

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Nhóm dự án**

**XÂY DỰNG HỆ THỐNG THEO DÕI**

**DỊCH BỆNH COVID-19 TẠI VIỆT NAM**

**DỰ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành**: **Hệ thống thông tin**

**HÀ NỘI - 2020**

**ĐẠI HỌC QUỐC GIA HÀ NỘI**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

**Nhóm dự án**

**XÂY DỰNG HỆ THỐNG THEO DÕI**

**DỊCH BỆNH COVID-19 TẠI VIỆT NAM**

**DỰ ÁN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY**

**Ngành: Hệ Thống Thông Tin**

**Cán bộ hướng dẫn: TS. Trần Mai Vũ**

**HÀ NỘI – 2020**

# Các thành viên trong nhóm:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên** | **Mã số sinh viên** | **Ghi chú** |
| 1 | Đỗ Văn Vinh | 14020556 | Nhóm trưởng |
| 2 | Phan Thế Minh | 16021314 |  |
| 3 | Nguyễn Trường Sơn | 16021332 |  |
| 4 | Đoàn Thị Thoa | 16022131 |  |
| 5 | Hoàng Văn Chính | 16028055 |  |
| 6 | Hoàng Mạnh Dũng | 16020899 |  |

# 

# Phân công công việc:

|  |  |
| --- | --- |
| **Công việc** | **Người làm** |
| Thu thập, xử lý, phân tích dữ liệu, thiết kế cơ sở dữ liệu | Đỗ Văn Vinh Đoàn Thị Thoa |
| Back-end | Phan Thế Minh Hoàng Mạnh Dũng |
| Front-end | Nguyễn Trường Sơn  Hoàng Văn Chính |

**TÓM TẮT**

Đại dịch Covid19 là một đại dịch bệnh truyền nhiễm với tác nhân là SARS-CoV-2 khởi nguồn vào tháng 12 năm 2019 với tâm dịch đầu tiên được ghi nhận là thành phố Vũ Hán thuộc miền trung Trung Quốc. Đến nay, chỉ sau khoảng 6 tháng sau ca mắc đầu tiên số ca mắc đã lên tới gần 8 triệu ca nhiễm và hơn 400.000 người tử vong, được ghi nhận trên 215 quốc gia và vùng lãnh thổ. Việc gia tăng không ngừng các ca nhiễm có ảnh hưởng không nhỏ đến nền kinh tế. Từ đó nhu cầu cập nhật diễn biến về tình hình Covid19 và những phân tích ảnh hưởng cũng gia tăng. Ngay cả những quốc gia có nền kinh tế, y tế rất phát triển như Mỹ , Anh hay Trung Quốc cũng đang rơi vào tình trạng khủng hoảng do nền kinh tế đang bị tàn phá nặng nề bởi Covid19. Trong khi đó Việt Nam là một nước có đường biên giới trên bộ dài khoảng 4.924 km tiếp giáp với Trung Quốc ở phía Bắc, do vậy việc ngăn chặn và phòng chống dịch ở nước ta càng trở nên khó khăn hơn bao giờ hết. Để có thể kiểm soát dịch chỉ với khoảng 300 ca nhiễm và 0 ca tử vong là cả sự đánh mất mát về sức người, sức của. Do vậy, trong đề tài khóa luận này nhóm em muốn xây dựng hệ thống để mọi người có thể theo dõi dịch bệnh Covid19 một cách dễ dàng và trực quan nhất đồng thời cũng đưa ra một số đánh giá liên quan đến khủng hoảng kinh tế có thể xảy ra tại Việt Nam dựa trên việc phân tích thống kê dữ liệu.

***Từ khóa***:  Covid19, thống kê dữ liệu

# LỜI CAM ĐOAN

Chúng tôi xin cam đoan đây là công trình nghiên cứu của nhóm chúng tôi, với sự hỗ trợ từ người thầy hướng dẫn khoa học TS Trần Mai Vũ.

Product hoàn toàn là sản phẩm do nhóm làm ra không sao tài liệu, công trình nghiên cứu của người khác. Việc sử dụng những dữ liệu và bài viết có trong luận văn được thu thập từ các nguồn thông tin khác nhau có ghi trong phần tài liệu tham khảo.

Nếu phát hiện bất kỳ sự gian lận nào, tôi xin chịu trách nhiệm trước Hội đồng, cũng như kết quả luận văn của mình.

Hà Nội, tháng 6 năm 2020

Sinh viên

Nhóm sinh viên

# LỜI CẢM ƠN

Chúng tôi muốn bày tỏ lòng biết ơn sâu sắc tới những người đã giúp đỡ chúng tôi trong quá trình làm luận văn, đặc biệt chúng tôi xin cám ơn TS Trần Mai Vũ, với lòng kiên trì, thầy đã chỉ bảo chúng tôi chi tiết và cho chúng tôi những định hướng quý báu trong từng bước của quá trình làm dự án. Đồng thời chúng tôi cũng xin gửi lời cảm ơn tới các giảng viên khoa Công nghệ thông tin – Trường Đại học Công nghệ - Đại học Quốc gia Hà Nội đã truyền đạt các kiến thức cho chúng tôi trong suốt thời gian học tập và nghiên cứu để có thể hoàn thành dự án này.

Chúng tôi cũng xin chân thành cảm ơn cơ quan, bạn bè, đồng nghiệp, gia đình và những người thân đã cùng chia sẻ, giúp đỡ, động viên, tạo mọi điều kiện thuận lợi để chúng tôi hoàn thành nhiệm vụ học tập và hoàn thành dự án này.

Hà Nội, tháng 06 năm 2020

Mục lục

[Các thành viên trong nhóm: 3](#_Toc43415419)

[Phân công công việc: 3](#_Toc43415420)

[LỜI CAM ĐOAN 2](#_Toc43415421)

[MỤC LỤC 4](#_Toc43415422)

[CHƯƠNG I: GIỚI THIỆU BÀI TOÁN 1](#_Toc43415423)

[1. Tính cấp thiết của đề tài 1](#_Toc43415424)

[2. Mục đích nghiên cứu 2](#_Toc43415425)

[3. Đối tượng nghiên cứu 2](#_Toc43415426)

[4. Phạm vi nghiên cứu 2](#_Toc43415427)

[CHƯƠNG 2: KHÁI QUÁT VỀ KHAI PHÁ DỮ LIỆU VÀ GIỚI THIỆU CÁC KỸ THUẬT ÁP DỤNG TRONG HỆ THỐNG 2](#_Toc43415428)

[2.1. Giới thiệu về khai phá dữ liệu 2](#_Toc43415429)

[2.1.1 Tại sao khai phá dữ liệu lại trở nên quan trọng? 2](#_Toc43415430)

[2.1.2 Tại sao cần khai phá dữ liệu 3](#_Toc43415431)

[2.1.3 Sơ lược về Khai phá Dữ liệu 4](#_Toc43415432)

[2.1.4. Quy trình CRISP-DM áp dụng trong dự án: 6](#_Toc43415433)

[2.2. Phân cụm dữ liệu và ứng dụng 8](#_Toc43415434)

[2.2.1 Mục đích của phân cụm dữ liệu 9](#_Toc43415435)

[2.2.2. Thuật toán phân cụm K-means 10](#_Toc43415436)

[2.3. Phân lớp 12](#_Toc43415437)

[2.3.1 Tìm hiểu chung về phân lớp 12](#_Toc43415438)

[2.3.2 Các phương pháp phân lớp 16](#_Toc43415439)

[2.3.3 Mô hình Deep Neural Network 16](#_Toc43415440)

[2.4. Mô hình hồi quy tuyến tính (Linear Regression Model) 17](#_Toc43415441)

[2.5. Mạng thần kinh nhân tạo (Artificial neural network) 18](#_Toc43415442)

[2.5.1 Khái niệm 18](#_Toc43415443)

[2.5.2 Mô hình kiến trúc mạng ANN 18](#_Toc43415444)

[2.6. Xử lý ngôn ngữ tự nhiên (Natural Language Processing) 19](#_Toc43415445)

[2.6.1 Xử lý ngôn ngữ tự nhiên là gì? 19](#_Toc43415446)

[2.6.2 Ứng dụng của xử lý ngôn ngữ tự nhiên vào giám sát mạng xã hội 20](#_Toc43415447)

[2.6.3 Topic modeling: 21](#_Toc43415448)

[2.6.4 Bài toán phân loại văn bản (Text classification) 21](#_Toc43415449)

[2.7. Các công nghệ sử dụng: 28](#_Toc43415450)

[2.7.1. Anaconda 28](#_Toc43415451)

[2.7.2. Python3 29](#_Toc43415452)

[2.7.3. IDE: Jupyter NoteBook 30](#_Toc43415453)

[CHƯƠNG 3: PHƯƠNG PHÁP THU THẬP, ĐÁNH GIÁ, PHÂN TÍCH DỮ LIỆU VÀ QUÁ TRÌNH TRIỂN KHAI HỆ THỐNG THEO DÕI DỊCH BỆNH COVID-19 TẠI VIỆT NAM 31](#_Toc43415454)

[3.1. Xây dựng ý tưởng, xác định các chức năng cơ bản của hệ thống 31](#_Toc43415455)

[3.1.1 Xây dựng ý tưởng 31](#_Toc43415456)

[3.1.2 Các mục tiêu, tính năng cơ bản: 32](#_Toc43415457)

[3.1.3 Các mục tiêu, tính năng nâng cấp, phát triển: 32](#_Toc43415458)

[3.2. Thu thập dữ liệu 32](#_Toc43415459)

[3.2.1. Cơ sở lý thuyết: 32](#_Toc43415460)

[3.2.2. Triển khai thực tế 35](#_Toc43415461)

[3.3. Xử lý dữ liệu thô: 36](#_Toc43415462)

[3.3.1. Nhận xét tổng quan: 36](#_Toc43415463)

[3.3.2. Tiền xử lý dữ liệu: 37](#_Toc43415464)

[3.4. Phương pháp thực hiện: 37](#_Toc43415465)

[3.4.1. Tiền xử lý dữ liệu (Preprocessing Data) 38](#_Toc43415466)

[3.4.2. Vector hóa văn bản (Feature engineering) 40](#_Toc43415467)

[3.4.5. Xây dựng mô hình phân lớp văn bản 43](#_Toc43415468)

[3.6. Xây dựng hệ thống 44](#_Toc43415469)

[3.6.1 Thiết kế cơ sở dữ liệu 44](#_Toc43415470)

[3.6.2. Back-end 45](#_Toc43415471)

[3.6.3. Front-end 55](#_Toc43415472)

[CHƯƠNG 4: THỰC NGHIỆM VÀ CÁC ĐÁNH GIÁ LIÊN QUAN ĐẾN TÌNH HÌNH KINH TẾ CÓ THỂ ẢNH HƯỞNG TẠI THỊ TRƯỜNG VIỆT NAM 67](#_Toc43415473)

[KẾT LUẬN 68](#_Toc43415474)

[TÀI LIỆU THAM KHẢO 69](#_Toc43415475)

# CHƯƠNG I: GIỚI THIỆU BÀI TOÁN

## 1. Tính cấp thiết của đề tài

Bệnh viêm đường hô hấp cấp do chủng mới của virus corona được phát hiện lần đầu tiên tại thành phố Vũ Hán, tỉnh Hồ Bắc, Trung Quốc vào tháng 12/2019. Sau đó, dịch đã lan ra nhiều tỉnh thành của Trung Quốc. Đến nay, tất cả quốc gia / vùng lãnh thổ trên toàn cầu đều đã ghi nhận trường hợp mắc. Trong đó nước Mỹ đang là nước có số người nhiễm nhiều nhất.

Đến nay đã có gần 8 triệu người mắc bệnh, trong đó hơn 400.000 người đã tử vong. Qua dịch bệnh lần này cho ta thấy rõ bộ mặt thực tế của y tế toàn cầu là yếu kém như thế nào khi đối mặt với dịch bệnh mới mang tính toàn cầu. Đặc biệt là các nước phương Tây và Mỹ khi họ luôn tự hào mình là các nước phát triển và có nền kĩ thuật tiên tiến nhưng lại gánh chịu hậu quả nặng nề nhất tính đến thời điểm hiện tại.

Điển hình như Mỹ, do các bước đi thiếu quyết đoán của chính phủ đã khiến nước này nhanh chóng trở thành quốc gia có số người bị mắc nhiều nhất thế giới. Kéo theo đó là các hệ lụy liên quan khác như kinh tế, chính trị, xã hội, v.v… Đỉnh điểm nhất là các vụ bạo loạn diễn ra gần đây đã khiến nước Mỹ diễn ra bạo động trên toàn quốc, tàn phá nền kinh tế đã kiệt quệ do hậu quả của dịch bệnh.

Cổ nhân có câu: “phòng bệnh hơn chữa bệnh”. Vì vậy, việc theo dõi dịch bệnh COVID-19 là việc cấp thiết hiện nay để mọi người có thể đánh giá chi tiết và toàn diện về dịch bệnh và các hậu quả nó có thể gây ra để phòng tránh và khắc phục hậu quả tốt nhất. Ngày nay công nghệ thông tin đang ngày một tiếp cận đến nhiều người dân hơn bao giờ hết, cùng với đó là mọi người có xu hướng ở nhà tránh tiếp xúc nhiều nhất có thể nên một hệ thống theo dõi dịch COVID-19 và các đánh giá liên quan trên nền tảng ứng dụng web là điều mà nhiều người mong muốn có được để dễ dàng theo dõi cũng như tìm biện pháp đối phó cho riêng mình cũng như toàn xã hội. Đó cũng chính là lý do chúng tôi chọn đề tài: ***“XÂY DỰNG HỆ THỐNG THEO DÕI DỊCH BỆNH COVID-19 VÀ CÁC ĐÁNH GIÁ LIÊN QUAN ĐẾN KHỦNG HOẢNG KINH TẾ CÓ THỂ XẢY RA TẠI VIỆT NAM”*** để xây dựng, nghiên cứu phát triển và ứng dụng vào thực tiễn.

## 2. Mục đích nghiên cứu

Xây dựng một trang web thuần Việt bao gồm các bản đồ tình số ca bệnh, thống kê các bài báo chính thống cũng như đưa ra một số biểu đồ báo cáo cung cấp tri thức cho cá nhân và doanh nghiệp. Các bài báo được cập nhật theo thời gian thực và lấy từ các nguồn chính thống đảm bảo cung cấp thông tin chính xác kịp thời. Giao diện thuần Việt với biểu đồ, bản đồ và số liệu giúp thân thiện với người dùng.

## 3. Đối tượng nghiên cứu

Đối tượng nghiên cứu: dịch bệnh COVID-19 và các đánh giá liên quan đến khủng hoảng kinh tế có thể xảy ra.

## 4. Phạm vi nghiên cứu

Phạm vi nghiên cứu: chủ yếu nghiên cứu trên phạm vi Việt Nam, một phần số liệu về ca mắc, hồi phục và tử vong trên phạm vi thế giới.

# CHƯƠNG 2: KHÁI QUÁT VỀ KHAI PHÁ DỮ LIỆU

# VÀ GIỚI THIỆU CÁC KỸ THUẬT ÁP DỤNG TRONG HỆ THỐNG

## 2.1. Giới thiệu về khai phá dữ liệu

### 2.1.1 Tại sao khai phá dữ liệu lại trở nên quan trọng?

Ngày nay số lượng người sử dụng các thiết bị thông minh như smartphone, tablet hay PC, laptop có kết nối Internet để tìm kiếm thông tin, giải trí, trò chuyện, mua sắm… trên toàn thế giới đang gia tăng nhanh chóng. Ngoài ra sự xuất hiện của thuật ngữ IoT miêu tả sự kết nối giữa tất cả các thiết bị với nhau bằng Internet, cho phép trao đổi, truyền tải dữ liệu. IoT hỗ trợ con người rất nhiều lĩnh vực. Các công ty cũng ứng dụng IoT trong các hoạt động kinh doanh, sản xuất với mục đích tìm kiếm cơ hội gia tăng lợi nhuận, phân tích ngày càng lớn, từ dữ liệu người dùng, dữ liệu khách hàng đến dữ liệu thị trường, dữ liệu sản xuất, … đa dạng và phức tạp hơn. Theo tập đoàn công nghệ Cisco, khối lượng Big Data dự báo trong những năm tới, hay trong năm 2020 có thể đạt ơn 500 Zettabytes một năm. Bên cạnh đó khả năng thu thập và lưu trữ thông tin của các Hệ thống Thông tin cũng tăng nhanh một cách chóng mặt. Hơn nữa, việc áp dụng tin học vào các hoạt động sản xuất, kinh doanh và nhiều lĩnh vực hoạt động khác đã tạo ra cho chúng ta một lượng dữ liệu lưu trữ khổng lồ. Sự bùng nổ này đã dẫn tới một yêu cầu cấp thiết là cần có những kỹ thuật cũng như công cụ mới để chuyển đổi lượng dữ liệu khổng lồ kia thành những tri thức có ích với con người. Từ đó, các kỹ thuật Khai phá Dữ liệu đã trở thành một lĩnh vực quan trọng trong nền CNTT thế giới hiện nay nói chung và Việt Nam nói riêng.

Nguồn dữ liệu Big Data là nguồn lực quan trọng nhưng để tận dụng hiệu quả dữ liệu để đạt được giá trị thì Data mining là công cụ không thể thiếu, nó giúp chúng ta hiểu được các tập dữ liệu đang thể hiện cái gì, đang cung cấp các thông tin, kiến thức hữu ích nào, ...

### 2.1.2 Tại sao cần khai phá dữ liệu

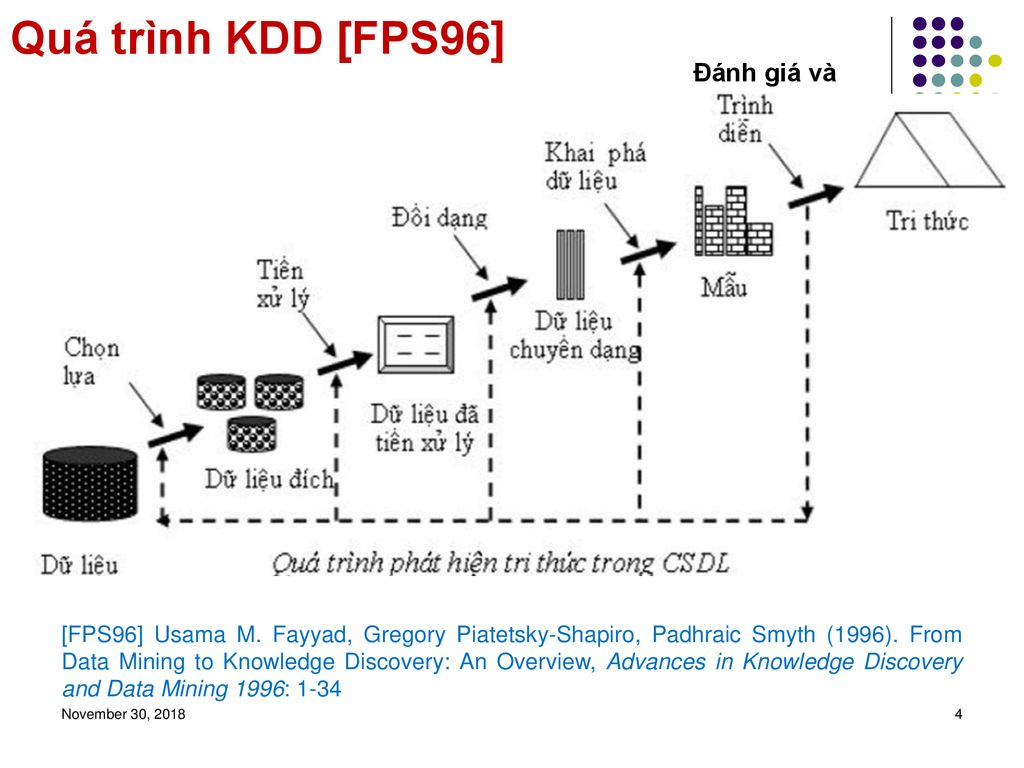
Từ khi bắt đầu đại dịch Covid19 (tháng 12/2019) cho đến nay lượng thông tin về đại dịch này tăng nhanh một cách chóng mặt. Trên khắp các trang báo điện tử hay mạng xã hội Covid19 luôn là đề tài nóng bỏng. Chính vì vậy mà hiện nay lượng dữ liệu mà về Covid19 thu thập và lưu trữ được trong các kho dữ liệu là rất lớn, nhiều khi vượt quá khả năng quản lý. Do đó xảy ra tình trạng khủng hoảng phân tích dữ liệu tác nghiệp để cung cấp thông tin với yêu cầu chất lượng ngày càng cao cho những người ra quyết định tài chính, thương mại, khoa học trong tình hình dịch bệnh diễn biến phức tạp… Đúng như John Naisbett đã cảnh báo “Chúng ta đang chìm ngập trong dữ liệu mà vẫn đói tri thức”. Hay Jim Gray (chuyên gia của Microsoft, giải thưởng Turing 1998) cũng đã từng nói “Chúng ta đang ngập trong dữ liệu khoa học, dữ liệu y tế, dữ liệu nhân khẩu học, dữ liệu tài chính, và các dữ liệu tiếp thị. Con người không có đủ thời gian để xem xét dữ liệu như vậy. Sự chú ý của con người đã trở thành nguồn tài nguyên quý giá. Vì vậy, chúng ta phải tìm cách tự động phân tích dữ liệu, tự động phân loại nó, tự động tóm tắt nó, tự động phát hiện và mô tả các xu hướng trong nó, và tự động chỉ dẫn các dị thường”.

Với khối lượng dữ liệu tăng nhanh và khổng lồ như vậy rõ ràng các phương pháp thủ công truyền thống áp dụng để phân tích dữ liệu sẽ không hiệu quả, tốn kém và dễ dẫn đến những sai lệch. Do đó để có thể khai phá hiệu quả các dữ liệu lớn cần những kỹ thuật mới, các kỹ thuật khai phá dữ liệu Data mining. Khai phá dữ liệu là một lĩnh vực khoa học nhằm tự động hóa khai thác thông tin, tri thức hữu ích, tiềm ẩn trong các CSDL từ đó cung cấp cái nhìn đa chiều tổng quan và chính xác cho cá nhân, tổ chức. Các kết quả nghiên cứu cùng với những ứng dụng thành công trong khai phá dữ liệu, khám phá tri thức cho thấy khai phá dữ liệu là một lĩnh vực tiềm năng, mang lại nhiều lợi ích. Do đó nhóm đã chọn hướng khai phá dữ liệu để đưa ra cái nhìn tổng quan, chính xác cho mọi người trong thời gian đại dịch đang diễn ra này.

### 2.1.3 Sơ lược về Khai phá Dữ liệu

Khai phá dữ liệu là quá trình khảo sát và phân tích một khối lượng lớn các dữ liệu được lưu trữ trong các CSDL cũng như kho dữ liệu, … để từ đó trích xuất ra các thông tin và tri thức quan trọng, có giá trị tiềm ẩn bên trong. Những dữ liệu này được thu từ nhiều nguồn: Từ các phần mềm hệ thống, nghiệp vụ và các phần mềm khác như: tài chính, kế toán, các hệ thống quản lý tài nguyên doanh nghiệp, các hệ thống quản lý khách hàng hay từ các công cụ lưu trữ thông tin khác. Đây là những khối dữ liệu lớn nhưng những thông tin mà nó nắm giữ vẫn chưa được thể hiện ra rõ ràng với người dùng. Kích thước của khối dữ liệu là rất lớn và tăng nhanh với một tốc độ chóng mặt chiếm nhiều dung lượng lưu trữ. Khai phá dữ liệu là ngành khoa học giúp trích xuất ra các mẫu điển hình có giá trị và biến chúng thành những tri thức hữu ích.

Quá trình này bao gồm một số bước được thể hiện trong hình dưới

******

Hình 1.0.1 *Các bước xây dựng một hệ thống khai phá dữ liệu*

Khi tiến hành xây dựng một hệ thống khai phá dữ liệu như Hình 1.0.2, ta thấy rằng để khai phá dữ liệu thì dữ liệu phải được chuẩn bị bởi các bước và các kỹ thuật khác nhau và sẽ ảnh hưởng đến kết quả của hệ thống.

Các bước được trình bày như sau:

* **Thu thập dữ liệu**: là bước đầu tiên trong quá trình khai phá dữ liệu. Đây là bước được khai thác trong một cơ sở dữ liệu, một kho dữ liệu và thậm chí các dữ liệu từ các nguồn ứng dụng.
* **Trích lọc dữ liệu**: Ở bước này các dữ liệu liên quan trực tiếp đến nhiệm vụ của quá trình sẽ được thu thập từ các nguồn dữ liệu ban đầu.
* **Tiền xử lý dữ liệu**: có nhiệm vụ làm sạch, loại bỏ nhiễu, rút gọn và rời rạc hóa dữ liệu. Giai đoạn này là giai đoạn hay bị sao lãng, nhưng thực tế nó là một bước rất quan trọng trong quá trình khai phá dữ liệu. Một số lỗi thường mắc phải trong khi gom dữ liệu là tính không đủ chặt chẽ, logic. Vì vậy, dữ liệu thường chứa các giá trị vô nghĩa và không có khả năng kết nối dữ liệu. Giai đoạn này sẽ tiến hành xử lý những dạng dữ liệu không chặt chẽ nói trên. Những dữ liệu dạng này được xem như thông tin dư thừa, không có giá trị. Bởi vậy, đây là một quá trình rất quan trọng vì dữ liệu này nếu không được “làm sạch – tiền xử lý” thì sẽ gây nên những kết quả sai lệch nghiêm trọng.
* **Chuyển đổi dữ liệu**: nhằm chuẩn hóa và làm mịn dữ liệu để chuyển dữ liệu về dạng thuận lợi nhất phục vụ cho việc khai phá.
* **Khai phá dữ liệu**: dùng các kỹ thuật phân tích để khai thác dữ liệu, trích chọn các mẫu thông tin cần thiết… Công đoạn này được xem là mất thời gian nhất và cũng là quan trọng nhất trong cả quá trình.
* **Đánh giá và biểu diễn tri thức**: Các thông tin và mối liên hệ giữa chúng vừa khám phá trong công đoạn trước được biểu diễn dưới các dạng trực quan đồng thời được đánh giá theo những tiêu chí nhất định.

Đó là 6 giai đoạn của quá trình khám phá tri thức, trong đó giai đoạn 5 - **khai phá dữ liệu (hay còn được gọi là Data Mining)** là giai đoạn được quan tâm nhiều nhất.

**Khai phá dữ liệu** là một giai đoạn quan trọng trong quá trình khám phá tri thức. Về bản chất là giai đoạn duy nhất tìm ra được thông tin mới, thông tin tiềm ẩn có trong cơ sở dữ liệu chủ yếu phục vụ cho mô tả và dự đoán.

Mô tả dữ liệu là tổng kết hoặc diễn tả những đặc điểm chung của những thuộc tính dữ liệu trong kho dữ liệu mà con người có thể hiểu được.

Dự đoán là dựa trên những dữ liệu hiện thời để dự đoán những quy luật được phát hiện từ các mối liên hệ giữa các thuộc tính của dữ liệu, trên cơ sở đó chiết xuất ra các mẫu, dự đoán được những giá trị chưa biết hoặc những giá trị tương lai của các biến quan tâm.

### 2.1.4. Quy trình CRISP-DM áp dụng trong dự án:

Trước đây chúng ta thường triển khai các hoạt động Data mining một cách không hiệu quả, mơ hồ, áp dụng những quy trình, phương pháp không hợp lý, đặc biệt thường không có một hệ thống Data mining chuẩn hóa, luôn thay đổi liên tục rất tốn kém và mất thời gian. Nhận thấy được vấn đề, các tập đoàn công nghệ, công ty chuyên về tư vấn giải pháp phân tích như IBM, NCR, Daimler AG, Teradata đã nghiên cứu và phát triển CRIP-DM. Hiện nay, phần mềm hỗ trợ khai thác Data mining hoàn chỉnh và hiệu quả nhất theo mô hình CRISP-DM chính là IBM SPSS Modeler.

CRISP-DM gồm 6 giai đoạn (bước) chính như sau:

**1. Business understanding / Research understanding phase**

Đầu tiên, chúng ta phải xác định nhu cầu của việc triển khai Data mining là gì và tìm hiểu liệu Data mining sẽ cung cấp thông tin cho chúng ta như thế nào, mục đích khi áp dụng Data mining là gì?

Khi đã xác định được chính xác nhu cầu, và mục đích, chúng ta sẽ có thể quyết định được nguồn lực xây dựng hệ thống Data mining như lựa chọn phần mềm, công cụ Data mining, sau đó phát triển các kế hoạch, dự án Data mining cho từng chiến lược, giai đoạn cụ thể, tiếp theo là thiết lập các tiêu chí đánh giá hệ thống Data mining dựa trên những mục đích nhu cầu đã xác định trước.

Kết thúc giai đoạn này, chúng ta sẽ có một kế hoạch hoàn chỉnh vừa đạt được mục đích Data mining vừa đạt được mục tiêu nghiên cứu.

**2. Data understanding phase**

Những bước đầu tiên trong giai đoạn Data understanding là xác định các loại dữ liệu, các nguồn dữ liệu khác nhau và tiến hành thu thập, tổng hợp, tích hợp dữ liệu, sử dụng metadata ở quy trình Data mining đầu tiên

Tiếp theo dùng các phương pháp phân tích để mô tả ban đầu và làm quen với bộ dữ liệu, xây dựng các báo cáo, truy vấn dữ liệu, lập các bảng, biểu đồ hay còn gọi là các công cụ trực quan hóa để khám phá các thuộc tính của dữ liệu, đảm bảo dữ liệu sẽ mang lại giá trị phân tích đối chiếu với các mục tiêu ở giai đoạn Business understanding, và tìm hiểu tổng quan các thông tin hữu ích tiềm ẩn có thể đạt được thông qua bộ dữ liệu.

Sau cùng là đánh giá chất lượng của bộ dữ liệu bằng cách trả lời những câu hỏi như: “những dữ liệu cần phân tích đã thu nhập đủ chưa?”, “Có missing value tại các cột dữ liệu hay không?” Nếu bộ dữ liệu thực sự chưa đạt yêu cầu, chưa đầy đủ, hay dữ liệu không liên quan đến mục đích nghiên cứu thì chúng ta phải quay lại giai đoạn 1 Business understanding để tìm hiểu kỹ hơn, xác định lại các vấn đề đã gặp.

**3. Data preparation phase**

Đây là giai đoạn theo các chuyên gia là tốn nhiều thời gian nhất trong quy trình CRISP-DM, thành quả của giai đoạn này sẽ giúp chúng ta có một bộ dữ liệu hoàn chỉnh cuối cùng để tiến hành xây dựng các mô hình hay còn gọi là model tập trung phân tích và khai thác thông tin. Data preparation gồm các bước tương tự như Data cleaning, Data integration, Data selection, Data transformation

Select: chọn lọc ra các dữ liệu cần phân tích ví dụ như chọn mẫu, chọn các biến, các cột, dòng dữ liệu có thể liên quan với mục tiêu, cần thiết để phân tích nghiên cứu, sự cân đối của bộ dữ liệu, tránh trường hợp khối lượng dữ liệu quá lớn đưa vào model sẽ làm chậm tốc độ phân tích, ảnh hưởng đến toàn quá trình, đặc biệt loại ra các dữ liệu dư thừa, trùng lặp.

Clean: làm sạch bộ dữ liệu. Dữ liệu sau khi chọn cần được kiểm tra để tìm ra và loại bỏ dữ liệu “bị nhiễu”, dữ liệu không hợp lý,... đặc biệt là xử lý missing value, thêm giá trị vào các dữ liệu chứa missing value (filling in missing value) hay thay đổi mising value bằng các biến giả định phù hợp để chuẩn bị đưa vào model phân tích.

Transform, construct: chuyển đổi, cấu trúc bộ dữ liệu. Điển hình là điều chỉnh cột dữ liệu theo biến mới, theo điều kiện phân tích, thay đổi giá trị trong từng cột dữ liệu sao cho phù hợp hơn, hay xây dựng các cột dữ liệu mới với loại biến, giá trị mới,...

Integrate, format: tích hợp, và điều chỉnh bộ dữ liệu. Quá trình tích hợp cũng cần xây dựng các điều kiện phù hợp để hạn chế sai sót, tránh việc bỏ sót các dữ liệu quý giá. Điều chỉnh dữ liệu nhằm tạo ra bộ dữ liệu gọn hơn, dễ quan sát hơn, tránh bị nhầm lẫn.

**4. Modeling**

Sau khi có một bộ dữ liệu hoàn chỉnh và đầy đủ, chúng ta sẽ tiến hành chọn ra, thiết lập các model dựa trên các thuật toán phù hợp với mục đích phân tích và nghiên cứu. Các thuật toán kỹ thuật trong Data minning điển hình như Association Rule, Clustering, Classification, Outlier Detection, Prediction, …

Các model sẽ tiến hành phân tích dữ liệu và cung cấp các kết quả là những thông tin hữu ích có giá trị. Để đánh giá mức độ chính xác của kết quả thì khi áp dụng model các chuyên gia khuyến khích kiểm tra, thử nghiệm model bằng bộ dữ liệu tương đương.

Kết quả nhận được sau khi chạy các model cần được xem xét cẩn thận, neus kết quả bất hợp lý, hay thông tin cung cấp không phù hợp với thwucj tế, hay bị sai lệch quá nhiều, thì cần quay lại bước Data prepation để kiemr tra dữ liệu chọn ra đã đúng hay chưa? Các dữ liệu nhiễu đã được loại bỏ chưa? Dữ liệu đã được chuyển đổi điều chỉnh hợp lý chưa? Nếu kết quả đã được kiểm chứng, thì đến giai đoạn tiếp theo đánh giá đo lường chi tiết dựa trên mục đích nghiên cứu.

**5. Evaluation**

Kiểm tra, đánh giá kết quả có được từ model với mục tiêu nghiên cứu. Đối chiếu kết quả với những mục tiêu đề ra ban đầu trong bước Business Understanding/ Research Understanding, để xem liệu kết quả, thông tin từ phân tích có hữu ích, có góp phần cho các ý tưởng nghiên cứu khác khả thi hay không

Đây cũng là cơ sở đánh giá sau cùng sự hiệu quả của các bước còn lại trong quy trình CRISP-DM và toàn hệ thống Data minning.

**6. Deployment**

Bước cuối cùng là triển khai kết quả thành các chiến lược, giải pháp cụ thể, trình bày kết quả bằng các công cụ trực quan hóa như đồ thị, biểu đồ, các báo cáo, ... để các người xem dễ hình dung. Đây cũng là giai đoạn cần bàn luận về các cách thwucs cải thiện, phát triển hệ thống Data minning trong tương lai đặc biệt là thiết lập các model, quy trình Data mining chuẩn hóa đã được kiểm chứng để phục vụ lần nghiên cứu, phân tích sắp tới.

## 2.2. Phân cụm dữ liệu và ứng dụng

Phân cụm dữ liệu là một kỹ thuật phát triển mạnh mẽ trong nhiều năm trở lại đây do các ứng dụng và lợi ích to lớn của nó trong các lĩnh vực trong thực tế. Ở một mức cơ bản nhất người ta định nghĩa phân cụm dữ liệu như sau:

Phân cụm dữ liệu là một kỹ thuật trong Data mining nhằm tìm kiếm, phát hiện các cụm, các mẫu dữ liệu tự nhiên tiềm ẩn và quan trọng trong tập dữ liệu lớn để từ đó cung cấp thông tin, tri thức cho việc ra quyết định.

Do đó, phân cụm dữ liệu là quá trình phân chia một tập dữ liệu ban đầu thành các cụm dữ liệu sao cho các đối tượng trong một cụm thì “tương tự” nhau và các đối tượng trong các cụm khác nhau thì “phi tương tự” với nhau. Số cụm dữ liệu được xác định bằng kinh nghiệm hoặc bằng một số phương pháp phân cụm.

Sau khi xác định các đặc tính của dữ liệu, người ta đi tìm cách thích hợp để xác định "khoảng cách" giữa các đối tượng, hay là phép đo tương tự dữ liệu. Đây chính là các hàm để đo sự giống nhau giữa các cặp đối tượng dữ liệu, thông thường các hàm này hoặc là để tính độ tương tự (Similar) hoặc là tính độ phi tương tự (Dissimilar) giữa các đối tượng dữ liệu. Giá trị của hàm tính độ đo tương tự càng lớn thì sự giống nhau giữa đối tượng càng lớn và ngược lại, còn hàm tính độ phi tương tự tỉ lệ nghịch với hàm tính độ tương tự.

Trong quá trình phân cụm dữ liệu thì vấn đề trở ngại lớn nhất đó là nhiễu (noise). Nhiễu xuất hiện do trong quá trình thu thấp thông tin, dữ liệu thiếu chính xác hoặc không đầy đủ. Vì vậy chúng ta cần phải khử nhiễu trong quá trình tiến hành phân cụm dữ liệu.

### 2.2.1 Mục đích của phân cụm dữ liệu

Mục tiêu của phương pháp phân cụm dữ liệu là quá trình nhóm các điểm dữ liệu trong cơ sở dữ liệu thành các cụm sao cho những điểm dữ liệu trong cùng một cụm có độ tương đồng lớn và những điểm không cùng một cụm có sự tương đồng là rất nhỏ. Điểm mạnh của phân cụm dữ liệu là đưa ra được những cấu trúc có ích hoặc những cụm các đối tượng tìm thấy trực tiếp từ dữ liệu mà không cần bất kì một tri thức cơ sở nào. Giống như cách tiếp cận học máy, phân cụm dữ liệu được hiểu như là phương pháp “học không có thầy”(unsupervised learning). Phân cụm dữ liệu không đòi hỏi phải định nghĩa trước các mẫu dữ liệu huấn luyện. Vì thế, có thể coi phân cụm dữ liệu là một cách học bằng quan sát (learning by observation). Trong phương pháp này sẽ không thể biết kết quả các cụm thu được sẽ như thế nào khi bắt đầu quá trình. Vì vậy, cần có một chuyên gia để đánh giá các cụm thu được. Phân cụm đặc biệt hiệu quả khi ta không biết về thông tin của các cụm, hoặc khi ta quan tâm tới những thuộc tính của cụm mà chưa biết hoặc biết rất ít về những thông tin đó. Ngoài ra, phân cụm dữ liệu còn có thể được sử dụng như một bước tiền xử lí cho các thuật toán khai phá dữ liệu khác.

Một vài ví dụ về ý nghĩa thực tiễn của phân cụm dữ liệu có thể áp dụng trong dự án:

* Xác định các nhóm người bệnh nhằm cung cấp thông tin cho điều trị bệnh trong ý tế.
* Khám phá các vị trí địa lý có đông người nhiễm bệnh thuận lợi cho phân vùng cách ly.
* Xác định nhóm mặt hàng mà mọi người gia tăng nhu cầu trong tình hình dịch bệnh...

Như vậy, phân cụm dữ liệu là một phương pháp xử lý thông tin quan trọng và phổ biến, nó nhằm khám phá mối liên hệ giữa các mẫu dữ liệu bằng cách tổ chức chúng thành các cụm tương tự.

Các bước của một bài toán phân cụm dữ liệu gồm:

* Xây dựng hàm tính độ tương tự.
* Xây dựng các tiêu chuẩn phân cụm.
* Xây dựng mô hình cho cấu trúc dữ liệu.
* Xây dựng thuật toán phân cụm và xác lập các điều kiện khởi tạo.
* Xây dựng các thủ tục biểu diễn và đánh giá kết quả phân cụm

**Các kiểu dữ liệu và độ đo tương tự trong phép phân cụm**

* Phân cụm dữ liệu dựa trên kích thước miền:
* Thuộc tính liên tục.
* Thuộc tính rời rạc
* Phân loại dữ liệu dựa trên hệ đo:
* Thuộc tính định danh, thuộc tính thứ tự, thuộc tính khoảng, thuộc tính tỷ lệ, ...

**Các yêu cầu với kỹ thuật phân cụm**

* Khả năng mở rộng.
* Thích nghi với các kiểu dữ liệu khác nhau.
* Khám phá ra các cụm với hình thù bất kỳ.
* Tối thiểu lượng tri thức cần cho xác định tham số vào.
* Ít nhạy cảm với thứ tự của dữ liệu vào.
* Thích nghi với dữ liệu nhiễu cao.
* Ít nhạy cảm với tham số đầu vào.
* Thích nghi với dữ liệu đa chiều.
* Dễ hiểu, dễ cài đặt và khả thi.

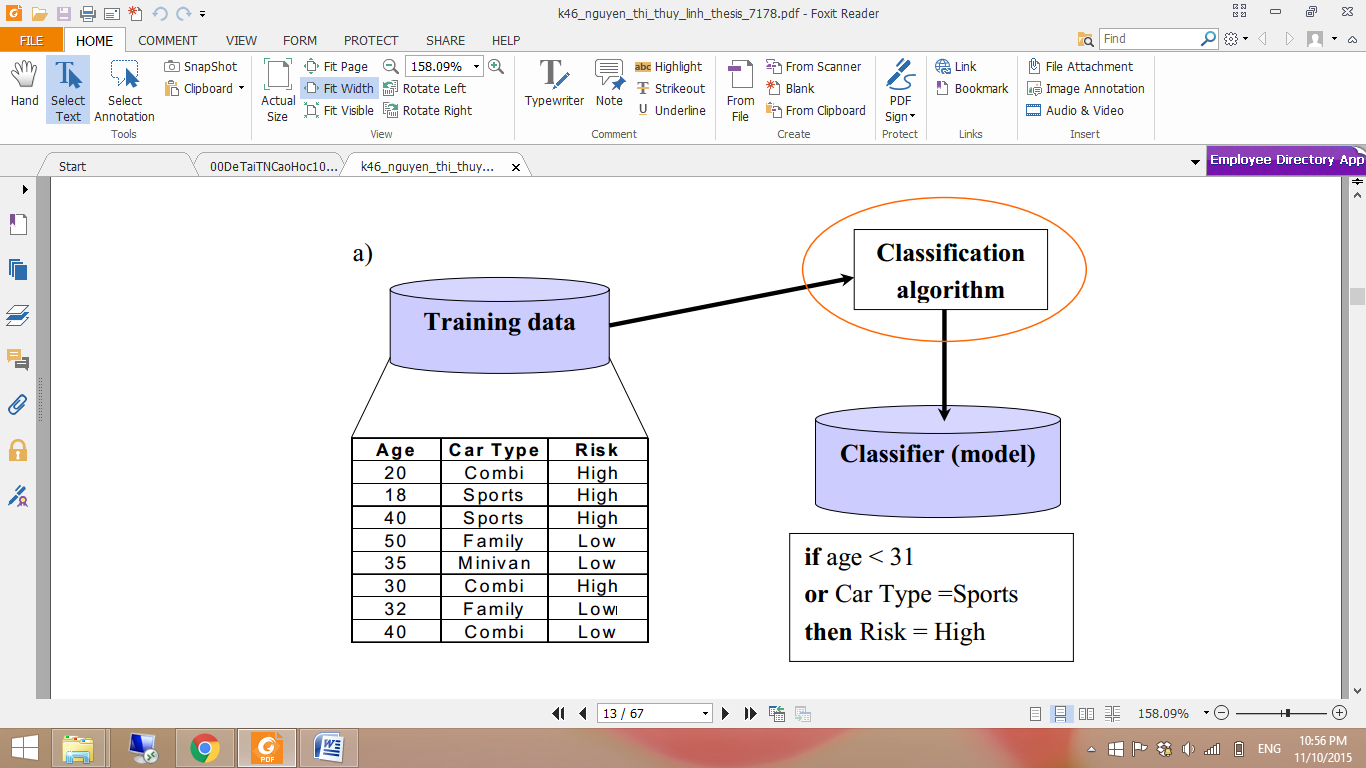
## 2.3. Phân lớp

### 2.3.1 Tìm hiểu chung về phân lớp

***Phân lớp dữ liệu*** (*classification*) là một trong những hướng nghiên cứu chính của khai phá dữ liệu. Thực tế đặt ra nhu cầu là từ một cơ sở dữ liệu với nhiều thông tin ẩn con người có thể trích rút ra các quyết định nghiệp vụ thông minh. Phân lớp và dự đoán là hai dạng của phân tích dữ liệu nhằm trích rút ra một mô hình mô tả các lớp dữ liệu quan trọng hay dự đoán xu hướng dữ liệu tương lai. Phân lớp dự đoán giá trị của những nhãn xác định (*categorical label*) hay những giá trị rời rạc (*discrete value*), có nghĩa là phân lớp thao tác với những đối tượng dữ liệu mà có bộ giá trị là biết trước. Trong khi đó, dự đoán lại xây dựng mô hình với các hàm nhận giá trị liên tục. Ví dụ mô hình phân lớp dự báo thời tiết có thể cho biết thời tiết ngày mai là mưa, hay nắng dựa vào những thông số về độ ẩm, sức gió, nhiệt độ, … của ngày hôm nay và các ngày trước đó. Hay nhờ các luật về xu hướng mua hàng của khách hàng trong siêu thị, các nhân viên kinh doanh có thể ra những quyết sách đúng đắn về lượng mặt hàng cũng như chủng loại bày bán… Một mô hình dự đoán có thể dự đoán được lượng tiền tiêu dùng của các khách hàng tiềm năng dựa trên những thông tin về thu nhập và nghề nghiệp của khách hàng. Trong những năm qua, phân lớp dữ liệu đã thu hút sự quan tâm các nhà nghiên cứu trong nhiều lĩnh vực khác nhau như học máy (*machine learning*), hệ chuyên gia (*expert system*), thống kê (*statistics*)... Công nghệ này cũng ứng dụng trong nhiều lĩnh vực khác nhau như: thương mại, nhà băng, maketing, nghiên cứu thị trường, bảo hiểm, y tế, giáo dục... Phần lớn các thuật toán ra đời trước đều sử dụng cơ chế dữ liệu cư trú trong bộ nhớ (*memory resident*), thườngthao tác với lượng dữ liệu nhỏ. Một số thuật toán ra đời sau này đã sử dụng kỹ thuật cư trú trên đĩa cải thiện đáng kể khả năng mở rộng của thuật toán với những tập dữ liệu lớn lên tới hàng tỉ bản ghi. Quá trình phân lớp dữ liệu gồm hai bước:

* ***Bước thứ nhất (learning)***

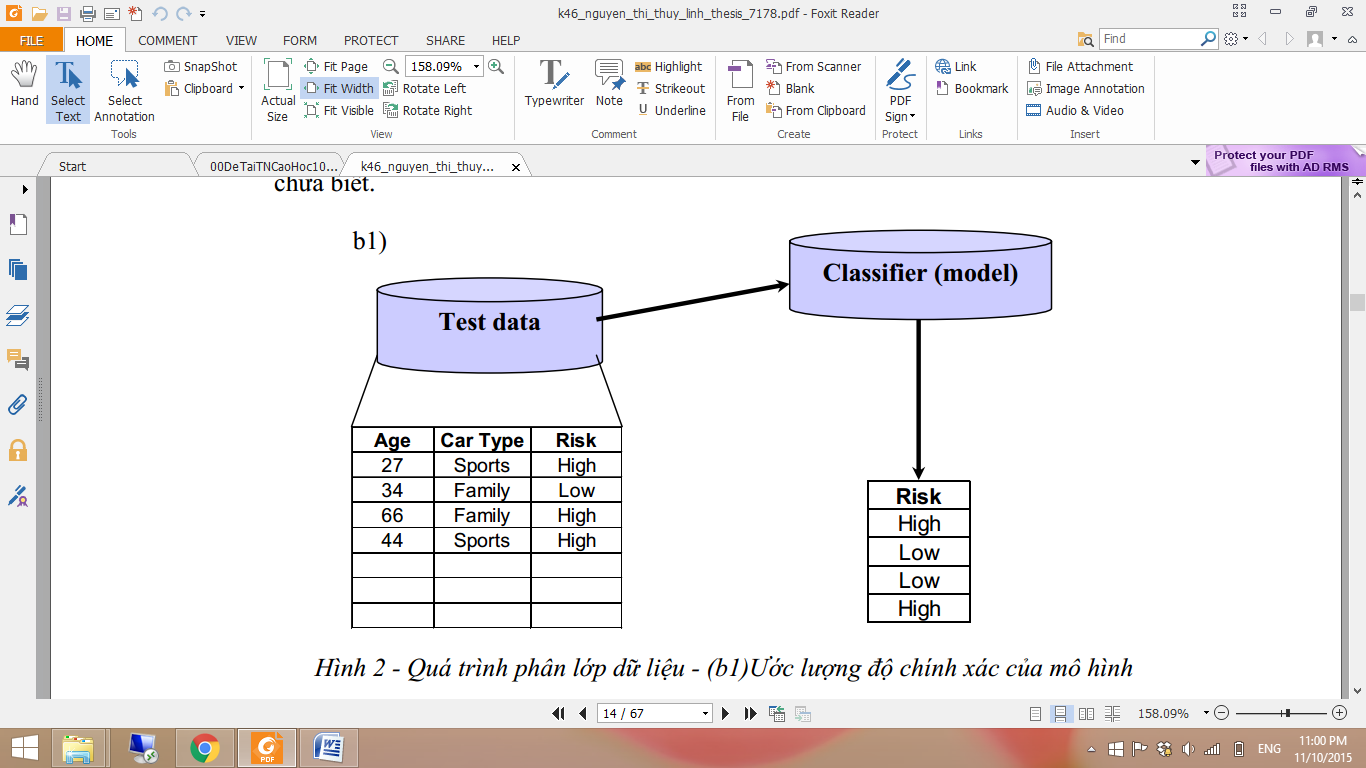
Quá trình học nhằm xây dựng một mô hình mô tả một tập các lớp dữ liệu hay các khái niệm định trước. Đầu vào của quá trình này là một tập dữ liệu có cấu trúc được mô tả bằng các thuộc tính và được tạo ra từ tập các bộ giá trị của các thuộc tính đó. Mỗi bộ giá trị được gọi chung là một phần tử dữ liệu (*data tuple*), có thể là các *mẫu* (*sample*), *ví dụ* (*example*), *đối tượng* (*object*), *bản ghi* (*record*) hay *trường hợp* (*case*). Dự án sử dụng các thuật ngữ này với nghĩa tương đương. Trong tập dữ liệu này, mỗi phần tử dữ liệu được giả sử thuộc về một lớp định trước, lớp ở đây là giá trị của một thuộc tính được chọn làm *thuộc tính gán nhãn lớp* hay *thuộc tính phân lớp* (*class label attribute*). Đầu ra của bước này thường là các quy tắc phân lớp dưới dạng:



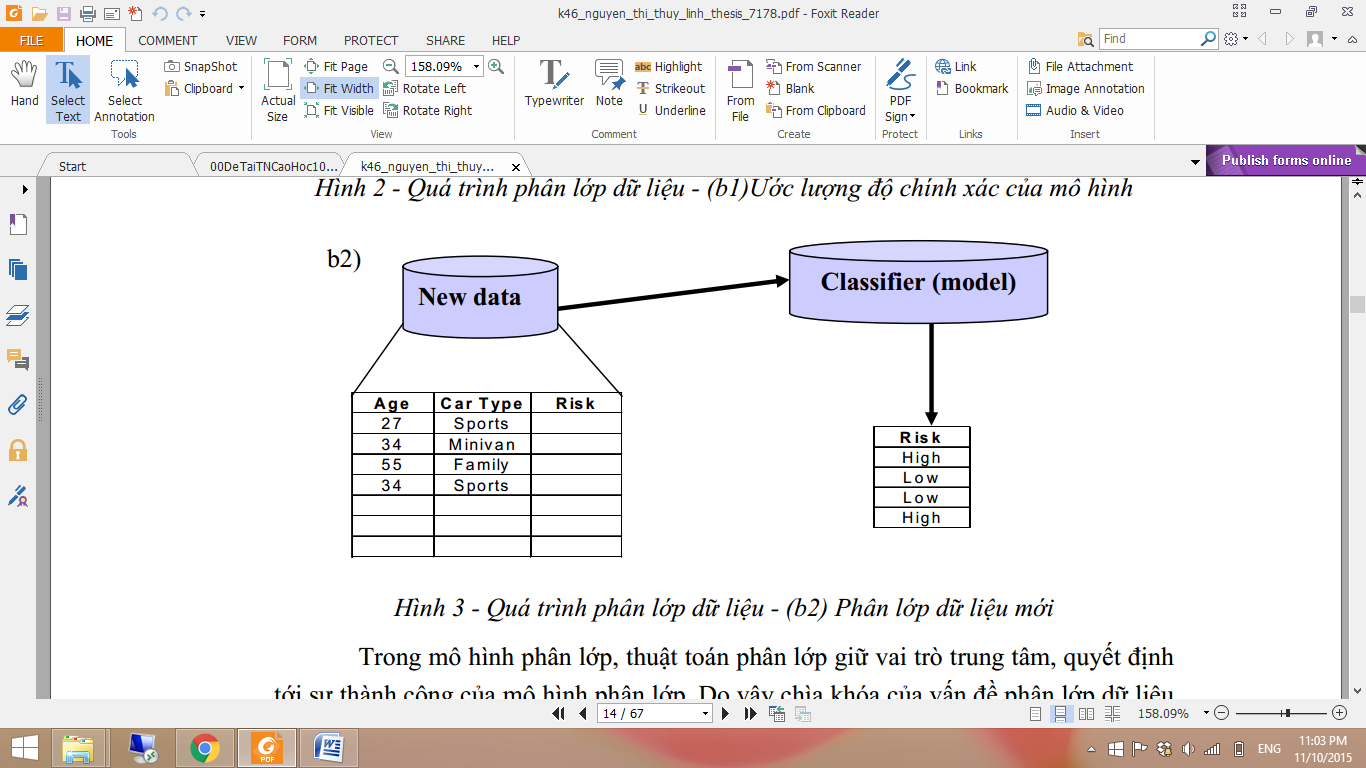
Hình 2.3.1 Quá trình phân lớp dữ liệu- (a) Bước xây dựng mô hình phân lớp

* ***Bước thứ hai (classification)***

Bước thứ hai dùng mô hình đã xây dựng ở bước trước để phân lớp dữ liệu mới. Trước tiên độ chính xác mang tính chất dự đoán của mô hình phân lớp vừa tạo ra được ước lượng. **Holdout** là một kỹ thuật đơn giản để ước lượng độ chính xác đó. Kỹ thuật này sử dụng một tập dữ liệu kiểm tra với các mẫu đã được gán nhãn lớp. Các mẫu này được chọn ngẫu nhiên và độc lập với các mẫu trong tập dữ liệu đào tạo. Độ chính xác của mô hình trên *tập dữ liệu kiểm tra* đã đưa là tỉ lệ phần trăm các các mẫu trong tập dữ liệu kiểm tra được mô hình phân lớp đúng (so với thực tế). Nếu độ chính xác của mô hình được ước lượng dựa trên tập dữ liệu đào tạo thì kết quả thu được là rất khả quan vì mô hình luôn có xu hướng “quá vừa” dữ liệu. Quá vừa dữ liệu là hiện tượng kết quả phân lớp trùng khít với dữ liệu thực tế vì quá trình xây dựng mô hình phân lớp từ tập dữ liệu đào tạo có thể đã kết hợp những đặc điểm riêng biệt của tập dữ liệu đó. Do vậy cần sử dụng một tập dữ liệu kiểm tra độc lập với tập dữ liệu đào tạo. Nếu độ chính xác của mô hình là chấp nhận được, thì mô hình được sử dụng để phân lớp những dữ liệu tương lai, hoặc những dữ liệu mà giá trị của thuộc tính phân lớp là chưa biết.



Hình 2.3.2 Quá trình phân lớp dữ liệu - (b1) Ước lượng độ chính xác của mô hình



Hình 2.3.4 Quá trình phân lớp dữ liệu - (b2) Phân lớp dữ liệu mới

Trong mô hình phân lớp, thuật toán phân lớp giữ vai trò trung tâm, quyết định tới sự thành công của mô hình phân lớp. Do vậy chìa khóa của vấn đề phân lớp dữ liệu là tìm ra được một thuật toán phân lớp nhanh, hiệu quả, có độ chính xác cao và có khả năng mở rộng được. Trong đó khả năng mở rộng được của thuật toán được đặc biệt trú trọng và phát triển. Có thể liệt kê ra đây các kỹ thuật phân lớp đã được sử dụng trong những năm qua:

* *Phân lớp cây quyết định (Decision tree classification)*.
* *Bộ phân lớp Bayesian (Bayesian classifier)*
* *Mô hình phân lớp K-hàng xóm gần nhất (K-nearest neighbor classifier)*.
* *Mạng nơron*.
* *Phân tích thống kê.*
* *Các thuật toán di truyền*.
* *Phương pháp tập thô (Rough set Approach)*

**Một số yếu tố để lựa chọn mô hình phân lớp**

Trong từng ứng dụng cụ thể cần lựa chọn mô hình phân lớp phù hợp. Việc lựachọn đó căn cứ vào sự so sánh các mô hình phân lớp với nhau, dựa trên các tiêu chuẩnsau:

• ***Độ chính xác dự đoán*** (*predictive accuracy*): Độ chính xác là khả năng của mô hình để dự đoán chính xác nhãn lớp của dữliệu mới hay dữ liệu chưa biết.

• ***Tốc độ*** (*speed*): Tốc độ là những chi phí tính toán liên quan đến quá trình tạo ra và sử dụng môhình.

• ***Sức mạnh*** (*robustness*): Sức mạnh là khả năng mô hình tạo ta những dự đoán đúng từ những dữ liệu*noise* hay dữ liệu với những giá trị thiếu.

• ***Khả năng mở rộng*** (*scalability*): Khả năng mở rộng là khả năng thực thi hiệu quả trên lượng lớn dữ liệu của môhình đã học.

• ***Tính hiểu được*** (*interpretability*): Tính hiểu được là mức độ hiểu và hiểu rõ những kết quả sinh ra bởi mô hình đãhọc.

• ***Tính đơn giản*** (*simplicity*): Tính đơn giản liên quan đến kích thước của cây quyết định hay độ cô đọng củacác luật.

Trong luận văn này, đặc biệt quan tâm tới vấn đề chất lượng phân lớpvà tính có thể hiểu được.

**2.3.1 Phương pháp đánh giá độ chính xác của mô hình phân lớp**

Nhìn chung, việc lựa chọn thuật toán phân loại cho bài toán phát hiện hành vi giao thông vẫn đơn thuần được hỗ trợ bởi bằng chứng thực nghiệm. Phần lớn các nghiên cứu đã sử dụng xác thực chéo để thống kê và đánh giá hiệu suất của các thuật toán phân loại với một bộ dữ liệu cụ thể. Kết quả phân loại với một phương pháp cụ thể được thể hiện trong một ma trận cụ thể Mnxn (n là số hành vi được phát hiện). Trong đó Mixj là số hoạt động thực tế trong lớp i được phân loại thành lớp j. Các giá trị sau có thể thu được từ ma trận Mnxn trong vấn đề phân loại:

Table 1 Ma trận thống kê

|  |  |  |  |
| --- | --- | --- | --- |
|  | Thực tế | | |
| Dự đoán |  | True | False |
| Positives | True Positives | False Positives |
| Negatives | True Negatives | False Negatives |

- TP (True Positives): là số lượng các hành vi được phân loại đúng giống như hành vi thực hiện (số lượng hành viTăng tốc được hệ thống phân loại là Tăng tốc khi xe đang Tăng tốc).

- TN (True Negatives): là số lượng các hành vi được phân loại đúng không giống như hành vi thực hiện (số lượng hành viTăng tốc được hệ thống phân loại là Tăng tốc khi một người đang Di chuyển).

- FP (False Positives): là số lượng các hành vi được phân loại sai giống như hành vi thực hiện (số lượng hành viTăng tốc được hệ thống phân loại là Di chuyển khi một người đang Di chuyển).

- FN (False Negatives): là số lượng các hành vi được phân loại sai không giống như hành vi thực hiện (số lượng hành viTăng tốc được hệ thống phân loại là Di chuyển khi một người đang Tăng tốc).

Kết quả nhận dạng của hệ thống được tính theo công thức sau:

Accuracy = (2.1)

Giá trị class precision là tỷ lệ phân loại đúng một hành vi giống như hành vi đang thực hiện với tổng số các hành vi thực hiện:

Precision = (2.2)

Giá trị class recall là tỷ lệ phân loại đúng một hành vi giống như hành vi đang thực hiện với tổng số các hành vi được nhận dạng là hành vi đó:

Recall = (2.3)

### 2.3.2 Các phương pháp phân lớp

Có rất nhiều phương pháp phân lớp, mỗi phương pháp phân lớp đều có cách tính toán khác nhau. Sự khác nhau cơ bản của các phương pháp này là ở thuật toán học quy nạp. Tuy nhiên, nhìn một cách tổng quan thì các phương pháp đó đều phải thực hiện một số bước chung như sau: đầu tiên, mỗi phương pháp sẽ dựa trên các thông tin của các mẫu để biểu diễn mẫu thành dạng vector; sau đó, tuỳ từng phương pháp mà ta sẽ áp dụng công thức và phương thức tính toán khác nhau để thực hiện việc phân loại.

### 2.3.3 Mô hình Deep Neural Network

Deep neural network là hệ thống cấu trúc thần kinh phức tạp gồm nhiều đơn vị neural network mà trong đó, ngoài các lớp nguồn vào (input), nguồn ra (output) thì có hơn một lớp ẩn (hidden layer). Mỗi lớp này sẽ thực hiện một kiểu phân loại và sắp xếp riêng trong một quá trình ta gọi là “phân cấp tính năng” và mỗi lớp đảm nhiệm một trọng trách riêng, output của lớp này sẽ là input của lớp sau.

Deep Neural Network được xây dựng với mục đích mô phỏng hoạt động não bộ phức tạp của con người và được áp dụng vào nhiều lĩnh vực khác nhau, mang lại thành công và những hiệu quả đáng kinh ngạc cho con người.

#### Mạng thần kinh nhân tạo (Artificial neural network)

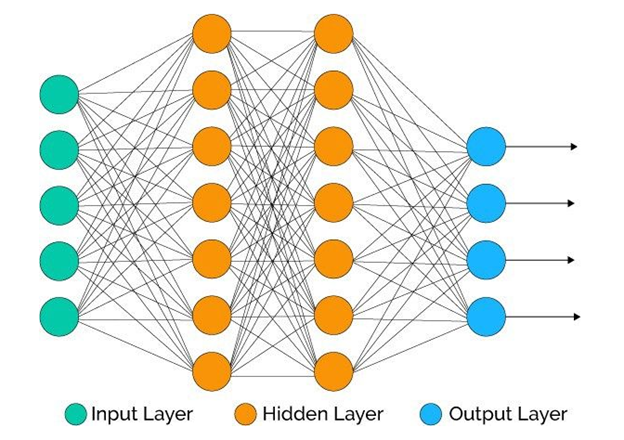
##### Khái niệm

Mạng thần kinh nhân tạo (ANN hay còn được gọi là Neural Network) là một mô hình học tập đặc biệt được mô phỏng theo cách các nơron thần kinh liên kết với nhau trong bộ não của con người. Chúng sử dụng một mạng lưới các nút (mô phỏng theo các khớp thần kinh) để xử lý dữ liệu. Dữ liệu đầu vào sẽ chạy qua toàn bộ mạng lưới ANN và cho ra một loạt các kết quả đầu ra. Sau đó, các kết quả đầu ra này sẽ được so sánh với các dữ liệu mà hệ thống đã được học tập trước đó.

Ví dụ:

Khi ta muốn huấn luyện cho máy tính phân biệt và nhận diện được mặt người, chúng ta cần đưa vào một lượng ảnh đủ lớn chụp các mẫu mặt người để máy tính có thể có đủ lượng dữ liệu cần thiết cho việc học tập. Sau đó, chúng ta sẽ gán nhãn những bức ảnh nào là mặt người thực sự. Hệ thống máy tính sẽ dựa theo những bức ảnh mà chúng ta đã gán nhãn này trên toàn bộ hệ thống mạng thần kinh để có thể cho ra kết quả chuẩn xác nhất. Qua thời gian huấn luyện, dần dần mạng neural sẽ cải thiện độ chính xác của các kết quả mà nó đưa ra.

##### Mô hình kiến trúc mạng ANN



Hình 2.5.1 MÔ HÌNH MẠNG ANN

Một mô hình mạng ANN gồm có 3 thành phần chính: Input layer và Output layer chỉ gồm 1 layer, hidden layer có thể có 1 hay nhiều layer tuỳ vào bài toán cụ thể. ANN hoạt động theo hướng mô tả lại hoạt động của hệ thần kinh với các neuron được kết nối với nhau:

* Lớp nhập (input layer) cung cấp cho mạng các số liệu cần thiết. Số lượng nơron trong lớp nhập tương ứng với số lượng thông số đầu vào được cung cấp cho mạng và các thông số đầu vào này được giả thiết ở dạng vector.
* Lớp ẩn (hidden layer) chứa các neuron ẩn giúp kết nối giá trị đầu vào đến giá trị đầu ra. Một mạng neuron có thể có một hoặc nhiều lớp ẩn chịu trách nhiệm chính cho việc xử lý các neuron của lớp nhập và đưa các thông tin đến neuron của lớp xuất. Các neuron này thích ứng với việc phân loại và nhận diện mối liên hệ giữa thông số đầu vào và thông số đầu ra.
* Lớp xuất (output layer) chứa các neuron đầu ra nhằm chuyển thông tin đầu ra của các tính toán từ ANN đến người dùng. Một ANN có thể được xây dựng để có nhiều thông số đầu ra.

Số nơron của lớp nhập và lớp xuất sẽ do bài toán quyết định, số nơron lớp ẩn và số lớp ẩn sẽ do người nhập quyết định. Tuy nhiên, việc chọn loại và số lượng của thông số đầu vào có ảnh hưởng lớn đến chất lượng của mạng

Trong một mạng lưới ANN, trừ input layer thì tất cả các node thuộc các layer khác đều full-connected với các node thuộc layer trước nó. Mỗi node thuộc hidden layer nhận vào ma trận đầu vào từ layer trước và kết hợp với trọng số để ra được kết quả.

## 2.4. Xử lý ngôn ngữ tự nhiên (Natural Language Processing)

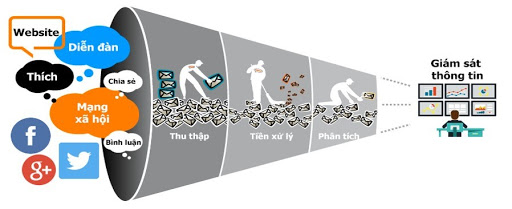
### 2.4.1 Xử lý ngôn ngữ tự nhiên là gì?

Xử lý ngôn ngữ tự nhiên (NLP), là một lĩnh vực khoa học máy tính, kỹ thuật thông tin và trí tuệ nhân tạo tập trung vào nghiên cứu các tương tác về mặt ngôn ngữ giữa máy tính và con người, cụ thể hơn là làm thế nào để lập trình cho máy tính xử lý và phân tích một lượng lớn dữ liệu ngôn ngữ tự nhiên. Trong trí tuệ nhân tạo thì xử lý ngôn ngữ tự nhiên là một trong những phần khó nhất vì nó liên quan đến việc phải hiểu ý nghĩa ngôn ngữ-công cụ hoàn hảo nhất của tư duy và giao tiếp.

Nói cách khác, NLP quan tâm đến việc làm thế nào để máy tính hiểu và tận dụng được các tập dữ liệu sẵn có dưới dạng ngôn ngữ tự nhiên. NLP đã được ứng dụng rộng rãi trong thực tế nhằm: tiết kiệm sức lao động, thúc đẩy các ngành nghề kinh doanh mới, và giúp các nhà hoạch định chiến lược trong việc đưa ra quyết định

### 2.4.2 Ứng dụng của xử lý ngôn ngữ tự nhiên vào giám sát mạng xã hội

Ứng dụng giám sát mạng xã hội có nhiệm vụ tìm hiểu và phân tích ý kiến của người dùng về các chủ đề trên mạng như các trang tin tức, diễn đàn, blogs và các mạng xã hội phổ biến như Twitter, Facebook, … Mục đích cuối cùng của nó là tìm hiểu xem người dùng đang quan tâm về chủ đề gì có thái độ ra sao về chủ đề đó. Những thông tin như vậy sẽ giúp các nhà kinh doanh hiểu được nhu cầu của người dùng từ đó đưa ra các chiến lược cạnh tranh.



Giám sát mạng xã hội có thể được chia thành hai bước: thu thập và phân tích dữ liệu. Thu thập dữ liệu liên quan đến việc tìm và lấy tất cả các bài báo, bình luận từ các trang web, diễn đàn, blogs, các bài đăng, bình luận của người dùng trên mạng xã hội, lưu trữ và phân loại chúng vào các danh mục phù hợp. Một chương trình thu thập thông tin tốt là chương trình có thể thu thập thông tin từ nhiều nguồn khác nhau, chắt lọc, tổng hợp và phân loại chúng theo một cách rõ ràng và thuận tiện nhất cho việc phân tích.

Bộ phân tích có nhiệm vụ xác định và phân loại ý kiến người dùng dựa trên tập dữ liệu đã được tổng hợp từ bước trước. Cụ thể hơn, với mỗi bài đăng hay bình luận, xác định xem nó là phản hồi tích cực, tiêu cực hay trung lập. Nhưng trong một số trường hợp khác, người ta có thể muốn biết chi tiết hơn ý kiến của người dùng về một khía cạnh cụ thể. Khi đó, một bộ phân tích cảm xúc với từng khía cạnh của sản phẩm/dịch vụ (ABSA) sẽ được triển khai. ABSA bao gồm ba tác vụ chính: phân loại khía cạnh, trích xuất từ khóa và phân loại cảm xúc.

* Phân loại khía cạnh là xác định xem bài đăng hay bình luận liên quan đến khía cạnh nào của sản phẩm hay dịch vụ. Đó có thể là chất lượng, giá thành các món ăn với một nhà hàng, dịch vụ chăm sóc khách hàng với một khách sạn hoặc là tốc độ đường truyền với một nhà cung cấp dịch vụ Internet.
* Trích xuất từ khóa chịu trách nhiệm trích xuất các từ hay cụm từ đại diện cho các khía cạnh trong câu/đoạn/bình luận đầu vào. Các từ, cụm từ này thường là các danh từ, cụm danh từ trong câu.
* Tác vụ cuối cùng liên quan đến việc phân loại cảm xúc của người dùng. Việc phân loại cảm xúc này không dừng lại ở mức câu/đoạn/bình luận mà là phân loại cảm xúc người dùng trên từng khía cạnh của sản phẩm được nhắc đến trong câu/đoạn/bình luận. ABSA là một bài toán khó không chỉ bởi chúng ta thiếu dữ liệu gán nhãn để huấn luyện mô hình mà còn bởi vì tính không cấu trúc, nhiều ẩn dụ, từ “lóng”, và sai chính tả của dữ liệu mạng xã hội.

### 2.4.3. Topic modeling:

Topic modeling là một kỹ thuật học máy không giám sát, có khả năng quét một tập hợp các tài liệu, phát hiện các mẫu từ và cụm từ trong đó và tự động phân cụm các nhóm từ và các biểu thức tương tự đặc trưng nhất cho một bộ tài liệu.

Ngày nay người ta nói rất nhiều về trí tuệ nhân tạo, cùng với các thuật ngữ như học máy và Xử lý ngôn ngữ tự nhiên (NLP). Đặc biệt trong một công ty xử lý hàng trăm, thậm chí hàng ngàn tương tác của khách hàng mỗi ngày. Phân tích một lượng lớn dữ liệu từ các bài đăng trên mạng xã hội, email, trò chuyện, phản hồi khảo sát, v.v…., không phải là một nhiệm vụ dễ dàng.

Đó là lý do tại sao nhiều người hào hứng về ý nghĩa trí tuệ nhân tạo có thể có trong các nhiệm vụ hàng ngày của họ, cũng như cho các doanh nghiệp nói chung. Phân tích văn bản do AI cung cấp sử dụng nhiều phương pháp hoặc thuật toán để xử lý ngôn ngữ một cách tự nhiên, một trong số đó là Topic modeling - được sử dụng để tự động phát hiện các chủ đề từ văn bản.

Trong dự án này, chúng tôi sẽ xem xét kỹ thuật phân tích chủ đề: Topic modeling. Topic modeling là một kỹ thuật học máy không có giám sát, nói cách khác, một kỹ thuật không cần đào tạo.

Topic modeling là một kỹ thuật học máy tự động phân tích dữ liệu văn bản để xác định các chủ đề cho một bộ tài liệu. Đây là một mô hình học không giám sát trong học máy bởi vì nó không yêu cầu một danh sách các thẻ hoặc dữ liệu đào tạo được xác định trước mà con người đã phân loại trước đó.

Vì topic modeling không có yêu cầu đào tạo, nên có thể nhanh chóng và dễ dàng bắt đầu phân tích dữ liệu. Nhưng vẫn đảm bảo có thể đảm bảo bạn sẽ nhận được kết quả tốt.

### 2.4.4 Bài toán phân loại văn bản (Text classification)

**Định nghĩa phân loại văn bản:**

Phân loại văn bản là sự phân loại không cấu trúc các tài liệu văn bản dựa trên một tập hợp của một hay nhiều loại văn bản đã được định nghĩa trước. Trong nhiều lĩnh vực, phân loại văn bản là một trong những bài toán được ứng dụng rộng rãi của xử lý ngôn ngữ tự nhiên, với mục đích là tự động phân loại các tài liệu vào một hoặc nhiều thể loại được xác định.

**Giới thiệu chung về bài toán phân loại văn bản trong xử lý ngôn ngữ tự nhiên**:

Ứng dụng của công nghệ phân loại hiện nay đang phát triển rất mạnh ở rất nhiều lĩnh vực (học thuật, kinh doanh, bảo mật, y tế...) và các đối tượng (nhà nghiên cứu xã hội, chính phủ và các tổ chức phi lợi nhuận khác). Vì các tổ chức này sở hữu một lượng lớn dữ liệu không có cấu trúc và việc xử lý dữ liệu sẽ trở nên dễ dàng hơn rất nhiều nếu như các dữ liệu này được chuẩn hóa bởi các chủ đề/nhãn. Nền tảng công nghệ để thực hiện bài toán phân loại văn bản chính là trí tuệ nhân tạo (Artificial Intelligence – AI) và học sâu (Deep Learning).

Bên cạnh đó, trong những năm gần đây, ứng dụng mạng xã hội vào hoạt động kinh tế đang đạt được những lợi ích chưa từng có. Các doanh nghiệp có thể thu thập các đánh giá của khách hàng về các sản phẩm và dịch vụ của họ. Điều này là cơ sở để tái phát minh các chiến lược kinh doanh kế tiếp, sao cho phù hợp với người tiêu dùng. Việc phân tích xã hội sử dụng một lượng dữ liệu khổng lồ từ các mạng xã hội để đưa ra các quyết định chiến lược. Các doanh nghiệp sử dụng các kỹ thuật và công cụ học máy để xác định các mẫu và xu hướng, từ đó có được các thông tin hữu ích. Việc gán nhãn cho nội dung hoặc sản phẩm sử dụng các danh mục cũng là một cách để cải thiện việc duyệt web hoặc xác định nội dung liên quan trên trang web. Các lĩnh vực như thương mại điện tử, truyền thông có thể sử dụng công nghệ tự động để phân loại và gán nhãn cho nội dung và sản phẩm. Hay một hệ thống phản ứng khẩn cấp có thể được thực hiện hiệu quả hơn bằng cách phân loại các cuộc trò chuyện, trao đổi trên phương tiện truyền thông xã hội. Các nhà chức trách có thể theo dõi và phân loại tình huống khẩn cấp để phản ứng nhanh nếu có bất kỳ tình huống nào phát sinh. Đây là trường hợp phân loại rất chọn lọc.

Bài toán phân loại văn bản là một trong những nội dung của xử lý ngôn ngữ tự nhiên. Đó là vấn đề xây dựng các mô hình mà có thể phân loại các tài liệu mới vào các chủ đề đã được xác định trước. Đây là một quá trình xử lý phức tạp, không chỉ liên quan đến việc huấn luyện các mô hình mà cần thực hiện nhiều bước khác nhau như: tiền xử lý dữ liệu, chuyển đổi dữ liệu, giảm kích thước của dữ liệu. Để phân loại văn bản cần sử dụng các kỹ thuật khác nhau và triển khai trong các hệ thống phức tạp. Hiện nay, có hai hướng phát triển hệ thống phân loại: phát triển các hệ thống phân loại mới, hoặc cải tiến các hệ thống có sẵn để đạt kết quả tốt hơn.

Ở Việt Nam, cộng đồng nghiên cứu trí tuệ nhân tạo xử lý ngôn ngữ tự nhiên để áp dụng vào bài toán xử lý tiếng Việt đang rất phát triển. Do đặc thù của tiếng Việt và sự khác biệt về bộ ngôn ngữ trong các thư viện, công cụ hỗ trợ, nên cộng đồng các nhà nghiên cứu AI của Việt Nam đã xây dựng và phát triển một số các thuật toán, thư viện và công cụ dành riêng cho tiếng Việt. Một số doanh nghiệp và trường đại học hoạt động trong lĩnh vực công nghệ thông tin đã và đang nghiên cứu đưa bài toán phân loại văn bản vào ứng dụng thực tế như: Framgia, Đại học Lê Quý Đôn, FPT, Đại học Khoa học Công nghệ thuộc Đại học Quốc gia Việt Nam ….

**Việc phân loại văn bản có nhiều ứng dụng thực tiễn**

* Ví dụ: Trong lĩnh vực toàn mạng trong thời gian gian diễn ra dịch Covid19 việc thu thập thông tin trên các trang báo điện tử, diễn đàn, mạng xã hội,… để theo dõi tình hình, nắm bắt thông tin là một trong những nhiệm vụ mới, quan trọng trên không gian mạng. Do lượng thông tin thu thập được qua mạng Internet là rất lớn, nên đã đặt ra yêu cầu phân tích và xử lý lượng dữ liệu khổng lồ trên một cách tự động và nhanh chóng giúp cơ quan nhanh chóng phát hiện và xác định những đối tượng đăng bài sai sự thật gây hoang mang dư luận.

**Các cách xử lý dữ liệu ngôn ngữ tự nhiên**

Vì dữ liệu đầu vào là văn bản nên trong quá trình xử lý chúng ta cần đưa dữ liệu này về một dạng mà các mô hình có thể học một cách dễ dàng. Có 2 hướng tiếp cận:

* Biểu diễn văn bản bằng các từ quan trọng, các từ này sẽ ảnh hưởng tới việc văn bản thuộc class nào.
* Biểu diễn văn bản thành một ma trận (văn bản ngắn thì có thể biểu diễn thành một mảng 2 chiều) hoặc thành một vector (văn bản dài có thể thực hiện theo cách này - khá lý tưởng).

a. Biểu diễn thành ma trận:

* Bước 1: Biến 1 từ (word) thành một vector (có thể sử dụng word2vec training sẵn)
* Bước 2: Ghép các từ lại thành 1 mảng, như vậy ta được mảng 2 chiều (ma trận): l x d (l = length là chiều dài của văn bản hay số từ; d = dimension là chiều của không gian vector mà 1 từ (word) được biểu diễn).

b. Biểu diễn thành vector: Cho một mô hình học từ nhiều văn bản khác nhau và có thể ánh xạ một văn bản thành 1 vector.

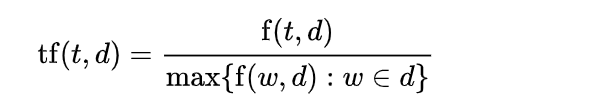
#### ***2.6.4.1 TF-IDF Vector as features (Term Frequency - Inverse Document Frequency): biểu diễn văn bản về dạng số***.

1. **Khái niệm**

TF-IDF (Term Frequency – Inverse Document Frequency) là 1 kĩ thuật sử dụng trong khai phá dữ liệu văn bản. Trọng số này được sử dụng để đánh giá tầm quan trọng của một từ trong một văn bản. Giá trị cao thể hiện độ quan trọng cao và nó phụ thuộc vào số lần từ xuất hiện trong văn bản nhưng bù lại bởi tần suất của từ đó trong tập dữ liệu. Một vài biến thể của tf-idf thường được sử dụng trong các hệ thống tìm kiếm như một công cụ chính để đánh giá và sắp xếp văn bản dựa vào truy vấn của người dùng. Tf-idf cũng được sử dụng để lọc những từ stopwords trong các bài toán như tóm tắt văn bản và phân loại văn bản.

1. **TF là gì?**

TF: Term Frequency (Tần suất xuất hiện của từ) là số lần từ xuất hiện trong văn bản. Vì các văn bản có thể có độ dài ngắn khác nhau nên một số từ có thể xuất hiện nhiều lần trong một văn bản dài hơn là một văn bản ngắn. Như vậy, term frequency thường được chia cho độ dài văn bản (tổng số từ trong một văn bản).

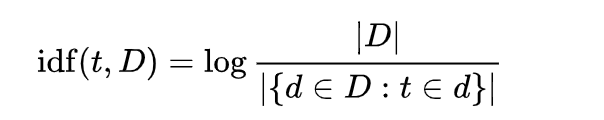


Trong đó:

* tf (t, d): tần suất xuất hiện của từ t trong văn bản d
* f (t, d): Số lần xuất hiện của từ t trong văn bản d
* max ({f (w, d): w ∈ d}): Số lần xuất hiện của từ có số lần xuất hiện nhiều nhất trong văn bản d

1. **IDF là gì?**

IDF: Inverse Document Frequency (Nghịch đảo tần suất của văn bản), giúp đánh giá tầm quan trọng của một từ. Khi tính toán TF, tất cả các từ được coi như có độ quan trọng bằng nhau. Nhưng một số từ như “is”, “of” và “that” thường xuất hiện rất nhiều lần nhưng độ quan trọng là không cao. Như thế chúng ta cần giảm độ quan trọng của những từ này xuống.



Trong đó:

* idf (t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D : t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

Cơ số logarit trong công thức này không thay đổi giá trị idf của từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Việc sử dụng logarit nhằm giúp giá trị tf-idf của một từ nhỏ hơn, do chúng ta có công thức tính tf-idf của một từ trong 1 văn bản là tích của tf và idf của từ đó.

Cụ thể, chúng ta có công thức tính tf-idf hoàn chỉnh như sau:

**tf\_idf (t, d, D) = tf(t,d) x idf(t,D)**

Khi đó: Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó).

#### 2.6.4.2 Dùng SVD (singular value decomposition) giảm số chiều ma trận

Giải thuật SVD được Golub và Kahan giới thiệu năm 1965, đó là một công cụ phân rã ma trận hiệu quả được sử dụng để giảm hạng (hay số chiều) của ma trận. Kỹ thuật này được áp dụng vào nhiều bài toán xử lý văn bản khác nhau như tóm tắt văn bản, phát hiện sao chép, lập chỉ mục và truy vấn. SVD cho phép phân tích một ma trận phức tạp thành ba ma trận thành phần. Mục đích nhằm đưa việc giải quyết bài toán liên quan đến ma trận lớn, phức tạp về những bài toán nhỏ hơn.

A = USVT

Trong đó

* U: là ma trận trực giao cấp m x r (m số từ chỉ mục) các vectơ dòng của U là các vector từ chỉ mục.
* S: là ma trận đường chéo cấp r x r có các giá trị suy biến (singular value) *σ1 >= σ2* >= *σ3…* >= *σr* với r = **rank(A).**
* V: là ma trận trực giao cấp r x n (n số văn bản trong tập văn bản) - các vector cột của V là các vector văn bản.

Hạng của ma trận A là các số dương trên đường chéo của ma trận S. Giả sử hạng của ma trận A là r hay rank(A) = r thì số Frobenius của A là



Ta có thể sử dụng SVD để xấp xỉ ma trận A với n giá trị đơn:

Ak = UkSkVkT

Ma trận xấp xỉ Ak =UkSkVkT có hạng là k với k << r, trong đó:

* Uk, Vk là ma trận trực giao
* Sk là ma trận chéo cấp k x k
* r là hạng của A
* k là số chiều được chọn trong mô hình giảm lược (k ≤ r).

Giảm lược số chiều, lựa chọn k là tới hạn. Đúng như ý tưởng, chúng ta muốn một giá trị k đủ lớn để phù hợp mọi đặc tính cấu trúc thực của dữ liệu, nhưng đủ nhỏ để lọc ra các chi tiết không phù hợp hay các chi tiết không quan trọng.

Việc tính toán phân rã ma trận với SVD đòi hỏi thời gian tính toán cao, vì vậy để rút ngắn thời gian tính toán có thể dùng giải pháp tính toán song song.

#### 2.6.4.3 Dùng PCA để giảm chiều ma trận số.

1. **PCA là gì?**

Phân tích thành phần chính (Principal Component Analysis – PCA) là một thuật toán Machine Learning thống kê sử dụng phép biến đổi một tập hợp các dữ liệu từ một không gian nhiều chiều qua không gian ít chiều hơn. Việc làm này nhằm tối ưu việc thể hiện sự biến thiên của tập dữ liệu.

1. **Cách tiến hành phương pháp PCA**

Với dữ liệu cần phân tích ban đầu phụ thuộc nhiều biến, vấn đề là các biến này thường có tương quan với nhau sẽ bất lợi cho việc áp dụng các biến này để xây dựng các mô hình tính toán ví dụ: hồi quy, … và với số biến giải thích lớn chúng ta sẽ rất khó để có cái nhìn trực quan về dữ liệu.

Phương pháp PCA sẽ “chiếu” (biểu diễn) dữ liệu đa chiều lên một không gian có cơ sở trực giao, tức nếu ta xem mỗi cơ sở trong không gian mới là một biến thì hình ảnh của dữ liệu gốc trong không gian mới này sẽ được biểu diễn thông qua các biến độc lập (tuyến tính). Vấn đề: nếu chuyển dữ liệu ban đầu sang không gian mới thì những thông tin đáng quan tâm của dữ liệu ban đầu liệu có bị mất? Để giải quyết vấn đề này phương pháp PCA sẽ tìm không gian mới với tiêu chí cố gắng phản ánh được càng nhiều thông tin gốc càng tốt, và thước đo cho khái niệm “thông tin” ở đây là phương sai. Một điểm hay nữa là: do các biến trong không gian mới độc lập, nên ta có thể tính toán được tỷ lệ giải thích phương sai của từng biến mới đối với dữ liệu, điều này cho phép ta cân nhắc việc chỉ dùng số ít các biến để giải thích dữ liệu.

Nói một cách ngắn gọn, mục tiêu của PCA là tìm một không gian mới (với số chiều nhỏ hơn không gian cũ). Các trục tọa độ trong không gian mới được xây dựng sao cho trên mỗi trục, độ biến thiên của dữ liệu trên đó là lớn nhất có thể.



Hình 2.6.4.3 Ví dụ Kinh điển về hình ảnh con lạc đà

* Ví dụ minh họa PCA: phép chiếu lên các trục tọa độ khác nhau có thể cho cách nhìn rất khác nhau về cùng loại dữ liệu. Một ví dụ kinh điển hình ảnh về con lạc đà. Cùng là một con lạc đà nhưng nếu nhìn từ bên hông thì ta thấy được đầy đủ thông tin nhất, trong khi nhìn từ phía trước thì thật khó để nói nó là lạc đà.

**c. Ưu điểm của PCA**

* Giúp giảm số chiều của dữ liệu.
* Thay vì giữ lại các trục tọa độ của không gian cũ, PCA xây dựng một không gian mới với ít chiều hơn, nhưng lại có khả năng biểu diễn dữ liệu tốt tương đương không gian cũ, nghĩa là đảm bảo độ biến thiên (variability) của dữ liệu trên mỗi chiều mới.
* Các trục tọa độ trong không gian mới là tổ hợp tuyến tính của không gian cũ, do đó về mặt ngữ nghĩa, PCA xây dựng feature mới dựa trên các feature đã quan sát được. Điểm hay là các feature này vẫn biểu diễn tốt dữ liệu ban đầu.
* Trong không gian mới, các liên kết tiềm ẩn của dữ liệu có thể được khám phá, mà nếu đặt trong không gian cũ thì khó phát hiện hơn, hoặc những liên kết như thế không thể hiện rõ.

**d. Các ứng dụng tự nhiên**

* Giảm kích thước của dữ liệu.
* Nếu ta có thể giảm số chiều về 2 hoặc 3 chiều, ta có thể dùng các loại đồ thị để hiểu thêm về dữ liệu mà mình đang có, giúp ta nhìn dữ liệu trực quan hơn.
* Xử lý vấn đề tương quan giữa các biến trong dữ liệu ban đầu bằng cách sử dụng các biến mới trong không gian mà phương pháp PCA tìm được để mô tả dữ liệu.

1. **Mục đích chính**

PCA được sử dụng cho 2 mục tiêu:

* Giảm số lượng các biến bao gồm một bộ dữ liệu trong khi giữ lại các biến đổi trong dữ liệu.
* Xác định các mô hình ẩn trong dữ liệu và phân loại chúng theo nhiều cách thông tin, lưu trữ trong các dữ liệu.

## 2.5. Các công nghệ sử dụng:

### 2.5.1. Anaconda



**Anaconda là gì**: Anaconda là nền tảng mã nguồn mở (platform) về Khoa học dữ liệu (Data Science) trên Python thông dụng nhất hiện nay. Anaconda có hơn hơn 20 triệu người dùng trên toàn thế giới, Anaconda là cách nhanh nhất và dễ nhất để học Khoa học dữ liệu với Python hoặc R trên Windows, Linux và Mac OS X.

* Lợi ích của Anaconda:
* Dễ dàng tải hơn 7.500 packages về Python/R cho data science
* Quản lý thư viện, môi trường và dependency giữa các thư viện dễ dàng
* Dễ dàng phát triển mô hình machine learning và deep learning với scikit-learn, tensorflow, keras
* Xử lý dữ liệu tốc độ cao với numpy, pandas
* Hiển thị kết quả với Matplotlib, Bokeh

### 2.5.2. Python3



**Khái niệm**: Python là ngôn ngữ lập trình bậc cao, được tạo ra bởi Guido van Rossum và phát hành lần đầu vào năm 1991. Python được thiết kế hướng đối tượng với cú pháp đơn giản, dễ học. Nó tạo ra những chương trình với số lượng dòng code là ít nhất. Vì thế Python là 1 trong những ngôn ngữ lập trình phát triển nhanh nhất hiện nay (Nó có tốc độ tăng trưởng lớn nhất tính theo năm so với bất cứ ngôn ngữ lập trình nào, được thống kê bởi Học viên Stone River. Python trên toàn thế giới có tốc độ tăng trưởng lớn nhất trong 5 năm qua (7.0%), theo sau ngôn ngữ phổ biến Java, theo như PYPL công bố). Các ứng dụng sử dụng các công nghệ AI, Machine Learning, Blockchain… luôn ưu tiên sử dụng Python làm ngôn ngữ lập trình.

* Ưu điểm của Python
* Python là một ngôn ngữ có cấu trúc rõ ràng, cú pháp ngắn gọn và hình thức sáng sủa.
* Python có trên tất cả các nền tảng hệ điều hành từ UNIX, MS – DOS, Mac OS, Windows và Linux và các OS khác thuộc họ Unix.
* Ngôn ngữ lập trình python tương thích mạnh mẽ với Unix, hardware, third-party software với số lượng thư viện khổng lồ (400 triệu người sử dụng).
* Đặc biệt, ngôn ngữ lập trình Python với tốc độ xử lý cực nhanh, python có thể tạo ra những chương trình từ những script siêu nhỏ tới những phần mềm cực lớn như Blender 3D.
* Ứng dụng của Python:
* Xếp thứ 2/5 trong “Top ngôn ngữ đáng học nhất năm 2018” Python luôn nằm trong top đầu những ngôn ngữ lập trình phổ biến nhất. Bởi Python chính là nền tảng cho rất nhiều sản phẩm cực kỳ nổi tiếng: Google, Quora, Instagram, BitTorrent, Reddit, hay Dropbox. Bạn muốn phát triển một trang web, Python và framework Django đã tạo nên những ứng dụng tuyệt vời như Instagram, YouTube.
* Phù hợp lập trình với hệ điều hành Android. Bởi Android không đòi hỏi phải code quá nhiều, nó thường ngắn hơn Java từ 3-5 lần và ngắn hơn C++ 5-10 lần.
* Đặc biệt, hiện nay, ngôn ngữ lập trình Python là một trong những ngôn ngữ lập trình được sử dụng nhiều nhất trong việc phát triển trí tuệ nhân tạo. Python được nhiều người lựa chọn để lập trình trí tuệ nhân tạo vì tính phổ biến, dễ học và khả năng tương thích với công nghệ 4.0. Nó có thể được sử dụng một cách trơn tru với các cấu trúc dữ liệu và các thuật toán AI được sử dụng thường xuyên khác.

### 2.5.3. IDE: Jupyter NoteBook

Jupyter là một công cụ mã nguồn mở miễn phí với mục đích nhắm đến khoa học dữ liệu và giáo dục, giúp mọi người cùng học Python dễ dàng hơn. Jupyter có tính tương tác nên có thể sử dụng làm môi trường chạy thử và giảng dạy.

Tên Jupyter xuất phát từ cách chơi chữ Tôi là Python, bạn là R, chúng ta là Julia. Trước đây, Jupyter bắt nguồn từ dự án IPython, đến năm 2014 thì đổi tên sang Jupyter. Những người yêu thích Python – sử dụng Jupyter rất nhiều. Jupyter hỗ trợ rất nhiều các kernel cho các ngôn ngữ khác nhau, khoảng trên 40 ngôn ngữ trong đó có Python. Việc đổi tên từ IPython sang Jupyter cũng là vì mục đích hỗ trợ đa ngôn ngữ.

Jupyter notebook là công cụ cho phép bạn đưa cả code Python và các thành phần văn bản phức tạp như hình ảnh, công thức, video, biểu thức… vào trong cùng một file giúp cho việc trình bày trở lên dễ hiểu, giống như một file trình chiếu nhưng lại có thể thực hiện chạy code tương tác trên đó. Các file notebook này có thể được chia sẻ với mọi người và có thể thực hiện lại các công đoạn một cách nhanh chóng và chính xác như những gì tác giả của file này tạo ra. Đây là một công cụ tuyệt vời để bạn có thể tạo ra những thực nghiệm, ví dụ một cách nhanh và trực quan nhất.

**Vì sao chọn Jupyter notebook**?

Jupyter Notebook là một công cụ tuyệt vời để:

* Học và thử nghiệm với Python
* Xử lý / biến đổi dữ liệu (Data processing / transformations)
* Mô phỏng số học (vẽ biểu đồ)
* Thống kê mô hình
* Machine learning

# CHƯƠNG 3: PHƯƠNG PHÁP THU THẬP, ĐÁNH GIÁ,

# PHÂN TÍCH DỮ LIỆU VÀ QUÁ TRÌNH TRIỂN KHAI HỆ THỐNG THEO DÕI DỊCH BỆNH COVID-19 TẠI VIỆT NAM

## 3.1. Xây dựng ý tưởng, xác định các chức năng cơ bản của hệ thống

### Xây dựng ý tưởng

Trong tình hình dịch bệnh diễn biến phức tạp đòi hỏi nhu cầu tìm kiếm thông tin của mọi người tăng cao. Lợi dụng sự “khát” thông tin này nhiều người muốn trang mạng của mình điều hành tăng lượt truy cập đã đưa ra những thông tin giả, tin nhiễu gây hoang mang dư luận khiến người dân bị rơi vào một “đại dịch” khác đó là lạc lối giữa ma trận thông tin khổng lồ. Việc có quá nhiều tin giả và tin tức chưa qua kiểm chứng được đăng tải liên tục lên mạng xã hội khiến người dân khó nắm bắt được diễn biến hiện tại của dịch bệnh tại Việt Nam nói riêng và thế giới nói chung là tiền đề để nhóm xây dựng ý tưởng tạo một trang web tổng hợp cung cấp thông tin của những trang báo uy tín giúp người dân theo dõi được các tin tức liên quan chỉ trên một giao diện. Các bài báo được cập nhập liên tục theo thời gian thực đảm bảo thông tin bài báo không chỉ chính xác mà còn đúng đắn và kịp thời.

Thêm vào đó nhu cầu thông tin về số ca bệnh tại từng tỉnh thành cũng là một nhu cầu thiết yếu của mọi người trong những ngày dịch, giúp mọi người đưa ra quyết định có nên đến những nơi có khả năng lây nhiễm cao hay không? Nhóm đưa thông tin về số ca bệnh theo từng tỉnh, thành phố theo dạng bản đồ để giao diện thân thiện hơn với người sử dụng.

Bên cạnh đó để trang web có giá trị thương mại cao đòi hỏi phải có những phân tích báo cáo chuyên sâu hơn về những vấn đề liên quan đến dịch bệnh.

* Ví dụ các bài báo muốn có nhiều truy cập, lượt đọc thì việc bài báo có thông tin chính xác thôi là chưa đủ. Một bài viết có độ phổ biến cao phải là bài báo đánh trúng vào chủ đề mà được nhiều người quan tâm nhất từ đó việc phân tích được các chủ đề đang được quan tâm nhiều nhất là việc làm vô cùng cần thiết.
* Hoặc muốn giảm số ca nhiễm bệnh ta cần xác định được đâu là đối tượng chủ yếu mắc bệnh để đưa ra những phương án phòng bệnh chính xác và hiệu quả nhất. Từ đó biểu đồ phân tích về nhóm độ tuổi nhiễm bệnh phổ biến là vô cùng cần thiết.

### 3.1.2 Các mục tiêu, tính năng cơ bản:

* Bản đồ lây nhiễm dịch bệnh tại Việt Nam
* Số liệu thống kê các ca nhiễm, thông tin dịch bệnh tại Việt Nam
* Tin tức từ những nguồn tin chính thống

### 3.1.3 Các mục tiêu, tính năng nâng cấp, phát triển:

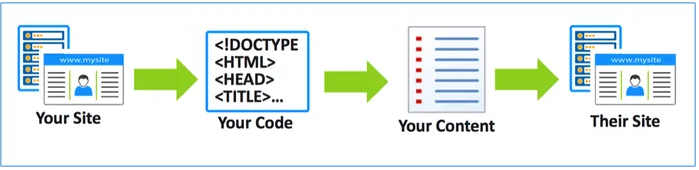
* Xác định các chủ đề mà mọi người đang bàn luận về dịch bệnh trên mạng xã hội
* Đánh giá sắc thái các bài viết có lượng tương tác cao trên mạng xã hội
* Báo cáo thống kê, phân tích chi tiết
* Đưa ra các mô hình dự đoán xu hướng

## 3.2. Thu thập dữ liệu

### 3.2.1. Cơ sở lý thuyết:

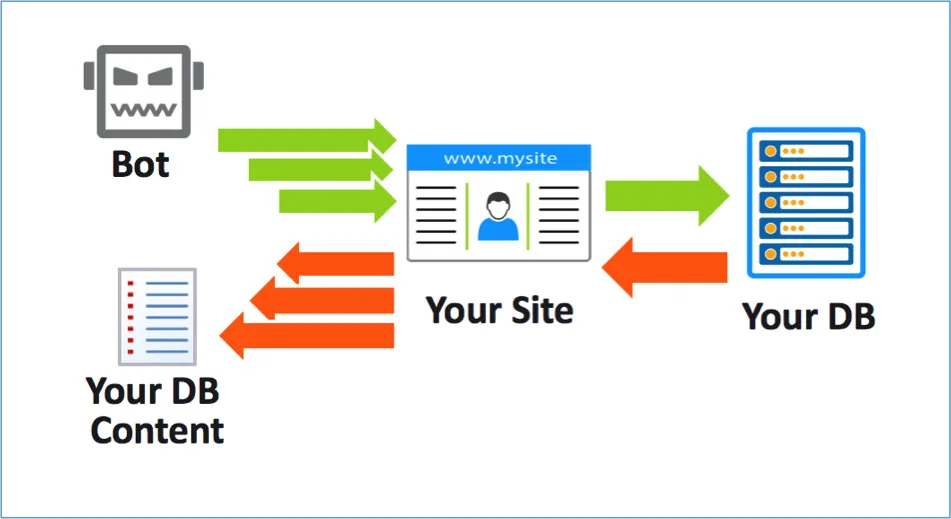
**Web Scraping là gì?**

Web scraping là một quá trình tự động thu thập thông tin từ website. Kiểu scraping phổ biến nhất là site scraping, tập trung vào sao chép nội dung web. Thông thường, các con bot sao chép dữ liệu bằng cách crawling. Crawl là một thuật ngữ mô tả quá trình thu thập dữ liệu trên website của các con bot. Các con bot truy cập vào mã nguồn website, phân tích cấu trúc, lấy nội dung phục vụ các nhu cầu khác nhau của con người.



Hình 3.2.1 Quá trình thu thập dư liệu của con bot len website

Một dạng scraping nâng cao hơn đó là database scraping. Nó khá giống với site scraping ngoài việc tạo ra con bot tương tác với phần ứng dụng nhằm lấy dữ liệu từ cơ sở dữ liệu của trang đó.



Hình 3.2.1 Kỹ thuật crawler dữ liệu từ web

Site scraping là một công cụ mạnh mẽ giúp tự động hóa việc thu thập và phát tán thông tin

Scraping dữ liệu không nhất thiết phải liên quan đến web. Scraping có thể đề cập đến việc trích xuất thông tin từ một hệ thống cục bộ, cơ sở dữ liệu chung hoặc thậm chí từ internet. Web Scaping cũng thực hiện việc tìm kiếm và thu thập thông tin nhưng khác với Web Crawling, Web Scraping không thu thập toàn bộ thông tin của một trang web mà chỉ thu thập những thông tin cần thiết, phù hợp với mục đích của người dùng. Trong Web Scraping chúng ta cũng phần nào sử dụng Web Crawler để thu thập dữ liệu, kết hợp với Data Extraction (trích xuất dữ liệu) để tập trung vào các nội dung cần thiết.

Ví dụ như đối với trang amazon.com, Web Crawling sẽ thu thập toàn bộ nội dung của trang web này (tên các sản phẩm, thông tin chi tiết, bảng giá, hướng dẫn sử dụng, các reviews và comments về sản phẩm, …). Tuy nhiên Web Scaping có thể chỉ thu thập thông tin về giá của các sản phẩm để tiến hành so sánh giá này với các trang bán hàng online khác.

**Một số trường hợp gặp phải khi scraping data**:

**Anti-bot detection**:

Khi thu thập dữ liệu thì có thể gặp những websites sử dụng các cơ chế chặn bot (dĩ nhiên họ sẽ ko chặn các Search engine nổi tiếng như Google, Bing, …). Họ có thể sẽ sử dụng các cơ chế:

* Phát hiện user-agents truy cập nhiều
* Giải pháp: dùng user-agent khác nhau hoặc dùng SE agents nổi tiếng cho mỗi lần request.
* Phát hiện IPs truy cập nhiều, giả sử 5 requests/s
* Giải pháp: dùng các dịch vụ IP rotator cho mỗi lần request.
* Phát hiện người dùng thật qua Javascript, đa số bot ko hỗ trợ JS
* Giải pháp: dùng headless browser như Splash, Selenium, PhantomJS, Puppeteer, … Có khá nhiều sites mình gặp dùng JS để detect robot như similarweb(.)com,…
* Sử dụng honeypot traps: ví dụ như các links đính kèm display: none, visibility: hidden,…
* Giải pháp: cài đặt cơ chế phát hiện các traps
* Sử dụng cookie, captcha để chặn, đa số sites dùng Cloudflare để chặn bot
* Giải pháp: Dùng script bypass Cloudflare

**Ghi kết quả dữ liệu (scraped data) quá nhiều**

Khi cào dữ liệu với nhiều spiders, ví dụ 1 spider cào được 1 record/2s, và nếu có 20 spiders thì có 10 records/s, lúc này việc ghi dữ liệu quá nhiều và liên tục vào DB sẽ làm cho DB của bị quá tải và giảm hiệu năng, có thể ảnh hưởng đến hiệu năng hoạt động. Khi đó, ta nên cân nhắc dùng:

* Bulk insert query: tức spider chỉ cần thực hiện 1 query để insert nhiều records.
* Bulk import file: tức là spider ghi dữ liệu vào 1 file với 1000 dữ liệu chẳng hạn, sau đó ta sử dụng lệnh import file đó vào DB. Ví dụ: MySQL (LOAD DATA LOCAL INFILE), MongoDB (mongoimport) Các DB engines nào cũng hỗ trợ 2 dạng trên, ví dụ như MySQL, MongoDB, …

**Cấu trúc site thay đổi**

Ví dụ như site thay đổi layout, tức HTML tags thay đổi, lúc này ta phải thay đổi các selectors để lấy đúng dữ liệu mong muốn. Trong tình huống này, spider cần có cơ chế phát hiện sự thay đổi cấu trúc site để thông báo cho chúng ta và dừng extract data của site đó. Khi đó, mỗi site ta cần hỗ trợ nhiều schemas để extract dữ liệu hơn.

* **Cách chủ website ngăn chặn site Scraping**

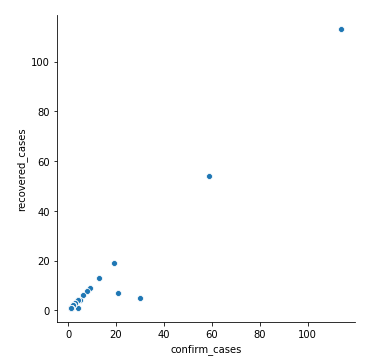
Chủ website có thể thực hiện các phương pháp sau để phân loại và giảm thiểu các con bot, bao gồm cả việc phát hiện Scraping Bot:

* Sử dụng công cụ phân tích – Các công cụ phân tích kiểm tra cấu trúc web request và thông tin header. Kết hợp các thông tin này với thông tin của các con bot trả về, chủ website có thể xác định đâu là con bot hợp pháp, đâu là con bot cần ngăn chặn.
* Triển khai cách tiếp cận “thách thức” (challenge- based) – Sử dụng các công nghệ web để đánh giá hành vi của client như nó có hỗ trợ cookie và JavaScript hay không? Chủ website cũng có thể sử dụng CAPTCHA để chặn các một vài cuộc tấn công.
* Lựa chọn cách tiếp cận hành vi – Hầu hết các con bot đều tự liên kết với các chương trình client gốc như JavaScript, Internet Explorer hay Chrome. Nếu đặc điểm của các con bot này khác biệt với client gốc, chủ website có thể sử dụng các điểm bất thường để phát hiện, ngăn chặn và giảm thiểu chúng.
* Sử dụng robots.txt – Chủ website có thể sử dụng robots.txt để bảo vệ website trước scraping bot, nhưng cách này không có hiệu quả lâu dài. Đây là tệp tin hướng dẫn các con bot thực hiện theo luật định sẵn. Trong một vài trường hợp, một vài con bot độc hại sẽ tìm kiếm thông tin trong robots.txt (thư mục riêng, trang quản trị) mà chủ website không muốn Google đánh chỉ mục và khai thác chúng.

### 3.2.2. Triển khai thực tế

#### 3.2.2.1. Thu thập và xử lý dữ liệu thống kê từ các nguồn uy tín

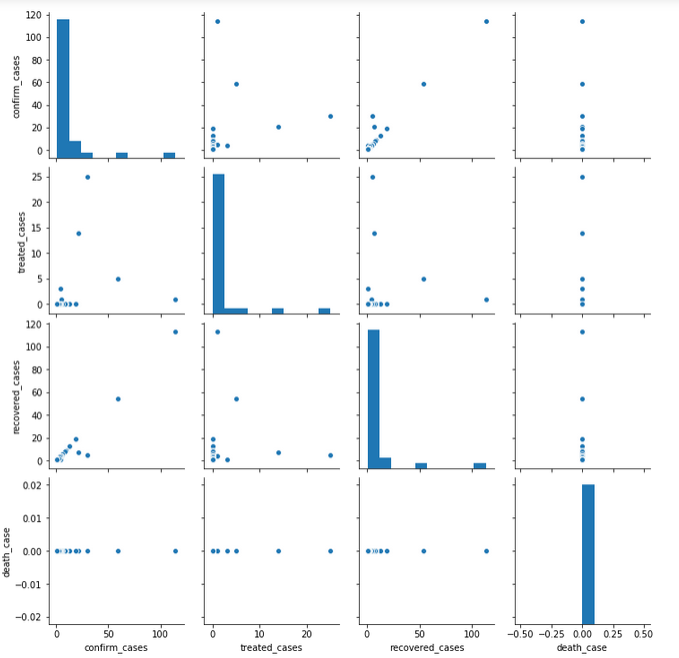
Trong phạm vi triển khai đề tài, thu thập dữ liệu thống kê số lượng các ca nhiễm tại Việt Nam. Việc xử lý các dữ liệu thống kê số lượng, thông tin các ca nhiễm bệnh tại Việt Nam khá đơn giản. Do tại Việt Nam, có hơn 300 ca nhiễm bệnh, bộ dữ liệu không nhiều.



*Ảnh: tương quan giữa số các nhiễm và số ca chữa khỏi*

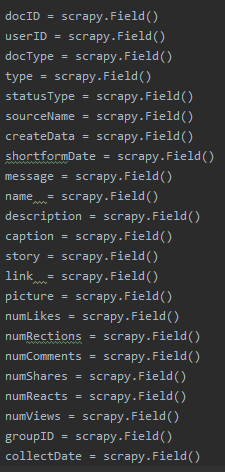
Có hai điểm dị thường trên biểu đồ, thực tế là do vụ Công ty Trường Sinh và quán Bar tại Thành phố Hồ Chí Minh khiến số ca nhiễm bệnh tại Việt Nam tăng nhanh.

Các biểu đồ khác thể hiện mối quan hệ giữa các trường trong bộ dữ liệu thống kê:



#### 3.2.2.2. Thu thập các dữ liệu trên mạng xã hội

Các trường chính của dữ liệu:



Hình 3.2.2.2 Các trường chính của dữ liệu

Gặp khó khăn, bị facebook detect được những con spider

Hiểu dữ liệu mình đang có: nhóm thu được rất nhiều dữ liệu các thảo luận công khai trên mạng xã hội, trong đó có cả những dữ liệu không cần thiết như các bài đăng bán hàng, check-in của người dùng, ...

## 3.3. Xử lý dữ liệu thô:

### 3.3.1. Nhận xét tổng quan:

Dữ liệu được lấy về, lưu trên một cơ sở dữ liệu đinh dạng .csv với các trường: docID, userID, caption, ... mà chưa qua bất kỳ khâu xử lý nào. Tiến hành tiền xử lý dữ liệu tách nội dung các bài đăng trên mạng xã hội ra những file text (.txt) khác nhau. Lọc các bài viết theo nguồn: bài đăng từ fanpage, bài đăng trong group, bài đăng cá nhân, .... Lọc các dạng bài đăng là kèm link, kèm ảnh, trạng thái, ... Phân loại nguồn dữ liệu theo giới tính, độ tuổi, ...

Tuy nhiên với số lượng các file văn bản (.txt) rất lớn (~100.000 bài mỗi ngày) với những chủ đề khác nhau. Đầu ra mong muốn là một file .csv với cột thứ nhất chứa tên các chủ đề thảo luận, cột thứ hai chứa số lượng bài viết liên quan tới chủ đề đó, cột thứ ba chứa phần trăm của chủ để trên tổng số lượng các bài viết.

Với lượng lớn dữ liệu như vậy, có 2 mục tiêu cần thực hiện:

* Nhóm các bài viết theo từng chủ đề khác nhau
* Đặt tên cho từng chủ đề, từng nhóm bài viết đó.

Việc nhóm các văn bản thành các chủ đề khác nhau có thể sử dụng các thuật toán phân cụm như K-means, thuật toán khai phá chủ đề LDA (Latent Dirichlet Allocation), ... Tuy nhiên việc sử dụng hai thuật toán trên đều phải biết trước giá trị k - số cụm để phân chia

Số cụm của bài toán không thể xác định rõ. Nguồn dữ liệu chủ yếu lấy từ các bài thảo luận công khai trên mạng xã hội.

### 3.3.2. Tiền xử lý dữ liệu:

Khi thu thập những bài thảo luận từ trên mạng xã hội, sẽ có trường hợp các bài thảo luận trùng nhau (hoặc gần giống nhau) nhưng lại lấy từ những nguồn khác nhau. (Người dùng, fanpage copy lại).

Dữ liệu thu được ở các bài ở dạng bài viết ảnh (chỉ thu thập được caption của ảnh), hoặc bài viết video.

Có nhiều bài đăng thu được ở dạng các bài bán hàng, check-in, ... Dữ liệu khá nhiễu, khó trích các đặc trưng.

## 3.4. Phương pháp thực hiện:

Do sự khó khăn trong dữ liệu, không xác định được số lượng chủ đề bài viết, các đặt tên phù hợp cho các chủ đề, ... Có nhiều phương pháp khác nhau để tiến hành nhóm các bài viết. Sau quá trình trao đổi, cả nhóm đã thống nhất chia bài viết thành các chủ đề chính (không chia thành những chủ đề thứ cấp nhỏ hơn). Và đưa ra một số biện pháp xử lý:

* **Thuật toán DBSCAN** (Density - based spatial clustering of applications with noise)

Kỹ thuật DBSCAN được đề xuất do việc không thể xác định số lượng chủ đề các bài viết trong tập dữ liệu, đồng thời chấp nhận dữ liệu nhiễu. Tuy nhiên, nhược điểm của DBSCAN là độ phức tạp cao, chạy chậm.

* **Thuật toán phân cụm** **K-means**: K-means sẽ phân cụm dữ liệu chủ đề bài viết vào số cụm k xác định.
* **Kỹ thuật phân loại văn bản** (Text classification): Kỹ thuật này có thể tự động “hiểu” được văn bản cho trước, xem văn bản đó thuộc thể loại nào. Xác định được ý nghĩa, đánh giá, bình luận của người dùng trên mạng xã hội.

Sau khi xem xét nguồn lực, cơ sở hạ tầng, nhóm đã quyết định chọn xử lý theo bài toán phân loại văn bản (Text classification)

### 3.4.1. Tiền xử lý dữ liệu (Preprocessing Data)

Một văn bản, một bài đăng ở dạng văn bản, làm sao máy tính có thể hiểu được văn bản để phân loại nó? Câu trả lời là: Máy tính chỉ có thể hiểu được dữ liệu dạng số, vì vậy cần chuyển dữ liệu dạng ký tự về dữ liệu dạng số, sau đó ta sẽ sử dụng dữ liệu dạng số này để huấn luyện cho máy tính phân loại các văn bản.

#### Chuẩn bị dữ liệu:

Trước hết, cần loại bỏ những ký tự đặc biệt trong văn bản gốc như: dấu chấm, dấu phẩy, dấu đóng/ mở ngoặc, chấm than, hỏi chấm, .... bằng cách sử dụng thư viện gensim trong Python.



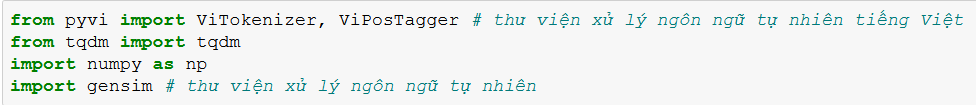
Hình 3.4.1.1 Dòng lệnh xóa các ký tự đặc biệt

Một đặc điểm quan trọng trong các văn bản tiếng Việt là một từ có nghĩa có thể được kết hợp bởi nhiều từ khác nhau, ví dụ như: hệ\_thống, triển\_khai, phân\_tích, .... khác với tiếng Anh và một số ngôn ngữ khác (các từ được phân cách với nhau bởi khoảng trắng, ví dụ như: hello world, big data analytics, .....) Vì vậy cần những kỹ thuật tách từ để đảm bảo ý nghĩa được toàn vẹn.



Hình 3.4.1.2 Dòng lệnh tách các từ tiếng việt

Các thư viện cần thiết cần được import để xử lý dữ liệu:



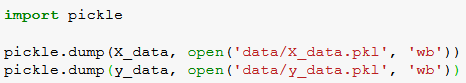
Hình 3.4.1.3 Thư viện xử lý dữ liệu

Đưa các bài đăng về dạng một cặp (x, y). Trong đó x là văn bản đã được xử lý, y là nhãn/ thể loại của văn bản đó. Chi tiết được thực hiện như sau:



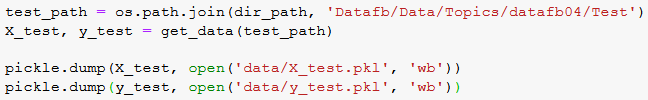
Hình 3.4.1.4 Chuyển đổi dữ liệu về dạng một cặp (x,y)

Để tiện cho việc sau này sử dụng các dữ liệu đã qua bước tiền xử lý, nhóm lưu dữ liệu vào file.pkl:



Hình 3.4.1.5. Lưu dữ liệu sau tiền xử lý vào file

Với dữ liệu test, nhóm cũng làm tương tự:



Hình 3.4.1.6. lưu dữ liệuTest vào file

### 3.4.2. Vector hóa văn bản (Feature engineering)

Sau khi xong các quá trình tiền xử lý dữ liệu, giờ cần đưa các văn bản đã được xử lý về dạng vector thuộc tính có dạng số học. Có nhiều hướng xử lý khác nhau để đưa văn bản về dữ liệu dạng số như:

* TF-IDF Vector as features (Term Frequency - Inverse Document Frequency)
* Word level
* N-Gram level
* Character level
* Text / NLP based features
* Topic Models as features
* Word Embeddings as features

Trong phạm vi dự án này, nhóm đã lựa chọn kỹ thuật TF-IDF Vector as features để biểu diễn văn bản về dạng dữ liệu số.

#### TF-IDF Vector as features (Term Frequency - Inverse Document Frequency)

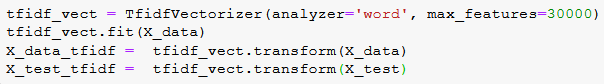
Công thức tính:

* TF(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
* IDF(t) = log\_e (Total number of documents / Number of documents with term t in it)

Thực hiện TF-IDF cho các cấp độ khác nhau của văn bản như sau:

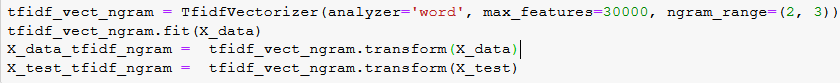
* Bước 1: Word Level TF-IDF:

Thực hiện tính toán dựa trên mỗi thành phần là một từ riêng lẻ



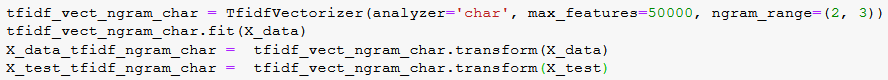
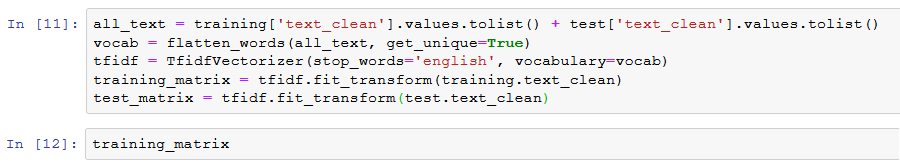
* Bước 2: N-gram Level TF-IDF:

Kết hợp *n* thành phần (từ) liên tiếp nhau. Ví dụ: "Việt\_Nam 63\_ngày liên\_tiếp không có ca\_nhiễm mới". Khi đó, 2-gram cho ta kết quả: {Việt\_Nam 63\_ngày, 63\_ngày liên\_tiếp, liên\_tiếp không có, không có ca\_nhiễm, ca\_nhiễm mới}. Mỗi phần từ là cặp 2 từ liên tiếp nhau.



* Bước 3: Character Level TF-IDF:

Dựa trên n-gram của ký tự.

Sau khi thực hiện kỹ thuật Term Frequency - Inverse Document Frequency, số chiều của vector bằng số từ được token (nhiều trường hợp lên tới 100.000, 1.000.000 hay thậm chí 10.000.000.000 chiều), đây là một vấn đề của việc biểu diễn văn bản dưới dạng vector.

Trong phạm vi dự án, với lượng dữ liệu thu được khoảng 100.000 bài đăng một ngày và với bộ từ điển bao gồm 30.000 từ, khi đó ma trận thu được sẽ có kích thước là:

100.000 x 30.000 = 3.000.000.000 (chiều)

Mỗi phần tử của ma trận được lưu dưới dạng float64 với 8 byte, khi đó bộ nhớ cần sử dụng:

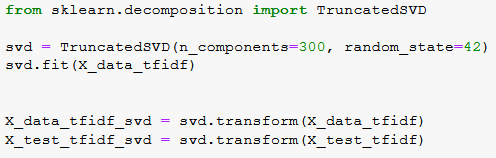
3.000.000.000 x 8 = 24.000.000.000 byte (tức 22,35 GB)

Với những khó khăn về cơ sở hạ tầng của nhóm, khó có thể lưu hết vào RAM để thực hiện tính toán, đặc biệt với những bộ từ điển đầy đủ và tối ưu hơn (số lượng từ trong từ điển lên tới >200.000 từ) thì bộ nhớ cần thiết còn tốn kém hơn rất nhiều.

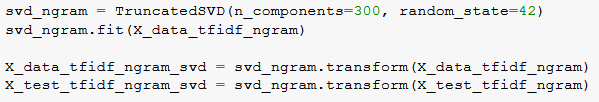
Để xử lý vấn đề giảm số chiều của ma trận, có một vài kỹ thuật như PCA (Principal Component Analysis), SVD (Singular Value Decomposition), ...

#### Thuật toán SVD để tiến hành giảm số chiều của ma trận:

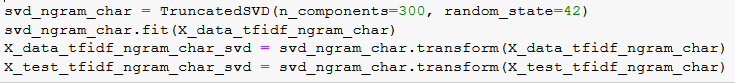
**Word level**



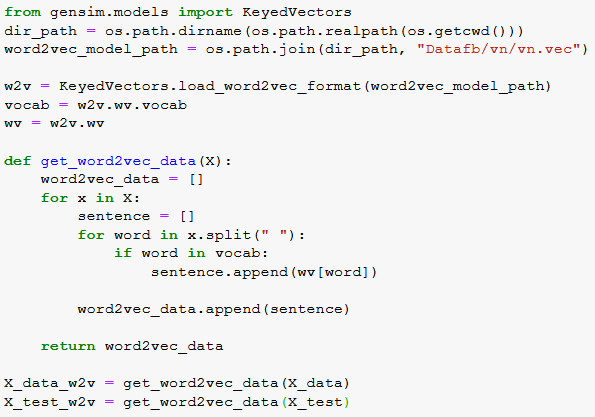
**N-gram level**



**N-gram char level**



Khi dùng thuật toán SVD, trên đây nhóm đã chuyển mỗi từ trong từ điển về một vector 300 chiều. Từ đó, có thể sử dụng cho các mô hình Deep Learning như Deep Neural Network hay Convolution Neural Networks để phân loại văn bản.



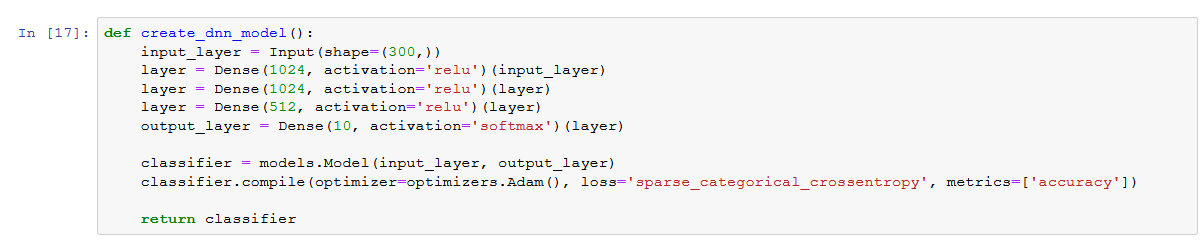
### 3.4.5. Xây dựng mô hình phân lớp văn bản

Trên thực trạng hiện nay, có khá nhiều kỹ thuật cho việc tự động phân lớp văn bản như:

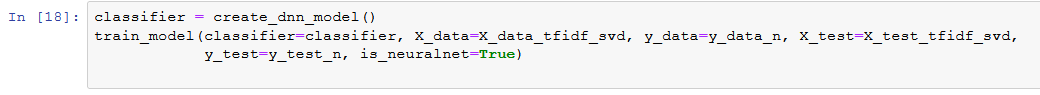
* Support Vector Machine
* Deep Neural Network:
* Convolutional Neural Network (CNN)
* Long Short Term Model (LSTM)
* Boosting Models

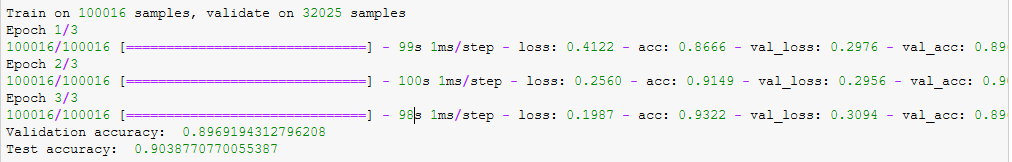
Trong dự án, nhóm lựa chọn mô hình Deep Neural Network:

Xây dựng mô hình:



Huấn luyện:

  
Kết quả:



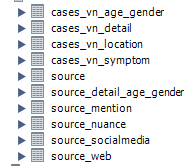
## 3.6. Xây dựng hệ thống

### 3.6.1 Thiết kế cơ sở dữ liệu

Công nghệ sử dụng: MySQL

Cơ sở dữ liệu bao gồm:

* Dữ liệu thống kê số lượng, thông tin các ca nhiễm tại Việt Nam.
* Dữ liệu phân tích trên mạng xã hội về dịch bệnh Covid-19.



### 3.6.2. Back-end

#### 3.6.2.1. Tổng quan về Nodejs

1. **Giới thiệu**

NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới.

NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ WIndow cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

Khi nói đến NodeJS thì phải nghĩ tới vấn đề Realtime. Realtime ở đây chính là xử lý giao tiếp từ client tới máy chủ theo thời gian thực. Giống như khi bạn lướt Facebook thì mỗi khi bạn comment hay like một topic nào đó thì ngay lập tức chủ topic và những người đã comment trên đó sẽ nhận được thông báo là bạn đã comment.

1. **Cách hoạt động của NodeJs**

Ý tưởng chính của Node js là sử dụng non-blocking, hướng sự vào ra dữ liệu thông qua các tác vụ thời gian thực một cách nhanh chóng. Bởi vì, Node js có khả năng mở rộng nhanh chóng, khả năng xử lý một số lượng lớn các kết nối đồng thời bằng thông lượng cao. Nếu như các ứng dụng web truyền thống, các request tạo ra một luồng xử lý yêu cầu mới và chiếm RAM của hệ thống thì việc tài nguyên của hệ thống sẽ được sử dụng không hiệu quả. Chính vì lẽ đó giải pháp mà Node js đưa ra là sử dụng luồng đơn (Single-Threaded), kết hợp với non-blocking I/O để thực thi các request, cho phép hỗ trợ hàng chục ngàn kết nối đồng thời.

Tính toán nhanh như sau: giả sử mỗi một luồng đính kèm 2MB dữ liệu được gửi lên server chạy trong hệ thống với 8GB RAM hệ thống, thì có tối đa khoảng 4000 kết nối đồng thời. Tuy nhiên với Node js thì nó có khả năng mở rộng ra cả hàng triệu kết nối cùng lúc. Thật vậy, cơ chế của Node js Application xử lý Model như sau:

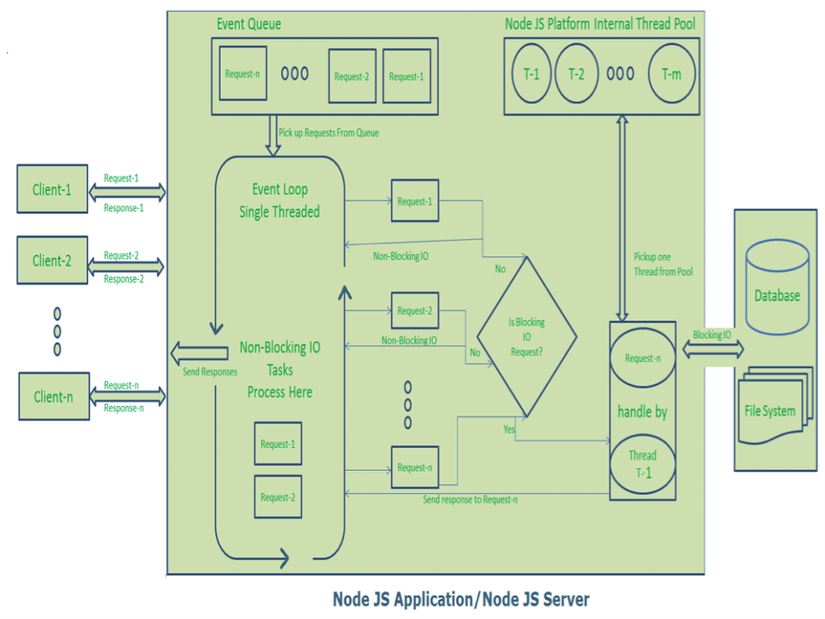
* Client gửi request đến Web Server
* Node js Web Service duy trì trong nội bộ một luồng giới hạn để cung cấp dịch vụ cho Client Request.
* Node js Web Service nhận tất cả các request và đặt chúng vào một trong Queue. Nó được gọi là một Event Queue.
* Node js Web Service nội bộ có một thành phần được gọi là "Event Loop".
* Event Loop chỉ sử dụng một luồng đơn để xử lý Model.
* Event Loop kiểm tra tất cả các Request đặt trong Event Queue. Nếu không có request nào thì chờ request đến vô thời hạn
* Nếu có request thì sẽ lấy một request từ Event Queue:
* Khởi động quá trình xử lý tiến trình từ client request
* Nếu Client Request không chứa nhiều Blocking I/O thì xử lý tất cả mọi thứ và chuẩn bị cho quá trình gửi lại phản hồi cho phía client.
* Nếu Client Request chứa nhiều Blocking I/O như việc tương tác với cơ sở dữ liệu, tập tin hệ thống, dịch vụ mở rộng, thì nó sẽ thực hiện theo các phương pháp tiếp cận khác nhau.

1. Kiểm tra các luồng sẵn có từ nội bộ bên trong của request gửi lên

2. Chọn một luồng và chỉ định cho client request tương ứng với luồng đó

3. Luồng đó phải có trách nhiệm với reuqest đó, xử lý nó, thực thi các hoạt động Blocking I/O, chuẩn bị các phản hồi và gửi lại cho Event Loop.

4. Event Loop gửi lại phản hồi tương ứng cho client.



Hình 3.6.2.1 sơ đồ nodejs

Mô tả sơ đồ như sau:

* Có n số lượng client gửi request lên web service. Chúng ta giả định rằng chúng đều truy cập đồng thời vào ứng dụng Web của chúng ta
* Chúng ta giả định client của chúng ta là Client-1, Client-2, ...client-n
* Web server của chúng ta duy trì một vùng các luồng có giới hạn và giả định rằng m là số luồng của vùng luồng đó.
* Node js Web service nhận các request từ client-1, client-2, ...client-n và đặt chúng vào trong Event Queue
* Node js Event Loop chọn các request theo dạng một đối một
* Event Loop chọn client-1 request-1
* Kiểm tra trong client-1 request-1 có không yêu cầu bất kì hoạt động Blocking I/O hoặc mất nhiều thời gian cho việc tính toán phức tạp
* Request này thì đơn giản và Non-blocking I/O, nó không có đòi hỏi một xử lý nào riêng biệt.
* Event Loop xử lý tất cả các giai đoạn để cung cấp cho hoạt động của client-1 request-1 (hoạt động ở đây nghĩa là các function của javascript) và chuẩn bị response-1.
* Event Loop gửi Response-1 đến Client-1
* Event Loop chọn client-2 request-2
* Kiểm tra trong client-2 request-2 có không yêu cầu bất kì hoạt động Blocking I/O hoặc mất nhiều thời gian cho việc tính toán phức tạp.
* Request này thì đơn giản và Non-blocking I/O, nó không có đòi hỏi một xử lý nào riêng biệt.
* Event Loop xử lý tất cả các giai đoạn để cung cấp cho hoạt động của client-2 request-2 và chuẩn bị response-2.
* Event Loop gửi Response-2 đến Client-2.
* Event Loop chọn client-n request-n
* Kiểm tra trong client-n request-n có không yêu cầu bất kì hoạt động Blocking I/O hoặc mất nhiều thời gian cho việc tính toán phức tạp.
* Request này thì phức tạp và nhiệm vụ Blocking I/O. Event Loop thì không thể xử lý request này ngay được.
* Event Loop lựa chọn Thread T-1 từ nội bộ khu vực Thread và chỉ định Client-n Request-n này ánh xạ với Thread T-1
* Thread T-1 đọc và xử lý Request-n, thực hiện Blocking IO cần thiết hoặc tính toán nhiệm vụ, và hoàn tất để chuẩn bị Response-n.
* Thread T-1 gửi Response-n này đến Event Loop
* Event Loop trong công đoạn này gửi Response-n này tới Client-n

Tại đây Client Request thì gọi đến một hoặc nhiều function javascript. Các function Javascript có thể gọi tới các function khác hoặc có thể sử dụng chức năng callback đặc thù.

Ý tưởng của Event Loop được hiểu như sau:

public class EventLoop {

while(true){

if (EventQueue nhận một JavaScript Function){

ClientRequest request = EventQueue.getClientRequest();

if (request requires BlokingIO or takes more computation time) {

chỉ định request đến Thread T1

} else {

Xử lý và chuẩn bị response

}

}

}

1. **NPM: The Node Package Manager**

Khi thảo luận về Node js thì một điều chắc chắn không nên bỏ qua là xây dựng package quản lý sử dụng các cộng cụ NPM mà mặc định với mọi cài đặt Node js. Ý tưởng của mô-đun NPM là khá tương tự như Ruby-Gems: một tập hợp các hàm có sẵn có thể sử dụng được, thành phần tái sử dụng, tập hợp các cài đặt dễ dàng thông qua kho lưu trữ trực tuyến với các phiên bản quản lý khác nhau.

Danh sách các mô-đun có thể tìm trên web [NPM package](https://npmjs.org/) hoặc có thể truy cập bằng cách sử dụng công cụ NPM CLI sẽ tự động cài đặt với Node js.

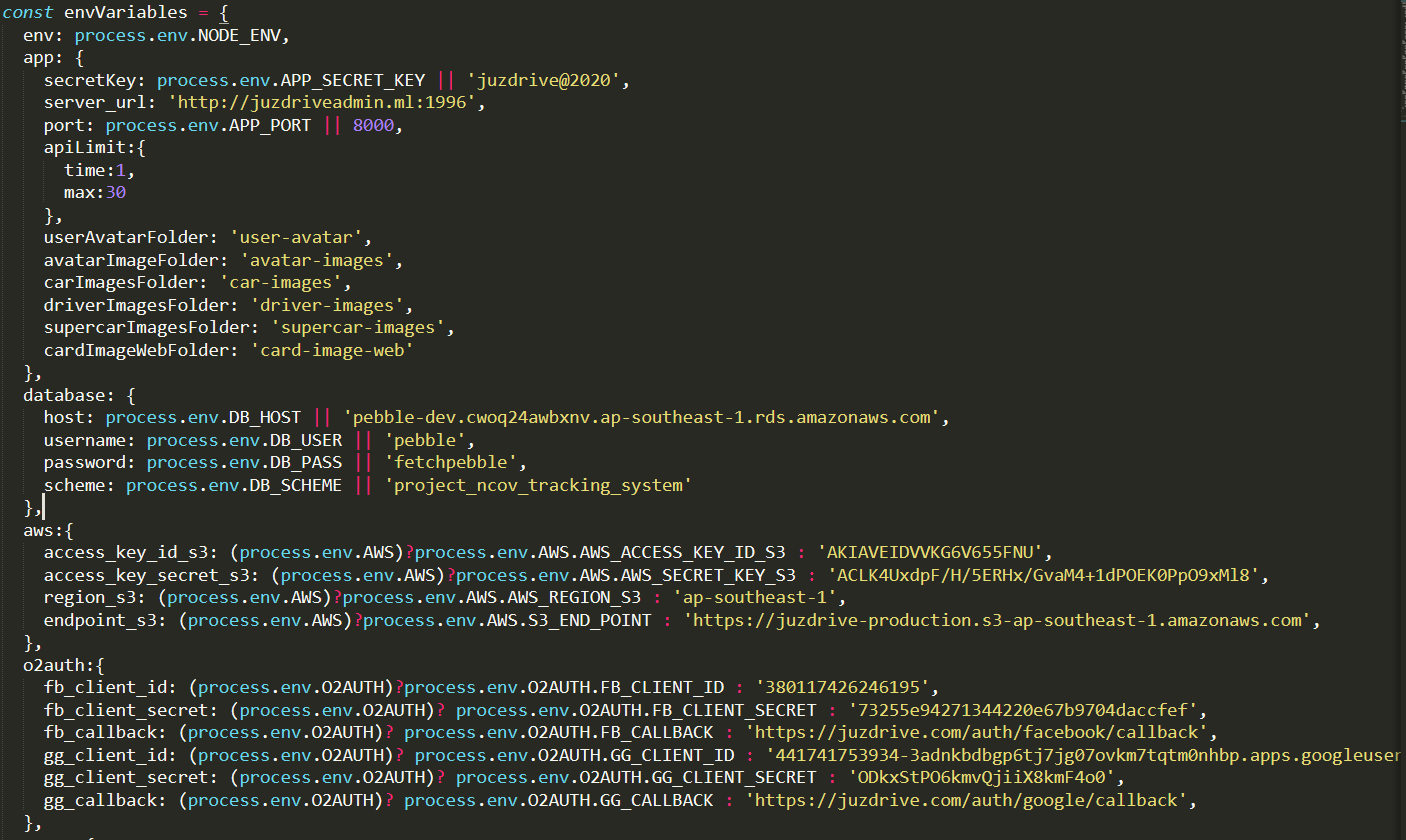
Một số module NPM phổ biến hiện nay:

* [expressjs.com/](http://expressjs.com/) - Express.js, một Sinatra-inspired web framework khá phát triển của Node.js, chứa rất nhiều các ứng dụng chuẩn của Node.js ngày nay.
* [connect](https://github.com/senchalabs/connect#readme) - Connect là một mở rộng của HTTP server framework cho Node.js, cung cấp một bộ sưu tập của hiệu suất cao "plugins" được biết đến như là trung gian; phục vụ như một nền tảng cơ sở cho Express
* [socket.io](http://socket.io/) and [sockjs](https://github.com/sockjs) - Hai thành phần Server-side websockets components nổi tiếng nhất hiện nay.
* [Jade](http://jade-lang.com/) - Một trong những engines mẫu, lấy cảm hứng từ HAML, một phần mặc định trong Express.js.
* [mongo](https://www.npmjs.com/package/mongodb) and [mongojs](https://github.com/mafintosh/mongojs) - MongoDB hàm bao để cung cấp các API cho cơ sở dữ liệu đối tượng trong MongoDB Node.js
* [redis](https://github.com/NodeRedis/node_redis) - thư viện Redis client.
* [coffee-script](https://www.npmjs.com/package/coffee-script) - CoffeeScript trình biên dịch cho phép developers viết các chương trình Node.js của họ dùng Coffee.
* [underscore](https://www.npmjs.com/package/underscore) ([lodash](https://lodash.com/), [lazy](https://www.npmjs.com/package/lazy)) - Thư viện tiện ích phổ biến nhất trong JavaScript, package được sử dụng với Node.js, cũng như hai đối tác của mình, hứa hẹn hiệu suất tốt hơn bằng cách lấy một cách tiếp cận thực hiện hơi khác nhau.
* [forever](https://www.npmjs.com/package/forever) - Có lẽ là tiện ích phổ biến nhất để đảm bảo rằng một kịch bản nút cho chạy liên tục. Giữ quá trình Node.js của bạn lên trong sản xuất đối mặt với bất kỳ thất bại không ngờ tới.

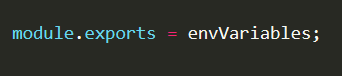
#### 3.6.2.2. Áp dụng nền tảng NodeJs để xây dựng hệ thống back-end cho dự án

1. **Tạo config**

Đường dẫn: Project/config/index.js



Tạo config với những giá trị như cổng giao tiếp(port), thông tin database(host, username, password, cheme)…



Cuối cùng sử dụng phương thức *module.exports* để trong các module khác có thể gọi đến các hàm/giá trị trong module này.

1. **Tạo kết nối tới database**

Đường dẫn: Project/src/common/DataConnection.js



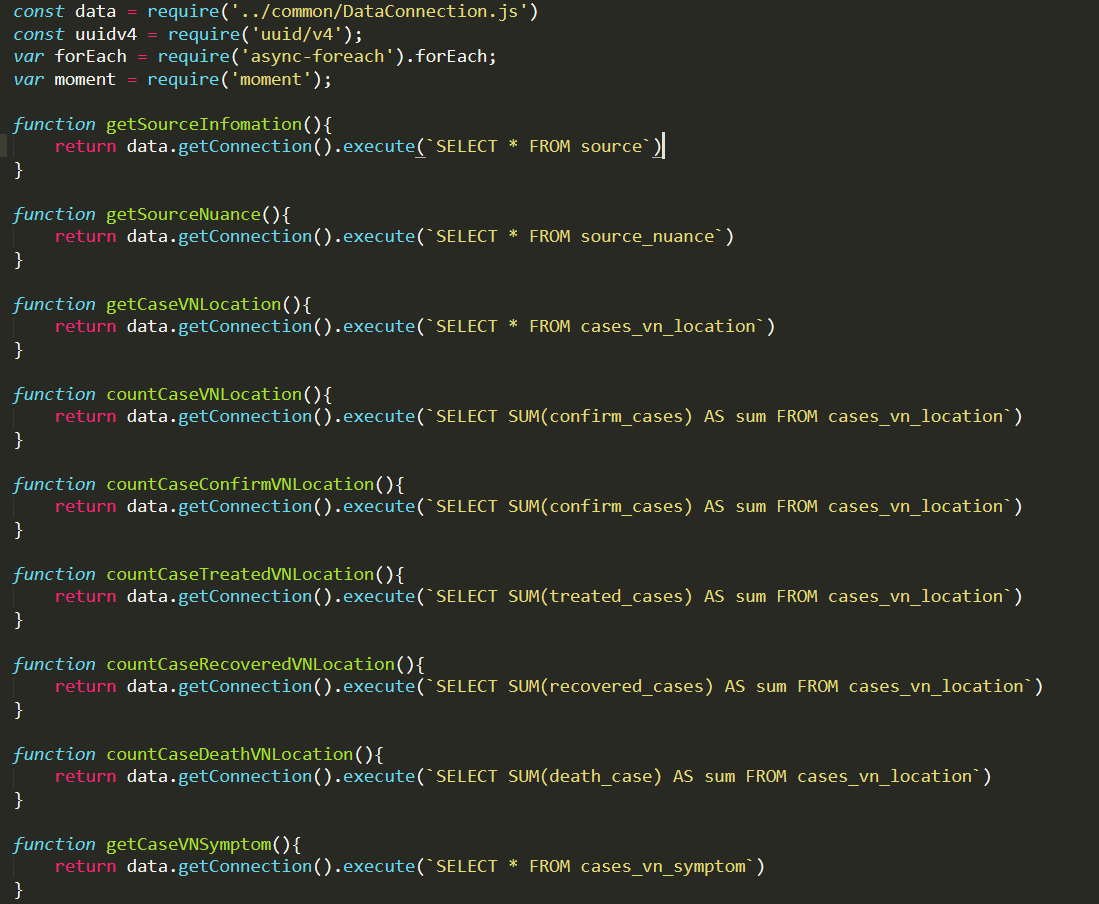
Chúng ta sử dụng hàm *require* để có thể sử các hàm trong module mặc định(mysql12/promise) của nodejs hay các module được xây dựng sau này(config)

Các giá trị của Database được gán giá trị chính bằng những giá trị đã được đưa vào từ phần connfig trước đó(Project/config/index.js)

Các hàm còn lại để kết nối với Database thông qua các giá trị được đưa vào.

1. **Tạo models**

Đường dẫn Project/src/models/info.js

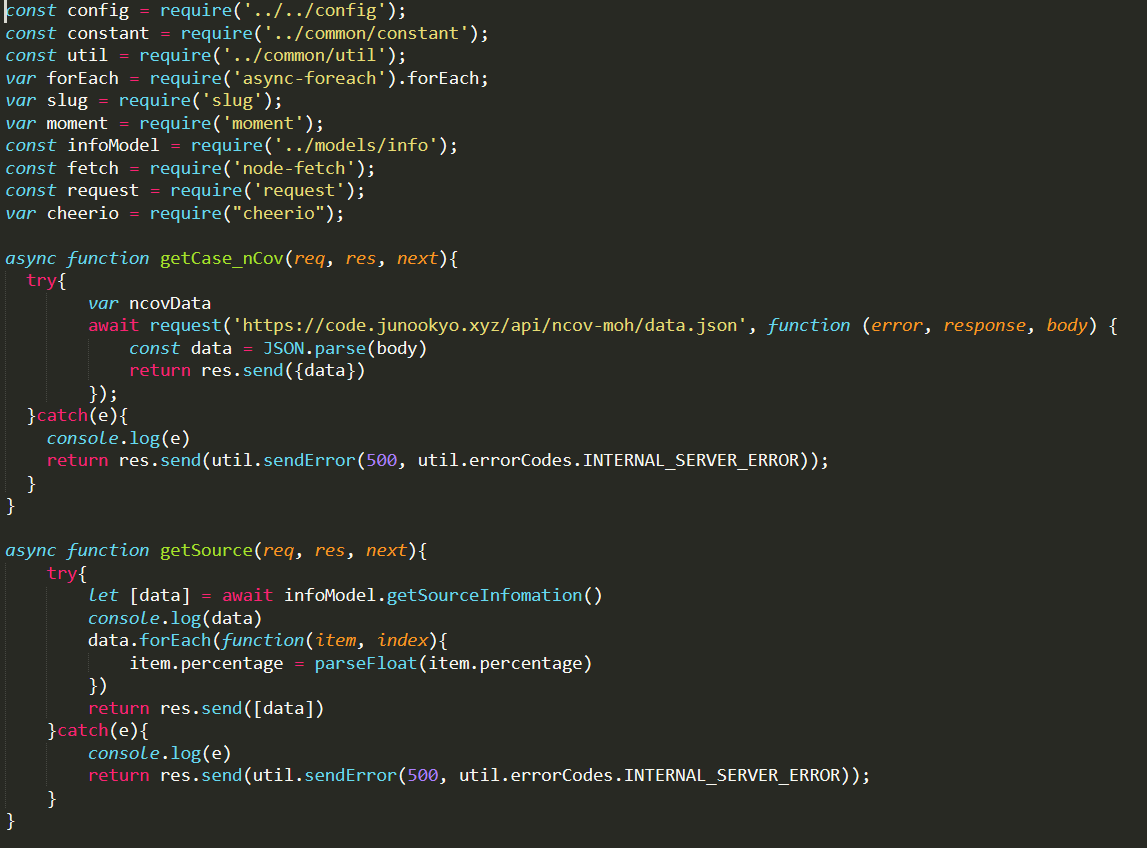


Tạo object *data* của module *DataConnection* thông qua hàm *require*.

Module này được tạo ra với mục đích lấy các dữ liệu cần thiết được lưu dữ trên database để phục vụ cho mục đích của dự án.

1. **Tạo controllers**

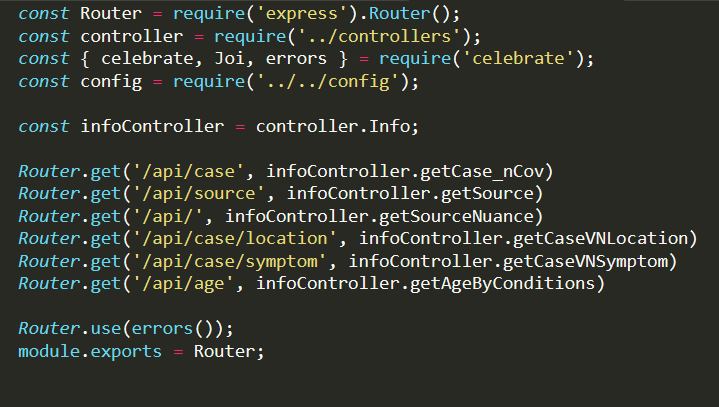
Đường dẫn: Project/src/controllers/info.js



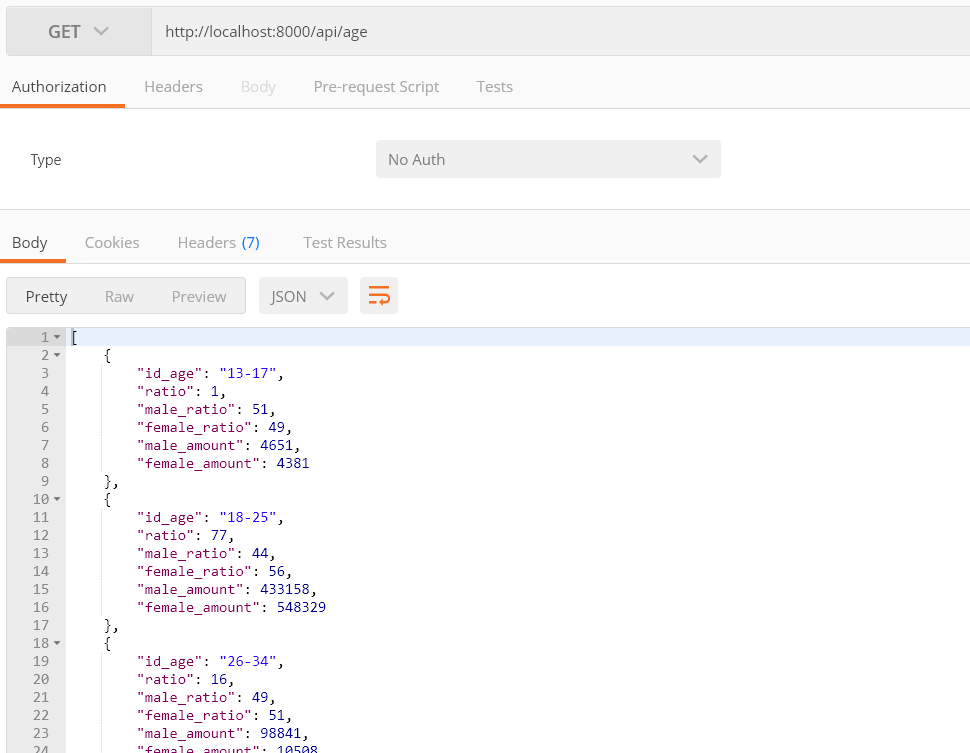
Các hàm trong controllers xử lý, định dạng dữ liệu trước khi được truyền ra và phòng tránh các ngoại lệ(exception), lỗi dữ liệu.

1. **Tạo routes**

Đường dẫn: Project/src/routes/info.js



Tạo các api khi client gọi đến thông qua đường dẫn được định nghĩa trong trong *Router.get(…)* sẽ được trả về giá trị được gán trong đó.

Ví dụ: Sử dụng postman để kiểm tra kết quả trả về của api

1. **Server.js**

Đường dẫn: Project/src/server.js



Phần main của dự án để khởi tạo, định nghĩa các đối tượng cần thiết để cho dự án hoạt động như: port, dataConnection, ...

### 3.6.3. Front-end

#### 3.6.3.1 Tổng quan về ReactJS

1. **Giới thiệu**

React JS là một framework hiển thị view chú ý đến hiệu năng (performance-minded) được tạo ra bởi Facebook. Rất nhiều đối thủ nặng ký về framework MVVM (Model-View-ViewModel) mất một thời gian lớn để hiển thị những lượng data lớn, như trong trường hợp những danh sách (list) và tương tự. Nhưng React đó không còn là vấn đề, vì nó chỉ hiển thị những gì thay đổi.

Ví dụ, nếu chúng ta đang xem một danh sách có 20 products được hiển thị bởi React, và chúng ta thay đổi product thứ 2, thì chỉ product đó được hiển thị lại, và 19 products còn lại vẫn giữ nguyên (không cần hiển thị lại). React cũng dùng cái gọi là “DOM ảo” (“virtual DOM”) để tăng hiệu năng bằng cách xuất ra một hiển thị ảo, sau đó kiểm tra sự khác biệt giữa hiển thị ảo và những gì có trên DOM và tạo một bản vá (a patch).

React yêu cầu tạo một class để định nghĩa những HTML cần hiển thị. Đó là những gì chúng ta cần tạo ra trong phần JavaScript.

Ví dụ 1:

var FirstMethod = React.createClass({

render: function() {

return <div>First Method</div>;

}

});

React.render(<FirstMethod />,document.body);

React.render(<FirstMethod />,document.body);

1. **Một số khái niệm trong React**

Sau đây là một số khái niệm cơ bản của ReactJS

* **Virtual-DOM**

DOM là tên gọi tắt của Document Object Model (Mô hình Đối tượng Tài liệu), là một chuẩn được định nghĩa bởi W3C dùng để truy xuất và thao tác trên code HTML hay XML bằng các ngôn ngữ lập trình thông dịch (scripting language) như Javascript.

DOM giúp thao tác với dữ liệu theo mô hình hướng đối tượng do các phần tử trong DOM có cấu trúc được định nghĩa thành các đối tượng, phương thức, thuộc tính để có thể truy xuất dễ dàng. Chúng được coi như các node và được biểu diễn dưới dạng DOM Tree.

Trong khi HTML là 1 đoạn code, DOM là một thể hiện trừu tượng của đoạn code đó trong bộ nhớ.

* **DOM tốt vậy thì sao lại có Virtual DOM?**

Chúng ta có thể thấy hình ở trên, HTML DOM được cấu trúc dạng cây. Thực sự rất tốt vì có thể duyệt cây rất dễ dàng. Nhưng ở đây không phải cứ dễ là tốc độ nhanh.

Tuy nhiên đừng hiểu nhầm việc đọc và ghi vào DOM của trình duyệt là chậm. Điều này không đúng. DOM nhanh. Việc cập nhật các node trong DOM không mất nhiều thời gian hơn việc thiết lập một thuộc tính trên một đối tượng JavaScript. Đó là một hoạt động đơn giản.

Điều chậm ở đây là layout mà các trình duyệt phải làm bất cứ khi nào DOM thay đổi. Mỗi khi DOM thay đổi, trình duyệt cần phải tính toán lại CSS, thực hiện dựng lại trang web. Đây là việc cần có thời gian.

Và hiện nay DOM Tree thực sự rất lớn. Việc những trang web SPA ngày càng được phát triển mạnh và đa dạng, vì vậy việc sửa đổi DOM Tree là liên tục không ngừng và sửa đổi rất nhiều.

Xem xét một DOM được tạo bởi hàng nghìn div, Có quá nhiều phương thức để xử lý các event click, submit, ... Điển hình của việc xử lý event trong jQuery sẽ là:

* Tìm tất cả các node liên quan đến event
* Cập nhật nó nếu thấy cần thiết

Chúng ta gặp 2 vấn đề:

* Thực sự rất khó để quản lý. Tưởng tượng xem nếu phải chỉnh sửa một đoạn xử lý event mà không nắm được context thì bạn sẽ phải bơi thật sâu trong code để có thể xem nó đang làm gì. Tốn thời gian và rủi ro cao.
* Nó không hiệu quả. Có nhất thiết cứ phải đi tìm tất cả những gì liên quan không? Hay có thể thông minh hơn bằng cách chỉ tìm node nào cần cập nhật.

ReactJs đến và cho chúng ta giải pháp, thay vì xử lý DOM Tree thủ công, chúng ta định nghĩa các component trông giống giống thế còn ReactJs sẽ thực hiện công việc ở tầng thấp hơn. Thực chất, công việc xử lý sẽ được HTML DOM API đảm nhiệm ở các tầng đó. Đây chính xác là cách mà Virtual DOM hoạt động.

Virtual DOM không được tạo ra bởi React tuy nhiên nó được React sử dụng và cung cấp miễn phí.

Một cách tổng quát thì nó là một định dạng dữ liệu JavaScript nhẹ được dùng để thể hiện nội dung của DOM tại một thời điểm nhất định nào đó. Nó có tất cả các thuộc tính giống như DOM nhưng không có khả năng tương tác lên màn hình như DOM.

DOM có thẻ div và các thẻ p ở trong, ReactJs sử dụng Virtual DOM bằng cách tạo ra các object React.div và React.p và khi tương tác, ta sẽ tương tác qua các object đó một cách nhanh chóng mà không phải đụng tới DOM hay DOM API của nó.

* **Lợi ích của Virtual Dom**

Việc tách biệt logic liên quan đến việc rendering ra khỏi DOM cho phép React chạy được trên nhiều môi trường khác nhau thay vì chỉ một môi trường duy nhất là trình duyệt (React Navtive hoặc SSR - Server-Side Rendering cho cả React). Chúng ta biết rằng Virtual DOM trên thực tế chỉ là một JavaScript object do đó chúng ta chỉ cần một môi trường cho chép chạy JavaScript và việc sử dụng Virtual DOM là hoàn toàn khả thi. Một số ví dụ có thể nói đến ở đây là: SSR sử dụng NodeJS hoặc embedded JavaScript khi xây dựng các ứng dụng native cho nền tảng Web cũng như Mobile. Công việc của chúng ta là thay đổi generation algorithm cho từng môi trường cụ thể nhằm thu được các output mong muốn cho từng môi trường đó.

Sử dụng Virtual DOM cho phép chúng ta tính toán các thay đổi trên đó (just JavaScript) và áp dụng đồng thời các thay đổi đó lên Actual DOM khi cần thiết. Phương pháp này sẽ hiệu quả hơn khá nhiều so với việc access Actual DOM element (sử dụng jQuery chẳng hạn) và tính toán các thay đổi trên đó.

* **Component**

React được xây dựng xung quanh các component, chứ không dùng template như các framework khác. Bạn có thể tạo ra một component bằng các gọi phương thức createClass của đối tượng React, điểm bắt đầu khi tiếp cận với thư viện này.

var Button = React.createClass({

render: function(){

return (

<input type="submit" />

);

}

});

**Cú pháp**

Để khởi tạo Component chúng ta sẽ có nhiều cách như sau:

* Dùng let, const trong JavaScript ES6 hoặc var ở các phiên bản JavaScript thấp hơn.
* Dùng Function.
* Dùng Class trong JavaScript ES6.

Chúng ta có thể tái sử dụng một component ở nhiều nơi, với các trạng thái hoặc các thuộc tính khác nhau, trong một component lại có thể chứa thành phần khác.

ReacJS tạo ra thẻ HTML không như chúng ta viết bình thường mà nó dùng Component để gói các thẻ HTML thành các object phân tầng để render

Trong các Component của React thì hàm render là quan trọng nhất. Nó chính là hàm xử lý việc tạo thẻ HTML cũng như là một minh chứng cho khả năng xử lý qua Virtual-DOM. Mọi thay đổi của dữ liệu ở bất cứ thời điểm nào đều cũng sẽ được Virtual-DOM xử lý và update ngay tức thì.

* **Hai dạng data cơ bản của React**
* Props
  + Chúng ta sử dụng props để gửi dữ liệu đến component.
  + Mọi component được coi là một hàm javascript thuần khiết (Pure Function).
  + Trong ReactJS, props tương đương với các tham số của hàm javascript thuần khiết.
  + Props là bất biến (không thể thay đổi được). Bởi vì điều này được phát triển trong khái niệm về các hàm thuần khiết. Trong các hàm thuần khiết, chúng ta không thể thay đổi dữ liệu của các tham số. Vì vậy, cũng không thể thay đổi dữ liệu của prop trong ReactJS.
  + Chúng ta không cần sử dụng this cho các hàm component cơ bản để truy cập props nhưng chúng ta phải sử dụng this để truy cập this.props.<prop\_name>.
* State
  + State giống như một kho lưu trữ dữ liệu cho các component trong ReactJS. Nó chủ yếu được sử dụng để cập nhật component khi người dùng thực hiện một số hành động như nhấp vào nút, nhập một số văn bản, nhấn một số phím, v.v.
  + React.Component là lớp cơ sở cho tất cả các lớp component cơ bản khác trong ReactJS. Bất cứ khi nào một lớp kế thừa lớp React.Component, hàm tạo (Constructor) của nó sẽ tự động gán thuộc tính state cho lớp với giá trị ban đầu được gán bằng null. Chúng ta có thể thay đổi nó bằng cách ghi đè hàm tạo (Constructor).
  + Trong nhiều trường hợp chúng ta cần cập nhật state. Để làm điều đó, chúng ta phải sử dụng phương thức setState và chúng ta không thể gán trực tiếp như thế này this.state = {'key': 'value'}.
  + Hãy cố gắng thử sử dụng khái niệm state trong component của chúng ta bằng cách thay đổi một ít mã code trong component ở trên mà chúng ta đã tạo.
  + Sau khi cập nhật component cùng với đoạn mã trên thì Ui người dùng cập nhật thêm một nút button. Khi click vào button chúng ta sẽ thấy một nội dung được hiển thị.
  + Trong đoạn mã trên chúng ta đã ghi đè hàm tạo (Constructor) và thiết lập giá trị state ban đầu: show\_technologiesbằng false. Trong khi rendering ra component React sẽ kiểm tra giá trị show\_technologies và nếu nó thiết lập giá trị là false thì React chỉ render ra nút button. Chúng ta sẽ liên kết sự kiện khi Click vào nút button. Bất cứ khi nào người dùng Click vào nút button state sẽ thay đổi thành: {"show\_technologies": true} bằng cách sử dụng phương thức setState.
  + Bây giờ, state đã được thay đổi, react sẽ render lại component với những thay đổi đó. Bất cứ khi nào state được cập nhật trong component, tất cả các component con của nó cũng sẽ render/show lại với những thay đổi mới nhất.
  + Đây là cách mà React xử lý state.

**Highcharts** là một thư viện biểu đồ được phát triển bằng JavaScript bởi Highsoft, một công ty được thành lập bởi Torstein Hønsi, cũng là người sáng tạo chính của Highcharts. Nó được phát hành lần đầu tiên vào năm 2009.

Highcharts có khả năng hiển thị các loại biểu đồ khác nhau, từ biểu đồ đường và cột cơ bản đến 3D và bản đồ nhiệt.

#### 3.6.3.2. Cài đặt Highchart trong ReactJS

npm install react-highcharts --save-dev

Sau khi cài đặt hoàn tất, gói sẽ nằm trong thư mục node\_modules. Bây giờ chúng ta sẽ tạo một component để hiển thị biểu đồ , giả sử đặt tên file là Graph.js và import Highcharts ở đó.

import React from 'react';

import Highcharts from 'highcharts';

#### 3.6.3.3 Vẽ đồ thị

Ví dụ, chúng ta hãy chọn biểu đồ hình tròn và minh họa thành phần của bầu khí quyển Trái đất với: Nitơ (78,1%), Oxy (20,9%), Argon (0,9%) và Khí Trace (0,1%). Chúng ta sẽ render biểu đồ này trong div với id = atmospheric-composition.

render() {

return (

<div id="atmospheric-composition">

</div>

);

}

Bây giờ hay tạo một biến state là series và gán một mảng cho nó một array bao gồm dữ liệu của thành phần bầu khi quyển Trái đát được cấu trúc như sau:

constructor(props) {

super(props);

this.state = {

series: [{

name: 'Gases',

data: [

{

name: 'Argon',

y: 0.9,

color: '#3498db'

},

{

name: 'Nitrogen',

y: 78.1,

color: '#9b59b6'

},

{

name: 'Oxygen',

y: 20.9,

color: '#2ecc71'

},

{

name: 'Trace Gases',

y: 0.1,

color: '#f1c40f'

}

]

}]

}

}

Biểu đồ Highcharts được khởi tạo bằng cách sử dụng hàm tạo Highcharts.chart(). Toàn bộ đối tượng (trong dấu ngoặc nhọn) được truyền dưới dạng tham số cho nó được gọi là **Option** Object; và chart, title và plotOptions là các đối tượng trong khi đó serieslà một mảng. Để dễ dàng xử lý, hàm tạo Highcharts.chart() được đặt bên trong hàm highChartsRender() để chúng ta có thể gọi ra để khởi tạo hàm tạo. Bên trong đối tượng chart, giá trị pie tương ứng với thuộc tính type biểu thị rằng biểu đồ là biểu đồ hình tròn và giá trị chuỗi atmospheric-composition tương ứng với thuộc tính renderTo chính là id của phần tử bên trong biểu đồ sẽ được hiển thị. Và để làm cho biểu đồ hình tròn của chúng ta trông giống như một chiếc bánh rán, tùy chọn innerSize bên trong đối tượng pie của đối tượng plotOptions được đặt thành 70%. Đối tượng cuối cùng là series được gán biến this.state.series là một array.

highChartsRender() {

Highcharts.chart({

chart: {

type: 'pie',

renderTo: 'atmospheric-composition'

},

title: {

verticalAlign: 'middle',

floating: true,

text: 'Earth\'s Atmospheric Composition',

style: {

fontSize: '10px',

}

},

plotOptions: {

pie: {

dataLabels: {

format: '{point.name}: {point.percentage:.1f} %'

},

innerSize: '70%'

}

},

series: this.state.series

});

}

Vì chúng ta cần render biểu đồ ngay sau khi thành phần Graph.js được chèn vào DOM, nên chúng ta sẽ gọi hàm highChartsRender() bên trong componentDidMount().

componentDidMount() {

this.highChartsRender();

}

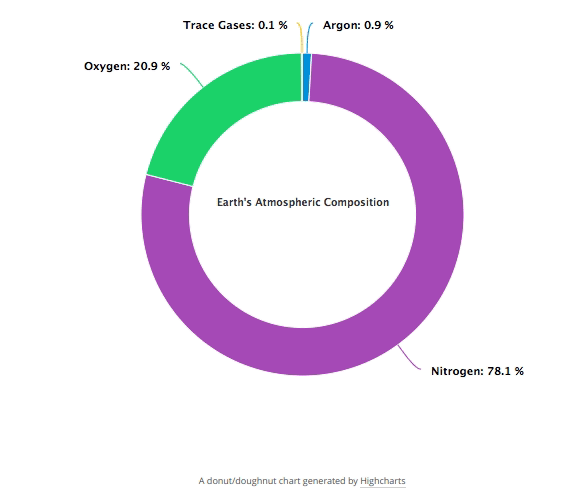
Bây giờ hãy import thành phần **Graph.js** bên trong file **index.js** của bạn

import Graph from './graph';

Và cuối cùng là render component Graph

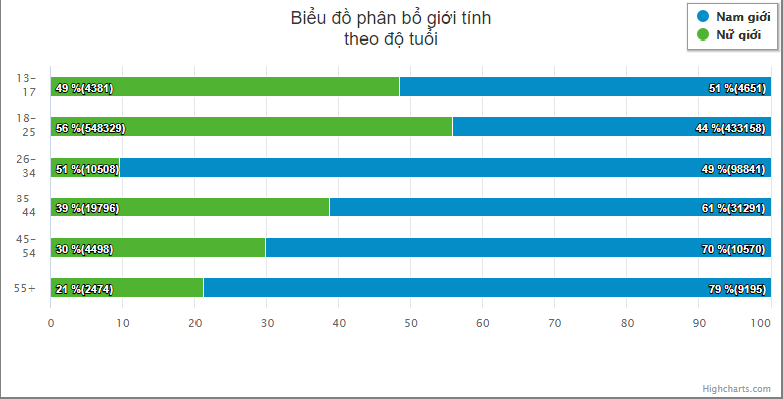
ReactDOM.render(<Graph/>, document.getElementById('root'));

Kết quả là:



**Ứng dụng trong dự án**

Điển hình: Biểu đồ phân bổ giới tính theo độ tuổi



#### 3.6.3.4 Cách gọi API từ ReactJs

1. **Cài đặt Axios để gọi API**

Chạy lệnh:

npm install axios ---save

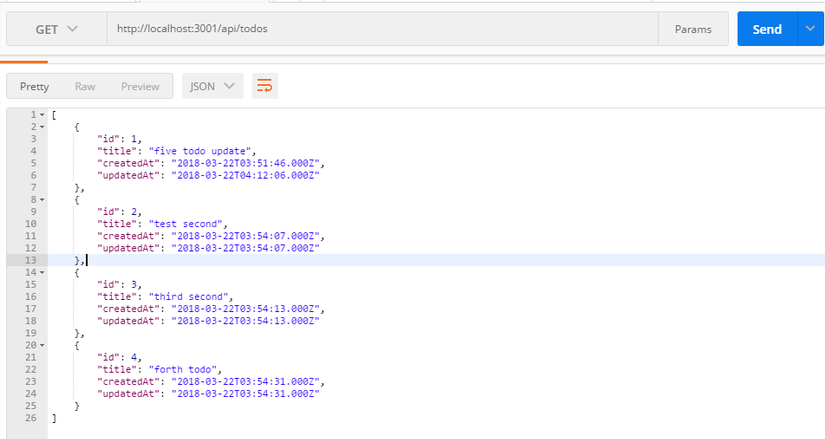
Code demo: [nodejs\_api](https://github.com/huuhv/nodejs_api)

Và fix port để chạy api, trong ví dụ này là 3001 chẳng hạn.

1. **Danh sách items**

Kết quả từ API trả về như sau:

URL: <http://localhost:3001/api/todos>



Trên frontend ta sẽ gọi bằng cách.

* File src/App.js mặc định sẽ là:

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

class App extends Component {

render() {

return (

<div className="App">

<header className="App-header">

<img src={logo} className="App-logo" alt="logo" />

<h1 className="App-title">Welcome to React</h1>

</header>

<p className="App-intro">

To get started, edit <code>src/App.js</code> and save to reload.

</p>

</div>

);

}

}

export default App;

* Và file src/App.js, ta sẽ sửa file này bằng cách import thêm thư viện axios gọi phương thức api theo kiểu restful api. Thêm function componentDidMount để gọi api từ Nodejs API.

import React, { Component } from 'react';

import logo from './logo.svg';

import './App.css';

import axios from 'axios';

class App extends Component {

constructor(){

super();

this.state ={items: []};

}

componentDidMount() {

axios.get('http://localhost:3001/api/todos')

.then(response => {

console.log(response.data);

this.setState({ items: response.data });

})

.catch(function (error) {

console.log(error);

})

}

render() {

return (

<div className="App">

<h1>Todo List</h1>

<div>

<table className="table table-hover" align="center">

<thead>

<tr>

<th width="10%">ID</th>

<th width="80%">Title</th>

<th width="10%">Actions</th>

</tr>

</thead>

<tbody>

{this.state.items.map(item =>

<tr key={item.id}>

<td>{item.id}</td>

<td>{item.title}</td>

<td><a href="#">Edit</a>&nbsp;<a href="#">Delete</a></td>

</tr>

)}

</tbody>

</table>

</div>

</div>

);

}

}

export default App;

Kết quả:



1. **ứng dụng trong dự án**

Gọi dữ liệu hiển thị trên frontend

#### 3.6.3.5 Google map React

1. **Khái quát chung**

google-map-react là một thành phần được viết trên một bộ nhỏ **API Google Maps**. Nó cho phép bạn kết xuất bất kỳ thành phần React nào trên Google Map. Nó hoàn toàn đẳng cấu và có thể kết xuất trên máy chủ. Ngoài ra, nó có thể kết xuất các thành phần bản đồ trong trình duyệt ngay cả khi API Google Maps không được tải. Nó sử dụng thuật toán di chuột nội bộ, có thể điều chỉnh mọi đối tượng trên bản đồ đều có thể di chuột được.

Với 1 ví dụ đơn giản bạn chỉ cần thêm lat với lng props đến GoogleMapReact component

Ví dụ trong jsbin:

import React, { Component } from 'react';

import GoogleMapReact from 'google-map-react';

const AnyReactComponent = ({ text }) => <div>{text}</div>;

class SimpleMap extends Component {

static defaultProps = {

center: {

lat: 59.95,

lng: 30.33

},

zoom: 11

};

render() {

return (

<div style={{ height: '100vh', width: '100%' }}>

<GoogleMapReact

bootstrapURLKeys={{ key: /\* YOUR KEY HERE \*/ }}

defaultCenter={this.props.center}

defaultZoom={this.props.zoom}

>

<AnyReactComponent

lat={59.955413}

lng={30.337844}

text="My Marker"

/>

</GoogleMapReact>

</div>

);

}

}

export default SimpleMap;

1. Cài đặt Google-map-react

npm:

npm install --save google-map-react

yarn:

yarn add google-map-react

1. **Đặc trưng**

**Hoạt động với các thành phần của bạn**

Thay vì các điểm đánh dấu mặc định của Google Maps, bóng bay và các thành phần bản đồ khác, bạn có thể hiển thị các thành phần phản ứng hoạt hình thú vị của mình trên bản đồ.

**Kết xuất đẳng cấu**

Nó ám trên máy chủ. *(Công cụ tìm kiếm chào mừng) (bạn có thể tắt javascript trong các công cụ phát triển trình duyệt và tải lại bất kỳ trang ví dụ nào để xem cách nó hoạt động)*

**Vị trí thành phần được tính độc lập với API Google Maps**

Nó kết xuất các thành phần trên bản đồ trước (và thậm chí không có) API Google Maps được tải.

**Tải xuống API Google Maps theo yêu cầu**

Không cần phải đặt một <script src= thẻ ở đầu trang. API Google Maps tải khi sử dụng Google Map React thành phần đầu tiên.

**Sử dụng API Google Maps**

Bạn có thể truy cập Google Maps map và maps các đối tượng bằng cách sử dụng onGoogleApiLoaded, trong trường hợp này bạn sẽ cần phải đặt yesIWantToUseGoogleMapApiInternalsthànhtrue

...

const handleApiLoaded = (map, maps) => {

// use map and maps objects

};

...

<GoogleMapReact

bootstrapURLKeys={{ key: /\* YOUR KEY HERE \*/ }}

defaultCenter={this.props.center}

defaultZoom={this.props.zoom}

yesIWantToUseGoogleMapApiInternals

onGoogleApiLoaded={({ map, maps }) => handleApiLoaded(map, maps)}

>

<AnyReactComponent

lat={59.955413}

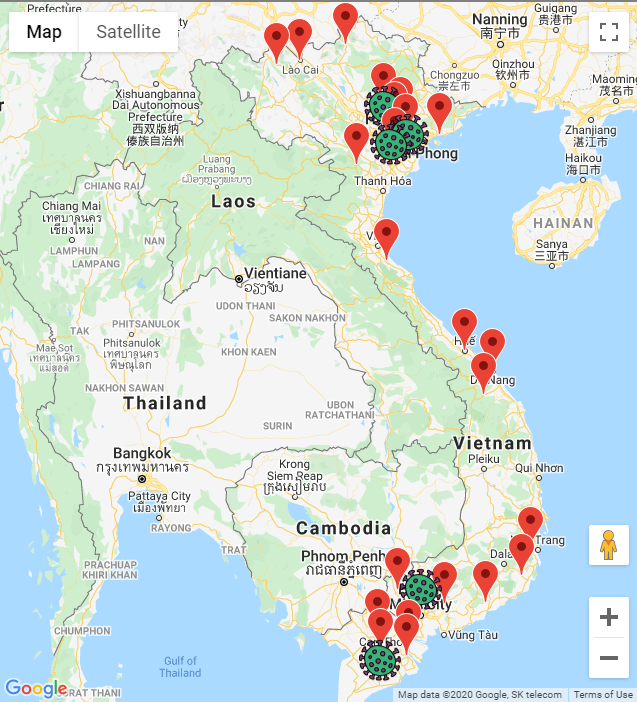
lng={30.337844}

text="My Marker"

/>

</GoogleMapReact>

**Ứng dụng trong dự án**



# CHƯƠNG 4: THỰC NGHIỆM VÀ CÁC ĐÁNH GIÁ LIÊN QUAN

# ĐẾN TÌNH HÌNH KINH TẾ CÓ THỂ ẢNH HƯỞNG TẠI THỊ TRƯỜNG VIỆT NAM

Tác động của Covid-19 cộng hưởng với đà suy giảm từ năm 2019 đã ảnh hưởng toàn diện đến tất cả các lĩnh vực kinh tế, xã hội của các quốc gia trên thế giới, trong đó có Việt Nam.

Khu vực dịch vụ chịu ảnh hưởng nặng nề nhất bởi dịch Covid-19, các hoạt động thương mại, du lịch... giảm mạnh.

Dự đoán hoạt động bán lẻ hàng hóa và doanh thu dịch vụ tiêu dùng giảm ở mức trung bình ngược lại thì hoạt động kinh doanh, thương mại điện tử tăng nhẹ do người dân sử dụng nhiều vì để tránh tiếp xúc với môi trường bên ngoài và người khác.

Giá cả thị trường biến động mạnh đặc biệt với những mặt hàng thiết bị y tế bảo vệ sức khỏe như khẩu trang ý tế, nước rửa tay, khử trùng... Giá thực phẩm tăng vì đánh vào tâm lý mua hàng hóa dự trữ của một số bộ phận người dân.

Các lĩnh vực như nông-lâm nghiệp-thủy sản gặp khó khăn trong việc xuất khẩu, đặc biệt là vận tải hàng không.

# KẾT LUẬN

Công nghệ thông tin đang thay đổi thế giới một cách mạnh mẽ, đặc biệt là mạng lưới thông tin linh hoạt và cực kì nhanh chóng. Đặc biệt là những thông tin liên quan đến đại dịch viêm đường hô hấp cấp Covid-19 đang diễn ra hiện nay.

Hệ thống theo dõi dịch bệnh covid-19 và các đánh giá liên quan đến khủng hoảng kinh tế có thể xảy ra tại Việt Nam đưa đến cho chúng ta một cái nhìn khách quan đến dịch bệnh dựa theo các thông tin chính thống như từ bộ y tế,..Với việc phân tích các dữ liệu nhận được chúng ta có thể thấy được những khu vực có người lây nhiễm hoặc có nguy cơ cao lây nhiễm để tránh và hạn chế di chuyển đến vùng dịch, thống kê độ tuổi có phần trăm lây nhiễm cao, thông qua các số liệu có thể thấy được hiệu quả ứng phó và xử lý trước dịch bệnh của nhà nước và bộ y tế.

Đối với hệ thống thông thường, việc xây dựng cơ sở dữ liệu là một bước cần thiết, tuy chưa thực sự tối ưu nhưng hệ thống có thể tồn tại trong một thời gian dài và đảm bảo tính linh động hiệu quả và việc phân tích dữ liệu cho hệ thống.

Thiết kế giao diện với bố cục hợp lý, màu sắc hài hòa tạo cảm giác thoải mái cho người sử dụng.

Trên đây là một bài phân tích và thiết kế hệ thống theo dõi dịch bệnh covid-19 và các đánh giá liên quan đến khủng hoảng kinh tế có thể xảy ra tại Việt Nam. Trong quá trình làm dự án em đã đạt được những kết quả như sau:

* Tiếp xúc và làm quen hơn với những công nghệ mới như ReactJs, NodeJs và việc phân tích dữ liệu.
* Hệ thống dễ sử dụng và thân thiện với người dùng, đáp ứng được nhu cầu thông tin cơ bản về dịch bệnh.
* Hoàn thành phân tích thiết kế hệ thống
* Hạn chế:
* Hệ thống có tính chuyên nghiệp chưa cao, chưa phân tích được chuyên sâu về ảnh hưởng của dịch covid-19 đến kinh tế Việt Nam.
* Hệ thống được tạo ra nhất thời trong thời gian dịch covid-19 diễn ra.
* Hướng phát triển:
* Nâng cấp hệ thống với những phân tích chuyên sâu hơn về ảnh hưởng của dịch bệnh.
* Mở rộng hệ thống với mục đích phân tích dữ liệu đa mục đích.
* Cải thiện giao diện người dùng, nâng cao tính chuyên nghiệp của hệ thống.

# TÀI LIỆU THAM KHẢO

1. “A study of sentiment analysis using deep learning techniques on Thai Twitter data” - Institute of Electrical and Electronics Engineers - Published in: 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE).
2. “Handling negative mentions on social media channels using deep learning” - Khuong Vo, Tri Nguyen, Dang Pham, Mao Nguyen, Minh Truong, Dinh Nguyen & Tho Quan - Pages 271-293 | Received 07 Mar 2018, Accepted 04 Jan 2019, Published online: 18 Jan 2019
3. SINGLE DOCUMENT AUTOMATIC TEXT SUMMARIZATION USING TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY (TF-IDF)” - Hans Christian; Mikhael Pramodana Agus; Derwin Suhartono - Computer Science Department, School of Computer Science, Bina Nusantara University, Jln. K.H. Syahdan No 9, Jakarta Barat, DKI Jakarta, 11480, Indonesia.

4. L. Vinh, S. Lee, H. Le, H. Ngo, H. Kim, M. Han, and Y.-K. Lee, “Semi-markov conditional random fields for accelerometer-based activity recognition,” Applied Intelligence, vol. 35, pp. 226–241, 2011.