ยินดีต้อนรับสู่ Colaboratory

ไฟล์ แก้ไข มุมมอง แทรก รันไทม์ เครื่องมือ ความช่วยเหลือ บันทึกการเปลี่ยนแปลงไม่ได้

แชร์

สารบัญ

+ ส่วน

+ โค้ด + ข้อความ | คัดลอกไปยังไดรฟ์

RAM ดิสก์ Gemini

ดับเบิลคลิก (หรือกด Enter) เพื่อแก้ไข

```
[1] # Install and Import ydata-profiling (Auro EDA tools)
    !pip install ydata-profiling  #ติดตั้ง ydata-profiling
    from ydata_profiling import ProfileReport  #import ProfileReport จาก package ชื่อ ydata_profiling
```

แสดงเอาต์พุตที่ซ่อนอยู่

```
[2] import numpy as np  #import numpy โดยหลังจากนี้เรียกใช้โดย np (lib สำหรับ คำนวณ)
    import tensorflow as tf  #import tensorflow โดยหลังจากนี้เรียกใช้โดย tf (lib สำหรับ machine learning)
    from tensorflow import keras  #import keras จาก package ชื่อ tensorflow (lib สำหรับ deep learning)
    import pandas as pd  #import pandas โดยหลังจากนี้เรียกใช้โดย pd (lib สำหรับ อ่าน csv)
    from matplotlib import pyplot as plt  #import pyplot จาก package ชื่อ matplotlib (lib สำหรับ สร้างกราฟ)
    %matplotlib inline  #แสดงกราฟทันทีที่สร้าง
```

```
[3] from google.colab import files  #import file จาก google.colab (module ของ google.colab)
    uploaded = files.upload()  #เปิดให้เรานำ file มาวาง
```

แสดงเอาต์พุตที่ซ่อนอยู่

```
[4] df = pd. read_csv("/content/xxx_dataset.csv")  #อ่านข้อมูลและเก็บใส่ตัวแปรที่ชื่อว่า df
```

```
# Generate the Profiling Report
profile = ProfileReport(
    df, title="xxx_dataset", html={ 'style': {'full_width': True}}, sort=None
)  #นำข้อมูล df มาใส่ form ที่ set ไว้ และเก็บค่าใน profile
```

```
[9] # The HTML report in an iframe
    profile.to_notebook_iframe() #นำข้อมูล profile มาแสดงใน iframe
```

Summarize dataset: 100% 29/29 [00:03<00:00, 3.81it/s, Completed]

Generate report structure: 100% 1/1 [00:07<00:00, 7.26s/it]

Render HTML: 100% 1/1 [00:01<00:00, 1.46s/it]

xxx_dataset

Overview Variables Interactions Correlations Missing values Sample

Overview

Brought to you by YData

Overview | Alerts 5 | Reproduction

Dataset statistics

Number of variables	11
Number of observations	400
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	34.5 KiB
Average record size in memory	88.3 B

Variable types

Categorical	7
Numeric	3
Boolean	1

Variables

Select Columns

```
[10] df.shape  #เป็น method สั่งเพื่อให้ return row,column
```

```
(400, 11)
```

```
[11] df.columns  #เป็น method สั่งเพื่อให้ return ชื่อ column ทั้งหมด
```

```
Index(['gender', 'age', 'hypertension', 'heart_disease', 'Marriage',
       'work_type', 'Living_type', 'avg_glucose', 'bmi', 'smoking_status',
       'illness'],
      dtype='object')
```

```
[12] df.isnull().any()  #เป็น method สั่งเพื่อให้ return ชื่อ column พร้อมตรวจสอบว่ามี column ใดไม่มีค่าไหม(ไม่มีค่าหรือ NaN) หากไม่มีจะ return true
```

```
                  0
```

gender	False
age	False
hypertension	False
heart_disease	False
Marriage	False
work_type	False
Living_type	False
avg_glucose	False
bmi	False
smoking_status	False
illness	False

dtype: bool

```
df.nunique()  #เป็น method สั่งเพื่อให้ return ชื่อ column ทั้งหมดที่ค่าไม่ซ้ำกันเลย
```

	0
gender	2
age	79
hypertension	2
heart_disease	2
Marriage	2
work_type	4
Living_type	2
avg_glucose	393
bmi	199
smoking_status	4
illness	2

dtype: int64

```
[14] df.describe(include = 'all')  #แสดงข้อมูลทั้งหมดที่ pandas วิเคราะมาโดยไม่สนว่ามีประโยชน์ไหม
```

	gender	age	hypertension	heart_disease	Marriage	work_type	Living_type	avg_glucose	bmi	smoking_status	illness
count	400	400.00000	400.000000	400.000000	400	400	400	400.000000	400.000000	400	400.000000
unique	2	NaN	NaN	NaN	2	4	2	NaN	NaN	4	NaN
top	Female	NaN	NaN	NaN	Yes	Private	Urban	NaN	NaN	never smoked	NaN
freq	214	NaN	NaN	NaN	306	231	201	NaN	NaN	164	NaN
mean	NaN	55.26780	0.180000	0.132500	NaN	NaN	NaN	119.391950	29.481750	NaN	0.500000
std	NaN	22.51279	0.384669	0.339458	NaN	NaN	NaN	54.377459	6.488354	NaN	0.500626
min	NaN	0.80000	0.000000	0.000000	NaN	NaN	NaN	56.070000	15.600000	NaN	0.000000
25%	NaN	44.00000	0.000000	0.000000	NaN	NaN	NaN	80.460000	25.575000	NaN	0.000000
50%	NaN	59.00000	0.000000	0.000000	NaN	NaN	NaN	97.665000	28.600000	NaN	0.500000
75%	NaN	74.25000	0.000000	0.000000	NaN	NaN	NaN	144.345000	33.025000	NaN	1.000000
max	NaN	82.00000	1.000000	1.000000	NaN	NaN	NaN	271.740000	48.900000	NaN	1.000000

```
df['gender'] = df['gender'].map({'Female': 1, 'Male': 0})
df['Marriage'] = df['Marriage'].map({'Yes': 1, 'No': 0})
df['work_type'] = df['work_type'].map({'Private': 0, 'Self-employed': 1, 'Govt_job': 2, 'children': 3})
df['Living_type'] = df['Living_type'].map({'Urban': 1, 'Rural': 0})
df['smoking_status'] = df['smoking_status'].map({'formerly smoked': 0, 'never smoked': 1, 'smokes': 2, 'Unknown': 3})
#แปลงค่าข้อมูลที่ระบุไว้
```

```
[17] df  #แสดงข้อมูลใน df
```

	gender	age	hypertension	heart_disease	Marriage	work_type	Living_type	avg_glucose	bmi	smoking_status	illness
0	NaN	67.0	0	1	NaN	NaN	NaN	228.69	36.6	NaN	1
1	NaN	80.0	0	1	NaN	NaN	NaN	105.92	32.5	NaN	1
2	NaN	49.0	0	0	NaN	NaN	NaN	171.23	34.4	NaN	1
3	NaN	79.0	1	0	NaN	NaN	NaN	174.12	24.0	NaN	1
4	NaN	81.0	0	0	NaN	NaN	NaN	186.21	29.0	NaN	1
...	...	...	...	...	...	...	...	...	...	...	...
395	NaN	54.0	0	1	NaN	NaN	NaN	140.28	37.1	NaN	0
396	NaN	67.0	0	0	NaN	NaN	NaN	244.28	29.4	NaN	0
397	NaN	53.0	0	0	NaN	NaN	NaN	124.16	31.7	NaN	0
398	NaN	47.0	0	0	NaN	NaN	NaN	93.55	31.4	NaN	0
399	NaN	44.0	0	0	NaN	NaN	NaN	99.34	33.1	NaN	0

400 rows × 11 columns

ขั้นตอนถัดไป: สร้างโค้ดด้วย df | ดูแผนภูมิที่แนะนำ | New interactive sheet

```
y = df['illness']  #นำข้อมูล illness มาเก็บใส่ตัวแปล y
X = df.drop(columns=['illness'])  #ลบข้อมูล illness ทิ้ง และเก็บข้อมูลที่เหลือทั้งหมดในตัวแปล y
```

```
[21] X.head()  #นำข้อมูลมาแสดง 5 row (5 เป็นค่า default)
```

	gender	age	hypertension	heart_disease	Marriage	work_type	Living_type	avg_glucose	bmi	smoking_status
0	NaN	67.0	0	1	NaN	NaN	NaN	228.69	36.6	NaN
1	NaN	80.0	0	1	NaN	NaN	NaN	105.92	32.5	NaN
2	NaN	49.0	0	0	NaN	NaN	NaN	171.23	34.4	NaN
3	NaN	79.0	1	0	NaN	NaN	NaN	174.12	24.0	NaN
4	NaN	81.0	0	0	NaN	NaN	NaN	186.21	29.0	NaN

```
[22] y.value_counts()  #นับค่าว่าเจอค่าดังกล่าวกี่ครั้ง
```

illness	count
1	200
0	200

dtype: int64

```
[25] รหัสที่สร้างอาจขึ้นอยู่กับใบอนุญาต | adityamanglik/Algorithm-Implementations | | 001ashwin/Recommendation-system-for-Dyslexic-Students
from sklearn. preprocessing import MinMaxScaler  #import MinMaxScaler จาก package sklearn. preprocessing
scaler = MinMaxScaler()  #สร้างตัวแปล scaler มาเก็บค่า 0 1
X_scaled=pd.DataFrame(scaler.fit_transform(X),columns=X.columns)  #สร้างตัวแปล scaler มาเก็บค่า 0 1 ที่เป็น DataFrame
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/_array_api.py:769: RuntimeWarning: All-NaN slice encountered
  return xp.asarray(numpy.nanmin(X, axis=axis))
/usr/local/lib/python3.10/dist-packages/sklearn/utils/_array_api.py:786: RuntimeWarning: All-NaN slice encountered
  return xp.asarray(numpy.nanmax(X, axis=axis))
```

```
X_train, x_test, y_train, y_test=train_test_split(X_scaled, y, test_size=0.30, random_state=42)
#นำ X_scaled มาหา illness โดยแบ่งอัตราส่วน เป็น 0.7 กับ 0.3 สำหรับตอนแทนค่า และมี seed ในการ random ค่าเป็น 42(seed เดิม = random ได้ค่าเดิม)
```

gender: Nominal
age: Continuous
hypertension: Nominal (Boolean)
heart_disease: Nominal (Boolean)
Marriage: Nominal
work_type: Nominal
Living_type: Nominal avg_glucose: Discrete
bmi: Discrete
smoking_status: Ordinal
illness: Nominal