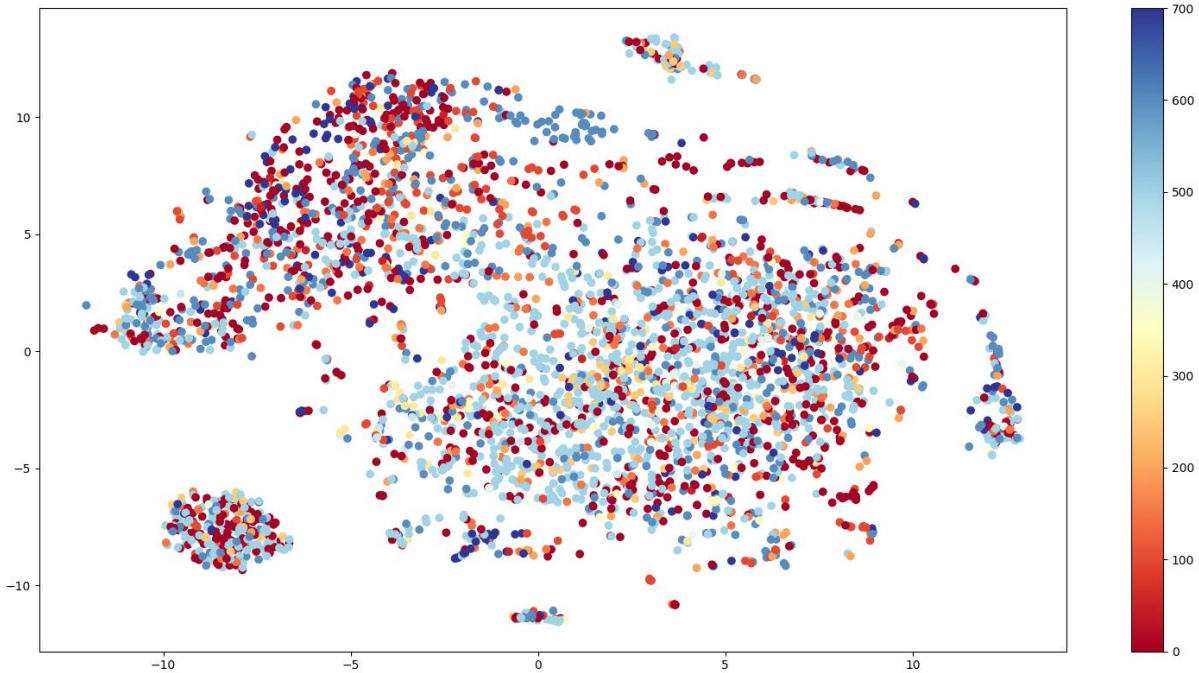
學號：R05921077 系級： 電機碩一 姓名：陳立杰

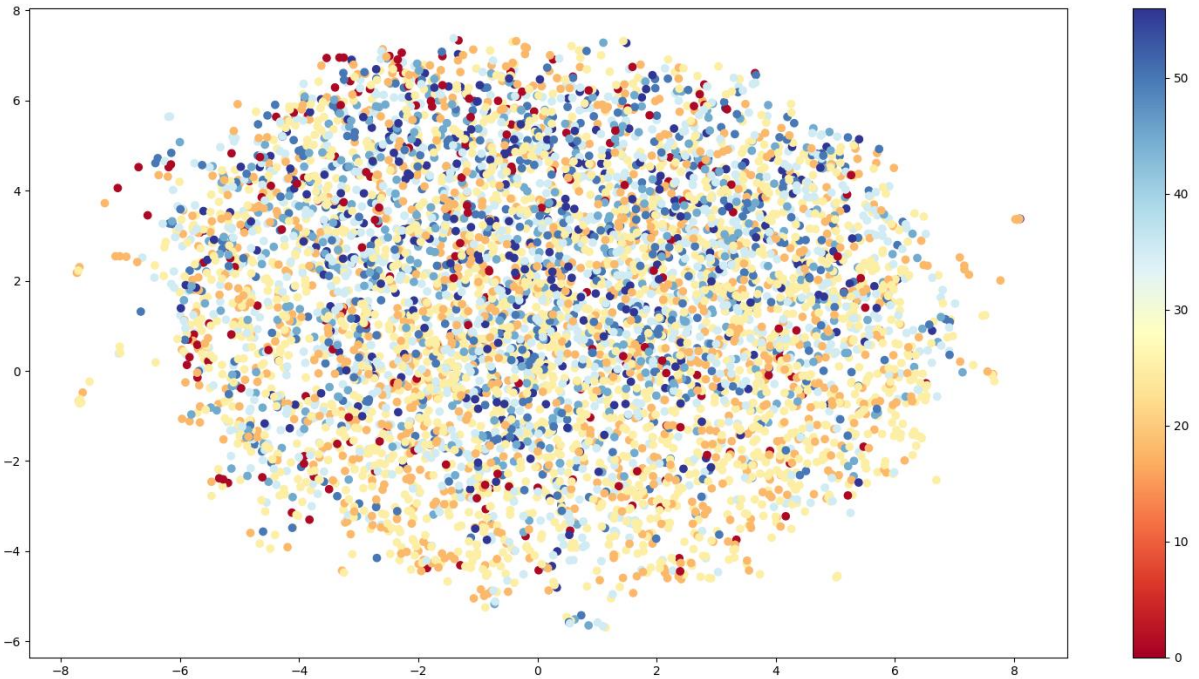
1. (1%)請比較有無 normalize(rating)的差別。並說明如何 normalize.
   我 是先做 rate[i]=(rate[i]-mean)/std 後拿來做 training，然後 predict 出一個值後再用 rate[i]=rate[i]*std+mean 來還原他的值，我發現 normalize 後 training 的速度會變快，原本大概要 train 2X 個 epoch，normalize 後只需 1X 個 epoch，但是準確度會稍微下降， 原本是 0.85716，normalize 後為 0.86223。
2. (1%)比較不同的 latent dimension 的結果。
   假設 k 為 latent dimension，k=2, 4, 8, 110, 120, 130 時在 public set 上的分數分別為 0.89915, 0.86864, 0.86361, 0.85819, 0.85716, 0.85729，從這些 data 可以看到比較大的 k 值的準確率會比較高，但是太大的時候準確度還是會稍微下降。
3. (1%)比較有無 bias 的結果。
   當 latent dimension= 120 時，沒有加 bias 的準確度為 0.85716，加上一些 bias 後準確度可以提升到 0.85414，也發現加入 age 這個 bias 之後的進步最明顯。
4. (1%)請試著用 DNN 來解決這個問題，並且說明實做的方法(方法不限)。並比較 MF 和 NN 的結果，討論結果的差異。
   我將 user embedding 以及 movie embedding concatenate 在一起再過 DNN 得出 rating，DNN 中有兩個 hidden layer，每個 layer 有 128 個 neural 且 dropout 0.1，output regression，當 k=8 時準確率為 NN 的 0.86285，而 MF 的準確度為 0.86361，稍微比 MF 好一些，但 NN 相較之下需要 train 非常久，大概需要 train 到 100 個 epoch，而 MF 大概只要 2X 個 epoch 就能 train 完了，所以如果 k 太大時 NN 就會非常難 train(k=latent dimension) ，應該是因為 NN 的參數量較多而且有 drop 才導致必須 train 那麼久。
5. (1%)請試著將 movie 的 embedding 用 tsne 降維後，將 movie category 當作 label 來作圖。
   Comedy 0
   Fantasy 50
   Action 100
   Romance 150
   Sci-Fi 200
   Documentary 250
   War 300
   Mystery 350
   Western 400
   Film-Noir 450
   Drama’,’Musical 500
   Thriller',’Horror’,’Crime 600
   Adventure,Animation,Children's 700



從這圖中可看到中間有很大的兩堆，上面那堆比較多喜劇、動作、恐怖片等，下面那堆則是比較多 drama 所以感覺很雜亂，因為 drama 很多不同的種類，而最上下分別有兩個很獨立的小堆，兩個小堆都有很多的 drama，下面那堆感覺比較偏喜劇、愛情的 drama，上面那堆則是比較偏神秘或戰爭類的 drama。

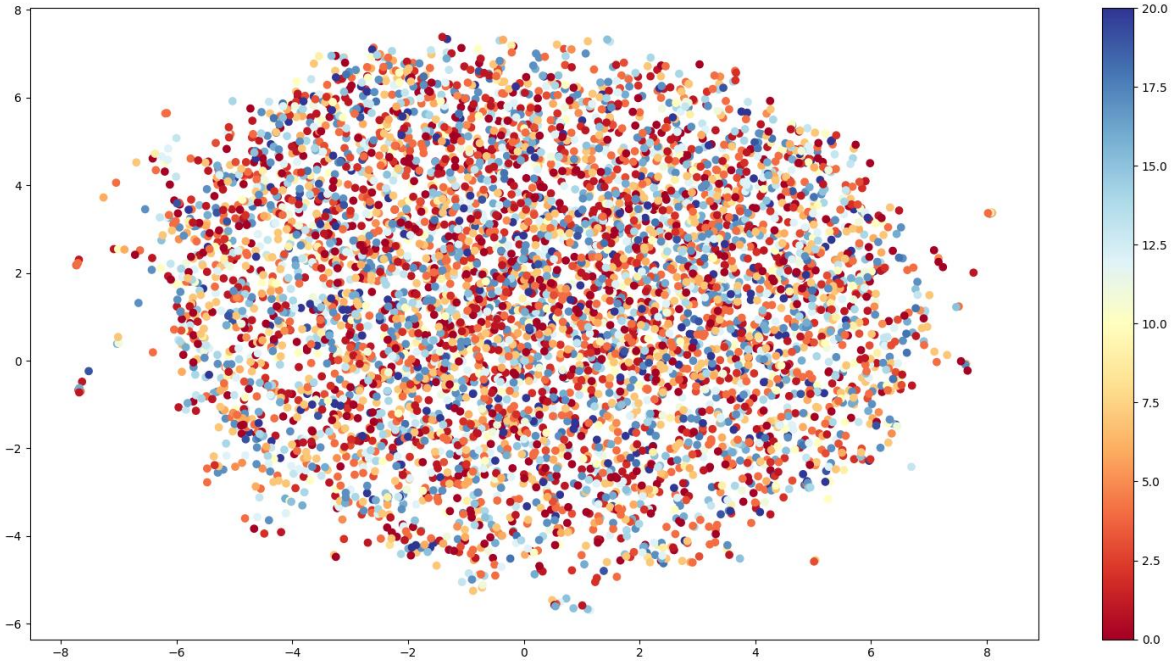
6. (BONUS)(1%)試著使用除了 rating 以外的 feature, 並說明你的作法和結果，結果好壞不會影響評分。

我把 users 的 age 跟 occupation 拿來當作額外的 feature 跟 tags，拿 users 的 embedding 用 tsne 降維作圖，下圖是 age 的結果:



很明顯的可以看到年紀較大也就是顏色較深的部分幾乎都集中在上半部，而年紀較小也就是較淺的點全部集中在下半部，代表年齡的確是有一些區分的效果，在做 bias 時確實也是加入 age 影響最大。

下圖是用 occupation 作圖的結果:



因為種類太多又不好分群，所以做出來的結果看不出什麼意義。