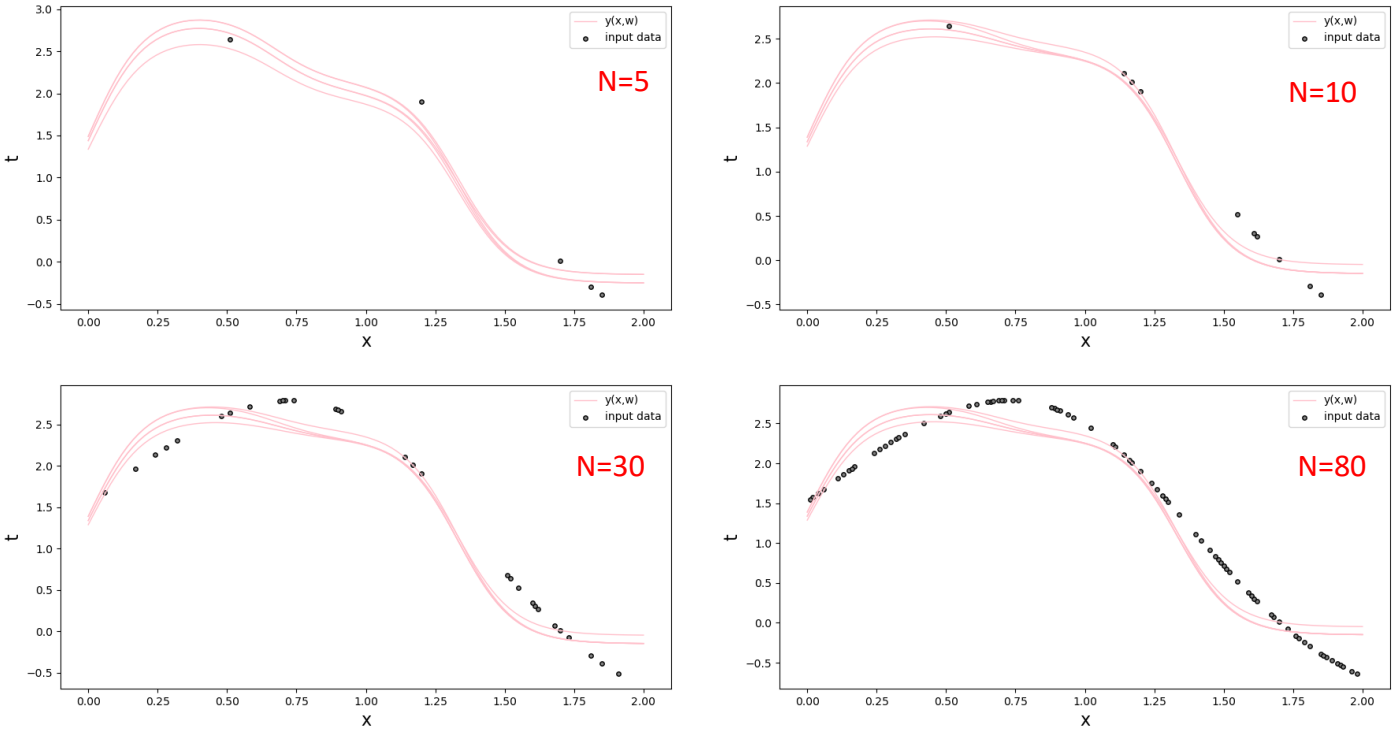
ML Hw2 報告

電機乙 0850736 楊登宇

1. Sequential Bayesian Learning

(1) Generate 5 curve samples

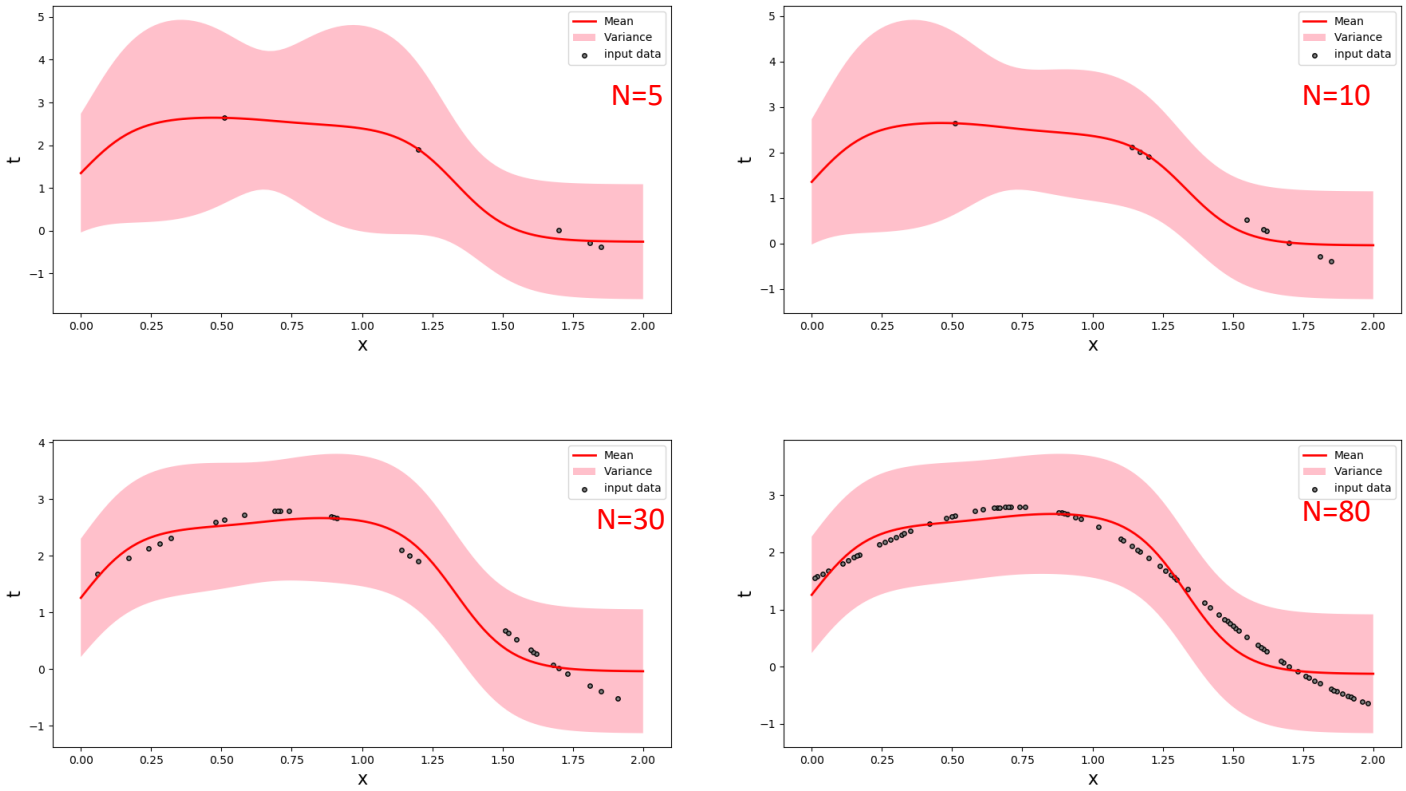


粉紅色的線為 y(x,w)，灰色的點為訓練資料的分布情形。

這題需要畫出 5 條 y(x,w)的線，但一開始我不太確定該如何給定其中的 w 參數。看課本的給法我一開始以為是隨機給定一組 w，但後來做完第三小題後才推測應該是要根據 sequential learning 時，我們所得到的 w 分布來取 w 的參數比較合理。

因此，根據第三小題所求出的 posterior distribution，我將 P(w)>0.8 的 w 分布來做取樣，取出 5 組的 w 參數來進行繪圖。從圖中可以發現，當給的訓練資料越多時，所取的 y(x,w)會相當貼近。但是不知道為何，所有的線在 x=1.5 後都會往水平線的趨勢走。不知道這樣的走向是否是正確的？

(2) Predictive distribution with mean curve and the region of variance



紅色線為 mean curve，粉紅色區域為 variance 加減平均後的區域，灰色點為訓練資料。這題求出訓練資料的平均與變異數相對簡單，不需要繁複的計算。我的作法是先將訓練資料的基底函數算出，將其放到Φ內，透過 posterior 分布的 mean 與 variance 公式即可透過簡單的矩陣乘法算出來。

其中 predict distribution 分布如下：

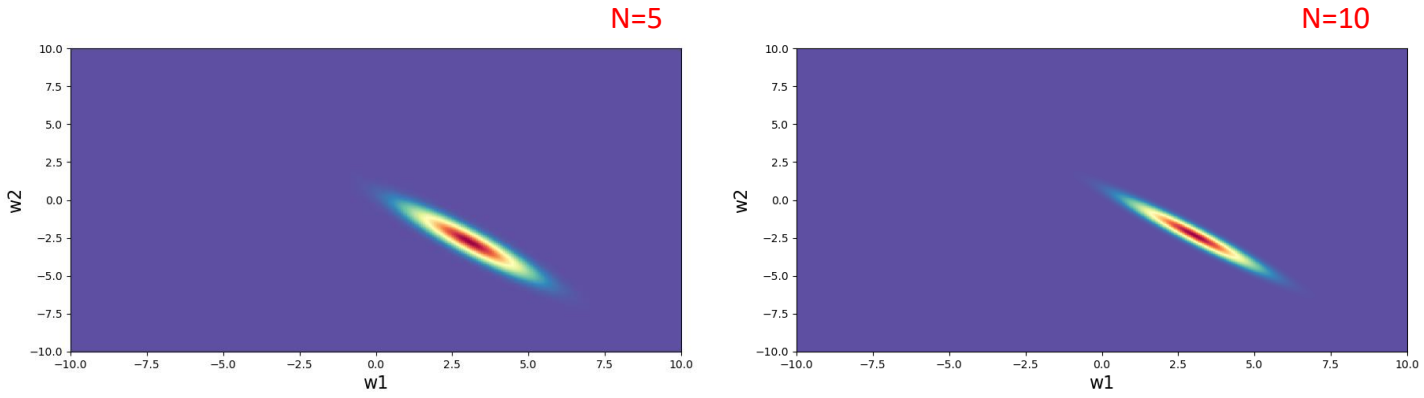
$$p(t|x) = \mathcal{N}\left(t|m(x), s^2(x)\right)$$

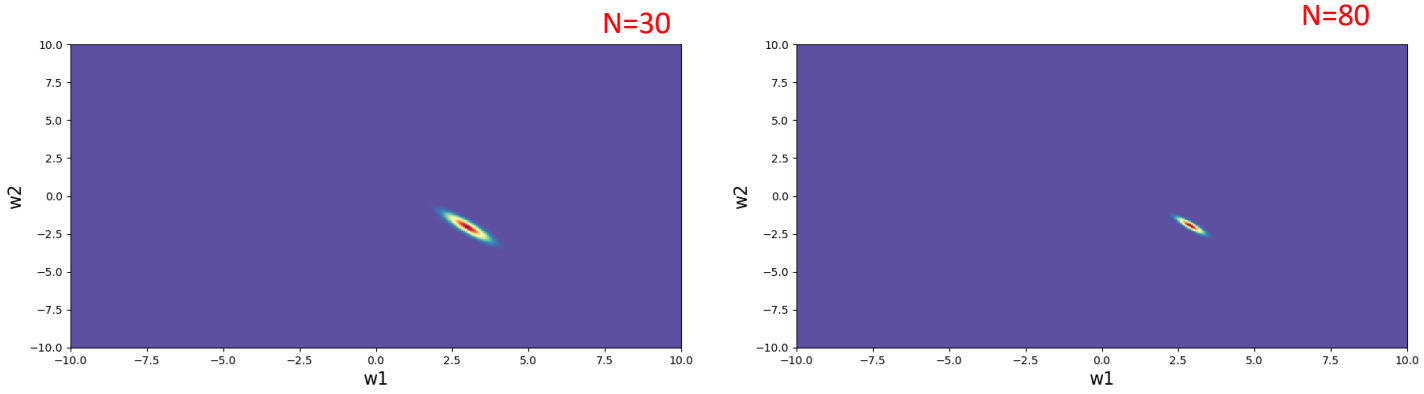
$$m(x) = \beta\phi(x)^T S_N \Phi^T t$$

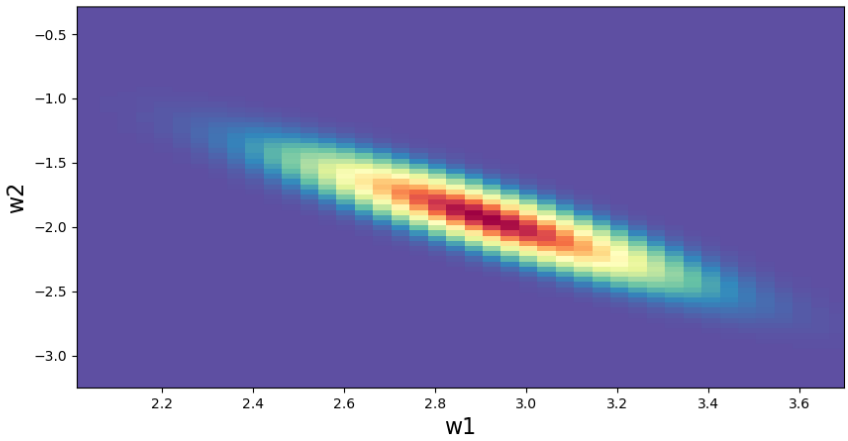
$$s^2(x) = \beta^{-1} + \phi(x)^T S_N \phi(x)$$

$${S_N}^{-1} = \alpha I + \Phi^T \Phi$$

(3) Select two weights and plot the corresponding prior distributions







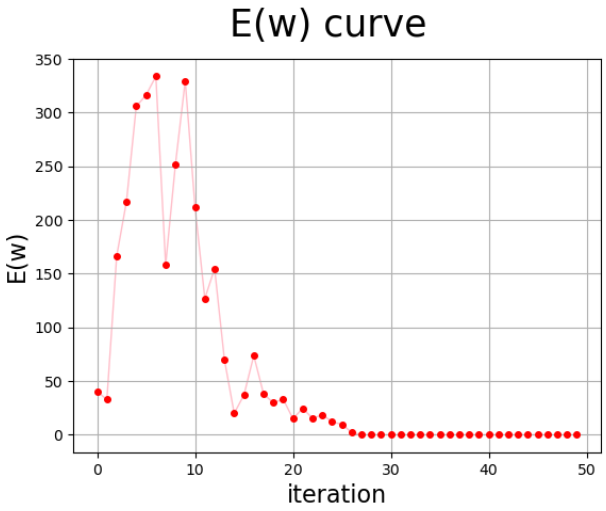
這個部份是我卡最久的地方，一開始看課本真的毫無頭緒，完全不懂這些圖是怎麼畫出來的，後來才看懂是先將一開始 prior 對 w 分布的情形(接近正圓)乘上得到訓練資料的 likelihood 分布，會得到 posterior 的分布。接著再將 posterior 分布更新成為下一個 prior 分布，繼續重複上述的過程。

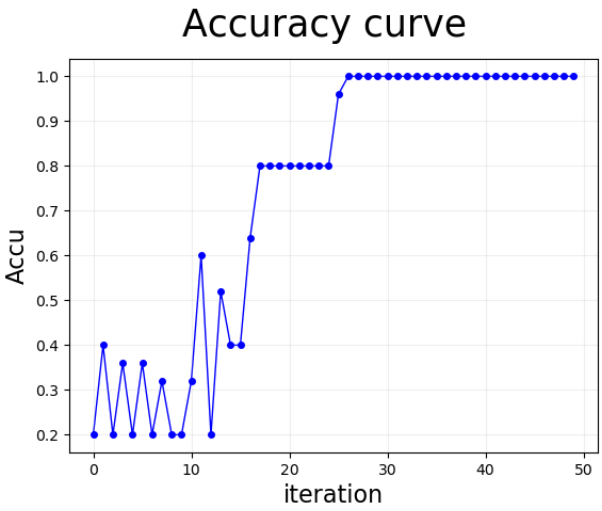
但一開始我怎麼做都畫不出來，先驗機率跟相似度機率乘完後得出的結果卻跟相似度分布一致。後來才發現算完一次分布後，要先進行 normalize 才可以。否則有些看起來分布情形為零的地方卻會有值，進而影響到後續的計算。

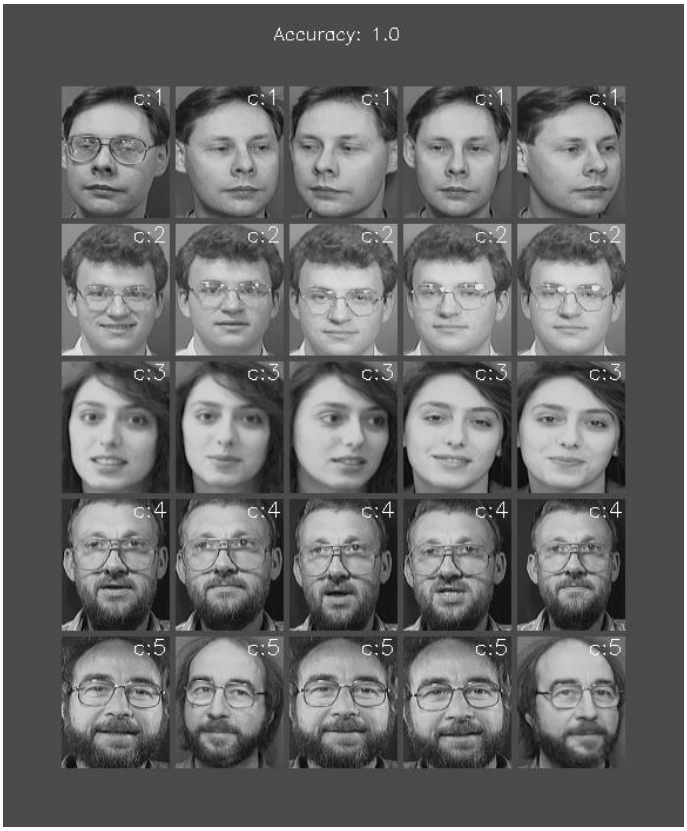
透過這次作業才確確實實的了解到 sequential learning 是透過不斷更新訓練資料，以達到收斂 w 的效果，進而去逼近結果。

2. Logistic Regression

(1)Gradient



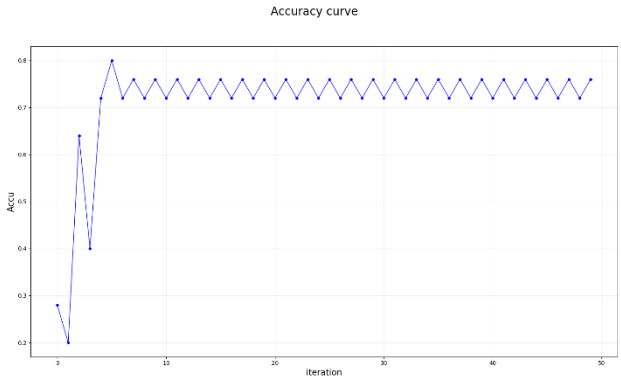


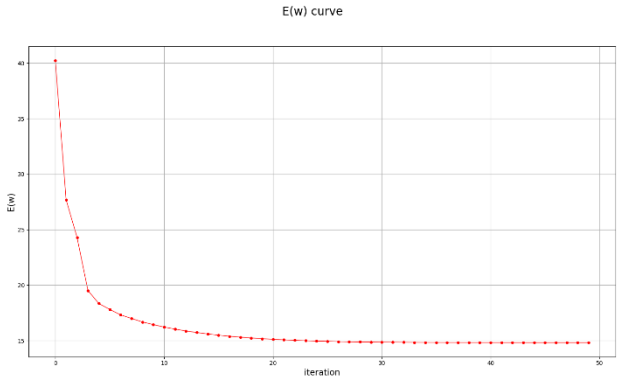


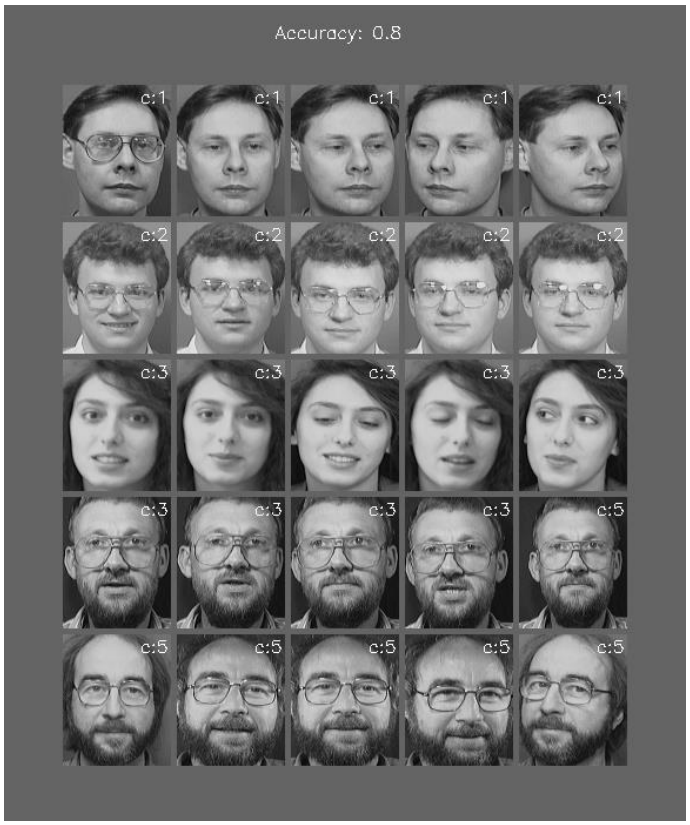
Gradient 使用一階微分來進行運算，所以整體而言要收斂的話會需要歷時較久的時間。雖然最後的做出的分類器效果幾乎可以維持在 0.8~1 之間，但是有時隨機取照片時仍會有準確率降到 0.2，但我目前也尚未確定究竟為何會如此。

(2)Newton

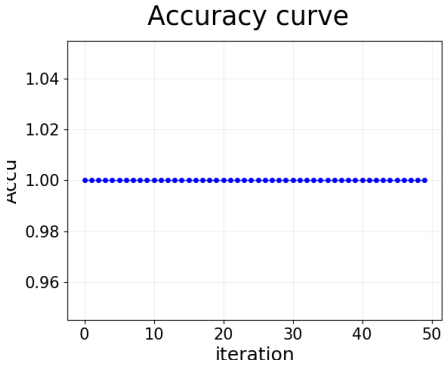
PCA 2

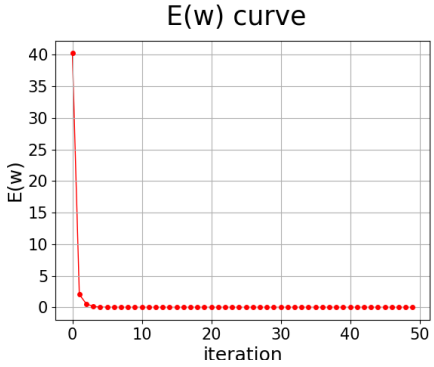


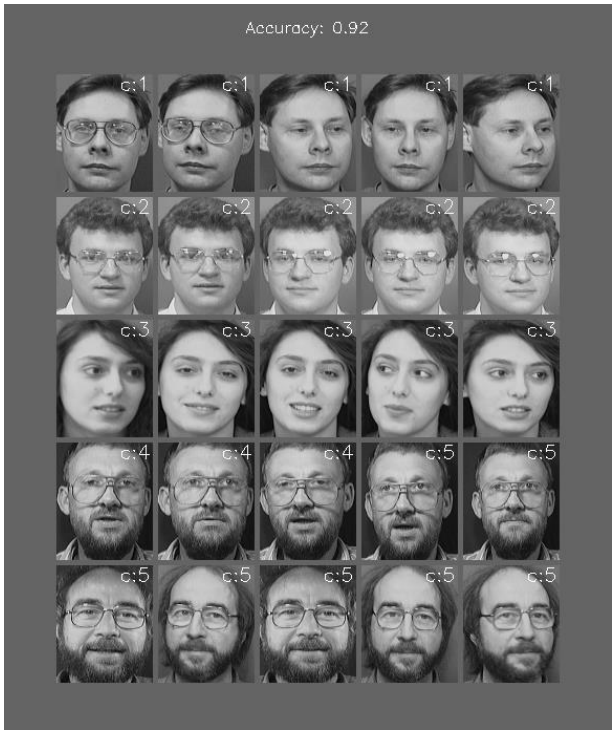




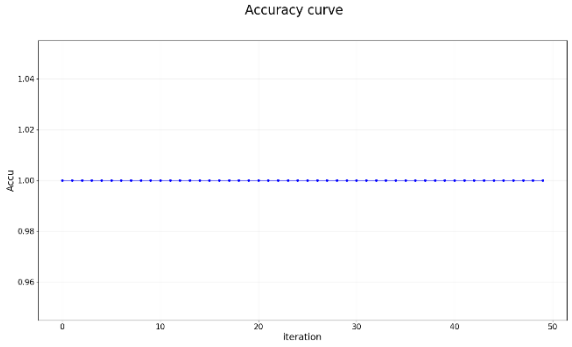
PCA 5

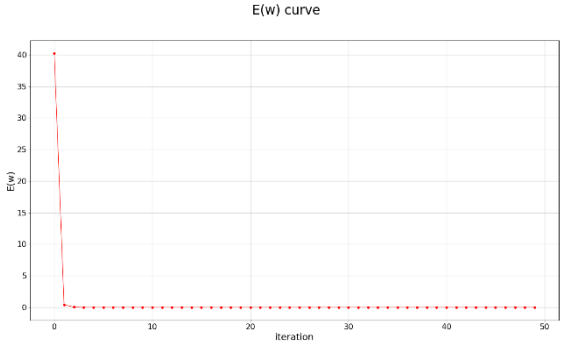

Accuracy curve
Accu
1.04
1.02
1.00
0.98
0.96
0
10
20
30
40
50
iteration

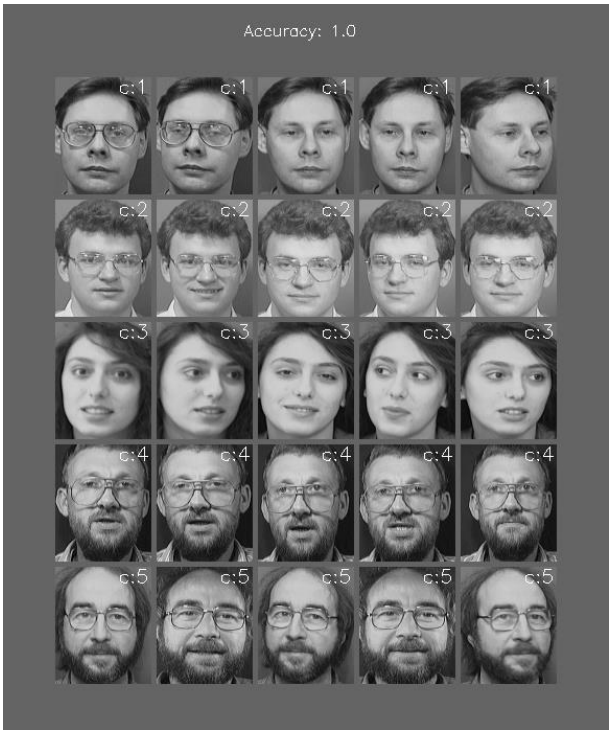

E(w) curve
E(w)
40
35
30
25
20
15
10
5
0
0
10
20
30
40
50
iteration


Accuracy: 0.92
c:1
c:2
c:3
c:4
c:5

PCA 10



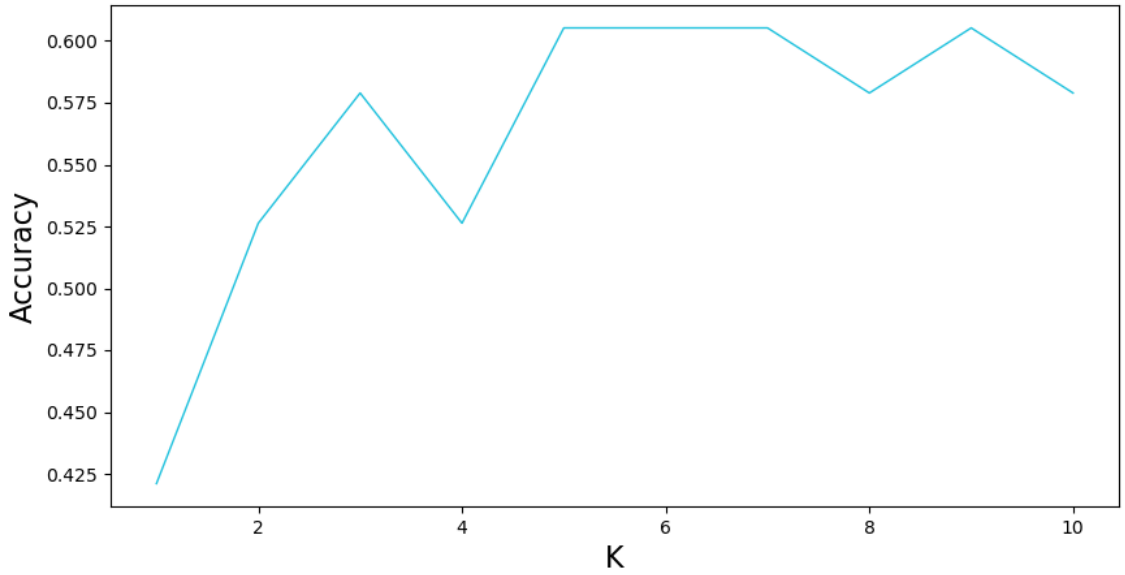




使用 PCA 降維時，會需要使用較大量的記憶體來做運算，所以一開始我用我的電腦跑的時候，因為照片便為 1 維資料，導致最後我要運算矩陣時，電腦容量不足，無法進行運算。幸好改用實驗室電腦後，得以解決這個問題。也因此凸顯出要進行 PCA 降維之前的運算也是需要先付出一筆計算效能。但是經過降維後，降低的資料量讓牛頓法可以更加的好計算，也是 PCA 最大得效益。

從降至 10 維、5 維、2 維的過程中可以發現，當維度降得越低，資料的效能也就更加失真，會使得後續在進行牛頓法做運算的時後效果減低，所以需要在資料量與計算的特徵值上來做取捨。

3. Nonparametric Methods



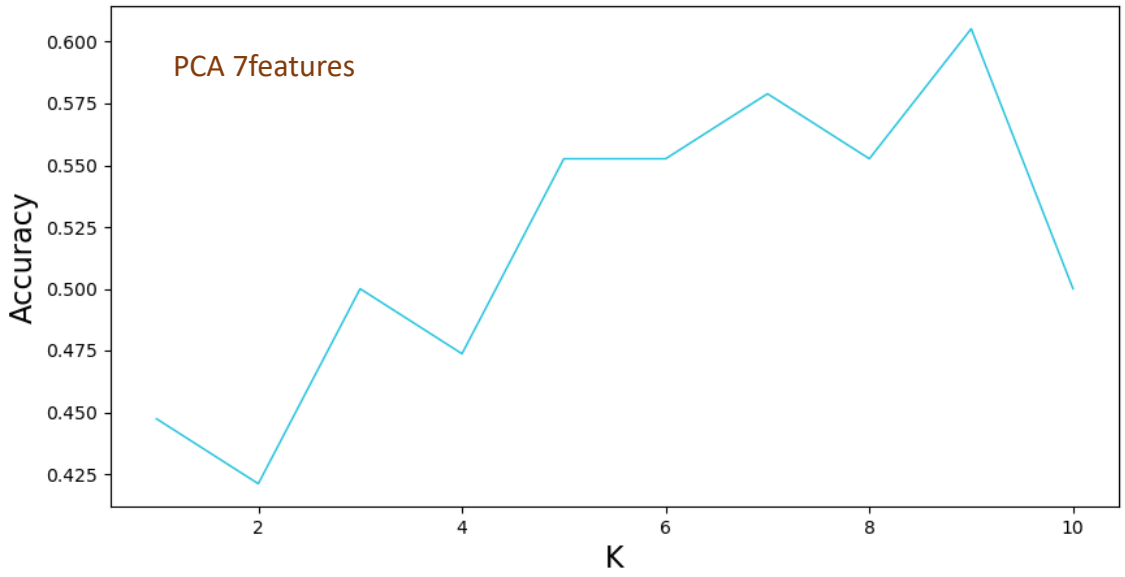
(1)K-nearest-neighbor classifier (with 9 features)

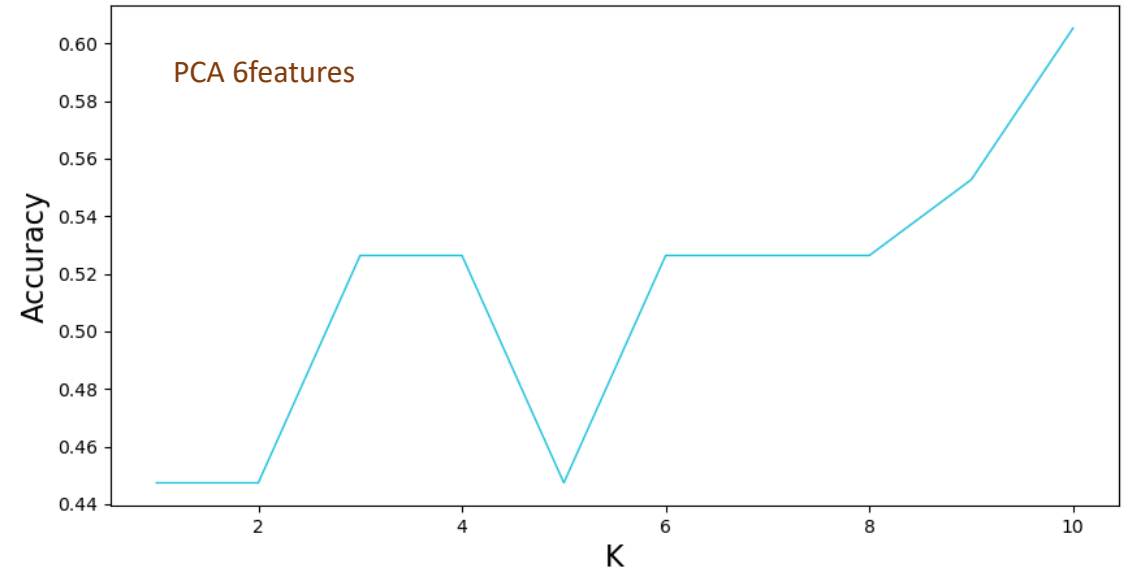
透過 k nearest neighbor 的簡單概念，將一個測試點帶入訓練資料的分布，計算該點與最靠近他的 K 個點做分類。找出這 K 個點大部分所屬的類別來進行該點的預測分類。不過因為原始資料含有一項是 true/false 的判斷，因此我將這一項的數值改為 true 的話就看成 1、false 則為 0，以便後續算距離時能有實質上的意義。接著將所有的數值進行標準化後，便將測試資料丟入計算尤拉距離，找出最靠近該測試點的 K 個點來做測試。

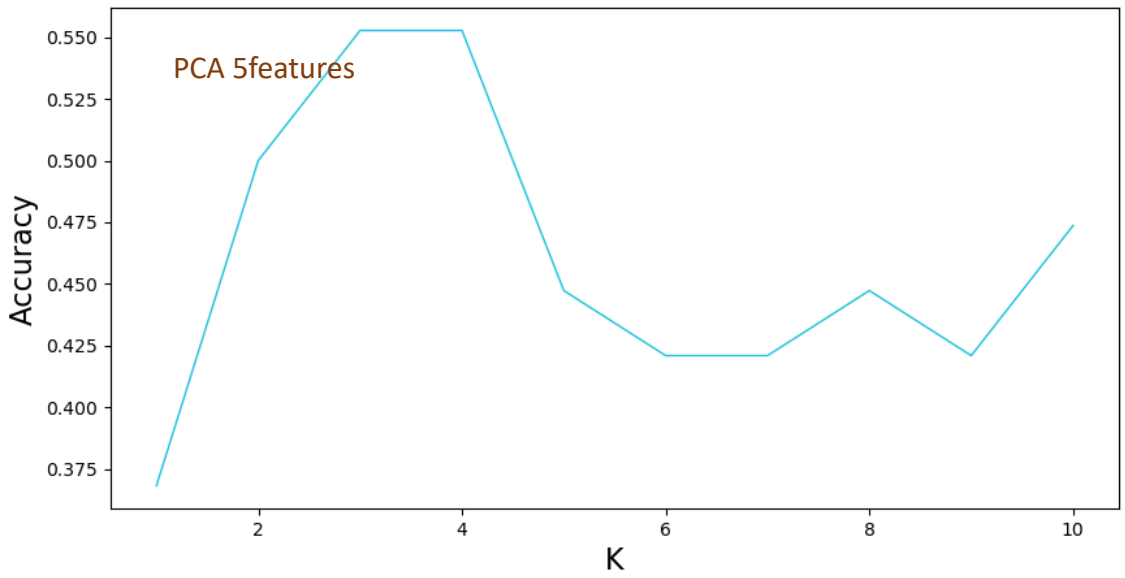
不過一開始在看資料的時候，不禁納悶真的需要把全部的資料都來拿來使用嗎？因為有些是 true/false 的判斷，有些則是一些世代分類的依據，而將這些資料拿來做為我計算距離的一個特徵點，感覺會有一些怪怪的地方。

因此，我除了使用全部 9 項特徵(扣除名字與分類)，也有試過排除世代與是否為傳說級的這兩項特徵。會發現排除這兩項所預測出的結果會更好。

(2)K-nearest-neighbor classifier (With PCA ,K=7)







使用 PCA 降維，但這邊一開始降維不是很懂怎麼樣給數據。一開始是把 train 跟 test 分開丟入降維，但發現了準確度卻也跟這下降非常多。一開始想說降維可能或多或少會使得準確度降低，但得出的準確度也下降太多了吧？後來才想到，若是將訓練資料與測試資料分開做 PCA 可能會有兩筆資料降維取的特徵向量不相同，進而去影響到後續做 k nearest neighbor 的準確性。
所以，我後來就把訓練資料跟測試資料一起做降維，然後在做預測時把訓練資料再獨立出來。這樣子嘗試後會發現準確度提升不少，PCA 後準確度最好的跟沒有使用降維方法的準確度差不多。才確定是之前分開丟資料取的特徵值真的會不一樣。
接著，看到上頁的三張圖片可以發現，雖然降維後可以使得資料量變少，但是降維是有上限的，降更多維度後準確度也會跟著降低。因此在降維與準確度存在著一個取捨關係，要能夠取到好的降維又同時能夠保持準確度是需要再根據不同的訓練模型與資料來做不同的選擇的。

雖然 k nearest neighbor 的演算法相當簡單，但對於分類也是有著不錯的準確度。但是，對於一些特別的特徵會不知道該如何數據化來來做後續計算距離的部分，例如這題 Legendary 就是我比較難去取數據的。不過後續我有再做過測試，發現不論該像我將 true/false 給的值距離拉大或縮小，對於後續測試的影響不會特別大。我猜可能是因為該項在 158 筆數據中幾乎都是 false，標準化後的值影響並不會特別顯著。也是因為我對該項做了測試，才知道標準差後的數值會比較好，不會受到特別極值造成的誤差。