**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

🙤🙧🟍🙥🙦



**DỰ ĐOÁN**

**GIÁ XE PHÂN KHỐI LỚN CŨ**

|  |  |  |
| --- | --- | --- |
| Sinh viên thực hiện: Nhóm 27 | | |
| STT | Họ tên | MSSV |
| 1 | Nguyễn Phú Trung | 21521592 |
| 2 | Lê Quốc Kiệt | 21521031 |
| 3 | Tạ Anh Khoa | 21522232 |

**TP. HỒ CHÍ MINH – 11/2023**

# 1. GIỚI THIỆU

Đề tài "Dự đoán giá xe phân khối lớn cũ" tập trung vào việc phân tích và dự đoán giá của các xe phân khối lớn đã qua sử dụng. Nhóm đã sử dụng phương pháp web scraping để thu thập dữ liệu từ trang web [Classic Trader](https://www.classic-trader.com/uk/cars/), tổng hợp thông tin về hơn 200 xe và 27 thông số khác nhau. Quá trình này được thực hiện bằng sự kết hợp giữa thư viện requests và BeautifulSoup trong ngôn ngữ lập trình Python.

Sau khi thu thập dữ liệu, nhóm em tiến hành quá trình làm sạch dữ liệu để loại bỏ các giá trị nhiễu và dữ liệu không cần thiết. Sau đó, chúng tôi sử dụng các phương pháp phân tích dữ liệu để thăm dò và tìm hiểu thông tin quan trọng về các yếu tố ảnh hưởng đến giá xe phân khối lớn đã qua sử dụng. Các phân tích này bao gồm việc xác định mối quan hệ giữa giá xe và các thông số như trạng thái kỹ thuật, động cơ, kích thước, và nhiều yếu tố khác.

Kết quả của đề tài này sẽ giúp nhóm xây dựng một mô hình dự đoán giá xe phân khối lớn cũ dựa trên dữ liệu thu thập và phân tích. Mô hình dự đoán này có thể hữu ích cho các người mua và bán xe phân khối lớn cũ để có cái nhìn sâu hơn về giá trị thực tế của các loại xe này trên thị trường.

Bộ dữ liệu do nhóm tự tiến hành đề xuất, được thu thập từ trang web [Classic Trader](https://www.classic-trader.com/uk/cars/), sau đó phân tích và thiết kế. Tất cả các thông tin và việc thực hiện đều không dựa trên đề tài nào khác.

# 2. MÔ TẢ BỘ DỮ LIỆU

## 2.1 Mô tả

Bộ dữ liệu xe phân khối lớn cũ là một tập hợp các thông tin về các xe phân khối lớn đã qua sử dụng, bao gồm các thông tin về hãng sản xuất, tên mẫu xe, kiểu xe, năm sản xuất, quảng đường đi, giá xe,..

Bộ dữ liệu xe phân khối lớn cũ của nhóm là bộ dữ liệu tự thu thập.

Bộ dữ liệu sau khi thực hiện thu thập ban đầu có 28 cột và 207 dòng. Tuy nhiên, sau khi thu thập nhóm nhận thấy dữ liệu ở một số cột (biến) bị khuyết với tỉ lệ cao (hơn 50%) hoặc chỉ có một giá trị (chiếm đến hơn 90%) nên nhóm đã quyết định thực hiện loại bỏ các cột trên.

Các biến đã bị loại bỏ: First-registration-date, Chassis-number, Engine-number, Gearbox-number, Matching-numbers, Previous-owners, Cafe-Racer, Starter, Gearbox, Gearshift, Front-break, Rear-break, Fuel-type.

Bộ dữ liệu chính thức bao gồm 15 cột trong đó: có 6 cột là biến số, 7 cột là biến phân loại và 2 cột vừa là biến số vừa là biến phân loại.

Giải thích các feature:

Make: Hãng sản xuất xe (PL)

Model: Tên mẫu xe (PL)

Type: Kiểu xe (PL)

Year-of-manufacture: Năm sản xuất (S)

Mileage: Quãng đường đã đi (S)

Engine-Type: Loại động cơ (PL)

Power: Công suất (S)

Cubic-capacity: Dung tích xi lanh (S)

Cylinders: Số xi lanh (S + PL)

Cooling: Hệ thống làm mát (PL)

Gears: Số cấp (S + PL)

Transmission: Hệ thống truyền động (PL)

Curb-weight: Trọng lượng xe không tải (S)

Exterior-color: Màu ngoại thất (PL)

Price: Giá bán (S)

* Trong đó:
* + (S): Biến số.
* + (PL): Biến phân loại.
* + (S + PL): Vừa là biến số vừa là biến phân loại.

Ví dụ về các feature:

Make: Ducati, Yamaha, Honda, Kawasaki…

Model: R 1100 RS, 999R, 999S…

Type: Chopper, Racing Motorcycle, Naked Bike…

Year-of-manufacture: 2019, 1996, 2000…

Mileage: 21,610 km, 1 km, 17,433 mls…

Engine-Type: Straight | 4-stroke | DOHC, 4 st, V | 4-Takt | DOHC…

Power: 40/55, 169/230 (Factory: 148/201), 77/105…

Cubic-capacity: 961, 744, 1172, 583…

Cylinders: 2, 4, 3, 1, 6…

Cooling: Fan, Air, Oil-air, water…

Gears: 6, 5, 4…

Transmission: Belt, Chain, Cardan…

Curb-weight: 0.164, 0.214, 0.207, 0.34…

Exterior-color: Red, Grey, Green, Blue, Others…

Price: 10,999, 6,988, 3,485, 61,142…

## 2.2 Phương pháp thu thập

Các công cụ và công nghệ hỗ trợ việc thu thập dữ liệu:

### 2.2.1 Jupyter notebook

Trong quá trình thu thập dữ liệu, nhóm sẽ làm việc với hai module BeautifulSoup và requests. Vì vậy, Jupyter notebook sẽ là công cụ không thể nào tuyệt vời hơn trong việc thu thập dữ liệu. Jupyter Notebook là môi trường lập trình tương tác cho phân tích dữ liệu, hỗ trợ ngôn ngữ Python cùng nhiều ngôn ngữ khác và tích hợp các thư viện phân tích mạnh mẽ, cùng khả năng tạo tài liệu trực quan và chia sẻ kết quả, giúp làm việc với dữ liệu trở nên hiệu quả và thuận tiện.

### 2.2.2 BeautifulSoup, requests

BeautifulSoup và requests là hai module được sử dụng rộng rãi để trích xuất dữ liệu từ các trang web.

#### a) BeautifulSoup

BeautifulSoup là một module được sử dụng để phân tích và xử lý mã HTML/XML. Tính năng nổi bật của BeautifulSoup đã được áp dụng vào việc thu thập dữ liệu trên là: phân tích cấu trúc của tài liệu HTML được lấy từ url của trang web, sau đó thực hiện tìm kiếm và khai thác dữ liệu mong muốn nhờ vào các phương thức find(), find\_all().

#### b) Requests

Requests là một module dùng để gửi và nhận các yêu cầu HTTP. Module này được sử dụng nhằm lấy script HMTL/XML của trang web mà chúng ta cần thu thập dữ liệu một cách dễ dàng mà không cần phải sử dụng trình duyệt web. Sau đó, kết hợp module BeautifulSoup để thực hiện việc tìm kiếm và truy xuất dữ liệu.

## 2.3 Nguồn dữ liệu

Để thu thập dữ liệu cho đồ án môn học, chúng em đã tham khảo qua nhiều trang web cũng như nhiều nguồn khác nhau. Cuối cùng, nhóm đã chọn trang web Classic Trader với đường dẫn: <https://www.classic-trader.com/> cùng với áp dụng một số filter để sàng lọc kết quả tìm kiếm để thu thập bộ dữ liệu với số lượng thích hợp và dữ liệu có ý nghĩa hơn. Đây là liên kết đến với trang đã được lọc và là nơi chứa dữ liệu mà nhóm đã thực hiện thu thập: [link](https://www.classic-trader.com/uk/motorcycle/search?yearBeginOfProduction=1995&priceMin=2500&priceMax=500000&currencyCode=gbp&pagination%5Bpage%5D=1). Classic Trader là một trang web đóng vai trò như là một international marketplace chuyên tổ chức mua bán các loại xe hơi, xe phân khối lớn cổ điển đã hoặc chưa qua sử dụng được thành lập từ 2013.

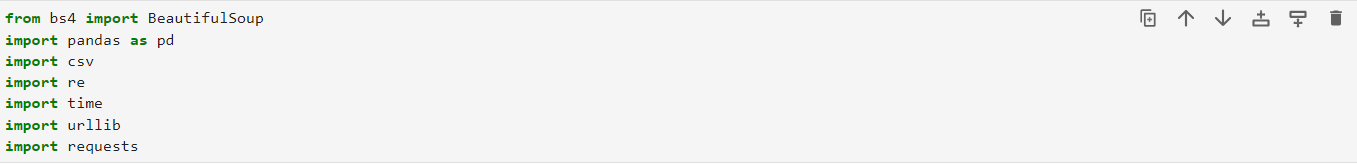
A screenshot of a computer

Description automatically generated

## 2.4 Tiến hành thu thập dữ liệu

Sau khi có đủ các thành phần để thu thập dữ liệu, nhóm bắt đầu tiến hành thu thập theo các bước sau:

* Import BeautifulSoup, requests cùng các module hỗ trợ khác:



* Khởi tạo các giá trị dùng để tương tác với số lượng trang nhất định của trang web:

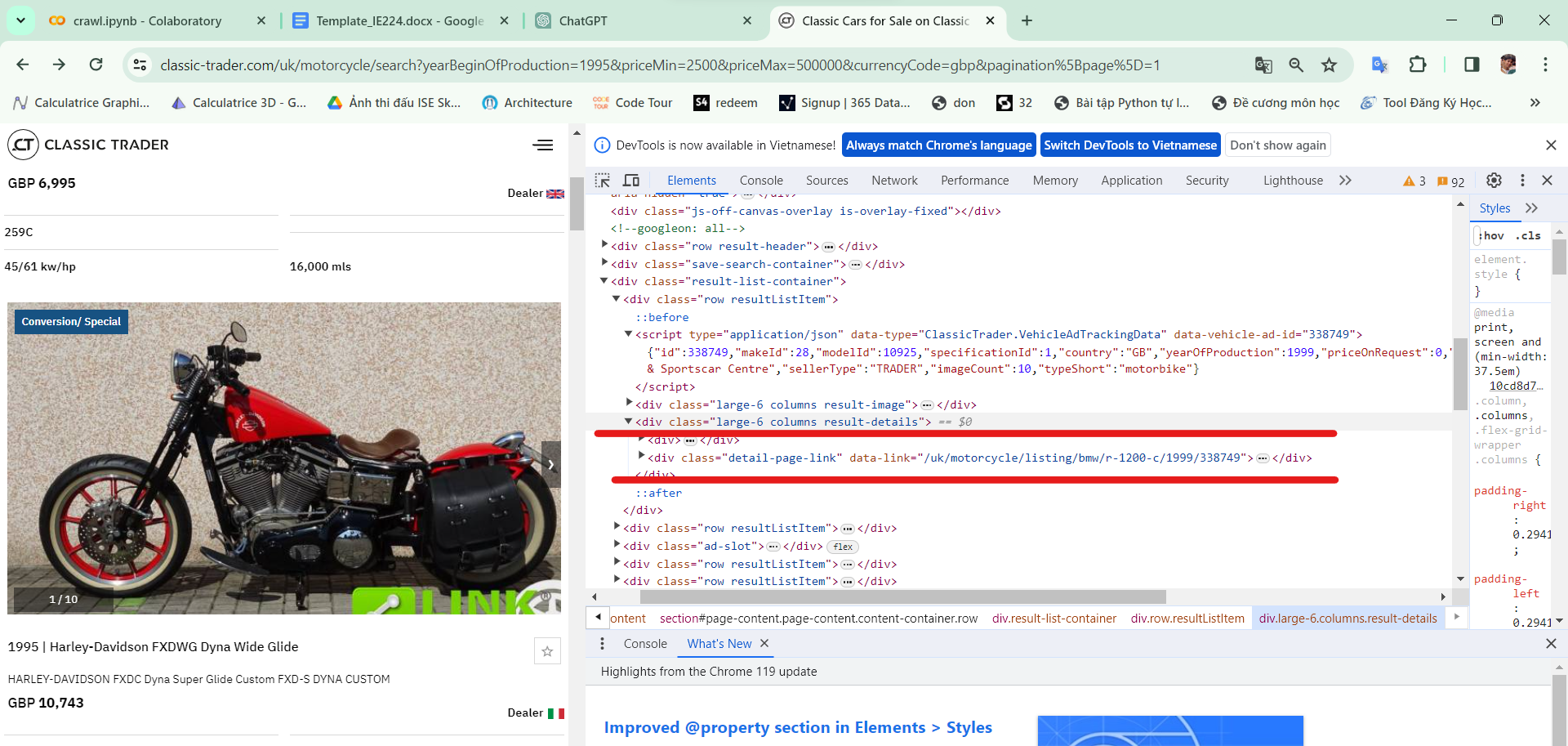
A screen shot of a computer

Description automatically generated

Ở đoạn code trên:

* base\_url: là biến dùng để lưu trữ url đến trang chủ của website, dùng để thực hiện ghép url để liên kết đến từng trang của website
* num\_Of\_Page: mục tiêu của nhóm là thu thập dữ liệu vào 14 trang đầu của trang web.
* base\_page\_link: là biến dùng để lưu trữ url (chưa hoàn chỉnh) đến từng trang của website tuy nhiên chưa bao gồm chỉ mục trang
* products\_pages: list được khởi tạo để chứa url của từng trang trong 14 trang
* vòng lặp for:
* thực hiện vòng lặp 14 lần – và biến i tương đương với chỉ số từ 1-14.
* thực hiện ghép từ số 1,2,..14 vào cuối base\_page\_link để tạo thành một url hoàn chỉnh đến lần lượt 14 trang và lưu vào products\_pages.
* Lấy các url con dẫn đến site chứa thông tin chi tiết của sản phẩm:

Ở mỗi ô chứa sản phẩm đều chứa một đoạn data-link mà khi kết hợp với đường dẫn trang chủ sẽ dẫn đến trang thông tin chi tiết của sản phẩm đó, vì vậy ý tưởng ở đây là truy cập vào từng trang chi tiết đó thông qua một list được khởi tạo để chứa các data-link đó. (hình bên dưới là ô sản phẩm cùng data-link của sản phẩm đó được gạch dưới trong đoạn script HTML bên phải).



Đầu tiên, tìm những vị trí chứa data-link và lưu lại data-link đó trong một list gọi là products\_link\_tails, việc lưu trữ những data-link này sẽ được thực hiện qua vòng lặp như sau:

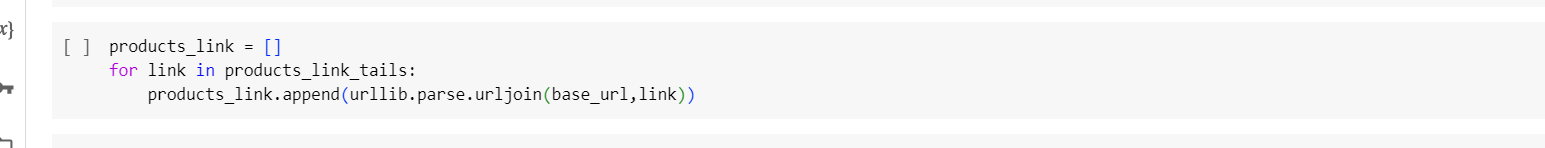
A white background with a black and white flag

Description automatically generated with medium confidence

Trong đoạn code trên:

* List products\_link\_tails: dùng để lưu data-link của từng sản phẩm
* Biến cnt: dùng để thực hiện việc truy xuất xen kẻ, do trong tệp HTML có đến 2 nơi chứa data-link nên sử dụng biến cnt nhằm lấy data-link không bị trùng lặp.
* Vòng lặp for:
* Duyệt qua từng url trong list products-pages
* Sau đó kết hợp lệnh thực hiện gửi request đến từng url và thực hiện trích xuất tệp HTML bằng BeautifulSoup
* Tìm các phần tử <div> có thuộc tính ‘data-link’ gán vào biến element
* Thực hiện thêm một vòng lặp for nhằm lấy được data-link không bị trùng lặp và thực hiện lưu trữ vào products\_link\_tails

Tiếp theo, thực hiện việc ghép url hoàn chỉnh bằng cách join base\_url với từng giá trị trong products\_link\_tails:



Sau khi vòng lặp kết thúc, chúng ta sẽ có được một list chứa toàn bộ data-link của các sản phẩm trong 14 trang đầu của trang web.

Khi đã có được đầy đủ liên kết, nhóm tiến hành thu thập dữ liệu bằng cách thu thập tên cột (biến) với ý tưởng: truy xuất đến một sản phẩm bất kì là thực hiện lấy tên biến bằng phương thức find\_all

A screen shot of a computer

Description automatically generated

Đoạn code trên thực hiện lấy các biến bằng cách gửi requests đến url\_sample sau đó thực hiện phân tích tệp HTML lấy được nhờ BeautifulSoup.

* Thực hiện lấy 27 giá trị phần tử <dt> (chứa thông tin cột trong HTML)
* Sau đó thực hiện một vòng for để lấy tên cột bằng thuộc tính .text
* Cuối cùng là thêm một cột “Price” vào list columns

Tiếp theo là xử lí tên cột:

A white background with a black border

Description automatically generated with medium confidence

Hàm convert\_string(s): thực hiện thay đổi chuỗi s qua ví dụ như sau:

* Chuỗi s ban đầu: s = “Van Toc (m/s)”
* s = “Van Toc “
* s = “Van Toc”
* s = “Van-Toc”

Sau khi thực hiện định dạng lại tên cột, dữ liệu tương ứng với tên cột sẽ được thu thập:

A screenshot of a computer

Description automatically generated

Đoạn code trên:

* Thực hiện tạo một file csv hoặc mở file csv có sẵn để lưu dataset: “PKL\_dataset.csv”.
* Mở file csv với mode write (‘w’) và thực hiện thêm dữ liệu:
* Tạo một đối tượng writer từ module csv để ghi dữ liệu vào tệp (“file”) vừa mở.
* Ghi một hàng vào tệp csv với columns là tên cột đã được xác định theo thứ tự.
* Vòng lặp for:
  + - Thực hiện gửi requests đến từng url đến từng sản phẩm.
    - Dùng time.sleep(3) để dừng 3 giây nhằm tránh việc gửi yêu cầu liên tiếp bị cấm hoặc quá tải cho máy chủ web.
    - Truy cập file HTML của sản phẩm bằng BeautifulSoup => gán vào biến ‘soup’.
    - Truy cập giá của từng sản phẩm: nằm ở phần tử <span> có thuộc tính class là “text-2xl font-semibold lg:text-3xl"

A screen shot of a computer

Description automatically generated

* + - Thực hiện tìm dữ liệu cho 27 biến bằng find\_all: Tìm tất cả các phần tử <dd> trong nội dung HTML.

A screenshot of a computer

Description automatically generated

* + - Lưu dữ liệu lấy được vào list row\_data bằng vòng lặp for
    - Thêm giá trị của biến price vào cuối row\_data.
    - Thêm bộ giá trị vừa lấy được vào file csv qua đối tượng writer.
* Việc ghi dữ liệu vào file csv kết thúc khi duyệt qua hết tất cả các url của từng sản phẩm.

Thống kê các số lượng biến khuyết:

Make 0

Model 0

Type 0

Year-of-manufacture 0

Mileage 6

Engine-Type 0

Power 0

Cubic-capacity 0

Cylinders 2

Cooling 5

Gears 19

Transmission 17

Curb-weight 40

Exterior-color 0

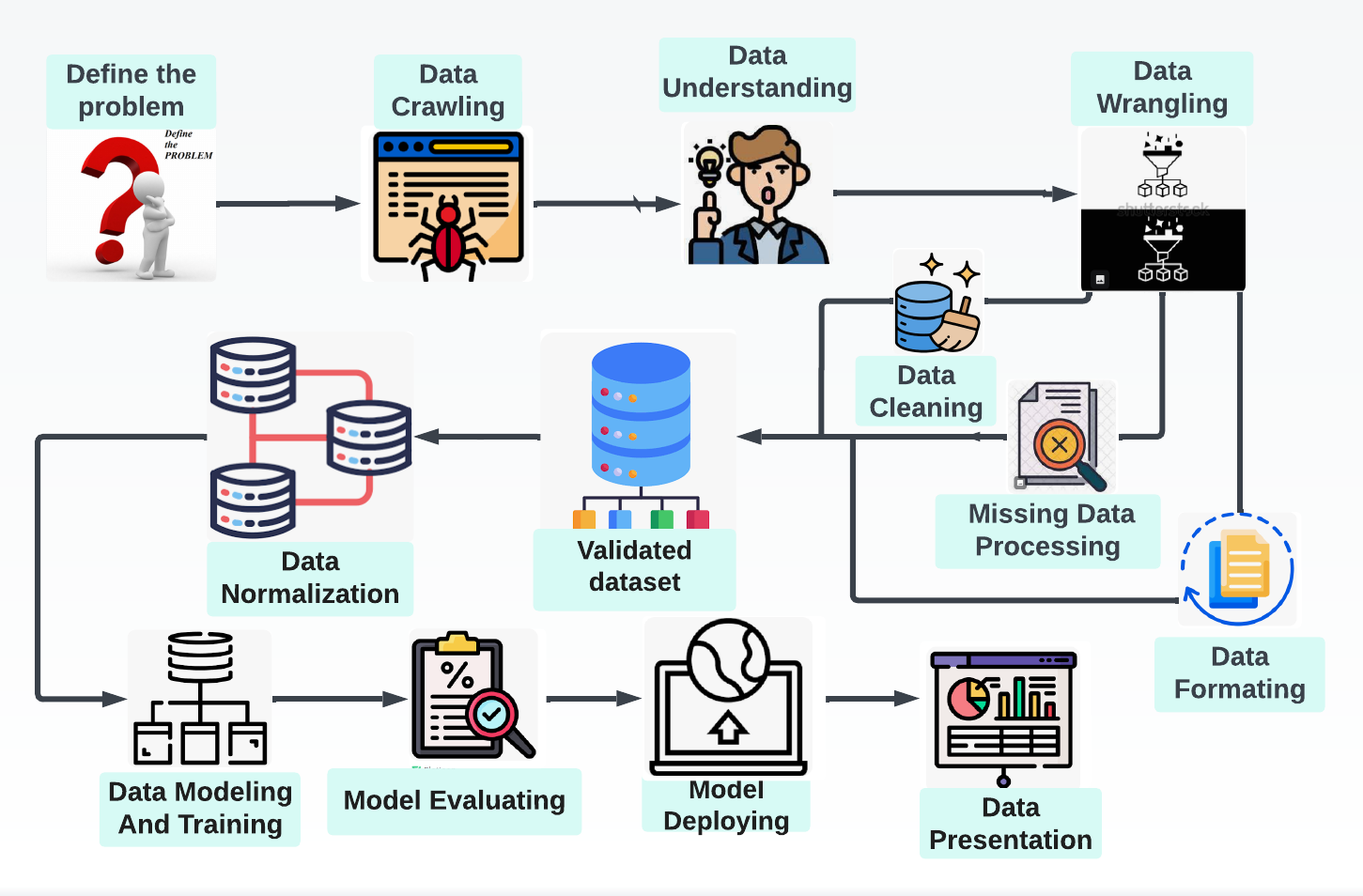
Price 0

# 3.  PHƯƠNG PHÁP PHÂN TÍCH (Có thể đặt lại nội dung tiêu đề)

Viết tốt được **6 điểm**

Phương pháp phân tích, có thể tiếp cận theo các cách sau

* Khuyến nghị tiếp cận theo quy trình hoặc các bước thực hiện phân tích dữ liệu đã được học. Hình:



Hình Quy trình phân tích dữ liệu

## Xác định vấn đề (Define the problem)

Nội dung đoạn văn dùng Style BT

* Gạch đầu dòng này dùng **Style G1**.

## Thu thập dữ liệu (Data crawling)

## Hiểu bộ dữ liệu (Data understanding)

## Sắp xếp dữ liệu (Data wrangling)

Nhằm mục tiêu có được bộ dữ liệu thích hợp, có ích cho việc phân tích, quá trình sắp xếp dữ liệu là vô cùng quan trọng. Quá trình này bao gồm

* Làm sạch dữ liệu (Data cleaning):
* Kiểm tra và xử lý các giá trị ngoại lệ trong bộ dữ liệu. Các giá trị này có thể do sai sót trong lúc thu thập.
* Loại bỏ đi các giá trị trùng lắp tránh việc trùng lắp dữ liệu ảnh hưởng đến kết quả dự đoán.
* Xử lí dữ liệu bị khuyết (Missing data processing):
* Phân tích mức độ khuyết của dữ liệu bằng cách thống kê số lượng các giá trị bị khuyết để hiểu đưa ra tiêu chí loại bỏ các cột trong bộ dữ liệu.
* Sử dụng phương pháp điền giá trị trung bình, trung vị, nhất suất một cách linh hoạt để thay vào các giá trị bị khuyết.
* Định dạng dữ liệu (Data formating):
* Tiến hành định dạng những cột chứa dữ liệu có thể chia thành nhiều biến bằng cách tách cột(biến) và giữ lại những cột(biến) cần thiết.
* Kiểm tra và chuyển đổi các biến số có đơn vị để đảm bảo tính đồng nhất về mặt đơn vị.
* Chuyển đổi các cột có nhiều giá trị nhưng chỉ thể hiện một thuộc tính và chọn cùng một đơn vị đồng nhất.

Kết quả thu được sau khi trải qua quá trình sắp xếp dữ liệu là một bộ dữ liệu đã được tối ưu hóa. Bộ dữ liệu này được xem là bộ dữ liệu đầy đủ có thể phục vụ cho việc chuẩn hóa, mô hình và phân tích.

## Chuẩn hóa dữ liệu (Data normalization)

## Mô hình hóa và training dữ liệu (Data modeling and training)

## Đánh giá mô hình (Model evaluating)

## Triển khai mô hình (Model deploying)

## Trình bày những phát hiện về dữ liệu (Data presentation)

# 4.  PHÂN TÍCH THĂM DÒ/SƠ BỘ (gợi ý)

Tập trình bày các phát hiện chính, không bắt buộc trình bày hết kết quả phân tích thăm dò các biến. Trình bày các biến quan trọng đã chọn lọc lại.

# 5.  KẾT QUẢ PHÂN TÍCH (gợi ý)

# 6.  KẾT LUẬN

Tóm tắt lại quá trình và kết quả hiện có.

Tóm tắt kết quả đạt được?

Viết 10 dòng, hoặc nửa trang.

Viết tốt được **2 điểm**

**Sinh viên trình bày không quá 10 trang cho tổng cộng 3 phần:**

**Giới thiệu.**

**Nội dung.**

**Kết luận.**

**(Không tính số trang tài liệu tham khảo)**

**Chú ý:**

* **Tại footer của báo cáo, nếu Họ tên SV dài có thể ghi tắt.   
  Ví dụ: Nguyễn Thị Thu Trang => Nguyễn T.T. Trang, hoặc N.T.T. Trang**
* **Nếu nhóm nhiều thành viên, không đủ chỗ ghi tên tại footer thì có thể không cần ghi.**

**TÀI LIỆU THAM KHẢO**

<https://www.classic-trader.com/>

…

**PHỤ LỤC PHÂN CÔNG NHIỆM VỤ**

|  |  |  |
| --- | --- | --- |
| **STT** | **Thành viên** | **Nhiệm vụ** |
| 1 | Nguyễn Phú Trung |  |
| 2 | Tạ Anh Khoa |  |
| 3 | Lê Quốc Kiệt |  |