AI VIET NAM
@aivietnam.edu.vn

AI VIET NAM – WARM-UP COURSE 2026

Tabular Data & Pandas

Nguyễn Quốc Thái Xỉn Quý Hùng Đinh Quang Vinh

I. Đề bài

Sử dụng data trong link này để hoàn thành các bài tập sau. Ta khởi tạo 1 pandas DataFrame như sau:

```
df = pd.read_csv("titanic_dataset.csv")
```

Câu 1: Cho biết có bao nhiêu cột chứa missing value?

A) 1

B) 2

C) 3

D) 4

Đáp án: C

Giải thích: Giả sử ta đã đọc file từ trước và lưu dưới dạng DataFrame trong pandas vào biến `df`. Có nhiều cách để kiểm tra những cột nào chứa missing values, một trong số đó là dòng lệnh:

```
df.isna().sum()
```

	0
PassengerId	0
Pclass	0
Name	0
Sex	0
Age	86
SibSp	0
Parch	0
Ticket	0
Fare	1
Cabin	327
Embarked	0

Hình 1: Kết quả sau khi chạy dòng lệnh câu 1

Ta quan sát kết quả, và thấy rằng cột **Age**, **Fare** và **Cabin** chứa missing values. Do đó, ta có tổng cộng 3 cột.

Câu 2: Chuyện gì xảy ra với df sau khi thực hiện đoạn code sau?

```
df["Age"] = df["Age"].fillna(df["Age"].mean())

df.drop(columns="Cabin")
```

A) Các missing value trong cột `Age` được thay bằng giá trị mean của cột `Age`, và cột `Cabin` bị xóa

B) Chương trình báo lỗi

C) Các missing value trong cột `Age` được thay bằng giá trị mean của cột `Age`

D) Không có chuyện gì xảy ra

Đáp án: C

Giải thích: Dòng đầu tiên của chương trình sẽ thay các giá trị missing trong cột `Age` bằng giá trị mean. Tuy nhiên, dòng lệnh thứ 2 sau khi thực thi sẽ không có chuyện gì xảy ra cả. Lý do là vì hàm drop sẽ không gán trực tiếp và `df`. Để có thay đổi xảy ra, ta có thể thực hiện dòng lệnh sau:

```
df = df.drop(columns="Cabin")
```

Câu 3: Để tìm những hành khách Nam lớn hơn 40 tuổi, đâu là đoạn code đúng?

A)

```
df[(df["Sex"] == "male") && (df["Age"] > 40)]
```

B)

```
df[(df["Sex"] == "male") and (df["Age"] > 40)]
```

C)

```
df[(df["Sex"] == "male") & (df["Age"] > 40)]
```

D) Cả 3 đáp án đều đúng

Đáp án: C

Giải thích: Chỉ duy nhất câu C là đúng syntax. Các câu khác nếu chạy sẽ khiến chương trình báo lỗi

Câu 4: Hãy cho biết có bao nhiêu hành khách nữ ở mỗi pClass? Đáp án ghi theo giá trị pClass tăng dần (1, 2, 3,...)

A) 50, 30, 72

B) 25, 55, 72

C) 30, 0, 123

D) 43, 45, 10

Đáp án: A

Giải thích: Ta chạy đoạn code sau để có đáp án

```
df[df["Sex"] == "female"]["Pclass"].value_counts().sort_index()
```

Câu 5: Có bao nhiêu hành khách đi cùng nhiều hơn 3 người (SibSp + Parch)?

A) 30

B) 40

C) 20

D) 50

Đáp án: C

Giải thích: Ta chạy đoạn code sau để có đáp án

```
len(df[(df["SibSp"] + df["Parch"]) > 3])
```

Câu 6: Cho biết tiền vé (Fare) trung bình của các hành khách đi một mình (Làm tròn 3 chữ số thập phân)

A) 40.321

B) 17.211

C) 22.863

D) 19.534

Đáp án: C

Giải thích: Ta chạy lệnh sau để có đáp án

```
round(df[(df["SibSp"] == 0) & (df["Parch"] == 0)]["Fare"].mean(), 3)
```

Câu 7: Đoạn code sau thực hiện điều gì?

```
df.loc[~(df["Embarked"] == "S"), "PassengerId":"Name"]
```

A) Lấy `PassengerId` và `Name` của các hành khách khởi hành ở `S`

B) Lấy `PassengerId`, `Pclass` và `Name` của các hành khách khởi hành ở `S`

C) Lấy `PassengerId` và `Name` của các hành khách không khởi hành ở `S`

D) Lấy `PassengerId`, `Pclass` và `Name` của các hành khách không khởi hành ở `S`

Đáp án: D

Giải thích:

- `~(df["Embarked"] == "S")`: Trả về các hành khách không đi cảng S
- `"PassengerId":"Name"`: Lấy tất cả các cột từ `PassengerId` đến `Name`. Do đó, nó sẽ bao gồm các cột `PassengerId`, `Pclass` và `Name`

Câu 8: Đâu là đoạn code tương tự với đoạn code này?

```
df.iloc[::2, 0:3]
```

A)

```
df.loc[::2, "PassengerId":"Name"]
```

B)

```
df.loc[::2, "PassengerId":"Sex"]
```

C)

```
df.loc[::2, 0:3]
```

D)

```
df.loc[::2, ("PassengerId", "Pclass", "Name")]
```

Đáp án: A

Giải thích: Ở hàm trên, cú pháp '0:3' sẽ lấy các giá trị từ PassengerId đến Name. Khác với loc, iloc không lấy giá trị cuối cùng, do đó đáp án B là sai. Ngoài ra, câu C và D đều sai cú pháp. Ta sẽ chọn đáp án A

Câu 9: Tạo ra cột 'Log_fare' bằng cách sử dụng hàm np.log1p của numpy. Hãy cho biết giá trị trung bình của cột 'Log_fare' (Làm tròn 3 chữ số thập phân)

A) 0.093

B) 2.018

C) 5.123

D) 3.016

Đáp án: D

Giải thích: Ta chạy đoạn code sau để có đáp án

```
import numpy as np

df["Log_fare"] = np.log1p(df["Fare"])

df["Log_fare"].mean()
```

Câu 10: Ta để ý rằng tên của các hành khách đều tuân theo 1 format nào đó và bao gồm danh xưng của họ. Chẳng hạn, hành khách có ID 892 có tên là 'Kelly, Mr. James', và danh xưng sẽ là 'Mr'. Hãy tạo ra một cột mới có tên là 'Title' trích xuất các danh xưng của họ và cho biết có bao nhiêu danh xưng khác nhau trong dữ liệu?

A) 6

B) 7

C) 8

D) 9

Đáp án: D

Giải thích: Ta chạy lệnh sau để có đáp án. Ý tưởng là ta tận dụng thông tin về dấu ',' và dấu '.' và sử dụng hàm `apply`: Với mỗi giá trị x của `Name`, ta tách tên theo dấu ',' rồi lấy giá trị thứ 2, sau đó tiếp tục tách theo dấu '.' rồi lấy giá trị đầu tiên. Ví dụ, với `'Kelly, Mr. James'`, ta tách ra thành `'Kelly'` và `'Mr. James'`. Ta tiếp tục tách `'Mr. James'` thành `'Mr'` và `'James'`. Cuối cùng, ta giữ giá trị `Mr`

```
df["Title"] = df["Name"].apply(lambda x: x.split(",")[1].split(".")[0].strip())
df["Title"].value_counts()
```