Sabre

ML- The Effective Clustering for Recommendation Engine

Sayan Banerjee

12/20/2019

Clustering for Recommendation

- What is the role of ”Clustering” in recommendation?
  - We have some sort of information about a target user.
  - We have historical information about many users and based on the similar information which is available for the current target user we have divided our whole user base into certain number of groups.
  - Now we can map our current user to a precalculated group by some calculations and recommend products which is/are popular/favorable in that group.
- The question is, we don’t know the reaction of the user on our recommendation because that would be based on our action/recommendation to the user.
- Only information we have at the time of mapping the current user to a precalculated group is, some demographic/session-based information about the user.
- So what we are doing is we are mapping the user to a group where the other users have broadly similar kind of demographic information and hoping that they all will have similar reactions to the similar recommendations.

Effective Clustering for Recommendation

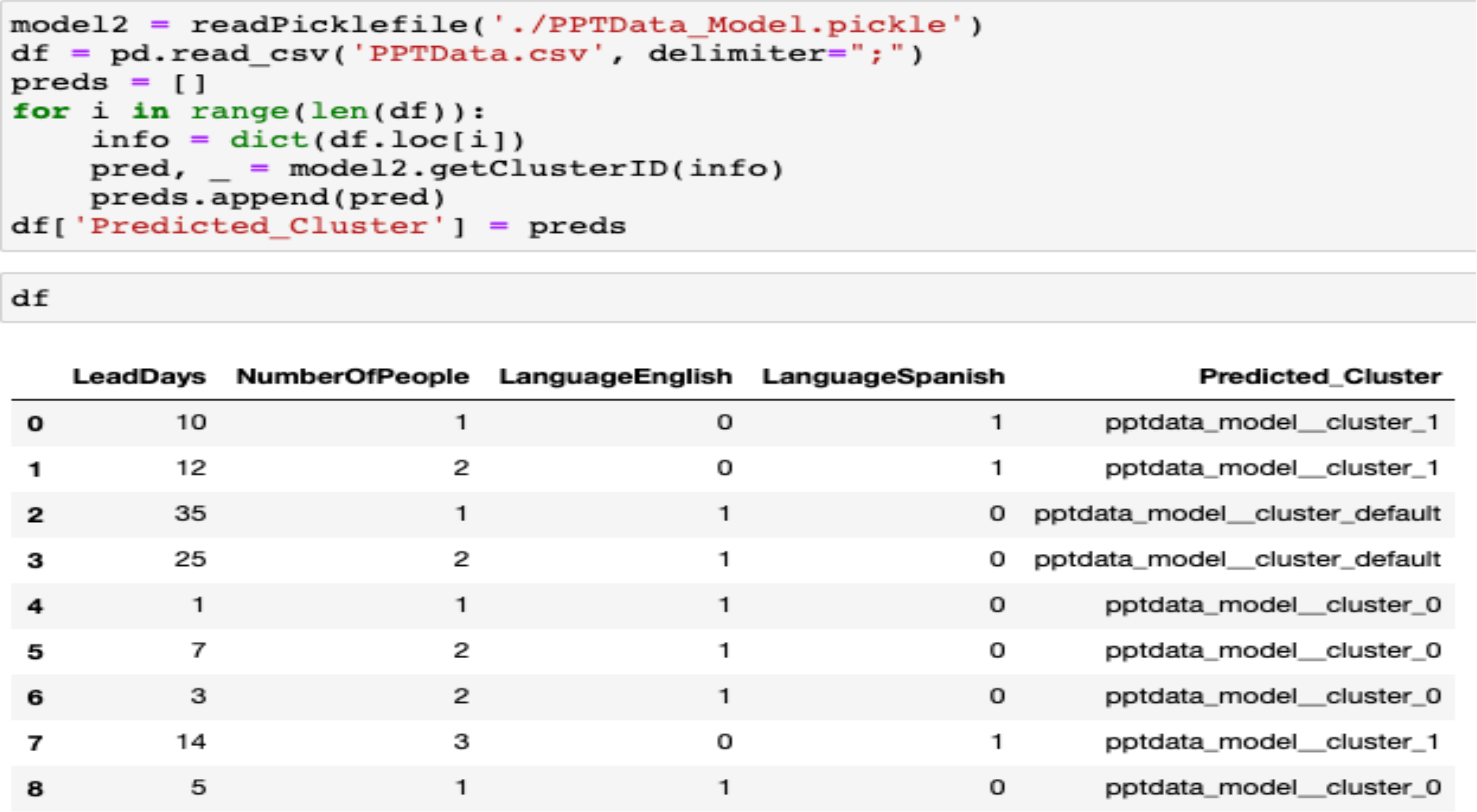
- Can we do better?
- Isn’t it make more sense if we find a way to put users in the same group who tend to have similar reactions to the similar products?
- And also have ability to map the current user to a group where the other users tends to have similar reactions to similar recommendations based on only the available demographic/session-based information about the user.
- Well, now we have ability to do just this, in a scalable, sustainable and adaptable way.
- Let me explain it to you with a toy example.

Toy Example - Data

- Suppose you have 9 datapoints which represents your whole historical user database.
- Historically you have sold 2 products. “Product-A” and ”Product-B”.
- 1 – Positive Reaction to a product, 0 – Negative/No Reaction to a product.
- Blue columns are the information which were/are/will be available from a session and red columns are the reaction of the user to a product in a session.
- I can only afford one cluster which may have less than 3 data points.

Lead Days	Number of People	Language	Product-A	Product-B
10	1	Spanish	0	1
12	2	Spanish	0	1
35	1	English	1	1
25	2	English	1	0
1	1	English	1	1
7	2	English	1	0
3	2	English	1	0
14	3	Spanish	1	0
5	1	English	1	1

Conventional Way – Using Only Blue Columns

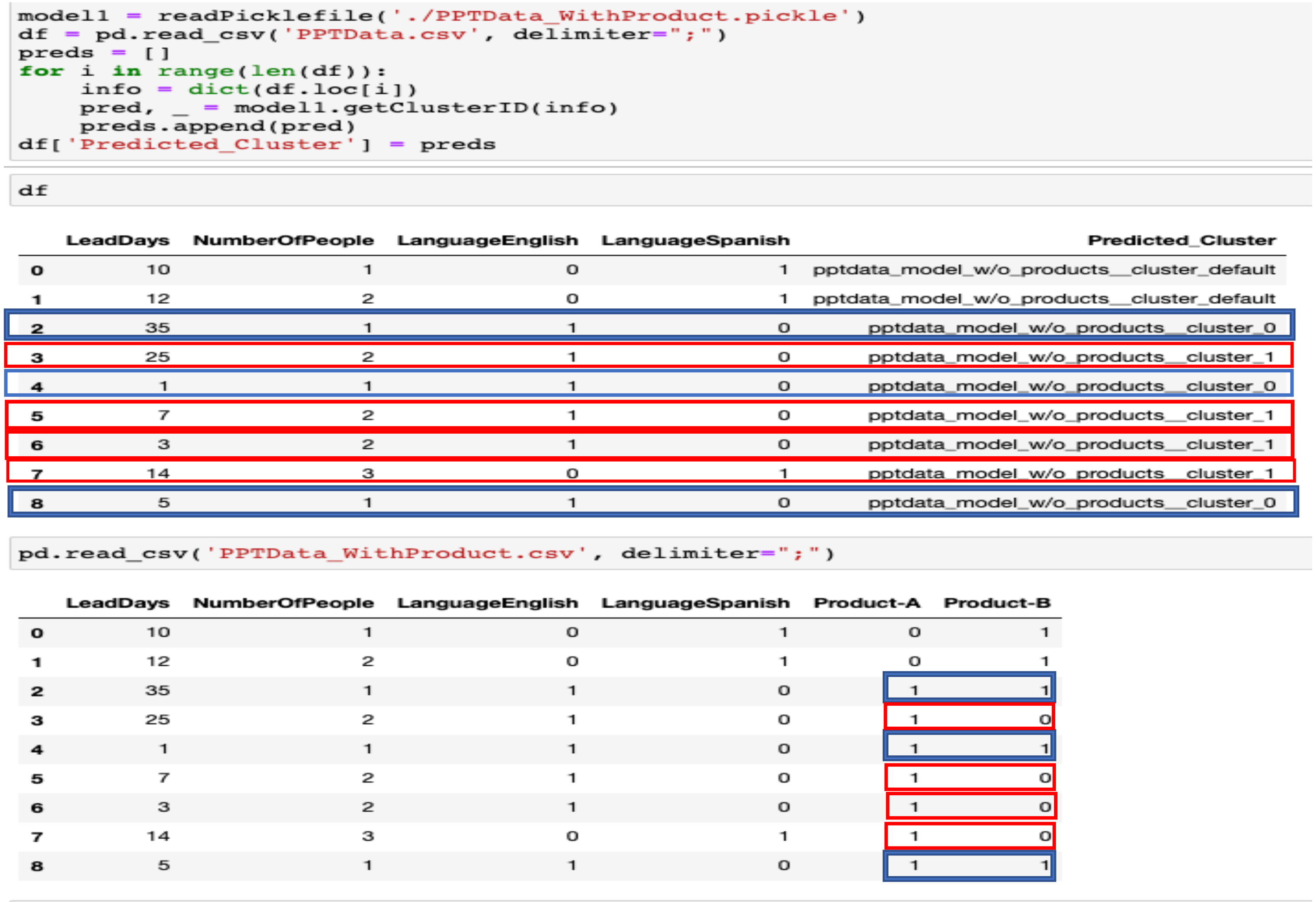


```
model2 = readPicklefile('./PPTData_Model.pickle')
df = pd.read_csv('PPTData.csv', delimiter=";")
preds = []
for i in range(len(df)):
    info = dict(df.loc[i])
    pred, _ = model2.getClusterID(info)
    preds.append(pred)
df['Predicted_Cluster'] = preds
```

```
df
```

	LeadDays	NumberOfPeople	LanguageEnglish	LanguageSpanish	Predicted_Cluster
0	10	1	0	1	pptdata_model__cluster_1
1	12	2	0	1	pptdata_model__cluster_1
2	35	1	1	0	pptdata_model__cluster_default
3	25	2	1	0	pptdata_model__cluster_default
4	1	1	1	0	pptdata_model__cluster_0
5	7	2	1	0	pptdata_model__cluster_0
6	3	2	1	0	pptdata_model__cluster_0
7	14	3	0	1	pptdata_model__cluster_1
8	5	1	1	0	pptdata_model__cluster_0

Effective Way – Using Blue and Red Columns Together at Training, but Only Blue Columns at Inference



```
model1 = readPicklefile('./PPTData_WithProduct.pickle')
df = pd.read_csv('PPTData.csv', delimiter=";")
preds = []
for i in range(len(df)):
    info = dict(df.loc[i])
    pred, _ = model1.getClusterID(info)
    preds.append(pred)
df['Predicted_Cluster'] = preds
```

```
df
```

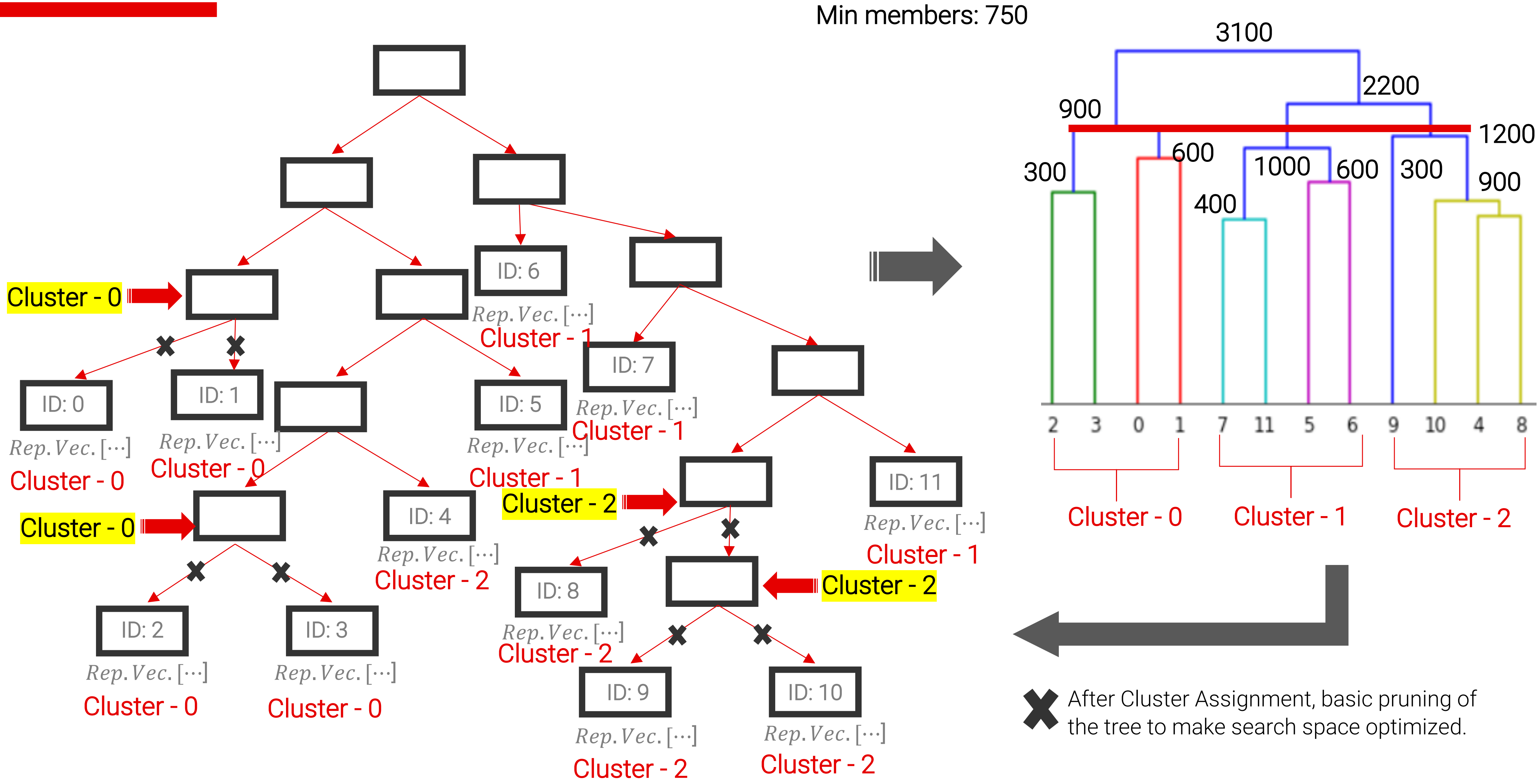
	LeadDays	NumberOfPeople	LanguageEnglish	LanguageSpanish	Predicted_Cluster
0	10	1	0	1	pptdata_model_w/o_products__cluster_default
1	12	2	0	1	pptdata_model_w/o_products__cluster_default
2	35	1	1	0	pptdata_model_w/o_products__cluster_0
3	25	2	1	0	pptdata_model_w/o_products__cluster_1
4	1	1	1	0	pptdata_model_w/o_products__cluster_0
5	7	2	1	0	pptdata_model_w/o_products__cluster_1
6	3	2	1	0	pptdata_model_w/o_products__cluster_1
7	14	3	0	1	pptdata_model_w/o_products__cluster_1
8	5	1	1	0	pptdata_model_w/o_products__cluster_0

```
pd.read_csv('PPTData_WithProduct.csv', delimiter=";")
```

	LeadDays	NumberOfPeople	LanguageEnglish	LanguageSpanish	Product-A	Product-B
0	10	1	0	1	0	1
1	12	2	0	1	0	1
2	35	1	1	0	1	1
3	25	2	1	0	1	0
4	1	1	1	0	1	1
5	7	2	1	0	1	0
6	3	2	1	0	1	0
7	14	3	0	1	1	0
8	5	1	1	0	1	1

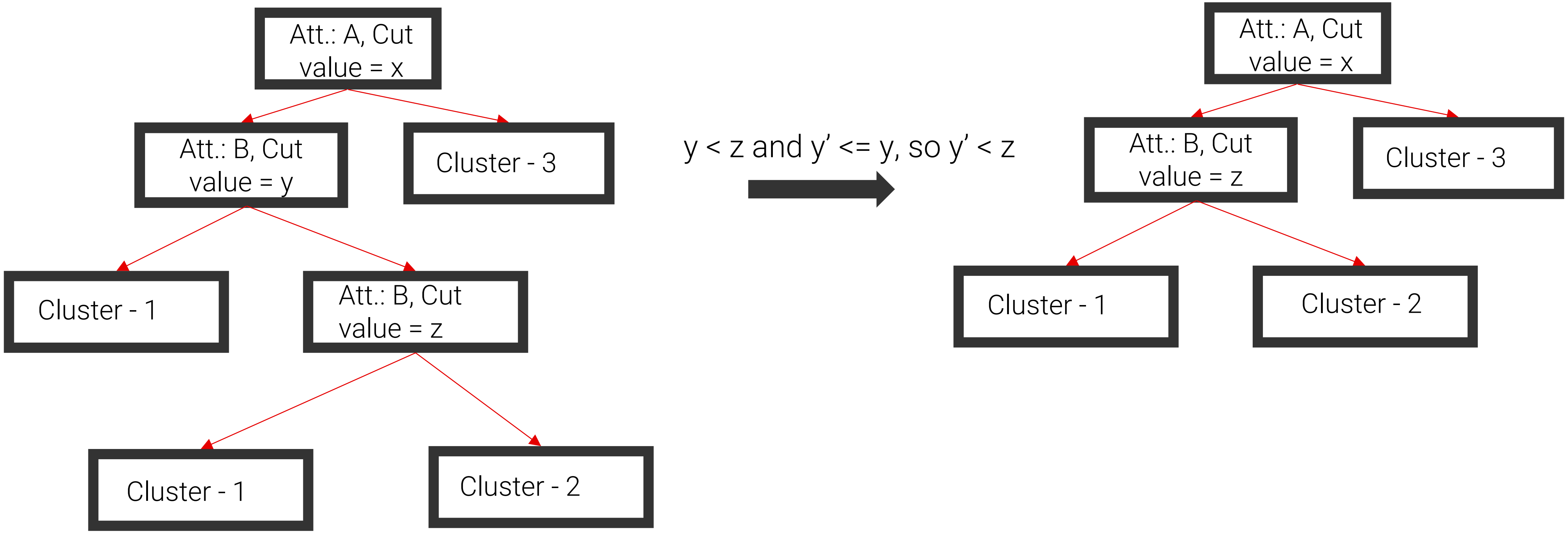
Appendix

Clustering Using CLTree and Hierarchical Clustering

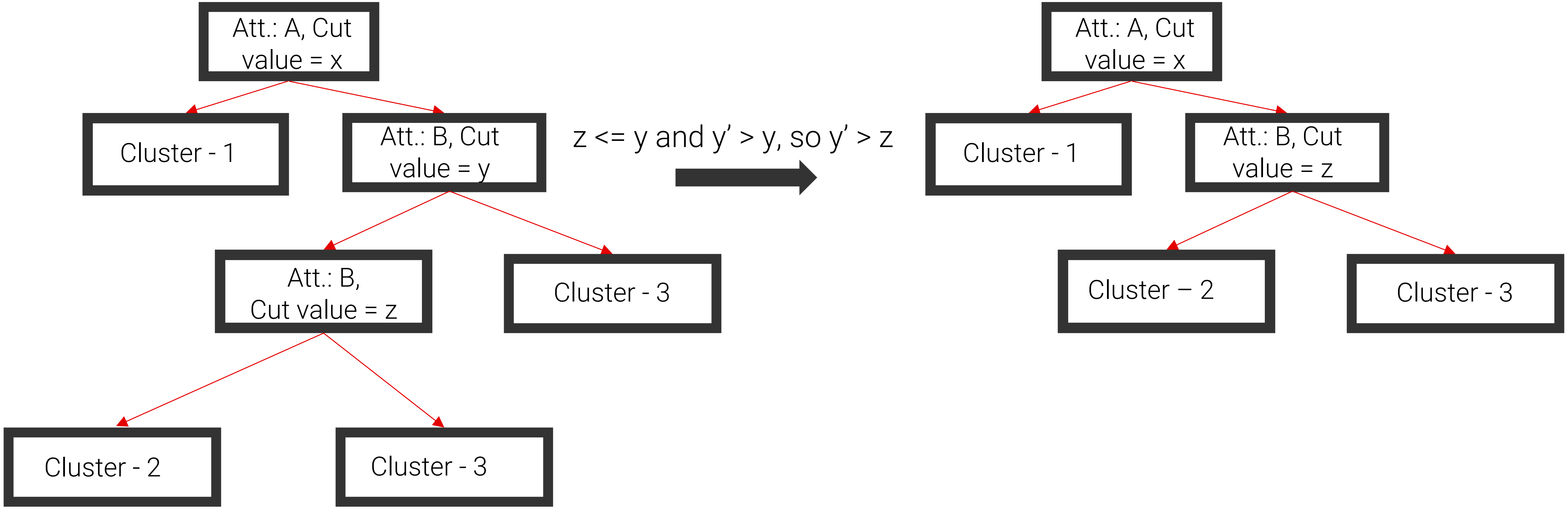


After Cluster Assignment, basic pruning of the tree to make search space optimized.

Search Space Optimization: pruningRedundantNodes() - 1



Search Space Optimization: pruningRedundantNodes() - 2



Sabre®