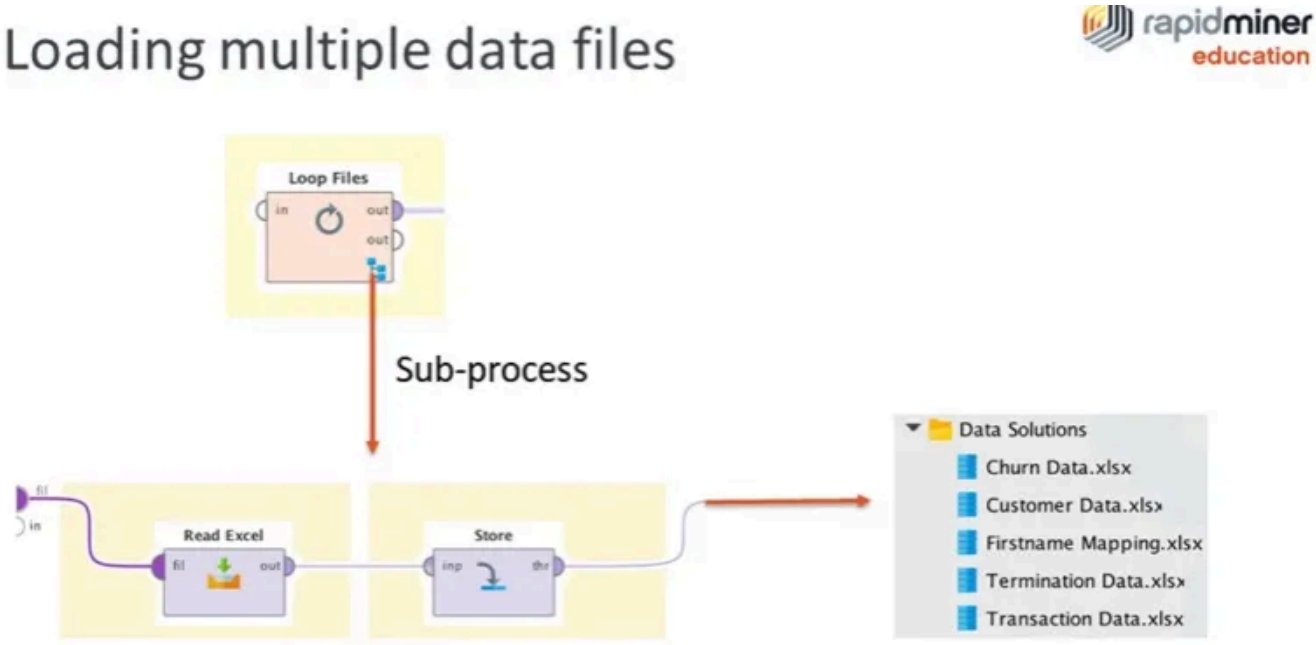
Rapid Miner -Data Engineering Master certificate(1)

KevinLuo · Follow

8 min read · Oct 18, 2021

Loops and Macros Introduction

Expand your ability to handle multiple data files with loops and macros. Instead of reading one file, programmatically loop over many files and store them appropriately.



Raw Data Files

Get a quick introduction to the multiple raw data files.

Extracting Data

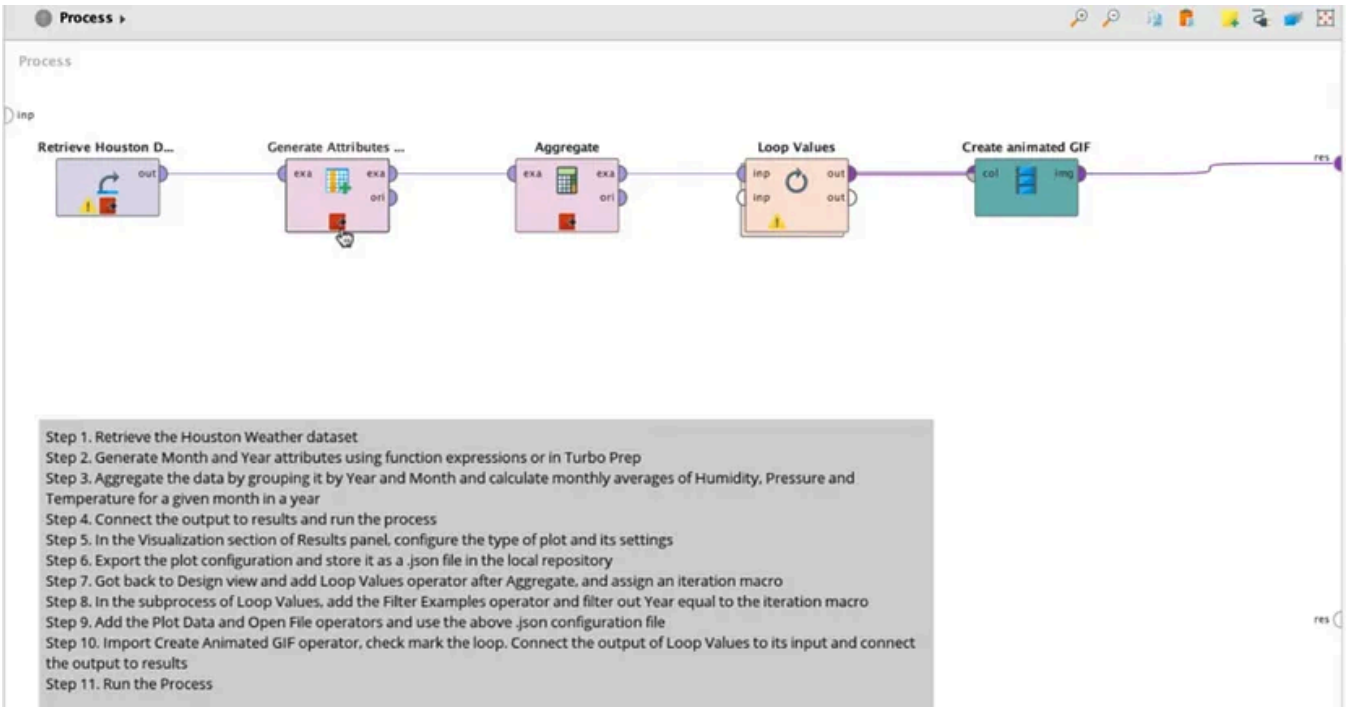
Watch and do! Loop over many files, and use macros to store them with useful names. Pay attention to how collections can be created. Collections will also be

revisited in the section on Text and Web.

Loop Attributes

Watch and do! Perform Feature Generation for many features at once.

Here is a fun example of Loop Values. This section lesson walks through an example of **creating such animation on a time-series weather dataset using some data prep, looping, and the "Create animated GIF" operator.** Open the process in Studio and check it out for yourself.



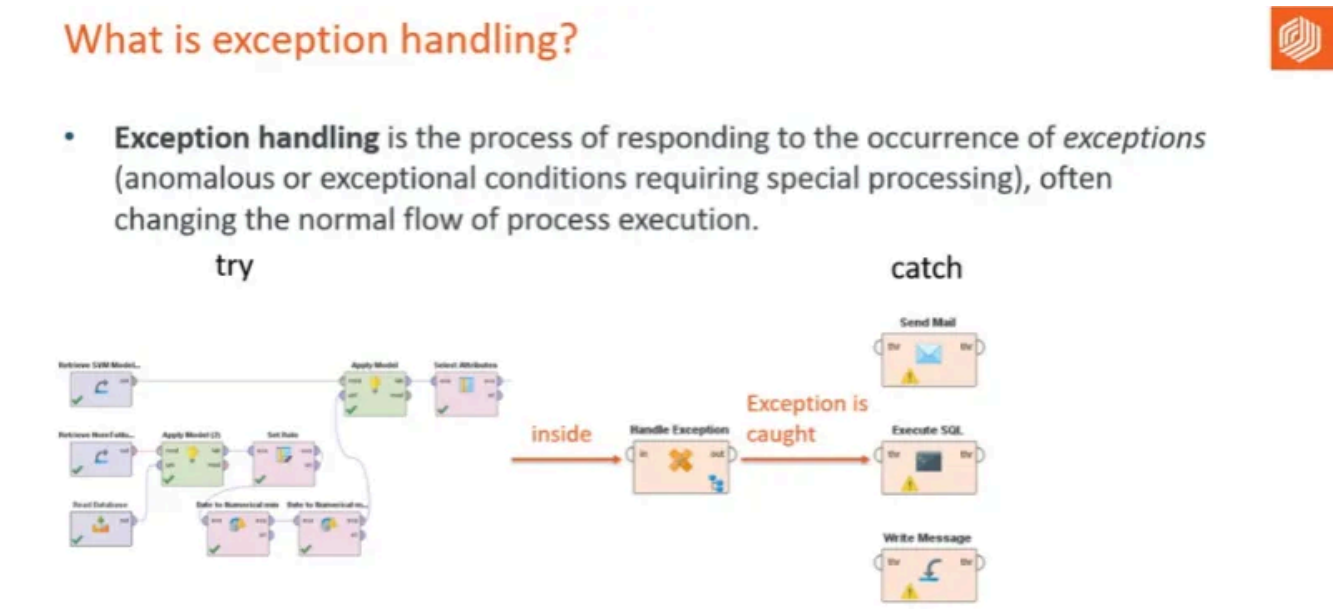
Automated Model Selection and Optimization

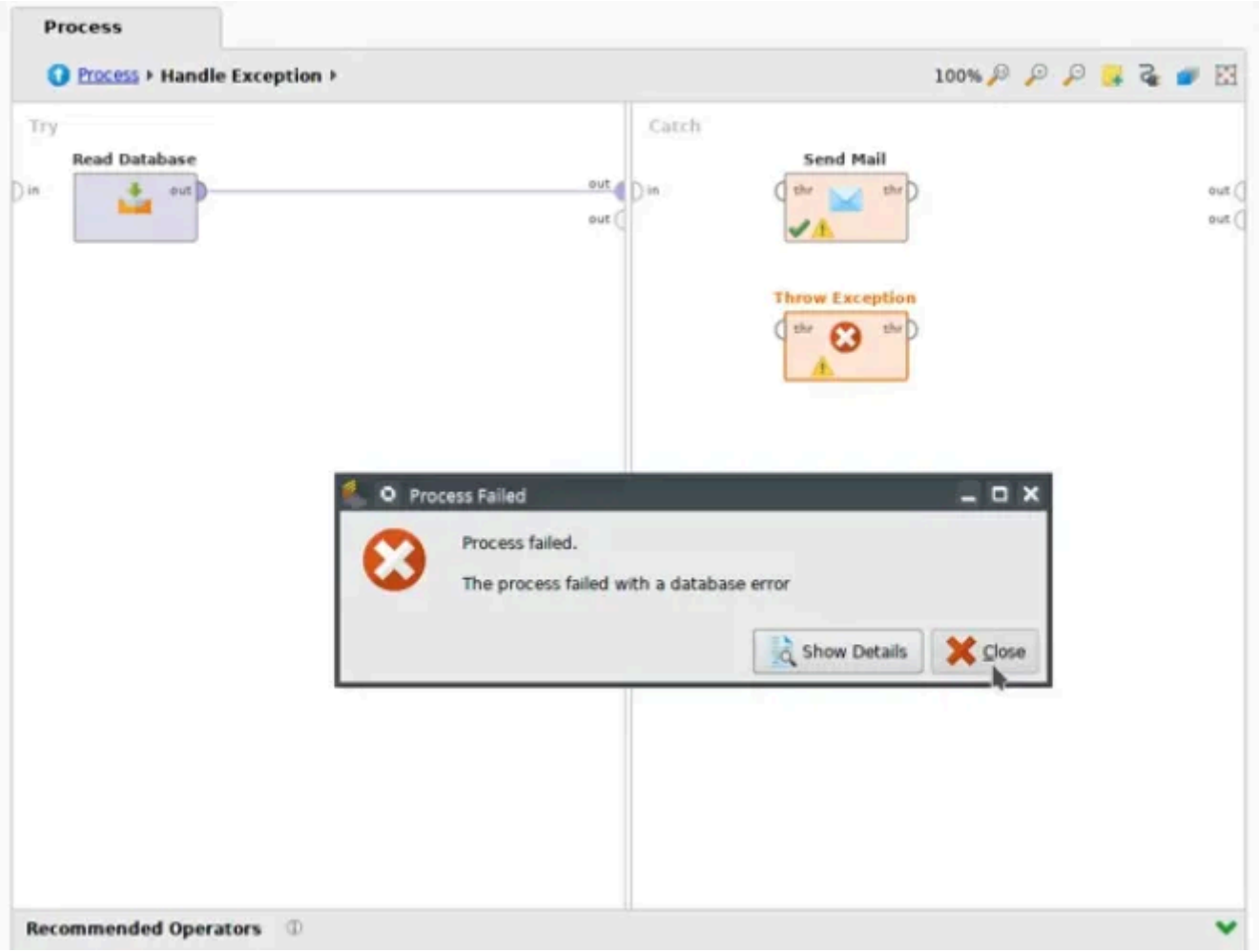
In this video on more advanced optimization, we also show you how to how to use '**select sub-process**', and how to remember and recall certain elements in a RapidMiner process. This module is split in two parts the first shows how to embed our most promisin models into one optimiaztion each. We then execute them one by one. To make sure we can comnpar them aftre we are then storing the best parameter combinations and repurpose them to create ROC curves with those parameters. This allows us to finally compare optimized models.

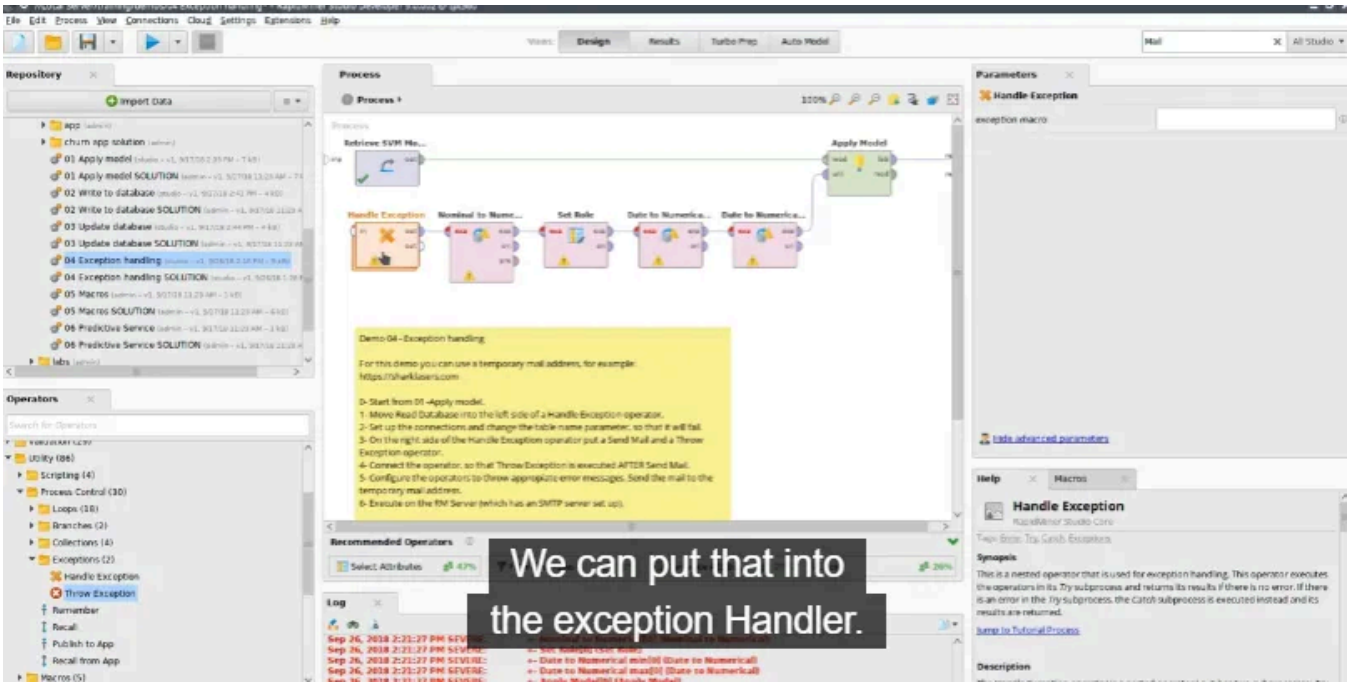
Error and Exception Handling

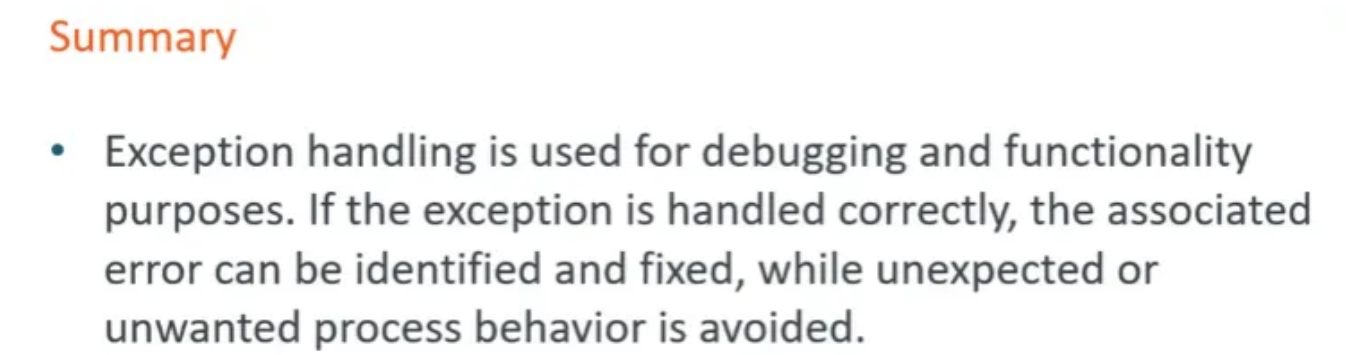
Exception handling in RapidMiner processes is a powerful way to react to unexpected situations or errors that only happen from time to time. You can execute operators when an exception occurs, and end the process using Throw Exception.

The demo also shows the Log panel and the specification of operator execution order.



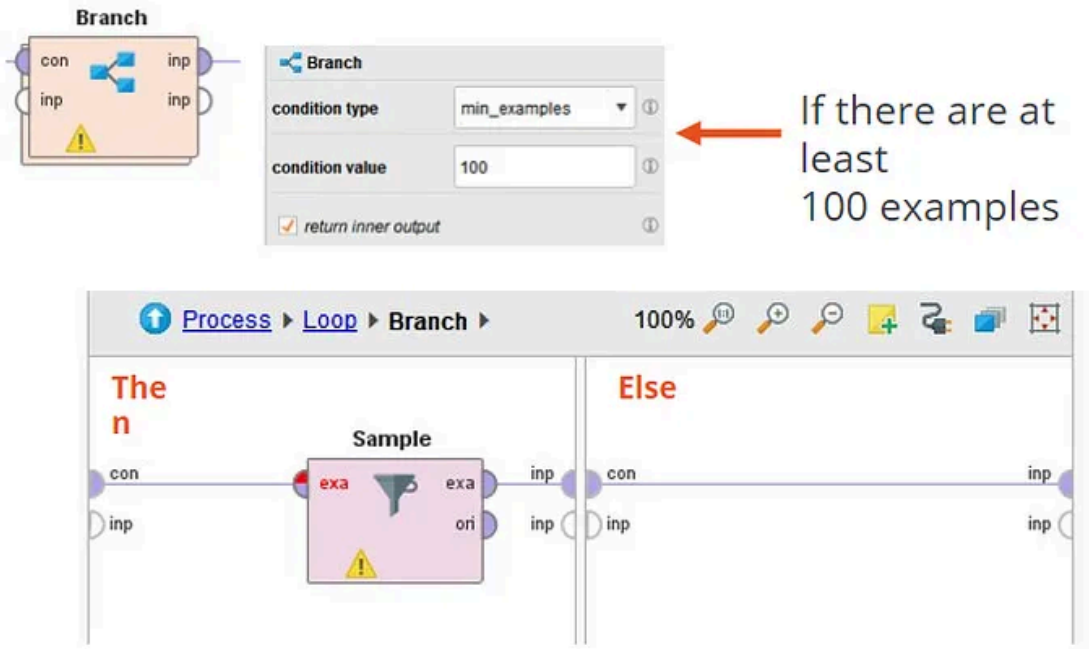




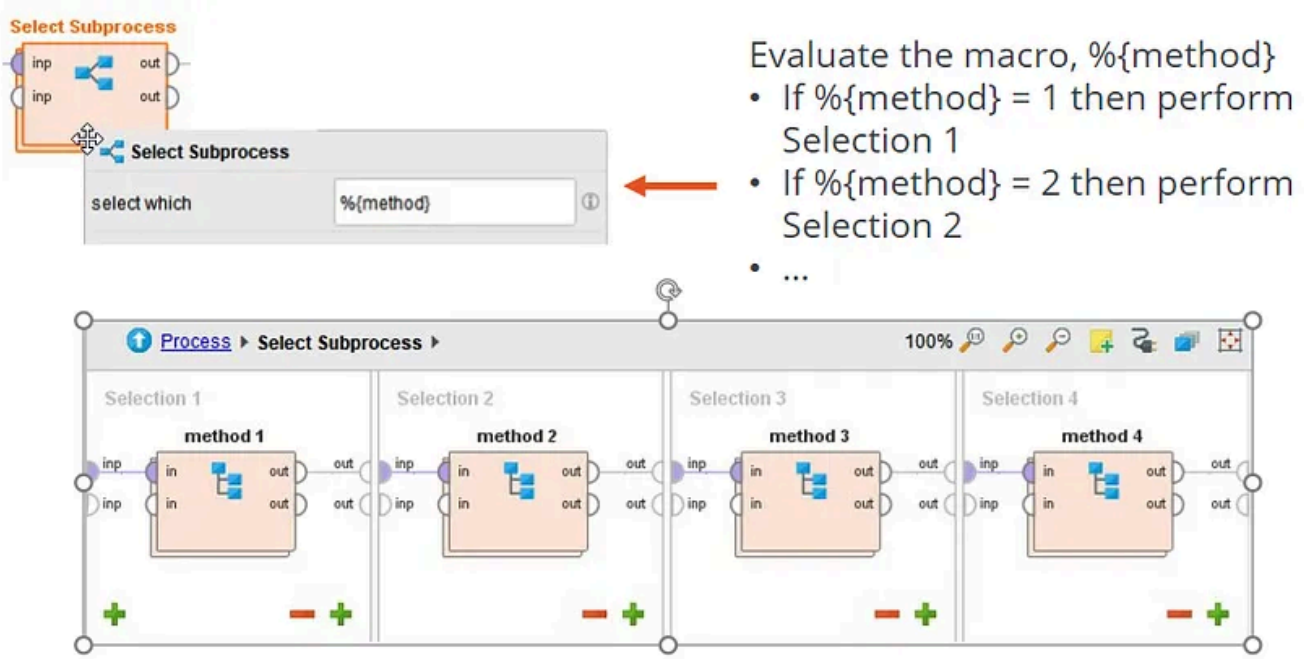


Branches

Branches are an important flow control structure. They have an if(condition, then, else) structure. Here is an example where we check the number of examples in the example set. If there are at least 100 examples, then the data is down-sampled. If there are less than 100 examples, then the data is passed on with no changes. Consider reading the documentation here.



Notice that the “Handle Exception” operator is like a specialized branch. These two operators have some similar capabilities. There is another operator, “Select Subprocess” that also has similar capabilities, it is a generalization of the Branch. Instead of exactly two options, it can have many options like a switch and case. Here we have an example where there are 4 possible cases or selections:

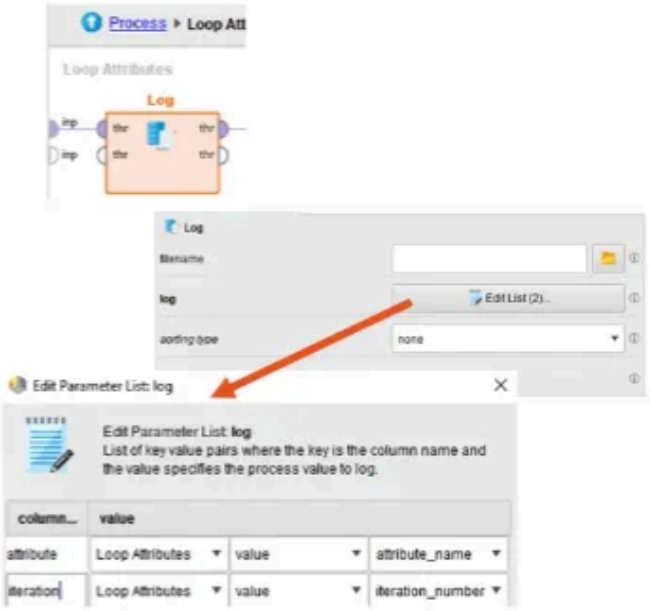


Logging Runtime Values

Watch and do! Implement logging in RapidMiner Processes.

Logging

- Logging should often be carefully designed, including when:
  - There are complex flow control structures
  - There are operations that are likely to fail
- There are many ways to implement logging in complex processes including:
  - Send to file
  - Send to standard output
  - Send to data and store



Process Control and Logging FAQs.

Click on the cards for a little extra information.

> *What is a macro anyway?*

Remember that a macro in RapidMiner is just a simple variable.
It is created within the process and its scope is the entire process.
Loops use macros, but macros can be used many other ways as well.
There will be more information on macros later in the course!

> *What are some of the most common loop operators?*

There are a variety of loop operators available and they each have their own advantages

Some of the most common operators are:

- Loop
- Loop Attributes
- Loop Values
- Loop Parameters
- Loop Files

- Loop Examples
- Loop Collection

> *A macro can keep a simple value within the process, but is it possible to remember and recall a more complex object?*

The Remember and Recall Operators can be used to “Remember” a reference to a complex object like an example set or a model. It keeps the object in the scope of the process, or the process store. The “Recall” operator can retrieve it from the process store with its name.

> *Parameter values are generally constant for a given process, but can they be mutable? Can they be set at runtime?*

The Set Parameters operator can access any other operator in the same process

> *Besides Handle Exception, are there other tools to handle operations that may have inconsistent results?*

Join Paths can be useful in a variety of situations. It will deliver the first non-null input. Sometimes this can be used instead of Handle Exception, or in other situations where you may want the process to bypass an operation if it is slow to return results. The Delay operator pauses process execution and can also be helpful in some of these situations.

> *The log operator can take operator parameters and values, are there other ways to add other information to the log?*

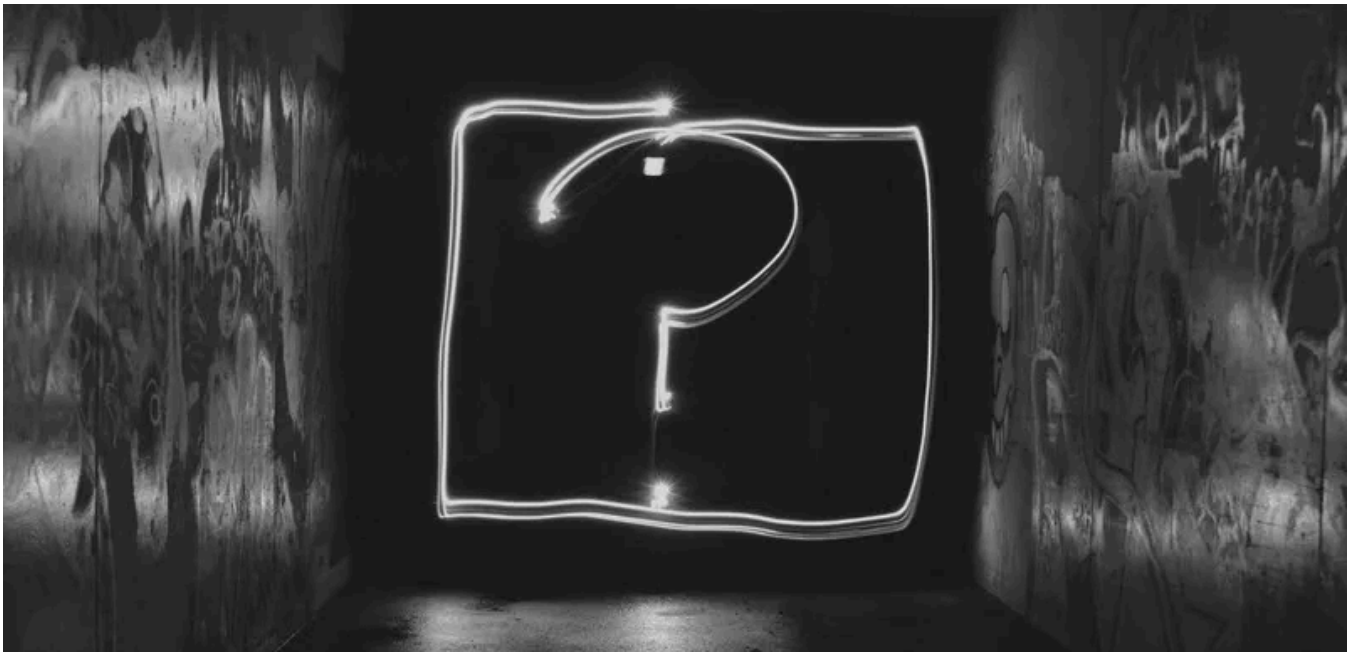
The “Extract Log Value” operator can be used to extract values from the data for logging. The “Provide Macro as Log Value” operator can provide macro values. These operators are typically used in combination with the log operator. They make the values easily accessible as an operator value. With the ability to perform computations on both data and macros, and use those values, these provide a lot of capability. Time can be logged directly with the log operator or indirectly from macros.

> *What are some good tools for debugging and inspecting loops?*

The most common tool is to set breakpoints. As you build more complex processes, it is more important to become familiar with the use of breakpoints. Remember that in the RapidMiner Studio Edit menu, there are tools for managing breakpoints in bulk, but can also be used to manage individual breakpoints. The right-click menu for a given operator is a good way to manage breakpoints one at a time. Remember that with breakpoints, you can not only inspect operator outputs, but also macro values and other context information.

The next most common tool for inspecting loops is to setup logging.

Take a simple and fast quiz on Flow Control



Finally, done the easy task section.

Which of the following are true statements about the Loop Attributes operator? (Select TWO correct answers)

Correct Answer

A. The loop will always have exactly as many iterations as there are attributes in the example set.

B. The loop will have as many iterations as selected attributes, and there is more than one way to select attributes. Your Answer ✔

C. Within a loop iteration, only the attribute of interest will be available.

D. Within a loop iteration, all attributes will be available, but the attribute of interest can be referenced with the attribute name macro Your Answer ✔

What is the name of the operator that can be used to manage operations that may fail? (Select One correct answer)

A. Manage Failures ✘

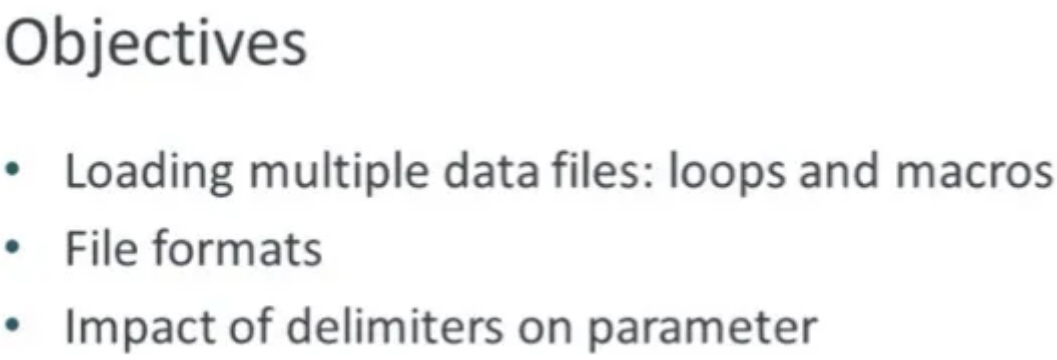
B. Try ✘

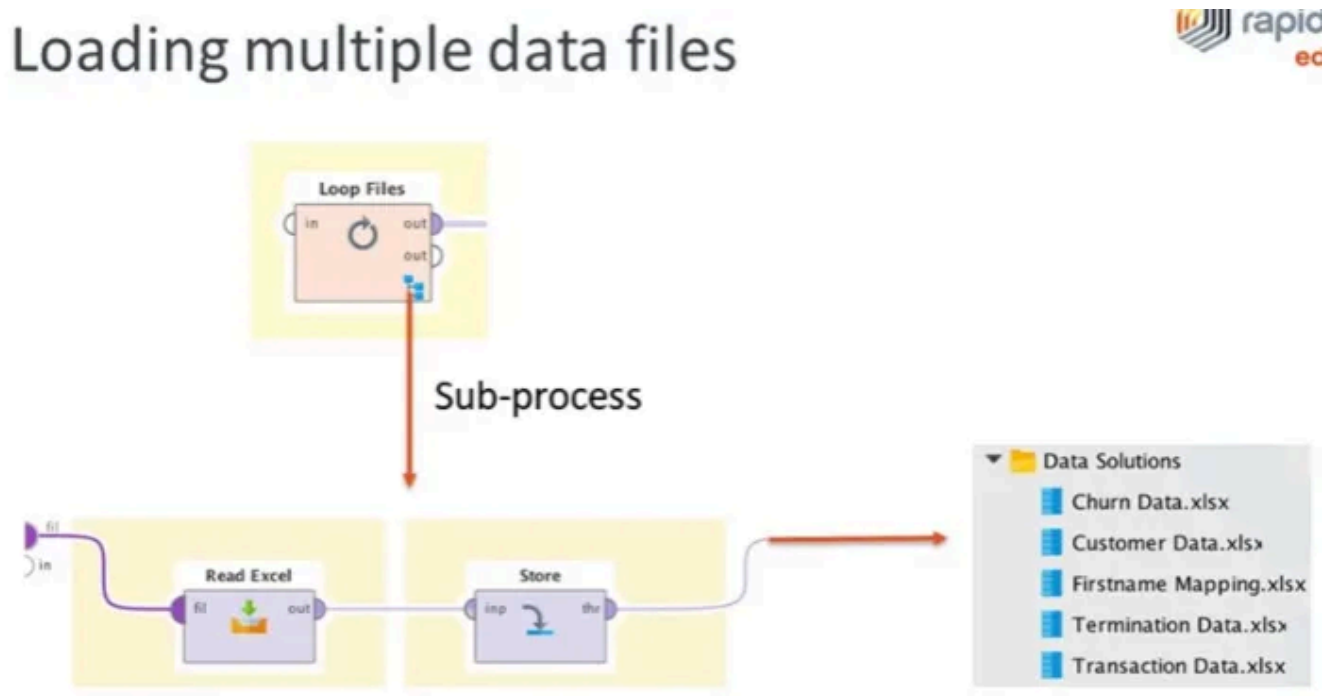
C. Catch ✘

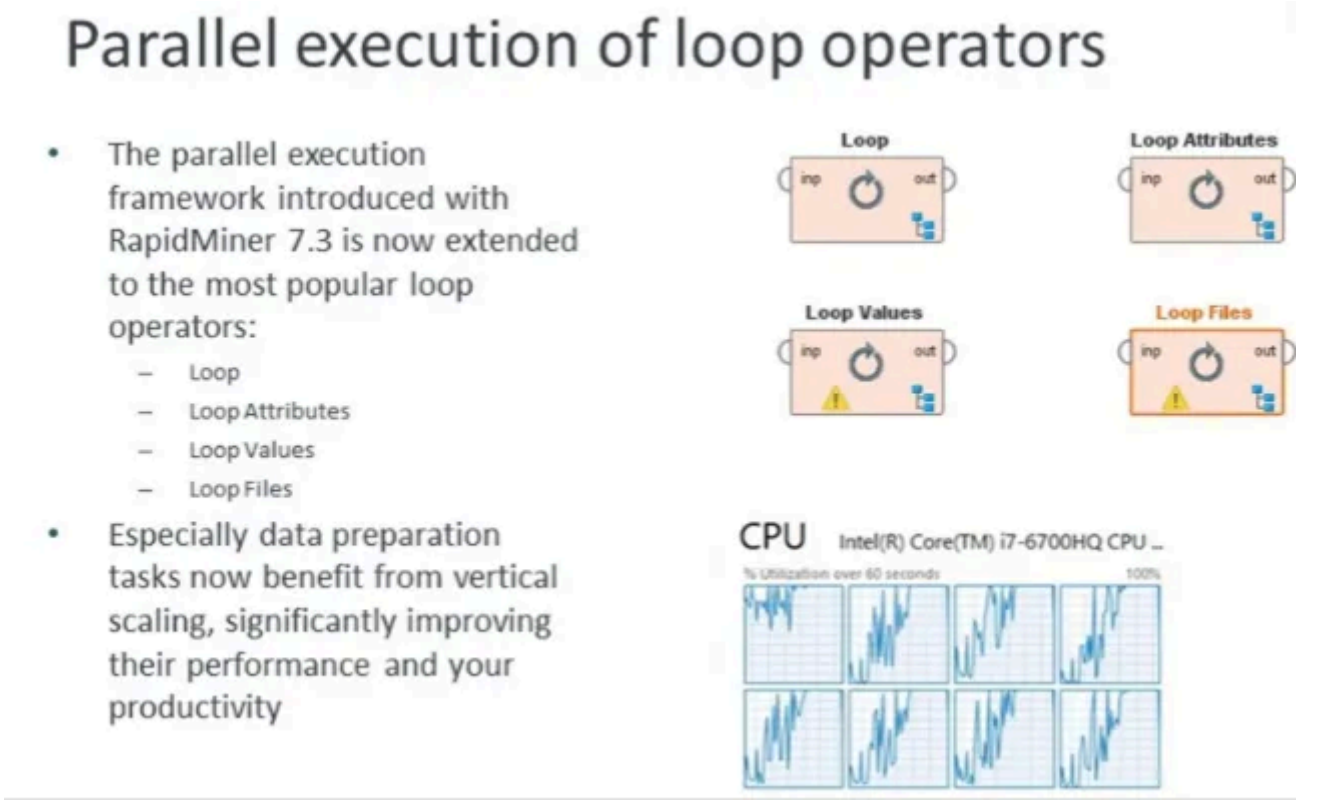
D. Handle Exception Your Answer ✔

Loops and Macros Introduction

Expand your ability to handle multiple data files with loops and macros. Instead of reading one file, programmatically loop over many files and store them appropriately.

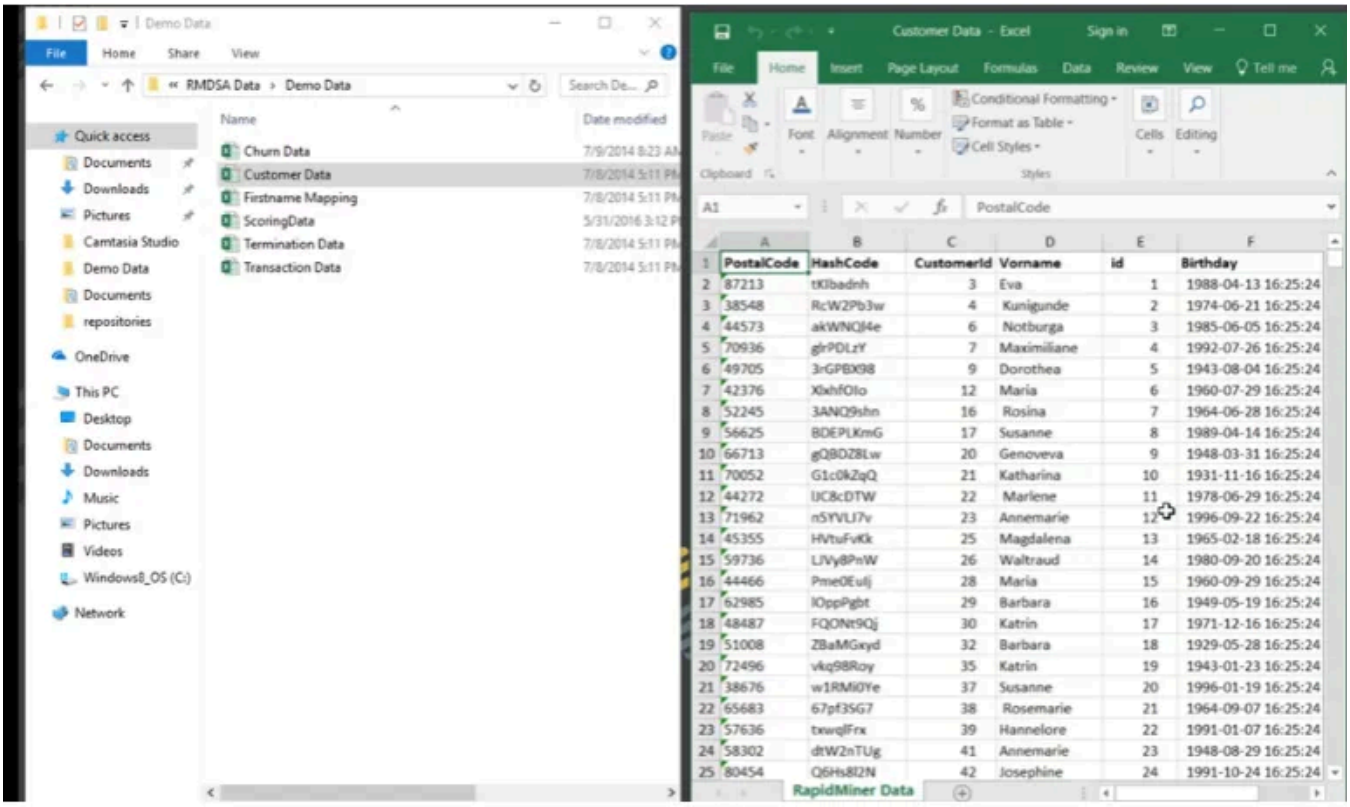






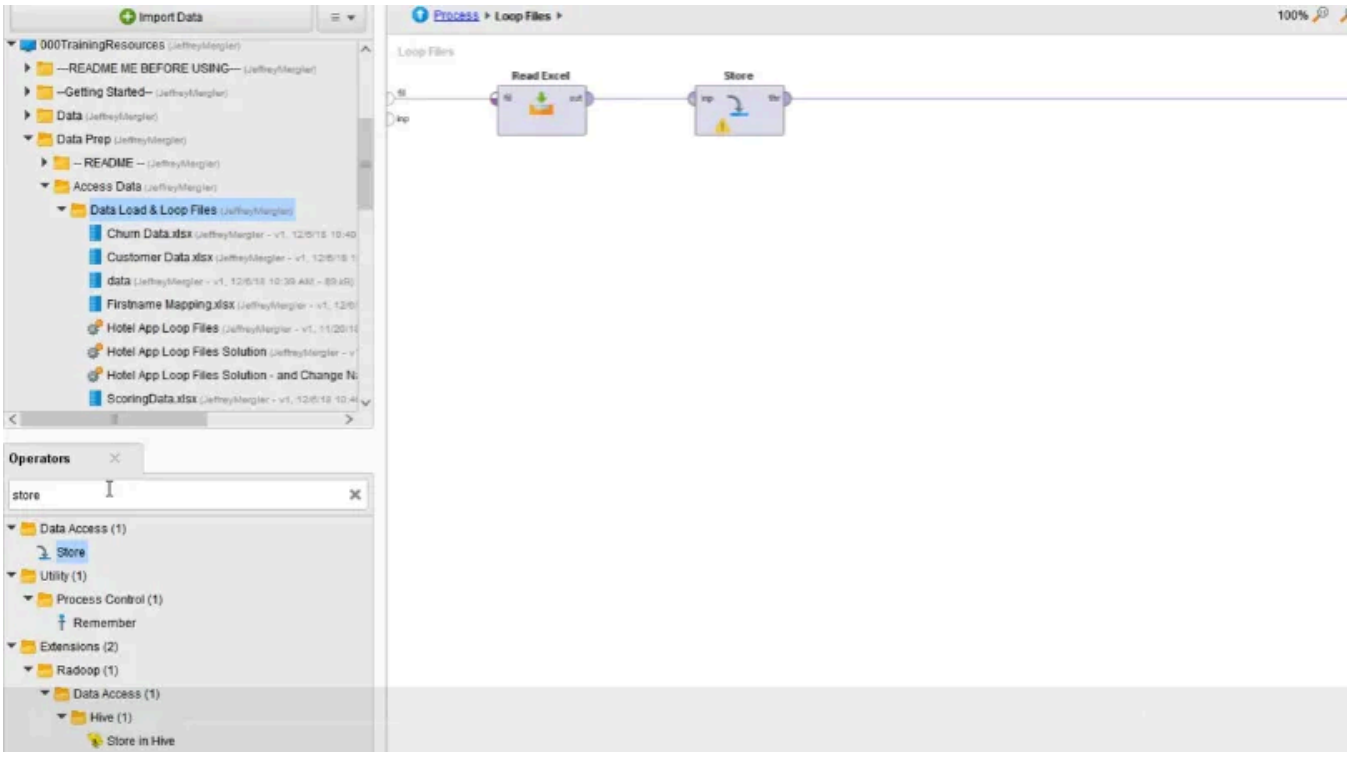
Raw Data Files

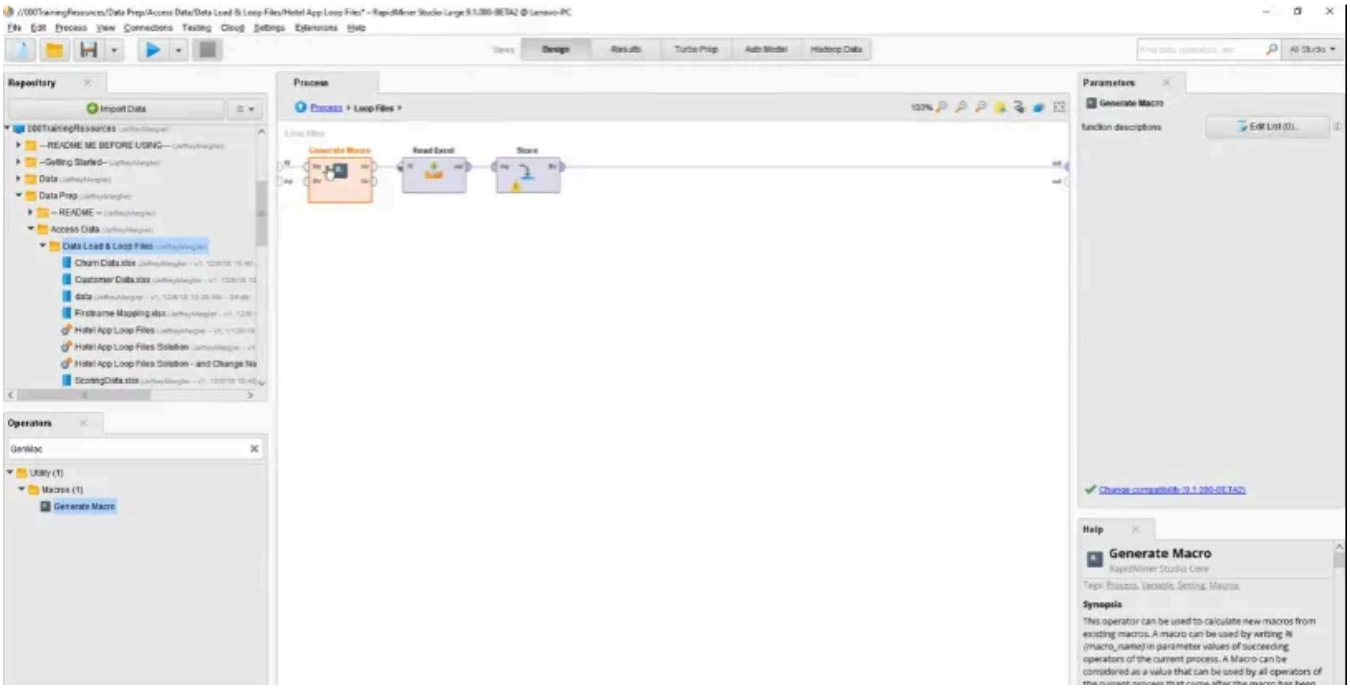
Get a quick introduction to the multiple raw data files.

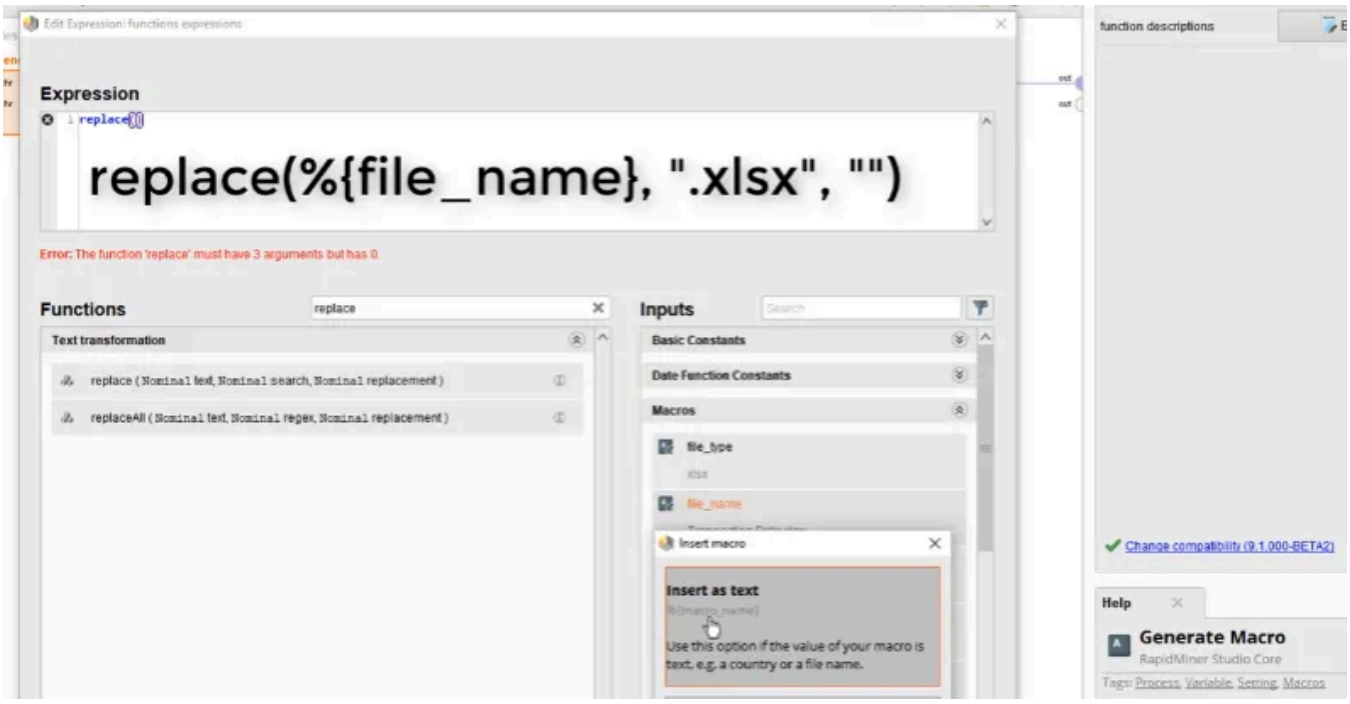


Extracting Data

Watch and do! Loop over many files, and use macros to store them with useful names. Pay attention to how collections can be created. Collections will also be revisited in the section on Text and Web.

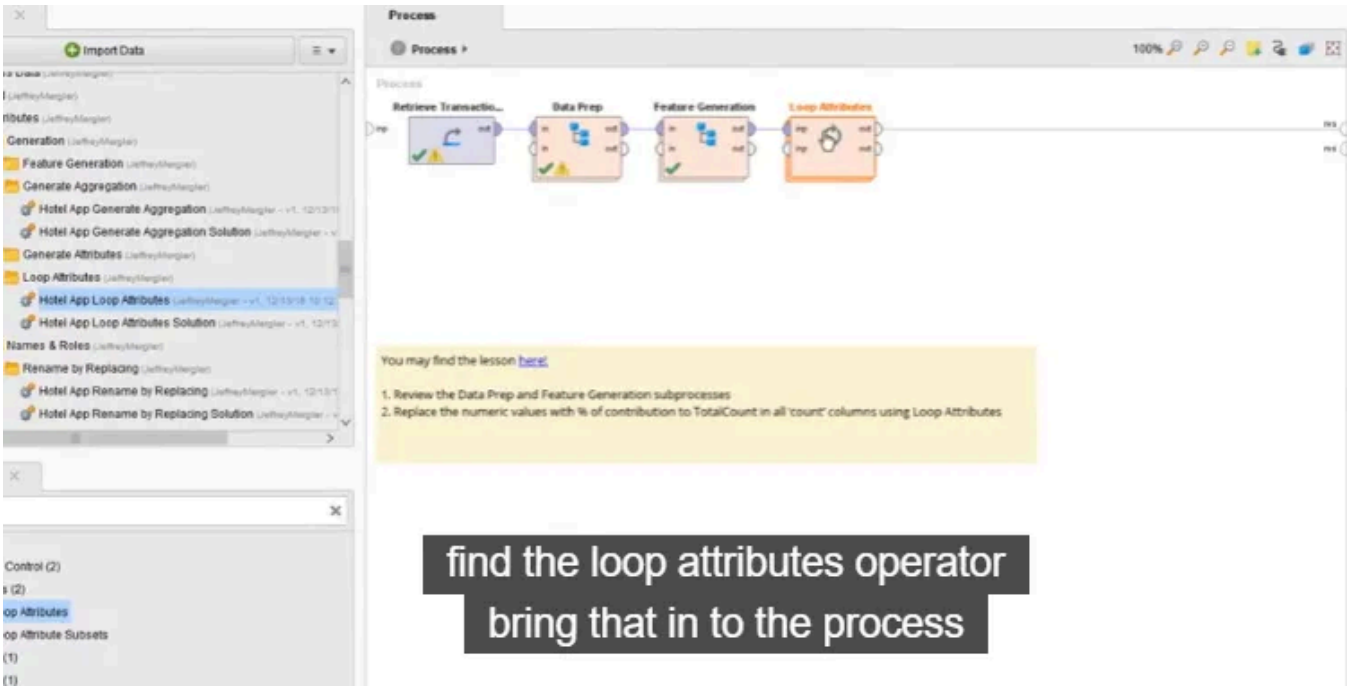


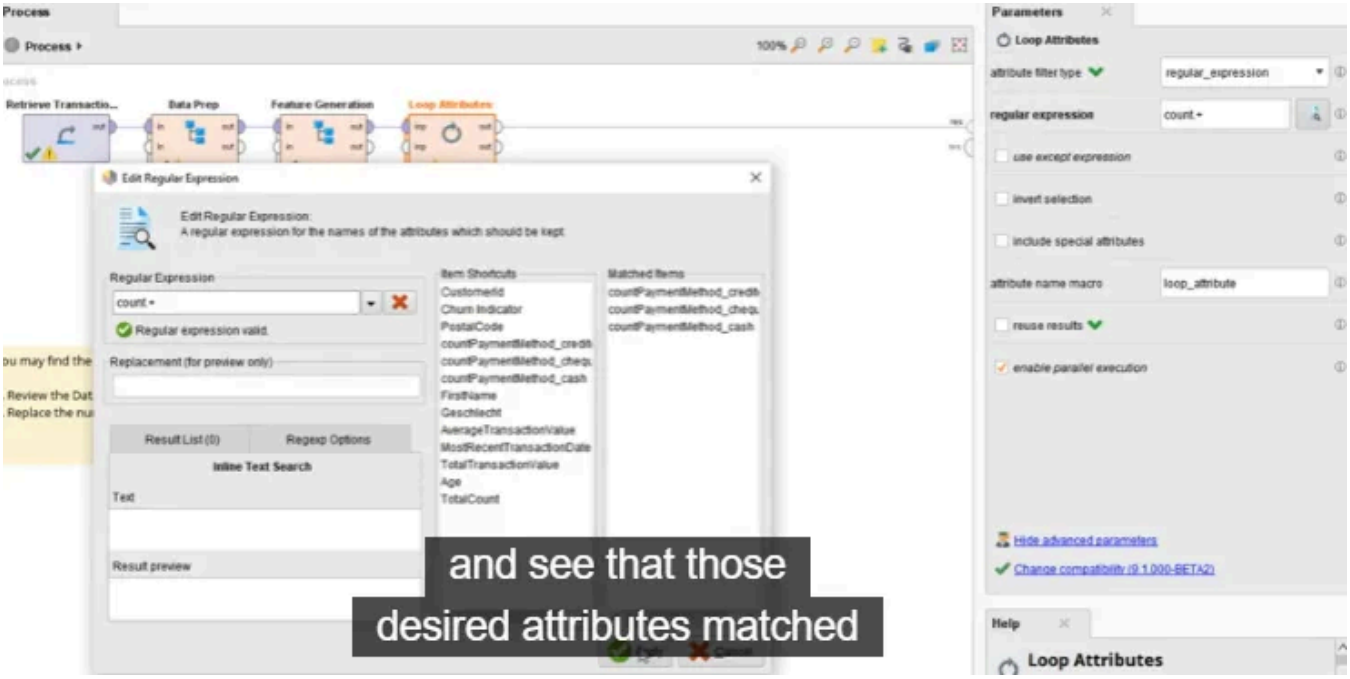


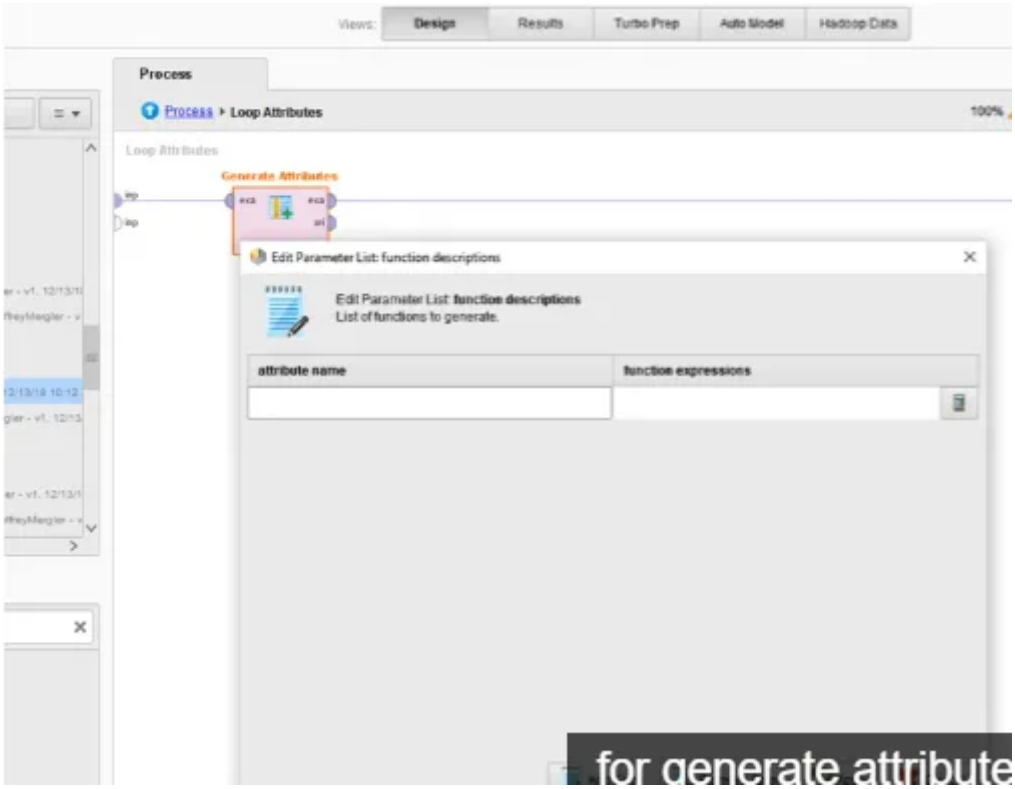


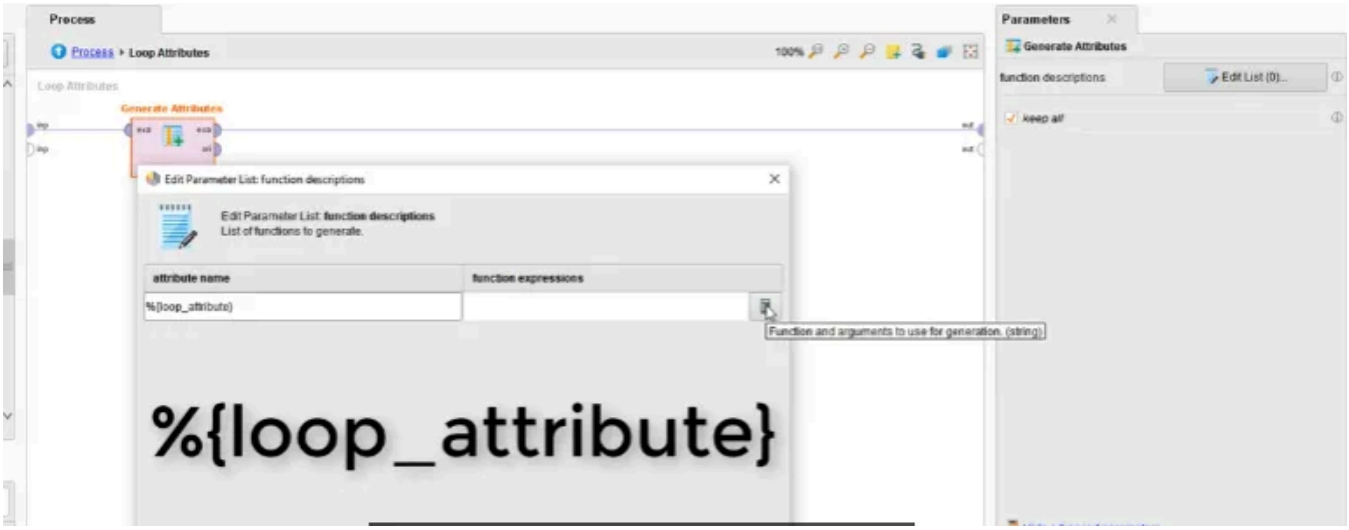
Loop Attributes

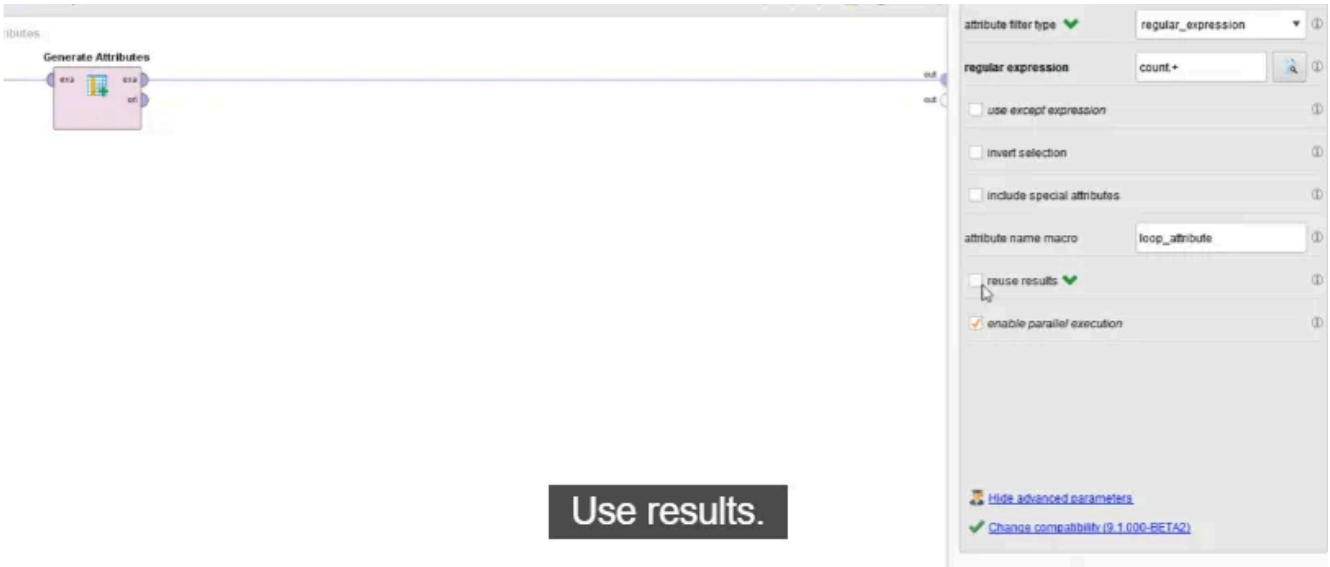
Watch and do! Perform Feature Generation for many features at once.

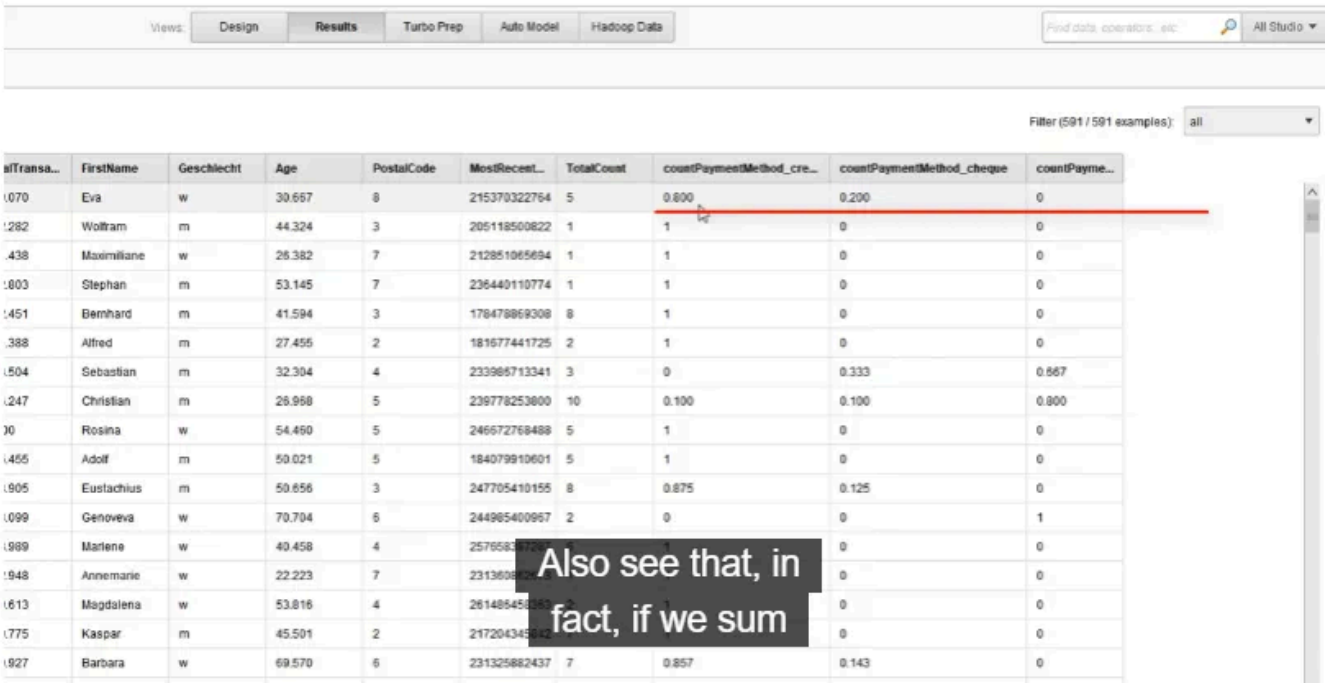

Import Data
Process
Process
Retrieve Transactio...
Data Prep
Feature Generation
Loop Attributes
Generation
Feature Generation
Generate Aggregation
Hotel App Generate Aggregation
Hotel App Generate Aggregation Solution
Generate Attributes
Loop Attributes
Hotel App Loop Attributes
Hotel App Loop Attributes Solution
Names & Roles
Rename by Replacing
Hotel App Rename by Replacing
Hotel App Rename by Replacing Solution
You may find the lesson here!
1. Review the Data Prep and Feature Generation subprocesses
2. Replace the numeric values with % of contribution to TotalCount in all 'count' columns using Loop Attributes
Control (2)
op Attributes
op Attribute Subsets
find the loop attributes operator
bring that in to the process


Process
Process
Retrieve Transactio...
Data Prep
Feature Generation
Loop Attributes
Edit Regular Expression
Edit Regular Expression
A regular expression for the names of the attributes which should be kept
Regular Expression
count
Regular expression valid.
Replacement (for preview only)
Result List (0)
Regexp Options
Inline Text Search
Text
Result preview
Item Shortcuts
CustomerId
Churn Indicator
PostalCode
countPaymentMethod_credit
countPaymentMethod_cheque
countPaymentMethod_cash
FirstName
Geschlecht
AverageTransactionValue
MostRecentTransactionDate
TotalTransactionValue
Age
TotalCount
Matched Items
countPaymentMethod_credit
countPaymentMethod_cheque
countPaymentMethod_cash
Parameters
Loop Attributes
attribute filter type
regular_expression
regular expression
count
use except expression
invert selection
include special attributes
attribute name macro
loop_attribute
reuse results
enable parallel execution
Hide advanced parameters
Change compatibility (9.1.000-BETA2)
Help
Loop Attributes
and see that those
desired attributes matched


Views:
Design
Results
Turbo Prep
Auto Model
Hadoop Data
Process
Process ▸ Loop Attributes
100%
Loop Attributes
Generate Attributes
Edit Parameter List: function descriptions
Edit Parameter List: function descriptions
List of functions to generate.
attribute name
function expressions
for generate attribute


Process
Process ▸ Loop Attributes
100%
Loop Attributes
Generate Attributes
Edit Parameter List: function descriptions
Edit Parameter List: function descriptions
List of functions to generate.
attribute name
function expressions
%{loop_attribute}
Function and arguments to use for generation. (string)
%{loop_attribute}
Parameters
Generate Attributes
function descriptions
Edit List (0)...
keep all

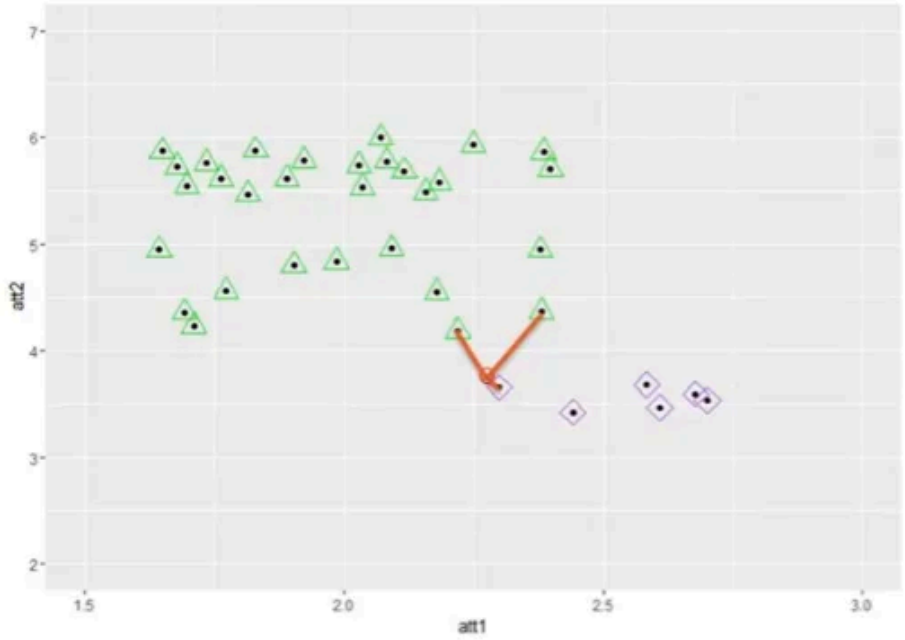

Generate Attributes
attribute filter type
regular_expression
regular expression
count.+
use except expression
invert selection
include special attributes
attribute name macro
loop_attribute
reuse results
enable parallel execution
Hide advanced parameters
Change compatibility (9.1.000-BETA2)
Use results.



Intro to Sampling

Learn ways to handle imbalanced data. Pay particular attention to how sampling works. There are times with very large data sets when data should be sampled down as a first-step, in order to make computation efficient.

Class sample size



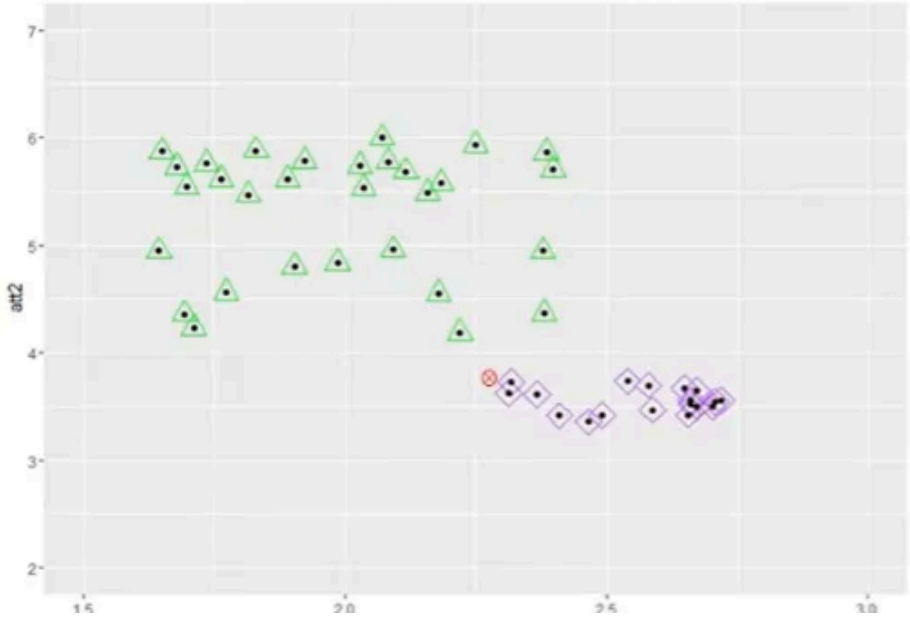
3-NN → Green

BUT

- One purple point is **really** close
- There are much more green points than purple ones

Sample numbers should be balanced!

Up-sampling (adding points + jitter)



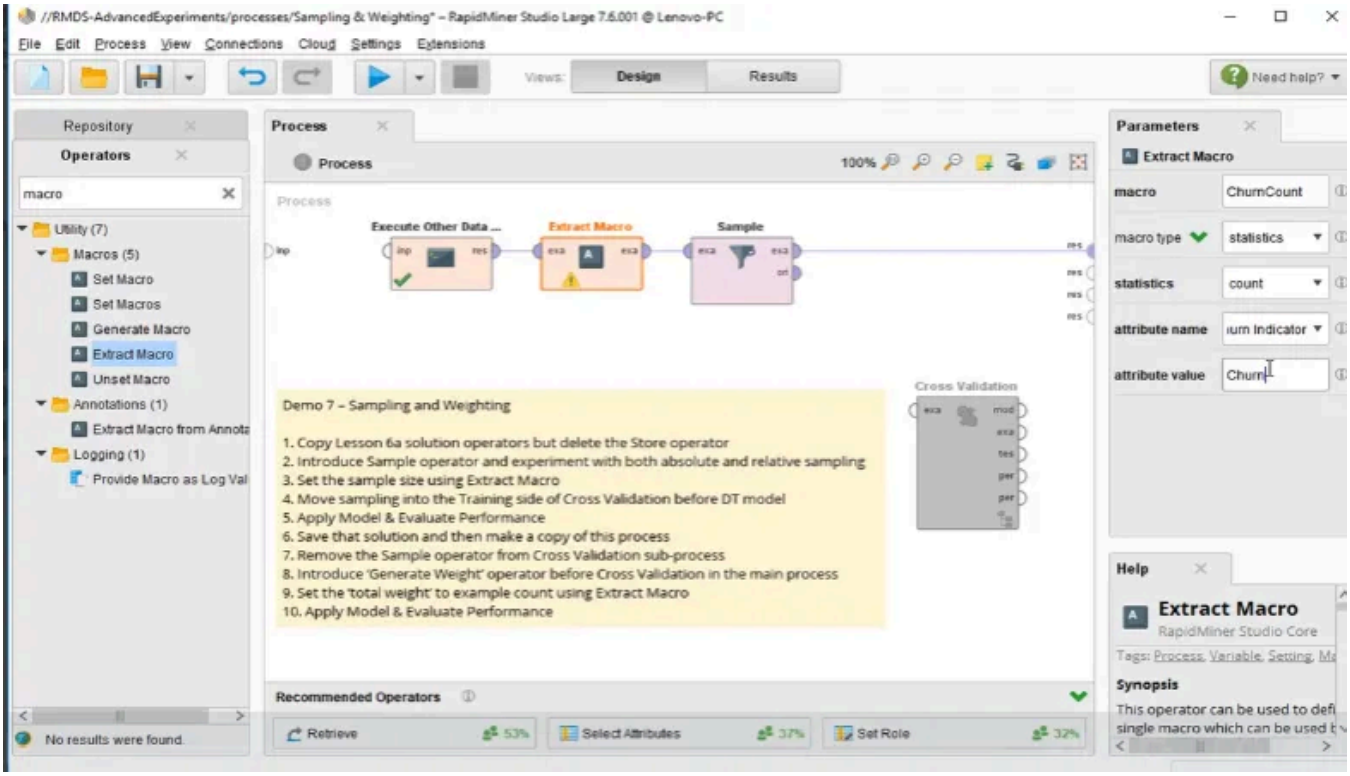
3-NN → Purple

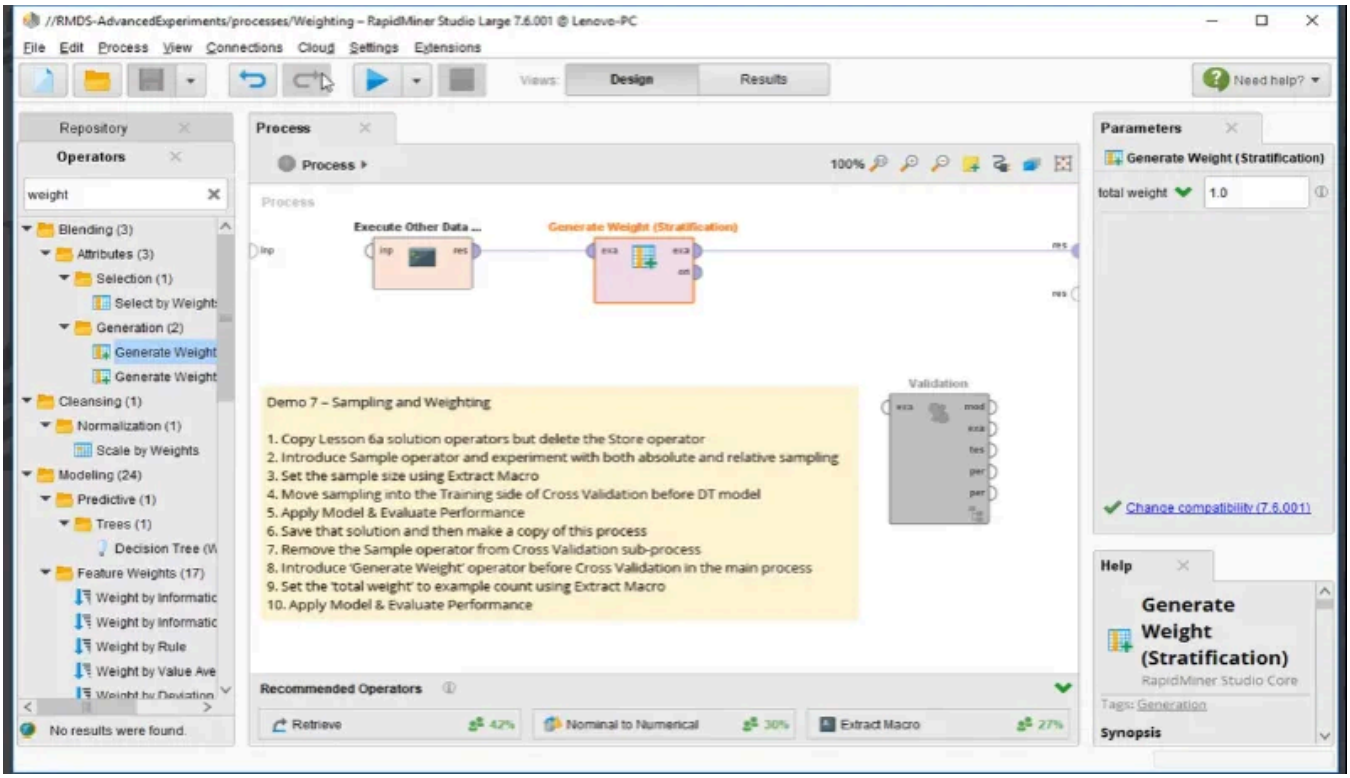
- There are more purple points
- As a whole they gained importance
- Classification near the green cluster is still accurate

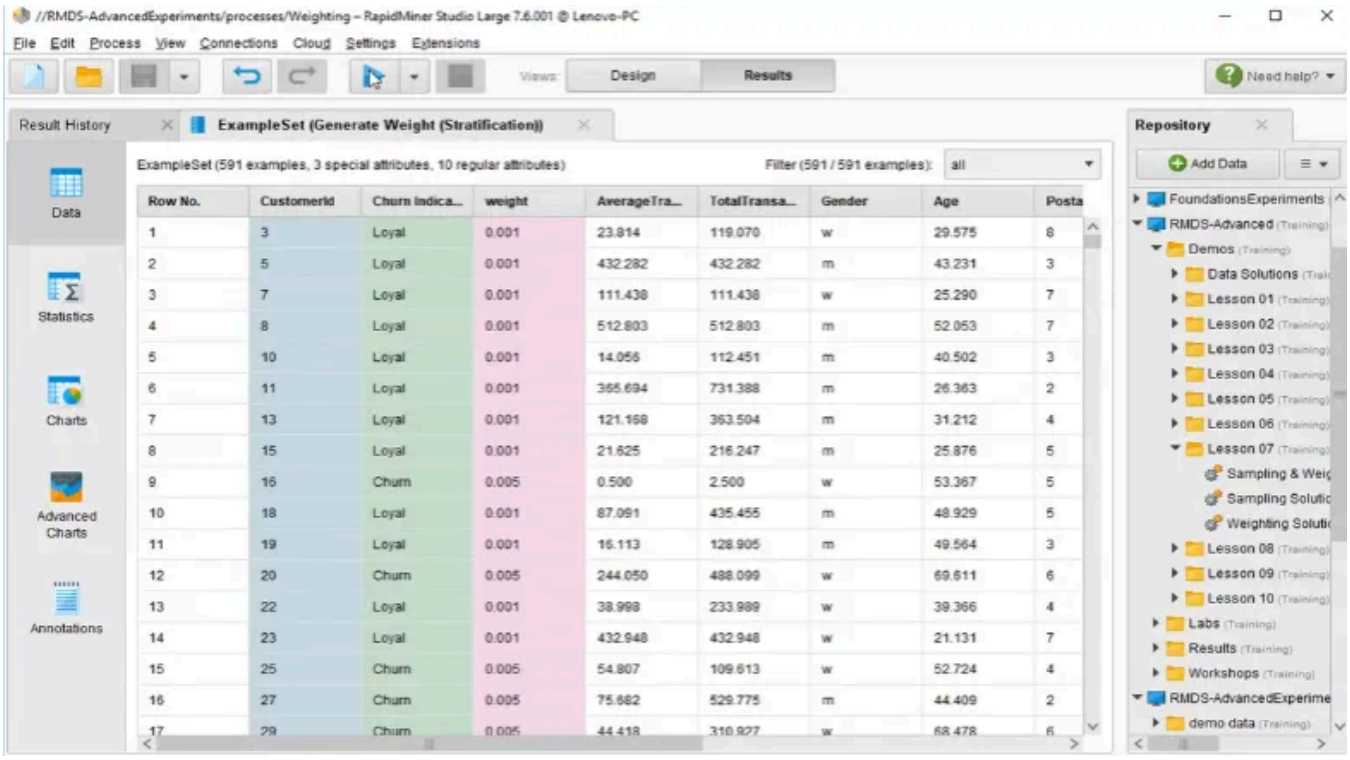
Generate Weight (Stratification)

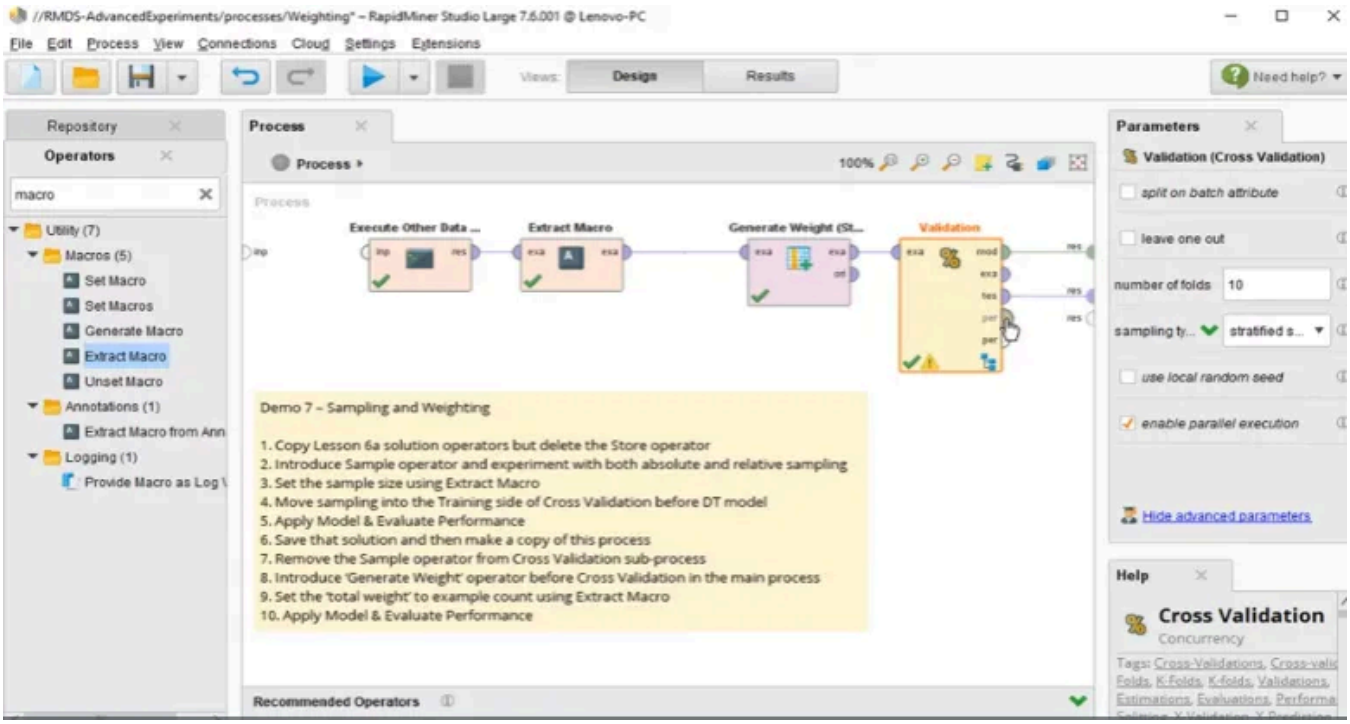
Sampling and Weighting

Watch and do! Downsample and weight your data in RapidMiner.





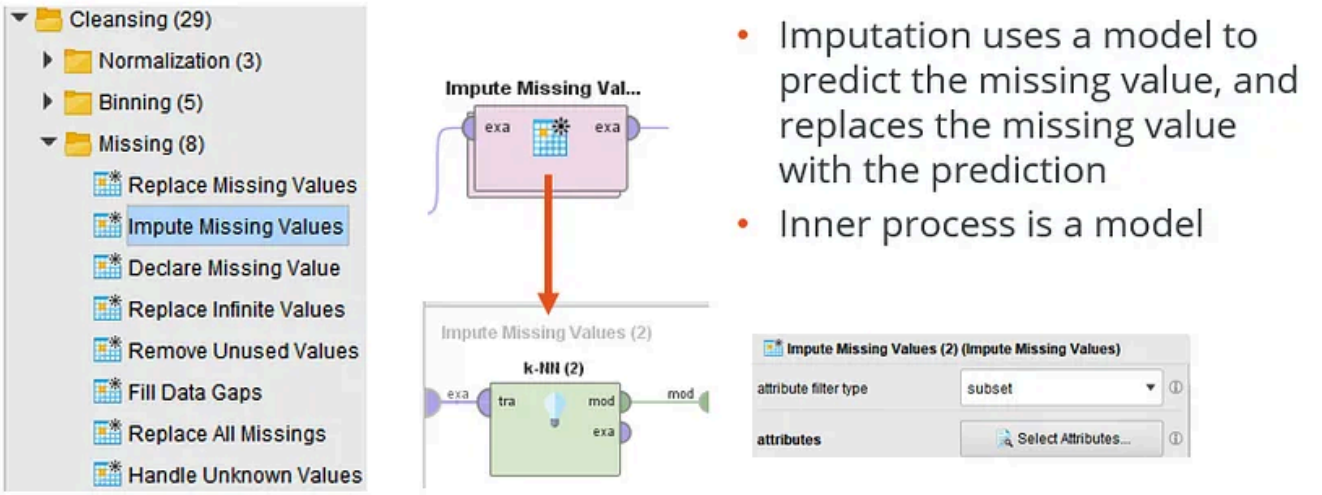




Replace Missing Values

Watch and do! Use RapidMiner to replace missing values in your data and get it ready for Machine Learning.

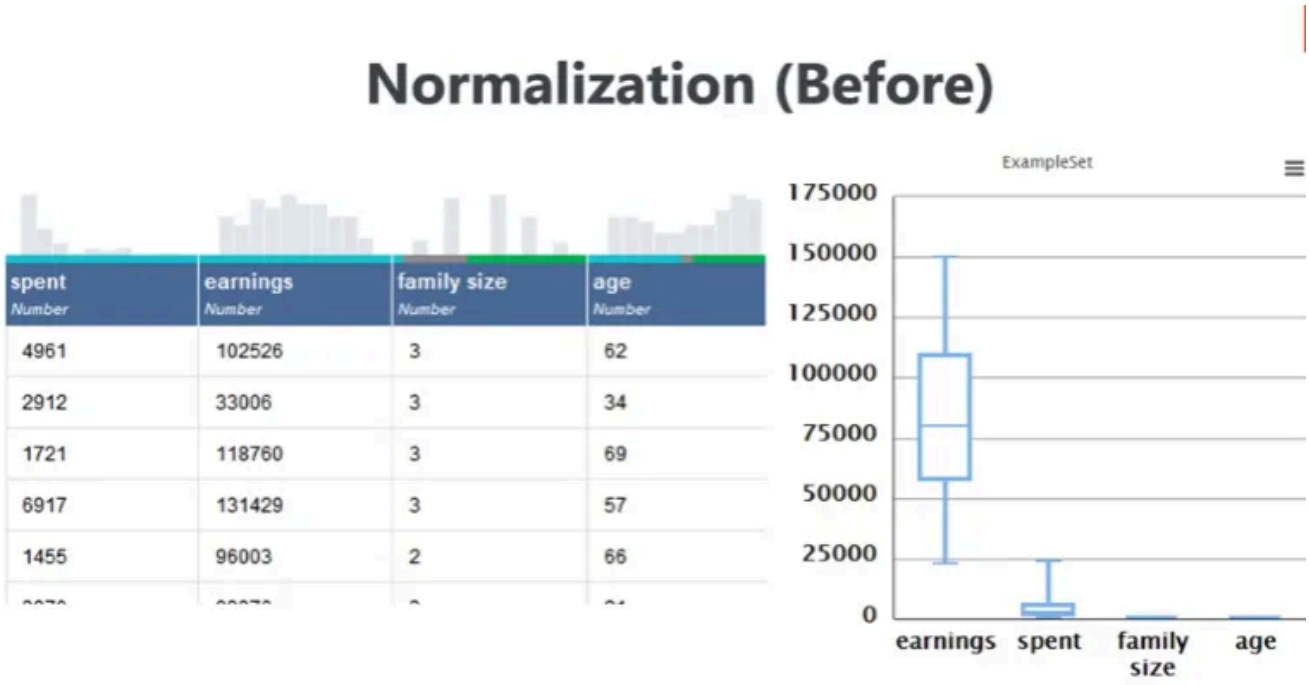
Imputation



Imputation is most valuable in small example sets that have a high proportion of missing values. It should be considered a specialized tool. This is a powerful technique, but care is needed. If a predictive model is based on predicted values, the variance and stability of the model can be hard to determine and control.

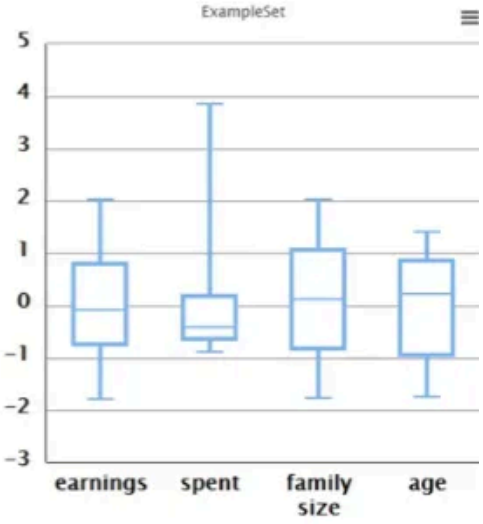
Normalizing Data

Learn about normalizing your data.

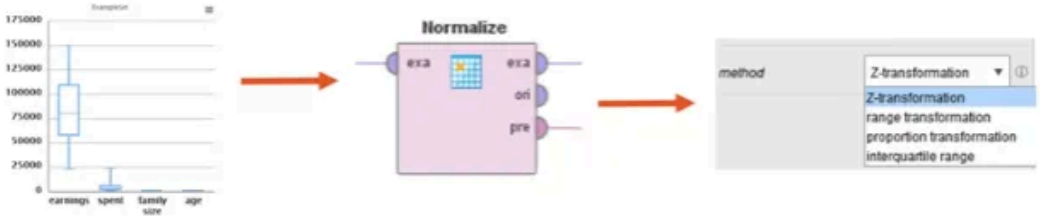


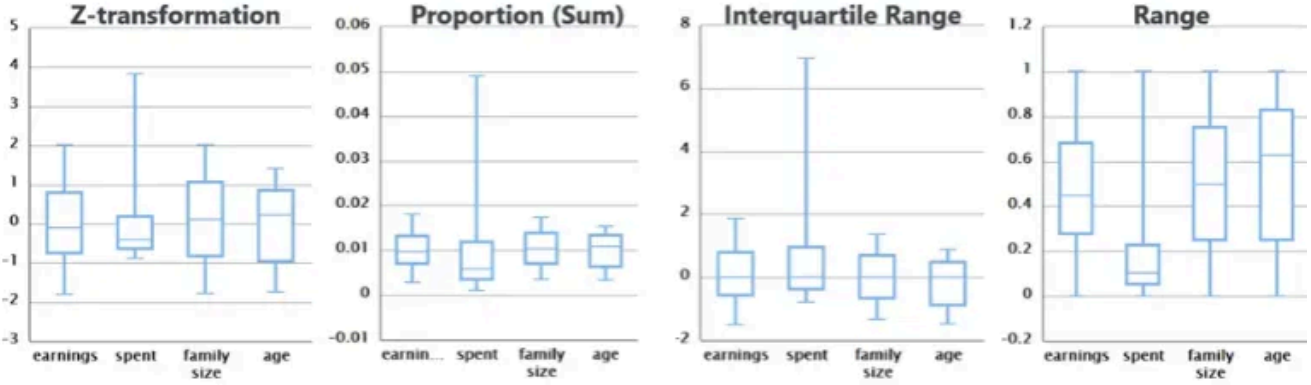
Normalization (After)

spent *Number*	earnings *Number*	family size *Number*	age *Number*
0.016	0.583	0.114	0.986
-0.397	-1.507	0.114	-0.645
-0.638	1.071	0.114	1.393
0.411	1.451	0.114	0.695
-0.691	0.387	-0.833	1.219

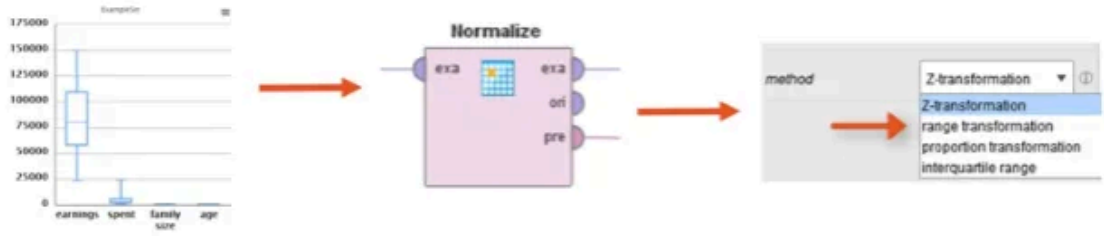


Normalization Methods

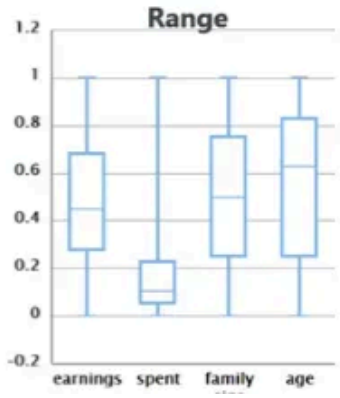




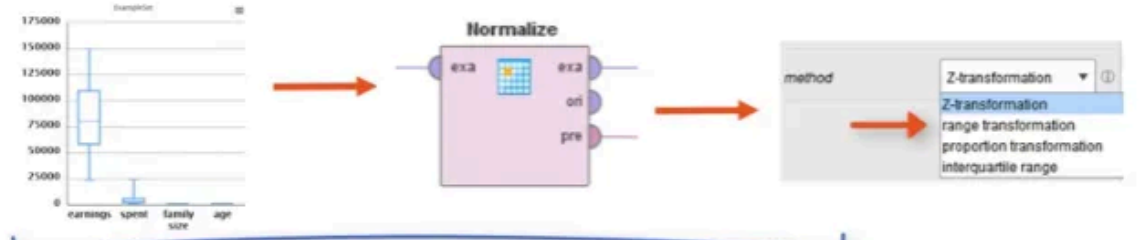
Normalization Methods



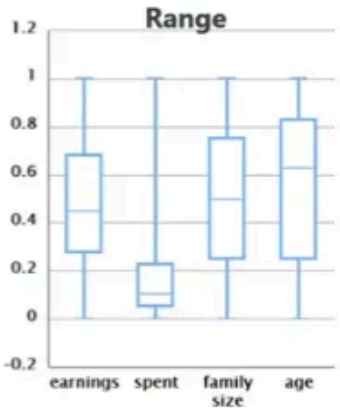
Set the max value to 1
Set the min value to 0
Other values are place in proportion to min and max



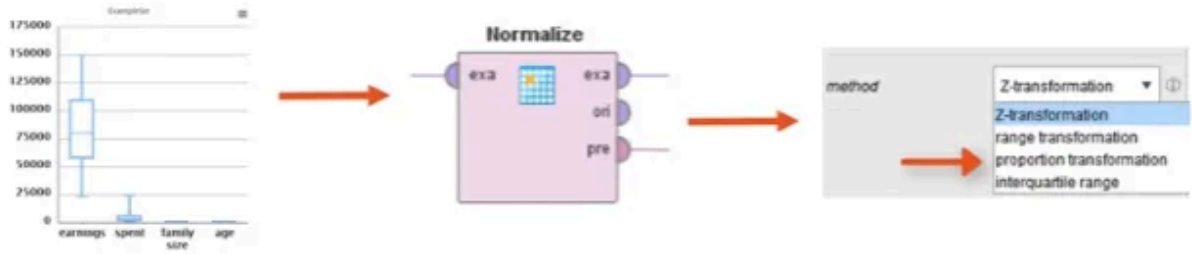
Normalization Methods

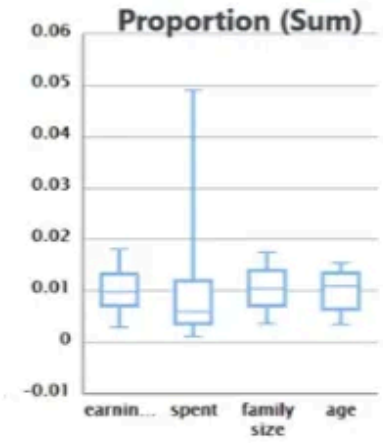


33006 → 0
96003 → ~0
118760 → ~0
?,000,000,000 → 1

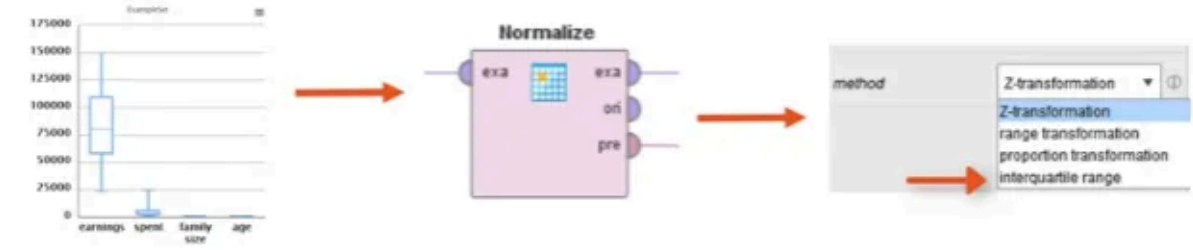


Normalization Methods

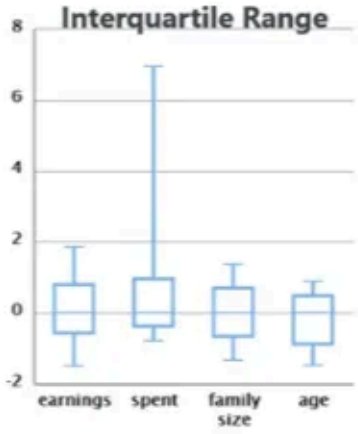




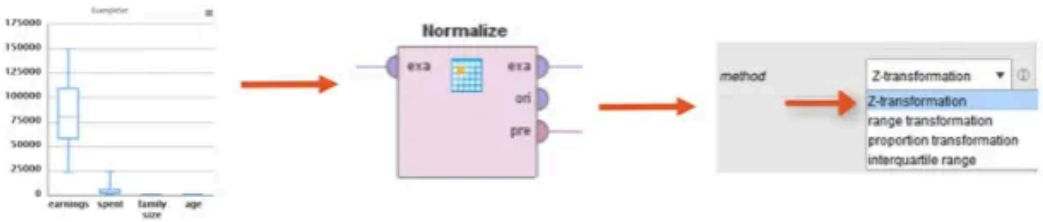
Normalization Methods

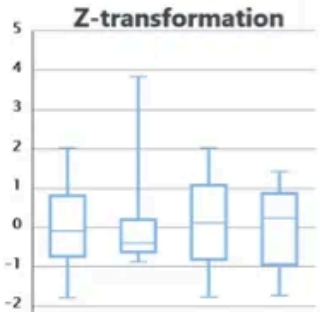


First quartile gets -1 and Third quartile gets +1



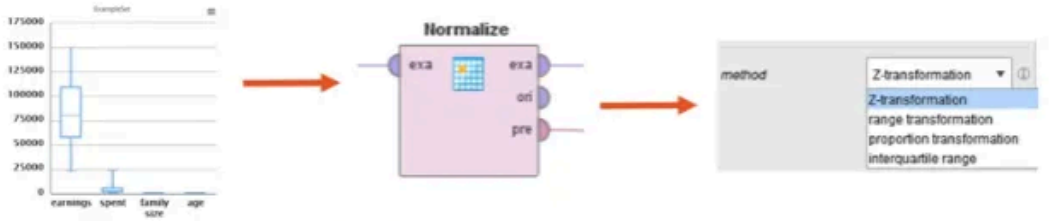
Normalization Methods

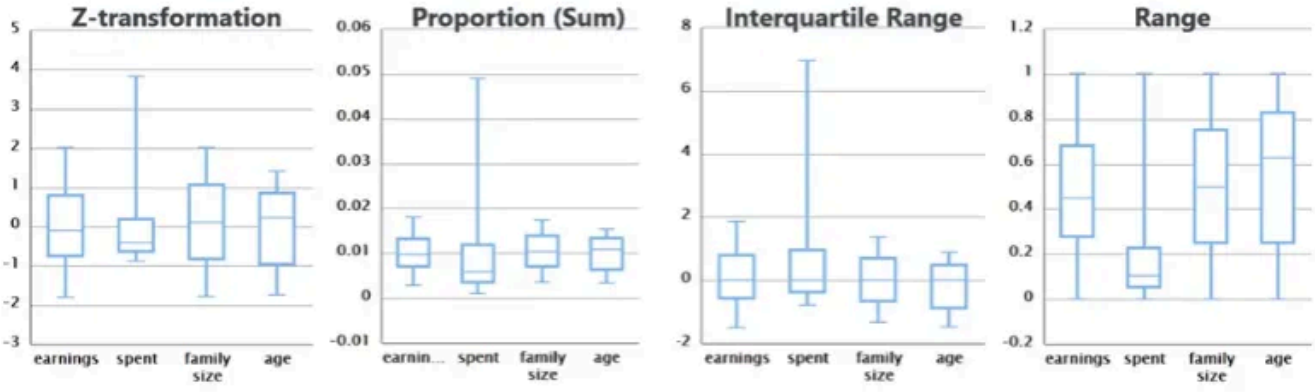




Subtract mean and divide by standard deviation

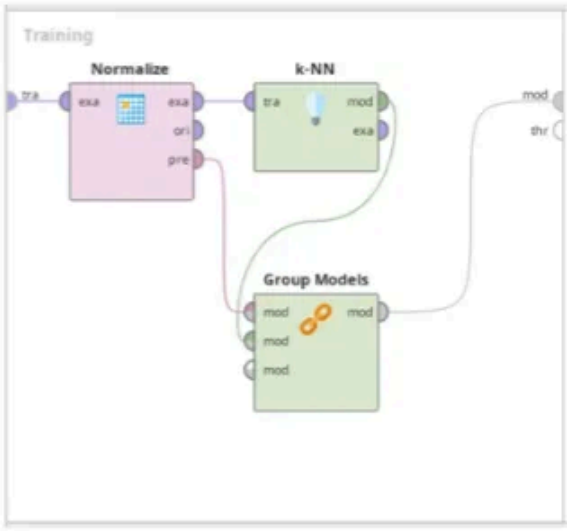
Normalization Methods

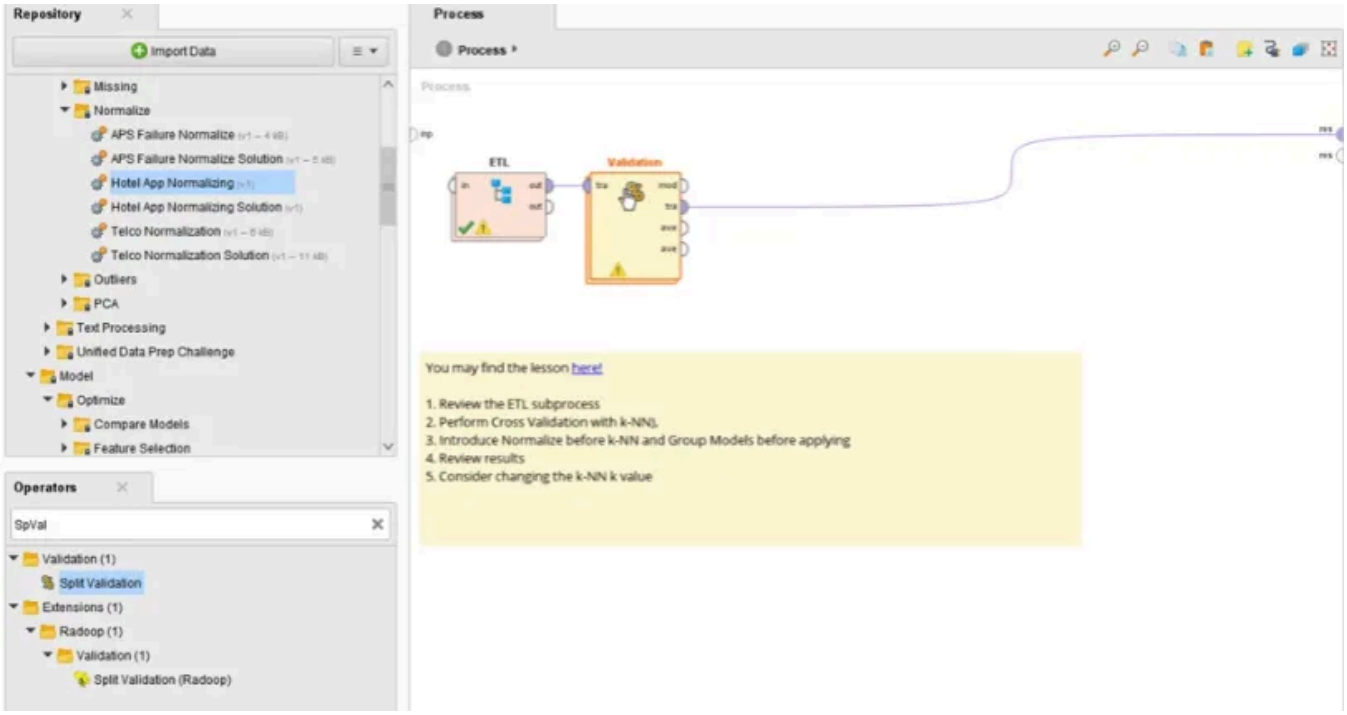


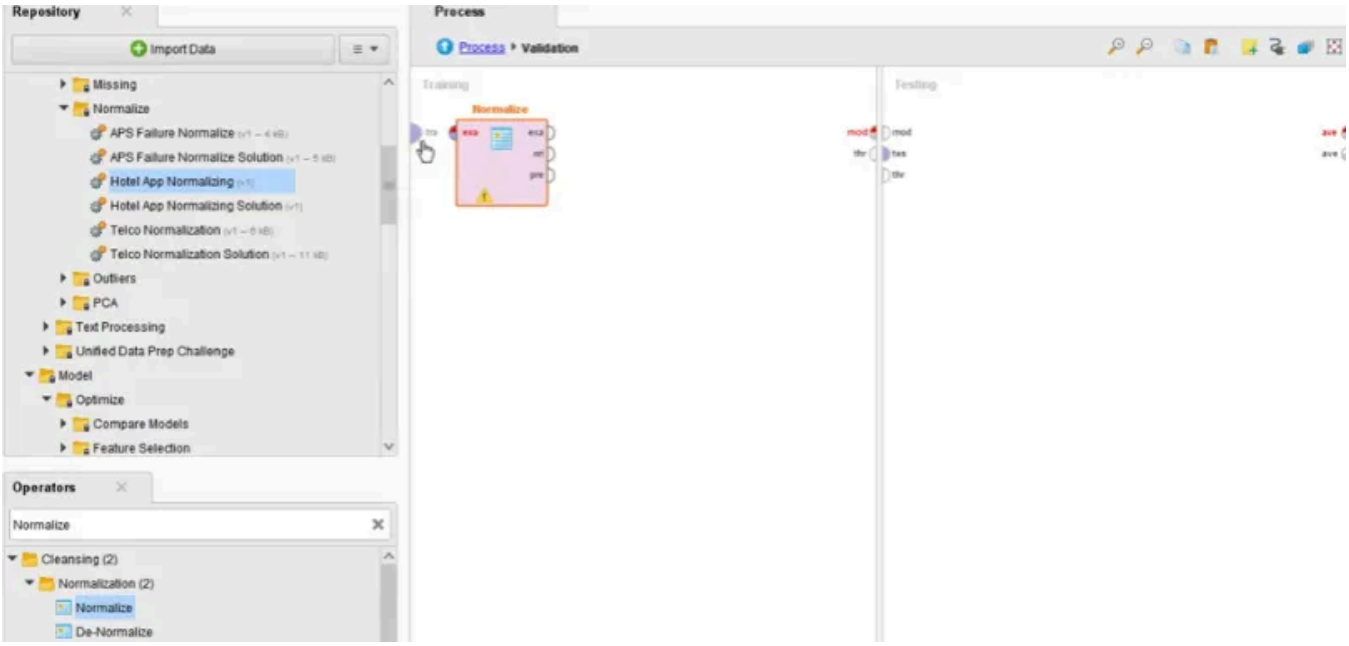


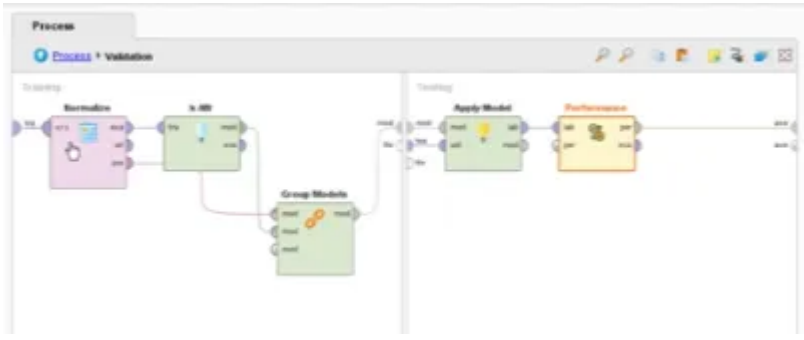
Group Models

This operator groups the given models into a **single** combined model. When this combined model is applied, it is equivalent to applying the original models in their respective order.

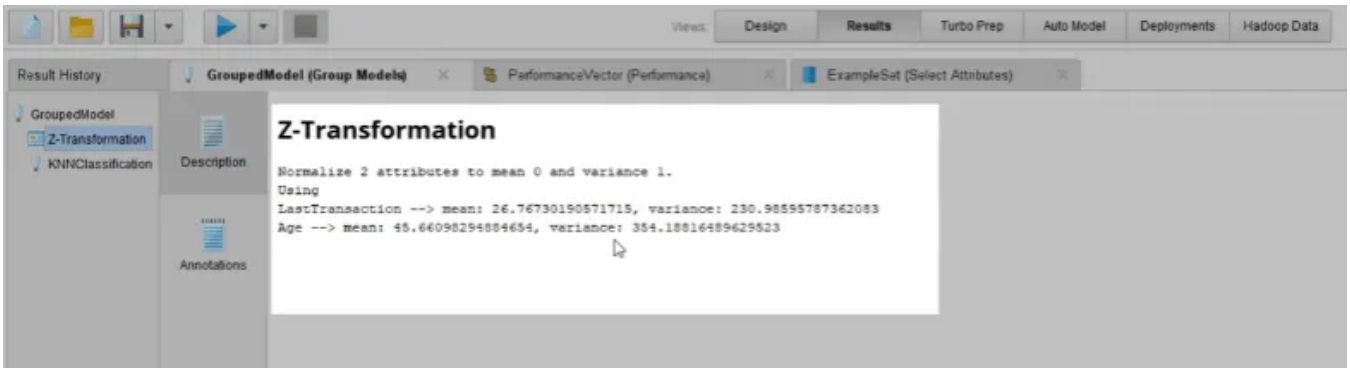


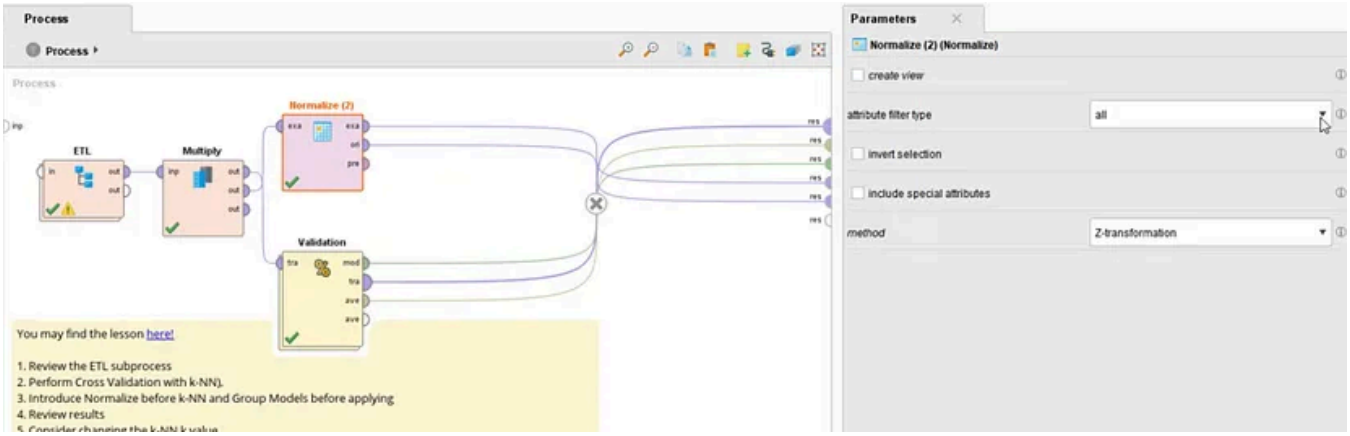


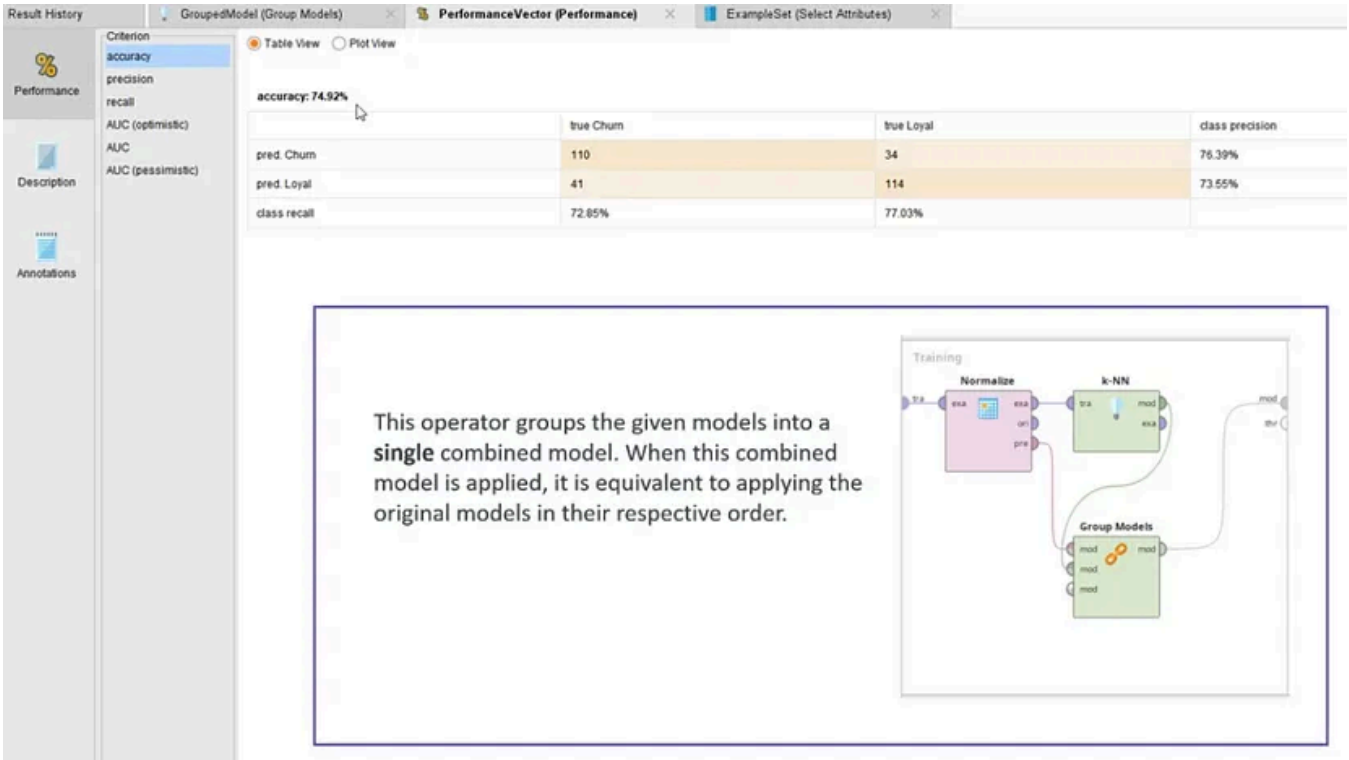




Normalize後的KNN處理，然後接下來就是從train model到透過testing model給模型評分。







Rapidminer | Machine Learning | Algorithms | Data Science | Feature Engineering

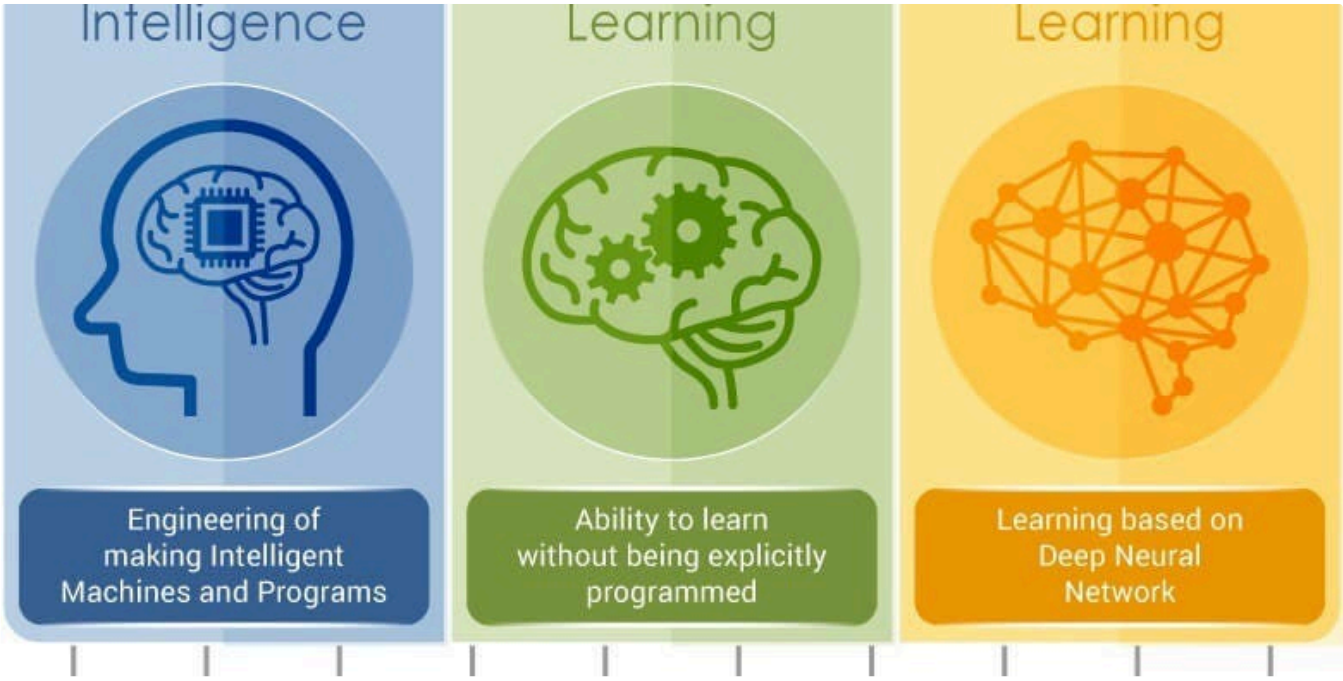
Follow

Written by KevinLuo

178 Followers

知曉很多種資料處理，可BI或AI化的軟體和工具。主要用的程式語言是python和R 偶爾用C++ Ig:(可在上面找到我) AIA第九屆經理人班 立志當個厲害的podcaster!

More from KevinLuo



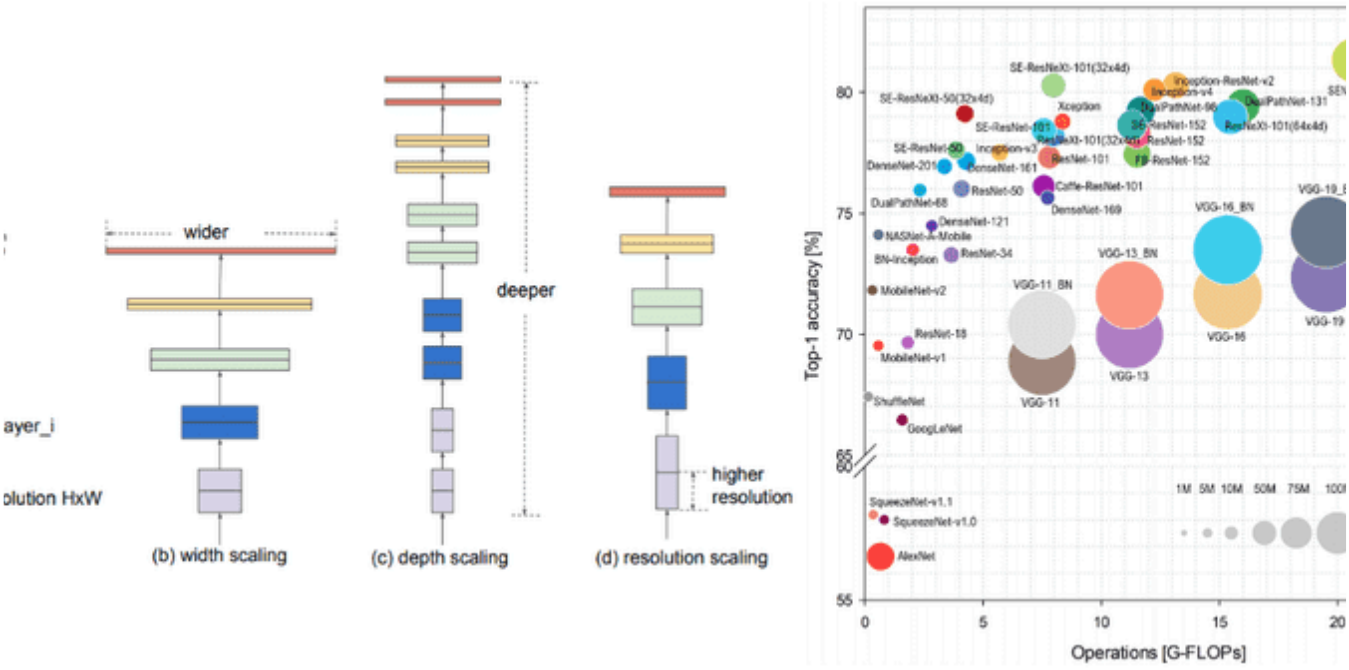

KevinLuo

Top 10 您應該要學會的深度學習演算法 (Fundamental Review Series)

首先，我們來簡單回想一下...

10 min read · Dec 1, 2021


12



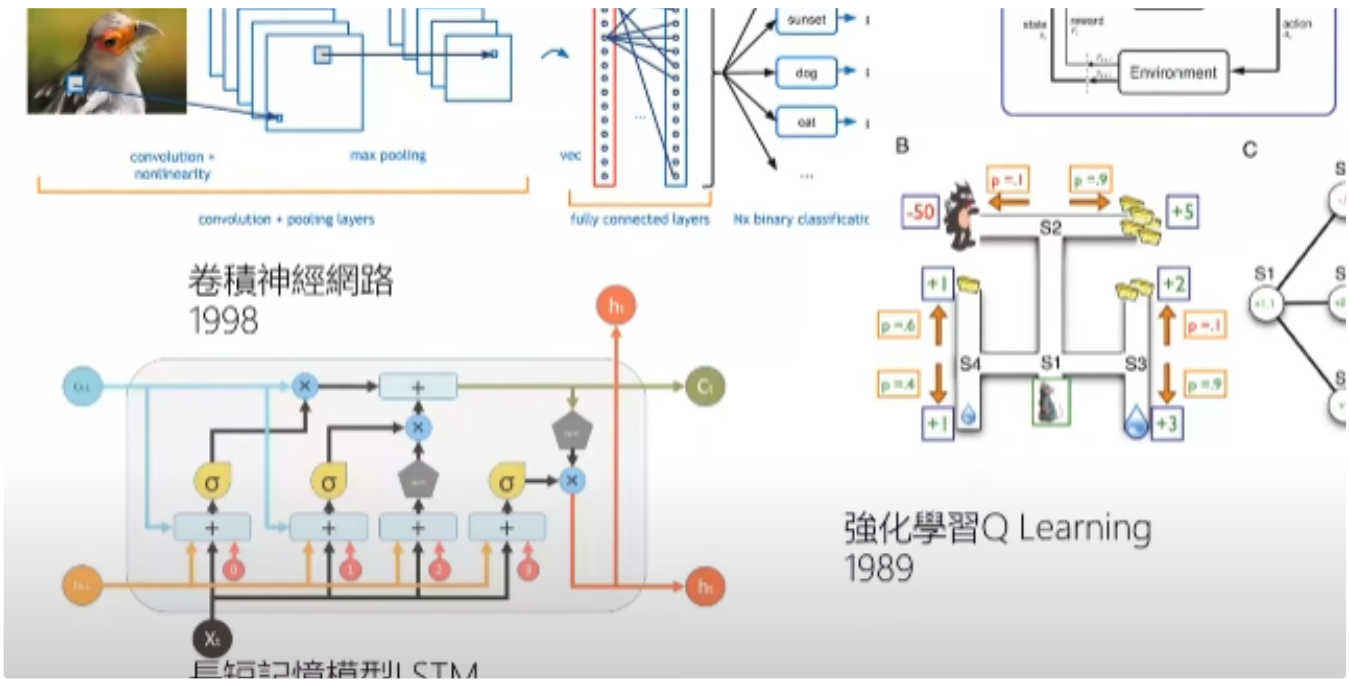
KevinLuo

好用的深度學習CNN預訓練模型框架總整理: 從AlexNet到EfficientNet(ML隨筆)

各位好，我是Kevin. 忽然發現頗久沒有更新我的medium...可能都快長蜘蛛網了(驚!

19 min read · Feb 16, 2022

56



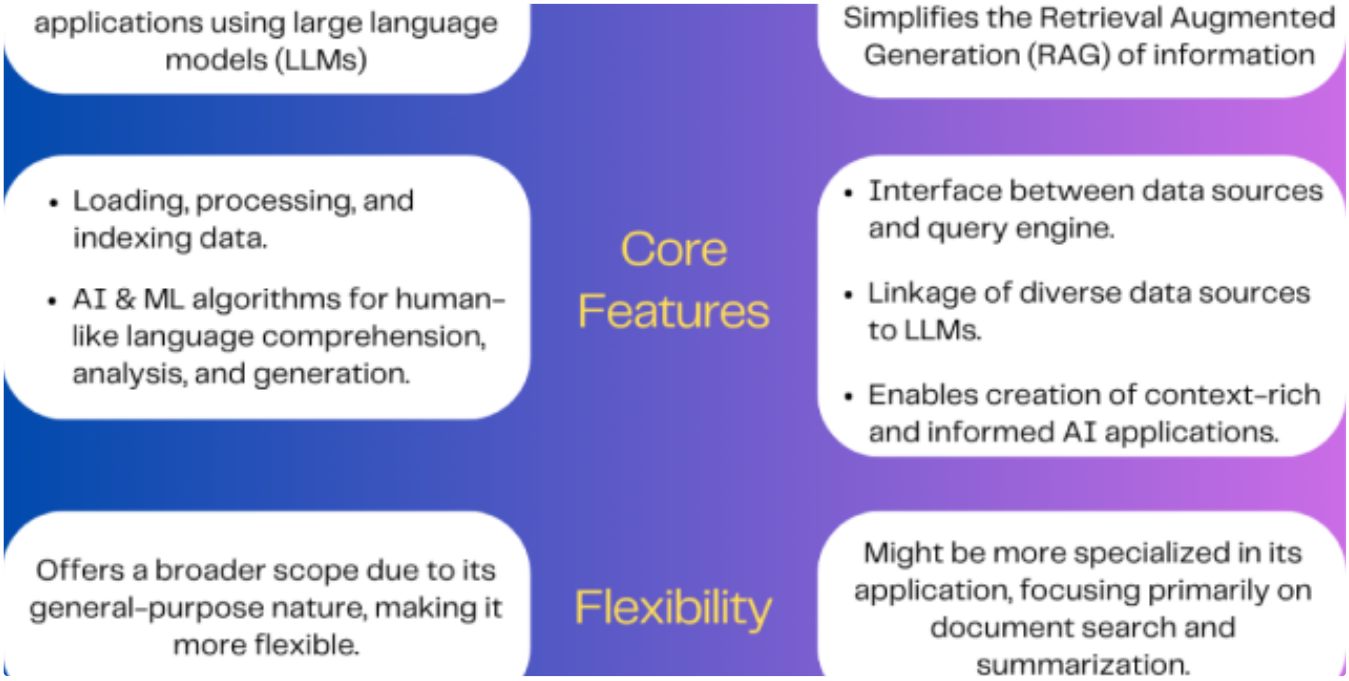
KevinLuo

深度學習-各種新優化器介紹-LookAhead&Ranger&LARS

Outline

22 min read · Oct 31, 2021

108



KevinLuo

LangChain V.S LlamaIndex

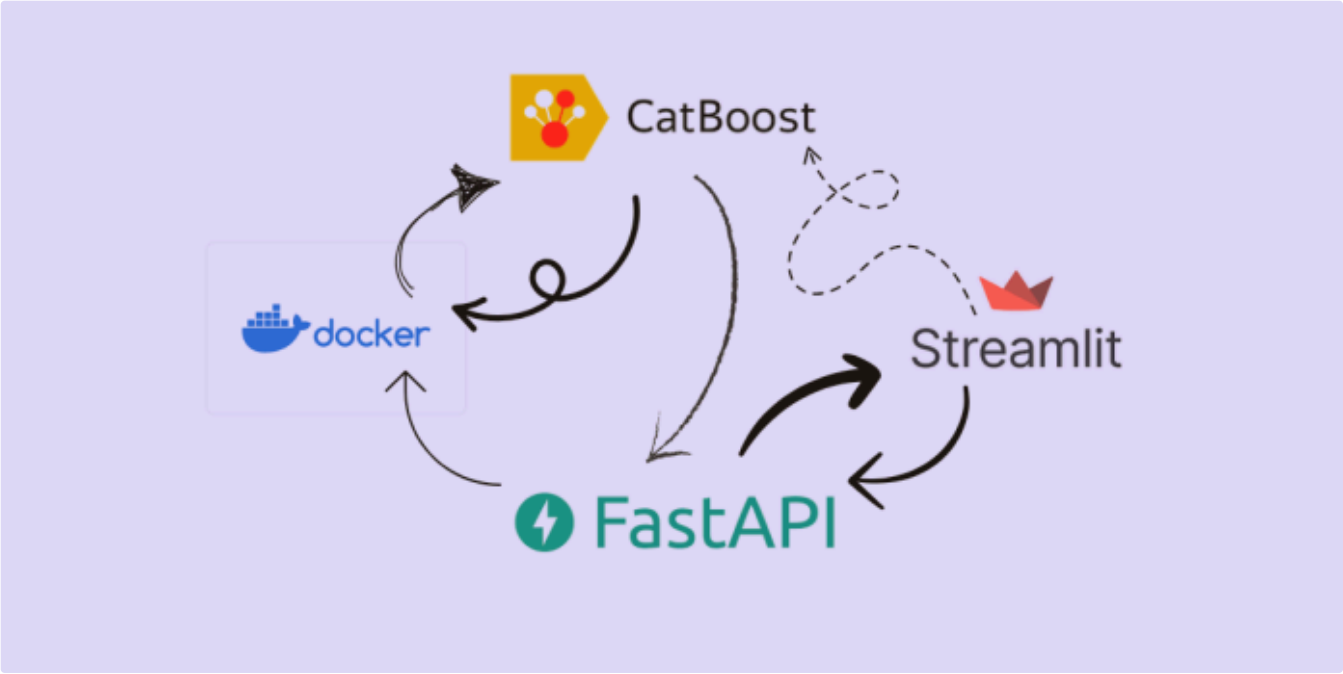
LangChain

3 min read · Mar 10, 2024

53

See all from KevinLuo

Recommended from Medium



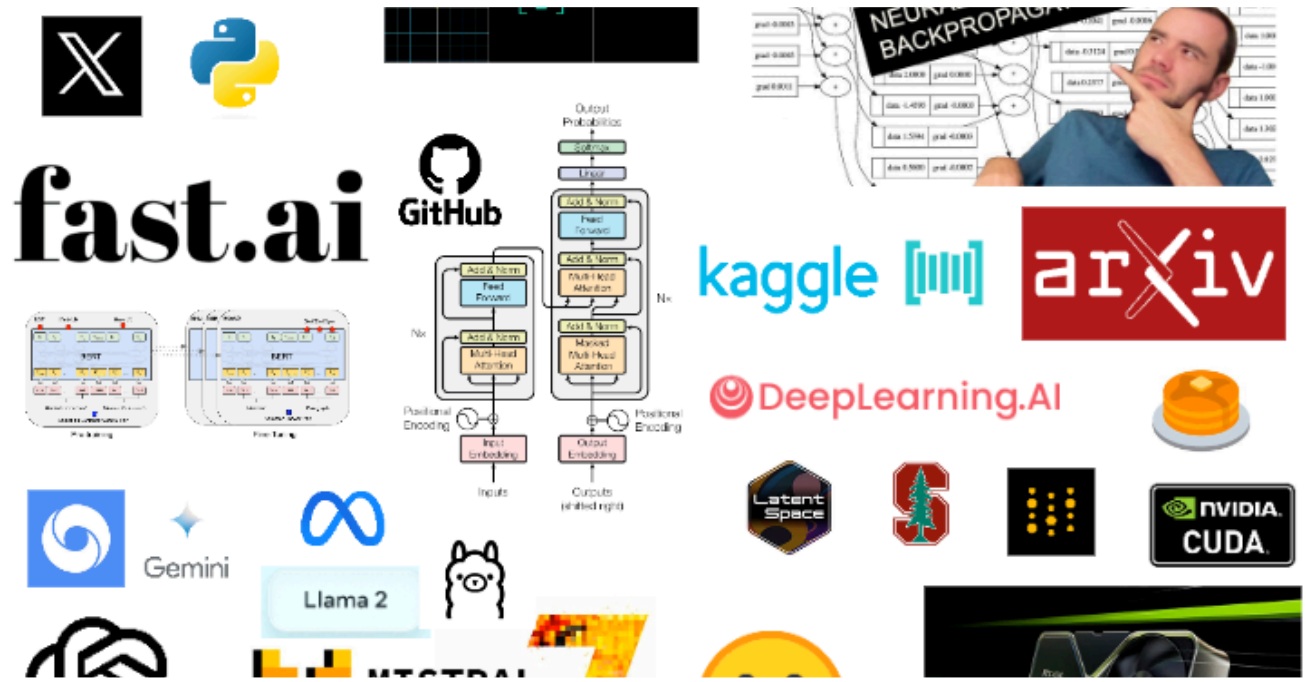
Ramazan Olmez

End-to-End Machine Learning Project: Churn Prediction

The main objective of this article is to develop an end-to-end machine learning project. For a model to be truly useful, it needs to be...

18 min read · Feb 23, 2024

204 1



Benedict Neo in bitgrit Data Science Publication

Roadmap to Learn AI in 2024

A free curriculum for hackers and programmers to learn AI

11 min read · Mar 11, 2024

11.1K 113

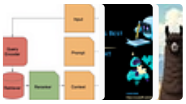
Lists

Predictive Modeling w/ Python

20 stories · 1141 saves

Practical Guides to Machine Learning

10 stories · 1373 saves

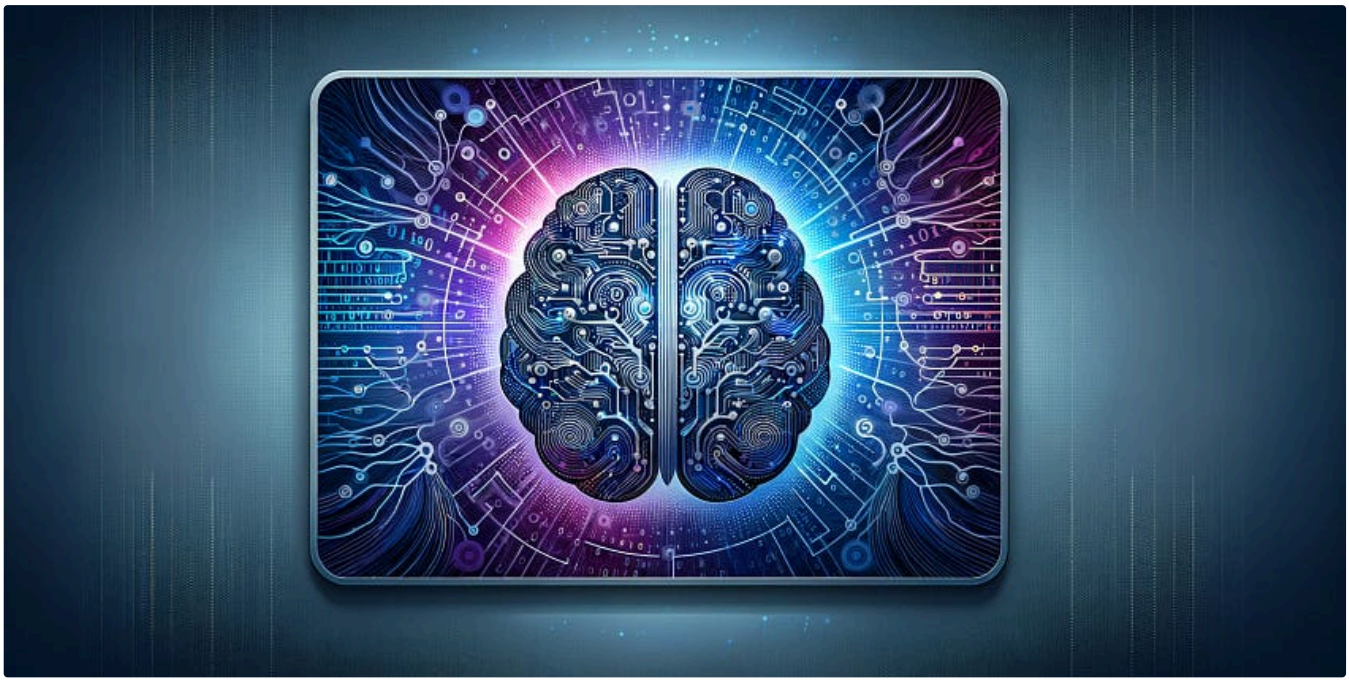


Natural Language Processing

1417 stories · 910 saves

data science and AI

40 stories · 140 saves



Cristian Leo in Towards Data Science

The Math Behind Neural Networks

Dive into Neural Networks, the backbone of modern AI, understand its mathematics, implement it from scratch, and explore its applications

· 28 min read · Mar 29, 2024

3.2K 21

Jason Roell in Stackademic

Ultimate Python Cheat Sheet: Practical Python For Everyday Tasks

(My Other Ultimate Guides)

34 min read · Jan 30, 2024

4.4K 38

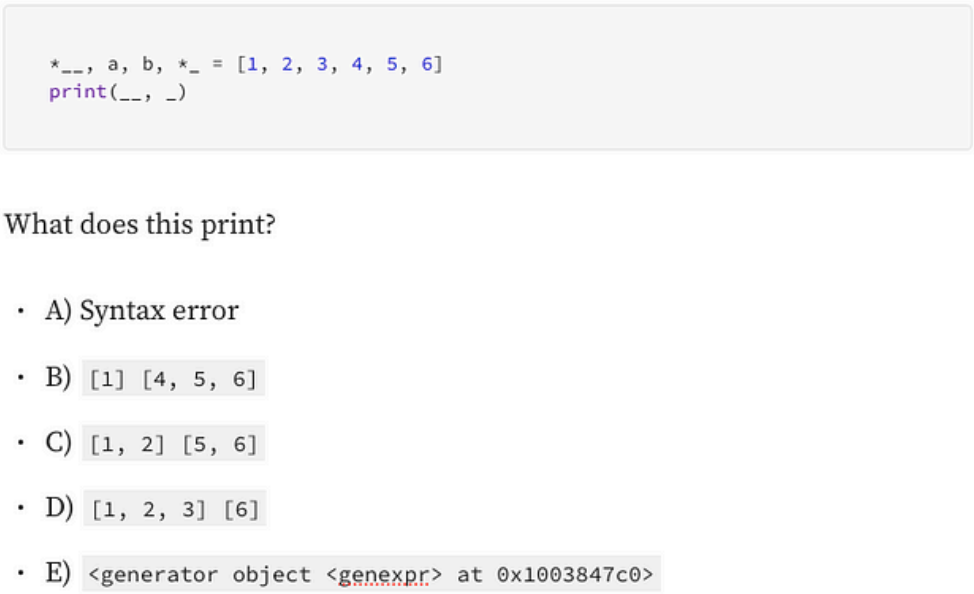
Somnath Singh in Level Up Coding

The Era of High-Paying Tech Jobs is Over

The Death of Tech Jobs.

✦ · 14 min read · Apr 1, 2024

10K 254



Liu Zuo Lin

You’re Decent At Python If You Can Answer These 7 Questions Correctly

No cheating pls!!

✦ · 6 min read · Mar 6, 2024

3.4K 19

See more recommendations