**HW1 Link Prediction**

**M10602141\_李弈鴻**

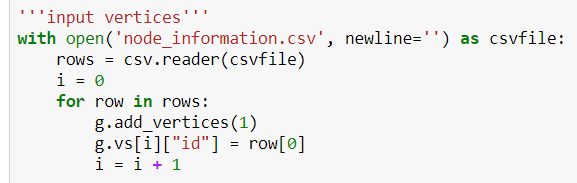
**Development :**

Language: Python( lib: igraph )

Tools: jupyter

**Code :**

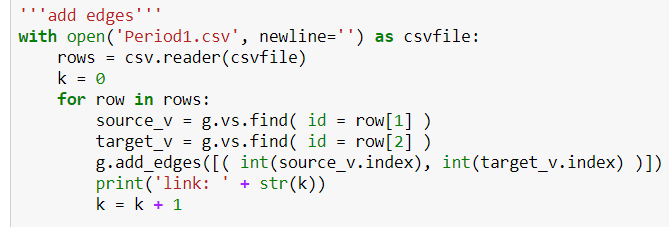
Step1



用node\_information.csv 進行vertices create

並且把每個node放置他們的id

Step2



用Period1.csv 解析後，設定彼此的edge

Step3



Edges設定完後將每兩個node都跑過一次shortest path

用的function是 Igraph-> shortest\_paths\_dijkstra()

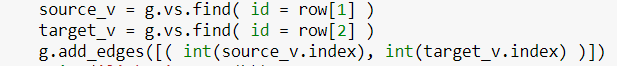
跑的時間非常非常久

並且得到的結果準確率相當低

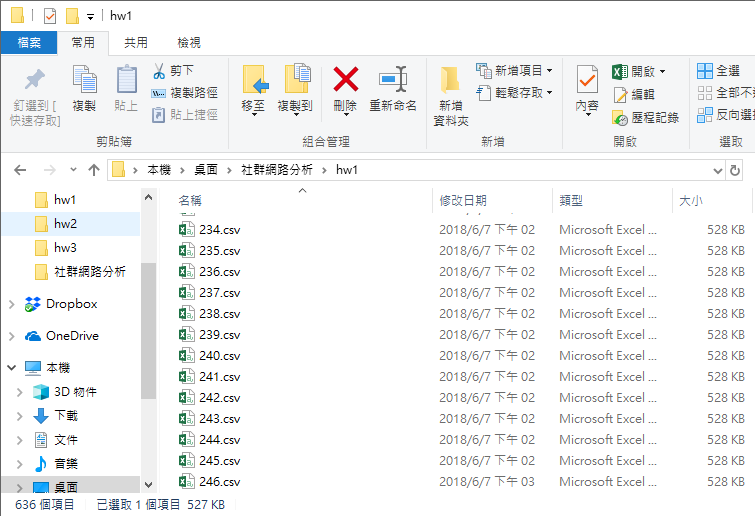
之後只用 Period1.csv 跟 Period.csv 做transitivity 才可以得到較高的準確率

**Problems:**

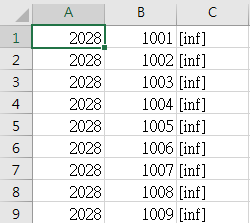
點的數量非常多且每個node名稱不是流水編號，在edges set時要去find()，可能是實驗室電腦比較不好，edges連接就要花大概一個多小時，之後才想到有binary search方式可以解決。



另外在跑shortest path時，也會花相當多的時間，當時測試要跑完27770個點要花一個月的時間，所以只用了一部份的資料去train。



->528kb代表shortest path都是無限



只有shortest path這個變數還是不夠的，shortest path是無限的data占了絕大多數，導致於判斷test data也絕大多數是0。

之後用edges set 兩個period 做transitivity才可以拿到比較好的準確率。

**Learn in this homework:**

因為目前電腦OS是windows 10，在灌igraph時相當不友善，許多網路上評論都是”disaster”來稱呼，igraph灌完不會出現問題。

可是在灌plot畫圖套件時會不斷發生error，花了許多時間去解決這個事情。

當資料量過多時，也會花比較多的時間去想說是什麼樣的因素會造成電腦跑程式速度很慢，以前寫程式比較不會遇到這些問題。有時候也會因為硬體性能而造成速度上差異，或是記憶體爆掉，需要更多的想法去解決這些事情。

這次寫這個也是相當有挑戰性，以前沒有碰過這部分的東西，許多演算法聽起來也相當有趣，再根據不同的特性去預測未來有沒有機會連接，一次摸到許多領域的東西，也學到了很多。