Data Prep.ipynb

File Edit View Insert Runtime Tools Help Saving...

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

```
[2]
import numpy as  np
import tensorflow as tf
from tensorflow import keras
import pandas as pd
from matplotlib import pyplot  as plt
%matplotlib inline
```

Upload you data / อัพโหลดไฟล์ข้อมูล

```
[3]
from google.colab import files
uploaded = files.upload() # download CSV file and upload to your colab
```

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

Auto EDA / ใช้ Library ydata-profiling เพื่อทำ EDA อัตโนมัติ

```
[5]
# Generate the Profiling Report
profile = ProfileReport(
    df, title="xxx_dataset", html={"style": {"full_width": True}}, sort=None
)
```

```
[6]
# The HTML report in an iframe
profile.to_notebook_iframe()
```

```
Summarize dataset: 100% 29/29 [00:07<00:00,  2.13it/s, Completed]
Generate report structure: 100% 1/1 [00:05<00:00,  5.59s/it]
Render HTML: 100% 1/1 [00:00<00:00,  1.21it/s]
```

xxx_dataset — Overview | Variables | Interactions | Missing values | Sample

Overview

Overview | Alerts 1 | 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

Data Preparation

```
[7]
df.shape
```

```
(400, 11)
```

```
[8]
df.columns
```

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

```
[9]
df.isnull().any()
```

```
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
```

[10]
```python
df.nunique()
```

```
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
```

[11]
```python
df.describe(include = 'all')
```

	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

[12]
```python
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})
```

[13]
```python
df
```

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

400 rows × 11 columns

Next steps: View recommended plots

[14]
```python
y = df ['illness']
x = df.drop(columns=['illness'])
```

[15]
```python
x.head()
```

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

Next steps: View recommended plots

[16]
```python
y.value_counts()
```

```
illness
1    200
0    200
Name: count, dtype: int64
```

[17]
```python
from sklearn.preprocessing import MinMaxScaler
from sklearn. preprocessing import MinMaxScaler

scale=MinMaxScaler()
x_scaled=pd.DataFrame(scale.fit_transform(x),columns=x.columns)
x_scaled
```

	gender	age	hypertension	heart_disease	Marriage	work_type	Living_type	avg_glucose	bmi	smoking_status
0	0.0	0.815271	0.0	1.0	1.0	0.000000	1.0	0.800389	0.630631	0.000000
1	0.0	0.975369	0.0	1.0	1.0	0.000000	0.0	0.231140	0.507508	0.333333
2	1.0	0.593596	0.0	0.0	1.0	0.000000	1.0	0.533964	0.564565	0.666667
3	1.0	0.963054	1.0	0.0	1.0	0.333333	0.0	0.547364	0.252252	0.333333
4	0.0	0.987685	0.0	0.0	1.0	0.000000	1.0	0.603422	0.402402	0.000000
...	...	...	...	...	...	...	...	...	...	...
395	1.0	0.655172	0.0	1.0	1.0	0.000000	1.0	0.390458	0.645646	0.000000
396	0.0	0.815271	0.0	0.0	1.0	0.666667	0.0	0.872676	0.414414	0.000000
397	0.0	0.642857	0.0	0.0	1.0	0.000000	1.0	0.315714	0.483483	0.333333
398	0.0	0.568966	0.0	0.0	1.0	0.000000	0.0	0.173784	0.474474	0.333333
399	0.0	0.532020	0.0	0.0	1.0	0.000000	1.0	0.200631	0.525526	0.333333

400 rows × 10 columns

Next steps: View recommended plots

Split Dataset into *Trainnig set* and *Testing set*

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x_scaled, y, test_size=0.30, random_state=42)
```