Nguyễn Tiến Anh – B20DCCN005

**Bài tập 2: Hiểu biết về xử lý dữ liệu và các thuật toán học máy cơ bản**

# Bài 2.1:

Method **train\_test\_split()**  trong Sklearn là phương thức được sử dụng để chia dữ liệu thành tập train, validation và tập test, nếu chúng ta set test\_size thì train\_size = 1- test\_size và ngược lại, nếu chúng ta không set cái nào thì mặc định sẽ là test\_size = 0.25 và train\_size= 0.75.

Cụ thể hơn, cú pháp thường sẽ là như sau:

X\_train,X\_test,y\_train,y\_test**=**train\_test\_split(X,y,random\_state**=**104, test\_size**=**0.25, shuffle**=**True)

* Trong đó:
* X là feature và y là label, khung dữ liệu sẽ được chia thành 4 yếu tố X\_train, X\_test, y\_train, y\_test, Bộ X\_train và y\_train được sử dụng để train và fitting the model. And, X\_test và y\_test dùng để kiểm tra model nếu chúng ta muốn dự đoán kết quả.
* Random\_state: Kiểm soát cách xáo trộn dữ liệu trước khi thực hiện phân tách
* Shuffle: Có nên xáo trộn dữ liệu trước khi chia tách hay không
* Stratify: một đối tượng giống như mảng, nếu không phải là None, sẽ xác định cách sử dụng phân chia phân tầng
* Note:
* Nên để train\_size > test\_size
* Train\_test\_split  phù hợp khi bạn có tập dữ liệu rất lớn, mô hình tốn kém để đào tạo hoặc yêu cầu ước tính hiệu suất mô hình một cách nhanh chóng,  quy trình train-test không phù hợp khi tập dữ liệu có sẵn nhỏ. Lý do là khi tập dữ liệu được chia thành tập huấn luyện và tập kiểm tra, sẽ không có đủ dữ liệu trong tập dữ liệu huấn luyện để mô hình học cách ánh xạ đầu vào thành đầu ra một cách hiệu quả. Cũng sẽ không có đủ dữ liệu trong bộ thử nghiệm để đánh giá hiệu quả hiệu suất của mô hình.
* Kết quả của bạn có thể thay đổi tùy theo tính chất ngẫu nhiên của thuật toán hoặc quy trình đánh giá hoặc sự khác biệt về độ chính xác bằng số
* Tại sao cần split dataset:
* Việc chia tách tập dữ liệu của bạn là điều cần thiết để đánh giá khách quan về hiệu suất dự đoán. Trong hầu hết các trường hợp, việc chia ngẫu nhiên tập dữ liệu của bạn thành ba tập hợp con là đủ:
  + Tập huấn luyện được áp dụng để huấn luyện hoặc điều chỉnh mô hình của bạn. Ví dụ: bạn sử dụng tập huấn luyện để tìm trọng số hoặc hệ số tối ưu cho hồi quy tuyến tính, hồi quy logistic hoặc mạng thần kinh.
  + Bộ xác thực được sử dụng để đánh giá mô hình không thiên vị trong quá trình điều chỉnh siêu tham số. Ví dụ: khi bạn muốn tìm số lượng nơ-ron tối ưu trong mạng nơ-ron hoặc hạt nhân tốt nhất cho máy vectơ hỗ trợ, bạn thử nghiệm với các giá trị khác nhau. Đối với mỗi cài đặt siêu tham số được xem xét, bạn điều chỉnh mô hình với tập huấn luyện và đánh giá hiệu suất của nó bằng tập xác thực.
  + Bộ kiểm tra là cần thiết để đánh giá khách quan mô hình cuối cùng. Bạn không nên sử dụng nó để điều chỉnh hoặc xác nhận.
* Trong những trường hợp ít phức tạp hơn, khi bạn không phải điều chỉnh siêu tham số, bạn có thể chỉ làm việc với các tập huấn luyện và kiểm tra.
* Cài đặt Sklearn:
  + pip install -u scikit-learn
  + nếu sử dụng anaconda hoặc miniconda có thể sử dụng: conda install -c anaconda scikit-learn
* Các chức năng khác trong gói sklearn.model\_selection liên quan đến select và validation bao gồm:
* Cross-validation (xác thực chéo)
  + Là một tập hợp các kỹ thuật kết hợp các thước đo hiệu suất dự đoán để có được ước tính mô hình chính xác hơn
  + Một trong những phương pháp xác thực chéo được sử dụng rộng rãi là xác thực chéo k-fold. Trong đó, bạn chia tập dữ liệu của mình thành k (thường là năm hoặc mười) tập con hoặc tập hợp gấp, có kích thước bằng nhau, sau đó thực hiện quy trình huấn luyện và kiểm tra k lần. Mỗi lần, bạn sử dụng một nếp gấp khác nhau làm tập kiểm tra và tất cả các nếp gấp còn lại làm tập huấn luyện. Điều này cung cấp k thước đo hiệu suất dự đoán và sau đó bạn có thể phân tích giá trị trung bình và độ lệch chuẩn của chúng
  + Bạn có thể triển khai xác thực chéo với KFold, StratifiedKFold, LeaveOneOut cũng như một số lớp và hàm khác từ sklearn.model\_selection
* Learning curves đôi khi được gọi là đường cong huấn luyện, cho thấy điểm dự đoán của tập huấn luyện và xác nhận phụ thuộc như thế nào vào số lượng mẫu huấn luyện. Bạn có thể sử dụng learning\_curve() để có được phần phụ thuộc này, điều này có thể giúp bạn tìm ra kích thước tối ưu của tập huấn luyện, chọn siêu tham số, so sánh các mô hình, v.v.
* Hyperparameter tuning còn được gọi là tối ưu hóa siêu tham số, là quá trình xác định tập hợp siêu tham số tốt nhất để xác định mô hình học máy của bạn. sklearn.model\_selection cung cấp cho bạn một số tùy chọn cho mục đích này, bao gồm GridSearchCV, RandomizedSearchCV, validation\_curve() và các tùy chọn khác.
* Example:
  + Example 1:

A screen shot of a computer program

Description automatically generated

* + Example 2:

A screen shot of a computer program

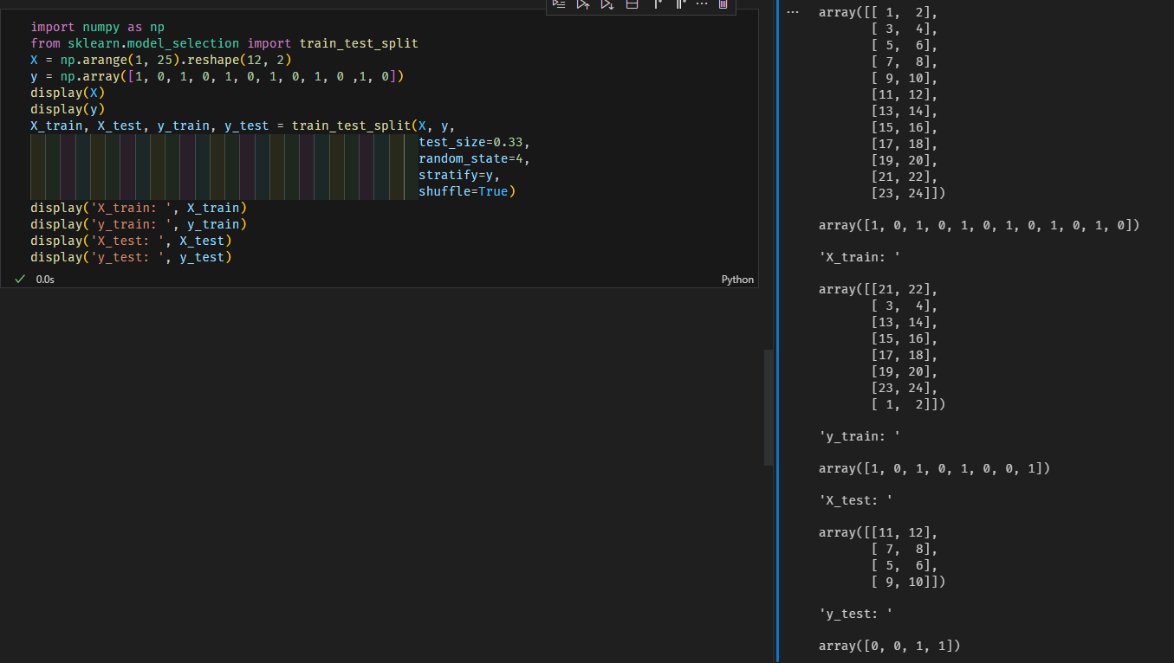
Description automatically generated

* + Example 3:

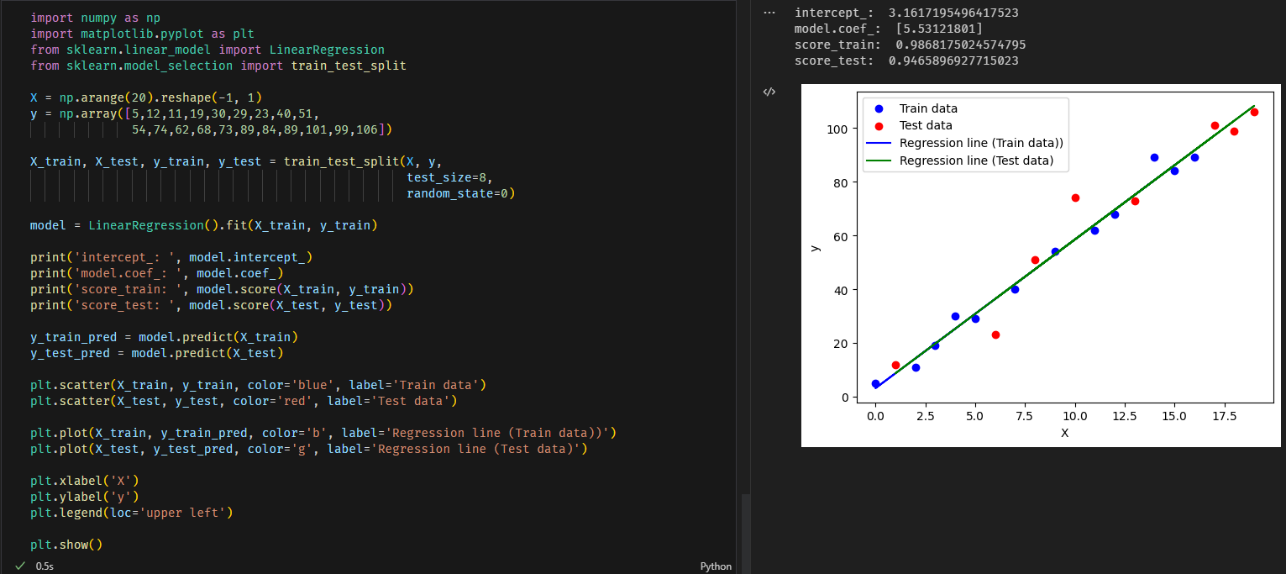
A screenshot of a computer

Description automatically generated

* + Example 4:



* + Example 5:



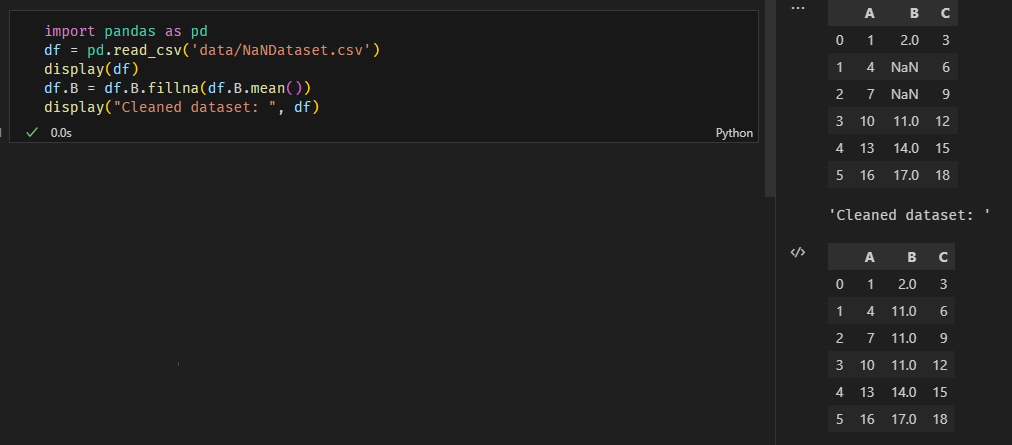
# Bài 2.2:

Trong học máy, một trong những nhiệm vụ đầu tiên bạn cần thực hiện là làm sạch dữ liệu. Rất hiếm khi bạn có một tập dữ liệu có thể sử dụng ngay lập tức

để đào tạo mô hình của bạn. Thay vào đó, bạn phải kiểm tra dữ liệu cẩn thận để tìm bất kỳ giá trị bị thiếu nào và loại bỏ chúng hoặc thay thế chúng bằng một số giá trị hợp lệ hoặc bạn phải chuẩn hóa chúng nếu có các cột có giá trị rất khác nhau. Các phần sau đây trình bày một số tác vụ phổ biến bạn cần thực hiện làm sạch dữ liệu.

1. Cleaning rows with NaNs:
   * Cách 1: Thay NaN bằng giá trị trung bình của các cột mà chúng thuộc về: df.B = df.B.fillna(df.B.mean())

Example:



* + Cách 2: Xóa các dòng có chứa NaN:

df.dropna()

df.reset\_index(drop=True)

Example:

A screen shot of a computer

Description automatically generated

1. Remove duplicate rows:

* Bước 1: tìm hàng trùng lặp: print(df.duplicated(keep=False))
* Bước 2: Loại bỏ hàng trùng lặp: df.drop\_duplicates(keep=’first’, inplace=True)
* **You can:** remove all duplicates in columns A and C and keep the last: df.drop\_duplicates(subset=['A', 'C'], keep='last', inplace=True)

Example:

A screenshot of a computer program

Description automatically generated

1. Normalizing Columns:

* Là một kỹ thuật thường được áp dụng trong quá trình làm sạch dữ liệu. Mục đích của việc chuẩn hóa là thay đổi giá trị của các cột số trong tập dữ liệu để sử dụng thang đo chung mà không sửa đổi sự khác biệt trong phạm vi giá trị.
* Chuẩn hóa là rất quan trọng đối với một số thuật toán để mô hình hóa dữ liệu một cách chính xác. Ví dụ: một trong các cột trong tập dữ liệu của bạn có thể chứa các giá trị từ 0 đến 1 trong khi cột khác có các giá trị từ 400.000 đến 500.000. Sự chênh lệch lớn về tỷ lệ các con số có thể gây ra vấn đề khi bạn sử dụng hai cột để huấn luyện mô hình của mình. Bằng cách sử dụng chuẩn hóa, bạn có thể duy trì tỷ lệ giá trị trong hai cột trong khi vẫn giữ chúng ở phạm vi giới hạn. Trong Pandas, bạn có thể sử dụng lớp MinMaxScaler để chia tỷ lệ từng cột thành một khoảng giá trị cụ thể.

Example:

A screen shot of a computer program

Description automatically generated

1. Removing Outliers

* Trong thống kê, điểm ngoại lệ là một điểm nằm cách xa các điểm được quan sát khác. Ví dụ: cho một tập hợp các giá trị—234, 267, 1, 200, 245, 300, 199, 250, 8999 và 245—rõ ràng là 1 và 8999 là các giá trị ngoại lệ. Họ nổi bật rõ ràng so với phần còn lại của các giá trị và chúng “nằm bên ngoài” hầu hết các giá trị khác trong tập dữ liệu; do đó từ ngoại lệ. Các ngoại lệ xảy ra chủ yếu do lỗi ghi hoặc lỗi thử nghiệm và trong học máy, điều quan trọng là phải loại bỏ chúng trước khi huấn luyện mô hình của bạn vì nó có thể làm sai lệch mô hình của bạn nếu bạn không làm như vậy. Có một số kỹ thuật để loại bỏ các giá trị ngoại lệ và trong chương này chúng ta thảo luận về hai trong số đó:
  + **Tukey Fences**: dựa trên Phạm vi liên vùng (IQR). IQR là sự khác biệt giữa tứ phân vị thứ nhất và thứ ba của một tập hợp các giá trị. Tứ phân vị đầu tiên, ký hiệu là Q1, là giá trị trong tập dữ liệu chứa 25% các giá trị bên dưới nó. Tứ phân vị thứ ba, ký hiệu là Q3, là giá trị trong tập dữ liệu chứa 25% các giá trị phía trên nó. Do đó, theo định nghĩa, IQR = Q3 – Q1.

Example:

A screenshot of a computer program

Description automatically generated

* + **Z-Score:** xác định các giá trị ngoại lệ là sử dụng phương pháp điểm Z. Một điểm Z cho biết điểm dữ liệu có bao nhiêu độ lệch chuẩn so với giá trị trung bình

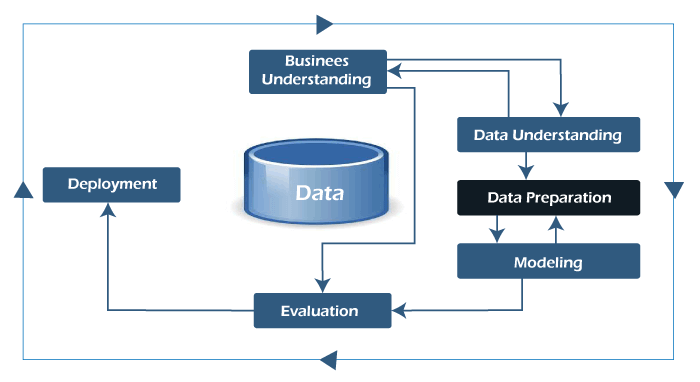
Example:

A screenshot of a computer

Description automatically generated

# Bài 2.3:

* Wrangling còn được gọi là trộn dữ liệu. Đó là quá trình chuyển đổi và ánh xạ dữ liệu từ một dạng dữ liệu "thô" sang định dạng khác để làm cho nó phù hợp và có giá trị hơn cho các mục đích tiếp theo khác nhau như phân tích. Mục tiêu của việc sắp xếp dữ liệu là đảm bảo dữ liệu có chất lượng và hữu ích. Các nhà phân tích dữ liệu thường dành phần lớn thời gian của họ cho quá trình sắp xếp dữ liệu so với việc phân tích dữ liệu thực tế.
* Quá trình sắp xếp dữ liệu có thể bao gồm việc trộn thêm, trực quan hóa dữ liệu, tổng hợp dữ liệu, đào tạo mô hình thống kê và nhiều ứng dụng tiềm năng khác. Việc sắp xếp dữ liệu thường tuân theo một tập hợp các bước chung, bắt đầu bằng việc trích xuất dữ liệu thô từ nguồn dữ liệu, "trộn" dữ liệu thô (ví dụ: sắp xếp) hoặc phân tích dữ liệu thành các cấu trúc dữ liệu được xác định trước và cuối cùng gửi nội dung kết quả vào một bồn chứa dữ liệu để lưu trữ và sử dụng trong tương lai.



Wrangling data (hoặc data wrangling) là quá trình chuẩn bị và biến đổi dữ liệu để làm cho nó phù hợp và sử dụng được cho các nhiệm vụ phân tích dữ liệu hoặc máy học. Các bước trong quá trình wrangling data thường bao gồm:

1. Thu thập dữ liệu: Thu thập dữ liệu từ các nguồn khác nhau như cơ sở dữ liệu, tệp tin, web, hoặc cảm biến. Dữ liệu có thể đến từ nhiều nguồn khác nhau và có định dạng và cấu trúc khác nhau.
2. Khám phá dữ liệu (Data Exploration): Trước khi bắt đầu quá trình wrangling, hãy thực hiện phân tích sơ bộ để hiểu cấu trúc và đặc điểm của dữ liệu. Điều này bao gồm việc kiểm tra thiếu dữ liệu, phân phối, giá trị ngoại lai, và các khía cạnh khác của dữ liệu.
3. Xử lý dữ liệu thiếu: Xử lý các giá trị dữ liệu bị thiếu bằng cách điền giá trị mặc định, loại bỏ dữ liệu thiếu, hoặc sử dụng các kỹ thuật impute để dự đoán giá trị thiếu dựa trên dữ liệu khác.
4. Xử lý dữ liệu ngoại lai: Xác định và xử lý các giá trị ngoại lai (outliers) có thể làm ảnh hưởng đến kết quả phân tích. Có thể xóa giá trị ngoại lai, biến đổi chúng hoặc sử dụng các kỹ thuật xử lý khác để ổn định dữ liệu.
5. Biến đổi và định dạng dữ liệu: Chuyển đổi dữ liệu thành định dạng thích hợp cho mô hình hoặc phân tích. Điều này có thể bao gồm chọn các đặc trưng quan trọng, mã hóa dữ liệu hạng mục thành số, chia dữ liệu thành các bin, và nhiều phương pháp biến đổi khác.
6. Tạo dữ liệu mới (Feature Engineering): Tạo ra các đặc trưng mới dựa trên dữ liệu hiện có để cải thiện khả năng dự đoán của mô hình. Điều này có thể bao gồm việc tạo tổng hợp các đặc trưng, tạo đặc trưng tương tác, và sử dụng các phép biến đổi dữ liệu sáng tạo.
7. Tạo tập dữ liệu kiểm tra (Test Dataset): Chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu suất của mô hình sau khi huấn luyện.
8. Lưu trữ và quản lý dữ liệu: Lưu trữ dữ liệu một cách có cấu trúc và bảo quản phiên bản của dữ liệu gốc và tất cả các biến đổi đã thực hiện để có khả năng tái sử dụng và kiểm tra lại.
9. Kiểm tra và đánh giá dữ liệu: Kiểm tra dữ liệu sau khi đã được wrangling để đảm bảo rằng nó đủ chất lượng cho mục tiêu phân tích hoặc máy học.

Quá trình wrangling data có vai trò quan trọng trong quá trình chuẩn bị dữ liệu cho các dự án phân tích dữ liệu và máy học. Nó giúp đảm bảo dữ liệu là đáng tin cậy, có ý nghĩa và phù hợp để tạo ra các mô hình và phân tích có giá trị.

Example: Data replacing in data wrangling

A screenshot of a computer

Description automatically generated

Example: Filtering data in Data Wrangling

A screenshot of a computer program

Description automatically generated

Example: Creating First Dataframe to Perform Merge Operation using Data Wrangling

A screenshot of a computer

Description automatically generated

Example: Creating Second Dataframe to Perform Merge operation using Data Wrangling

A screenshot of a computer

Description automatically generated

Example: Data Wrangling Using Merge Operation:

A screenshot of a computer program

Description automatically generated

Example: Creating dataframe to use Grouping methods[Car selling datasets]:

A screen shot of a computer

Description automatically generated

Example: Creating New Datasets Using the Concatenation of Two Datasets In Data Wrangling.

# Bài 2.4:

# Bài 2.5:

# Bài 2.6:

# Bài 2.7:

# Bài 2.8:

# Bài 2.9:

# Bài 2.10:

# Bài 2.11:

# Bài 2.12:

# Bài tập Deployment: