

---

# Data Mining

## SPSS Clementine 12.0

### 6. *k*-Means Algorithm

**Spring 2010**

Instructor: Dr. Masoud Yaghini

# Outline

---

---

- **K-Means Algorithm in Clementine**
- **K-Means Node**
- **References**

---

---

# **K-Means Algorithm in Clementine**

# Overview

---

- The ***k*-means method** is a clustering method, used to group records based on similarity of values for a set of input fields.
- The basic idea is to try to discover  $k$  clusters, such that the records within each cluster are similar to each other and distinct from records in other clusters.
- *K*-means is an iterative algorithm; an initial set of clusters is defined, and the clusters are repeatedly updated until
  - no more improvement is possible or
  - the number of iterations exceeds a specified limit.

# Primary Calculations

---

- Input fields are recoded before the values are input to the algorithm including:
  - **Scaling of Range Fields**
  - **Numeric Coding of Symbolic Fields**
  - **Encoding of Flag Fields**

# Scaling of Range Fields

---

- To compensate for the effect of scale, range fields are transformed so that they all have the same scale.
- In Clementine, range fields are rescaled to have values between 0 and 1.
- The transformation used is

$$x_i' = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}}$$

# Numeric Coding of Symbolic Fields

---

- Clementine recode a **symbolic field** as a group of numeric fields with one numeric field for each category or value of the original field.
- For each record, the value of the derived field corresponding to the category of the record is set to 1.0, and all the other derived field values are set to 0.0.
- Such derived fields are sometimes called **indicator fields**, and this recoding is called **indicator coding**.

# Numeric Coding of Symbolic Fields

---

- For example, consider the following data, where  $x$  is a symbolic field with possible values A, B, and C:

Record #	$X$	$X_1'$	$X_2'$	$X_3'$
1	B	0	1	0
2	A	1	0	0
3	C	0	0	1

- There is a problem.

# Numeric Coding of Symbolic Fields

---

- For algorithms that use the **Euclidean distance** to measure differences between records, the difference between two records with different values  $i$  and  $j$  for the **set** is

$$\sqrt{\sum_{k=1}^J (x_{k1} - x_{k2})^2}$$

- where  $J$  is the number of categories, and  $x_{kn}$  is value of the derived indicator for category  $k$  for record  $n$ .

# Numeric Coding of Symbolic Fields

---

- The values will be different on two of the derived indicators,  $x_i$  and  $x_j$ .
- The sum will be,

$$\sqrt{(1 - 0)^2 + (0 - 1)^2} = \sqrt{2} \approx 1.414,$$

- which is larger than 1.0.
- That means that based on this coding, set fields will have more weight in the model than range fields that are rescaled to 0-1 range.

# Numeric Coding of Symbolic Fields

---

- To account for this bias, Clementine applies a scaling factor to the derived set fields, such that a difference of values on a set field produces a Euclidean distance of **1.0**.
- The default scaling factor is:

$$\sqrt{\frac{1}{2}} \approx 0.707$$

- This value gives the desired result:

$$\sqrt{\left(\sqrt{\frac{1}{2}} - 0\right)^2 + \left(0 - \sqrt{\frac{1}{2}}\right)^2} = \sqrt{\frac{1}{2} + \frac{1}{2}} = 1$$

# Encoding of Flag Fields

---

- **Flag fields** are a special case of symbolic fields.
- They have only two values in the set, they can be handled in a slightly more efficient way than other set fields.
- Flag fields are represented by a single numeric field, taking the value of 1.0 for the “true” value and 0.0 for the “false” value.
- Blanks for flag fields are assigned the value 0.5.

# Main Steps of Algorithm

---

- 1. Select initial cluster centers
- 2. Assign each record to the nearest cluster
- 3. Update the cluster centers based on the records assigned to each cluster
- 4. Repeat steps 2 and 3 until either:
  - In step 3, there is no change in the cluster centers from the previous iteration, or
  - The number of iterations exceeds the maximum iterations parameter

# Selecting Initial Cluster Centers

---

- Initial cluster centers are chosen using a maximin algorithm:
  - 1. Initialize the first cluster center as the values of the input fields for the first data record.
  - 2. For each data record, compute the minimum (Euclidean) distance between the record and each defined cluster center.
  - 3. Select the record with the largest minimum distance from the defined cluster centers. Add a new cluster center with values of the input fields for the selected record.
  - 4. Repeat steps 2 and 3 until  $k$  cluster centers have been added to the model.

# Assigning Records to Clusters

---

- In each iteration of the algorithm, each record is assigned to the cluster whose center is closest.
- Closeness is measured by the usual squared Euclidean distance:

$$d_{ij} = \|X_i - C_j\|^2 = \sum_{q=1}^Q (x_{qi} - c_{qj})^2$$

- where  $X_i$  is the vector of encoded input fields for record  $i$ ,
- $C_j$  is the cluster center vector for cluster  $j$ ,
- $Q$  is the number of encoded input fields,
- $x_{qi}$  is the value of the  $q$ th encoded input field for the  $i$ th record, and
- $c_{qj}$  is the value of the  $q$ th encoded input field for the  $j$ th cluster.

# Updating Cluster Centers

---

- After records have been (re)assigned to their closest clusters, the cluster centers are updated.
- The cluster center is calculated as the mean vector of the records assigned to the cluster:

$$C_j = \overline{X}_j$$

- where the components of the mean vector are calculated in the usual manner:

$$\overline{x}_{qj} = \frac{\sum_{i=1}^{n_j} x_{qi}(j)}{n_j}$$

- where  $n_j$  is the number of records in cluster  $j$ ,
- $x_{qi}(j)$  is the  $q$ th encoded field value for record  $i$  which is assigned to cluster  $j$ .

# Blank Handling

---

- In k-means, blanks are handled by substituting **neutral** values for the missing ones.
- For **range** and **flag fields** with missing values (blanks and nulls), the missing value is replaced with 0.5.
- For **set fields**, the derived indicator field values are all set to 0.0.

# Effect of Options

---

- **Maximum Iterations**

- The maximum iterations parameter controls how long the algorithm will continue searching for a stable cluster solution.
- The algorithm will repeat the classify/update cycle no more than the number of times specified.
- If and when this limit is reached, the algorithm terminates and produces the current set of clusters as the final model.

# Effect of Options

---

- **Error Tolerance**

- The error tolerance parameter provides another means of controlling how long the algorithm will continue searching for a stable cluster solution.
- The maximum change in cluster means for an iteration  $t$  is calculated as

$$\max_J || C_j(t) - C_j(t - 1) ||$$

- where  $C_j(t)$  is the cluster center vector for the  $j$ th cluster at iteration  $t$
  - $C_j(t - 1)$  is the cluster center vector at the previous iteration.
- If the maximum change is less than the specified tolerance for the current iteration, the algorithm terminates and produces the current set of clusters as the final model.

# Effect of Options

---

- **Encoding Value for Sets**

- The encoding value for **sets** parameter controls the relative weighting of set fields in the *k*-means algorithm.
- The default value is:  $\sqrt{0.5} \approx 0.707$
- It provides an equal weighting between range fields and set fields.
- To emphasize **set fields** more heavily, you can set the encoding value closer to 1.0
- To emphasize **range fields** more, set the encoding value closer to 0.0.

# Model Summary Statistics

---

- Cluster proximities are calculated as the Euclidean distance between cluster centers:

$$d_{ij} = \|C_i - C_j\| = \sqrt{\sum_{q=1}^Q (c_{qi} - c_{qj})^2}$$

# Primary Calculations

---

- The value of the distance field for each record, if requested, is calculated as the Euclidean distance between the record and its assigned cluster center:

$$d_{ij} = \|X_i - C_j\| = \sqrt{\sum_{q=1}^Q (x_{qi} - c_{qj})^2}$$

---

# **K-Means Node**

# Introduction

---

- The K-Means node provides a method of **cluster analysis**.
- K-Means models **do not use a target field**.
- **Requirements:**
  - To train a K-Means model, you need one or more *In* fields.
  - Fields with direction *Out*, *Both*, or *None* are ignored.
- **Strengths:**
  - The K-Means model is often the fastest method of clustering for large datasets.

# K-Means Node Model Options

The screenshot shows the 'K-Means' dialog box with the following settings:

- Model name:  Auto  Custom
- Use partitioned data
- Number of clusters: 5
- Generate distance field
- Show cluster proximity
- Cluster label:  String  Number
- Label prefix: cluster
- Optimize:  Speed  Memory

At the bottom, there are tabs for 'Fields', 'Model', 'Expert', and 'Annotations'. Below the tabs are buttons for 'OK', 'Execute', 'Cancel', 'Apply', and 'Reset'.

# K-Means Node Model Options

---

- **Model name.**
  - You can generate the model name automatically based on the target or ID field (or model type in cases where no such field is specified) or specify a custom name.
- **Use partitioned data.**
  - If a partition field is defined, this option ensures that data from only the training partition is used to build the model.
- **Specified number of clusters.**
  - Specify the number of clusters to generate. The default is 5.

# K-Means Node Model Options

---

- **Generate distance field.**
  - If this option is selected, the model nugget will include a field containing the distance of each record from the center of its assigned cluster.
- **Show cluster proximity.**
  - Select this option to include information about distances between cluster centers in the model nugget output.
- **Cluster label.**
  - Specify the format for the generated cluster membership field. Cluster membership can be indicated as a String with the specified Label prefix (for example "Cluster 1", "Cluster 2", and so on), or as a Number.

# K-Means Node Model Options

---

- **Optimize.**

- Select options designed to increase performance during model building based on your specific needs.
- Select **Speed** to instruct the algorithm to never use disk spilling in order to improve performance.
- Select **Memory** to instruct the algorithm to use disk spilling when appropriate at some sacrifice to speed. This option is selected by default.

# K-Means Node Model Options

The screenshot shows the 'K-Means' dialog box with the following settings:

- Mode:**  Simple,  Expert
- Stop on:**  Default,  Custom
- Maximum iterations:** 20
- Change tolerance:** 0.0
- Encoding value for sets:** 0.70711

The 'Expert' tab is selected in the bottom tab bar. The 'Execute' button is highlighted with a green play icon.

# K-Means Node Expert Options

---

- **Stop on.**
  - Specify the stopping criterion to be used in training the model.
- **Maximum Iterations.**
  - This option allows you to stop model training after the number of iterations specified.
- **Change tolerance.**
  - This option allows you to stop model training when the largest change in cluster centers for an iteration is less than the level specified.

# K-Means Node Expert Options

---

- **Encoding value for sets.**

- Specify a value between 0 and 1.0 to use for recoding set fields as groups of numeric fields.
- The default value is the square root of 0.5 (approximately 0.707107), which provides the proper weighting for recoded flag fields.
- Values closer to 1.0 will weight set fields more heavily than numeric fields.

# K-Means Model Nuggets

---

- K-Means model nuggets contain all of the information captured by the clustering model
- When you execute a stream containing a K-Means model nugget, the node adds two new fields containing:
  - the cluster membership and
  - distance from the assigned cluster center for that record.
- The new field names are derived from the model name, prefixed by
  - \$KM- for the cluster membership and
  - \$KMD- for the distance from the cluster center.

# K-Means Model Tab

---

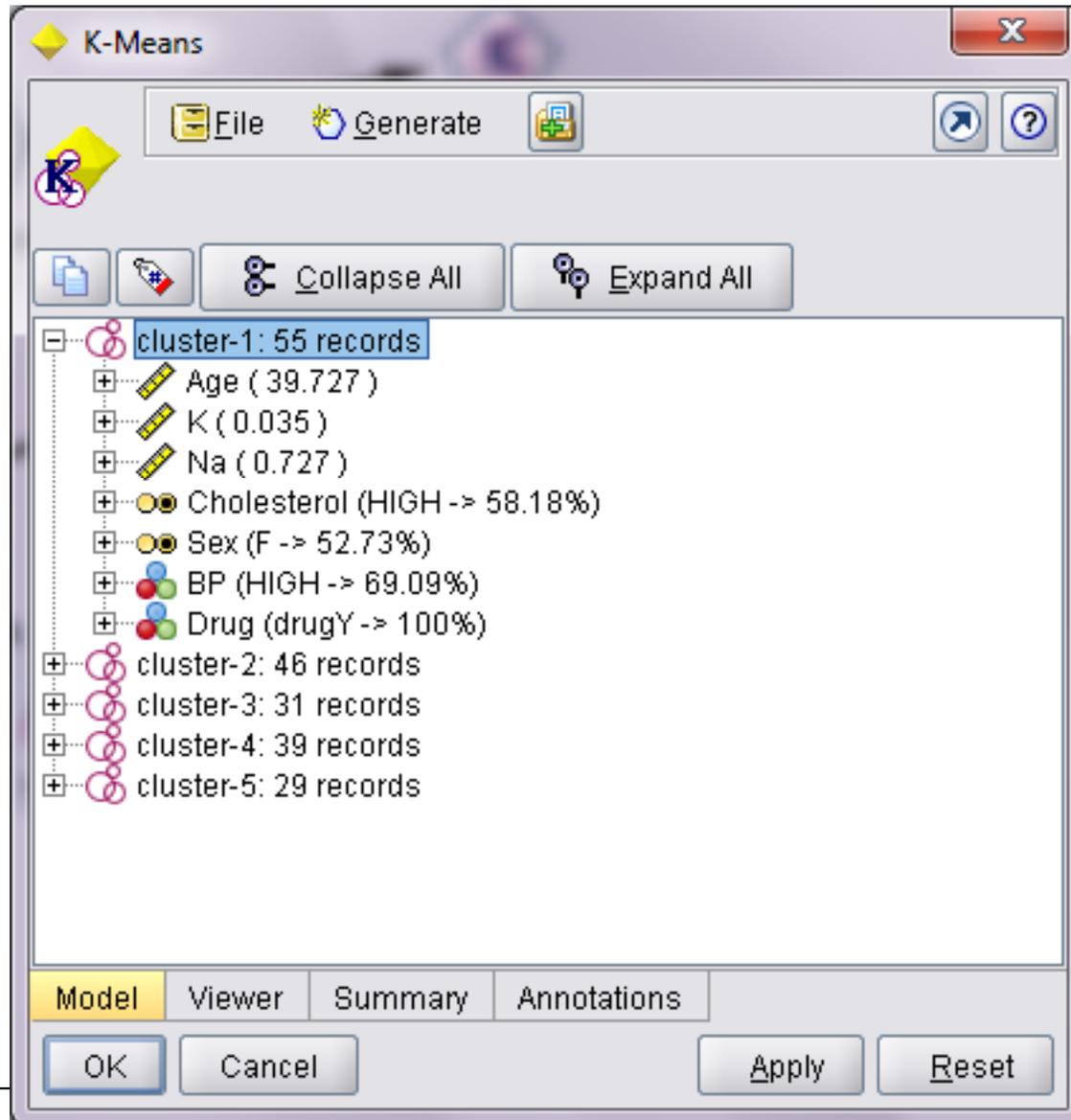
- The Model tab for a K-Means model nugget contains detailed information about the clusters defined by the model.
- Clusters are labeled and the **number of records** assigned to each cluster is shown.
- Each cluster is described by its center, which can be thought of as the **prototype** for the cluster.
- For **scale fields**, the mean value for training records assigned to the cluster is given;
- For **symbolic fields**, the proportion for each distinct value is reported.

# K-Means Model Tab

---

- If you requested Show cluster proximity in the K-Means node used to generate the model nugget, each cluster description will also contain its proximities from every other cluster.

# K-Means Model Tab

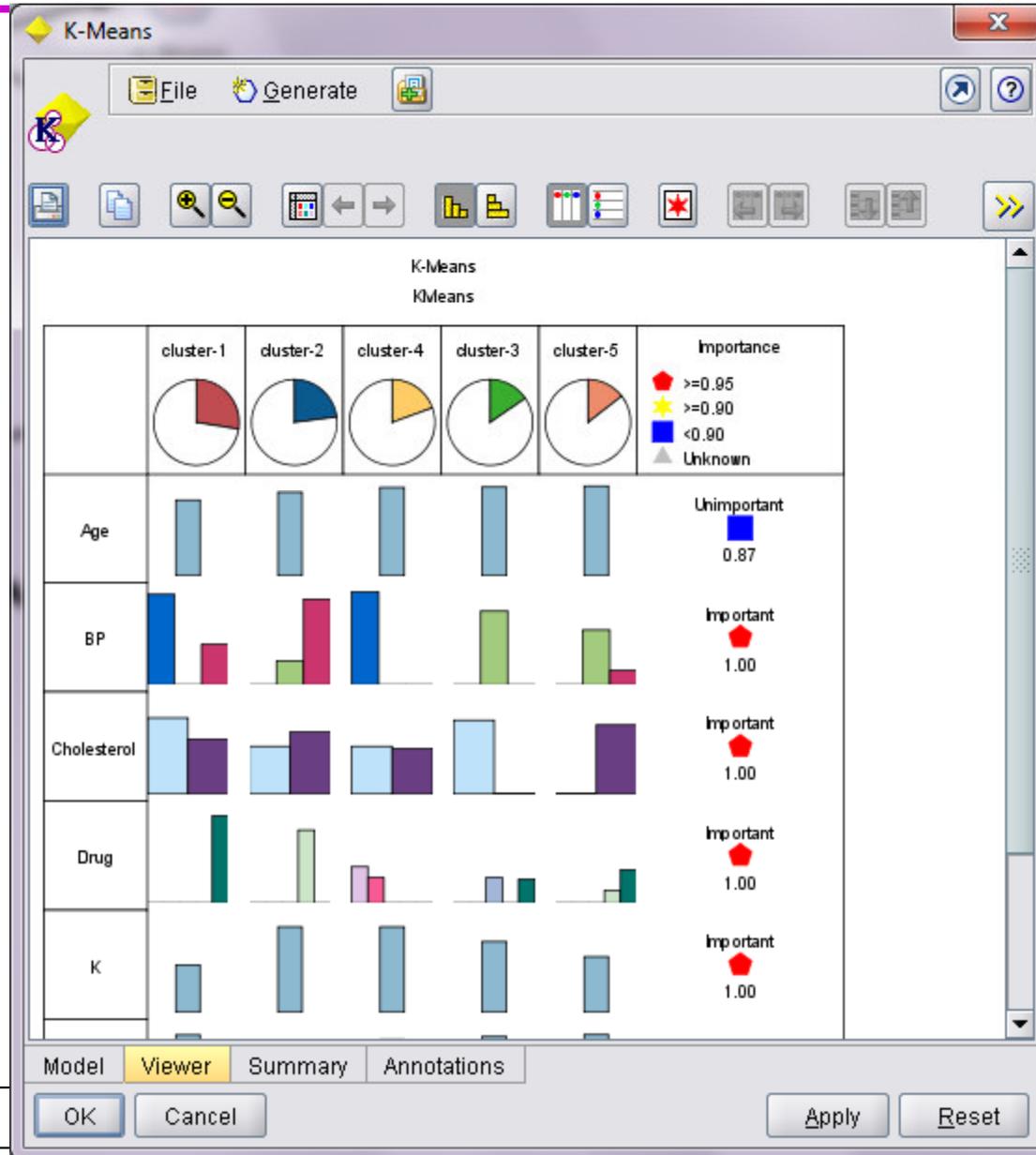


# K-Means Viewer Tab

---

- The **Viewer tab** shows a graphical display of summary statistics and distributions for fields between clusters.
- By default, the clusters are displayed on the  $x$  axis and the fields on the  $y$  axis.

# K-Means Viewer Tab



Clementine

# K-Means Viewer Tab

---

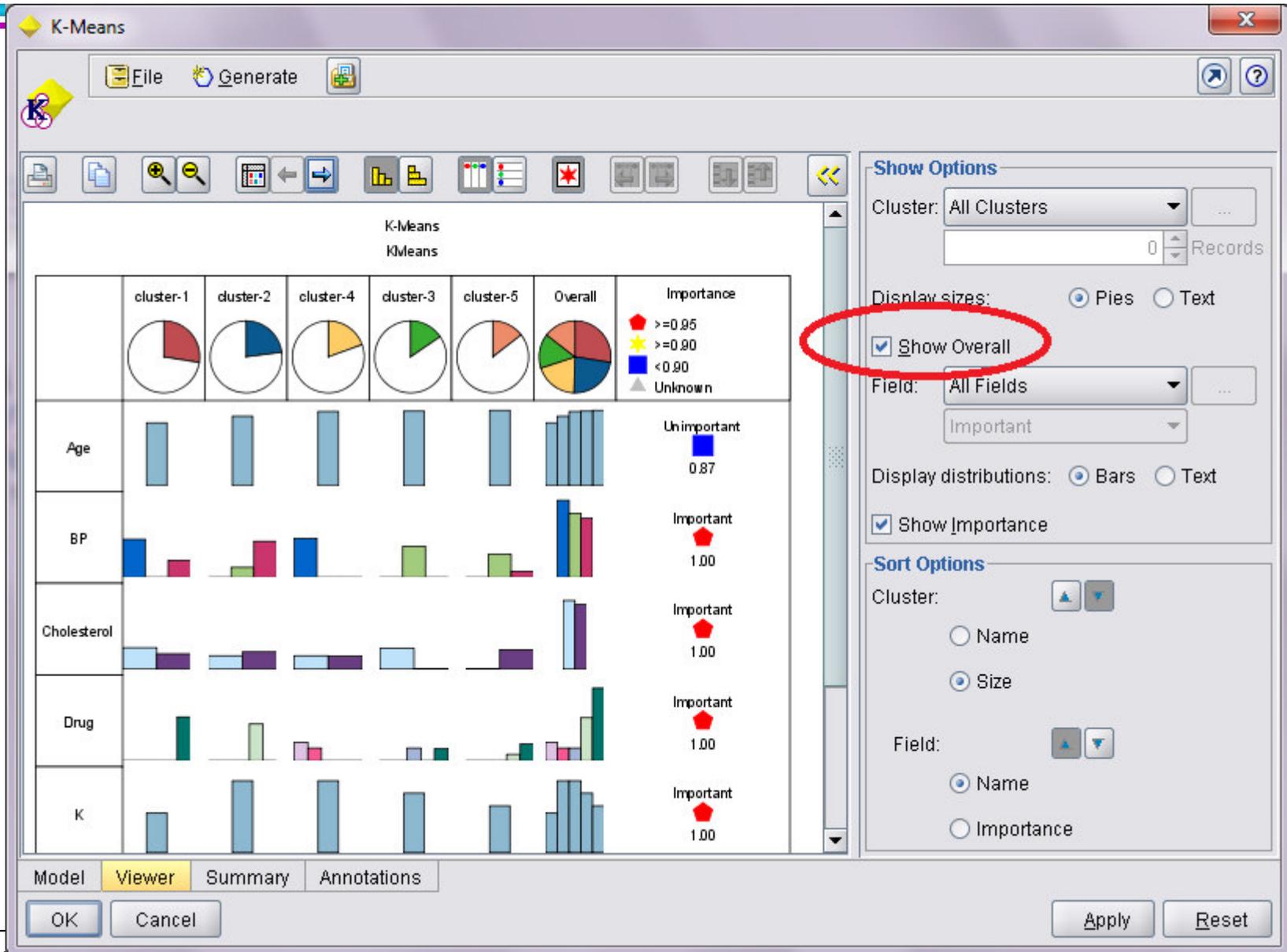
- The **cluster axis** lists each cluster in cluster number order and by default includes an **Importance column**.
- The **Importance column** displays the overall importance of the field to the model.
- It is displayed as 1 minus the *p value* (probability value from the **t test** or **chi-square test** used to measure importance).

# K-Means Viewer Tab

---

- An **Overall column** can be added using options on the **expanded dialog** box.
- The **Overall column** displays the values (represented by bars) for all clusters in the dataset and provides a useful comparison tool.
- Expand the dialog box using the **yellow arrow** button and select the **Show Overall** option.

# K-Means Viewer Tab



# K-Means Viewer Tab

---

- The **field axis** lists each field (variable) used in the analysis and is sorted alphabetically.
- Both discrete fields and scale fields are displayed by default.
- The **individual cells** of the table shows summaries of a given field's values for the records in a given cluster.

# Understanding the Cluster View

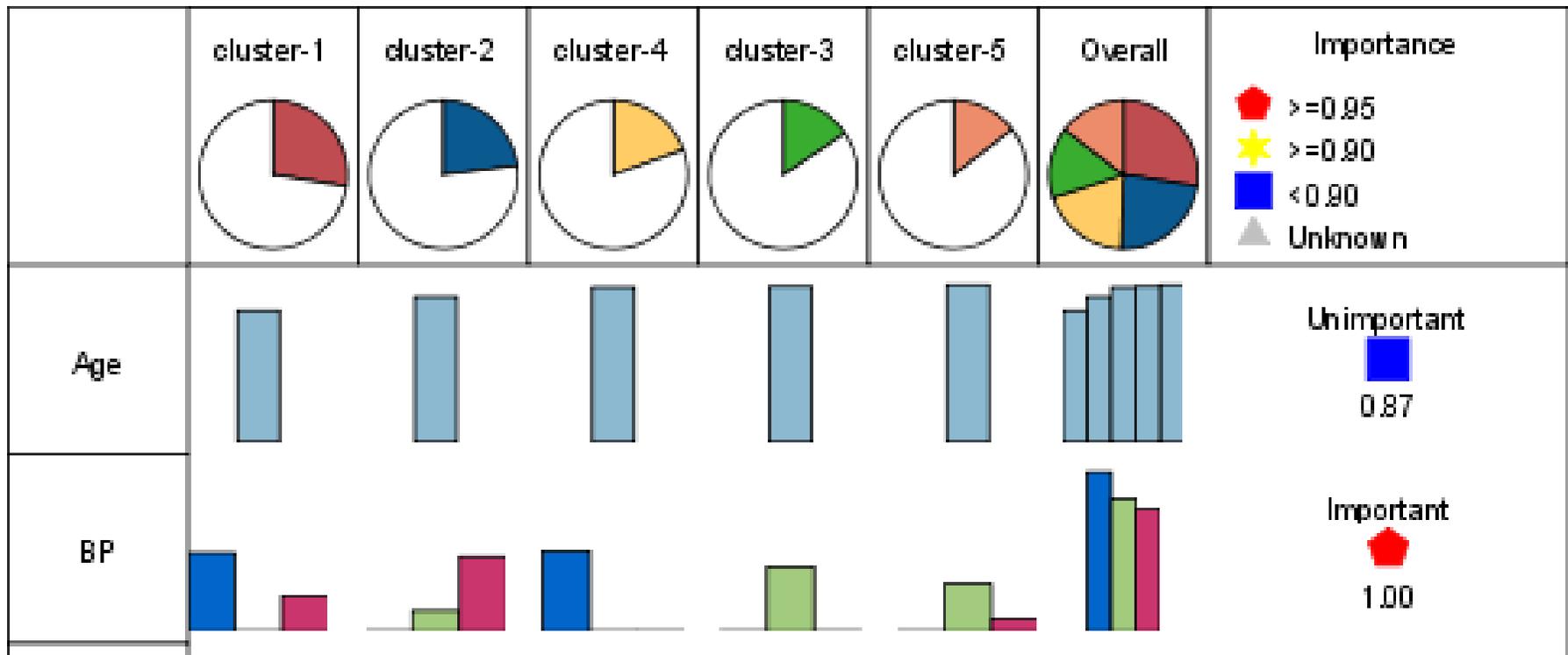
---

---

- There are two approaches to interpreting the results in a cluster display:
  - Examine clusters to determine characteristics unique to that cluster.
    - ◆ Does one cluster contain all the high-income borrowers?
    - ◆ Does this cluster contain more records than the others?
  - Examine fields across clusters to determine how values are distributed among clusters.
    - ◆ Does one's level of education determine membership in a cluster?
    - ◆ Does a high credit score distinguish between membership in one cluster or another?

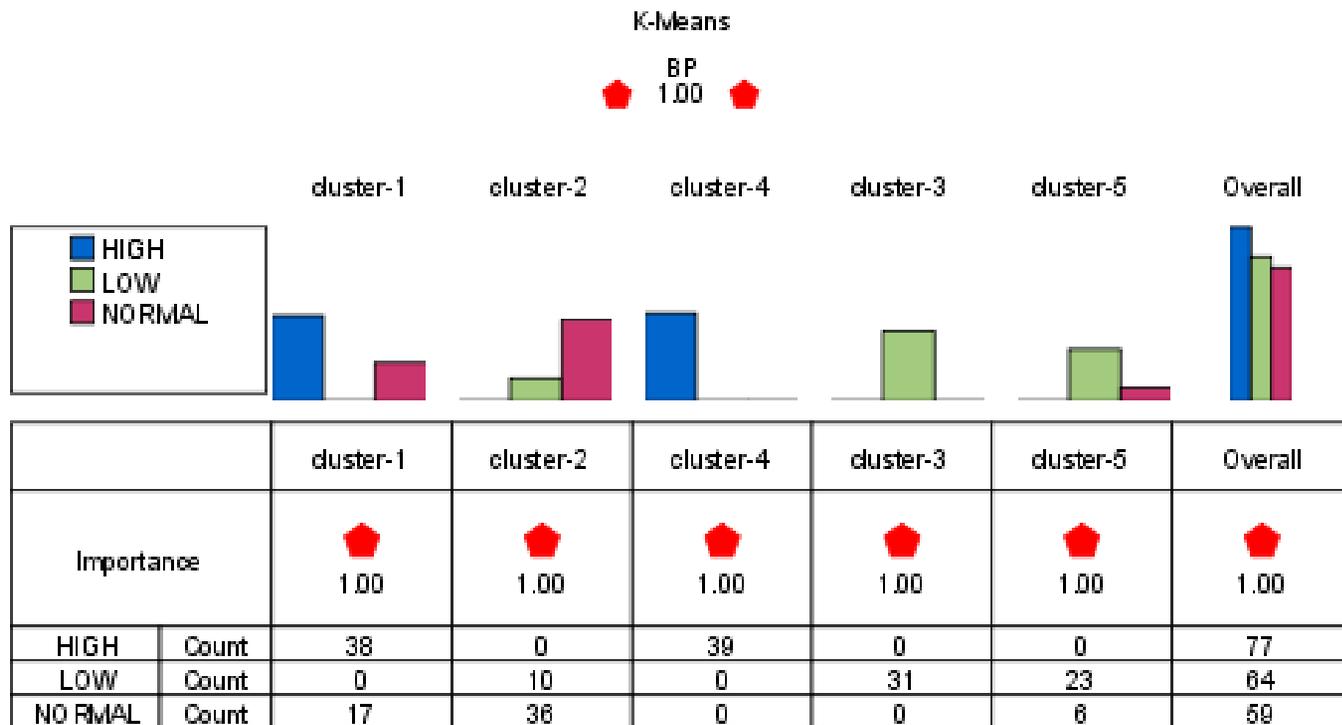
# Understanding the Cluster View

- For example, examine *BP* (blood pressure) field, notice that clusters 2 and 5 contain entirely different values for the *BP* field.



# Understanding the Cluster View

- This information, combined with the importance level, tells you that blood pressure is an important determinant of membership in a cluster.
- You can double-click the field for a more detailed view, displaying actual values and statistics.



# What Is Importance?

---

- The higher the importance measure, the less likely the variation for a field between clusters is due to chance
- Importance is calculated as **1 minus the significance value** of a statistical test.
- For **categorical variables**
  - the test is a **chi-squares test**.
  - The null hypothesis is within-cluster distributions of category counts are the same across cluster.
  - If this categorical variable is really influential in determining cluster, the null hypothesis will be rejected and the significance level will be close to zero.
  - Hence the Importance index is close to one.

# What Is Importance?

---

---

- For **continuous variables**
  - The test is a **Student's  $t$  test**.
  - The null hypothesis is within-cluster means are the same across cluster.
  - If this continuous variable is really influential in determining cluster, the null hypothesis will be rejected and the significance level will be close to zero.
  - Hence the Importance index is close to one.

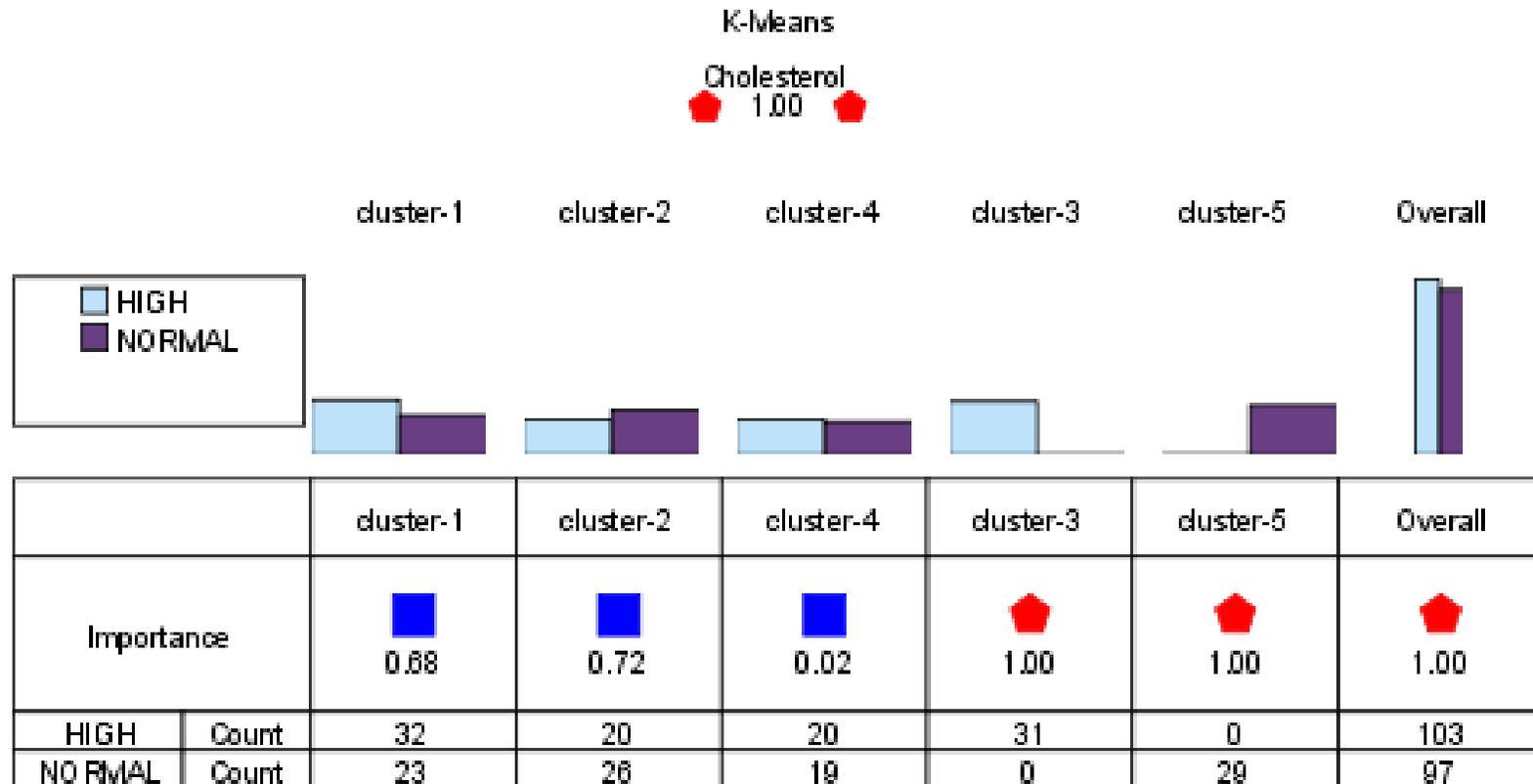
# Reading the Display for Discrete Fields

---

- For discrete fields, or sets, the **Top View** displays distribution charts indicating the category counts of the field for each cluster.
- **Drill-down** (by double-clicking or using the expanded tab options) to view actual counts for each value within a cluster.
- These counts indicate the number of records with the given value that fall into a specific cluster.

# Reading the Display for Discrete Fields

- Drill-down view for a discrete field



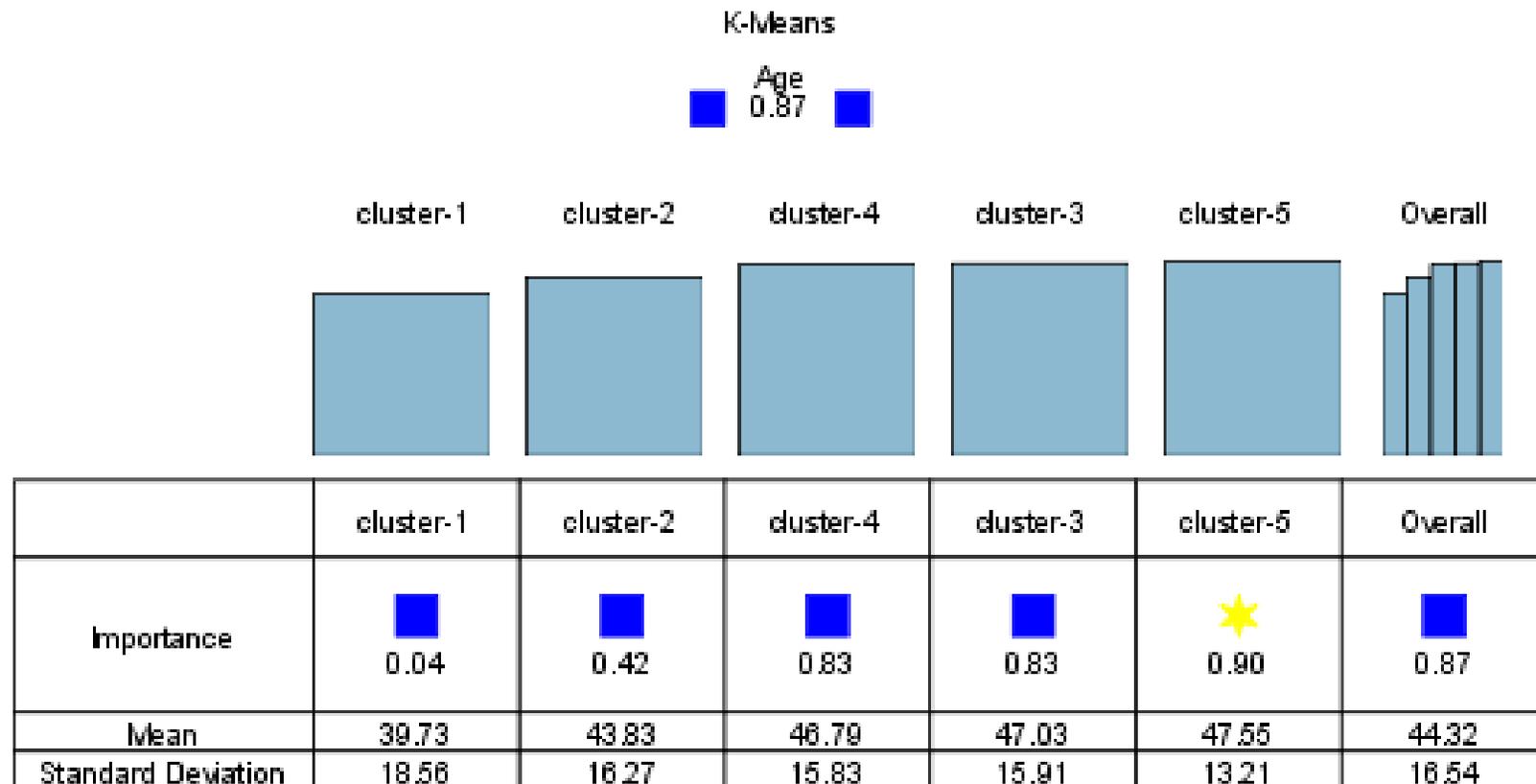
# Reading the Display for Scale Fields

---

- For scale fields, the **Viewer** displays bars representing the mean value of a field for each cluster.
- The **Overall** column compares these mean values, but is not a histogram indicating frequency distribution.
- Drill-down (by double-clicking or using the expanded tab options) to view the actual mean value and standard deviation of the field for each cluster.

# Reading the Display for Scale Fields

- Drill-down view for a scale field



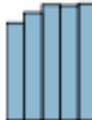
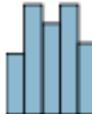
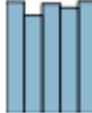
# Reading Cluster Details

---

- You can view detailed information about a single cluster by drilling-down into the display.
- This is an effective way to quickly examine a cluster of interest and determine which field(s) might contribute to the cluster's uniqueness.
- Compare the **Cluster** and **Overall** charts by field and use the importance levels to determine fields that provide separation or commonality between clusters.
- Using the mouse or the keyboard, you can drill-down to view more details for a field or cluster.

# Reading Cluster Details

K-Means  
cluster-3

	cluster-3	Overall	Importance
			<ul style="list-style-type: none"> <li><span style="color: red;">★</span> <math>\geq 0.95</math></li> <li><span style="color: yellow;">★</span> <math>\geq 0.90</math></li> <li><span style="color: blue;">■</span> <math>&lt; 0.90</math></li> <li><span style="color: gray;">▲</span> Unknown</li> </ul>
Age			Unimportant <span style="color: blue;">■</span> 0.87
BP			Important <span style="color: red;">★</span> 1.00
Cholesterol			Important <span style="color: red;">★</span> 1.00
Drug			Important <span style="color: red;">★</span> 1.00
K			Important <span style="color: red;">★</span> 1.00
Na			Important <span style="color: red;">★</span> 1.00
Sex			Unimportant <span style="color: blue;">■</span> 0.83

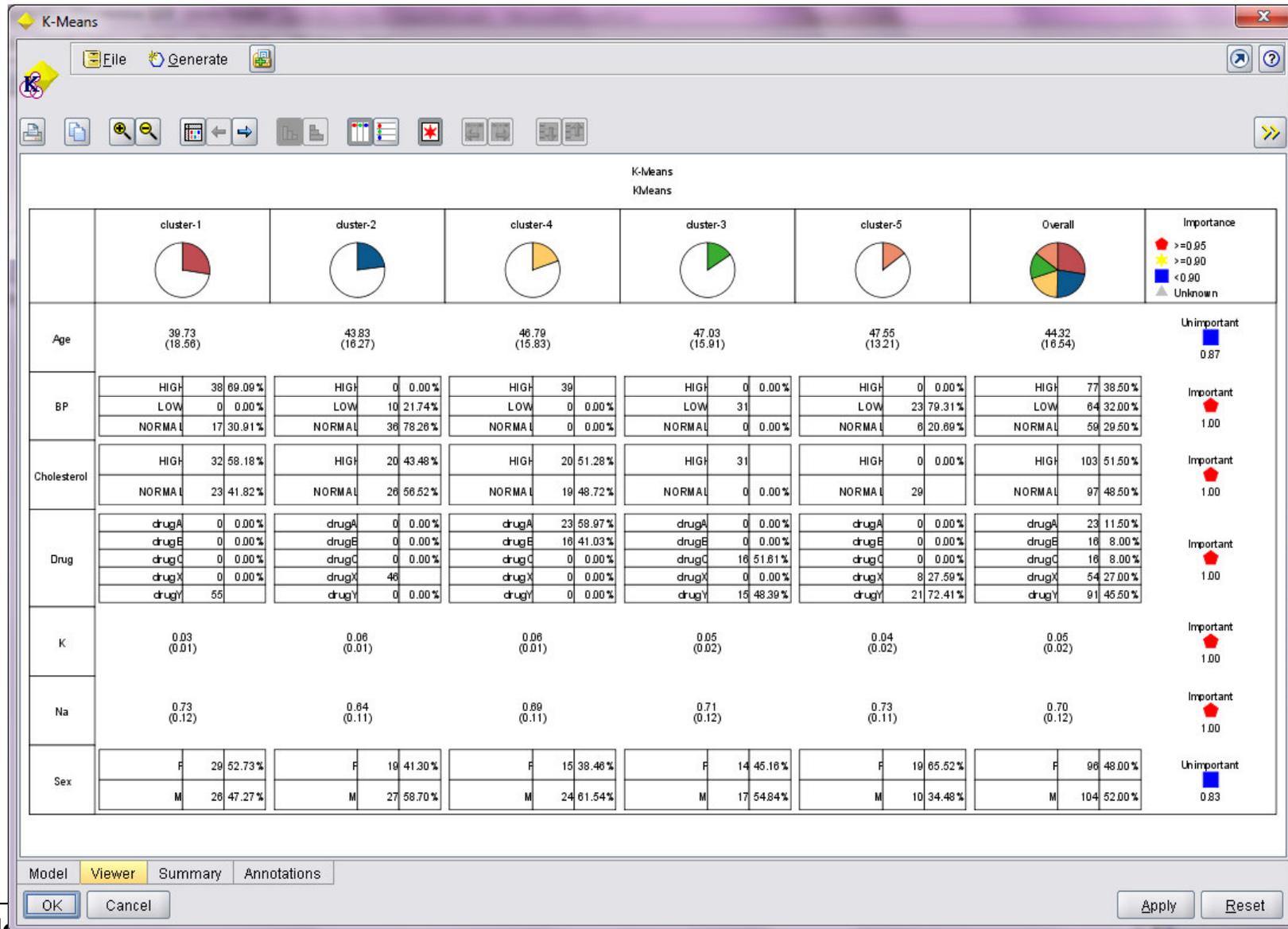
Clementine

# Viewing Clusters As Text

---

- Information in the Cluster Viewer can also be displayed as text, where all values are displayed as numerical values instead of as charts.
- The text view, while different in appearance, operates in the same manner as the graphical view.
- To view as text:
  - Click the **yellow arrow** at the top of the **Viewer** to expand for more options.
  - For both **Display sizes** and **Display distributions**, you can select to view results as text.

# Reading Cluster Details



# K-Means Model Summary

---

- The Summary tab for a K-Means model nugget contains information about the training data, the estimation process, and the clusters defined by the model.
- The number of clusters is shown, as well as the iteration history.

# K-Means Model Summary

The screenshot displays the 'K-Means' dialog box in IBM SPSS Clementine. The window title is 'K-Means'. The interface includes a menu bar with 'File', 'Generate', and a help icon. Below the menu bar are buttons for 'Collapse All' and 'Expand All'. The main area shows a tree view of the model structure with the following components:

- Analysis
  - Number of clusters: 5
    - Iteration
    - Error
- Fields
- Build Settings
- Training Summary

The 'Iteration' and 'Error' table is expanded and shows the following data:

Iteration	Error
1	0.811
2	0.245
3	0.157
4	0.1
5	0.043
6	0.072
7	0.096
8	0.0

At the bottom of the dialog, there are tabs for 'Model', 'Viewer', 'Summary' (which is selected), and 'Annotations'. Below the tabs are buttons for 'OK', 'Cancel', 'Apply', and 'Reset'.

---

# References

# References

---

- Integral Solutions Limited., **Clementine® 12.0 Algorithms Guide**, 2007. (Chapter 7)
- Integral Solutions Limited., **Clementine® 12.0 Modeling Nodes**, 2007. (Chapter 11)



The end