

Homework 1

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

```
1 import requests
2 import time
3 import pandas as pd
```

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

```
1 response.status_code

200
```

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

	year	weeknum	province	new_case	total_case	new_case_excludeabroad	total_c
0	2022	49	ปราชินบุรี	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	

```
1 # count the number of rows and columns
2 rows = len(df.axes[0])
3 cols = len(df.axes[1])
4
5 print(f"Rows = {rows}, Columns = {cols} ")

Rows = 79, Columns = 10
```

Homework 2

ML model using sklearn :

```
1 from sklearn.linear_model import LogisticRegression
2 from sklearn.model_selection import train_test_split
3 import pandas as pd
4 import numpy as np
```

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

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
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	

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

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

```
1 # preview data types
2 df2.dtypes
```

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

```
1 # prepare & split data
2 x = df2.drop('Outcome' , axis = 1)
3 y = df2['Outcome']
4
5 x_tra, x_tes, y_tra, y_tes = train_test_split(
6     x, y , test_size = 0.25, random_state = 42
7 )
```

```
1 # train model
2 model = LogisticRegression()
3 model.fit(x_tra, y_tra)
4
5 # test model
6 p = model.predict(x_tes)
7 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
 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 0 1 1 1 0 0 1 0 1 0 1 1 0 0 1 0 1 0
 0 0 1 0 0 1 0 0 1 0 0 0 0 0 0 0 0 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 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(

```
1 # model evaluation
2 model.score(x_tes, y_tes)
```

```
0.7291666666666666
```

