k-Medoids Clustering

k-Medoids clustering is a useful alternative to the popular k-Means clustering algorithm. Like k-Means, it groups items into k distinct clusters so that items within the same cluster are more similar to each other than items within different clusters. k-Medoids clustering has the advantage over k-Means in that it chooses a prototypical item for each cluster, rather than computing a theoretical mean for each cluster.

k-Medoids clustering is particularly useful when there is a need to understand the nature of each cluster by identifying its prototypical member. Please keep in mind the following best practices for cluster analysis:

1) Clustering works best when items are clustered ***by their nature*** – similar to how you might find books, movies or music grouped together in the public library or a well-organized store. Using an item's characteristics will result in clusters being distinguished by the most discriminating features of the items they contain.
2) The Cluster model is really ***just the beginning*** of the analytical process: It's up to the model creator to study the resulting model and understand which characteristics differentiate one cluster from another.
3) Once the cluster model is built, then business measures, such as Revenue per Customer or Transaction Frequency, can be applied. In this way, the relationship between the intrinsic nature defined by the clusters and key performance indicators can be established.

For more detailed information on k-Medoids clustering, also known as PAM (Partitioning Around Medoids) clustering, please consult the k-Mediods Wikipedia page [3].

This R Script has two functional modes:

- The user specifies the exact number of clusters.
- The user specifies a maximum number of clusters, without specifying an exact number, then the model optimally clusters the data in such a way that the Average Silhouette Width [4] is maximized.

How to Deploy to MicroStrategy:

Prerequisite: Please follow the instructions in the R Integration Pack User Guide [1] for configuring your MicroStrategy environment with R and that the R Script functions have been installed in your MicroStrategy project(s).

1) Download the kMedoidsClustering.R file from the Off-The-Shelf R Script Repository. [2].
2) From the R console, run the kMedoidsClustering.R script to verify the script runs correctly. For details, see the "**Running from the R Console**" section below.
3) Cut-and-paste the appropriate metric expression below in any MicroStrategy metric editor. Map the arguments to the appropriate MicroStrategy metrics. A description of each output is shown below.
4) Use the new metric in reports, dashboards and documents.

Metric Expressions:

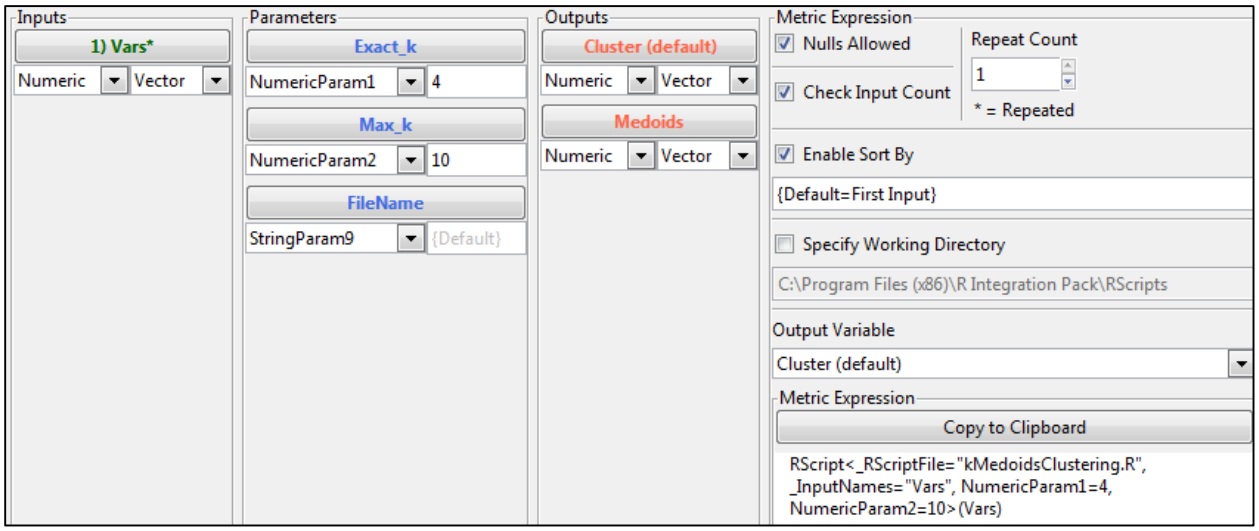
1) **Cluster:** Returns the cluster to which the record belongs:

```
RScript<_RScriptFile="kMedoidsClustering.R",_InputNames="Vars",NumericParam1=4,
NumericParam2=10>(Vars)
```

2) **Medoids:** Returns the cluster for which the record is the prototype (A value of 0 means the record is not prototypical of any cluster):

```
RScript<_RScriptFile="kMedoidsClustering.R",_InputNames="Vars",[_OutputVar]="Me
doids",NumericParam1=4, NumericParam2 = 10>(Vars)
```

Analytic Signature:



Inputs:

Vars: The numeric variables that are used to cluster the data. Since the Vars argument is a repeated input, it can be mapped to any number of MicroStrategy metrics. In this way, we enable our algorithm to consider any number of variables when generating predictions.

Parameters:

Exact_k: Uses NumericParam2 with a default of 4. This parameter specifies the number of clusters. If Exact_k is greater than 0, then the data will be grouped into Exact_k clusters. If not, then the number of clusters, less than or equal to the maximum number specified in Max_k, that best partitions the data will be used.

Max_k: Uses NumericParam1 with a default of 10. By using this parameter without a valid Exact_k value, the clustering algorithm will find the optimal number of clusters between 2 and the Max_k value.

FileName: Uses StringParam9 with a default of "". This parameter specifies the name of the .Rdata file used to persist relevant objects from the R environment isuch as the model and the data used to create it, allowing for additional inspection and analysis. Please note the R Script automatically appends the ".Rdata" file extension to this file name. If the default of ""is used, then no Rdata file is created.

Outputs:

Cluster: A numeric value representing the cluster to which that record was assigned.

Medoid: A numeric value representing the cluster for which that record is prototypical. A value of 0 indicates that the record is not prototypical of any cluster.

Additional Results Generated by the R Script:

One file is stored in the working directory:

Rdata File: This file persists the state of several objects from the R environment for later inspection, analysis, and reuse, including df (a data frame containing the data read in from MicroStrategy), and model (the k-medoids model object).

Running from the R Console:

In addition to processing data from MicroStrategy during execution of a report or dashboard, the R script is also configured to run from the R console. Running the script for the R Console verifies that the script is functioning as expected, a good practice when initially deploying this analytic to a new system (for more details, see "Configuring dual execution modes" in [1]).

When run from the R Console, if the script is executing properly, a "Success!" message will appear in the console. If a "Success!" message does not appear, then please note the error in order to take appropriate action. For common pitfalls, please consult the **Troubleshooting** section below.

Troubleshooting:

This section covers certain situations you might encounter but it's not intended as a comprehensive list of possible errors

1) If an error occurs, the report will either fail with an error message, or return nulls as the output. In these cases, please refer to the RScriptErrors.log file generated for further guidance and the DSSErrors.log. Please consult the User Guide [1] and the R documentation for additional guidance.
2) If a non-numeric variable is passed in to the Vars argument, the report will fail with an error message indicating that a variable with unexpected type was passed in. This can be remedied by using all numeric variables as inputs.

Example:

If you're using MicroStrategy version 9.4.1 or higher and would like to try this example yourself, please download the StorePerformance.mstr file from this location and import it into your MicroStrategy environment:
http://download-codeplex.sec.s-msft.com/Download?ProjectName=rintegrationpack&DownloadId=837963

Store Data Layout:

A retail organization wants to identify poor performing stores in order to detect the store managers that are most in need of training. Traditionally, they have identified poor performers by using standard business measures such as Revenue or Revenue per Square Foot. Such measures simply assume that all stores have equal revenue potential which is not a valid assumption because of differences in the nature of each store and the communities they serve.

There are specific characteristics that would allow each store to be clustered by its nature. These characteristics can include the size and age of the store plus the population density, median income and the number of local competitors in its area. It would be unfair to focus solely on the stores with the lowest overall performance since stores in more populous and prosperous areas would have an advantage over stores in smaller, less affluent locations or locations with higher numbers of competitors. Instead, by clustering stores by their nature, the performance of stores within each cluster can be easily compared and the lowest performers identified.

Using the Store Characteristic grid, let's build a four-cluster k-Mediods clustering model, by inserting a new metric named "Cluster (R)" with this expression that returns the cluster for each store and then add it to the grid.

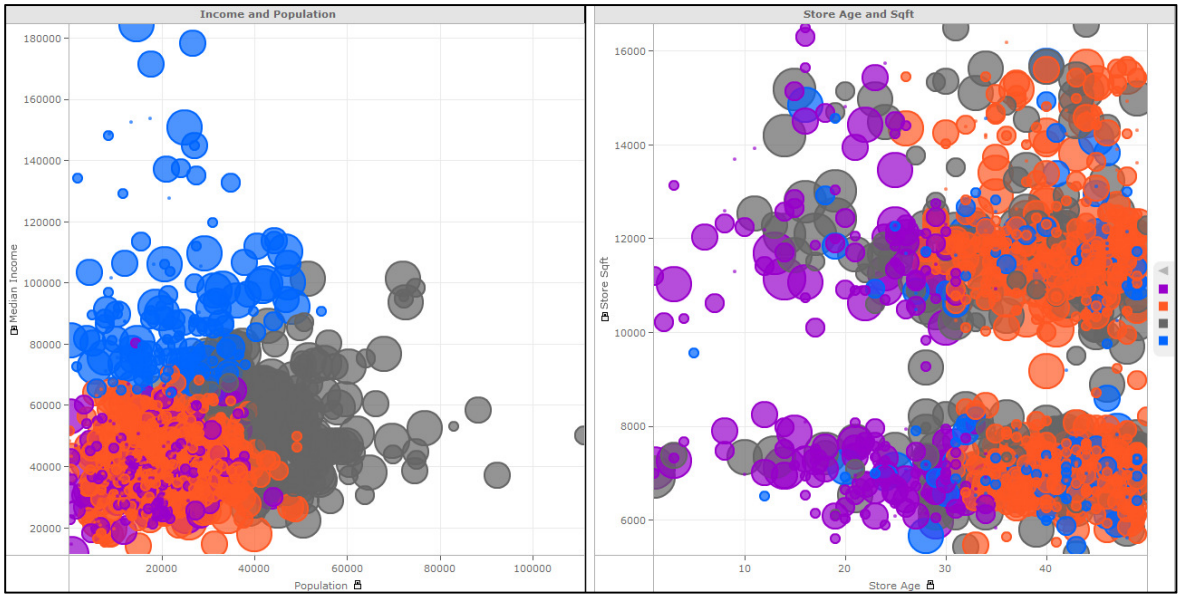
```
RScript<NumericParam1=4, NumericParam2=10,_RScriptFile="kMedoidsClustering.R">([Median
Income],Population,[Store Sqft],[Store Age],Competitors)
```

This is the result:

Store Characteristics							
Store	City	Store Sqft	Population	Competitors	Store Age	Median Income	Clusters(R)
10001	DOTHAN	7,109	35759	6	14	$38,008	1
10005	BOAZ	6,957	15693	4	14	$33,993	1
10011	BIRMINGHAM	12,605	19484	1	8	$54,491	1
10012	FORT PAYNE	11,907	4575	2	32	$38,899	2
10019	SHEFFIELD	7,501	9042	3	45	$37,147	2
10029	OPELIKA	14,256	22214	4	46	$37,923	2
10034	TALLASSEE	6,549	13179	4	38	$41,203	2
10038	ANNISTON	8,036	20150	3	33	$25,135	1
10039	HUNTSVILLE	7,114	21509	1	41	$60,686	2
10054	DECATUR	12,097	34434	5	40	$38,538	3
10055	DOTHAN	11,199	14470	3	31	$60,075	2
10056	BIRMINGHAM	11,462	19686	1	29	$31,884	1
10069	EUFAULA	7,502	13277	2	43	$27,106	2
10078	ANNISTON	10,973	19801	3	43	$50,954	2
10083	FOLEY	6,862	26767	3	33	$39,141	2
10085	DECATUR	5,429	30545	4	32	$54,565	3
10087	MOBILE	14,512	13967	1	33	$23,747	2
10092	TUSCALOOSA	6,504	39878	5	42	$17,806	2
10100	FAIRHOPE	7,496	27829	5	44	$54,296	3
10101	TALLADEGA	7,414	26822	2	41	$35,734	2
10108	PRATTVILLE	7,972	26954	3	14	$52,863	1
10109	CARROLLTON	7,416	3460	1	34	$27,124	1
10112	DEMOPOLIS	7,931	8395	1	26	$35,480	1
10113	MOBILE	11,772	14269	1	40	$48,566	2
10114	BESSEMER	6,784	27139	3	30	$25,591	1
10118	SELMA	11,061	24268	4	29	$33,488	1
10125	HALEYVILLE	12,904	13485	5	35	$31,198	2
10131	HUNTSVILLE	6,754	21509	4	34	$60,686	2
10144	MOBILE	6,416	33607	5	29	$49,568	3
10150	GREENVILLE	11,918	14074	5	33	$31,849	2
10152	MOBILE	12,334	17847	3	8	$55,483	1
10158	RUSSELLVILLE	6,707	11150	4	40	$31,016	2
10164	SYLACAUGA	10,951	18472	5	24	$33,340	1
20006	PALMER	7,070	25176	5	25	$73,230	4

Clusters Layout:

This layout contains two scatterplots. The leftmost scatterplot shows Stores by Median Income and Population; the rightmost scatterplot shows Stores by Age and Sq. Feet. Stores in both plots are sized by the number of local competitors. For each of the two visualizations, add the Cluster metric that was just created to the Color By field. Set the coloring to be Purple for 1, Orange for 2, Gray for 3, and Blue for 4. The following visualizations result:



From the left side plot, it's clear that the gray stores in Cluster 4 consists of stores in areas with higher median income and the blue Cluster 3 stores dominate areas with higher population density. Purple Cluster 1 and Orange Cluster 2 stores are mixed together as stores that are in less affluent and less populated areas.

Using the right side plot, it's easy to distinguish Cluster 1 and Cluster 2 stores. Purple Cluster 1 stores are clearly lower in age than the orange Cluster 2 stores. So while both cluster 1 and 2 represent stores in relatively poorer, smaller towns, Cluster 1 stores are newer than Cluster 2.

As it turns out, the number of competitors and the size of the store were not needed to describe each cluster. The nature of the local population and the age of the store are the key characteristics identified by the k-Medoids algorithm to cluster stores effectively.

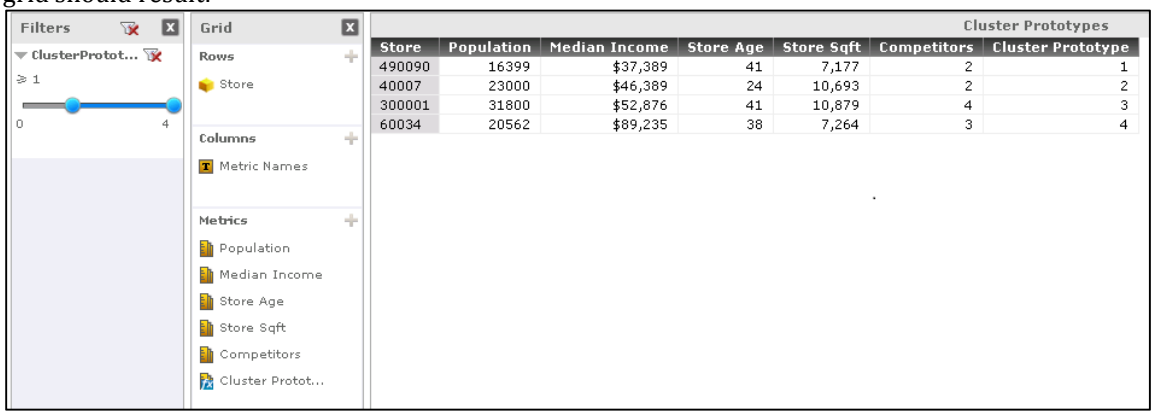
Now that the nature of the clusters is understood, we can apply business measures in order to identify the lowest performers in each cluster.

Cluster Prototypes Layout:

The primary benefit associated with k-medoids clustering is the ability to understand the nature of each cluster by identifying a prototypical example from the dataset. You can use the Medoids output from the kMedoidsClustering.R script to understand the prototypical store for each cluster. On the Cluster Prototypes layout, create a metric with the following definition and name it Cluster Prototype.

```
RScript<NumericParam1=4,NumericParam2=10,_RScriptFile="kMedoidsClustering.R",_OutputVa
r="Medoids">([Median Income],Population,[Store Sqft],[Store Age],Competitors)
```

Now, add this metric to the columns and filter panel. Set the Cluster Prototype filter to 1 to 4. The following grid should result:



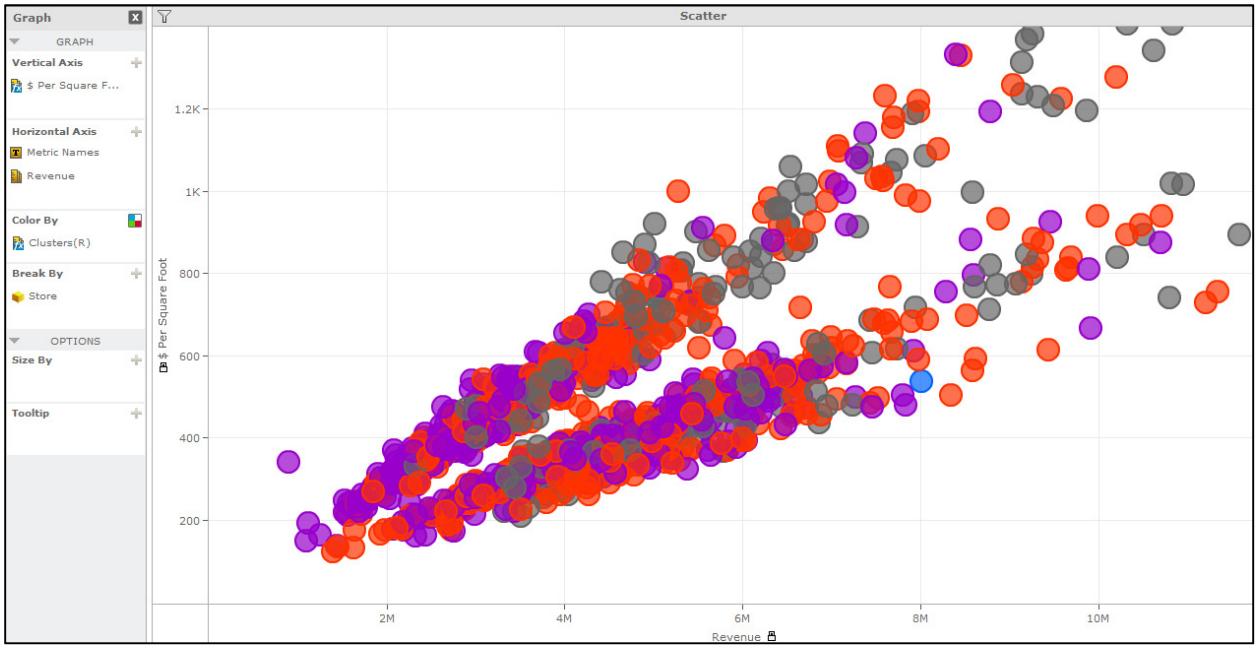
Cluster Prototypes

Store	Population	Median Income	Store Age	Store Sqft	Competitors	Cluster Prototype
490090	16399	$37,389	41	7,177	2	1
40007	23000	$46,389	24	10,693	2	2
300001	31800	$52,876	41	10,879	4	3
60034	20562	$89,235	38	7,264	3	4

Observe that the prototypical store for Cluster 3 is in a highly populated area. The prototypical store for Cluster 4 is in a high income area, with a median income of almost $90,000. Also, you can see that the prototypical stores for Clusters 1 and 2 are both in relatively low population areas with relatively low median income. However, these stores are differentiated by their ages; the prototype for Cluster 1 is 41 months, compared to just 24 months for Cluster 2's prototype. With this analysis, we can confirm the nature of the clusters we identified in the previous layout.

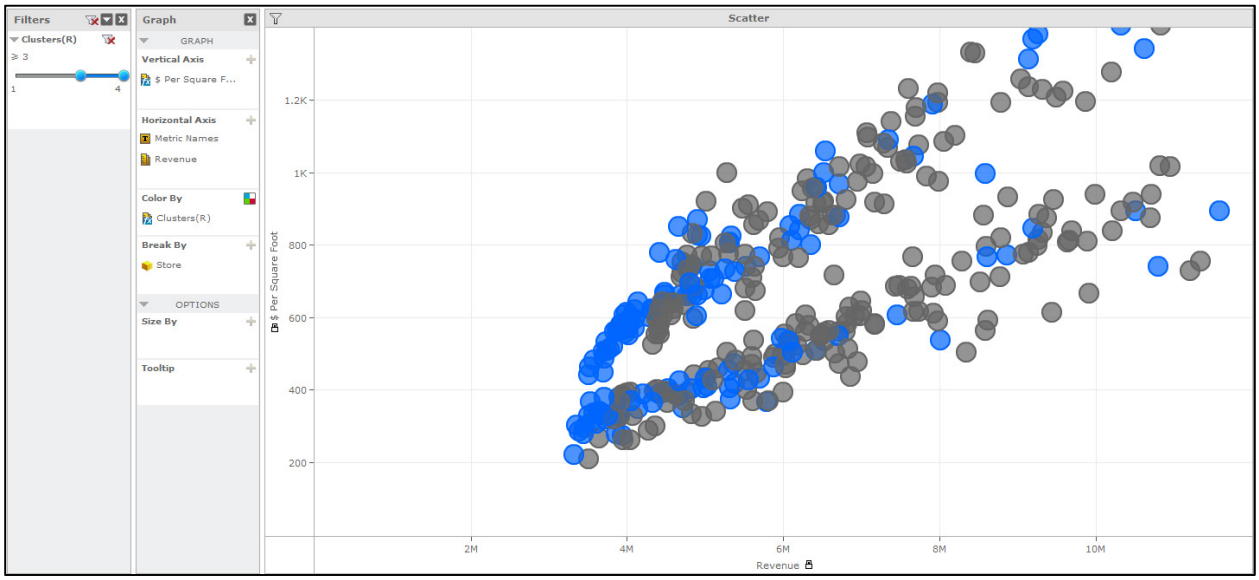
Poor Performers Layout:

On this layout, observe that there are only a handful of stores with Revenue below $2M and Revenue per Sq. Foot less than $200. Intuitively, we can identify these stores as poor performers and administer our training to the managers of these stores.



Now that we have clustered the stores, we can see which stores have lower Revenue and Revenue per Sq. Feet compared to their peers. In essence, we can use the clustering results to benchmark what the expected Revenue should be.

Add the Cluster (R) clustering metric that was created to the Color By field, and the filtering area. Set the Cluster (R) filter to Clusters 3 and 4. Observe that there are now several stores that although they have revenue greater than $3.5M, are clearly performing poorer than their peers. These stores should be considered poor performers and their managers should also receive training. Without the clustering, we would have missed that these stores performed poorly.



Next Steps:

As noted in the introduction section, the identification of clusters is just the beginning of the analysis. The next step would be to analyze how each cluster performs based on key performance indicators such as Revenue, Profit, and Sales Volume. For users of MicroStrategy Analytics Enterprise, this process can be accomplished in either of two ways:

- Creating a data mart from the Clustering report and modeling the Cluster output as an attribute.
- Creating a custom group based on the Cluster output.

Once the Cluster output has been either modeled as an attribute or used to define a custom group, it can be linked to business performance by placing the attribute or custom group on rows and KPIs of interest on the columns. With this report, you can see which clusters are most and least beneficial to your organization.

Please see Chapters 4 and 13 of the MicroStrategy Advanced Reporting Guide for more information on how to use the outputs of a clustering model to analyze performance relative to important business measures.

References:

1) MicroStrategy R Integration Pack User Guide: https://rintegrationpack.codeplex.com/documentation
2) R Integration Pack Analytic Shelf: Off-The-Shelf R Script Repository: https://rintegrationpack.codeplex.com/wikipage?title=R%20Script%20%22Shelf%22&referringTitle=Home#
3) http://en.wikipedia.org/wiki/k-medoids
4) http://en.wikipedia.org/wiki/Silhouette_(clustering)

