This project is about a news article recommendation system where users are provided a comfort while rating a news article. For example, while rating a movie, one may like its acting but not direction, cast but not songs, cinematography but not the script. In that case, it is not easy for a user to directly rate a movie or a news article for that matter. He rates several criteria, and then, the ratings of those criteria are aggregated using fuzzy technique to arrive at the final rating.

We have table 1 which contains user ratings in linguistic variables as follows.

Table 1. Linguistic Ratings

|  |  |  |
| --- | --- | --- |
| **User** | **NewsArticle1** | **NewsArticle2** |
|  | (Criteria1,Criteria2, Criteria3, Criteria4, Criteria5) | (Criteria1,Criteria2, Criteria3, Criteria4, Criteria5) |
| User1 | Good, Bad, Good, Average, Average, Bad, Good | Good, Average, Average, Average, Bad, Bad, Good |
| User2 | Bad, Bad, Average, Average, Good, Good, Bad | Average, Average, Average, Good, Good, Bad, Bad |
| User3 | Average, Average, Good, Good, Average, Average, Average | Good, Good, Average, Bad, Average, Average, Average |

Now in table 2, we fuzzify the ratings of each criterion for each news article.

Table 2. Fuzzified ratings

|  |  |  |
| --- | --- | --- |
| **User** | **NewsArticle1** | **NewsArticle2** |
|  | (Criteria1,Criteria2, Criteria3, Criteria4, Criteria5) | (Criteria1,Criteria2, Criteria3, Criteria4, Criteria5) |
| User1 | (0.2,0.3,0.4),(0.0,0.1,0.2),(0.2,0.3,0.4),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.0,0.1,0.2),(0.2,0.3,0.4) | (0.2,0.3,0.4),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.0,0.1,0.2),(0.0,0.1,0.2),(0.2,0.3,0.4) |
| User2 | (0.0,0.1,0.2),(0.0,0.1,0.2),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.2,0.3,0.4),(0.2,0.3,0.4),(0.0,0.1,0.2) | (0.1,0.2,0.3),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.2,0.3,0.4), (0.2,0.3,0.4), (0.0,0.1,0.2), (0.1,0.2,0.3) |
| User3 | (0.1,0.2,0.3),(0.1,0.2,0.3),(0.2,0.3,0.4),(0.2,0.3,0.4),(0.1,0.2,0.3),(0.1,0.2,0.3),(0.1,0.2,0.3) | (0.2,0.3,0.4),(0.2,0.3,0.4),(0.1,0.2,0.3),(0.0,0.1,0.2), (0.1,0.2,0.3), (0.1,0.2,0.3), (0.1,0.2,0.3) |

Here, bad = (0.0,0.1,0.2), average = (0.1,0.2,0.3), and good = (0.2,0.3,0.4).

Next step is to solve them. First, we take the average of membership values for each criterion. Suppose criteria 1 belongs to news article 1, and is rated as (a,b,c) by user 1 and (d,e,f) by user 2, then, criteria 1's final fuzzified ratings for news article 1 would be ((a+d)/2,(b+e)/2,(c+f)/2) if there are 2 users. Note that this rating for criteria 1 or any other criteria for that matter of news article 1 (or any other for that matter) would be same for all the users.

Therefore, we obtain average ratings in the following table.

Table 3. Averaged Fuzzified Ratings

|  |  |  |
| --- | --- | --- |
| **Criteria** | **News 1** | **News 2** |
| Criteria1 | (0.1,0.2,0.3) | (0.2,0.3,0.4) |
| Criteria2 | (0.0,0.1,0.2) | (0.1,0.2,0.3) |
| Criteria3 | (0.1,0.2,0.3) | (0.1,0.2,0.3) |
| Criteria4 | (0.1,0.2,0.3) | (0.1,0.2,0.3) |
| Criteria5 | (0.1,0.2,0.3) | (0.1,0.2,0.3) |
| Criteria6 | (0.1,0.2,0.3) | (0.0,0.1,0.2) |
| Criteria7 | (0.1,0.2,0.3) | (0.1,0.2,0.3) |

Next job is to defuzzify news article 1 and 2. The formula for triangular defuzzification is (a+2b+c)/4 which means criteria 1 for news article 1 can be defuzzified as (0.1+2(0.2)+(0.3))/4. So, we get the following table.

Table 4. Defuzzified Ratings

|  |  |  |
| --- | --- | --- |
| **Criteria** | **News 1** | **News 2** |
| Criteria1 | 0.2 | 0.3 |
| Criteria2 | 0.1 | 0.2 |
| Criteria3 | 0.2 | 0.2 |
| Criteria4 | 0.2 | 0.2 |
| Criteria5 | 0.2 | 0.2 |
| Criteria6 | 0.2 | 0.1 |
| Criteria7 | 0.2 | 0.2 |

We can find the final rating of news article 1 by multiplying the matrix with triangular fuzzy normalized weight 0.3.

[0.2 0.1 0.2 0.2 0.2 0.2 0.2] X [0.3]=0.3(0.2+0.1+0.2+0.2+0.2+0.2+0.2)=0.39

Table 5. Final Ratings

|  |  |  |
| --- | --- | --- |
| **Final Rating** | **News 1** | **News 2** |
|  | 0.39 | 0.42 |

/\*Query to insert data of table fuzzifiedratings to table defuzzifiedaverageratings of database for criteria 1

insert into defuzzifiedaverageratings(newsarticle1, newsarticle2, newsarticle3, newsarticle4, newsarticle5) (select avg(newsarticle1a)+2\*avg(newsarticle1b)+avg(newsarticle1c),avg(newsarticle2a)+2\*avg(newsarticle2b)+avg(newsarticle2c),avg(newsarticle3a)+2\*avg(newsarticle3b)+avg(newsarticle3c),avg(newsarticle4a)+2\*avg(newsarticle4b)+avg(newsarticle4c),avg(newsarticle5a)+2\*avg(newsarticle5b)+avg(newsarticle5c) from fuzzifiedratings where criteria=1);

/\* Query to insert data of table fuzzifiedratings to table defuzzifiedaverageratings of database for criteria 2

insert into defuzzifiedaverageratings(newsarticle1, newsarticle2, newsarticle3, newsarticle4, newsarticle5) (select avg(newsarticle1a)+2\*avg(newsarticle1b)+avg(newsarticle1c),avg(newsarticle2a)+2\*avg(newsarticle2b)+avg(newsarticle2c),avg(newsarticle3a)+2\*avg(newsarticle3b)+avg(newsarticle3c),avg(newsarticle4a)+2\*avg(newsarticle4b)+avg(newsarticle4c),avg(newsarticle5a)+2\*avg(newsarticle5b)+avg(newsarticle5c) from fuzzifiedratings where criteria=2);

/\* Query to insert data of table fuzzifiedratings to table defuzzifiedaverageratings of database for criteria 3

insert into defuzzifiedaverageratings(newsarticle1, newsarticle2, newsarticle3, newsarticle4, newsarticle5) (select avg(newsarticle1a)+2\*avg(newsarticle1b)+avg(newsarticle1c),avg(newsarticle2a)+2\*avg(newsarticle2b)+avg(newsarticle2c),avg(newsarticle3a)+2\*avg(newsarticle3b)+avg(newsarticle3c),avg(newsarticle4a)+2\*avg(newsarticle4b)+avg(newsarticle4c),avg(newsarticle5a)+2\*avg(newsarticle5b)+avg(newsarticle5c) from fuzzifiedratings where criteria=3);

/\* Query to insert data of table fuzzifiedratings to table defuzzifiedaverageratings of database for criteria 4

insert into defuzzifiedaverageratings(newsarticle1, newsarticle2, newsarticle3, newsarticle4, newsarticle5) (select avg(newsarticle1a)+2\*avg(newsarticle1b)+avg(newsarticle1c),avg(newsarticle2a)+2\*avg(newsarticle2b)+avg(newsarticle2c),avg(newsarticle3a)+2\*avg(newsarticle3b)+avg(newsarticle3c),avg(newsarticle4a)+2\*avg(newsarticle4b)+avg(newsarticle4c),avg(newsarticle5a)+2\*avg(newsarticle5b)+avg(newsarticle5c) from fuzzifiedratings where criteria=4);

/\* Query to insert data of table fuzzifiedratings to table defuzzifiedaverageratings of database for criteria 5

insert into defuzzifiedaverageratings(newsarticle1, newsarticle2, newsarticle3, newsarticle4, newsarticle5) (select avg(newsarticle1a)+2\*avg(newsarticle1b)+avg(newsarticle1c),avg(newsarticle2a)+2\*avg(newsarticle2b)+avg(newsarticle2c),avg(newsarticle3a)+2\*avg(newsarticle3b)+avg(newsarticle3c),avg(newsarticle4a)+2\*avg(newsarticle4b)+avg(newsarticle4c),avg(newsarticle5a)+2\*avg(newsarticle5b)+avg(newsarticle5c) from fuzzifiedratings where criteria=5);

In table defuzzifiedaverageratings of database, we should already have the values of criteria indicating their serial numbers before final ratings of news articles are inserted. For that, the following query suffices.

Insert into defuzzifiedaverageratings (criteria) values(1),(2),(3),(4),(5);

Table finalratings of the database contains final ratings. For that following query is executed.

insert into finalratings(newsarticle1,newsarticle2,newsarticle3,newsarticle4,newsarticle5) (select 0.3\*sum(newsarticle1),0.3\*sum(newsarticle2),0.3\*sum(newsarticle3),0.3\*sum(newsarticle4),0.3\*sum(newsarticle5) from defuzzifiedaverageratings);