ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC BÁCH KHOA

CHƯƠNG 5: XỬ LÝ DỮ LIỆU

Khoa Công nghệ thông tin

Tài liệu tham khảo

- Các khoá học về KHDL và học máy trên internet

Nội dung

- Làm sạch dữ liệu (Data cleaning)
- Chuẩn hoá dữ liệu (Data normalization)

Làm sạch dữ liệu

- Xử lý dữ liệu trống (missing data)
- Xử lý dữ liệu ngoại lệ (outliers)

DATA CLEANING

DATA MANIP.
Find data
Load data
Merge/join data
Data dictionary

→

DATA CLEANING
Missing treatment
Cleaning
Outliers detection

EDA
Range, Distribution
Classes, Freq., Perc.
Anomalies, Correlation
Relationship, Patterns

TRANSFORMATION
Fea. construction
Fea. selection
Fea. transformation
Train/Test/Val. sep.

Missing : Data is not available

Type of Missing	Solution
Small % of observation	- Leave missing (treat as a category) - Delete missing - Single value or Model-based imputation
High % of observation	- Leave missing (treat as a category) or Consider remove fea.
Small % at multiple fea.	- Delete missing and contact data source people

Cleaning: Data is available but looks strange (Use EDA to detect)

Type of problems:	Solution
String: - Typing error: 123a,... - Wrongly data manip. - Non-unique value: Hanoi or Ha noi or Ha_noi or HaNoi - Abbrev.: Chicago or IL	- Correct manually (using script) and/or contact data source people - Check previous step (data manip.)
Format/type: date, temperature, numeric vs string	- Convert manually using script

Outliers: Data is available but looks irregular (Use EDA to detect)

Outlier detection	Reason and treatment
Unreasonable value: Negative age, extremely high/low value	- Input errors -> treat as missing and/or contact data source people Scaling error: check unit
Zero value (income, date)	- Probably missing value which is converted in to numeric
Irregular pattern	- Could have valuable information
Others	- Consider Transformation

Easy Data Science @ s-lab.vn - khoahocdulieudedang@gmail.com – Follow us on Facebook **Học Machine Learning và Deep Learning**

Làm sạch dữ liệu

Xử lý dữ liệu trống (missing data)

Dữ liệu trống là gì

- Là khi biến không nhận giá trị nào trong một quan sát (observation)
- Được thể hiện bởi các ô trống (hoặc giá trị NaN) trong bảng dữ liệu

Ví dụ: Titanic Dataset on Kaggle (dữ liệu được thu thập sau khi vụ tai nạn xảy ra)

PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	0	3	Braund, Mr.	male	22	1	0	A/5 21171	7.25		S
2	1	1	Cumings, Mr	female	38	1	0	PC 17599	71.2833	C85	C
3	1	3	Heikkinen, M	female	26	0	0	STON/O2.	7.925		S
4	1	1	Futrelle, Mrs	female	35	1	0	113803	53.1	C123	S
5	0	3	Allen, Mr. W	male	35	0	0	373450	8.05		S
6	0	3	Moran, Mr. J	male		0	0	330877	8.4583		Q
7	0	1	McCarthy, M	male	54	0	0	17463	51.8625	E46	S
8	0	3	Palsson, Mas	male	2	3	1	349909	21.075		S
9	1	3	Johnson, Mrs	female	27	0	2	347742	11.1333		S
10	1	2	Nasser, Mrs.	female	14	1	0	237736	30.0708		C
11	1	3	Sandstrom, I	female	4	1	1	PP 9549	16.7	G6	S
12	1	1	Bonnell, Mis	female	58	0	0	113783	26.55	C103	S
13	0	3	Saundercock	male	20	0	0	A/5. 2151	8.05		S
14	0	3	Andersson, M	male	39	1	5	347082	31.275		S
15	0	3	Vestrom, Mi	female	14	0	0	350406	7.8542		S
16	1	2	Hewlett, Mrs	female	55	0	0	248706	16		S
17	0	3	Rice, Master	male	2	4	1	382652	29.125		Q
18	1	2	Williams, Mr	male		0	0	244373	13		S
19	0	3	Vander Plank	female	31	1	0	345763	18		S
20	1	3	Masselmani,	female		0	0	2649	7.225		C

Ý nghĩa các biến của Titanic Dataset

Tên biến	Ý nghĩa
Passengerid	Passenger's ID
Pclass	Passenger's socio-economic status (1st ~ Upper; 2nd ~ Middle; 3rd ~ Lower)
Survived	Survival (0 = No; 1 = Yes)
Name	Name
Sex	Sex
Age	Age
Sibsp	Number of Siblings/Spouses Aboard
Parch	Number of Parents/Children Aboard
Ticket	Ticket Number
Fare	Passenger Fare
Cabin	Cabin number
Embarked	Port of Embarkation (C = Cherbourg; Q = Queenstown; S = Southampton)

Thống kê dữ liệu trống với Pandas

```
import pandas as pd
df=pd.read_csv('titanic.csv')
df.isnull().sum()
```

```
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

3 cột có dữ liệu trống

Các kiểu dữ liệu trống

1. Dữ liệu trống hoàn toàn ngẫu nhiên:

- Xác suất dữ liệu bị trống là như nhau đối với mọi quan sát
- Không có mối quan hệ nào giữa dữ liệu trống và các dữ liệu khác

```
# Vi du: cac hanh khach ko co du lieu ve dia diem len tau
df[df['Embarked'].isnull()]
```

Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	Icard, Miss. Amelie	female	38.0	0	0	113572	80.0	B28	NaN
1	Stone, Mrs. George Nelson (Martha Evelyn)	female	62.0	0	0	113572	80.0	B28	NaN

Các kiểu dữ liệu trống (tt)

2. Dữ liệu trống một cách không ngẫu nhiên:

- Dữ liệu bị trống một cách có hệ thống
- Có mối quan hệ giữa dữ liệu trống và các dữ liệu khác

```
import numpy as np
df['cabin_null']=np.where(df['Cabin'].isnull(),1,0)
# Tim phan tram hanh khach ko co du lieu ve cho ngoi
df['cabin_null'].mean() # → 0.7710437710437711
```

```
# Tim phan tram hanh khach ko co du lieu ve cho ngoi
# phan theo tinh trang song/chet
df.groupby(['Survived'])['cabin_null'].mean()
```

```
Survived
0    0.876138
1    0.602339
Name: cabin_null, dtype: float64
```

→ Những người ko sống sót bị trống dữ liệu về chỗ ngồi nhiều hơn những người sống sót

Tại sao dữ liệu bị trống?

- Do được thu thập từ nhiều nguồn, có nguồn tồn tại những quan sát không đầy đủ dữ liệu
- Do quá trình thu thập dữ liệu, đặc biệt là từ các khảo sát (survey)
  - Người trả lời không muốn điền thông tin cá nhân (ví dụ: lương, tuổi)
  - Đối tượng được khảo sát không còn sống nữa
  - Người nhập liệu nhập sai → thông tin không hợp lệ (coi như bỏ trống)

Các kỹ thuật xử lý dữ liệu trống

1. Xóa các quan sát có bất kỳ phần tử dữ liệu nào bị bỏ trống

→ nguy hiểm (các quan sát bị loại bỏ có thể chứa thông tin quan trọng)

2. Thay thế giá trị của dữ liệu trống bằng:

- Giá trị trung bình/trung vị/giá trị xuất hiện nhiều nhất (Mean/ Median/Mode imputation)
- Giá trị được lấy ngẫu nhiên từ các giá trị khác (Random sample imputation)
- Giá trị tại đuôi của phân bố dữ liệu (End of distribution imputation)

Thay thế bằng Mean/Median/Mode

- Kỹ thuật này giả định rằng dữ liệu trống hoàn toàn ngẫu nhiên
- Thay thế dữ liệu trống bằng Mean/Median/Mode của cột dùng **hàm fillna()** của đối tượng DataFrame

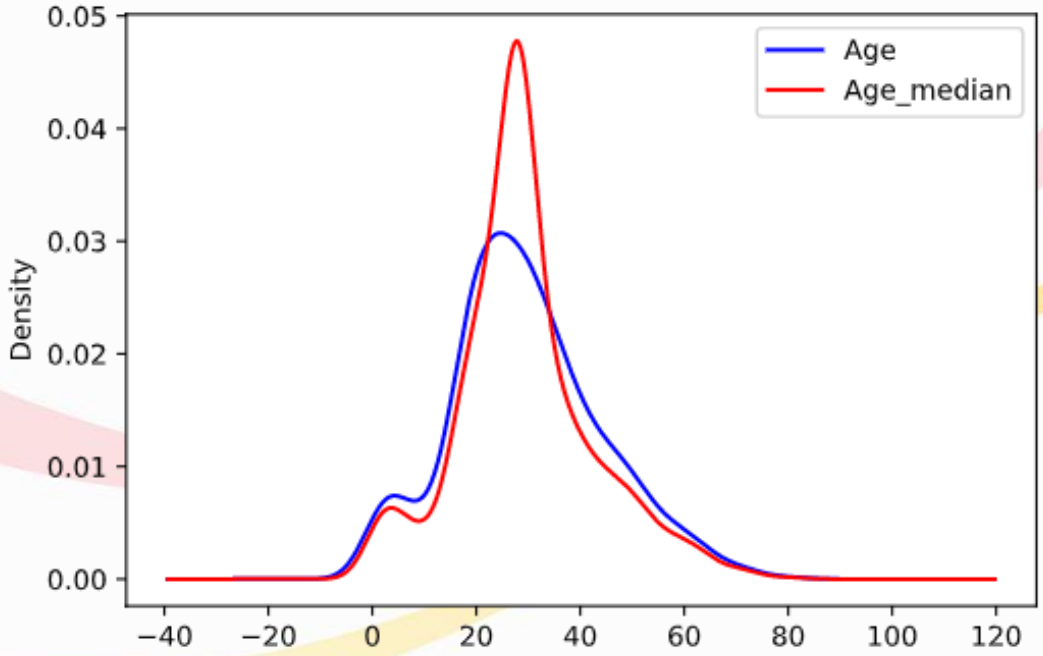
```
df=pd.read_csv('titanic.csv',usecols=['Age','Fare','Survived'])
# Phan tram du lieu trong cua moi cot
df.isnull().mean()
```

```
Survived    0.000000
Age         0.198653
Fare        0.000000
dtype: float64
```

```
# dien du lieu trong cua cot Age bang gia tri Median
median=df['Age'].median()          # → 28.0
df['Age_median']=df['Age'].fillna(median)
# in ra do lech chuan truoc va sau khi dien du lieu trong
print(df['Age'].std())       # → 14.526497332334044
print(df['Age_median'].std()). # → 13.019696550973194
```

Thay thế bằng Mean/Median/Mode (tt)

```
# ve ham mat do xac suat cua cot Age truoc va sau khi dien
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(111)
df['Age'].plot(kind='kde', color='blue')
df['Age_median'].plot(kind='kde', color='red')
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```



Thay thế bằng Mean/Median/Mode (tt)

- Ưu điểm:
  - Dễ thực hiện (ít bị ảnh hưởng bởi giá trị ngoại lệ)
  - Nhanh chóng nhận được dataset hoàn chỉnh
- Nhược điểm:
  - Làm thay đổi phương sai của dữ liệu

Bài tập 1: Thực hiện thay thế các NaN trong cột Age bằng Mean và Mode. Nhận xét kết quả.

Thay thế bằng giá trị ngẫu nhiên

- Kỹ thuật này cũng giả định rằng dữ liệu trống hoàn toàn ngẫu nhiên
- Thay thế dữ liệu trống bằng giá trị ngẫu nhiên của cột tương ứng
- Dùng **hàm dropna()** của đối tượng DataFrame để bỏ qua các NaN values

```
# tiep tuc vi du trong slide truoc: xu ly cac NaN trong cot Age
df['Age'].isnull().sum() # → 177
# lay ngau nhien tu cot Age mot gia trị khac NaN,
# ket qua se ko lap lai sau moi lan thuc hien lenh
df['Age'].dropna().sample()
```

```
824    2.0
Name: Age, dtype: float64
```

Thay thế bằng giá trị ngẫu nhiên (tt)

```
# lay ngau nhien tu cot Age n gia trị khac NaN
# ket qua se lap lai sau moi lan thuc hien lenh
random_samples = df['Age'].dropna().sample(n=df['Age'].isnull().sum(),random_state=0)
random_samples
```

```
423    28.00
177    50.00
305     0.92
292    36.00
889    26.00
       ...
539    22.00
267    25.00
352    15.00
99     34.00
689    15.00
Name: Age, Length: 177, dtype: float64
```

Thay thế bằng giá trị ngẫu nhiên (tt)

```
# chi so cua cac khach hang bi trong du lieu Age
df[df['Age'].isnull()].index
```

```
Int64Index([  5,  17,  19,  26,  28,  29,  31,  32,  36,  42,
            ...
            832, 837, 839, 846, 849, 859, 863, 868, 878, 888],
           dtype='int64', length=177)
```

```
# gan lai index cho series ngau nhien vua tao
random_samples.index = df[df['Age'].isnull()].index
```

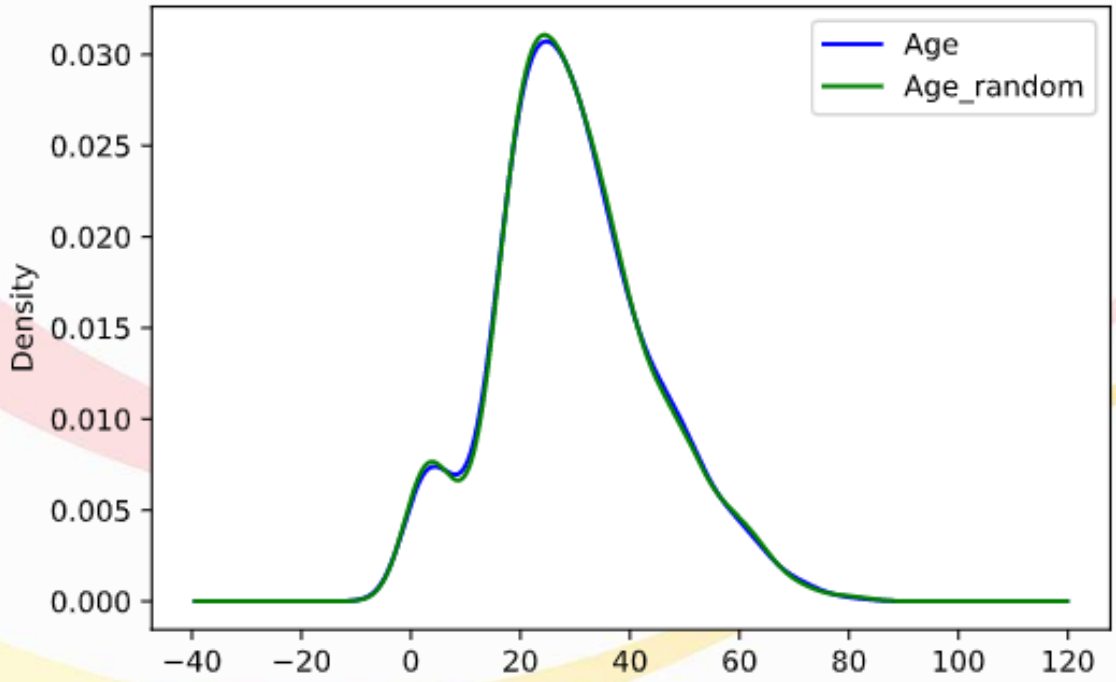
Thay thế bằng giá trị ngẫu nhiên (tt)

```
# Thay thế dữ liệu trống bằng các giá trị ngẫu nhiên của cột
df['Age_random']=df['Age']
df.loc[df['Age'].isnull(), 'Age_random']=random_samples
df.tail()
```

	Survived	Age	Fare	Age_random
886	0	27.0	13.00	27.0
887	1	19.0	30.00	19.0
888	0	NaN	23.45	15.0
889	1	26.0	30.00	26.0
890	0	32.0	7.75	32.0

Thay thế bằng giá trị ngẫu nhiên (tt)

```
# ve ham mat do xac suat cua cot Age truoc va sau khi dien
import matplotlib.pyplot as plt
fig = plt.figure()
ax = fig.add_subplot(111)
df['Age'].plot(kind='kde', ax=ax, color='blue')
df['Age_random'].plot(kind='kde', ax=ax, color='green')
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```



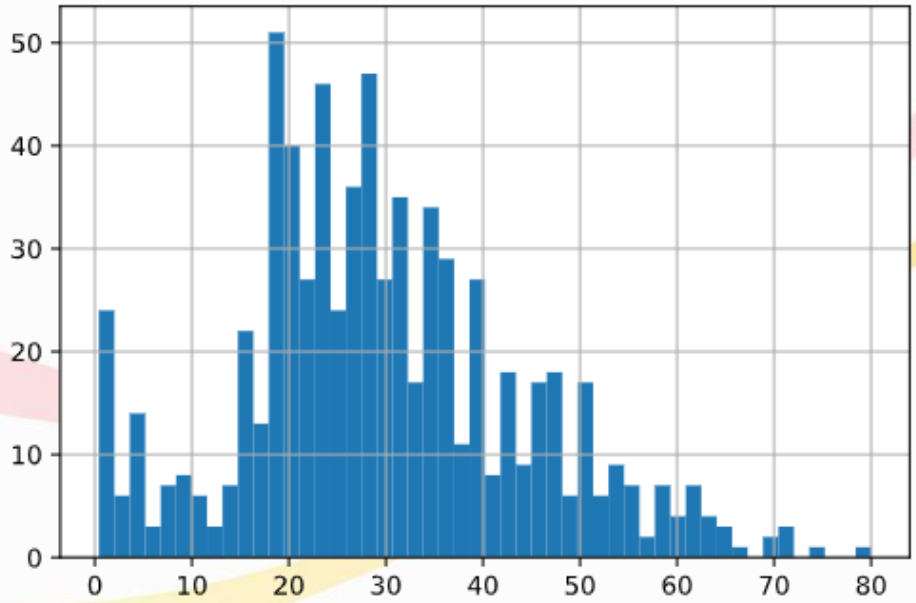
Thay thế bằng giá trị ngẫu nhiên (tt)

- Ưu điểm:
  - Dễ thực hiện
  - Phương sai của dữ liệu ít bị biến đổi
- Nhược điểm:
  - Không phải lúc nào dữ liệu trống cũng có tính ngẫu nhiên

Thay thế bằng giá trị đuôi của phân bố

- Kỹ thuật này được dùng khi có nghi ngờ rằng: **dữ liệu trống một cách KHÔNG ngẫu nhiên** (vd: người già dễ bị chết → ko có thông tin về Tuổi). Do đó cần nắm bắt thông tin quan trọng này
- Thay thế dữ liệu trống bằng giá trị ở đuôi của phân bố dữ liệu

```
df.Age.hist(bins=50)
# gia tri o duoi cua phan bo (bien Age theo phan bo chuan)
extreme = df.Age.mean() + 3*df.Age.std() # →extreme=73.27
```

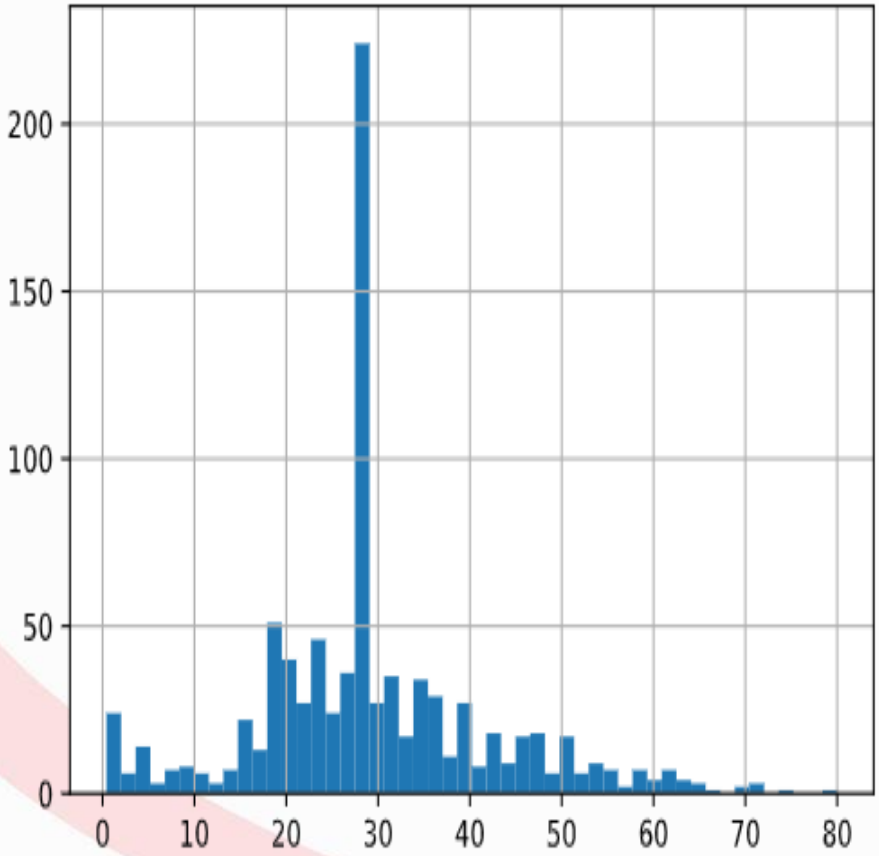


Thay thế bằng giá trị đuôi của phân bố (tt)

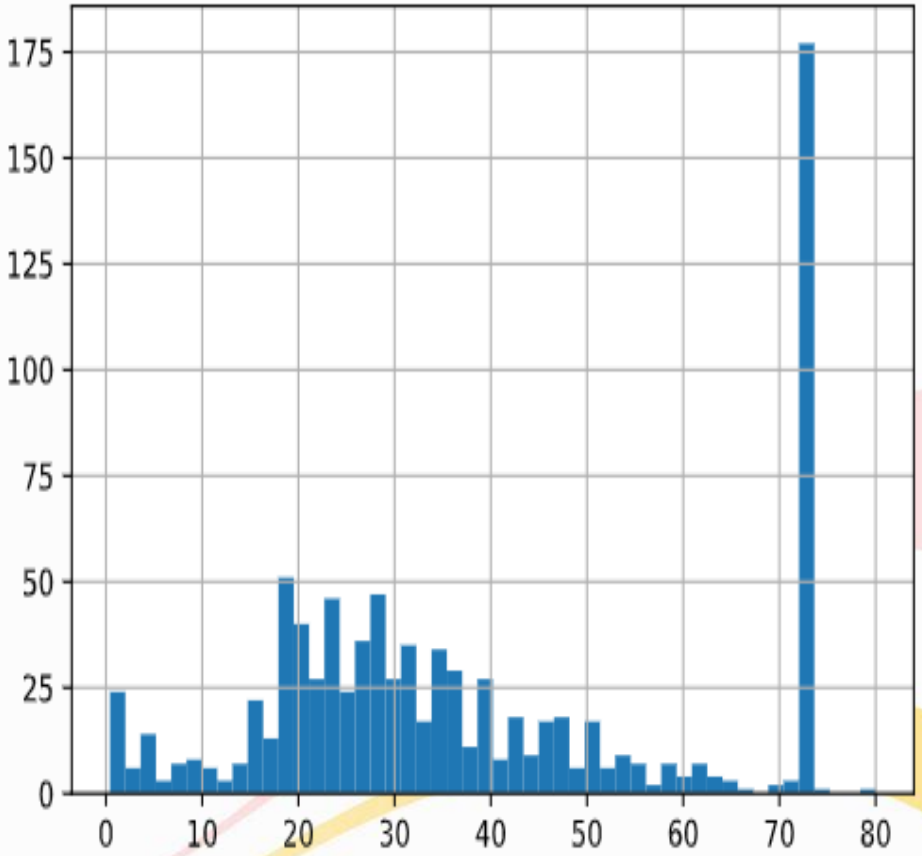
```
# định nghia hàm thay thế dữ liệu theo 2 cách:
# Median & End of Distribution
def impute_nan(df,variable,median,extreme):
  df[variable+"_end_dist"]=df[variable].fillna(extreme)
  df[variable].fillna(median,inplace=True)

# gọi hàm thay thế dữ liệu
impute_nan(df,'Age',df.Age.median(),extreme)
```

Thay thế bằng giá trị đuôi của phân bố (tt)

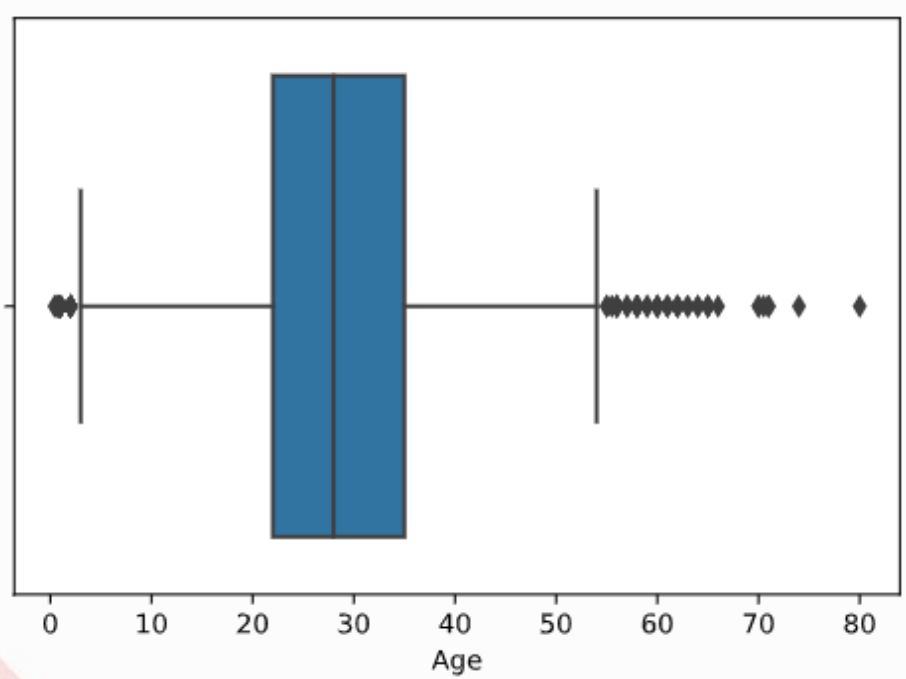


Histogram của biến Age sau khi thay NaN bằng Median

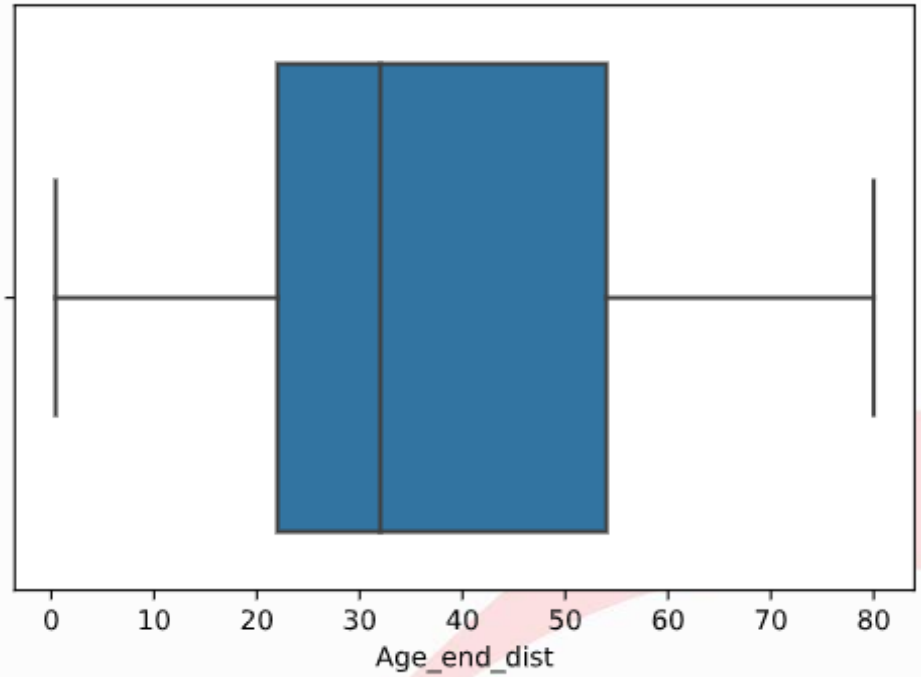


Histogram của biến Age sau khi thay NaN bằng đuôi của phân bố

Thay thế bằng giá trị đuôi của phân bố (tt)



Boxplot của biến Age sau khi thay NaN bằng Median



Boxplot của biến Age sau khi thay NaN bằng đuôi của phân bố

Thay thế bằng giá trị đuôi của phân bố (tt)

- Ưu điểm:
  - Dễ thực hiện
  - Nắm bắt được sự quan trọng của dữ liệu trống (nếu có nghi vấn)
- Nhược điểm:
  - Có thể làm méo mó phân bố dữ liệu của biến
  - Nếu sự khuyết dữ liệu không quan trọng → kỹ thuật này làm giảm năng lực dự báo (của mô hình)
  - Nếu số lượng dữ liệu trống là lớn → kỹ thuật này làm che giấu các giá trị ngoại lệ thực sự
  - Nếu số lượng dữ liệu trống là nhỏ → kỹ thuật này tạo ra một giá trị ngoại lệ ngoài dự tính

Tổng kết

- Mỗi kỹ thuật xử lý dữ liệu trống có ưu/nhược điểm riêng
- Cần lựa chọn kỹ thuật hợp lý dựa trên tính chất của tập dữ liệu và tính chất của dữ liệu trống (ngẫu nhiên/không ngẫu nhiên)

Làm sạch dữ liệu

Xử lý dữ liệu ngoại lệ (outliers)

Xử lý dữ liệu ngoại lệ

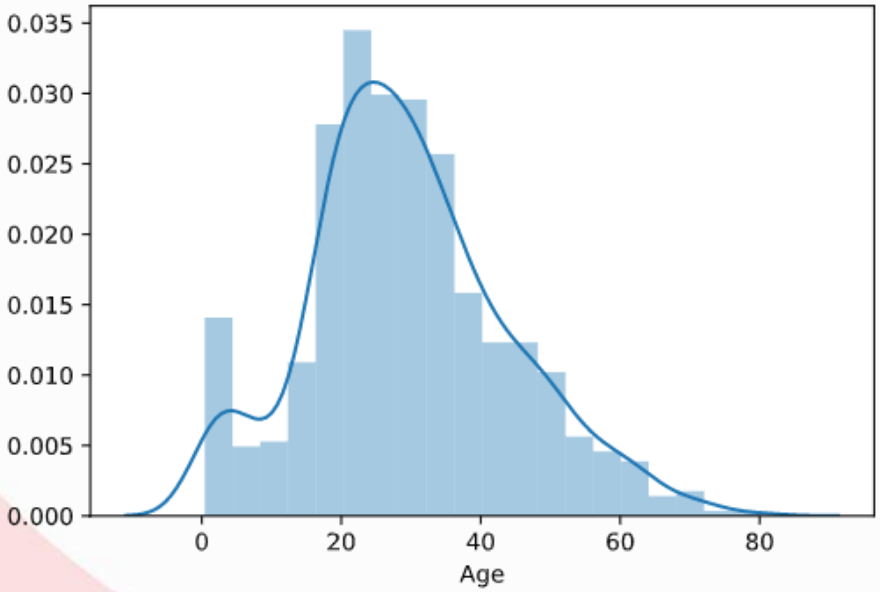
- Dữ liệu ngoại lệ là (các) giá trị quá lớn hoặc quá nhỏ so với các giá trị khác của tập dữ liệu
- Dữ liệu ngoại lệ tác động xấu (tuỳ mức độ) đến hiệu suất của các thuật toán học máy

Which Machine LEarning Models Are Sensitive To Outliers?

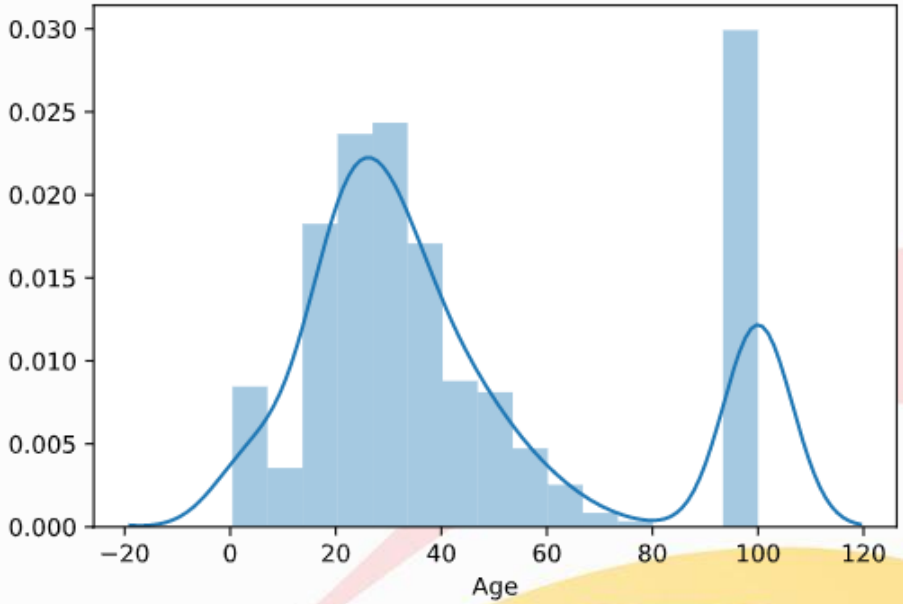
1. Naivye Bayes Classifier--- Not Sensitive To Outliers
2. SVM-------- Not Sensitive To Outliers
3. Linear Regression---------- Sensitive To Outliers
4. Logistic Regression------- Sensitive To Outliers
5. Decision Tree Regressor or Classifier---- Not Sensitive
6. Ensemble(RF,XGboost,GB)------- Not Sensitive
7. KNN------------------------- Not Sensitive
8. Kmeans---------------------- Sensitive
9. Hierarichal----------------- Sensitive
10. PCA------------------------ Sensitive
11. Neural Networks------------- Sensitive

Ví dụ giả định

Nếu ta thay thế các NaN trong cột Age bằng giá trị 100 thì sẽ làm thay đổi hoàn toàn hàm phân bố xác suất của dữ liệu



Trước khi thay thế



Sau khi thay thế

Xử lý dữ liệu ngoại lệ

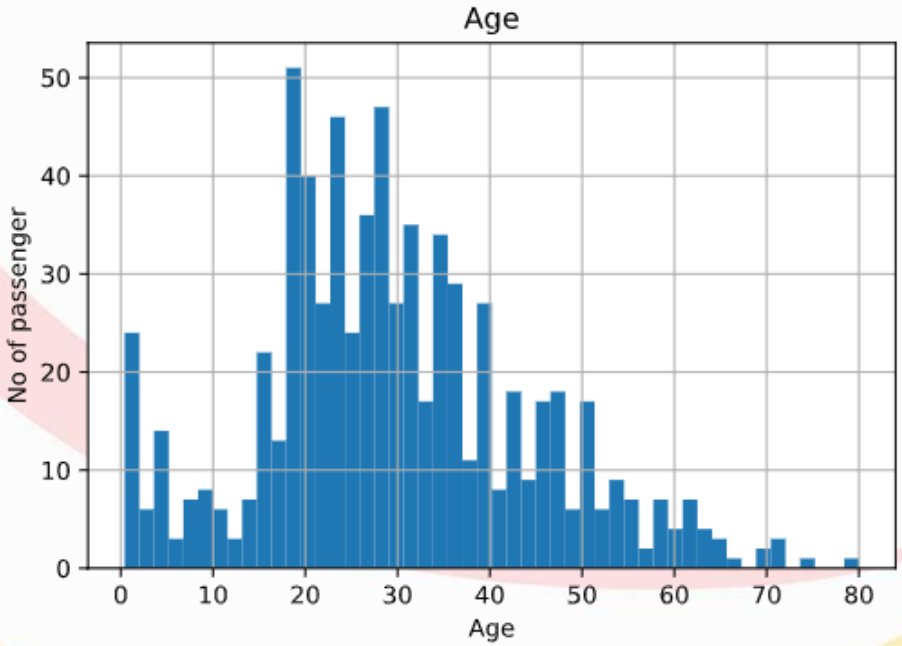
Giải pháp xử lý thông thường là:

- Xác định các giá trị biên trên và biên dưới của dữ liệu
- Thay thế giá trị ngoại lệ bằng 1 trong 2 giá trị trên

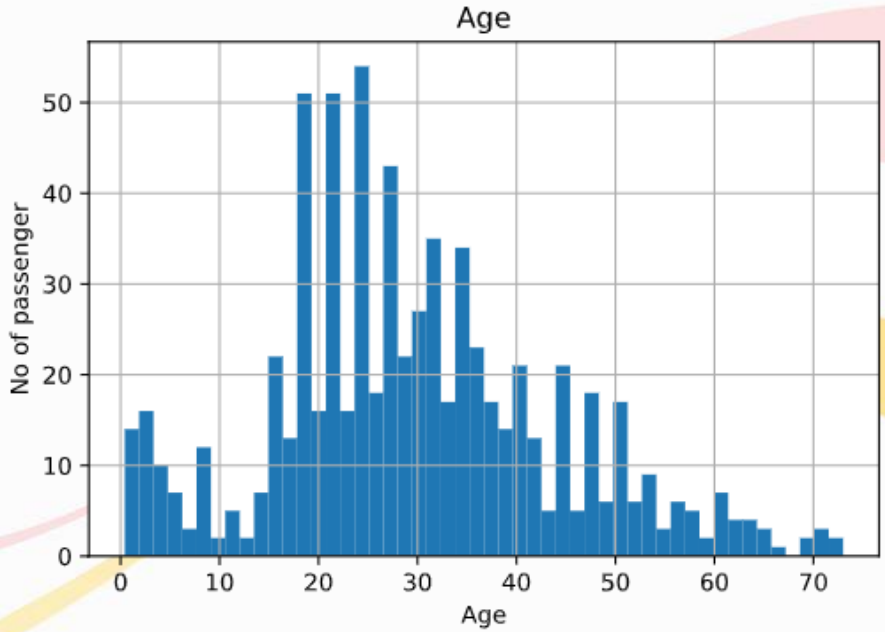
Xác định các giá trị biên

Dựa vào hàm phân bố xác suất của dữ liệu:

- **Nếu dữ liệu có dạng phân bố chuẩn:**
  - Biên trên = GTTB + 3*Độ lệch chuẩn
  - Biên dưới = GTTB - 3*Độ lệch chuẩn



Phân bố tuổi ban đầu



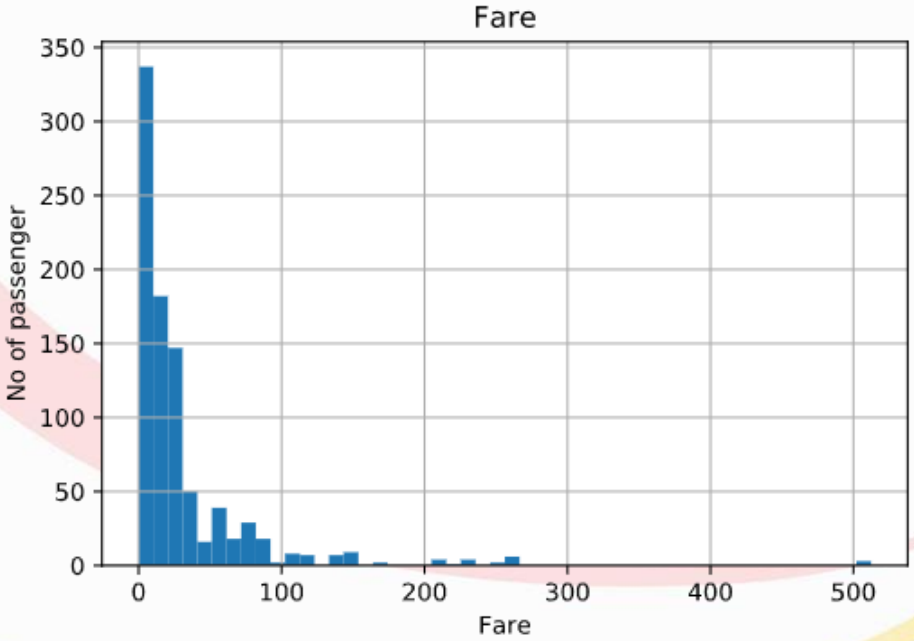
Phân bố tuổi sau khi thay thế các giá trị > biên trên cho 73

Xác định các giá trị biên

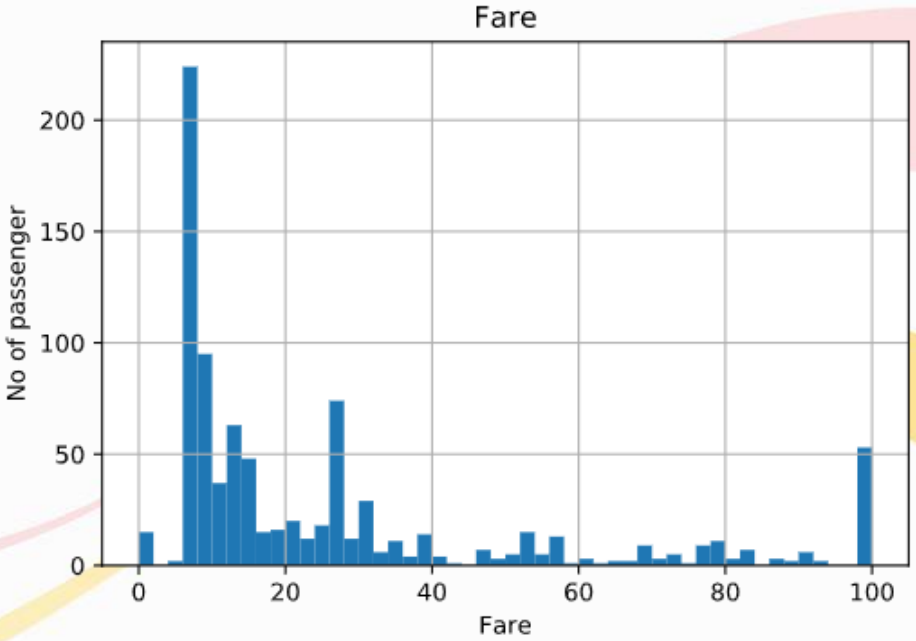
Dựa vào hàm phân bố xác suất của dữ liệu:

- **Nếu dữ liệu có dạng phân bố lệch (skewed):**
  - Biên trên = Quantile 75 + 3*IQR
  - Biên dưới = Quantile 25 - 3*IQR

(IQR: Interquantile range)



Phân bố giá vé ban đầu



Phân bố giá vé sau khi gán các giá trị > biên trên cho 100

Demo tác dụng của xử lý ngoại lệ

Demo thuật toán hồi quy logistic trong bài toán dự báo sự sống/chết (Survived) của hành khách lên tàu Titanic dựa trên Độ tuổi (Age) và Giá vé (Fare) mà họ đã mua trong 2 trường hợp:

- Không xử lý ngoại lệ
- Có xử lý ngoại lệ