

# CIDM 6355 Week 8 RapidMiner Lab - Clustering with K-Means Algorithm

## Learning Objectives:

- Understand and apply k-means algorithm to create clusters in RapidMiner
- Interpret the clusters generated by a k-means model and explain their significance, if any
- Visualize clusters
- Evaluate different k-means models.

**Dataset:** Please download the dataset titled "City\_Data.csv" from WT Class to use it with this lab session.

**Requirements:** The same requirements with previous Lab sessions, especially screenshots must be taken with date and time there.

**Attention:** Due to multiple reasons such as random selection of initial centroids, k-means algorithm may generate different results each time. If you find your result is different from the one provided in this instruction, do not worry as long as you follow the instruction correctly.

## Business Understanding

It will be useful to segment metropolitan cities in the United States into groups that are expected to share similarity. In this lab session, we are going to cluster 325 cities into a few groups based on characteristics such as climate, recreation and job opportunity.

## Data Understanding

This dataset contains various data on 325 metropolitan cities in the United States. Each metropolitan area is scored in each of the following five categories (attributes):

- Living Cost
- Jobs
- Climate
- Health Care
- Recreation

The score in each category ranges from 0 to 100 points (where 100 is most favorable and 0 is least favorable in the given category). Note that the scores in each category are normalized such that the 50th percentile point is close to the average for all metropolitan areas (Note: normalization is very important for k-means clustering, which will be covered in Week 9). As a data scientist, you are invited to cluster 325 cities into no more than 10 groups based on the nine attributes above.

## Data Preparation

1. Import your data to RapidMiner using the Read CSV operator and run it to check if any missing values and/or outliers in this dataset by examining the Statistics.

**Deliverable 1:** Please write down the average (i.e., grand average) for all the five attributes (round them the third decimal place). All these numbers below are the centroid for all 325 cities (i.e., population centroid).

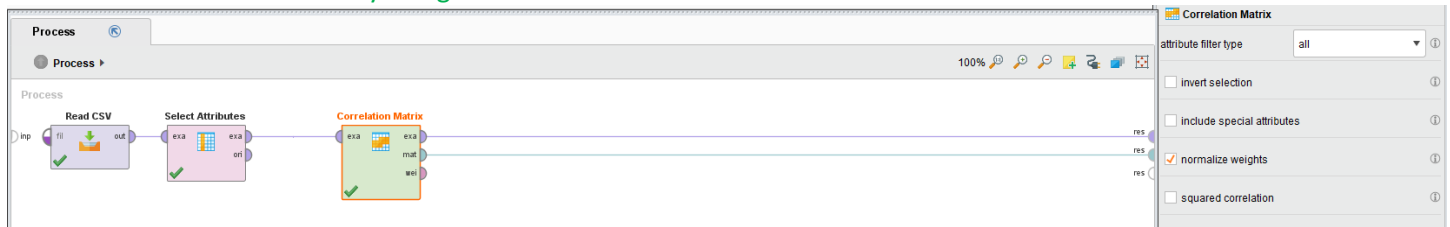
Attributes	Cost_living	Jobs	Climate	Health_Care	Recreation
Average					

2. Add the Select Attributes operator to unselect the polynomial attribute [Note: because the first column is a

polynomial attribute, it should be excluded in the Correlation Matrix or the K-Means model]

3. Add the Correlation Matrix operator to check if any highly-correlated attributes (absolute value greater than 0.85).

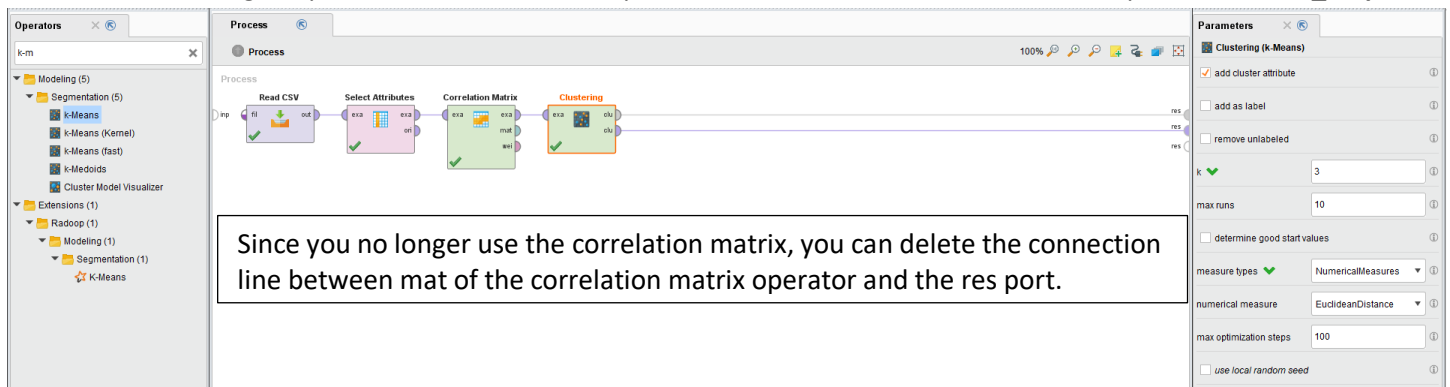
**Questions to think about:** What potential problems do we have if we cluster with highly-correlated attributes? What solutions do we have? You may Google the answers.



## Modeling

4. Modeling using K-Means Algorithm

4.1 This step is straightforward and simply requires you to build the process shown in the screenshot below. Find the K-Means operator under Modeling → Segmentation or simply type K-Means in the search box. Also notice that the configuration of the "Clustering" operator -- with **k being 3** and distance measure being the **Euclidean distance**. After running the process, observe the example set and cluster model results. Save the process as **Lab8\_Step4.1**.



- 4.2 Under the ExampleSet results, notice that there exists a new attribute "Cluster" with each observation assigned to either cluster\_0, cluster\_1 or cluster\_2.

Result History

ExampleSet (Clustering)

Cluster Model (Clustering)

Data

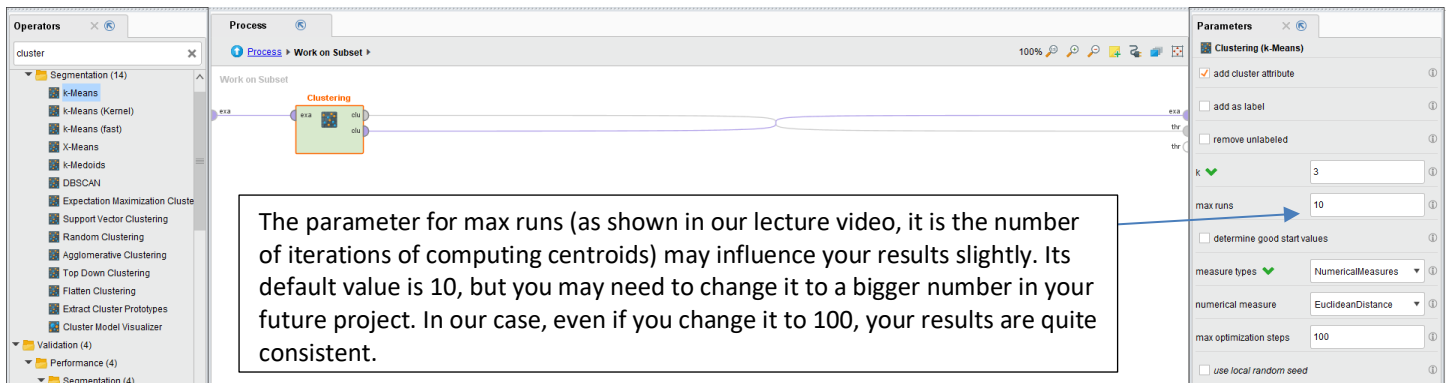
ExampleSet (325 examples, 2 special attributes, 5 regular attributes)

Row No.	id	cluster	Cost_Living	Jobs	Climate	Health_Care	Recreation
1	1	cluster_0	54.680	74.780	75.920	91.500	100
2	2	cluster_0	21.250	75.070	16.430	84.700	99.710

- 4.3 However, there is a problem here: the example set does not include city names. Even though we can use the Row No. or id to identify city name, it is not convenient. Now, we are going to use the operator Work on Subset to get the city name back to the example set. The Work on Subset operator is a nested operator that can filter certain attributes at the subset level, while keeping all the attributes at the higher level. The screenshot below shows you the parameters for this operator (the parameters are similar to the operator Select Attributes)



4.4 When you double click the operator Work on Subset, you can see subprocess, in which you can add the K-Means operator with the same parameters as in Step 4.1. **Note: Pay attention to the link of input and output ports. Their types should be matched. In this case, the second “clu” of the K-Means Clustering should be linked to “exa”.**



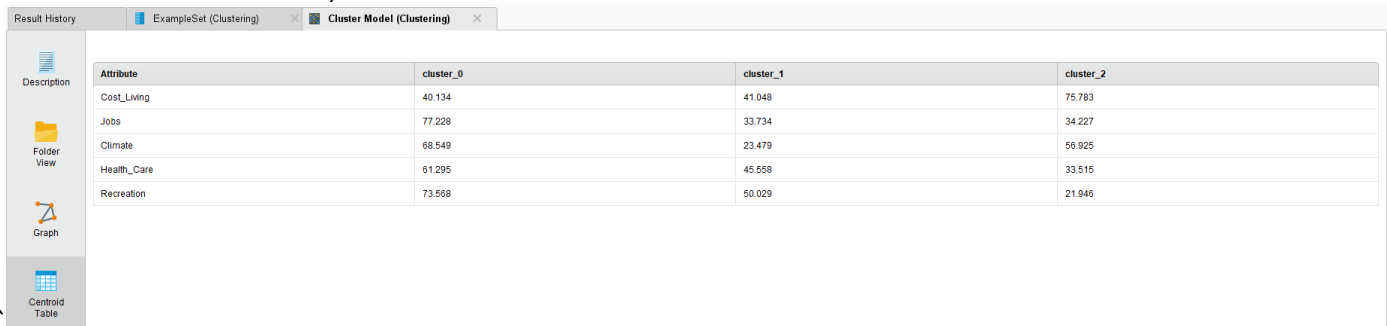
The parameter for max runs (as shown in our lecture video, it is the number of iterations of computing centroids) may influence your results slightly. Its default value is 10, but you may need to change it to a bigger number in your future project. In our case, even if you change it to 100, your results are quite consistent.

4.5 Save the process as **Lab8\_Step4.5**. Run this process and you will find that the example set includes the city name in the last column. Observe the results.

**Deliverable 2: Take a screenshot of your ExampleSet with date and time (Screenshot 1).**

4.6 Under the Cluster Model results, in the Description view, you can see the numbers of individuals in each clusters.

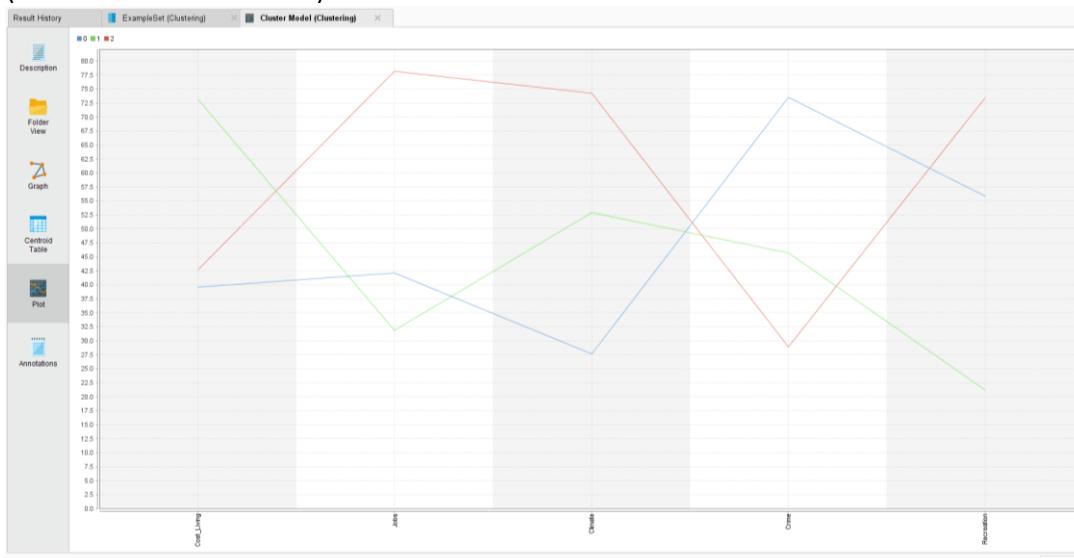
4.7 Under the Cluster Model results, in the Centroid Table view, you can see the centroid values for each cluster (see the screenshot below).



Attribute	cluster_0	cluster_1	cluster_2
Cost_Living	40.134	41.048	75.783
Jobs	77.228	33.734	34.227
Climate	68.549	23.479	56.925
Health_Care	61.295	45.558	33.515
Recreation	73.568	50.029	21.946

Note: the centroid value on each attribute for each cluster is the average of all cities in this cluster on the same attribute. For example, the centroid value of living cost in cluster\_0 is 40.134, which is average score in the living cost of all 128 cities in cluster\_0.

4.8 Under the Cluster Model results, in the Plot view, you can see a line chart to present the centroid values for each cluster (see the screenshot below).



Deliverable 3: based on the results in 4.5-4.8, please clearly describe each cluster, including cluster size, example cities, at least three out of five dimensions/attributes, and an appropriate name. For example, Cluster 0 includes 128 cities such as New Orleans, LA and Long Island, NY have highest scores in job opportunities, climate, healthcare, and recreation. However, this group of cities have quite high living cost. We can name this group of cities ..... [Even though the description or name is quite subjective, please use the centroid table to justify your description]

## Evaluation

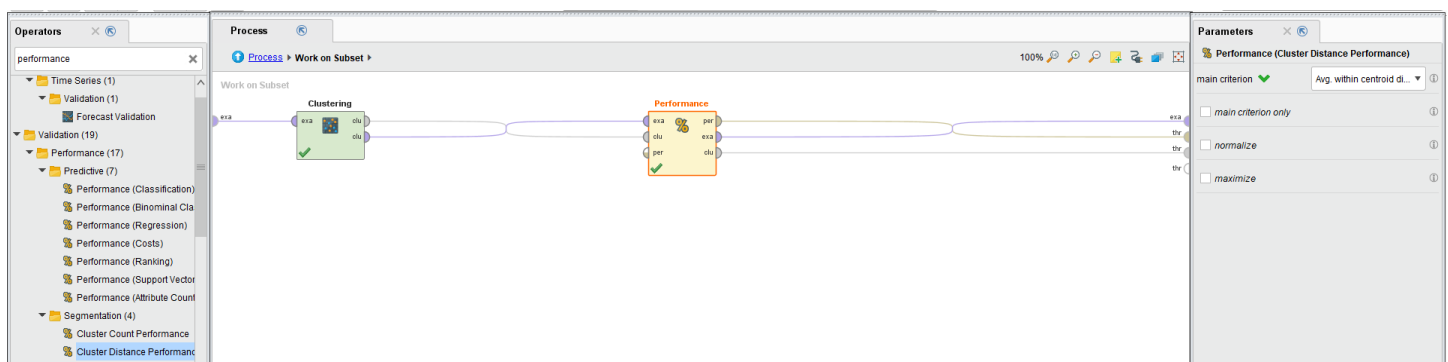
### 5. Evaluate your DM model

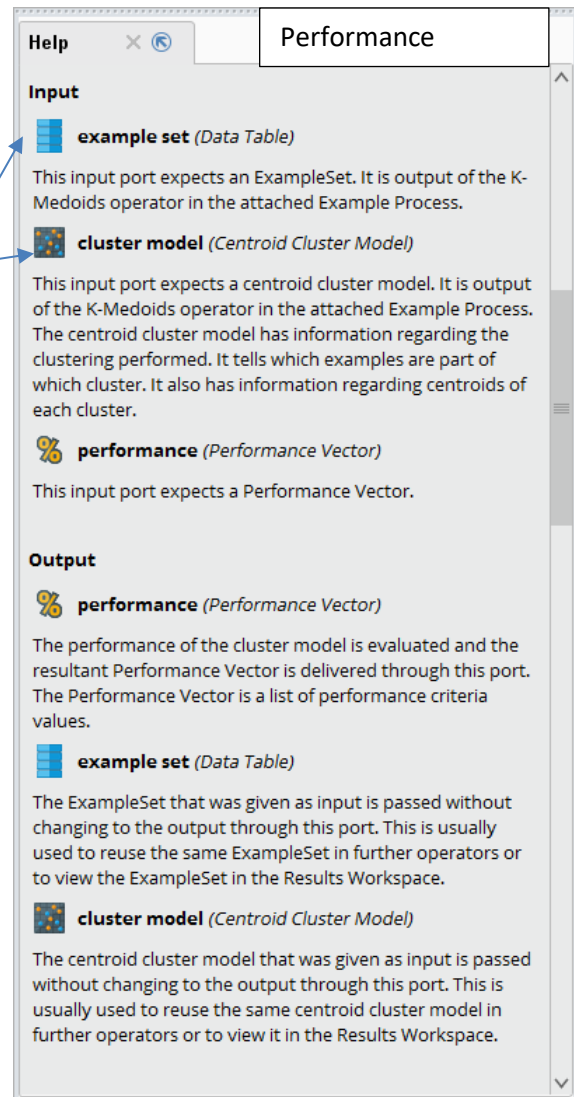
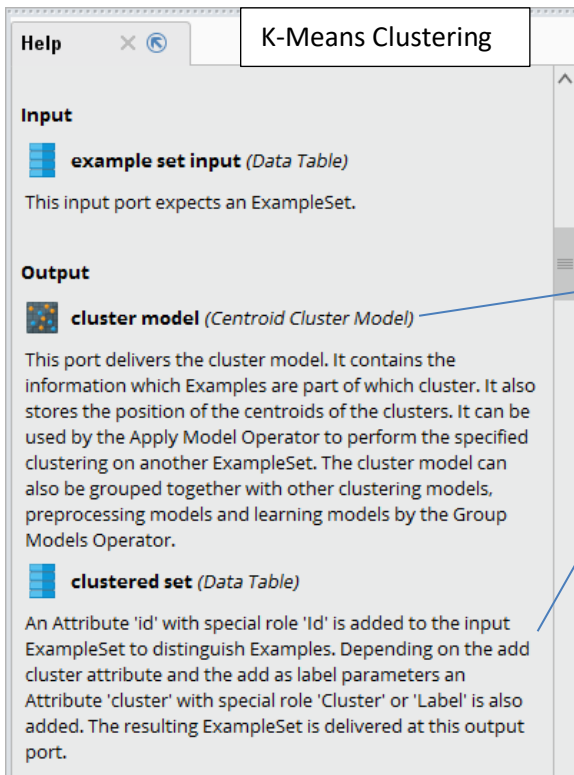
In this step, you will play with the number of clusters and decide how many clusters should be chosen for this case. Add a “Cluster Distance Performance” operator to your process in Step 1. The process is show below. RapidMiner does not have SSE performance operator. But the distance performance is very similar to SSE.

There are many measures used to evaluate your clustering model. For details, please read [this chapter](#). Here, we are going to use average within centroid distance and Davies-Bouldin index (DBI). For details, please read this [documentation](#). Basically, average within centroid distance is the average within cluster distance is calculated by averaging the distance between the centroid and all examples of a cluster. The algorithms that produce clusters with low intra-cluster distances (high intra-cluster similarity) and high inter-cluster distances (low inter-cluster similarity) will have a low Davies-Bouldin index. The clustering algorithm that produces a collection of clusters with the smallest Davies-Bouldin index is usually considered the best algorithm based on this criterion. However, the two measures may conflict because the average within centroid distance only considers intra-cluster distance, but Davies-Bouldin index considers both intra-cluster and inter-cluster distances.

5.1 Find the Cluster Distance Performance operator under Validation → Performance → Segmentation or simply type performance in the search box.

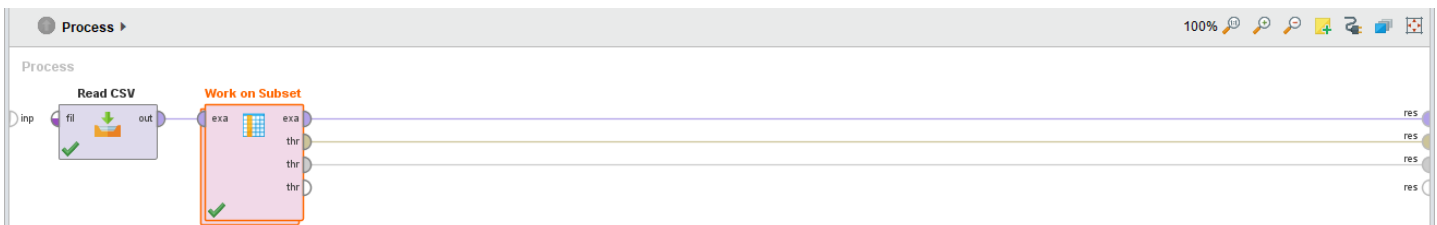
5.2 Add the Cluster Distance Performance operator into the subprocess of Work on Subset. **Note: Pay attention to the link of input and output ports to make them matched.**





Note: you can see the detail about the two operators in the Help file.

5.3 Add one more connection between the thr port of Work on Subset operator and the res port. Doing so, we are going to have three results. Save the process as **Lab8\_Step5.3** and run it.



5.4 In the Performance Vector, you can find the overall average within centroid distance and average within centroid distance for each cluster. In addition, you can also find Davies Bouldin. All of them are negative in RM, even though they are mathematically positive.

Note: Both average within centroid distance and Davies–Bouldin index are mathematically positive. However, in RapidMiner, they are shown as negative numbers when you use the Performance operator because they are multiplied by -1 to allow using them for optimization. The reason for multiplying by -1: The Performance (Cluster Distance Performance) calculates the average distance within centroids. The smaller the distances are the better the clustering

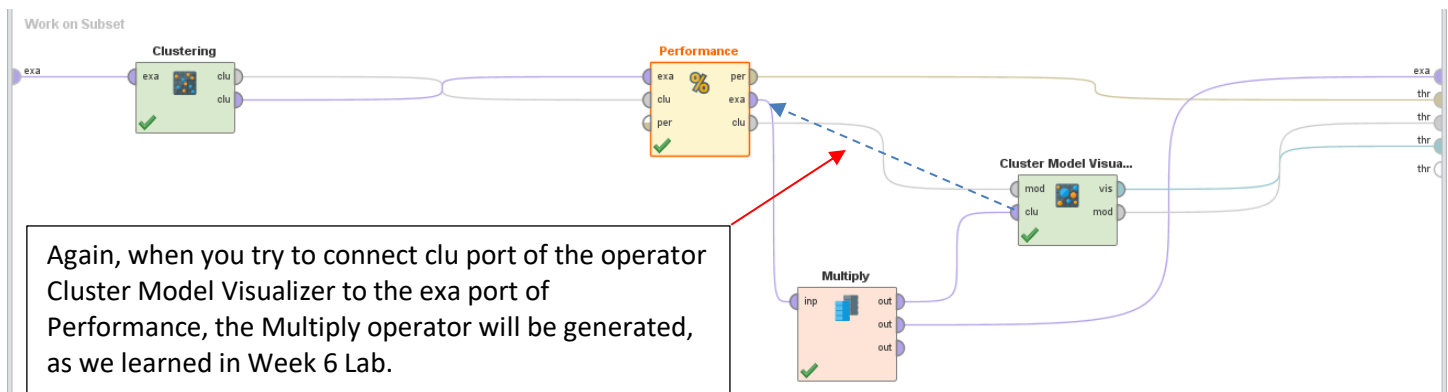
works (in theory). But our optimization operators always try to maximize the performance of an algorithm. This means if you don't multiply by -1, the optimization algorithm would always prefer cluster results with a higher average within centroid distance. This is the same reason why Davies–Bouldin index is negative in RapidMiner. However, they are displayed as positive numbers when you use Cluster Model Visualizer (**Attention: In RM 9.9, Davies–Bouldin index and Average Cluster Distance are no longer shown when using this operator**).

## 5.5 Visualize Clustering Models

The operator Cluster Model Visualizer uses visualization tools for centroid-based cluster models to capture the essential details of each cluster. According to RM documentation, this operator includes the following visualization tools:

- Overview: shows the size of all found clusters, together with some information about the clusters and their quality.
- Heat map: displays a decision tree describing the main difference between the clusters.
- Centroid Chart: shows the values for the cluster centroids in a parallel chart.
- Centroid table: shows the values for the cluster centroids in a table.
- Scatter plot: with a choice of cluster, displays a scatter plot in terms of the two most important Attributes.

5.5.1 We add the operator Cluster Model Visualizer into the operator Work on Subset and then make the following connections.



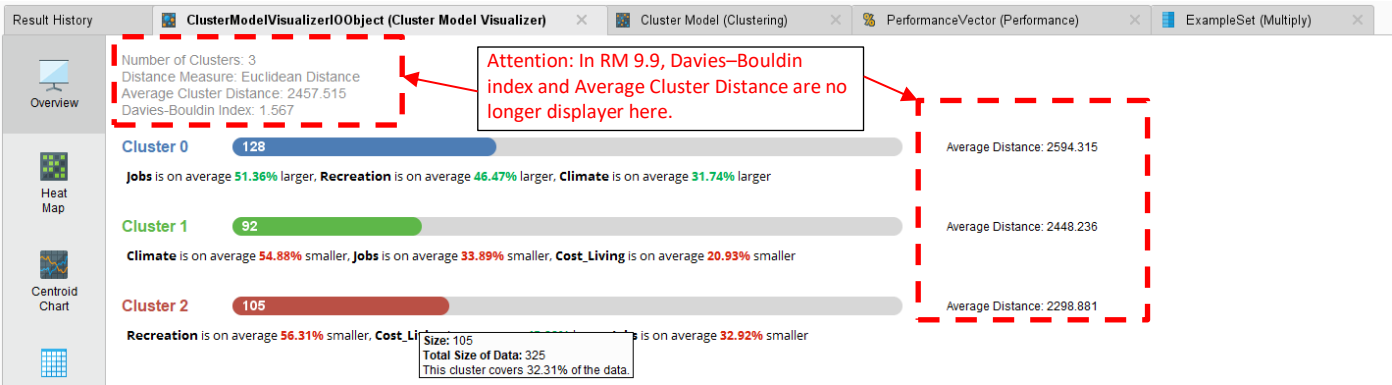
5.5.2 Because we have four outputs in the Work on Subset, we need to add an additional connection for the operator Work on Subset.



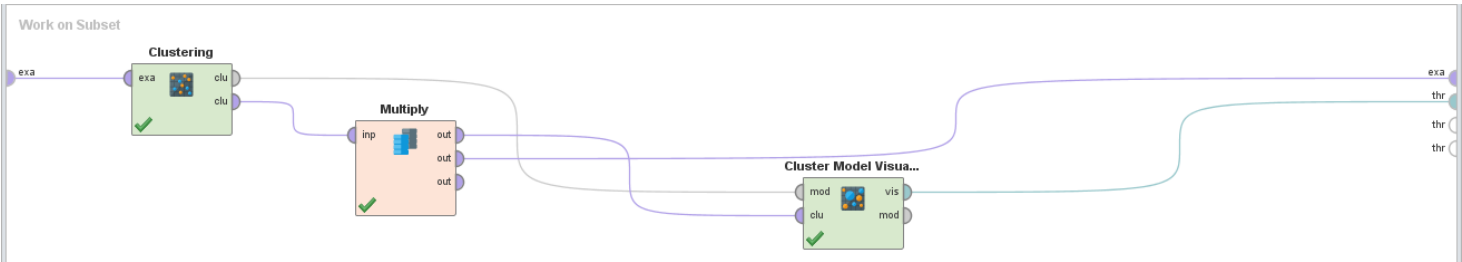
5.5.3 Save this process as **Lab8\_Step 5.5** and then run it. You will see four results (see the screenshot as below).

5.5.4 On the top of the Overview view, you can find that both average cluster distance and Davis-Bouldin Index are positive now, which is different from results generated by the Performance Operator. Then, you can see the

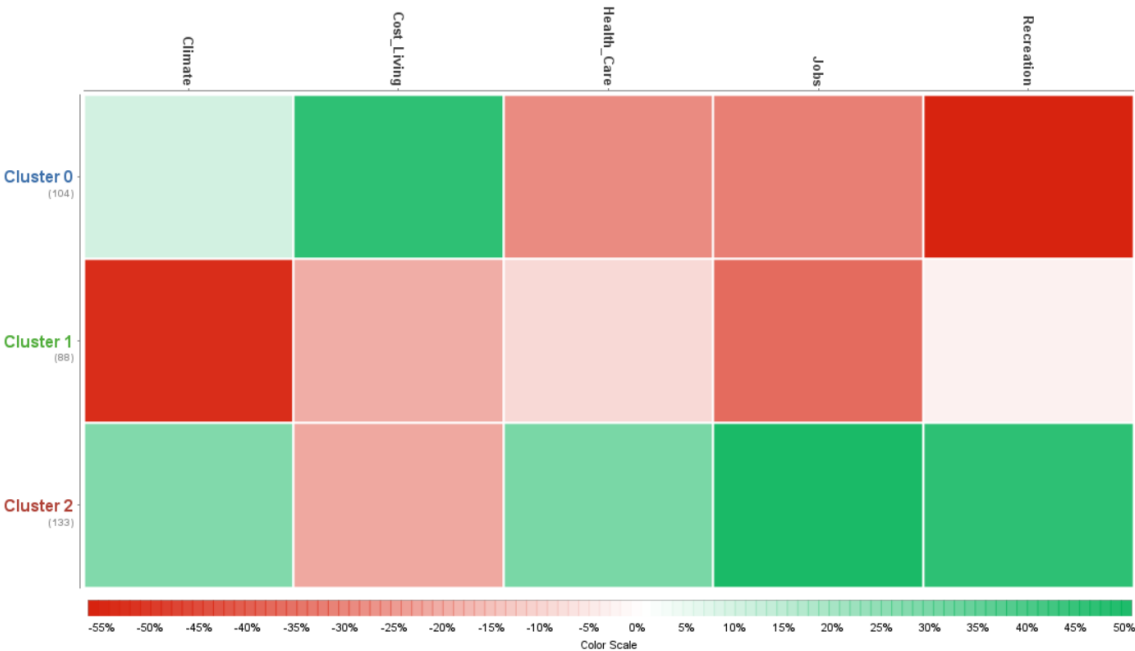
size of each cluster, their significant attributes, and average distance (positive again). For example, Cluster 0 may be a group of “ideal” cities because their scores in job opportunity, recreation, and climate are higher than the average.



5.5.5 You may realize that the Cluster Model Visualizer tab covers all the information generated in Cluster Model and Performance. If you want to simplify your process, you can drop the Performance operator. However, when you are asked to obtain average centroid distance and Davis-Bouldin index, you have to use the results generated by the Performance operator.



5.5.6 The heat map further highlights the difference among the three clusters. When you move cursor to any piece, you can find the comparison between the cluster centroid and the overall centroid (see the table in Step 1).



A red color indicates that the cluster average (cluster centroid) is smaller than grand average (population centroid): the darker, the much smaller.

A green color indicates that the cluster average (cluster centroid) is greater than grand average (population centroid): the darker, the much greater.

**Question to think about:** Based on the heat map above, among all the five attributes, which one is the most important differentiator? Which one is the least important differentiator?

- Recreation is the most important differentiator because it has three very distinct colors across the three clusters in the heat map: dark red in Cluster 0, almost white in Cluster 1 (indicating that the cluster centroid is almost the same with the population centroid at this dimension), and dark green in Cluster 2.
- Health\_Care is the least important differentiator because it has three very light colors across the three clusters in the heat map, which means that the cluster centroid is slightly different from the population centroid at this dimension and Health\_Care cannot effectively distinguish the three cluster centroids with the population centroid.

Note: sometimes the least important differentiator attribute is not listed in the head map.

### 6. Compare multiple clustering models

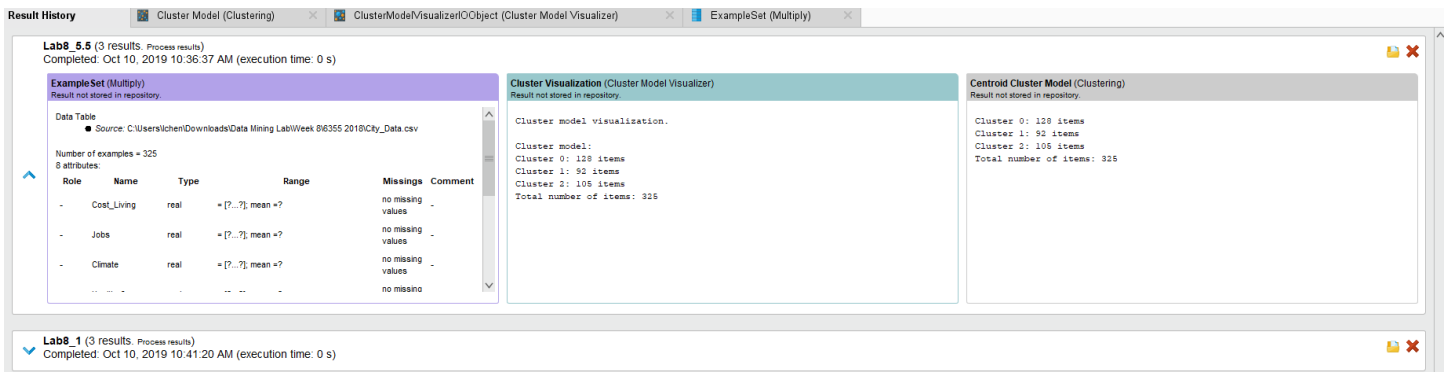
6.1 Now, please change the k from 3 to 4 (after you change the number of clusters, you’d better to press **Enter** on your keyboard to make it really works). Take a look at Overview, Heat Map, and Centroid Table under the Cluster Model Visualizer for the 4-cluster model (see the example in Step 5.5.4, 5.5.6, and 5.5.7).

6.2 Change the k to 2-15 respectively (after you change the number of clusters, you’d better to press **Enter** on your keyboard to make it really works) while keeping other parameters the same. For each k value, please write down its Avg. within centroid distance and Davies Bouldin index in the following table (round your answer to the third decimal place). You do not need to submit this table, but you need to use it to answer the following questions. In order to be consistent, please use positive numbers. **When they are converted as positive numbers, the smaller, the better.**

	Avg. within centroid distance	Davies Bouldin Index
K=2		
K=3		
K=4		
K=5		
K=6		
K=7		
K=8		
K=9		
K=10		
K=11		
K=12		
K=13		
K=14		
K=15		

**Deliverable 4:** Take a screenshot of your Result History page with date and time (see an example as below) (Screenshot 2).





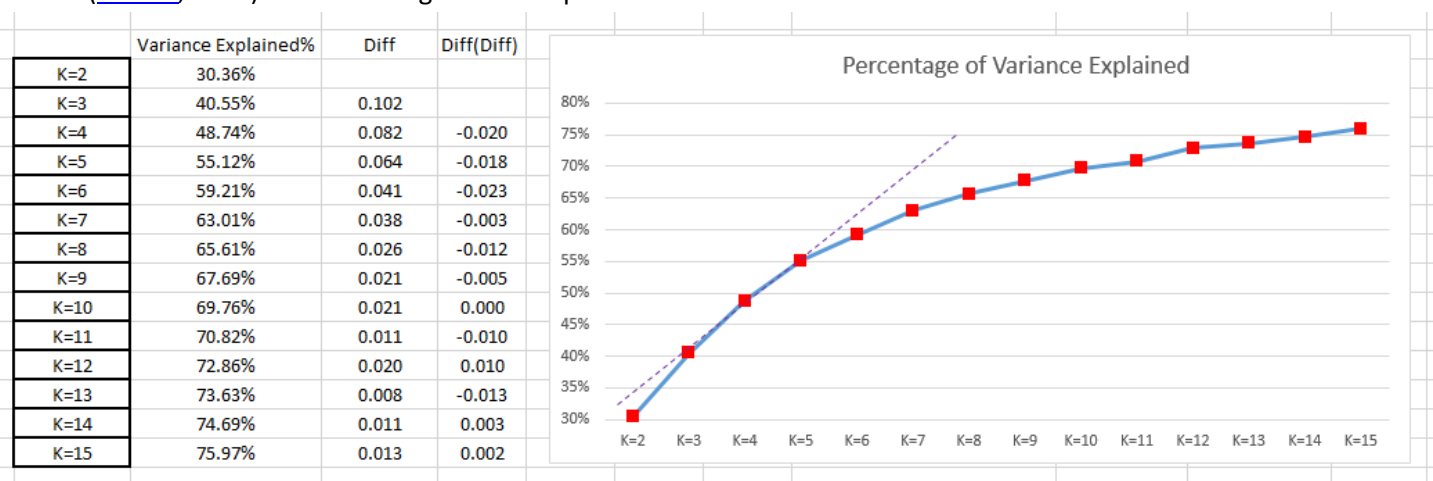
**Deliverable 5:** Based on the table above, when k increases, what happen to Avg. within centroid distance (increasing or decreasing)? What about Davies Bouldin Index? Imagine an extreme case, when k=325, what would Avg. within centroid distance be? What potential problem will we encounter if we only use Avg. within centroid distance as the main criterion for evaluating clustering models? Please answer all the question in this deliverable.

6.3 Deep Understanding: According to RapidMiner, the average within cluster distance is calculated by averaging the distance between the centroid and all examples of a cluster. However, “distance” in this definition is not Euclidean distance (even we chose Euclidean distance in the parameter), but **squared** Euclidean distance or a cluster’s average SSE (Sum of Squared Error): SSE divided by the cluster size. Please refer to the Excel Sheet Distance Calculation, which uses k=3 as an example to illustrate how average within centroid distance (all are positive in the Excel), SSE, SST, and is calculated. Basically, Avg. within centroid distance\*cluster size=SSE.

## 7. Further Thinking and Reading

### 7.1 Elbow Method

How to evaluate a clustering model or find a best k is a challenging question, especially when your data is not very clustered. We need to consider both logical (does it make sense) and mathematical issues (can it be supported by statistical criteria). Even mathematical consideration only involves many indices. One useful method is elbow method. “One should choose a number of clusters so that adding another cluster doesn’t give much better modeling of the data. More precisely, if one plots the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters is chosen at this point, hence the ‘elbow criterion’” ([Source](#), 2017). The following is an example in our case.



Sometimes, if the data is not very clustered, the elbow chart does not have a clear elbow. Even though our data is not very clustered, we still find that a good k could be one between 3 and 5.

Of course, we can also use average within centroid distance and/or DBI as the index instead of percentage of variance explained (see [Sum of Squared Errors or average within centroid distance](#) or [DBI](#) for more details). Replace the y-axis by average within centroid distance and/or DBI you get in the table under Step 6.2.

**Deliverable 6: Draw an elbow chart using average within centroid distance or DBI for k=2-15. Take a screenshot of your elbow chart with date and time (Screenshot 3). Observe your elbow chart and then discuss which k is the best and why.**

## 7.2 Some other performance measures in RM

As shown in the right screenshot, RM includes four operators for segmentation (cluster) performance. What we used so far is the second cluster distance performance. In addition to the distance measure, we also have density and item distribution measures.

### 7.2.1 Cluster Count Performance

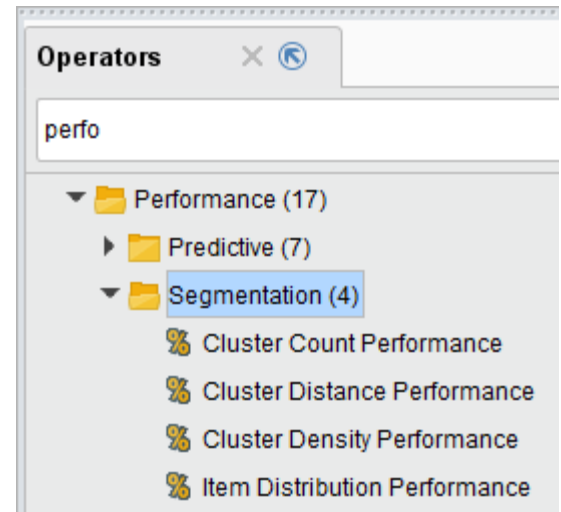
This is a very simple operator. It takes a cluster model as input and returns a performance vector that has the 'Number of clusters' and 'Cluster Number Index' criteria. The 'Number of clusters' criteria contains the number of clusters. The 'Cluster Number Index' criteria builds a derived index from the number of clusters by using the formula  $1 - (k / n)$  with k as the number of clusters and n as the number of examples. This can be used for optimizing the coverage of a cluster result with respect to the number of clusters.

### 7.2.2 Cluster Density Performance

The centroid cluster model has information regarding the clustering performed. It tells which examples are parts of which cluster. It also has information regarding centroids of each cluster. The Cluster Density Performance operator takes this centroid cluster model and clustered set as input and evaluates the performance of the model based on the cluster densities. It is important to note that this operator also requires a Similarity Measure object as input. This operator is used for evaluation of non-hierarchical cluster models based on the average within cluster similarity/distance. It is computed by averaging all similarities / distances between each pair of examples of a cluster. RM uses Avg. within cluster distance (again, a positive number which is displayed as a negative number in this operator for optimization), which is the average of squared difference between any two data points in the same cluster. The smaller distance, the greater density.

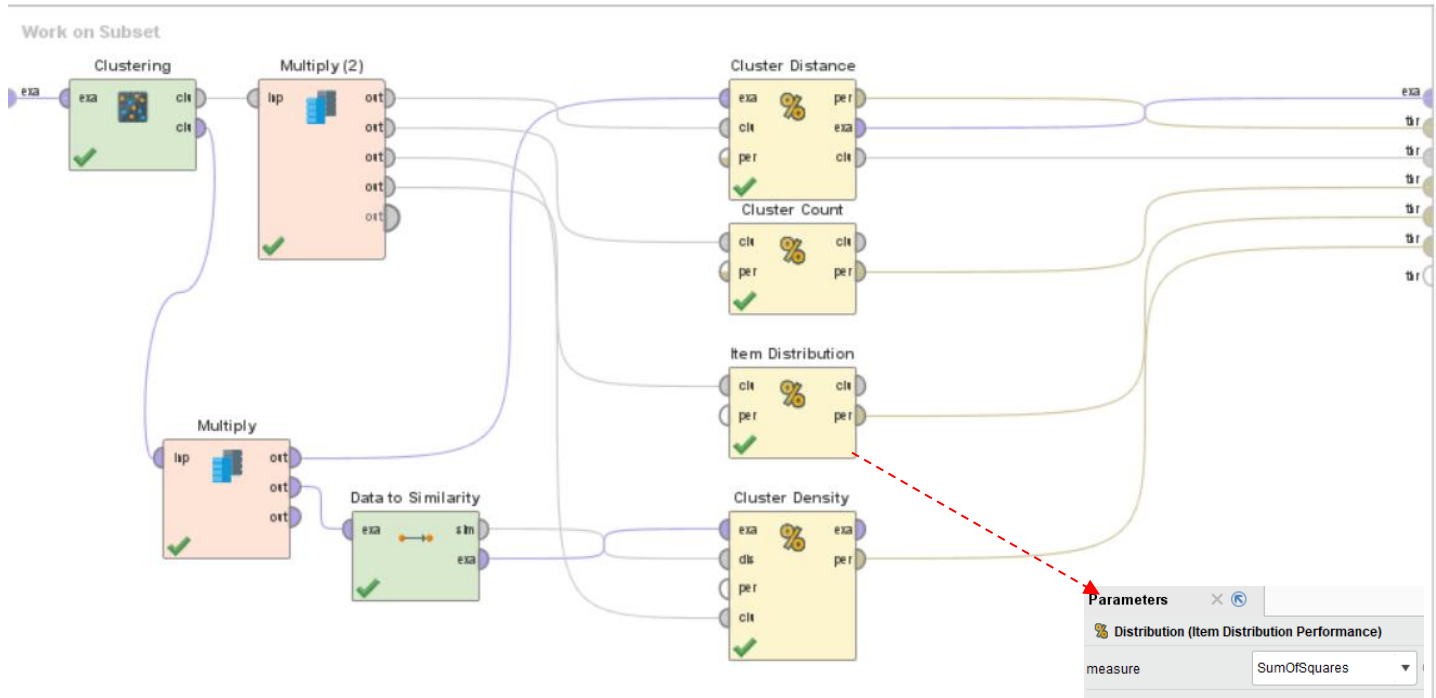
### 7.2.3 Item Distribution Performance

Usually, when we cluster items into multiple groups, we hope each group has similar size, rather than one group including 90% of items and many groups with a single item. The Item Distribution Performance operator takes this cluster model as input and evaluates the performance of the model based on the distribution of examples i.e. how well the examples are distributed over the clusters. Two distribution measures are supported: **Sum of Squares** and **Gini Coefficient**. These distribution measures are explained in the parameters. Flat clustering creates a flat set of clusters without any explicit structure that would relate clusters to each other. Hierarchical clustering, on the other hand, creates a hierarchy of clusters. This operator can only be applied on models produced by operators that produce flat cluster models e.g., K-Means or K-Medoids operators. It **cannot** be applied on models created by the operators that produce a hierarchy of clusters e.g., the Agglomerative Clustering operator (to be discussed in the next class).



You can add all the four operators into your process (see the screenshot as below) to obtain all the performance measures, which will help you make a more comprehensive decision when selecting the best k. Please change the

four performance operators; otherwise you may confused which is which.



**Deliverable 7:** Use k=3 to include all the four performance operators in your process. Take a screenshot of the description view of Cluster Density Performance and Item Distribution Performance with date and time (two screenshots in total: Screenshot 4 and Screenshot 5) and then briefly discuss each result.

Note: Item Distribution Performance, when you chose the sum of square, you got 0.340, which is computed as below, similar to the formula used to compute Gini Index in Week 4.

Cluster size	Distribution	Square	Sum of squares
128	$128/325=0.394$	$0.394*0.394=0.155$	$0.155+0.104+0.080=0.340$
105	$105/325=0.323$	$0.323*0.323=0.104$	
92	$92/325=0.283$	$0.283*0.283=0.080$	

**Additional Tip:** The operator write CSV or write Excel is helpful for you to write ExampleSet in RM into your local computer.

