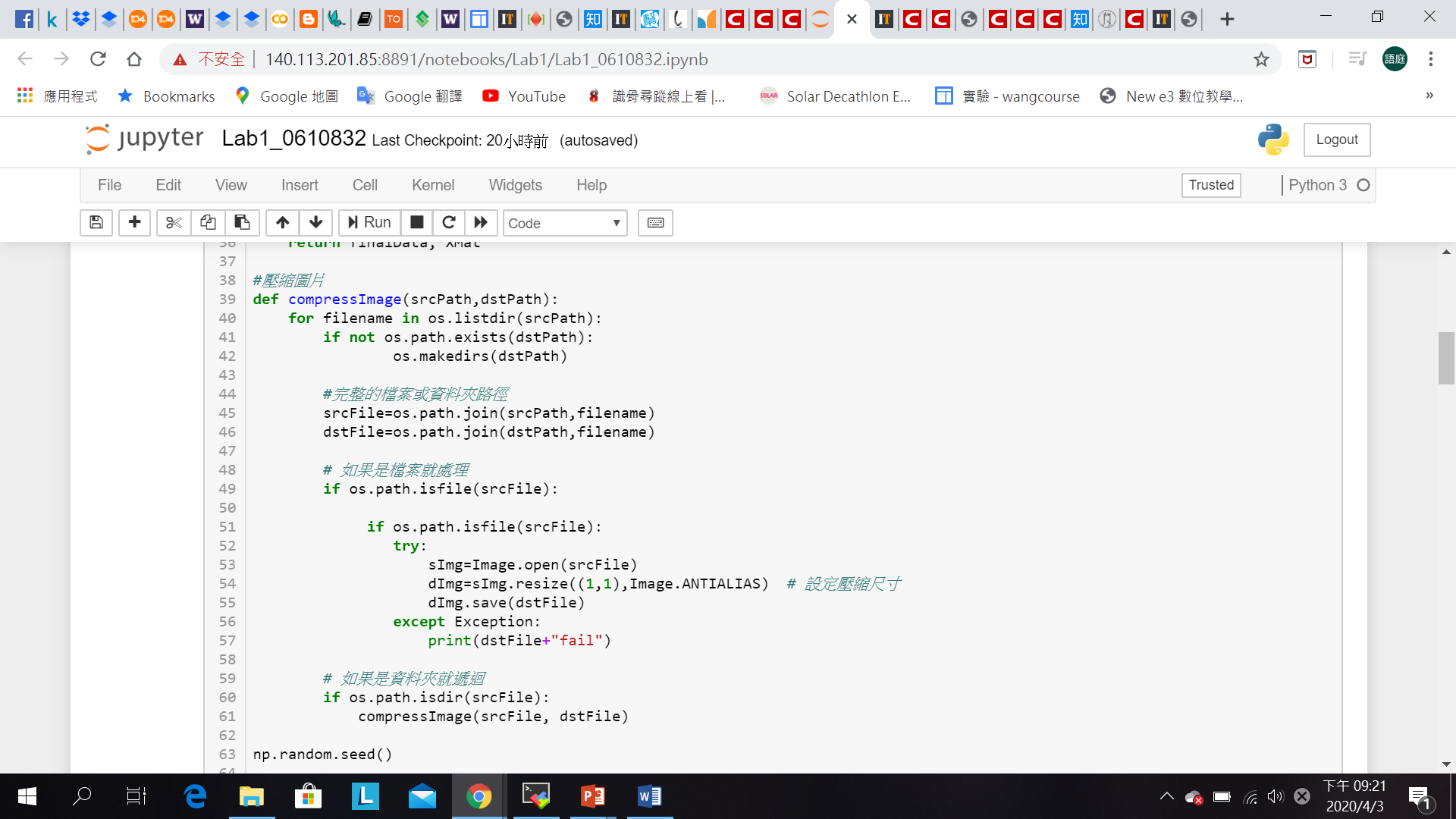
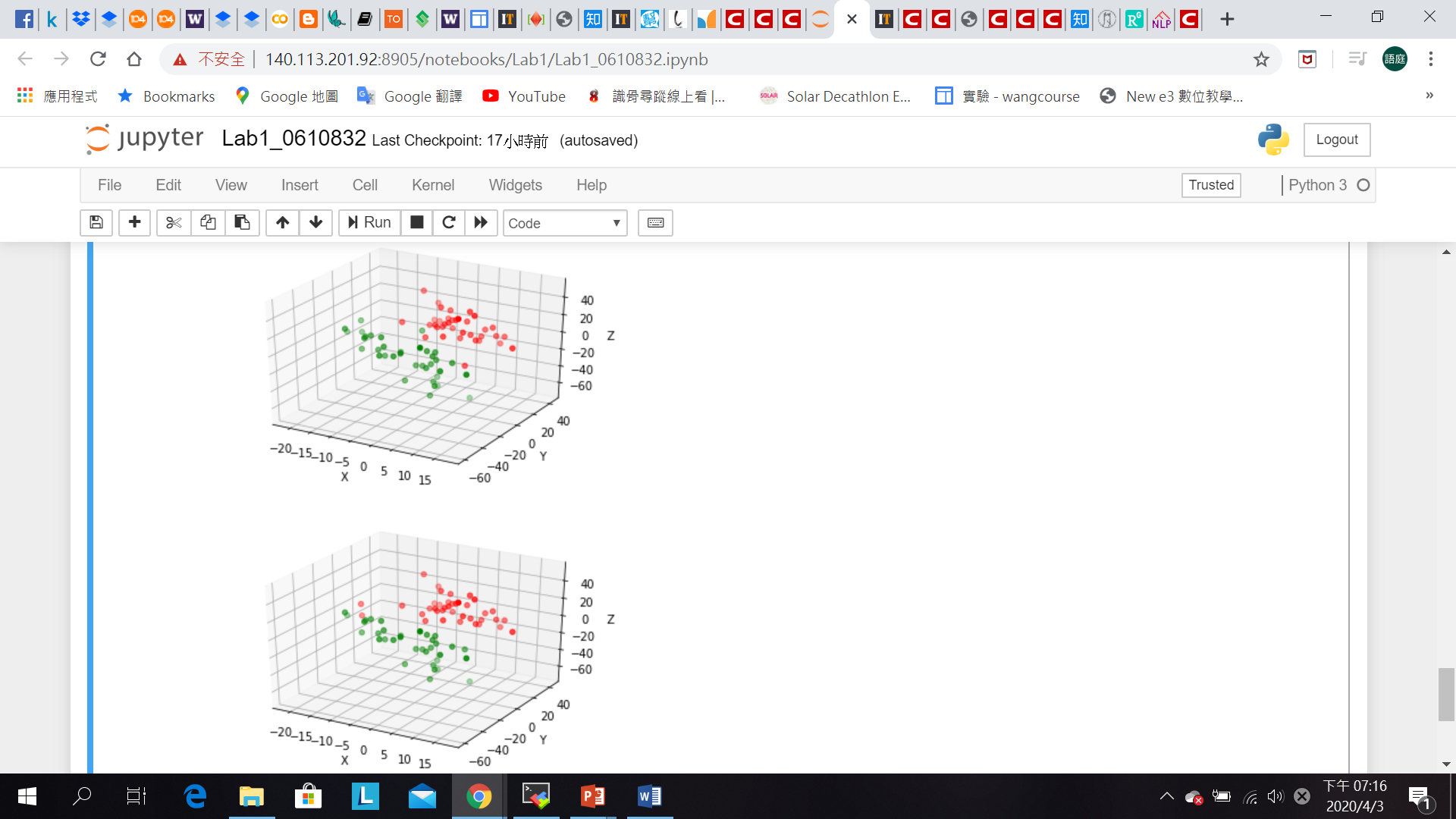
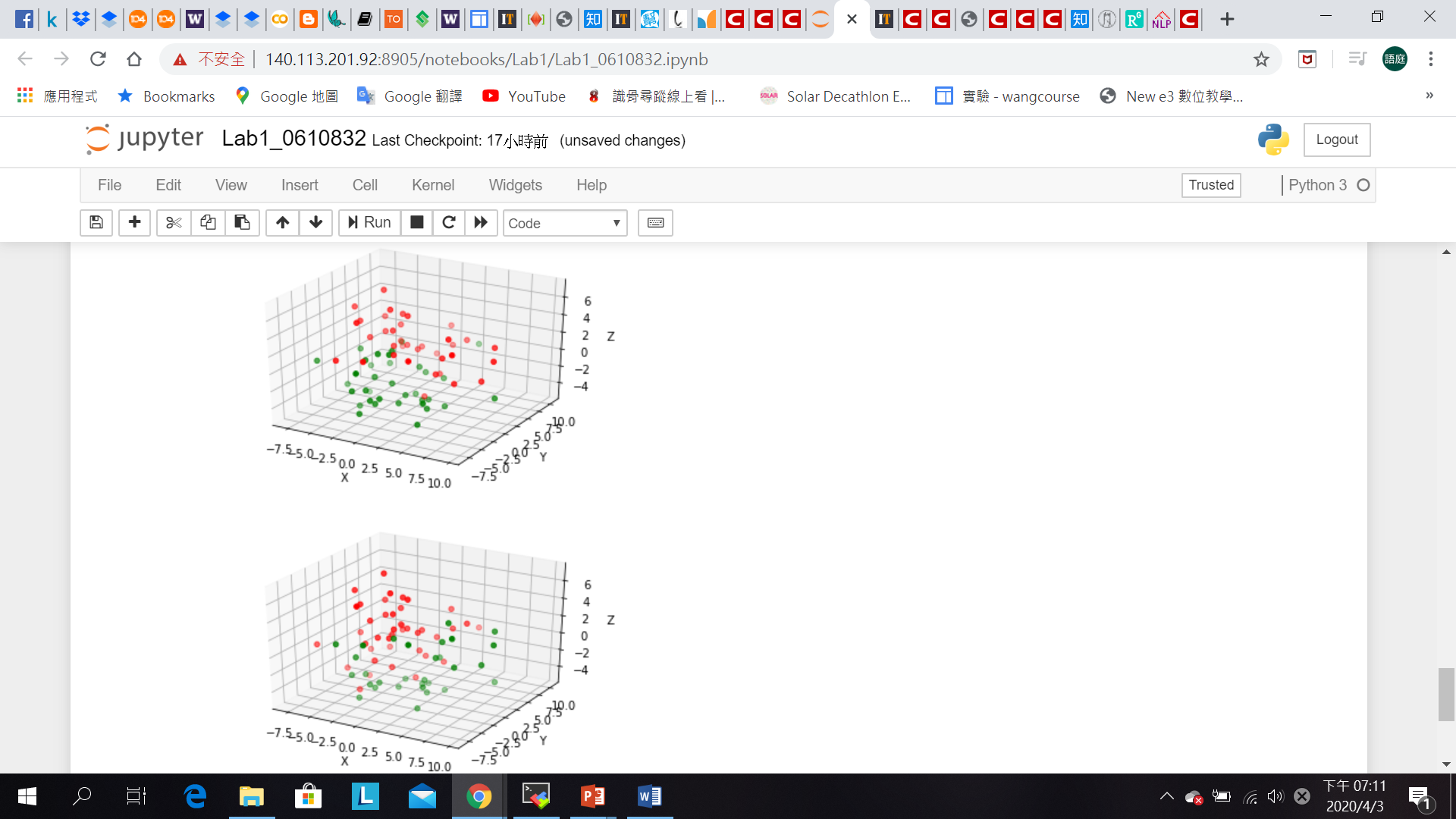
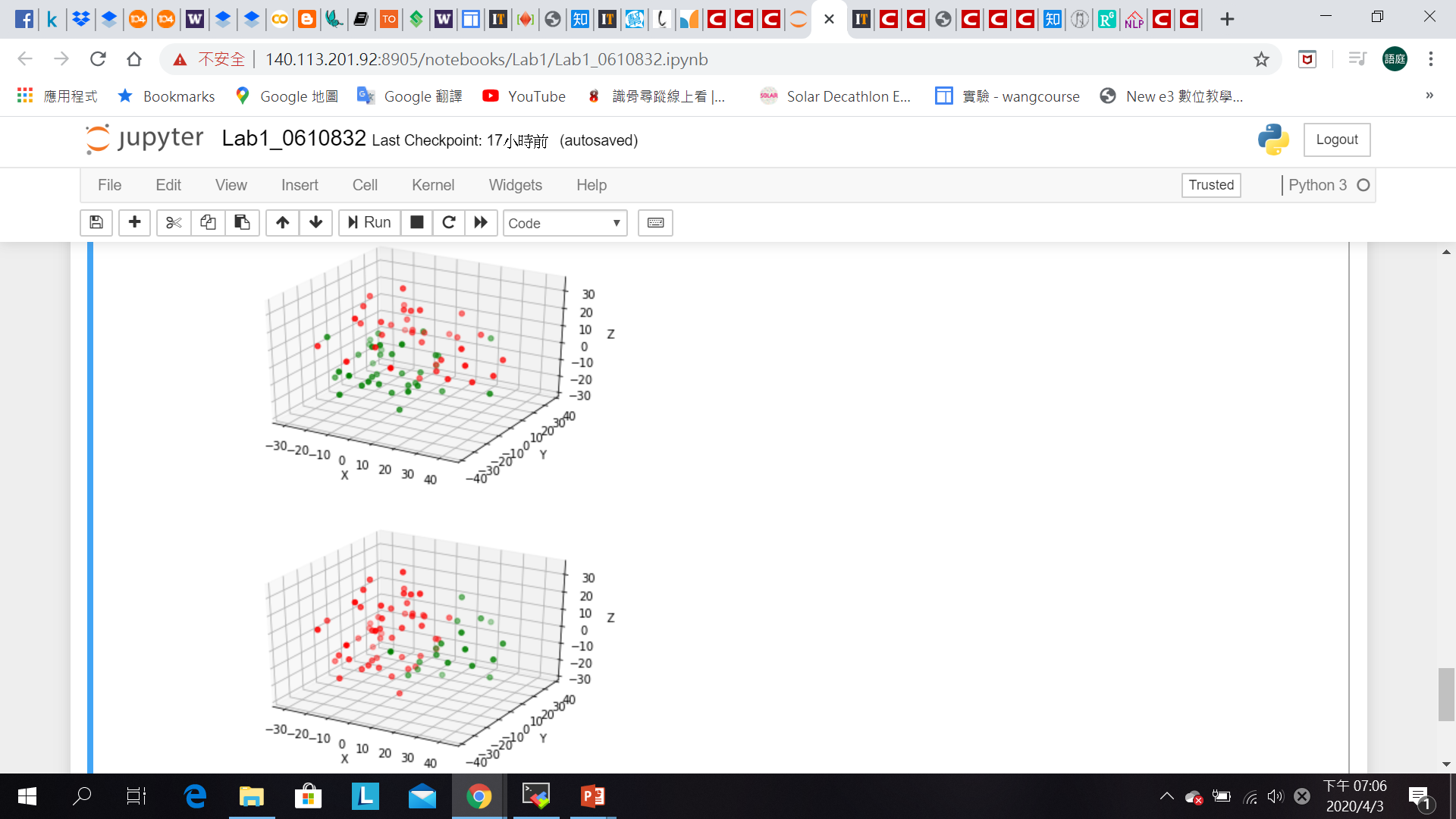
Lab1 PCA+K-MEANS

1. 壓縮和讀檔



一開始在做PCA時因為memory不夠，無法儲存這麼大的投影矩陣，所以決定在讀檔處理圖片時，就先壓縮圖片。

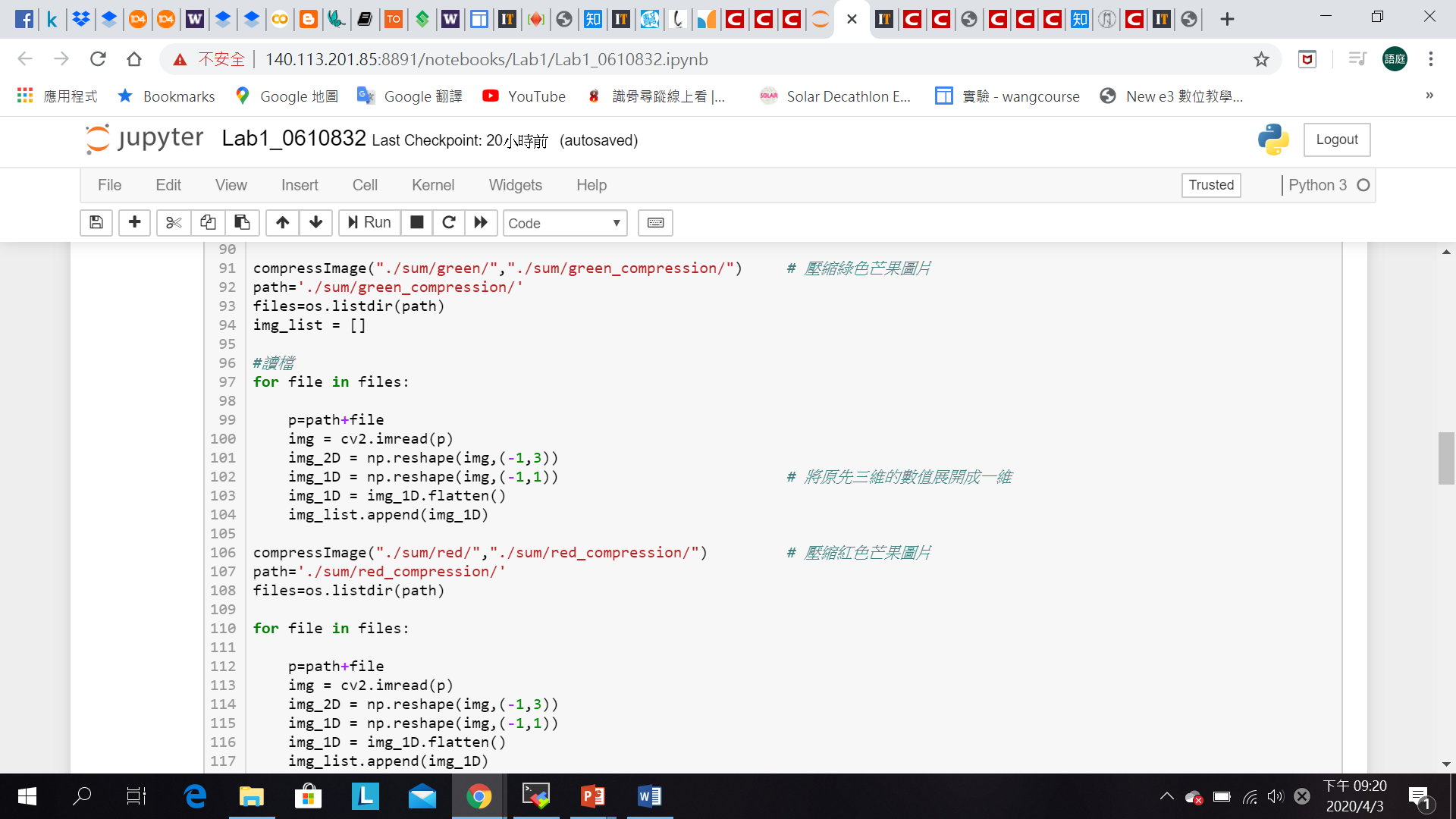


5\*5

1\*1

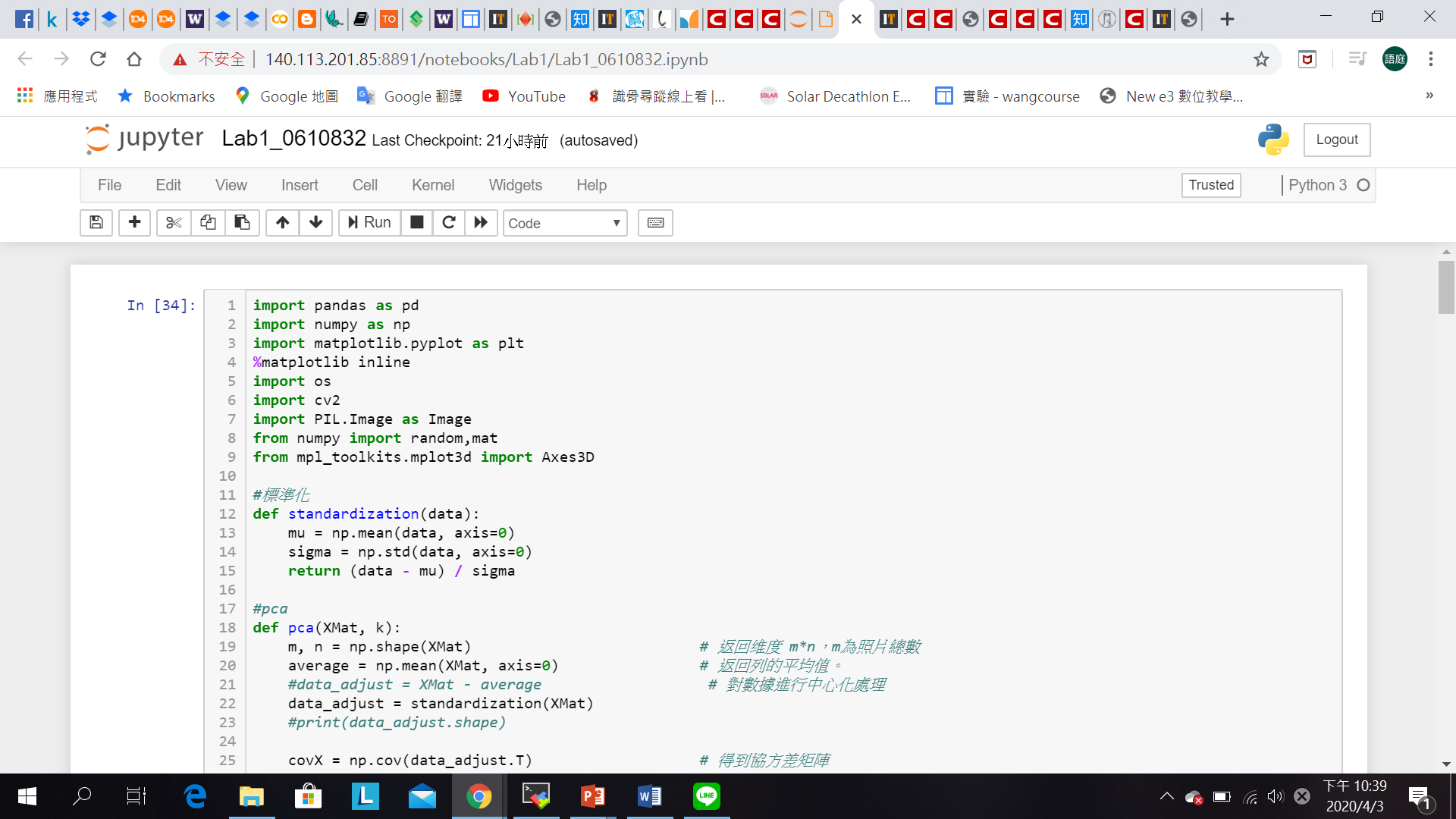
25\*25

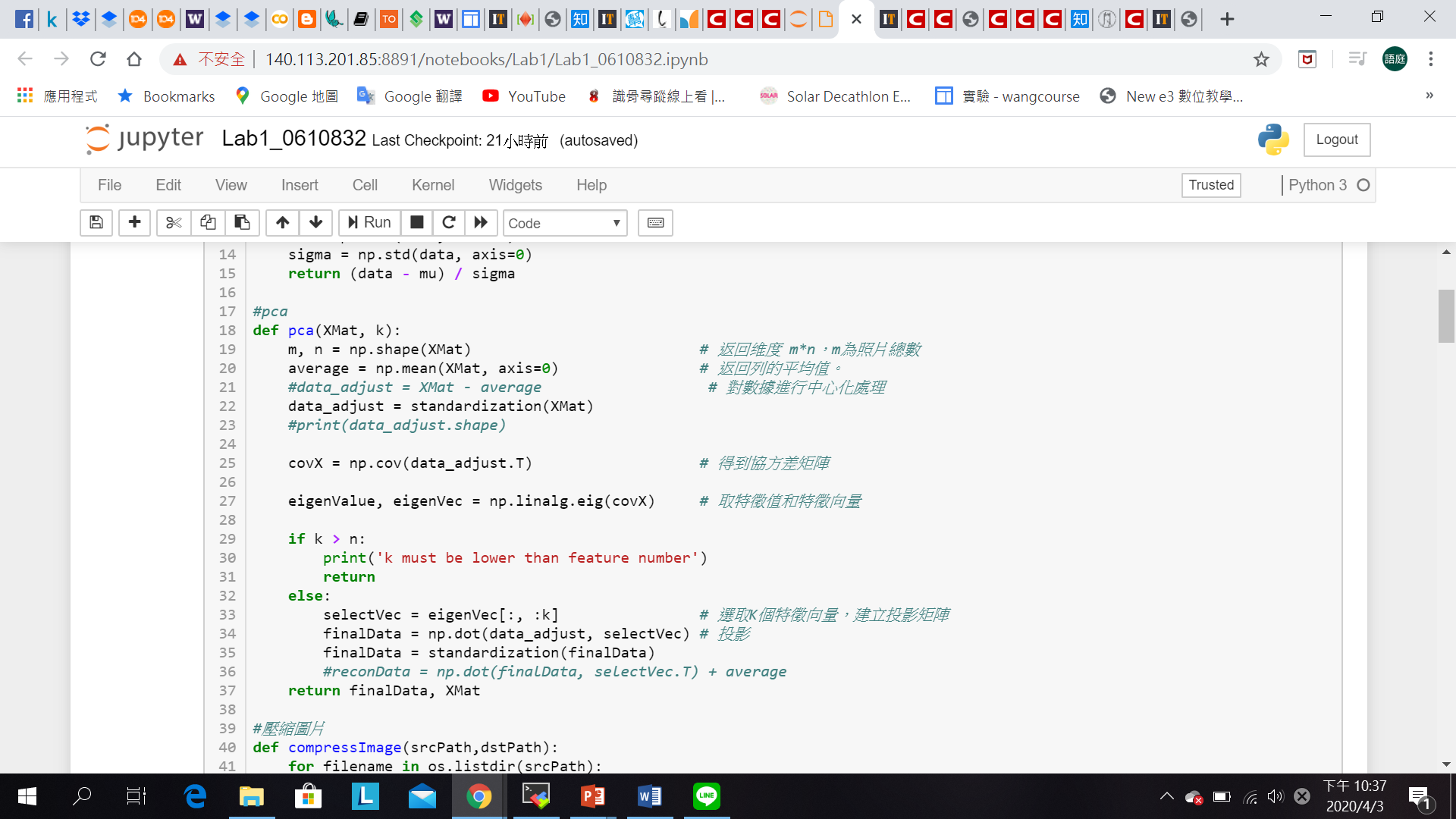
嘗試將不一樣大的圖片統一壓縮成不同的大小後，發現壓縮得越小，在分類的表現也會越好，同時注意到當我把圖片壓縮成1\*1時，經過PCA降維處理後，在三維空間的分布跟前面相比呈現比較大的區別。

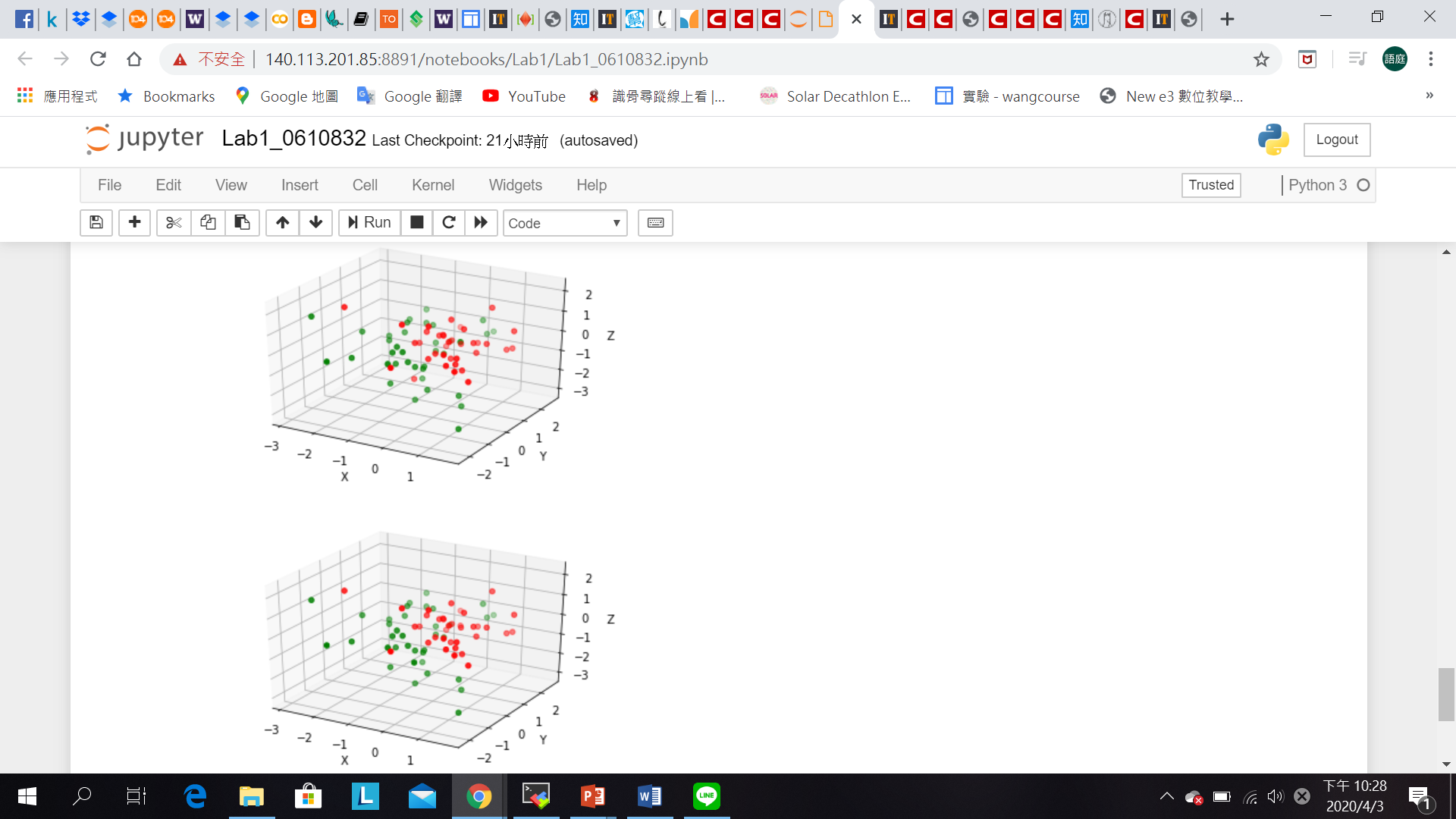
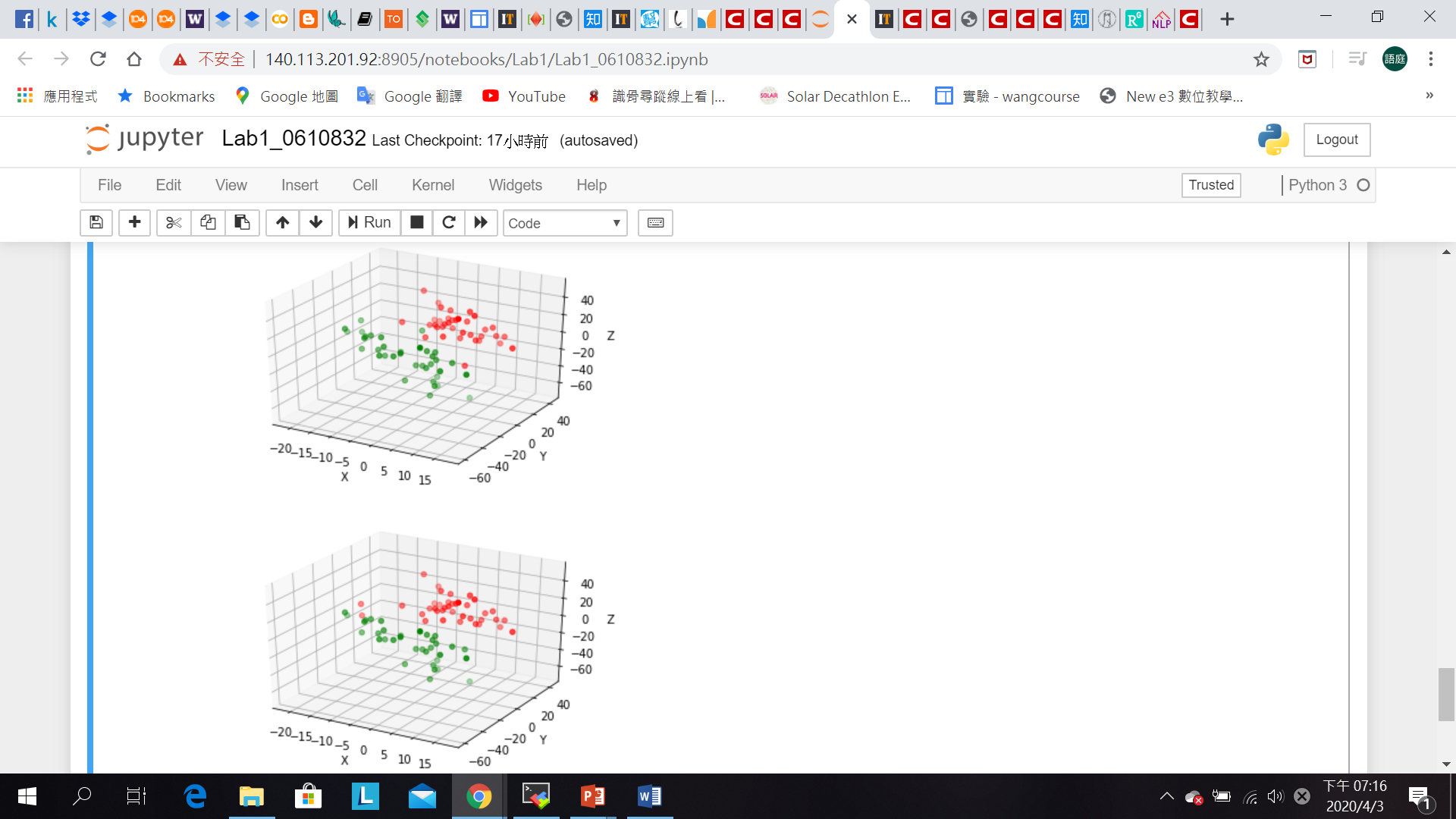


原本我是將每張圖片降維成二維array，大小是一張圖片壓縮過後的element個數\*每個樣本的屬性值個數(3)，然後對每張照片分別進行PCA處理，但結果顯示紅綠分布很明顯混雜在一起，所以後來改成先將每一張照片樣本的所有元素拉為一為陣列，再將代表每一張照片的array組成matrix，最後一起丟入PCA進行處理。

1. PCA



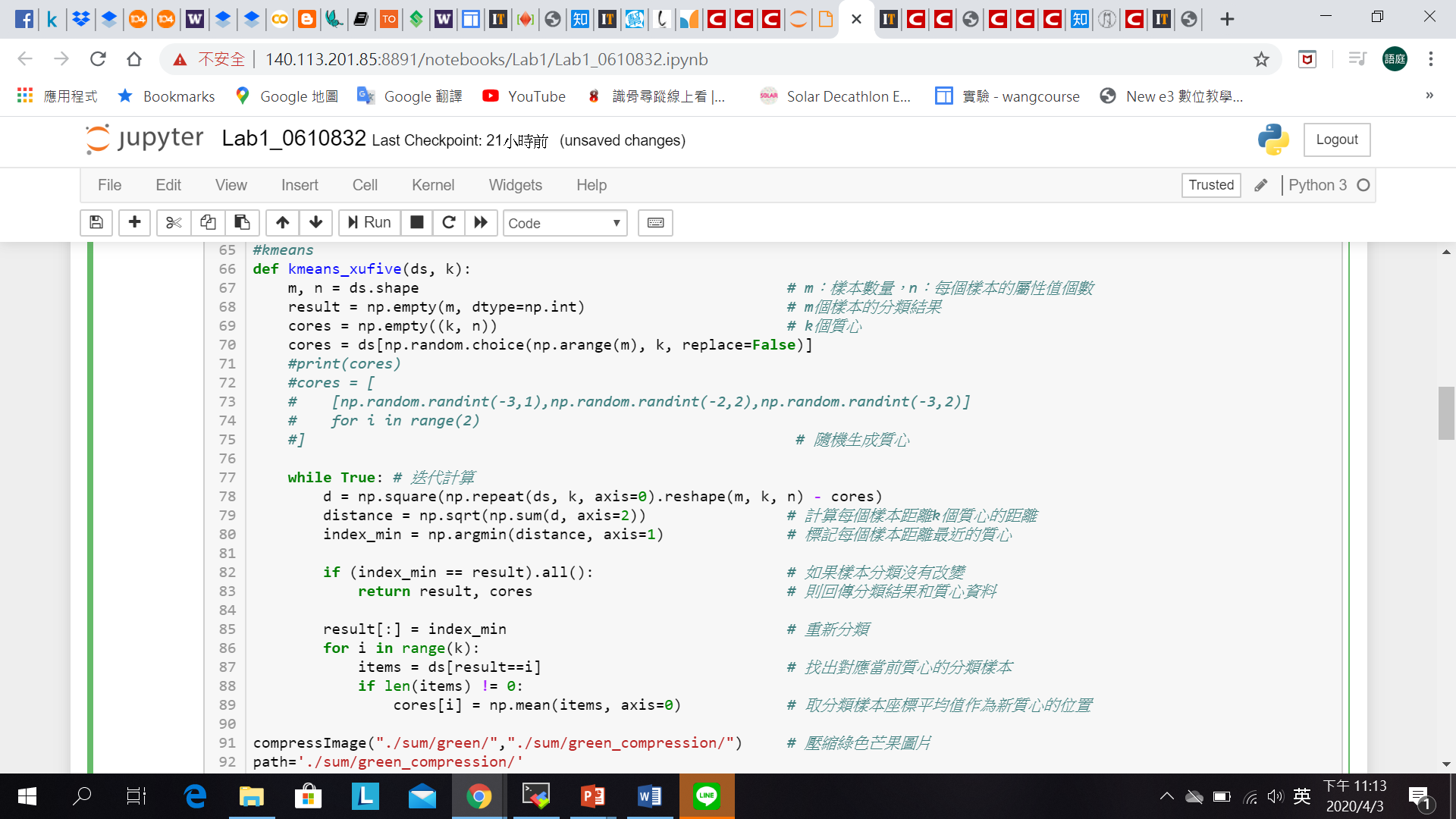


針對要準備進行PCA處理的數據，一開始我是採中心化處理，讓他們與平均相減，後來我又嘗試標準化的處理，分類效果也有提升。

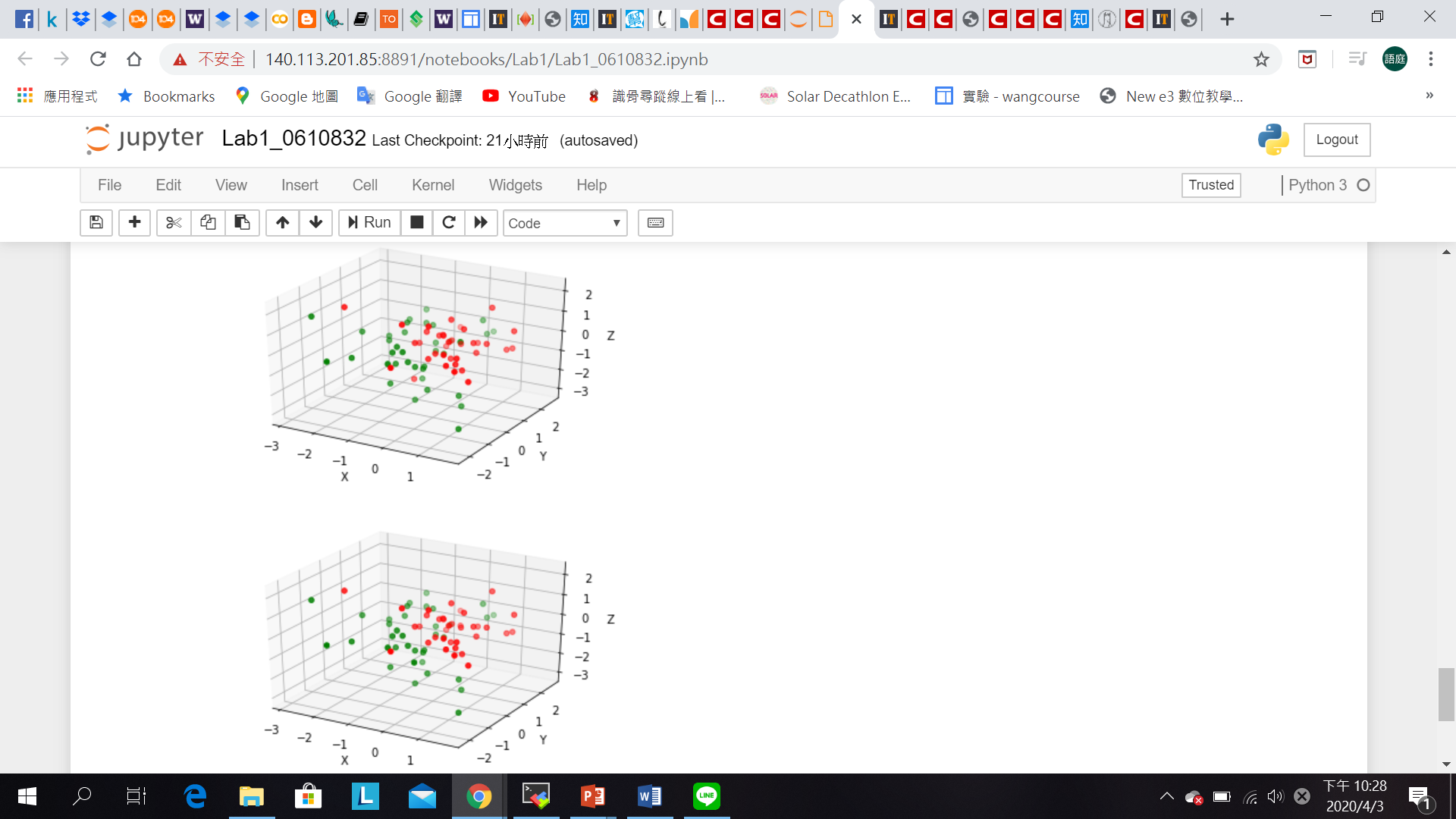
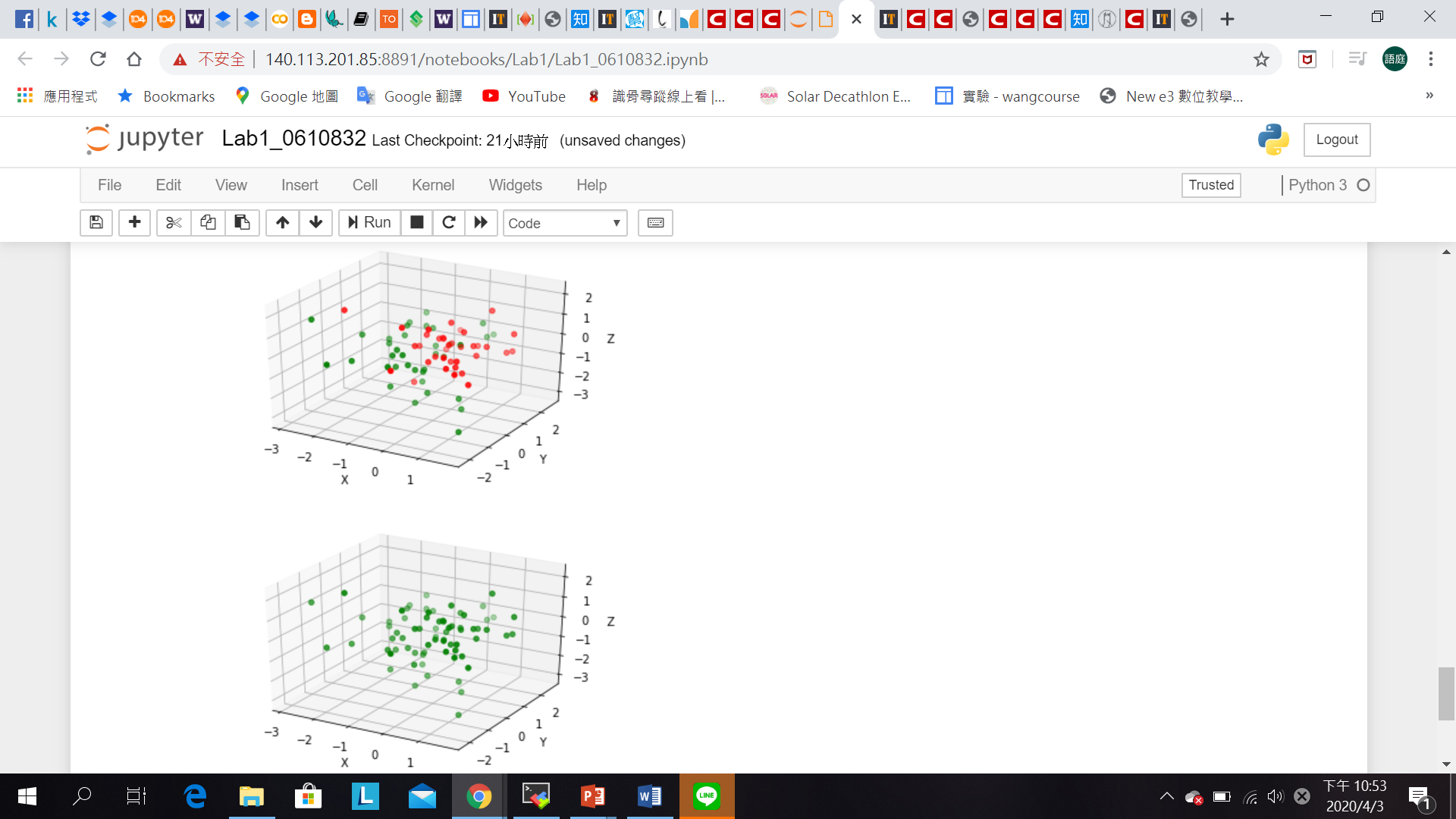
標準化

中心化

1. K-MEANS



原先我是隨機生成兩個點做為分類的起始點，但後來發現這樣很容易讓分類不均，甚至遇到全部都分成同一類，所以後來我改成隨機取72個樣本數中的兩個點做為起始點，分類效果明顯好了很多。



隨機在空間中取點

隨機從樣本點取點