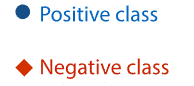
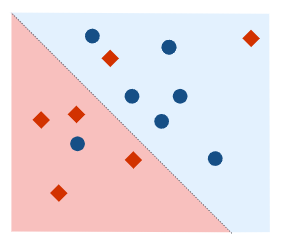
CPSC 4830

Data Mining for Data Analytics

Spring 2024 Assignment 2

Part 1

1. Bagging and Boosting
2. Given the graph below. The line is the classifier and the arrow refer to the positive orientation.
3. Circle all the point(s) whose weight will be increased as a result of Adaboost.
4. Draw in the same figure a possible classifier that we could select at the next boosting iteration. Please indicate both the classifying line and positive orientation.





1. Let be an i.i.d. sample from a distribution. To estimate the median of the distribution, you can compute the sample median of the data. How do we compute an estimate of the variance of the sample median?

-> To estimate the variance of the sample median bootstrapping is used. Bootstrapping is a resampling method where you repeatedly sample with replacement from the original dataset to create multiple bootstrap samples. Then after computing the sample median for each bootstrap sample, we can estimate the variance of the sample median.

1. Given the following code, explain the random forest model being used.

rf\_clf = RandomForestClassifier(n\_estimators=100, criterion="entropy", min\_samples\_split=10, n\_jobs = -1, random\_state=42)

->

1. N estimators: This parameter specifies the number of decision trees in the random forest. In this case, `n\_estimators=100` means that the random forest will consist of 100 decision trees.

2. `criterion`: This parameter specifies the function to measure the quality of a split. "entropy" is used here, which means the decision trees in the random forest will use information gain based on entropy to determine the best split at each node.

3. `min\_samples\_split`: This parameter specifies the minimum number of samples required to split an internal node. Here, `min\_samples\_split=10` means that a node will only be split if it contains at least 10 samples.

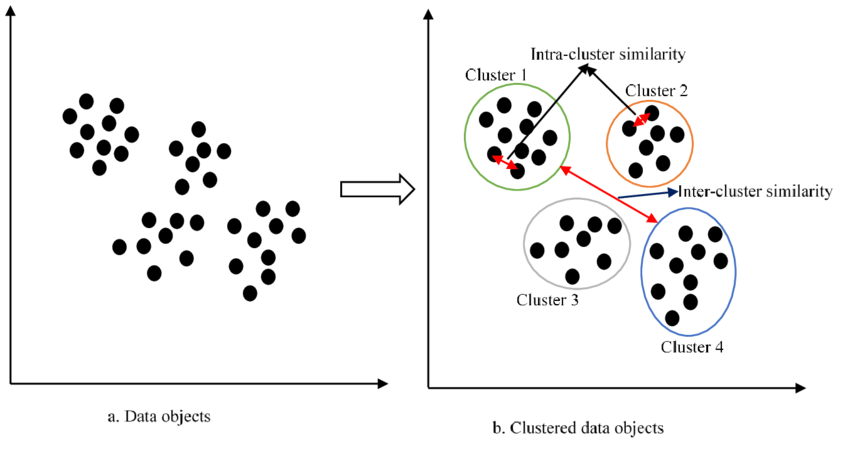
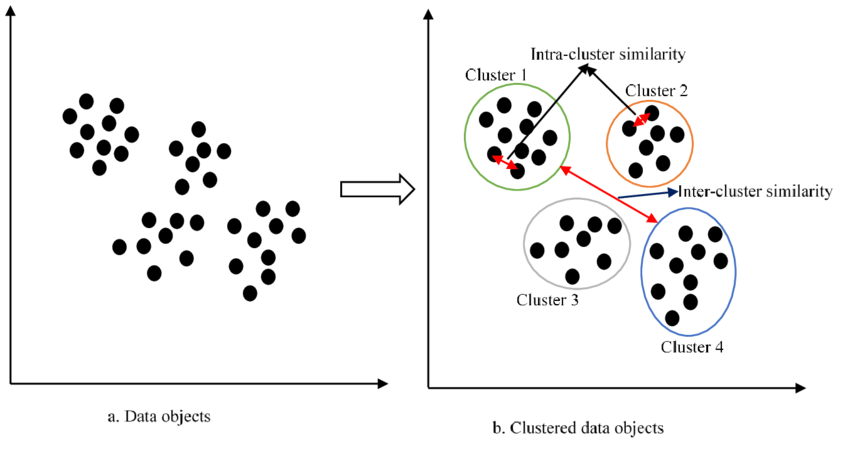
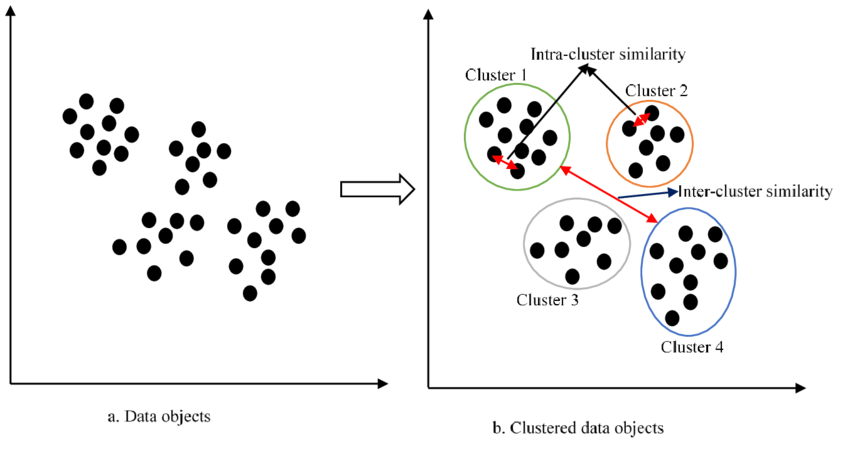
4. `n\_jobs`: This parameter specifies the number of jobs to run in parallel for both fitting and predicting. Setting `n\_jobs=-1` means that the computation will be distributed across all available CPU cores.

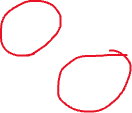
5. `random\_state`: This parameter is used to set the random seed for reproducibility. By setting `random\_state=42`, the random number generator will be initialized with a fixed seed value, ensuring that the results are reproducible across multiple runs.

1. K-means
2. Given the following dataset in the following graph, what is the possible number of groups? List out your answer.

->2 & 4.

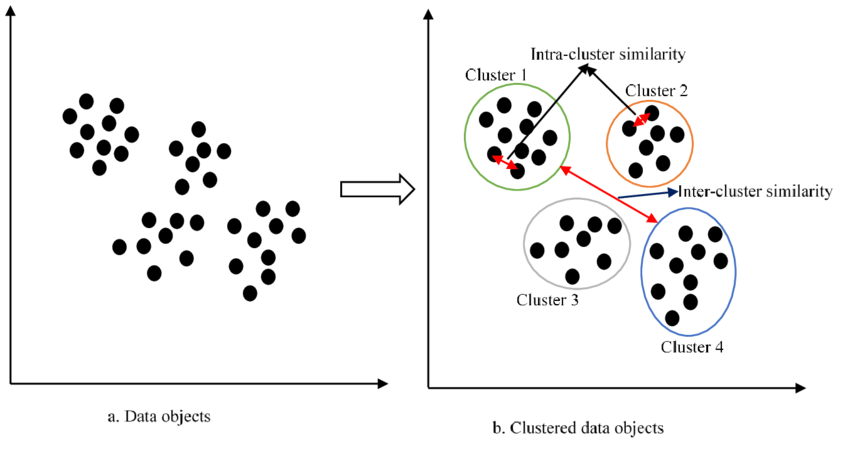
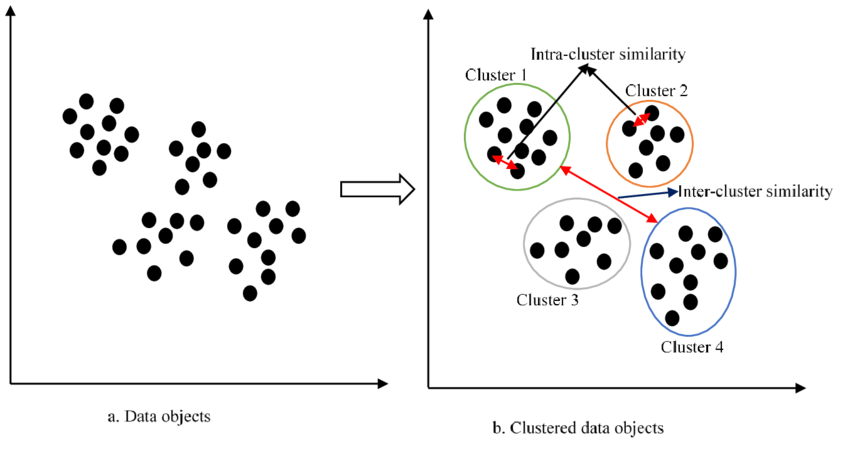
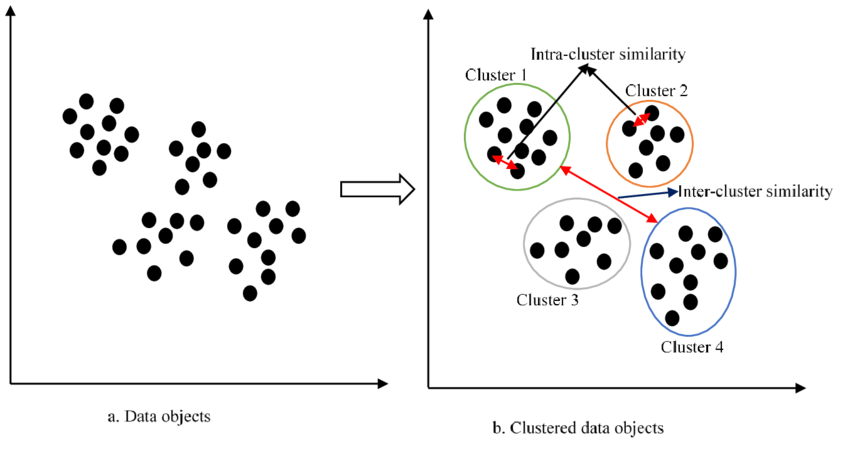
1. Suppose k=2 and “X” is the initial seeds (at the upper left and bottom right corner), and we are using K-means to separate the group. Draw the approximate position of “X” (calculation is not required) AND the line of separation for each step until the algorithm terminates.



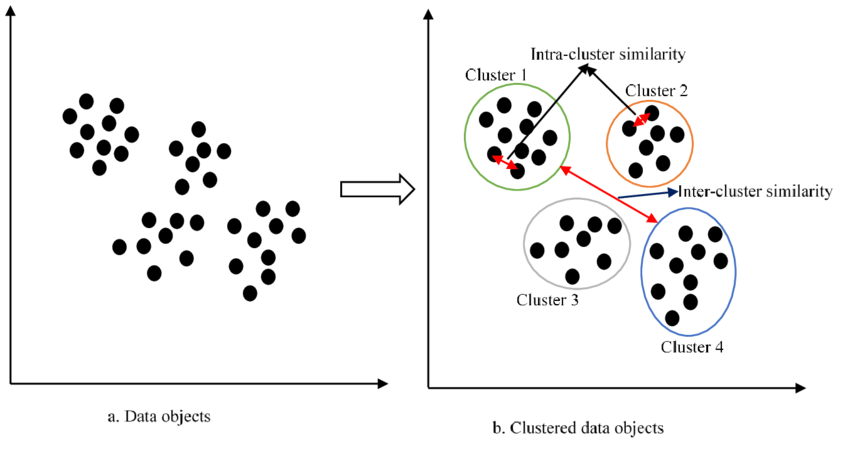
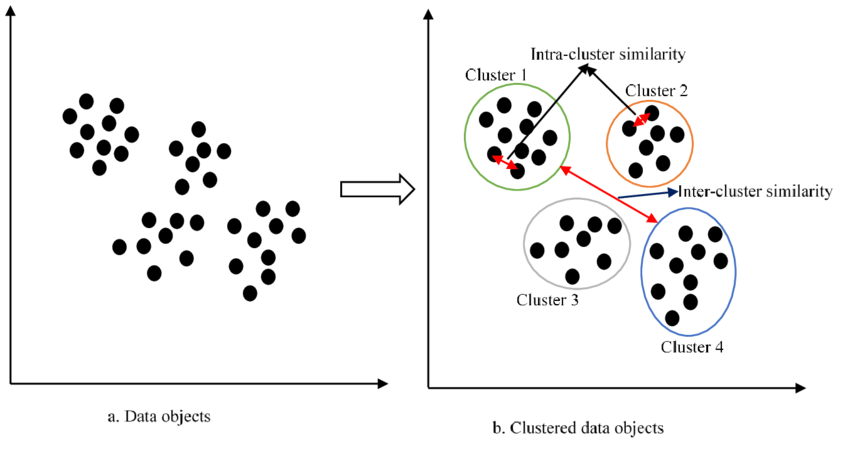
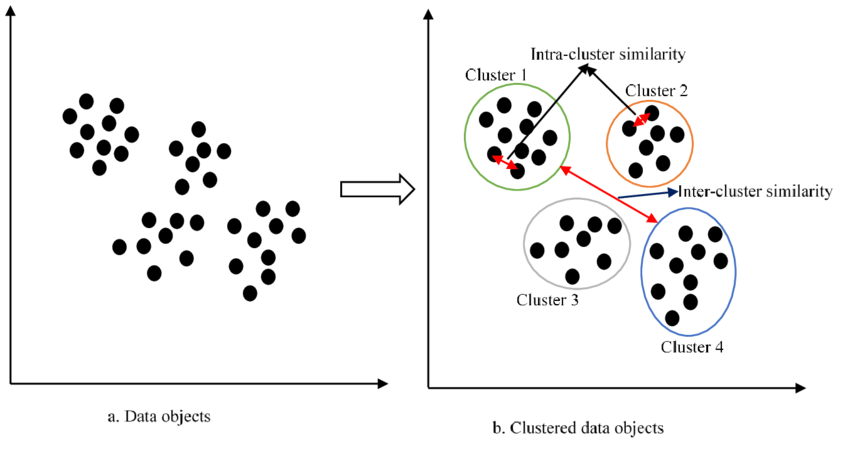


X

X







1. Given three clusters, A, B and C, containing a total of six points, where each point is defined by an integer value in one dimension, A = {0, 3, 12}, B = {2, 8} and C = {15}, which two clusters will be merged at the next iteration of HAC when using Euclidean distance and Single linkage.

-> In Hierarchical Agglomerative Clustering (HAC) with single linkage, at each iteration, the two clusters that are closest together based on the minimum distance between any two points (one from each cluster) are merged into a single cluster. Here, we'll compute the distances between clusters using Euclidean distance.

Given the clusters:

- A = {0, 3, 12}

- B = {2, 8}

- C = {15}

We calculate the distances between all pairs of clusters using single linkage:

1. Distance between A and B:

- Minimum distance between any point in A and any point in B:

- min(|0 - 2|, |0 - 8|, |3 - 2|, |3 - 8|, |12 - 2|, |12 - 8|) = min(2, 5, 1, 5, 10, 4) = 1

- Distance between A and B: 1

2. Distance between A and C:

- Minimum distance between any point in A and any point in C:

- min(|0 - 15|, |3 - 15|, |12 - 15|) = min(15, 12, 3) = 3

- Distance between A and C: 3

3. Distance between B and C:

- Minimum distance between any point in B and any point in C:

- min(|2 - 15|, |8 - 15|) = min(13, 7) = 7

- Distance between B and C: 7

Now, we compare the distances between all pairs of clusters:

- Distance between A and B: 1

- Distance between A and C: 3

- Distance between B and C: 7

The smallest distance is between clusters A and B (distance = 1). Therefore, at the next iteration of HAC, clusters A and B will be merged.

1. K-Means Clustering

Consider performing K-Means Clustering on a one-dimensional dataset containing four data points: {-2, 1, 9, 18} using k = 2, Euclidean distance, and the initial cluster centers are c1 = 3 and c2 = 21.

1. What are the initial cluster assignments? (That is, which examples are in cluster c1 and which examples are in cluster c2?)

-> To determine the initial cluster assignments, we need to calculate the distances between each data point and the initial cluster centers (c1 = 3 and c2 = 21) and assign each data point to the cluster with the nearest center. Here's the calculation:

For c1 = 3:

- Distance from -2: |3 - (-2)| = 5

- Distance from 1: |3 - 1| = 2

- Distance from 9: |3 - 9| = 6

- Distance from 18: |3 - 18| = 15

For c2 = 21:

- Distance from -2: |21 - (-2)| = 23

- Distance from 1: |21 - 1| = 20

- Distance from 9: |21 - 9| = 12

- Distance from 18: |21 - 18| = 3

Based on these distances, we can assign the data points to their nearest cluster center:

- Cluster c1: {-2, 1, 9}

- Cluster c2: {18}

So, the initial cluster assignments are:

- Cluster c1: {-2, 1, 9}

- Cluster c2: {18}

(ii) What are the new cluster centers after making the assignments in (i)?

- After making the initial cluster assignments, we need to update the cluster centers by calculating the mean of the data points in each cluster. Here's how we can do it:

For Cluster c1: {-2, 1, 9}

- New cluster center: Mean of {-2, 1, 9} = (-2 + 1 + 9) / 3 = 8 / 3 ≈ 2.67

For Cluster c2: {18}

- New cluster center: Mean of {18} = 18

So, the new cluster centers are:

- c1: 2.67

- c2: 18

1. KNN
2. Consider a set of five training examples given as values, where are the two attribute values (positive integers) and is the binary class label:

{((0, 0), −1), ((3, 7), +1), ((3, 3), +1), ((6, 3), −1), ((2, 5), −1)}.

Classify a test example at coordinates (2, 4) using a k-NN classifier with k = 3 and Euclidean distance. Your answer should be either +1 or -1.

-> To classify the test example at coordinates (2, 4) using a k-NN classifier with k = 3 and Euclidean distance, we need to follow these steps:

1. Calculate the Euclidean distance between the test example and each of the training examples.

2. Select the k nearest neighbors based on the smallest Euclidean distances.

3. Determine the majority class label among the k nearest neighbors.

4. Assign this majority class label to the test example.

Let's calculate the Euclidean distances:

- Distance between (2, 4) and (0, 0):

\( \sqrt{(2 - 0)^2 + (4 - 0)^2} = \sqrt{4 + 16} = \sqrt{20} \)

- Distance between (2, 4) and (3, 7):

\( \sqrt{(2 - 3)^2 + (4 - 7)^2} = \sqrt{1 + 9} = \sqrt{10} \)

- Distance between (2, 4) and (3, 3):

\( \sqrt{(2 - 3)^2 + (4 - 3)^2} = \sqrt{1 + 1} = \sqrt{2} \)

- Distance between (2, 4) and (6, 3):

\( \sqrt{(2 - 6)^2 + (4 - 3)^2} = \sqrt{16 + 1} = \sqrt{17} \)

- Distance between (2, 4) and (2, 5):

\( \sqrt{(2 - 2)^2 + (4 - 5)^2} = \sqrt{0 + 1} = 1 \)

Now, let's select the k = 3 nearest neighbors:

1. (2, 4) → (2, 5) with distance 1

2. (2, 4) → (3, 3) with distance \( \sqrt{2} \)

3. (2, 4) → (0, 0) with distance \( \sqrt{20} \)

Among these three nearest neighbors, we have:

- 1 neighbor with class label +1 (from (2, 5)).

- 2 neighbors with class label -1 (from (3, 3) and (0, 0)).

Since the majority of the nearest neighbors have class label -1, we classify the test example at coordinates (2, 4) as -1.

So, the classification result for the test example at coordinates (2, 4) using a k-NN classifier with k = 3 and Euclidean distance is -1.

1. Given the following code, explain the KNN model being used.

knn = KNeighborsClassifier(n\_neighbors=10, metric='cosine')

-> The code provided initializes a k-Nearest Neighbors (KNN) classifier model using scikit-learn (`KNeighborsClassifier`).

Here's an explanation of the parameters used:

1. `n\_neighbors`: This parameter specifies the number of neighbors to consider when making predictions. In this case, `n\_neighbors=10` means that the KNN algorithm will consider the 10 nearest neighbors to the query point when making predictions.

2. `metric`: This parameter specifies the distance metric used to measure the distance between instances. Here, `'cosine'` is used, which means that the cosine similarity metric is used to calculate the distance between instances. Cosine similarity measures the cosine of the angle between two vectors and is often used for text data or when the magnitude of the vectors is not important.

By setting `n\_neighbors=10` and `metric='cosine'`, the KNN model will classify new data points based on the class labels of the 10 nearest neighbors to the query point, using cosine similarity as the distance metric. This means that the model will find the 10 most similar data points to the query point based on cosine similarity and assign the class label that is most common among those 10 neighbors to the query point.

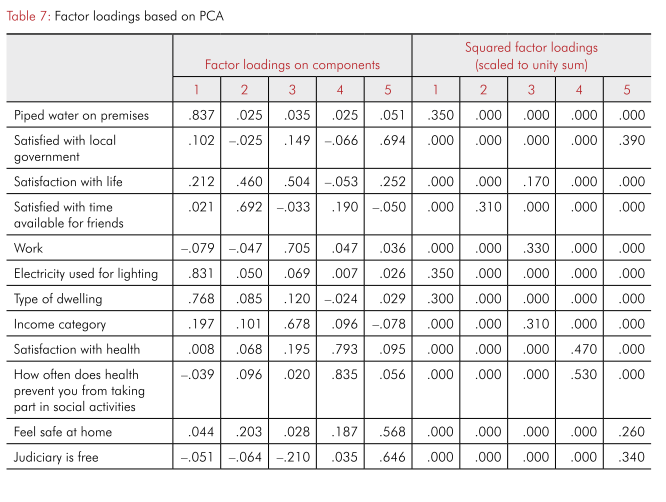
PCA

1. What are the normal ways to choose the number of eigenvalues? Suggest 2 methods.

-> 1. Eigenvalue Criterion: Pick all eigenvalues > 1  
2. Proportion of variance explained criterion: Calculate the % of all eigenvalues in total and cut off with a % that you like.  
3. Scree Plot Criterion (Elbow algorithm again)  
In general we are finding the intersection of the asymptotes of the scree plot.

1. Name the latent variables for PC1 and PC2 in the following table.

Further, If the number of eigenvalues being selected is different, say include PC3, in general and not in this particular case, will it change your latent variables name(s)? Explain within 2 sentences.



1. Association Rule

For the following transaction data, generate rules using Apriori Algorithm.

Given Support = 50% and Confidence = 70%.

|  |  |
| --- | --- |
| Transaction ID | Items Purchased |
| 1 | Book1, Book2, Book3 |
| 2 | Book2, Book3, Book4 |
| 3 | Book2, Book5, Book6 |
| 4 | Book2, Book4, Book6 |
| 5 | Book1, Book3, Book6 |

Step 1 Frequency Table

|  |  |  |
| --- | --- | --- |
| Item | Frequency | Support |
| Book1 | 2 |  |
| Book2 | 4 |  |
| Book3 | 3 |  |
| Book4 | 2 |  |
| Book5 | 1 |  |
| Book6 | 3 |  |

Step 2 Support Check

|  |  |  |
| --- | --- | --- |
| Item | Frequency | Support |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Step 3 Two items candidate set and frequency table.

|  |  |  |
| --- | --- | --- |
| Item | Frequency | Support |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Step 4 Two items Support Check

|  |  |  |
| --- | --- | --- |
| Item | Frequency | Support |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

And if you need to draw Step 5,6,7,8, please use the margin.

Last Step: Generate rules by confidence check.

1. Collaborative Filtering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Book 1 | Book 2 | Book 3 | Book 4 | Book 5 |
| Alvin | 4 | 3 | 4 | 2 |  |
| Bob | 5 |  | 1 |  | 2 |
| Calvin | 3 | 5 | 3 | 3 | 1 |
| Davy | 1 | 1 | 5 | 4 |  |
| Edwin | 2 |  | 2 | 3 | 3 |

The above table shows a dataset of user-item ratings. The rating is in a 1-to-5 scale with 5 represents strongly like the item. Missing value means no rating.

1. By using Cosine Similarity, calculate the similarity between Alvin and Calvin. Conclude your answer by stating if their taste is similar or dissimilar.

You may follow these steps:

Step 1: Calculate the mean score by user

Step 2: Adjust the user score

Step 3: Cosine Similarity

Step 4: Conclusion

1. Simplify the table which only contain Book 1, Book 3 and Book 4 as show below. For simplicity, use the score to do calculation directly without adjusting the user mean. By Item-Item collaborative algorithm, estimate the ratings for Book 4 by Bob.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Book 1 | Book 3 | Book 4 |
| Alvin | 4 | 4 | 2 |
| Bob | 5 | 1 |  |
| Calvin | 3 | 3 | 3 |
| Davy | 1 | 5 | 4 |
| Edwin | 2 | 2 | 3 |

1. How does KNN play the role in item-item collaboratively filtering? Suppose k=1, explain how it affect the formula in part b. You do not need to do any calculation in the part.



* In this case we consider only 1 item instead of 2 as of above and that will be computationally efficient and can provide accurate predictions.