

## CSC 466 Lab 5 Report

### **Authors:**

Sophia Chung, [spchung@calpoly.edu](mailto:spchung@calpoly.edu)

Aditi Gajjar, [agajjar@calpoly.edu](mailto:agajjar@calpoly.edu)

Anagha Sikha, [arsikha@calpoly.edu](mailto:arsikha@calpoly.edu)

## **Introduction**

In this report, we evaluate four unique collaborative filtering methods applied to the Jester dataset from the Jester project at UC Berkeley. The methods utilize different combinations of similarity measures and predictors. Our main objective is to find out which method best predicts user ratings for jokes which we answer through specific research questions comparing the methods' effectiveness.

## **Collaborative Filtering Methods**

We used N Nearest Neighbors as our approach for all four methods. This approach looks at a given user and their joke ratings, and then identifies the N nearest neighbors based on the chosen similarity metric (Pearson correlation or cosine similarity). We wanted to logically focus on users with similar joke preferences or rating styles.

We also incorporated a normalization process in order to scale values of predicted ratings to between -10 and 10, which guarantees the predicted ratings are on a comparable scale with the actual ratings. By normalizing the predictions in this manner, we ensured that the evaluation of the following methods accurately reflects their prediction accuracy.

### **Method 1**

Our first method uses Pearson correlation for the similarity measure between users and utilizes weighted sums as the predictor. The Pearson correlation reflects the statistical correlation between the users' rating patterns and focuses on measuring the strength and direction of this linear relationship. The weighted sum takes into account the influence of neighboring users based on their similarity to the chosen user. However, weighted sums are insensitive to the variability in users' ratings (users having different baseline preferences) and are very sensitive to outliers.

### **Method 2**

Our second method uses cosine similarity for the similarity measure between users and uses weighted sums for rating predictions. Cosine similarity measures the collinearity of the users' rating patterns. This metric focuses on the direction rather than the magnitude of the vector and is useful in looking at user preferences in an angular way such as the direction of user rating vectors. However, cosine similarity does not account for the jokes that certain users did not rate.

### **Method 3**

Similar to method 1, method 3 utilizes Pearson correlation for the similarity measure between users but uses adjusted weighted sums for rating predictions. Unlike weighted sums, adjustment weighted sums account for the user's approach to rating the jokes by

taking the average ratings of users, and thus, considering potential biases or trends in the user ratings.

## **Method 4**

Similar to method 2, method 4 utilizes cosine similarity for the similarity measure, and similar to method 3, method 4 uses adjusted weighted sums as the predictor.

## **Research Questions**

Question 1: For each method, which values of size and repeats maximize the F1 score?

- The F1 score is useful for identifying both positive and negative cases of recommendation and strikes a balance between precision and recall. This metric is also a well-established measure; thus, we want to compare this metric among the different methods in order to determine which method is the best.

Question 2: After tuning the values for size and repeats, which method results in the highest F1 score?

- We aim to see how different predictors (weighted sum vs adjusted weighted sum) and the different similarity measures (Pearson correlation vs cosine similarity) impact the metrics.

## **Experiments**

We used random sampling, repetition, and normalization to minimize randomness, ensuring that our findings are reliable and reflective of the methods' true capabilities. To begin with, we systematically evaluated different collaborative filtering methods by generating random test cases for each method. For each evaluation, we specify the size of the test set and the number of repetitions. This random sampling, coupled with multiple iterations, significantly reduces the risk of bias by ensuring a diverse range of user-joke pairs. Also, we normalize the predicted ratings to a consistent -10 to +10 scale, enhancing comparability across different methods.

To evaluate the performance of our collaborative filtering methods, we implemented two evaluation methods, a random sampling method that generated random test cases (CFRandom), and a user-specified test method that took in an inputted list of test cases (CFList). These evaluation methods computed the mean absolute error (MAE), confusion matrix, precision, recall, F1 score, and overall accuracy.

CFList would yield the same results as CFRandom given the same input. We decided to run our experiments using CFRandom due to the method's ability to provide a representative and unbiased sample of user-joke interactions with repetition.

For all the methods, we followed this process to find the best metrics: increasing repeats, finding the best repeat, then increasing size, and finding the best size. From all the results, we selected which experiment had ideal metrics or a combination of low MAE mean, low MAE standard deviation, high accuracy, and high F1 score.

After running different sizes of  $N$ , we decided to hardcode  $N$  to be 5, which provides a good balance between the quality of recommendations and computational efficiency. Looking at the 5 nearest neighbors considers a sufficient number of neighbors for robust recommendations while keeping the computation time reasonable, and looks at a diverse enough set of neighbors.

## Method 1: Pearson Correlation x Weighted Sum

| Size | Repeats | Metrics  |
|------|---------|--|
| 5    | 3       | MAE mean: 7.14237341737723<br>MAE std: 0.8524075436379666<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 9 1<br>True 3 2<br>Precision: 0.6666666666666666<br>Recall: 0.4<br>F1 score: 0.5<br>Overall accuracy:<br>0.7333333333333333                        |
| 5    | 10      | MAE mean: 7.048281777067521<br>MAE std: 2.2534591772016923<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 30 12<br>True 7 1<br>Precision: 0.07692307692307693<br>Recall: 0.125<br>F1 score: 0.09523809523809525<br>Overall accuracy: 0.62                   |
| 10   | 3       | MAE mean: 9.481975559829142<br>MAE std: 1.3856248691957664<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 17 6<br>True 5 2<br>Precision: 0.25<br>Recall: 0.2857142857142857<br>F1 score: 0.2666666666666666<br>Overall accuracy:<br>0.6333333333333333      |
| 25   | 3       | MAE mean:<br>5.947037765117297<br>MAE std: 0.7266329469040322<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 47 7<br>True 18 3<br>Precision: 0.3<br>Recall: 0.14285714285714285<br>F1 score: 0.19354838709677416<br>Overall accuracy:<br>0.6666666666666666 |

Table 1: Tuning Results from Pearson Correlation x Weighted Sum Method

## Method 2: Cosine Similarity x Weighted Sum

| Size | Repeats | Metrics  |
|------|---------|--|
| 5    | 3       | MAE Mean: 10.484431974711184<br>MAE Std: 0.337906789984688<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 4 3<br>True 7 1<br>Precision: 0.25<br>Recall: 0.125<br>F1 score: 0.16666666666666666<br>Overall accuracy: 0.3333333333333333                  |
| 5    | 10      | MAE Mean: 9.370092022918367<br>MAE Std: 1.5645866026730517<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 20 14<br>True 13 3<br>Precision: 0.17647058823529413<br>Recall: 0.1875<br>F1 score: 0.1818181818181818<br>Overall accuracy: 0.46              |
| 10   | 3       | MAE Mean: 8.335580799378405<br>MAE Std: 0.803435940264885<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 17 3<br>True 8 2<br>Precision: 0.4<br>Recall: 0.2<br>F1 score: 0.26666666666666666<br>Overall accuracy: 0.6333333333333333                     |
| 25   | 3       | MAE Mean: 6.324512421918359<br>MAE Std: 0.6415648041333166<br>Confusion matrix:<br>Predicted False True<br>Actual<br>False 47 13<br>True 14 1<br>Precision: 0.07142857142857142<br>Recall: 0.06666666666666667<br>F1 score: 0.0689655172413793<br>Overall accuracy: 0.64 |

Table 2: Tuning Results from Cosine Similarity x Weighted Sum Method

### Method 3: Pearson Correlation x Adjusted Weighted Sum

| Size | Repeats | Metrics   |
|------|---------|---|
| 5    | 3       | MAE mean: 10.420695865225055      Precision: 0.5<br>MAE std: 1.10630757447246      Recall: 0.3333333333333333<br>Confusion matrix:      F1 score: 0.4<br>Predicted False True      Overall accuracy: 0.6<br>Actual<br>False    7    2<br>True    4    2                                       |
| 5    | 10      | MAE mean: 8.525062048279056      Precision: 0.1333333333333333<br>MAE std: 2.58309659959855      Recall: 0.18181818181818182<br>Confusion matrix:      F1 score: 0.15384615384615383<br>Predicted False True      Overall accuracy: 0.56<br>Actual<br>False    26   13<br>True    9    2      |
| 10   | 3       | MAE mean: 6.993720072938085      Precision: 0.7142857142857143<br>MAE std: 0.6362753472749851      Recall: 0.625<br>Confusion matrix:      F1 score: 0.6666666666666666<br>Predicted False True      Overall accuracy:<br>Actual      0.8333333333333334<br>False    20   2<br>True    3    5 |
| 25   | 3       | MAE mean: 8.070722931734466      Precision: 0.6666666666666666<br>MAE std: 0.3562820900157139      Recall: 0.15384615384615385<br>Confusion matrix:      F1 score: 0.25<br>Predicted False True      Overall accuracy: 0.68<br>Actual<br>False    47   2<br>True    22   4                    |

*Table 3: Tuning Results from Pearson Correlation x Adjusted Weighted Sum Method*

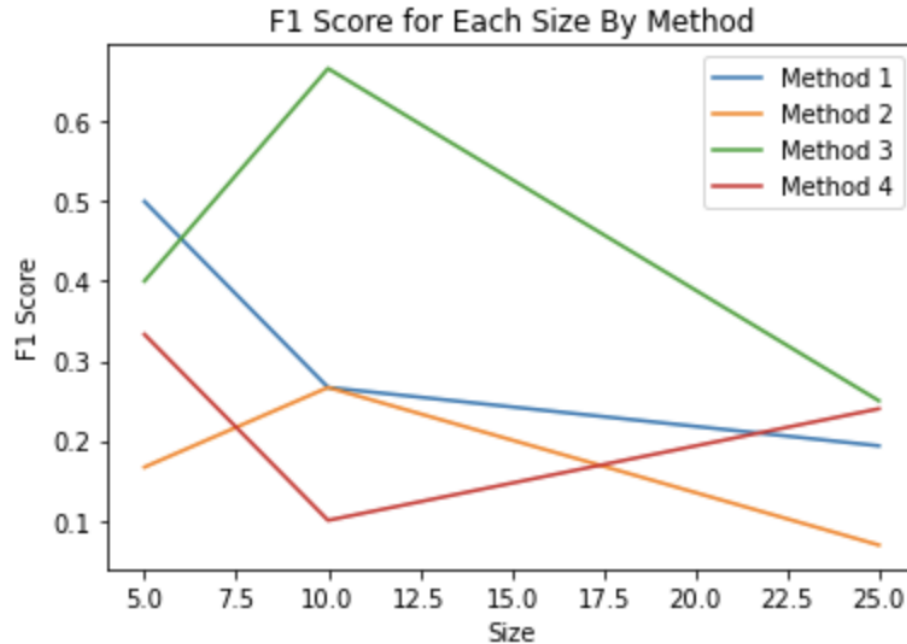
## Method 4: Cosine Similarity x Adjusted Weighted Sum

| Size | Repeats | Metrics   |
|------|---------|---|
| 5    | 3       | MAE mean: 6.6981471870664135      Precision: 0.3333333333333333<br>MAE std: 1.2909568203091843      Recall: 0.3333333333333333<br>Confusion matrix: <b>F1 score: 0.3333333333333333</b><br>Predicted False True      Overall accuracy:<br>Actual      0.4666666666666667<br>False      5      4<br>True      4      2 |
| 5    | 10      | MAE mean: 7.995128514427732      Precision: 0.2727272727272727<br>MAE std: 2.22662091102235      Recall: 0.23076923076923078<br>Confusion matrix:      F1 score: 0.24999999999999994<br>Predicted False True      Overall accuracy: 0.64<br>Actual<br>False      29      8<br>True      10      3                     |
| 10   | 3       | MAE mean: 7.178871441289935      Precision: 0.1<br>MAE std: 1.4413485528068426      Recall: 0.1<br>Confusion matrix:      F1 score: 0.10000000000000002<br>Predicted False True      Overall accuracy: 0.4<br>Actual<br>False      11      9<br>True      9      1  |
| 25   | 3       | MAE mean: 6.774160867934932      Precision: 0.3333333333333333<br>MAE std: 1.3769096769173423      Recall: 0.1875<br>Confusion matrix:      F1 score: 0.24000000000000005<br>Predicted False True      Overall accuracy:<br>Actual      0.7466666666666667<br>False      53      6<br>True      13      3             |

*Table 4: Tuning Results from Cosine Similarity x Adjusted Weighted Sum Method*

## Results

**Question 1: For each method, which values of size and repeats maximize the F1 score?**

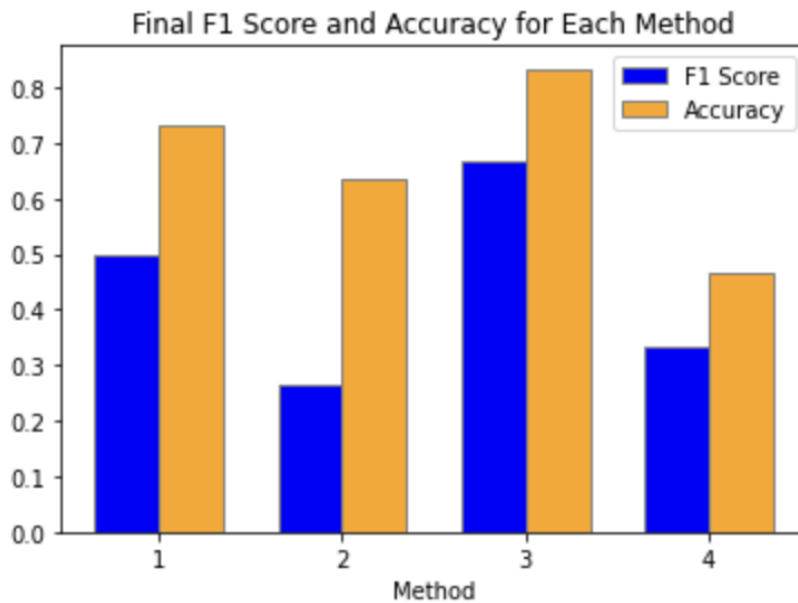


*Figure 1: Comparing Size and F1 Score for Each Method*

Based on the graph presented above, we can observe that the F1 score generally decreases as the size value increases. However, there are some exceptions to this trend. For instance, when the size value is 10 for method 3, the F1 score increases compared to when the size value is 3 but then declines as the size value gets larger. Similarly, for method 4, the F1 score initially decreases as the size value increases, but after reaching a size value of 10, it starts to increase.

Additionally, we conducted hyperparameter tuning to measure the impact of different repetition values. Tables 1-4 show that generally increasing the number of repeats resulted in a lower F1 score.

**Question 2: After tuning size and repeats, which method results in the highest F1 score?**



*Figure 2: Comparing F1 Score and Accuracy for Each Method*

From Figure 2, we see that method 3 has the highest F1 score and overall accuracy. For each of the methods, we observe that overall accuracy is always larger than F1 score.

## Conclusions

| Method | Input Parameters       | F1 Score |
|--------|------------------------|----------|
| 1      | Size = 5, Repeats = 3  | 0.5      |
| 2      | Size = 10, Repeats = 3 | 0.2666   |
| 3      | Size = 10, Repeats = 3 | 0.6666   |
| 4      | Size = 5, Repeats = 3  | 0.3333   |

*Table 5: Input Parameters to Achieve Highest F1 Score for Each Method*

For Method 1, we found that using a size of 5 and 3 repetitions provided the best results that balanced a low MAE mean, a low MAE standard deviation, and the highest F1-score of 0.5. It resulted in an overall accuracy of 0.7333.

For Method 2, we found that using a size of 10 and 3 repetitions provided the best results that balanced the lowest MAE mean, a low MAE standard deviation, and the highest F1-score of 0.2667. It resulted in an overall accuracy of 0.6333.

For Method 3, we found that using a size of 10 and 3 repetitions provided the best results that balanced the lowest MAE mean, a low MAE standard deviation, and the highest F1-score of 0.6667. It had an overall accuracy of about 0.833.

For Method 4, we found that using a size of 5 and 3 repetitions provided the best results that balanced a low MAE mean, MAE standard deviation, and the highest F1-score of 0.3333. It had an overall accuracy of 0.467.

Overall, we found that Method 3 was the best method out of all our methods. This method had the lower MAE mean, the lowest MAE standard deviation, the highest accuracy, and the highest F1 score out of all the methods.

Method 3's balanced approach to rating predictions is likely the key to its success. Unlike standard weighted sums, the adjusted weighted sums take into account the average ratings of users. This difference makes the method more reasonable as it accounts for individual user biases or trends in their ratings, providing a more accurate prediction. For instance, if a user generally rates jokes higher or lower than average, this method adjusts predictions accordingly, leading to more accurate and personalized recommendations.

The use of Pearson correlation further enhances the method's effectiveness. By measuring the strength and direction of the linear relationship between users' rating patterns, Pearson correlation effectively identifies users with similar tastes and preferences. This similarity measure, when combined with the nuanced predictions of adjusted weighted sums, creates a powerful tool for predicting user ratings. By accounting for user-specific rating tendencies and identifying similar users, this method optimizes collaborative filtering in this context.

Thus, we conclude that employing Pearson correlation as the similarity measure and an adjusted weighted sum as the predictor tends to yield better results in predicting user preferences for humor in our recommendation system.