

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**



**ĐỒ ÁN MÔN HỌC: MẠNG XÃ HỘI**

**XÂY DỰNG MÔ HÌNH DỰ ĐOÁN LIÊN KẾT NGƯỜI DÙNG TRÊN MẠNG  
XÃ HỘI MEETUP SỬ DỤNG CÔNG NGHỆ DỮ LIỆU LỚN GOOGLE CLOUD**

**GVHD: ThS. Nguyễn Thị Anh Thư**

**Sinh viên thực hiện: Nhóm 9**

Nguyễn Hoàng Minh	20521609
Tạ Nhật Minh	20521614
Nguyễn Thiện Thuật	20521998
Ngô Văn Khải	19520614
Lý Số Ly	19521136
Phạm Mạnh Lợi	19521772

**TP. Hồ Chí Minh, ngày 3 tháng 12 năm 2023**

## LỜI CẢM ƠN

Đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến quý Thầy cô giảng viên Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP. HCM nói chung và quý thầy cô khoa Hệ thống Thông tin nói riêng, đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Đặc biệt, nhóm chúng em xin gửi lời cảm ơn và lòng biết ơn sâu sắc nhất tới giảng viên Nguyễn Thị Anh Thư, người đã hướng dẫn cho em trong suốt thời gian làm đề tài. Cô đã trực tiếp hướng dẫn tận tình, sửa chữa và đóng góp nhiều ý kiến quý báu giúp nhóm chúng em hoàn thành tốt báo cáo môn học của mình.

Trong thời gian một học kỳ thực hiện đề tài, nhóm chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới từ thầy cô, bạn bè cũng như nhiều nguồn tài liệu tham khảo, để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, vì kiến thức chuyên môn còn hạn chế và bản thân còn thiếu nhiều kinh nghiệm thực tiễn nên nội dung của báo cáo không tránh khỏi những thiếu sót, em rất mong nhận được sự góp ý, chỉ bảo thêm của quý thầy cô nhằm hoàn thiện những kiến thức của mình để nhóm chúng em có thể dùng làm hành trang thực hiện tiếp các đề tài khác trong tương lai cũng như là trong việc học tập và làm việc sau này.

Thành phố Hồ Chí Minh, 3 tháng 12 năm 2023

Nhóm sinh viên thực hiện.

# MỤC LỤC

<b>LỜI CẢM ƠN</b>	<b>2</b>
<b>MỤC LỤC</b>	<b>3</b>
<b>CHƯƠNG 1. TÌM HIỂU CÔNG NGHỆ DỮ LIỆU LỚN GOOGLE CLOUD 6</b>	
<b>1.1. Giới thiệu</b>	<b>6</b>
<b>1.2. Đánh giá ưu nhược điểm</b>	<b>7</b>
1.2.1. Ưu điểm của Google Cloud Platform	7
1.2.2. Nhược điểm của Google Cloud Platform	7
<b>1.3. Ứng dụng</b>	<b>9</b>
<b>1.4. Cách thức triển khai các thành phần theo kiến trúc dữ liệu lớn 10</b>	
1.4.1. Ingest	10
1.4.2. Process	12
1.4.3. Data sources (Data warehouse, data lake, lakehouse)	13
1.4.4. Enrich	15
<b>CHƯƠNG 2. GOOGLE CLOUD XỬ LÝ DỮ LIỆU MẠNG XÃ HỘI 16</b>	
<b>2.1. Kiến trúc tổng quan</b>	<b>16</b>
<b>2.2. Các thành phần chính</b>	<b>17</b>
2.2.1. Process & Ingest	17
2.2.1.1. Cloud DataFlow	17
2.2.2. Data Lake	19
2.2.2.1. Cloud Storage	19
2.2.3. Data Warehouse	22
2.2.3.1. BigQuery	22
2.2.3.2. Neo4J aura	26
2.2.3.3. MongoDB Atlas	28
2.2.4. Enrich	32
2.2.4.1. AutoML	32

2.2.4.2. Vertex AI	34
2.2.4.3. NetworkX	35
2.2.4.4. Spark GraphX	36
2.2.4.5. PyTorch Geometric	38
<b>CHƯƠNG 3. PHÂN TÍCH VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN LIÊN KẾT NGƯỜI DÙNG TRÊN MEETUP SỬ DỤNG GOOGLE CLOUD</b>	<b>40</b>
<b>3.1. Giới thiệu bộ dữ liệu</b>	<b>40</b>
<b>3.2. Thống kê dữ liệu</b>	<b>41</b>
<b>3.3. Lưu trữ dữ liệu</b>	<b>44</b>
3.3.1. Tổng quan quá trình lưu trữ dữ liệu	44
<b>3.4. Tiền xử lý và phân tích dữ liệu</b>	<b>50</b>
3.4.1. Trích xuất và tiền xử lý dữ liệu	50
3.4.2. Các độ đo trung tâm	53
3.4.2.1. Độ đo degree centrality	53
3.4.2.2. Độ đo closeness centrality	54
3.4.2.3. Độ đo betweenness centrality	56
3.4.2.4. Độ đo eigenvector centrality	58
3.4.2.5. Độ đo Page Rank	60
3.4.3. Phát hiện cộng đồng	62
3.4.3.1. Thuật toán phát hiện cộng đồng Louvain	62
<b>3.5. Xây dựng mô hình dự đoán liên kết</b>	<b>64</b>
3.5.1. Dự đoán dựa trên độ tương đồng cục bộ	64
3.5.1.1. Thuật toán Jaccard	64
3.5.1.2. Thuật toán Adamic-Adar	65
3.5.1.3. Thuật toán Preferential Attachment	67
3.5.2. Dự đoán dựa trên độ tương đồng toàn cục	69
3.5.2.1. Thuật toán Hitting time	69
3.5.2.1. Thuật toán Katz Global	71

3.5.3. Dự đoán dựa trên máy học	73
3.5.3.1. Xử lý dữ liệu huấn luyện mô hình	73
3.5.3.2. Xây dựng mô hình dự đoán liên kết	75
3.5.3.3. Kết quả thử nghiệm	79
<b>CHƯƠNG 4. TỔNG KẾT</b>	<b>81</b>
<b>TÀI LIỆU THAM KHẢO</b>	<b>82</b>

# CHƯƠNG 1. TÌM HIỂU CÔNG NGHỆ DỮ LIỆU LỚN GOOGLE CLOUD

## 1.1. Giới thiệu

**Google Cloud Platform (GCP)**, do Google cung cấp, là một bộ dịch vụ điện toán đám mây chạy trên cùng cơ sở hạ tầng mà Google sử dụng nội bộ cho các sản phẩm dành cho người dùng cuối của mình, chẳng hạn như Google Tìm kiếm, Gmail, Google Drive và YouTube. Bên cạnh một bộ công cụ quản lý, nó còn cung cấp một loạt dịch vụ đám mây mô-đun bao gồm điện toán, lưu trữ dữ liệu, phân tích dữ liệu và học máy. Đăng ký yêu cầu chi tiết thẻ tín dụng hoặc tài khoản ngân hàng.

Google Cloud Platform cung cấp cơ sở hạ tầng dưới dạng dịch vụ, nền tảng dưới dạng dịch vụ và môi trường điện toán không có máy chủ.

Vào tháng 4 năm 2008, Google công bố App Engine, một nền tảng để phát triển và lưu trữ các ứng dụng web trong các trung tâm dữ liệu do Google quản lý, đây là dịch vụ điện toán đám mây đầu tiên của công ty. Dịch vụ này bắt đầu được cung cấp rộng rãi vào tháng 11 năm 2011. Kể từ khi công bố App Engine, Google đã bổ sung nhiều dịch vụ đám mây vào nền tảng này.

Google Cloud Platform là một phần của Google Cloud, bao gồm cơ sở hạ tầng đám mây công cộng Google Cloud Platform, cũng như Google Workspace (G Suite), phiên bản Android và Chrome OS dành cho doanh nghiệp cũng như giao diện lập trình ứng dụng (API) dành cho máy học và lập bản đồ doanh nghiệp dịch vụ.

Với sự ra đời của AI sắp đến gần, Google đã chứng tỏ mình là công ty dẫn đầu thị trường trong việc sử dụng và triển khai các kỹ thuật thu thập Dữ liệu lớn, cũng như các phân tích cần thiết cho các dịch vụ dựa trên Dữ liệu lớn của riêng họ, chẳng hạn như Phân phối quảng cáo, và Phân tích.

## 1.2. Đánh giá ưu nhược điểm

### 1.2.1. Ưu điểm của Google Cloud Platform

- Tính linh hoạt và khả năng mở rộng:

Một trong những ưu điểm lớn nhất của GCP là tính linh hoạt và khả năng mở rộng. Cho dù bạn là một công ty khởi nghiệp nhỏ hay một doanh nghiệp đã thành lập, Google Cloud đều cung cấp nhiều dịch vụ và tài nguyên phù hợp với nhu cầu của bạn. Với khả năng tăng hoặc giảm quy mô theo yêu cầu, bạn có thể quản lý khối lượng công việc của mình một cách hiệu quả và tránh các chi phí không cần thiết.

- Độ tin cậy và hiệu suất:

Google nổi tiếng với cơ sở hạ tầng mạnh mẽ và mạng lưới trung tâm dữ liệu toàn cầu. Điều này mang lại độ tin cậy và hiệu suất vượt trội trên GCP. Bạn có thể mong đợi thời gian ngừng hoạt động tối thiểu, thời gian phản hồi nhanh và tính sẵn sàng cao, đảm bảo các ứng dụng và dịch vụ của bạn vẫn trực tuyến và người dùng có thể truy cập được.

- Công nghệ:

Google luôn đi đầu trong đổi mới công nghệ và GCP cũng không ngoại lệ. Với các công cụ và dịch vụ phức tạp, chẳng hạn như BigQuery để phân tích dữ liệu và học máy, GCP hỗ trợ các doanh nghiệp tận dụng những tiến bộ mới nhất. Điều này cho phép bạn đi trước đối thủ và mở ra những cơ hội mới.

### 1.2.2. Nhược điểm của Google Cloud Platform

- Chi phí:

Mặc dù GCP cung cấp rất nhiều tính năng và dịch vụ nhưng việc điều hướng qua chúng có thể hơi khó khăn, đặc biệt đối với người mới bắt đầu. Có thể có một lộ trình học tập liên quan

đến việc hiểu và sử dụng hiệu quả tất cả các tài nguyên mà GCP cung cấp. Tuy nhiên, với tài liệu sẵn có và cộng đồng trực tuyến, rào cản này có thể được khắc phục bằng thời gian và công sức.

- Cơ cấu giá:

Mặc dù GCP cung cấp cấu trúc định giá linh hoạt nhưng việc hiểu và ước tính chi phí một cách chính xác vẫn có thể phức tạp. Tùy thuộc vào mô hình và yêu cầu sử dụng của bạn, bạn có thể thấy khó khăn khi dự đoán chi phí hàng tháng của mình trên GCP. Tuy nhiên, việc lập kế hoạch và giám sát cẩn thận có thể giúp bạn tối ưu hóa chi tiêu của mình.

- Tùy chọn hỗ trợ hạn chế:

So với một số nhà cung cấp dịch vụ đám mây khác, GCP có các tùy chọn hỗ trợ tương đối hạn chế. Mặc dù nó cung cấp nhiều kênh hỗ trợ khác nhau, bao gồm tài liệu, diễn đàn và hỗ trợ cộng đồng, nhưng khả năng hỗ trợ trực tiếp và được cá nhân hóa có thể bị hạn chế. Đây có thể là mối lo ngại đối với các doanh nghiệp cần hỗ trợ ngay lập tức hoặc gặp các vấn đề kỹ thuật phức tạp.

Google Cloud Platform sở hữu một số lợi thế khiến nó trở thành lựa chọn hấp dẫn cho các doanh nghiệp đang tìm kiếm giải pháp điện toán đám mây đáng tin cậy và có thể mở rộng. Từ tính linh hoạt và công nghệ tiên tiến đến độ tin cậy, GCP có rất nhiều điều thú vị. Tuy nhiên, điều quan trọng là phải xem xét những thách thức tiềm ẩn, chẳng hạn như lộ trình học tập, cơ cấu giá cả và các tùy chọn hỗ trợ hạn chế. Bằng cách cân nhắc ưu và nhược điểm, bạn có thể đưa ra quyết định sáng suốt về việc liệu Google Cloud Platform có phù hợp với tổ chức của mình hay không.

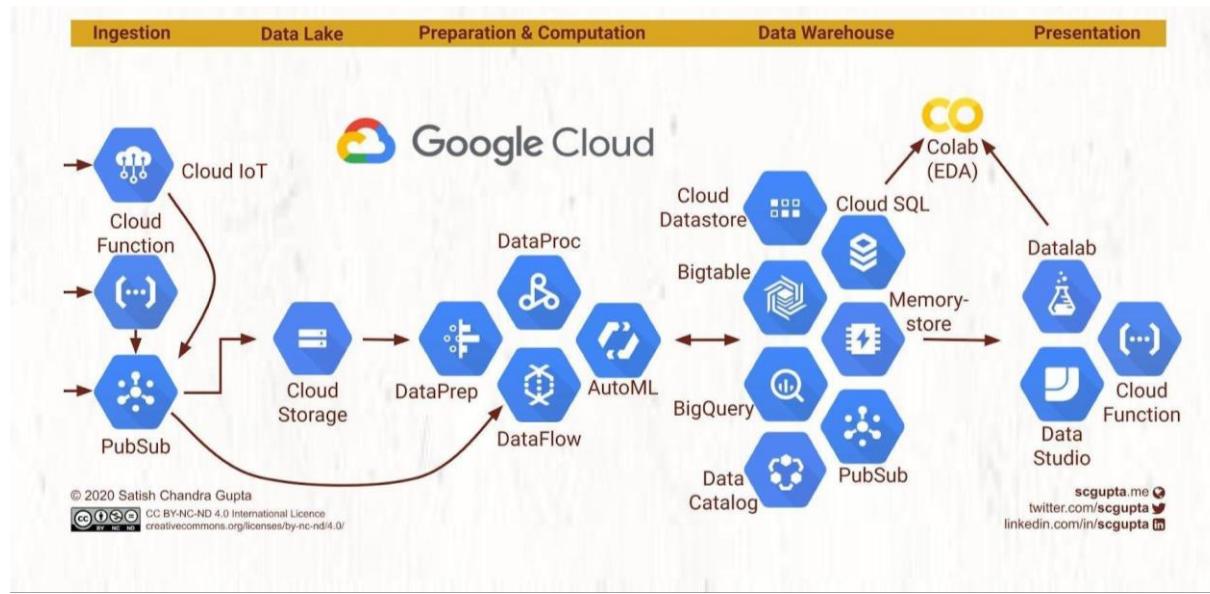
### 1.3. Ứng dụng

Google Cloud có nhiều ứng dụng và dịch vụ đa dạng để hỗ trợ các công việc liên quan đến lưu trữ, tính toán, phân tích dữ liệu và phát triển ứng dụng:

- **Lưu trữ dữ liệu:** Google Cloud Storage cung cấp dịch vụ lưu trữ đám mây với khả năng mở rộng linh hoạt và độ tin cậy cao. Nó cho phép người dùng lưu trữ, truy xuất và quản lý dữ liệu trong môi trường đám mây.
- **Tính toán đám mây:** Google Compute Engine cung cấp khả năng tính toán đám mây mạnh mẽ. Người dùng có thể tạo và quản lý các máy ảo để chạy ứng dụng, xử lý dữ liệu và triển khai hệ thống.
- **Machine Learning và AI:** Google Cloud Machine Learning cung cấp các công cụ và dịch vụ để xây dựng, huấn luyện và triển khai các mô hình máy học. Đồng thời, nó còn cung cấp các dịch vụ AI khác như Vision API, Speech-to-Text API và Translation API.
- **Cơ sở dữ liệu:** Google Cloud cung cấp nhiều dịch vụ cơ sở dữ liệu bao gồm Cloud SQL (MySQL và PostgreSQL), Cloud Firestore (cơ sở dữ liệu NoSQL), và Bigtable (cơ sở dữ liệu cột rộng).
- **Phân tích dữ liệu:** Google BigQuery là một dịch vụ phân tích dữ liệu mạnh mẽ. Điều này cho phép người dùng truy vấn và phân tích cấu trúc dữ liệu lớn với tốc độ nhanh chóng và khả năng mở rộng.
- **Internet of Things (IoT):** Google Cloud IoT cung cấp cho người dùng các dịch vụ và công cụ để kết nối, quản lý và thu thập dữ liệu từ các thiết bị IoT.
- **Phát triển ứng dụng:** Google Cloud Platform cung cấp cho người dùng môi trường phát triển ứng dụng linh hoạt với các dịch vụ như App Engine, Kubernetes Engine và Cloud Functions.

Ngoài ra, Google Cloud còn có các dịch vụ khác như dịch vụ mạng, bảo mật, giao tiếp và giải pháp doanh nghiệp để hỗ trợ các nhu cầu đa dạng của người dùng.

## 1.4. Cách thức triển khai các thành phần theo kiến trúc dữ liệu lớn



### 1.4.1. Ingest

Data ingestion trong công nghệ Google Cloud là quá trình thu thập và chuyển đổi dữ liệu từ các nguồn bên ngoài vào các dịch vụ và công cụ tính toán của Google Cloud để phân tích và lưu trữ. Google Cloud hiện nay cung cấp nhiều dịch vụ và công cụ khác nhau để hỗ trợ việc ingest dữ liệu. Sau đây là một số phương pháp thu thập dữ liệu tốt nhất được khuyên dùng bởi Google Cloud:

- Xác định nguồn dữ liệu để tiếp nhận

Dữ liệu thường thu thập từ một nhà cung cấp hoặc dịch vụ đám mây khác hay từ việc nhập dữ liệu:

- Với dữ liệu thu thập từ nhà cung cấp khác, có thể sử dụng các dịch vụ Cloud Data Fusion, Storage Transfer Service, BigQuery Transfer Service.
- Với việc nhập dữ liệu, tùy thuộc vào độ lớn của bộ dữ liệu và kỹ năng của người làm việc với chúng mà có thể xem xét việc sử dụng Cloud Data Fusion với một

đầu nối phù hợp như Java Database Connectivity (JDBC) cho những người quen thao tác trên giao diện hoặc sử dụng Transfer Appliance hay Storage Transfer Service cho những dữ liệu lớn.

- Xác định nguồn dữ liệu streaming (theo luồng) hoặc batch (hàng loạt)

Tùy thuộc vào nhu cầu sử dụng dữ liệu và độ cần thiết trong việc dùng theo luồng hay hàng loạt để có thể chọn những dịch vụ phù hợp được cung cấp.

Ví dụ: cần sử dụng cho dịch vụ global streaming với độ trễ thấp có thể dùng Pub/Sub. Nếu cần về phân tích và báo cáo có thể dùng stream data into BigQuery.

- Tiếp nhận dữ liệu bằng các công cụ tự động

Google Cloud cung cấp các công cụ tự động để hỗ trợ những nhu cầu tiếp nhận dữ liệu một cách tự động từ người dùng.

Ví dụ: bạn có thể sử dụng dịch vụ Pub/Sub hoặc Data Flow để thu thập và xử lý dữ liệu theo thời gian thực hoặc hàng loạt.

- Sử dụng công cụ di chuyển dữ liệu để tiếp nhận từ một data warehouse khác

Nếu bạn muốn tiếp nhận dữ liệu từ một data warehouse khác, Google Cloud cung cấp các công cụ di chuyển dữ liệu giúp bạn chuyển đổi và tiếp nhận dữ liệu từ nguồn đó vào hệ thống.

- Ước lượng nhu cầu tiếp nhận dữ liệu

Trước khi tiếp nhận dữ liệu, bạn cần ước lượng nhu cầu tiếp nhận dữ liệu của mình, bao gồm khối lượng dữ liệu, tần suất tiếp nhận và yêu cầu thời gian thực (nếu có). Điều này sẽ giúp bạn chọn các công cụ và phương pháp phù hợp.

- Sử dụng công cụ phù hợp để tiếp nhận dữ liệu theo lịch trình

Google Cloud cung cấp các công cụ giúp bạn tiếp nhận dữ liệu theo lịch trình định kỳ. Ví dụ, bạn có thể sử dụng dịch vụ Cloud Functions hoặc App Engine để tạo các ứng dụng và kịch bản để tự động tiếp nhận dữ liệu theo lịch trình.

- Xem xét nhu cầu tiếp nhận dữ liệu từ máy chủ FTP/SFTP

Nếu bạn cần tiếp nhận dữ liệu từ máy chủ FTP/SFTP, bạn cần xem xét và đánh giá nhu cầu tiếp nhận dữ liệu từ máy chủ này và sử dụng các công cụ phù hợp để tiếp nhận dữ liệu.

- Sử dụng các kết nối Apache Kafka để tiếp nhận dữ liệu

Apache Kafka connectors là các công cụ được sử dụng để tiếp nhận dữ liệu từ các nguồn dữ liệu theo luồng bằng cách kết nối với hệ thống Apache Kafka. Ingestion (hoặc tiếp nhận dữ liệu) là quá trình thu thập và chuyển đổi dữ liệu từ các nguồn khác nhau vào hệ thống để tiếp tục xử lý và lưu trữ.

#### 1.4.2. Process

Process data trong công nghệ Google Cloud là quá trình xử lý và biến đổi dữ liệu sau khi nó đã được ingest vào hệ thống. Google Cloud cung cấp nhiều dịch vụ và công cụ để xử lý dữ liệu theo các yêu cầu và mục đích khác nhau. Dưới đây là một số dịch vụ và công cụ phổ biến trong việc xử lý dữ liệu trong Google Cloud:

- **Khám phá các phần mềm mã nguồn mở bạn có thể sử dụng trong Google Cloud:** Google Cloud hỗ trợ sử dụng nhiều phần mềm mã nguồn mở phổ biến như Hadoop, Spark, TensorFlow và nhiều công cụ khác. Bằng cách tận dụng các công cụ này, bạn có thể thực hiện xử lý dữ liệu phức tạp và triển khai các mô hình học máy.
- **Xác định nhu cầu xử lý dữ liệu ETL hoặc ELT:** Trước khi xử lý dữ liệu, bạn cần xác định rõ nhu cầu xử lý dữ liệu ETL (Trích xuất, Chuyển đổi, Tải lên) hoặc ELT (Trích xuất, Tải lên, Chuyển đổi). Điều này sẽ giúp bạn lựa chọn các công cụ và dịch vụ phù hợp để xử lý quy trình ETL hoặc ELT của mình.

- **Sử dụng framework phù hợp cho trường hợp sử dụng dữ liệu:** Google Cloud cung cấp các framework như Dataflow, Apache Beam và Cloud Composer để hỗ trợ xử lý dữ liệu. Bằng cách chọn framework phù hợp với trường hợp sử dụng dữ liệu của bạn, bạn có thể thực hiện các phép biến đổi, tính toán và xử lý dữ liệu một cách hiệu quả.
- **Đảm bảo sự kiểm soát tương lai với máy chủ thực thi:** Trong quá trình xử lý dữ liệu, quan trọng để bạn giữ được sự kiểm soát và lựa chọn máy chủ thực thi phù hợp. Google Cloud cung cấp các dịch vụ như Kubernetes và App Engine để giúp bạn duy trì sự kiểm soát và quản lý quy trình xử lý dữ liệu của mình.
- **Sử dụng Data Flow để ingest dữ liệu từ nhiều nguồn:** Dataflow là một dịch vụ xử lý dữ liệu dựa trên luồng trong Google Cloud. Bạn có thể sử dụng Dataflow để thu thập dữ liệu từ nhiều nguồn khác nhau và xử lý nó theo thời gian thực hoặc hàng loạt.
- **Khám phá, xác định và bảo vệ dữ liệu nhạy cảm:** Trong quá trình xử lý dữ liệu, quan trọng để khám phá, xác định và bảo vệ dữ liệu nhạy cảm. Google Cloud cung cấp các công cụ và dịch vụ để giúp bạn khám phá và bảo vệ dữ liệu nhạy cảm trong quá trình xử lý.

#### 1.4.3. Data sources (Data warehouse, data lake, lakehouse)

Trong Google Cloud, Data sources (Nguồn dữ liệu) thường liên quan đến các hệ thống và nền tảng mà bạn có thể sử dụng để lưu trữ, quản lý, và truy cập dữ liệu. Các nguồn dữ liệu phổ biến hiện nay đặc biệt như Data Warehouse, Data Lake và Lakehouse.

**Data Warehouse (Kho dữ liệu):** Là một hệ thống lưu trữ dữ liệu có cấu trúc và được tối ưu hóa để phục vụ cho việc truy vấn và phân tích dữ liệu. Dữ liệu trong Data Warehouse thường đã được làm sạch và biến đổi để phục vụ nhu cầu phân tích.

### Ứng dụng:

- Thường được sử dụng để lưu trữ dữ liệu lịch sử và dữ liệu kinh doanh quan trọng.
- Sử dụng cho các mục đích phân tích dự báo, thống kê, và báo cáo.
- Ví dụ: Google BigQuery là dịch vụ Data Warehouse phổ biến trong Google Cloud.

**Data Lake (Hồ dữ liệu):** là một hệ thống lưu trữ dữ liệu linh hoạt và không có cấu trúc cụ thể. Nó cho phép bạn lưu trữ dữ liệu ở mọi định dạng (cấu trúc, bất cấu trúc, semi-cấu trúc) mà không cần biến đổi dữ liệu trước khi lưu trữ.

### Ứng dụng:

- Dữ liệu lớn và đa dạng, chẳng hạn như log, hình ảnh, video, văn bản và dữ liệu nguồn mở, có thể được lưu trữ trong Data Lake.
- Sử dụng cho việc chuẩn bị dữ liệu trước khi xử lý và phân tích (ETL).
- Ví dụ: Google Cloud Storage, Amazon S3 là các dịch vụ phổ biến cho Data Lake.

**Lakehouse:** là một mô hình kết hợp cả Data Warehouse và Data Lake để kết hợp lợi ích của cả hai. Nó cho phép bạn lưu trữ dữ liệu nguyên gốc trong Data Lake và cũng có cấu trúc để dễ dàng truy cập và truy vấn như Data Warehouse.

### Ứng dụng:

- Giúp cân bằng tính linh hoạt của Data Lake và khả năng phân tích và truy cập dữ liệu của Data Warehouse.
- Đảm bảo tính nhất quán và hiệu suất trong việc truy vấn và phân tích dữ liệu lớn.
- Ví dụ: Databricks Delta Lake là một trong những giải pháp Lakehouse phổ biến.

Mỗi nguồn dữ liệu này có ưu điểm và hạn chế riêng, và việc lựa chọn phụ thuộc vào mục tiêu và yêu cầu cụ thể của dự án của bạn. Đối với kiến trúc dữ liệu lớn, có thể kết hợp các nguồn này để đảm bảo tính toàn vẹn và hiệu suất trong quá trình quản lý và sử dụng dữ liệu.

#### 1.4.4. Enrich

Trong Google Cloud, Enrich thường liên quan đến việc cải thiện và mở rộng giá trị của dữ liệu bằng cách sử dụng dịch vụ và công cụ máy học như AutoML và Vertex AI.

**AutoML (Automated Machine Learning):** Là các công cụ và dịch vụ trong đó máy tính tự động học cách xây dựng và triển khai các mô hình máy học tùy chỉnh mà không cần kiến thức sâu về machine learning. AutoML cung cấp một cách tự động hóa quá trình phát triển mô hình máy học, bao gồm việc lựa chọn thuật toán, đào tạo mô hình, và tinh chỉnh siêu tham số.

#### Ứng dụng:

AutoML được sử dụng để xây dựng các mô hình máy học tùy chỉnh cho nhiều loại nhiệm vụ, như phân loại hình ảnh, phân tích văn bản, và dự đoán dữ liệu. Giúp giảm đáng kể thời gian và kiến thức cần thiết để phát triển các mô hình máy học. **Ví dụ:** Google Cloud AutoML bao gồm AutoML Vision cho phân loại hình ảnh, AutoML Natural Language cho phân tích ngôn ngữ tự nhiên, và AutoML Tables cho phân tích dữ liệu có cấu trúc.

**Vertex AI:** là một nền tảng AI và machine learning toàn diện trong Google Cloud. Nó cung cấp các công cụ để xây dựng, đào tạo và triển khai mô hình máy học một cách dễ dàng và hiệu quả. Vertex AI cung cấp cả các dịch vụ AutoML và dịch vụ máy học tùy chỉnh thông qua một giao diện đồ họa dễ sử dụng.

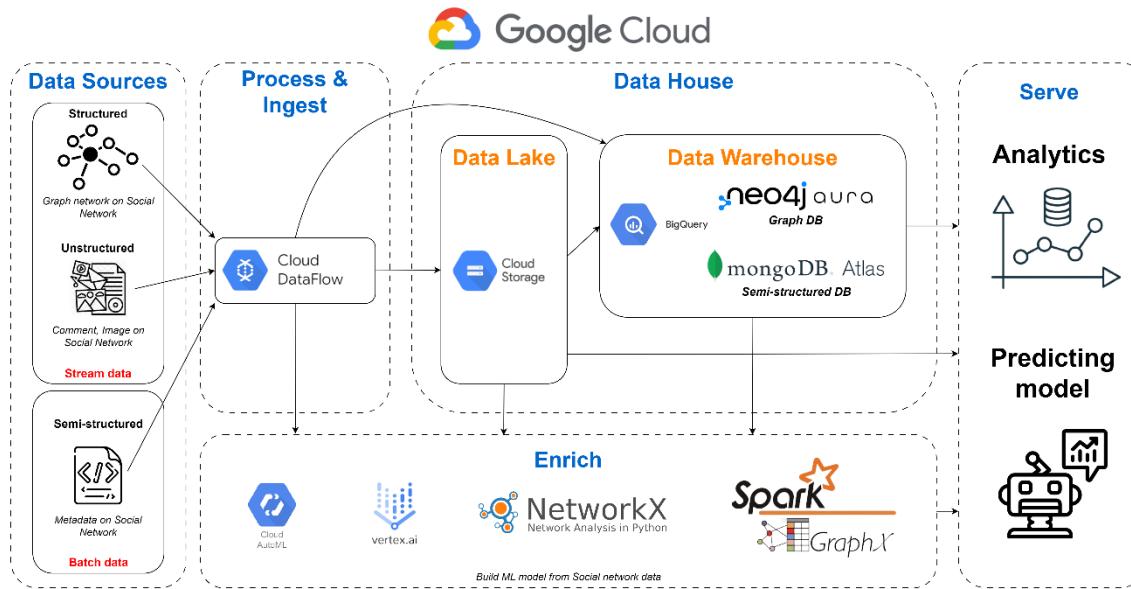
#### Ứng dụng:

- Vertex AI được sử dụng cho việc phát triển và triển khai các ứng dụng dựa trên máy học và trí tuệ nhân tạo (AI) trong các dự án kinh doanh.
- Nó hỗ trợ việc đào tạo và triển khai mô hình máy học trên dữ liệu của bạn và cung cấp các công cụ giám sát và quản lý mô hình. **Ví dụ:** Bạn có thể sử dụng Vertex AI để đào tạo mô hình máy học cho phân loại hình ảnh, phân tích ngôn ngữ tự nhiên, dự đoán dữ liệu, và nhiều loại nhiệm vụ khác.

# CHƯƠNG 2. GOOGLE CLOUD XỬ LÝ DỮ LIỆU

## MẠNG XÃ HỘI

### 2.1. Kiến trúc tổng quan



Mô hình tổng quan do chúng tôi đề xuất sử dụng công nghệ dữ liệu lớn Google Cloud kết hợp với 2 cơ sở dữ liệu Neo4J và MongoDB để xử lý dữ liệu mạng xã hội như hình trên. Có tất cả 4 thành phần chính: Process & Ingest, Data House, Enrich và Serve. Đầu tiên, tất cả dữ liệu từ mạng xã hội, được chia thành 3 nhóm chính: Structured Data (dữ liệu dạng đồ thị...), Unstructured Data (Bình luận, ảnh, video... được đăng trên Mạng xã hội) và Semi-Structured Data (Metadata,...), sẽ được DataFlow xử lý và quản lý các Data pipeline để lưu và lấy dữ liệu. Dữ liệu sẽ được lưu vào DataHouse với 2 thành phần chính là DataLake và DataWarehouse. DataLake sẽ làm nhiệm vụ lưu trữ tạm thời tất cả các loại dữ liệu của mạng xã hội để phục vụ cho việc lấy và chuyển dữ liệu vào Data Warehouse làm nhiệm vụ phân tích cụ thể. Còn DataWarehouse, chúng tôi sử dụng Neo4J để lưu trữ dữ liệu dạng đồ thị và MongoDB để lưu trữ dữ liệu dạng bán cấu trúc, BigQuery được sử dụng như một cầu nối để trích xuất dữ liệu từ Data

Lake và lưu vào 2 Database còn lại. Ở phần Enrich, chúng tôi sẽ lấy các dữ liệu cần thiết từ DataHouse và sử dụng các thư viện, các công cụ để hỗ trợ cho việc phân tích và xây dựng các mô hình dự đoán trên mạng xã hội. Cuối cùng là Serve, ở đây, chúng tôi sẽ đưa ra các kết quả từ việc phân tích dữ liệu bằng các đồ thị trực quan và insights từ chúng, thêm vào đó là các mô hình dự đoán để đưa ra các gợi ý cho người dùng.

## 2.2. Các thành phần chính

### 2.2.1. Process & Ingest

#### 2.2.1.1. *Cloud DataFlow*

Process & Ingest trong Google Cloud là quá trình và công cụ để xử lý và nhập dữ liệu vào các dịch vụ và công cụ của Google Cloud. Đây là quá trình quan trọng để thu thập, chuẩn hóa và chuyển đổi dữ liệu từ các nguồn khác nhau vào hệ thống Google Cloud để tiếp tục xử lý và phân tích.

Quá trình và công cụ Process & Ingest trong Google Cloud giúp bạn đảm bảo dữ liệu được đưa vào hệ thống một cách hiệu quả và đáng tin cậy, từ đó tạo ra cơ sở dữ liệu mạnh mẽ để phân tích, trích xuất thông tin và đưa ra quyết định.

Ví dụ: Cloud Dataflow là một dịch vụ xử lý dữ liệu phân tán mạnh mẽ, cho phép bạn xử lý, biến đổi và chuyển đổi dữ liệu theo cách tùy chỉnh. Cloud Pub/Sub là một dịch vụ hàng đợi tin nhắn đám mây, giúp bạn nhận và gửi dữ liệu theo thời gian thực. Cloud Storage là một dịch vụ lưu trữ đám mây, giúp bạn lưu trữ và quản lý dữ liệu lớn.

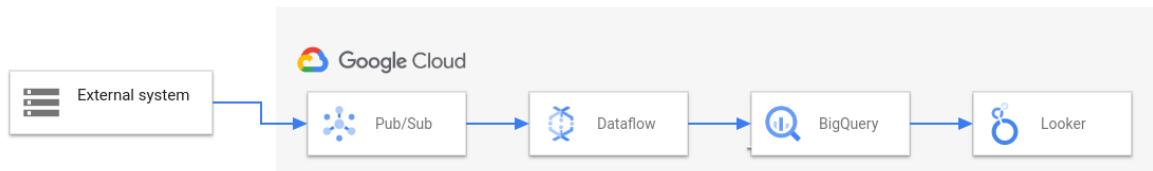
Phương thức hoạt động:

Dataflow sử dụng mô hình đường ống dữ liệu, trong đó dữ liệu di chuyển qua một loạt các giai đoạn. Các giai đoạn có thể bao gồm đọc dữ liệu từ nguồn, biến đổi và tổng hợp dữ liệu, và ghi kết quả vào đích.

Các đường ống có thể từ rất đơn giản đến phức tạp hơn. Ví dụ, một đường ống có thể thực hiện các công việc sau:

- Di chuyển dữ liệu nguyên thủy đến đích.
- Biến đổi dữ liệu để sử dụng được bởi hệ thống đích.
- Tổng hợp, xử lý và làm giàu dữ liệu để phân tích.
- Kết hợp dữ liệu với dữ liệu khác.

Với Dataflow, bạn có thể xây dựng các giải pháp ETL (Extract, Transform, Load) và BI (Business Intelligence), nơi dữ liệu được thu thập từ nguồn khác nhau, xử lý và lưu trữ vào BigQuery, và sau đó phân tích và truy vấn dữ liệu từ BigQuery bằng các công cụ như Looker.



Sơ đồ này cho thấy các giai đoạn sau:

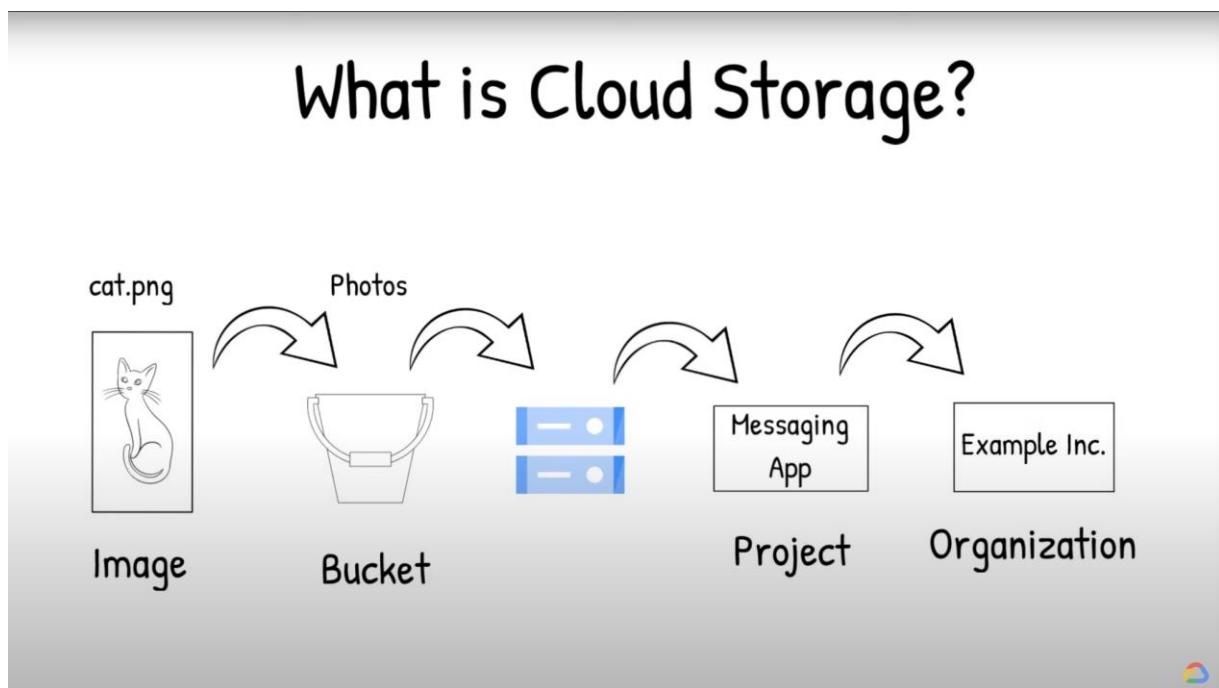
- Pub/Sub thu thập dữ liệu từ hệ thống bên ngoài.
- Dataflow đọc dữ liệu từ Pub/Sub và ghi vào BigQuery. Trong giai đoạn này, Dataflow có thể biến đổi hoặc tổng hợp dữ liệu.
- BigQuery hoạt động như một kho dữ liệu, cho phép các nhà phân tích chạy các truy vấn tạm thời trên dữ liệu.
- Looker cung cấp thông tin phân tích thời gian thực từ dữ liệu được lưu trữ trong BigQuery.

## 2.2.2. Data Lake

### 2.2.2.1. Cloud Storage

Cloud Storage là hình thức lưu trữ dữ liệu trên máy chủ ngoại tuyến. Dịch vụ này do một nhà cung cấp bên thứ ba quản lý, đảm bảo dữ liệu luôn truy cập được qua internet công khai hoặc riêng. Cloud Storage giúp tổ chức lưu trữ và truy cập dữ liệu mà không cần xây dựng trung tâm dữ liệu riêng.

Phương thức hoạt động



Cloud Storage sử dụng các máy chủ từ xa để lưu trữ dữ liệu, như các tệp tin, dữ liệu kinh doanh, video hoặc hình ảnh. Người dùng tải dữ liệu lên máy chủ thông qua kết nối internet, nơi nó được lưu trữ trên một máy ảo trên một máy chủ vật lý. Để đảm bảo tính sẵn có và cung cấp tính dự phòng, nhà cung cấp đám mây thường phân tán dữ liệu vào nhiều máy ảo trong các trung tâm dữ liệu được đặt tại khắp nơi trên thế giới. Nếu nhu cầu lưu trữ tăng, nhà cung cấp đám mây sẽ triển khai thêm máy ảo để xử lý tải lượng. Người dùng có thể truy cập dữ liệu trong Cloud

Storage thông qua kết nối internet và phần mềm như giao diện web, trình duyệt hoặc ứng dụng di động qua giao diện lập trình ứng dụng (API).

Cloud Storage có bốn mô hình khác nhau:

### **Public (Công cộng)**

Cloud Storage công cộng là mô hình trong đó tổ chức lưu trữ dữ liệu trong các trung tâm dữ liệu của nhà cung cấp dịch vụ cũng được sử dụng bởi các công ty khác. Dữ liệu trong Cloud Storage công cộng được phân tán trên nhiều vùng và thường được cung cấp theo hình thức đăng ký hoặc trả theo sử dụng. Cloud Storage công cộng được coi là "linh hoạt", có nghĩa là dữ liệu lưu trữ có thể được mở rộng hoặc thu nhỏ tùy thuộc vào nhu cầu của tổ chức. Các nhà cung cấp đám mây công cộng thường cho phép dữ liệu được truy cập từ bất kỳ thiết bị nào như điện thoại thông minh hoặc giao diện web.

### **Private (Riêng)**

Cloud Storage riêng là mô hình mà tổ chức sử dụng các máy chủ và trung tâm dữ liệu riêng để lưu trữ dữ liệu trong mạng của chính họ. Ngoài ra, tổ chức cũng có thể làm việc với các nhà cung cấp dịch vụ đám mây để cung cấp các máy chủ riêng và kết nối riêng không được chia sẻ bởi bất kỳ tổ chức nào khác. Các đám mây riêng thường được sử dụng bởi các tổ chức yêu cầu kiểm soát dữ liệu cao hơn và có yêu cầu tuân thủ và bảo mật nghiêm ngặt.

### **Hybrid (Kết hợp)**

Mô hình đám mây kết hợp là sự kết hợp giữa mô hình lưu trữ đám mây riêng và công cộng. Mô hình lưu trữ đám mây kết hợp cho phép tổ chức quyết định lưu trữ dữ liệu nào trong đám mây nào. Dữ liệu nhạy cảm và dữ liệu phải tuân thủ nghiêm ngặt có thể được lưu trữ trong đám mây riêng, trong khi dữ liệu ít nhạy cảm được lưu trữ trong đám mây công cộng. Mô hình lưu trữ đám mây đám mây kết hợp thường có một lớp điều phối để tích hợp giữa hai đám mây.

Mô hình đám mây kết hợp cung cấp tính linh hoạt và cho phép tổ chức mở rộng với đám mây công cộng nếu cần.

### **Multicloud (Đa đám mây)**

Mô hình lưu trữ đa đám mây là khi một tổ chức thiết lập nhiều mô hình đám mây từ nhiều nhà cung cấp dịch vụ đám mây (công cộng hoặc riêng). Tổ chức có thể chọn mô hình đa đám mây nếu một nhà cung cấp đám mây cung cấp các ứng dụng độc quyền, tổ chức yêu cầu dữ liệu được lưu trữ tại một quốc gia cụ thể, các nhóm khác nhau được đào tạo về các đám mây khác nhau, hoặc tổ chức cần đáp ứng các yêu cầu khác nhau mà không được nêu trong Thỏa thuận Mức dịch vụ của nhà cung cấp. Mô hình đa đám mây cung cấp tính linh hoạt và tính dự phòng cho tổ chức.

Các loại dữ liệu trong Cloud Storage:

- **Đối tượng**

Lưu trữ đối tượng là một kiến trúc lưu trữ dữ liệu cho các kho lưu trữ lớn chứa dữ liệu không có cấu trúc. Nó xác định mỗi mảnh dữ liệu là một đối tượng, lưu trữ nó trong một kho lưu trữ riêng, và đóng gói nó với siêu dữ liệu và một định danh duy nhất để dễ dàng truy cập và khôi phục.

- **Tệp**

Lưu trữ tệp tổ chức dữ liệu theo định dạng phân cấp của tệp và thư mục. Lưu trữ tệp phổ biến trong máy tính cá nhân, nơi dữ liệu được lưu trữ dưới dạng tệp và những tệp đó được tổ chức trong các thư mục. Lưu trữ tệp giúp dễ dàng tìm kiếm và khôi phục các mục dữ liệu cụ thể khi cần. Lưu trữ tệp thường được sử dụng trong các thư mục và kho dữ liệu.

- **Khối**

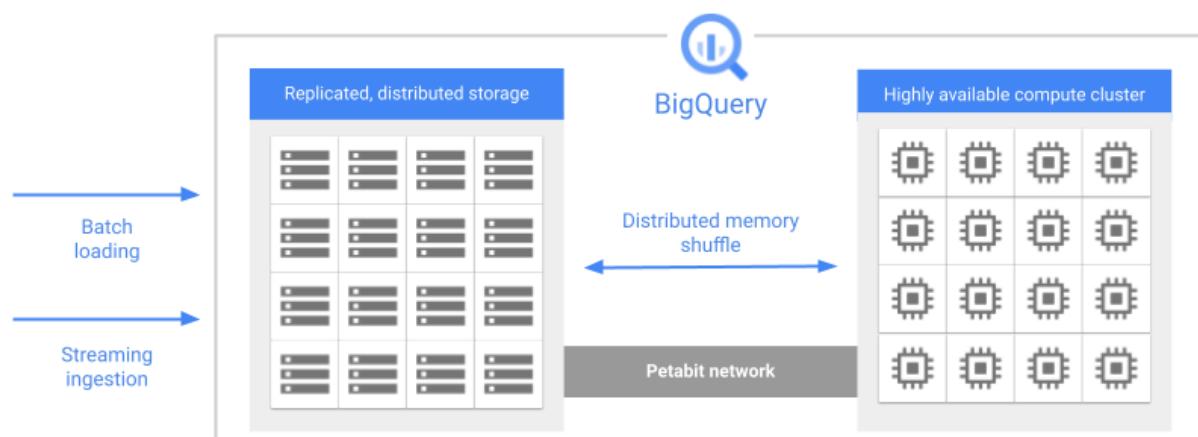
Lưu trữ khối chia dữ liệu thành các khối, mỗi khối có một định danh duy nhất, sau đó lưu trữ những khối đó như các mảnh riêng biệt trên máy chủ. Mạng đám mây lưu trữ những khối đó ở bất kỳ đâu mà nó hiệu quả nhất cho hệ thống. Lưu trữ khối thích hợp nhất cho các khối lượng dữ liệu lớn yêu cầu độ trễ thấp như các tải công việc đòi hỏi hiệu suất cao hoặc cơ sở dữ liệu.

### 2.2.3. Data Warehouse

#### 2.2.3.1. *BigQuery*

BigQuery là một hệ thống lưu trữ và phân tích dữ liệu doanh nghiệp được quản lý hoàn toàn. Với BigQuery, bạn có thể sử dụng truy vấn SQL để phân tích dữ liệu mà không cần quản lý cơ sở hạ tầng. Nó có khả năng mở rộng và cho phép truy vấn hàng terabytes trong vài giây và hàng petabytes trong vài phút. Bạn có thể lưu trữ dữ liệu trong BigQuery hoặc truy cập dữ liệu từ các nguồn bên ngoài. BigQuery cũng cung cấp các công cụ mạnh mẽ như BigQuery ML và BI Engine để phân tích và hiểu dữ liệu.

Phương thức hoạt động:



Khi bạn chạy một truy vấn, hệ thống truy vấn phân phõi công việc song song trên nhiều máy làm việc, quét các bảng liên quan trong lưu trữ, xử lý truy vấn và sau đó thu thập kết quả. BigQuery thực hiện truy vấn hoàn toàn trong bộ nhớ, sử dụng mạng petabit để đảm bảo dữ liệu di chuyển cực kỳ nhanh đến các nút làm việc.

Một số tính năng của Storage BigQuery: tính quản lý, tính bền vững, tính bảo mật và có tính hiệu quả cao

Các loại dữ liệu:

- a. Dữ liệu bảng: Dữ liệu trong BigQuery chủ yếu là dữ liệu bảng, bao gồm các loại bảng khác nhau như bảng tiêu chuẩn, bảng sao chép, bảng chụp ảnh và tầng nhìn vật liệu.

Standard tables (Bảng tiêu chuẩn): chứa dữ liệu có cấu trúc. Mỗi bảng đều có một schema, và mỗi cột trong schema đều có một kiểu dữ liệu. BigQuery lưu trữ dữ liệu theo định dạng cột.

Row	invoice_and_item_number	date	store_number	store_name	address
1	INV-17405400008	2019-02-07	3814	Costco Wholesale #788 / WDM	7205 Mills Civic Pkwy
2	S31021500029	2016-03-01	2602	Hy-Vee Food Store / Webster City	823 2ND ST
3	S31201800066	2016-03-14	2191	Keokuk Spirits	1013 MAIN
4	INV-19479400030	2019-05-20	5505	Liquor Barn II	721 Central Ave W
5	INV-19542900066	2019-05-22	5257	MAD Ave Quik Shop	405, Madison Ave
6	S20335700024	2014-07-28	3679	FRANKLIN STREET FLORAL & GIFT	103 FRANKLIN ST
7	S10466400001	2013-02-06	2626	Hy-Vee Drugstore / University / DSM	4100 UNIVERSITY AVE
8	S05337200019	2012-05-02	4593	The Cue Liquors	1404 MAIN ST
9	S13138000001	2013-07-02	2603	Hy-Vee Wine and Spirits / Bettendorf	2890 DEVILS GLEN ROAD
10	INV-09163000071	2017-12-12	2513	Hy-Vee Food Store #2 / Iowa City	812 S 1st Ave
11	INV-30841300002	2020-10-07	4312	I-80 Liquor / Council Bluffs	2411 S 24TH ST #1
12	INV-31028700021	2020-10-14	4247	Fareway Stores #879 / Belmond	512 River Ave N



Numeric Types	String-Like Types	Temporal Types	Complex Types
INT64	STRING	DATE	ARRAY
FLOAT64	BYTES	DATETIME	STRUCT
NUMERIC	GEOGRAPHY	TIME	
BIGNUMERIC		TIMESTAMP	
BOOL			



Table clones (Bảng sao chép): là các bản sao nhẹ, có thể ghi của các bảng tiêu chuẩn. BigQuery chỉ lưu trữ phần khác biệt giữa một bảng sao chép và bảng gốc của nó.

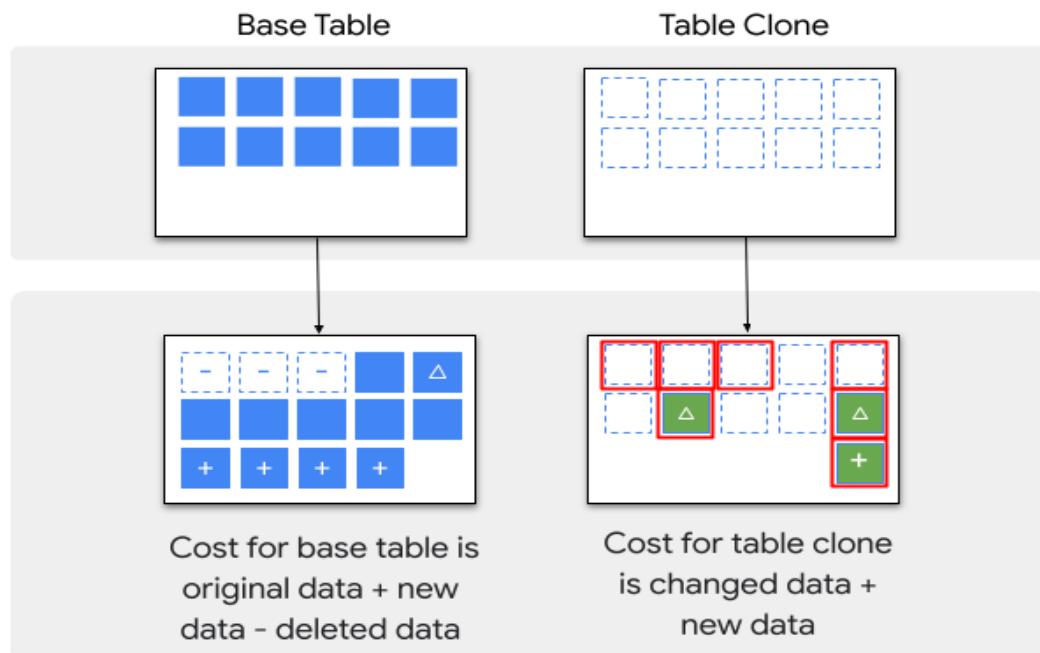
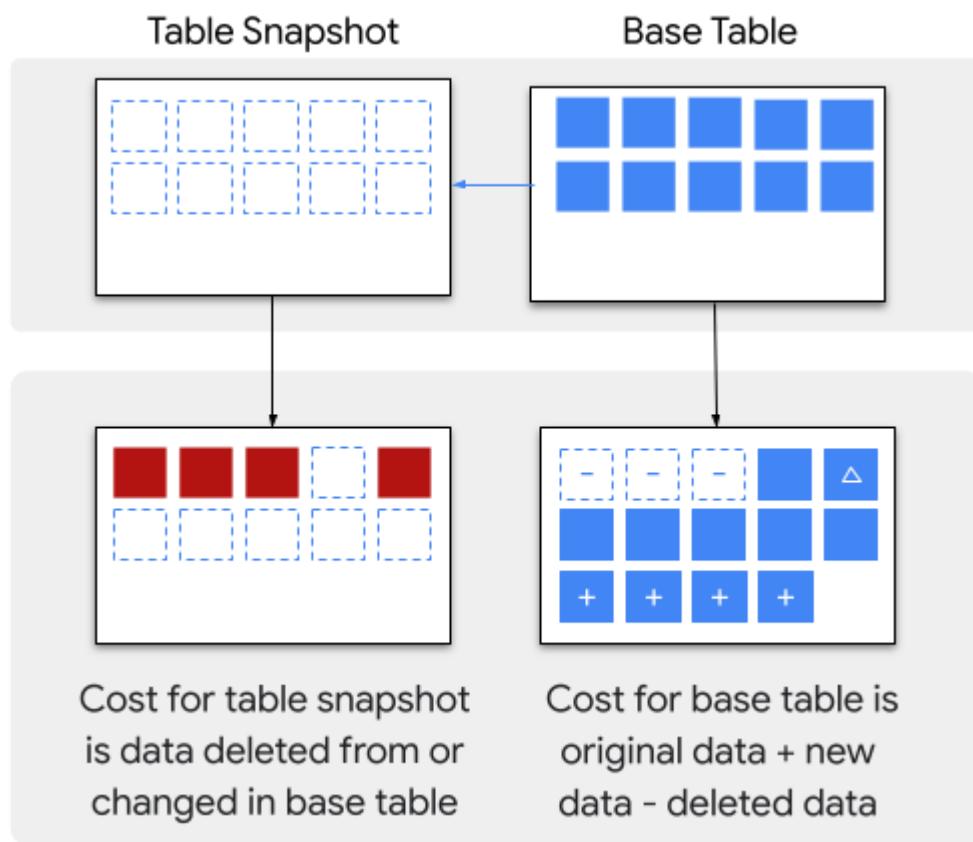


Table snapshots (Bảng chụp ảnh): là một bản sao của bảng tại một thời điểm cụ thể. Khi tạo một bảng chụp ảnh, dữ liệu trong bảng được sao lưu lại dưới dạng tài nguyên chỉ đọc. Điều này cho phép bạn truy cập và thao tác với dữ liệu trong bảng chụp ảnh mà không làm ảnh hưởng đến bản gốc của bảng. Bạn có thể sử dụng bảng chụp ảnh để xem dữ liệu tại một thời điểm cụ thể trong quá khứ hoặc để so sánh với phiên bản hiện tại của bảng.



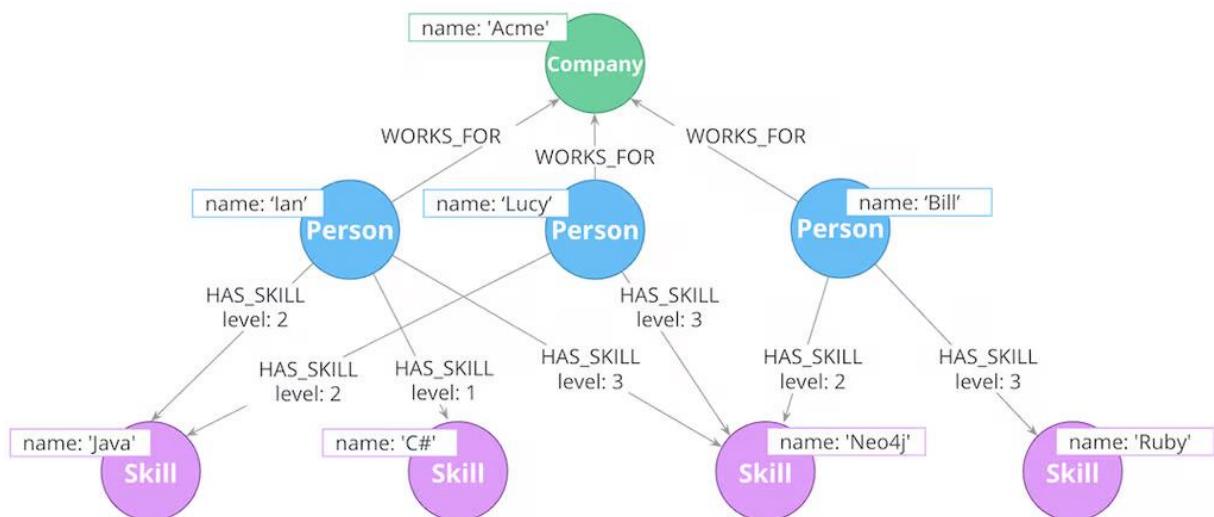
Siêu dữ liệu (Metadata): Khi bạn tạo bất kỳ đối tượng lâu dài nào trong BigQuery, chẳng hạn như bảng, tầng nhìn hoặc hàm tự định nghĩa (UDF), BigQuery lưu trữ dữ liệu mô tả về đối tượng đó. Điều này cũng đúng cho các tài nguyên không chứa dữ liệu bảng, như UDF và tầng nhìn logic.

Dữ liệu mô tả bao gồm thông tin về cấu trúc bảng, thông số phân vùng và nhóm cụm của bảng, thời gian hết hạn của bảng và các thông tin khác. Loại dữ liệu mô tả này có thể nhìn thấy và có thể được cấu hình khi tạo tài nguyên. Ngoài ra, BigQuery lưu trữ các dữ liệu mô tả mà nó sử dụng bên trong để tối ưu hóa các truy vấn.

### 2.2.3.2. Neo4J aura

Neo4j Aura là một nền tảng đồ thị được cung cấp dưới dạng dịch vụ đám mây, nhanh chóng, có khả năng mở rộng, luôn hoạt động và hoàn toàn tự động.

Aura bao gồm AuraDB, dịch vụ cơ sở dữ liệu đồ thị dành cho các nhà phát triển xây dựng ứng dụng thông minh, và AuraDS, dịch vụ khoa học dữ liệu đồ thị dành cho các nhà khoa học dữ liệu xây dựng mô hình dự đoán và quy trình phân tích.



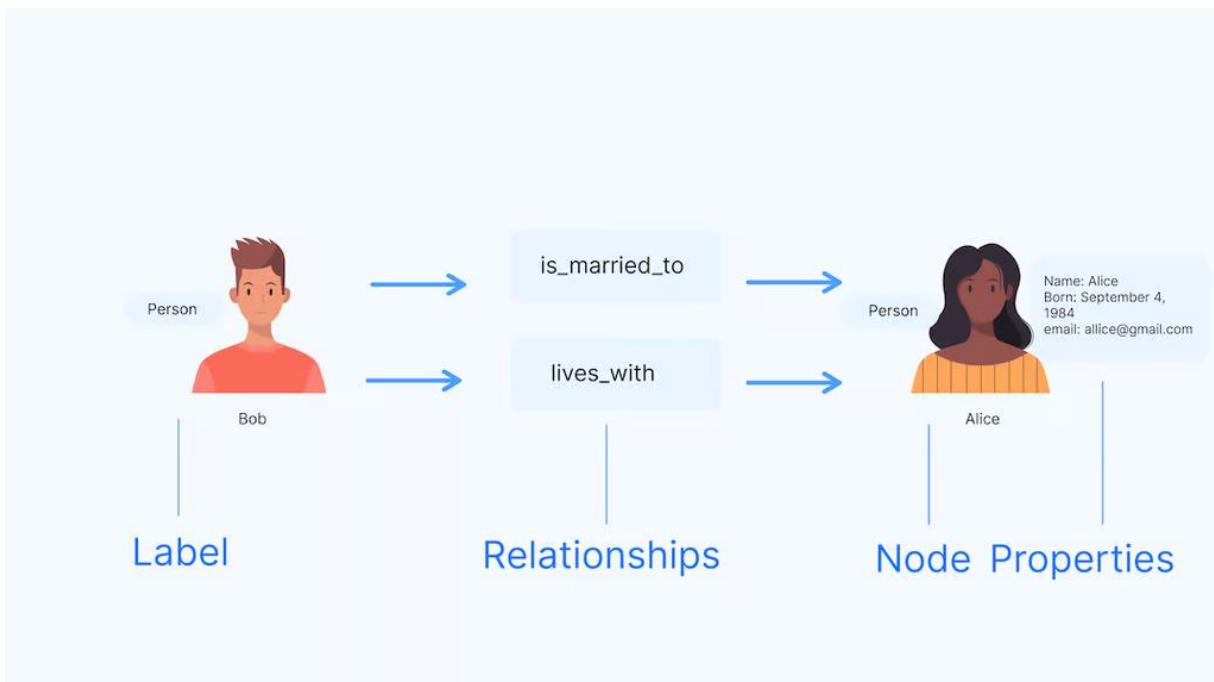
Các định nghĩa:

- Nút: Các phần tử dữ liệu chính (ví dụ: Jack hoặc các thành viên khác của vòng kết nối tình bạn) được kết nối với nhau bằng các mối quan hệ. Các nút có thể có nhãn và thuộc tính (giải thích bên dưới).

- Mối quan hệ: Mô tả các kết nối giữa các nút và kết nối chúng với nhau (ví dụ: Jack đã "kết hôn với" Jane). Các mối quan hệ có thể có một hoặc nhiều thuộc tính.
- Nhãn: Thể hiện vai trò của các nút (ví dụ: Jane là "người.") Nhãn được sử dụng để nhóm các nút. Mỗi nút có thể có nhiều nhãn. Nhãn cũng được lập chỉ mục để đẩy nhanh quá trình tìm kiếm các nút trong biểu đồ.
- Thuộc tính: Thuộc tính của các nút và các mối quan hệ liên quan đến các cặp tên hoặc giá trị.

Các loại dữ liệu:

Cơ sở dữ liệu Neo4j cho phép lưu trữ dữ liệu dưới dạng các cặp khóa-giá trị, nghĩa là các thuộc tính có thể có bất kỳ loại giá trị nào (chuỗi, số hoặc boolean.) Cấu trúc dữ liệu biểu đồ ban đầu có vẻ hơi phức tạp, nhưng nó rất đơn giản và tự nhiên



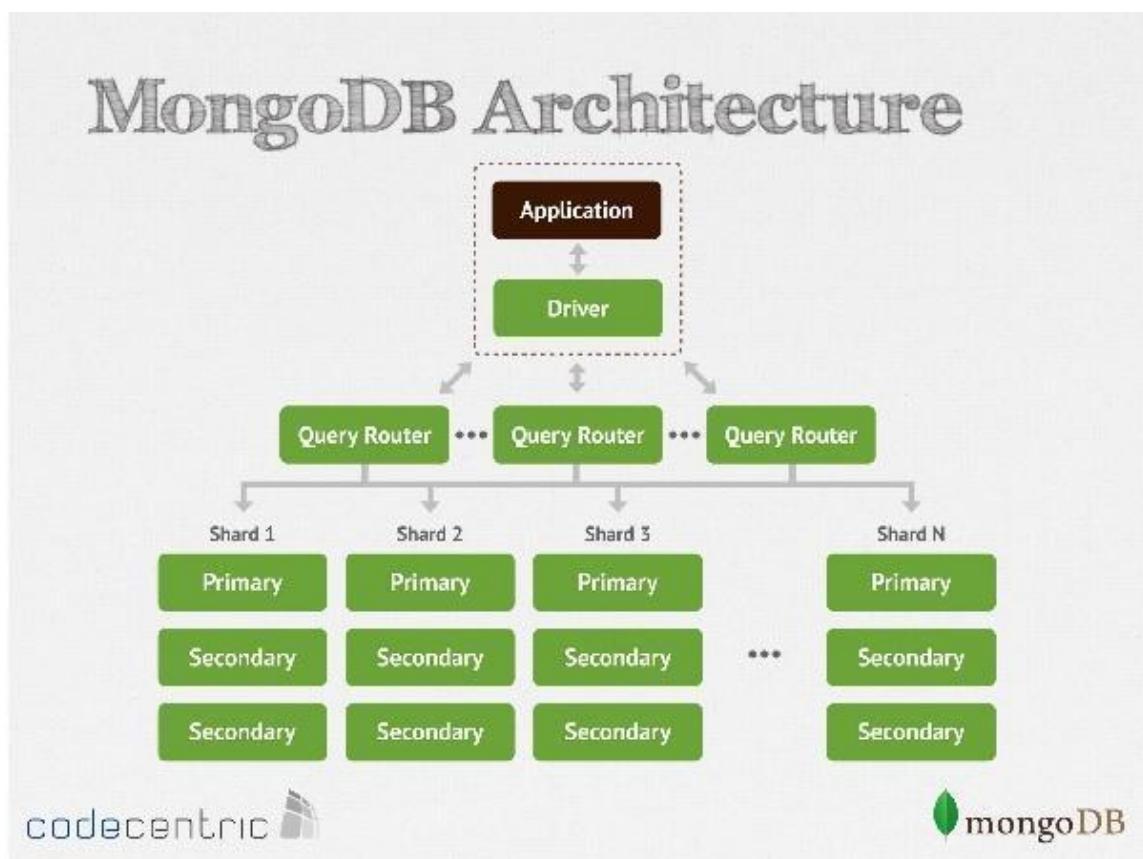
### 2.2.3.3. MongoDB Atlas

#### Kiến trúc tổng quan

MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++.

Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.

MongoDB có mô hình dữ liệu linh hoạt, khác với mô hình dữ liệu của các cơ sở dữ liệu quan hệ phải khai báo lược đồ quan hệ trước khi insert dữ liệu.



### Tại sao lại sử dụng Mongo

MongoDB: là cơ sở dữ liệu phi quan hệ vì vậy dữ liệu của nó được lưu trữ dưới dạng document. Với dạng lưu trữ này giúp việc lưu trữ dữ liệu linh hoạt hơn và phù hợp hơn với các yêu cầu phát triển dữ liệu trong thực tế.

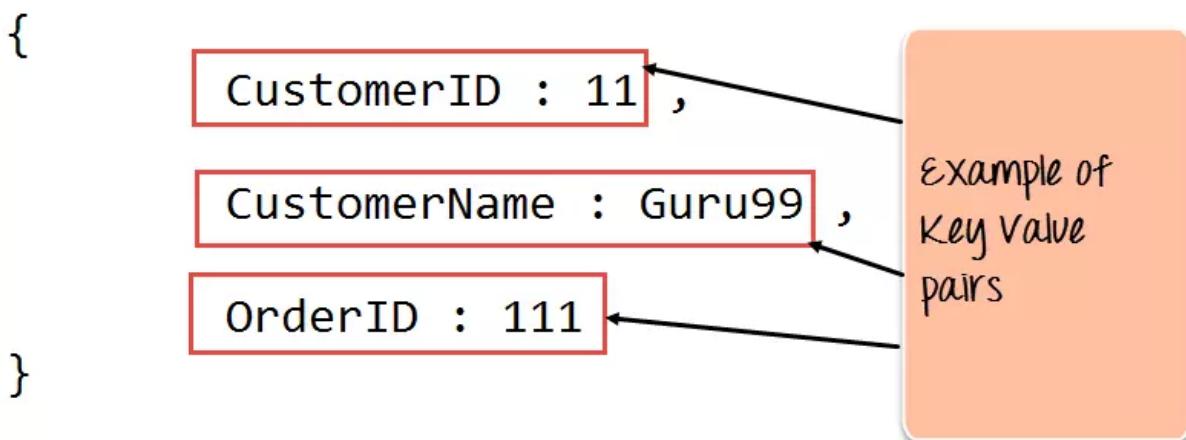
Truy vấn đặc biệt: MongoDB hỗ trợ bạn tìm kiếm dữ liệu theo trường, truy vấn theo miền, hoặc tìm kiếm các biểu thức thông thường. Các truy vấn được thực hiện để có thể trả về các trường cụ thể trong document.

Đánh chỉ mục (Index): Đánh chỉ mục để giúp việc tìm kiếm nhanh chóng hơn và trong MongoDB có thể đánh chỉ mục cho bất kỳ trường nào.

Bản sao (Replication): MongoDB luôn sẵn sàng cung cấp bộ các bản sao. Một bộ bản sao bao gồm hai hoặc nhiều thể hiện của MongoDB. Mỗi bộ bản sao có thể đóng vai trò là bản sao chính (bản sao thứ cấp) hoặc bản sao phụ (bản sao sơ cấp) bất cứ lúc nào. Bản sao thứ cấp là server chính tương tác với client để thực hiện việc đọc/ghi. Bản sao sơ cấp một bản sao dữ liệu của bản sao thứ cấp. Khi bản sao thứ cấp thất bại, bản sao thứ cấp tự động được chuyển qua bản sao sơ cấp để chuyển thành thứ cấp sau đó trở thành server thứ cấp.

Cân bằng tải: MongoDB sử dụng khái niệm sharding để chia cắt dữ liệu thành nhiều thể hiện của MongoDB. MongoDB có thể chạy trên nhiều server cân bằng tải, và hoặc sao chép dữ liệu để hệ thống luôn hoạt động ngay cả khi có lỗi phần cứng.

### Cách hoạt động Mongo



MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.

Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.

Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.

### Các loại dữ liệu trong Mongo



Nếu như mô hình dữ liệu trong hệ quản trị cơ sở dữ liệu quan hệ là các bảng quan hệ thì trong hệ quản trị cơ sở dữ liệu phi quan hệ lại được lưu trữ dưới nhiều dạng khác nhau, dưới đây là các dạng lưu trữ và giới.

Key - Value: dữ liệu lưu trữ kiểu này chúng ta sẽ dựa vào key để lấy value. Dữ liệu dạng này có tốc độ truy vấn nhanh và đặc biệt Key - Value được sử dụng để làm cache cho dữ liệu (ví dụ như Redis).

Document: Mỗi object được lưu trữ trong cơ sở dữ liệu dưới dạng document. Dữ liệu sẽ được lưu trữ dưới dạng BSON/JSON/XML dưới database. Với dạng này chúng ta có thể dễ dàng thêm, sửa, xóa trường một cách linh hoạt vì vậy nó khắc phục được cấu trúc cứng nhắc của Schema.

Column - Family: Dữ liệu được lưu trữ dạng cột khác với SQL là dạng hàng, mỗi hàng có key/id riêng. Đặc biệt mỗi hàng trong một bảng lại có số lượng cột khác nhau. Với dạng lưu trữ này sẽ thích hợp cho việc ghi một số lượng lớn dữ liệu.

Graph: Đây là kiểu cơ sở dữ liệu đồ thị, dữ liệu sẽ được lưu trữ theo từng node. Node chính là các thực thể hoặc là đối tượng. Properties là các thuộc tính liên quan đến từng Node, nó sẽ được đặt tên sao cho có quan hệ gần gũi với Node nhất. Edges là cạnh nối các Node, nó biểu thị cho quan hệ giữa các Node. Với dạng lưu trữ này thường được sử dụng trong các mạng nơron, mạng xã hội.... Ưu điểm của nó là sử dụng các thuật toán duyệt node để tăng tốc độ truy vấn.

#### 2.2.4. Enrich

##### 2.2.4.1. AutoML

Automated Machine Learning cung cấp các phương pháp và quy trình để cung cấp Machine Learning cho những người không phải là chuyên gia về Machine Learning, nhằm nâng cao hiệu quả của Machine Learning và đẩy nhanh nghiên cứu về Machine Learning.

Học máy (ML) đã đạt được những thành công đáng kể trong những năm gần đây và ngày càng có nhiều ngành học dựa vào nó. Tuy nhiên, thành công này chủ yếu dựa vào các chuyên gia về máy học của con người để thực hiện các nhiệm vụ sau:

- Tiền xử lý và làm sạch dữ liệu.
- Lựa chọn và xây dựng các tính năng phù hợp.
- Lựa chọn mô hình phù hợp
- Tối ưu hóa siêu tham số mô hình.
- Thiết kế cấu trúc liên kết của mạng lưới thần kinh (nếu sử dụng học sâu).
- Mô hình học máy sau xử lý
- Phân tích nghiêm túc các kết quả thu được.

Vì độ phức tạp của những nhiệm vụ này thường vượt quá khả năng của những người không phải là chuyên gia ML, nên sự phát triển nhanh chóng của các ứng dụng học máy đã tạo ra nhu cầu về các phương pháp học máy có sẵn có thể sử dụng dễ dàng và không cần kiến thức chuyên môn. Chúng tôi gọi khu vực nghiên cứu kết quả nhằm tới mục tiêu tự động hóa tiến bộ của máy học là AutoML.

Trong những năm gần đây, một số gói có sẵn đã được phát triển để cung cấp khả năng học máy tự động. Chúng tôi đã phát triển:

- AutoWEKA là một cách tiếp cận để lựa chọn đồng thời thuật toán học máy và các siêu tham số của nó; kết hợp với gói WEKA, nó sẽ tự động tạo ra các mô hình tốt cho nhiều tập dữ liệu khác nhau.
- Auto-sklearn là một phần mở rộng của AutoWEKA sử dụng thư viện Python scikit-learn, đây là một sự thay thế tùy ý cho các trình phân loại và hồi quy scikit-learn thông thường.
- Auto-PyTorch dựa trên khung học tập sâu PyTorch và cùng tối ưu hóa các siêu tham số và kiến trúc thần kinh.

Các gói AutoML nổi tiếng khác bao gồm:

- AutoGluon là phương pháp xếp chồng nhiều lớp của các mô hình ML đa dạng.
- H2O AutoML cung cấp khả năng lựa chọn và tổng hợp mô hình tự động cho nền tảng phân tích dữ liệu và học máy H2O.
- MLBox là thư viện AutoML có ba thành phần: tiền xử lý, tối ưu hóa và dự đoán.
- TPOT là trợ lý khoa học dữ liệu giúp tối ưu hóa quy trình học máy bằng lập trình di truyền.
- TransmogrifAI là một thư viện AutoML chạy trên Spark.

#### 2.2.4.2. Vertex AI

Vertex AI là một nền tảng trí tuệ nhân tạo hợp nhất cung cấp tất cả các dịch vụ đám mây của Google dưới một mái nhà. Với Vertex AI, bạn có thể xây dựng các mô hình ML hoặc triển khai và mở rộng quy mô chúng một cách dễ dàng bằng cách sử dụng công cụ tùy chỉnh và được đào tạo trước. Khi phát triển các giải pháp ML trên Vertex AI, bạn có thể tận dụng AutoML và các thành phần ML nâng cao khác để nâng cao đáng kể năng suất và khả năng mở rộng.

Google cũng tập trung biến Vertex AI thành một nền tảng thân thiện cho người mới và là giải pháp tiết kiệm thời gian cho các chuyên gia. Đó là lý do nó có thể huấn luyện các mô hình một cách dễ dàng và yêu cầu ít dòng mã hơn 80%.

##### Các tính năng chính của Vertex AI

Mặc dù Vertex AI có sẵn rất nhiều tính năng nhưng sau đây là một số tính năng chính của nó:

Toàn bộ quy trình làm việc ML trong một giao diện người dùng hợp nhất: Vertex AI cung cấp một giao diện người dùng và API hợp nhất cho tất cả các dịch vụ Google Cloud liên quan đến AI. Ví dụ: trong Vertex AI, bạn có thể sử dụng AutoML để đào tạo và so sánh các mô hình cũng như lưu trữ chúng trong kho lưu trữ mô hình trung tâm.

Tích hợp với tất cả các khung nguồn mở: Vertex AI tích hợp với các khung nguồn mở thường được sử dụng, chẳng hạn như PyTorch và TensorFlow, đồng thời nó cũng hỗ trợ các công cụ khác thông qua các vùng chứa tùy chỉnh.

Truy cập vào các API được đào tạo trước cho video, tầm nhìn và các API khác: Vertex AI giúp dễ dàng tích hợp video, dịch thuật và xử lý ngôn ngữ tự nhiên với các ứng dụng hiện có. AutoML trao quyền cho các kỹ sư đào tạo các mô hình được tùy chỉnh để đáp ứng nhu cầu kinh doanh của họ với chuyên môn và nỗ lực tối thiểu.

Tích hợp dữ liệu từ đầu đến cuối và AI: Vertex AI được tích hợp với Dataproc, Dataflow và BigQuery nguyên bản thông qua Vertex AI Workbench. Bạn có thể xây dựng/chạy các mô hình ML trong BigQuery hoặc bạn có thể sử dụng dữ liệu xuất từ BigQuery sang Vertex AI Workbench và thực thi các mô hình ML từ đó.

#### 2.2.4.3. NetworkX

Phân tích biểu đồ có thể được sử dụng để xác định cường độ và hướng của mối quan hệ giữa các đối tượng trong biểu đồ. Nhu cầu về các công cụ để phân tích mối quan hệ có tiềm năng gần như vô hạn do vai trò ngày càng tăng của mạng trong hệ sinh thái thông tin của chúng ta. Ảnh hưởng của mạng xã hội đến mọi thứ, từ quyết định mua hàng đến bầu cử quốc gia đã thúc đẩy sự quan tâm đến việc phân tích biểu đồ. Nó đặc biệt hữu ích trong việc khám phá các mối quan hệ không rõ ràng do tính phức tạp của mạng hoặc số lượng đường dẫn giữa các nút.

Có nhiều cách sử dụng phân tích mạng biểu đồ, chẳng hạn như phân tích các mối quan hệ trong mạng xã hội, phát hiện mối đe dọa mạng và xác định những người có nhiều khả năng mua sản phẩm nhất dựa trên sở thích chung.

Trong thế giới thực, các nút có thể là người, nhóm, địa điểm hoặc những thứ như khách hàng, sản phẩm, thành viên, thành phố, cửa hàng, sân bay, bến cảng, tài khoản ngân hàng, thiết bị, điện thoại di động, phân tử hoặc trang web.

Các nút NetworkX có thể là bất kỳ đối tượng nào có thể băm được, nghĩa là giá trị của nó không bao giờ thay đổi. Đây có thể là chuỗi văn bản, hình ảnh, đối tượng XML, toàn bộ biểu đồ và các nút tùy chỉnh. Gói cơ sở bao gồm nhiều chức năng để tạo, đọc và ghi biểu đồ ở nhiều định dạng.

NetworkX có khả năng hoạt động trên các đồ thị rất lớn với hơn 10 triệu nút và 100 triệu cạnh. Gói cốt lõi, là phần mềm miễn phí theo giấy phép BSD, bao gồm các cấu trúc dữ liệu để biểu diễn những thứ như đồ thị đơn giản, đồ thị có hướng và đồ thị có các cạnh song song và các

vòng tự lặp. NetworkX cũng có một cộng đồng lớn các nhà phát triển duy trì gói cốt lõi và đóng góp cho hệ sinh thái bên thứ ba.

Trong số các ứng dụng chính của NetworkX là:

- Nghiên cứu cấu trúc và động lực của mạng lưới xã hội, sinh học và cơ sở hạ tầng
- Môi trường lập trình tiêu chuẩn hóa cho đồ thị
- Phát triển nhanh chóng các dự án hợp tác, đa ngành
- Tích hợp với các thuật toán và mã viết bằng C, C++ và FORTRAN
- Làm việc với các tập dữ liệu lớn không chuẩn

NetworkX được coi là tương đối dễ cài đặt và sử dụng, đặc biệt đối với các nhà phát triển Python.

Phân tích biểu đồ rất hữu ích để đạt được những điều sau:

- Phát hiện tội phạm tài chính như rửa tiền
- Xác định các giao dịch và hoạt động gian lận
- Thực hiện phân tích người ảnh hưởng trong cộng đồng mạng xã hội
- Thực hiện phân tích đề xuất từ xếp hạng hoặc mua hàng của khách hàng
- Xác định điểm yếu của lưới điện, lưới nước và mạng lưới giao thông
- Tối ưu hóa các tuyến đường trong ngành hàng không, bán lẻ và sản xuất

#### 2.2.4.4. *Spark GraphX*

GraphX là thành phần mới nhất trong Spark. Đó là một đa đồ thị có hướng, có nghĩa là nó chứa cả các cạnh và đỉnh và có thể được sử dụng để biểu diễn một loạt các cấu trúc dữ liệu. Nó cũng có các thuộc tính liên quan gắn liền với mỗi đỉnh và cạnh.

GraphX hỗ trợ một số toán tử cơ bản và một biến thể được tối ưu hóa của API Pregel. Ngoài những công cụ này, nó còn bao gồm một bộ sưu tập thuật toán ngày càng tăng giúp bạn phân tích dữ liệu của mình.

Spark GraphX là hệ thống xử lý đồ thị mạnh mẽ và linh hoạt nhất hiện nay. Nó có một thư viện thuật toán ngày càng tăng có thể áp dụng cho dữ liệu của bạn, bao gồm PageRank, các thành phần được kết nối, SVD++ và số lượng tam giác.

Ngoài ra, Spark GraphX còn có thể xem và thao tác với biểu đồ và tính toán. Bạn có thể sử dụng RDD để chuyển đổi và nối các biểu đồ. Thuật toán đồ thị lặp tùy chỉnh cũng có thể được viết bằng API Pregel.

Mặc dù Spark GraphX vẫn giữ được tính linh hoạt, khả năng chịu lỗi và dễ sử dụng nhưng nó mang lại hiệu suất tương đương với các bộ xử lý đồ thị chuyên dụng nhanh nhất.

GraphX cho phép bạn áp dụng trực tiếp các chức năng lọc và ánh xạ cơ bản trên các tập hợp các đỉnh và cạnh. Tuy nhiên, nó cũng cho phép bạn xác định các hàm tùy chỉnh được gọi là Hàm do người dùng xác định (UDF) có thể được sử dụng theo cách tương tự như các hoạt động tích hợp sẵn.

- Toán tử kết cấu

Toán tử đảo ngược - Khi đảo ngược tất cả các cạnh trong biểu đồ, nó sẽ tạo ra một biểu đồ mới. Nó có thể hữu ích khi cố gắng tính toán PageRank nghịch đảo.

Toán tử đồ thị con - Toán tử đồ thị con chọn các đỉnh và cạnh quan tâm. Chúng ta có thể sử dụng toán tử này để giới hạn đồ thị ở các đỉnh và cạnh mà chúng ta quan tâm, điều này sẽ loại bỏ các liên kết bị hỏng.

Toán tử mặt nạ - Nó xây dựng một đồ thị con bằng cách trả về một đồ thị chứa các đỉnh và cạnh được tìm thấy trong đồ thị đầu vào. Chúng ta có thể sử dụng nó với toán tử đồ thị con để hạn chế biểu đồ dựa trên các tiêu chí cụ thể.

- Tham gia các nhà khai thác

Một trong những cách tốt nhất để kéo dữ liệu từ nhiều nguồn vào một biểu đồ là sử dụng toán tử nối. Nó rất hữu ích khi bạn có thêm thuộc tính người dùng mà bạn muốn hợp nhất với biểu đồ hiện có hoặc nếu bạn muốn kéo các thuộc tính đỉnh từ biểu đồ này sang biểu đồ khác. Có hai toán tử Join: joinvertices và Outerjoinvertices.

- Tin nhắn tổng hợp

Việc tổng hợp được xử lý bằng thao tác tổng hợp Messages trong GraphX. Tại đỉnh đích của chúng, nó tổng hợp các tin nhắn bằng hàm sendMsg do người dùng xác định.

#### 2.2.4.5. *PyTorch Geometric*

PyTorch Geometric là một framework deep learning tập trung vào việc xây dựng mạng lưới thần kinh dựa trên đồ thị. Nó được thiết kế để sử dụng cùng với PyTorch, một thư viện deep learning phổ biến. Không giống như các khung học sâu khác, PyTorch Geometric cho phép tạo mạng thần kinh dựa trên đồ thị, rất hữu ích cho các tác vụ như dự đoán liên kết, phân loại nút và phân loại đồ thị.

PyTorch Geometric được thiết kế để mang lại hiệu quả cao và thân thiện với người dùng. Nó cung cấp một số tính năng chính, chẳng hạn như trình tải dữ liệu, các lớp tích chập đồ thị và thư viện các mạng thần kinh đồ thị được xác định trước. Ngoài ra, nó còn cung cấp một loạt công cụ để xây dựng mạng thần kinh dựa trên biểu đồ, chẳng hạn như khả năng xây dựng biểu đồ từ dữ liệu và dễ dàng tạo các lớp chập biểu đồ tùy chỉnh.

PyTorch Geometric khác với các framework deep learning khác ở chỗ nó được thiết kế đặc biệt cho các mạng thần kinh dựa trên đồ thị. Điều này làm cho nó trở thành lựa chọn lý tưởng cho các nhiệm vụ như dự đoán liên kết, phân loại nút và phân loại biểu đồ. Ngoài ra, nó còn cung cấp một loạt công cụ để xây dựng mạng nơ-ron dựa trên biểu đồ, giúp việc tạo các lớp chập biểu đồ tùy chỉnh trở nên dễ dàng hơn.

PyTorch Geometric là một thư viện mạnh mẽ để tạo mạng lưới thần kinh đồ thị (GNN) có thể được sử dụng cho nhiều ứng dụng. Nó được thiết kế để có hiệu quả cao và có khả năng mở rộng, cho phép người dùng xây dựng GNN cho các ứng dụng của riêng họ một cách nhanh chóng và dễ dàng.

PyTorch Geometric cung cấp nhiều tính năng hữu ích để xây dựng GNN, bao gồm nhiều cấu trúc dữ liệu khác nhau để biểu diễn đồ thị, nhiều lớp khác nhau để xây dựng GNN và một loạt bộ dữ liệu để đào tạo và thử nghiệm. Nó cũng cung cấp một loạt công cụ để đào tạo và đánh giá GNN, chẳng hạn như một loạt các hàm mất dữ liệu, trình tối ưu hóa và số liệu.

Để sử dụng PyTorch Geometric cho các ứng dụng GNN, trước tiên người dùng phải tạo một đối tượng biểu đồ, đối tượng này thể hiện dữ liệu sẽ được sử dụng trong GNN. Đối tượng biểu đồ này có thể được tạo bằng cách sử dụng cấu trúc dữ liệu của thư viện, chẳng hạn như lớp Biểu đồ hoặc Dữ liệu. Sau khi đối tượng biểu đồ được tạo, người dùng có thể tạo mô hình GNN bằng cách thêm các lớp, chẳng hạn như lớp tích chập, lớp gộp và các lớp được kết nối dày đặc. Cuối cùng, người dùng có thể huấn luyện và đánh giá mô hình GNN bằng các công cụ đánh giá và đào tạo của thư viện.

PyTorch Geometric là một thư viện mạnh mẽ để tạo GNN và có thể được sử dụng cho nhiều ứng dụng. Bằng cách làm theo các bước được nêu ở trên, người dùng có thể nhanh chóng và dễ dàng tạo GNN cho các ứng dụng của riêng mình.

# CHƯƠNG 3. PHÂN TÍCH VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN LIÊN KẾT NGƯỜI DÙNG TRÊN MEETUP SỬ DỤNG GOOGLE CLOUD

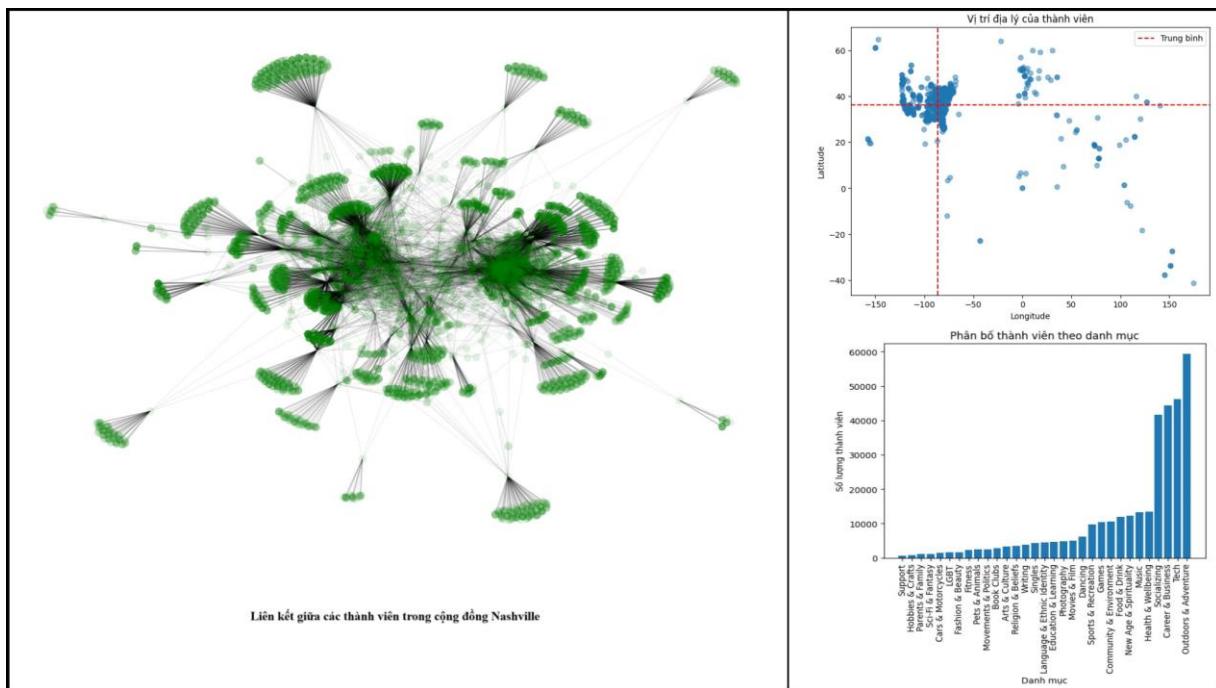
## 3.1. Giới thiệu bộ dữ liệu

Bộ dữ liệu được chọn là bộ dữ liệu về nhóm cộng đồng tại Nashville, được lấy từ kaggle: [Nashville Meetup Network | Kaggle](#)

Trong đó, dữ liệu có chứa 2 phần dữ liệu chính:

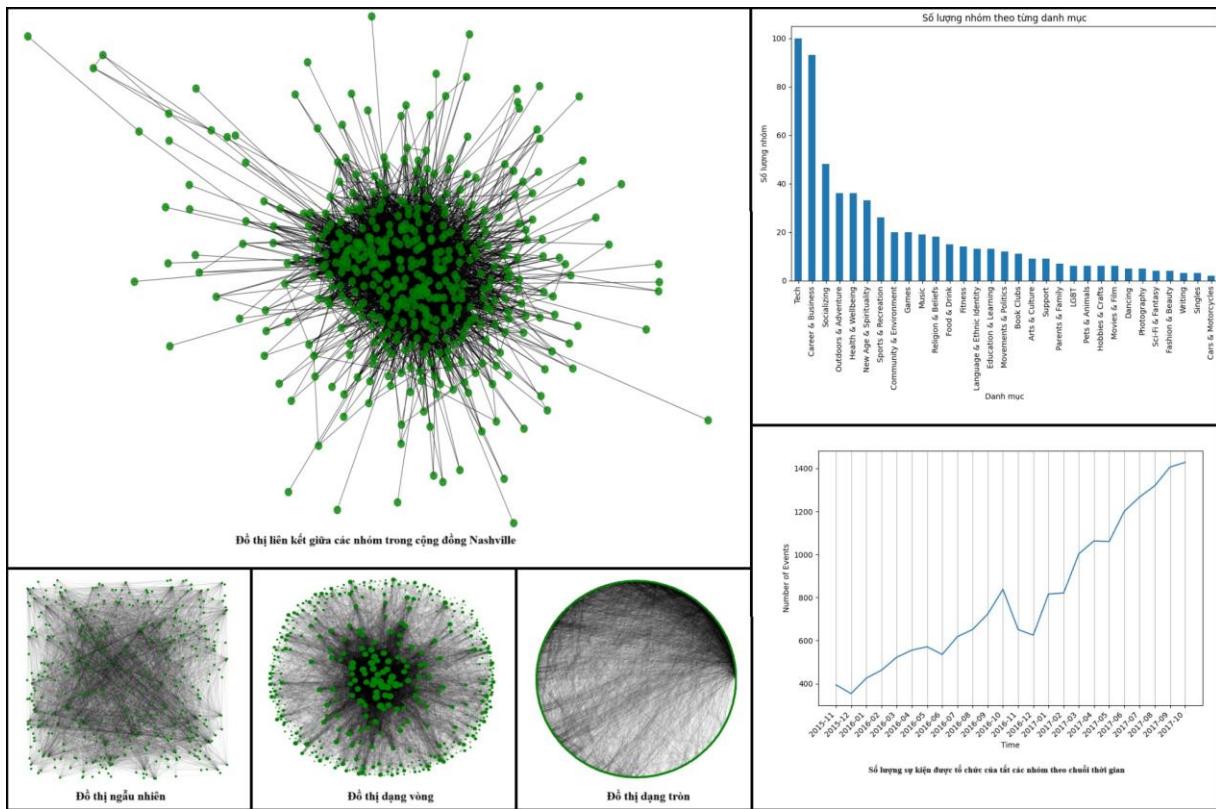
- Dữ liệu đồ thị:
  - member-to-group-edges.csv: Chứa thông tin các cạnh để xây dựng đồ thị giữa các thành viên với các nhóm có trong mạng. Trọng số thể hiện số sự kiện mà thành viên đó đã tham gia trong nhóm.
  - group-edges.csv: Chứa thông tin các cạnh để xây dựng đồ thị giữa các nhóm có trong mạng. Trọng số thể hiện số thành viên chung giữa hai nhóm.
  - member-edges.csv: Chứa thông tin các cạnh để xây dựng đồ thị giữa các thành viên có trong mạng. Trọng số thể hiện số nhóm chung mà cả hai cùng tham gia.
  - rsvps.csv: Dữ liệu gốc về việc tham gia của thành viên vào sự kiện nhóm, đã được tổng hợp để tạo thành member-to-group-edges.csv.
- Dữ liệu thông tin mô tả:
  - meta-groups.csv: Bao gồm các thông tin về nhóm, gồm tên nhóm, id nhóm, số lượng thành viên, và thông tin về các danh mục.
  - meta-members.csv: Bao gồm các thông tin về thành viên, gồm tên và id thành viên, địa chỉ, và vị trí địa lý của người dùng.
  - meta-events.csv: Bao gồm thông tin về các sự kiện, gồm tên và id sự kiện, id group tổ chức sự kiện và thời gian diễn ra.

### 3.2. Thống kê dữ liệu



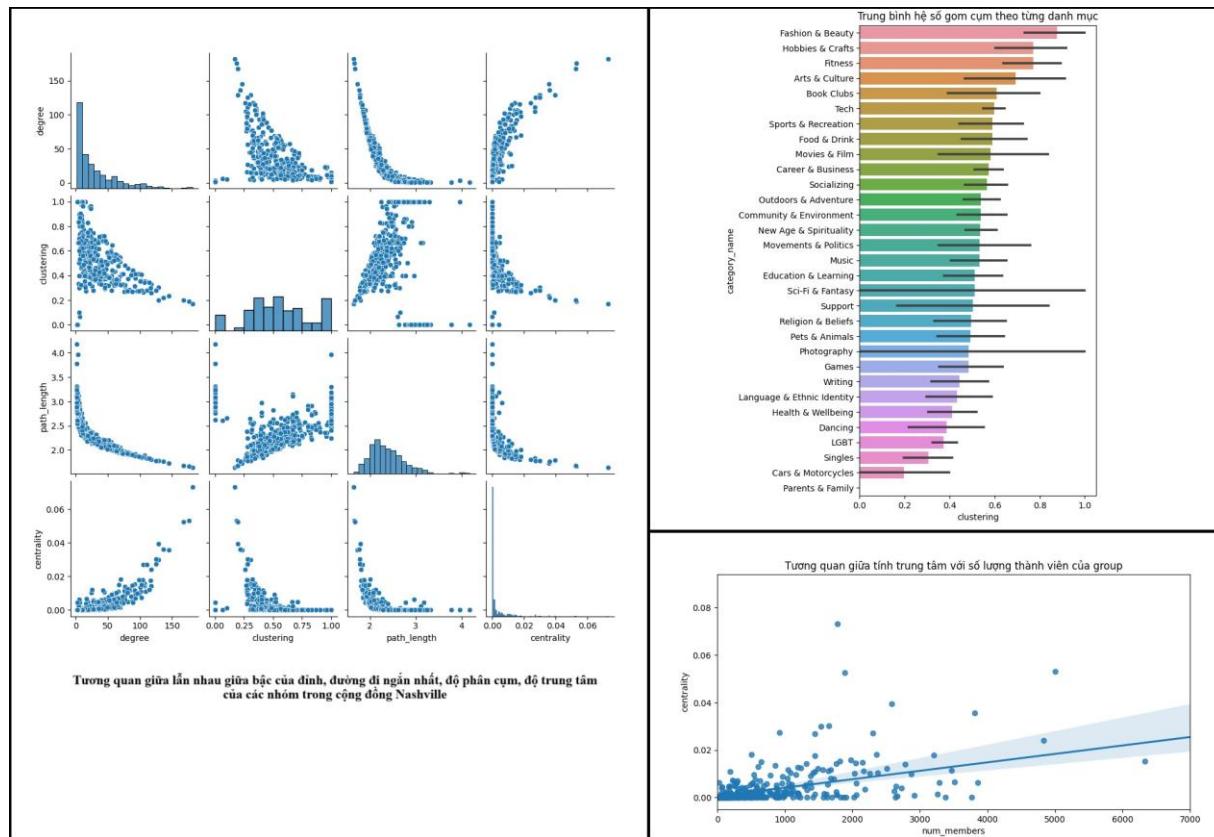
Biểu đồ 1. Trực quan liên kết và thông tin phân bố của các thành viên trong mạng

Từ biểu đồ trên, ta có thể hình tượng hóa được mô hình liên kết giữa các thành viên tham gia vào mạng xã hội Meetup. Đồng thời nắm bắt được về sự phân bố về mặt địa lý cũng như sự phân bố thành viên trong các loại nhóm trong mạng. Từ đồ thị vị trí địa lý, ta có thể thấy hầu hết các thành viên có vị trí trong thành phố Nashville (36, -86) và một phần nhỏ các thành viên đến từ các thành phố lân cận. Các thành viên tham gia vào các nhóm về công việc, xã hội, công nghệ cũng như các hoạt động ngoài trời - thám hiểm chiếm đa số, trong khi các nhóm về hỗ trợ hay ché tạo có khá ít người tham gia. Qua đó ta có thể nắm bắt các vấn đề được các người dùng ở đây quan tâm đến.



Biểu đồ 2. Trực quan mô hình liên kết và thông tin phân loại - hoạt động của các nhóm

Mô hình liên kết các nhóm trong cộng đồng mạng Meetup được thể hiện qua biểu đồ 2. Mô hình được trực quan theo nhiều dạng đồ thị khác nhau. Số lượng các nhóm thuộc chủ đề công nghệ, công việc, xã hội chiếm phần lớn, và nó tỉ lệ thuận với số lượng thành viên quan tâm đến các lĩnh vực này. Đồng thời, số lượng các hoạt động được tổ chức bên trong các group ngày càng có sự gia tăng về số lượng. Điều này cho thấy sự phát triển nhanh chóng của cộng đồng Meetup qua thời gian.



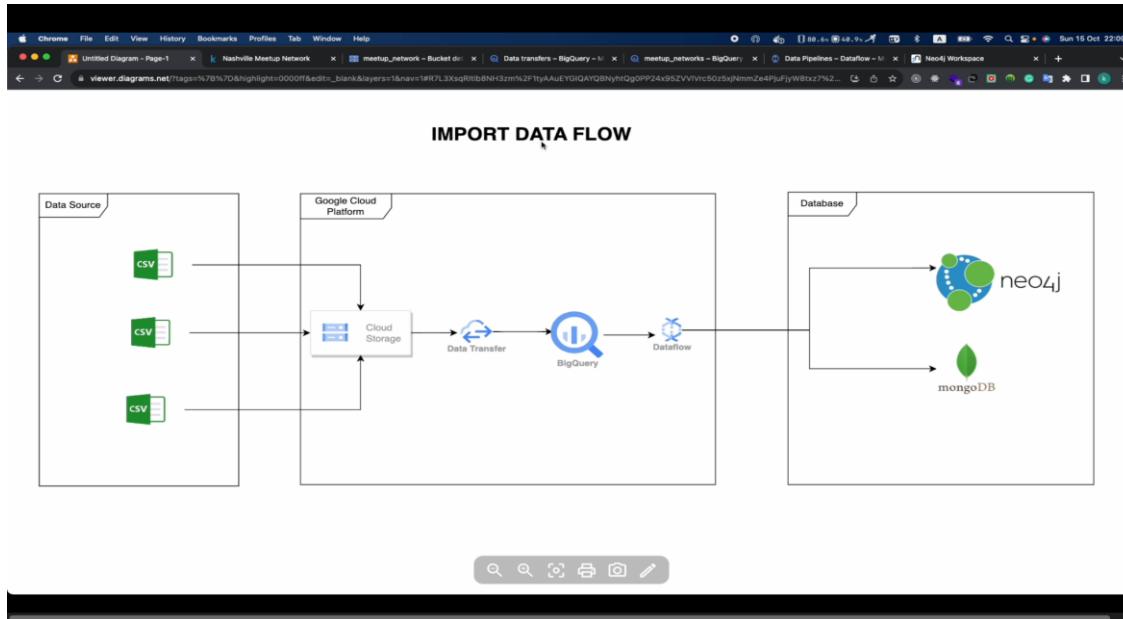
Biểu đồ 3. Tương quan giữa các độ đo của mô hình liên kết các nhóm trong cộng đồng.

Qua biểu đồ ta có thể thấy được sự tương quan của các thông số như hệ số gom cụm, bậc của đỉnh, độ trung tâm của các node trong đồ thị liên kết giữa các nhóm trong mạng Nashville. Dựa vào biểu đồ trên ta thấy cộng đồng về chủ đề thời trang & sắc đẹp dễ dàng liên kết với các cộng đồng khác, trong đó cộng đồng về gia đình thì thường sẽ đứng một mình. Đồng thời thông qua biểu đồ tương quan giữa số lượng thành viên và tính trung tâm của nhóm, ta thấy chưa hẳn 1 cộng đồng với đông đảo thành viên sẽ tạo thành ảnh hưởng lớn, yếu tố số lượng thành viên là 1 yếu tố quan trọng nhưng không phải là yếu tố duy nhất ảnh hưởng đến độ “nổi tiếng” của cộng đồng đó.

### 3.3. Lưu trữ dữ liệu

#### 3.3.1. Tổng quan quá trình lưu trữ dữ liệu

- Gồm 3 thành phần chính: Data source, Google Cloud Platform and Database



- Lưu các file CSV vào Google Cloud Storage ( Các file CSV lấy từ Kaggle)

Name	Type	Created	Storage class	Last modified	Public access	Version history	Error
group-edges.csv	text/csv	Oct 5, 2023, 10:19:58 AM	Standard	Oct 5, 2023, 10:19:58 AM	Not public	—	—
job-specification.json	application/json	Oct 9, 2023, 8:35:27 AM	Standard	Oct 9, 2023, 8:35:27 AM	Not public	—	—
member-edges.csv	text/csv	Oct 5, 2023, 10:20:36 AM	Standard	Oct 5, 2023, 10:20:36 AM	Not public	—	—
member-to-group-edges.csv	text/csv	Oct 15, 2023, 8:40:26 AM	Standard	Oct 15, 2023, 8:40:26 AM	Not public	—	—
meta-groups.csv	text/csv	Oct 14, 2023, 10:01:21 PM	Standard	Oct 14, 2023, 10:01:21 PM	Not public	—	—
meta-members.csv	text/csv	Oct 14, 2023, 10:01:29 PM	Standard	Oct 14, 2023, 10:01:29 PM	Not public	—	—
neo4j-connection-info.json	application/json	Oct 8, 2023, 6:06:04 PM	Standard	Oct 8, 2023, 6:06:04 PM	Not public	—	—

- Sử dụng Data Transfer để đẩy dữ liệu từ Cloud Storage vào BigQuery.

- Khởi tạo các table trước khi đẩy dữ liệu

The screenshot shows the Google Cloud BigQuery schema editor for the 'meta-members' table. The schema includes:

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags
member_id	NUMERIC	NULLABLE				
name	STRING	NULLABLE				
hometown	STRING	NULLABLE				
city	STRING	NULLABLE				
state	STRING	NULLABLE				
lat	FLOAT	NULLABLE				
lon	FLOAT	NULLABLE				

A separate query editor window displays the following SQL code:

```

1 SELECT * FROM `neural-orbit-309209.meetup.networks.meta-members` LIMIT 1000
2
3 SELECT * FROM `neural-orbit-309209.meetup.networks.meta-groups` LIMIT 1000
4
5 SELECT * FROM `neural-orbit-309209.meetup.networks.member-to-group-edges` LIMIT 1000
6
7 delete from `neural-orbit-309209.meetup.networks.member-to-group-edges` where 1=1
  
```

- Tạo Transfer để đẩy dữ liệu từ file csv vào Bigquery

The screenshot shows the 'Create transfer' configuration page. The destination settings are set to 'meetup.networks' and the destination table is 'meta-members'. The data source details indicate a Cloud Storage URI of 'meetup.network/meta-members.csv'. The transfer options include a file format of 'CSV'. A success message 'Transfer deleted' is displayed at the bottom.

- Dữ liệu đã được đẩy thành công vào Bigquery

The screenshot shows the Google Cloud Platform interface with multiple tabs open. The main focus is on the 'meta-members' schema in the BigQuery 'meetup\_network' dataset. The schema includes fields: member\_id (NUMERIC), name (STRING), hometown (STRING), city (STRING), state (STRING), lat (FLOAT), and lon (FLOAT). Below the schema, a query results table displays data from the 'meta-groups' table, showing columns: group\_id, group\_name, num\_members, and category\_id. The results show various groups like 'Nashville Young Professionals', 'Nashville Online Entrepreneurs', etc., categorized under 'Career & Busi'.

group_id	group_name	num_members	category_id
1	18562307	3210	2
2	4126912	1532	2
3	19266990	1444	2
4	4705492	2307	2
5	1776274	1649	2
6	1358081	1447	2
7	1492619	2163	2
8	18414590	690	2
9	16954142	1084	2

- Dùng Data Flow để đẩy dữ liệu từ Bigquery vào 2 cơ sở dữ liệu Neo4J và MongoDB

- Khởi tạo cơ sở dữ liệu MongoDB cùng các bảng

The screenshot shows the MongoDB Compass interface. A modal window titled 'Create Database' is open, prompting for the database name ('meetup-network') and collection name ('meta-groups'). The 'Advanced Collection Options' section is collapsed. At the bottom right of the modal are 'Cancel' and 'Create Database' buttons.

- Sử dụng Dataflow để tự động đẩy dữ liệu từ Bigquery vào các bảng trong MongoDB vào 00h00 hàng ngày

Free trial status: \$9,749,840.37 credit and 42 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

**Pipeline name \***: mongodb-import-meetup-networks

**Regional endpoint \***: us-central1 (Iowa)

**Dataflow template \***: BigQuery to MongoDB

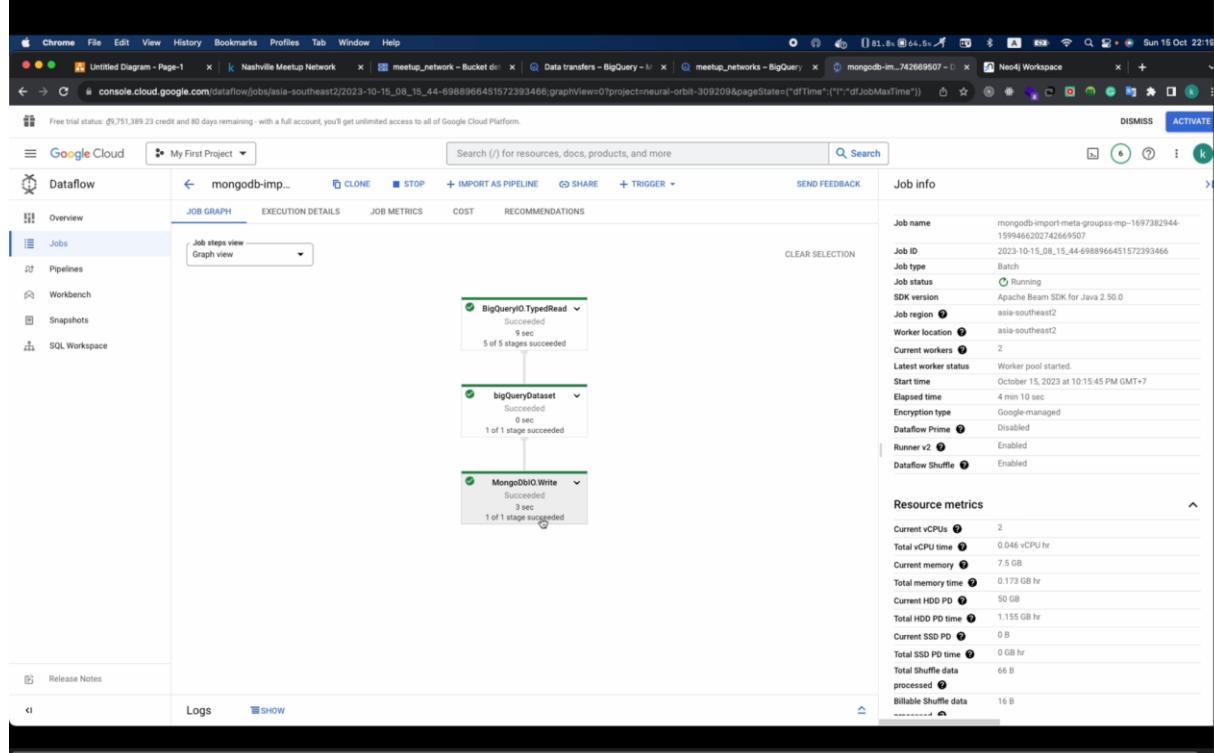
**Schedule your pipeline**

**Repeat \***: Daily

**At time \***: 00:00

**Timezone \***: Indochina Time (ICT)

Run daily at 12:00 AM ICT



- Kiểm tra dữ liệu đã được lưu trữ trong MongoDB

The screenshot shows the MongoDB Compass interface connected to a cluster. The left sidebar shows databases and collections, with 'meetup-network.meta-members' selected. The main area displays the contents of the collection, showing four documents. Each document contains the following fields:

```

{
  "_id": ObjectId("652c82d894724c627bb9a6fa"),
  "member_id": "13864081.00000000",
  "name": "Drew Carpenter",
  "city": "Lexington",
  "state": "KY",
  "lat": 38.06,
  "lon": -84.25
}

{
  "_id": ObjectId("652c82d894724c627bb9a6fb"),
  "member_id": "13864081.00000000",
  "name": "Les. Thompson",
  "city": "Cincinnati",
  "state": "OH",
  "lat": 39.11,
  "lon": -84.5
}

{
  "_id": ObjectId("652c82d894724c627bb9a6fc"),
  "member_id": "148121882.00000000",
  "name": "Olivia",
  "city": "Cincinnati",
  "state": "OH",
  "lat": 39.11,
  "lon": -84.5
}

{
  "_id": ObjectId("652c82d894724c627bb9a6fd"),
  "member_id": "148121882.00000000",
  "name": "Teresa",
  "city": "Cincinnati",
  "state": "OH",
  "lat": 39.11,
  "lon": -84.5
}

```

- Sử dụng Dataflow để tự động đẩy dữ liệu từ Bigquery vào Neo4J vào 00h hằng ngày

The screenshot shows the Google Cloud Platform Dataflow interface. On the left, there's a sidebar with 'Dataflow' selected and other options like Overview, Jobs, Pipelines, Workbench, Snapshots, and SQL Workspace. The main area is titled 'Create pipeline from template'.

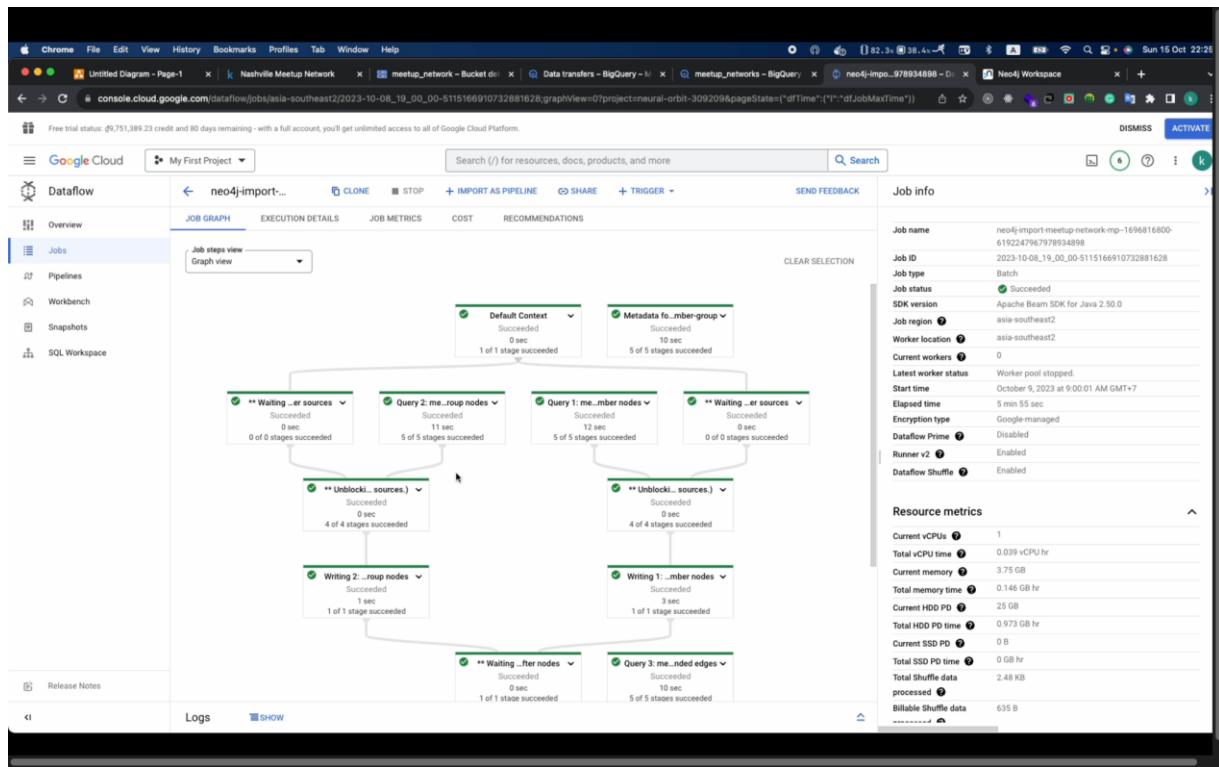
**Pipeline name \***: Neo4j-import-meetup-networks

**Regional endpoint \***: us-central1 (Iowa)

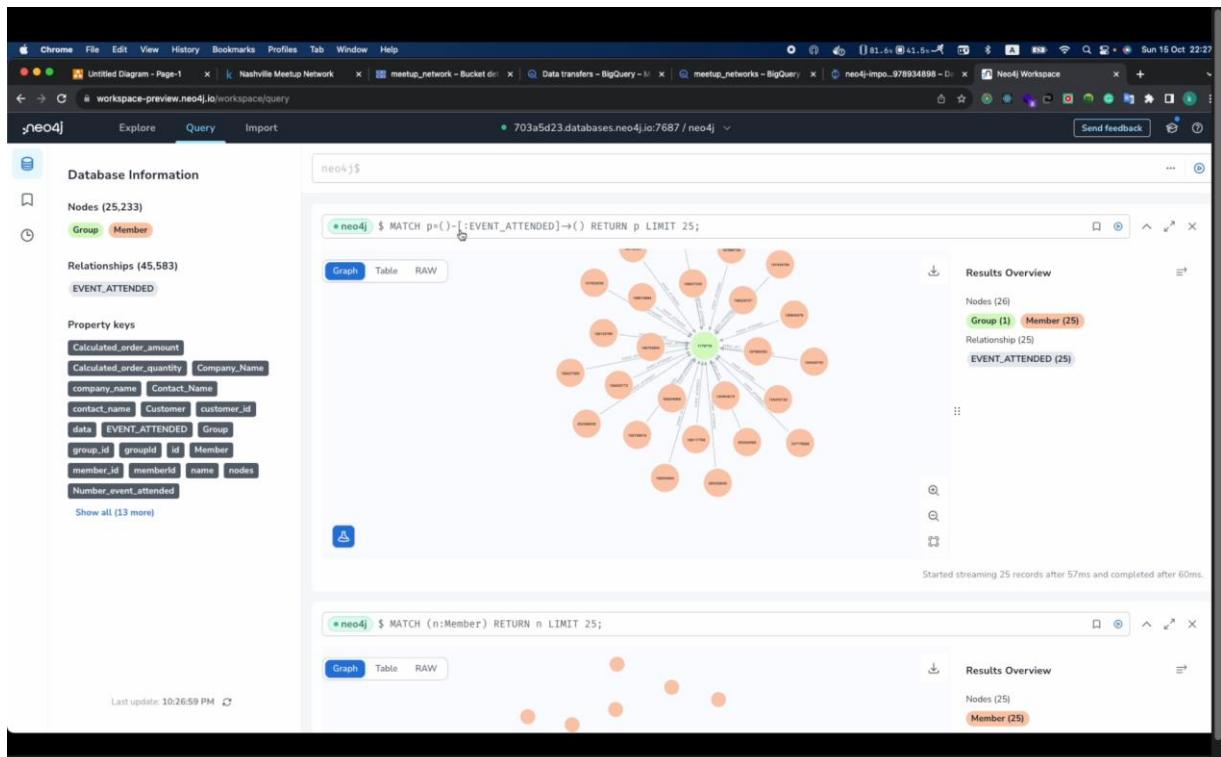
**Dataflow template \***: Google Cloud to Neo4j

**Schedule your pipeline**

- Repeat \***: Daily
- At time \***: 00:00
- Timezone \***: Indochina Time (ICT)



- Kiểm tra dữ liệu đã được lưu trữ trong Neo4J



### 3.4. Tiền xử lý và phân tích dữ liệu

#### 3.4.1. Trích xuất và tiền xử lý dữ liệu

Trích xuất dữ liệu là quá trình rút trích những thông tin cần thiết từ nguồn dữ liệu, trong khi tiền xử lý dữ liệu là quá trình chuẩn bị và làm sạch dữ liệu trước khi phân tích hoặc sử dụng cho các mục đích khác. Đây là hai bước quan trọng để có thể triển khai việc phân tích và ứng dụng thuật toán một cách hiệu quả.

##### Trích xuất dữ liệu:

- Kết nối với neo4j: Để có thể trích xuất dữ liệu từ neo4j thì bước đầu tiên là kết nối với server neo4j chứa dữ liệu cần dùng. Sử dụng thư viện GraphDatabase trong neo4j của Google Colab.

```
from neo4j import GraphDatabase

uri = "neo4j+s://5ee17651.databases.neo4j.io"
password = "Xt79kmF2M9V0c5zPouJBIFcihMd792ExxQmwbclsqDM"

graphdb = GraphDatabase.driver(uri,
                               auth=("neo4j", password))
session = graphdb.session()
```

- Gọi lệnh Query để lấy dữ liệu: sử dụng hàm session.run(query) để gọi lệnh truy vấn trong python. Khi đó giá trị gọi về sẽ có dạng key-value với n node.
- Ví dụ: Xuất tất cả giá trị của 25 node bất kì trong mạng neo4j

```

query = "MATCH (n:member) RETURN n LIMIT 25;"|_
nodes = session.run(query)
for node in nodes:
    properties = node["n"]
    for key in properties.keys():
        value = properties[key]
        print(key, ':', value)
    print('-----')

member_id : 2069
hometown : Brentwood
city : Brentwood
name : Wesley Duffee-Braun
lon : -86.79
state : TN
lat : 36.0
-----
```

- Lưu dữ liệu về dataframe để sử dụng: sử dụng hàm session.run(query) để truy vấn thông tin kế tiếp lưu và các biến group1, group2, weight. Cuối cùng sử dụng hàm pd.DataFrame() để gộp thành 1 dataframe hoàn chỉnh cho group. Làm tương tự với member để có được dataframe cho member.

```

query = "MATCH p=()-[r:group_edges]->() RETURN p, r.weight AS weight;"|_
result = session.run(query)

group1 = []
group2 = []
weight=[]
for record in result:
    start_node = record["p"].nodes[0]["group_id"]
    end_node = record["p"].nodes[-1]["group_id"]
    weight_node = record["weight"]

    group1.append(start_node)
    group2.append(end_node)
    weight.append(weight_node)

df_gr_edge= pd.DataFrame({'group1': group1, 'group2':group2, 'weight': weight})|_
df_gr_edge[0:50]

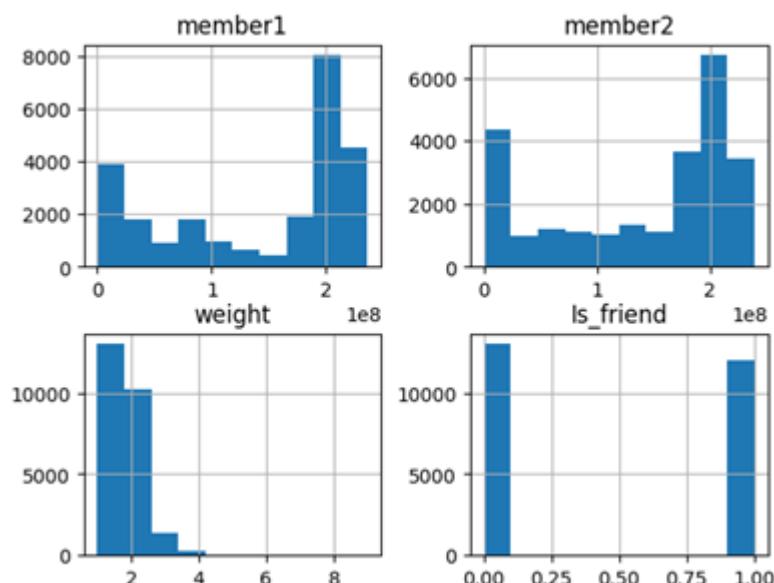
   group1  group2  weight
0  339011  19728145      24
1  339011  10016242       2
```

### Tiền xử lý dữ liệu:

- Lọc dữ liệu: Các dữ liệu liên kết của người dùng bị thiếu thông tin không thể dùng để xây dựng mô hình dự đoán nên chúng tôi đã tiến hành loại bỏ chúng ra khỏi bộ dữ liệu huấn luyện.

- Điền dữ liệu thiếu: Với các dữ liệu bị thiếu, chúng tôi thực hiện điền giá trị bị thiếu bằng giá trị có nhiều nhất trong thuộc tính đó.

Sau đó, chúng tôi thực hiện khảo sát dữ liệu và gán nhãn bạn bè giữa các liên kết.



Hình biểu đồ phân bố các giá trị của thuộc tính

Từ bảng khảo sát liên kết với weight là số nhóm chung của 2 user. Để đảm bảo dữ liệu cân bằng, chúng tôi quy định với các member có tham gia chung 2 group trở lên sử là bạn bè. Kết quả thu được 1 bộ dữ liệu với hơn 25000 liên kết giữa các người dùng sử dụng cho việc dự đoán liên kết.

### 3.4.2. Các độ đo trung tâm

#### 3.4.2.1. Độ đo degree centrality

**Khái niệm:** Degree centrality là một khái niệm quan trọng trong phân tích mạng lưới (network analysis). Nó đo lường mức độ quan trọng của một đỉnh (node) trong mạng dựa trên số lượng kết nối (edges) mà đỉnh đó có. Độ đo degree centrality đơn giản là số lượng kết nối mà một đỉnh có trong mạng.

Degree centrality được sử dụng để xác định các đỉnh quan trọng trong mạng. Degree centrality càng cao, có nghĩa là đỉnh đó có số lượng kết nối (liên kết, cạnh) với các đỉnh khác trong mạng lưới.

**Bảng 1:** Kết quả top 5 thành viên có degree centrality cao nhất trong mạng

Node	Degree Centrality
6160486	996
195657825	806
85557392	761
204669023	577
190939281	566

**Bảng 2:** Kết quả top 5 nhóm có degree centrality cao nhất trong mạng

Node	Degree Centrality
19728145	182
18955830	176
1187715	168
18506072	145
339011	136

### 3.4.2.2. Độ đo closeness centrality

**Khái niệm:** Closeness centrality là một khái niệm trong phân tích mạng lưới (network analysis) để đo mức độ tiếp cận của một đỉnh (node) với tất cả các đỉnh khác trong mạng. Nó đo lường độ xa gần của một đỉnh dựa trên khoảng cách ngắn nhất (shortest path) từ đỉnh đó đến tất cả các đỉnh khác trong mạng.

$$c_{clos}(x) = \frac{1}{\sum_y d(y, x)}$$

- Trong đó:

- $d(y, x)$ : Độ dài đường đi ngắn nhất từ đỉnh  $y$  đến đỉnh  $x$ .

Closeness centrality thường được sử dụng để xác định các đỉnh trung tâm và quan trọng trong mạng. Closeness centrality càng cao, có nghĩa là đỉnh đó có mức độ tiếp cận gần hơn với các đỉnh khác trong mạng lưới.

**Bảng 3:** Kết quả top 5 thành viên có closeness centrality cao nhất trong mạng

Node	Closeness Centrality
6160486	0.387242941791565
221191725	0.3752533213240261
195657825	0.3702099300233256
127601602	0.3610464171586416
5900662	0.36020750027018267

**Bảng 4:** Kết quả top 5 nhóm có closeness centrality cao nhất trong mạng

Node	Closeness Centrality
19728145	0.6074766355140186
18955830	0.600263852242744
1187715	0.5963302752293578
18506072	0.5781448538754765
339011	0.5666251556662516

### 3.4.2.3. Độ đo betweenness centrality

**Khái niệm:** Betweenness centrality là một khái niệm trong phân tích mạng lưới (network analysis) để đo mức độ quan trọng của một đỉnh (node) trong việc trung gian giao tiếp giữa các đỉnh khác trong mạng. Nó đo lường độ tầm ảnh hưởng của một đỉnh trong việc kiểm soát luồng thông tin hoặc tương tác giữa các cặp đỉnh khác trong mạng.

$$c_{\text{bet}}(x) = \sum_{y,z \neq x, \sigma_{yz} \neq 0} \frac{\sigma_{yz}(x)}{\sigma_{yz}}$$

▪ Trong đó:

- $\sigma_{yz}$ : Số đường đi ngắn nhất từ  $y$  đến  $z$ .
- $\sigma_{yz}(x)$ : Số đường đi ngắn nhất từ  $y$  đến  $z$  đi qua  $x$ .

Betweenness centrality thường được sử dụng để xác định các đỉnh trung tâm và quan trọng trong mạng. Betweenness centrality càng cao, có nghĩa là đỉnh đó có vai trò trung gian quan trọng hơn trong mạng lưới.

**Bảng 5:** Kết quả top 5 thành viên có betweenness centrality cao nhất trong mạng

Node	Betweenness Centrality
6160486	0.2029501983670458
195657825	0.12884112885866053
226754592	0.09381998676399958
85557392	0.08988957356016605
190939281	0.06817455092321273

**Bảng 6:** Kết quả top 5 nhóm có betweenness centrality cao nhất trong mạng

Node	Betweenness Centrality
19728145	0.0731614126896249
18955830	0.05311218560607821
1187715	0.05250332056091229
18243826	0.03948797055642313
339011	0.03625139635677649

#### 3.4.2.4. Độ đo eigenvector centrality

**Khái niệm:** Eigenvector centrality là một phương pháp đo lường tầm quan trọng của các đỉnh trong một mạng lưới. Nó xem xét không chỉ số lượng kết nối của một đỉnh mà còn tầm quan trọng của các đỉnh mà đỉnh đó kết nối với.

$$c_{\text{eig}}(x) = \frac{1}{\lambda} \sum_{y \rightarrow x} c_{\text{eig}}(y)$$

▪ Trong đó:

- $c_{\text{eig}}$ : hối tụ đến vector eigen của ma trận kè  $A$ .
- $\lambda$ : hối tụ đến giá trị đặc biệt của vector eigen của ma trận kè  $A$ .
- $\lambda = \|c_{\text{eig}}\|_2 = \sqrt{\sum_{r=1}^n |c_r|^2}$  với  $c$  là vector có các thành phần  $c_1, c_2, \dots, c_r$ .

Kết quả của eigenvector centrality có thể được hiểu là một phân bố tương đối của tầm quan trọng trong mạng, và các đỉnh có giá trị cao hơn sẽ có ảnh hưởng lớn hơn. Eigenvector centrality càng cao, có nghĩa là đỉnh đó có tầm quan trọng và ảnh hưởng càng lớn trong mạng lưới.

**Bảng 7:** Kết quả top 5 thành viên có eigenvector centrality cao nhất trong mạng

Node	Eigenvector Centrality
85557392	0.2413830940262894
203818557	0.18485667889934462
184389351	0.17953068141380235
211585840	0.1688574855800939
208569099	0.16549283105204365

**Bảng 8:** Kết quả top 5 nhóm có eigenvector centrality cao nhất trong mạng

Node	Eigenvector Centrality
18955830	0.14645235841457377
1187715	0.14583075790052097

19728145	0.14546248933085848
4126912	0.14064239117518768
18506072	0.13917812377816893

### 3.4.2.5. Độ đo Page Rank

**Khái niệm:** PageRank là một độ đo trong mạng, đo lường sự quan trọng của một đỉnh trong mạng dựa trên số lượng liên kết đến nó và sự quan trọng của những đỉnh liên kết đó. Nó được sử dụng để xác định mức độ ảnh hưởng của một đỉnh trong mạng và thường được sử dụng trong các thuật toán tìm kiếm và xếp hạng trang web.

$$P(i) = \sum_{j \in B(i)} \frac{P(j)}{O_j}$$

- Trong đó:
  - $B(i)$  là tập các trang có liên kết trả về trang  $i$ .
  - $O_j$  là số liên kết ra từ trang  $j$  (outbound link).

Khi kết quả PageRank của một đỉnh trong mạng càng cao, tức là đỉnh đó có số lượng liên kết đến và sự quan trọng của những đỉnh liên kết đó càng cao, cho thấy mức độ ảnh hưởng của đỉnh đó trong mạng cũng càng cao. Điều này có thể đảm bảo rằng nó là một đỉnh quan trọng và có sự ảnh hưởng lớn trong mạng.

**Bảng 9:** Kết quả top 5 thành viên có page rank cao nhất trong mạng

Node	Page Rank
226754592	0.019616222613307414
6160486	0.017184304105386804
195657825	0.013057277282691796
85557392	0.01015041277644306
214466652	0.0092871040047275

**Bảng 10:** Kết quả top 5 nhóm có page rank cao nhất trong mạng

Node	Page Rank
19728145	0.14645235841457377
10016242	0.14583075790052097
18955830	0.017811732153714566
1187715	0.015032622951214306
16487812	0.014968222592691201

### 3.4.3. Phát hiện cộng đồng

#### 3.4.3.1. Thuật toán phát hiện cộng đồng Louvain

Louvain là một thuật toán được sử dụng để phát hiện cộng đồng trong đồ thị của mô hình mạng. Giải thuật này đã giành được nhiều đánh giá tích cực trong cộng đồng khoa học máy tính và được áp dụng nhiều trong các ứng dụng thực tế.

Thuật toán này tập trung vào việc tối ưu hóa sự tương tác bên trong các cụm và giữ cho mỗi quan hệ giữa các cụm một cách yếu nhất có thể. Nó được sử dụng rộng rãi để phân tích mạng xã hội, mạng lưới thông tin, và các hệ thống phức tạp khác.

#### Cách thức hoạt động:

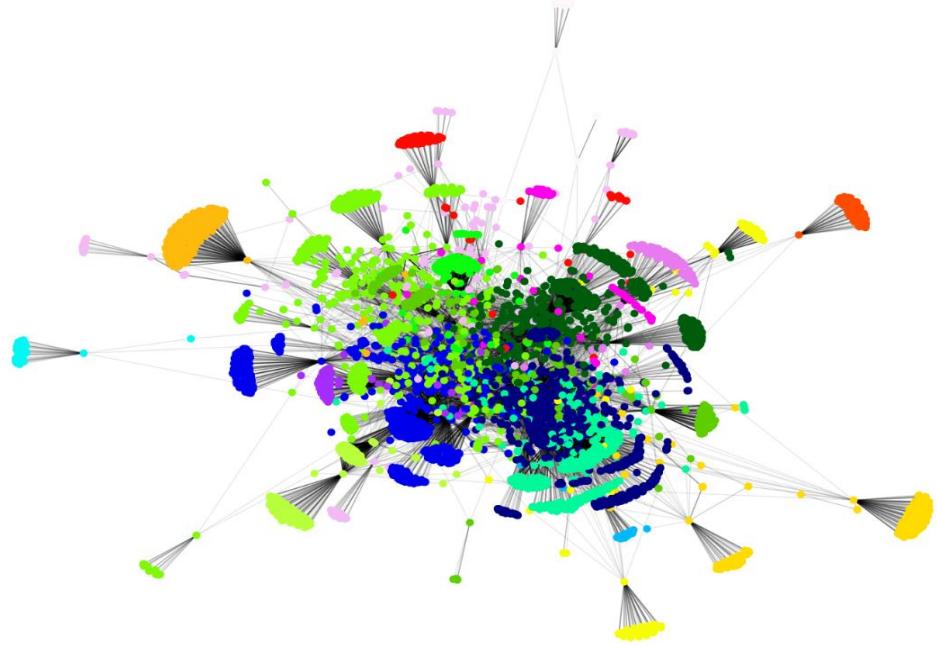
- Bước 1: Khởi tạo mỗi đỉnh trong đồ thị là một cộng đồng riêng biệt.
- Bước 2: Lặp lại cho đến khi không thể tối ưu hóa được nữa, bao gồm 2 pha con:
  - Pha 1: Kiểm tra tất cả các đỉnh kề với đỉnh hiện tại và xem liệu việc chuyển đỉnh đã chọn sang cộng đồng khác có tăng giá trị modularity trong đồ thị hay không. Nếu có, chuyển đỉnh sang cộng đồng đó.
  - Pha 2: Trộn các cộng đồng trong đồ thị và lặp lại pha 1

Các bước trên được lặp lại cho đến khi không thể tối ưu hóa được giá trị modularity trong đồ thị.

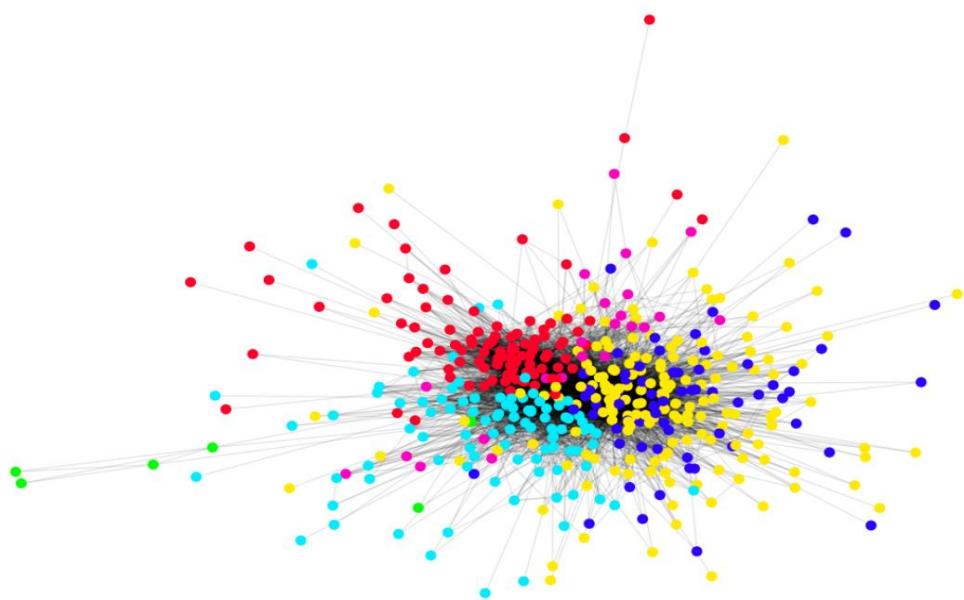
- Công thức tính modularity trong đồ thị:

$$\text{modularity} = [\text{tổng các cạnh trong cộng đồng}] - [\text{tổng bậc của các đỉnh trong cộng đồng}]^2 / 2m$$

trong đó: m là tổng số cạnh trong đồ thị.



Kết quả thuật toán phát hiện cộng đồng Louvain đối với các thành viên.



Kết quả thuật toán phát hiện cộng đồng Louvain đối với các nhóm.

### 3.5. Xây dựng mô hình dự đoán liên kết

#### 3.5.1. Dự đoán dựa trên độ tương đồng cục bộ

##### 3.5.1.1. Thuật toán Jaccard

Jaccard là một phép đo đánh giá sự tương đồng hoặc khác biệt giữa hai tập hợp dữ liệu. Nó được sử dụng để đo lường mức độ giống nhau giữa hai tập hợp bằng cách tính tỉ lệ giữa số lượng phần tử chung và tổng số phần tử không trùng lặp trong hai tập hợp. Đây thường được áp dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên, khai thác dữ liệu và học máy để so sánh sự tương đồng giữa văn bản, tập hợp từ vựng, hoặc các đặc trưng dữ liệu khác.

Cụ thể hệ số Jaccard được tính như sau:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$

Trong đó,  $0 \leq J(A, B) \leq 1$ . Nếu giao điểm của A và B rỗng thì  $J(A, B) = 0$ . Khi hệ số Jaccard càng tiến gần về 1 thì hai điểm càng có khả năng liên kết với nhau.

**Bảng 11:** Kết quả dự đoán Top 5 cặp thành viên dễ liên kết với nhau nhất dựa trên Jaccard.

Source	Target	Jaccard
183871346	45382252	1.0
222643307	215317302	1.0
223154086	196076811	1.0

223154086	227796243	1.0
223154086	220226916	1.0

**Bảng 12:** Kết quả dự đoán Top 5 cặp nhóm dễ liên kết với nhau nhất dựa trên Jaccard.

Source	Target	Jaccard
25327081	18849449	1.000000
25326528	25482608	1.000000
22817838	18779992	0.527273
21533726	16447162	0.517241
18729267	18549827	0.500000

### 3.5.1.2. Thuật toán Adamic-Adar

Mức độ tương đồng Adamic-Adar là một phương pháp đo lường độ quan trọng của sự chung của các đỉnh trong mạng lưới (graph). Nó dựa trên cơ sở rằng những đỉnh ít chung với các đỉnh khác sẽ có đóng góp lớn hơn cho tính chất độc đáo của một liên kết.

Cụ thể, nó sử dụng đồ thị kết nối để tính toán giá trị tương đồng giữa các đỉnh bằng cách gán trọng số cao hơn cho các liên kết qua các đỉnh ít liên kết hơn. Phương pháp này thường được sử dụng trong việc đo lường sự tương đồng giữa các người dùng hoặc các đối tượng trong các mạng xã hội hoặc mạng lưới thông tin.

Mức độ tương đồng Adamic-Adar được tính như sau:

$$A(x, y) = \sum_{u \in N(x) \cap N(y)} \frac{1}{\log |N(u)|}$$

Trong đó, x và y là hai đỉnh trong đồ thị, N(x) và N(y) là tập hợp các đỉnh kề (các đỉnh được kết nối trực tiếp) với x và y tương ứng.

Ý tưởng chính của công thức này là đánh giá độ tương đồng giữa hai đỉnh x và y dựa trên đóng góp thông tin của các đỉnh chung. Nếu hai đỉnh có nhiều đỉnh kề chung ít liên kết hơn, tức là đỉnh kề đó không phổ biến, thì đóng góp thông tin của chúng sẽ cao hơn và dẫn đến giá trị tương đồng cao hơn theo công thức Adamic-Adar.

**Bảng 13:** Kết quả dự đoán Top 5 cặp thành viên dễ liên kết nhất dựa trên Adamic-Adar.

Source	Target	Adamic-Adar
234684445	6160486	40.513492
6160486	5900662	35.876661
6160486	53121132	28.873096
207061281	211585840	28.793302
85557392	73895512	27.537759

**Bảng 14:** Kết quả dự đoán Top 5 cặp nhóm dễ liên kết nhất dựa trên Adamic-Adar.

Source	Target	Adamic-Adar
11077852	1776274	15.689848
18243826	18562307	14.739374
1728035	18955830	14.392966
18506072	18589616	13.658258
19728145	18589616	13.073634

### 3.5.1.3. Thuật toán Preferential Attachment

Preferential Attachment là một khái niệm trong mạng lưới, mô tả mối quan hệ giữa sự phát triển của các liên kết trong một mạng. Nó cho biết rằng các nút có nhiều kết nối sẽ có khả năng cao hơn để tạo ra các kết nối mới so với những nút có ít kết nối. Điều này có thể giúp dự đoán xu hướng phát triển của mạng và hiểu rõ về sự phát triển của các liên kết trong môi trường mạng cụ thể.

$$PA = d_x d_y$$

Trong đó,  $d_x$  là bậc của đỉnh  $x$ ,  $d_y$  là bậc của đỉnh  $y$ .

**Bảng 15:** Kết quả dự đoán Top 5 cặp thành viên dễ liên kết nhất dựa trên Adamic-Adar.

Source	Target	Preferential Attachment
195657825	6160486	802776
195657825	85557392	613366
204669023	6160486	574692
6160486	190939281	563736
6160486	191758521	461148

**Bảng 16:** Kết quả dự đoán Top 5 cặp nhóm dễ liên kết nhất dựa trên Adamic-Adar.

Source	Target	Preferential Attachment
10016242	19728145	19474
1728035	18955830	19184
19728145	18589616	18200
10016242	1187715	17976
19728145	18616278	16926

### 3.5.2. Dự đoán dựa trên độ tương đồng toàn cục

#### 3.5.2.1. Thuật toán Hitting time

Thuật toán Hitting time là một khái niệm trong lý thuyết đồ thị và phân tích mạng, được sử dụng để đo lường thời gian cần thiết để đi từ một đỉnh đến một đỉnh khác trong một mạng lưới.

Ý tưởng của thuật toán Hitting Time là tính toán thời gian trung bình mà một random walker (bước ngẫu nhiên) đi từ một nút nguồn đến một nút đích trong đồ thị. Khi tính toán thời gian hitting time của một cặp nút, ta sử dụng tất cả các đường đi có thể có từ nút nguồn đến nút đích, và trọng số của đường đi ở đây được xác định bởi xác suất di chuyển từ một nút sang nút kế tiếp (sử dụng xác suất chuyển từ ma trận xác suất chuyển).

Thời gian hitting time cho cặp nút  $(u, v)$  được tính bằng công thức:

$$\text{hit\_time}(u, v) = \text{sum}[\text{shortest}(u, i) + \text{shortest}(v, i)] / (2 * (n-2))$$

Trong đó:

- $\text{shortest}(u, i)$  là thời gian ngắn nhất từ nút  $u$  đến nút  $i$ .
- $\text{shortest}(v, i)$  là thời gian ngắn nhất từ nút  $v$  đến nút  $i$ .
- $n$  là số nút có trong đồ thị mạng.

**Bảng 17:** Kết quả dự đoán Top 5 cặp thành viên dễ liên kết với nhau nhất dựa trên Hitting time

Source	Target	Hitting Time
211551769	154764282	1.2621828343891703e-

211551769	183097071	1.2621828343891703e-
211551769	8809394	1.2621828343891703e-
211551769	212660721	1.2621828343891703e-
211551769	56356372	1.2621828343891703e-

**Bảng 18:** Kết quả dự đoán Top 5 cặp nhóm dễ liên kết với nhau nhất dựa trên Hitting time

Source	Target	Hitting Time
19292162	24556869	4.833728021980107e-
535553	24556869	4.833728021980107e-
19194894	24556869	4.833728021980107e-

19728145	24556869	4.833728021980107e-
18850080	24556869	4.833728021980107e-

### 3.5.2.1. Thuật toán Katz Global

Thuật toán Katz global là một thuật toán trong phân tích mạng lưới, được sử dụng để đo lường tầm quan trọng của các đỉnh trong mạng dựa trên sự kết hợp giữa tầm quan trọng của các đỉnh kề cận và các đỉnh từ xa.

Katz Global Function gán điểm cho mỗi đỉnh dựa trên mối quan hệ của nó với các đỉnh khác trong mạng. Nó không chỉ tính đến mối quan hệ trực tiếp mà còn tính đến mối quan hệ gián tiếp thông qua các đỉnh khác. Điểm số cho mỗi đỉnh được tính toán bằng cách cộng tổng trọng số của các đỉnh láng giềng trực tiếp, được nhân với một tham số gọi là "hệ số giảm". Hệ số giảm xác định sự ảnh hưởng của mối quan hệ gián tiếp đối với điểm số của đỉnh.

Katz Global Function có thể được biểu diễn bằng phương trình:

$$\textcolor{red}{x} = \beta \textcolor{blue}{A} \textcolor{red}{x} + \textcolor{brown}{a} \textcolor{violet}{b}$$

với:

- $x$  là vector biểu thị điểm số của các đỉnh.
- $A$  là ma trận kề của mạng xã hội.
- $\beta$  là hệ số giảm.
- $b$  là vector biểu thị điểm số ban đầu hoặc mức độ quan trọng được gán cho mỗi đỉnh.

- $\alpha$  là hệ số tỷ lệ.

Bằng cách tính toán lặp lại Katz Global Function, ta có thể thu được điểm số của các đỉnh, cung cấp một đại lượng đo lường sự tâm trung hoặc quan trọng của chúng trong mạng. Các đỉnh có điểm số Katz cao được coi là tâm trung hoặc có ảnh hưởng hơn.

**Bảng 19:** Kết quả dự đoán Top 5 cặp thành viên dễ liên kết với nhau nhất dựa trên Katz score

Source	Target	Katz score
183566364	28246202	0.06364533048488252
183566364	73498632	0.053907016046973136
183566364	5900662	0.05280196470369172
183566364	88375412	0.0459561228101436
183566364	76743532	0.03993978494669909

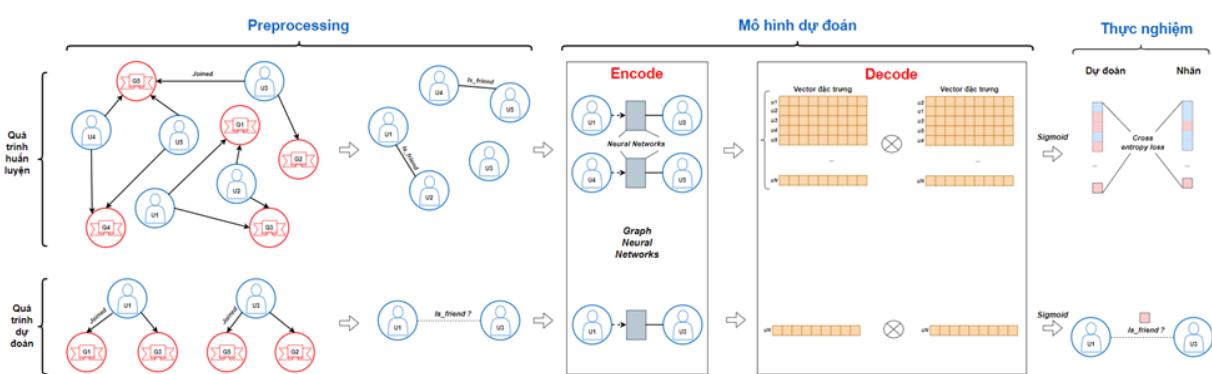
**Bảng 20:** Kết quả dự đoán Top 5 cặp nhóm dễ liên kết với nhau nhất dựa trên Katz score

Source	Target	Katz score
252576	1274150	0.03935477898092444
6335372	1274150	0.03089331721996152

18589616	1274150	0.02954961958950432
252576	18672773	0.024845454521672722
252576	20348898	0.024601756623378258

### 3.5.3. Dự đoán dựa trên máy học

Quy trình xây dựng mô hình dự đoán liên kết của chúng tôi bao gồm các thành phần chính sau: 1) Tiền xử lý dữ liệu, 2) Xây dựng mô hình dự đoán, 3) Thực nghiệm. Được mô tả như hình bên dưới.



Mô hình tổng quan dự đoán liên kết dựa trên mô hình máy học có giám sát

#### 3.5.3.1. Xử lý dữ liệu huấn luyện mô hình

Để xây dựng được mô hình dự đoán liên kết (link prediction), chúng tôi xác định các thuộc tính được sử dụng như sau:

- o `Nodes` - Members (bằng ID)
- o `Edges` - Is\_friend (1: yes, 0: no)
- o `Node Features` - hometown, city, state, lat, lon (tất cả thuộc tính có của node (member))
  - o `Labels` - Is\_friend (1: yes, 0: no)
    - Node feature + Node index
      - o Chuyển các giá trị category sang numeric: Đối với các thuộc tính có giá trị category (ví dụ: thuộc tính 'city': Brentwood, Nashville, Antioch) sẽ được chuyển sang giá trị numeric mới có thể dùng được để huấn luyện mô hình (ví dụ: thuộc tính 'city': 0, 1, 2)
      - o Bên cạnh đó, vì ID member bị thua nhiều nên chúng tôi cũng thực hiện đánh index cho member để thuận tiện cho việc tìm kiếm.
      - o Dữ liệu node features sẽ có chiều (24962,4) (số member, số thuộc tính)
        - Label
          - o Chúng tôi thực hiện join dữ liệu edge\_member với dữ liệu member\_metadata theo member cột 1 để xác định các thuộc tính của member có trong file edge nhằm lấy ra nhãn Is\_friend tương ứng với member feature đó.
          - o Dữ liệu nhãn ứng với dữ liệu metadata sẽ có chiều (24962) (số member)
            - Edges + Label edge index
              - o Nhãn Edges ở tập huấn luyện sẽ được chúng tôi lấy trong file edge\_member với số chiều là (2, 1563) (2, số liên kết)

- o Nhãn Edges ở tập dự đoán sẽ được dùng để dự đoán liên kết, được chúng tôi join dữ liệu edge\_member với dữ liệu member\_metadata theo member cột 2 để làm dữ liệu đầu ra cho mô hình. Có số chiều là (2, 7489) (2, số liên kết)

- Chia dữ liệu huấn luyện

- o Dữ liệu được thành 2 tập train và test theo tỉ lệ (7:3) cho việc huấn luyện và đánh giá mô hình. Sau đó sẽ được chia thành định dạng của pytorch để phù hợp với yêu cầu đầu vào của mô hình dự đoán.

```
data_train
Data(x=[24962, 4], edge_index=[2, 1563], y=[24962], valid_mask=[24962], edge_label=[17472], edge_label_index=[2, 17472])
```

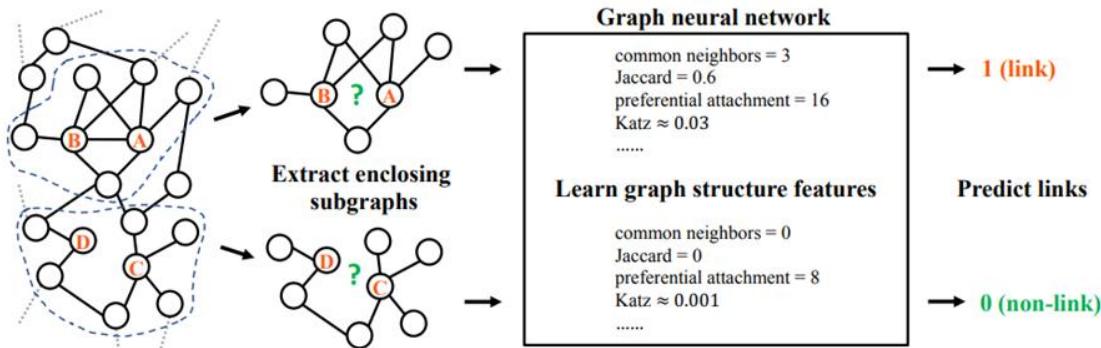
Hình ảnh dữ liệu huấn luyện theo định dạng pytorch

```
data_test
Data(x=[24962, 4], edge_index=[2, 1563], y=[24962], valid_mask=[24962], edge_label=[7489], edge_label_index=[2, 7489])
```

Hình ảnh dữ liệu kiểm thử theo định dạng pytorch.

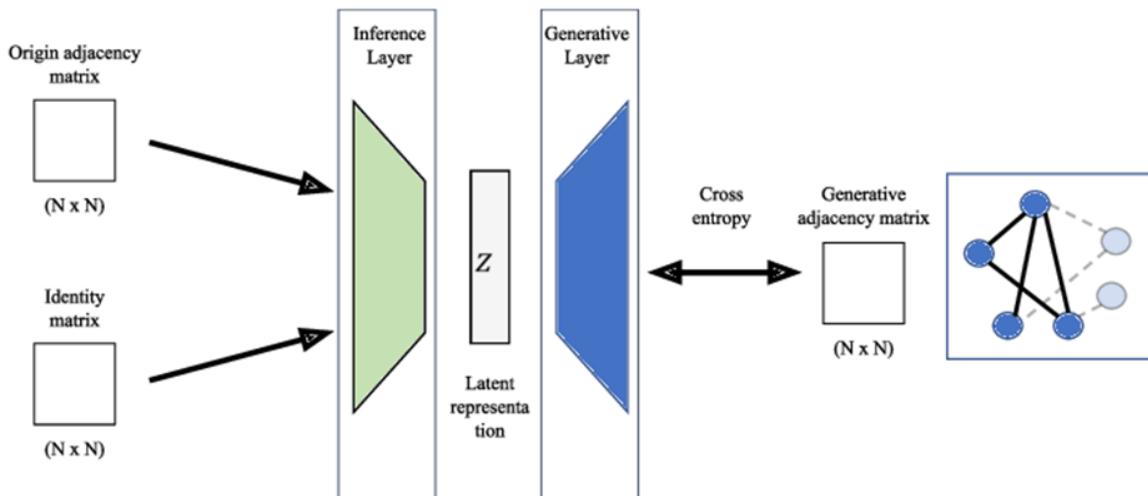
### 3.5.3.2. Xây dựng mô hình dự đoán liên kết

- Tổng quan



**Figure 1:** The SEAL framework. For each target link, SEAL extracts a local enclosing subgraph around it, and uses a GNN to learn general graph structure features for link prediction. Note that the heuristics listed inside the box are just for illustration – the learned features may be completely different from existing heuristics.

Mô hình dự đoán liên kết dựa trên GNN



Mô hình dự đoán liên kết dựa trên mô hình autoencoder

Về xây dựng mô hình dự đoán, đầu tiên, chúng tôi sẽ chuyển các thuộc tính của các node và các liên kết giữa các member dựa vào các mô hình trích xuất đặc trưng trên graph để tạo ra các vec-tor đặc trưng. Các vec-tor này sẽ tương ứng với từng đặc trưng của người dùng. Vec-tor được tạo ra sẽ được dùng để nhân tích vô hướng với nhau và dùng hàm kích hoạt softmax để thể hiện xác suất liên kết giữa hai người dùng. Xác suất này sẽ được so sánh với nhãn của nó để tính lỗi ( thông qua công thức Cross entropy) nhằm để cải thiện mô hình.

Phần kiến trúc mô hình, chúng tôi thiết kế như sau:

- o Encoder: 2 layer Graph Convolutional layer
- o Decode: Nhân tích vô hướng từng cặp theo các đặc trưng của node
- o Hàm lỗi sử dụng: BCEWithLogitsLoss (Hàm lỗi này là sự kết hợp của Sigmoid layer và hàm Binary Cross Entropy)

- Các phương pháp rút trích đặc trưng:

- o GCN: là một phương pháp có thể mở rộng cho việc học bán giám sát trên dữ liệu đồ thị. Phương pháp này dựa trên một biến thể hiệu quả của mạng nơ-ron tích chập mà hoạt động trực tiếp trên đồ thị. Mô hình này có tỉ lệ truyền tính theo số cạnh đồ thị và học các hidden layer representations dùng để trích xuất local graph structure và features of nodes.

- o GraphSAGE: là một framework tổng quát có tên là GraphSAGE (SAmple and aggreGatE) để embedding cho các node trên đồ thị lớn. GraphSAGE sử dụng thông tin đặc trưng của đỉnh để tạo ra nhúng đại diện cho các đỉnh trước đó chưa được quan sát. Thay vì huấn luyện từng embedding riêng lẻ cho từng đỉnh, GraphSAGE huấn luyện một tập hợp các hàm tổng hợp thông tin từ vùng lân cận của mỗi đỉnh.

- Hàm lỗi dự đoán

Công thức hàm Sigmoid dùng để tích xác xuất liên kết như sau

$$\sigma(x) = 1 / (1 + \exp(-x))$$

Trong đó:

- o  $x$  là giá trị đầu vào (có giá trị liên tục).
- o  $\exp$  là hàm mũ với cơ số  $e$  ( $2.71828\dots$ ).
- o  $\sigma(x)$  là giá trị đầu ra của hàm Sigmoid (nằm trong khoảng 0 đến 1).

Chúng tôi sử dụng hàm Cross entropy để tính lỗi dự đoán, với công thức như sau:

$$H(P, Q) = - \sum P(x) \log(Q(x))$$

Trong đó:

- $P(x)$  là xác suất thật của nhãn  $x$ .
- $Q(x)$  là xác suất dự đoán của nhãn  $x$ .
- $\Sigma$  là dấu tổng.
- Thuật toán tối ưu mô hình

Chúng tôi sử dụng thuật toán tối ưu Adam để cải thiện mức độ hiệu quả của mô hình, với công thức như sau:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

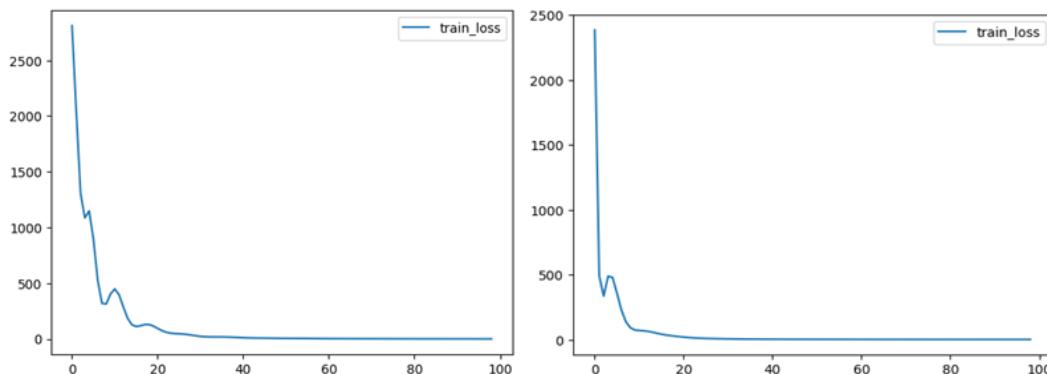
Trong đó:

- $\alpha$  (alpha) là tốc độ học (learning rate), quy định tốc độ cập nhật của các tham số.
- $\theta_t$  là giá trị hiện tại của các tham số mô hình.
- $g$  là gradient của hàm mất mát đối với các tham số.
- $m_t$  và  $v_t$  là các ước lượng gradient bậc nhất (first moment) và gradient bậc hai (second moment) tại thời điểm  $t$ .

- $\beta_1$  và  $\beta_2$  là hệ số giảm dần (decay rate) cho first moment và second moment tương ứng.
- $m_{\hat{m}}$  và  $v_{\hat{v}}$  là các ước lượng được điều chỉnh của first moment và second moment để khắc phục sự thiếu chuẩn xác ban đầu.
- $t$  là chỉ số bước tối ưu hóa (iteration).
- $\epsilon$  (epsilon) là một giá trị nhỏ được thêm vào trong mẫu số để tránh chia cho 0.

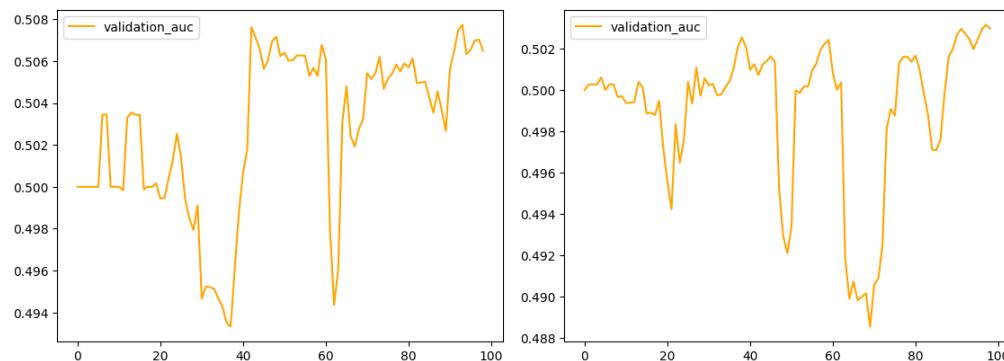
### 3.5.3.3. Kết quả thử nghiệm

Tham số huấn luyện: hidden\_channels = 128, 64; lr=0.01; epoch=100



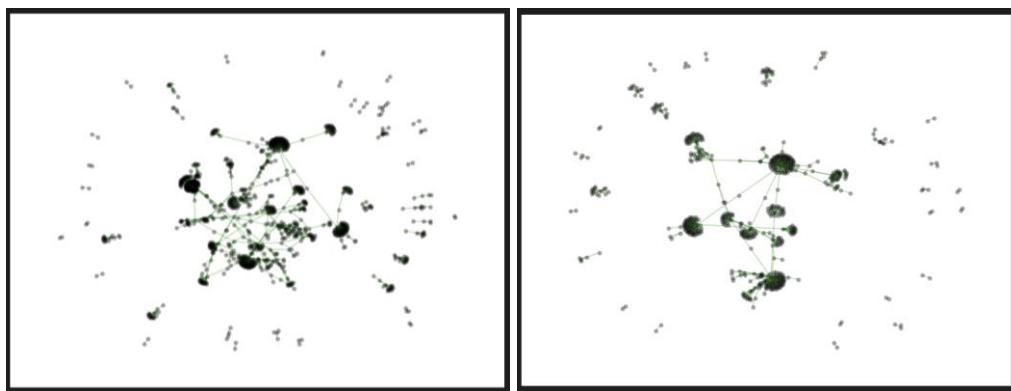
Đồ thị hàm lỗi khi huấn luyện trên mô hình GraphSAGE và GCN

**Nhận xét:** Từ đồ thị hàm lỗi cho thấy mô hình tối ưu rất nhanh trong khoảng 30 epoch đầu và ổn định không giảm đối với các epoch về sau.



Hình. Đồ thị kết quả AUC trên tập dữ liệu kiểm thử của mô hình GraphSAGE và GCN

**Nhận xét:** Độ chính xác của 2 mô hình giao động từ 49% đến 51%, mô hình không có khả năng dự đoán trên bộ dữ liệu. Có thể là do đặc trưng của thuộc tính không đủ (chỉ có 4 đặc trưng) hoặc dữ liệu quá thưa thớt nên mô hình không thể trích xuất tốt đặc trưng của từng member để dự đoán.



Đồ thị kết quả dự đoán liên kết trên tập dữ liệu kiểm thử của mô hình GraphSAGE và GCN

## CHƯƠNG 4. TỔNG KẾT

Qua đề tài thực hiện, nhóm đã thu nhận được rất nhiều kiến thức quý giá. Nhóm đã tìm hiểu về công nghệ dữ liệu lớn và Google Cloud Platform, đặc biệt là ưu nhược điểm của nền tảng này. Điều này giúp chúng ta có cái nhìn tổng quan về khả năng và giới hạn của Google Cloud khi làm việc với dữ liệu lớn trong môi trường mạng xã hội.

Nhóm cũng đã nghiên cứu thêm về triển khai cấu trúc dữ liệu lớn trong môi trường Google Cloud. Qua việc tìm hiểu về quá trình tiếp nhận dữ liệu (ingest), xử lý dữ liệu (process), nguồn dữ liệu (data source) và bổ sung thông tin (enrich) để xử lý và phân tích dữ liệu mạng xã hội một cách hiệu quả.

Nhóm còn tìm hiểu thêm về các loại cloud database như Neo4j Aura, MongoDB Atlas, Cloud Storage để lưu trữ dữ liệu đám mây. Và cuối cùng nhóm đã ứng dụng và thực hiện phân tích và xây dựng mô hình dự đoán liên kết người dùng trên Meetup sử dụng Google Cloud. Qua quá trình này, chúng tôi đã áp dụng các thành phần chính của Google Cloud và tiến hành thống kê dữ liệu, lưu trữ dữ liệu và phân tích dữ liệu mạng xã hội. Điều này giúp chúng tôi có cái nhìn rõ ràng về cách áp dụng các công nghệ và kỹ thuật của Google Cloud để tạo ra giá trị từ dữ liệu mạng xã hội.

Từ luận văn này, nhóm đã học được rất nhiều kiến thức quan trọng về mạng xã hội và cách sử dụng Google Cloud để xử lý dữ liệu lớn. Điều này sẽ đóng góp vào sự phát triển và áp dụng công nghệ trong lĩnh vực mạng xã hội, từ việc phân tích dữ liệu đến dự đoán và đưa ra quyết định thông minh. Nhóm hy vọng rằng luận văn này cung cấp một cơ sở vững chắc và đóng góp vào sự phát triển của mạng xã hội và công nghệ dữ liệu lớn.

## TÀI LIỆU THAM KHẢO

[1]	Google Cloud (2022). Use graphs for smarter AI with Neo4j and Google Cloud Vertex AI. Truy xuất từ: <a href="https://cloud.google.com/blog/products/ai-machine-learning/analyze-graph-data-on-google-cloud-with-neo4j-and-vertex-ai">https://cloud.google.com/blog/products/ai-machine-learning/analyze-graph-data-on-google-cloud-with-neo4j-and-vertex-ai</a>
[2]	Google Cloud (2023). Analyze your data. Truy xuất từ: <a href="https://cloud.google.com/architecture/framework/system-design/data-analytics">https://cloud.google.com/architecture/framework/system-design/data-analytics</a>
[3]	Google Cloud (2023). Application development resources. Truy xuất từ: <a href="https://cloud.google.com/architecture/application-development">https://cloud.google.com/architecture/application-development</a>
[4]	Google Cloud (2023). Overview of BigQuery storage. Truy xuất từ: <a href="https://cloud.google.com/bigquery/docs/storage_overview">https://cloud.google.com/bigquery/docs/storage_overview</a>
[5]	Google Cloud (2023). AI and machine learning resources. Truy xuất từ: <a href="https://cloud.google.com/architecture/ai-ml">https://cloud.google.com/architecture/ai-ml</a>

[6]	Itbrief (2023). Neo4j and Google Cloud extend partnership with new integration. Truy xuất từ: <a href="https://itbrief.com.au/story/neo4j-and-google-cloud-extend-partnership-with-new-integration">https://itbrief.com.au/story/neo4j-and-google-cloud-extend-partnership-with-new-integration</a>
[7]	Neo4j (2018). Google Cloud - Neo4j Causal Cluster Launch Demo. Truy xuất từ: <a href="https://www.youtube.com/watch?v=TomnIoeVGIg">https://www.youtube.com/watch?v=TomnIoeVGIg</a>
[8]	Kaggle (2023). Nashville Meetup Network. Truy xuất từ: <a href="https://www.kaggle.com/datasets/stkbailey/nashville-meetup?resource=download&amp;select=rsvps.csv">https://www.kaggle.com/datasets/stkbailey/nashville-meetup?resource=download&amp;select=rsvps.csv</a>
[9]	Prnewswire (2023). Neo4j and Google Cloud Extend Strategic Partnership With New Native Integration to Google Cloud's BigQuery Cloud Data Warehouse. Truy xuất từ: <a href="https://www.prnewswire.com/in/news-releases/neo4j-and-google-cloud-extend-strategic-partnership-with-new-native-integration-to-google-clouds-bigquery-cloud-data-warehouse-301784227.html">https://www.prnewswire.com/in/news-releases/neo4j-and-google-cloud-extend-strategic-partnership-with-new-native-integration-to-google-clouds-bigquery-cloud-data-warehouse-301784227.html</a>
[10]	Arxiv (2018). Link Prediction Based on Graph Neural Networks. Truy xuất từ: <a href="https://arxiv.org/pdf/1802.09691.pdf">https://arxiv.org/pdf/1802.09691.pdf</a>
[11]	Arxiv (2017). Modeling Relational Data with Graph Convolutional Networks. Truy xuất từ: <a href="https://arxiv.org/pdf/1703.06103.pdf">https://arxiv.org/pdf/1703.06103.pdf</a>

- |      |  |
|------|--|
| [12] | Arxiv (2017). SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS. Truy xuất từ:<br><a href="https://arxiv.org/pdf/1609.02907v4.pdf">https://arxiv.org/pdf/1609.02907v4.pdf</a> |
|------|--|