TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA ĐIỆN – ĐIỆN TỬ**



**PHẠM HOÀNG TẤN DUY**

**CẬP NHẬT DỮ LIỆU COVID-19**

**HẰNG NGÀY, LƯU TRỮ**

**VÀ XÂY DỰNG**

**MÔ HÌNH DỰ ĐOÁN**

**ĐỒ ÁN TỔNG HỢP**

**KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM

**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

**KHOA ĐIỆN – ĐIỆN TỬ**



**PHẠM HOÀNG TẤN DUY**

**CẬP NHẬT DỮ LIỆU COVID-19**

**HẰNG NGÀY, LƯU TRỮ**

**VÀ XÂY DỰNG**

**MÔ HÌNH DỰ ĐOÁN**

**ĐỒ ÁN TỔNG HỢP**

**KỸ THUẬT ĐIỆN TỬ - VIỄN THÔNG**

Người hướng dẫn

**TS. Nguyễn Thành Nam**

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2021**

**LỜI CẢM ƠN**

Lời đầu tiên tôi xin cảm ơn các Quý thầy cô khoa Điện – Điện tử của trường Đại học Tôn Đức Thắng đã nhiệt tình giảng dạy, truyền đạt và hướng dẫn những kiến thức quý báu cho tôi trong suốt khoảng thời gian theo học.

Và đặc biệt là tôi xin chân thành cảm ơn Thầy **TS. Nguyễn Thành Nam**, thầy đã tận tình, tận tâm giúp đỡ và hướng dẫn trực tiếp tôi trong quá trình thực hiện và hoàn thiện Đồ án tổng hợp. Dưới sự hướng dẫn của Thầy, bản thân tôi không những không ngừng tiếp thu được những kiến thức mới mà còn học tập được những kinh nghiệm quý báu và phong cách làm việc chuyên nghiệp cùng với vận dụng những kiến thức đã được học trong quá trình thực hiện đồ án. Đây là một hành trang tốt đẹp và rất quý báu cho tôi để sẵn sàng hơn trong quá trình học tập và làm việc sau này.

Bên cạnh đó, tôi muốn xin chân thành cảm ơn gia đình, bạn bè đã sát cánh cùng tôi trong khoảng thời gian học tập tại trường Đại học Tôn Đức Thắng và đã đóng góp ý kiến cũng như nghị lực, hỗ trợ tôi trong khoảng thời gian thực hiện đồ án.

*TP. Hồ Chí Minh, ngày 21 tháng 12 năm 2021*

*Tác giả*

Công trình được hoàn thành tại Trường Đại học Tôn Đức Thắng

Cán bộ hướng dẫn khoa học: TS. Nguyễn Thành Nam

*(Ghi rõ học hàm, học vị, họ tên và chữ ký)*

Đồ án tổng hợp được bảo vệ tại **Hội đồng đánh giá Đồ án tổng hợp của Trường Đại học Tôn Đức Thắng** vào ngày… /…/……

Xác nhận của Chủ tịch Hội đồng đánh giá Đồ án tổng hợp và Trưởng khoa quản lý chuyên ngành sau khi nhận Đồ án tổng hợp đã được sửa chữa (nếu có).

**CHỦ TỊCH HỘI ĐỒNG TRƯỞNG KHOA**

**…………………………. ………………………………**

**CÔNG TRÌNH ĐƯỢC HOÀN THÀNH**

**TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Tôi xin cam đoan đây là công trình nghiên cứu của riêng tôi và được sự hướng dẫn khoa học của TS. Nguyễn Thành Nam. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong Đồ án tổng hợp còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung Đồ án tổng hợp của mình.** Trường Đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 21 tháng 12 năm 2021*

*Tác giả*

*Phạm Hoàng Tấn Duy*

(Trang này dùng để đính kèm Nhiệm vụ Đồ án tốt nghiệp có chữ ký của Giảng viên hướng dẫn)

|  |  |
| --- | --- |
| TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  **KHOA ĐIỆN – ĐIỆN TỬ**  ------------------- | CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM  Độc lập – Tự do – Hạnh phúc  ---------------------- |

**LỊCH TRÌNH LÀM ĐỒ ÁN TỐT NGHIỆP**

Họ tên sinh viên: PHẠM HOÀNG TẤN DUY

Lớp: 15040201 MSSV: 41502101

Tên đề tài: Cập nhật dữ liệu COVID-19 hằng ngày, lưu trữ và xây dựng mô hình dự đoán.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tuần/Ngày** | **Khối lượng** | | **GVHD ký** |
| **Đã thực hiện** | **Tiếp tục thực hiện** |
| **1**  **(11/10-17/10)** | -Nhận tờ nhiệm vụ | -Hiểu rõ yêu cầu nội dung đề tài, quy định thực hiện đồ án |  |
| **2**  **(18/10-24/10)** | -Tìm hiểu về virus và Vắc-xin COVID-19.  -Tìm hiểu về khái niệm Big Data và Data Mining. | -Tìm hiểu về ngành Khoa học Dữ liệu và Trí tuệ nhân tạo. |  |
| **3**  **(25/10-31/10)** | -Tìm hiểu về Machine Learning.  -Tìm hiểu SQL và Cơ sở dữ liệu MySQL.  -Tìm hiểu về API. | -Tìm hiểu Ngôn ngữ Lập trình Python và các thư viện sử dụng.  -Tìm nguồn dữ liệu cho đồ án từ API, Kaggle … |  |
| **4**  **(01/11-07/11)** | -Tìm hiểu về Docker và Apache Airflow. | -Thiết kế kiến trúc hệ thống và sơ đồ khối cho đồ án. |  |
| **5**  **(08/11-14/11)** | -Tìm hiểu về phần mềm Power BI.  -Thu thập dữ liệu và sử dụng Power BI để vẽ Dashboard. | -Tiếp tục và hoàn thiện Dashboard cho dữ liệu tổng quát.  -Thu thập dữ liệu chi tiết và dữ liệu Vắc-xin dịch COVID-19. |  |
| **6**  **(15/11-21/11)** | -Vẽ biểu đồ chi tiết và Vắc-xin bằng Jupyter Notebook | -Hoàn thiện các biểu đồ để khai thác đầy đủ thông tin. |  |
| **7**  **(22/11-28/11)** | -Hoàn thiện những phần đã làm.  -Tìm hiểu về ETL. | -Chuẩn bị nội dung để báo cáo tiến độ đồ án 50%. |  |
| **Kiểm tra giữa kỳ** | Đánh giá khối lượng hoàn thành……..%  được tiếp tục/không tiếp tục thực hiện ĐATN | | |
| **9**  **(29/11-05/12)** | -Tìm hiểu lý thuyết thuật toán Machine Learning. | -Lựa chọn thuật toán phù hợp cho đồ án. |  |
| **10**  **(06/12-12/12)** | -Áp dụng thuật toán SVR và Polynomial Regression vào đồ án. | -Áp dụng thuật toán Bayesian Regression vào đồ án và so sánh kết quả. |  |
| **11**  **(13/12-19/12)** | -Áp dụng thuật toán Polynomial Regression để dự đoán số ca nhiễm tại Việt Nam. | -Hoàn thiện các mô hình dự đoán.  -Đánh giá độ chính xác của các thuật toán. |  |
| **12**  **(20/12-26/12)** | -Sử dụng phần mềm Task Scheduler để thiết lập tự động hóa quy trình ETL bằng file BAT. | -Tìm hiểu Dockerfiles và Docker-compose.  -Tìm hiểu cách sử dụng Docker để cài Apache Airflow. |  |
| **13**  **(27/12-02/01)** | -Áp dụng Apache Airflow để triển khai tự động hóa quy trình ETL. | -Hoàn thiện và theo dõi đồ án. |  |
| **14**  **(03/01-09/01)** | -Tiến hành viết báo cáo. | -Tiếp tục viết báo cáo.  -Xem lại nội dung lý thuyết. |  |
| **15**  **(10/01-16/01)** | -Hoàn thiện báo cáo | -Tiến hành làm PowerPoint. |  |
| Nộp Đồ án tốt nghiệp | Đã hoàn thành……..% Đồ án tốt nghiệp  được bảo vệ/không được bảo vệ ĐATN | | |

**CẬP NHẬT DỮ LIỆU COVID-19 HẰNG NGÀY, LƯU TRỮ**

**VÀ XÂY DỰNG MÔ HÌNH DỰ ĐOÁN**

**TÓM TẮT**

Trong bối cảnh tình hình dịch COVID-19 đang diễn biến phức tạp trên toàn cầu và ở Việt Nam, gây ra nhiều sự mất mát, khó khăn và ảnh hưởng tiêu cực đến nền kinh tế toàn cầu nên tôi quyết định sử dụng sự kiện này để làm đề tài nghiên cứu cho Đồ án Tổng hợp của tôi cho HK1 năm học 2021-2022. Tuy nhiên, đề tài lúc đăng ký có tên là “Streaming COVID-19 data, lưu trữ và xây dựng mô hình dự đoán”, do đề tài của tôi chỉ cập nhật data mỗi ngày vào khung giờ nhất định chứ không cập nhật real-time nên Giáo viên phản biện có yêu cầu tôi đổi tên đề tài cho phù hợp là: Cập nhật dữ liệu COVID-19 hằng ngày, lưu trữ và xây dựng mô hình dự đoán.

**[THESIS TITLE]**

**ABSTRACT**

**(BOLD, SIZE 16)**

(Times New Roman – 13) ……………………………………………………………

………………………………………………………………………………………...

………………………………………………………………………………………...

………………………………………………………………………………………...

………………………………………………………………………………………...

MỤC LỤC

DANH MỤC HÌNH VẼ xiv

DANH MỤC BẢNG BIỂU xvii

DANH MỤC CÁC CHỮ VIẾT TẮT xviii

CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 1

1.1 Giới thiệu chung 1

1.1.1 Giới thiệu về COVID-19 1

1.1.2 Giới thiệu về Vắc-xin COVID-19 2

1.2 Mục đích thực hiện đề tài 3

1.3 Yêu cầu của đề tài 3

1.4 Phương pháp thực hiện 3

CHƯƠNG 2. LÝ THUYẾT ĐỀ TÀI 4

2.1 Giới thiệu về Big Data và Khoa học Dữ liệu 4

2.1.1 Big Data 4

2.1.2 Khoa Học Dữ Liệu (Data Science) 7

2.2 Giới thiệu về Data Mining và Machine Learning 8

2.2.1 Data Mining 8

2.2.2 Machine Learning 10

2.3 Giới thiệu Python và các thư viện của Python 12

2.3.1 Python 12

2.3.2 Các thư viện của Python 13

2.4 Giới thiệu SQL và Cơ sở dữ liệu MySQL 14

2.4.1 SQL 14

2.4.2 Cơ sở dữ liệu MySQL 16

2.5 Giới thiệu Jupyter Notebook, API và Kaggle 17

2.5.1 Jupyter Notebook 17

2.5.2 API 17

2.5.3 Kaggle 18

2.6 Giới thiệu về Docker và Apache Airflow 19

2.6.1 Docker 19

2.6.2 Apache Airflow 20

CHƯƠNG 3. THU THẬP, PHÂN TÍCH VÀ BIỂU ĐỒ 21

3.1 Thiết kế hệ thống 21

3.1.1 Kiến trúc hệ thống 21

3.1.2 Sơ đồ khối 22

3.2 Tổng quát về dịch COVID-19 23

3.2.1 Thu thập dữ liệu từ RapidAPI và lưu trữ vào Cơ sở dữ liệu MySQL 23

3.2.2 Thiết kế Dashboard bằng phần mềm Power BI 25

3.3 Chi tiết về dịch COVID-19 32

3.3.1 Giới thiệu 32

3.3.2 Biểu đồ và phân tích với Python 32

3.3.2.1 So sánh Việt Nam và Châu Á 36

3.3.2.2 So sánh Việt Nam và Đông Nam Á (South East Asia) 38

3.4 Tình hình tiêm ngừa Vắc-xin COVID-19 40

3.4.1 Tình hình tiêm ngừa Vắc-xin COVID-19 của Thế giới 40

3.4.2 So sánh Việt Nam và Châu Á 44

3.4.3 So sánh Việt Nam và Đông Nam Á (South East Asia) 46

3.5 Tự động hóa tiến trình Extract-Transform-Load 48

3.5.1 Giới thiệu về Extract-Transform-Load 48

3.5.2 Tự động hóa bằng Apache Airflow và triển khai bằng Docker 49

3.5.3 Tự động hóa bằng Task Scheduler 52

CHƯƠNG 4. XÂY DỰNG MÔ HÌNH DỰ ĐOÁN 53

4.1 Thuật toán SVM, Linear Regression và Bayesian Ridge Regression 53

4.1.1 Thuật toán SVM 53

4.1.2 Thuật toán Linear Regression 57

4.1.2.1 Giới thiệu Linear Regression 57

4.1.2.2 GIới thiệu OLS 59

4.1.3 Thuật toán Bayesian Ridge Regression 60

4.2 Áp dụng vào mô hình dự đoán 62

4.2.1 Áp dụng thuật toán SVR 62

4.2.2 Áp dụng thuật toán Linear Regression 63

4.2.3 Áp dụng thuật toán Bayesian Ridge Regression 65

4.3 Kết quả dự đoán và so sánh 66

4.4 Áp dụng Polynomial Regression dự đoán số ca nhiễm ở Việt Nam 68

CHƯƠNG 5. KẾT LUẬN 71

5.1 Kết luận 71

5.2 Hướng phát triển 72

TÀI LIỆU THAM KHẢO 73

PHỤ LỤC A MÃ NGUỒN PYTHON 1

# DANH MỤC HÌNH VẼ

[**Hình 1‑1. Hình ảnh của virus COVID-19 1**](#_Toc93131278)

[**Hình 1‑2. Các loại Vắc-xin COVID-19 2**](#_Toc93131279)

[**Hình 2‑1. Big Data 4**](#_Toc93131280)

[**Hình 2‑2. 5V’s of Big Data 6**](#_Toc93131281)

[**Hình 2‑3. Data Science và các lĩnh vực liên quan 8**](#_Toc93131282)

[**Hình 2‑4. Các bước thực hiện Data Mining 9**](#_Toc93131283)

[**Hình 2‑5. Các loại thuật toán được sử dụng trong Machine Learning 11**](#_Toc93131284)

[**Hình 2‑6. Lựa chọn thuật toán phù hợp 11**](#_Toc93131285)

[**Hình 2‑7. Hình ảnh định nghĩa SQL 15**](#_Toc93131286)

[**Hình 2‑8. Định nghĩa về API 18**](#_Toc93131287)

[**Hình 2‑9. Quy trình thực thi của hệ thống sử dụng Docker. 20**](#_Toc93131288)

[**Hình 2‑10. Kiến trúc của Apache Airflow 20**](#_Toc93131289)

[**Hình 3‑1. Kiến trúc hệ thống 21**](#_Toc93131290)

[**Hình 3‑2. Sơ đồ khối 22**](#_Toc93131291)

[**Hình 3‑3. Định dạng JSON của Response 23**](#_Toc93131292)

[**Hình 3‑4. Dữ liệu trong MySQL 24**](#_Toc93131293)

[**Hình 3‑5. Tính năng của Power BI 25**](#_Toc93131294)

[**Hình 3‑6. Dashboard về số ca nhiễm, ca tử vong... 26**](#_Toc93131295)

[**Hình 3‑7. Dashboard về tỷ lệ tử vong, tỷ lệ mắc bệnh... 27**](#_Toc93131296)

[**Hình 3‑8. Dashboard về tổng ca nhiễm của các châu lục 29**](#_Toc93131297)

[**Hình 3‑9. Dashboard về tổng ca tử vong của các châu lục 30**](#_Toc93131298)

[**Hình 3‑10. Dashboard về tổng ca hồi phục của các châu lục 31**](#_Toc93131299)

[**Hình 3‑11. Tương quan giữa các khu vực trên Thế giới 32**](#_Toc93131300)

[**Hình 3‑12. Số ca nhiễm, tử vong và đường Moving Average của Thế giới 32**](#_Toc93131301)

[**Hình 3‑13. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Thế giới 33**](#_Toc93131302)

[**Hình 3‑14. Tỉ lệ tử vong theo ngày và trung bình của Thế giới 33**](#_Toc93131303)

[**Hình 3‑15. Top 10 các Tỉnh có số ca cao nhất trên Thế giới 34**](#_Toc93131304)

[**Hình 3‑16. Top 10 Bang của Mỹ có số ca cao nhất 34**](#_Toc93131305)

[**Hình 3‑17. Top 10 nước và phần còn lại về số ca nhiễm 35**](#_Toc93131306)

[**Hình 3‑18. Top 10 tỉnh và phần còn lại về số ca nhiễm 35**](#_Toc93131307)

[**Hình 3‑19. Số ca nhiễm, tử vong và đường Moving Average của Châu Á 36**](#_Toc93131308)

[**Hình 3‑20. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Châu Á 36**](#_Toc93131309)

[**Hình 3‑21. Số ca nhiễm và tử vong giữa Việt Nam và các nước Châu Á 37**](#_Toc93131310)

[**Hình 3‑22. Biểu đồ cột giữa Việt Nam và các nước châu Á 37**](#_Toc93131311)

[**Hình 3‑23. Số ca nhiễm, tử vong và Moving Average của Đông Nam Á 38**](#_Toc93131312)

[**Hình 3‑24. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Đông Nam Á 38**](#_Toc93131313)

[**Hình 3‑25. Số ca nhiễm và tử vong giữa Việt Nam và các nước Đông Nam Á 39**](#_Toc93131314)

[**Hình 3‑26. Biểu đồ cột giữa Việt Nam và các nước Đông Nam Á 39**](#_Toc93131315)

[**Hình 3‑27. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin 40**](#_Toc93131316)

[**Hình 3‑28. Biểu đồ số lượng Quốc gia sử dụng Vắc-xin đơn 40**](#_Toc93131317)

[**Hình 3‑29. Biểu đồ top 10 Quốc gia có tổng số Vắc-xin đã tiêm 41**](#_Toc93131318)

[**Hình 3‑30. Biểu đồ top 10 Quốc gia có số Vắc-xin tiêm mỗi ngày 41**](#_Toc93131319)

[**Hình 3‑31. Biểu đồ top 10 Quốc gia có số người đã tiêm đầy đủ 41**](#_Toc93131320)

[**Hình 3‑32. Biểu đồ số lượng Quốc gia sử dụng Vắc-xin kết hợp 42**](#_Toc93131321)

[**Hình 3‑33. Biểu đồ số lượng Vắc-xin mỗi ngày của Thế giới 42**](#_Toc93131322)

[**Hình 3‑34. Biểu đồ cộng dồn lượng Vắc-xin được tiêm mỗi ngày của Thế giới 43**](#_Toc93131323)

[**Hình 3‑35. Biểu đồ phần trăm dân số Thế giới được tiêm đầy đủ 43**](#_Toc93131324)

[**Hình 3‑36. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin ở Châu Á 44**](#_Toc93131325)

[**Hình 3‑37. Biểu đồ số lượng Vắc-xin mỗi ngày của các nước Châu Á 44**](#_Toc93131326)

[**Hình 3‑38. Biểu đồ số lượng Vắc-xin cộng dồn của các nước Châu Á 45**](#_Toc93131327)

[**Hình 3‑39. Biểu đồ phân phối tiêm Vắc-xin mỗi ngày của Việt Nam 45**](#_Toc93131328)

[**Hình 3‑40. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin ở Đông Nam Á 46**](#_Toc93131329)

[**Hình 3‑41. Biểu đồ số lượng Vắc-xin mỗi ngày của Đông Nam Á 46**](#_Toc93131330)

[**Hình 3‑42. Biểu đồ số lượng Vắc-xin cộng dồn của Đông Nam Á 47**](#_Toc93131331)

[**Hình 3‑43. Biểu đồ Scatter về tổng số lượng Vắc-xin của Đông nam Á 47**](#_Toc93131332)

[**Hình 3‑44. Quy trình thực hiện ETL 48**](#_Toc93131333)

[**Hình 3‑45. Các thành phần của Docker 49**](#_Toc93131334)

[**Hình 3‑46. Cách tạo Docker Container từ Dockerfile 50**](#_Toc93131335)

[**Hình 3‑47. Log của DAG khi đã thực hiện thành công 51**](#_Toc93131336)

[**Hình 3‑48. Dữ liệu được lưu trữ trong CSDL Postgres 51**](#_Toc93131337)

[**Hình 3‑49. File BAT sử dụng trong Task Scheduler 52**](#_Toc93131338)

[**Hình 3‑50. Thiết lập trong Task Scheduler 52**](#_Toc93131339)

[**Hình 4‑1. Siêu phẳng (hyper-plane) trong SVM 53**](#_Toc93131340)

[**Hình 4‑2. Margin của siêu phẳng trong SVM 54**](#_Toc93131341)

[**Hình 4‑3. Siêu phẳng có chứa tín hiệu nhiễu 55**](#_Toc93131342)

[**Hình 4‑4. Non-linear trong SVM 56**](#_Toc93131343)

[**Hình 4‑5. Mô hình Hồi quy tuyến tính Linear Regression 58**](#_Toc93131344)

[**Hình 4‑6. R-Square (R2) của Mô hình hồi quy tuyến tính 58**](#_Toc93131345)

[**Hình 4‑7. Phương pháp Ordinary Least Square 59**](#_Toc93131346)

[**Hình 4‑8. So sánh các trọng số của mô hình 61**](#_Toc93131347)

[**Hình 4‑9. Curve Fitting với Hồi quy Bayesian Ridge 61**](#_Toc93131348)

[**Hình 4‑10. Dự đoán của thuật toán SVR 62**](#_Toc93131349)

[**Hình 4‑11. Dự đoán của thuật toán Polynomial Regression 63**](#_Toc93131350)

[**Hình 4‑12. Các mô hình Underfitted, well-fitted and overfitted 64**](#_Toc93131351)

[**Hình 4‑13. Dự đoán của thuật toán Bayesian Ridge Regression 65**](#_Toc93131352)

[**Hình 4‑14. Dự đoán số ca nhiễm ở Việt Nam 68**](#_Toc93131353)

# DANH MỤC BẢNG BIỂU

[Bảng 4‑1. Bảng so sánh kết quả dự đoán 66](#_Toc92983596)

[Bảng 4‑2. Bảng so sánh và sai số theo ngày của các thuật toán so với thực tế 67](#_Toc92983597)

[Bảng 4‑3. Bảng kết quả dự đoán số ca nhiễm của Việt Nam 69](#_Toc92983598)

[Bảng 4‑4. Bảng so sánh và sai số theo ngày của các thuật toán so với thực tế 70](#_Toc92983599)

# DANH MỤC CÁC CHỮ VIẾT TẮT

API Application Programming Interface

BI Business Intelligence

HTTP Hyper Text Transfer Protocol

SARS Severe Acute Respiratory Syndrome

SMTP Simple Mail Transfer Protocol

SQL Structured Query Language

SVR Support Vector Regression

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu chung

### Giới thiệu về COVID-19

COVID-19 (viết tắt của **CO**rona**VI**rus **D**isease 20**19)** là bệnh đường hô hấp cấp tính truyền nhiễm gây ra bởi chủng virus corona SARS-CoV-2 và các biến thể, là loại virus mới gây viêm đường hô hấp cấp ở người và lây lan từ người sang người, được phát hiện điều tra ổ dịch bắt nguồn từ khu chợ hải sản và động vật ở Vũ Hán (Wuhan), tỉnh Hồ Bắc (Hubei), Trung Quốc vào tháng 12/2019, là virus.



Hình 1‑1. Hình ảnh của virus COVID-19

Phương thức lây truyền chính của COVID-19 là từ người sang người bằng 3 cách: Truyền trực tiếp (qua hô hấp từ các giọt dịch khi tiếp xúc gần), Truyền tiếp xúc (mầm bệnh bám trên 1 vật thể rồi người khác chạm vào và đưa vào hệ hô hấp như miệng, mũi), Truyền khí dung (mầm bệnh bay lửng lơ trong không khí, người khác hít phải hoặc tiếp xúc qua mắt khi đưa tay dụi mắt).

Do cách thức lây nhiễm qua đường hô hấp khiến COVID-19 là căn bệnh rất dễ lây lan trong những không gian hẹp, kín, những chỗ đông người và tiếp xúc gần. Để phòng chống và hạn chế việc lây lan của dịch COVID-19, Bộ Y tế gửi đến thông điệp 5K (**K**hẩu trang, **K**hử khuẩn, **K**hoảng cách, **K**hông tụ tập và **K**hai báo Y tế) trong trạng thái “Bình thường mới”.

### Giới thiệu về Vắc-xin COVID-19

Vắc-xin COVID-19 là chế phẩm sinh học cung cấp khả năng miễn dịch, loại Vắc-xin truyền tin RNA (mRNA) dạy các tế bào cách tạo protein giúp kích hoạt phản ứng miễn dịch bên trong cơ thể, mang lại lợi ích cho người được tiêm chủng bằng cách cung cấp khả năng bảo vệ các loại virus mà không phải chịu những hậu quả nghiêm trọng có thể xảy ra khi mắc bệnh.

Để phòng chống, tiếp cận an toàn và hiệu quả COVID-19 là yếu tố quan trọng để chấm dứt đại dịch COVID-19, hiện tại có nhiều Vắc-xin đã được chứng minh và đang phát triển, Tổ chức Sức khỏe Thế giới WHO (**W**orld **H**ealth **O**rganization) cùng với các đối tác đang làm việc để phát triển, sản xuất và triển khai Vắc-xin an toàn và hiệu quả. Hiện nay, đã có nhiều loại Vắc-xin được WHO phê duyệt và chứng nhận an toàn để cấp phép sử dụng như AstraZeneca, Moderna, Pfizer/BioNTech, Sputnik V, Sinopharm, Janssen, SinoVac …

Vào đầu tháng 1/2020, ở nhiều quốc gia trên Thế giới đã bắt đầu triển khai việc tiêm Vắc-xin để phòng chống COVID-19, tiêu biểu như Anh, Pháp, Mỹ… Và cho đến hiện nay, để sống chung với dịch thì đã có nhiều quốc gia tham gia tiêm ngừa Vắc-xin trong đó có Việt Nam và đưa Thế giới trở lại trạng thái “Bình thường mới”.



Hình 1‑2. Các loại Vắc-xin COVID-19

## Mục đích thực hiện đề tài

Trong bối cảnh tình hình dịch COVID-19 đang diễn biến rất phức tạp cùng với nhiều biến thể và đặc biệt là biến thể Omicron đã gây ra không ít sự khó khăn, mất mát và ảnh hưởng tiêu cực đến kinh tế cũng như gây nên sự rối loạn trong cuộc sống của Thế giới nói chung và Việt Nam nói riêng.

Bên cạnh đó cùng với sự phát triển mạnh về Khoa Học Công Nghệ, nhiều quốc gia trên Thế giới như Anh, Mỹ… đã nghiên cứu và sản xuất nhiều loại Vắc-xin nhằm ngăn chặn sự lây lan nhanh của các biến thể COVID-19 và đưa Thế giới trở lại “Bình thường mới”.

Mục đích thực hiện đề tài là đem đến một cái nhìn tổng quát và chi tiết về tình hình dịch COVID-19 cùng với tình hình tiêm chủng Vắc-xin ngừa COVID-19 ở trên Thế giới, bên cạnh đó là Việt Nam so với khu vực Châu Á và Đông Nam Á.

## Yêu cầu của đề tài

Yêu cầu đề tài là tìm hiểu về Dữ liệu lớn (Big Data), tìm hiểu về Data Mining, sử dụng phần mềm Power BI để vẽ biểu đồ để thể hiện được cái nhìn tổng quát về dịch COVID-19, sử dụng ngôn ngữ lập trình Python cùng các thư viện đi kèm để thực hiện phân tích, vẽ biểu đồ để có cái nhìn chi tiết hơn về dịch COVID-19 và tình hình tiêm ngừa Vắc-xin của các quốc gia trên Thế giới nói chung và Việt Nam nói riêng, lưu trữ dữ liệu bằng Cơ sở dữ liệu MySQL và sử dụng thuật toán Machine Learning để dự đoán số ca nhiễm của Thế giới và Việt Nam trong thời gian gần.

## Phương pháp thực hiện

Phương pháp thực hiện là:

* Sử dụng ngôn ngữ lập trình Python để trích xuất dữ liệu từ API và nguồn dữ liệu từ các trang web uy tín, tin cậy như Kaggle.
* Sử dụng MySQL là Cơ sở dữ liệu để lưu trữ dữ liệu.
* Sử dụng các thư viện trong Python và phần mềm Power BI để vẽ các biểu đồ về tình hình dịch COVID-19 và Vắc-xin.

# LÝ THUYẾT ĐỀ TÀI

## Giới thiệu về Big Data và Khoa học Dữ liệu

### Big Data

Big Data nghĩa là Dữ liệu lớn dùng để mô tả khối lượng dữ liệu lớn, bao gồm có cấu trúc và không có cấu trúc, lượng dữ liệu này cung cấp thông tin cho doanh nghiệp, tổ chức theo cơ sở hằng ngày, hằng giờ, hằng phút, thậm chí hằng giây.

Trong kinh doanh, Big Data dùng để phân tích để tìm ra những cái nhìn sâu sắc hay thầm hiểu (được gọi là insight trong ngành Marketing) dẫn đến các quyết định tốt hơn và chiến lược kinh doanh. Trong ngành Y tế, Big Data được sử dụng để chẩn đoán, theo dõi và dự đoán tình hình sức khỏe của bệnh nhân. Trong ngành Quản lý thiên tai, Big Data giúp những nhà khí tượng học có thể theo dõi, dự đoán để đưa ra những cảnh báo giúp cho con người có thế tránh những thiệt hại về người và của do thiên tai xảy ra.

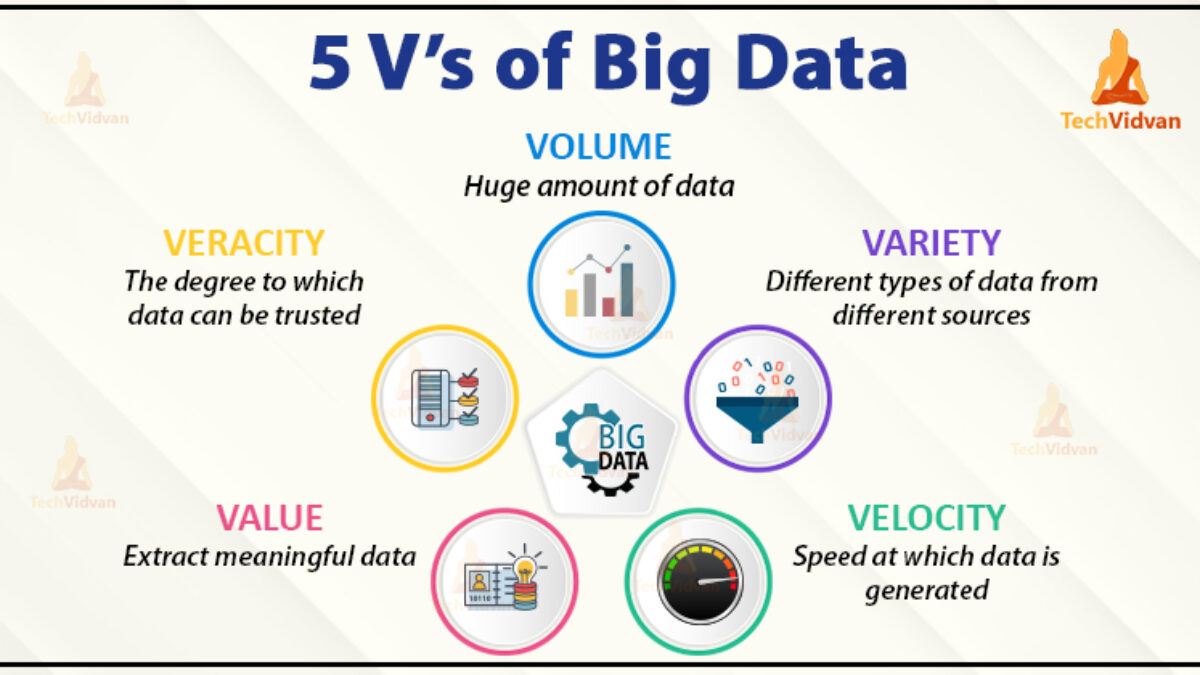
Tuy nhiên, do những bộ dữ liệu này quá lớn làm cho các phần mềm xử lý dữ liệu truyền thống không quản lý được, nhưng những khối dữ liệu khổng lồ này có thể giải quyết được rất nhiều vấn đề về kinh doanh hay sức khỏe mà trước đây không thể làm được giúp cho ngành Big Data ngày càng phát triển và đem đến nhiều điểm tích cực cho doanh nghiệp và xã hội.



Hình 2‑1. Big Data

Big Data có 5 đặc trưng cơ bản (5V’s) gồm:

* Volume: Là sự tăng trưởng về mặt khối lượng, Big Data không ngừng phát triển về mặt kích thước (dung lượng) khi mà nhu cầu thu thập dữ liệu ngày càng lớn dẫn đến khối lượng dữ liệu sẽ phát triển ngày càng khổng lồ theo thời gian, dưới định dạng hình ảnh, âm thanh, video … trên các kênh truyền thông, mạng xã hội. Dung lượng dữ liệu của 1 hệ thống thông tin rất lớn có thể lên đến hàng Terabyte (TB) và Petabyte (PB).
* Velocity: Là sự tăng trưởng về mặt tốc độ truyền dữ liệu bên cạnh sự tăng trưởng về mặt khối lượng. Dữ liệu được thu thập thông tin qua các bài post, thông báo chỉ trong vài giây trên các nền tảng mạng xã hội. Sự chuyển động của dữ liệu hiện tại gần như là thực tế (real-time) và tốc độ cập nhật thông tin rất nhỏ chỉ còn đơn vị hàng Mili giây.
* Variety: Là sự tăng trưởng về tính đa dạng của dữ liệu, đề cập đến sự không nhất quán và không chắc chắn. Bản chất của dữ liệu không còn đơn giản là dạng số hay chữ như truyền thống, mà đã phát triển thành cả dữ liệu cấu trúc, bán cấu trúc và phi cấu trúc như video, hình ảnh, âm thanh, từ các cảm biến, file log của các thiết bị thu. Dữ liệu của hệ thống thông tin hay doanh nghiệp không còn đơn giản mà tính đa dạng cũng ngày càng tăng lên theo thời gian làm cho tính phức tạp của dữ liệu ngày càng tăng.
* Veracity: Là tính xác thực của dữ liệu, cùng với sự tăng trưởng nhanh về mặt khối lượng, tốc độ và đa dạng của dữ liệu thì việc đảm bảo tính xác thực, tính chính xác của dữ liệu là rất quan trọng. Với sự phát triển của xu hướng thời đại 4.0, mạng xã hội và người dùng di động làm cho việc xác định độ tin cậy và xác thực của dữ liệu ngày một khó khăn hơn, vì thế bài toán phân tích và loại bỏ dữ liệu thiếu chính xác và nhiễu là rất quan trọng của Big Data.
* Value: Cuối cùng nhưng là quan trọng nhất, giá trị thông tin mà Big Data mang lại là tính chất quan trọng nhất của xu hướng công nghệ Big Data. Một hệ thống Big Data sẽ không được gọi là hiệu quả khi không mang lại nhiều giá trị cho doanh nghiệp, vì vậy doanh nghiệp phải hoạch định được những giá trị thông tin để giải quyết vấn đề, bài toán hoặc mô hình hoạt động kinh doanh của mình.



Hình 2‑2. 5V’s of Big Data

Ứng dụng của Big Data: Big Data mang lại nhiều giá trị cho rất nhiều ngành khác nhau, tiêu biểu như Ngân Hàng, Y Tế, Thương Mại Điện Tử, Digital Marketing, ngăn chặn nội dung xấu/đen…

* Ngành Ngân Hàng: Big Data là nền tảng của ngân hàng kỹ thuật số, áp dụng Machine Learning và AI để phát hiện các hoạt động gian lận, hỗ trợ xử lý, báo cáo và phân tích lượng dữ liệu khổng lồ từ các hoạt động giao dịch hằng ngày nhằm đảm bảo an ninh cho ngân hàng, sử dụng kỹ thuật phân cụm để tìm ra khu vực có nhu cầu giao dịch cao để thành lập chi nhánh mới …
* Ngành Y Tế: Theo dõi tình trạng bệnh nhân, đánh giá các triệu chứng và xác định nhiều bệnh thời gian đầu, lưu trữ hồ sơ bệnh lý, thông tin nhạy cảm được bảo mật, dự đoán trước về dấu hiệu phát dịch, bùng dịch …
* Ngành Thương Mại Điện Tử: Thu thập dữ liệu, thông tin về nhu cầu, thói quen mua hàng hoặc tìm kiếm của khách hàng, tạo mô hình tiếp thị, quản lý các sản phẩm được xem và mua nhiều nhất để đưa ra các chương trình khuyến mãi để tăng hiệu suất, lập báo cáo theo nhiều tiêu chí, phân tích hành vi mua hàng, xu hướng mua hàng để cung cấp dịch vụ nhanh và tốt nhất …
* Ngành Retail (Bán lẻ): Nhờ sự phát triển của Big Data giúp doanh nghiệp xây dựng mô hình chi tiêu của khách hàng, so sánh tỷ lệ cung – cầu đề sản xuất tránh việc tung ra sản phẩm không được sử dụng, bố trí sản phẩm trưng bày dựa vào thói quen và nhu cầu mua hàng …
* Ngành Digital Marketing: Phân tích thị trường, đối thủ cạnh tranh và đánh giá mục tiêu kinh doanh, tạo báo cáo sau chiến dịch quảng cáo để tăng hiệu suất, tập trung vào chủ đề tìm kiếm nâng cao và tư vấn cho chủ doanh nghiệp để thực hiện chiến lược xếp hạng trên Google.
* Ngăn chặn nội dung xấu/đen: Các Extension của các nền tảng trình duyệt như Google Chrome, Microsoft Edge, Mozilla Firefox … có nhiều Addin phục vụ cho việc lọc nội dung (content filtering) có phù hợp hay không, ngăn chặn các tổ chức xấu lợi dụng để trục lợi.

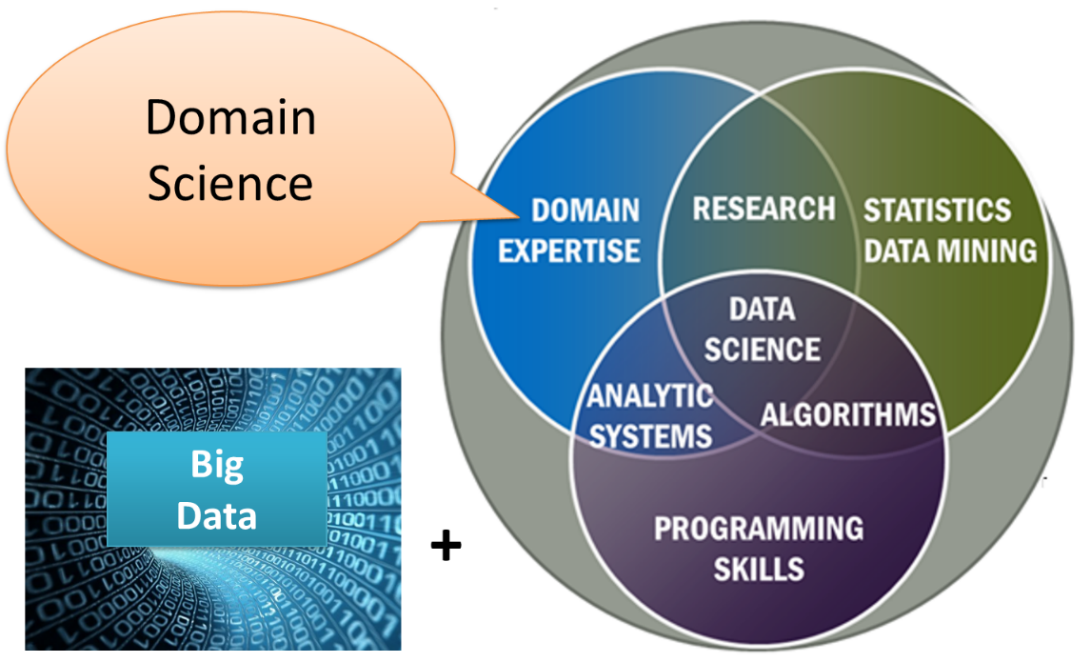
### Khoa Học Dữ Liệu (Data Science)

Khoa Học Dữ Liệu là ngành khoa học về việc quản trị, lưu trữ và phân tích dữ liệu, trích xuất các giá trị từ dữ liệu để tìm ra các insight, tri thức hành động và quyết định. Khoa học Dữ liệu gồm 3 phần chính: Tạo và quản trị, phân tích dữ liệu và chuyển kết quả phân tích thành giá trị hành động.

Việc phân tích lại dựa vào 3 nguồn tri thức: Toán học (Thống kê), Công nghệ Thông tin (Máy học) và tri thức của lĩnh vực ứng dụng. Các lĩnh vực của Khoa học dữ liệu: Học máy (Machine Learning), Khai thác dữ liệu (Data mining), Lập trình (Programming), Thống kê (Statistic), Phân tích (Analyzing).

Khoa học Dữ liệu yêu cầu thực hiện các quan sát, đặt câu hỏi, phân tích kết quả và đưa ra khuyến nghị. Mục đích chính là biến đổi một lượng lớn dữ liệu chưa qua xử lý, giúp các doanh nghiệp định vị được mô hình kinh doanh, giúp giảm chi phí, gia tăng hiệu quả làm việc, nắm bắt được cơ hội cơ hội, tránh rủi ro trên thị trường và làm gia tăng lợi thế cạnh tranh. Quy trình làm việc của ngành Khoa học Dữ liệu:

* Thu thập dữ liệu (Data Capture).
* Phân tích khám phá EDA (**E**xploratory **D**ata **A**nalysis) để tìm ra những insight.
* Phân tích sau cùng (Final Analysis) để đưa ra những quyết định.
* Quản trị dữ liệu (Data Manage) để giảm chi phí và bảo mật dữ liệu.
* Báo cáo (Report).



Hình 2‑3. Data Science và các lĩnh vực liên quan

## Giới thiệu về Data Mining và Machine Learning

### Data Mining

Data Mining nghĩa là Khai phá Dữ liệu, là quá trình phân loại, trích xuất, sắp xếp tập dữ liệu lớn để thiết lập các mối liên hệ, xác định mẫu để đưa ra hướng giải quyết vấn đề nhờ phân tích dữ liệu, sử dụng những dữ liệu có giá trị tiềm ẩn từ bên trong lượng lớn dữ liệu được lưu trữ trong Cơ sở dữ liệu.

Chu trình khai phá dữ liệu là quá trình phức tạp bao gồm kho dữ liệu lớn, chuyên sâu cùng với các công nghệ tính toán, Data Mining không chỉ giới hạn trong việc trích xuất dữ liệu mà còn được dùng để làm sạch dữ liệu, phân tích mẫu và chuyển đổi. Trong Data Mining có nhiều tham số quan trọng khác nhau như Phân tích chuỗi giá trị (Value chain analysis), Phân cụm (Clustering), Phân loại (Classification), Dự đoán (Prediction) và Quy tắc kết hợp.

Để thực hiện Data Mining bao gồm 7 bước:

* Bước 1: Làm sạch dữ liệu, làm sạch để dữ liệu không có dư thừa và bất thường.
* Bước 2: Tích hợp dữ liệu, kết hợp nhiều nguồn dữ liệu lại làm một để dễ quản lý và sử dụng.
* Bước 3: Lựa chọn dữ liệu, trích xuất dữ liệu từ Cơ sở dữ liệu đa chiều.
* Bước 4: Chuyển đổi dữ liệu để phân tích tóm tắt, tổng hợp và báo cáo.
* Bước 5: Khai phá dữ liệu, trích xuất dữ liệu từ dữ liệu hiện có.
* Bước 6: Đánh giá mẫu, phân tích mẫu trong dữ liệu.
* Bước 7: Trình bày thông tin, dữ liệu sẽ được thể hiện. trình bày dưới dạng cây, biểu đồ, bảng và ma trận …



Hình 2‑4. Các bước thực hiện Data Mining

Hiện nay có nhiều công cụ được tạo ra để phục vụ, hỗ trợ cho việc Data Mining trở nên nhanh chóng và chính xác hơn, có thể kể đến như RapidMiner, Weka, KNime, Apache Mahout … Data Mining có rất nhiều ứng dụng để phục vụ, đáp ứng nhu cầu cho nhiều lĩnh vực:

* Phân tích và quản lý rủi ro trong doanh nghiệp.
* Phát hiện gian lận trong ngân hàng để ngăn chặn rủi ro bảo mật.
* Phát hiện bệnh cho bệnh nhân ở nhiều giai đoạn giúp bác sĩ theo dõi và chữa trị kịp thời.
* Phân tích thị trường chứng khoán để đưa ra kế hoạch đầu tư.

### Machine Learning

Machine Learning là lĩnh vực con của Trí tuệ nhân tạo (AI - **A**rtificial **I**ntelligence) và là nền tảng của lĩnh vực Học sâu (Deep Learning), sử dụng thuật toán để máy tính có thể học từ dữ liệu để thực hiện công việc bằng việc lập trình, thay vì được lập trình để thực hiện công việc một cách rõ ràng.

Các bước thực hiện Machine Learning workflow gồm: Data Collection (thu thập dữ liệu, để đạt hiệu quả và đúng đắn phải thu thập từ nguồn dữ liệu chính thống), Preprocessing (tiền xử lý dữ liệu như chuẩn hóa, loại bỏ thuộc tính dư thừa, gán nhãn, trích xuất đặc trưng …), Training Model (huấn luyện mô hình), Evaluating Model (đánh giá mô hình, đánh giá độ chính xác để nhận định mô hình tốt hay chưa) và Improve (cải thiện mô hình để có được kết quả chính xác hơn, giảm thiểu sai số).

Machine Learning có rất nhiều ứng dụng đã mang lại rất nhiều lợi ích cho các doanh nghiệp, tổ chức, cho ngành Y Tế cùng rất nhiều ngành hay lĩnh vực khác:

* Xử lý ảnh (Image Processing): Gắn thẻ hình ảnh (Image Tagging), Nhận dạng ký tự (Optical Character Recognition), Xe tự hành (Self-driving vehicle) …
* Phân tích văn bản (Text Analysis): Khai thác thông tin (Information Extraction), Lọc spam (Spam filtering), Phân tích ngữ nghĩa (Sentiment Analysis) …
* Khai phá dữ liệu (Data Mining): Phát hiện bất thường (Anomaly Detection), Gom nhóm (Grouping), Dự đoán (Predictions), Phát hiện quy luật (Association rules)
* Trò chơi điện tử và Robot.

Có 2 loại thuật toán cơ bản trong Machine Learning: Học có giám sát (Supervised Learning) và Học không có giám sát (Unsupervised Learning)

* Học có giám sát: Thuật toán Học có giám sát được chia thành là Phân loại (Classification) và Hồi quy (Regression) với các thuật toán là: Support Vector Machines (SVMs), Linear Regression, Logistic Regression, k-Nearest Neighbors (kNN), Neural Network …
* Học không có giám sát: Thuật toán Học không có giám sát được chia thành là Phân cụm (Clustering), Biểu diễn và giảm số chiều (Representation and dimensional reduction) với các thuật toán là: K-mean, Spectral Clustering, Hierarchical clustering-Apriori (Association Rule Mining) …



Hình 2‑5. Các loại thuật toán được sử dụng trong Machine Learning

Việc lựa chọn thuật toán để áp dụng vào bài toán Machine Learning cũng rất quan trọng, giúp tiết kiệm thời gian, tải nguyên máy tính và đem lại kết quả chính xác hơn. Dữ liệu có dãn (label) thì sử dụng thuật toán Học có giám sát (Supervised), nếu không có thì sử dụng thuật toán Học không có giám sát (Unsupervised). Trong thuật toán học có giám sát, dữ liệu dạng phân loại (Category) hay dạng số lượng (Quantity) để lựa chọn thuật toán Phân loại (Classification) hay Hồi quy (Regression). Trong thuật toán học không giám sát, dữ liệu dạng phân cụm (Cluster) hay dạng Chiều (Dimensionality) để lựa chọn thuật toán Phân tích Phân cụm (Cluster Analysis) hay Giảm kích thước (Dimensionality reduction).



Hình 2‑6. Lựa chọn thuật toán phù hợp

## Giới thiệu Python và các thư viện của Python

### Python

Python là ngôn ngữ lập trình bậc cao được tạo ra bởi Guido van Rossum và được ra mắt lần đầu vào năm 1991, mã nguồn mở (Open-source), là ngôn ngữ thông dịch (Interpreter), dễ học dễ đọc và chạy đa nền tảng. Python có cấu trúc gọn gàng, hỗ trợ nhiều mẫu đa lập trình như: Lập trình hàm (Functional Programming), Lập trình hướng đối tượng OOP (**O**bject-**O**riented **P**rogramming) …

* Ưu điểm của Python:
* Là ngôn ngữ dễ học cho người mới và được nhiều lập trình viên sử dụng.
* Cú pháp đơn giản dễ làm quen.
* Thích hợp cho các chương trình nhỏ và dự án lớn nhờ khả năng mở rộng.
* Sử dụng được trên nhiều Hệ điều hành: Windows, UNIX, Mac OS, Linux.
* Nhược điểm của Python:
* Python không có một số thuộc tính như: Public, Private, Protected …
* Mặc dù tốc độ xử lý nhanh, tuy nhiên vẫn chậm hơn Java, C++ …

Cùng với Ưu và Nhược điểm, Python có nhiều tính năng nổi bật được nhiều lập trình viên sử dụng:

* Phát triển phần mềm (Software Development)
* Phát triển Web phía máy chủ (Backend Development) và API cùng với framework hỗ trợ như Flask, Django, FastAPI, Uvicorn, Falcon …
* Python được sử dụng để thu thập và xử lý dữ liệu lớn cùng với thư viện hỗ trợ như Requests, Scrapy, BeautifulSoup, Pandas, PySpark, Numpy …
* Python được sử dụng để khai phá dữ liệu và trực quan hóa cùng với thư viện hỗ trợ như Matplotlib, Bokeh, Seaborn, Plotly …
* Python được sử dụng cho lĩnh vực Trí tuệ nhân tạo, Machine Learning và Deep Learning cùng với nhiều thư viện hỗ trợ như Tensorflow, Scikit-learn, Keras, PyTorch, SciPy, Theano …
* Python được sử dụng trong lĩnh vực xử lý ảnh (Image Processing) cùng với thư viện như OpenCV, PIL, Pillow …
* Python được sử dụng trong lĩnh vực xử lý ngôn ngữ tự nhiên (**N**atural **L**anguage **P**rocessing – NLP) cùng với nhiều thư viện hỗ trợ như NLTK, Spacy, Gensim, Pattern, TextBlob, CoreNLP …
* Python được nhiều trường học sử dụng làm ngôn ngữ lập trình để dạy học vì tính dễ học, dễ đọc và đơn giản.

### Các thư viện của Python

Python là ngôn ngữ mã nguồn mở, có nhiều thư viện hỗ trợ nhằm giúp lập trình viên dễ dàng hơn trong việc lập trình. Trong đề tài này, tôi có sử dụng một số thư viện để hỗ trợ:

* **Pandas**: Là công cụ dùng để phân tích và thao tác dữ liệu mã nguồn mở nhanh, mạnh, dễ sử dụng và linh hoạt, cung cấp nhiều tính năng hữu ích cho người dùng, đặc biệt là trong ngành Khoa học Dữ liệu.
* **Numpy (Num**eric **Py**thon**)**: Là thư viện toán học phổ biến và mạnh mẽ của Python, cho phép người sử dụng làm việc, thao tác hiệu quả với ma trận 1 chiều, 2 chiều hay đa chiều (multidimensional) và mảng với tốc độ xử lý nhanh gấp nhiều lần so với sử dụng “core Python” đơn thuần, hoạt động nhanh trên mảng cùng với các phép toán học, logic, sắp xếp, các phép biến đổi Fourier, đại số tuyến tính, mô phỏng ngẫu nhiên …
* **Matplotlib**: Trong lĩnh vực Khoa học dữ liệu trực quan hóa (Visualization) là một bước quan trọng và Matplotlib là giải pháp hiệu quả cho người dùng Python. Matplotlib là thư viện vẽ đồ thị, biểu đồ, hình ảnh trực quan tĩnh và động rất mạnh mẽ cho người làm việc với Python và Numpy.
* **Seaborn**: Bên cạnh Matplotlib thì Python có Seaborn cũng là thư viện dùng để trực quan hóa dữ liệu, vẽ đồ thị và biểu đồ. Seaborn được xây dựng dựa trên thư viện Matplotlib và Pandas (tích hợp chặt chẽ với cấu trúc dữ liệu của Pandas), cung cấp cho người dùng Python nhiều chức năng để vẽ nhiều dạng biểu đồ, thống kê đẹp hơn so với Matplotlib. Các chức năng vẽ biểu đồ của Seaborn hoạt động trên Dataframe và mảng chứa tập dữ liệu.
* **Requests**: Là thư viện sử dụng HTTP (**H**yper **T**ext **T**ransfer **P**rotocol) đơn giản của Python, dùng để gửi lệnh HTTP qua các Server, dịch vụ Web API, cung cấp cho người dùng gửi các loại lệnh như GET (để lấy dữ liệu), POST (để thêm dữ liệu), PUT (để cập nhật dữ liệu), DELETE (để xóa dữ liệu) …
* **Scikit-learn**: Là thư viện mạnh mẽ dành cho các thuật toán Machine Learning được biết bằng ngôn ngữ Python, là một công cụ đơn giản và hiệu quả cho phân tích dự đoán, xây dựng trên thư viện Numpy, SciPy và Matplotlib, cung cấp các công cụ để xử lý các bài toán Machine Learning và mô hình thống kê (Statistical Modeling) như: Classification, Clustering, Regression, và Dimensionality reduction.

## Giới thiệu SQL và Cơ sở dữ liệu MySQL

### SQL

SQL: Viết tắt của **S**tructured **Q**uery **L**anguage, là Ngôn ngữ Truy vấn dữ liệu mang tính Cấu trúc theo tiêu chuẩn ANSI/ISO, dùng để giao tiếp với bất cứ Hệ thống Quản trị Cơ sở dữ liệu Quan hệ RDBMS (**R**elational **D**ata**b**ase **M**anagement **S**ystem) như Oracle, MySQL, PostgreSQL, Microsoft SQL Server … Cơ sở dữ liệu này được quy thành nhiều bảng, mỗi bảng gồm nhiều cột (gọi là trường) và hàng (gọi là bản ghi), các bảng này có mối liên kết chặt chẽ với nhau do người dùng định nghĩa nên được gọi là có cấu trúc.

SQL được sử dụng rất nhiều ở các công ty vừa và lớn để xây dựng hệ thống lưu trữ Cơ sở dữ liệu, giúp người dùng truy vấn thông tin nhanh, chính xác và quản lý hiệu quả, bảo trì, bảo mật thông tin dễ dàng hơn.

Không giống như các dạng lưu trữ dữ liệu truyền thống như tài liệu giấy hay Excel sẽ có nhiều rủi ro như bị mất, chỉnh sửa như xóa, sửa … và dữ liệu không được bảo mật cao, dễ bị rò rỉ thông tin bảo mật ra bên ngoài. Còn với SQL, sau khi lưu thông tin vào cơ sở dữ liệu, người dùng chỉ cần sử dụng câu truy vấn SQL để tương tác với các bảng là đã trích xuất được thông tin cần thiết, việc thêm, sửa, xóa cũng dễ dàng hơn rất nhiều.



Hình 2‑7. Hình ảnh định nghĩa SQL

SQL có nhiều Ưu điểm giúp cho người dùng dễ dàng thao tác với cơ sở dữ liệu:

* Thao tác thay đổi, cập nhật dữ liệu dễ dàng trong Cơ sở dữ liệu.
* Truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ phức tạp.
* Tạo thủ tục lưu trữ, chế độ view và chức năng trong Cơ sở dữ liệu.
* Nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL.
* Chia user, thiết lập quyền truy cập trên các bảng, thủ tục và view.

Các lệnh SQL cơ bản được chia thành 4 loại: DDL, DML, DCL và TCL

* DDL (**D**ata **D**efinition **L**anguage): Ngôn ngữ định nghĩa dữ liệu, ví dụ: CREATE, ALTER, TRUNCATE, DROP, RENAME ...
* DML (**D**ata **M**anipulation **L**anguage): Ngôn ngữ thao tác dữ liệu, ví dụ: SELECT, INSERT, UPDATE, DELETE, MERGE, LOCK TABLE ...
* DCL (**D**ata **C**ontrol **L**anguage): Ngôn ngữ điều khiển dữ liệu, ví dụ: GRANT, REVOKE.
* TCL (**T**ransaction **C**ontrol **L**anguage): Ngôn ngữ kiểm soát giao dịch, ví dụ: COMMIT, ROLLBACK. SAVEPOINT …

### Cơ sở dữ liệu MySQL

MySQL là Hệ quản trị Cơ sở dữ liệu mã nguồn miễn phí (free-source) phổ biến thế giới, được nhiều nhà phát triển ưa chuộng trong quá trình xây dựng và phát triển ứng dụng, sử dụng SQL, có tốc độ cao, ổn định, dễ dàng sử dụng và hoạt động được trên nhiều Hệ điều hành như Windows, Ubuntu, Linux, MacOS. Hệ quản trị Cơ sở dữ liệu này hoạt động dựa trên mô hình tiêu chuẩn Client (Máy khách) – Server (Máy chủ).

MySQL phù hợp cho việc phát triển các ứng dụng có sử dụng truy cập Cơ sở dữ liệu trên Internet nhờ tốc độ và tính bảo mật cao. Hiện nay, nhiều Website có lượng truy cập lớn như Facebook, Twitter, Google hay Amazon … đều đang sử dụng MySQL để hỗ trợ lưu trữ dữ liệu để đáp ứng cho những ứng dụng kinh doanh. Ưu và nhược điểm của MySQL:

* Ưu điểm:
* Dễ dàng sử dụng: Tốc độ xử lý cao và ổn định, hoạt động trên nhiều Hệ điều hành cung cấp nhiều hàm tiện ích.
* Tính bảo mật và an toàn cao: MySQL phù hợp với các ứng dụng có truy cập Internet nhờ sở hữu nhiều tính năng bảo mật cấp cao.
* Cho phép khôi phục: MySQL cho phép các giao dịch (Transaction) khôi phục, cam kết (Commit) và hồi phục sự cố.
* Khả năng mở rộng mạnh mẽ: MySQL có khả năng xử lý với lượng dữ liệu lớn và có thể mở rộng.
* Nhược điểm:
* MySQL bị hạn chế dung lượng, tốc độ truy vấn sẽ giảm dần nếu lượng dữ liệu truy vấn lớn, để cải thiện cần áp dụng nhiều biện pháp như chia sẻ database ra nhiều server hoặc tạo cache MySQL.

## Giới thiệu Jupyter Notebook, API và Kaggle

### Jupyter Notebook

Jupyter Notebook là một ứng dụng Web mã nguồn mở cho phép người dùng tạo và chia sẻ code, phương trình, hình ảnh hóa (visualization). Ứng dụng của Jupyter Notebook: làm sạch, chuyển đổi dữ liệu, lập mô hình thống kê, trực quan hóa dữ liệu mô phỏng … Jupyter Notebook cho phép người dùng thực thi code theo từng ô lệnh (cell) mà không cần phải chạy hoàn toàn 1 file như những ứng dụng viết code khác.

### API

API là viết tắt của **A**pplication **P**rogramming **I**nterface, nghĩa là Giao diện Lập trình Ứng dụng, là trung gian phần mềm cho phép 2 ứng dụng giao tiếp với nhau, có tính bảo mật và an toàn cao. API có thể sử dụng cho hệ điều hành (Operating System), hệ thống cơ sở dữ liệu (Database System), hệ thống trên nền tảng web (Web-based System), phần cứng máy tính … API là một tập hợp các quy trình, giao thức và công cụ để xây dựng ứng dụng phần mềm. Hiện nay có rất nhiều API được ra đời nhằm giúp cho các lập trình viên như Google Maps API, Youtube API, Twitter API, Amazon Product Advertising API …

* Ưu điểm của API:
* Giao tiếp 2 chiều phải được xác nhận.
* Kết nối mọi lúc nhờ Internet.
* Hỗ trợ chức năng RESTful.
* Mã nguồn mở, giao tiếp API là 2 chiều phải được xác nhận nên thông tin đáng tin cậy.
* Nhược điểm của API:
* Đòi hỏi kiến thức chuyên sâu.
* Tốn nhiều chi phí vận hành, nâng cấp, chỉnh sửa và phát triển.
* Gặp vấn đề về bảo mật nếu bị tấn công hệ thống.

API giúp cho việc phát triển ứng dụng trở nên dễ dàng, linh hoạt và tiết kiệm thời gian nhờ những đặc điểm sau: Sử dụng mã nguồn mở, hỗ trợ XML và JSON, có khả năng đáp ứng đầy đủ thành phần HTTP như URL, request/response headers …, mô hình WebAPI hỗ trợ MVC như: unit test, IoC container, routing, controller, action result, model binder …



Hình 2‑8. Định nghĩa về API

### Kaggle

Kaggle là một công ty con dưới sự điều hành của Google LLC – là một tập đoàn công nghệ đa quốc gia được thành lập tại Mỹ, là doanh nghiệp lớn chuyên về cung cấp các dịch vụ và sản phẩm liên quan đến mạng Internet như dịch vụ công nghệ thông tin, quảng bá trực tuyến, phần mềm, phần cứng cũng như nổi bất nhật là công cụ tìm kiếm thông tin.

Kaggle là một cộng đồng trực tuyến, là nơi những nhà Khoa học dữ liệu cùng với mọi đối tượng có thể sử dụng để thực hành về Machine Learning, Deep Learning và tương tác với nhau như một cộng đồng. Kaggle cung cấp rất nhiều tập dữ liệu (được gọi là Dataset) về mọi lĩnh vực như thể thao, y tế, tài chính … và một nền tảng (platform) trực tuyến để mọi người có thể chia sẻ kiến thức, cũng như code của bản thân. Kaggle còn cung cấp nơi để mọi người có thể luyện thêm kỹ năng bằng tham gia thi đấu các cuộc thi do Kaggle tổ chức.

## Giới thiệu về Docker và Apache Airflow

### Docker

Docker là phần mềm cho phép lập trình viên triển khai, dựng và kiểm thử ứng dụng một cách nhanh chóng và là nền tảng mã nguồn mở. Docker đóng gói phần mềm vào các đơn vị tiêu chuẩn hóa được gọi là Container chứa mọi thứ mà phần mềm cần để chạy bao gồm thư viện, mã, công cụ hệ thống và thời gian chạy. Docker giúp lập trình viên nhanh chóng triển khai và thay đổi quy mô ứng dụng vào bất kỳ môi trường nào và đảm bảo rằng mã của bạn sẽ chạy được bằng cách sử dụng Docker.

* Cách thức hoạt động của Docker: Docker hoạt động bằng cách cung cấp phương thức tiêu chuẩn để chạy mã, là hệ điều hành của Container (trên nền tảng ảo hóa). Docker được cài đặt trên từng máy chủ và cung cấp các lệnh đơn giản dễ dàng sử dụng để xây dựng, khởi động hoặc dừng Container.
* Lý do nên sử dụng Docker: Cho phép vận chuyển mã nhanh hơn, tiêu chuẩn hóa các hoạt động ứng dụng, tiết kiệm tiền bằng cách cải thiện khả năng vận dụng tài nguyên.
* Những thành phần chính của Docker: Docker Engine (thành phần chính của Docker, là công cụ để đóng gói ứng dụng), Docker Hub (chứa hàng ngàn public images có thể sử dụng dễ dàng từ cộng đồng), Images (là khuôn mẫu để tạo Container), Container (là instance của image, có thể thao tác dựa trên Docker API hoặc Docker CLI), Docker Client (giúp người dùng giao tiếp với Docker host), Docker Daemon (lắng nghe yêu cầu từ Docker Client để quản lý Container, Image, Network và Volumes bằng REST API), Dockerfiles (tập tin để chỉ dẫn để build một image)và Volumes (phần dữ liệu được tạo ra khi Container được khởi tạo).
* Khi nào nên sử dụng Docker: Khi build ứng dụng cần scale một cách linh hoạt, thiết lập máy local và server trên cùng một môi trường nhanh chóng, triển khai kiến trúc Microservices …



Hình 2‑9. Quy trình thực thi của hệ thống sử dụng Docker.

### Apache Airflow

Apache Airflow là phần mềm mã nguồn mở (open-source), được sử dụng để sắp xếp các luồng làm việc dưới dạng chu kỳ theo hướng DAG (**D**irected **A**cyclic **G**raph) của những tác vụ theo hướng lập trình. Một lịch trình Airflow được sử dụng để lên lịch những quy trình làm việc và đường ống dữ liệu (data pipeline). Giao diện Airflow cho phép người dùng quản lý, theo dõi những đường ống chạy, giám sát tiến trình và xử lý sự cố khi cần thiết.

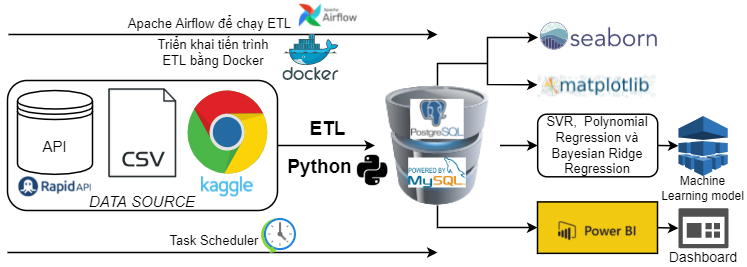


Hình 2‑10. Kiến trúc của Apache Airflow

# THU THẬP, PHÂN TÍCH VÀ BIỂU ĐỒ

## Thiết kế hệ thống

### Kiến trúc hệ thống

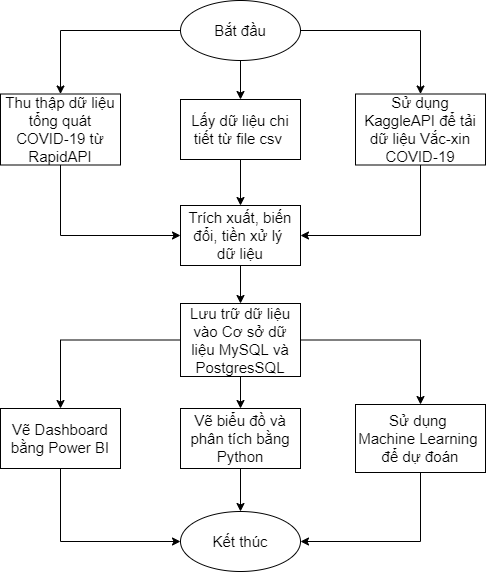


Hình 3‑1. Kiến trúc hệ thống

Hình 3-1 là kiến trúc hệ thống của Đồ án để giới thiệu những công nghệ, thuật toán Machine Learning, phần mềm được sử dụng trong Đồ án và quy trình công việc:

* Sử dụng Task Scheduler để lập lịch tự động và triển khai Apache Airflow bằng Docker để chạy tác vụ ETL.
* Thu thập dữ liệu từ các trang web được cung cấp bởi bên thứ 3 (third-party) uy tín, chính xác và được xác nhận.
* Thu thập dữ liệu thông qua API bằng phương thức HTTP.
* Thu thập dữ liệu từ nền tảng đa người dùng Kaggle.
* Sử dụng ngôn ngữ lập trình Python để trích xuất, tiền xử lý (preprocessing)
* Lưu trữ dữ liệu vào Cơ sở dữ liệu MySQL và PostgreSQL bằng Python.
* Sử dụng thư viện Matplotlib và Seaborn để phân tích và vẽ biểu đồ.
* Sử dụng thuật toán SVM, Polynomial Regression và Bayesian Ridge Regression để dự đoán.
* Sử dụng phần mềm Power BI để vẽ Dashboard báo cáo.

### Sơ đồ khối



Hình 3‑2. Sơ đồ khối

Hình 3-2 là sơ đồ khối mô tả chi tiết quy trình của Đồ án: Sử dụng Docker cài đặt và thiết lập Apache Airflow cùng với phần mềm Task Scheduler để tự động hóa quy trình ETL. Thu thập dữ liệu tổng quát về dịch COVID-19 từ trang web RapidAPI, thu thập dữ liệu từ nền tảng trực tuyến Kaggle về dữ liệu Vắc-xin và thu thập dữ liệu chi tiết về dịch COVID-19 từ Trường Đại học Johns Hopkins. Trích xuất, tiền xử lý, biến đổi dữ liệu bằng ngôn ngữ lập trình Python và lưu trữ dữ liệu bằng Cơ sở dữ liệu MySQL. Sử dụng phần mềm Power BI để vẽ các Dashboard, sử dụng thư viện Matplotlib và Seaborn để phân tích và vẽ biểu đồ và sử dụng Machine Learning để dự đoán số ca nhiễm trong tương lai gần của Thế giới và Việt Nam.

## Tổng quát về dịch COVID-19

Để có được cái nhìn tổng quát về dịch COVID-19, trong chương này tôi sử dụng Python để thu thập dữ liệu, lưu trữ dữ liệu vào Cơ sở dữ liệu MySQL và vẽ Dashboard bằng phần mềm Power BI.

### Thu thập dữ liệu từ RapidAPI và lưu trữ vào Cơ sở dữ liệu MySQL

Để thu thập được dữ liệu về dịch COVID-19 như tổng số ca nhiễm, tổng số ca tử vong, tổng số ca hồi phục, tỷ lệ tử vong, tỷ lệ mắc bệnh … trong đồ án này tôi sử dụng nền tảng cung cấp API là RapidAPI.

RapidAPI là được phát triển từ cuộc thi Hackathon năm 2015, được tạo ra bởi các nhà phát triển dành cho các nhà phát triển để họ có một nơi để truy cập vào các API và Microservices cũng như xây dựng các ứng dụng, phần mềm hiệu quả và dễ dàng hơn. RapidAPI là nơi các nhà phát triển có thể tìm, thử nghiệm và kết nối với các API – tất cả chỉ cần 1 tài khoản, 1 API key và 1 SDK (**S**oftware **D**evelopment **K**it). Các nhà phát triển có thể chia sẻ và cộng tác làm việc, RapidAPI cho phép các nhà phát triển chia sẻ và đăng ký API công khai. RapidAPI cung cấp rất nhiều API như dữ liệu về Thể thao, dữ liệu về Y tế, dữ liệu về Khoa học …

Trong Đồ án này, tôi sử dụng “VACCOVID - coronavirus, Vắc-xin and treatment tracker” của RapidAPI, là 1 API miễn phí và đã được xác thực thông tin bởi RapidAPI để phân tích và vẽ Dashboard.

Để trích xuất được dữ liệu từ API, tôi sử dụng ngôn ngữ lập trình Python cùng với thư viện Requests. Để sử dụng thư viện Requests, tôi cần cung cấp 2 thông số (parameter) là URL (**U**niform **R**esource **L**ocator), phương thức (method) “GET” và đưa phản hồi (response) trả về thành định dạng lưu trữ JSON (**J**ava**S**cript **O**bject **N**otation).



Hình 3‑3. Định dạng JSON của Response

Sau khi có được đầy đủ dữ liệu về COVID-19 của các quốc gia trên thế giới, tôi sử dụng Python để giao tiếp với Cơ sở dữ liệu MySQL để lưu trữ. Tôi sử dụng 2 từ khóa “TRUNCATE” (xóa tất cả dữ liệu của 1 bảng) và “INSERT” (thêm dữ liệu vào bảng) để cập nhật dữ liệu mới cho bảng “SUMMARY\_COVID\_DATA” của Cơ sở dữ liệu.



Hình 3‑4. Dữ liệu trong MySQL

Hình 3-4 là dữ liệu thống kê về dịch COVID-19 được cập nhật vào MySQL, đây là dữ liệu được tính đến ngày 24/12/2021 cùng với những cột như:

* NAME: tên quốc gia.
* CONTINENT: tên châu lục.
* TOTAL\_CASES: tổng số ca nhiễm.
* TOTAL\_DEATHS: tổng số ca tử vong.
* TOTAL\_RECOVERED: tổng số ca hồi phục.
* FATALITY\_RATE: tỷ lệ tử vong.
* INFECTION\_RISK: nguy cơ lây nhiễm.
* SERIOUS\_CRITICAL: số ca nguy kịch.
* TOTAL\_TESTS: tổng số xét nghiệm.
* TEST\_PERCENTAGE: tỷ lệ xét nghiệm
* RECOVERY\_PROPORATION: tỷ lệ hồi phục.
* POPULATION: dân số.

### Thiết kế Dashboard bằng phần mềm Power BI

Power BI là phần mềm thuộc nhóm Kinh doanh Thông minh (BI **- B**usiness **I**ntelligence) của Microsoft, Power BI được phát triển với mục đích sử dụng dữ liệu để đưa ra các báo cáo và hiển thị thông tin chi tiết của doanh nghiệp một cách trực quan. Power BI có ở cả 3 nền tảng Website, Desktop và Thiết bị di động.



Hình 3‑5. Tính năng của Power BI

Power BI có khả năng kết nối được với nhiều loại dữ liệu khác nhau như Excel, Cơ sở dữ liệu (MySQL, Oracle, PostgreSQL, SQL Server …), nền tảng Azure, dịch vụ Online (Google Analytics, Salesforce) … Những tính năng của Power BI: Trí tuệ nhân tạo, mô hình hóa, tự động làm sạch dữ liệu, kết nối với API, hỗ trợ triển khai Hybrid …

* Ưu điểm của Power BI:
* Tích hợp Python và R để nâng cao chất lượng trực quan hóa.
* Tính năng Power Query giúp dễ dàng sử dụng và làm sạch dữ liệu.
* Được xây dựng trên nền tảng Trí tuệ nhân tạo và Machine Learning.
* Cài đặt lịch trình tự động cập nhật.

Đầu tiên, để có cái nhìn tổng quát về tình hình dịch COVID-19 trên thế giới, tôi có thiết kế Dashboard (Hình 3-6) về tổng số ca nhiễm, tổng số ca tử vong, tổng số ca hồi phục… ở các châu lục và quốc gia.



Hình 3‑6. Dashboard về số ca nhiễm, ca tử vong...

Tính đến ngày 26/12/2021 trên toàn thế giới, tổng số ca nhiễm là 279 triệu ca, tổng số ca tử vong là 5 triệu ca, tổng số ca hồi phục là 250 triệu ca. Tôi sử dụng biểu đồ hình bánh (pie chart) (Hình 3-6[1], Hình 3-6[2], Hình 3-6[3]) để thể hiện tỷ lệ về tổng số ca nhiễm, ca tử vong và ca hồi phục của từng khu vực châu lục: châu Á, châu Âu, châu Mỹ (Nam Mỹ, Bắc Mỹ), Phi và châu Đại Dương:

Về tỷ lệ số ca nhiễm, qua biểu đồ tôi thấy được châu Á và châu Âu là 2 châu lục có số ca nhiễm lớn nhất lần lượt là 84.1 và 83.3 triệu ca chiếm 30.05% và 29.75% so với toàn thế giới.

Về tỷ lệ số ca tử vong, qua biểu đồ tôi thấy được châu Á và châu Âu là 2 châu lục có số ca tử vong lớn nhất lần lượt là 1.5 và 1.25 triệu ca chiếm 27.85% và 23.06% so với toàn thế giới.

Về tỷ lệ số ca hồi phục, qua biểu đồ tôi thấy được châu Á và châu Âu là 2 châu lục có số ca hồi phục lớn nhất lần lượt là 81.4 và 72.6 triệu ca chiếm 32.51% và 29.01% so với toàn thế giới.

Tôi sử dụng biểu đồ cột (bar chart) (Hình 3-6[4], Hình 3-6[5], Hình 3-6[6]) cùng với đường trung bình (average line) để so sánh tổng số ca nhiễm, tử vong và hồi phục của 15 quốc gia có số ca nhiều nhất:

Ở biểu đồ tổng số ca nhiễm: Mỹ, Ấn Độ và Brazil là 3 quốc gia có số ca nhiễm cao nhất lần lượt là 53, 34.8 và 22.2 triệu ca cùng với đường trung bình là 12.88.

Ở biểu đồ tổng số ca tử vong: Mỹ, Brazil và Ấn Độ là 3 quốc gia có số ca tử vong cao nhất lần lượt là 0.83, 0.62 và 0.48 triệu ca cùng với đường trung bình là 0.258.

Ở biểu đồ tổng số ca hồi phục: Mỹ, Ấn Độ và Brazil là 3 quốc gia có số ca hồi phục cao nhất lần lượt là 41, 34.2 và 21.4 triệu ca cùng với đường trung bình là 11.44.

Dashboard dưới đây (Hình 3-7) thể hiện về số ca dương tính, số ca nguy kịch của ngày 24/12/2021 và tỷ lệ tử vong, tỷ lệ mắc bệnh tính đến ngày 24/12/2021.



Hình 3‑7. Dashboard về tỷ lệ tử vong, tỷ lệ mắc bệnh...

Hình 3-7[1] thể hiện mối tương quan giữa tỷ lệ tử vong, tỷ lệ lây nhiễm và dân số của 10 quốc gia có tỷ lệ lây nhiễm cao nhất. Bar chart và line chart để thể hiện tỷ lệ lây nhiễm (INFECTION\_RISK), tỷ lệ tử vong (FATALITY\_RATE) và dân số (POPULATION)

Andorra, Montenegro và Seychelles là 3 quốc gia có tỷ lệ lây nhiễm cao nhất lần lượt là 28.06%, 25.92% và 24.38%, có tỷ lệ tử vong lần lượt là 0.64%, 1.47% và 0.54%. Tỉ lệ mắc bệnh cao do dân số của 3 quốc gia này rất nhỏ, với tỷ lệ giữa số ca mắc và dân số là: Andorra (21571/77447), Montenegro (161944/ 628183) và Seychelles (24047/ 99245).

Hình 3-7[2] thể hiện tổng số ca nhiễm, tử vong và hồi phục của 10 quốc gia có tổng số ca nhiễm cao nhất:

Biểu đồ cột xếp chồng (stacked column chart) để thể hiện sự tương quan giữa tổng số ca nhiễm, tử vong và hồi phục của từng quốc gia.

Mỹ, Ấn Độ và Brazil là 3 quốc gia có tổng số ca nhiễm + tổng số ca tử vong + tổng số ca hồi phục cao nhất trên thế giới, với thống kê lần lượt là 95, 69 và 44 triệu ca.

Hình 3-7[3] thể hiện mối tương quan giữa dân số, số ca nhiễm và tỷ lệ tử vong của 10 quốc gia có số ca nhiễm cao nhất:

Nga là quốc gia có tỷ lệ tử vong cao nhất với tỷ lệ tử vong 2.92%, tiếp đến là Brazil và Iran có tỷ lệ tử vong là 2.78% và 2.12%.

Hình 3-7[4] thể hiện mối tương quan giữa dân số, tổng số xét nghiệm và tỷ lệ xét nghiệm của 10 quốc gia có tổng số xét nghiệm nhiều nhất:

“Stacked column và line chart” để thể hiện dân số (POPULATION), tổng số xét nghiệm (TOTAL\_TESTS) và tỷ lệ xét nghiệm (TEST\_PERCENTAGE).

Trung Quốc (1.44 tỷ), Ấn Độ (1.4 tỷ) và Mỹ (333.8 triệu) là 3 quốc gia có dân số nhiều nhất thế giới có tỷ lệ xét nghiệm lần lượt là 11.12%, 48% và 239%.

Dashboard dưới đây (Hình 3-8) thể hiện tỷ trọng về tổng số ca nhiễm của các quốc gia tại châu Âu, châu Á, châu Mỹ (Nam Mỹ và Bắc Mỹ), châu Phi và châu Đại Dương.



Hình 3‑8. Dashboard về tổng ca nhiễm của các châu lục

Ở châu Âu (Hình 3-8[1]): Vương quốc Anh, Nga và Pháp có số ca nhiễm cao nhất là 11.9, 10.4 và 9.1 triệu ca chiếm tỷ trọng 16.85%, 14.7% và 12.88%.

Ở châu Á (Hình 3-8[2]): Ấn Độ, Thổ Nhĩ Kỳ và Iran có số ca nhiễm cao nhất là 34.8, 9.3 và 6.2 triệu ca chiếm tỷ trọng 46.98%, 12.54% và 8.35%.

Ở Bắc Mỹ (Hình 3-8[3]): Mỹ, Mexico và Canada có số ca nhiễm cao nhất là 53, 3.95 và 1.95 triệu ca chiếm tỷ trọng 84.42%, 6.29% và 3.12%.

Ở Nam Mỹ (Hình 3-8[4]): Brazil, Argentina và Colombia có số ca nhiễm cao nhất là 22.2, 5.45 và 5.12 triệu ca chiếm tỷ trọng 56.35%, 13.82% và 12.98%.

Ở châu Phi (Hình 3-8[5]): Nam Phi, Morocco và Tunisia có số ca nhiễm cao nhất là 3.4, 0.95 và 0.72 triệu ca chiếm tỷ trọng 42.2%, 11.83% và 8.95%.

Ở châu Đại Dương (Hình 3-8[6]): Úc, Fiji và Polynesia thuộc Pháp có số ca nhiễm cao nhất là 301.2, 52.8 và 46.4 nghìn ca chiếm tỷ trọng 65%, 11.4% và 10.01%.

Dashboard dưới đây (Hình 3-9) thể hiện tỷ trọng về tổng số ca tử vong của các quốc gia tại châu Âu, châu Á, châu Mỹ (Nam Mỹ và Bắc Mỹ), châu Phi và châu Đại Dương.



Hình 3‑9. Dashboard về tổng ca tử vong của các châu lục

Ở châu Âu (Hình 3-9[1]): Nga, Vương quốc Anh và Ý có số ca tử vong cao nhất là 33.2, 147.8 và 136.5 nghìn ca chiếm tỷ trọng 23.53%, 11.47% và 10.59%.

Ở châu Á (Hình 3-9[2]): Ấn Độ, Indonesia và Iran có số ca tử vong cao nhất là 479.7, 144.05 và 131.3 nghìn ca chiếm tỷ trọng 43.42%, 13.04% và 11.89%.

Ở Bắc Mỹ (Hình 3-9[3]): Mỹ, Mexico và Canada có số ca tử vong cao nhất là 837.8, 298.7 và 30.1 nghìn ca chiếm tỷ trọng 68.02%, 24.26% và 2.45%.

Ở Nam Mỹ (Hình 3-9[4]): Brazil, Peru và Colombia có số ca tử vong cao nhất là 618.4, 202.5 và 129.7 nghìn ca chiếm tỷ trọng 51.95%, 17.01% và 10.9%.

Ở châu Phi (Hình 3-9[5]): Nam Phi, Morocco và Ai Cập có số ca tử vong cao nhất lần lượt là 90.7, 25.5 và 21.5 nghìn ca chiếm tỷ trọng 45.92%, 12.91% và 10.91%.

Ở châu Đại Dương (Hình 3-9[6]): Úc, Fiji và Polynesia thuộc Pháp có số ca tử vong cao nhất là 2.2, 0.7 và 0.4 nghìn ca chiếm tỷ trọng 49.21%, 15.66% và 14.29%.

Dashboard dưới đây (Hình 3-10) thể hiện tỷ trọng về tổng số ca hồi phục của các quốc gia tại châu Âu, châu Á, châu Mỹ (Nam Mỹ và Bắc Mỹ), châu Phi và châu Đại Dương.



Hình 3‑10. Dashboard về tổng ca hồi phục của các châu lục

Ở châu Âu (Hình 3-10[1]): Vương quốc Anh, Nga và Pháp có số ca hồi phục cao nhất là 9.96, 9.22 và 7.79 triệu ca chiếm tỷ trọng 16.13%, 14.94% và 12.62%.

Ở châu Á (Hình 3-10[2]): Ấn Độ, Thổ Nhĩ Kỳ và Iran có số ca hồi phục cao nhất là 34.2, 8.92 và 6.02 triệu ca chiếm tỷ trọng 47.55%, 12.4% và 8.37%.

Ở Bắc Mỹ (Hình 3-10[3]): Mỹ, Mexico và Canada có số ca hồi phục cao nhất là 41, 3.3 và 1.81 triệu ca chiếm tỷ trọng 82.8%, 6.66% và 1.23%.

Ở Nam Mỹ (Hình 3-10[4]): Brazil, Argentina và Colombia có số ca hồi phục cao nhất là 21.4, 5.26 và 4.96 triệu ca chiếm tỷ trọng 56.72%, 13.93% và 13.13%.

Ở châu Phi (Hình 3-10[5]): Nam Phi, Ma-rốc và Tunisia có số ca hồi phục cao nhất là 3.1, 0.93 và 0.69 triệu ca chiếm tỷ trọng 42.14%, 12.73% và 9.44%.

Ở châu Đại Dương (Hình 3-10[6]): Úc, Fiji và Polynesia thuộc Pháp có số ca hồi phục cao nhất lần lượt là 229.3, 51.2 và 45.7 nghìn ca chiếm tỷ trọng 59.31%, 13.25% và 11.83%.

## Chi tiết về dịch COVID-19

### Giới thiệu

Ở phần 3.3, là cái nhìn chi tiết về tình hình dịch COVID-19 của Thế giới khi mà virus SARS-CoV-2 đã hoành hành khắp cả 5 châu lục và khiến hàng trăm triệu người lây nhiễm, hàng triệu người tử vong ảnh hưởng tiêu cực đến Y tế, Kinh tế …

Nguồn dữ liệu sử dụng đến từ Kho dữ liệu của Trung tâm Hệ thống Khoa học và Kỹ thuật (CSSE) tại trường Đại học Johns Hopkins.

### Biểu đồ và phân tích với Python



Hình 3‑11. Tương quan giữa các khu vực trên Thế giới

Hình 3-11 sử dụng bảng dạng gradient để thể hiện mối tương quan giữa các khu vực về tổng số ca nhiễm, ca tử vong, ca hồi phục, ca dương tính … Màu càng đậm cho thấy số lượng càng lớn và ngược lại. Tính đến ngày 26/12/2021: Châu Âu và Châu Á có tổng số ca nhiễm, tử vong và hồi phục cao nhất. Nam Mỹ là nơi có tỷ lệ tử vong cao nhất với 3.02%, tiếp đến là Châu Phi (2.39%) và Bắc Mỹ (1.96%).



Hình 3‑12. Số ca nhiễm, tử vong và đường Moving Average của Thế giới

Hình 3-12 sử dụng biểu đồ đường (line chart) để thể hiện số lượng ca nhiễm và ca tử vong cộng dồn (cumulative) theo thời gian cùng với đường Moving Average 7 ngày (MA7). Đường MA7 đi sát với đường số ca nhiễm và tử vong cho thấy tăng đều, nhanh và ít biến động.



Hình 3‑13. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Thế giới

Hình 3-13 sử dụng biểu đồ thanh (bar chart) để thể hiện số ca nhiễm và tử vong theo từng ngày của Thế giới cùng với đường MA7. Số ca nhiễm cao nhất trong 1 ngày hơn 1.4 triệu ca và số ca tử vong cao nhất trong 1 ngày là hơn 20 nghìn ca.



Hình 3‑14. Tỉ lệ tử vong theo ngày và trung bình của Thế giới

Hình 3-14 sử dụng biểu đồ đường để thể hiện tỷ lệ tử vong theo ngày và trung bình của Thế giới. Tỉ lệ tử vong trung bình là hơn 0.03, thấp hơn nhiều so với những dịch bệnh khác. Tỉ lệ tử vong tăng nhanh từ tháng 2/2020 đến 6/2020 và giảm mạnh đến gần cuối năm 2020, cùng với việc các quốc gia trên Thế giới triển khai việc tiêm Vắc-xin giúp cho tỷ lệ tử vong duy trì mức 0.02 và giảm dần theo thời gian.

Bên cạnh thông tin về các Quốc gia thì thông tin về các Tỉnh/Bang của các Quốc gia này cũng khai thác để có được cái nhìn chi tiết hơn.



Hình 3‑15. Top 10 các Tỉnh có số ca cao nhất trên Thế giới

Hình 3-15 sử dụng bảng gradient để thể hiện sự tương quan về tỷ lệ lây nhiễm, tỷ lệ tử vong, số ca nhiễm và tử vong giữa các Tỉnh/Bang của các Quốc gia. Anh, Maharashtra và California là 3 tỉnh có số cao nhiễm cao nhất (10.1, 6.65 và 5.29 triệu ca).



Hình 3‑16. Top 10 Bang của Mỹ có số ca cao nhất

Hình 3-16 sử dụng bảng gradient để thể hiện sự tương quan về tỷ lệ lây nhiễm, tỷ lệ tử vong, số ca nhiễm và tử vong giữa các Bang của Mỹ. California, Texas và Florida là 3 bang có số ca nhiễm và tử vong cao nhất tại Mỹ (5.3, 4,48 và 3.83 triệu ca). Texas có tỷ lệ lây nhiễm cao nhất hơn 4.1 triệu người cùng tỷ lệ tử vong là 0.016%, New Jersey có tỷ lệ lây nhiễm thấp nhất 328.2 nghìn người cùng tỷ lệ tử vong cao là 0.02%.



Hình 3‑17. Top 10 nước và phần còn lại về số ca nhiễm

Hình 3-17 sử dụng biểu đồ cột (bar chart) để thể hiện số ca nhiễm của top 10 nước có số ca nhiễm cao nhất và các nước còn lại sẽ là “Others”.



Hình 3‑18. Top 10 tỉnh và phần còn lại về số ca nhiễm

Hình 3-18 sử dụng biểu đồ cột (bar chart) để thể hiện số ca nhiễm của top 10 tỉnh có số ca nhiễm cao nhất và các nước còn lại sẽ là “Others”.

#### So sánh Việt Nam và Châu Á



Hình 3‑19. Số ca nhiễm, tử vong và đường Moving Average của Châu Á

Hình 3-19 sử dụng biểu đồ đường (line chart) để thể hiện số lượng ca nhiễm và ca tử vong cộng dồn (cumulative) theo thời gian cùng với đường Moving Average 7 ngày (MA7). Số ca nhiễm và tử vong tăng nhanh vào tháng 5/2021 và vẫn chưa có dấu hiệu dừng lại. Đường MA7 đi sát với đường số ca nhiễm và tử vong cho thấy tăng đều, nhanh và ít biến động.



Hình 3‑20. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Châu Á

Hình 3-20 sử dụng biểu đồ đường và cột (bar and line chart) để thể hiện số ca nhiễm và tử vong theo từng ngày của Châu Á cùng với đường MA7. Đầu tháng 3/2021 là thời điểm dịch COVID-19 bùng phát mạnh ở Ấn Độ và khu vực Đông Nam Á dẫn đến số ca nhiễm và tử vong tăng mạnh. Số ca nhiễm cao nhất trong 1 ngày gần 900 nghìn ca và số ca tử vong cao nhất trong 1 ngày là hơn 8 nghìn ca.



Hình 3‑21. Số ca nhiễm và tử vong giữa Việt Nam và các nước Châu Á

Hình 3-21 sử dụng biểu đồ đường thể hiện tốc độ gia tăng số ca nhiễm và tử vong cộng dồn theo ngày giữa Việt Nam và các quốc gia khác ở châu Á. Ấn Độ là quốc gia có dịch bùng phát mạnh vào tháng 4/2021 dẫn đến số ca nhiễm và tử vong tăng nhanh. So sánh giữa Việt Nam và các nước châu Á, Việt Nam có tốc độ gia tăng ít hơn về cả số ca nhiễm và tử vong, đây là 1 biểu đồ tốt dành cho Việt Nam.



Hình 3‑22. Biểu đồ cột giữa Việt Nam và các nước châu Á

Hình 3-22 sử dụng biểu đồ cột để so sánh về số ca nhiễm, tử vong và hồi phục giữa Việt Nam và các nước Châu Á. Ấn Độ là quốc gia có số ca nhiễm, tử vong và hồi phục cao nhất ở Châu Á.

#### So sánh Việt Nam và Đông Nam Á (South East Asia)



Hình 3‑23. Số ca nhiễm, tử vong và Moving Average của Đông Nam Á

Hình 3-23 sử dụng biểu đồ đường (line chart) để thể hiện số lượng ca nhiễm và ca tử vong cộng dồn (cumulative) theo thời gian cùng với đường Moving Average 7 ngày (MA7). Số ca nhiễm và tử vong tăng nhanh vào tháng 6/2021 và vẫn chưa có dấu hiệu dừng lại. Đường MA7 đi sát với đường số ca nhiễm và tử vong cho thấy tăng đều, nhanh và ít biến động.



Hình 3‑24. Số ca nhiễm, tử vong mỗi ngày và đường MA7 của Đông Nam Á

Hình 3-24 sử dụng biểu đồ đường và cột (bar and line chart) để thể hiện số ca nhiễm và tử vong theo từng ngày của Đông Nam Á cùng với đường MA7. Số ca nhiễm cao nhất trong 1 ngày gần 900 nghìn ca và số ca tử vong cao nhất trong 1 ngày là hơn 8 nghìn ca. Số ca nhiễm và số ca tử vong luôn ở mức rất cao (hơn 500 ca/ngày) từ tháng 5/2021 đến tháng 9/2021.



Hình 3‑25. Số ca nhiễm và tử vong giữa Việt Nam và các nước Đông Nam Á

Hình 3-25 sử dụng biểu đồ đường thể hiện tốc độ gia tăng số ca nhiễm và tử vong cộng dồn theo ngày giữa Việt Nam và các quốc gia Đông Nam Á. Đầu tháng 6/2021, dịch bùng phát mạnh ở khu vực Đông Nam Á, nhiều quốc gia bị ảnh hưởng như Indonesia, Philippines, Malaysia, Thailand và Việt Nam. So sánh giữa Việt Nam và các nước Đông Nam Á, Việt Nam có tốc độ gia tăng về cả số ca nhiễm và tử vong chỉ sau Indonesia, Philippines, Malaysia và Thailand.



Hình 3‑26. Biểu đồ cột giữa Việt Nam và các nước Đông Nam Á

Hình 3-26 sử dụng biểu đồ cột để so sánh về số ca nhiễm, tử vong và hồi phục giữa Việt Nam và các nước Đông Nam Á. Indonesia là quốc gia có số ca nhiễm, tử vong và hồi phục cao nhất ở Đông Nam Á. Việt Nam đứng thứ 5 trong khu vực.

## Tình hình tiêm ngừa Vắc-xin COVID-19

### Tình hình tiêm ngừa Vắc-xin COVID-19 của Thế giới

Các bảng và biểu đồ được thu thập và cập nhật đến ngày 27/12/2021.



Hình 3‑27. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin

Hình 3-27 là bảng sơ lược về tiến độ tiêm chủng Vắc-xin COVID-19 trên toàn Thế giới, đã có 223/232 quốc gia tham gia tiêm ngừa, 18 loại Vắc-xin được sử dụng, hơn 9105 triệu liều Vắc-xin đã được tiêm, hơn 3274 triệu người đã được tiêm và hơn 2591 triệu người đã tiêm đủ 2 mũi (đối với loại Vắc-xin yêu cầu 2 mũi). Đã có 223 tham gia tiêm ngừa Vắc-xin trên toàn Thế giới cho thấy được tốc độ triển khai tiêm ngừa Vắc-xin là rất nhanh.



Hình 3‑28. Biểu đồ số lượng Quốc gia sử dụng Vắc-xin đơn

Hình 3-28 sử dụng biểu đồ cột ngang (horizontal bar chart) để thể hiện số lượng quốc gia sử dụng Vắc-xin đơn. Dẫn đầu là Oxford/AstraZeneca (của Anh) được 23 quốc gia sử dụng, Pfizer/BioTech (của Mỹ và Đức) được 7 quốc gia sử dụng và Sinopharm/Beijing (của Trung Quốc) được 3 quốc gia sử dụng.



Hình 3‑29. Biểu đồ top 10 Quốc gia có tổng số Vắc-xin đã tiêm



Hình 3‑30. Biểu đồ top 10 Quốc gia có số Vắc-xin tiêm mỗi ngày



Hình 3‑31. Biểu đồ top 10 Quốc gia có số người đã tiêm đầy đủ

Hình 3-29 sử dụng biểu đồ cột để thể hiện tổng số lượng Vắc-xin đã được tiêm của top 10 quốc gia. Trung Quốc là quốc gia dẫn đầu với hơn 2.7 tỷ liều, tiếp đến là Ấn Độ và Mỹ với 1.5 tỷ và 500 triệu liều. Hình 3-30 sử dụng biểu đồ cột để thể hiện số lượng Vắc-xin tiêm mỗi ngày cao nhất của top 10 quốc gia. Trung Quốc là quốc gia dẫn đầu với hơn 22 triệu liều, tiếp đến là Ấn Độ và Bangladesh với 10 và 4 triệu liều. Hình 3-31 sử dụng biểu đồ cột để thể hiện tổng số người đã được tiêm đầy đủ của top 10 quốc gia. Trung Quốc là quốc gia dẫn đầu với hơn 1.2 tỷ người, tiếp đến là Ấn Độ và Mỹ với 577 và 200 triệu người.



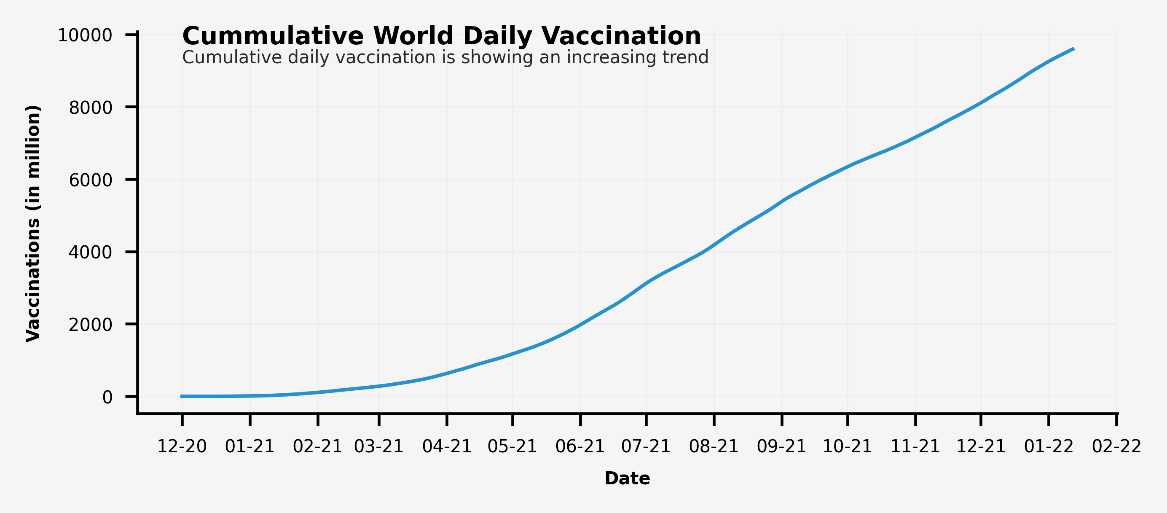
Hình 3‑32. Biểu đồ số lượng Quốc gia sử dụng Vắc-xin kết hợp

Hình 3-32 sử dụng biểu đồ cột ngang (horizontal bar chart) để thể hiện số lượng quốc gia sử dụng Vắc-xin đơn. Dẫn đầu là kết hợp giữa Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioTech được 24 quốc gia sử dụng, kết hợp giữa Moderna, Oxford/AstraZeneca, Pfizer/BioTech được 15 quốc gia sử dụng và kết hợp giữa Oxford/AstraZeneca, Pfizer/BioTech được 11 quốc gia sử dụng.



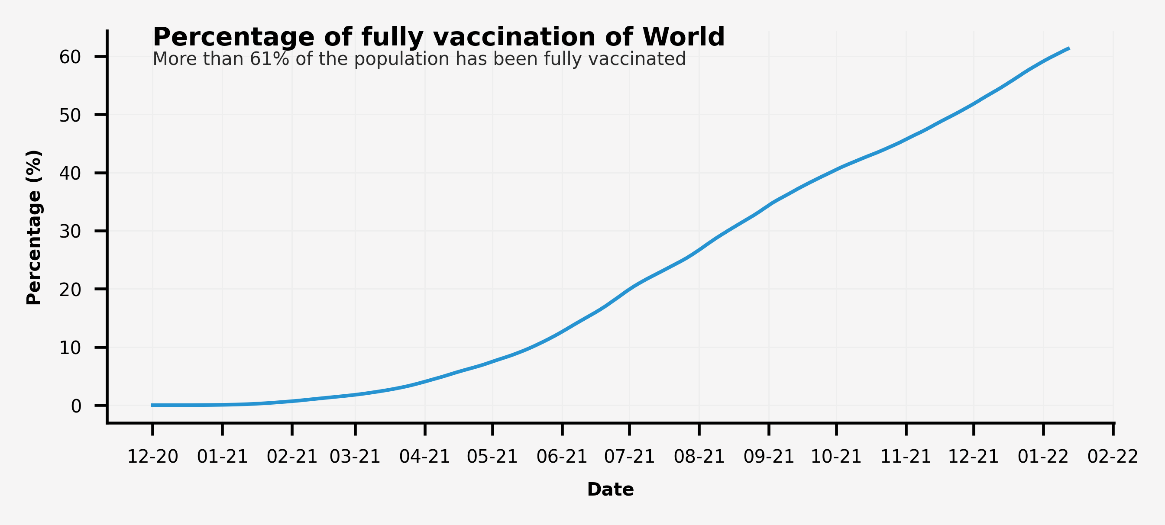
Hình 3‑33. Biểu đồ số lượng Vắc-xin mỗi ngày của Thế giới

Hình 3-33 sử dụng biểu đồ đường để thể hiện số lượng Vắc-xin được tiêm mỗi ngày của Thế giới. Số lượng Vắc-xin tăng mạnh vào đầu tháng 3/2021, đạt đỉnh vào đầu tháng 7/2021 với số lượng 43.9 triệu liều/ngày và duy trì số lượng Vắc-xin tiêm mỗi ngày là hơn 20 triệu liều từ tháng 5/2021 đến nay cho thấy được Thế giới đang rất tích cực phủ rộng Vắc-xin trên toàn Thế giới.



Hình 3‑34. Biểu đồ cộng dồn lượng Vắc-xin được tiêm mỗi ngày của Thế giới

Hình 3-34 sử dụng biểu đồ đường để thể hiện số lượng Vắc-xin được tiêm mỗi ngày cộng dồn của Thế giới. Cho đến tháng 1/2022 đã có gần 10000 triệu liều Vắc-xin đã được tiêm trên toàn Thế giới, là 1 con số mang đến sự tích cực trong bối cảnh dịch COVID-19 vẫn đang hoành hành khắp 5 châu lục.



Hình 3‑35. Biểu đồ phần trăm dân số Thế giới được tiêm đầy đủ

Hình 3-35 sử dụng biểu đồ đường để thể hiện số phần trăm dân số đã được tiêm đầy đủ (đa số các loại Vắc-xin yêu cầu 2 mũi). Biểu đồ này chỉ mang tính chất tham khảo vì cho đến nay đã có nhiều quốc gia đã triển khai tiêm mũi 3 (mũi bổ sung) cho dân số của họ cho nên số phần trăm ở biểu đồ này sẽ lớn hơn so với thực tế.

**Nhận xét và đánh giá:** Thế giới đã trải qua hơn 2 năm chống chọi với dịch COVID-19 tuy nhiên Vắc-xin đã được phát triển, sản xuất và triển khai tiêm ngừa đã và đang từng bước đưa Thế giới trở lại trạng thái “Bình thường mới”.

### So sánh Việt Nam và Châu Á



Hình 3‑36. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin ở Châu Á

Hình 3-36 là bảng sơ lược về tiến độ tiêm chủng Vắc-xin COVID-19 ở Châu Á, đã có 50/50 quốc gia tham gia tiêm ngừa, 18 loại Vắc-xin được sử dụng, hơn 6104 triệu liều Vắc-xin đã được tiêm, hơn 1803 triệu người đã được tiêm và hơn 1326 triệu người đã tiêm đủ 2 mũi (đối với loại Vắc-xin yêu cầu 2 mũi). Tất cả các quốc gia và vùng lãnh thổ đều đã tham gia tiêm ngừa Vắc-xin vì dịch COVID-19 đã lây lan rộng khắp lãnh thổ của khu vực châu Á.



Hình 3‑37. Biểu đồ số lượng Vắc-xin mỗi ngày của các nước Châu Á

Hình 3-37 là biểu đồ đường của các quốc gia tại châu Á có số lượng Vắc-xin tiêm mỗi ngày cùng với Việt Nam. Trung Quốc là quốc gia có số lượng mũi tiêm một ngày cao nhất khi đạt đinh 22.4 triệu liều/ngày. Tiếp đến là Ấn Độ với gần 10 triệu liều/ngày. Cuộc chạy đua Vắc-xin ở Châu Á bắt đầu từ tháng 5/2021 cũng là lúc dịch bùng phát mạnh ở khu vực châu Á nói chung và Việt Nam nói riêng.



Hình 3‑38. Biểu đồ số lượng Vắc-xin cộng dồn của các nước Châu Á

Hình 3-38 là biểu đồ đường thể hiện số lượng Vắc-xin cộng dồn của Việt Nam và các nước châu Á. Trung Quốc là quốc gia có xu hướng tăng mạnh nhất so với phần còn lại, tăng mạnh vào đầu tháng 7/2021 đến giữa tháng 9/2021, tính cho đến ngày 26/12/2021 đã tiêm được hơn 2.5 tỷ liều. Tiếp đến là Ấn Độ, có xu hướng tăng từ tháng 8/21 cho đến nay và đạt được hơn 1 tỷ mũi tiêm.



Hình 3‑39. Biểu đồ phân phối tiêm Vắc-xin mỗi ngày của Việt Nam

Hình 3-39 là biểu đồ cột thể hiện số lượng Vắc-xin phân phối mỗi ngày của Việt Nam. Có hơn 140 ngày mà Việt Nam tiêm được 0.2 triệu liều/ngày và khoảng 5 ngày Việt Nam tiêm được từ 1.6 đến 1.8 triệu liều/ngày. Từ 0.6 đến 1.4 triệu liều/ngày là số lượng tiêm đạt được trong 20 ngày.

### So sánh Việt Nam và Đông Nam Á (South East Asia)



Hình 3‑40. Bảng sơ lược về tiến độ tiêm chủng Vắc-xin ở Đông Nam Á

Hình 3-40 là bảng sơ lược về tiến độ tiêm chủng Vắc-xin COVID-19 trên toàn Thế giới, đã có 10/11 quốc gia tham gia tiêm ngừa, 10 loại Vắc-xin được sử dụng, hơn 754 triệu liều Vắc-xin đã được tiêm, hơn 352 triệu người đã được tiêm và hơn 263 triệu người đã tiêm đủ 2 mũi (đối với loại Vắc-xin yêu cầu 2 mũi). Đã có 10 quốc gia khu vực Đông Nam Á tham gia tiêm ngừa Vắc-xin cho thấy các quốc gia khu vực này đang rất tích cực để tăng tỷ lệ phủ Vắc-xin của họ.



Hình 3‑41. Biểu đồ số lượng Vắc-xin mỗi ngày của Đông Nam Á

Hình 3-41 là biểu đồ đường thể hiện số lượng Vắc-xin tiêm mỗi ngày của các Quốc gia Đông Nam Á. Indonesia là quốc gia có số lượng Vắc-xin tiêm mỗi ngày cao nhất khi đạt đỉnh là 1.9 triệu/ngày), đứng thứ 2 là Việt Nam khi đạt đỉnh là gần 1.6 triệu liều. Số liều Vắc-xin tăng mạnh vào đầu tháng 7/2021 cũng chính là thời điểm dịch bùng phát mạnh ở khu vực Đông Nam Á, dẫn đến các quốc gia triển khai tiêm ngừa nhanh chóng để chống lại dịch COVID-19.



Hình 3‑42. Biểu đồ số lượng Vắc-xin cộng dồn của Đông Nam Á

Hình 3-42 là biểu đồ đường thể hiện số lượng Vắc-xin cộng dồn của Việt Nam và các nước Đông Nam Á. Indonesia là quốc gia có xu hướng tăng mạnh nhất so với phần còn lại, tăng mạnh vào đầu tháng 7/2021 đến nay, tính cho đến ngày 26/12/2021 đã tiêm được hơn 250 triệu liều. Tiếp đến là Việt Nam, có xu hướng tăng từ tháng 9/2021 đến nay và đạt được gần 150 triệu mũi tiêm.



Hình 3‑43. Biểu đồ Scatter về tổng số lượng Vắc-xin của Đông nam Á

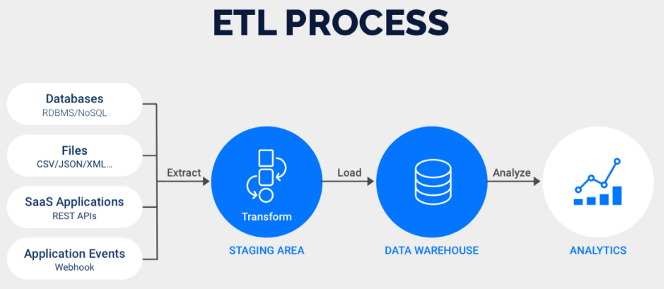
Hình 3-43 là biểu đồ Scatter thể hiện tổng số lượng Vắc-xin của các nước Đông Nam Á. Biểu đồ này có tê tương tác trực tiếp để xem số lượng Vắc-xin đã tiêm được của từng quốc gia. Ví dụ ngày 4/7/2021, Indonesia có 46.33 triệu liều, Philippines có 11.7 triệu liều, Thái Lan có 11.06 triệu liều và Việt Nam có 3.903 triệu liều.

## Tự động hóa tiến trình Extract-Transform-Load

### Giới thiệu về Extract-Transform-Load

**E**xtract-**T**ransform-**L**oad viết tắt là ETL, nghĩa là Trích xuất-Biến đổi-Tải, là quy trình sao chép, trích xuất, vận chuyển dữ liệu từ một hoặc nhiều nguồn, biến đổi phù hợp và đưa vào hệ thống đích đến.

Extract là quá trình đọc dữ liệu từ nhiều nguồn khác nhau, ví dụ như từ API, từ file excel, csv hay từ một Cơ sở dữ liệu khác, trong giai đoạn này dữ liệu được thu thập từ nhiều nguồn khác nhau. Transform là quá trình biến đổi dữ liệu được trích xuất thành dạng cần có trước khi được đưa vào Cơ sở dữ liệu. Load là quá trình ghi chép, đưa dữ liệu vào Cơ sở dữ liệu đích.



Hình 3‑44. Quy trình thực hiện ETL

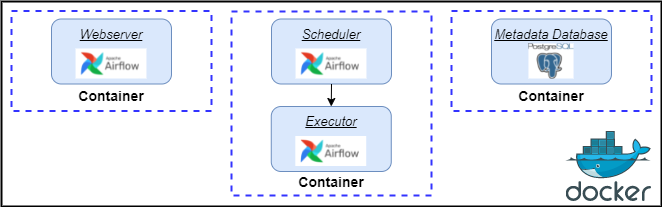
Giai đoạn Trích xuất – Extract: Dữ liệu đến từ rất nhiều nguồn cả có cấu trúc (structured) và không cấu trúc (unstructured), ví dụ như ứng dụng bán hàng, hệ thống quản lý khách hàng, nền tảng lưu trữ dữ liệu …

Giai đoạn Chuyển đổi – Transform: Áp dụng các quy định và quy tắc để áp dụng nhằm đảm bảo khả năng truy cập và chất lượng dữ liệu, quy trình chuyển đổi dữ liệu gồm: làm sạch dữ liệu (data cleaning), tiêu chuẩn hóa (standardize), xác minh, sắp xếp, loại bỏ dữ liệu trùng lặp (remove duplicate) …

Giai đoạn Tải: Tải dữ liệu vào một đích mới, dữ liệu có thể được tải tất cả cùng một lúc (tải đầy đủ) hoặc theo từng khoảng theo lịch trình thời gian (tải tăng dần).

### Tự động hóa bằng Apache Airflow và triển khai bằng Docker

Trong phần này, tôi sử dụng Docker và Apache Airflow để cài đặt và thiết lập một quy trình ETL từ các nguồn dữ liệu đến Cơ sở dữ liệu. Để cài đặt Apache Airflow, tôi sử dụng Docker để tạo các Container chứa các image của các thành phần cần thiết.



Hình 3‑45. Các thành phần của Docker

Hình 3-45 là các thành phần kiến trúc của Docker bao gồm: Webserver, Scheduler, Executor và Metadata Database. Mỗi thành phần được chứa trong 1 Container hay được gọi là Image, những Container này giao tiếp được với nhau khi cùng được chạy trên cùng 1 Docker Engine.

* Webserver: là giao diện web để quản lý các quy trình công việc (DAG), quản lý môi trường, các luồng môi trường Airflow và thực hiện các hành động quản trị cung cấp khả năng quản lý người dùng, vai trò và các cấu hình khác nhau.
* Scheduler: là bộ lập lịch giám sát tất cả các tác vụ và DAG, sau đó kích hoạt các thực thể (instances) của tác vụ ngay khi các phần phụ thuộc (dependencies) đã hoàn tất. Đằng sau mỗi scheduler là thực hiện những quy trình phụ, dùng để giám sát và đồng bộ hóa với tất cả các DAG trong thư mục DAG đã chỉ định.
* Executor: là thành phần thực thi các tác vụ trong khi Scheduler tổ chức, sắp xếp các tác vụ. Có nhiều loại thực thi đi kèm với Airflow như SequentialExecutor, LocalExecutor, CeleryExecutor và KubernetesExecutor.
* Metadata Database: là Cơ sở dữ liệu lưu trữ các cấu hình, như là các biến và kết nối, thông tin người dùng, vai trò và chính sách. Là nguồn cung trung thực của Airflow Scheduler cho tất cả các siêu dữ liệu (metadata) liên quan đến DAG, thời gian biểu, số liệu thống kê từ các lần chạy DAG. Airflow sử dụng SQLAlchemy và ORM (**O**bject **R**elational **M**apping) bằng Python để kết nối và tương tác với Cơ sở siêu dữ liệu ở lớp ứng dụng ở bên dưới. Cơ sở siêu dữ liệu của Airflow có tổng cộng 30 bảng được lưu trữ trên lược đồ.

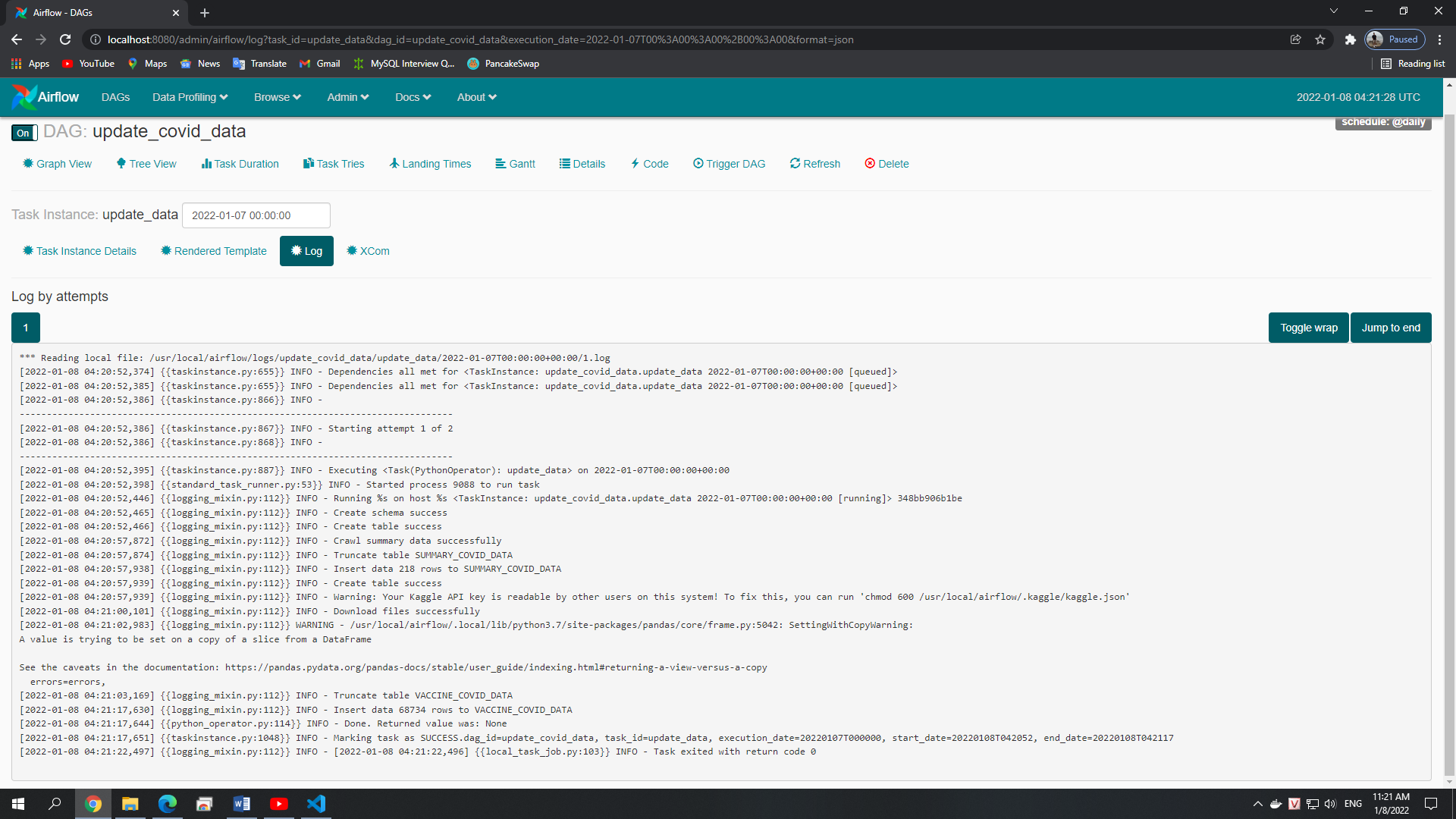
Các Container của Docker được định nghĩa bằng công cụ có tên là Docker-Compose, là công cụ để định nghĩa và chạy multi-container cho ứng dụng Docker. Với Docker-Compose, lập trình viên sử dụng file YAML để thiết lập các services cho ứng dụng. Sau đó dùng những command để create và run những thiết lập đó. Sử dụng khá đơn giản chỉ với 3 bước:

* Khai báo môi trường cho ứng dụng bằng Dockerfile (là một dạng text dùng để thiết lập cấu trúc cho docker image nhờ chứa một tập hợp các câu lệnh ví dụ như RUN, COPY, FROM, CMD …).
* Khai báo các services cần thiết để chạy ứng dụng trong file docker-compose.yaml.
* Chạy docker-compose up để khởi động và chạy ứng dụng.



Hình 3‑46. Cách tạo Docker Container từ Dockerfile

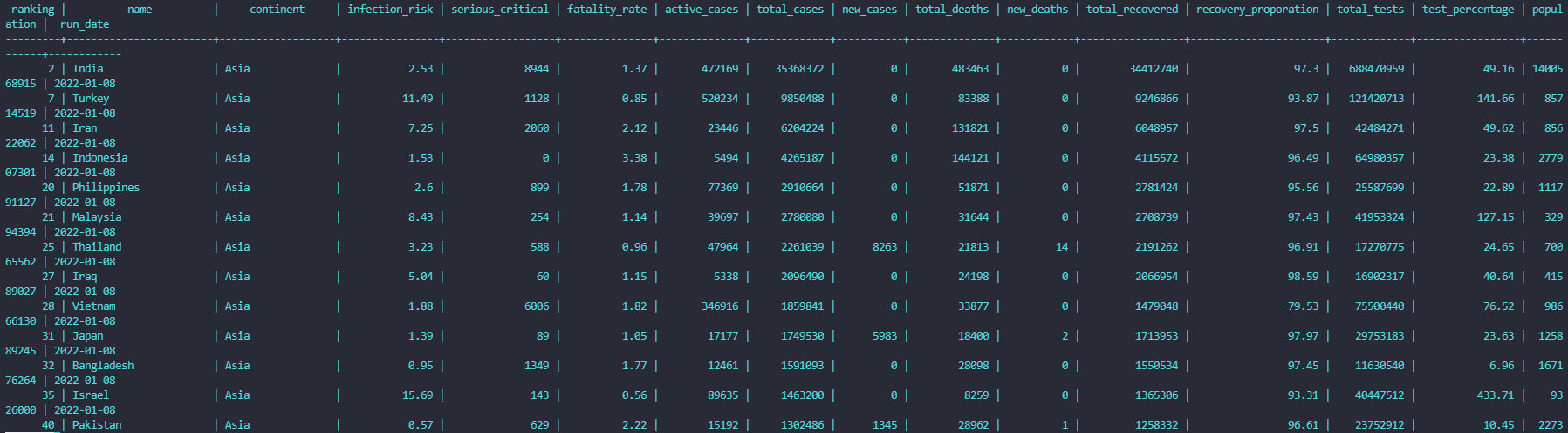
Để khởi động Docker, cần thiết lập các cấu hình cần thiết cho 3 Container: Webserver, Scheduler và Metadata Database trong file “docker-compose.yaml” để định nghĩa các Image như khai báo môi trường, phiên bản Airflow, thiết lập thư mục, các thư viện cần thiết. Sử dụng terminal để khởi động Docker bằng lệnh “docker-compose up –build”. Sử dụng port localhost:8080 cho Webserver và Scheduler, Cơ sở dữ liệu PostgreSQL để lưu trữ Metadata.



Hình 3‑47. Log của DAG khi đã thực hiện thành công

Hình 3-47 là log của DAG sau khi đã thực hiện thành công, các thiết lập của DAG bao gồm: retries=1 (số lần chạy lại khi bị lỗi), retry\_delay=timedelta(minutes=5) (thời gian để chạy lại khi bị lỗi), start\_date=datetime(2021, 1, 1) (ngày bắt đầu chạy DAG).

Dữ liệu sau quy trình ETL sẽ được lưu trữ vào Container chứa Cơ sở dữ liệu PostgreSQL. Để kết nối vào Cơ sở dữ liệu PostgreSQL cần thực thi câu lệnh của Docker trên Terminal bằng cách sử dụng tên Container và bật psql (giao diện dòng lệnh cho Postgres, với cú pháp là “docker exec -it [container\_name] psql -U [postgres\_user]”, trong trường hợp này cú pháp sẽ là “docker exec -it covid-19\_postgres\_1 psql -U airflow”.

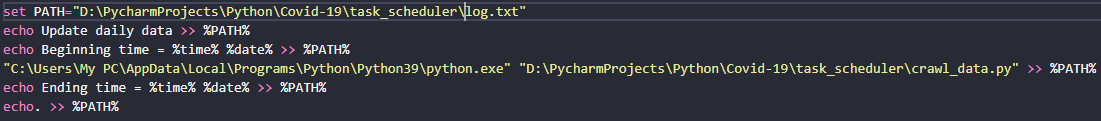


Hình 3‑48. Dữ liệu được lưu trữ trong CSDL Postgres

### Tự động hóa bằng Task Scheduler

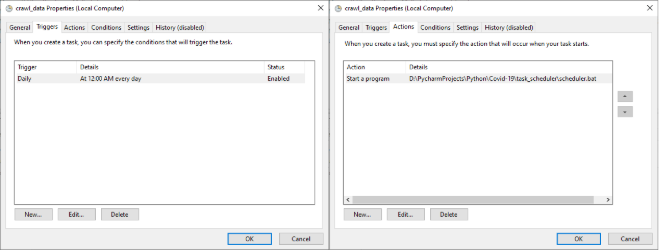
Task Scheduler là một công cụ có sẵn của Windows, giúp người dùng thực hiện một số tác vụ và hành động trên máy tính một cách tự động. Task Scheduler có nhiều tính năng hỗ trợ cho máy tính, khả năng tự động hóa tác vụ người dùng muốn chạy trên Windows một cách hiệu quả và nhanh chóng.

Task Scheduler cho phép người dùng thiết lập những cài đặt, yêu cầu cũng như điều kiện để kích hoạt tự động, cung cấp lịch sử thực thi để người dùng quản lý. Task Scheduler cho phép người dùng sử dụng nhiều loại định dạng file để chạy như .exe, .bat hoặc bằng những câu lệnh.



Hình 3‑49. File BAT sử dụng trong Task Scheduler

Hình 3-49 là nội dung file scheduler.bat để kích hoạt tự động chạy file Python bằng Task Scheduler và sử dụng để ghi file log của chương trình chạy.



Hình 3‑50. Thiết lập trong Task Scheduler

Hình 3-50 là thiết lập “Triggers” và “Actions” trong Task Scheduler. Với “Triggers”, tác vụ sẽ được khởi chạy vào lúc 12:00AM hằng ngày. Với “Actions”, tác vụ sẽ khởi chạy file “scheduler.bat”.

# XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

## Thuật toán SVM, Linear Regression và Bayesian Ridge Regression

### Thuật toán SVM

SVM viết tắt của **S**upport **V**ector **M**achines là thuật toán máy học (Machine Learning) có giám sát, được sử dụng phổ biến trong các bài toán phân lớp (classification) và hồi quy (regression). SVMs được chia thành 2 loại:

* SVM: dùng cho các bài toán phân lớp.
* SVR (**S**upport **V**ector **R**egression): dùng cho các bài toán hồi quy.

SVM có nhiều loại bài toán con như: Linear, Non-linear, Soft-margin, Hard-margin, binary-class và multi-class. Với bài toán SVM, ý tưởng là tìm một siêu phẳng (hyper-plane) để phân tách các điểm dữ liệu, siêu phẳng này sẽ chia không gian dữ liệu thành các vùng khác nhau và mỗi vùng sẽ chứa một loại dữ liệu.



Hình 4‑1. Siêu phẳng (hyper-plane) trong SVM

Trong một không gian có rất nhiều siêu phẳng (Hình 4-1), để tối ưu hóa và mang lại kết quả chính xác cần phải tìm ra siêu phẳng tối ưu nhất (Optimal hyperplane). Siêu phẳng này được biểu diễn bằng hàm số:  (với W và X là vector <W.X> là tích vô hướng) hay  (WT là ma trận chuyển vị). Cách chọn siêu phẳng tối ưu:



Hình 4‑2. Margin của siêu phẳng trong SVM

Trong hình 4-2, siêu phẳng phân tách tập dữ liệu thành 2 lớp (chấm đỏ và chấm xanh) H0 thỏa mãn <W.X> + b =0, siêu phẳng này tạo ra thêm 2 nửa không gian (half-space) dữ liệu: Không gian chứa dữ liệu lớp chấm đỏ Xi thỏa mãn và không gian chứa dữ liệu chấm xanh Xj thỏa mãn.

Chọn hai siêu phẳng lề H1 đi qua điểm thuộc lớp chấm đỏ và H2 đi qua điểm thuộc lớp chấm xanh đều song song với H0.

* H1 : <W.X> + b =-1
* H2 : <W.X> + b =1

Khoảng cách từ H1 đến H0 là d-

Khoảng cách từ H2 đến H0 là d+

m = d- + d+ được gọi là mức lề

* **Siêu phẳng tối ưu** là siêu phẳng phân tách có lề (margin) lớn nhất, trong lý thuyết học máy: Một siêu phẳng tối ưu sẽ cực tiểu hóa giới hạn lỗi mắc phải.

Để tìm được siêu phẳng tối ưu, trong thuật toán SVM cần phải tìm được margin (lề):

Khoảng cách từ 1 điểm Xk đến siêu phẳng H0: trong đó ||W|| là độ dài vector W: ||W||=<W.W>= 

Khoảng cách từ một điểm Xi nằm trên H1 và H0: 

Khoảng cách từ một điểm Xj nằm trên H2 và H0: 

🡪 Các điểm nằm trên 2 mặt phẳng H1 và H2 được gọi là Support Vector.

Cực tiểu hóa (minimization): với điều kiện:

* , nếu yi=-1
* , nếu yi=1
* Bài toán **Hard-margin** của SVM.

Ở hình 4-2, việc xác định siêu phẳng trong điều kiện lý tưởng khi mà tập dữ liệu có thể phân tách tuyến tính, tìm được 2 siêu phẳng lề H1 và H2 mà không có điểm nhiễu nào nằm giữa chúng. Vậy trong trường hợp có nhiễu, bài toán cần sử dụng biến **slack.**



Hình 4‑3. Siêu phẳng có chứa tín hiệu nhiễu

Với trường hợp có tín hiệu nhiễu, cần sử dụng biến slack:

* , nếu yi=-1
* , nếu yi=-1

Cực tiểu hóa: 

Với điều kiện:

* 
* 

C: tham số xác định mức độ phạt (penalty degree) đối với lỗi, là một tham số quan trọng mà cần phải tối ưu trong các bài toán SVM.

* Bài toán **Soft-margin** của SVM

Với bài toán có không gian dữ liệu phi tuyến tính (non-linear) khi không tìm được siêu phẳng H0­ Chúng ta cần phải biểu diễn (ánh xạ) dữ liệu từ dữ liệu từ không gian ban đầu X sang không gian F bằng một hàm ánh xạ phi tuyến để giải quyết bài toán:

* 
* 
* Bài toán **Non-linear** của SVM



Hình 4‑4. Non-linear trong SVM

### Thuật toán Linear Regression

#### Giới thiệu Linear Regression

Thuật toán Linear Regression (Hồi quy tuyến tính) là thuật toán với phương pháp tiếp cận tuyến tính để dự đoán mối quan hệ giữa biến phụ thuộc Y (còn được gọi là biến kết cục) trên trục tung Oy dự trên các biến độc lập X (còn được gọi là biến giải thích) trên trục hoành Ox trong mô hình. Hồi quy tuyến tính được phát triển thành Mô hình hồi quy tuyến tính (Linear Regression Model) là công cụ quan trọng trong lĩnh vực Kinh tế lượng, là phương pháp thống kê hồi quy và dự báo dữ liệu.

Phương pháp Hồi quy Tuyến tính tổng quát: Mô hình Hồi quy Tuyến tính bội (Multiple Regression model) với dạng Mô hình Hồi quy tổng thể (Population Regression model) với n-1 biến giải thích có dạng:

(1)

Trong đó:

* Y: biến phụ thuộc (dependent variable).
* X: các biến giải thích hay biến độc lập (explanatory/independent variables).
* βn: hệ số của các biến độc lập, trong đó β1 là hệ số tự do.
* u: hạng nhiễu hay sai số ngẫu nhiên (random hay stochastic error term).
* i: ký hiệu cho quan sát thứ i trong tổng thể.

Phương trình (1) rút gọn thành mô hình tổng thể (population model):

 Với 

* βx: thành phần tất định (deterministic component), là 1 trung bình có điều kiện (conditional mean) của Yi, nghĩa là E(Yi|X): giá trị trung bình của Y khi X là giá trị được cho trước.
* ui: thành phần phi hệ thống (nonsystematic component).

Phương trình: Y = β0 + β1\*X, Với β1 là độ dốc (slope); β0 là chặn (intercept)

* β1 = Tương quan \* ( Độ lệch chuẩn của Y / Độ lệch chuẩn của X)
* β0 = Trung bình (Y) – B1 \* Trung bình (X)



Hình 4‑5. Mô hình Hồi quy tuyến tính Linear Regression

Với: Độ lệch chuẩn của X.

Với: Độ lệch chuẩn của Y.

Độ mạnh của Mô hình hồi quy tuyến tính: được đo lường thông qua **hệ số xác định r2** (coefficient of determination) hay **R-Square (R2)** với dao động từ 0 đến 1.

Công thức tính R-Square (R2): ***R2* = (TSS – RSS)/TSS.**



Hình 4‑6. R-Square (R2) của Mô hình hồi quy tuyến tính

* Tổng các diện tích TSS (**T**otal **S**um of **S**quares): phép đo tổng hợp biến thiên trong tỷ lệ đáp ứng / biến phụ thuộc và là số lượng biến thiên có trong đáp ứng trước khi hồi quy được thực hiện.
* **R**esidual **S**um of **S**quares (RSS): RSS đo lường số phương sai không giải thích được sau khi thực hiện mô hình hồi quy trong một tập dữ liệu.
* TSS – RSS: là hiệu số giữa TSS và RSS để đo lường mức độ thay đổi trong đáp ứng được giải thích.

#### GIới thiệu OLS

OLS là viết tắt của **O**rdinary **L**east **S**quare mô hình hồi quy bằng phương pháp bình phương cực tiểu hay được gọi là mô hình Pooled OLS (Pool) là trường hợp đặc biệt của phương pháp bình phương cực tiểu tổng quát (**G**eneralized **L**east **S**quares method – GLS) để tìm đường hồi quy gần nhất với giá trị liên tục của biến phụ thuộc (tổng bình phương của các nhiễu hay sai số ∑ui2 nhỏ nhất). Công thức tổng quát:

Yi = b1 + b2X2i+ b3X3i +…+ bkXki+ui

Phương sai và sai số chuẩn: , trong đó:

* Phương sai (Variance): Tỷ lệ của tổng bình phương phần dư (RSS) chia cho (n – k) với bậc tự do, với n là cỡ mẫu và k là số tham số hồi quy ước lượng.
* 𝜎̂ (**S**tandard **E**rror of the **R**egression - SER): Sai số chuẩn của hồi quy, là độ lệch chuẩn của các giá trị Y xoay quanh đường hồi quy.



Hình 4‑7. Phương pháp Ordinary Least Square

### Thuật toán Bayesian Ridge Regression

Kỹ thuật hồi quy Bayes (Bayesian Regression) được sử dụng để bao gồm các tham số chính quy hóa trong quá trình ước lượng: tham số chính quy hóa không được định cứng mà sẽ được điều chỉnh theo dữ liệu có sẵn. Điều này thực hiện bằng cách giới thiệu cò mồi không thông tin trên các siêu tham số của mô hình. Các *l2* chính quy hóa được sử dụng trong hồi quy và phân loại Ridge bằng với việc tìm 1 ước lượng hậu kỳ tối đa Gaussian trên các hệ số ** với độ chính xác ****. Hồi quy Ridge giúp giải quyết một số vấn đề của OLS bằng cách áp đặt hình phạt (penalty) với kích cỡ của các hệ số, hệ số Ridge tối thiểu hóa tổng bình phương (RSS) còn lại:



Thay vì đặt  thủ công, cho nó như 1 biến ngẫu nhiên ước lượng từ dữ liệu. Với mô hình xác suất đầy đủ, đầu ra ***y***là Gaussian được phân phối xung quanh:  với là biến ngẫu nhiên được ước tính từ dữ liệu.

* Ưu điểm của Hồi quy Bayes:
* Thích ứng với dữ liệu trong tầm tay.
* Sử dụng để bao gồm các tham số chính quy hóa trong quy trình ước tính.
* Nhược điểm của Hồi quy Bayes:
* Việc suy diễn mô hình có thể tốn nhiều thời gian.

Hồi Quy Bayesian Ridge ước tính một mô hình xác suất của bài toán Hồi quy, giá trị trước cho hệ số**đưa ra bởi Gaussian hình cầu:. Các mồivà được chọn là phân bố Gamma, liên hợp cho độ chính xác Gaussian, mô hình kết quả gọi là Hồi quy Bayesian Ridge. Các tham số, và  được ước tính phù hợp với mô hình, các tham số chính quy hóa và  ước tính bằng tối đa hóa khả năng cận biên log (log marginal likelihood). Giá trị ban đầu của tối đa hóa có được đặt bằng siêu tham số alpha\_init và lambda\_init. Có 4 siêu tham số , ,  và  của các phân phối Gamma và , mặc định .

So sánh với ước lượng OLS, trọng số có xu hướng dịch chuyển về phía số không, điểu giúp ổn định thuật toán. Việc ước lượng hóa mô hình được lặp đi lặp lại tối đa hóa khả năng log biên của các quan sát.



Hình 4‑8. So sánh các trọng số của mô hình

Hình 4-8 là biểu đồ so sánh các dự đoán và độ không chắc chắn của Hồi quy Bayesian Ridge cho hồi quy một chiều bằng phương pháp Polynomial Feature và ước lượng OLS.

Việc lựa chọn các tham số khởi tạo tham số chính quy hóa (alpha, lambda) khá quan trọng. Do các tham số chính quy hóa được xác định bởi quy trình lặp đi lặp lại phụ thuộc vào biến khởi tạo khi điều chỉnh đường cong Hồi quy Bayesian Ridge.



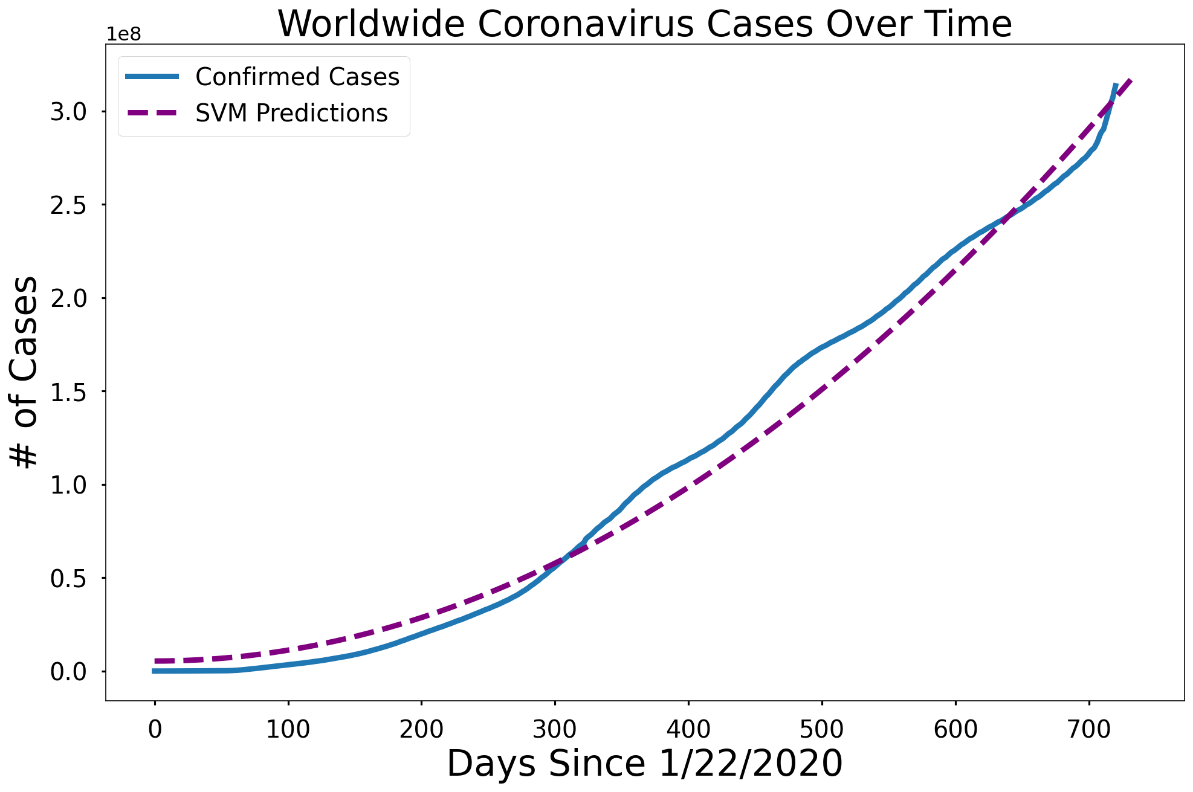
Hình 4‑9. Curve Fitting với Hồi quy Bayesian Ridge

Hình 4-9 là ví dụ hình sin được tính bằng đa thức được sử dụng các cặp giá trị ban đầu. Với cặp mặc định (=1.9, =1) độ lệch đường cong quá lớn và phương sai nhỏ. Thay vào đó là =1, =10-3 để giảm độ lệch.

## Áp dụng vào mô hình dự đoán

Trong phần này, tôi sử dụng 3 thuật toán: SVR, Linear Regression và Linear Regression để dự đoán số ca nhiễm của Thế giới trong 15 ngày tiếp theo.

### Áp dụng thuật toán SVR



Hình 4‑10. Dự đoán của thuật toán SVR

Hình 4-10 là biểu đồ đường thể hiện kết quả dự đoán của thuật toán SVR. Đường màu xanh là số ca nhiễm thực tế và đường màu tím gạch đứt là số ca nhiễm dự đoán.

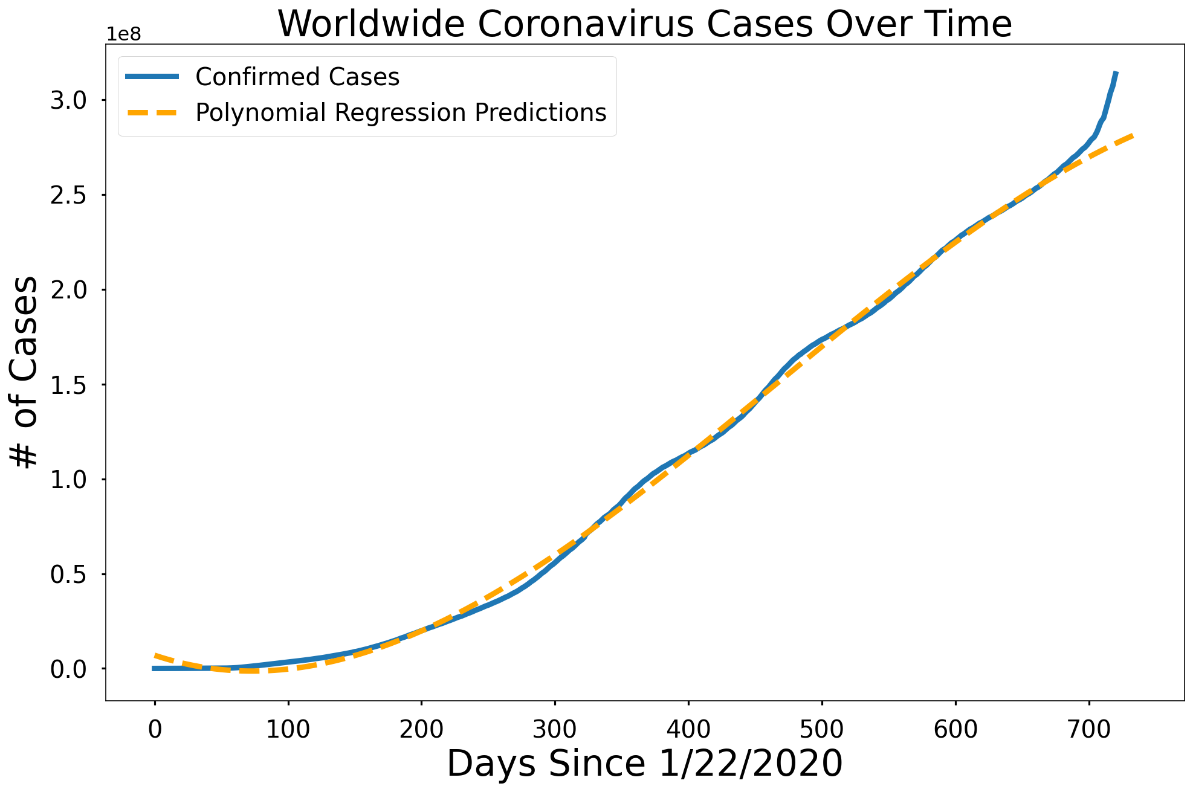
Tham số được sử dụng trong mô hình SVR:

* kernel: “poly”, gamma (hệ số Kernel) = 0.01, epsilon=1, degree (hàm của đa thức Kernel)=2, C (tham số chính quy)=0.1

Thông số kỹ thuật (metric) của mô hình thuật toán Polynomial Regression:

* Lỗi trung bình tuyệt đối (Mean Absolute Error): MAE=9748265.19
* Lỗi bình phương trung bình (Mean Squared Error): MSE=105786660194903
* R-Square= 0.98
* Hệ số chặn: intercept=5371633.73

### Áp dụng thuật toán Linear Regression



Hình 4‑11. Dự đoán của thuật toán Polynomial Regression

Hình 4-11 là biểu đồ đường thể hiện kết quả dự đoán của thuật toán Polynomial Regression. Đường màu xanh là số ca nhiễm thực tế và đường màu cam gạch đứt là số ca nhiễm dự đoán.

Do số lượng feature đầu vào để training nhỏ (hơn 700) nên cần sử dụng Polynomial Features để cải thiện các feature của thuật toán.

* Giới thiệu về **Polynomial Feature** và công dụng:

Trong các bài toán với số lượng dữ liệu đầu vào nhỏ, tạo Polynomial Feature để cải thiện thuật toán. Để cải thiện các feature của thuật toán bằng nhiều cách khác nhau, có thể kết hợp nhiều feature lại thành 1. Cách thường được sử dụng nhất là thêm vào các bậc cao hơn cho cùng 1 feature để nâng bậc cho hàm: *y=f(x)=b0+b1x1* thành hàm bậc 2: *y=f(x)=b0+b1x1+b2x12*hay bậc 3: *y=f(x)=b0+b1x1+b2x12+ b3x13.*

Một điều quan trọng trong triển khai hồi quy đa thức (Polynomial Regression) là việc lựa chọn mức độ tối ưu của hàm hồi quy đa thức. Việc lựa chọn này không có quy tắc cụ thể, tùy thuộc vào từng bài toán, có 2 vấn đề cần chú ý khi lựa chọn bậc cho hàm (degree): underfitting and overfitting.



Hình 4‑12. Các mô hình Underfitted, well-fitted and overfitted

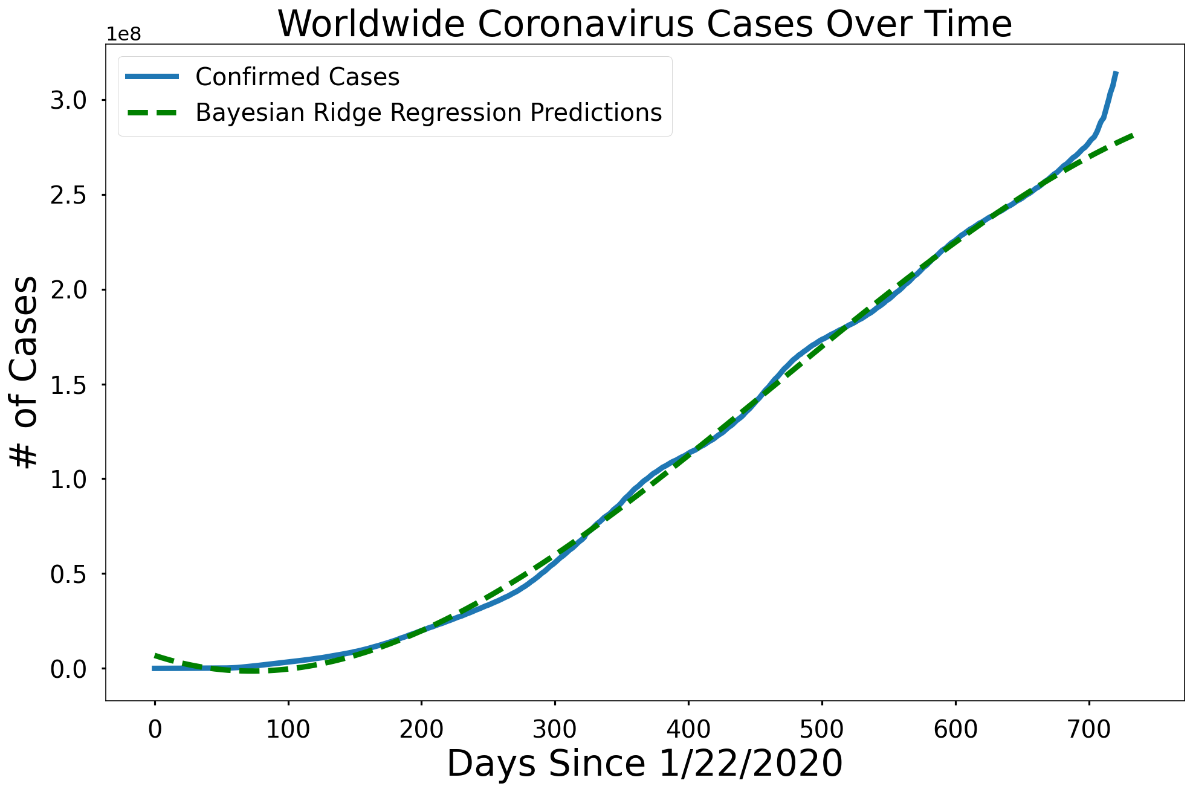
Hình 4-12 là các ví dụ về các trường hợp Underfitting, Overfitting và Well-fitting khi sử dụng bậc 1, 2, 3 và 5. Underfitting là khi một mô hình có kết quả dự đoán không chính xác, không thể nắm bắt chính xác các yếu tố phụ thuộc của dữ liệu. Overfitting là khi một mô hình học quá tốt dữ liệu hiện có, tuy nhiên không khái quát tốt và có R2 thấp khi có dữ liệu mới.

Trong bài toán dự đoán này, dữ liệu training đầu vào là số ngày nên tôi dùng Polynomial Features để nâng lên bậc 3 (degree=3) nên thuật toán này có tên Polynomial Regression, phương trình có dạng: *y=f(x)=b0+b1x1+b2x12+ b3x13.*

Thông số kỹ thuật (metric) của mô hình thuật toán Polynomial Regression:

* Lỗi trung bình tuyệt đối (Mean Absolute Error): MAE=8417567.3
* Lỗi bình phương trung bình (Mean Squared Error): MSE=159201722308772
* R-Square=0.998
* Hệ số chặn: intercept=0
* Hệ số góc: coefficients=(6.962e+6, -2.349e+5, 1.745e+3, -1.247e+0)
* y=f(x)= 6.962e+6 -2.349e+5*x1*+1.745e+3*x12*-1.247e+0*x13*

### Áp dụng thuật toán Bayesian Ridge Regression



Hình 4‑13. Dự đoán của thuật toán Bayesian Ridge Regression

Hình 4-13 là biểu đồ đường thể hiện kết quả dự đoán của thuật toán Bayesian Ridge Regression. Đường màu xanh là số ca nhiễm thực tế và đường màu xanh lá gạch đứt là số ca nhiễm dự đoán.

Sử dụng tham số:

* n\_iter (số lần lặp tối đa)=40
* =[1e-7, 1e-6, 1e-5, 0.0001,0.001]
* tol = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]
* normalize = [True, False]

Thông số kỹ thuật (metric) của mô hình thuật toán Bayesian Ridge Regression:

* Lỗi trung bình tuyệt đối (Mean Absolute Error): MAE=8367661.38
* Lỗi bình phương trung bình (Mean Squared Error): MSE=158067576289895
* R-Square=0.998
* Hệ số chặn: intercept=0
* Hệ số góc: coefficients=(6.843e+6, -2.335e+5, 1.741e+3, -1.244e+0)

## Kết quả dự đoán và so sánh

|  |  |  |  |
| --- | --- | --- | --- |
| Ngày | SVR | Polynomial | Bayesian Ridge |
| 12/01/2022 | 307974931 | 277142061 | 277222710 |
| 13/01/2022 | 308814912 | 277476946 | 277558670 |
| 14/01/2022 | 309656057 | 277809916 | 277892724 |
| 15/01/2022 | 310498366 | 278140965 | 278224863 |
| 16/01/2022 | 311341840 | 278470085 | 278555080 |
| 17/01/2022 | 312186478 | 278797268 | 278883368 |
| 18/01/2022 | 313032280 | 279122507 | 279209719 |
| 19/01/2022 | 313879246 | 279445794 | 279534127 |
| 20/01/2022 | 314727377 | 279767122 | 279856582 |
| 21/01/2022 | 315576671 | 280086483 | 280177079 |
| 22/01/2022 | 316427130 | 280403870 | 280495609 |
| 23/01/2022 | 317278753 | 280719275 | 280812164 |
| 24/01/2022 | 318131541 | 281032691 | 281126739 |
| 25/01/2022 | 318985492 | 281344111 | 281439324 |
| 26/01/2022 | 319840608 | 281653527 | 281749913 |

Bảng 4‑1. Bảng so sánh kết quả dự đoán

Bảng 4-1 là bảng kết quả dự đoán của 3 thuật toán SVR, Polynomial Regression và Bayesian Ridge Regression. Thời gian dự đoán là 15 ngày từ 12/1/2022 đến 26/1/2022. Với thuật toán SVR, số ca nhiễm dự đoán tăng từ 307.9 triệu đến 319.8 triệu với độ dao động ∆=11.865.677, trung bình 791.045 ca nhiễm/ngày. Với thuật toán Polynomial Regression, số ca nhiễm dự đoán tăng từ 277.1 triệu đến 281.6 triệu với độ dao động ∆=4.511.466, trung bình 300.764 ca nhiễm/ngày. Với thuật toán Bayesian Ridge Regression, số ca nhiễm dự đoán tăng từ 277.2 triệu đến 281.7 triệu với độ dao động ∆=4.527.203, trung bình 301.813 ca nhiễm/ngày. Thuật toán SVR là thuật toán cho ra dự đoán có số ca nhiều nhất gấp 2 lần so với Polynomial Regression và Bayesian Ridge Regression.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ngày | Thực tế | SVR | %Δ | Polynomial | %Δ | Bayesian | %Δ |
| 28/12/2021 | 282870 | 295514 | 4.47 | 271894 | 3.88 | 271959 | 3.86 |
| 29/12/2021 | 284576 | 296337 | 4.13 | 272256 | 4.33 | 272323 | 4.31 |
| 30/12/2021 | 286550 | 297161 | 3.70 | 272617 | 4.86 | 272685 | 4.84 |
| 31/12/2021 | 288221 | 297985 | 3.39 | 272976 | 5.29 | 273045 | 5.27 |
| 01/01/2022 | 289348 | 298811 | 3.27 | 273334 | 5.53 | 273403 | 5.51 |
| 02/01/2022 | 290217 | 299639 | 3.25 | 273689 | 5.70 | 273759 | 5.67 |
| 03/01/2022 | 292741 | 300467 | 2.64 | 274043 | 6.39 | 274114 | 6.36 |
| 04/01/2022 | 295285 | 301296 | 2.04 | 274394 | 7.07 | 274467 | 7.05 |
| 05/01/2022 | 297798 | 302127 | 1.45 | 274744 | 7.74 | 274818 | 7.72 |
| 06/01/2022 | 300355 | 302959 | 0.87 | 275092 | 8.41 | 275167 | 8.39 |
| 07/01/2022 | 303270 | 303792 | 0.17 | 275439 | 9.18 | 275514 | 9.15 |
| 08/01/2022 | 305334 | 304626 | 0.23 | 275783 | 9.68 | 275859 | 9.65 |
| 09/01/2022 | 307343 | 305461 | 0.61 | 276125 | 10.16 | 276203 | 10.13 |
| 10/01/2022 | 310629 | 306298 | 1.39 | 276466 | 11.00 | 276545 | 10.97 |
| 11/01/2022 | 313498 | 307136 | 2.03 | 276805 | 11.70 | 276884 | 11.68 |

Bảng 4‑2. Bảng so sánh và sai số theo ngày của các thuật toán so với thực tế

\*Đơn vị: Nghìn ca

Bảng 4-2 là bảng so sánh số ca nhiễm thực tế và số ca nhiễm dự đoán của 3 thuật toán SVR, Polynomial Regression và Bayesian Ridge Regression cùng với phần trăm sai số (%Δ) của từng thuật toán so với thực tế, số ca nhiễm sử dụng đơn vị là nghìn ca. Với thuật toán SVR có %Δ cao nhất là 4.47% vào ngày 28/12/2021 và thấp nhất là 0.17% vào ngày 7/1/2022. Với thuật toán Polynomial Regression có %Δ cao nhất là 4.47% vào ngày 28/12/2021 và thấp nhất là 3.88% vào ngày 7/1/2022. Với thuật toán Bayesian Ridge Regression có %Δ cao nhất là 11.68% vào ngày 11/1/2022 và thấp nhất là 3.86% vào ngày 28/12/2021. Trong khoảng 15 ngày từ ngày 28/12/2021 đến ngày 11/1/2022, SVR là thuật toán có khả năng dự đoán tốt hơn so với 2 thuật toán còn lại khi mà %Δ trung bình là 2.243% so với 7.39% của Polynomial Regression và 7.37% của Bayesian Ridge Regression.

**Nhận xét và đánh giá:** Các kết quả dự đoán sẽ có tỷ lệ sai số trong mức cho phép và chỉ mang tính chất tham khảo nhằm giúp cho việc đối mặt với số ca nhiễm trong tương lai gần. Trong cả 3 mô hình dự đoán SVR, Polynomial Regression và Bayesian Ridge Regression đều sử dụng để dự đoán số ca nhiễm chỉ trong 15 ngày tiếp theo vì số ca nhiễm tăng hay giảm còn phụ thuộc vào nhiều yếu tố khác như khả năng tiêm ngừa, mức an sinh, điều kiện kinh tế và thói quen sinh hoạt của từng khu vực trên Thế giới. Số ca nhiễm dự đoán dựa theo dữ liệu có sẵn, tuy nhiên khả năng bùng phát dịch Toàn cầu vẫn đang ở trước mắt nên mọi người phải luôn đề cao tinh thần tự giác và bảo vệ bản thân.

## Áp dụng Polynomial Regression dự đoán số ca nhiễm ở Việt Nam



Hình 4‑14. Dự đoán số ca nhiễm ở Việt Nam

Hình 4-14 là biểu đồ đường thể hiện kết quả dự đoán số ca nhiễm trong 30 ngày tiếp theo của Việt Nam sử dụng thuật toán Polynomial Regression. Đường màu xanh là số ca nhiễm thực tế và đường màu đỏ gạch đứt là số ca nhiễm dự đoán.

Thông số kỹ thuật (metric) của mô hình thuật toán Polynomial Regression:

* Lỗi trung bình tuyệt đối (Mean Absolute Error): MAE=135543.06
* Lỗi bình phương trung bình (Mean Squared Error): MSE=25902047353
* R-Square=0.925
* Hệ số chặn: intercept=0
* Hệ số góc: coefficients=(-1.177e+5, 2.996e+3, -1.519e+1, 2.002e-2)
* y=f(x)= -1.167e+5+2.97e+3*x1*-1.508e+1*x12*+1.99e-2*x13*

|  |  |  |  |
| --- | --- | --- | --- |
| Ngày | Kết quả | Ngày | Kết quả |
| 12/01/2022 | 1651869 | **27/01/2022** | 1843105 |
| 13/01/2022 | 1664219 | **28/01/2022** | 1856314 |
| 14/01/2022 | 1676627 | **29/01/2022** | 1869581 |
| 15/01/2022 | 1689090 | **30/01/2022** | 1882907 |
| 16/01/2022 | 1701611 | **31/01/2022** | 1896291 |
| 17/01/2022 | 1714188 | **01/02/2022** | 1909733 |
| 18/01/2022 | 1726822 | **02/02/2022** | 1923234 |
| 19/01/2022 | 1739513 | **03/02/2022** | 1936794 |
| 20/01/2022 | 1752261 | **04/02/2022** | 1950413 |
| 21/01/2022 | 1765066 | **05/02/2022** | 1964091 |
| 22/01/2022 | 1777928 | **06/02/2022** | 1977828 |
| 23/01/2022 | 1790848 | **07/02/2022** | 1991624 |
| 24/01/2022 | 1803826 | **08/02/2022** | 2005479 |
| 25/01/2022 | 1816861 | **09/02/2022** | 2019394 |
| 26/01/2022 | 1829954 | **10/02/2022** | 2033369 |

Bảng 4‑3. Bảng kết quả dự đoán số ca nhiễm của Việt Nam

Bảng 4-3 là bảng kết quả dự đoán số ca nhiễm sử dụng thuật toán Polynomial Regression với thời gian dự đoán là 30 ngày từ ngày 12/01/2022 đến ngày 10/02/2022. Số ca nhiễm dự đoán tăng từ 1.652 triệu đến 203.33 triệu với độ dao động ∆=381.500, trung bình 12.716 ca nhiễm/ngày. Với số ca nhiễm trung bình này, con số này gần bằng với số ca nhiễm mỗi ngày thực tế của Việt Nam trong khoảng 5 tháng gần đây.

|  |  |  |  |
| --- | --- | --- | --- |
| Ngày | Thực tế | Polynomial | %Δ |
| 28/12/2021 | 1680985 | 1473288 | 12.356 |
| 29/12/2021 | 1694874 | 1484808 | 12.394 |
| 30/12/2021 | 1714742 | 1496382 | 12.734 |
| 31/12/2021 | 1731257 | 1508011 | 12.895 |
| 01/01/2022 | 1746092 | 1519695 | 12.966 |
| 02/01/2022 | 1763040 | 1531434 | 13.137 |
| 03/01/2022 | 1778976 | 1543228 | 13.252 |
| 04/01/2022 | 1800704 | 1555076 | 13.641 |
| 05/01/2022 | 1817721 | 1566981 | 13.794 |
| 06/01/2022 | 1843563 | 1578940 | 14.354 |
| 07/01/2022 | 1859841 | 1590956 | 14.457 |
| 08/01/2022 | 1876394 | 1603026 | 14.569 |
| 09/01/2022 | 1899575 | 1615153 | 14.973 |
| 10/01/2022 | 1914393 | 1627335 | 14.995 |
| 11/01/2022 | 1930428 | 1639574 | 15.067 |

Bảng 4‑4. Bảng so sánh và sai số theo ngày của các thuật toán so với thực tế

Bảng 4-4 là bảng so sánh kết quả dự đoán và phần trăm sai số của thuật toán Polynomial Regression so với dữ liệu thực tế ở Việt Nam. %Δ lớn nhất là 15.067% vào ngày 11/01/2022 và %Δ thấp nhất là 12.356% vào ngày 28/12/2021 với %Δ trung bình là 13.706%. Số ca nhiễm dự đoán thấp hơn so với số ca nhiễm thực tế có thể đến từ việc dữ liệu đầu vào có sự biến động mạnh vào tháng 4/2021 dẫn đến mô hình thuật toán cho ra kết quả không chính xác.

**Nhận xét và đánh giá:** Kết quả dự đoán có %Δ khá lớn từ 12% đến 15% trong khoảng thời gian từ ngày 28/12/2021 đến ngày 11/01/2022 cho thấy mô hình thuật toán không được chính xác như mong đợi. Kết quả dự đoán chỉ mang tính chất tham khảo để Việt Nam có những bước chuẩn bị trước với số ca nhiễm trong tương lai, tuy nhiên việc bùng phát dịch vẫn có thể xảy ra nên mọi người hãy đề cao sự tự giác, tự bảo vệ bản thân và tuân thủ các biện pháp phòng dịch theo thông điệp “5K”.

# KẾT LUẬN

## Kết luận

Dịch COVID-19 đang diễn ra một cách phức tạp và khó lường ở khắp mọi nơi trên Thế giới, Châu Âu và Châu Á là 2 châu lục bị ảnh hưởng nhiều nhất cùng với số ca nhiễm và số ca tử vong rất cao. Tiếp đến là khu vực Bắc Mỹ cũng có số ca nhiễm và tử vong cao, tuy nhiên Mỹ là quốc gia chiếm tỷ trọng cao nhất về số ca nhiễm (chiếm hơn 80%) và số ca tử vong (chiếm gần 70%). Bên cạnh số ca nhiễm và tử vong cao thì số ca hồi phục cũng là 1 tín hiệu tốt khi đã có gần 90% số người mắc bệnh đã hồi phục. Tuy số ca nhiễm cao nhưng tỷ lệ tử vong của dịch COVID-19 so với các đại dịch khác là nhỏ hơn nhiều, khi mà tỷ lệ tử vong chỉ 1.78% (so với dịch SARS-2003 có tỷ lệ tử vong 10%, dịch MERS-2012 có tỷ lệ tử vong 34.4%, dịch Ebola có tỷ lệ tử vong 43.9%). Sự nguy hiểm của dịch COVID-19 đó là tốc độ lây lan rất nhanh và sức khỏe của người bệnh chuyển biến xấu rất nhanh khiến cho cả Thế giới lo sợ và hoang mang, tuy nhiên cùng với sự phát triển của Vắc-xin COVID-19 đã đem đến sự an tâm, hạn chế việc lây lan cũng như giảm nhẹ các triệu chứng khi mắc bệnh và giúp cho Thế giới trở lại được trạng thái “Bình thường mới”.

Việt Nam cũng là quốc gia bị ảnh hưởng không ít bởi đại dịch COVID-19 khi số ca nhiễm và cả tử vong ngày một tăng và đã trải qua nhiều đợt giãn cách xã hội nhằm truy vết và ngăn chặn sự lây lan của dịch. Ở khu vực châu Á, Việt Nam thuộc top 10 quốc gia bị ảnh hưởng chỉ đứng sau Ấn Độ, Iran, Nhật Bản … và ở khu vực Đông Nam Á, Việt Nam thuộc top 5 quốc gia có số ca nhiễm và tử vong cao trong khu vực. Bên cạnh đó, Việt Nam đang thực hiện rất tốt về quy định phòng chống dịch, cũng như khả năng ứng phó và điều trị nhanh của các lực lượng tuyến đầu.

Để có được trạng thái “Bình thường mới” như hiện nay, Vắc-xin đóng vai trò rất quan trọng. Vắc-xin đã được triển khai và sử dụng bởi rất nhiều quốc gia trên Thế giới, trong đó có Việt Nam. Tỷ lệ tiêm ngừa (1 mũi và 2 mũi) của các quốc gia tình tới thời điểm hiện tại là rất cao (từ 70-150% dân số). Việt Nam đang triển khai việc tiêm Vắc-xin nhanh khi đã có hơn 130% dân số đã được tiêm ngừa so với khu vực châu Á nói chung và Đông Nam Á nói riêng.

* Ưu điểm:
* Đồ án này giúp tôi có thêm nhiều kiến thức về xử lý dữ liệu, thống kê và trau dồi kỹ năng vẽ biểu đồ bằng phần mềm Power Bi và sử dụng các thư viện Python để trực quan hóa.
* Đồ án này giúp tôi có cái nhìn rõ hơn về đại dịch COVID-19 đang diễn ra về mặt tổng quát và chi tiết, so sánh được Việt Nam với các quốc gia ở Châu Á và khu vực Đông Nam Á, theo dõi được tình trạng tiêm chủng Vắc-xin COVID-19 của Thế giới nói chung, giữa Việt Nam và Châu Á, Đông Nam Á nói riêng.
* Nhờ vào các thuật toán Machine Learning, tôi có thể xây dựng được mô hình dự đoán về số ca nhiễm trong thời gian gần của Thế giới và Việt Nam.
* Nhược điểm:
* Kiến thức và kĩ năng phân tích dữ liệu của tôi còn hạn chế nên vẫn chưa thể khai thác hết tất cả những thông tin cần thiết.
* Do số lượng dữ liệu đầu vào không đủ lớn, dẫn đến sai số khiến cho kết quả dự đoán không được chính xác như mong đợi.

## Hướng phát triển

Trong đề tài này, hướng phát triển của tôi là:

* Thu thập thông tin theo hướng thời gian thực (real-time) để dữ liệu có thể cập nhật sớm nhất có thể.
* Sử dụng ngôn ngữ Python để tạo ra các báo cáo bằng file pdf.
* Gửi email cho những người đã được cài đặt sẵn thông qua giao thức SMTP.
* Sử dụng Power BI để tạo ra các Dashboard tự động cập nhật.
* Triển khai các báo cáo bằng các nền tảng Web/App cho Desktop và Mobile.

# TÀI LIỆU THAM KHẢO

**Tiếng Việt**

Vũ Hữu Tiệp (2018), *Machine Learning cơ bản.*

**Tiếng Anh**

D. J. C. MacKay(1992),*Bayesian Interpolation, Computation and Neural Systems,* Vol. 4, No. 3**.**

M. E. Tipping (2001), *Sparse Bayesian Learning and the Relevance Vector Machine,* Journal of Machine Learning Research, Vol. 1

Platt, John (1999), *Probabilistic outputs for support vector machines and comparison to regularizedlikelihood methods.*

Chih-Chung Chang and Chih-Jen Lin (2001), *A Library for Support Vector Machines.*

Andreas C. Müller and Sarah Guido (2016), *Introduction to Machine Learning with Python: A Guide for Data Scientists,* O'Reilly Media*.*

O Theobald (2017), *Machine Learning For Absolute Beginners: A Plain English Introduction (Second Edition) (Machine Learning From Scratch Book 1)*

O Theobald (2019), *Machine Learning with Python: A Practical Beginners’ Guide (Machine Learning From Scratch Book 2).*

O Theobald (2017), *Data Analytics for Absolute Beginners: Make Decisions Using Every Variable: (Introduction to Data, Data Visualization, Business Intelligence & Machine Learning).*

Brady Ellison (2021), *Python for Data Analysis: A Beginners Guide to Master the Fundamentals of Data Science and Data Analysis by Using Pandas, Numpy and Ipython.*

Wes McKinney (2012), *Python for Data Analysis: A Beginners Guide to Master the Fundamentals of Data Science and Data Analysis by Using Pandas, Numpy and Ipython.*

Cole Nussbaumer Knaflic (2015), *Storytelling with Data: A Data Visualization Guide for Business Professionals*, Wiley; 1st edition.

Ian Miell and Aidan Hobson Sayers (2016), *Docker in Practice 1st Edition*, Manning.

Mark Reed (2020), Docker: *The Ultimate Beginners Guide to Learn Docker Step-by-Step (Computer Programming),* Publishing Factory.

Christian Leornardo (2020), *Docker: Docker for the Absolute Beginner.*

Bas P. Harenslak and Julian Rutger de Ruiter (2021), *Data Pipelines with Apache Airflow,* Manning.

Dan Clark (2017), *Beginning Power BI: A Practical Guide to Self-Service Data Analytics with Excel 2016 and Power BI Desktop,* Apress

1. MÃ NGUỒN PYTHON

**Code lấy dữ liệu tổng hợp:**

import requests

from datetime import date

from connect\_mysql import query

import kaggle

import os

import pandas as pd

import numpy as np

WORKING\_PATH = os.path.dirname(os.path.dirname(os.path.realpath(\_\_file\_\_)))

# use rapid api to get info for all countries

def get\_summary\_data():

params = ['asia', 'africa', 'europe',

'northamerica', 'southamerica', 'australia']

sum\_data = []

for param in params:

url = f"https://vaccovid-coronavirus-vaccine-and-treatment-tracker.p.rapidapi.com/api/npm-covid-data/{param}"

headers = {

'x-rapidapi-host': "vaccovid-coronavirus-vaccine-and-treatment-tracker.p.rapidapi.com",

'x-rapidapi-key': "282e773d97msh20602f20182196ep1bbea3jsn8b10ac27c324"}

response = requests.request("GET", url, headers=headers).json()

for country in response:

rank = country['rank']

name = country['Country']

continent = country['Continent']

infection\_risk = country['Infection\_Risk']

serious\_critical = country['Serious\_Critical']

active\_cases = country['ActiveCases']

fatality\_rate = country['Case\_Fatality\_Rate']

test\_percentage = country['Test\_Percentage']

recovery\_proporation = country['Recovery\_Proporation']

total\_cases = country['TotalCases']

new\_cases = country['NewCases']

total\_deaths = country['TotalDeaths']

new\_deaths = country['NewDeaths']

total\_recovered = country['TotalRecovered']

total\_tests = country['TotalTests']

population = country['Population']

run\_date = date.today().strftime('%Y-%m-%d')

info = (rank, name, continent, infection\_risk, serious\_critical, fatality\_rate, active\_cases, total\_cases,

new\_cases, total\_deaths, new\_deaths, total\_recovered, recovery\_proporation, total\_tests, test\_percentage, population, run\_date)

sum\_data.append(info)

print('Crawl summary data successfully')

query(keyw='Truncate', table\_name='SUMMARY\_COVID\_DATA')

query(keyw='Insert', table\_name='SUMMARY\_COVID\_DATA', num\_cols=17, data\_arr=sum\_data)

def get\_vaccine\_data():

try:

kaggle.api.authenticate()

kaggle.api.dataset\_download\_files(

'gpreda/covid-world-vaccination-progress', path=f'{WORKING\_PATH}/data', unzip=True)

print('Download files successfully')

file\_path = fr'{WORKING\_PATH}\data\country\_vaccinations.csv'

df = pd.read\_csv(file\_path)

use\_cols = ['country', 'date', 'total\_vaccinations', 'people\_vaccinated',

'people\_fully\_vaccinated', 'daily\_vaccinations', 'vaccines']

final\_df = df[use\_cols]

final\_df = final\_df.replace({np.NAN: None})

final\_df['run\_date'] = date.today().strftime('%Y-%m-%d')

except Exception as err:

print(err)

def get\_full\_vaccines():

df\_with\_vaccine = pd.read\_csv(

rf'{WORKING\_PATH}\data\country\_vaccinations.csv', usecols=['country', 'vaccines'])

df\_with\_vaccine = df\_with\_vaccine.groupby(by='country').max()

df\_no\_vaccine = pd.read\_csv(

'https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations.csv')

df\_no\_vaccine['vaccines'] = ''

df\_no\_vaccine = df\_no\_vaccine.set\_index('location')

### update 'vaccines' column from df\_with\_vaccine to df\_no\_vaccine

list\_country = df\_with\_vaccine.index

for country in list\_country:

vaccine = df\_with\_vaccine.loc[country, 'vaccines']

df\_no\_vaccine.loc[country, 'vaccines'] = vaccine

### reformat the vaccine df before update to dB

df\_no\_vaccine = df\_no\_vaccine.reset\_index()

final\_df\_vaccine = df\_no\_vaccine[df\_no\_vaccine['vaccines'] != '']

final\_df\_vaccine.rename(columns={'location': 'country'}, inplace=True)

use\_cols = ['country', 'iso\_code', 'date', 'total\_vaccinations',

'people\_vaccinated', 'people\_fully\_vaccinated', 'daily\_vaccinations',

'total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred',

'people\_fully\_vaccinated\_per\_hundred', 'daily\_vaccinations\_per\_million', 'vaccines']

final\_df\_vaccine = final\_df\_vaccine[use\_cols]

final\_df\_vaccine = final\_df\_vaccine.replace({np.NaN: None})

# update daily vaccine data

vaccine\_data = [tuple(row) for row in final\_df\_vaccine.values]

query(keyw='Truncate', table\_name='VACCINE\_COVID\_DATA')

query(keyw='Insert', table\_name='VACCINE\_COVID\_DATA', num\_cols=12, data\_arr=vaccine\_data)

final\_df\_vaccine.to\_csv(rf'{WORKING\_PATH}\data\full\_vaccinations.csv', index=False)

def main():

get\_summary\_data()

get\_vaccine\_data()

get\_full\_vaccines()

main()

**Code vẽ biểu đồ chi tiết:**

import numpy as np

import matplotlib.pyplot as plt

import plotly.express as px

import os

# for offline ploting

from plotly.offline import plot

import pandas as pd

from sklearn.linear\_model import LinearRegression, BayesianRidge

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

from sklearn.preprocessing import PolynomialFeatures

from sklearn.svm import SVR

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error

import datetime

import operator

from task\_scheduler.connect\_mysql import query

plt.style.use('seaborn-poster')

%matplotlib inline

from IPython.display import set\_matplotlib\_formats

set\_matplotlib\_formats('retina')

import warnings

warnings.filterwarnings('ignore')

WORKING\_PATH = os.getcwd()

from IPython.display import Image

Image(filename = rf'{WORKING\_PATH}\data\covid.jpg')

PLOT\_SIZE = (16, 10)

confirmed\_df = pd.read\_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_confirmed\_global.csv')

deaths\_df = pd.read\_csv('https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_time\_series/time\_series\_covid19\_deaths\_global.csv')

updated\_date = (datetime.datetime.today()-datetime.timedelta(days=2)).strftime('%m-%d-%Y')

latest\_data = pd.read\_csv(f'https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse\_covid\_19\_data/csse\_covid\_19\_daily\_reports/{updated\_date}.csv')

sum\_country = query(keyw='Read', table\_name='SUMMARY\_COVID\_DATA')[['NAME', 'CONTINENT', 'TOTAL\_CASES', 'TOTAL\_DEATHS', 'TOTAL\_RECOVERED', 'ACTIVE\_CASES', 'SERIOUS\_CRITICAL', 'FATALITY\_RATE']] ### query from database and get specified columns

df\_continents\_cases = sum\_country[['CONTINENT', 'TOTAL\_CASES', 'TOTAL\_DEATHS', 'TOTAL\_RECOVERED', 'ACTIVE\_CASES', 'SERIOUS\_CRITICAL']]

df\_continents\_cases = df\_continents\_cases.groupby(by='CONTINENT').sum()

### gradient table of continents about confirmed, deaths, ...

df\_continents\_cases['FATALITY\_RATE (per 100)'] = np.round(100\*df\_continents\_cases['TOTAL\_DEATHS']/df\_continents\_cases['TOTAL\_CASES'],2)

df\_continents\_cases.style.background\_gradient(cmap='Blues',subset=['TOTAL\_CASES'])\

.background\_gradient(cmap='Reds',subset=['TOTAL\_DEATHS'])\

.background\_gradient(cmap='Greens',subset=['TOTAL\_RECOVERED'])\

.background\_gradient(cmap='Purples',subset=['ACTIVE\_CASES'])\

.background\_gradient(cmap='Pastel1\_r',subset=['SERIOUS\_CRITICAL'])\

.background\_gradient(cmap='YlOrBr',subset=['FATALITY\_RATE (per 100)'])\

.format('{:.2f}')\

.format('{:.0f}',subset=['TOTAL\_CASES','TOTAL\_DEATHS','TOTAL\_RECOVERED','ACTIVE\_CASES', 'SERIOUS\_CRITICAL'])

### top 30 country with gradient table

sum\_country.drop(['CONTINENT'], axis=1).sort\_values('TOTAL\_CASES', ascending= False).head(30).set\_index(['NAME']).style.background\_gradient(cmap='Blues',subset=['TOTAL\_CASES'])\

.background\_gradient(cmap='Reds',subset=['TOTAL\_DEATHS'])\ .background\_gradient(cmap='Greens',subset=['TOTAL\_RECOVERED'])\

.background\_gradient(cmap='Purples',subset=['ACTIVE\_CASES'])\ .background\_gradient(cmap='Pastel1\_r',subset=['SERIOUS\_CRITICAL'])\ .background\_gradient(cmap='YlOrBr',subset=['FATALITY\_RATE'])\

.format('{:.2f}')\ .format('{:.0f}',subset=['TOTAL\_CASES','TOTAL\_DEATHS','TOTAL\_RECOVERED','ACTIVE\_CASES', 'SERIOUS\_CRITICAL'])

### create a plot map to show about the confirmed, deaths and recovered status

def plot\_map(df, col):

df = df[df[col]>0]

fig = px.choropleth(df, locations='NAME', locationmode='country names',

color=col, hover\_name='NAME',

title=col, hover\_data=[col], color\_continuous\_scale='matter')

fig.show()

### scatter plot with for top 20 country with total deaths and total cases

fig = px.scatter(sum\_country.sort\_values('TOTAL\_DEATHS', ascending=False).iloc[:20, :],

x='TOTAL\_CASES', y='TOTAL\_DEATHS', color='NAME', size='TOTAL\_CASES',

height=700, text='NAME', log\_x=True, log\_y=True,

title='Deaths vs Confirmed')

fig.update\_traces(textposition='top center')

fig.update\_layout(showlegend=False)

fig.update\_layout(xaxis\_rangeslider\_visible=True)

fig.show()

cols = confirmed\_df.keys()

### get data from 22/1/2021 to updated\_date

confirmed = confirmed\_df.loc[:, cols[4]:cols[-1]]

deaths = deaths\_df.loc[:, cols[4]:cols[-1]]

dates = confirmed.keys() ### get list of date

total\_cases = []

total\_deaths = []

mortality\_rate = []

for day in dates:

confirmed\_sum = confirmed[day].sum()

death\_sum = deaths[day].sum()

# confirmed, deaths

total\_cases.append(confirmed\_sum) ### total cases day by day

total\_deaths.append(death\_sum) ### total deaths day by day

# calculate mortality rates

mortality\_rate.append(death\_sum/confirmed\_sum) ### mortality rates day by day

### get the list of daily increase

def daily\_increase(data):

d = []

for i in range(len(data)):

if i == 0:

d.append(data[0])

else:

d.append(data[i]-data[i-1])

return d

### list of moving average

def moving\_average(data, window\_size):

moving\_average = []

for i in range(len(data)):

if i + window\_size < len(data):

moving\_average.append(np.mean(data[i:i+window\_size]))

else:

moving\_average.append(np.mean(data[i:len(data)]))

return moving\_average

# window size

window = 7

# confirmed cases

world\_daily\_increase = daily\_increase(total\_cases)

world\_confirmed\_avg = moving\_average(total\_cases, window)

world\_daily\_increase\_avg = moving\_average(world\_daily\_increase, window)

# deaths

world\_daily\_death = daily\_increase(total\_deaths)

world\_death\_avg = moving\_average(total\_deaths, window)

world\_daily\_death\_avg = moving\_average(world\_daily\_death, window)

### since 22/1/2020

num\_of\_days = np.array([i for i in range(len(dates))])

total\_cases = np.array(total\_cases).reshape(-1, 1)

total\_deaths = np.array(total\_deaths).reshape(-1, 1)

### line plot for world cases and moving average

plt.figure(figsize=PLOT\_SIZE)

plt.plot(num\_of\_days, total\_cases, linewidth=4.5)

plt.plot(num\_of\_days, world\_confirmed\_avg, linestyle='dashed', color='red', linewidth=4.5)

plt.title('Total of Coronavirus Cases Over Time', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(['Worldwide Coronavirus Cases', 'Moving Average {} Days'.format(window)], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

### line plot for death cases and moving average

plt.figure(figsize=PLOT\_SIZE)

plt.plot(num\_of\_days, total\_deaths, linewidth=4.5)

plt.plot(num\_of\_days, world\_death\_avg, linestyle='dashed', color='red', linewidth=4.5)

plt.title('Total of Coronavirus Deaths Over Time', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(['Worldwide Coronavirus Deaths', 'Moving Average {} Days'.format(window)], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

### bar plot for daily world cases and line plot moving average

plt.figure(figsize=PLOT\_SIZE)

plt.bar(num\_of\_days, world\_daily\_increase)

plt.plot(num\_of\_days, world\_daily\_increase\_avg, color='red', linestyle='dashed', linewidth=4.5)

plt.title('World Daily Increases in Confirmed Cases', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(['Moving Average {} Days'.format(window), 'World Daily Increase in COVID-19 Cases'], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

### bar plot for daily world cases and line plot moving average

plt.figure(figsize=PLOT\_SIZE)

plt.bar(num\_of\_days, world\_daily\_death)

plt.plot(num\_of\_days, world\_daily\_death\_avg, color='red', linestyle='dashed')

plt.title('World Daily Increases in Confirmed Deaths', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(['Moving Average {} Days'.format(window), 'World Daily Increase in COVID-19 Deaths'], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

unique\_countries = list(latest\_data['Country\_Region'].unique()) ### get list of unique country

country\_confirmed\_cases = []

country\_death\_cases = []

country\_active\_cases = []

country\_incidence\_rate = []

country\_mortality\_rate = []

no\_cases = []

for i in unique\_countries:

cases = latest\_data[latest\_data['Country\_Region']==i]['Confirmed'].sum()

if cases > 0:

country\_confirmed\_cases.append(cases) ### list country has cases

else:

no\_cases.append(i) ### list country has no cases

for i in no\_cases:

unique\_countries.remove(i) ### final list country has cases

# sort countries by the number of confirmed cases

unique\_countries = [k for k, v in sorted(zip(unique\_countries, country\_confirmed\_cases), key=operator.itemgetter(1), reverse=True)]

for i in range(len(unique\_countries)):

country\_confirmed\_cases[i] = latest\_data[latest\_data['Country\_Region']==unique\_countries[i]]['Confirmed'].sum() country\_death\_cases.append(latest\_data[latest\_data['Country\_Region']==unique\_countries[i]]['Deaths'].sum()) country\_incidence\_rate.append(latest\_data[latest\_data['Country\_Region']==unique\_countries[i]]['Incident\_Rate'].sum()) country\_mortality\_rate.append(country\_death\_cases[i]/country\_confirmed\_cases[i])

### bar chart to compare confirmed cases and percentage with others

total\_world\_cases = np.sum(country\_confirmed\_cases)

us\_confirmed = latest\_data[latest\_data['Country\_Region']=='US']['Confirmed'].sum()

outside\_us\_confirmed = total\_world\_cases - us\_confirmed

plt.figure(figsize=(16, 9))

plt.barh('United States', us\_confirmed)

plt.barh('Outside United States', outside\_us\_confirmed)

plt.title('# of Total Coronavirus Confirmed Cases', size=20)

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

plt.figure(figsize=(16, 9))

plt.barh('United States', us\_confirmed/total\_world\_cases)

plt.barh('Outside United States', outside\_us\_confirmed/total\_world\_cases)

plt.title('# of Coronavirus Confirmed Cases Expressed in Percentage', size=20)

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

# Only show 10 provinces with the most confirmed cases, the rest are grouped into the other category

visual\_unique\_provinces = []

visual\_confirmed\_cases2 = []

others = np.sum(province\_confirmed\_cases[10:])

for i in range(len(province\_confirmed\_cases[:10])):

visual\_unique\_provinces.append(unique\_provinces[i])

visual\_confirmed\_cases2.append(province\_confirmed\_cases[i])

visual\_unique\_provinces.append('Others')

visual\_confirmed\_cases2.append(others)

total\_cases\_asia = []

total\_deaths\_asia = []

mortality\_rate\_asia = []

for day in dates:

confirmed\_sum\_asia = confirmed\_df\_asia[day].sum()

deaths\_sum\_asia = deaths\_df\_asia[day].sum()

# confirmed, deaths

total\_cases\_asia.append(confirmed\_sum\_asia) ### total cases day by day

total\_deaths\_asia.append(deaths\_sum\_asia) ### total deaths day by day

# calculate mortality rates

mortality\_rate\_asia.append(deaths\_sum\_asia/confirmed\_sum\_asia) ### mortality rates day by day

# asia confirmed cases

asia\_daily\_increase = daily\_increase(total\_cases\_asia)

asia\_confirmed\_avg = moving\_average(total\_cases\_asia, window)

asia\_daily\_increase\_avg = moving\_average(asia\_daily\_increase, window)

# asia deaths cases

asia\_daily\_death = daily\_increase(total\_deaths\_asia)

asia\_death\_avg = moving\_average(total\_deaths\_asia, window)

asia\_daily\_death\_avg = moving\_average(asia\_daily\_death, window)

# vietnam confirmed cases

vietnam\_daily\_increase = daily\_increase(total\_cases\_vietnam)

vietnam\_confirmed\_avg = moving\_average(total\_cases\_vietnam, window)

vietnam\_daily\_increase\_avg = moving\_average(vietnam\_daily\_increase, window)

# vietnam deaths cases

vietnam\_daily\_death = daily\_increase(total\_deaths\_vietnam)

vietnam\_death\_avg = moving\_average(total\_deaths\_vietnam, window)

vietnam\_daily\_death\_avg = moving\_average(vietnam\_daily\_death, window)

colors = ['red', 'green', 'purple', 'brown', 'gray', 'black']

plt.figure(figsize=PLOT\_SIZE)

for item, color in zip(dict\_top6\_asia.values(), colors):

plt.plot(num\_of\_days, item[0], color=color, linewidth=4.5)

plt.plot(num\_of\_days, total\_cases\_vietnam, linewidth=4.5)

plt.title('Vietnam and Top 6 Country with Confirmed Cases', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(top6\_country\_asia+['Vietnam'], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

colors = ['red', 'green', 'purple', 'brown', 'gray', 'black']

plt.figure(figsize=PLOT\_SIZE)

for item, color in zip(dict\_top6\_asia.values(), colors):

plt.plot(num\_of\_days, item[1], color=color, linewidth=4.5)

plt.plot(num\_of\_days, total\_deaths\_vietnam, linewidth=4.5)

plt.title('Vietnam and Top 6 Country with Deaths Cases', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(top6\_country\_asia+['Vietnam'], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

colors = ['red', 'green', 'purple', 'brown', 'gray', 'black']

plt.figure(figsize=PLOT\_SIZE)

for item, color in zip(dict\_top6\_asia.values(), colors):

plt.plot(num\_of\_days, item[2], color=color, linewidth=4.5)

plt.plot(num\_of\_days, mortality\_rate\_vietnam, linewidth=4.5)

plt.title('Vietnam and Top 6 Country with Mortality Rate', size=30)

plt.xlabel('Days Since 22/1/2020', size=30)

plt.ylabel('# of Cases', size=30)

plt.legend(top6\_country\_asia+['Vietnam'], prop={'size': 15})

plt.xticks(size=20)

plt.yticks(size=20)

plt.show()

days\_in\_future = 15

future\_forcast = np.array([i for i in range(len(dates)+days\_in\_future)]).reshape(-1, 1)

adjusted\_dates = future\_forcast[:-days\_in\_future]

start = '1/22/2020'

start\_date = datetime.datetime.strptime(start, '%m/%d/%Y')

future\_forcast\_dates = []

for i in range(len(future\_forcast)):

future\_forcast\_dates.append((start\_date + datetime.timedelta(days=i)).strftime('%m/%d/%Y'))

days\_since\_1\_22 = np.array([i for i in range(len(dates))]).reshape(-1, 1)

X\_train\_confirmed, X\_test\_confirmed, y\_train\_confirmed, y\_test\_confirmed = train\_test\_split(days\_since\_1\_22, total\_cases, test\_size=0.08, shuffle=False)

### train SVR model and fit it to predict

svm\_confirmed = SVR(shrinking=True, kernel='poly', gamma=0.01, epsilon=1, degree=2, C=0.1)

svm\_confirmed.fit(X\_train\_confirmed, y\_train\_confirmed)

test\_svm\_pred = svm\_confirmed.predict(X\_test\_confirmed)

svm\_pred = svm\_confirmed.predict(future\_forcast)

r\_sq = svm\_confirmed.score(X\_train\_confirmed, y\_train\_confirmed)

print('MAE:', mean\_absolute\_error(test\_svm\_pred, y\_test\_confirmed))

print('MSE:',mean\_squared\_error(test\_svm\_pred, y\_test\_confirmed))

print('coefficient of determination:', r\_sq)

print('intercept:', svm\_confirmed.intercept\_)

# transform our data for polynomial regression with degree=3

poly = PolynomialFeatures(degree=3)

poly\_X\_train\_confirmed = poly.fit\_transform(X\_train\_confirmed)

poly\_X\_test\_confirmed = poly.fit\_transform(X\_test\_confirmed)

poly\_future\_forcast = poly.fit\_transform(future\_forcast)

# linear polynomial regression

linear\_model = LinearRegression(normalize=True, fit\_intercept=False)

linear\_model.fit(poly\_X\_train\_confirmed, y\_train\_confirmed)

test\_linear\_pred = linear\_model.predict(poly\_X\_test\_confirmed)

linear\_pred = linear\_model.predict(poly\_future\_forcast)

r\_sq = linear\_model.score(poly\_X\_train\_confirmed, y\_train\_confirmed)

print('MAE:', mean\_absolute\_error(test\_linear\_pred, y\_test\_confirmed))

print('MSE:',mean\_squared\_error(test\_linear\_pred, y\_test\_confirmed))

print('coefficient of determination:', r\_sq)

print('intercept:', linear\_model.intercept\_)

print('coefficients:', linear\_model.coef\_)

# bayesian ridge polynomial regression

tol = [1e-6, 1e-5, 1e-4, 1e-3, 1e-2]

alpha\_1 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]

alpha\_2 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]

lambda\_1 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]

lambda\_2 = [1e-7, 1e-6, 1e-5, 1e-4, 1e-3]

normalize = [True, False]

bayesian\_grid = {'tol': tol, 'alpha\_1': alpha\_1, 'alpha\_2' : alpha\_2, 'lambda\_1': lambda\_1, 'lambda\_2' : lambda\_2,

'normalize' : normalize}

bayesian = BayesianRidge(fit\_intercept=False)

bayesian\_search = RandomizedSearchCV(bayesian, bayesian\_grid, scoring='neg\_mean\_squared\_error', cv=3, return\_train\_score=True, n\_jobs=-1, n\_iter=40, verbose=1)

bayesian\_search.fit(poly\_X\_train\_confirmed, y\_train\_confirmed)

bayesian\_confirmed = bayesian\_search.best\_estimator\_

test\_bayesian\_pred = bayesian\_confirmed.predict(poly\_X\_test\_confirmed)

bayesian\_pred = bayesian\_confirmed.predict(poly\_future\_forcast)

r\_sq = bayesian\_confirmed.score(poly\_X\_train\_confirmed, y\_train\_confirmed)

print('MAE:', mean\_absolute\_error(test\_bayesian\_pred, y\_test\_confirmed))

print('MSE:',mean\_squared\_error(test\_bayesian\_pred, y\_test\_confirmed))

print('coefficient of determination:', r\_sq)

print('intercept:', bayesian\_confirmed.intercept\_)

print('coefficients:', bayesian\_confirmed.coef\_)

**Code vẽ biểu đồ Vắc-xin:**

import os

import warnings

import numpy as np

import pandas as pd

from datetime import timedelta

import PIL

from IPython.display import Image

import matplotlib.dates as mdates

import matplotlib.pyplot as plt

import matplotlib.ticker as ticker

from matplotlib.patches import Patch

from matplotlib.offsetbox import OffsetImage,AnnotationBbox

from matplotlib.animation import FuncAnimation

from IPython.display import HTML

import plotly.graph\_objs as go

from plotly.offline import iplot

import plotly.express as px

import seaborn as sns

import textwrap

import pycountry

import urllib

from task\_scheduler.connect\_mysql import query

warnings.filterwarnings('ignore')

WORKING\_PATH = os.getcwd()

Image(filename = rf'{WORKING\_PATH}\data\vaccine.jpg')

df\_vaccine = pd.read\_csv(rf'{WORKING\_PATH}\data\full\_vaccinations.csv')

df\_vaccine['date'] = pd.to\_datetime(df\_vaccine['date'])

countries = list(df\_vaccine['country'].unique()) ### list of unique country

vaccines\_name = list(df\_vaccine['vaccines'].unique())

### create a dataframe to get all the lastest data of each unique country

data\_lastest = pd.DataFrame()

for country in countries:

temp = df\_vaccine[df\_vaccine['country']==country].reset\_index(drop=True)

temp = temp.iloc[-1, :]

data\_lastest = data\_lastest.append(temp, ignore\_index=True)

### create a overview board about the vaccine status

plt.rcParams['figure.dpi'] = 600

fig = plt.figure(figsize=(5, 0.8), facecolor='#f6f5f5')

gs = fig.add\_gridspec(1, 1)

gs.update(wspace=0, hspace=0)

background\_color = '#f6f5f5'

ax0 = fig.add\_subplot(gs[0, 0])

ax0.set\_facecolor(background\_color)

for s in ['top','right', 'left', 'bottom']:

ax0.spines[s].set\_visible(False)

ax0.set\_xticks([])

ax0.set\_yticks([])

total\_vaccine = round(data\_lastest['total\_vaccinations'].sum()/1000000)

people\_vaccined = round(data\_lastest['people\_vaccinated'].sum()/1000000)

number\_vaccines = len(set(data\_lastest['vaccines'].to\_list()))

lst\_vaccines = set([y.strip() for i in data\_lastest['vaccines'].to\_list() for y in i.split(',')])

people\_fully\_vaccined = round(data\_lastest['people\_fully\_vaccinated'].sum()/1000000)

ax0.grid(which='major', axis='y', zorder=0, color='#EEEEEE')

ax0.text(-0.12, 0.8, 'General Overview', color='black', fontsize=7, ha='left', weight='bold', va='bottom')

ax0.text(-0.12, 0.79, 'A quick glance of world vaccination progress', color='#292929', fontsize=5, ha='left', va='top')

ax0.text(0, 0, len(countries), color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0, 0, 'country with\nVaccination Program', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.2, 0, len(lst\_vaccines), color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.2, 0, 'vaccines\nused', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.4, 0, total\_vaccine, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.4, 0, 'million of\ntotal vaccinations', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.65, 0, people\_vaccined, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.65, 0, 'million of\npeople vaccinated', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.9, 0, people\_fully\_vaccined, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.9, 0, 'million of people\nfully vaccinated', color='dimgray', fontsize=6, ha='center', va='top', weight='bold')

plt.show()

### create bar chart to show vaccine information of top 10 country

def draw\_trace\_bar(data, feature, title, xlab, ylab, color):

data = data.sort\_values(feature, ascending=False).head(10)

trace = go.Bar(

x = data['Country'],

y = data[feature],

marker=dict(color=color),

text=data['Country']

)

data = [trace]

layout = dict(title = title,

xaxis = dict(title = xlab, showticklabels=True, tickangle=45,

zeroline=True, zerolinewidth=1, zerolinecolor='grey',

showline=True, linewidth=2, linecolor='black', mirror=True,

tickfont=dict(

size=10,

color='black'),),

yaxis = dict(title = ylab, gridcolor='lightgrey', zeroline=True, zerolinewidth=1, zerolinecolor='grey',

showline=True, linewidth=2, linecolor='black', mirror=True),

plot\_bgcolor = 'white', paper\_bgcolor = 'white',

hovermode = 'closest'

)

fig = dict(data = data, layout = layout)

iplot(fig, filename='draw\_trace')

### get daily vaccine day by day

daily\_vaccination = df\_vaccine.groupby('date')['daily\_vaccinations'].sum() / 1000000

background\_color = '#f6f5f5'

years = mdates.YearLocator()

months = mdates.MonthLocator()

years\_fmt = mdates.DateFormatter('%m-%y')

### boxenplot to show daily vaccination on weekday

daily\_vaccination = pd.DataFrame(daily\_vaccination).reset\_index(drop=False)

daily\_vaccination['weekday'] = daily\_vaccination['date'].dt.dayofweek

weekday = {0: 'Mon', 1:'Tue', 2:'Wed', 3:'Thu', 4:'Fri', 5:'Sat', 6:'Sun'}

daily\_vaccination['weekday'] = daily\_vaccination['weekday'].map(weekday)

background\_color = '#f6f5f5'

color\_map = ['#2693d1' for \_ in range(7)]

sns.set\_palette(sns.color\_palette(color\_map))

plt.rcParams['figure.dpi'] = 600

fig = plt.figure(figsize=(6, 3), facecolor='#f6f5f5')

gs = fig.add\_gridspec(1, 1)

gs.update(wspace=0, hspace=0)

ax0 = fig.add\_subplot(gs[0, 0])

ax0.set\_facecolor(background\_color)

for s in ['right', 'top']:

ax0.spines[s].set\_visible(False)

ax0.text(-0.5, peak\_vaccination+4, 'World Weekday Daily Vaccination Distribution', color='black', fontsize=7, ha='left', va='bottom', weight='bold')

ax0.text(-0.5, peak\_vaccination+4, 'Daily vaccination distribution from Monday to Friday are about the same',

color='#292929', fontsize=5, ha='left', va='top')

ax0\_sns = sns.boxenplot(ax=ax0, x=daily\_vaccination['weekday'], y=daily\_vaccination['daily\_vaccinations'],

zorder=3, linewidth=1, saturation=1)

ax0\_sns.set\_xlabel('Day', fontsize=5, weight='bold')

ax0\_sns.set\_ylabel('Vaccinations (in million)', fontsize=5, weight='bold')

ax0\_sns.tick\_params(labelsize=5)

plt.show()

comb\_vaccine = data\_lastest.groupby('vaccines')['people\_fully\_vaccinated'].sum().sort\_values(ascending=False)

comb\_vaccine = comb\_vaccine/1000000

comb\_vaccine = comb\_vaccine[:5]

top\_vaccine = comb\_vaccine.head(1).index[0]

background\_color = '#f6f5f5'

color\_map = ['lightgray' for \_ in range(26)]

color\_map[0] = '#2693d1'

sns.set\_palette(sns.color\_palette(color\_map))

plt.rcParams['figure.dpi'] = 600

fig = plt.figure(figsize=(5, 3), facecolor='#f6f5f5')

gs = fig.add\_gridspec(1, 1)

gs.update(wspace=0, hspace=0)

ax0 = fig.add\_subplot(gs[0, 0])

ax0.set\_facecolor(background\_color)

for s in ['right', 'top']:

ax0.spines[s].set\_visible(False)

ax0.tick\_params(axis = 'y', which = 'both', left = False)

ax0.text(0, -0.72, 'Top 5 Vaccines - Fully Vaccinated People', color='black', fontsize=7, ha='left', va='bottom', weight='bold')

ax0.text(0, -0.7, f'{top\_vaccine} is the most vaccine that has fully vaccinated people',

color='#292929', fontsize=5, ha='left', va='top')

ax0\_sns = sns.barplot(ax=ax0, y=comb\_vaccine.index, x=comb\_vaccine, zorder=2, orient='h',

linewidth=0.5, edgecolor='black', saturation=1)

ax0\_sns.set\_xlabel('People (in million)', fontsize=5, weight='bold')

ax0\_sns.set\_ylabel('Vaccine', fontsize=5, weight='bold')

ax0.grid(which='major', axis='x', zorder=0, color='#EEEEEE', lw=0.3)

ax0.grid(which='major', axis='y', zorder=0, color='#EEEEEE', lw=0.3)

ax0\_sns.tick\_params(labelsize=5)

ax0.set\_yticklabels([textwrap.fill(e, 35) for e in comb\_vaccine.index])

for p in ax0.patches:

value = f'{p.get\_width():,.1f}'

x = p.get\_x() + p.get\_width() + 20

y = p.get\_y() + p.get\_height() / 2

ax0.text(x, y, value, ha='center', va='center', fontsize=5,

bbox=dict(facecolor='none', edgecolor='black', boxstyle='round', linewidth=0.3))

plt.show()

### get daily vaccine day by day

daily\_vaccination\_asia = df\_vaccine\_asia.groupby(['country', 'date'], as\_index=False)['daily\_vaccinations'].sum()

daily\_vaccination\_asia['daily\_vaccinations'] = daily\_vaccination\_asia['daily\_vaccinations'] / 1000000

background\_color = '#f6f5f5'

years = mdates.YearLocator()

months = mdates.MonthLocator()

years\_fmt = mdates.DateFormatter('%m-%y')

### create a overview board about the vaccine status in SEA

plt.rcParams['figure.dpi'] = 600

fig = plt.figure(figsize=(5, 0.8), facecolor='#f6f5f5')

gs = fig.add\_gridspec(1, 1)

gs.update(wspace=0, hspace=0)

background\_color = '#f6f5f5'

ax0 = fig.add\_subplot(gs[0, 0])

ax0.set\_facecolor(background\_color)

for s in ['top','right', 'left', 'bottom']:

ax0.spines[s].set\_visible(False)

ax0.set\_xticks([])

ax0.set\_yticks([])

num\_country = len(data\_lastest\_sea['country'].values)

total\_vaccine = round(data\_lastest\_sea['total\_vaccinations'].sum()/1000000)

people\_vaccined = round(data\_lastest\_sea['people\_vaccinated'].sum()/1000000)

number\_vaccines = len(set(data\_lastest\_sea['vaccines'].to\_list()))

lst\_vaccines = set([y.strip() for i in data\_lastest\_sea['vaccines'].to\_list() for y in i.split(',')])

people\_fully\_vaccined = round(data\_lastest\_sea['people\_fully\_vaccinated'].sum()/1000000)

ax0.grid(which='major', axis='y', zorder=0, color='#EEEEEE')

ax0.text(-0.12, 0.8, 'General Overview', color='black', fontsize=7, ha='left', weight='bold', va='bottom')

ax0.text(-0.12, 0.79, 'A quick glance of SEA vaccination progress', color='#292929', fontsize=5, ha='left', va='top')

ax0.text(0, 0, num\_country, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0, 0, 'country with\nVaccination Program', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.2, 0, len(lst\_vaccines), color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.2, 0, 'vaccines\nused', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.4, 0, total\_vaccine, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.4, 0, 'million of\ntotal vaccinations', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.65, 0, people\_vaccined, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.65, 0, 'million of\npeople vaccinated', color='dimgray', fontsize=5, ha='center', va='top', weight='bold')

ax0.text(0.9, 0, people\_fully\_vaccined, color='#2693d7', fontsize=20, ha='center', weight='bold', va='bottom')

ax0.text(0.9, 0, 'million of people\nfully vaccinated', color='dimgray', fontsize=6, ha='center', va='top', weight='bold')

plt.show()