AGENDA

1. Unsupervised learning Association rules
2. Unsupervised learning CLUSTER ANALYSIS
3. Deployment

Data Science Academy

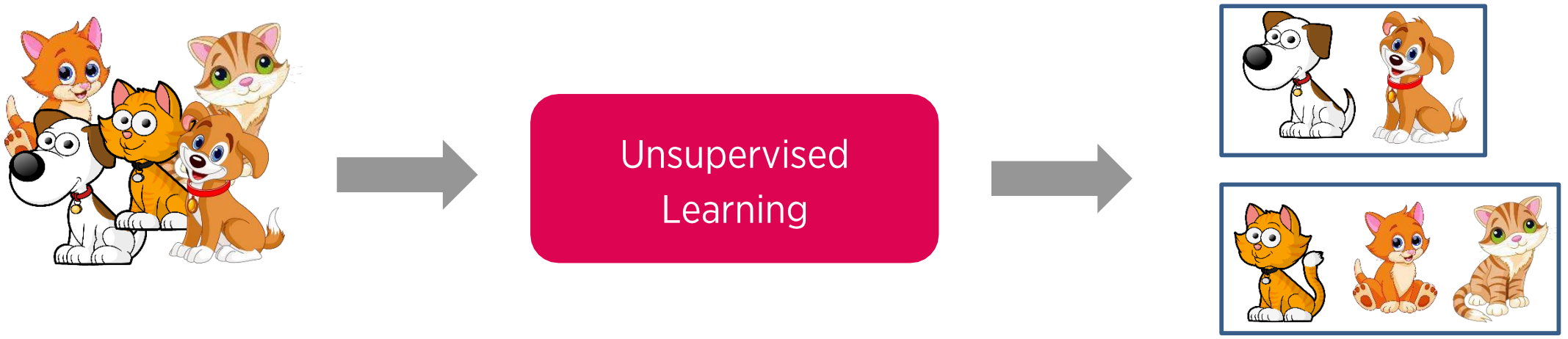
I
Unsupervised learning
Association rules

Unsupervised Learning

- An unsupervised machine learning algorithm makes use of input data without any labels – in other words, no teacher (label) telling the child (computer) when it is right or when it has made a mistake so that it can self-correct.
- Unlike supervised learning that tries to learn a function that will allow us to make predictions given some new unlabeled data, unsupervised learning tries to learn the basic structure of the data to give us more insight into the data

Unsupervised Learning

- For instance, suppose it is given an image having both dogs and cats which have not seen ever. Thus, the machine has no idea about the features of dogs and cats so we can't categorize it in dogs and cats. But it can categorize them according to their similarities, patterns and differences i.e., we can easily categorize the above picture into two parts.
- First may contain all pics having **dogs** in it and second part may contain all pics having **cats** in it. Here you didn't learn anything before , means no training data or examples.



Unsupervised Learning methods

- Clustering
  - K-means clustering
  - Two-step algorithm
  - Kohonen network
  - Anomaly algorithm
- Association rule
  - Apriori algorithm
  - Sequence algorithm

Association rules

- Association rules are if-then statements that help to show the probability of relationships between data items within large data sets in various types of databases. Association rule mining has a number of applications and is widely used to help discover sales correlations in transactional data or in medical data sets.
- Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrence, in a database. It identifies frequent if-then associations, which are called association rules.
- An association rule has two parts: **an antecedent** (if) and **a consequent** (then).

Necessary terms

Term	Description
Antecedents	One or more IF choices
Consequents	One or more THEN subsequent choices
Confidence	% of the time the IF choices result in the THEN choices
Rule support	% of all transactions that have both IF and THEN choices
Antecedent support	% of all transactions that have the IF choices
Lift	How many times more likely a customer is to choose the IF with the THEN than just the IF alone

Apriori algorithm

- The Apriori node discovers association rules in the data. Association rules are statements of the form.

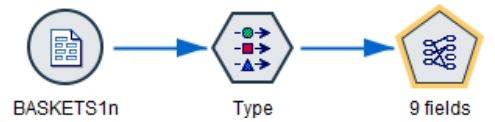
  if **ancedent(s)** then **consequent(s)**

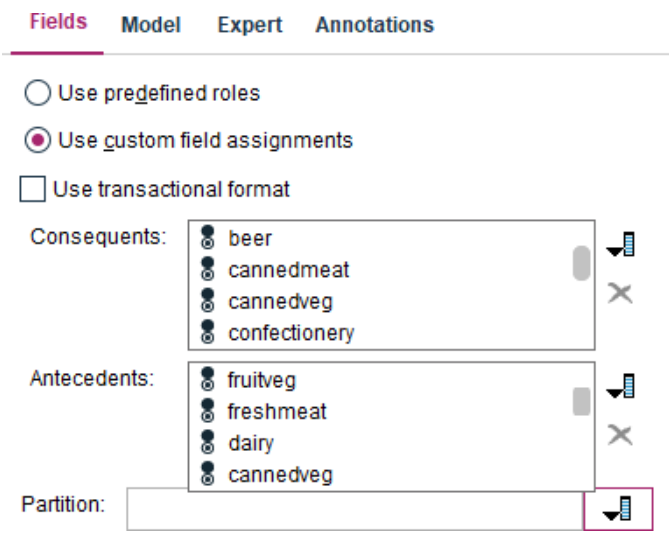
- For example, “if a customer purchases a razor and after shave, then that customer will purchase shaving cream with 80% confidence”. Apriori extracts a set of rules from the data, pulling out the rules with the highest information content. Apriori offers five different methods of selecting rules and uses a sophisticated indexing scheme to efficiently process large datasets.

Market Basket analysis with Apriori algorithm

1. Open “BASKETS1n,txt” ;
2. Add **Type** node and set **Role**:

   fruitveg, freshmeat, ..., confectionary -> **Both**
3. Connect **Apriori** node from **Modelling** palette
   - **Consequents**: fruitveg, freshmeat, ..., confectionary
   - **Antecedents**: fruitveg, freshmeat, ..., confectionary
4. Run





Output

- **Support** displays antecedent support – that is , the proportion of IDs for which the antecedents are true, based on the training data.
- **Confidence** displays the ratio of rule support to antecedent support. This indicates the proportion of IDs with the specified antecedent(s) for which the consequent(s) is/are also true

Consequent	Antecedent	Support %	Confidence %
frozenmeal	beer cannedveg	16.7	87.425
cannedveg	beer frozenmeal	17.0	85.882
beer	frozenmeal cannedveg	17.3	84.393

Sequence algorithm

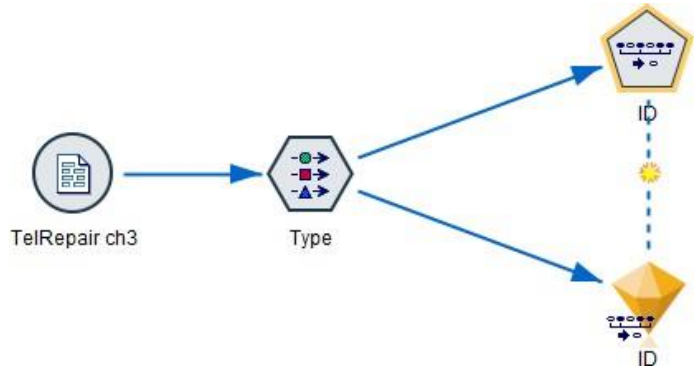
- The **Sequence node** discovers patterns in sequential or time-oriented data, in the format bread -> cheese.
- The elements of a sequence are **item sets** that constitute a single transaction. For example, if a person goes to the store and purchases bread and milk and then a few days later returns to the store and purchases some cheese, that person's buying activity can be represented as two item sets. The first item set contains bread and milk, and the second one contains cheese.
- A **sequence** is a list of item sets that tend to occur in a predictable order.
- The Sequence node detects frequent sequences and creates a generated model node that can be used to make predictions.

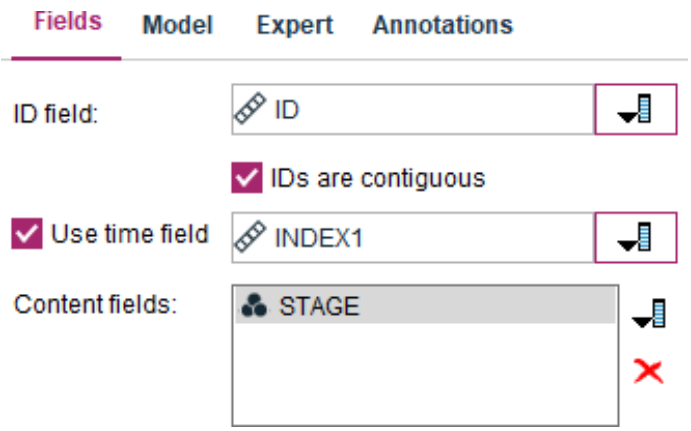
Sequence algorithm

1. Open “TelRepair ch3.txt” ;
2. Connect **Type** node ;
3. Add **Sequence** node from Modelling palette;
4. In the fields -> **Id field:** ID; **Use time field:** INDEX1;

   Content fields: STAGE
5. In the model section > **Minimum rule support**: 1;

   Minimum rule confidence: 75
6. Run





Output

Antecedent	Consequent	Support %	Confidence %
160	299	4.0	100.0
180 185	210	5.6	100.0
150 195	210	9.733	100.0
145 180	210	1.6	100.0
140 160	299	1.333	100.0
190 180	210	3.467	100.0
190 115	210	2.933	100.0
135 120	210	2.8	100.0
190 120	210	4.267	100.0
195 165	210	5.067	100.0
170 185	210	3.2	100.0
150 135	210	2.8	100.0
90 160	299	3.867	100.0
185	210	2.0	100.0

II
Unsupervised learning
CLUSTER ANALYSIS

Data Science
Academy

Cluster Analysis

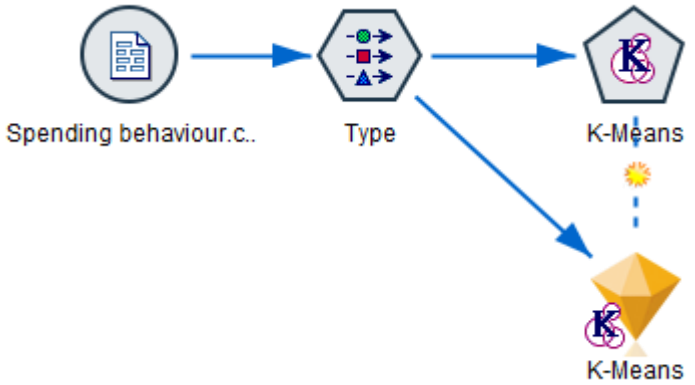
- The cluster analysis groups the population members in homogeneous classes or clusters (or groups). The researcher does not know the groups in advance; they will be created during the clustering procedure
- The cluster analysis technique computes the distances between cases and then builds a similarity or proximity matrix based on those distances. The cases that are close to one another are included in the same cluster.
- The members of every cluster should be similar to one another and different from the members of the other clusters.

K-Means Clustering

- The K-Means cluster (also called disjoint cluster) can be used when we have a big number of sample cases.
- For this analysis, the researcher establishes the number of clusters from the beginning and the program generates one final solution with that number of classes.
- The K-Means cluster can only work with continuous or ordinal variables.
- Furthermore, it is very advisable to standardize the initial variables before running the procedure, to make the interpretation easier.

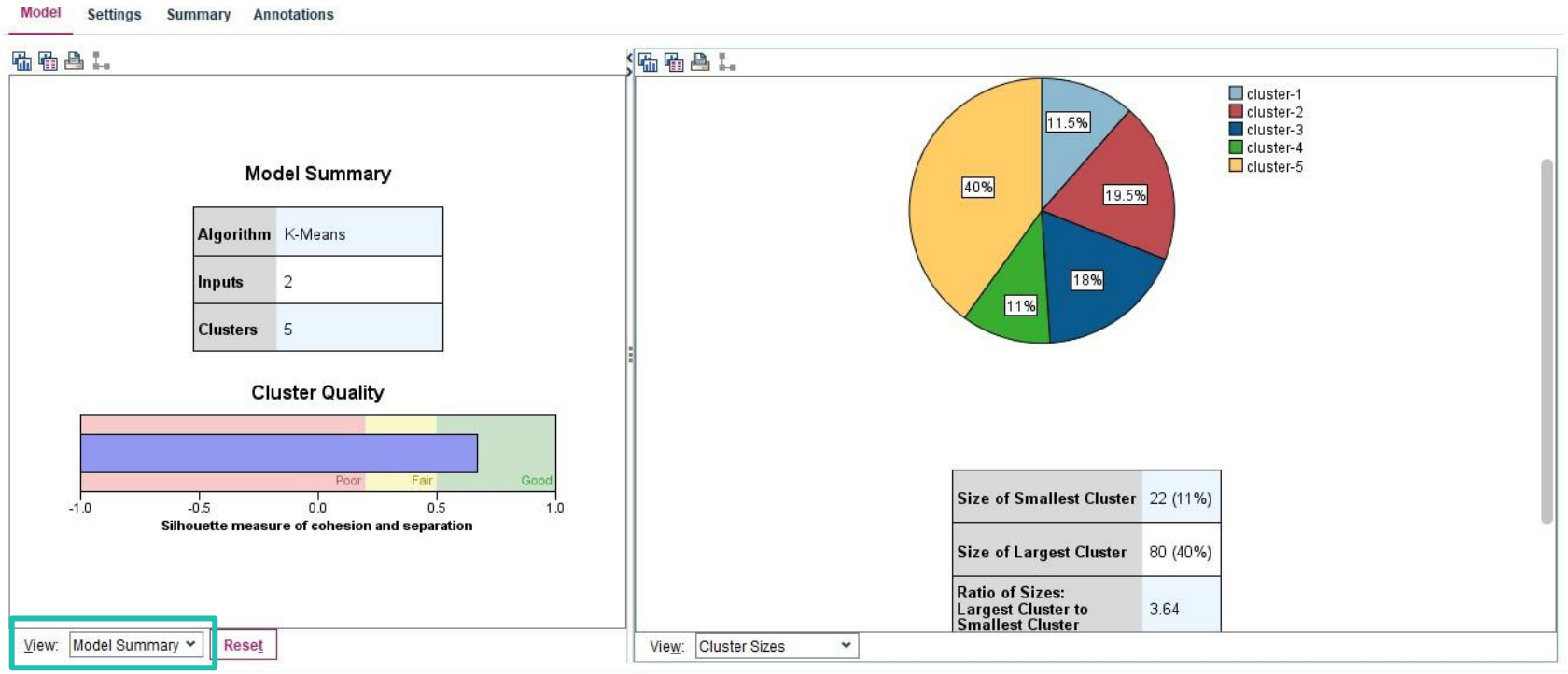
Customer segmentation using K-Means algorithm

1. Open “Spending behavior.csv” ;
2. Connect **Type** node;
3. Connect **K-Means** node from **Modeling** palette;
4. **Fields** -> **Inputs** -> Annual income, Spending_score
5. **Model** -> **Number of clusters**: 5
6. Run



Customer segmentation using K-Means algorithm

1. Change **View** section to **Clusters**;
2. Select all Cluster names by ctrl



Customer segmentation using K-Means algorithm

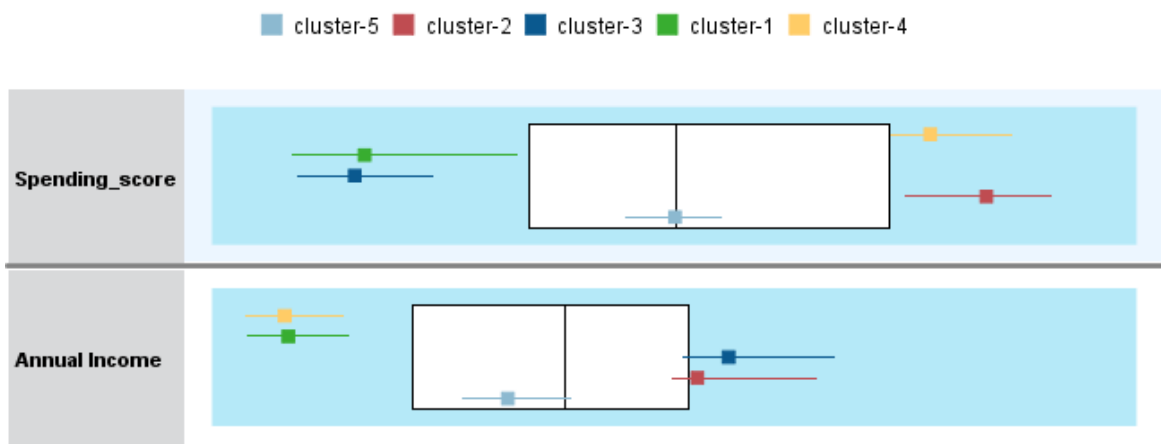
Clusters

Input (Predictor) Importance

1.0 0.8 0.6 0.4 0.2 0.0

Cluster	cluster-5	cluster-2	cluster-3	cluster-1	cluster-4
Label					
Description					
Size	40.0% (80)	19.5% (39)	18.0% (36)	11.5% (23)	11.0% (22)
Inputs	Spending_score 49.71	Spending_score 82.13	Spending_score 17.58	Spending_score 20.91	Spending_score 79.36
	Annual Income 55,087.50	Annual Income 86,538.46	Annual Income 87,750.00	Annual Income 26,304.35	Annual Income 25,727.27

Cluster Comparison



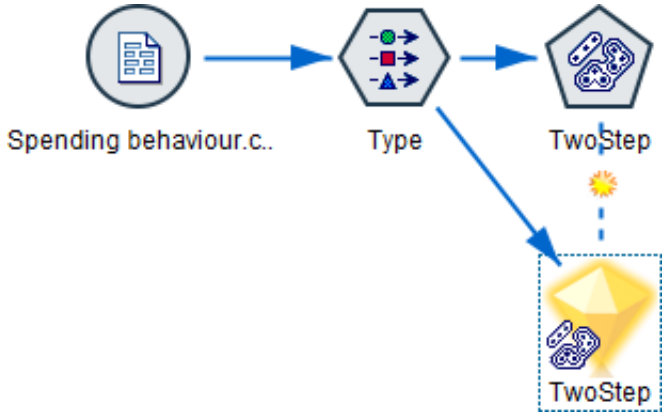
Two-Step algorithm

- The Two-Step Cluster node provides a form of cluster analysis. It can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Two-Step Cluster models do not use a target field.

- Two-Step Cluster is a two-step clustering method. Two-Step Cluster analysis identifies groupings by running pre-clustering first and then by running hierarchical methods. Two-Step clustering can handle scale and ordinal data in the same model, and it automatically selects the number of clusters.

Customer segmentation using Two-Step algorithm

1. Open “Spending behavior.csv” ;
2. Connect **Type** node;
3. Connect **Two-Step** node from **Modeling** palette;
4. **Fields** -> **Inputs** -> Annual income, Spending_score
5. Run



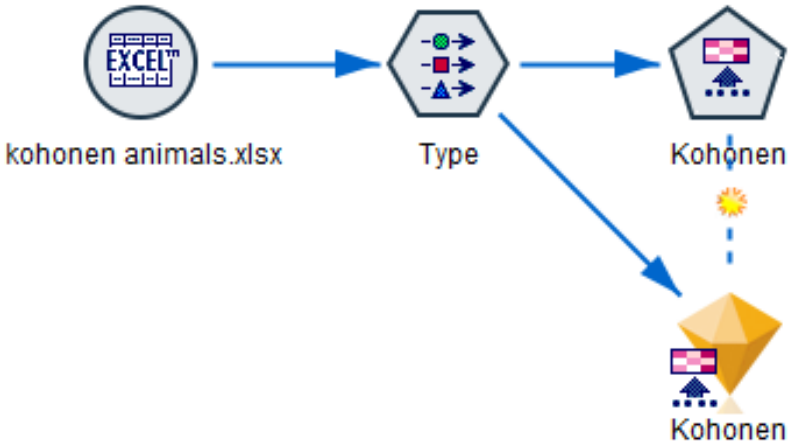
Kohonen Network

- Kohonen networks are a type of neural networks that perform clustering, also known as a **knet** or a **self-organizing map**. This type of network can be used to cluster the dataset into distinct groups when you don't know what those groups are at the beginning. Records are grouped so that records within a group or cluster tend to be similar to each other, and records in different groups are dissimilar.

- The basic units are **neurons**, and they are organized into two layers; the **input layer** and the **output layer** (also called the **output map**). All of the input neurons are connected to all of the output neurons, and these connections have **strengths**, or **weights**, associated with them. During training , each unit competes with all of the others to "win" each record.

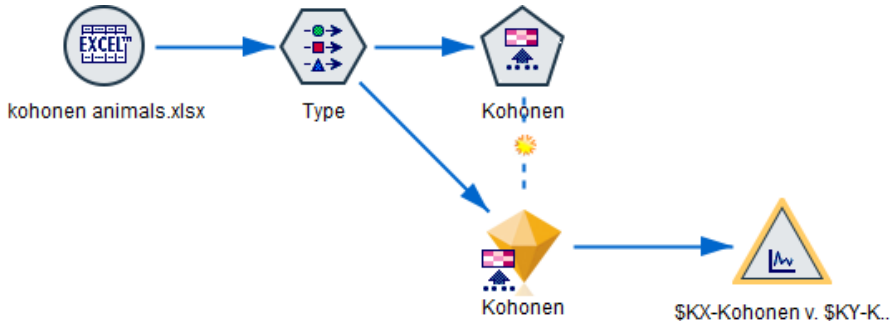
Finding similar animals using Kohonen (Self Organizing Map)

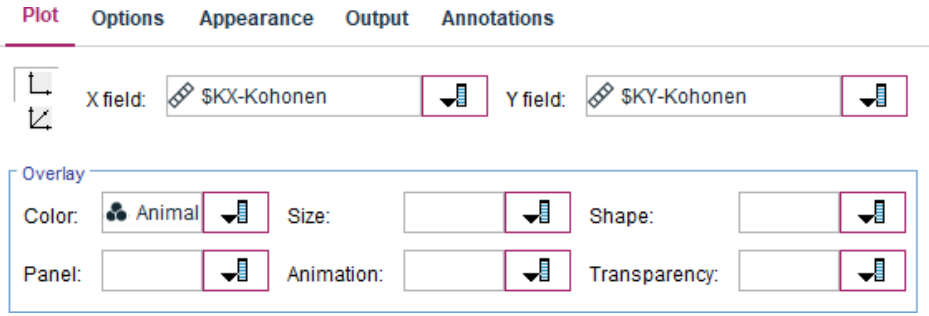
1. Open “kohonen animals.xlsx” ;
2. Connect **Type** node;
3. Connect **Kohonen** node from **Modeling** palette;
4. **Fields** -> **Inputs** -> All variables excluding *Animal*
5. Run



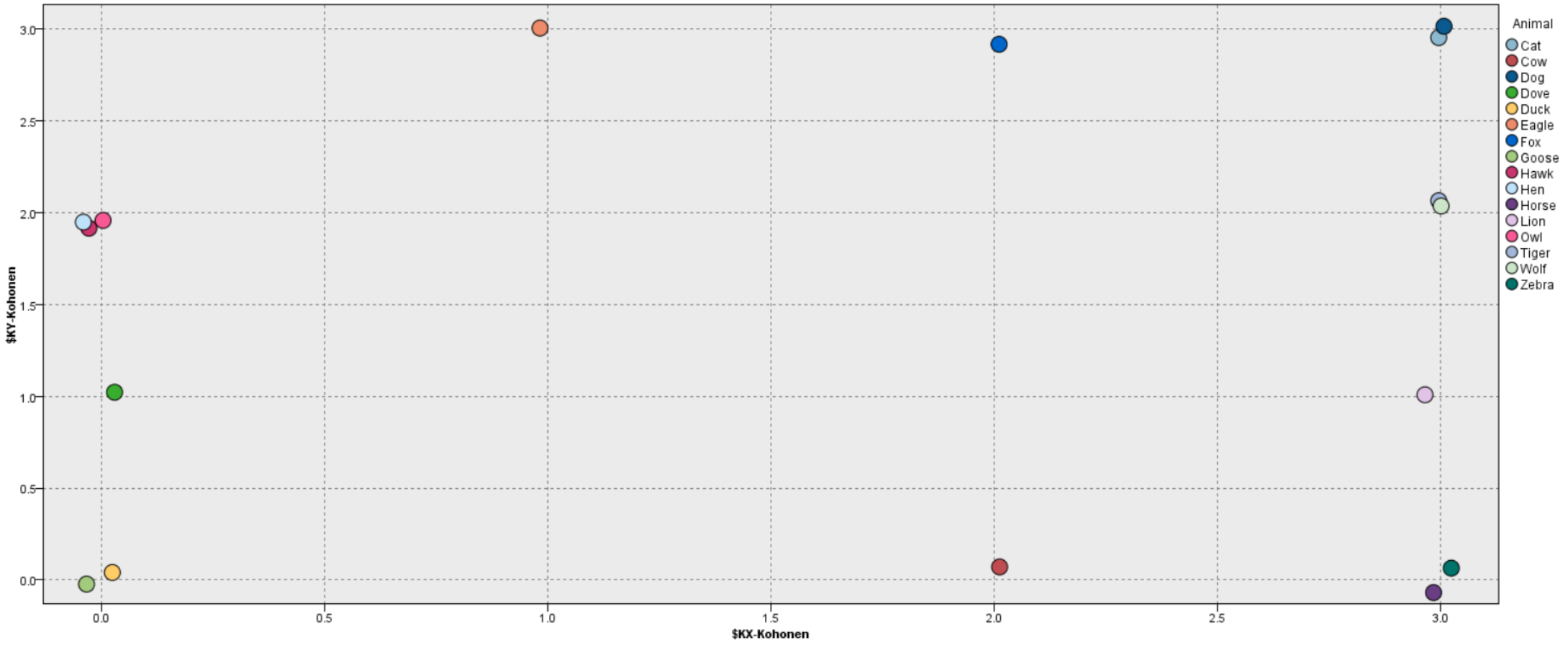
Mapping results of Kohonen algorithm

1. Connect **Plot** node to **Model Nugget**;
2. **X field**: KX-Kohonen
3. **Y field**: KY-Kohonen
4. **Color**: Animal
5. Activate **Jitter** in the **Options** tab
6. Run





Output



OK

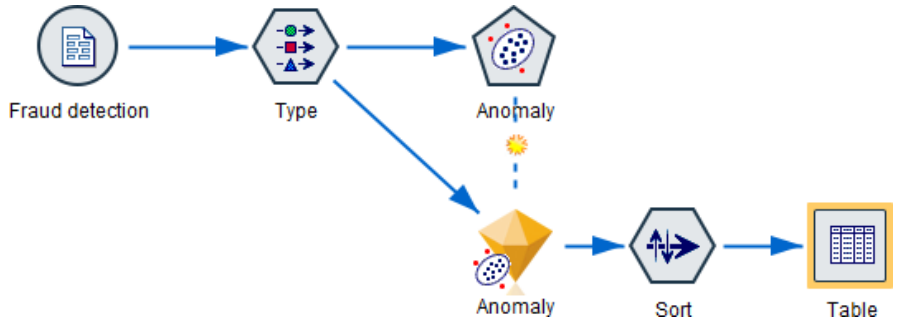
Anomaly algorithm

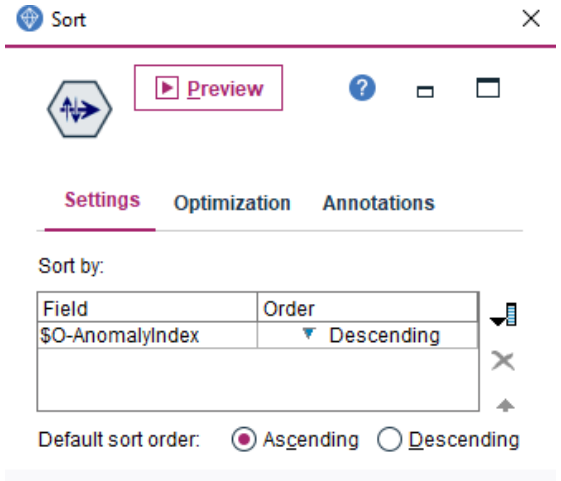
- Anomaly detection models are used to identify outliers, or unusual cases, in the data. Unlike other modeling methods that store rules about unusual cases, anomaly detection models store information on what normal behavior looks like. This makes it possible to identify outliers even if they do not conform to any known pattern, and it can be particularly useful in applications, such as fraud detection, where new patterns may constantly be emerging. Anomaly detection is an unsupervised method, which means that it does not require a training dataset containing known cases of fraud to use as a starting point.

- While traditional methods of identifying outliers generally look at one or two variables at a time, anomaly detection can examine large numbers of fields to identify clusters or peer groups into which similar records fall. Each record can then be compared to others in its peer group to identify possible anomalies. The further away a case is from the normal center, the more likely it is to be unusual. For example, the algorithm might lump records into three distinct clusters and flag those that fall far from the center of any one cluster.

Fraud detection using Anomaly algorithm

1. Open “Fraud detection.csv” ;
2. Connect **Type** node;
3. Add **Anomaly** node from **Modelling** palette
4. Add **Sort** node from **Record Ops**
5. Sort by -> $O-Anomaly index; Order Descending
6. Add **Table** node
7. Run





Output

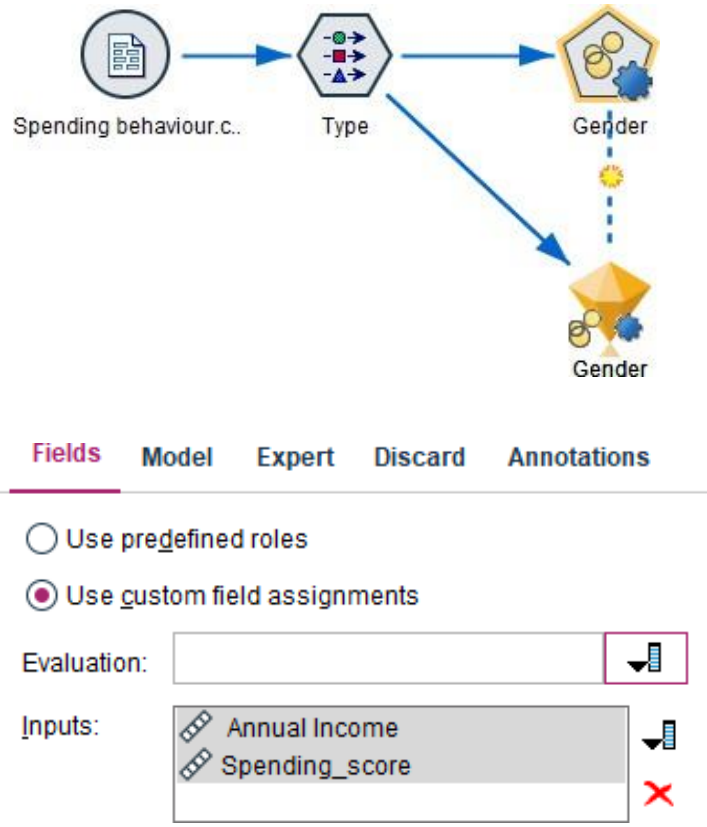
$O-Anomaly	$O-AnomalyIndex	$O-PeerGroup	$O-Field-1	$O-FieldImpact-1	$O-Field-2	$O-FieldImpact-2	$O-Field-3	$O-FieldImpact-3
T	1387.823	2	num_root	0.387	num_compromised	0.384	su_attempted	0.120
T	1102.465	2	num_root	0.380	num_compromised	0.380	su_attempted	0.151
T	929.091	2	src_bytes	0.999	service	0.000	flag	0.000
T	675.861	2	urgent	0.931	num_failed_logins	0.065	dst_bytes	0.001
T	333.275	2	su_attempted	0.500	num_compromised	0.185	num_root	0.183
T	325.602	2	su_attempted	0.512	num_failed_logins	0.298	dst_bytes	0.112
T	319.470	2	urgent	0.987	num_access_files	0.006	dst_host_count	0.002
T	295.244	2	su_attempted	0.564	num_root	0.159	num_compromised	0.157
T	277.752	2	num_failed_logins	0.915	dst_host_srv_diff_host_rate	0.078	dst_host_count	0.002
T	244.203	2	num_file_creations	0.860	num_access_files	0.131	duration	0.002
T	228.380	2	su_attempted	0.729	root_shell	0.105	num_access_files	0.079
T	171.819	2	num_file_creations	0.988	duration	0.004	logged_in	0.002
T	169.278	2	num_failed_logins	0.992	diff_srv_rate	0.003	service	0.001
T	162.319	2	num_compromised	0.396	num_root	0.312	su_attempted	0.268
T	135.201	2	num_file_creations	0.986	duration	0.004	logged_in	0.002
T	126.702	2	num_file_creations	0.963	is_guest_login	0.015	hot	0.012
T	123.952	2	num_file_creations	0.896	dst_host_srv_diff_host_rate	0.079	duration	0.007
T	118.655	2	num_shells	0.723	root_shell	0.202	num_file_creations	0.039
T	118.584	2	num_shells	0.724	root_shell	0.202	num_file_creations	0.039
T	118.066	2	num_shells	0.727	root_shell	0.203	num_file_creations	0.039

Auto Cluster node

- The Auto Cluster node estimates and compares clustering models, which identify groups of records that have similar characteristics. The node works in the same manner as other automated modeling nodes, allowing you to experiment with multiple combinations of options in a single modeling pass. Models can be compared using basic measures with which to attempt to filter and rank the usefulness of the cluster models, and provide a measure based on the importance of particular fields.

Building several clustering models using Auto Cluster

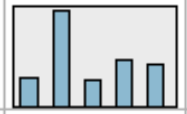
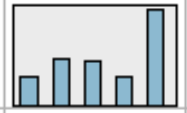
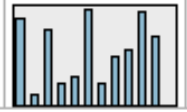
1. Open “Spending behavior.csv” ;
2. Connect **Type** node;
3. Connect **Auto Cluster** node from **Modeling** palette
4. Use custom field assignments -> **Inputs**: Annual Income, Spending_score
5. Run



Resulting Model nugget of Auto Cluster

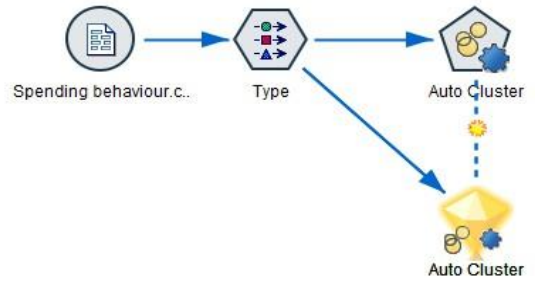
Model | Summary | Annotations

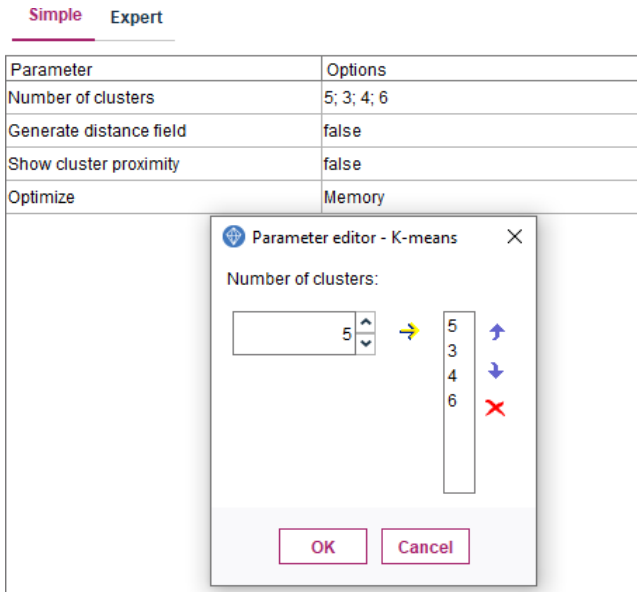
Sort by: Use | Ascending | Descending | Delete Unused Models | View: Training set

Use?	Graph	Model	Build Time (mins)	Silhouette	Number of Clusters	Smallest Cluster (N)	Smallest Cluster (%)	Largest Cluster (N)	Largest Cluster (%)	Smallest/ Largest	Importance
☑		TwoStep 1	< 1	0.670	5	22	11	81	40	0.272	0.0
☐		K-means 1	< 1	0.669	5	22	11	80	40	0.275	0.0
☐		Kohonen 1	< 1	0.423	11	3	1	32	16	0.094	0.0

Building different sized K-Means models using Auto Cluster

1. Open “Spending behavior.csv” ;
2. Connect Type node;
3. Connect Auto Cluster from Modeling palette;
4. Use custom field assignments -> Inputs: Annual Income, Spending_score;
5. Expert -> Select only K-Means;
6. Model parameters: Default -> Specify;
7. Number of clusters: Specify:3; 4; 5; 6;
8. OK
9. Run





III
Deployment

Data Science
Academy

Deployment

- Deployment is where developed and validated models are put into use to impact decisions and improve business outcomes.
- Deploying models into applications can require complex time – consuming coding, SPSS Modeler simplifies this process with code free deployment.
- The Modeler platform allows for many different deployment modes including ad-hoc, batch, real-time and streaming.
- In the simplest case we would run a model manually on new data and write the resulting scores to a file that we distribute through existing workflow mechanisms.

Deployment Planning

A successful deployment of data mining results requires that the right information reaches the right people.

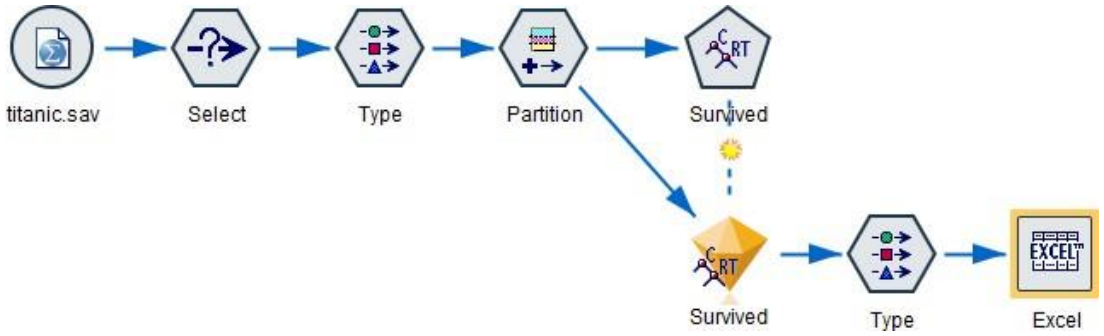
- Decision makers. Decision makers need to be informed of the recommendations and proposed changes to the site and provided with short explanations of how these changes will help.
- Web developers. People who maintain the Web site will have to incorporate the new recommendations and organization of site content. Inform them of what changes could happen because of future studies, so they can lay the groundwork now.
- Database experts. The people who maintain the customer, purchase and product databases should be kept apprised of how the information from the databases is being used and what attributes may be added to the databases in future projects.

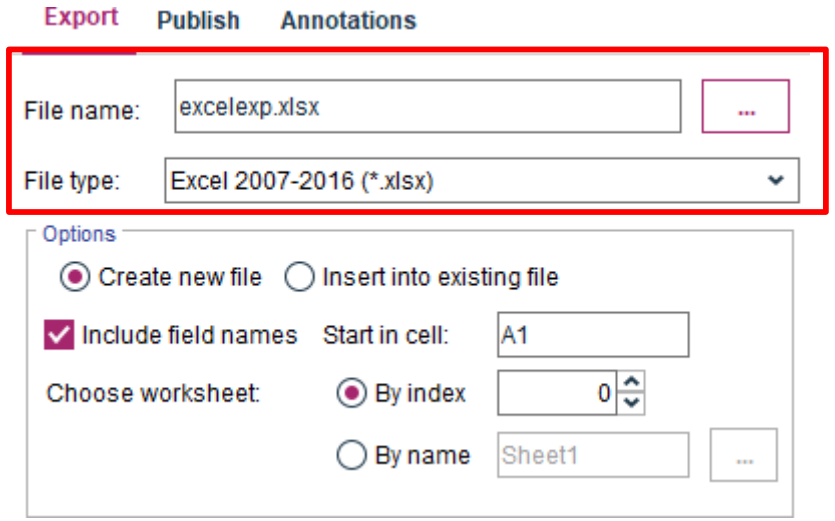
Export Palette

Buttons	Description
Database 	The Database export node writes data to an ODBC-compliant relational data source. In order to write to an ODBC data source, the data source must exist and you must have written permission for it.
Flat File 	The Flat File export node outputs data to a delimited text file. It is useful for exporting data that can be read by other analysis or spreadsheet software.
Statistics Export 	The Statistics Export node outputs data in IBM SPSS Statistics *.sav* format. The *.sav* files can be read by SPSS Statistics Base and other products.
Data Collection 	The IBM SPSS Data Collection export node outputs data in the format used by Data Collection market research software. The Data Collection Data Library must be installed to use this node.
SAS 	The SAS export node outputs data in SAS format, to be read into SAS or a SAS-compatible software package
Excel 	The Excel export node outputs data in Microsoft Excel format (*.xls*). Optionally, you can choose to launch Excel automatically and open the exported file when the node is executed.
XML Export 	The XML export node outputs data to a file in XML format. You can optionally create an XML source node to read the exported data back into the stream.

Exporting Model Results

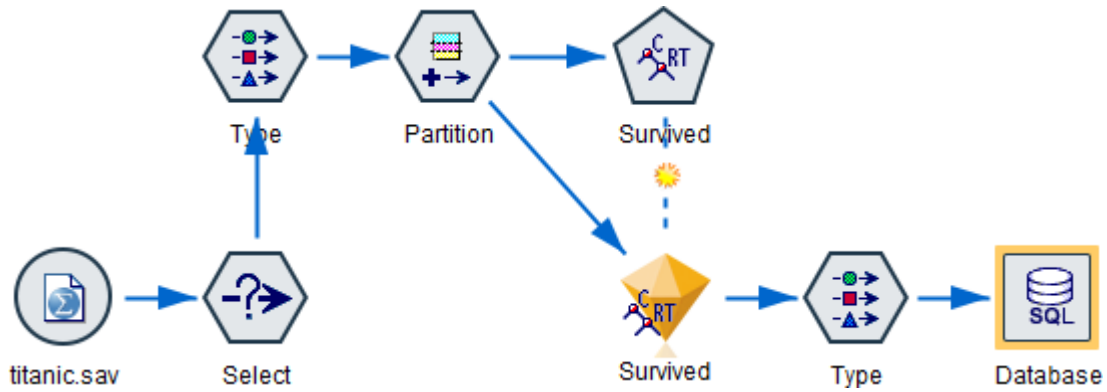
1. Build **C&RT model** on "**titanic.sav**" dataset as shown in stream
2. Add **Type** to Modelling Nugget (without Type node export nodes will not work).
3. Add **Excel** export node from **Export palette.**
4. Edit name and destination of file name box.

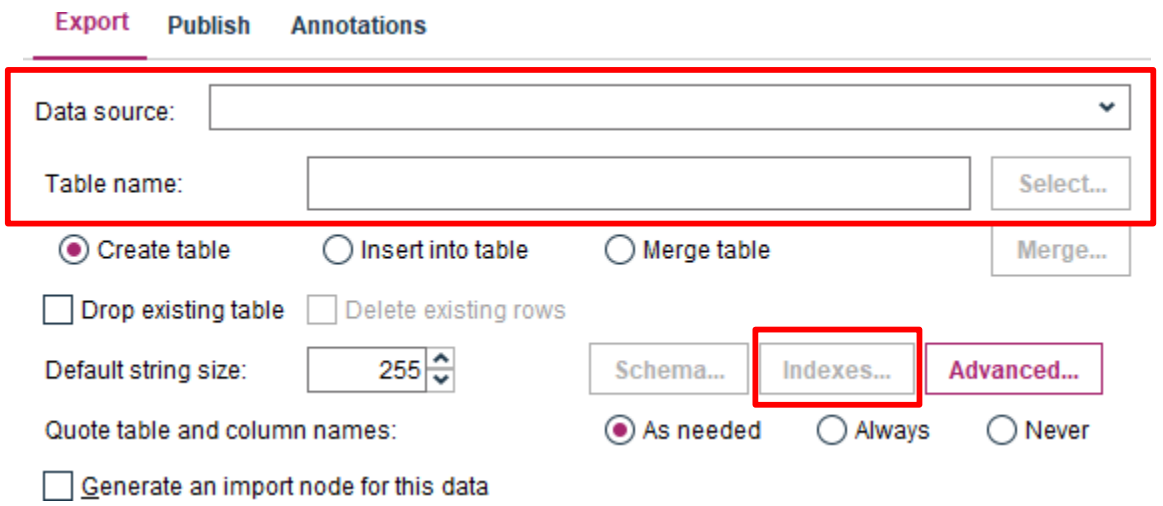




Export of scored data to SQL Server

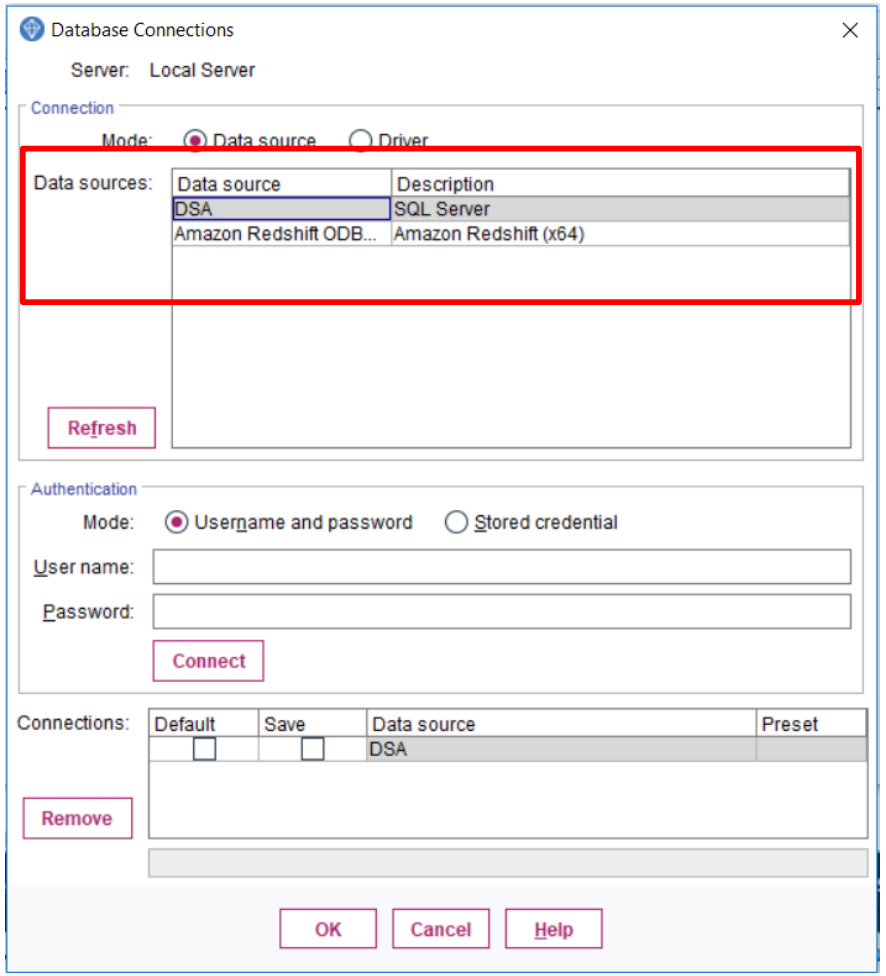
1. Build **C&RT model** on “titanic.sav” dataset as shown in stream
2. Add **Type** to Modeling Nugget (without Type node export nodes will not work).
3. Add **Database export** node from **Export palette**;
4. Select <**Add new database connection**> in **Data Source** section;





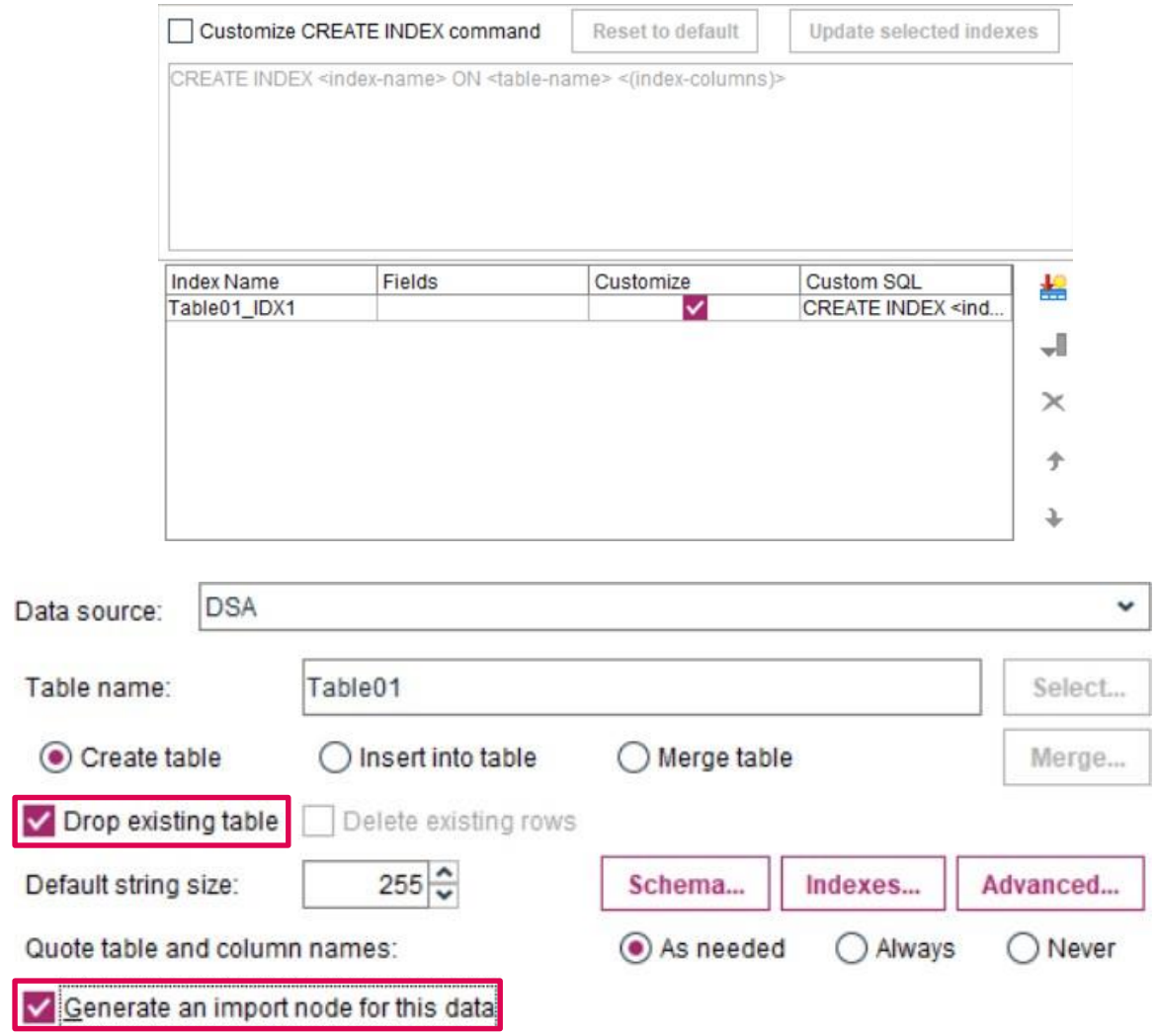
Export of scored data to SQL Server

1. Select your **Data Source** name
2. Click **OK**
3. Type **Table**



Export of scored data to SQL Server

1. In order to change storage types of fields click **Indexes** button and define appropriate SQL storage types. (For Azerbaijani alphabet texts you can use nvarchar, nchar, ntext and etc., storage types);
2. Enable **Drop existing table** while second time you need to overwrite table;
3. Enable **Generate an import node** option for testing export results.

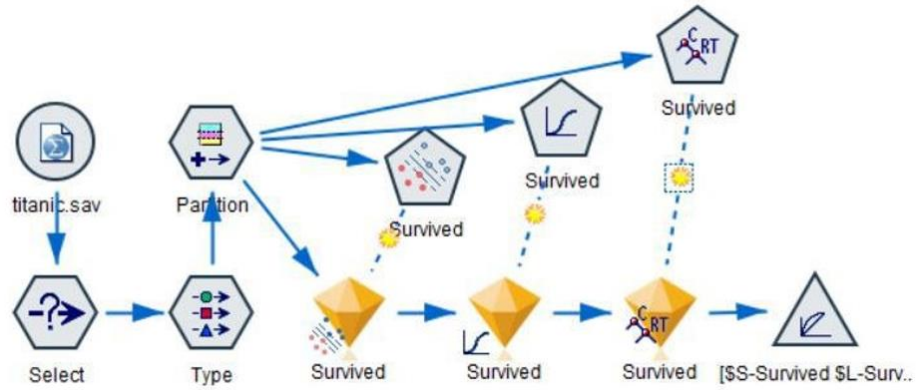


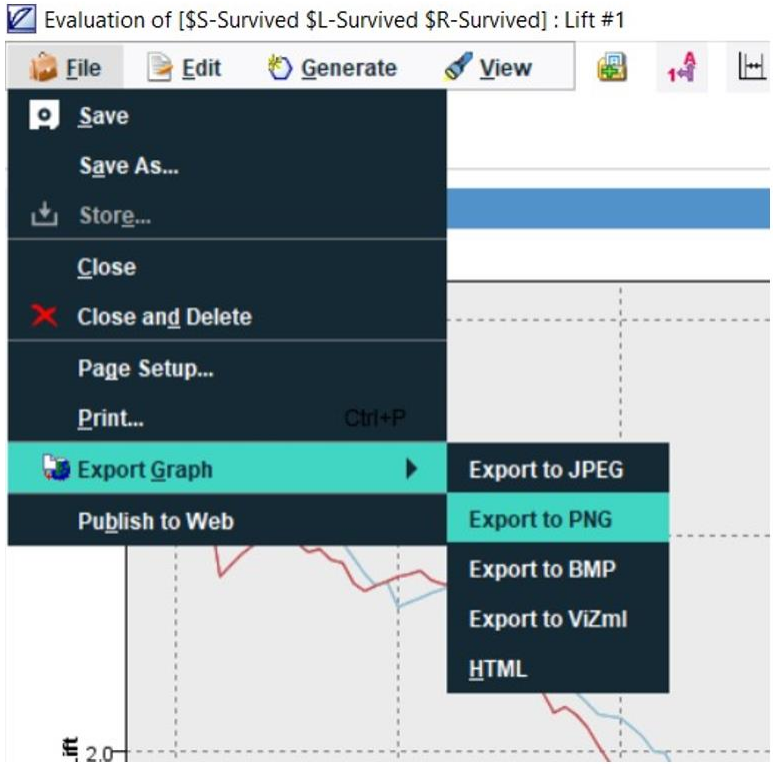
Preparing a Final Presentation

- In addition to the project report, you may also need to present the project findings to a team of sponsors or related departments. If this is the case, you could use much of the same information in your report but presented from a broader perspective. The charts and graphs in IBM SPSS Modeler can easily be exported for this type of presentation.

Exporting Graphs

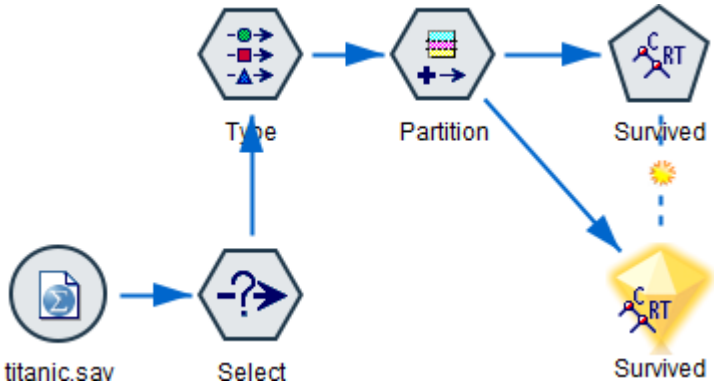
- Open **ensemble dataset**
- Select **File** button from **Menu bar**
- Choose **Export Graph**
- Export to PNG

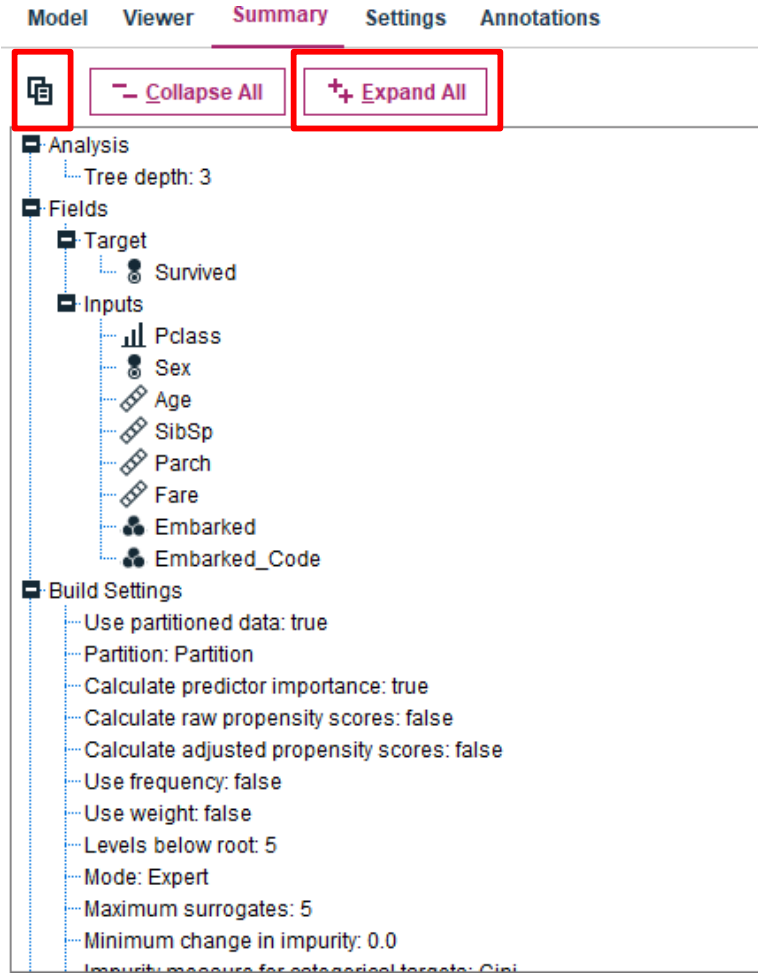




Exporting model parameters

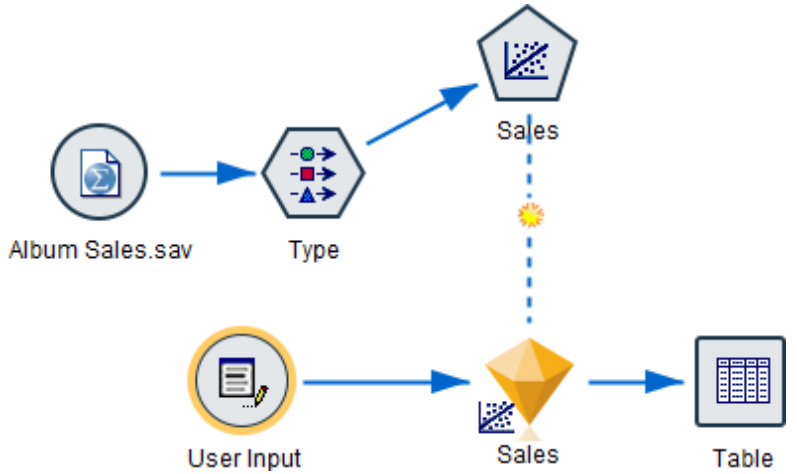
1. Build **C&RT model** on "**titanic.sav**" dataset as shown in stream
2. Double click on **Modeling nugget** ;
3. Click **Expand All** in **Summary tab**;
4. Click **copy to clipboard** button on top right.





Scoring Linear regression inside Modeler

1. Build **Regression model of "Album Sales.sav"** as shown in stream
2. Add **User Input** node from **Source palette**;
3. Type in **Fields:** *Adverts, Airplay, Attract, Sales*
4. Select **Storage**: *Real* for all of them
5. Write appropriate numbers in Values section;
6. Make connection as shown in stream
7. Add **Table** node to see scored data
8. Run



Data | Filter | Types | Annotations

Field	Storage	Values
Adverts	Real	10 15 250
Airplay	Real	42 18 74
Attract	Real	10 9 9
Sales	Real	

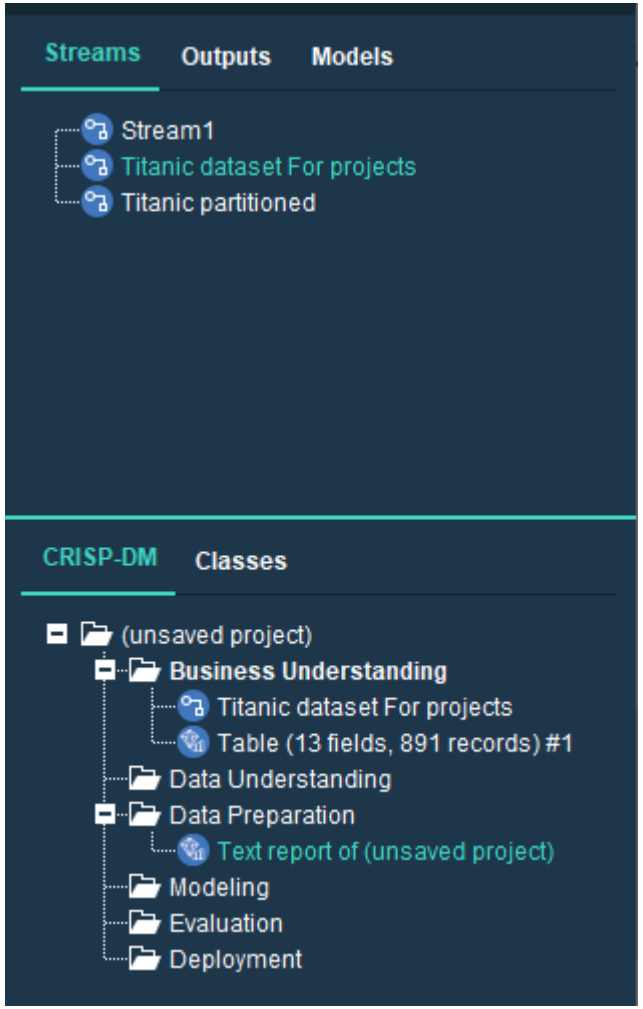
Reporting in SPSS Modeler

- One of the most useful features of projects is the ability to generate reports based on the project items and annotations. This is a critical component of effective data mining, as discussed throughout the CRISP-DM methodology. You can generate a report directly into one of several file types or to an output window on the screen for immediate viewing. From there, you can print, save, or view the report in a web browser. You can distribute saved reports to others in your organization.

- Reports are often generated from project files several times during the data mining process for distribution to those involved in the project. The report culls information about the objects referenced from the project file as well as any annotations created. You can create reports based on either the Classes view or CRISP-DM view.

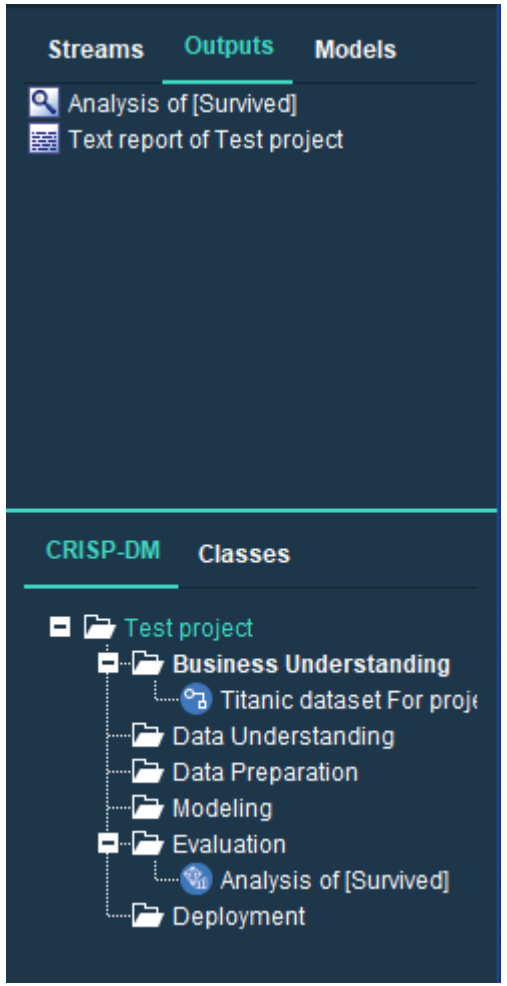
Working with CRISP-DM folder

1. Right Click any stream name that you want to add to Project folder and choose add to project option.
2. If you haven't saved stream save it and it will automatically be added to project folder
3. Go to Outputs tab and right click any output and choose add to project option
4. Apply same procedures to Models tab.
5. Right click on project folder and save it.



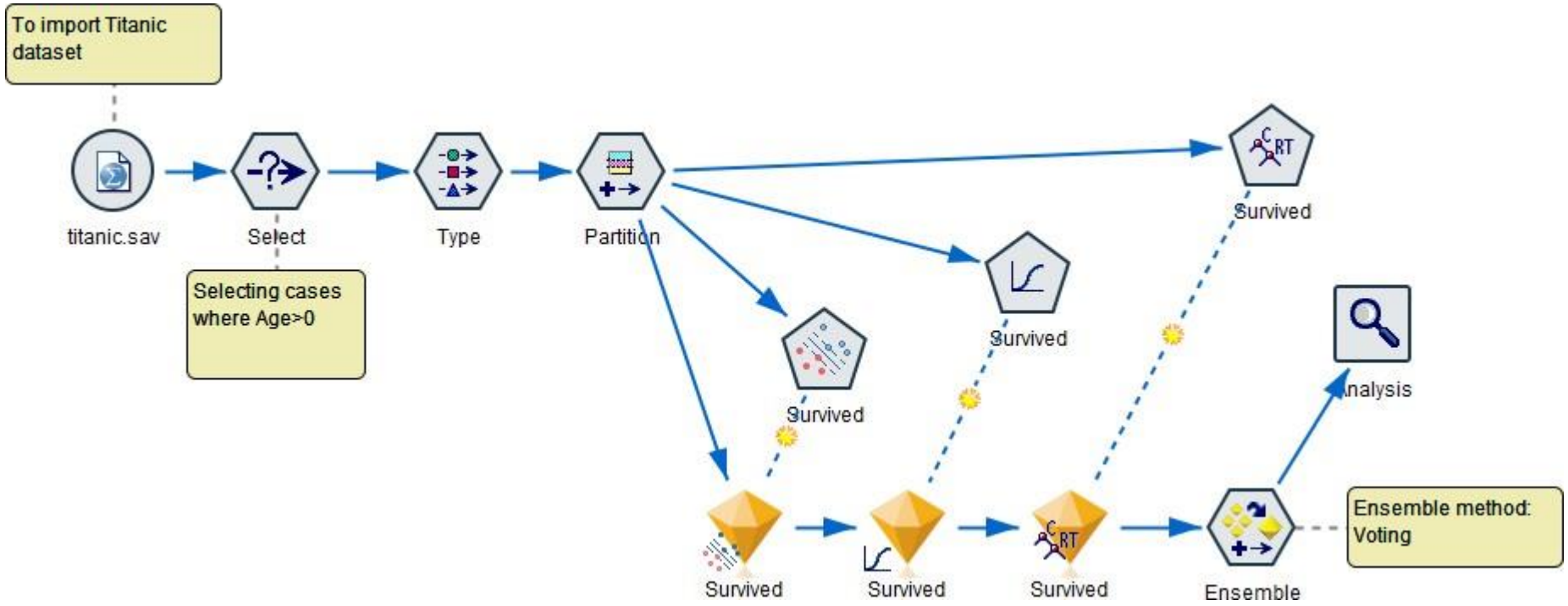
Project Reporting

1. Right click on **Project name**
2. Choose **Project report**
3. Select **Generate report** in opened window



Commenting for understandable streams

1. To add comment to any node right click node
2. Choose new comment and type



Conducting a Final Project Review

You should conduct a brief interview with those significantly involved in the data mining process. Questions to consider during these interviews include the following:

- What are your overall impressions of the project?
- What did you learn during the process-both about data mining in general and the data available?
- Which parts of the project went well? Where did difficulties arise? Was there information that might have helped ease the confusion?

After the data mining results have been deployed, you might also interview those affected by the results such as customers or business partners. Your goal here should be to determine whether the project was worthwhile and offered benefits it set out to create.

Thank you!

Data Science
Academy