**User-Based CF:**

Dataset裡面有userID、itemID、rate，第一主要工作是取出user之間對於item共同的評價，根據這些來去計算出各user與其他user的相似度，而這裡我用Cosine Similarity meature (Fig. 1)的方式來去計算相似度，根據相似度去找出與user最為相像的人，並推測user在觀看此電影時，會做出與相似者一樣的評價，為了以防有些相似者因為共同電影少，所造成的相似度高的假象，因此採用DiscountSim(Fig 2.)方法，Threshold設5。

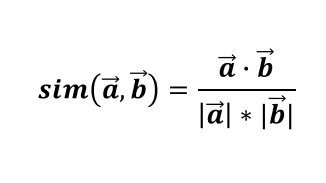


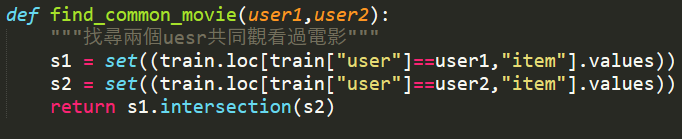
Fig 2.

Fig 1.

**實做說明:**

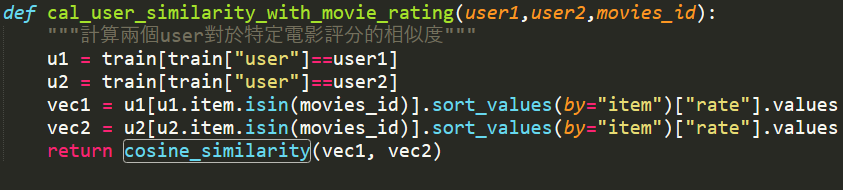
**Find\_common\_movie(user1,user2) :**

**此function主要取出兩個user共同觀看過的電影，透過個別取出user1和user2的itemID對其做交集**



**cal\_user\_similarity\_with\_movie\_rating(user1,user2,moviesid):**

**此function主要是取出user1跟user2共同觀看的電影評價，並將結果傳至cosine\_similarity**

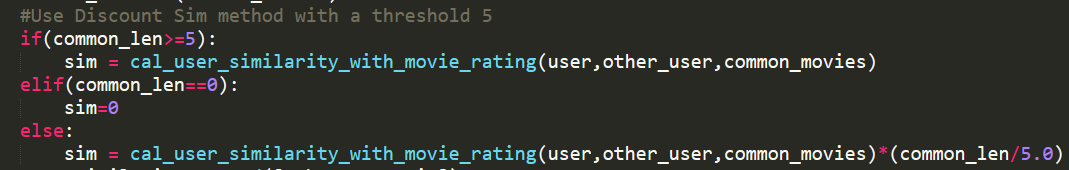


**cosine\_similarity(vec1, vec2):**

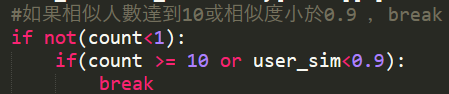
**此function主要是實現Cosine Similarity meature**



這裡主要是運用Discount Sim，且門檻為5

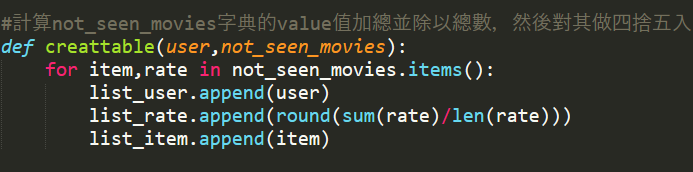


若取出相似者人數大於10或者相似者的相似度小於0.9，則跳出迴圈，避免一些相似度較低者加入分析。



**creattable(user,not\_seen\_movies):**

**此function主要是針對user1未評價的電影來去取出相似者set裡的評價做平均，並做四捨五入。**



然後我們做了一個評估程式，此模組使用MSE:

A\_data:0.0770046669495121

B\_data: 0.10930701475114082