'PA'])

feature\_df['collaboration'] = df['collaboration']

print("\nExtracted Features:")

print(feature\_df)

# Step 4: Train/Test Split

X = feature\_df[['CN', 'JC', 'AA', 'PA']]

y = feature\_df['collaboration']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

# Step 5: Train a Logistic Regression Model

model = LogisticRegression()

model.fit(X\_train, y\_train)

# Step 6: Predictions and Results

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

print("\n--- Model Evaluation ---")

print(f"Accuracy: {accuracy:.2f}")

print("Classification Report:")

print(classification\_report(y\_test, y\_pred))

# Step 7: Predict future collaborations

future\_pairs = [('B', 'E'), ('C', 'F'), ('D', 'G')]

future\_features = [extract\_features(a1, a2) for a1, a2 in future\_pairs]

future\_pred = model.predict(future\_features)

print("\nPredicted Future Collaborations:")

for pair, pred in zip(future\_pairs, future\_pred):

print(f"{pair}: {'Likely to Collaborate' if pred == 1 else 'Unlikely to Collaborate'}")