2023 年臺灣國際科學展覽會
優勝作品專輯

作品編號	190026
參展科別	電腦科學與資訊工程
作品名稱	**Predict the precise time that the sunset cloud appeared.**
得獎獎項	

就讀學校	高雄市立高雄高級中學
指導教師	盧政良、李麗偵
作者姓名	莊博丞、邱律銘、許祐嘉

關鍵詞　晚霞、雲層、爬蟲

作者簡介

Abstract

Sunset cloud is one of the amazing natural phenomena on Earth. There is a huge economic chance hidden in the back. Therefore, we want to create a system to Predict the accurate time that the sunset cloud appeared, in order to help the tourism industry in Taiwan.

In this study, a computational model will be proposed using Cauchy's equation, theorems related to refraction and reflection, and other auxiliary formulas contributed by the research paper. For the automation part, including temperature, pressure, and humidity, we use the government's open data platform and satellite open data for static web crawling. For the cloud height, we use dynamic web crawlers to crawl relevant data from AccuWeather website for calculation. We calculate the crawled data and the proposed model in a 15-minute interval to provide users with an accurate time to watch the sunset clouds.

By using this model and the results of the crawler's data acquisition calculations, we can obtain prediction results with nearly 90% accuracy. Therefore, we are able to provide users with the accurate time of the sunset clouds appeared.

摘要

雲彩是其中一個在世界上最奇妙的自然現象。在其中也隱藏著巨大的觀光經濟利益。因此，我們想要建立一個系統以預測晚霞雲彩出現的時間，以幫助台灣的觀光業。

本研究將藉由柯西公式、折射反射相關定理以及其他由論文貢獻的輔助公式提出一個計算模型，以計算預測晚霞雲彩出現的時間以及光的路徑。自動化的部分，包含溫度、壓力以及濕度，我們藉由政府的公開資料平台以及衛星公開資料進行靜態網頁爬蟲抓取。雲層高度我們則是透過動態網頁爬蟲，逐一從 AccuWeather 公開網站上爬取相關資料以利計算。我們將爬取的資料以及所提出的模型計算後以 15 分鐘作爲一個區隔，提供使用者準確的時間觀賞雲彩。

透過此模型以及爬蟲擷取資料計算得到的結果，我們可以得到接近 90%準確率的預測結果。因此，我們能夠準確地為用戶提供正確日落雲彩出現的時間。

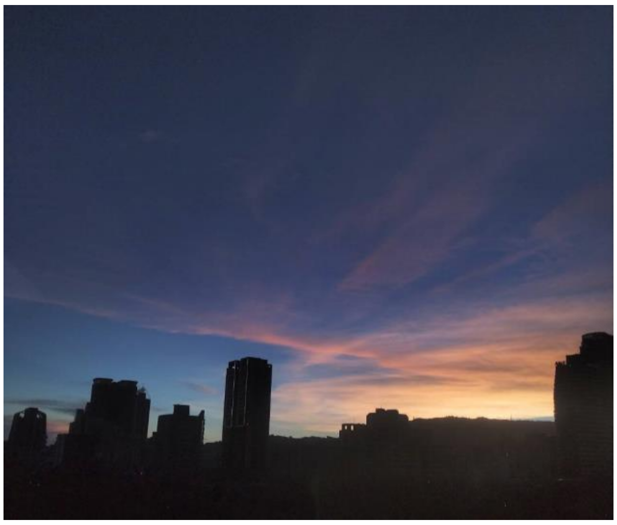
1. Introduction

1.1 Motivation

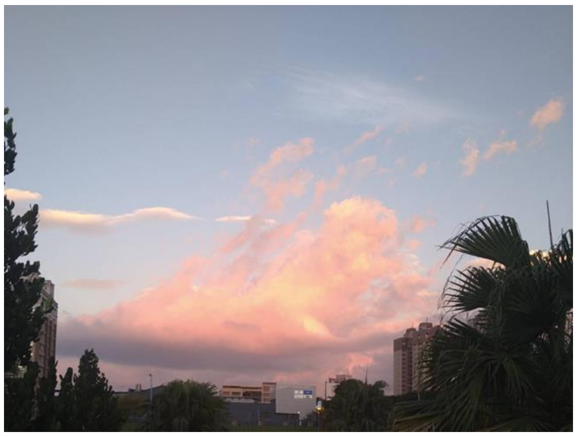
Taiwan is the most beautiful island in the world. Sunset cloud is also one of the most amazing phenomena taking place in Taiwan. Many tourists often search for a place in Taiwan to look for the beautiful sunset cloud including dating, family trip or even proposal. There is a huge economic chance hidden in the back. Based on the above reasons, we start to think of **how to help tourists watch the sunset cloud more conveniently and can also help the tourism industry which the pandemic had a huge impact on**. Therefore, we want to build a system to predict the accurate time that the sunset cloud would appear and can also predict whether the sunset cloud would appear. It could not only help tourists to choose the correct time and date to watch the phenomenon but also bring crowds to the merchant around the tourist attractions.

1.2 How the sunset cloud appears?

First, we have to know how the sunset cloud appeared. The light from the sun is scattered during traveling through the atmosphere, and the light with a longer wavelength, usually red light, has a greater refraction angle. Therefore, the different wavelengths of light will reflect from the different horizontal levels of the color cloud, which makes the color different. Here comes the picture (Picture 1-1) illustrating it. From the daily observation, we notice that color cloud usually appears after rain or typhoon, especially with highly water vapor pressure.

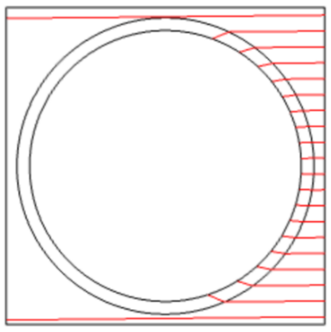


The picture above (Picture 1-1) is the sunset with some colorful clouds but without rich water vapor. The reason this scenery appears that we think is because of the height of the cloud, making the light can have enough scattered before being reflected.



▲Picture 1-2

The picture above (Picture 1-2), in contrast, is of the rich water vapor in the atmosphere. Therefore, a multi-color cloud could be created even if in a low-level atmosphere. This kind of cloud is cumulonimbus, which usually appears in the low-level atmosphere.
Note: this is pictured at 8/23, 18:30.



▲Picture 1-3

In the illustration (Picture 1-3) above, we can see the different positions on the Earth, which symbolize the different time of the day, and has a different light route, the red line. The different light routes are caused by the different refraction in the atmosphere. However, different refraction phenomena are not only due to the different time, but also the different atmosphere variants, such as air pressure, air temperature, humidity, and so on. Fortunately, we got access to the Formosa No.7 satellite, which contained the data we need, including the air pressure and temperature.

2 Theory

The theories we used can be categorized with light refraction, which both scatter and refraction are caused by the refraction index. Therefore, we use the following equation (Equation 1):

$$N_n \times sin\theta_n = N_{n+1} \times sin\theta_{n+1}$$

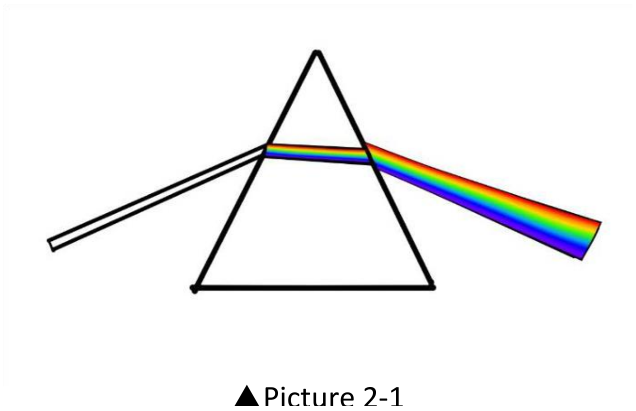
$$N_1 sin\theta_1 = N_2 sin\theta_2$$

▲Equation 1

where N_n is the current medium (level) refraction index and θ_n is the incident angle. Similarly, n+1 is the next medium (level). The equation is given by Fundamentals of Optics [4]. And the below are some detailed descriptions.

2.1 Light refraction and scatter

Light would refract through different media. However, different air densities or air molecules can be considered to be different medium, too. Therefore, the light refract in the atmosphere is the multi-scattering. Hence, we have to use integration to calculate it, which you can see it at equation 3. Besides, the density of atmosphere is an exponential distribution.



▲Picture 2-1

If the sunlight beam going through the triangle glass, it will produce a rainbow-like light phase, which is called scatter, and it is caused by the different wavelength. Or just like picture 2-1.

2.2 Refraction index

Since the atmosphere refraction index would be majority affected by the wavelength, our purpose as well, we use Cauchy Equation [3] to calculate our refraction index to decrease the miscalculation between different wavelengths. And the below is the Cauchy Equation (Equation 2):

$$n_{air}(\lambda,\, T,\, v,\, p) \approx 1 + \left(\frac{77.6 \cdot 10^{-6}}{T}\right)\left(1 + \frac{7.52 \cdot 10^{-3}}{\lambda^2}\right)\left(p + \frac{4810 \cdot v}{T}\right)$$

▲Equation 2

where λ is wavelength (µm), using 700 nm or 0.7 µm as red-light wavelength, T is temperature (Kelvin), p is air pressure (mbar), and v is the pressure of water vapor (mbar). However, we couldn't get the v directly, so we use the below equation [5]:

$$v = \frac{-1}{pg}\int_{p1}^{p2} M_p \, dp$$

▲Equation 3

where v is precipitable water vapor (cm), g is the acceleration due to the earth's gravity (986.665 cm/sec^2), ρ is the density of liquid water ($g \cdot cm^{-3}$), p is the atmospheric pressure along altitude ($mbar$), P_1 and P_2 are the atmospheric pressure at the surface and the top of the atmosphere respectively. M_p is mixing ratio at the pressure level, p . The integration is from the surface at P_1 up to the pressure designated by P_2 which depends on the final altitude reached by the radiosonde observation. M_p can be calculated by below (Equation 4):

$$Mp = \frac{0.622e}{p - e} \tag{4}$$

▲Equation 4

where e is the actual vapor pressure ($mbar$). The actual vapor pressure is obtained as the product of the saturated vapor pressure (e_s) and the relative humidity (RH) at pressure p , or as the equation (Equation 5) below:

$$e = \frac{RH \cdot e_x}{100} \tag{5}$$

▲Equation 5

The saturated vapor pressure value depends only on the air temperature T in degree Celsius. The saturated vapor pressure (in $mbar$) is calculated according to below(Equation 6) :

$$e_x = 6.112 \cdot \exp\left(\frac{17.67 \cdot T}{T + 243.5}\right) \tag{6}$$

▲Equation 6

After using the equation mentioned earlier, we can use the refraction index we calculated to simulate the routes in different light wavelengths, temperature, air pressure and humidity. See **(2.6) Simulation** for more details.

2.3 The relation between time and solar altitude

To compute solar altitude angle, we need the observer's latitude, time, and date of the year. After accessing the mentioned variants, we can compute the solar altitude angle by the following formula (Equation 7) [6]:

$$\theta = \sin^{-1}(sin\delta\ sin\phi + cos\delta\ cos\phi\ cos\omega) \tag{7}$$

▲Equation 7

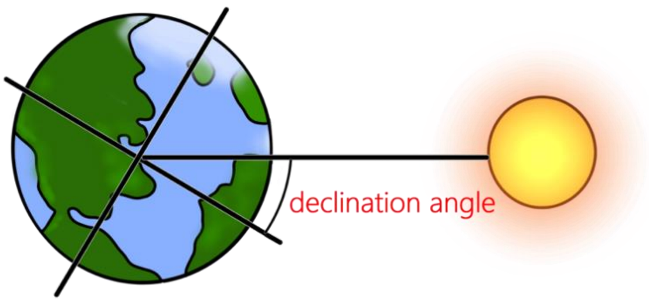
where δ is the declination angle, φ is the latitude angle, and ω is the hour angle.

The declination angle changing depends on seasons, due to the earth's revolution. It is the angle between the latitude of direct sunlight and the earth’s equator. The northern latitude to the earth’s equator is positive, and the opposite is negative. If the axis of the earth were not askew, the declination angle would not exist (the following picture). The declination angle depends on the date of the year(d) used in the following formula, not related to the observer’s location. For instance, d=1 means 1 of January.

$$\delta = -0.40928\ \cos\left(\frac{2\pi}{365(d+10)}\right) \tag{8}$$

▲Equation 8

Declination angle can transform the local solar time into sun’s moving angle,



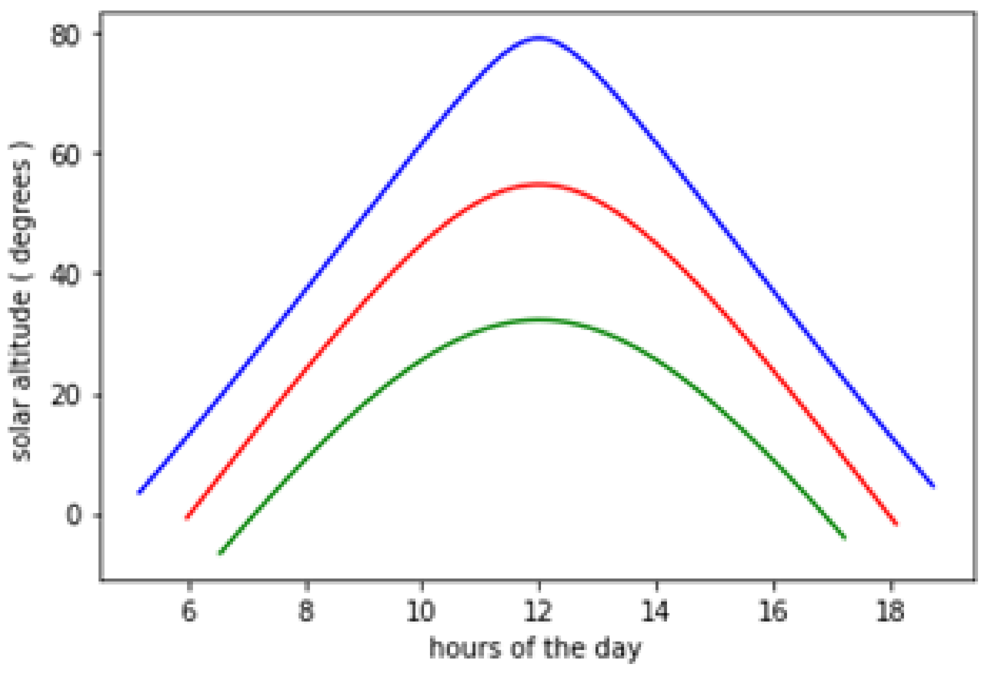
▲Picture 2-2

noted that sun travel 2π/24 rad. per hour. In the midday, declination angle is zero and therefore it can be calculated easily through the below equation, which t is the local time.

$$\omega = \frac{\pi}{12}(t-12) \tag{9}$$

▲Equation 9

By the above formulas, we could easily compute with Python, and draw the data as a linear graph. The following picture (Picture 2-3) is a linear graph when the observer is in 23.5°N .



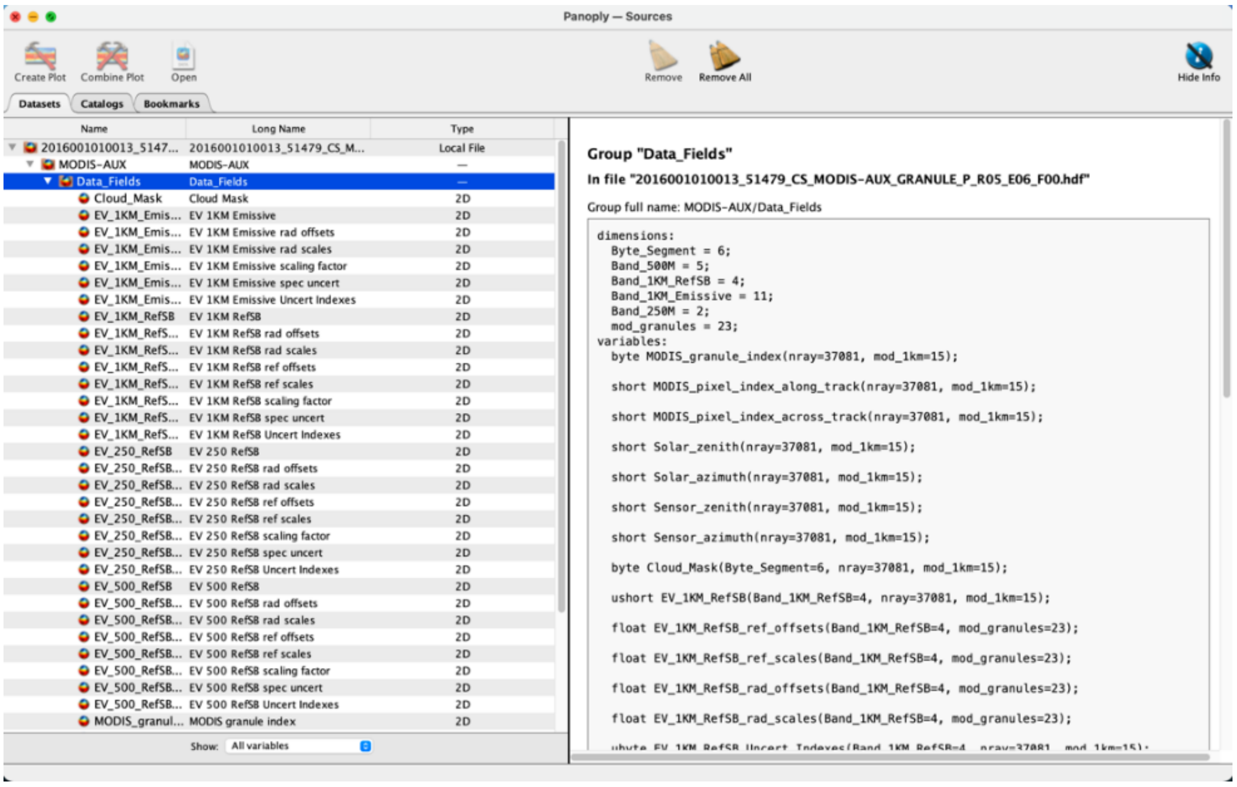
▲Picture 2-3

Where the horizontal axis is time of the day (hours), and the vertical axis is solar altitude angle computed by the above formulas (degrees). The blue line is the day of summer solstice; the rea one is the day of spring equinox; the green one is the day of winter solstice.

3. Approach

3.1 Panoply

In this research, we use a lot of data obtained from satellites or other open data platforms. These data were usually stored in some unfamiliar type, including NetCDF, HDF, and GRIB. Although these types can be read into Python, we should understand the framework of the data before coding. Panoply is the best application released by NASA for data preprocessing in this situation. It can read several data types, including most of the open data in astronomy and meteorology. Panoply is able to turn single data not only into plots but also export into CSV. [8]



Name	Long Name	Type
2016001010013_5147...	2016001010013_51479_CS_M...	Local File
MODIS-AUX	MODIS-AUX	—
Data_Fields	Data_Fields	—
Cloud_Mask	Cloud Mask	2D
EV_1KM_Emis...	EV 1KM Emissive	2D
EV_1KM_Emis...	EV 1KM Emissive rad offsets	2D
EV_1KM_Emis...	EV 1KM Emissive rad scales	2D
EV_1KM_Emis...	EV 1KM Emissive scaling factor	2D
EV_1KM_Emis...	EV 1KM Emissive spec uncert	2D
EV_1KM_Emis...	EV 1KM Emissive Uncert Indexes	2D
EV_1KM_RefSB	EV 1KM RefSB	2D
EV_1KM_RefS...	EV 1KM RefSB rad offsets	2D
EV_1KM_RefS...	EV 1KM RefSB rad scales	2D
EV_1KM_RefS...	EV 1KM RefSB ref offsets	2D
EV_1KM_RefS...	EV 1KM RefSB ref scales	2D
EV_1KM_RefS...	EV 1KM RefSB scaling factor	2D
EV_1KM_RefS...	EV 1KM RefSB spec uncert	2D
EV_1KM_RefS...	EV 1KM RefSB Uncert Indexes	2D
EV_250_RefSB	EV 250 RefSB	2D
EV_250_RefSB...	EV 250 RefSB rad offsets	2D
EV_250_RefSB...	EV 250 RefSB rad scales	2D
EV_250_RefSB...	EV 250 RefSB ref offsets	2D
EV_250_RefSB...	EV 250 RefSB ref scales	2D
EV_250_RefSB...	EV 250 RefSB scaling factor	2D
EV_250_RefSB...	EV 250 RefSB spec uncert	2D
EV_250_RefSB...	EV 250 RefSB Uncert Indexes	2D
EV_500_RefSB	EV 500 RefSB	2D
EV_500_RefSB...	EV 500 RefSB rad offsets	2D
EV_500_RefSB...	EV 500 RefSB rad scales	2D
EV_500_RefSB...	EV 500 RefSB ref offsets	2D
EV_500_RefSB...	EV 500 RefSB ref scales	2D
EV_500_RefSB...	EV 500 RefSB scaling factor	2D
EV_500_RefSB...	EV 500 RefSB spec uncert	2D
EV_500_RefSB...	EV 500 RefSB Uncert Indexes	2D
MODIS_granul...	MODIS granule index	2D

Group "Data_Fields"

In file "2016001010013_51479_CS_MODIS-AUX_GRANULE_P_R05_E06_F00.hdf"

Group full name: MODIS-AUX/Data_Fields

```
dimensions:
  Byte_Segment = 6;
  Band_500M = 5;
  Band_1KM_RefSB = 4;
  Band_1KM_Emissive = 11;
  Band_250M = 2;
  mod_granules = 23;
variables:
  byte MODIS_granule_index(nray=37081, mod_1km=15);

  short MODIS_pixel_index_along_track(nray=37081, mod_1km=15);

  short MODIS_pixel_index_across_track(nray=37081, mod_1km=15);

  short Solar_zenith(nray=37081, mod_1km=15);

  short Solar_azimuth(nray=37081, mod_1km=15);

  short Sensor_zenith(nray=37081, mod_1km=15);

  short Sensor_azimuth(nray=37081, mod_1km=15);

  byte Cloud_Mask(Byte_Segment=6, nray=37081, mod_1km=15);

  ushort EV_1KM_RefSB(Band_1KM_RefSB=4, nray=37081, mod_1km=15);

  float EV_1KM_RefSB_ref_offsets(Band_1KM_RefSB=4, mod_granules=23);

  float EV_1KM_RefSB_ref_scales(Band_1KM_RefSB=4, mod_granules=23);

  float EV_1KM_RefSB_rad_offsets(Band_1KM_RefSB=4, mod_granules=23);

  float EV_1KM_RefSB_rad_scales(Band_1KM_RefSB=4, mod_granules=23);

  ubyte EV_1KM_RefSB_Uncert_Indexes(Band_1KM_RefSB=4, nray=37081, mod_1km=15);
```

▲Picture 3-1 The interface of Panoply

3.2 Web Crawling

In this research, in order to obtain all the data we need, we used a lot of web crawling skills with python. Each site has its own most suitable crawling method. Here are the differences between web crawling in different types of pages and the method we choose to deal with.

	Speed	Best Module to Use	Written Language
Dynamic Pages	Slow	Wget, Requests, Beautifulsoup	Javascript
Static Pages	Fast	Selenium	HTML

▲the differences between web crawling in different types of pages

In static pages, we use Requests and Beautifulsoup modules to get the contents of the websites. In the content we got, we could usually locate the information we need by its tag. Some websites provide the download link we need so that we could right download the data. However, some websites didn't provide the link we want. Instead, they provide some important information on their website. The information they provide could combine to become the download link we want. (See **3.4 Data Access – Pressure and Temperature**) After obtaining the download link, we could use Wget Module to download the file.

The same method isn't fit in dynamic pages. In dynamic pages, including Accuweather.com mentioned in **3.5 Data Access – Cloud Height**, we tend to use Selenium Module to crawl for the information. The way we crawl in dynamic pages is to simulate real users and copy the information on the website. Therefore, it is slow and heavy.

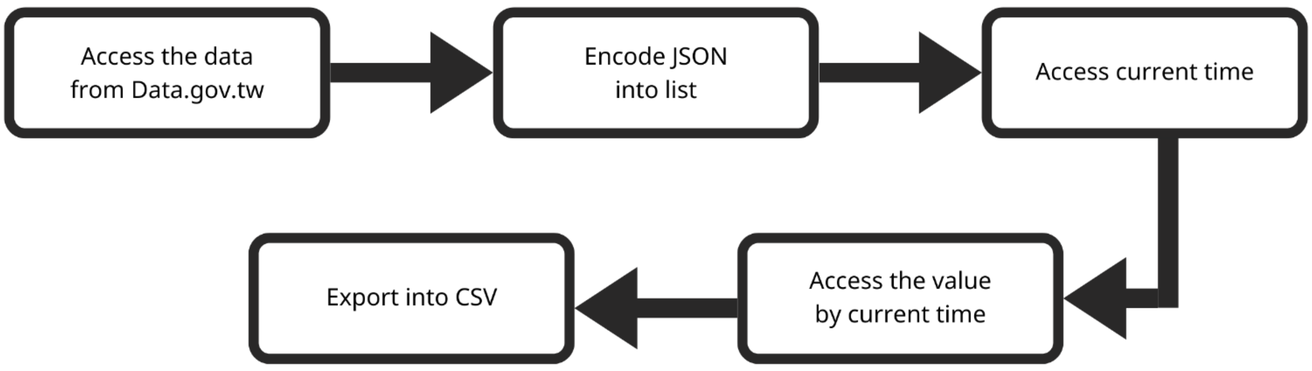
3.3 Data Access - Sunrise and Sunset Time

The most important cause of the stunning colorful cloud is the light right from the sun. Therefore, it is important for us to know the time of sunrise and sunset.

3.3.1 Data.gov.tw

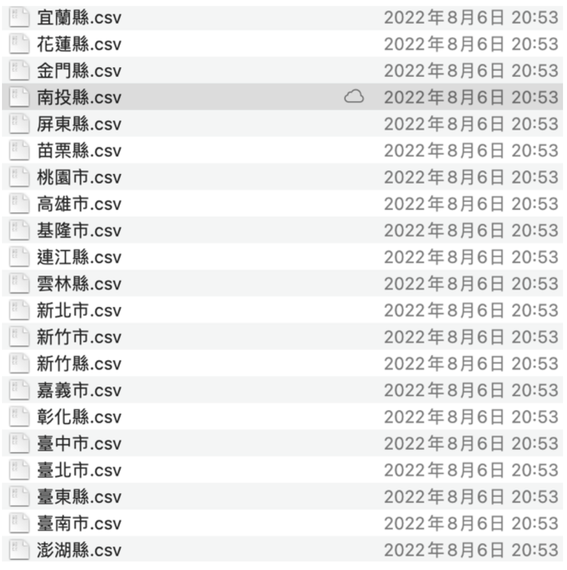
Data.gov.tw [9] is an online open data platform established by the Taiwanese government in order to release data to the public based on "The Freedom of Government Information Law". The platform provides multiple types of data, including CSV, XML, and JSON.

3.3.2 Data access

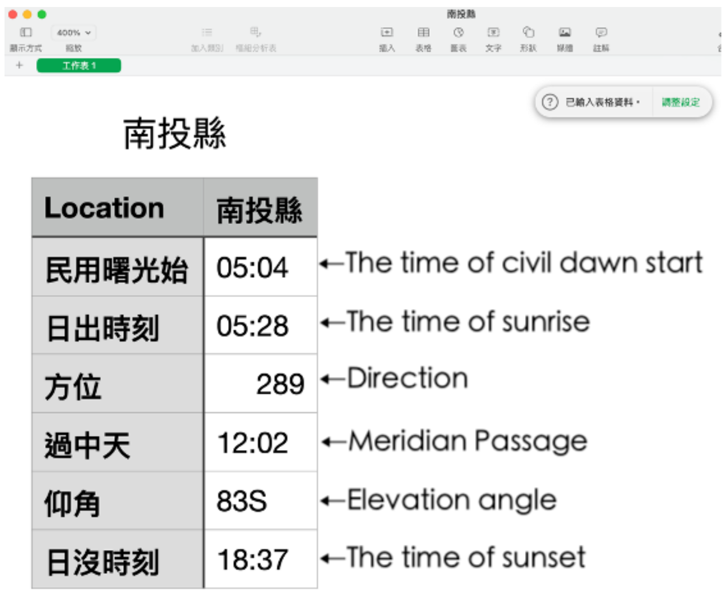

Access the data from Data.gov.tw
Encode JSON into list
Access current time
Access the value by current time
Export into CSV

▲Picture 3-2 The process chart of accessing sunset and sunrise value

The platform provides a permanent link for users to download the latest data. After downloading the data in JSON from the link, we encode it into a dictionary in Python. In the dictionary, it is easy for us to access the current data by importing the current time with Time module. Due to the data which has already been sorted out in cities, we export the current data in CSV files. And below are the codes (Picture 2-5) and the results of accessing the time of sunrise and sunset (Picture 2-3,4).



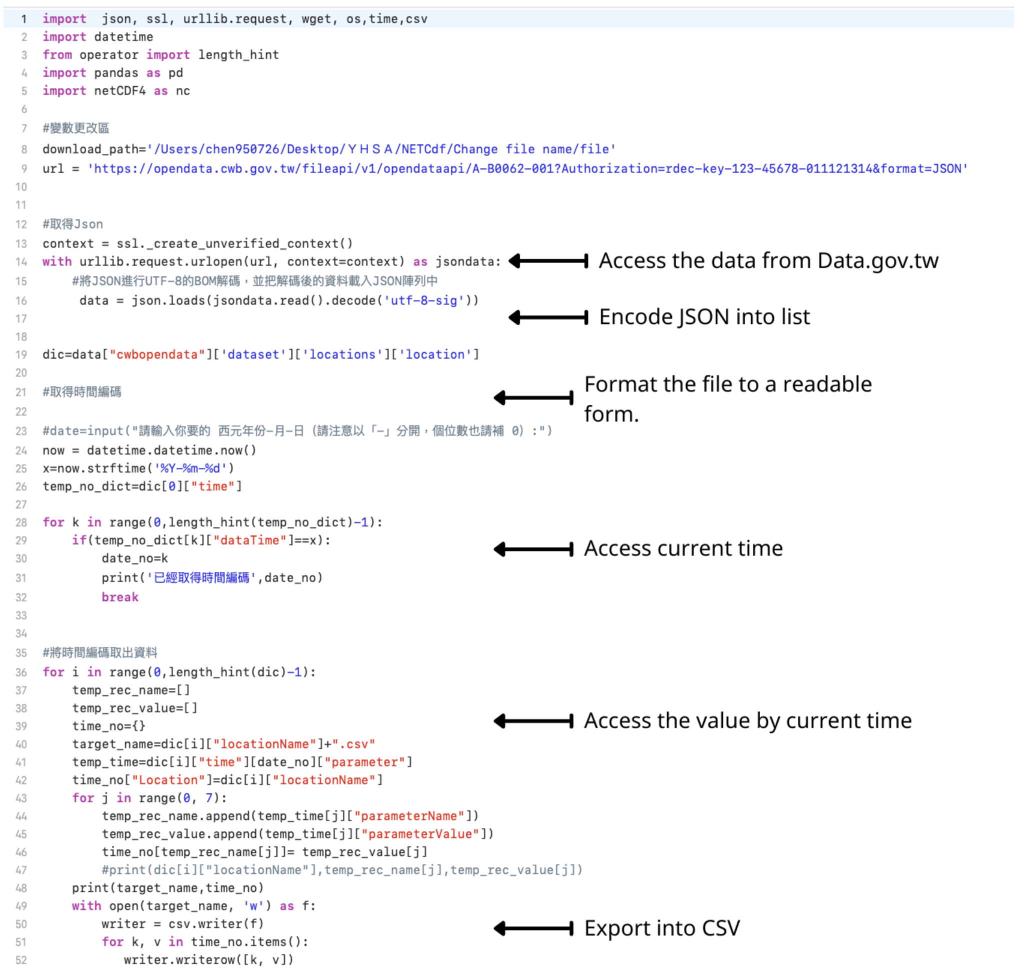
宜蘭縣.csv	2022年8月6日 20:53
花蓮縣.csv	2022年8月6日 20:53
金門縣.csv	2022年8月6日 20:53
南投縣.csv	2022年8月6日 20:53
屏東縣.csv	2022年8月6日 20:53
苗栗縣.csv	2022年8月6日 20:53
桃園市.csv	2022年8月6日 20:53
高雄市.csv	2022年8月6日 20:53
基隆市.csv	2022年8月6日 20:53
連江縣.csv	2022年8月6日 20:53
雲林縣.csv	2022年8月6日 20:53
新北市.csv	2022年8月6日 20:53
新竹市.csv	2022年8月6日 20:53
新竹縣.csv	2022年8月6日 20:53
嘉義市.csv	2022年8月6日 20:53
彰化縣.csv	2022年8月6日 20:53
臺中市.csv	2022年8月6日 20:53
臺北市.csv	2022年8月6日 20:53
臺東縣.csv	2022年8月6日 20:53
臺南市.csv	2022年8月6日 20:53
澎湖縣.csv	2022年8月6日 20:53



南投縣

Location	南投縣	
民用曙光始	05:04	←The time of civil dawn start
日出時刻	05:28	←The time of sunrise
方位	289	←Direction
過中天	12:02	←Meridian Passage
仰角	83S	←Elevation angle
日沒時刻	18:37	←The time of sunset

▲Picture 3-3,4 The result of accessing sunset and sunrise time


Access the data from Data.gov.tw
Encode JSON into list
Format the file to a readable form.
Access current time
Access the value by current time
Export into CSV

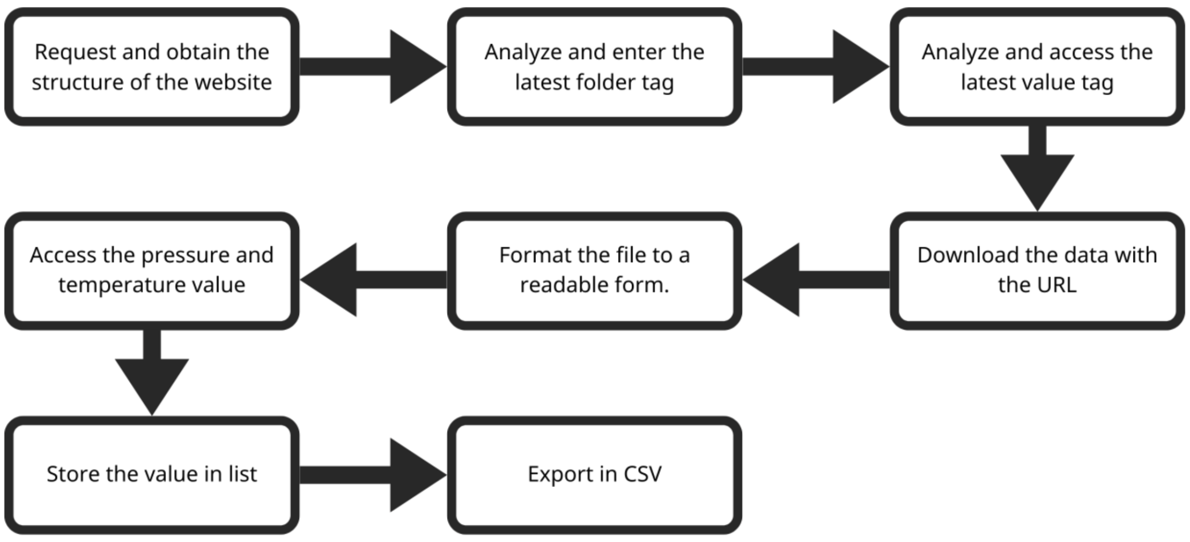
```python
import  json, ssl, urllib.request, wget, os,time,csv
import datetime
from operator import length_hint
import pandas as pd
import netCDF4 as nc

#變數更改區
download_path='/Users/chen950726/Desktop/ＹＨＳＡ/NETCdf/Change file name/file'
url = 'https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/A-B0062-001?Authorization=rdec-key-123-45678-011121314&format=JSON'


#取得Json
context = ssl._create_unverified_context()
with urllib.request.urlopen(url, context=context) as jsondata:
    #將JSON進行UTF-8的BOM解碼，並把解碼後的資料載入JSON陣列中
     data = json.loads(jsondata.read().decode('utf-8-sig'))


dic=data["cwbopendata"]['dataset']['locations']['location']

#取得時間編碼

#date=input("請輸入你要的 西元年份-月-日 (請注意以「-」分開，個位數也請補 0) :")
now = datetime.datetime.now()
x=now.strftime('%Y-%m-%d')
temp_no_dict=dic[0]["time"]

for k in range(0,length_hint(temp_no_dict)-1):
    if(temp_no_dict[k]["dataTime"]==x):
        date_no=k
        print('已經取得時間編碼',date_no)
        break


#將時間編碼取出資料
for i in range(0,length_hint(dic)-1):
    temp_rec_name=[]
    temp_rec_value=[]
    time_no={}
    target_name=dic[i]["locationName"]+".csv"
    temp_time=dic[i]["time"][date_no]["parameter"]
    time_no["Location"]=dic[i]["locationName"]
    for j in range(0, 7):
        temp_rec_name.append(temp_time[j]["parameterName"])
        temp_rec_value.append(temp_time[j]["parameterValue"])
        time_no[temp_rec_name[j]]= temp_rec_value[j]
        #print(dic[i]["locationName"],temp_rec_name[j],temp_rec_value[j])
    print(target_name,time_no)
    with open(target_name, 'w') as f:
        writer = csv.writer(f)
        for k, v in time_no.items():
           writer.writerow([k, v])
```

▲Picture 3-5 The code of accessing sunset and sunrise time

3.4 Data Access – Pressure and Temperature

Based on the below paragraph **(3.4.1 FORMOSAT-7)**, we can access the data of current atmospheric pressure and temperature with Formosat-7. There are two different systems that we can access target data, including Formosat-7 TDPC, and Formosat-7 TROPS. After our discussion, Formosat-7 TDPC is our first choice, according to its stability and more advanced property.

The only way we can access the latest data is through its official download site. In this situation, we tend to use Python for our coding language, due to its rich module database, consisting of Wget, Request, and NetCDF4.



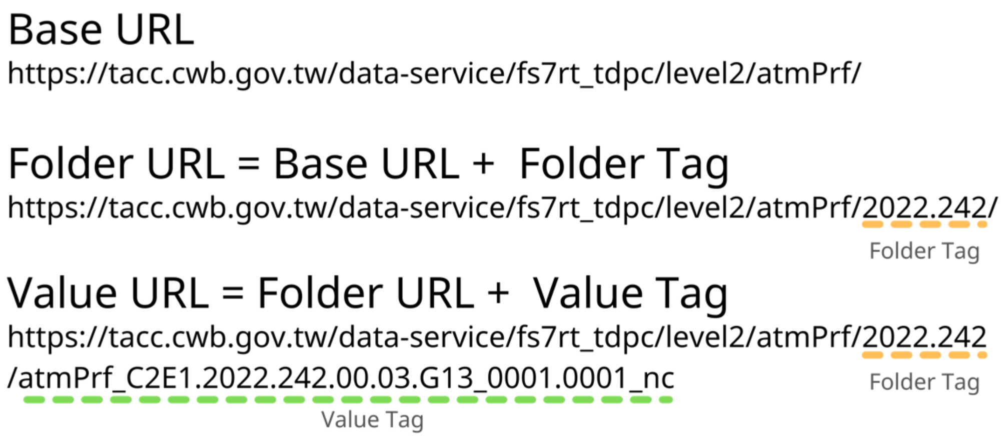
Request and obtain the structure of the website
Analyze and enter the latest folder tag
Analyze and access the latest value tag
Download the data with the URL
Format the file to a readable form.
Access the pressure and temperature value
Store the value in list
Export in CSV

▲Picture 3-6 The process chart of accessing pressure and pressure

3.4.1 FORMOSAT-7

The Formosat-7 [10] is a major international collaborative space program between Taiwan and the USA, aimed at creating a high-reliability meteorological satellite system. All six satellites of the Formosat-7 system orbit the Earth for about 97 minutes, and could provide about 4000 data per day between 50 degrees north and south latitudes. By measuring radio occultation signals, atmospheric parameters, including refractivity, dry pressure and temperature, humidity, and the electron concentration of the ionosphere can be derived. Users can approach the data from Taiwan Data Processing Center (https://tacc.cwb.gov.tw).

3.4.2 Download the file



Base URL
https://tacc.cwb.gov.tw/data-service/fs7rt_tdpc/level2/atmPrf/

Folder URL = Base URL + Folder Tag
https://tacc.cwb.gov.tw/data-service/fs7rt_tdpc/level2/atmPrf/2022.242/
Folder Tag

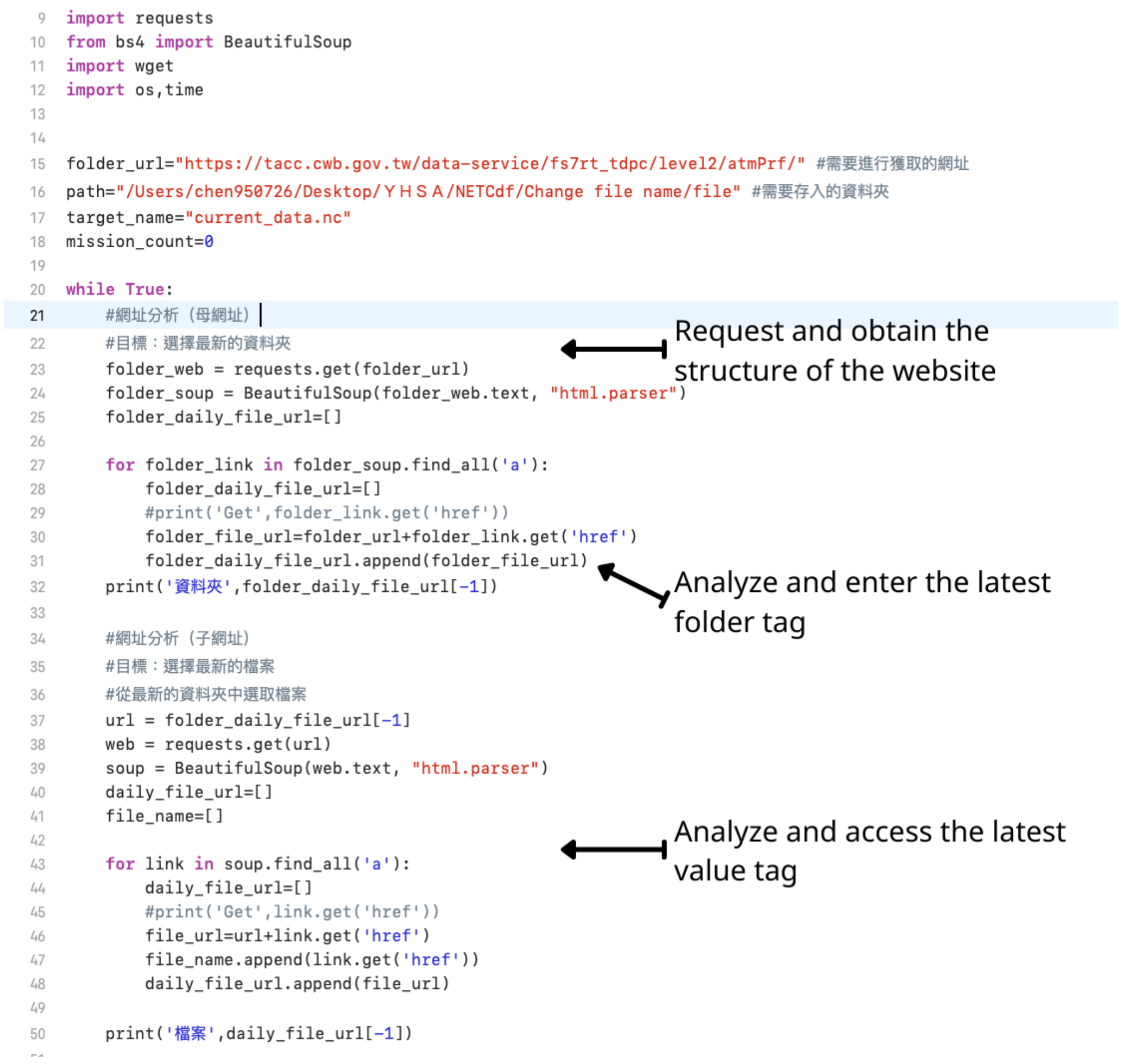
Value URL = Folder URL + Value Tag
https://tacc.cwb.gov.tw/data-service/fs7rt_tdpc/level2/atmPrf/2022.242
/atmPrf_C2E1.2022.242.00.03.G13_0001.0001_nc
Folder Tag
Value Tag

▲Picture 3-7 The characteristic of value URL and folder URL

First, we have to get the folder URL. In this situation, we have to import the Requests Module to access the website information. Due to the characteristic of the URL under this condition, the target we need to get was only the name of the latest file (folder tag). After reaching the target with Beautiful Soup Module, we combine it with the base URL to get the folder URL. After then, we do the exact same steps we did above to get the value URL. Last but not least, we download the file with the value URL with Wget Module.

3.4.3 Analyze the file to extract the Information

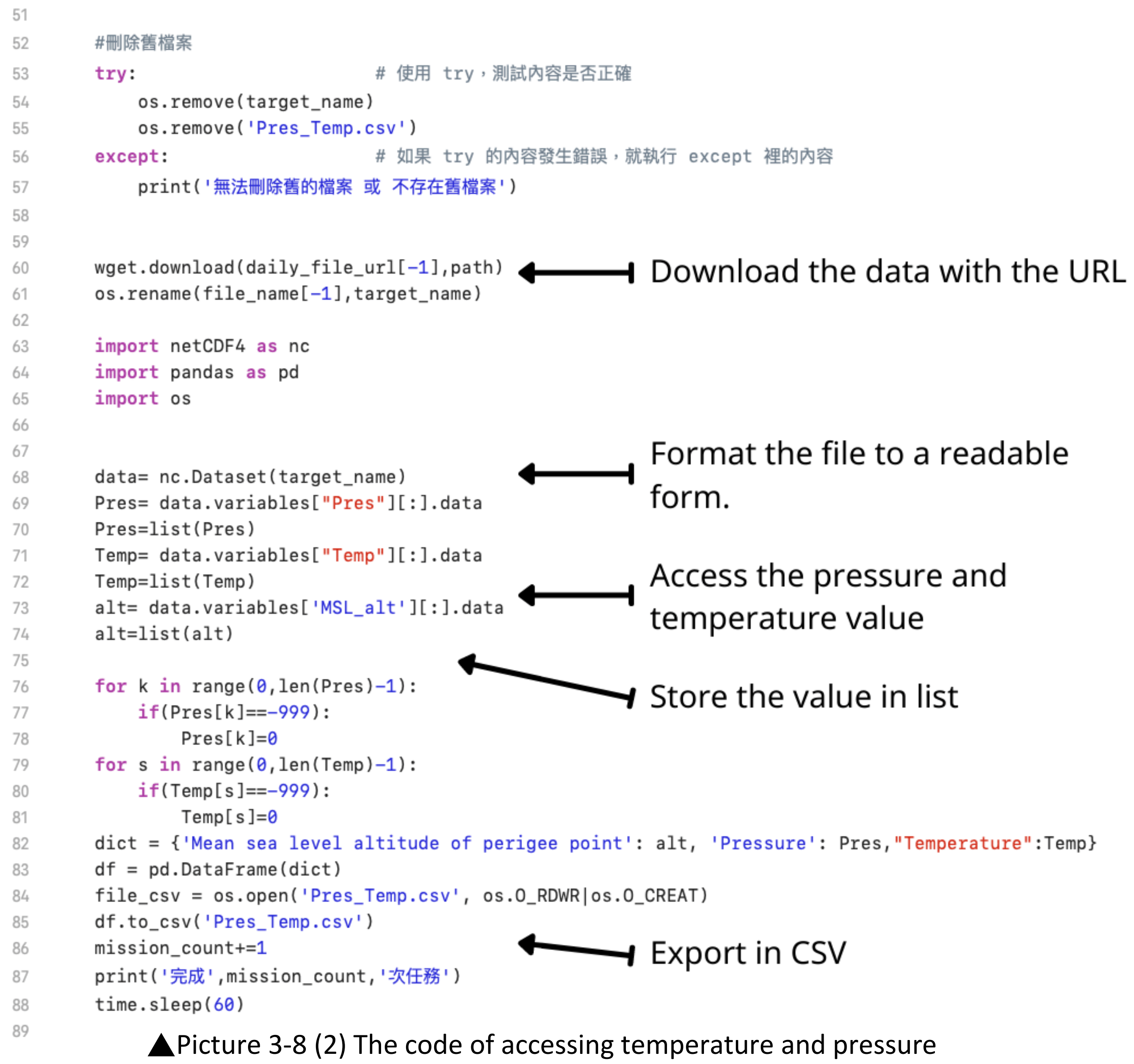
After achieving the file, we have to analyze and extract it to get the information we need. First, we analyze it with Panoply, an app released by NASA. After we got the information we need, we import and read the file into python by netCDF4 module. The data we took out would be stored in the array and exported into CSV for later simulating.



```python
import requests
from bs4 import BeautifulSoup
import wget
import os,time


folder_url="https://tacc.cwb.gov.tw/data-service/fs7rt_tdpc/level2/atmPrf/" #需要進行獲取的網址
path="/Users/chen950726/Desktop/ＹＨＳＡ/NETCdf/Change file name/file" #需要存入的資料夾
target_name="current_data.nc"
mission_count=0

while True:
    #網址分析（母網址）
    #目標：選擇最新的資料夾
    folder_web = requests.get(folder_url)
    folder_soup = BeautifulSoup(folder_web.text, "html.parser")
    folder_daily_file_url=[]

    for folder_link in folder_soup.find_all('a'):
        folder_daily_file_url=[]
        #print('Get',folder_link.get('href'))
        folder_file_url=folder_url+folder_link.get('href')
        folder_daily_file_url.append(folder_file_url)
    print('資料夾',folder_daily_file_url[-1])

    #網址分析（子網址）
    #目標：選擇最新的檔案
    #從最新的資料夾中選取檔案
    url = folder_daily_file_url[-1]
    web = requests.get(url)
    soup = BeautifulSoup(web.text, "html.parser")
    daily_file_url=[]
    file_name=[]

    for link in soup.find_all('a'):
        daily_file_url=[]
        #print('Get',link.get('href'))
        file_url=url+link.get('href')
        file_name.append(link.get('href'))
        daily_file_url.append(file_url)

    print('檔案',daily_file_url[-1])
```

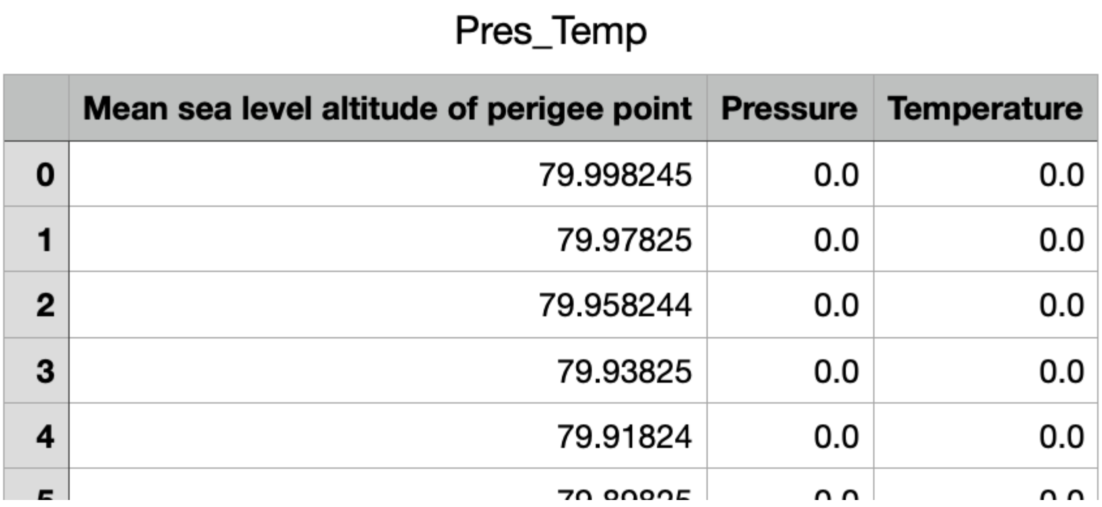
Request and obtain the structure of the website
Analyze and enter the latest folder tag
Analyze and access the latest value tag

▲Picture 3-8 (1) The code of accessing temperature and pressure


Download the data with the URL
Format the file to a readable form.
Access the pressure and temperature value
Store the value in list
Export in CSV

```python

#刪除舊檔案
try:                          # 使用 try，測試內容是否正確
    os.remove(target_name)
    os.remove('Pres_Temp.csv')
except:                       # 如果 try 的內容發生錯誤，就執行 except 裡的內容
    print('無法刪除舊的檔案 或 不存在舊檔案')


wget.download(daily_file_url[-1],path)
os.rename(file_name[-1],target_name)

import netCDF4 as nc
import pandas as pd
import os


data= nc.Dataset(target_name)
Pres= data.variables["Pres"][:].data
Pres=list(Pres)
Temp= data.variables["Temp"][:].data
Temp=list(Temp)
alt= data.variables['MSL_alt'][:].data
alt=list(alt)

for k in range(0,len(Pres)-1):
    if(Pres[k]==-999):
        Pres[k]=0
for s in range(0,len(Temp)-1):
    if(Temp[s]==-999):
        Temp[s]=0
dict = {'Mean sea level altitude of perigee point': alt, 'Pressure': Pres,"Temperature":Temp}
df = pd.DataFrame(dict)
file_csv = os.open('Pres_Temp.csv', os.O_RDWR|os.O_CREAT)
df.to_csv('Pres_Temp.csv')
mission_count+=1
print('完成',mission_count,'次任務')
time.sleep(60)

```

▲Picture 3-8 (2) The code of accessing temperature and pressure



Pres_Temp

	Mean sea level altitude of perigee point	Pressure	Temperature
0	79.998245	0.0	0.0
1	79.97825	0.0	0.0
2	79.958244	0.0	0.0
3	79.93825	0.0	0.0
4	79.91824	0.0	0.0
5	[illegible]	[illegible]	[illegible]

▲Picture 3-9 The result of accessing pressure and temperature

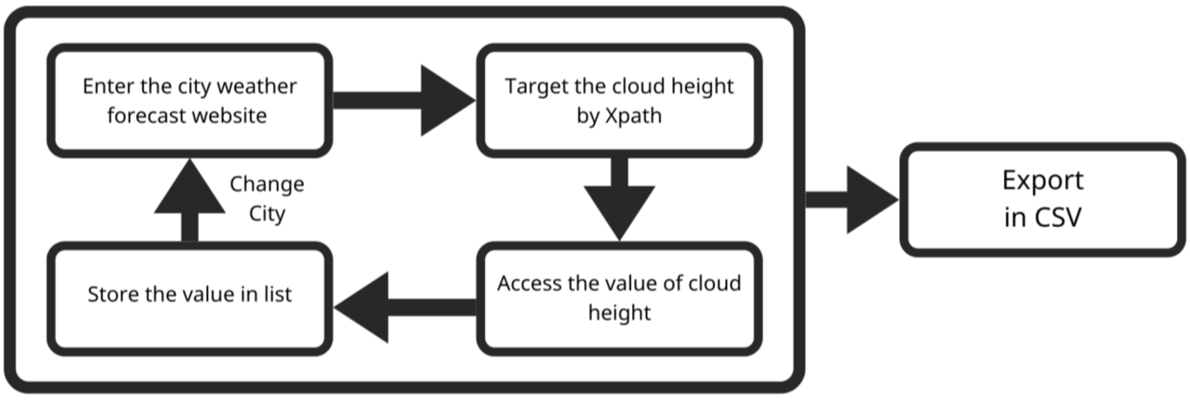
3.5 Data Access – Cloud Height

In order to calculate the route of light, we need the position of the cloud to calculate the reflect angle of the light reflected from the cloud. There are more details at **2.7 Stimulate**. Therefore, if we want to simulate the stunning clouds, it is important for us to obtain the height of the cloud.

3.5.1 AccuWeather

There are many different ways to access the data, including commercial weather forecasts, aviation weather forecasts, and the data released from the satellite. AccuWeather is a company aimed at providing accurate commercial weather forecasts, which also runs a free, advertising-supported website (AccuWeather.com). The website is based on weather information from numerous sources, including National Weather Service (NWS) and other reliable meteorological organizations [11].

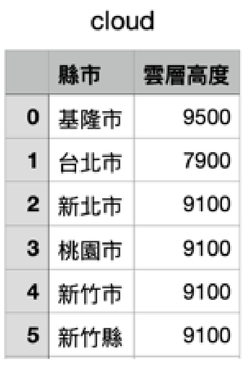
3.5.2 Data access


Enter the city weather forecast website
Target the cloud height by Xpath
Change City
Access the value of cloud height
Store the value in list
Export in CSV

▲Picture 3-9 The process chart of accessing cloud height

There are several modules for users to crawl on the internet, including Requests, which we used in the above paragraph. AccuWeather.com is an international website, which also runs on a dynamic web page. Therefore, selenium is our first choice in order to read the information on a dynamic web page.

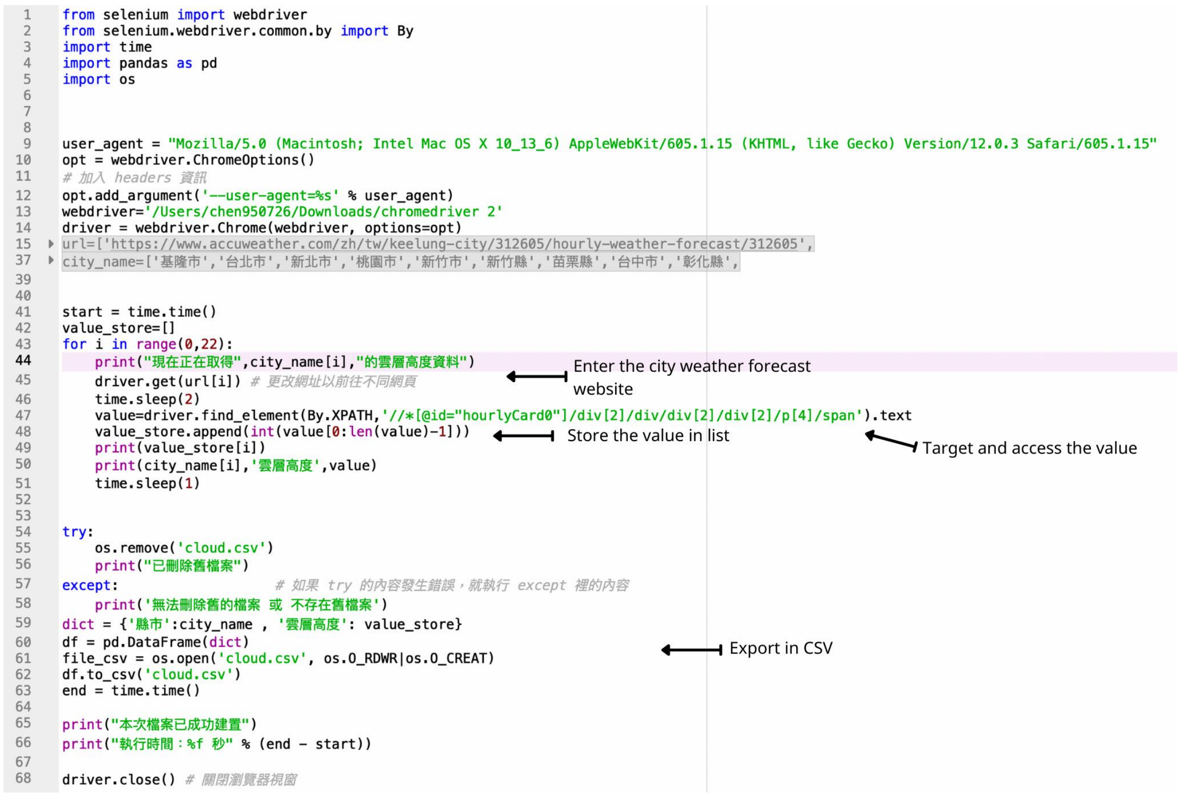
Firstly, we enter the website, where we want to access the target with Python and Selenium. Secondly, we acquire it with XPath, which can target our value. After we got the value, we stored it in the list until we finish running all the cities we want. Finally, we stored all the values in CSV (Picture 3-10).



cloud

	縣市	雲層高度
0	基隆市	9500
1	台北市	7900
2	新北市	9100
3	桃園市	9100
4	新竹市	9100
5	新竹縣	9100

▲Picture 3-10 The result of accessing the height of cloud



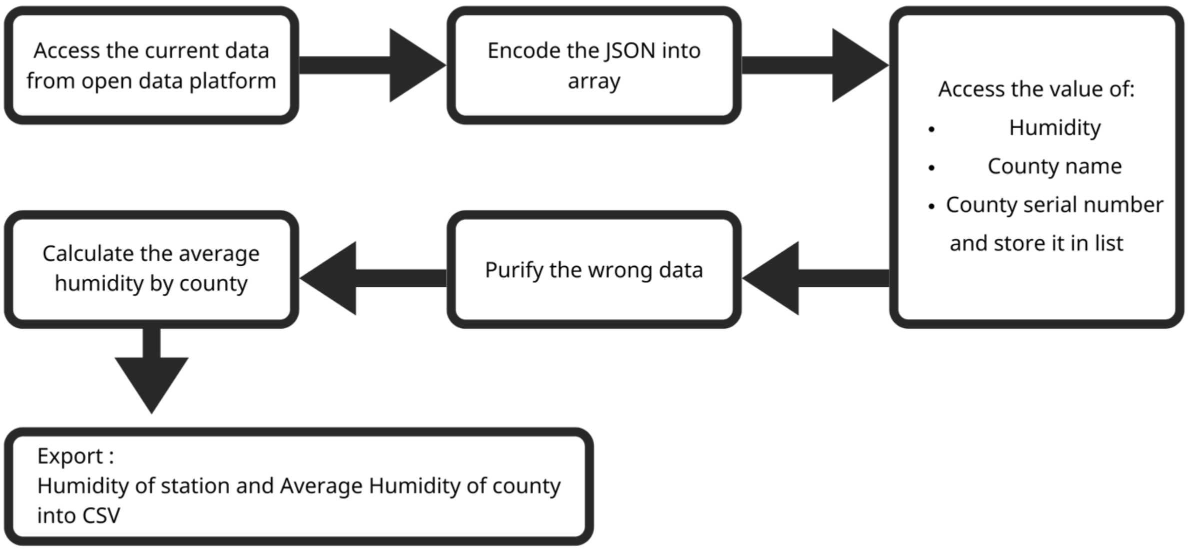
```python
from selenium import webdriver
from selenium.webdriver.common.by import By
import time
import pandas as pd
import os



user_agent = "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_6) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/12.0.3 Safari/605.1.15"
opt = webdriver.ChromeOptions()
# 加入 headers 資訊
opt.add_argument('--user-agent=%s' % user_agent)
webdriver='/Users/chen950726/Downloads/chromedriver 2'
driver = webdriver.Chrome(webdriver, options=opt)
url=['https://www.accuweather.com/zh/tw/keelung-city/312605/hourly-weather-forecast/312605',
city_name=['基隆市','台北市','新北市','桃園市','新竹市','新竹縣','苗栗縣','台中市','彰化縣',


start = time.time()
value_store=[]
for i in range(0,22):
    print("現在正在取得",city_name[i],"的雲層高度資料")
    driver.get(url[i]) # 更改網址以前往不同網頁
    time.sleep(2)
    value=driver.find_element(By.XPATH,'//*[@id="hourlyCard0"]/div[2]/div/div[2]/div[2]/p[4]/span').text
    value_store.append(int(value[0:len(value)-1]))
    print(value_store[i])
    print(city_name[i],'雲層高度',value)
    time.sleep(1)


try:
    os.remove('cloud.csv')
    print("已刪除舊檔案")
except:                 # 如果 try 的內容發生錯誤，就執行 except 裡的內容
    print('無法刪除舊的檔案 或 不存在舊檔案')
dict = {'縣市':city_name , '雲層高度': value_store}
df = pd.DataFrame(dict)
file_csv = os.open('cloud.csv', os.O_RDWR|os.O_CREAT)
df.to_csv('cloud.csv')
end = time.time()

print("本次檔案已成功建置")
print("執行時間：%f 秒" % (end - start))

driver.close() # 關閉瀏覽器視窗
```

Enter the city weather forecast website
Store the value in list
Target and access the value
Export in CSV

▲Picture 3-11 The code of accessing the height of cloud

3.6 Data Access - Humidity

In the atmosphere, refraction is caused by spatial variations, including temperature, pressure, and humidity, with humidity being the most important. Therefore, it is important for us to access the current humidity.


Access the current data from open data platform
Encode the JSON into array
Access the value of:
• Humidity
• County name
• County serial number
and store it in list
Purify the wrong data
Calculate the average humidity by county
Export :
Humidity of station and Average Humidity of county into CSV

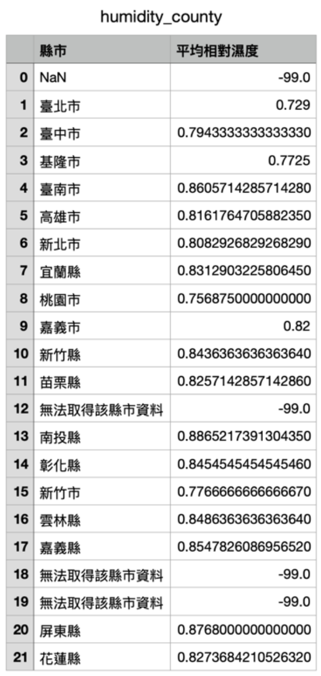
▲Picture 3-12 The process chart of accessing humidity

3.6.1 Open Weather Data Platform

Open Weather Data Platform (opendata.cwb.gov.tw) is an open data platform, founded by the Central Weather Bureau (CWB) in Taiwan for users to access the current data of meteorological information. It provides multiple different data, county weather forecast and meteorological observations data was included [12]. In this report, we will be using meteorological observations data of automatic weather stations , which provides humidity in its dataset.

3.6.2 Data access

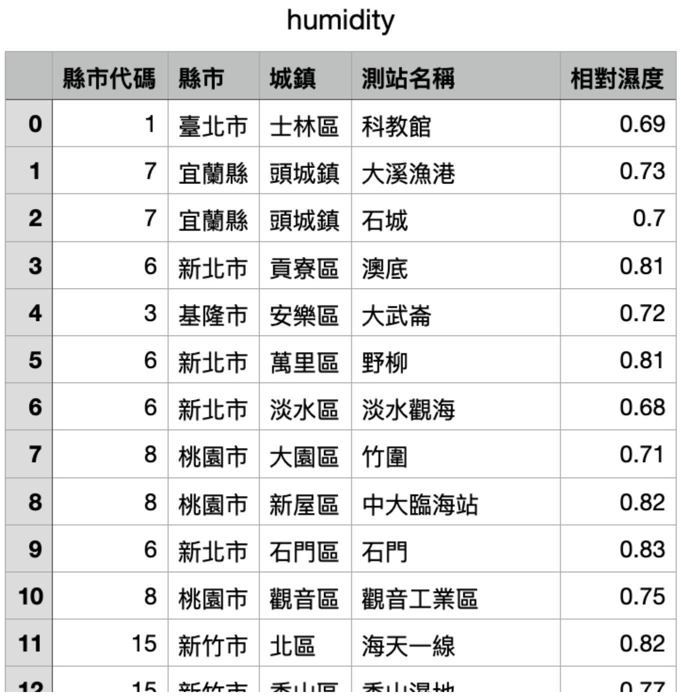
It is the same way for us to access the data as we did in 2.2.2. First of all, we reached the data from the URL and encode the JSON into the array. After reaching the data, we obtained the value that we need, including humidity, county name, county number, and the automatic weather station name. Getting all the values, the next step we did is to calculate the average humidity by county. First, we extract the humidity with the same county number and store it in list. Second, we counted the quantity of the same county number at the same time. After calculations, we can get the average humidity. At last, we export two data: humidity of station and average humidity of county (Picture 3-13,14).



humidity_county

	縣市	平均相對濕度
0	NaN	-99.0
1	臺北市	0.729
2	臺中市	0.7943333333333330
3	基隆市	0.7725
4	臺南市	0.8605714285714280
5	高雄市	0.8161764705882350
6	新北市	0.8082926829268290
7	宜蘭縣	0.8312903225806450
8	桃園市	0.7568750000000000
9	嘉義市	0.82
10	新竹縣	0.8436363636363640
11	苗栗縣	0.8257142857142860
12	無法取得該縣市資料	-99.0
13	南投縣	0.8865217391304350
14	彰化縣	0.8454545454545460
15	新竹市	0.7766666666666670
16	雲林縣	0.8486363636363640
17	嘉義縣	0.8547826086956520
18	無法取得該縣市資料	-99.0
19	無法取得該縣市資料	-99.0
20	屏東縣	0.8768000000000000
21	花蓮縣	0.8273684210526320

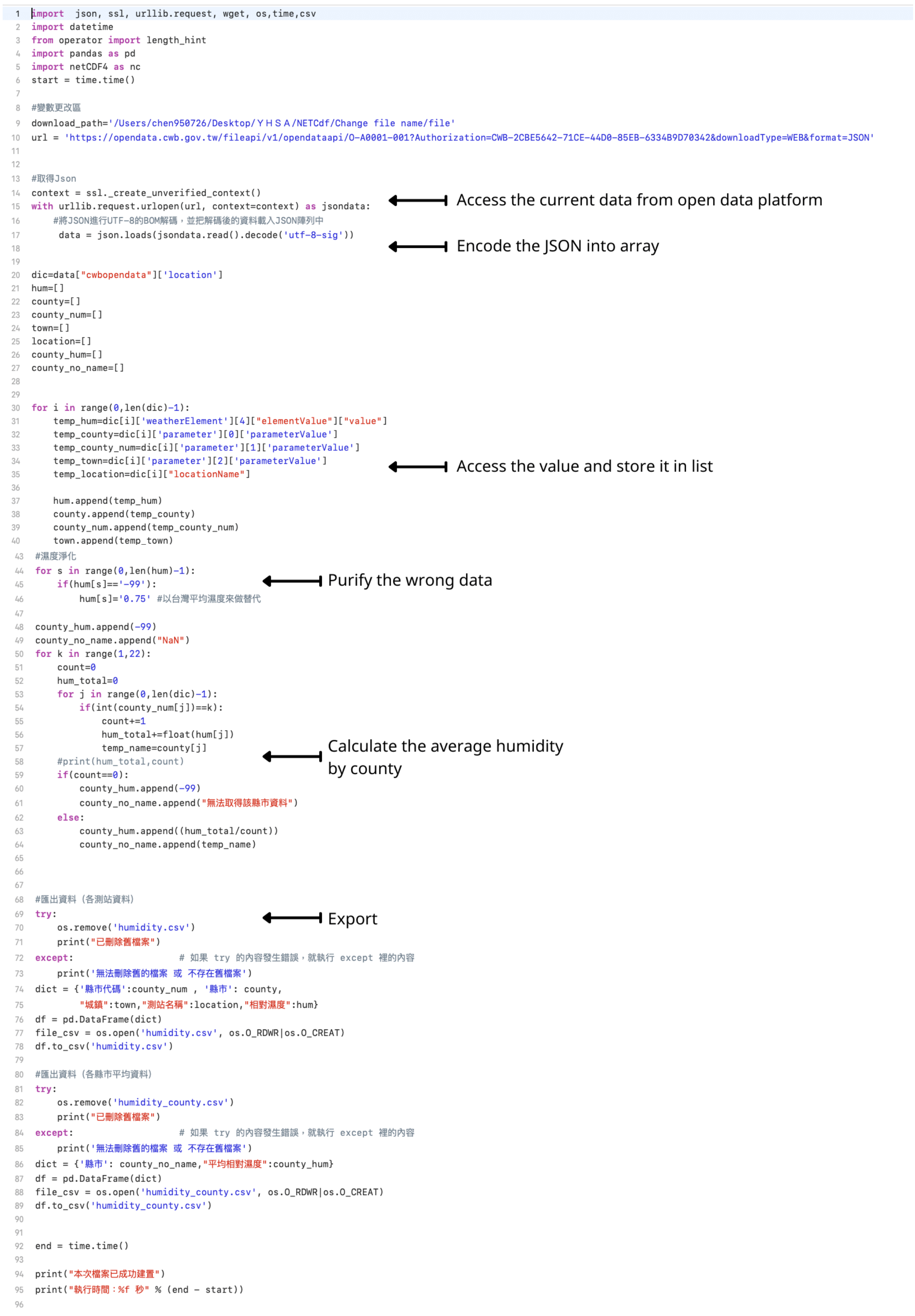
▲Picture 3-13 Average humidity of



humidity

	縣市代碼	縣市	城鎮	測站名稱	相對濕度
0	1	臺北市	士林區	科教館	0.69
1	7	宜蘭縣	頭城鎮	大溪漁港	0.73
2	7	宜蘭縣	頭城鎮	石城	0.7
3	6	新北市	貢寮區	澳底	0.81
4	3	基隆市	安樂區	大武崙	0.72
5	6	新北市	萬里區	野柳	0.81
6	6	新北市	淡水區	淡水觀海	0.68
7	8	桃園市	大園區	竹圍	0.71
8	8	桃園市	新屋區	中大臨海站	0.82
9	6	新北市	石門區	石門	0.83
10	8	桃園市	觀音區	觀音工業區	0.75
11	15	新竹市	北區	海天一線	0.82

▲Picture 3-14 Humidity of station


Access the current data from open data platform
Encode the JSON into array
Access the value and store it in list
Purify the wrong data
Calculate the average humidity by county
Export

```python
import  json, ssl, urllib.request, wget, os,time,csv
import datetime
from operator import length_hint
import pandas as pd
import netCDF4 as nc
start = time.time()

#變數更改區
download_path='/Users/chen950726/Desktop/Ｙ Ｈ Ｓ Ａ/NETCdf/Change file name/file'
url = 'https://opendata.cwb.gov.tw/fileapi/v1/opendataapi/O-A0001-001?Authorization=CWB-2CBE5642-71CE-44D0-85EB-6334B9D70342&downloadType=WEB&format=JSON'


#取得Json
context = ssl._create_unverified_context()
with urllib.request.urlopen(url, context=context) as jsondata:
    #將JSON進行UTF-8的BOM解碼，並把解碼後的資料載入JSON陣列中
     data = json.loads(jsondata.read().decode('utf-8-sig'))


dic=data["cwbopendata"]['location']
hum=[]
county=[]
county_num=[]
town=[]
location=[]
county_hum=[]
county_no_name=[]


for i in range(0,len(dic)-1):
    temp_hum=dic[i]['weatherElement'][4]["elementValue"]["value"]
    temp_county=dic[i]['parameter'][0]['parameterValue']
    temp_county_num=dic[i]['parameter'][1]['parameterValue']
    temp_town=dic[i]['parameter'][2]['parameterValue']
    temp_location=dic[i]["locationName"]

    hum.append(temp_hum)
    county.append(temp_county)
    county_num.append(temp_county_num)
    town.append(temp_town)
#濕度淨化
for s in range(0,len(hum)-1):
    if(hum[s]=='-99'):
        hum[s]='0.75' #以台灣平均濕度來做替代

county_hum.append(-99)
county_no_name.append("NaN")
for k in range(1,22):
    count=0
    hum_total=0
    for j in range(0,len(dic)-1):
        if(int(county_num[j])==k):
            count+=1
            hum_total+=float(hum[j])
            temp_name=county[j]
    #print(hum_total,count)
    if(count==0):
        county_hum.append(-99)
        county_no_name.append("無法取得該縣市資料")
    else:
        county_hum.append((hum_total/count))
        county_no_name.append(temp_name)



#匯出資料 (各測站資料)
try:
    os.remove('humidity.csv')
    print("已刪除舊檔案")
except:                     # 如果 try 的內容發生錯誤，就執行 except 裡的內容
    print('無法刪除舊的檔案 或 不存在舊檔案')
dict = {'縣市代碼':county_num , '縣市': county,
        "城鎮":town,"測站名稱":location,"相對濕度":hum}
df = pd.DataFrame(dict)
file_csv = os.open('humidity.csv', os.O_RDWR|os.O_CREAT)
df.to_csv('humidity.csv')

#匯出資料 (各縣市平均資料)
try:
    os.remove('humidity_county.csv')
    print("已刪除舊檔案")
except:                     # 如果 try 的內容發生錯誤，就執行 except 裡的內容
    print('無法刪除舊的檔案 或 不存在舊檔案')
dict = {'縣市': county_no_name,"平均相對濕度":county_hum}
df = pd.DataFrame(dict)
file_csv = os.open('humidity_county.csv', os.O_RDWR|os.O_CREAT)
df.to_csv('humidity_county.csv')


end = time.time()

print("本次檔案已成功建置")
print("執行時間：%f 秒" % (end - start))
```

▲Picture 3-12 The code of accessing humidity

3.7 Refraction index

Mainly, we use python to compute the whole calculation, and csv file to connect different calculations. And the below are the calculations mentioned at [1.2].

First, the Cauchy's equation (2)

```
def Cauchy_equation(air_pressure,air_temperature,wavelength,relative_humidity):

  T = air_temperature
  w = wavelength
  a = air_pressure
  v = pressure_of_water_vapour(relative_humidity,T)

  return 1+(0.0000776/T)*(1+(0.00752)/w/w)*(a+4810*v/T)
```

air_temperature and air_pressure are given by **2.3 Data Accessing – Pressure and Temperature**, which is related with the altitude, different altitude layers would have different temperature and pressure.

```
def pressure_of_water_vapour(relative_humidity,temperature):

  RH= relative_humidity
  T = temperature

  es=6.112*np.exp(17.67*T/(T+243.5))   # saturated vapor
  e = RH*es/100                        # actual vapor pressure
  Mp = 0.662*e/(a-e)                     # mixing ratio at the pressure level, p

  return -1/986.665*Mp
```

As for the integration at (3), because the temperature and pressure alter in the small gap, 20 meters, we choose it as the gap in numerical method for the integration, or as the code below.

```
for i in range(0,3899):

    air_pressure       = float(pressure[i])
    air_temperature    = float(temperature[i])+273.0
    wavelength         = 700*1000.0
    relative_humidity  = 0.7

    refractive_index.append(Cauchy_equation(
        air_pressure       = air_pressure,
        air_temperature    = air_temperature,
        relative_humidity  = relative_humidity,
        wavelength         = wavelength
        )
    )
```

Because the colorful cloud is often caused by the red-light, we use 700 nm as the wavelength of light. As for the humidity, we use the data accessed from **2.5 Data Access – Humidity**.

3.8 Simulation

When thinking how to present the simulation, we didn't have any access to the professional software which can run the complex simulation. We didn't know how to conduct it as well. Therefore, we use the simple but powerful way "Matplotlib" to conduct it.

3.8.1 Light route in atmosphere

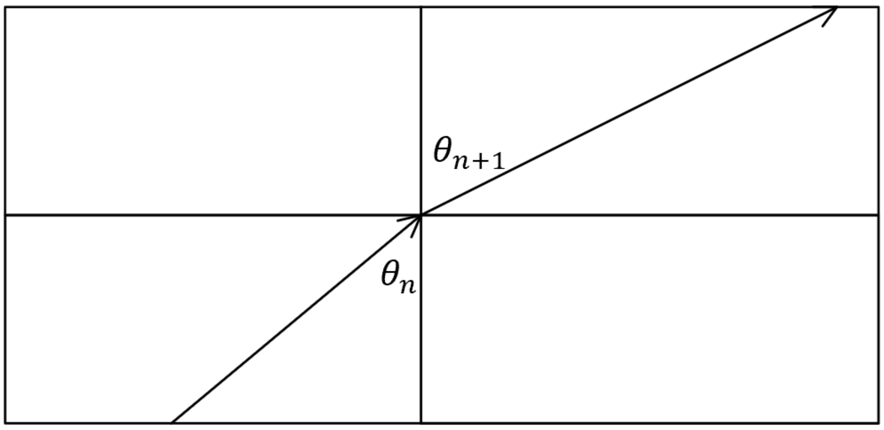
The main idea is to use the light vector reflecting from the target cloud to reverse the light route during traveling through the atmosphere. It would work is because that the thickness of troposphere is very thin to atmosphere. Therefore, the light reflected from the cloud would have enough distance to be refracted. After we reverse the light route, we can calculate the incident angle and therefore access the time from the time-incident angle relation at **2.7.2**. Also noted that the main colorful cloud reflects the near-red light, so we use the refraction index which has been adjusted to the red-light wavelength of 700 nm by the Cauchy Equation mentioned at **1.2.2.**

How we use the refraction index, which mentioned at **1.2.2,** adjusted to the near-red light refraction index, is as below equation:

$$N_n \cdot sin\theta_n = N_{n+1} \cdot sin\theta_{n+1} \Rightarrow sin\theta_{n+1} = \frac{N_n \times sin\theta_n}{N_{n+1}}$$

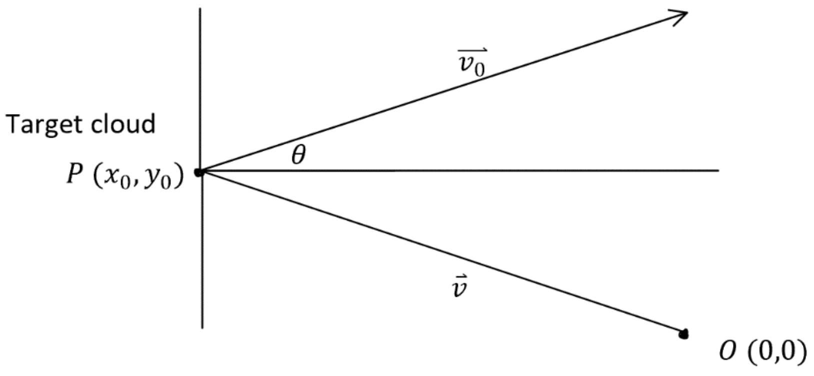
▲Equation 10

where N_n and N_{n+1} are the refraction index in the current and the next level.



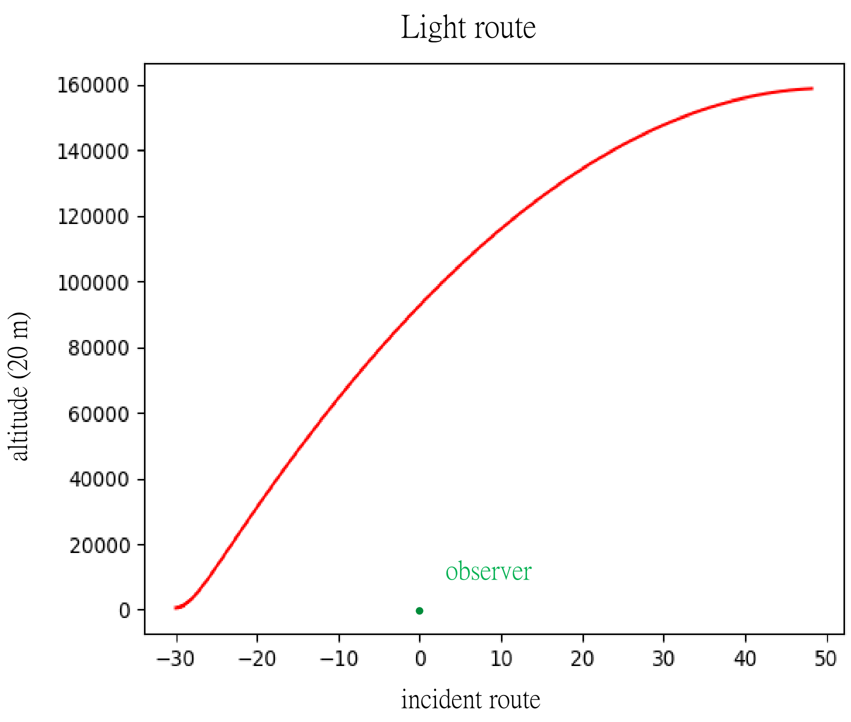
▲Picture 3-13

But there are some initial setups you need to know before understanding the code. The observer O is at $(0,0)$ and the target cloud P we want to observe is at (x_0, y_0). If we match the O and P together, we can get the vector $\overrightarrow{OP}$ or $\vec{v}$ which can be used to calculate the reflect angle θ and reflect vector $\overrightarrow{v_0}$.



▲Picture 3-14

After we have the reflect angle θ , we can use the equation above, using the current and the next level refraction index N_n to calculate the next level's refraction angle θ_{n+1} , and continuously doing it until reaching the top of the atmosphere. As the result, we can connect all the vector together, and get the picture below to simulate the light route.



▲Picture 3-15

In above picture, the line beams would travel in the atmosphere until they are reflected by the target cloud at (-30,40), which would be received by the observer at (0,0). Due to the reflection, the light beams would be seen from the near-horizonal direction imaged by humans' brain. Consequently, the pretty scene of the phenomenon would appear in our sight.

3.8.2 Time-incident angle relation

We calculate the solar altitude by the formula at (1.2.3). The incident angle is the solar altitude angle's supplementary angle. And we can easily compute it and list the correspondence between the incident angle and time of the day. We build two lists: one is the time and the other is the solar altitude angle. Between two adjacent materials, there is 15-minute interval. Because we only discuss about the cloud of dawn, we search the data from midday to sunset. While the incident angle (rads) is given, the program would find the location of it and print two times which the incident angle given is within the data corresponded by.

```
m=int((sunset-sunrise)/30)
m0=int((sunset-sunrise)/15)
while 1<int(m0-m) :
 if vector < solaraltitude[int((m+m0)/2)-1]:
  m0=int((m+m0)/2)
 else:
  m=int((m+m0)/2)
```

Above is how we search the location of data from two lists. Where sunrise is time of sunrise, sunset is that of sunset, and vector is the incident angle given. Because our time interval is 15-minute, m is the location of midday data and m0 is the location of sunset in the lists. Noted that our predict time is a 15-minute interval because of the unpredictable variants such as user favor, and mis simulation.

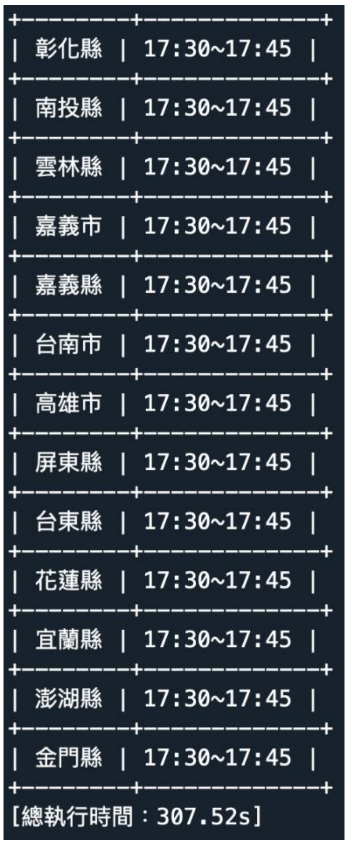
We take the following time-incident angle table for example:

Time	Incident angle
18:00	1.4114465988757618
18:15	1.4669588037239287

If the best appreciation incident angle (rads) is 1.45, the code will print 18:00~18:15. In this way, the user can get the most appropriate time to watch the phenomenon

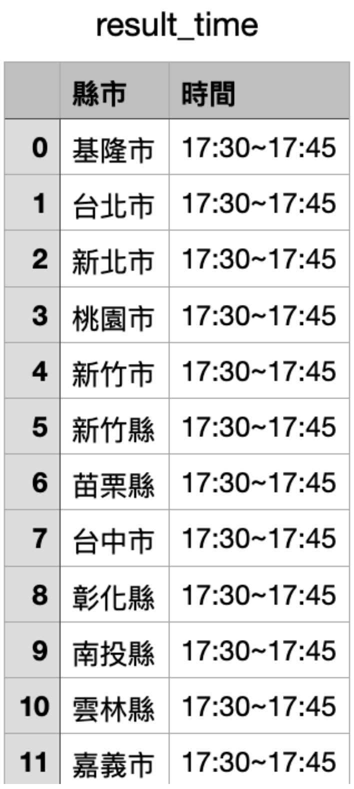
3.9 Output

Here are the output results we got (Picture 3-16). The results can also output into CSV type (Picture 3-17) for future applications.



```
+--------+-------------+
| 彰化縣 | 17:30~17:45 |
+--------+-------------+
| 南投縣 | 17:30~17:45 |
+--------+-------------+
| 雲林縣 | 17:30~17:45 |
+--------+-------------+
| 嘉義市 | 17:30~17:45 |
+--------+-------------+
| 嘉義縣 | 17:30~17:45 |
+--------+-------------+
| 台南市 | 17:30~17:45 |
+--------+-------------+
| 高雄市 | 17:30~17:45 |
+--------+-------------+
| 屏東縣 | 17:30~17:45 |
+--------+-------------+
| 台東縣 | 17:30~17:45 |
+--------+-------------+
| 花蓮縣 | 17:30~17:45 |
+--------+-------------+
| 宜蘭縣 | 17:30~17:45 |
+--------+-------------+
| 澎湖縣 | 17:30~17:45 |
+--------+-------------+
| 金門縣 | 17:30~17:45 |
+--------+-------------+
[總執行時間：307.52s]
```

▲Picture 3-16 Output Result Example



result_time

	縣市	時間
0	基隆市	17:30~17:45
1	台北市	17:30~17:45
2	新北市	17:30~17:45
3	桃園市	17:30~17:45
4	新竹市	17:30~17:45
5	新竹縣	17:30~17:45
6	苗栗縣	17:30~17:45
7	台中市	17:30~17:45
8	彰化縣	17:30~17:45
9	南投縣	17:30~17:45
10	雲林縣	17:30~17:45
11	嘉義市	17:30~17:45

▲Picture 3-17 Output result in CSV Example

The reason we choose 15 minutes as a prediction gap is that most of the sunset clouds often last about 15 to 20 minutes. It is meaningless to predict less than 10 minutes. Although predicting less than 10 minutes can get a more precise result, it can't create more worthwhile. Instead, it could cause more unnecessary problems. On the other hand, predicting more than 20 minutes is also pointless. While our model's accuracy might improve rapidly in the wake of the time increase, it would also lose the significance of the existence of this prediction.

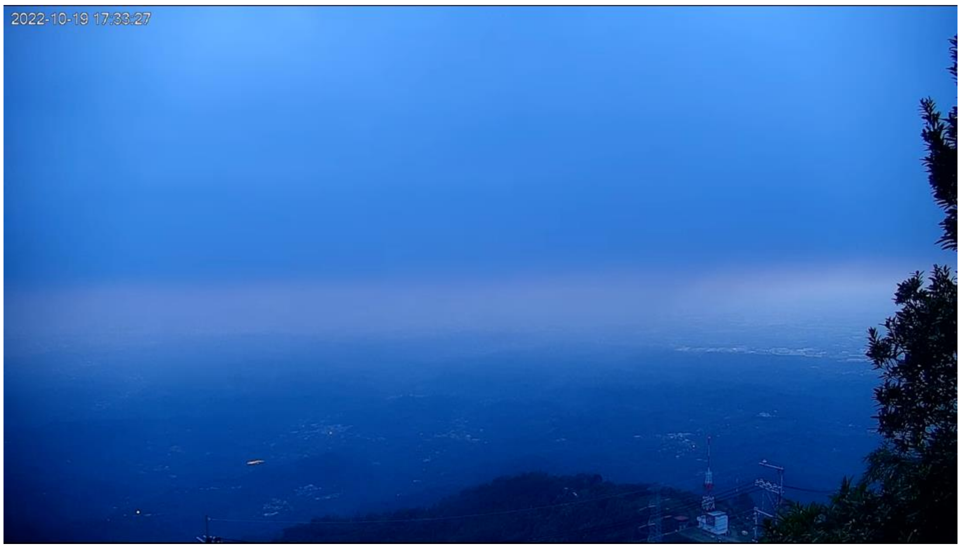
4. Result

As for the result, we use our program to predict the colorful cloud in these days. Here comes our result.

No.	Date	County	Observe time	Predict time	Temp	Press	Humidity	Result
1	8/26	Kaohsiung	18:23	18:15~18:30	31°	1008.1	70%	Correct
2	8/27	Kaohsiung	18:25	18:15~18:30	31°	1005.6	76%	Correct
3	8/27	Kaohsiung	18:52	18:15~18:30	31°	1005.6	76%	Correct
4	8/29	Kaohsiung	18:17	18:15~18:30	27°	1088.1	93%	Correct
5	10/19	Nantou	17:38	17:30~17:45	19.8°	902.8	88%	Correct
6	10/19	Chiayi	17:33	17:30~17:45	13.8	766.1	97%	Incorrect
7	10/19	New Taipei City	17:38	17:30~17:45	21.4	1018.0	68%	Correct
8	10/19	Taipei	17:39	17:30~17:45	21.9	1016.6	66%	Correct
9	10/23	Taoyuan	17:39	17:30~17:45	23	1018.0	83%	Correct

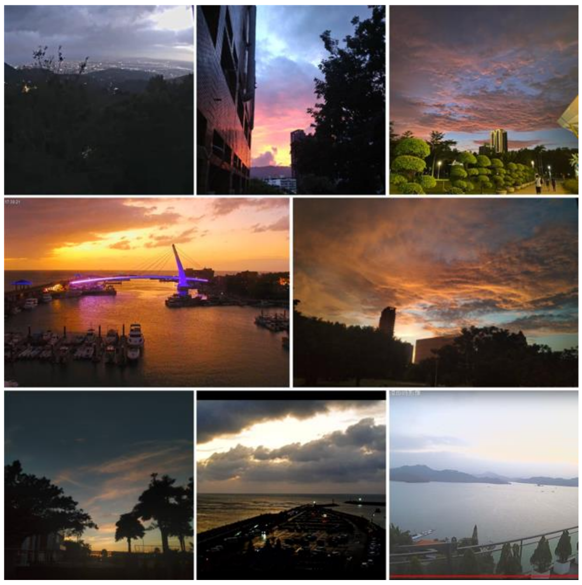
Note:
No.5 Observed in Sun Moon Lake National Scenic Area Administration
No.6 Observed in Alishan National Scenic Area
No.7 Observed in Tamsui
No.8 Observed in Yangmingshan National Park

Accuracy: 88%



▲Picture 4-1 The photo of No.6

As for the wrong prediction of No.6, which is observed in the Alishan National Scenic Area, we could clearly see in picture 4-1 that there still have some light at the edge of the horizon. However, due to the altitude of the observation location being too high, which the altitude of Alishan is nearly 2,400 meters, the weather often comes with a large amount of fog, which would impact the watching experience of the sunset cloud.



▲Picture 4-1 The photo of the observation, including No.1~5,7~9

In terms of our results, it appears that our system can predict over 85% of the results. Most of the predicted times did come with the sunset cloud appearing. But the model we create still has some problems. We can’t predict the result when there is extreme weather, including typhoons, heavy rain, and fog. It might come with the wrong prediction under these results.

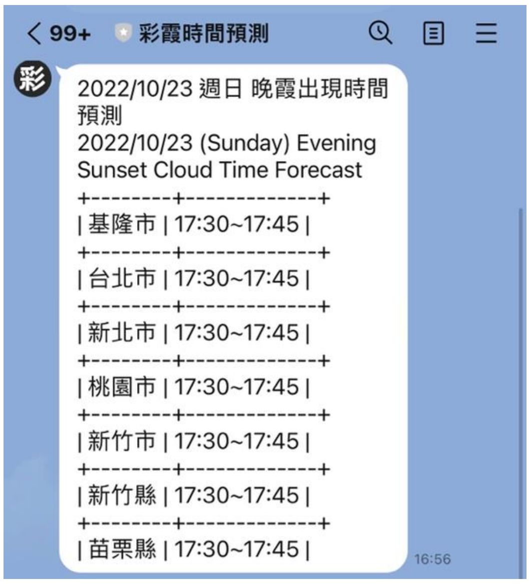
5. Conclusion and Future Prospect

In conclusion, our model to predict the exist time of the sunset cloud is stable and reliable. It is important to promote this prediction system to the public in order to satisfy our main purpose when designing this system, helping the tourism industry and tourists to watch the sunset cloud more conveniently.

5.1 Line Bot

Due to the predicting system, which is designed for Taiwan, it is better to promote to the public through Line. Line is an social application that about 95.7% of people in Taiwan used every day. Its robot account system is also good for us to announce our daily predictions.

Picture 5-1 is an example that we could use our prediction system throughout Line. People who have our official Line account can receive daily predictions every day.



2022/10/23 週日 晚霞出現時間預測
2022/10/23 (Sunday) Evening Sunset Cloud Time Forecast

```
+--------+-------------+
| 基隆市 | 17:30~17:45 |
+--------+-------------+
| 台北市 | 17:30~17:45 |
+--------+-------------+
| 新北市 | 17:30~17:45 |
+--------+-------------+
| 桃園市 | 17:30~17:45 |
+--------+-------------+
| 新竹市 | 17:30~17:45 |
+--------+-------------+
| 新竹縣 | 17:30~17:45 |
+--------+-------------+
| 苗栗縣 | 17:30~17:45 |
```

▲Picture 5-1 Example of Line

5.2 Cooperation with tourist attractions

We could also collaborate with national scenic areas or other tourist attractions that are known for sunset cloud watching, including Si Zih Bay and Yong-an Fish Harbor. The administrative center of the tourist attraction can hold a party or a bazaar to gather people. They can also organize mini-concerts to attract tourists to watch the stunning sunset cloud with beautiful, elegant music during the predicted time.

▲Picture 5-2 Yong-an Fish Harbor is known for its sunset view

6. References

[1] Bruneton, E., & Neyret, F. (2008). Precomputed Atmospheric Scattering. *Computer Graphics Forum*, *27*(4), 1079–1086. https://doi.org/10.1111/j.1467-8659.2008.01245.x

[2] Lopes, Diogo A. R. Fernandes, António Ramires (Ed.). (2014). *Atmospheric Scattering - State of the Art*. Instituto Politécnico de Leiria. https://repositorium.sdum.uminho.pt/bitstream/1822/30959/1/epcg2014_final%20submission_no_numbers_19.pdf

[3] Trager, Scott. "The Earth's atmosphere: seeing, background, absorption & scattering" (PDF). S.C. Trager. Retrieved 31 May 2022.

[4] Jenkins, F. A., White, H. E., & Brukhard, D. G. (1958). Fundamentals of Optics. *American Journal of Physics*, *26*(4), 272. https://doi.org/10.1119/1.1996127

[5] Phokate, S. (2017). Atmospheric water vapor: Distribution and Empirical estimation in the atmosphere of Thailand. *Journal of Physics: Conference Series*, *901*, 012051. https://doi.org/10.1088/1742-6596/901/1/012051

[6] A Solar Altitude Angle Model for Efficient Solar Energy Predictions. (2020, March 4). In *Dr. Sergio Herrería-Alonso*. https://www.mdpi.com/1424-8220/20/5/1391/pdf?version=1583316673

[7] 穆磐石. (2018, October 15). *Li Shangyin, "Going Up to the Pleasure Garden"（李商隱登樂遊原）*. Https://Www.Nd.Edu/. Retrieved August 29, 2022, from http://sites.nd.edu/peter-moody/2018/10/15/li-shangyin-going-up-to-the-pleasure-garden%EF%BC%88%E6%9D%8E%E5%95%86%E9%9A%B1%EF%BC%8C-%E7%99%BB%E6%A8%82%E9%81%8A%E5%8E%9F%EF%BC%89/

[8] NASA GISS. (n.d.). *NASA GISS: Panoply 5 netCDF, HDF and GRIB Data Viewer*. National Aeronautics and Space Administration. Retrieved August 30, 2022, from https://www.giss.nasa.gov/tools/panoply/

[9] 政府資料開放平臺. (n.d.). *政府資料開放平臺-關於平臺*. Retrieved August 30, 2022, from https://data.gov.tw/about

[10] National Space Organization. (n.d.). *FORMOSAT-7 - National Space Organization*. © 2019 National Applied Research Laboratories National Space Organization. Retrieved August 30, 2022, from https://www.nspo.narl.org.tw/inprogress.php?c=20022301&ln=en

[11] Wikipedia contributors. (2022, July 29). *AccuWeather*. Wikipedia. Retrieved August 30, 2022, from https://en.wikipedia.org/wiki/AccuWeather

[12] Central Weather Bureau. (n.d.). *opendata.cwb.gov.tw*. Https://Opendata.Cwb.Gov.Tw. Retrieved August 30, 2022, from https://opendata.cwb.gov.tw/about/opendata

【評語】190026

This project proposes a computational model using Cauthy’s equation，theorems and based on the temperature，pressure，and humidity provided by the government platform，to predict the accurate time when the sunset cloud will appear. The project has its novelty. The idea is good. The run experiments to justify the performance of the proposed model. The time is divided into time slot (15 mins/timeslot). The project has its novelty. The idea is good.

Some comments are given below:

More explanation is suggested to reason why Cauthy’s equation，theorems are applied. It would be better if more experiments are run to study the performance of the proposed model.