ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

Knowledge Discovery from Databases

PP2: Recommender System

Nikolaos Sakellaris

ics21096

Thessaloniki, 2024

Περιεχόμενα

Introduction

The aim of the work was to implement and evaluate a collaborative filtering item-item recommendation system for movies using the Pearson coefficient as a similarity measure. The implementation was done in the Python programming language. The MovieLens data was used, specifically the ***ratings.csv*** file from the small subset of 100,000 records.

Four prediction functions were implemented in total, including:

- Weighted Mean

- Weighted Mean with adjustment of user's mean rating and bias removal

- Weighted Mean with weighting based on the number of common users who have rated the items

- Weighted Mean with weighting based on the variance of each item's ratings

The closest N items are selected using Pearson similarity.

The program splits the data into a training set T = 80% and a test set 100-T = 20%. Then, it displays in each case: a) Mean Absolute Error (MAE), b) Mean Precision (macro average precision), and c) Mean Recall (macro average recall). To calculate the evaluation metrics, a movie is considered relevant if its rating is >=3.

Pre-processing and Data Reduction

Due to the large volume of data (100,000 records), it was deemed necessary to use a data reduction algorithm to save time and optimize the performance of the final model.

The file ***filter-ratings.py*** includes code that:

- Initially, modifies the timestamp column and converts it into a recognizable date-time format.

- Then, applies a simple form of data reduction by keeping 1 out of every 5 records. This reduces the total volume of data to 20% of the original (from 100,000 to 20,000 records).

The results were saved in the file ***ratings-reduced.csv.***

Recommender Script

For a fixed training set of 80% and a test set of 20%, and for the values of N = (5, 10, 15, 20, 25), the experiment is conducted 5 times in total. A comparison is made between the 4 prediction functions, and the evaluation metrics are reported as the averages of all executions.

The prediction functions and the execution code of the experiment are contained in the file recommender.py. Specifically, we have the following functions:

1. **Weighted Average Function:**

```
    def predict_weighted_average(self, user_ratings, item_index, N):
        # Get N most similar items
        similar_items = np.argsort(self.similarity_matrix[item_index])[-N:]

        # Filter out items without user ratings
        valid_similar_items = similar_items[user_ratings[similar_items] != 0]

        # Calculate weighted average prediction
        numerator = np.sum(self.similarity_matrix[item_index,
valid_similar_items] * user_ratings[valid_similar_items])
        denominator = np.sum(np.abs(self.similarity_matrix[item_index,
valid_similar_items]))

        if denominator == 0:
            return 0  # Avoid division by zero

        prediction = numerator / denominator

        # Replace NaN with 0
        return np.nan_to_num(prediction)
```

2. **Weighted Adjusted Average minus the neighbors bias:**

```
    def predict_weighted_average_adjusted(self, user_ratings, item_index, N):
        # Get N most similar items
        similar_items = np.argsort(self.similarity_matrix[item_index])[-N:]

        # Filter out items without user ratings
        valid_similar_items = similar_items[user_ratings[similar_items] != 0]

        # Calculate adjusted weighted average prediction
        user_avg = np.mean(user_ratings)
        numerator = np.sum((self.similarity_matrix[item_index,
valid_similar_items] * (user_ratings[valid_similar_items] - user_avg)))
        denominator = np.sum(np.abs(self.similarity_matrix[item_index,
valid_similar_items]))
```

```
        if denominator == 0:
          return user_avg  # Return user's average rating if denominator is 0

        prediction = user_avg + (numerator / denominator)

        #Replace NaN with 0
        return np.nan_to_num(prediction)
```

3. **Weighted Average based on the number of Common Users:**

```
    def predict_weighted_average_common_users(self, user_ratings, item_index, N):
        # Get N most similar items
        similar_items = np.argsort(self.similarity_matrix[item_index])[-N:]

        # Filter out items without user ratings
        valid_similar_items = similar_items[user_ratings[similar_items] != 0]

        # Calculate weights based on the number of common users
        common_users_count = np.sum(self.ratings[:, valid_similar_items] != 0, axis=0)

        # Check if there are common users
        if np.any(common_users_count):
            weights = common_users_count / np.max(common_users_count)  # Normalize weights
        else:
            # If there are no common users, assign equal weights to all items
            weights = np.ones_like(common_users_count) / len(common_users_count)

        # Calculate weighted average prediction with common users-based weights
        numerator = np.sum(self.similarity_matrix[item_index, valid_similar_items] * user_ratings[valid_similar_items] * weights)
        denominator = np.sum(np.abs(self.similarity_matrix[item_index, valid_similar_items]) * weights)

        if denominator == 0:
            return 0  # Avoid division by zero

        prediction = numerator / denominator

        # Replace NaN with 0
        return np.nan_to_num(prediction)
```

The weighting of each neighbor is determined by the number of common users who have rated both the target item and the neighboring item. The more common users, the greater the weighting factor. The weight of each neighbor is normalized by dividing it by the maximum number of common users among all neighbors. If there are no common users for the specific items, each neighbor is assigned the same weighting factor.

common_users_count = np.sum(self.ratings[:, valid_similar_items] != 0, axis=0)

if np.any(common_users_count):

weights = common_users_count / np.max(common_users_count)

else:

np.ones_like(common_users_count) / len(common_users_count)

4. **Weighted Average based on the Variance of ratings of each item:**

```
    def predict_weighted_average_variance(self, user_ratings, item_index, N):
        # Get N most similar items
        similar_items = np.argsort(self.similarity_matrix[item_index])[-N:]

        # Filter out items without user ratings
        valid_similar_items = similar_items[user_ratings[similar_items] != 0]

        # Calculate weights based on the variance of ratings
        weights = np.var(self.ratings[:, valid_similar_items], axis=0)

        # Calculate weighted average prediction with variance-based weights
        numerator = np.sum(self.similarity_matrix[item_index,
valid_similar_items] * user_ratings[valid_similar_items] * weights)
        denominator = np.sum(np.abs(self.similarity_matrix[item_index,
valid_similar_items]) * weights)

        if denominator == 0:
            return 0  # Avoid division by zero

        prediction = numerator / denominator

        # Replace NaN with 0
        return np.nan_to_num(prediction)
```

The weighting of each neighbor is determined by the difference in ratings for that neighboring item. The greater the difference, the greater the weighting factor. The weight is calculated using the difference in ratings for each neighboring item. If the difference is 0, a very small value is added to avoid division by 0.

weights = np.var(self.ratings[:, valid_similar_items], axis=0) + 1e-9

The experiment was executed a total of 5 times for different values of N. For each experiment, precision was maintained at 5 decimal places, and the average of all prediction functions as well as the average of each evaluation metric (MAE, mean precision, mean recall) was calculated.

The code produces results separately for each function, which are printed in the following format in the console:

```
Experiment for N=5 (1/5):
    Original Weighted Average (MAE): 3.5127958172812326
    Original Weighted Average Precision: 0.4667534655285932
    Original Weighted Average Recall: 0.4969693922130952
    Weighted Average with Adjustment (MAE): 3.4662397927839144
    Weighted Average with Adjustment Precision: 0.4667534655285932
    Weighted Average with Adjustment Recall: 0.4969693922130952
    Weighted Average with Variance (MAE): 3.513066993091251
    Weighted Average with Variance Precision: 0.4667534655285932
    Weighted Average with Variance Recall: 0.4969693922130952
    Weighted Average with Common Users (MAE): 3.5130334562535612
    Weighted Average with Common Users Precision: 0.4667534655285932
    Weighted Average with Common Users Recall: 0.4969693922130952

Experiment for N=10 (2/5):
    Original Weighted Average (MAE): 3.4166116974494285
    Original Weighted Average Precision: 0.5134238464679461
    Original Weighted Average Recall: 0.5022638602259987
    Weighted Average with Adjustment (MAE): 3.3646190420186404
    Weighted Average with Adjustment Precision: 0.5134238464679461
    Weighted Average with Adjustment Recall: 0.5022638602259987
    Weighted Average with Variance (MAE): 3.4166966946299504
    Weighted Average with Variance Precision: 0.5134238464679461
    Weighted Average with Variance Recall: 0.5022638602259987
    Weighted Average with Common Users (MAE): 3.4166083871941417
    Weighted Average with Common Users Precision: 0.5141234456251413
    Weighted Average with Common Users Recall: 0.5024005095837467
```

Experiment 1

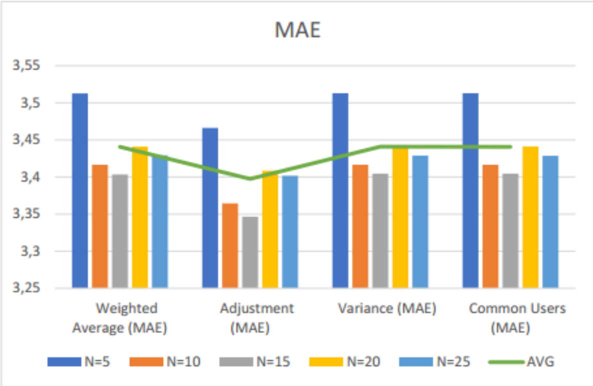
Below is the table showing the results of each function during the execution of the 1st experiment for the defined values of N. Then, the average of these results is calculated:

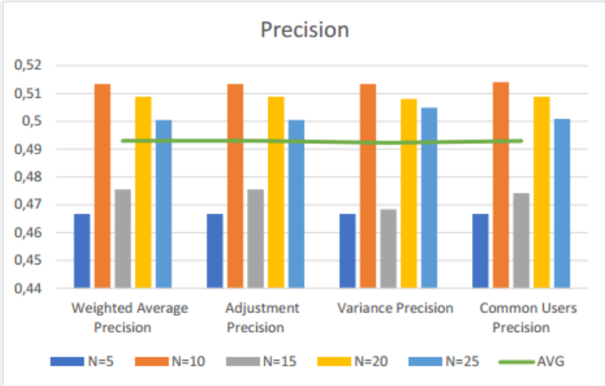
Experiment 1	N=5	N=10	N=15	N=20	N=25	AVG
Weighted Average (MAE)	3,5128	3,41661	3,40356	3,4411	3,42915	3,440644
Weighted Average Precision	0,46675	0,51342	0,47554	0,50887	0,5005	0,493016
Weighted Average Recall	0,49697	0,50226	0,49292	0,50289	0,50017	0,499042
Adjustment (MAE)	3,46624	3,36462	3,34658	3,40838	3,40168	3,3975
Adjustment Precision	0,46675	0,51342	0,47554	0,50887	0,5005	0,493016
Adjustment Recall	0,49697	0,50226	0,49292	0,50289	0,50017	0,499042
Variance (MAE)	3,51307	3,4167	3,40456	3,44157	3,42893	3,440966
Variance Precision	0,46675	0,51342	0,46838	0,50803	0,50488	0,492292
Variance Recall	0,49697	0,50226	0,49115	0,50259	0,50163	0,49892
Common Users (MAE)	3,51303	3,41661	3,40443	3,44136	3,4288	3,440846
Common Users Precision	0,46675	0,51412	0,47421	0,50887	0,5009	0,49297
Common Users Recall	0,49697	0,5024	0,49262	0,50289	0,5003	0,499036

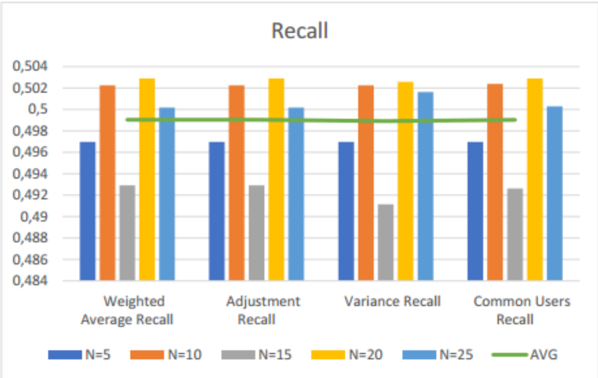
Average Total MAE	3,429989
Average Total Precision	0,492824
Average Total Recall	0,49901

Where "**Weighted Average**" denotes the *Original Weighted Average* function, "**Adjustment**" denotes the *Weighted Average with Adjustment function*, "**Variance**" denotes the *Weighted Average with Variance* function, and "**Common Users**" denotes the *Weighted Average with Common Users* function.

The values of these functions are represented graphically in the following plots:







Observations:

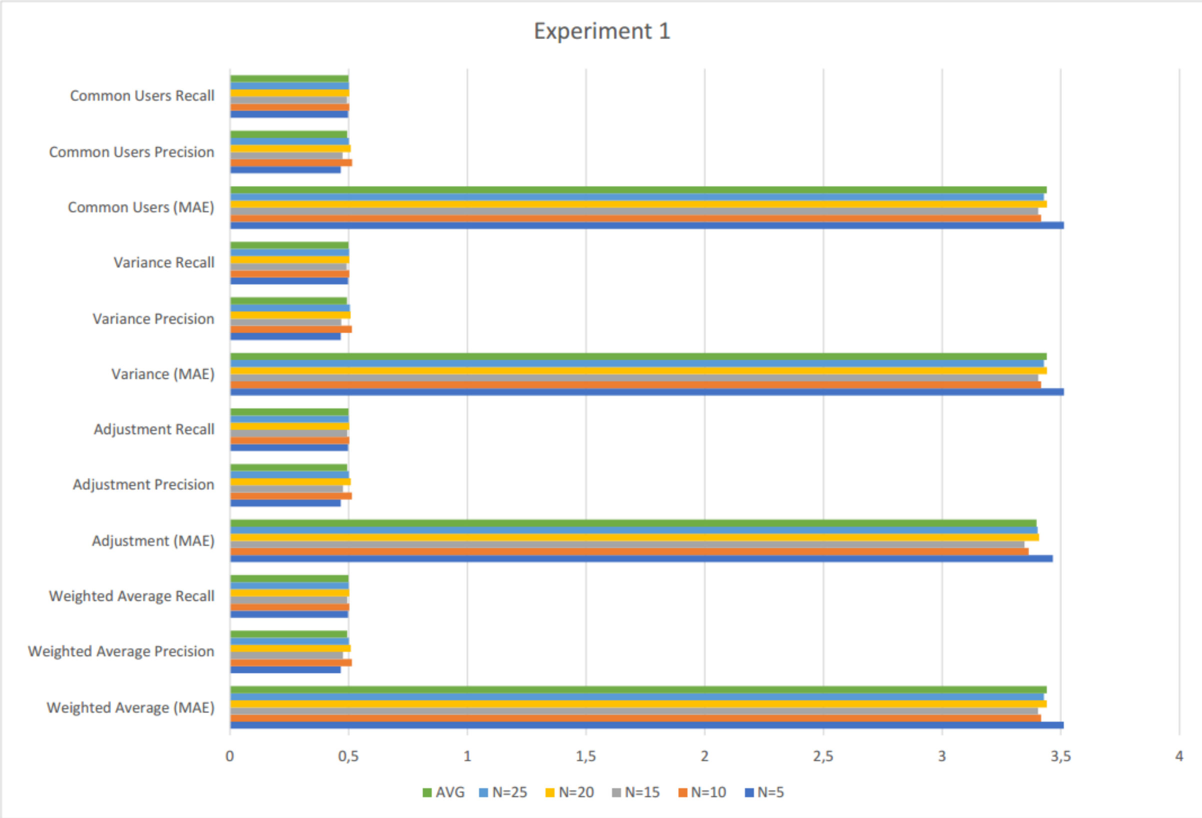
Based on the values we observe, there is a relative balance in the results.

Specifically, the **MAE** values range from **3.35** to **3.5**, encountered when **N=5**.

Regarding Precision, we notice a small variation in the deviation of the values, with cases of **N=10, 20, 25** being close to **0.5**, while for **N=5, 15**, they do not exceed **0.47**. The average value is around **0.49**.

For Recall, the values vary much less among them and generally range from **0.49** to **0.5**.

The average overall **MAE** is **3.42**, while **Precision** and **Recall** are **0.49**.



Overall, in the above diagram, it is observed that the values of Precision and Recall do not differ significantly both among the functions and among the different values of N each time. On the contrary, MAE shows a significant difference compared to the other two metrics, but the values of the functions are still quite close, with the only slight variation being for N=5.

Similar observations could be made for experiments 2-5. The diagrams do not differ significantly, while the MAE, Precision, and Recall metrics almost exactly range within the same values.

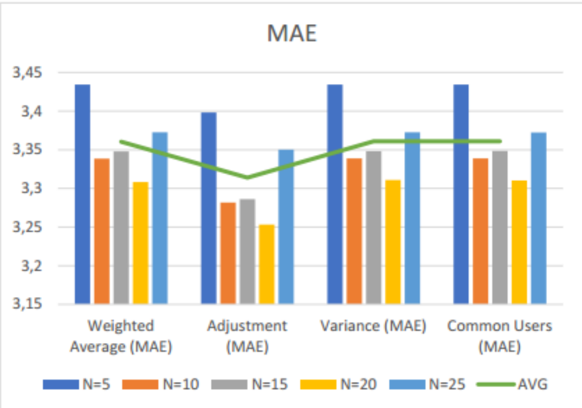
Experiment 2

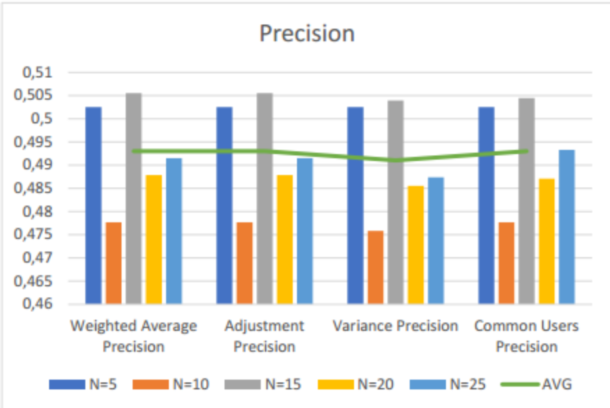
2nd Experiment results:

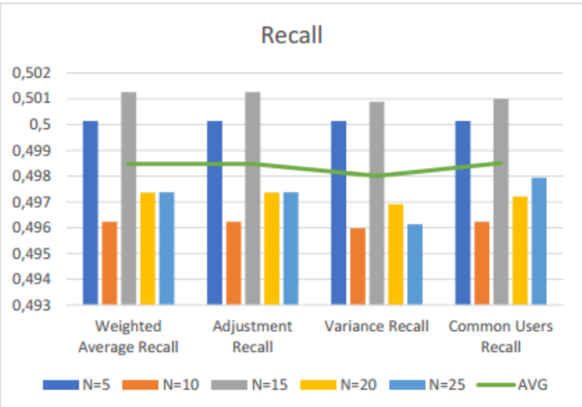
Experiment 2	**N=5**	**N=10**	**N=15**	**N=20**	**N=25**	**AVG**
Weighted Average (MAE)	3,43442	3,33855	3,34778	3,30835	3,37248	3,360316
Weighted Average Precision	0,50252	0,47768	0,50556	0,48786	0,49152	0,493028
Weighted Average Recall	0,50014	0,49623	0,50126	0,49736	0,49737	0,498472
Adjustment (MAE)	3,3982	3,28171	3,28608	3,25328	3,35022	3,313898
Adjustment Precision	0,50252	0,47768	0,50556	0,48786	0,49152	0,493028
Adjustment Recall	0,50014	0,49623	0,50126	0,49736	0,49737	0,498472
Variance (MAE)	3,43435	3,33884	3,34826	3,31082	3,37245	3,360944
Variance Precision	0,50252	0,47582	0,50395	0,48556	0,48739	0,491048
Variance Recall	0,50014	0,49597	0,50088	0,49691	0,49613	0,498006
Common Users (MAE)	3,43438	3,33891	3,34837	3,31027	3,37228	3,360842
Common Users Precision	0,50252	0,47768	0,50449	0,48711	0,49329	0,493018
Common Users Recall	0,50014	0,49623	0,501	0,49721	0,49794	0,498504

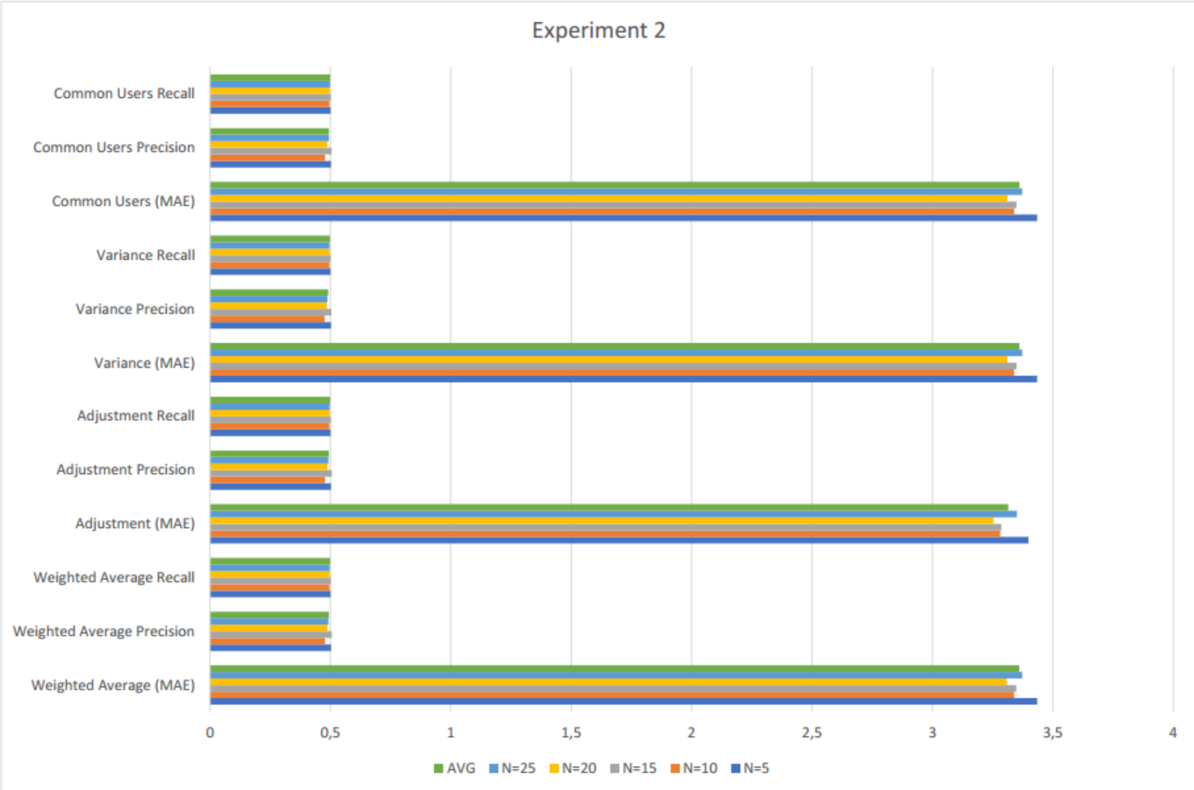
Average Total MAE	3,349
Average Total Precision	0,4925305
Average Total Recall	0,4983635

Related graphic representations:









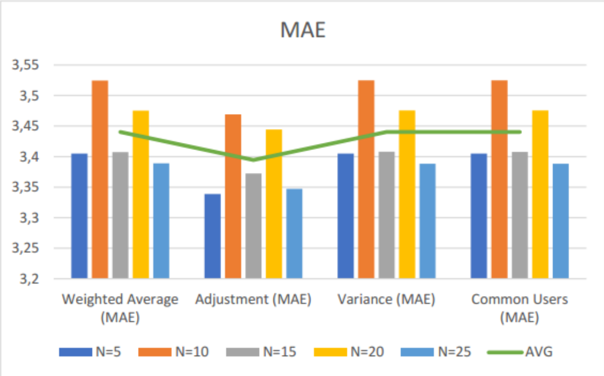
Experiment 3

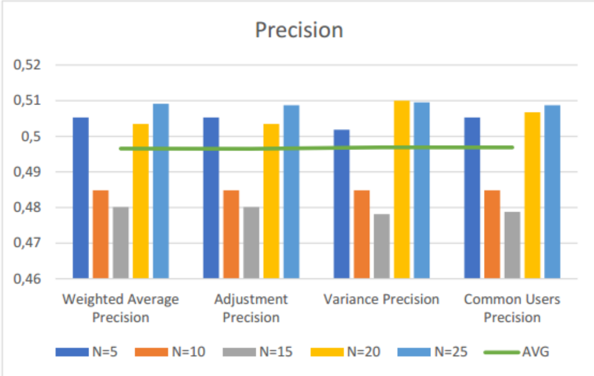
3rd Experiment results:

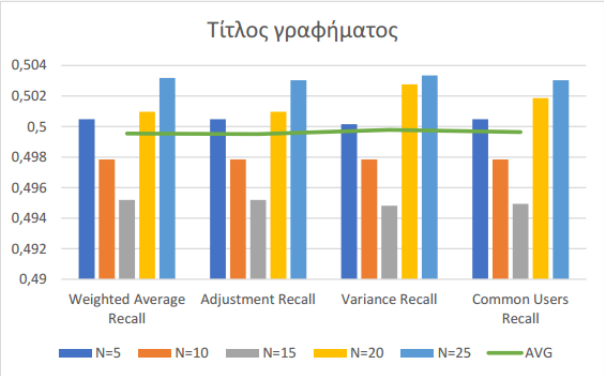
Experiment 3	**N=5**	**N=10**	**N=15**	**N=20**	**N=25**	**AVG**
Weighted Average (MAE)	3,40514	3,52458	3,40739	3,47534	3,38893	3,440276
Weighted Average Precision	0,5053	0,48481	0,48011	0,50346	0,50914	0,496564
Weighted Average Recall	0,50048	0,49785	0,4952	0,50097	0,50319	0,499538
Adjustment (MAE)	3,33879	3,46933	3,37243	3,44456	3,34713	3,394448
Adjustment Precision	0,5053	0,48481	0,48011	0,50346	0,50873	0,496482
Adjustment Recall	0,50048	0,49785	0,4952	0,50097	0,50304	0,499508
Variance (MAE)	3,40512	3,52483	3,40789	3,47554	3,38847	3,44037
Variance Precision	0,50186	0,48481	0,47813	0,50999	0,50954	0,496866
Variance Recall	0,50016	0,49785	0,49481	0,50276	0,50335	0,499786
Common Users (MAE)	3,40508	3,52477	3,40766	3,47556	3,38841	3,440296
Common Users Precision	0,5053	0,48481	0,4788	0,5067	0,50873	0,496868
Common Users Recall	0,50048	0,49785	0,49494	0,50187	0,50304	0,499636

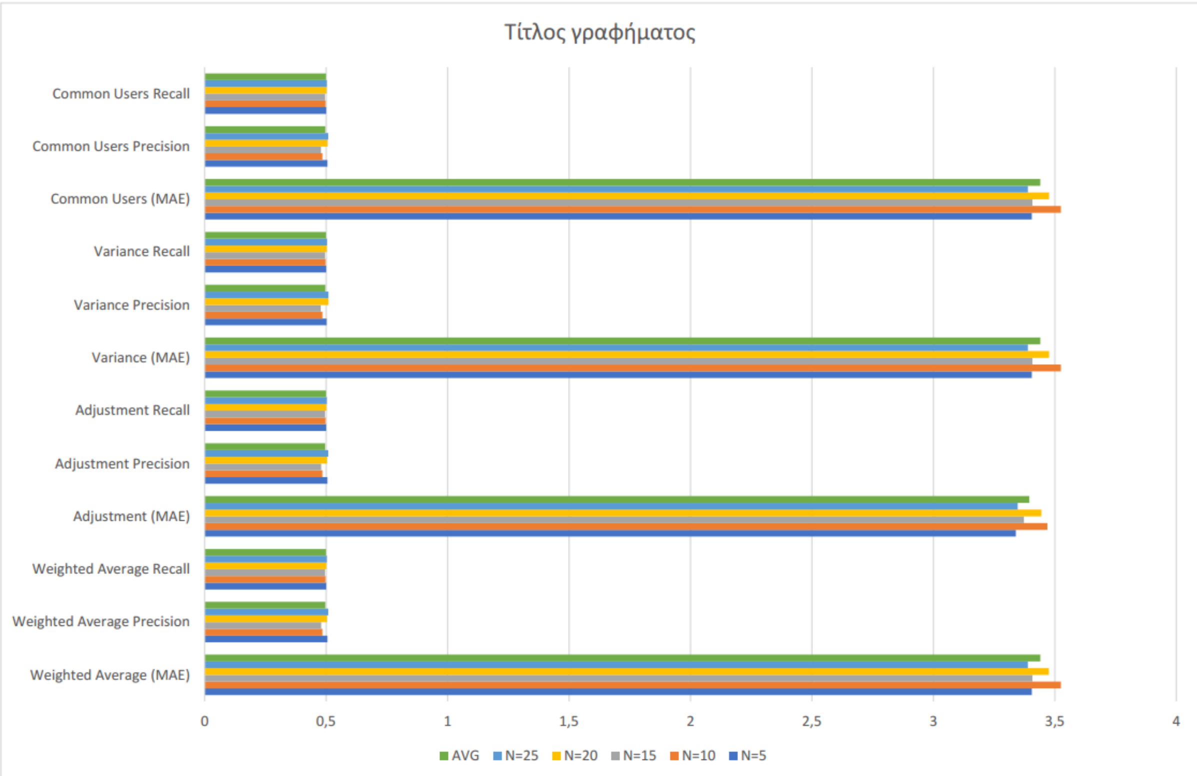
Average Total MAE	3,428848
Average Total Precision	0,496695
Average Total Recall	0,499617

Related graphic representations:









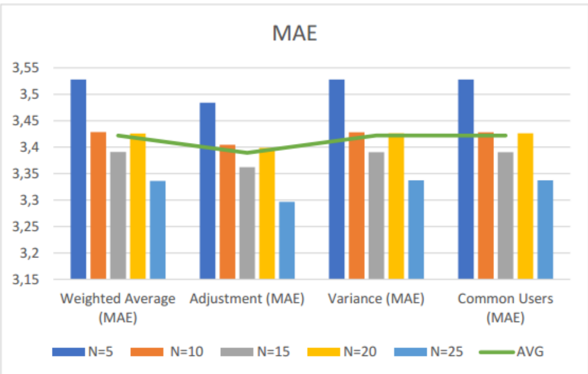
Experiment 4

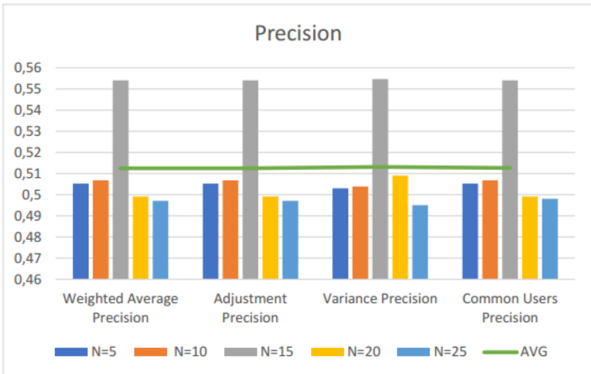
4th Experiment results:

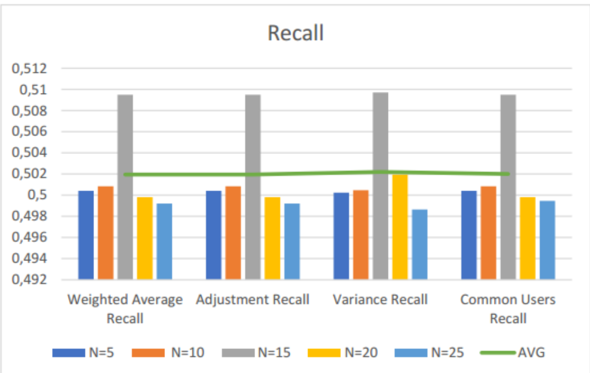
Experiment 4	N=5	N=10	N=15	N=20	N=25	AVG
Weighted Average (MAE)	3,52795	3,42848	3,39101	3,42564	3,33612	3,42184
Weighted Average Precision	0,5053	0,5068	0,55409	0,49919	0,49717	0,51251
Weighted Average Recall	0,50041	0,50085	0,50949	0,49982	0,49921	0,501956
Adjustment (MAE)	3,48417	3,40434	3,36231	3,39867	3,29672	3,389242
Adjustment Precision	0,5053	0,5068	0,55409	0,49919	0,49717	0,51251
Adjustment Recall	0,50041	0,50085	0,50949	0,49982	0,49921	0,501956
Variance (MAE)	3,52792	3,4283	3,39066	3,42621	3,33742	3,422102
Variance Precision	0,50303	0,5039	0,55465	0,50907	0,49513	0,513156
Variance Recall	0,50023	0,50047	0,50971	0,50195	0,49864	0,5022
Common Users (MAE)	3,52792	3,4283	3,39064	3,42633	3,33714	3,422066
Common Users Precision	0,5053	0,5068	0,55409	0,49919	0,49808	0,512692
Common Users Recall	0,50041	0,50085	0,50949	0,49982	0,49947	0,502008

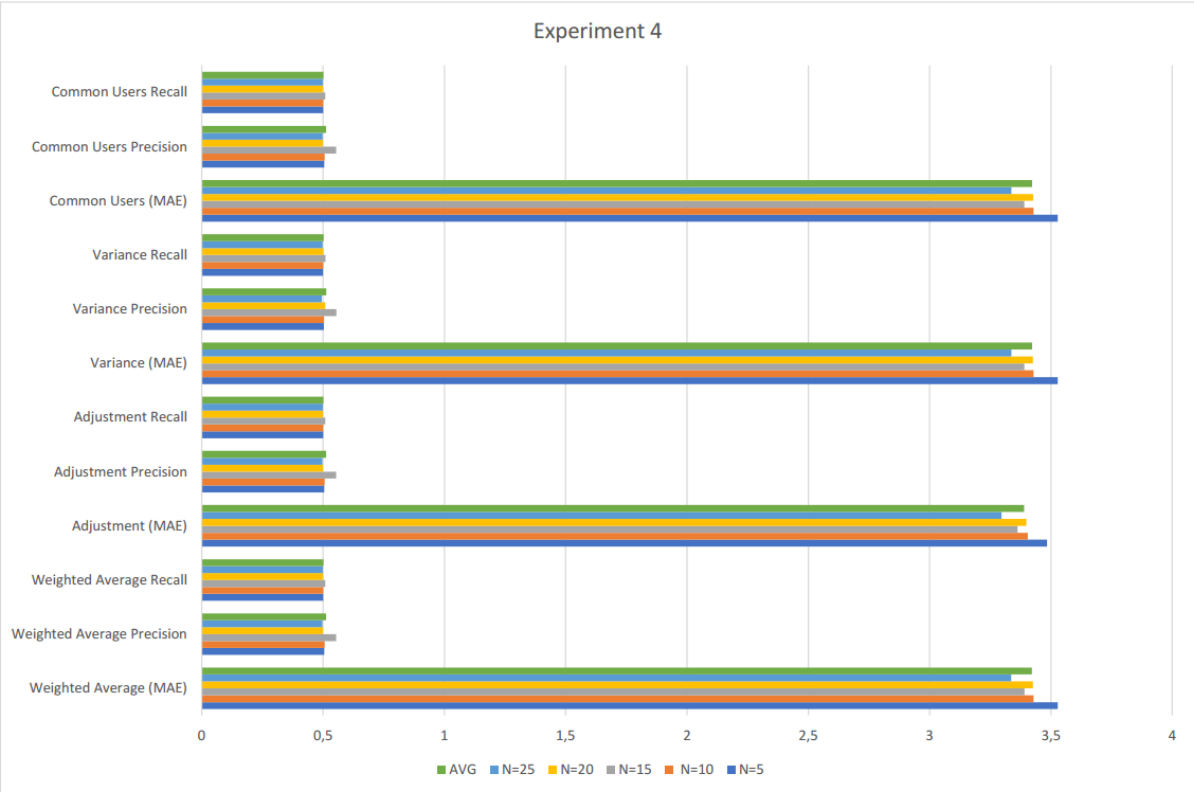
Average Total MAE	3,413813
Average Total Precision	0,512717
Average Total Recall	0,50203

Related graphic representations:









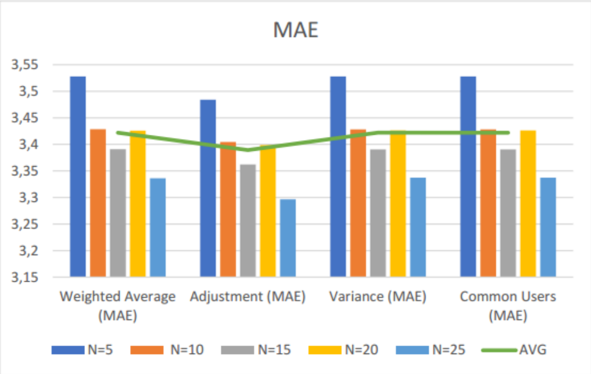
Experiment 5

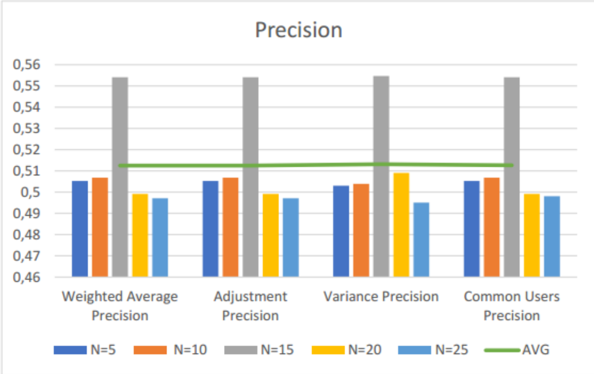
5th Experiment results:

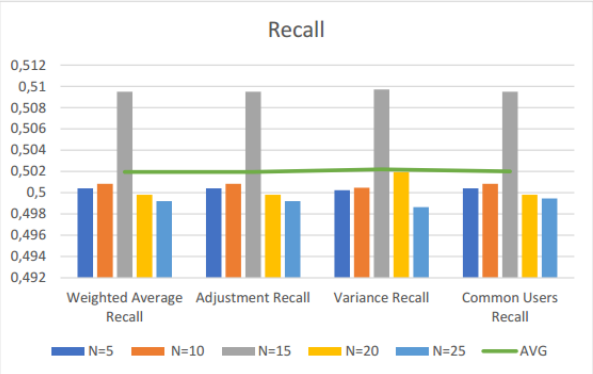
Experiment 5	N=5	N=10	N=15	N=20	N=25	AVG
Weighted Average (MAE)	3,5128	3,41661	3,40356	3,4411	3,42915	3,440644
Weighted Average Precision	0,46675	0,51342	0,47554	0,50887	0,5005	0,493016
Weighted Average Recall	0,49697	0,50226	0,49292	0,50289	0,50017	0,499042
Adjustment (MAE)	3,46624	3,36462	3,34658	3,40838	3,40168	3,3975
Adjustment Precision	0,46675	0,51342	0,47554	0,50887	0,5005	0,493016
Adjustment Recall	0,49697	0,50226	0,49292	0,50289	0,50017	0,499042
Variance (MAE)	3,51307	3,4167	3,40456	3,44157	3,42893	3,440966
Variance Precision	0,46675	0,51342	0,46838	0,50803	0,50488	0,492292
Variance Recall	0,49697	0,50226	0,49115	0,50259	0,50163	0,49892
Common Users (MAE)	3,51303	3,41661	3,40443	3,44136	3,4288	3,440846
Common Users Precision	0,46675	0,51412	0,47421	0,50887	0,5009	0,49297
Common Users Recall	0,49697	0,5024	0,49262	0,50289	0,5003	0,499036

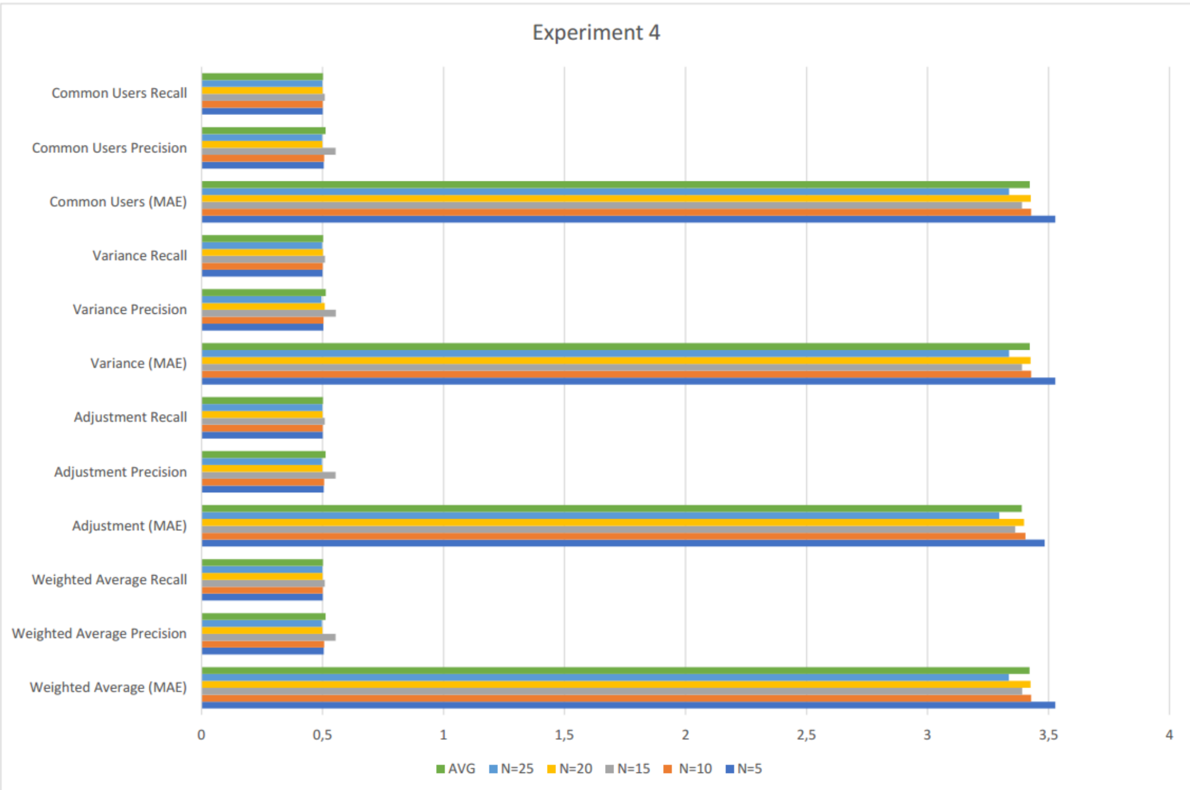
Average Total MAE	3,429989
Average Total Precision	0,492824
Average Total Recall	0,49901

Related graphic representations:









Final Results Average

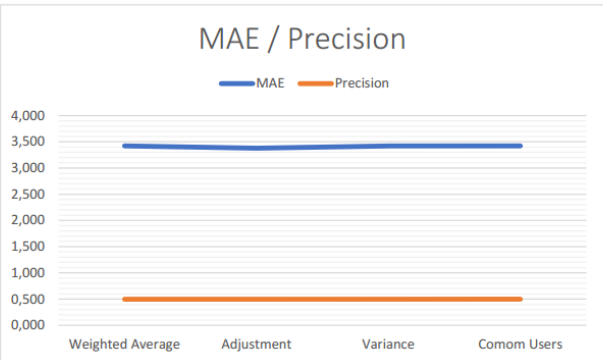
For the sake of simplifying the presentation of the data, rounding to 3 decimal places was performed. The results of the AVG (Average) column of each experiment were considered together, resulting in the Total Average column, as well as the total values for MAE, Precision, and Recall.

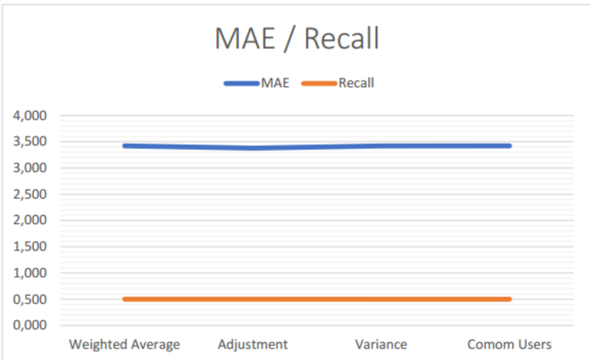
Functions	Exp 1	Exp 2	Exp 3	Exp 4	Exp 5	Total Average
Weighted Average (MAE)	3,441	3,360	3,440	3,422	3,441	3,421
Weighted Average Precision	0,493	0,493	0,497	0,513	0,493	0,498
Weighted Average Recall	0,499	0,498	0,500	0,502	0,499	0,500
Adjustment (MAE)	3,398	3,314	3,394	3,389	3,398	3,379
Adjustment Precision	0,493	0,493	0,496	0,513	0,493	0,498
Adjustment Recall	0,499	0,498	0,500	0,502	0,499	0,500
Variance (MAE)	3,441	3,361	3,440	3,422	3,441	3,421
Variance Precision	0,492	0,491	0,497	0,513	0,492	0,497
Variance Recall	0,499	0,498	0,500	0,502	0,499	0,500
Common Users (MAE)	3,441	3,361	3,440	3,422	3,441	3,421
Common Users Precision	0,493	0,493	0,497	0,513	0,493	0,498
Common Users Recall	0,499	0,499	0,500	0,502	0,499	0,500

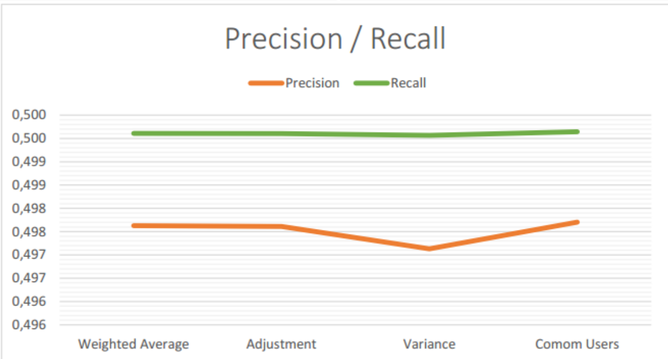
Total Average MAE	3,410
Total Average Precision	0,498
Total Average Recall	0,500

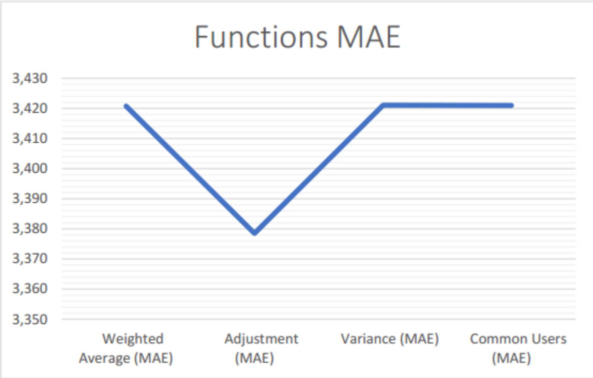
Functions/Values	Weighted Average	Adjustment	Variance	Comom Users
MAE	3,421	3,379	3,421	3,421
Precision	0,498	0,498	0,497	0,498
Recall	0,500	0,500	0,500	0,500

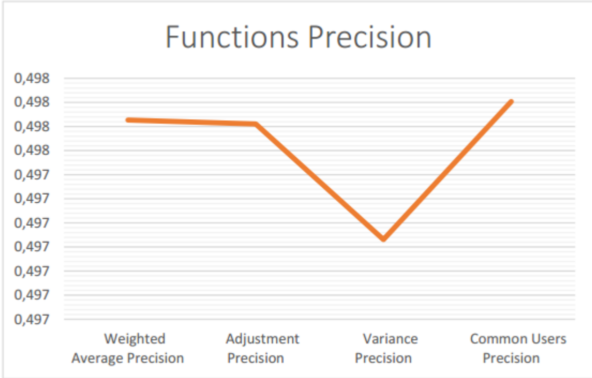
The respective plots represent the aggregate of results for the average of the 5 experiments:

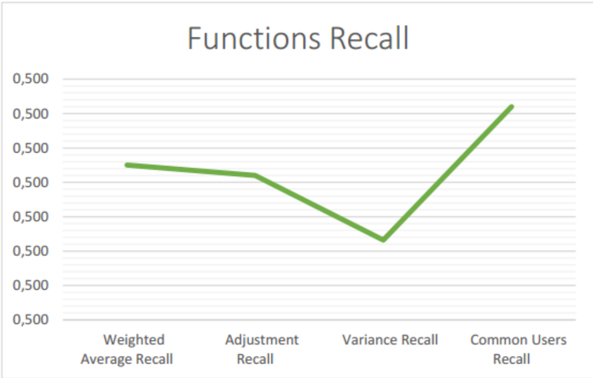












Observations:

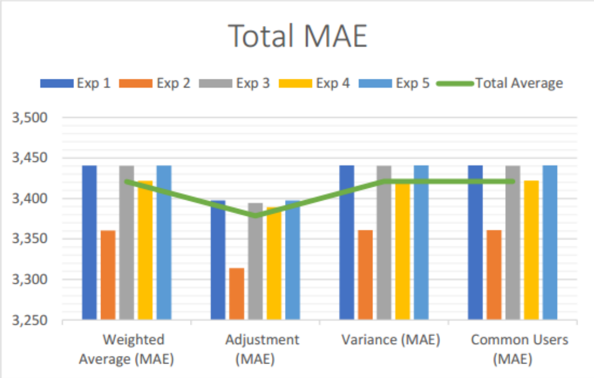
In the above plots, the variations in the mean values for MAE, Precision, and Recall are discernible. These variations are negligible, as the resulting curve does not change by more than a few decimal units in any case. Specifically, for MAE, the variation is approximately 0.05, while for the other two metrics, it is comparatively much smaller, approaching zero.

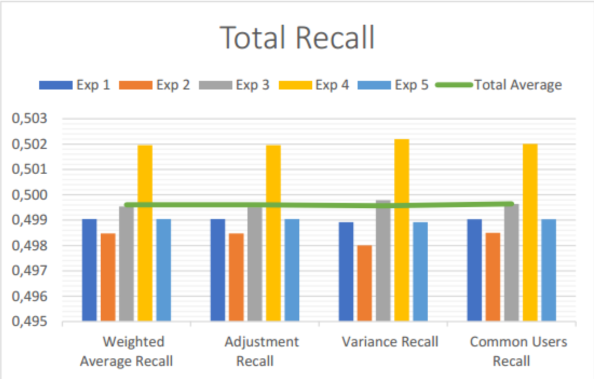
In conclusion, we could arrive at the following values:

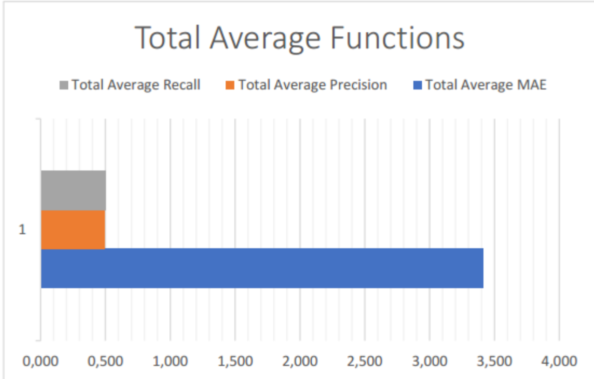
MAE = 3.40

Precision = 0.49

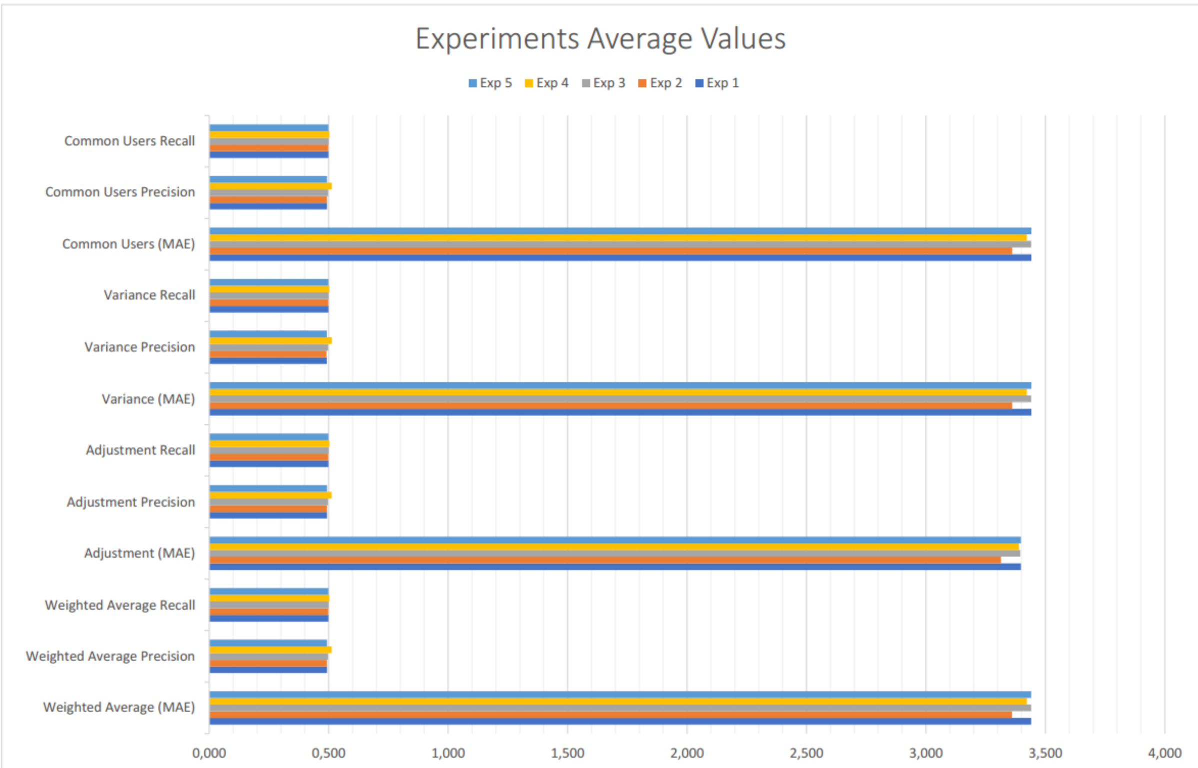
Recall = 0.50







Finally, the overall mean values for each function are presented, as well as for each different experiment case. A small observation in these experiments could be the relatively higher value for experiment 4. However, even this value does not significantly deviate from the average, as this difference is at the level of a few decimal digits.



Files

The final deliverable file contains:

- The experiment code in the files ***filter-ratings.py*** and ***recommender.py*** in the scripts folder.

- The results in Excel format and PDF files in the results folder.

- The original and reduced datasets ***ratings.csv*** and ***ratings-reduced.csv***.