**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI**

**Giảng viên hướng dẫn: Phạm Thị Thu Thúy**

**Sinh viên thực hiện: Huỳnh Công Lợi**

**Mã số sinh viên: 62133105**

KHÁNH HOÀ, 2024

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN TỐT NGHIỆP**

**ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI**

Giảng viên hướng dẫn: Phạm Thị Thu Thúy

Sinh viên thực hiện: Huỳnh Công Lợi

Mã số sinh viên: 62133105

KHÁNH HOÀ, 2024

A screenshot of a white paper with text and numbers

Description automatically generatedA screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU THEO DÕI TIẾN ĐỘ VÀ ĐÁNH GIÁ ĐỒ ÁN TỐT NGHIỆP**

***(Dùng cho CBHD và nộp cùng báo cáo Đồ án của sinh viên)***

**Tên đề tài:** ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI

**Giảng viên hướng dẫn:** TS. Phạm Thị Thu Thúy

**Sinh viên được hướng dẫn:** Huỳnh Công Lợi **MSSV:** 62133105

**Khóa:** 62 **Ngành:** Công nghệ thông tin

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Lần KT** | **Ngày** | | **Nội dung** | | | **Nhận xét của GVHD** |
| 1 | 05/02 | | Xác nhận đề tài và giảng viên phụ trách hướng dẫn | | |  |
| 2 | 19/02 | | Trình bày các công việc dự kiến sẽ thực hiện | | |  |
| 3 | 26/02 | | Tìm hiểu cơ sở dữ liệu đồ thị NEO4J và các thức hoạt động của một ứng dụng mạng xã hội  Báo cáo tiến độ công việc cho giảng viên phụ trách hướng dẫn lần 1 | | |  |
| 4 | 14/03 | | Tìm hiểu về ngôn ngữ truy vấn Cypher và các công cụ hỗ trợ.  Báo cáo tiến độ công việc cho giảng viên phụ trách hướng dẫn lần 2 | | |  |
| 5 | 06/05 | | Tìm hiểu về các mô hình phân lớp và bài toán dự đoán liên kết  Thiết kế và cài đặt hệ thống | | |  |
| 6 | 11/05 | | Chỉnh sửa lại chương trình theo yêu cầu của giảng viên hướng dẫn lần 1 | | |  |
| 7 | 15/05 | | Chỉnh sửa lại chương trình theo yêu cầu của giảng viên hướng dẫn lần 2 | | |  |
| 8 | 17/05 | | Chỉnh sửa lại chương trình theo yêu cầu của giảng viên hướng dẫn lần 3 | | |  |
| 9 | 20/05 | | Báo cáo tiến độ công việc cho giảng viên phụ trách hướng dẫn lần 3 | | |  |
| 10 | 24/05 | | Chỉnh sửa báo cáo và cập nhật tiến độ công việc cho giảng viên phụ trách hướng dẫn lần 4 | | |  |
| 11 | 29/05 | | Chỉnh sửa báo cáo và cập nhật tiến độ công việc cho giảng viên phụ trách hướng dẫn lần 5 | | |  |
| **Kiểm tra giữa tiến độ của Trưởng BM** | | | | | | |
| Ngày kiểm tra:  ………………….. | | Đánh giá công việc hoàn thành: …% | | | Ký tên  ……………....... | |
| Được tiếp tục: | | Không tiếp tục: |
| 8 | |  | |  |  | |
| 9 | |  | |  |  | |
| 10 | |  | |  |  | |
| 11 | |  | |  |  | |
| 12 | |  | |  |  | |
| 13 | |  | |  |  | |
| 14 | |  | |  |  | |
| 15 | |  | |  |  | |

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

|  |  |  |
| --- | --- | --- |
| Điểm hình thức: ……/10 | Điểm nội dung: ……/10 | **Điểm tổng kết**: ……. /10 |

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: | Không được bảo vệ: |
|  | *Khánh Hòa, ngày……tháng……năm….* | | |
|  | **Giáo viên hướng dẫn**  *(Ký và ghi rõ họ tên)* | | |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN TỐT NGHIỆP**

**(Dùng cho CBHD và nộp cùng báo cáo ĐA/KLTN của sinh viên)**

**Tên đề tài:** ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI

**Chuyên ngành:** Công nghệ thông tin

**Họ và tên sinh viên:** Huỳnh Công Lợi **MSSV:** 62133105

**Giảng viên hướng dẫn:** TS. Phạm Thị Thu Thúy

**Cơ quan công tác:** Trường Đại học Nha Trang

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số (%)** | **Mô tả chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| **Xây dựng đề cương nghiên cứu** | **10** |  |  |  |  |  |
| **Tinh thần và thái độ làm việc** | **10** |  |  |  |  |  |
| **Kiến thức và kỹ năng làm việc** | **10** |  |  |  |  |  |
| **Nội dung và kết quả đạt được** | **40** |  |  |  |  |  |
| **Kỹ năng viết và trình bày báo cáo** | **30** |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: | Không được bảo vệ: |
|  | *Khánh Hòa, ngày……tháng……năm….* | | |
|  | **Cán bộ hướng dẫn**  *(Ký và ghi rõ họ tên)* | | |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN TỐT NGHIỆP**

**(Dùng cho thành viên Hội đồng bảo vệ ĐA/KLTN)**

**Tên đề tài:** ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI

**Chuyên ngành:** Công nghệ thông tin

**Họ và tên sinh viên:** Huỳnh Công Lợi **MSSV:** 62133105

**Họ tên thành viên HĐ:**

**Cơ quan công tác:**

**Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số (%)** | **Mô tả chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| **Xây dựng đề cương nghiên cứu** | **10** |  |  |  |  |  |
| **Tinh thần và thái độ làm việc** | **10** |  |  |  |  |  |
| **Kiến thức và kỹ năng làm việc** | **10** |  |  |  |  |  |
| **Nội dung và kết quả đạt được** | **40** |  |  |  |  |  |
| **Kỹ năng viết và trình bày báo cáo** | **30** |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung** (sau khi sinh viên hoàn thành ĐA/KLTN):

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: | Không được bảo vệ: |
|  | *Khánh Hòa, ngày……tháng……năm….* | | |
|  | **Cán bộ chấm điểm**  *(Ký và ghi rõ họ tên)* | | |

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

**PHIẾU CHẤM ĐIỂM ĐỒ ÁN TỐT NGHIỆP**

**(Dùng cho CBPB và nộp cùng báo cáo ĐA/KLTN của sinh viên)**

**Tên đề tài:** ỨNG DỤNG MÔ HÌNH PHÂN LỚP ĐỂ DỰ ĐOÁN LIÊN KẾT TRÊN MẠNG XÃ HỘI

**Chuyên ngành:** Công nghệ thông tin

**Họ và tên sinh viên:** Huỳnh Công Lợi **MSSV:** 62133105

**Người phản biện (học hàm, học vị, họ và tên):**

**Cơ quan công tác:**

1. **Phần đánh giá và cho điểm của người hướng dẫn (tính theo thang điểm 10)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Tiêu chí đánh giá** | **Trọng số (%)** | **Mô tả chất lượng** | | | | **Điểm** |
| **Giỏi** | **Khá** | **Đạt yêu cầu** | **Không đạt** |
| **9 - 10** | **7 - 8** | **5 - 6** | **< 5** |
| **Xây dựng đề cương nghiên cứu** | **10** |  |  |  |  |  |
| **Tinh thần và thái độ làm việc** | **10** |  |  |  |  |  |
| **Kiến thức và kỹ năng làm việc** | **10** |  |  |  |  |  |
| **Nội dung và kết quả đạt được** | **40** |  |  |  |  |  |
| **Kỹ năng viết và trình bày báo cáo** | **30** |  |  |  |  |  |
| **ĐIỂM TỔNG** | | | | | |  |

*Ghi chú: Điểm tổng làm tròn đến 1 số lẻ.*

**Nhận xét chung:**

|  |  |  |  |
| --- | --- | --- | --- |
| Đồng ý cho sinh viên: | | Được bảo vệ: | Không được bảo vệ: |
|  | *Khánh Hòa, ngày……tháng……năm….* | | |
|  | **Cán bộ phản biện**  *(Ký và ghi rõ họ tên)* | | |

**II. Phần nhận xét cụ thể (dựa theo phiếu chấm điểm và khung tiêu chí đánh giá theo Rubric)**

**II.1. Hình thức thuyết minh (tỉ trọng 30%)**

***\* Trình bày*** (*Rõ ràng, mạch lạc? Biểu bảng, hình vẽ trình bày rõ ràng, đúng quy cách?)*

***\* Bố cục và lập luận*** *(Bố cục hợp lý? Tỉ trọng giữa các phần? Cơ sở lập luận?)*

***\* Văn phong*** *(Gọn gàng, súc tích hay rườm rà, khó hiểu? Lỗi văn phạm và chính tả?)*

**II.2. Nội dung thuyết minh** (tỉ trọng 30%)

***\* Mục tiêu nghiên cứu*** *(Trình bày rõ ràng? Ý nghĩa khoa học và thực tiễn?*

*Tính khả thi?)*

***\* Tổng quan tài liệu*** *(Phân tích và đánh giá? Độ tin cậy và chất lượng nguồn tài liệu?)*

***\* Phương pháp nghiên cứu* (***Hiện đại?**Phù hợp với mục tiêu và nội dung nghiên cứu? Mô tả? Đánh giá và so sánh với các phương pháp khác?)*

**II.3. Kết quả nghiên cứu (**tỉ trọng 20%)

***\* Kết quả đạt được*** *(Độ**tin cậy? Tính sáng tạo? Giá trị khoa học và thực tiễn?)*

***\* Kết luận*** *(Đáp ứng mục tiêu nghiên cứu? Quan điểm của cá nhân?)*

**II.4. MỨC ĐỘ TRÍCH DẪN VÀ SAO CHÉP** (tỉ trọng 20%)

**\* Mức độ trích dẫn *(****Đúng quy định? Trung thực, đầy đủ, rõ ràng? Sắp xếp tài liệu tham khảo?)*

**\* Mức độ sao chép** *(Tỉ lệ sao chép? Hình thức sao chép?)*

# LỜI CẢM ƠN

Hành trình nghiên cứu đề tài "Ứng dụng mô hình phân lớp để dự đoán liên kết trên mạng xã hội" đã khép lại, để lại trong tôi những dấu ấn sâu sắc và lòng biết ơn vô bờ bến.

Trước tiên, tôi xin gửi lời tri ân sâu sắc đến Trường Đại học Nha Trang, nơi đã tạo điều kiện và môi trường học tập tuyệt vời để tôi có thể thực hiện nghiên cứu này. Nhờ sự hỗ trợ của nhà trường, tôi đã có cơ hội tiếp cận với các nguồn tài nguyên và kiến thức quý báu, góp phần không nhỏ vào sự thành công của đề tài.

Tiếp theo, tôi xin cảm ơn Khoa Công nghệ Thông tin, Trường Đại học Nha Trang, đặc biệt là cô giáo Phạm Thị Thu Thúy, Trưởng khoa. Sự tận tâm dìu dắt và hỗ trợ của cô trong suốt quá trình nghiên cứu đã giúp tôi vượt qua những khó khăn và hoàn thành đề tài một cách trọn vẹn nhất. Những lời khuyên quý giá và niềm tin tưởng của cô là nguồn động lực lớn lao đối với tôi.

Tôi cũng xin gửi lời tri ân đến gia đình thân yêu của mình. Nhờ sự động viên, khích lệ và niềm tin tưởng của gia đình, tôi đã có thêm động lực để vượt qua những thử thách và hoàn thành đề tài một cách tốt nhất.

Đặc biệt, tôi xin cảm ơn những người bạn thân yêu của mình. Sự ủng hộ và chia sẻ của các bạn đã giúp tôi vượt qua những giai đoạn khó khăn trong quá trình nghiên cứu.

Với đề tài này, tôi hy vọng có thể mang đến những đóng góp nhỏ bé cho cộng đồng khoa học và thực tiễn. Ứng dụng mô hình phân lớp để dự đoán liên kết trên mạng xã hội có tiềm năng mở ra nhiều hướng nghiên cứu mới trong lĩnh vực khoa học dữ liệu và trí tuệ nhân tạo, góp phần giải quyết các vấn đề thực tế trong đời sống xã hội.

Cuối cùng, xin trân trọng cảm ơn một lần nữa tất cả những ai đã đồng hành và hỗ trợ tôi trong suốt quá trình nghiên cứu. Kính chúc quý thầy cô sức khỏe, dồi dào nhiệt huyết và thành công trong sự nghiệp giảng dạy. Chúc các bạn sinh viên luôn học tập tốt và gặt hái được nhiều thành công trong tương lai.

Xin chân thành cảm ơn.

# LỜI CAM ĐOAN

Tôi xin cam đoan rằng đề tài "Ứng dụng mô hình phân lớp để dự đoán liên kết trên mạng xã hội" là sản phẩm nghiên cứu độc lập của bản thân, được thực hiện dựa trên quá trình nghiên cứu và phân tích cẩn thận.

Để hoàn thành đề tài, tôi đã sử dụng các nguồn tài liệu và thông tin hợp lệ, công khai từ các nguồn đáng tin cậy. Nguồn tài liệu bao gồm sách, báo, bài báo khoa học, trang web chính thức của các tổ chức uy tín.

Tôi xin khẳng định rằng tất cả các số liệu và kết quả trình bày trong đề tài đều là sự thật và được minh bạch hoàn toàn. Tôi đã thực hiện đầy đủ các quy trình nghiên cứu khoa học, đảm bảo tính chính xác và khách quan của dữ liệu.

Tôi cam đoan chịu hoàn toàn trách nhiệm về tính chính xác và trung thực của nội dung đề tài.

|  |
| --- |
| Nha Trang, ngày 31 tháng 05 năm 2024 |
| Sinh viên cam đoan |
| **Huỳnh Công Lợi** |

**MỤC LỤC**

[LỜI CẢM ƠN xiv](#_Toc168301049)

[LỜI CAM ĐOAN xv](#_Toc168301050)

[DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ CÁI VIẾT TẮT 3](#_Toc168301051)

[DANH MỤC CÁC BẢNG 4](#_Toc168301052)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ 5](#_Toc168301053)

[CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT 7](#_Toc168301054)

[1.1. Tổng quan đề tài 7](#_Toc168301055)

[1.1.1. Lý do chọn đề tài 7](#_Toc168301056)

[1.1.2. Mục tiêu và phạm vi nghiên cứu 7](#_Toc168301057)

[1.1.3. Tình hình nghiên cứu 8](#_Toc168301058)

[1.2. Cơ sở lý thuyết 9](#_Toc168301059)

[1.2.1. Cơ sở dữ liệu đồ thị 9](#_Toc168301060)

[1.2.2. Cơ sở dữ liệu đồ thị Neo4J – Neo4J Desktop 10](#_Toc168301061)

[1.2.3. Ngôn ngữ truy vấn Cypher 17](#_Toc168301062)

[1.2.4. Bài toán dự đoán liên kết 17](#_Toc168301063)

[1.2.4.1 Giới thiệu bài toán 17](#_Toc168301064)

[1.2.4.2 Ứng dụng 18](#_Toc168301065)

[1.2.4.3 Phương pháp dự đoán liên kết 18](#_Toc168301066)

[1.2.5. Mô hình phân lớp 19](#_Toc168301067)

[1.2.4.1 Mô hình phân lớp hồi quy (Logistic Regression) 19](#_Toc168301068)

[1.2.4.2 Mô hình phân lớp Naive Bayes 22](#_Toc168301069)

[1.2.4.3 Mô hình phân lớp cây quyết định (Decision Tree) 26](#_Toc168301070)

[1.2.6. Ứng dụng mô hình phân lớp trong bài toán dự đoán liên kết 29](#_Toc168301071)

[1.2.6.1 Xây dựng mô hình phân lớp 29](#_Toc168301072)

[1.2.6.2 Tập huấn luyện và tập kiểm tra 29](#_Toc168301073)

[1.2.6.3 Các metrics đánh giá mô hình 30](#_Toc168301074)

[CHƯƠNG 2. TRIỂN KHAI BÀI TOÁN DỰ ĐOÁN LIÊN KẾT TRÊN CƠ SỞ DỮ LIỆU ĐỒ THỊ NEO4J 32](#_Toc168301075)

[2.1. Thu thập bộ dữ liệu 32](#_Toc168301076)

[2.2. Cơ sở dữ liệu đồ thị 36](#_Toc168301077)

[2.2.1. Môi trường cài đặt cơ sở dữ liệu 36](#_Toc168301078)

[2.2.2. Tạo cơ sở dữ liệu 37](#_Toc168301079)

[2.2.3. Nạp dữ liệu vào cơ sở dữ liệu đồ thị Neo4J 38](#_Toc168301080)

[CHƯƠNG 3. CÀI ĐẶT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH 44](#_Toc168301081)

[3.1. Môi trường cài đặt và triển khai chương trình 44](#_Toc168301082)

[3.2. Tạo tập dữ liệu huấn luyện và kiểm tra 45](#_Toc168301083)

[3.3. Xây dựng các tính năng dự đoán liên kết 50](#_Toc168301084)

[3.4. Đánh giá mô hình 56](#_Toc168301085)

[3.5. Chương trình thực nghiệm 57](#_Toc168301086)

[3.6. Đánh giá kết quả của chương trình 63](#_Toc168301087)

[KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN 77](#_Toc168301088)

[DANH MỤC TÀI LIỆU THAM KHẢO 78](#_Toc168301089)

# DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ CÁI VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **Ký hiệu**  **(chữ viết tắt)** | **Chữ viết đầy đủ** | **Nghĩa tiếng việt** |
| ACID | Atomicity – Consistency – Isolation - Durability | Tính nguyên tử - Tính nhất quán - Tính cô lập - Tính bền vững |
| API | Application Programming Interface | Giao diện lập trình ứng dụng |
| DBMS | Database Management System | Hệ quản trị cơ sở dữ liệu |
| SQL | Structured Query Language | Ngôn ngữ truy vấn có cấu trúc |

# DANH MỤC CÁC BẢNG

[Bảng 2.1 Chú thích hình 2.18 43](#_Toc167803923)

[Bảng 3.1 Mô tả tiêu chí lọc để hiển thị trên bảng kết quả 51](#_Toc167803924)

[Bảng 3.2 Kết quả chạy của các mô hình phân lớp 63](#_Toc167803925)

[Bảng 3.3 Kết quả dự đoán của mô hình Logistic hồi quy (tỉ lệ 8:2) 65](#_Toc167803926)

[Bảng 3.4 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 8:2) 65](#_Toc167803927)

[Bảng 3.5 Kết quả dự đoán của mô hình Naive Bayes (tỉ lệ 8:2) 67](#_Toc167803928)

[Bảng 3.6 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 8:2) 67](#_Toc167803929)

[Bảng 3.7 Kết quả dự đoán của mô hình Cây quyết định (tỉ lệ 8:2) 70](#_Toc167803930)

[Bảng 3.8 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 8:2) 70](#_Toc167803931)

[Bảng 3.9 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 5:5) 70](#_Toc167803932)

[Bảng 3.10 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 5:5) 70](#_Toc167803933)

[Bảng 3.11 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 5:5) 71](#_Toc167803934)

[Bảng 3.12 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 2:8) 71](#_Toc167803935)

[Bảng 3.13 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 2:8) 71](#_Toc167803936)

[Bảng 3.14 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 2:8) 72](#_Toc167803937)

[Bảng 3.15 Kết quả hiệu suất của các mô hình phân lớp 72](#_Toc167803938)

[Bảng 3.16 Kết quả thời gian xử lý của các mô hình phân lớp 75](#_Toc167803939)

# DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 1.1 Ví dụ về cơ sở dữ liệu đồ thị mạng xã hội [5] 9](#_Toc168264394)

[Hình 1.2 Thành phần của một cơ sở dữ liệu đồ thị [5] 10](#_Toc168264395)

[Hình 1.3 Mô hình cơ sở dữ liệu đồ thị Neo4j [5] 11](#_Toc168264396)

[Hình 1.4 Các phiên bản và hệ điều hành 13](#_Toc168264397)

[Hình 1. 5 Form điền thông tin người dùng để tải ứng dụng 13](#_Toc168264398)

[Hình 1.6 Hướng dẫn sử dụng Active key để đăng nhập 14](#_Toc168264399)

[Hình 1.7 Lựa chọn quyền sử dụng của người dùng 14](#_Toc168264400)

[Hình 1.8 Lựa chọn nơi lưu trữ ứng dụng của người dùng 15](#_Toc168264401)

[Hình 1.9 Ảnh mô tả tiến trình cài đặt của ứng dụng 15](#_Toc168264402)

[Hình 1.10 Hướng dẫn người dùng đăng nhập bằng Software key 16](#_Toc168264403)

[Hình 1.11 Giao diện cơ bản của ứng dụng 16](#_Toc168264404)

[Hình 1.12 Cơ sở dữ liệu đồ thị của ví dụ Movie DBMS 17](#_Toc168264405)

[Hình 1.13 Các phương pháp dự đoán liên kết [5] 19](#_Toc168264406)

[Hình 2.1 Công cụ Graph API của Facebook 32](#_Toc168264407)

[Hình 2.2 Trang sử dụng các ứng dụng của Facebook dành cho developers 32](#_Toc168264408)

[Hình 2.3 Ảnh mô tả lựa chọn tạo ứng dụng trong Facebook 33](#_Toc168264409)

[Hình 2.4 Cấp quyền muốn sử dụng để khai thác thông tin người dùng 34](#_Toc168264410)

[Hình 2.5 Ảnh mô tả việc chọn các quyền muốn sử dụng trong ứng dụng 34](#_Toc168264411)

[Hình 2.6 Chọn các quyền muốn sử dụng trong ứng dụng 35](#_Toc168264412)

[Hình 2.7 Thực hiện truy vấn để lấy các thông tin của người dùng 35](#_Toc168264413)

[Hình 2.8 Tập dữ liệu của người dùng Facebook sau khi lấy được 36](#_Toc168264414)

[Hình 2.9 Giao diện của cơ sở dữ liệu đồ thị Neo4J 36](#_Toc168264415)

[Hình 2.10 Tạo cơ sở dữ liệu đồ thị mới “fb\_friend\_data” trong Project 37](#_Toc168264416)

[Hình 2.11 Khởi động cơ sở dữ liệu đồ thị “fb\_friend\_data” 37](#_Toc168264417)

[Hình 2.12 Giao diện sau khi đã khởi động thành công cơ sở dữ liệu 37](#_Toc168264418)

[Hình 2.13 Kết quả của truy vấn nạp người dùng Facebook vào Neo4j 38](#_Toc168264419)

[Hình 2.14 Kết quả tạo mối quan hệ giữa các người dùng với nhau 39](#_Toc168264420)

[Hình 2.15 Ảnh mô tả mối quan hệ cùng nơi ở thường trú 40](#_Toc168264421)

[Hình 2.16 Ảnh mô tả mối quan hệ cùng quê quán 41](#_Toc168264422)

[Hình 2.17 Ảnh mô tả mối quan hệ cùng độ tuổi 42](#_Toc168264423)

[Hình 2.18 Mô hình cơ sở dữ liệu đồ thị của người dùng Facebook trong Neo4j 42](#_Toc168264424)

[Hình 3.1 Giao diện hệ thống khuyến nghị kết bạn 58](#_Toc168264425)

[Hình 3.2 Giao diện lựa chọn người dùng và mô hình phân lớp 58](#_Toc168264426)

[Hình 3.3 Giao diện hiển thị kết quả sau khi nhấn nút “Tìm Bạn” 59](#_Toc168264427)

[Hình 3.4 Giao diện dự đoán của "Phương Phan" và "Logistic Regression" 59](#_Toc168264428)

[Hình 3.5 Giao diện dự đoán của "Nguyễn Thành Đạt" và "Naive Bayes" 60](#_Toc168264429)

[Hình 3.6 Giao diện dự đoán của "Kei Kyosaki" và "Decision Tree" 60](#_Toc168264430)

[Hình 3.7 Giao diện khi lưu kết quả của chương trình vào file Excel 61](#_Toc168264431)

[Hình 3.8 Giao diện báo lỗi khi người dùng chưa chạy chương trình "Tìm Bạn" 62](#_Toc168264432)

[Hình 3.9 Kết quả in các bảng dữ liệu của chương trình vào trong Excel 62](#_Toc168264433)

[Hình 3.10 Biểu đồ về độ chính xác của các mô hình phân lớp 73](#_Toc168264434)

[Hình 3.11 Biểu đồ về độ bao phủ của các mô hình phân lớp 73](#_Toc168264435)

[Hình 3.12 Biểu đồ về hệ số F1 của các mô hình phân lớp 74](#_Toc168264436)

[Hình 3.13 Biểu đồ mô tả thời gian huấn luyện của các mô hình phân lớp 75](#_Toc168264437)

[Hình 3.14 Biểu đồ mô tả thời gian dự đoán của các mô hình phân lớp 76](#_Toc168264438)

# CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI VÀ CƠ SỞ LÝ THUYẾT

## Tổng quan đề tài

### Lý do chọn đề tài

Mạng xã hội (MXH) là dịch vụ kết nối các thành viên có cùng sở thích trên internet, không giới hạn về không gian và thời gian. Hiện nay, có hàng trăm MXH như MySpace, Twitter, Pinterest, Facebook, trong đó Facebook phổ biến nhất với 64 triệu tài khoản ở Việt Nam, đứng thứ 7 toàn cầu. MXH đã trở thành một phần tất yếu trong cuộc sống của nhiều người, đặc biệt là giới trẻ và người về hưu.[1]

Trong khoa học máy tính, cơ sở dữ liệu đồ thị, như Neo4j, được sử dụng để biểu diễn các MXH và hệ thống liên quan như hệ thống gợi ý và quản lý mạng, nhờ khả năng lưu trữ và truy vấn dữ liệu hiệu quả.

Mục tiêu của đề tài này là áp dụng các mô hình phân loại như Logistic Regression, Naive Bayes và Decision Tree để dự đoán các liên kết trên MXH, đặc biệt là Facebook và mạng xã hội sẽ được minh họa bằng việc sử dụng cơ sở dữ liệu đồ thị Neo4j. Đồng thời, đánh giá độ tin cậy của mỗi tài khoản người dùng dựa trên hoạt động của họ trên MXH. Từ kết quả thu được, đề tài sẽ đề xuất các cải tiến cho các thuật toán phân loại nhằm nâng cao độ chính xác trong việc dự đoán các liên kết trên MXH.

### Mục tiêu và phạm vi nghiên cứu

- Tìm hiểu cách lưu trữ dữ liệu và triển khai mạng xã hội trên cơ sở dữ liệu đồ thị Neo4j.

- Ứng dụng các mô hình phân lớp để dự đoán liên kết trên cơ sở dữ liệu Neo4j

- Thiết kế hệ thống giới thiệu bạn bè dựa trên cơ sở dữ liệu đồ thị Neo4j và các mô hình phân sử dụng.

* Đối tượng nghiên cứu

- Khai phá dữ liệu trên mạng xã hội Facebook sử dụng công cụ hỗ trợ Graph API.

- Cơ sở dữ liệu đồ thị Neo4j và ngôn ngữ truy vấn Cypher.

- Bài toán dự đoán liên kết và các mô hình phân lớp được sử dụng để khuyến nghị kết bạn.

* Phạm vi nghiên cứu

- Cơ sở dữ liệu đồ thị người dùng từ mạng xã hội Facebook.

* Phương pháp nghiên cứu

- Nghiên cứu các tài liệu và thông tin liên quan đến chủ đề nghiên cứu như: Nghiên cứu về mạng xã hội; Nghiên cứu tài liệu về cơ sở dữ liệu đồ thi Neo4j, ngôn ngữ truy vấn Cypher; Nghiên cứu về bài toán dự đoán liên kết; Các mô hình phân lớp để giải quyết bài toán khuyến nghị.

- Thu thập dữ liệu từ mạng xã hội, cụ thể là Facebook.

- Xây dựng chương trình hệ thống khuyến nghị kết bạn.

- Thực nghiệm và phân tích, đánh giá kết quả thực nghiệm.

### Tình hình nghiên cứu

Bài toán dự đoán liên kết đang có nhiều ứng dụng quan trọng trong thực tế, đặc biệt là trong các lĩnh vực như mạng xã hội, hệ gợi ý, và mối quan hệ trong mạng sinh học. Các nghiên cứu gần đây đã đưa ra những phương pháp và kết quả đáng chú ý trong việc giải quyết bài toán này.

Cụ thể là bài báo "Link Prediction in Social Networks using Machine Learning" của Govinda K, Rajkumar Rajasekaran, V.S.R.P. Venkata Krishna Varun, Davya Vuyyuru và Battula Thirumaleshwari Devi [2] đã sử dụng các kỹ thuật máy học như học sâu và học máy để dự đoán liên kết trong mạng xã hội. Phương pháp này đã chứng minh được hiệu quả của việc áp dụng các kỹ thuật máy học trong việc dự đoán mối quan hệ giữa các cá nhân trên mạng xã hội.

Tiếp đến là nghiên cứu "Link Prediction for Social Network Analysis Using Random Forest and XG-Boost Algorithm" của Ved Prakash Chaubey, Ajay Sharma, Tushar Sharma; Shamneesh Sharma và Aman Kumar [3] đã áp dụng các thuật toán Random Forest và XG-Boost để dự đoán liên kết trong mạng xã hội. Công trình này đã chứng minh được hiệu quả của việc sử dụng các phương pháp máy học tập trong việc dự đoán mối quan hệ giữa các nút trong mạng.

Ngoài ra, bài báo "Link Prediction in Dynamic Social Networks Using Deep Learning" của Fateme Mohamady và Sina Dami [4] đã nghiên cứu về việc sử dụng học sâu để dự đoán liên kết trong mạng xã hội động. Phương pháp này đã mở ra những triển vọng mới trong việc hiểu và dự đoán sự phát triển của mối quan hệ trong mạng xã hội theo thời gian.

Tổng hợp lại, các nghiên cứu này đã đóng góp vào việc hiểu sâu hơn về cách thức hoạt động của mạng xã hội và cung cấp các phương pháp để dự đoán liên kết, từ đó có thể áp dụng vào thực tiễn như xây dựng các hệ thống khuyến nghị kết bạn hiệu quả hơn trên mạng xã hội.

## Cơ sở lý thuyết

### Cơ sở dữ liệu đồ thị

Cơ sở dữ liệu đồ thị thuộc vào loại cơ sở dữ liệu NoSQL, nơi dữ liệu được tổ chức và lưu trữ dưới dạng các nút, cạnh và thuộc tính. Khác với cơ sở dữ liệu quan hệ, cơ sở dữ liệu đồ thị tập trung vào mối quan hệ giữa các dữ liệu, giúp nó xử lý tốt các dữ liệu có mối quan hệ phức tạp và các truy vấn liên quan đến chúng.

A diagram of a movie character

Description automatically generated

Hình 1.1 Ví dụ về cơ sở dữ liệu đồ thị mạng xã hội [5]

Đồ thị đơn giản nhất có thể chỉ chứa một nút duy nhất, trong đó thông tin được lưu trữ dưới dạng các thuộc tính. Mỗi nút có thể bắt đầu với một số lượng nhỏ thuộc tính và mở rộng lên hàng triệu thuộc tính khác.

A diagram of a data flow

Description automatically generated

Hình 1.2 Thành phần của một cơ sở dữ liệu đồ thị [5]

Trong hình 2, chúng ta thấy một đồ thị với các nút và các mối quan hệ. Dữ liệu được lưu trữ không chỉ trong các nút mà còn trong các mối quan hệ. Các nút được tổ chức thành các mối quan hệ, trong khi mỗi nút cũng có các thuộc tính của riêng nó..

### Cơ sở dữ liệu đồ thị Neo4J – Neo4J Desktop

Neo4j đã trở thành một trong những hệ quản trị cơ sở dữ liệu đồ thị phổ biến nhất kể từ khi ra mắt vào năm 2007, và phiên bản 1.0 đã được công bố vào năm 2010. Được DB-Engines xếp hạng cao, Neo4j là lựa chọn hàng đầu cho nhiều ứng dụng với cơ sở dữ liệu đồ thị.

Neo4j tập trung vào việc lưu trữ dữ liệu dưới dạng một mạng lưới các mối quan hệ, trong đó các mục thông tin được biểu diễn dưới dạng các nút và các mối quan hệ giữa chúng. Các đặc điểm quan trọng của Neo4j bao gồm:

* Mô hình dữ liệu hướng đồ thị, dễ hiểu và trực quan.
* Khả năng mở rộng mà không ảnh hưởng đến hiệu suất truy vấn, tính toàn vẹn và độ tin cậy của dữ liệu.
* Tuân thủ nguyên tắc ACID.
* Hỗ trợ nhiều ngôn ngữ lập trình như Python, Java, Scala, và JavaScript.
* Sử dụng ngôn ngữ truy vấn Cypher dễ học và mạnh mẽ.
* Lập chỉ mục bằng Apache Lucene để tăng cường hiệu suất tìm kiếm và truy vấn.

A diagram of data flow

Description automatically generated

Hình 1.3 Mô hình cơ sở dữ liệu đồ thị Neo4j [5]

Các thành phần trong cơ sở dữ liệu Neo4J gồm:

* Nút (Node)
* Mối quan hệ (Relationship)
* Tập thuộc tính (Properties)
* Nhãn (Lable)
* NEO4J Desktop

Neo4j Desktop là một ứng dụng miễn phí và đa nền tảng cho nhà phát triển và quản trị viên cơ sở dữ liệu đồ thị Neo4j. Với giao diện thân thiện, người dùng có thể dễ dàng quản lý và chạy nhiều cơ sở dữ liệu đồ thị cùng một lúc. Điều này bao gồm cài đặt, quản lý, và thực hiện các hoạt động quản trị trên cơ sở dữ liệu, cũng như tạo và chạy các ứng dụng Neo4j.

Các tính năng chính của Neo4j Desktop bao gồm:

* Quản lý nhiều cơ sở dữ liệu đồ thị đồng thời.
* Giao diện người dùng đơn giản và thân thiện.
* Cài đặt và cấu hình cơ sở dữ liệu một cách dễ dàng.
* Cung cấp công cụ trực quan hóa và phân tích dữ liệu.
* Hỗ trợ cho các plugin của Neo4j để mở rộng chức năng của ứng dụng.

Bên cạnh đó, Neo4j Desktop cũng cho phép tạo dự án, tổ chức các thực thể và mối quan hệ trong cơ sở dữ liệu, và quản lý quyền truy cập cũng như xác định hạn chế và ràng buộc trên dữ liệu. DBMS của Neo4j được xây dựng trên kiến trúc lưu trữ đồ thị, giúp tối ưu hóa việc truy vấn và thao tác trên dữ liệu đồ thị.

* Cài đặt NEO4J Desktop

Để cài đặt NEO4J Desktop ta có thể truy cập trang web:

***https://neo4j.com/download-center/#desktop***

Sau đó lựa chọn phiên bản phù hợp để tải.

A screenshot of a computer

Description automatically generated

Hình 1.4 Các phiên bản và hệ điều hành

A screenshot of a form

Description automatically generatedSau đó chúng ta cần phải điền các thông tin cần thiết để có thể tải.

Hình 1. 5 Form điền thông tin người dùng để tải ứng dụng

Để kích hoạt ứng dụng có thể sử dụng “Active key” hoặc bạn cũng có thể tạo khóa từ bên trong ứng dụng bằng cách điền vào biểu mẫu ở bên phải màn hình ứng dụng.

Graphical user interface, text, application

Description automatically generated

Hình 1.6 Hướng dẫn sử dụng Active key để đăng nhập

Sau khi quá trình tải và chạy thì bạn cần phải thao tác chọn lựa một vài thông tin cần thiết như hình ảnh

A screenshot of a computer setup

Description automatically generated

Hình 1.7 Lựa chọn quyền sử dụng của người dùng

A screenshot of a computer

Description automatically generated

Hình 1.8 Lựa chọn nơi lưu trữ ứng dụng của người dùng

Graphical user interface, text, application

Description automatically generated

Hình 1.9 Ảnh mô tả tiến trình cài đặt của ứng dụng

Sau khi cài đặt xong, người dùng sẽ tiến hành khởi động ứng dụng và ở màn hình sẽ hiển thị nơi để người dùng lựa chọn đăng ký người dùng bằng cách điền form thông tin hoặc đăng nhập bằng Software key mà ta đã lấy được ở hình 1.6.

A screenshot of a computer

Description automatically generated with medium confidence

Hình 1.10 Hướng dẫn người dùng đăng nhập bằng Software key

Sau khi đăng nhập thành công, giao diện sẽ hiển thị như trong hình minh họa. Người dùng sẽ được đưa vào một dự án mẫu có tên là "Example Project" ngay từ ban đầu. Trong dự án này, có một phần "Movie DBMS" cung cấp hướng dẫn về các mệnh đề cơ bản trong ngôn ngữ truy vấn Cypher như MATCH, CREATE và nhiều hơn nữa.

Graphical user interface, text, application, email

Description automatically generated

Hình 1.11 Giao diện cơ bản của ứng dụng

A screenshot of a computer

Description automatically generated

Hình 1.12 Cơ sở dữ liệu đồ thị của ví dụ Movie DBMS

### Ngôn ngữ truy vấn Cypher

Cypher, ngôn ngữ truy vấn dành cho Neo4j, tương tự như SQL về cách cấu trúc các truy vấn. Được xây dựng từ nhiều mệnh đề khác nhau, Cypher được thiết kế để đơn giản nhưng mạnh mẽ, giúp người dùng dễ dàng làm việc với Neo4j.Các từ khóa quan trọng trong Cypher bao gồm:

* Create: Tạo nút và mối quan hệ.
* Match: Biểu thức truy vấn.
* Return: Trả về kết quả.
* Where: Điều kiện lọc.
* Delete: Xóa nút, mối quan hệ và thuộc tính.
* Set: Đặt giá trị cho thuộc tính.
* With: Chia nhỏ câu truy vấn thành các phần.

### Bài toán dự đoán liên kết

#### **Giới thiệu bài toán**

Bài toán dự đoán liên kết là một nhiệm vụ quan trọng trong lĩnh vực khai phá tri thức đồ thị, với mục tiêu dự đoán sự tồn tại của một liên kết tiềm ẩn giữa hai thực thể trong một đồ thị. Nói cách khác, bài toán này nhằm xác định xem hai thực thể có nên được kết nối với nhau hay không, và loại liên kết nào là phù hợp nhất.

Bài toán dự đoán liên kết có nhiều ứng dụng thực tế trong các lĩnh vực khác nhau như:

* Khuyến nghị hệ thống: Dự đoán các sản phẩm, phim ảnh, âm nhạc mà người dùng có thể quan tâm.
* Mạng xã hội: Gợi ý kết bạn, theo dõi hoặc tạo nhóm cho người dùng.
* Phân tích sinh học: Dự đoán tương tác protein-protein hoặc protein-DNA.
* Tìm kiếm thông tin: Xác định các tài liệu có liên quan đến một chủ đề nhất định.

#### **1.2.4.2 Ứng dụng**

Có nhiều phương pháp khác nhau để giải quyết bài toán dự đoán liên kết, có thể được phân thành hai nhóm chính:

* Phương pháp dựa trên đặc trưng: Các phương pháp này sử dụng các đặc trưng của các thực thể và các liên kết hiện có trong đồ thị để dự đoán sự tồn tại của các liên kết mới.
* Phương pháp học máy: Các phương pháp này sử dụng các thuật toán học máy để học hỏi từ dữ liệu và đưa ra dự đoán cho các liên kết mới.

#### **1.2.4.3 Phương pháp dự đoán liên kết**

Có nhiều kỹ thuật dự đoán liên kết khác nhau. Hiện nay có 3 phương pháp dự đoán liên kết đại diện như sau: phương pháp đầu tiên dựa trên học máy (machine learning) để tạo mô hình phân loại nhị phân. Thứ hai, phương pháp tiếp cận dựa trên mô hình cấu trúc topo. Cuối cùng là tiếp cận xác xuất theo mô hình Markov, Bayes, Random

A diagram of a model

Description automatically generated

Hình 1.13 Các phương pháp dự đoán liên kết [5]

### Mô hình phân lớp

#### **1.2.4.1 Mô hình phân lớp hồi quy (Logistic Regression)**

Logistic Regression (hay còn được gọi là hồi quy logitsic) là một mô hình phân loại được sử dụng để dự đoán xác suất của một biến phụ thuộc nhị phân dựa trên các biến độc lập.

Mô hình hồi quy logistic dựa trên ý tưởng của hồi quy tuyến tính, nhưng sử dụng một hàm logistic để giới hạn kết quả trong khoảng từ 0 đến 1, phản ánh xác suất dự đoán của lớp positive .

Trong hồi qui tuyến tính chúng ta dựạ vào một hàm hồi qui giả thuyết

để dự báo biến mục tiêu liên tục y. Vì giá trị của y có thể vượt ngoài khoảng [0, 1] nên trong hồi qui Logistic cần một hàm số có tác dụng chiếu giá trị dự báo lên không gian xác suất nằm trong khoảng [0, 1] và đồng thời tạo ra tính phi tuyến cho phương trình hồi qui nhằm giúp nó có đường biên phân chia giữa hai nhóm tốt hơn. Đó chính là hàm Sigmoid hoặc hàm Logistic. [6]

Đầu ra dự đoán của logistic regression thường được viết chung dưới dạng:

[6]

Trong đó

* : là tích vô hướng được sử dụng trực tiếp để dự đoán output
* : là hàm sigmoid được tính bằng công thức

[6]

Với 𝑧=𝛽0+𝛽1𝑥1+𝛽2𝑥2+...+𝛽𝑛𝑥𝑛 là một tổ hợp tuyến tính của các đặc trưng (𝑥1,𝑥2,...,𝑥𝑛), và 𝛽0,𝛽1,...,𝛽𝑛​ là các tham số của mô hình

Mô hình hồi quy logistic phân lớp (logistic regression for classification) được sử dụng để dự đoán xác suất của một trong hai lớp (binary classification).

Dưới đây là các bước hoạt động của mô hình này:

1. **Thu thập và tiền xử lý dữ liệu**:
   * **Thu thập dữ liệu**: Bước đầu tiên là thu thập dữ liệu chứa các đặc trưng (features) và nhãn (labels) để đào tạo mô hình.
   * **Tiền xử lý dữ liệu**: Làm sạch dữ liệu, xử lý các giá trị bị thiếu, mã hóa các biến phân loại, chuẩn hóa hoặc chuẩn hóa các đặc trưng nếu cần.
2. **Khởi tạo tham số**:
   * **Khởi tạo trọng số (weights) và hệ số chặn (bias)**: Ban đầu, các trọng số và hệ số chặn được khởi tạo bằng các giá trị nhỏ, thường là các giá trị ngẫu nhiên hoặc bằng 0.
3. **Tính toán hàm số tuyến tính**:
   * **Hàm số tuyến tính**: Tính giá trị tuyến tính bằng cách nhân trọng số với các đặc trưng và cộng với hệ số chặn.
   * Công thức:

[6]

trong đó 𝑧 là giá trị tuyến tính, 𝑤 là trọng số, 𝑥 là các đặc trưng, và 𝑏 là hệ số chặn.

1. **Áp dụng hàm sigmoid**:
   * **Hàm sigmoid**: Áp dụng hàm sigmoid lên giá trị 𝑧 để chuyển đổi nó thành xác suất.
   * Công thức của hàm sigmoid:

[6]

Kết quả là một giá trị giữa 0 và 1, đại diện cho xác suất của lớp dương (positive class).

1. **Tính toán hàm mất mát (loss function)**:
   * **Hàm mất mát log (log-loss)**: Đo lường sự khác biệt giữa giá trị dự đoán và giá trị thực tế.
   * Công thức:

[6]

trong đó 𝑦 là nhãn thực tế và là xác suất dự đoán.

1. **Cập nhật tham số bằng cách tối ưu hóa (optimization)**:
   * **Gradient Descent**: Sử dụng phương pháp gradient descent để cập nhật trọng số và hệ số chặn nhằm giảm thiểu hàm mất mát. Công thức cập nhật:
     + [7]
     + [7]

Trong đó, 𝛼 là tốc độ học (learning rate), và ​ là đạo hàm của hàm mất mát theo trọng số và hệ số chặn.

* + **Lặp lại**: Quá trình tính toán hàm số tuyến tính, áp dụng hàm sigmoid, tính toán hàm mất mát, và cập nhật tham số được lặp lại nhiều lần qua nhiều epoch (chu kỳ) cho đến khi hàm mất mát hội tụ hoặc đạt đến một ngưỡng xác định trước.

1. **Dự đoán và phân loại**:
   * **Dự đoán**: Sau khi mô hình đã được huấn luyện, sử dụng trọng số và hệ số chặn đã học để tính toán xác suất cho các mẫu mới.
   * **Phân loại**: Áp dụng một ngưỡng (threshold) để quyết định nhãn của mẫu. Ví dụ, nếu xác suất lớn hơn 0.5, phân loại mẫu vào lớp dương; nếu nhỏ hơn hoặc bằng 0.5, phân loại mẫu vào lớp âm.
2. **Đánh giá mô hình**:
   * **Đánh giá**: Sử dụng các chỉ số đánh giá như độ chính xác của từng lớp (precision), độ nhạy (recall), F1-score để đánh giá hiệu suất của mô hình trên tập kiểm tra.

#### **1.2.4.2 Mô hình phân lớp Naive Bayes**

Naive Bayes dựa trên định lý Bayes với một giả định độc lập giữa các dự đoán. Nói một cách đơn giản, một phân loại Naive Bayes giả định rằng sự hiện diện của một tính năng cụ thể trong class không liên quan đến sự hiện hiện của tính năng khác. Vì mô hình Naive Bayes sử dụng xác suất để đưa ra dự đoán nên nó tính xác suất của một mẫu thuộc vào mỗi lớp dựa trên các đặc trưng của mẫu đó , sau đó chọn lớp có xác suất cao nhất làm dự đoán.

Định lý Bayes cung cấp một cách để tính toán xác suất P(c|x) từ P(c), P(x) và P(x|c). Hãy nhìn công thức dưới.

Trong đó:

* 𝑃(𝑦∣𝑋) là xác suất có điều kiện của lớp 𝑦 dựa trên các đặc trưng 𝑋.
* 𝑃(𝑋∣𝑦) là xác suất của các đặc trưng 𝑋 dựa trên lớp 𝑦.
* 𝑃(𝑦) là xác suất tiên nghiệm của lớp 𝑦.
* 𝑃(𝑋) là xác suất của các đặc trưng 𝑋 (có thể được bỏ qua trong nhiều trường hợp vì nó không thay đổi theo lớp).

Việc tính toán 𝑝(𝑥𝑖|𝑐) phụ thuộc vào loại dữ liệu. [Có ba loại được sử dụng phổ biến](http://scikit-learn.org/dev/modules/classes.html#module-sklearn.naive_bayes) là: Gaussian Naive Bayes, Multinomial Naive Bayes, và Bernoulli Naive .

* Phân loại các biến thể của Naive Bayes

a. **Multinomial Naive Bayes (Naive Bayes đa thức)**

* Loại dữ liệu đầu vào: Dữ liệu rời rạc, thường là các tần suất xuất hiện của các từ hoặc đặc trưng rời rạc khác.
* Ứng dụng: Thường được sử dụng trong phân loại văn bản, phân loại tài liệu, phân loại email dựa trên từ vựng, hoặc bất kỳ tác vụ nào mà dữ liệu có thể được biểu diễn dưới dạng vector đặc trưng rời rạc.
* Công thức tính: Sử dụng phân phối đa thức để ước lượng xác suất. Xác suất của một mẫu x thuộc lớp y được tính bằng:

[7]

Trong đó:

* + là xác suất xuất hiện của đặc trưng thứ 𝑖 trong mẫu 𝑥 khi biết mẫu thuộc lớp 𝑦.
  + là là số lượng đặc trưng trong mẫu.

**b. Gaussian Naive Bayes**

* Loại dữ liệu đầu vào: Dữ liệu liên tục, giả định là tuân theo phân phối Gaussian (phân phối chuẩn).
* Ứng dụng: Thích hợp cho các đặc trưng có giá trị liên tục như chiều cao, cân nặng, nhiệt độ, độ dài, v.v.
* Công thức tính: Ước lượng xác suất sử dụng phân phối Gaussian. Xác suất của một mẫu x thuộc lớp y được tính bằng:

[7]

Trong đó:

* + là giá trị trung bình của các đặc trưng trong lớp y
  + là phương sai của các đặc trưng trong lớp y

**c. Bernoulli Naive Bayes**

* Loại dữ liệu đầu vào: Dữ liệu nhị phân, tức là mỗi đặc trưng chỉ có hai giá trị có thể (0 hoặc 1).
* Ứng dụng: Thường được sử dụng trong phân loại dữ liệu có tính nhị phân như phân loại văn bản thành hai nhóm (chứa từ khóa cụ thể hoặc không chứa từ khóa đó), phân loại email là spam hoặc không spam dựa trên sự xuất hiện của từ khóa, v.v.
* Công thức tính: Sử dụng phân phối Bernoulli để ước lượng xác suất. Xác suất của một mẫu x thuộc lớp y được tính bằng:

[7]

Trong đó:

* + là xác suất xuất hiện của đặc trưng thứ 𝑖 trong mẫu 𝑥 khi biết mẫu thuộc lớp 𝑦.
  + là giá trị của đặc trưng thứ 𝑖 (0 hoặc 1).
  + 𝑛 là số lượng đặc trưng trong mẫu.
* Gaussian Naive Bayes là một biến thể của mô hình phân lớp Naive Bayes, giả định rằng các đặc trưng (features) tuân theo phân phối chuẩn (Gaussian distribution). Đây là các bước hoạt động của mô hình Gaussian Naive Bayes:

1. **Thu thập và tiền xử lý dữ liệu**:
   * **Thu thập dữ liệu**: Thu thập một tập dữ liệu chứa các đặc trưng và nhãn để huấn luyện mô hình.
   * **Tiền xử lý dữ liệu**: Làm sạch dữ liệu, xử lý các giá trị thiếu, mã hóa các biến phân loại, và chuẩn hóa đặc trưng nếu cần.
2. **Phân chia dữ liệu theo lớp**:
   * **Phân loại theo nhãn**: Chia tập dữ liệu thành các nhóm dựa trên các nhãn lớp khác nhau.
3. **Tính toán các tham số thống kê**:
   * **Tính toán trung bình và phương sai**: Đối với mỗi đặc trưng và mỗi lớp, tính toán trung bình (𝜇) và phương sai () của đặc trưng đó.
     + [7]
     + [7]

Trong đó, ​ và ​ là trung bình và phương sai của đặc trưng 𝑗trong lớp 𝑐, ​ là số mẫu trong lớp 𝑐, và ​ là giá trị của đặc trưng 𝑗 cho mẫu 𝑖 trong lớp 𝑐.

1. **Xác suất tiên nghiệm (Prior probability)**:
   * **Tính toán xác suất tiên nghiệm**: Tính toán xác suất tiên nghiệm của mỗi lớp dựa trên tần suất xuất hiện của các lớp trong tập dữ liệu huấn luyện.
     + [7]

Trong đó, 𝑃(𝑐)là xác suất tiên nghiệm của lớp 𝑐, ​ là số mẫu trong lớp 𝑐, và 𝑁 là tổng số mẫu.

1. **Xác suất có điều kiện (Conditional probability)**:
   * **Tính toán xác suất có điều kiện**: Giả định các đặc trưng tuân theo phân phối chuẩn, tính toán xác suất có điều kiện của mỗi đặc trưng 𝑗 cho một giá trị cụ thể 𝑥𝑗trong lớp 𝑐 bằng cách sử dụng hàm mật độ xác suất của phân phối chuẩn:

[7]

1. **Tính toán xác suất hậu nghiệm (Posterior probability)**:
   * **Áp dụng định lý Bayes**: Sử dụng định lý Bayes để tính toán xác suất hậu nghiệm cho mỗi lớp 𝑐*c*:

[7]

Trong đó, là vector các đặc trưng của mẫu.

1. **Dự đoán**:
   * **Dự đoán lớp**: Chọn lớp có xác suất hậu nghiệm lớn nhất làm nhãn dự đoán cho mẫu:

[7]

1. **Đánh giá mô hình**:
   * **Đánh giá**: Sử dụng các chỉ số như độ chính xác của từng lớp (precision), độ nhạy (recall), F1-score để đánh giá hiệu suất của mô hình trên tập kiểm tra.

#### **1.2.4.3 Mô hình phân lớp cây quyết định (Decision Tree)**

Cây Quyết Định là một phương pháp học máy dựa trên việc tạo ra một cây quyết định có cấu trúc phân cấp để học từ dữ liệu. Cây quyết định phân loại dữ liệu bằng cách đặt các câu hỏi và dựa vào câu trả lời để đưa ra quyết định.

Cây Quyết Định tạo ra các quy tắc quyết định dựa trên dữ liệu đào tạo. Mỗi nút trên cây đại diện cho một thuộc tính (hoặc điều kiện) và mỗi nhánh đại diện cho một giá trị của thuộc tính đó. Các quy tắc được học từ dữ liệu và được sử dụng để phân loại các mẫu mới.

* Dưới đây là các bước hoạt động của mô hình cây quyết định:

1. **Thu thập và tiền xử lý dữ liệu**:
   * **Thu thập dữ liệu**: Thu thập một tập dữ liệu bao gồm các đặc trưng và nhãn để huấn luyện mô hình.
   * **Tiền xử lý dữ liệu**: Làm sạch dữ liệu, xử lý các giá trị thiếu và mã hóa các biến phân loại nếu cần thiết.
2. **Chọn đặc trưng phân chia**:
   * **Lựa chọn đặc trưng**: Tại mỗi nút của cây, chọn một đặc trưng để phân chia dữ liệu. Quyết định này dựa trên một tiêu chí chọn lựa như thông tin thu được (Information Gain), tỷ lệ thông tin (Information Gain Ratio), hay chỉ số Gini (Gini Index).
3. **Phân chia dữ liệu**:
   * **Phân chia dữ liệu**: Dữ liệu tại mỗi nút được chia thành các tập con dựa trên giá trị của đặc trưng đã chọn. Mỗi tập con sẽ trở thành một nhánh (branch) của nút đó.
4. **Đánh giá tiêu chí phân chia**:
   * **Tiêu chí thông tin thu được**: Đo lường sự giảm bớt độ hỗn loạn (entropy) do phân chia dữ liệu theo đặc trưng đã chọn.
     + Entropy:

[8]

* + - Information Gain:

[8]

Trong đó:

Tính toán hệ số Entropy của biến mục tiêu S có N phần tử với phần tử thuộc lớp c cho trước:

[8]

Tính hàm số Entropy tại mỗi thuộc tính: với thuộc tính x, các điểm dữ liệu trong S được chia ra K child node ,, …, với số điểm trong mỗi child node lần lượt là , ,…, , ta có:

[8]

* + **Chỉ số Gini**: Đo lường mức độ thuần nhất của dữ liệu.

[8]

1. **Xây dựng cây quyết định**:
   * **Tạo nút gốc (root)**: Bắt đầu với toàn bộ dữ liệu và chọn đặc trưng đầu tiên để phân chia.
   * **Phân nhánh**: Tạo các nhánh cho mỗi giá trị của đặc trưng đã chọn và tiếp tục phân chia dữ liệu ở mỗi nhánh con.
   * **Đệ quy**: Lặp lại quá trình chọn đặc trưng và phân chia cho đến khi một trong các điều kiện dừng được thỏa mãn:
     + Tất cả các mẫu trong tập con thuộc về một lớp duy nhất.
     + Không còn đặc trưng nào để phân chia.
     + Độ sâu tối đa của cây đã đạt được.
     + Số lượng mẫu trong tập con nhỏ hơn một ngưỡng định trước.
2. **Gán nhãn cho lá**:
   * **Nhãn lớp**: Khi không thể phân chia thêm, gán nhãn lớp cho mỗi nút lá dựa trên lớp chiếm đa số trong tập con của nút đó.
3. **Dự đoán**:
   * **Đi qua cây**: Để dự đoán nhãn cho một mẫu mới, bắt đầu từ nút gốc và đi qua cây theo các đặc trưng của mẫu đó cho đến khi đến một nút lá. Nhãn của nút lá là nhãn dự đoán.
4. **Đánh giá mô hình**:
   * **Đánh giá**: Sử dụng các chỉ số như độ chính xác của từng lớp (precision), độ nhạy (recall), F1-score để đánh giá hiệu suất của mô hình trên tập kiểm tra.

### Ứng dụng mô hình phân lớp trong bài toán dự đoán liên kết

#### **1.2.6.1 Xây dựng mô hình phân lớp**

Với ba mô hình phân loại được đề cập trước (Logistic Regression, Naive Bayes, Decision Tree), quy trình xây dựng mô hình chung bao gồm các bước sau:

1. Chuẩn bị dữ liệu:

* Tiền xử lý dữ liệu:
  + Chuyển đổi dữ liệu đồ thị sang dạng phù hợp cho mô hình học máy.
  + Xử lý các giá trị thiếu và nhiễu trong dữ liệu.
  + Mã hóa các biến phân loại (nếu có).
* Chia tập dữ liệu:
  + Chia tập dữ liệu thành tập huấn luyện và tập kiểm tra.
  + Tập huấn luyện được sử dụng để huấn luyện mô hình, trong khi tập kiểm tra được sử dụng để đánh giá hiệu suất của mô hình.

1. Lựa chọn mô hình:

* Lựa chọn mô hình phân loại phù hợp dựa trên đặc điểm của bài toán và dữ liệu.
* Có thể thử nghiệm với các mô hình khác nhau để tìm ra mô hình cho kết quả tốt nhất.

1. Huấn luyện mô hình:

* Sử dụng tập huấn luyện để huấn luyện mô hình phân loại đã chọn.
* Điều chỉnh các tham số của mô hình để tối ưu hóa hiệu suất.

1. Đánh giá mô hình:

* Sử dụng tập kiểm tra để đánh giá hiệu suất của mô hình trên dữ liệu chưa nhìn thấy.
* Sử dụng các metrics đánh giá phù hợp như độ chính xác, độ bao phủ, hệ số F1 v.v.

#### **1.2.6.2 Tập huấn luyện và tập kiểm tra**

Tập huấn luyện:

* Tập huấn luyện đóng vai trò quan trọng trong việc giúp mô hình học cách phân biệt giữa các cặp thực thể có liên kết và không có liên kết.
* Cần đảm bảo tập huấn luyện có đủ số lượng mẫu và được phân bố đều giữa các lớp (có liên kết và không có liên kết).
* Chất lượng của tập huấn luyện ảnh hưởng trực tiếp đến hiệu suất của mô hình.

Tập kiểm tra:

* Tập kiểm tra được sử dụng để đánh giá hiệu suất của mô hình trên dữ liệu chưa nhìn thấy.
* Tập kiểm tra cần được lựa chọn một cách ngẫu nhiên từ dữ liệu thực tế và không được sử dụng trong quá trình huấn luyện mô hình.
* Kích thước của tập kiểm tra nên đủ lớn để đảm bảo tính chính xác của đánh giá.

#### **1.2.6.3 Các metrics đánh giá mô hình**

Để đánh giá hiệu suất của mô hình dự đoán liên kết, có thể sử dụng một số metrics sau:

Trong bài toán dự đoán liên kết để đề xuất kết bạn ta có 2 lớp: lớp có liên kết được chuẩn đoán Positive và lớp không có liên kết được chuẩn đoán là Negative:

* TP (True Positive): Số lượng dự đoán chính xác. Là khi mô hình dự đoán đúng một người có liên kết.
* TN (True Negative): Số lương dự đoán chính xác một cách gián tiếp. Là khi mô hình dự đoán đúng một người không có liên kết, tức là việc không chọn trường hợp có liên kết là chính xác.
* FP (False Positive - Type 1 Error): Số lượng các dự đoán sai lệch. Là khi mô hình dự đoán một người có liên kết và người đó hoàn toàn không có liên kết.
* FN (False Negative - Type 2 Error): Số lượng các dự đoán sai lệch một cách gián tiếp. Là khi mô hình dự đoán một người không có liên kết nhưng người đó có liên kết, tức là việc không chọn trường hợp có liên kết là sai.

Từ các chỉ số này, ta có thể đánh giá mức độ tin cậy của một mô hình:

* Độ chính xác (Precision): Tỷ lệ các dự đoán có liên kết được mô hình dự đoán chính xác là có liên kết.

[9]

* Độ bao phủ (Recall): Tỷ lệ các cặp thực thể có liên kết thực sự được mô hình dự đoán là có liên kết.

[9]

* Hệ số F1 (F1-score) : Trung bình điều hòa giữa độ chính xác và độ bao phủ.

[9]

Lựa chọn metrics phù hợp phụ thuộc vào mục tiêu cụ thể của bài toán.

Ví dụ:

Nếu mục tiêu là tối ưu hóa số lượng dự đoán chính xác, độ chính xác là metrics phù hợp.

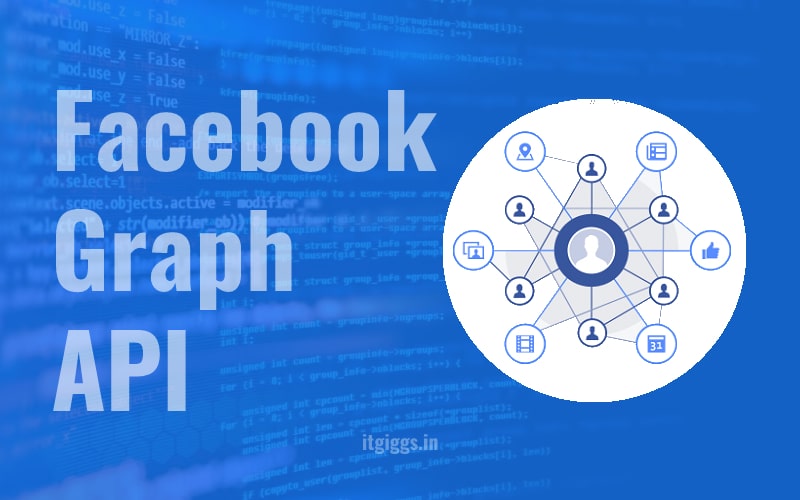
Nếu mục tiêu là tối ưu hóa số lượng các cặp thực thể có liên kết được dự đoán chính xác, độ bao phủ là metrics phù hợp.

Nếu mục tiêu là cân bằng giữa độ chính xác và độ bao phủ, hệ số F1 là metrics phù hợp.

# TRIỂN KHAI BÀI TOÁN DỰ ĐOÁN LIÊN KẾT TRÊN CƠ SỞ DỮ LIỆU ĐỒ THỊ NEO4J

## Thu thập bộ dữ liệu

Tác giả thu thập dữ liệu từ mạng xã hội Facebook, nền tảng phổ biến nhất tại Việt Nam với một lượng dữ liệu phong phú. Để minh họa cho các thuật toán của các mô hình phân lớp, tác giả sử dụng công cụ Graph API [10] để thu thập dữ liệu từ Facebook.



Hình 2.1 Công cụ Graph API của Facebook

A screenshot of a computer

Description automatically generatedTrước khi sử dụng Graph API, ta tạo tài khoản cá nhân thành tài khoản Facebook developer bằng cách vào trang <https://developers.facebook.com/async/registration>.

Hình 2.2 Trang sử dụng các ứng dụng của Facebook dành cho developers

Sau đó ta vào trang https://developers.facebook.com/apps/ và nhấn nút "Tạo ứng dụng". Đây là bước tạo ra ứng dụng Facebook, tài khoản người dùng nào chấp nhận liên kết với ứng dụng này sẽ có quyền truy cập và chia sẽ các thông tin mà ứng dụng yêu cầu (ở đây là thông tin về bạn bè). Sau đó sẽ được giao diện như hình 2.2, chọn và bấm tiếp tục.

A screenshot of a computer

Description automatically generated

Hình 2.3 Ảnh mô tả lựa chọn tạo ứng dụng trong Facebook

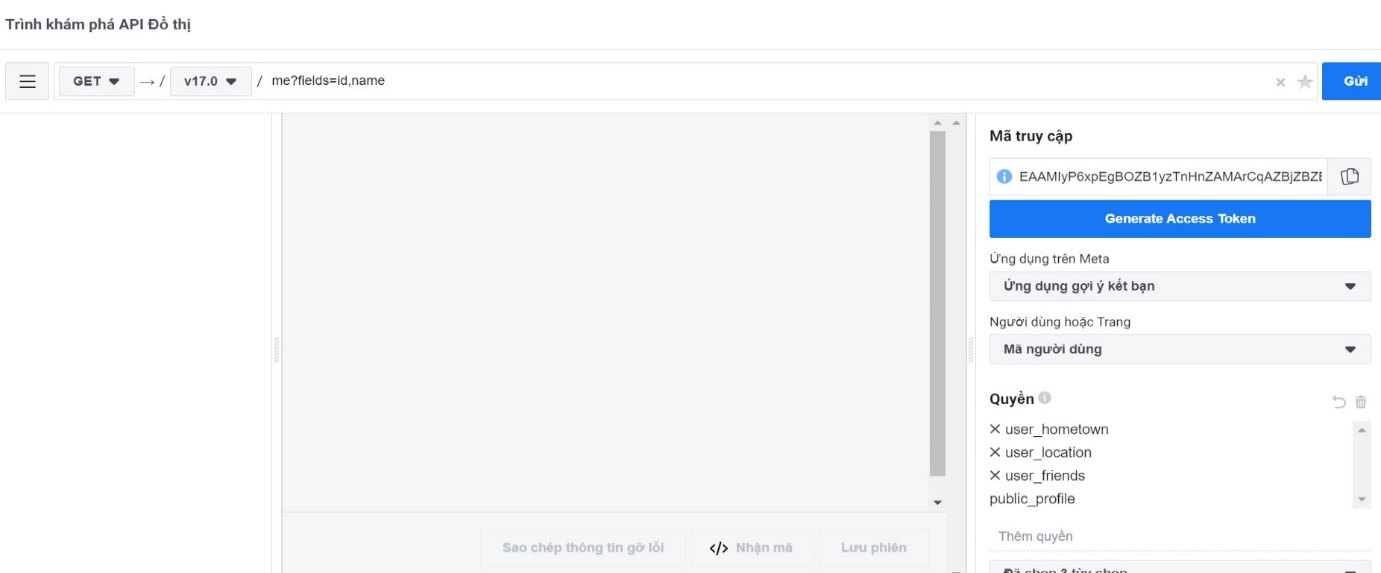
Sau khi tạo xong ứng dụng ta sẽ vào ứng dụng và chọn “Trường hợp sử dụng” ở menu công cụ ở bên trái, sau đó sẽ tiến hành thêm quyền để có thể sử dụng thêm dữ liệu về người dùng Facebook cho các hoạt động như hình 2.4.

A screenshot of a computer

Description automatically generated

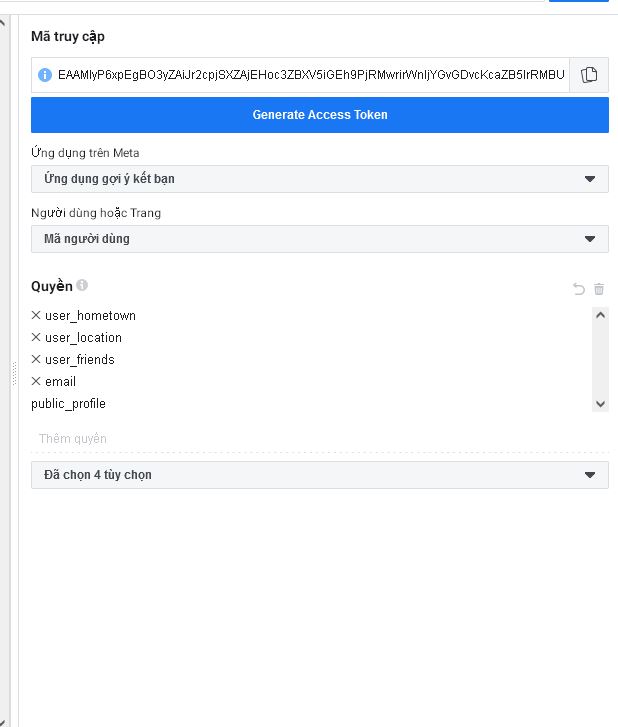
Hình 2.4 Cấp quyền muốn sử dụng để khai thác thông tin người dùng

Bước tiếp theo ta vào trang <https://developers.facebook.com/tools/explorer/> là công cụ khám phá API đồ thị và ta sẽ có giao diện như hình 2.5.



Hình 2.5 Ảnh mô tả việc chọn các quyền muốn sử dụng trong ứng dụng

Chọn và thêm quyền "user\_friends" cùng các quyền cần thiết khác. Hình dưới đây hiển thị màn hình yêu cầu quyền của ứng dụng. Khi ứng dụng được phát hành rộng rãi, hệ thống sẽ hiển thị thông báo xin quyền tương tự khi người dùng tham gia.



Hình 2.6 Chọn các quyền muốn sử dụng trong ứng dụng

Sau khi được cấp quyền truy cập, tác giả sử dụng trang Graph API để kiểm tra API của Facebook, thu được tổng số bạn bè thực tế và danh sách những bạn bè đã chấp nhận ứng dụng..

A screenshot of a computer

Description automatically generated

Hình 2.7 Thực hiện truy vấn để lấy các thông tin của người dùng

Thông tin Facebook bao gồm ba phần chính:

- Nút (Nodes): Các thực thể cơ bản như người dùng, bình luận, hình ảnh, trang.

- Cạnh (Edges): Mối quan hệ giữa các nút như hình ảnh của trang, mối quan hệ bạn bè, mối quan hệ chung sở thích.

- Tệp (Files): Thông tin chi tiết về các nút như ngày sinh, giới tính, email, số điện thoại, địa chỉ.

A screenshot of a computer

Description automatically generatedĐồ án đã sử dụng Graph API để thu thập dữ liệu từ 372 người dùng Facebook với các thông tin sau: friend\_id, friend\_name, friend\_gender, url, friend\_birthday, relationship\_status, hometown\_id, hometown\_name, location\_id, location\_name. Dữ liệu này gồm 11 cột, 373 dòng và có dung lượng 53KB, được lưu dưới định dạng file .csv.

Hình 2.8 Tập dữ liệu của người dùng Facebook sau khi lấy được

## Cơ sở dữ liệu đồ thị

### Môi trường cài đặt cơ sở dữ liệu

Cơ sở dữ liệu đồ thị Neo4j Desktop phiên bản 1.5.9 sẽ được cài đặt trên máy tính có CPU Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz, Ram: 04GB, hệ điều hành Windows 11 Pro-64bit.

A screenshot of a computer

Description automatically generated

Hình 2.9 Giao diện của cơ sở dữ liệu đồ thị Neo4J

### Tạo cơ sở dữ liệu

Sau khi cài đặt xong ta sẽ tiến hành tạo Project mới để lưu trữ dữ liệu như hình 2.10, tiếp đến ta sẽ tiến hành khởi động chạy như hình 2.11 và kết quả như hình 2.12.

A screenshot of a computer

Description automatically generated

Hình 2.10 Tạo cơ sở dữ liệu đồ thị mới “fb\_friend\_data” trong Project

A screenshot of a computer

Description automatically generated

Hình 2.11 Khởi động cơ sở dữ liệu đồ thị “fb\_friend\_data”

A screenshot of a computer

Description automatically generated

Hình 2.12 Giao diện sau khi đã khởi động thành công cơ sở dữ liệu

### Nạp dữ liệu vào cơ sở dữ liệu đồ thị Neo4J

* Đưa dữ liệu vào cơ sở đồ thị Neo4J

Lấy dữ liệu từ file csv fb\_friends\_data.csv và đưa vào Neo4j với mỗi người gồm các thông tin (friend\_id, friend\_name, friend\_gender, url, friend\_birthday, relationship\_status, location\_id, location\_name, hometown\_id, hometown\_name)

*1:* LOAD CSV WITH HEADERS FROM "file:///fb\_friends\_data.csv" AS nodes

*2:* CREATE (p:Person {friend\_id: nodes.friend\_id,friend\_name: nodes.friend\_name, friend\_gender:nodes.friend\_gender,url:nodes.url,friend\_birthday:nodes.friend\_birthday,relationship\_status:nodes.relationship\_status,location\_id:nodes.location\_id, location\_name:nodes.location\_name,hometown\_id:nodes.hometown\_id, hometown\_name:nodes.hometown\_name})

A screenshot of a computer code

Description automatically generated

Hình 2.13 Kết quả của truy vấn nạp người dùng Facebook vào Neo4j

* Tạo mối quan hệ giữa người dùng với nhau
* Mối quan hệ bạn bè

*1:* MATCH (p1:Person), (p2:Person)

*2:* WHERE p1.friend\_id < p2.friend\_id AND rand() > 0.95

*3:* WITH p1, p2

*4:* LIMIT 3500

*5:* MERGE (p1)-[:FRIEND]->(p2)

*6:* MERGE (p2)-[:FRIEND]->(p1)

A screenshot of a computer

Description automatically generated

Hình 2.14 Kết quả tạo mối quan hệ giữa các người dùng với nhau

* + Mối quan hệ cùng nơi thường trú với nhau

*1:* LOAD CSV WITH HEADERS FROM "file:///fb\_friends\_data.csv" AS nodes

*2:* WITH nodes

*3:* merge (a:Person{friend\_id: nodes.friend\_id,friend\_name: nodes.friend\_name})

*4:* merge(l:Location{location\_id:nodes.location\_id,  
location\_name:nodes.location\_name})

*5:* merge (a)-[:LIVE]->(l)

*6:* return a, l

A screenshot of a computer

Description automatically generated

Hình 2.15 Ảnh mô tả mối quan hệ cùng nơi ở thường trú

* + Mối quan hệ cùng quê quán với nhau

*1:* LOAD CSV WITH HEADERS FROM "file:///fb\_friends\_data.csv" AS nodes

2: WITH nodes

*3:* merge (a:Person{friend\_id: nodes.friend\_id,friend\_name: nodes.friend\_name})

*4:* merge(h:Hometown{hometown\_id:nodes.hometown\_id,  
hometown\_name:nodes.hometown\_name})

*5:* merge (a)-[:FROM]->(h)

*6:* return a, h

A screenshot of a computer

Description automatically generated

Hình 2.16 Ảnh mô tả mối quan hệ cùng quê quán

* + Mối quan hệ cùng độ tuổi với nhau

Với xác định độ tuổi thì sau khi lấy dữ liệu friend\_birthday ra dưới dạng date dd/mm/yyyy thì ta sẽ tiến hành lấy dữ liệu friend\_birthday đó sử dụng hàm split để tách dữ liệu ra để lấy năm sinh và sử dụng toInterger để chuyển về số nguyên và lưu vào birth\_year sau đó ta sẽ tạo node để chứa giá trị của birth\_year với năm sinh đến năm sinh + 10.

*1:* LOAD CSV WITH HEADERS FROM "file:///fb\_friends\_data.csv" AS nodes

*2:* WITH nodes, toInteger(split(nodes.friend\_birthday, '/')[0]) AS birth\_year

*3:* MERGE (age:Age {birth\_year: birth\_year})

*4:* ON CREATE SET age.age\_range =   
toString(birth\_year) + "-" + toString(birth\_year + 10)

*5:* merge (a:Person{friend\_id: nodes.friend\_id,friend\_name: nodes.friend\_name})

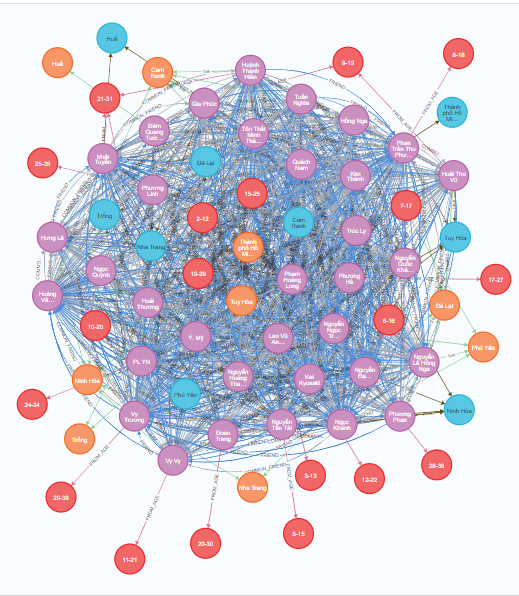
*6:* MERGE (a)-[:FROM\_AGE]->(age)

*7:* RETURN a, age

A screenshot of a computer

Description automatically generated

Hình 2.17 Ảnh mô tả mối quan hệ cùng độ tuổi



Hình 2.18 Mô hình cơ sở dữ liệu đồ thị của người dùng Facebook trong Neo4j

Kết quả cơ sở dữ liệu đồ thị đạt được là 475 Nút, 8,390 Relationship gồm:

+ 4 Node Labels (Person, Age,Hometown, Location)

+ 4 Relationship types( FRIEND, FROM\_AGE, from, live).

**Chú thích:**

|  |
| --- |
| **D:\TAI LIEU CAO HOC\NGHIEN CUU LUÂN VAN\NGHIEN CUU LUÂN VAN 28.3\file hình nền PP\nut person.png**  Màu tím thể hiện Nút của một User Facebook  Màu tím thể hiện Nút của một User facebook |
| Màu vàng thể hiện Nút nơi thường trú  Màu vàng thể hiện Nút nơi làm việc |
| Màu đỏ thể hiện Nút quê quán  Màu đỏ thể hiện Nút nơi sinh sống |
| Màu đỏ thể hiện Nút độ tuổi  Màu đỏ thể hiện Nút nơi sinh sống |
| D:\TAI LIEU CAO HOC\NGHIEN CUU LUÂN VAN\NGHIEN CUU LUÂN VAN 28.3\file hình nền PP\canh ss.png  Cạnh, thể hiện mối quan hệ cùng nơi sinh sống  Cạnh, thể hiện mối quan hệ cùng nơi sinh sống |
| Cạnh, thể hiện mối quan hệ cùng quê quán  Cạnh, thể hiện mối quan hệ cùng nơi làm việc |
| Cạnh, thể hiện mối quan hệ cùng độ tuổi  Cạnh, thể hiện mối quan hệ cùng nơi làm việc |
| Cạnh, thể hiện mối quan hệ bạn bè  **Hình 3.23. Giá trị tương tự phù hợp theo các mối quan hệ của User Diep Tieu My**Cạnh, thể hiện mối quan hệ bạn bè |

Bảng 2.1 Chú thích hình 2.18

# CÀI ĐẶT VÀ THỰC NGHIỆM CHƯƠNG TRÌNH

## Môi trường cài đặt và triển khai chương trình

* Ngôn ngữ lập trình

Python 3.8: Python là ngôn ngữ lập trình phổ biến trong lĩnh vực khoa học dữ liệu và học máy do cú pháp rõ ràng và có nhiều thư viện hỗ trợ mạnh mẽ.

* Thư viện và công cụ

NumPy: Thư viện hỗ trợ tính toán số học và xử lý mảng.

Pandas: Thư viện cung cấp các cấu trúc dữ liệu và công cụ phân tích dữ liệu.

Scikit-learn: Thư viện cung cấp các công cụ học máy đơn giản và hiệu quả.

NetworkX: Thư viện dùng để tạo, thao tác và nghiên cứu cấu trúc, động lực và chức năng của các mạng phức tạp.

Matplotlib: Thư viện vẽ đồ thị 2D trong Python.

Seaborn: Thư viện vẽ biểu đồ thống kê, xây dựng trên Matplotlib.

TensorFlow hoặc PyTorch: Một trong hai thư viện này sẽ được sử dụng để xây dựng và huấn luyện mô hình học sâu, tùy theo yêu cầu cụ thể của đề tài.

Neo4j: Thư viện giao tiếp với cơ sở dữ liệu đồ thị Neo4j.

PyQt5: Thư viện để phát triển giao diện người dùng đồ họa (GUI) cho ứng dụng.

XlsxWriter: Thư viện để tạo và chỉnh sửa các tệp Excel.

datetime: Thư viện cung cấp các lớp để xử lý ngày và thời gian.

time: Thư viện hỗ trợ đo thời gian thực thi và các chức năng liên quan đến thời gian.

Các thư viện cụ thể từ Scikit-learn:

* train\_test\_split: Chức năng chia dữ liệu thành tập huấn luyện và tập kiểm tra.
* GaussianNB: Thuật toán Naive Bayes phân phối chuẩn.
* LogisticRegression: Thuật toán hồi quy logistic.
* DecisionTreeClassifier: Thuật toán cây quyết định.
* precision\_score, recall\_score, f1\_score, confusion\_matrix: Các công cụ đánh giá hiệu suất của mô hình.
* LabelEncoder: Công cụ mã hóa nhãn dữ liệu.
* euclidean\_distances: Chức năng tính toán khoảng cách Euclidean giữa các điểm.
* Cơ sở dữ liệu

Neo4j: Một hệ quản trị cơ sở dữ liệu đồ thị, phù hợp với việc lưu trữ và xử lý dữ liệu mạng xã hội.

* Công cụ phát triển

VS Code: Một trình soạn thảo mã nguồn mở, mạnh mẽ và dễ sử dụng, hỗ trợ nhiều ngôn ngữ lập trình và tích hợp nhiều công cụ.

* Hệ thống quản lý phiên bản

Git: Công cụ quản lý phiên bản mã nguồn, giúp theo dõi các thay đổi và cộng tác phát triển dự án.

## Tạo tập dữ liệu huấn luyện và kiểm tra

* Bước 1: Truy vấn cơ sở dữ liệu:

Sau khi đã nạp toàn bộ dữ liệu vào cơ sở dữ liệu Neo4j ta sẽ thực hiện một truy vấn để lấy danh sách các người dùng không phải là bạn bè của người dùng được chọn. Trong truy vấn này, chúng ta tìm kiếm các người dùng mà người dùng được chọn chưa kết bạn với họ.

**Mã giả:**

1. *Định nghĩa truy vấn để:*

*- Tìm người dùng 'kei' qua tên 'selected\_user'*

*- Tìm những người dùng 'potentialFriend' không phải là bạn bè của 'kei'*

*- Khớp các bạn chung giữa 'kei' và 'potentialFriend' (nếu có)*

*- Đếm số bạn chung và gán vào biến 'common\_friends\_count'*

*- Trả về ID, tên, ngày sinh, giới tính, loại bạn bè, tình trạng quan hệ, ID quê quán, và ID địa điểm của 'potentialFriend'*

1. *Chạy truy vấn với tham số là tên của 'selected\_user'*
2. *Lưu kết quả truy vấn vào biến 'data'*

Với biến ‘loại bạn bề’ trên sẽ được sử dụng để trả về 2 loại giá trị đó là 1 và 2 nếu

biến 'common\_friends\_count' > 0 thì biến ‘loại bạn bè’ sẽ có giá trị là 1 và ngược lại nếu < 0 thì biến ‘loại bạn bè’ sẽ có giá trị là 2.

* Bước 2: Xử lý dữ liệu từ kết quả truy vấn:

Dữ liệu được trích xuất từ kết quả truy vấn bao gồm friend\_id, age (được tính từ ngày sinh), gender (được chuyển đổi thành số), friend (loại bạn bè), relationship\_status (được chuyển đổi thành số), hometown\_id và location\_id.

Cụ thể:

* + age được tính từ friend\_birthday thông qua hàm calculate\_age(birthdate\_str)
  + gender được chuyển đổi nếu friend\_gender là Male thì là 1 ngược lại là Female thì là 0
  + friend biến biểu diễn cho người dùng có bạn chung với nhau hay không nếu có thì 1 ngược lại nếu không có bạn chung thì là 2
  + relationship\_status được chuyển đổi nếu relationship\_status là “Độc thân” thì là 1 hoặc nếu là “Đang hẹn hò” thì là 2 và cuối cùng nếu là “Đã kết hôn” thì là 3
  + Nếu hometown\_id hay location\_id của người dùng trong danh sách có sự trùng nhau với dữ liệu của người dùng được chọn và biến friend là 2 thì sẽ chuyển biến friend thành 1

**Mã giả:**

*1: Nếu 'data' không rỗng:*

*2: Trích xuất ID của những người không phải bạn bè và in chúng ra*

*3: Khởi tạo danh sách 'X' và 'y' cho bộ đặc trưng và nhãn*

*4: Vòng lặp xử lý từng bản ghi trong 'data':*

*5: Kiểm tra xem tất cả các trường dữ liệu có tồn tại:*

*6: - Nếu đủ thông tin:*

*7: Tính tuổi từ ngày sinh*

*8: Chuyển đổi tình trạng mối quan hệ thành số:*

*9: - "Độc thân" thành 1*

*10: - "Đang hẹn hò" thành 2*

*11: - "Đã kết hôn" thành 3*

*12: - Các trường hợp khác thành 0*

*13: Chuyển đổi giới tính thành số:*

*14: - "MALE" thành 1*

*15: - "FEMALE" thành 0*

*16: Chuyển friend thành 1*

*17: - Nếu location\_id hay hometown\_id của 2 người dùng giống nhau*

*18: - Nếu friend là 2*

* Bước 3: Chuẩn bị dữ liệu cho huấn luyện và dự đoán:

Dữ liệu từ kết quả truy vấn được xử lý để chuẩn bị cho quá trình huấn luyện và dự đoán. Các trường thông tin được chuyển đổi thành dạng số và được lưu vào các mảng X và y. Mảng X chứa các thông tin về người dùng không phải là bạn bè, trong khi mảng y chứa giới tính của họ (1 là có bạn chung. 2 là không có bạn chung).

Trong trường hợp không đủ dữ liệu trả về các mảng X và y để sử dụng trong quá trình huấn luyện và dự đoán thì các mảng này sẽ được trả về trống.

**Mã giả:**

*1: Thêm bộ (ID bạn, tuổi, giới tính, tình trạng mối quan hệ, ID quê hương, ID địa điểm) vào 'X'*

*2: Thêm chỉ báo có bạn chung (1 nếu có, 2 nếu không) vào 'y'*

*3: In 'selected\_user', 'X', và 'y'*

*4: Trả về 'X' và 'y'*

*5: Nếu 'data' rỗng:*

*6: In thông báo không tìm thấy người dùng không phải bạn bè*

*7: Trả về danh sách rỗng cho 'X' và 'y'*

* Bước 4: Tạo tập dữ liệu huấn luyện và kiểm tra

Khi đã có được mảng dữ liệu X, y ta sẽ tạo tập dữ liệu huấn luyện và kiểm tra nằm trong phương thức find\_friends của lớp Starter

Với X là tập dữ liệu đặc trưng và y là tập nhãn tương ứng. Hàm train\_test\_split từ sklearn.model\_selection được sử dụng để chia dữ liệu thành tập huấn luyện và tập kiểm tra. Tham số test\_size=0.2 chỉ định rằng 20% dữ liệu sẽ được sử dụng cho việc kiểm tra và random\_state=42 đảm bảo tính nhất quán trong việc chia dữ liệu. Sau khi chia, các tập dữ liệu huấn luyện và kiểm tra sẽ được sử dụng để huấn luyện mô hình và đánh giá hiệu suất của mô hình.

**Mã giả:**

*1: Phân chia dữ liệu 'X' và 'y' thành tập huấn luyện và tập kiểm thử:*

*2: Sử dụng hàm train\_test\_split với các tham số:*

*3: - X là dữ liệu đầu vào*

*4: - y là nhãn*

*5: - test\_size là 0.2 (20% dữ liệu cho kiểm thử)*

*6: - random\_state là 42 để đảm bảo kết quả nhất quán qua các lần chạy*

*7: Lưu kết quả phân chia vào các biến 'X\_train', 'X\_test', 'y\_train', 'y\_test'*

* Bước 5: Huấn luyện mô hình

Đầu tiền ta sẽ sử dụng hàm encode\_features(X) để duyệt qua từng dòng của ma trận dữ liệu X: Hàm này duyệt qua từng dòng của ma trận dữ liệu X, biểu diễn các mẫu dữ liệu. Sau đó sẽ mã hóa các đặc trưng của mỗi dòng và sẽ trả về ma trận dữ liệu đã được mã hóa: Kết quả là một ma trận dữ liệu mới với các đặc trưng đã được mã hóa.

**Mã giả:**

*1: Khởi tạo danh sách rỗng X\_encoded và mã hóa nhãn*

*2: Duyệt qua mỗi hàng trong X:*

*3: Khởi tạo danh sách rỗng encoded\_row*

*4: Duyệt qua mỗi giá trị trong hàng:*

*5: Kiểm tra nếu giá trị là chuỗi:*

*6: - Kiểm tra nếu cột i chưa có trong mã\_hóa\_nhãn, thì tạo một LabelEncoder mới.*

*7: - Lấy tất cả các giá trị chuỗi duy nhất trong cột i của X và huấn luyện   
LabelEncoder với các giá trị này.*

*8: - Mã hóa giá trị hiện tại bằng LabelEncoder và thêm giá trị đã mã hóa vào dòng\_đã\_mã\_hóa.*

*9: Ngược lại:*

*10: Thêm giá trị số vào encoded\_row*

*11: Thêm encoded\_row vào X\_encoded*

*12: Trả về X\_encoded*

Tiếp đến ta sẽ sử dụng hàm train(X\_train, y\_train) với việc gọi hàm encode\_features để mã hóa các đặc trưng của dữ liệu huấn luyện X\_train và điền các giá trị thiếu (nếu có) bằng cách sử dụng SimpleImputer, hàm này điền các giá trị thiếu trong dữ liệu đã được mã hóa bằng các giá trị mặc định được chỉ định (ở đây là 0.0).

Cuối cùng ta sẽ tiến hành huấn luyện mô hình trên dữ liệu đã được mã hóa và điền giá trị thiếu.

**Mã giả:**

*1: Gọi hàm mã\_hóa\_tính\_năng trên X\_train và lưu vào X\_train\_encoded*

*2: Huấn luyện mô hình với X\_train\_ encoded và y\_train*

## Xây dựng các tính năng dự đoán liên kết

Sau khi huấn luyện mô hình thành công ta sẽ tiến hành dự đoán liên kết trên mô hình vừa được huấn luyện

* Bước 1: Dự đoán kết quả đặc trưng của tập kiểm tra sử dụng mô hình đã huấn luyện

Đoạn mã giả sử hàm model.predict để dự đoán kết quả của tập kiểm tra. Nếu dự đoán nhãn cho điểm dữ liệu là 1 (có khả năng kết bạn) thì sẽ tiếp tục xử lý nếu không sẽ bỏ qua.

* Bước 2: Tính toán ngưỡng kết nối

Đoạn mã sử dụng hàm calculate\_distance để tính toán khoảng cách giữa người dùng được chọn và các người dùng khác trong danh sách.

Thêm khoảng cách vào danh sách distances.

Sau đó tính ngưỡng kết nối dựa trên trung bình và độ lệch chuẩn theo danh sách khoảng cách vừa tính được.

* Bước 3: Kiểm tra nếu khoảng cách giữa người dùng và điểm dữ liệu hiện tại

Nếu khoảng cách giữa người dùng và điểm dữ liệu hiện tại nhỏ hơn hoặc bằng ngưỡng kết nối, kiểm tra thêm các điều kiện:

* Nếu điểm dữ liệu phù hợp với tiêu chí quê quán và nơi sinh sống, thêm đề xuất kết bạn vào dah sách recommendations.
* Nếu điểm dữ liệu chỉ phù hợp với tiêu chí quê quán, thêm đề xuất kết bạn vào danh sách recommendations.
* Nếu điểm dữ liệu chỉ phù hợp với tiêu chí nơi sinh sống, thêm đề xuất kết bạn vào danh sách recommendations.

|  |
| --- |
| Tiêu chí |
| Cùng quê quán, cùng nơi ở |
| Cùng quê quán |
| Cùng nơi ở |
| Ngẫu nhiên |

Bảng 3.1 Mô tả tiêu chí lọc để hiển thị trên bảng kết quả

* Bước 4: Xây dựng danh sách gợi ý kết bạn:

Dựa trên thông tin được thu thập từ các bước trên, đoạn mã sẽ xây dựng một danh sách gợi ý kết bạn, bao gồm thông tin về các người dùng mà bạn có thể muốn kết bạn dựa trên các tiêu chí như quê quán, nơi sinh sống, độ tuổi và giới tính.

**Mã giả:**

*1: Hàm dự\_đoán(self, X\_test, user\_data)*

*2: Lấy user\_location\_id từ user\_data*

*3: Lấy user\_hometown\_id từ user\_data*

*4: Lấy user\_age sau khi đã chuyển đổi tuổi của user\_data*

*5: Lấy user\_gender sau khi đã chuyển đổi giới tính của user\_data*

*6: Khởi tạo danh sách recommendations rỗng*

*7: Khởi tạo danh sách distances rỗng*

*8: Gọi hàm encode\_features để mã hóa X\_test và lưu vào X\_test\_encoded*

*9: Gọi hàm model.predict để dự đoán X\_test\_encoded và lưu vào predictions*

*10: Duyệt qua mỗi điểm dữ liệu (data\_point) trong X\_test:*

*11: Nếu predictions[i] == 1:*

*12: Lấy test\_location\_id từ data\_point tại vị trí 5*

*13: Lấy test\_hometown\_id từ data\_point tại vị trí 4*

*14: Lấy test\_age từ data\_point tại vị trí 1*

*15: Lấy test\_gender từ data\_point tại vị trí 2*

*16: Tạo vector đặc trưng cho người dùng được chọn lưu trong user\_vector*

*17: Tạo vector đặc trưng cho người dùng trong danh sách lưu trong test\_vector*

*18: Tính khoảng cách giữa user\_vector và test\_vector*

*19: Thêm khoảng cách vào danh sách distances*

*20: Tính mean\_distance là trung bình của danh sách distances*

*21: Tính std\_distance là độ lệch chuẩn của danh sách distances*

*22: Tính linking\_threshold là mean\_distance cộng với std\_distance*

*23: Nếu khoảng cách <= linking\_threshold:*

*24: Nếu độ dài của data\_point lớn hơn 5 và giá trị tại vị trí 4 là hometown\_id và giá trị tại vị trí 5 là location\_id của user\_data:*

*25: Lấy friend\_id từ data\_point*

*26: Lấy friend\_data từ connector sử dụng friend\_id*

*27: In "Cùng quê quán và nơi sinh sống:" và friend\_data*

*28: Nếu friend\_data tồn tại:*

*29: Chuyển friend\_data thành từ điển friend\_data\_dict*

*30: Lấy friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name từ friend\_data\_dict*

*31: Nếu friend\_name tồn tại:*

*32: Thêm (friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name, "Cùng quê quán và nơi sinh sống", distance, "Gợi ý kết bạn") vào recommendations*

*33: Ngược lại:*

*34: In "No data found for friend with ID: {friend\_id}"*

*35: Ngược lại:*

*36: In "No data found for friend with ID: {friend\_id}"*

*37: Ngược lại nếu độ dài của data\_point lớn hơn 4 và giá trị tại vị trí 4 là hometown\_id:*

*38: Lấy friend\_id từ data\_point*

*39: Lấy friend\_data từ connector sử dụng friend\_id*

*40: In "Cùng quê quán:" và friend\_data*

*41: Nếu friend\_data tồn tại:*

*42: Chuyển friend\_data thành từ điển friend\_data\_dict*

*43: Lấy friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name từ friend\_data\_dict*

*44: Nếu friend\_name tồn tại:*

*45: Thêm (friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name, "Cùng quê quán", distance, "Gợi ý kết bạn") vào recommendations*

*46: Ngược lại:*

*47: In "No data found for friend with ID: {friend\_id}"*

*48: Ngược lại:*

*49: In "No data found for friend with ID: {friend\_id}"*

*50: Ngược lại nếu độ dài của data\_point lớn hơn 5 và giá trị tại vị trí 5 là location\_id của user\_data:*

*51: Lấy friend\_id từ data\_point*

*52: Lấy friend\_data từ connector sử dụng friend\_id*

*53: In "Cùng nơi sinh sống:" và friend\_data*

*54: Nếu friend\_data tồn tại:*

*55: Chuyển friend\_data thành từ điển friend\_data\_dict*

*56: Lấy friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name từ friend\_data\_dict*

*57: Nếu friend\_name tồn tại:*

*58: Thêm (friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name, "Cùng nơi sinh sống", distance, "Gợi ý kết bạn") vào recommendations*

*59: Ngược lại:*

*60: In "No data found for friend with ID: {friend\_id}"*

*61: Ngược lại:*

*62: Lấy friend\_id từ data\_point*

*63: Lấy friend\_data từ connector sử dụng friend\_id*

*64: Nếu friend\_data tồn tại:*

*65: Chuyển friend\_data thành từ điển friend\_data\_dict*

*66: Lấy friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name từ friend\_data\_dict*

*67: Nếu friend\_name tồn tại:*

*68: Thêm (friend\_name, friend\_birthday, friend\_gender, hometown\_name, location\_name, "Ngẫu nhiên", distance, "Gợi ý kết bạn") vào recommendations*

*69: Ngược lại:*

*70: In "No data found for friend with ID: {friend\_id}"*

*71: Ngược lại:*

*72: In "No data found for friend with ID: {friend\_id}"*

*73: Trả về recommendations*

Công thức tổng quát tính điểm liên kết từ đoạn mã giả trên như sau:

Giả sử rằng mỗi người dùng 𝑢 được đại diện bởi một vector đặc trưng 𝑢=[𝑢hometown, 𝑢 location, 𝑢age, 𝑢gender] và mỗi đối tượng kết bạn tiềm năng 𝑣 cũng được đại diện bởi một vector đặc trưng 𝑣=[𝑣hometown, 𝑣location, 𝑣age, 𝑣gender]

Khoảng cách Euclidean giữa hai vector đặc trưng 𝑢 và 𝑣 được tính như sau:

𝑑(𝑢,𝑣) = (1)

Các bước tính điểm liên kết

* **Tính toán khoảng cách Euclidean**: Tính khoảng cách Euclidean giữa vector đặc trưng của người dùng 𝑢 và từng vector đặc trưng của các đối tượng kết bạn tiềm năng

𝑑(𝑢,) = (2)

* **Tính ngưỡng liên kết**: Tính giá trị trung bình () và độ lệch chuẩn () của các khoảng cách Euclidean đã tính được. Ngưỡng liên kết () được xác định bằng tổng của giá trị trung bình và độ lệch chuẩn.

(3)

(4)

* **So sánh với ngưỡng liên kết**: So sánh khoảng cách Euclidean của mỗi đối tượng kết bạn tiềm năng với ngưỡng liên kết để xác định những đối tượng có thể kết bạn.

𝑑(𝑢,)

Tính khoảng cách giữa người dùng 𝑢 “Kei Kyosaki” và người dùng “Nguyễn Cát Khánh Phương”:

Với 𝑢=[‘352’, ‘179’,17,0] và 𝑣=[‘109’,’109’,23.0].

Từ công thức (1) (2) (3) (4) ta có được kết quả như sau:

𝑑(𝑢,) = 252.95256472311166.

= 305.9344947676756.

= 52.98193004456394.

= 358.9164248122395.

Vậy với 𝑑(𝑢,) thì người dùng sẽ có mặt trong danh sách đề xuất kết bạn với người dùng 𝑢.

## Đánh giá mô hình

Hàm evaluate được sử dụng để đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm tra.

* Bước 1 Mã hóa đặc trưng của dữ liệu kiểm tra:

Hàm này gọi encode\_features để mã hóa các đặc trưng của dữ liệu kiểm tra X\_test.

* Bước 2 Dự đoán kết quả:

Sử dụng mô hình đã được huấn luyện, hàm dự đoán kết quả dựa trên dữ liệu kiểm tra đã được mã hóa.

* Bước 3 Tính các điểm đánh giá:

Sử dụng các hàm tính toán các điểm đánh giá như accuracy\_score, precision\_score, recall\_score, và f1\_score để tính toán độ chính xác, độ chính xác, độ bao phủ, và hệ số F1.

* Bước 4 Trả về các điểm đánh giá:

Kết quả cuối cùng trả về là một bộ các điểm đánh giá bao gồm độ chính xác (precision), độ bao phủ (recall), và hệ số F1 (F1-score) để hiển thị lên giao diện chương trình.

**Mã giả:**

*1: Hàm đánh\_giá(self, X\_test, y\_test)*

*2: X\_test\_encoded = self.encode\_features(X\_test)*

*3: predictions = self.model.predict(X\_test\_encoded)*

*4: confusion\_mat = confusion\_matrix(y\_test, predictions)*

*5: TN = confusion\_mat[1, 1]*

*6: FP = confusion\_mat[1, 0]*

*7: FN = confusion\_mat[0, 1]*

*8: TP = confusion\_mat[0, 0]*

*9: In thông báo "Matrix:", confusion\_mat*

*10: In thông báo "True Positives (TP):", TP*

*11: In thông báo "True Negatives (TN):", TN*

*12: In thông báo "False Positives (FP):", FP*

*13: In thông báo "False Negatives (FN):", FN*

*14: precision = precision\_score(y\_test, predictions)*

*15: recall = recall\_score(y\_test, predictions)*

*16: f1 = f1\_score(y\_test, predictions)*

*17: Trả về precision, recall, f1*

## Chương trình thực nghiệm

Các bước thực hiện hệ thống cụ thể như sau:

Bước 1: Lựa chọn người dùng muốn gợi ý kết bạn

Bước 2: Lựa chọn mô hình phân lớp muốn sử dụng

A screenshot of a computer

Description automatically generated

Hình 3.1 Giao diện hệ thống khuyến nghị kết bạn

A screenshot of a computer

Description automatically generated

Hình 3.2 Giao diện lựa chọn người dùng và mô hình phân lớp

Bước 3: Nhất nút “TÌM BẠN”, sau đó chương trình sẽ hiển thị thông tin người dùng được chọn ở bước 1, danh sách người dùng phù hợp để gợi ý kết bạn và mô tả các thông số hiệu suất của mô hình phân lớp.

A screenshot of a computer

Description automatically generated

Hình 3.3 Giao diện hiển thị kết quả sau khi nhấn nút “Tìm Bạn”

Sau khi chương trình xử lý các dữ liệu sẽ hiển thị kết quả dự đoán của người dùng và mô hình phân lớp được chọn.

A screenshot of a computer

Description automatically generated

Hình 3.4 Giao diện dự đoán của "Phương Phan" và "Logistic Regression"

A screenshot of a computer

Description automatically generated

Hình 3.5 Giao diện dự đoán của "Nguyễn Thành Đạt" và "Naive Bayes"

A screenshot of a computer

Description automatically generated

Hình 3.6 Giao diện dự đoán của "Kei Kyosaki" và "Decision Tree"

Bước 4: Sau khi đã lấy được danh sách người dùng phù hợp với người dùng được chọn bạn đầu thì người dùng có thể chọn xuất ra file tài liệu Excel để lưu trữ bằng cách nhấn vào nút “In kết quả vào Excel”.

A screenshot of a computer

Description automatically generated

Hình 3.7 Giao diện khi lưu kết quả của chương trình vào file Excel

A screenshot of a computer

Description automatically generated

Hình 3.8 Giao diện báo lỗi khi người dùng chưa chạy chương trình "Tìm Bạn"

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Hình 3.9 Kết quả in các bảng dữ liệu của chương trình vào trong Excel

## Đánh giá kết quả của chương trình

Xét theo tập dữ liệu của người dùng được chọn là “Nguyễn Lê Hồng Nga” (với tỉ lệ chia dữ liệu giữa dữ liệu huấn luyện và dữ liệu kiểm tra là 8:2) và ngưỡng liên kết của mỗi mô hình, ta có được là kết quả của danh sách người dùng được dự đoán có thể đề xuất kết bạn như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Mô hình phân lớp | Logistic Regression | Naive Bayes | Decision Tree |
| Số dự đoán phù hợp | 50 người dùng | 53 người dùng | 49 người dùng |
| Ngưỡng liên kết | 303.1774397329405 | 302.2685542089184 | 304.25904074219125 |

Bảng 3.2 Kết quả chạy của các mô hình phân lớp

Để đánh giá hiệu suất của các mô hình phân lớp ta sẽ đánh giá theo hai phương pháp, cụ thể là: So sánh các tiêu chí đánh giá (Độ chính xác, Độ bao phủ, Hệ số F1); So sánh thời gian xử lý của từng mô hình phân lớp (thời gian huấn luyện và thời gian dự đoán). Với tỉ lệ chia dữ liệu của các tập dữ liệu huấn luyện và tập dữ liệu kiểm tra lần lượt là 8:2, 5:5 và 2:8.

Xét theo tập dữ liệu của người dùng được chọn là “Nguyễn Lê Hồng Nga” với tỉ lệ chia dữ liệu là 8:2, ta có được là danh sách người dùng trong tập dữ liệu kiểm tra đó là 72 người dùng có khả năng kết bạn.

* So sánh các tiêu chí đánh giá (Độ chính xác, Độ bao phủ, Hệ số F1) thì ta có như sau:
* Mô hình Logistic Hồi quy (Logistic Regression)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| friend\_id | friend\_name | friend | Dự đoán | So sánh |
| 100856157 | Công Lợi | 1 | 1 | T |
| 100367707 | Nguyễn Cát Khánh Phương | 1 | 1 | T |
| 100298486 | Quách Nam | 1 | 1 | T |
| 100206041 | Hoàng Gia Kts | 1 | 2 | F |
| 100591751 | Nhi Nhi | 1 | 1 | T |
| 100102913 | Duyen Tran | 1 | 2 | F |
| 100951649 | Thuy Huynh | 1 | 1 | T |
| 100537247 | Minh Như Teuk | 1 | 1 | T |
| 100161967 | Ha Linh | 1 | 2 | F |
| 100571039 | Công Thành | 1 | 1 | T |
| 100675942 | Nguyễn Ngọc Thảo Quyên | 1 | 1 | T |
| 100032369 | Luong Ngoc Loi | 2 | 2 | T |
| 100416401 | Liễu Huyền | 1 | 1 | T |
| 100341100 | Nguyễn Quốc Khánh | 1 | 1 | T |
| 100746988 | Trương Phước Kháng | 1 | 1 | T |
| 100593928 | Huỳnh Minh Huy | 1 | 1 | T |
| 100128895 | Đàm Quang Tường | 1 | 2 | F |
| 100334725 | Thy Thy | 1 | 1 | T |
| 100416367 | Tuyết Quyên | 1 | 1 | T |
| 100016746 | Quy Nguyen | 2 | 2 | T |
| 100362038 | Quynh Quynh | 1 | 1 | T |
| 100875703 | Helen Nguyễn | 1 | 1 | T |
| 100199385 | Yu Na | 1 | 2 | F |
| 100488692 | Phượng Cấn Thị | 1 | 1 | T |
| 100872241 | Trần Quốc | 1 | 1 | T |
| 100436705 | Phạm Hoàng Long | 1 | 1 | T |
| 100747859 | Van Anh | 1 | 1 | T |
| 100072317 | Trần Nguyên Đăng Vũ | 1 | 2 | F |
| 100512231 | Trần Thị Kiều Sâm | 1 | 1 | T |
| 100456770 | Văn Minh Hoan | 1 | 1 | T |
| 100343699 | Trang Lê | 1 | 1 | T |
| 100453442 | Nguyễn Cườngg | 1 | 1 | T |
| 100026949 | Trương Phi Hoàng | 2 | 2 | T |
| 100440449 | Dương Hồng Chi | 1 | 1 | T |
| 100607196 | Nguyễn Kim Tài | 1 | 1 | T |
| 100628809 | Nguyễnn Uyênn | 1 | 1 | T |
| 100884526 | Trần Hồng Giang | 1 | 1 | T |
| 100024946 | Thảo Nguyên Vũ | 2 | 2 | T |
| 100641068 | Lê Sung Chôi | 1 | 1 | T |
| 100474957 | Nhầm Duyên | 1 | 1 | T |
| 100051611 | Linh Lê | 1 | 2 | F |
| 100549204 | Nguyễn Xuân Hoài An | 1 | 1 | T |
| 100341710 | Vongoc Quynhchi | 1 | 1 | T |
| 100606378 | Pun Oăn | 1 | 1 | T |
| 100566101 | Trúc Quỳnh | 1 | 1 | T |
| 100655372 | Tuan Tran | 1 | 1 | T |
| 100679515 | Nguyễn Lê Vân | 1 | 1 | T |
| 100925729 | Thanh Hoàng Nguyễn | 1 | 1 | T |
| 100357873 | Khien Pham | 1 | 1 | T |
| 100848972 | Ngọc Trầm | 1 | 1 | T |
| 100266239 | Rei Nguyen | 1 | 1 | T |
| 100893934 | Vũ Lê | 1 | 1 | T |
| 100201300 | Chu Le Dung | 1 | 2 | F |
| 100722422 | Lương Tuấn Minh | 1 | 1 | T |
| 100020742 | Cẩm Tiên | 2 | 2 | T |
| 100738549 | Nguyễn Chi Lê | 1 | 1 | T |
| 100671693 | Duy Hoàng | 1 | 1 | T |
| 100406193 | Thanh Tâm Lưu Thị | 1 | 1 | T |
| 100923735 | Lie THANH | 1 | 1 | T |
| 100317963 | Công Minh | 1 | 1 | T |
| 100481276 | Tôn Nữ Thị Lài | 1 | 1 | T |
| 100902198 | Tuyết Nhung | 1 | 1 | T |
| 100730083 | Liên Liên Truong | 1 | 1 | T |
| 100855437 | Lê Anh Tuấn | 1 | 1 | T |
| 100517338 | Đoan Trang | 1 | 1 | T |
| 100437894 | Duong Minh Thien | 1 | 1 | T |
| 100033013 | Thảo Vi | 2 | 2 | T |
| 100951924 | Art Phạm | 1 | 1 | T |
| 100243619 | Thái Nhi | 1 | 1 | T |
| 100345920 | William Nguyễn | 1 | 1 | T |
| 100223813 | Khánh Nhật Trần | 1 | 1 | T |
| 100849255 | MI HU | 1 | 1 | T |

Bảng 3.3 Kết quả dự đoán của mô hình Logistic hồi quy (tỉ lệ 8:2)

Từ bảng dữ liệu trên ta có được bảng giá trị TP, TN, FP, FN tương ứng

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 58 | 0 | 6 | 8 |

Bảng 3.4 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 8:2)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + Độ chính xác (Precision): 1
  + Độ bao phủ (Recall): 0.878787879
  + Hệ số F1 (F1-score): 0.935483871
* Mô hình giả định Bayes (Naive Bayes)

Tương tự với mô hình Logistic Regression ta cũng sẽ có 1 bảng dữ liệu và bảng giá trị TP, TN, FP, FN tương ứng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| friend\_id | friend\_name | friend | Dự đoán | So sánh |
| 100856157 | Công Lợi | 1 | 1 | T |
| 100367707 | Nguyễn Cát Khánh Phương | 1 | 1 | T |
| 100298486 | Quách Nam | 1 | 1 | T |
| 100206041 | Hoàng Gia Kts | 1 | 2 | F |
| 100591751 | Nhi Nhi | 1 | 1 | T |
| 100102913 | Duyen Tran | 1 | 2 | F |
| 100951649 | Thuy Huynh | 1 | 1 | T |
| 100537247 | Minh Như Teuk | 1 | 1 | T |
| 100161967 | Ha Linh | 1 | 2 | F |
| 100571039 | Công Thành | 1 | 1 | T |
| 100675942 | Nguyễn Ngọc Thảo Quyên | 1 | 1 | T |
| 100032369 | Luong Ngoc Loi | 2 | 2 | T |
| 100416401 | Liễu Huyền | 1 | 1 | T |
| 100341100 | Nguyễn Quốc Khánh | 1 | 1 | T |
| 100746988 | Trương Phước Kháng | 1 | 1 | T |
| 100593928 | Huỳnh Minh Huy | 1 | 1 | T |
| 100128895 | Đàm Quang Tường | 1 | 1 | F |
| 100334725 | Thy Thy | 1 | 1 | T |
| 100416367 | Tuyết Quyên | 1 | 1 | T |
| 100016746 | Quy Nguyen | 2 | 1 | T |
| 100362038 | Quynh Quynh | 1 | 1 | T |
| 100875703 | Helen Nguyễn | 1 | 1 | T |
| 100199385 | Yu Na | 1 | 2 | F |
| 100488692 | Phượng Cấn Thị | 1 | 1 | T |
| 100872241 | Trần Quốc | 1 | 1 | T |
| 100436705 | Phạm Hoàng Long | 1 | 1 | T |
| 100747859 | Van Anh | 1 | 1 | T |
| 100072317 | Trần Nguyên Đăng Vũ | 1 | 2 | F |
| 100512231 | Trần Thị Kiều Sâm | 1 | 1 | T |
| 100456770 | Văn Minh Hoan | 1 | 1 | T |
| 100343699 | Trang Lê | 1 | 1 | T |
| 100453442 | Nguyễn Cườngg | 1 | 1 | T |
| 100026949 | Trương Phi Hoàng | 2 | 2 | T |
| 100440449 | Dương Hồng Chi | 1 | 1 | T |
| 100607196 | Nguyễn Kim Tài | 1 | 1 | T |
| 100628809 | Nguyễnn Uyênn | 1 | 1 | T |
| 100884526 | Trần Hồng Giang | 1 | 1 | T |
| 100024946 | Thảo Nguyên Vũ | 2 | 1 | T |
| 100641068 | Lê Sung Chôi | 1 | 1 | T |
| 100474957 | Nhầm Duyên | 1 | 1 | T |
| 100051611 | Linh Lê | 1 | 2 | F |
| 100549204 | Nguyễn Xuân Hoài An | 1 | 1 | T |
| 100341710 | Vongoc Quynhchi | 1 | 1 | T |
| 100606378 | Pun Oăn | 1 | 1 | T |
| 100566101 | Trúc Quỳnh | 1 | 1 | T |
| 100655372 | Tuan Tran | 1 | 1 | T |
| 100679515 | Nguyễn Lê Vân | 1 | 1 | T |
| 100925729 | Thanh Hoàng Nguyễn | 1 | 1 | T |
| 100357873 | Khien Pham | 1 | 1 | T |
| 100848972 | Ngọc Trầm | 1 | 1 | T |
| 100266239 | Rei Nguyen | 1 | 1 | T |
| 100893934 | Vũ Lê | 1 | 1 | T |
| 100201300 | Chu Le Dung | 1 | 1 | F |
| 100722422 | Lương Tuấn Minh | 1 | 1 | T |
| 100020742 | Cẩm Tiên | 2 | 2 | T |
| 100738549 | Nguyễn Chi Lê | 1 | 1 | T |
| 100671693 | Duy Hoàng | 1 | 1 | T |
| 100406193 | Thanh Tâm Lưu Thị | 1 | 1 | T |
| 100923735 | Lie THANH | 1 | 1 | T |
| 100317963 | Công Minh | 1 | 1 | T |
| 100481276 | Tôn Nữ Thị Lài | 1 | 1 | T |
| 100902198 | Tuyết Nhung | 1 | 1 | T |
| 100730083 | Liên Liên Truong | 1 | 1 | T |
| 100855437 | Lê Anh Tuấn | 1 | 1 | T |
| 100517338 | Đoan Trang | 1 | 1 | T |
| 100437894 | Duong Minh Thien | 1 | 1 | T |
| 100033013 | Thảo Vi | 2 | 2 | T |
| 100951924 | Art Phạm | 1 | 1 | T |
| 100243619 | Thái Nhi | 1 | 1 | T |
| 100345920 | William Nguyễn | 1 | 1 | T |
| 100223813 | Khánh Nhật Trần | 1 | 2 | T |
| 100849255 | MI HU | 1 | 1 | T |

Bảng 3.5 Kết quả dự đoán của mô hình Naive Bayes (tỉ lệ 8:2)

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 59 | 4 | 2 | 7 |

Bảng 3.6 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 8:2)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + Độ chính xác (Precision): 0.936507937
  + Độ bao phủ (Recall): 0.893939394
  + Hệ số F1 (F1-score): 0.914728682
* Mô hình Cây quyết định (Decision Tree)

Tương tự với mô hình Logistic Regression và Naive Bayes ta cũng sẽ có 1 bảng dữ liệu và bảng giá trị TP, TN, FP, FN tương ứng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| friend\_id | friend\_name | friend | Dự đoán | So sánh |
| 100856157 | Công Lợi | 1 | 1 | T |
| 100367707 | Nguyễn Cát Khánh Phương | 1 | 1 | T |
| 100298486 | Quách Nam | 1 | 1 | T |
| 100206041 | Hoàng Gia Kts | 1 | 2 | F |
| 100591751 | Nhi Nhi | 1 | 1 | T |
| 100102913 | Duyen Tran | 1 | 2 | F |
| 100951649 | Thuy Huynh | 1 | 1 | T |
| 100537247 | Minh Như Teuk | 1 | 1 | T |
| 100161967 | Ha Linh | 1 | 2 | F |
| 100571039 | Công Thành | 1 | 1 | T |
| 100675942 | Nguyễn Ngọc Thảo Quyên | 1 | 1 | T |
| 100032369 | Luong Ngoc Loi | 2 | 2 | T |
| 100416401 | Liễu Huyền | 1 | 1 | T |
| 100341100 | Nguyễn Quốc Khánh | 1 | 1 | T |
| 100746988 | Trương Phước Kháng | 1 | 1 | T |
| 100593928 | Huỳnh Minh Huy | 1 | 1 | T |
| 100128895 | Đàm Quang Tường | 1 | 2 | F |
| 100334725 | Thy Thy | 1 | 1 | T |
| 100416367 | Tuyết Quyên | 1 | 1 | T |
| 100016746 | Quy Nguyen | 2 | 2 | T |
| 100362038 | Quynh Quynh | 1 | 1 | T |
| 100875703 | Helen Nguyễn | 1 | 1 | T |
| 100199385 | Yu Na | 1 | 2 | F |
| 100488692 | Phượng Cấn Thị | 1 | 1 | T |
| 100872241 | Trần Quốc | 1 | 1 | T |
| 100436705 | Phạm Hoàng Long | 1 | 1 | T |
| 100747859 | Van Anh | 1 | 1 | T |
| 100072317 | Trần Nguyên Đăng Vũ | 1 | 2 | F |
| 100512231 | Trần Thị Kiều Sâm | 1 | 1 | T |
| 100456770 | Văn Minh Hoan | 1 | 1 | T |
| 100343699 | Trang Lê | 1 | 1 | T |
| 100453442 | Nguyễn Cườngg | 1 | 1 | T |
| 100026949 | Trương Phi Hoàng | 2 | 2 | T |
| 100440449 | Dương Hồng Chi | 1 | 1 | T |
| 100607196 | Nguyễn Kim Tài | 1 | 1 | T |
| 100628809 | Nguyễnn Uyênn | 1 | 1 | T |
| 100884526 | Trần Hồng Giang | 1 | 1 | T |
| 100024946 | Thảo Nguyên Vũ | 2 | 2 | T |
| 100641068 | Lê Sung Chôi | 1 | 1 | T |
| 100474957 | Nhầm Duyên | 1 | 1 | T |
| 100051611 | Linh Lê | 1 | 2 | F |
| 100549204 | Nguyễn Xuân Hoài An | 1 | 1 | T |
| 100341710 | Vongoc Quynhchi | 1 | 1 | T |
| 100606378 | Pun Oăn | 1 | 1 | T |
| 100566101 | Trúc Quỳnh | 1 | 1 | T |
| 100655372 | Tuan Tran | 1 | 1 | T |
| 100679515 | Nguyễn Lê Vân | 1 | 1 | T |
| 100925729 | Thanh Hoàng Nguyễn | 1 | 1 | T |
| 100357873 | Khien Pham | 1 | 1 | T |
| 100848972 | Ngọc Trầm | 1 | 1 | T |
| 100266239 | Rei Nguyen | 1 | 1 | T |
| 100893934 | Vũ Lê | 1 | 1 | T |
| 100201300 | Chu Le Dung | 1 | 2 | F |
| 100722422 | Lương Tuấn Minh | 1 | 1 | T |
| 100020742 | Cẩm Tiên | 2 | 2 | T |
| 100738549 | Nguyễn Chi Lê | 1 | 1 | T |
| 100671693 | Duy Hoàng | 1 | 1 | T |
| 100406193 | Thanh Tâm Lưu Thị | 1 | 1 | T |
| 100923735 | Lie THANH | 1 | 1 | T |
| 100317963 | Công Minh | 1 | 1 | T |
| 100481276 | Tôn Nữ Thị Lài | 1 | 1 | T |
| 100902198 | Tuyết Nhung | 1 | 1 | T |
| 100730083 | Liên Liên Truong | 1 | 1 | T |
| 100855437 | Lê Anh Tuấn | 1 | 1 | T |
| 100517338 | Đoan Trang | 1 | 1 | T |
| 100437894 | Duong Minh Thien | 1 | 1 | T |
| 100033013 | Thảo Vi | 2 | 2 | T |
| 100951924 | Art Phạm | 1 | 1 | T |
| 100243619 | Thái Nhi | 1 | 1 | T |
| 100345920 | William Nguyễn | 1 | 1 | T |
| 100223813 | Khánh Nhật Trần | 1 | 2 | T |
| 100849255 | MI HU | 1 | 1 | T |

Bảng 3.7 Kết quả dự đoán của mô hình Cây quyết định (tỉ lệ 8:2)

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 57 | 0 | 6 | 9 |

Bảng 3.8 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 8:2)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + Độ chính xác (Precision): 1
  + Độ bao phủ (Recall): 0.863636364
  + Hệ số F1(F1-score): 0.926829268

Xét theo tập dữ liệu của người dùng được chọn là “Nguyễn Lê Hồng Nga” với tỉ lệ chia dữ liệu là 5:5, ta có được là danh sách người dùng trong tập dữ liệu kiểm tra đó là 179 người dùng có khả năng kết bạn.

* So sánh các tiêu chí đánh giá (Độ chính xác, Độ bao phủ, Hệ số F1) thì ta có như sau:

Tương tự với kết quả của tỉ lệ chia dữ liệu là 8:2, ta sẽ có các bảng kết quả giá trị dự đoán của các mô hình như sau:

* Mô hình Logistic hồi quy:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 168 | 2 | 9 | 0 |

Bảng 3.9 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 5:5)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.988235294
    - Độ bao phủ (Recall): 1
    - Hệ số F1(F1-score): 0.99408284
* Mô hình Naive Bayes:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 168 | 5 | 6 | 0 |

Bảng 3.10 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 5:5)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.971098266
    - Độ bao phủ (Recall): 1
    - Hệ số F1 (F1-score): 0.985337243
* Mô hình cây quyết định:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 168 | 1 | 10 | 0 |

Bảng 3.11 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 5:5)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.99408284
    - Độ bao phủ (Recall): 1
    - Hệ số F1(F1-score): 0.997032641

Xét theo tập dữ liệu của người dùng được chọn là “Nguyễn Lê Hồng Nga” với tỉ lệ chia dữ liệu là 2:8, ta có được là danh sách người dùng trong tập dữ liệu kiểm tra đó là 286 người dùng có khả năng kết bạn.

* So sánh các tiêu chí đánh giá (Độ chính xác, Độ bao phủ, Hệ số F1) thì ta có như sau:

Tương tự với kết quả của tỉ lệ chia dữ liệu là 8:2, ta sẽ có các bảng kết quả giá trị dự đoán của các mô hình như sau:

* Mô hình Logistic hồi quy:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 271 | 9 | 6 | 0 |

Bảng 3.12 Kết quả giá trị dự đoán của mô hình Logistic hồi quy (tỉ lệ 2:8)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.967857143
    - Độ bao phủ (Recall): 1
    - Hệ số F1(F1-score): 0.983666062
* Mô hình Naive Bayes:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 271 | 11 | 4 | 0 |

Bảng 3.13 Kết quả giá trị dự đoán của mô hình Naive Bayes (tỉ lệ 2:8)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.960992908
    - Độ bao phủ (Recall): 1
    - Hệ số F1(F1-score): 0.980108499
* Mô hình cây quyết định:

|  |  |  |  |
| --- | --- | --- | --- |
| TP | FP | TN | FN |
| 271 | 9 | 6 | 0 |

Bảng 3.14 Kết quả giá trị dự đoán của mô hình Cây quyết định (tỉ lệ 2:8)

Thông qua bảng giá trị trên ta sẽ tính được các tiêu chí đánh giá như sau:

* + - Độ chính xác (Precision): 0.967857143
    - Độ bao phủ (Recall): 1
    - Hệ số F1(F1-score): 0.983666062

Từ các kết quả tính toán trên ta có được bảng tổng hợp kết quả và biểu đồ biễu diễn cho từng tiêu chí đánh giá như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Tiêu chí | Logistic hồi quy | Naive Bayes | Cây quyết định |
| Độ chính xác (1) | 1 | 0.936507937 | 1 |
| Độ bao phủ (1) | 0.878787879 | 0.893939394 | 0.863636364 |
| Hệ số F1 (1) | 0.935483871 | 0.914728682 | 0.926829268 |
| Độ chính xác (2) | 0.988235294 | 0.971098266 | 0.99408284 |
| Độ bao phủ (2) | 1 | 1 | 1 |
| Hệ số F1 (2) | 0.99408284 | 0.985337243 | 0.997032641 |
| Độ chính xác (3) | 0.967857143 | 0.960992908 | 0.967857143 |
| Độ bao phủ (3) | 1 | 1 | 1 |
| Hệ số F1 (3) | 0.983666062 | 0.980108499 | 0.983666062 |

Bảng 3.15 Kết quả hiệu suất của các mô hình phân lớp

Trong đó (1) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 8:2.

(2) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 5:5.

(3) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 2:8.

A graph with numbers and a number of numbers

Description automatically generated with medium confidence

Hình 3.10 Biểu đồ về độ chính xác của các mô hình phân lớp

Độ chính xác (Precision): Đây là tỷ lệ giữa số lượng các dự đoán đúng thuộc một lớp cụ thể và tổng số dự đoán của lớp đó.

* Logistic Hồi quy và Cây Quyết định cho thấy hiệu suất tốt nhất ở tỉ lệ 8:2, với độ chính xác tuyệt đối. Tuy nhiên, khi dữ liệu huấn luyện giảm, độ chính xác của chúng có xu hướng giảm.
* Naive Bayes cho thấy sự ổn định trong độ chính xác trên các tỉ lệ khác nhau, dù không đạt được hiệu suất cao nhất như hai mô hình kia.

A graph with different colored bars

Description automatically generated

Hình 3.11 Biểu đồ về độ bao phủ của các mô hình phân lớp

Độ bao phủ (Recall): Đây là tỷ lệ giữa số lượng dự đoán đúng thuộc một lớp cụ thể và tổng số mẫu thuộc lớp đó trong tập dữ liệu.

* Naive Bayes và Logistic Hồi quy thể hiện sự ổn định tốt hơn ở tỉ lệ 8:2 so với Cây Quyết định, cho thấy chúng có thể phát hiện các trường hợp tích cực hiệu quả hơn khi dữ liệu huấn luyện chiếm ưu thế.
* Ở hai tỉ lệ còn lại (5:5 và 2:8), cả ba mô hình đều đạt độ bao phủ là 1.0, cho thấy chúng đều có khả năng phát hiện tất cả các trường hợp tích cực khi tập dữ liệu cân bằng hoặc dữ liệu kiểm tra nhiều hơn.

A graph of numbers and a number of numbers

Description automatically generated with medium confidence

Hình 3.12 Biểu đồ về hệ số F1 của các mô hình phân lớp

Hệ số F1 (F1-score): Đây là một trung bình điều hòa giữa Precision và Recall.

* Logistic Hồi quy và Cây Quyết định thể hiện sự ổn định và cân bằng tốt nhất giữa độ chính xác và độ bao phủ qua các tỉ lệ khác nhau như ở tỉ lệ 8:2 thì Logistic Hồi quy có hệ số F1 cao nhất, ở tỉ lệ 5:5 thì Câu quyết định có hệ số F1 cao nhất còn ở tỉ lệ 2:8 thì Logistic Hồi quy và Cây Quyết định có hệ số F1 bằng nhau
* Naive Bayes cũng thể hiện sự ổn định nhưng không vượt trội như hai mô hình kia nên kết quả thường xếp sau 2 mô hình còn lại.
* So sánh thời gian xử lý của từng mô hình phân lớp (thời gian huấn luyện và thời gian dự đoán) sẽ tiến hành thêm các hàm đếm thời gian ở hàm *train* và hàm *predict* trong class *ClassifierModel*

Kết quả ta có được như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| Mô hình phân lớp | Logistichồi quy | Naive Bayes | Cây quyết định |
| Thời gian huấn luyện (1) | 0.1481 | 0.1389 | 0.4057 |
| Thời gian dự đoán (1) | 0.4235 | 0.55 | 0.33 |
| Thời gian huấn luyện (2) | 0.1655 | 0.0728 | 0.2035 |
| Thời gian dự đoán (2) | 1.0627 | 0.9816 | 0.9188 |
| Thời gian huấn luyện (3) | 0.0623 | 0.0295 | 0.0954 |
| Thời gian dự đoán (3) | 2.3891 | 2.2183 | 1.9925 |

Bảng 3.16 Kết quả thời gian xử lý của các mô hình phân lớp

Trong đó (1) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 8:2.

(2) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 5:5.

(3) là trường hợp chạy với tỉ lệ dữ liệu huấn luyện và kiểm tra là 2:8.A graph of numbers and a bar

Description automatically generated with medium confidence

Hình 3.13 Biểu đồ mô tả thời gian huấn luyện của các mô hình phân lớp

* Thời gian huấn luyện: Ở tất cả các tỉ lệ dữ liệu, Naive Bayes luôn có thời gian huấn luyện nhanh nhất, tiếp theo là Logistic Hồi quy và Cây Quyết định luôn có thời gian huấn luyện chậm nhất.

A graph of numbers and a bar

Description automatically generated with medium confidence

Hình 3.14 Biểu đồ mô tả thời gian dự đoán của các mô hình phân lớp

* Thời gian dự đoán: Cây Quyết định luôn có thời gian dự đoán nhanh nhất ở tất cả các tỉ lệ dữ liệu, tiếp theo là Naive Bayes và Logistic Hồi quy có thời gian dự đoán chậm nhất trong các trường hợp, đặc biệt rõ rệt ở tỉ lệ 2:8.

**Nhận xét:**

Logistic hồi quy thường có độ chính xác và F1-score cao, đặc biệt là khi tỷ lệ huấn luyện-kiểm tra là 8:2 và 5:5. Tuy nhiên, thời gian dự đoán của nó có thể chậm hơn so với các mô hình khác khi tỷ lệ huấn luyện-kiểm tra là 2:8.

Naive Bayes có thời gian huấn luyện nhanh nhất trong tất cả các tỉ lệ dữ liệu và có độ chính xác, F1-score tương đối cao, nhưng thấp hơn so với Logistic hồi quy và Cây quyết định trong một số trường hợp.

Cây quyết định có độ chính xác và F1-score cao, đặc biệt là khi tỷ lệ huấn luyện-kiểm tra là 5:5. Thời gian dự đoán của nó cũng là nhanh nhất trong nhiều trường hợp, nhưng thời gian huấn luyện có thể dài hơn so với Naive Bayes và Logistic hồi quy.

Dựa trên các kết quả trên, lựa chọn mô hình phụ thuộc vào các ưu tiên cụ thể:

* Nếu ưu tiên là thời gian huấn luyện nhanh, Naive Bayes là lựa chọn tốt.
* Nếu ưu tiên là độ chính xác và F1-score cao, Logistic hồi quy hoặc Cây quyết định có thể là lựa chọn tốt hơn.
* Nếu ưu tiên là thời gian dự đoán nhanh, Cây quyết định thường là lựa chọn tốt nhất.

# KẾT QUẢ VÀ HƯỚNG PHÁT TRIỂN

1. **Kết quả**

* Kết quả về mặt lý thuyết

Tìm hiểu về cách biểu diễn dữ liệu mạng xã hội dưới dạng đồ thị.

Tìm hiểu về cơ sở dữ liệu đồ thị Neo4j, ngôn ngữ truy vấn Cypher.

Tìm hiểu bài toán dự đoán liên kết trên mạng xã hội.

Tìm hiểu các mô hình phân lớp Hồi quy Logistic (Logistic Regression), Naive Bayes và Cây quyết định (Decision Tree).

* Kết quả về mặt ứng dụng

Ứng dụng có các chức năng:

- Kết nối cơ sở dữ liệu đồ thị Neo4j.

- Sử dụng mô hình phân lớp để dự đoán liên kết của từng User.

- Khuyến nghị kết bạn dựa trên các mối quan hệ trong xã hội của từng User.

- Chương trình đã xây dựng mối quan hệ dựa trên các mối quan hệ xã hội như: mối quan hệ bạn bè, mối quan hệ cùng nơi làm việc, cùng nơi sinh sống.

1. **Hạn chế**

Do hạn chế về mặt thời gian, kiến thức cũng như kinh nghiệm của bản thân, chương trình còn một số khuyết điểm:

- Chưa cải tiến được gì thêm trong thuật toán của các mô hình phân lớp.

- Giao diện khá đơn giản.

- Khai phá ít cơ sở dữ liệu, chưa khai thác hết các mối quan hệ trên mạng xã hội.

1. **Hướng phát triển**

**-** Nghiên cứu cải tiến và cài đặt biểu diễn mối quan hệ trên các bộ dữ liệu lớn hơn trong cơ sở dữ liệu đồ thị Neo4j.

- Nghiên cứu kết bạn dựa trên các mối quan hệ khác trong xã hội của từng User như cùng sở thích, cùng độ tuổi, cùng giới tính, cùng mối quanhệ huyết thống… nhằm đưa ra khuyến nghị chính xác cho người dùng.

# DANH MỤC TÀI LIỆU THAM KHẢO

|  |  |
| --- | --- |
|  | “Hãy là những người sử dụng mạng xã hội thông minh”, Công thông tin điện tử quận Bắc Từ Liêm, https://bactuliem.hanoi.gov.vn/xa-hoi/-/asset\_publisher/pRkZE481hymd/content/hay-la-nhung-nguoi-su-dung-mang-xa-hoi-thong-minh <https://neo4j.com/docs/> [Accessed May 28, 2024] |
|  | Govinda K, Rajkumar Rajasekaran, V.S.R.P. Venkata Krishna Varun, Davya Vuyyuru, Battula Thirumaleshwari Devi. "Link Prediction in Social Networks using Machine Learning.", 2023 |
|  | Ved Prakash Chaubey, Ajay Sharma, Tushar Sharma, Shamneesh Sharma, Aman Kumar. "Link Prediction for Social Network Analysis Using Random Forest and XG-Boost Algorithm.", 2023 |
|  | Fateme Mohamady, Sina Dami. "Link Prediction in Dynamic Social Networks Using Deep Learning.", 2024 |
|  | “What is a Graph Database?”, Neo4j the Graph database,  http://fooo.fr/~vjeux/github/github-recommandation/db/doc/manual/html/what-is-a-graphdb.html [Accessed May 28, 2024]. |
|  | “Logistic Regression”, Machine Learning cơ bản,  https://machinelearningcoban.com/2017/01/27/logisticregression/#mo-hinh-logistic-regression [Accessed May 28, 2024]. |
|  | "Naive Bayes Classifier", Machine Learning cơ bản,  https://machinelearningcoban.com/2017/08/08/nbc/ [Accessed May 28, 2024]. |
|  | "Decision Trees", Machine Learning cơ bản,  https://machinelearningcoban.com/2018/01/14/id3/ [Accessed May 28, 2024]. |
|  | “Các phương pháp đánh giá một mô hình phân lớp dữ liệu”, Học trực tuyến,  https://hoctructuyen123.net/cac-phuong-phap-danh-gia-mot-mo-hinh-phan-lop-du-lieu/ [Accessed May 28, 2024]. |
|  | “Đồ thị API”, Meta for Developers, https://developers.facebook.com/docs/graph-api/ [Accessed May 28, 2024]. |