Jinze Wan

101168891

User based:

200 neighboours, sim > 0: 0.6654059454293244

150 neighboours, sim > 0: 0.6654059454293244

100 neighboours, sim > 0: 0.6654059454293244

75 neighboours, sim > 0: 0.6654059454293244

70 neighboours, sim > 0.1: 0.6663258091299964

**70 neighboours, sim > 0: 0.6654059454293244**

70 neighboours, sim > -0.05: 0.67429761083426

70 neighboours, sim > -0.1: 0.6713011812166466

70 neighboours, sim > -0.15: 0.6872790658968316

70 neighboours, sim > -0.2: 0.7040701935623871

65 neighboours, sim > 0: 0.6654061569485494

60 neigihbours, sim > 0: 0.6654032302398318

50 neighboours, sim > 0: 0.6653794422636724

5 neighbours, sim > 0: 0.6881972556181207

2 neighbours, sim > 0: 0.761858768827512

Item based:

500 neighbours, sim > 0: 0.6639390844591836

400 neighbours, sim > 0: 0.6639385398069291

300 neighbours, sim > 0.2: 0.6648929016517194

300 neighbours, sim > 0.1: 0.6638940149646412

300 neighbours, sim > 0: 0.6639063038585276

**300 neighbours, sim > -0.1: 0.663803446604611**

300 neighbours, sim > -0.2: 0.669396621690839

250 neighbours, sim > 0: 0.6639432576561284

250 neighbours, sim > -0.1: 0.6638770972257245

200 neighbours, sim > 0: 0.6639798874133416

200 neighbours, sim > -0.05: 0.6639632287208611

200 neighbours, sim > -0.1: 0.6639522285220869

200 neighbours, sim > -0.2: 0.6697849157950609

150 neighbours, sim > 0: 0.6647590009591346

150 neighbours, sim > -0.1: 0.6647460435191779

100 neighbours, sim > 0: 0.667240043429692

75 neighbours, sim > 0: 0.6695156655279239

50 neighbours, sim > 0: 0.6739595436316902

20 neighbours, sim > 0: 0.691399992065137

10 neighbours, sim > 0: 0.713389704176813

5 neighbours, sim > 0.5: 0.7507624252208873

5 neighbours, sim > 0.2: 0.7482523550545657

5 neighbours, sim > 0.1: 0.7480487793598285

5 neighbours, sim > 0: 0.7479853874676544

5 neighbours, sim > -0.05: 0.747987349888184

5 neighbours, sim > -0.1: 0.7480066685217203

5 neighbours, sim > -0.2: 0.7480837976009204

2 neighbours, sim > 0: 0.8237192901907274

1. I implement an Item based recommendation system and an User based recommendation system. For Item based algorithm, I need to calculate the average rating of users, and then combine the user rating and the average rating to calculate the similarity between different items and predict the rating. For User based algorithm, I need to calculate the average rating of users, and then calculate the similarity between two different users and the predicted value of the target rating is finally obtained.
2. I reduced the number of loops as much as possible, and only calculated the average rating of the user related to the prediction and the similarity of items related to the prediction when predicting each rating
3. Make a 2d array for traversing the original data. If the score is not 0, it will be regarded as 0 and its rating will be predicted. Then the predicted rating will be compared with the actual rating. In this process, both the average rating and similiarty are calculated with the target rating as 0

When comparing the most accurately case, item based is more accurate than user based, but the difference is small (i.e. Only ~0.002 difference)



User based:

Top 10: 1.1756793268666776

Top 5: 0.7441896847595946

Top 2: 0.7707038240810274

Threshold based 0.2: 0.6749357011904494

Threshold based 0.1: 0.6663258091299964

Threshold based 0: 0.6654059454293244

Threshold based -0.05: 0.6742977138482814

Threshold based -0.1: 0.6713021633977755

Threshold based -0.15: 0.687282429592566

Item based:

Top 10: 1.0999421823534348

Top 5: 0.7482817213375311

Top 2: 0.8236830994801414

Threshold based 0.5: 0.675314677657937

Threshold based 0.2: 0.6649266814611896

Threshold based 0.1: 0.6639208156791818

Threshold based 0: 0.6639390844591836

Threshold based -0.05: 0.6639045138144726

Threshold based -0.1: 0.6638292070431066

Threshold based -0.2: 0.6692582505774116

Threshold based -0.3: 0.6821026774707774

Threshold based -0.4: 0.7638718410133911

Threshold based -0.5: 1.1154575248898422

In general, threshold-based is more accurate for this data. For user based algorithm, the most accurate case is when threshold set to about 0. For item based algorithm, the most accurate case is when Threshold is set to about -0.1.



The prediction rating is most accurate when the neighbour limit is set to about 300 and the threshold is set to about -0.1. With the increase of neighbour limit, the accuracy will get higher and higher until the maximum accuracy is about 300. After that, increasing the neighbour limit will decrease the accuracy. When the simlarity threshold is set at about -0.1, the accuracy will decrease when the threshold is raised from -0.1 to 0 or lowered from -0.1. When the threshold is raised from 0 to about 0.1, the accuracy will be improved. If the threshold is raised further, the accuracy decreases.

For user based case, no.

For item based case, yes, If a negative similarity of as little as -0.1 is included, the accuracy of the prediction can be improved.



User based:

150 neighbours, sim > 0: 39798ms

150 neighbours, sim > 0: 39559ms

100 neighbours, sim > 0: 39101ms

75 neighbours, sim > 0: 38561ms

70 neighbours, sim > 0.1: 38458ms

70 neighbours, sim > 0: 40045ms

70 neighboours, sim > -0.05: 39526ms

70 neighboours, sim > -0.1: 39372ms

70 neighboours, sim > -0.15: 39651ms

70 neighboours, sim > -0.2: 41728ms

65 neighbours, sim > 0: 38763ms

60 neighbours, sim > 0: 39207ms

50 neighbours, sim > 0: 39254ms

5 neighbours, sim > 0: 41794ms

2 neighbours, sim > 0: 40360ms

Item based:

150 neighbours, sim > 0.2: 47171ms

150 neighbours, sim > 0.1: 45718ms

150 neighbours, sim > 0: 46541ms

150 neighbours, sim > -0.1: 45851ms

150 neighbours, sim > -0.2: 47172ms

150 neighbours, sim > -0.5: 45429ms

100 neighbours, sim > 0: 45267ms

50 neighbours, sim > 0: 46084ms

20 neighbours, sim > 0: 45466ms

10 neighbours, sim > 0: 48869ms

5 neighbours, sim > 0: 47840ms

User based algorithms take less time to run than item based algorithms, and the number of neighbours and the threhold of similarity only have very limit influence on the running time.

I will choose user based algorithm. First, the user based algorithm takes less time to compute, which is important when you consider the large amount of data and the user experience. Second, in the case of fewer users and more items (films), considering the changes of film evaluation, user-based updating of neighbour requires less calculation than item-based updating. For example, in the sample data, there are only 165 users whose neighbor similarity needs to be calculated, and 4423 items need to be calculated if the item based algorithm is adopted.

In the case of the user based algorithm, when it comes to parameters, I will choose 70 neighbors and similarity greater than 0.



With more reviews, the accuracy of the algorithm will increase as the number of available neighbors increases. With less reviews, the algorithm will run into problems with Sparsity and Cold-start.