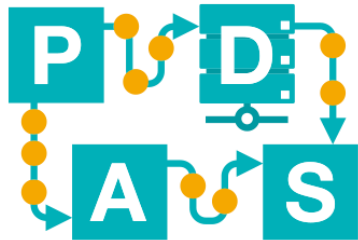


Data Mining

Decision Trees, Clustering, Association Rule

Harry Beyel

BPI-Instruction2



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Pen-and-paper Exercises

Decision Tree

Exercise 1

- Consider the following training dataset. Suppose *Outcome* is our response variable.
- Provide a decision tree that predicts the outcome. Use the information gain criterion for splitting.

District	Income	Previous Customer	Outcome
Suburban	High	No	Not responded
Suburban	High	Yes	Not responded
Rural	High	No	Responded
Urban	High	No	Responded
Urban	Low	No	Responded
Urban	Low	Yes	Not responded
Rural	Low	Yes	Responded
Suburban	High	Yes	Not responded
Suburban	Low	No	Responded
Urban	Low	No	Responded
Suburban	Low	Yes	Responded
Rural	High	Yes	Responded
Rural	Low	No	Responded
Urban	High	Yes	Not responded

Clustering

Exercise 2

	att_1	att_2
1	2	10
2	2	5
3	8	4
4	5	8
5	7	5
6	6	4
7	1	2
8	4	9

Provide three clusters for the data instances on the left side. The distance function is Euclidean distance. Suppose initially we assign $(2, 10)$, $(5, 8)$, and $(1, 2)$ as the center of each cluster, respectively. Use the *K-means* algorithm to find the clusters.

Clustering

Exercise 2

Algorithm: *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

- (1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
- (2) **repeat**
- (3) (re)assign each object to the cluster to which the object is the most similar, based on the mean value of the objects in the cluster;
- (4) update the cluster means, that is, calculate the mean value of the objects for each cluster;
- (5) **until** no change;

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{ip} - x_{jp})^2}$$
$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \dots + |x_{ip} - x_{jp}|$$

Association rules

Exercise 3

- Consider the following dataset:

$$D = [\{A\}^{50}, \{A, B\}^{30}, \{A, B, C\}^{20}, \{A, D\}^{20}, \{A, F, C\}^5]$$

- Consider *minimum support count* = 22. Draw the FP-tree after removing the infrequent items.
- Find the frequent itemsets (without calculating the conditional FP-trees)

Association rules

Exercise 4

1. Compute *support*, *confidence* and *lift* for the following rules:

- $\{A, B\} \Rightarrow \{E\}$
- $\{A\} \Rightarrow \{C\}$

2. Which of the following rules satisfies the following conditions:

- $Support \geq 0.5$
- $Confidence \geq 0.8$
- $0.9 < lift < 1.1$

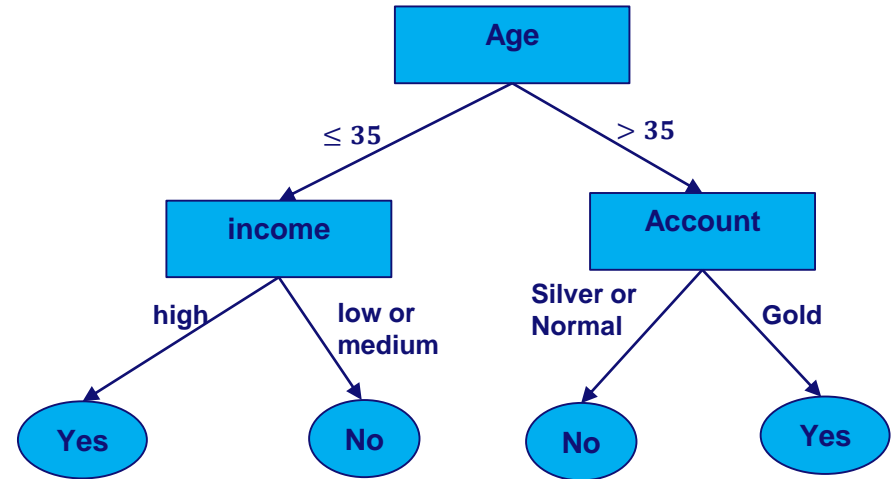
TID	Data items	frequency
1	A,B,E	10
2	C,A,D	25
3	C,B,D	15
4	C,A,B,E	20

Evaluation

Exercise 5 – part 1

Create the confusion matrix for the following classifier model using the test data in the table:

ID	Age	Account type	Income	Accepted (actual class)
1	35	Silver	high	Yes
2	63	Gold	high	No
3	42	Normal	high	No
4	30	Normal	medium	No
5	35	Gold	low	Yes
6	56	Gold	high	Yes
7	23	Normal	low	No
8	48	Gold	medium	No



Evaluation

Exercise 5 – part 1

- Confusion Matrix and performance measures

		predicted		
		yes	no	
target	yes	TP	FN	P
	no	FP	TN	N
		P'	N'	$P + N$

Measure	Formula
<i>accuracy</i>	$\frac{TP + TN}{TP + TN + FP + FN}$
<i>misclassification</i>	$1 - \text{accuracy}$
<i>recall</i>	$\frac{TP}{TP + FN}$
<i>precision</i>	$\frac{TP}{TP + FP}$
<i>F1 – measure</i>	$2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

Evaluation

Exercise 5 – part 2

Calculate *precision, recall, accuracy* **and** *F1 – measure* **based on the confusion matrix.**

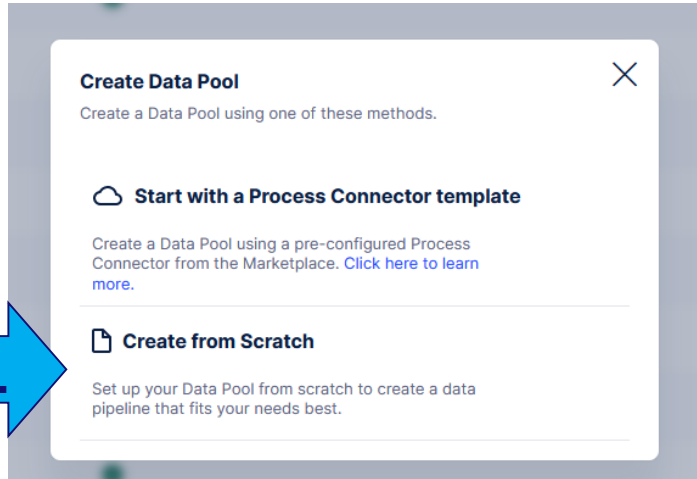


Creating Case-Situation Tables in Celonis

Create a Data Pool


The screenshot shows the Data Science Studio interface. On the left, a sidebar contains a menu with 'Data' highlighted. A blue arrow labeled '1.' points to this menu. A dropdown menu is open under 'Data', showing 'Data Integration', 'Machine Learning', and 'Task Mining'. A blue arrow labeled '2.' points to 'Data Integration'. In the top right corner, there is a search bar and a '+ New Data Pool' button. A blue arrow labeled '3.' points to this button. The main area displays a table of data pools with columns: Status, Data Connections, Created By, and Last Execution. The table contains several rows, each with a green status dot and a menu icon.

Create a Data Pool




Create Data Pool ✕

Create a Data Pool using one of these methods.

 **Start with a Process Connector template**

Create a Data Pool using a pre-configured Process Connector from the Marketplace. [Click here to learn more.](#)

 **Create from Scratch**

Set up your Data Pool from scratch to create a data pipeline that fits your needs best.

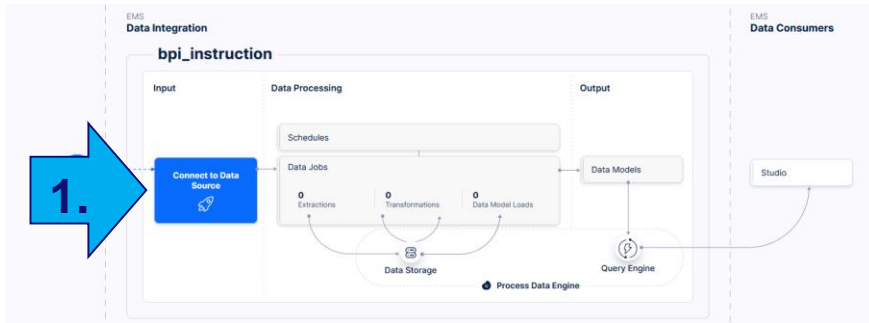


New Data Pool

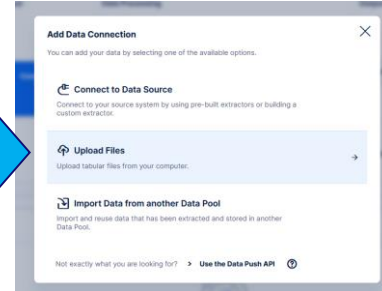
bpi_instruction

Cancel Save

File Upload

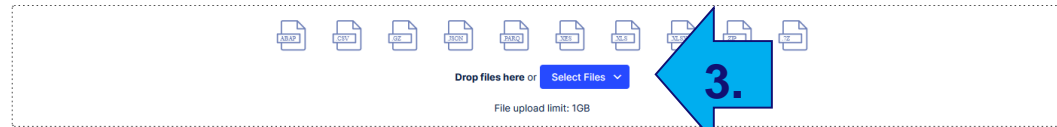


2.



File Uploads

[Go To SAP ABAP Generator](#)



Files

Filters: Status

Table Name	Type	Data Connection	Date	File Count	Status	Errors	Action
------------	------	-----------------	------	------------	--------	--------	--------

File Upload

Files

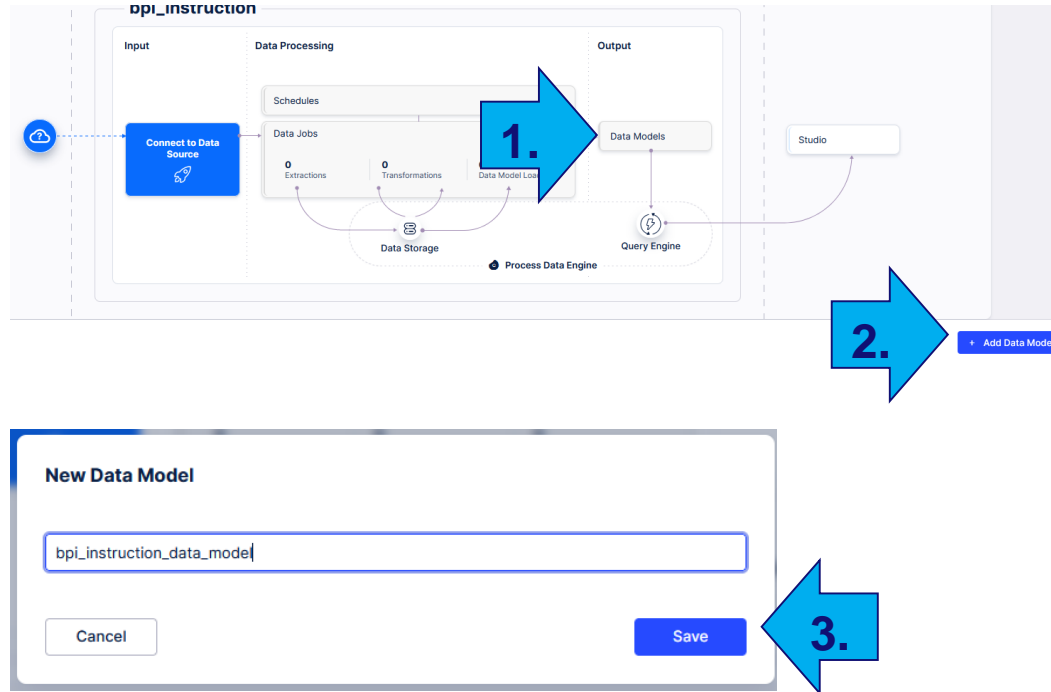
Filters: Status ▾

Table Name	Type	Data Connection	Date	File Count	Status	Errors	Action
Case_Table	Flat File	[Global]	05-04-2023 11:33	1			 
Activity_Table	Flat File	[Global]	05-04-2023 11:33	1			 



2.) Return to overview

Data Model



Data Model



Tables

Available Items (2)

Select all

Search

Activity_Table

Case_Table

+

+

Selected Items (0)

Clear all

Search

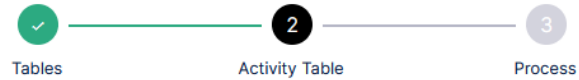
There are no selected items. Add some from the list of available items.



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Data Model



Activity Table

This table contains all information about the activities of your event log (activity name column, case ID column, timestamp column and optionally a sorting and end timestamp column).



TABLE NAME

Activity_Table

Case_Table



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Data Model

1.

2.

3.

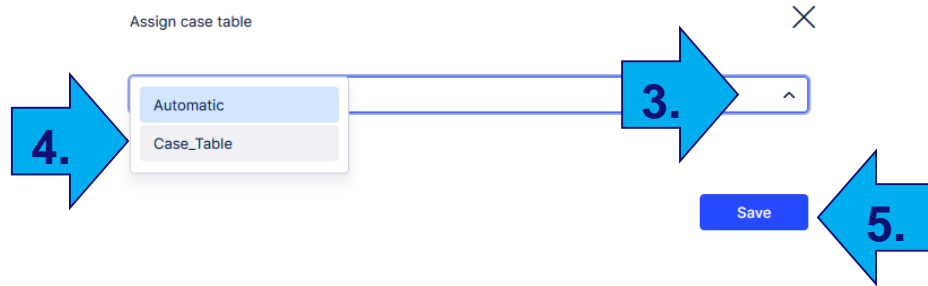
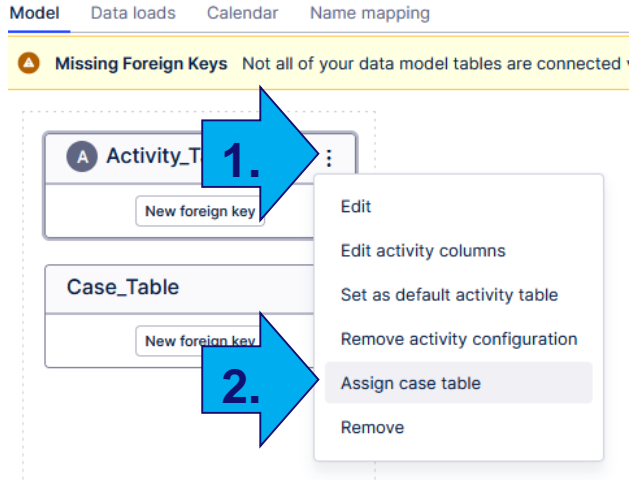
ABC	CASE ID	ABC	ACTIVITY	ABC	ORG-RESOURCE	DATE	COMPLETE TIMES...	ABC	EVENTORIGIN	0.1F	FIRSTWITHDRAW...	0.1F	CREDITSCORE	0.1F	NUMBEROFFERMS	0.1F	MONTHLYCOST	0.1F	OFFEREDAMOUNT
	Application_10003867...		A_Accepted		User_77		2016-11-25 16:33:09		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Complete		User_77		2016-11-25 16:35:38		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Concept		User_1		2016-11-25 15:32:40		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Create Application		User_1		2016-11-25 15:31:09		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Incomplete		User_29		2016-12-06 11:03:16		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Incomplete		User_90		2016-12-02 09:03:39		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	App	X Case ID	X Activity name		User		X Timestamp		App		Sorting		Sorting		Sorting		Sorting		Sorting
	Application_10003867...		A_Submitted		User_1		2016-11-25 15:31:11		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Validating		User_133		2016-12-05 20:02:03		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		A_Validating		User_53		2016-12-01 15:20:14		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		O_Accepted		User_123		2016-12-06 14:58:37		Offer		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		O_Create Offer		User_77		2016-11-25 16:35:29		Offer		5000		1080		58		100.25		5000
	Application_10003867...		O_Created		User_77		2016-11-25 16:35:29		Offer		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		O_Returned		User_53		2016-12-01 15:20:27		Offer		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		O_Sent (mail and online)		User_77		2016-11-25 16:35:38		Offer		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_10003867...		W_Validate application		User_90		2016-12-02 09:03:39		Workflow		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>
	Application_1001177986		A_Accepted		User_52		2016-06-25 16:01:30		Application		<Empty>		<Empty>		<Empty>		<Empty>		<Empty>

Back

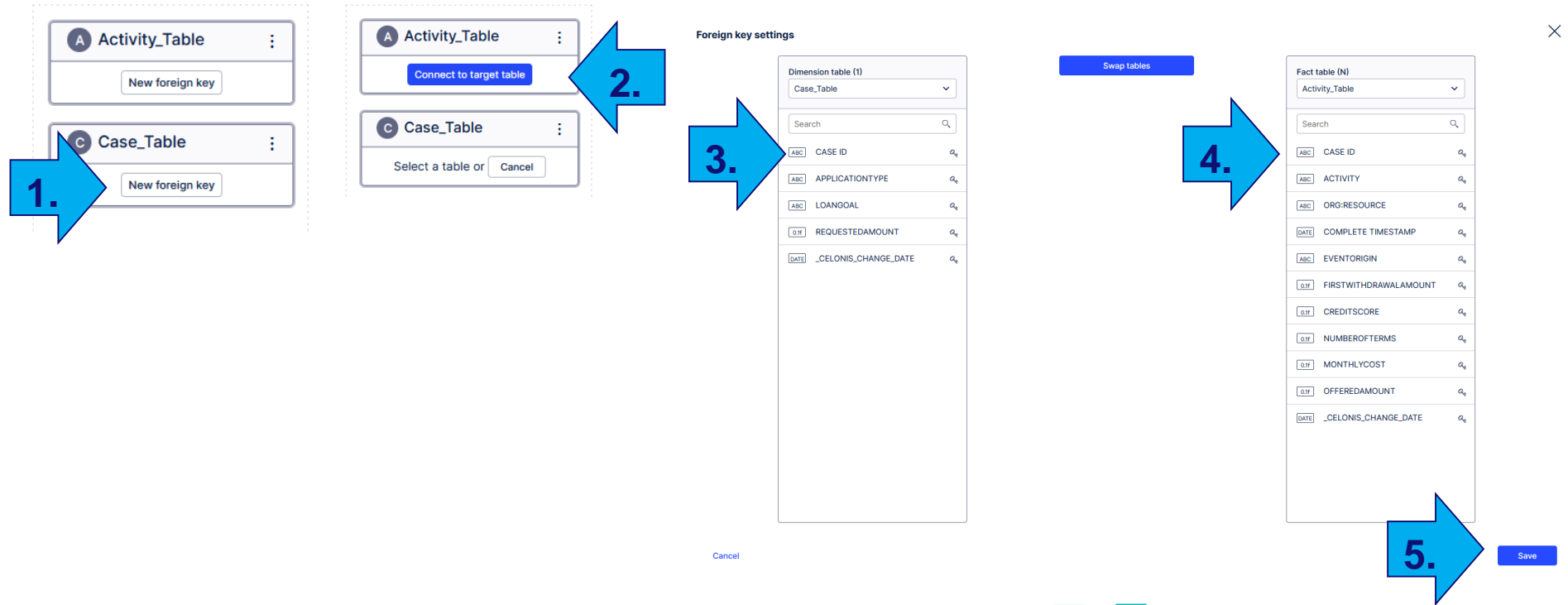
4.

Finish

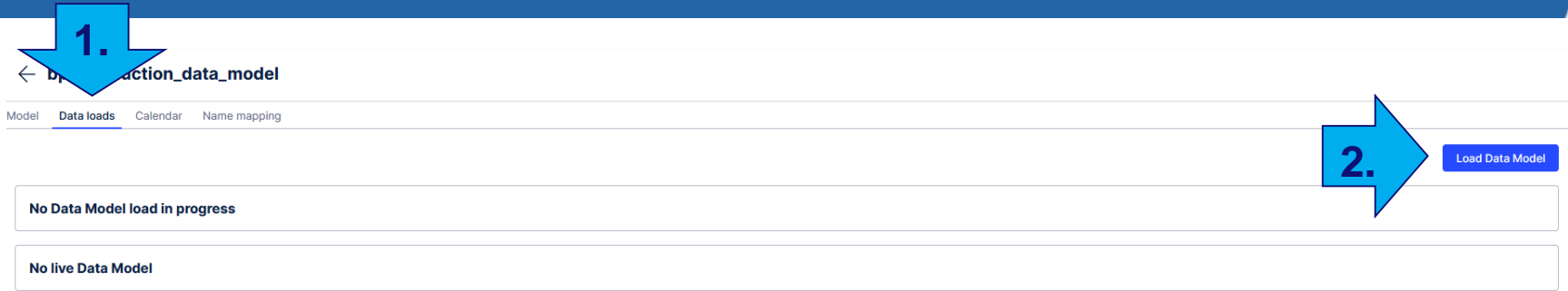
Data Model



Data Model



Data Model



1.

← Data loads action_data_model

Model Data loads Calendar Name mapping

Load Data Model

No Data Model load in progress

No live Data Model

2.

Studio

The screenshot illustrates the 'Create Space' workflow in the Studio application. It features a sidebar on the left with a menu icon (1), a top header with a 'Lecture 2' dropdown (2), and a central 'Create Space' dialog box. The dialog box includes a 'Space Icon' selector (3) and a 'Space Name' input field (4) with the text 'BPI'. A 'Create' button is located at the bottom right of the dialog.

1. Click the menu icon in the sidebar.

2. Click the 'Lecture 2' dropdown in the top header.

3. Click the 'Space Icon' selector in the 'Create Space' dialog.

4. Click the 'Create' button in the 'Create Space' dialog.

Studio

1.


BPI

Create Package


Create Package

A package typically represents the structure of an App or Instrument. It organizes all related assets and allows them to be built, published, and maintained together.

2.

Name 

bpi_instruction

Key 

bpi-instruction

[Learn about Studio on our Help Page](#)

3.

Create

Studio

Folder bpi_instruction is empty

Add your first asset and start building.



Create View

Build a visual interface with operational capabilities tailored for a business role



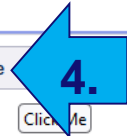
Create Analysis

Build a visual interface focused on process exploration and analytics



☒ DATA MODEL VARIABLE

New Data Model Variable



No variables created yet!

Click Me

NAME

bpi_instruction_analysis



KEY

bpi-instruction-analysis

SELECT DATA MODEL / KNOWLEDGE MODEL

☒ DATA MODEL VARIABLE

☐ KNOWLEDGE MODEL



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Studio

Create Variable



The data model represents the value of the new variable. Choose the data model you want to assign to the new variable.

Choose Data Model

Data Model	Parent Data Pool	Load Status
------------	------------------	-------------



bpi_instruction_data_model

bpi_instruction



Cancel

Next



Create Variable



This new variable can be referenced in any knowledge model or analysis by this unique key.

Key

bpi_instruction_data_model

Description (Optional)



Previous

Save

Studio

Create New Analysis

NAME 

bpi_instruction_analysis

KEY 

bpi-instruction-analysis

SELECT DATA MODEL / KNOWLEDGE MODEL

☒ DATA MODEL VARIABLE 

bpi_instruction_data_model

☐ KNOWLEDGE MODEL 



Cancel

Create



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Analysis

1. Create Package
bpl_instruction
bpl_instruction_analysis

4.98k of 4.98k cases selected 100%

New Sheet
A new sheet waiting to be built.

Proc
Detect and al
from the mo

COMPONENT + Edit

3.

2.

This analysis is empty.
Get started by adding a component to your analysis

Add component

COMPONENT + Edit PREVIEW

New component

PROCESS ANALYSIS COMPONENTS

- Process Explorer
- Variant Explorer
- Throughput Time Search
- Activity Explorer

MACHINE LEARNING COMPONENTS

- > Run ML Notebook

CHARTS AND TABLES

- OLAP Table
- Column Chart

4.

Analysis

The screenshot displays a software interface for configuring an OLAP Table component. On the left, a placeholder box contains a grid icon and the text "OLAP Table" and "You have no data yet". On the right, a "Component options" panel is open, showing a "General options" dropdown, a "Table title" input field, a "Component type" dropdown set to "OLAP Table", and sections for "DIMENSIONS" and "KPIs", each with an "Add" button. A large blue arrow with the number "1." points to the "Add" button in the "DIMENSIONS" section. The interface also features a top bar with a share icon, a toggle switch, and "Edit" and "PREVIEW" buttons.

Analysis

Add data

Source

All
Standard Process Dimension
Activity_Table
Case_Table

Search dimensions..



Eventtime in years
Eventtime in month
Timestamp of first activity in case
Timestamp of last activity in case
Total throughput time in days
Variants
Case when
CASE ID - Activity_Table
ACTIVITY - Activity_Table
ORG:RESOURCE - Activity_Table

Dimensions

KPIs

DIMENSIONS

1.

Custom dimension +

No dimensions yet

Select an existing dimension from the list or create a custom dimension



Chair of Process
and Data Science

RWTHAACHEN
UNIVERSITY

Analysis

The screenshot shows a software interface with a left sidebar and a main editor area. The sidebar contains a search bar labeled "Search dimensions.." and a list of dimension names: "Eventtime in years", "Eventtime in month", "Timestamp of first activity in case", "Timestamp of last activity in case", "Total throughput time in days", "Variants", "Case when", "CASE ID - Activity_Table", "ACTIVITY - Activity_Table", "ORG:RESOURCE - Activity_Table", "COMPLETE TIMESTAMP - Activity_Table", "EVENTORIGIN - Activity_Table", and "FIRSTWITHDRAWALAMOUNT - Activity_Table". The main editor area is titled "EDITOR" and contains a large black rectangle. A blue arrow points from the text "Insert dimension formulas" to the black rectangle. The interface also includes navigation icons at the top and bottom.

EDITOR

1

Search dimensions..

Eventtime in years

Eventtime in month

Timestamp of first activity in case

Timestamp of last activity in case

Total throughput time in days

Variants

Case when

CASE ID - Activity_Table

ACTIVITY - Activity_Table

ORG:RESOURCE - Activity_Table

COMPLETE TIMESTAMP - Activity_Table

EVENTORIGIN - Activity_Table

FIRSTWITHDRAWALAMOUNT - Activity_Table

Insert dimension formulas

DEVIEW

Accessing Columns in Table

- For later formulas, you need to refer to the columns of your table
- You refer to columns with `YourTable>YourColumn`

Pull Up Aggregation

- Pull-Up-functions are used to achieve nested aggregations and filters on aggregations
- The formula always consists of “PU_X (target_table, source_table.column ...)”
- A 1:N relationship between the target table and the table of the specified source column is required.
 - Therefore, your case table is, most of the time, the target table, your activity table the source table

Analysis

Function Name	General Formula
Case When	Case When ... Then ... Else ... End
COUNT	Count([Distinct] Column)
CALC_REWORK	CALC_REWORK(Condition, Column)
PU_AGGREGATION	PU_FIRST(/LAST) (Group, Column, Condition, Order)
CALC_THROUGHPUTTIME	CALC_THROUGHPUTTIME(Trace Crop, Transformed Timestamp to Integer)
DAYS_BETWEEN	DAYS_BETWEEN (Column, Column)
MATCH_ACTIVITIES	MATCH_ACTIVITIES(Column, STARTING[], NODE[], EXCLUDING[], ENDING [some Activities])

More formulas and explanations: <https://docs.celonis.com/en/pql-function-library.html>

Analysis

Function Name	Role in Creating the Situation Table's Variables
Case When	Creating conditional structures in other functions
COUNT/COUNT Distinct	Extracting trace features based on the times certain activities occur in the trace/ or checks just the existence of these activities in the trace
CALC_REWORK	Extracting trace features based on the times certain activities occur in the trace
PU_FIRST, PU_LAST	Extracting trace attributes based on the first/last trace's event attribute
CALC_THROUGHPUTTIME	Extracting the throughput time feature of a trace(between start and end events or any other pairs of the trace)
DAYS_BETWEEN	Extracting a time feature of a trace between a pair of its events (in days)
MATCH_ACTIVITIES	Extracting a binary trace attribute whether it starts/ ends with certain activities or includes /excludes these activities

More formulas and explanations: <https://docs.celonis.com/en/pql-function-library.html>

Analysis

Dimension Name	Dimension Formula
Case ID	"Case_Table"."CASE ID"
Application Type	"Case_Table"."APPLICATIONTYPE"
Loan Goal	"Case_Table"."LOANGOAL"
Requested Amount	"Case_Table"."REQUESTEDAMOUNT"
First Credit Score	PU_FIRST ("Case_Table", "Activity_Table"."CREDITSCORE")
Last Credit Score	PU_LAST ("Case_Table", "Activity_Table"."CREDITSCORE")

Analysis

Dimension Name	Dimension Formula
Having Call to Complete Application	<pre>COUNT(DISTINCT CASE WHEN "Activity_Table"."ACTIVITY"='W_Call incomplete files' THEN 1 END) Or alternatively: CASE WHEN MATCH_ACTIVITIES("Activity_Table"."ACTIVITY", NODE ['W_Call incomplete files'])=1 THEN 1 ELSE 0 END</pre>

Analysis

Dimension Name	Dimension Formula
Number of Offers	<pre>COUNT(CASE WHEN "Activity_Table"."ACTIVITY"='O_Create Offer' THEN "Activity_Table"."ACTIVITY" END) Or alternatively: CALC_REWORK ("Activity_Table"."ACTIVITY"='O_Create Offer',"Activity_Table"."ACTIVITY")</pre>

Analysis

Dimension Name	Dimension Formula
First Offer	PU_FIRST ("Case_Table", "Activity_Table"."OFFEREDAMOUNT")
Last Offer	PU_LAST ("Case_Table", "Activity_Table"."OFFEREDAMOUNT")
First Offers' Monthly Cost	PU_FIRST ("Case_Table","Activity_Table"."MONTHLYCOST")
Last Offers' Monthly Cost	PU_LAST ("Case_Table","Activity_Table"."MONTHLYCOST")

Analysis

Dimension Name	Dimension Formula
Throughput Times	<pre>DAYS_BETWEEN (MIN("Activity_Table"."COMPLETE TIMESTAMP"), MAX("Activity_Table"."COMPLETE TIMESTAMP")) Or alternatively: CALC_THROUGHPUT (CASE_START TO CASE_END, REMAP_TIMESTAMPS("Activity_Table"."COMPL ETE_TIMESTAMP", DAYS))</pre>

Analysis

Dimension Name	Dimension Formula
Results	<pre>CASE WHEN MATCH_ACTIVITIES ("Activity_Table"."ACTIVITY", NODE ['A_Cancelled']) = 1 THEN 'Cancelled by client' WHEN MATCH_ACTIVITIES ("Activity_Table"."ACTIVITY", NODE ['A_Denied']) = 1 THEN 'Denied by the bank' WHEN MATCH_ACTIVITIES ("Activity_Table"."ACTIVITY", NODE ['A_Pending']) = 1 THEN 'Successful completion' END</pre>

Export Situation Table

Make sure you are in edit mode.



4.98k of 4.98k cases selected 100%

COMPONENT + ☒ Edit

bpi_instruction_analysis
Last edited
2 hours ago by Harry Beyel
[Version Control](#)
Last dataload
-3h

Analysis settings

Saved formulas

Load script

Process explorer KPIs

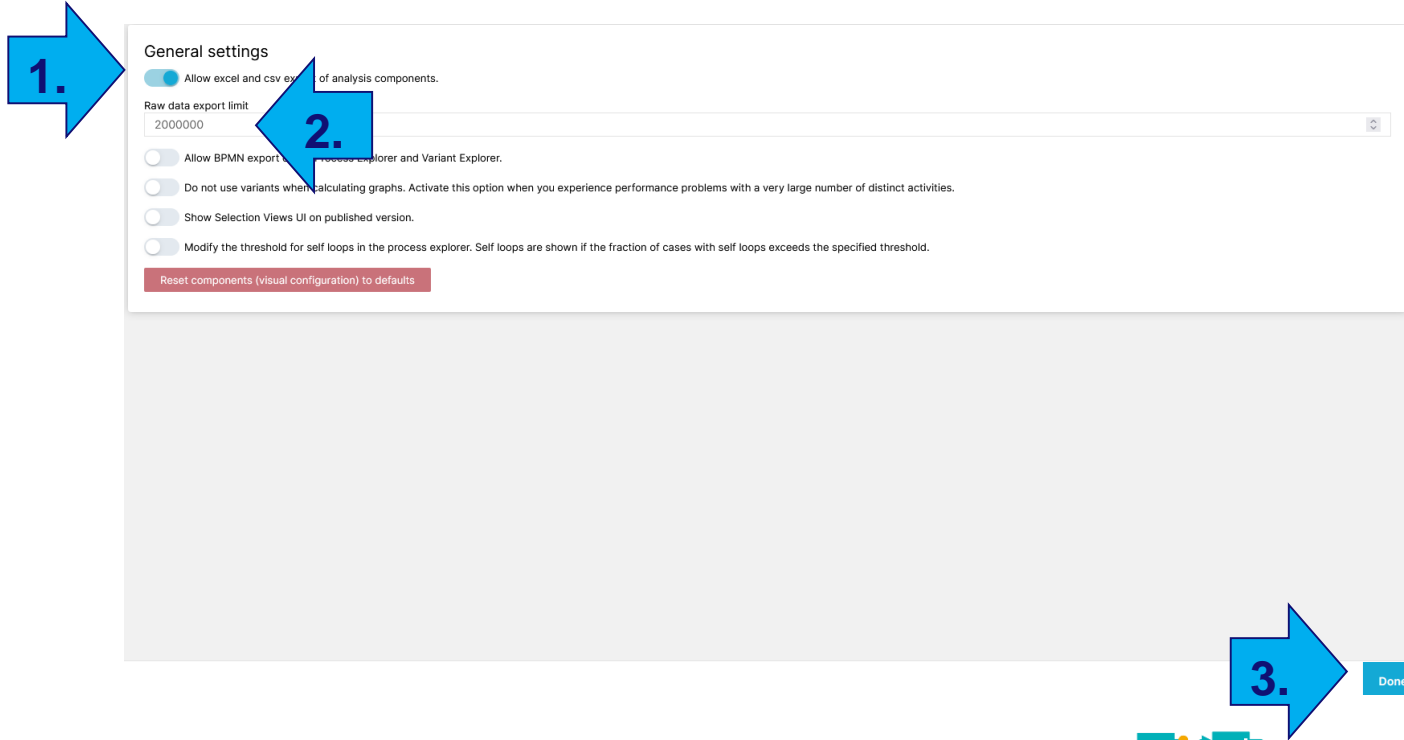
Variables

Keyboard shortcuts

Activate LiveReload

Applicati...	Loan Goal	Request...	First Cr...	Last Cre...	Having ...	Number ...	First Offer	Last Offer	First Off...	Last Off...	Through...	Results	
New credit	Car	5000	1080	1080	0	1	5000	5000	100.25	100.25	10.9774...	Succe...	
New credit	Existing I...	26000	972	972	0	1	26000	26000	264.74	264.74	28.4195...	Succe...	
New credit	Car	10000	896	896	0	1	25000	25000	535.22	535.22	35.8184...	Succe...	
New credit	Home im...	17000	0	0	0	1	17000	17000	319.2	319.2	21.8463...	Denie...	
New credit	Car	15000	0	0	0	1	15000	15000	157.59	157.59	31.4285...	Cance...	
Limit raise	Home im...	20000	0	0	0	1	20000	20000	350	350	16.9467...	Succe...	
New credit	Existing I...	15000	959	959	0	1	15000	15000	250	250	12.5998...	Succe...	
New credit	Existing I...	42000	0	0	0	1	42000	42000	420	420	34.4300...	Cance...	
Applicati...	Limit raise	Home im...	40000	881	881	0	1	40000	40000	400	400	8.05991...	Succe...
Applicati...	New credit	Car	14000	946	946	0	1	15000	15000	280	280	10.7404...	Succe...
Applicati...	New credit	Car	15700	781	781	0	1	15700	15700	343.63	343.63	6.89625	Succe...
Applicati...	New credit	Car	10000	1004	1004	0	1	10000	10000	180.08	180.08	10.1667	Succe...

Export Situation Table



1. Click the 'General settings' button.

2. Configure the settings:

- ☒ Allow excel and csv export of analysis components.
- Raw data export limit: 2000000
- ☐ Allow BPMN export of process explorer and Variant Explorer.
- ☐ Do not use variants when calculating graphs. Activate this option when you experience performance problems with a very large number of distinct activities.
- ☐ Show Selection Views UI on published version.
- ☐ Modify the threshold for self loops in the process explorer. Self loops are shown if the fraction of cases with self loops exceeds the specified threshold.

Reset components (visual configuration) to defaults

3. Click the 'Done' button.

Export Situation Table

[illegible]

Export Situation Table



**You can find your CSV-file
in your download folder.**



Working with Situation Tables from Celonis



About the data

1. **CASE ID:** The unique identifier assigned to each loan application
2. **Application Type:** The type of each loan application
3. **Loan Goal:** The goal of each loan application
4. **Requested Amount** The amount each loan application asks to be paid
5. **First Credit Score:** The first recorded credit of the application
6. **Last Credit Score:** The last recorded credit of the application
7. **Having Call to Complete Application:** Indicates if the loan application process includes the activity “W_Call incomplete files” which means the bank calls the applicant to provide further data/documents.
8. **Number Of Offers:** The number of times the bank offers a loan to an application. These offers are logged when the event’s activity is “O_Create Offer”.
9. **First Offer:** The amount of the first offer the bank provides to an application.
10. **Last Offer:** The amount of the last offer the bank provides to an application.
11. **First Offer’s Monthly Cost:** The monthly cost of the first offer the bank provides to an application.
12. **Last Offer’s Monthly Cost:** The monthly cost of the last offer the bank provides to an application.
13. **Throughput Time:** The time spent on each application from its first event to the last
14. **Results:** The outcome of each application. This could be one of the following:
 - i. **Successful completion:** the application includes the “A_Pending” activity. This activity describes a situation in which all documents have been received and the assessment is positive. The loan is confirmed and signed to be paid to the customer.
 - ii. **Denied by the bank:** the application includes “A_Denied” activity. This activity describes a situation in which the application doesn’t match the acceptance criteria.
 - iii. **Cancelled by client:** the application includes the “A_Cancelled” activity. This activity describes a situation in which the applicant does not get back to the bank after an offer was sent out

Import the data

- **Import the data in RapidMiner**
- **No date format needed**

Visualization

- Click on Visualization and create a bar graph (column), with an x-axis showing *Numbers of Offers*, and *Requested Amount* as value column
- What are your observations?

Decision Tree

- We want to predict *Result*. Therefore, we have to set the role.
- As the target role, select *label*.
- Connect the *Decision Tree* with the *Set Role* module and run the process.

Decision Tree – Parameters

- **For most modules, you can set different parameters.**
- **For now, set the maximal depth to 5**
- **Rerun the process**

Decision Tree –Check results

- To evaluate your results, you need to split the data. Otherwise, you will evaluate on data your model knows.
- To split data, use the *Split Data* module. Specify as partitions 0.8 and 0.2
- Next, apply the decision tree to the unseen data, using *Apply Model*
- Run the process

Decision Tree –Evaluating

- Checking every data instance by hand is impossible.
- Use the *Performance* module and link it with the labeled data from the *Apply Model* module
- Rerun the process.

Decision Tree – Cross Validation

- Instead of doing the previous steps, you can also use the **Cross Validation** operator
- You only have three modules on the “main” process page: **Retrieve**, **Set Role**, and **Cross Validation**. Link the first four outputs of the operator to the process output
- By double-clicking on **Cross Validation**, you can see the operator’s subprocess
- For training, use the prev. decision tree, for testing, the operators **Apply Model** and **Performance**
- Rerun the process

Decision Tree – Play with Parameters

- As seen earlier, we can change multiple parameters.
- Instead of changing parameters one by one and rerunning the process, we can do that with an operator: *Loop parameters*
- Drag *Cross Validation* into *Loop Parameters*
- Setting for *Loop Parameters*: Selected parameter is maximal depth, min=1, max=5, in 5 steps
- Input for *Loop Parameters*: *Set Role*
- Output for *Cross Validation*: *Model and performance*

Clustering – Getting Started

- Using *Select Attributes*, select the attributes *Numbers of Offers*, *Requested Amount*, *Throughput Times*
- Normalize the values using Z-transformation (advanced parameters)
- Let's cluster similar instances. To group them, we use the operator *Clustering (k-Means)*, with 4 centroids
- Run the process

Clustering – Visualize Clusters

- If we want to visualize the clusters, we cannot display every instance → Sampling is needed
- Connect the output clustered set with *Sample* operator. Set the sample size to 100 and the local random seed to 2023 (advanced parameter).
- Create a 3D-visualization of the clustered samples.

Clustering –Drawing Conclusions

- To draw conclusions related to unnormalized, or even original data, we have to join the clustered data with the original one
- To join data, we generate ids. Then we use the *Multiply* operator to multiply the original data with ids.
- Then, we run the process.
- Create a 3D-Visualization as before.

Frequent Itemsets

- We want to know which of the following variable values appear together: *Requested Amount*, *Having a Call to Complete Application*, *Results*
- In this process, we discretize the variables:
 - Requested amount:
 - Upper limit 9000: low request
 - Upper limit 20,000: medium request
 - Upper limit Infinity: High request
 - Having a Call to Complete Application:
 - Two operators are needed: *Numerical to Polynomial*, *Map*
 - If there was a call, denote it with “called,” otherwise with “not called”
- Use the module *FP-Growth*, by setting the frequency to 100

Association Rules

- Add to your former process the module *Create Association Rules*
- Set the minimum confidence to 0.8
- Connect the frequent set output with this module and run the process