**Python note from udemy python course**

**การติดตั้ง library ทำโดยการใช้ cmd**

คำสั่ง

Pip install pandas

Pip install jupyter

Pip install glob

Pip install numpy

**File handling**

Glob เป็น library ที่เอาไว้ใช้ในการ get ชื่อไฟล์ตามประเภทที่ต้องการมาเก็บไว้ใน list

Open เป็นคำสั่งที่ใช้ในการเปิดหรือสร้างหรือเพิ่มเติมข้อมูลในไฟล์ที่มีอยู่แล้วโดยมี pattern

File = Open(“ชื่อไฟล์”,”ประเภทคำสั่ง”)

ประเภทคำสั่ง – w,w+,r,r+,a,a+

File.read,File.write

Datetime เป็นฟังก์ชัน get ค่าเวลาที่ติดมากับ python การใช้เช่น now=datetime.datetime.now()

Strftime เป็น method เพื่อเปลี่ยนเวลาเป็น string ในแบบที่เราต้องการ เช่น now.strftime("%Y-%m-%d-%H-%M")

**Jupyter notebook – เป็น tool ที่ใช้ในการทำงานกับข้อมูลใช้กับ python**

การเปิดทำโดย cmd -> jupyter notebook(อันนี้รันบน tools jupyter)

หรือ cmd -> ipython (อันนี้รันบน cmd)

Jupyter key – ctrl+enter(รันคำสั่ง),shift+enter(ขึ้นคำสั่งใหม่)

Import os

Os.listdir() – นำค่าไฟล์ที่อยู่ในโฟลเดอร์มาแสดง

Import pandas – pandas เป็น library ของ python ที่ใช้ในการทำ datascience

Df1=pandas.read\_csv(“ชื่อไฟล์.csv”) – นำข้อมูลในไฟล์มาแสดง

Df1.set\_index(“ชื่อคอลัมน์”) -นำคอลัมน์นั้นมาไว้หน้าสุด

Df1.shape – ขนาดตาราง

Df1=pandas.read\_json(“ชื่อไฟล์.json”)

df3=pandas.read\_excel("supermarkets.xlsx",sheetname=0) 0= sheet1,1=sheet2

pandas.read\_csv? – เปิดคำอธิบาย

df2.loc["735 Dolores St","Country":"ID"] – แสดงค่าตามนี้

list(df2.loc[:,"Country"]) – สร้างlist ตามในวงเล็บ

df2.iloc[1:3,1:3]

df2.iloc[:5,1:3]

df2.ix[3,4]

df2.drop("City",1)

df2.columns

df2["Continent"]=df2.shape[0]\*["North America"] – เพิ่มคอลัมน์ต้องมีข้อมูลเท่ากันด้วย

df2["Continent"]=df2["Country"]+","+"North America" – แก้ไขข้อมูลในคอลัมน์ชื่อ continent

df2\_t=df2.T – สลับแถวเป็นหลัก

n=nom.geocode("3995 23rd St,San Francisco,CA 94114") – convert string to location

dfx["Address"]=dfx["Address"]+","+dfx["City"]+","+dfx["State"]+","+dfx["Country"] – เปลี่ยนค่าในคอลัมน์ address

dfx["Coordinates"]=dfx["Address"].apply(nom.geocode) – ใส่ค่า location เพิ่มไปในตาราง

dfx["Latitude"]=dfx["Coordinates"].apply(lambda x: x.latitude if x != None else None) – เพิ่มค่า latitude แต่ใส่เงื่อนไข

dfx["Longitude"]=dfx["Coordinates"].apply(lambda x: x.longitude if x != None else None)

**Numpy**

Numpy เป็น base library ของ pandas และ opencv และอีกหลายๆตัว

ใช้ทำ image processing เพราะมีค่า array มากไม่สะดวกในการทำใน jupyter notebook

import numpy

n=numpy.arange(27) –สร้าง array ขนาด 27 ช่อง

การจัดค่า array ของ numpy จะมีระบบชัดเจนทำให้ทำงานง่ายกว่า ใน python ปกติ

n.reshape(3,9)

import cv2

im\_g=cv2.imread("smallgray.png",0) – 0 คือใช้ใน gray scale , 1 คือใช้ใน bgr(blue,green,red)

cv2.imwrite("newsmallgray.png",im\_g) – สร้างภาพใหม่โดยเอาภาพเก่าไปใส่

**Folium**

เป็น library ที่ใช้ในการสร้าง map ของ python

Import folium

Map = folium.Map(location=[80,-100]) - สร้าง map เก็บไว้ในตัวแปร map

fg=folium.FeatureGroup(name="My Map")

fg.add\_child(folium.CircleMarker(location=[lt,ln], radius=8,color=color\_producer(el), fill\_color=color\_producer(el), popup=na+" "+str(el)+' m'))

map.add\_child(fg)

Map.save(“Map1.html”)

**Tkinter**

เป็น library สำหรับการสร้าง UI ของ python แบ่งเป็น grid row และ column

from tkinter import \*

window=Tk()

def km\_to\_miles():

print(e1\_value.get())

miles=float(e1\_value.get())\*1.6

t1.insert(END,miles)

b1=Button(window,text="Execute",command=km\_to\_miles)

b1.grid(row=0,column=0)

e1\_value=StringVar()

e1=Entry(window,textvariable=e1\_value)

e1.grid(row=0,column=1)

t1=Text(window,height=1,width=20)

t1.grid(row=0,column=2)

window.mainloop()

sqlite เป็น database ที่ไม่ต้องติดตั้งเพราะติดมากับ python

**การทำ python file เป็น .exe**

1.Pip install pyinstaller

2.Pyinstaller ไฟล์ที่ต้องการทำ

หมายเหตุ pyinstaller ไม่ support python 3.6

**การplot graph ใน jupyter notebook**

ใช้ bokeh – เป็น library ที่ใช้ในการเขียน graph ของ python ใช้กับ jupyter notebook

**Test plot**

from bokeh.charts import Scatter,output\_file,show

import pandas

df=pandas.DataFrame(columns=["X","Y"])

df["X"]=[1,2,3,4,5]

df["Y"]=[5,6,4,5,3]

p=Scatter(df, x="X", y="Y",title="Temperature", xlabel="Day of observation",ylabel="Temperature")

output\_file("Scatter\_charts.html")

show(p)

**test2**

from bokeh.plotting import figure,output\_file

p=figure(plot\_width=500,plot\_height=400)

p.triangle([1,2,3,4,5],[5,6,5,5,3], size=12,color="red",alpha=0.5)

**-figure เอาไว้ใช้เพื่อทำให้จุดที่ plot เป็นรูปตามที่ต้องการ**

**-size สามารถปรับเป็น size=[2,3,4,5,6] เพื่อให้แต่ละจุดใหญ่ไม่เท่ากัน**

**Web Scraping**

**-ใช้ในการ get ค่าจาก code ของ html ของ web ต่างๆ**

**-ใช้ร่วมกับ jupyter notebook**

import requests

from bs4 import BeautifulSoup

r=requests.get("http://pythonhow.com/example.html")

c=r.content

soup=BeautifulSoup(c,"html.parser") //จัด code ให้อยู่ในรูปที่สวยงาม

print(soup.prettify())

all=soup.find\_all("div",{"class":"cities"}) //ตัด code ใน tag มาไว้ใน list

all[2].find\_all("h2")[0].text //เลือก tag .ใน list มาแสดง

for item in all:

print(item.find\_all("p")[0].text)

web scraping แบบ multiple page ดูจากไฟล์ multiple ใน folder app7

**Financial data analysis**

ใช้ pandas datareader สามารถ get ค่าราคาหุ้นจากฐานข้อมูลต่างๆได้

โดยจะใช้ bokeh ในการวาดกราฟ candle-stick

**การสร้าง web app ด้วย python และ flask**

สร้าง folder static(เก็บไฟล์ css), templates(เก็บไฟล์ html)

ใช้ command line สั่ง

1. Python –m venv virtual (เพื่อสร้าง folder virtual ในการรัน web)
2. Check flask(โดย ไปที่ virtual\Scripts\activate) แล้วกด python แล้วลอง import flask
3. ถ้าไม่มี pip install flask
4. Run server โดย ไฟล์ python ที่อยู่นอก folder virtual(ในไฟล์ python นี้จะต้องมีคำสั่งให้ไปเรียกไฟล์ html)