Homework 1

API : Covid-19 cases by province (Week Number 48 : 28/11/2022-04/12/2022)

```
import requests
import time
import pandas as pd
```

```
url = 'https://covid19.ddc.moph.go.th/api/Cases/today-cases-by-provinces'
response = requests.get(url)
```

```
response.status_code
```

```
200
```

```
result = response.json()
df = pd.DataFrame.from_dict(result)
df.head(n = 10)
```

	year	weeknum	province	new_case	total_case	new_case_excludeabroad	total
0	2022	49	ปราจีนบุรี	5	52605	5	
1	2022	49	ราชบุรี	40	87263	40	
2	2022	49	ศรีสะเกษ	6	50754	6	
3	2022	49	นครศรีธรรมราช	75	129160	75	
4	2022	49	นครพนม	28	19367	28	
5	2022	49	ตราด	43	19288	43	

```
# count the number of rows and columns
rows = len(df.axes[0])
cols = len(df.axes[1])

print(f"Rows = {rows}, Columns = {cols} ")
```

```
Rows = 79, Columns = 10
```

Homework 2

ML model using sklearn :

```
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
import pandas as pd
import numpy as np
```

```
#df2 from https://www.kaggle.com/datasets/saurabh00007/diabetescsv/download?datasetVersionNumber=1
df2 = pd.read_csv('diabetes.csv')
df2.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigre
0	6	148	72	35	0	33.6	
1	1	85	66	29	0	26.6	
2	8	183	64	0	0	23.3	
3	1	89	66	23	94	28.1	
4	0	137	40	35	168	43.1	

```
# check null in each column
df2.isna().sum()
```

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

```python
# preview data types
df2.dtypes
```

```
Pregnancies                   int64
Glucose                       int64
BloodPressure                 int64
SkinThickness                 int64
Insulin                       int64
BMI                         float64
DiabetesPedigreeFunction    float64
Age                           int64
Outcome                       int64
dtype: object
```

```python
# prepare & split data
x = df2.drop('Outcome' , axis = 1)
y = df2['Outcome']

x_tra, x_tes, y_tra, y_tes = train_test_split(
    x, y , test_size = 0.25, random_state = 42
)
```

```python
# train model
model = LogisticRegression()
model.fit(x_tra, y_tra)

# test model
p = model.predict(x_tes)
print(p)
```

```
[0 0 0 0 0 0 0 1 1 1 0 1 0 0 0 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 1 1 1 1 1 1 1
 0 1 1 0 1 1 0 0 1 1 0 0 1 0 1 1 0 0 0 1 0 0 1 1 0 0 0 0 1 0 1 0 1 1 0 0 0
 0 1 0 0 0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1 0 0 1 0 1 0
 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 1 1 0 0 0 0 0 0 0 0 0
 0 1 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 1 0 0 1 1 1 0 0 1 1 0 0 0 0 0 1 1 0 1 1
 0 0 0 1 0 0 0]
/usr/local/lib/python3.8/dist-packages/sklearn/linear_model/_logistic.py:814: ConvergenceWarning: lbfgs failed to converge (status=
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  n_iter_i = _check_optimize_result(
```

```python
# model evaluation
model.score(x_tes, y_tes)
```

```
0.7291666666666666
```