

Rapid Miner -Data Engineering Master certificate(1)

KevinLuo · [Follow](#)

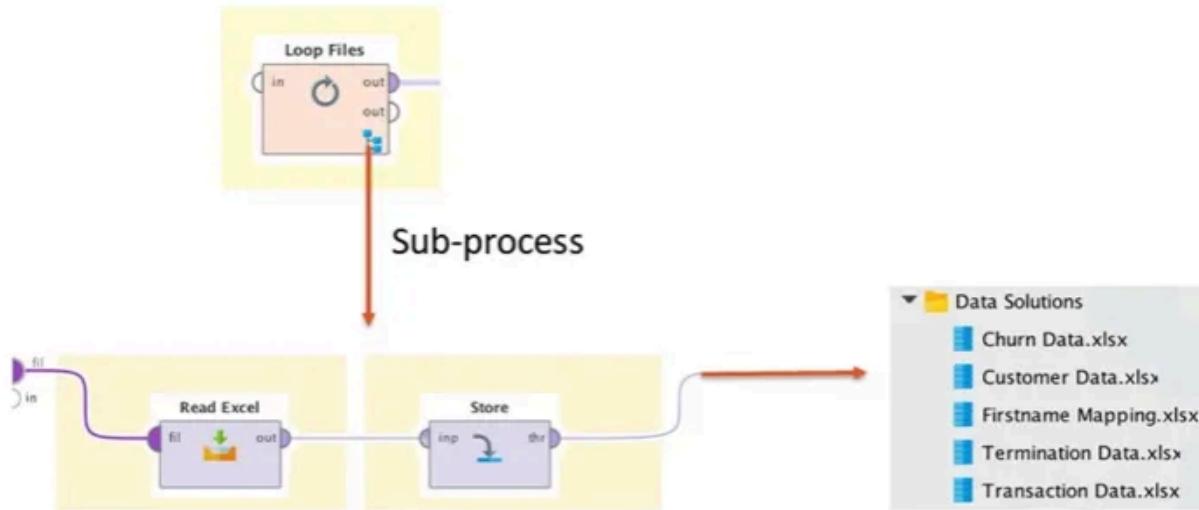
8 min read · Oct 18, 2021

 Listen Share

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.

Loading multiple data files



Raw Data Files

Get a quick introduction to the multiple raw data files.

Extracting Data

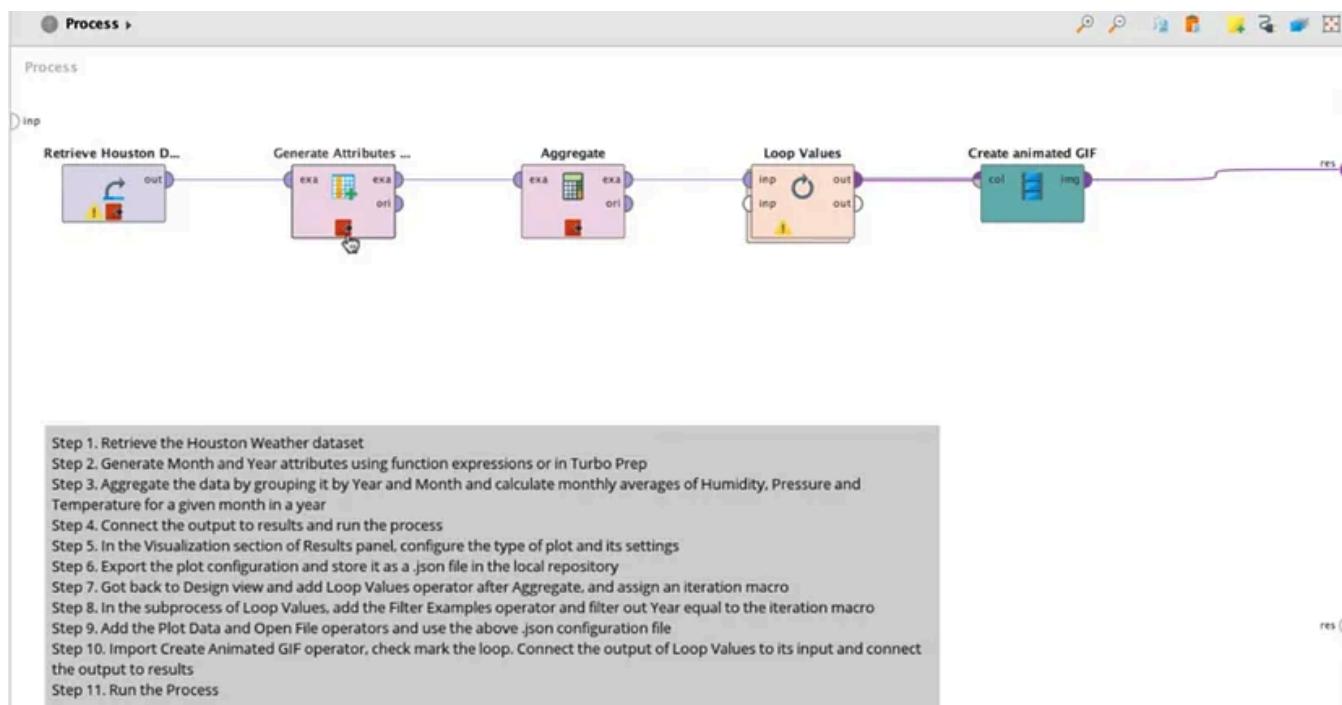
Watch and do! Loop over many files, and use macros to store them with useful names. Pay attention to how collections can be created. Collections will also be

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 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 promising models into one optimization each. We then execute them one by one. To make sure we can compare them after 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

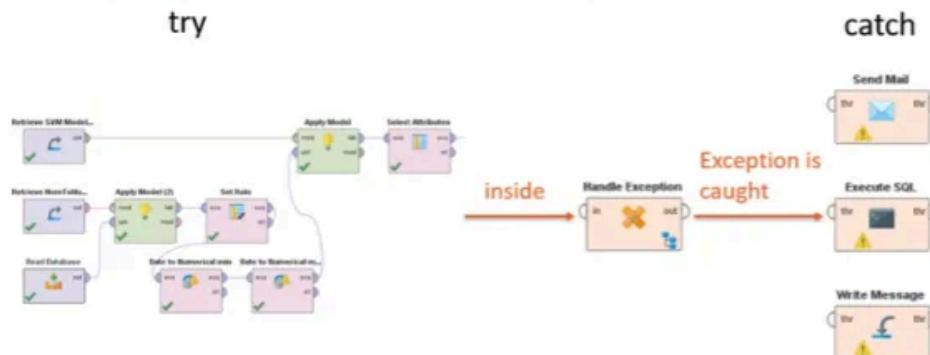
Exception handling in RapidMiner processes is a powerful way to react to unexpected situations or errors that only happen from time to time. You can execute operators when an exception occurs, and end the process using Throw Exception.

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

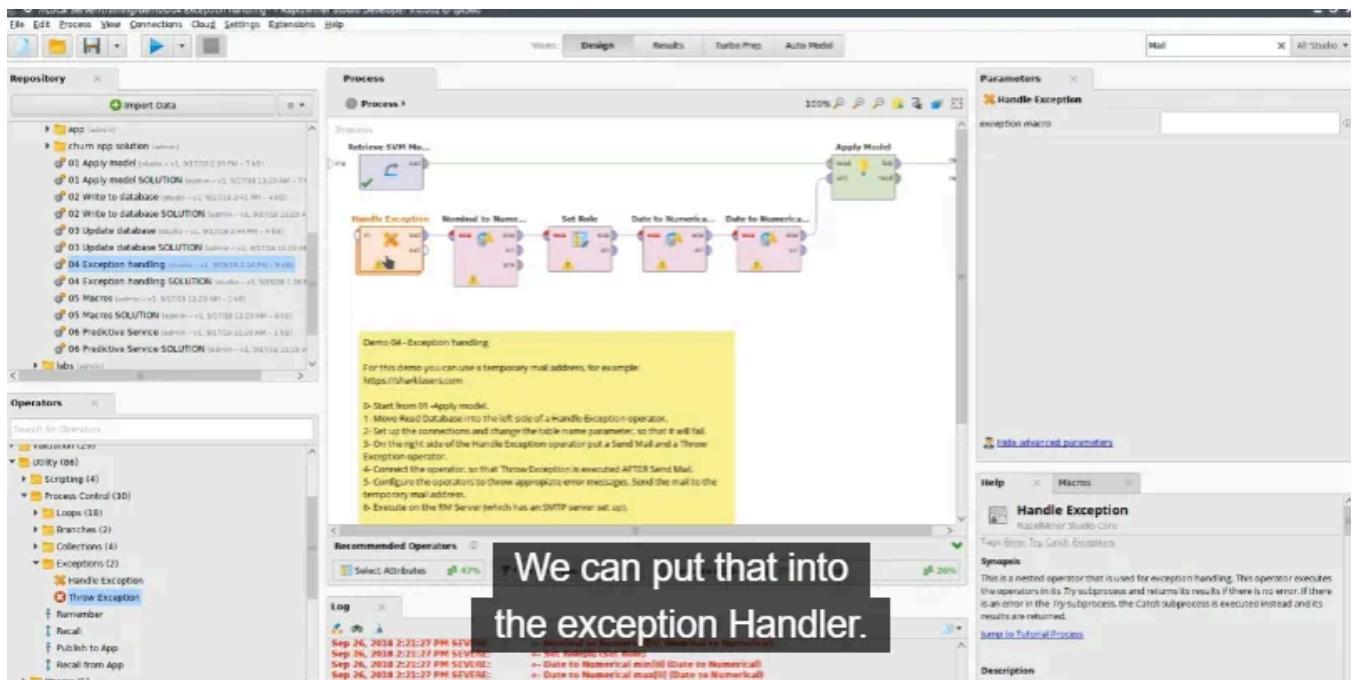
What is exception handling?



- **Exception handling** is the process of responding to the occurrence of *exceptions* (anomalous or exceptional conditions requiring special processing), often changing the normal flow of process execution.



The screenshot shows the KNIME interface with a process titled "Handle Exception". The "Try" section contains a "Read Database" operator. The "Catch" section contains a "Send Mail" operator and a "Throw Exception" operator. A "Process Failed" dialog box is open, stating "Process failed. The process failed with a database error." The "Show Details" button is visible in the dialog box.



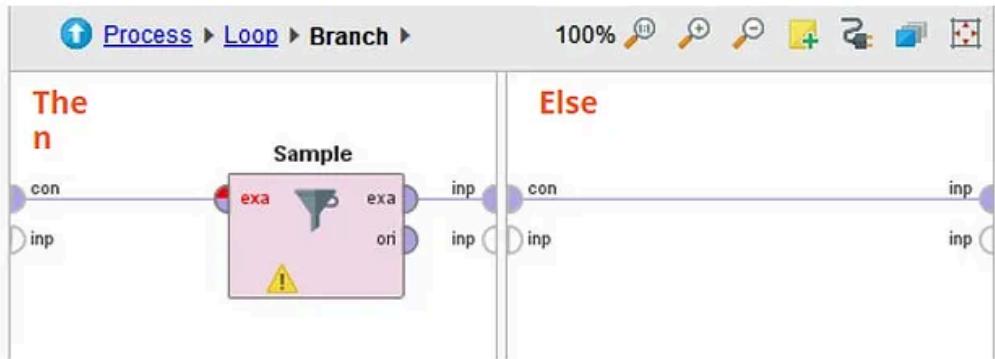
Summary

- Exception handling is used for debugging and functionality purposes. If the exception is handled correctly, the associated error can be identified and fixed, while unexpected or unwanted process behavior is avoided.

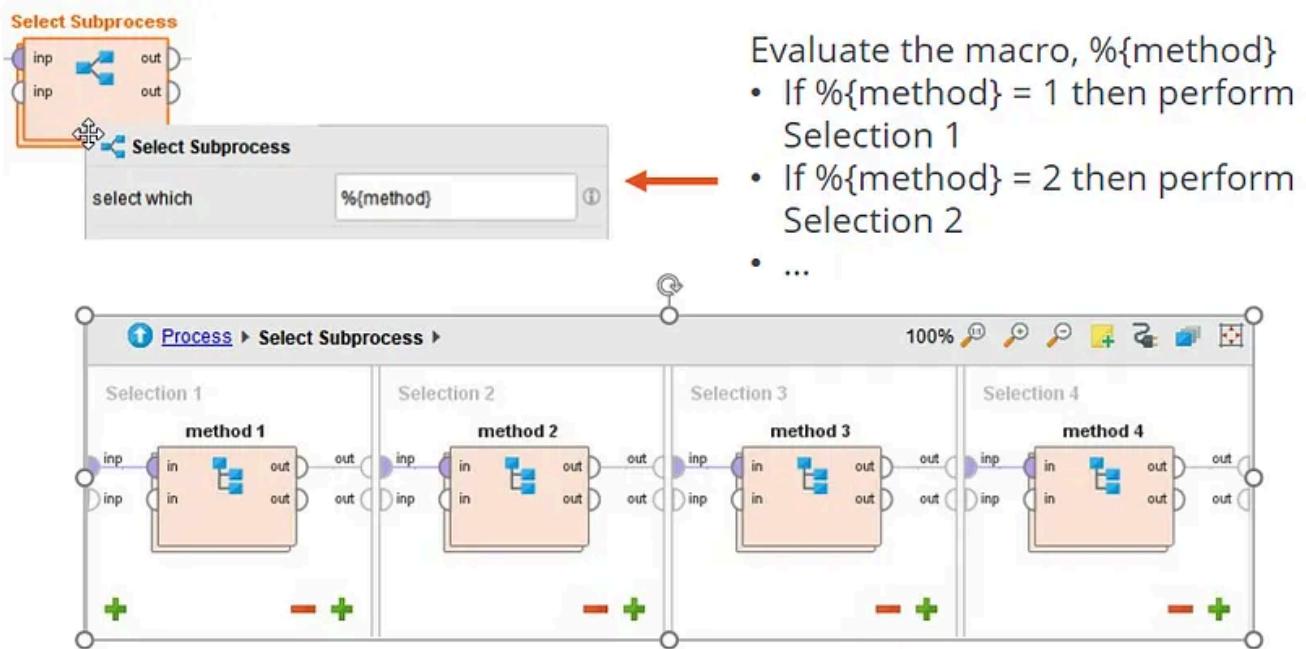
Branches

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 ✓

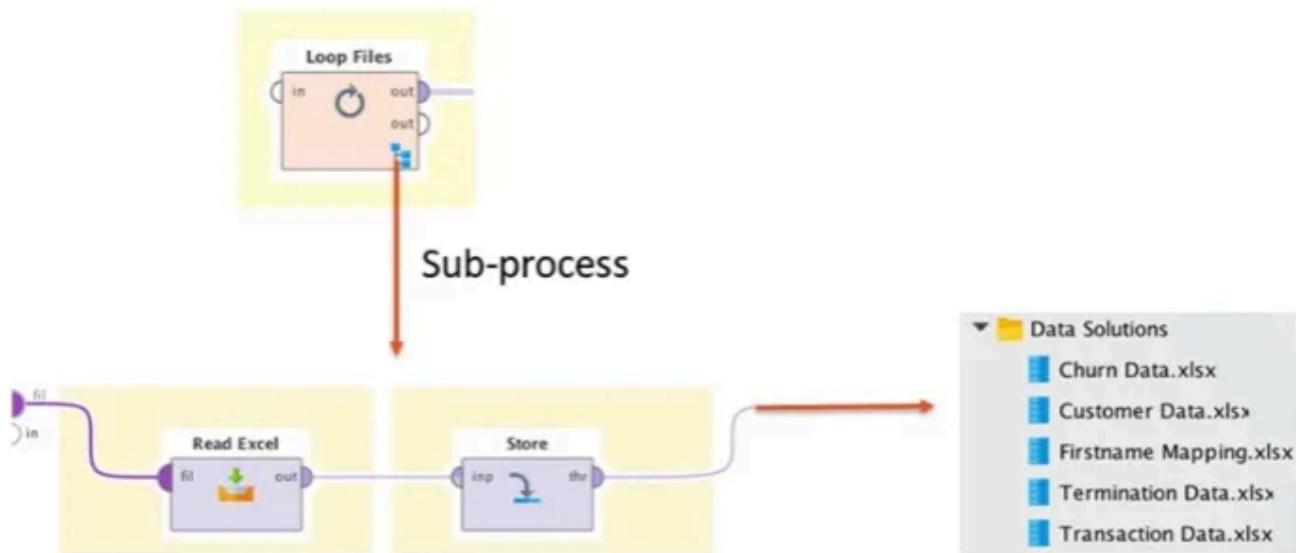
Loops and Macros Introduction

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.

Objectives

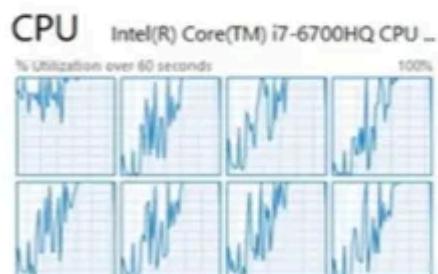
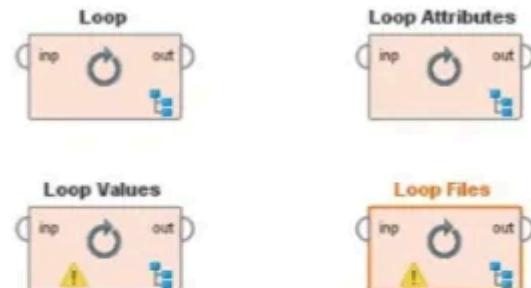
- Loading multiple data files: loops and macros
- File formats
- Impact of delimiters on parameter

Loading multiple data files



Parallel execution of loop operators

- The parallel execution framework introduced with RapidMiner 7.3 is now extended to the most popular loop operators:
 - Loop
 - Loop Attributes
 - Loop Values
 - Loop Files
- Especially data preparation tasks now benefit from vertical scaling, significantly improving their performance and your productivity



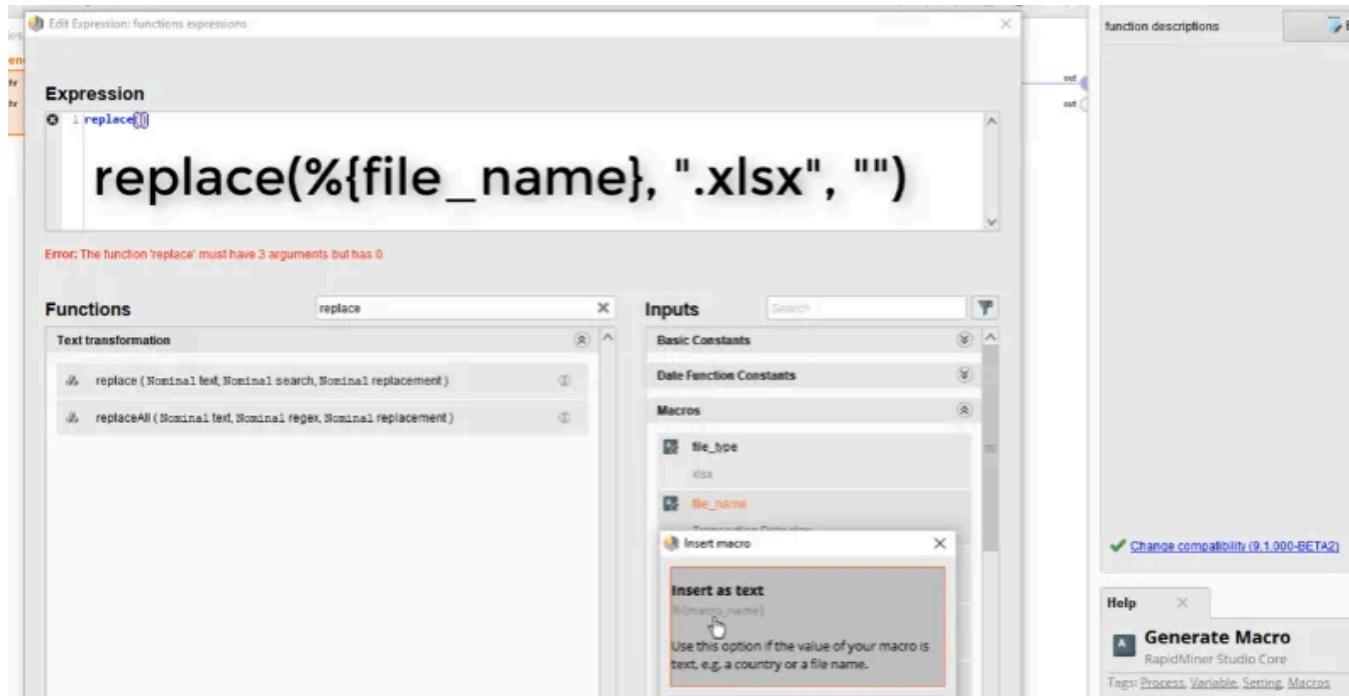
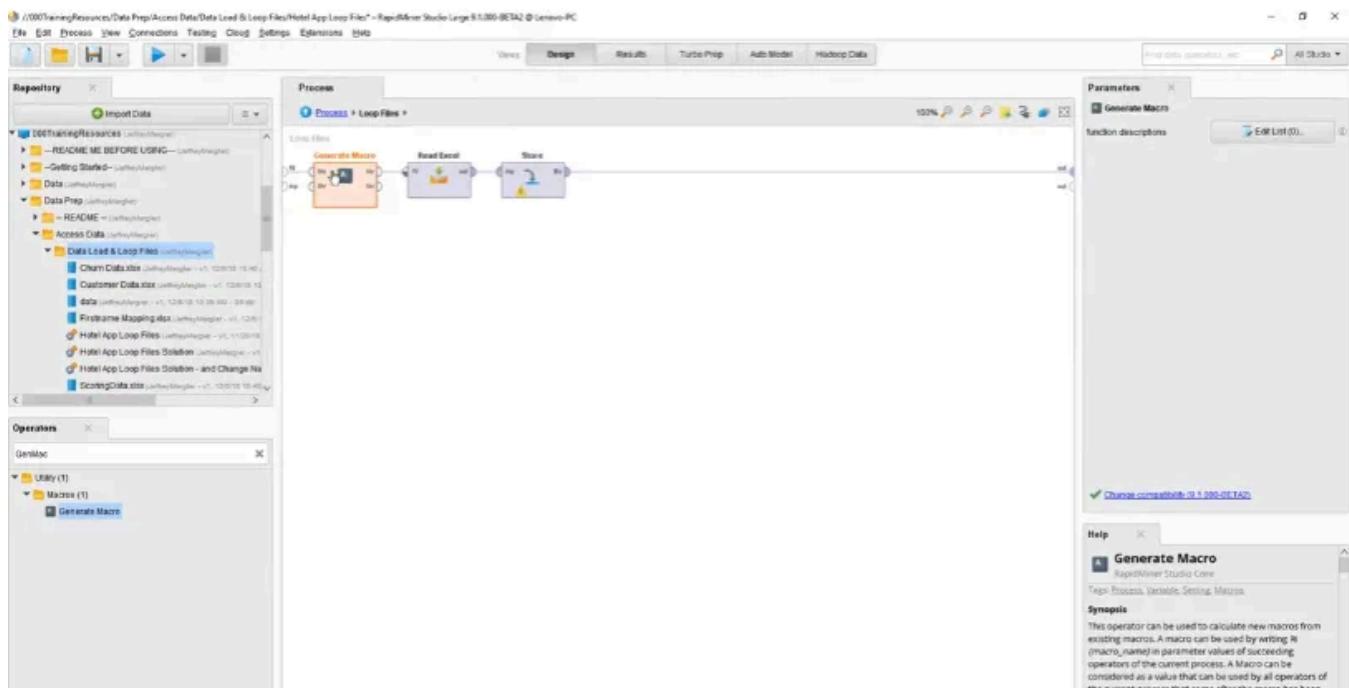
Raw Data Files

Get a quick introduction to the multiple raw data files.

	PostalCode	HashCode	CustomerId	Vorname	id	Birthday
1	87213	tKibdnh	3	Eva	1	1988-04-13 16:25:24
2	38548	RcW2Pb3w	4	Kunigunde	2	1974-06-21 16:25:24
3	44573	akWNQj4e	6	Notburga	3	1985-06-05 16:25:24
4	70936	gkPDLzY	7	Maximiliane	4	1992-07-26 16:25:24
5	49705	3nGPBX98	9	Dorothea	5	1943-08-04 16:25:24
6	42376	XkhfOlo	12	Maria	6	1960-07-29 16:25:24
7	52245	3ANQ9shn	16	Rosina	7	1964-06-28 16:25:24
8	56625	BDEPLkmG	17	Susanne	8	1989-04-14 16:25:24
9	66713	gOB0Z8Lw	20	Genoveva	9	1948-03-31 16:25:24
10	70052	G1c0kZqQ	21	Katharina	10	1931-11-16 16:25:24
11	44272	UIC8cDTW	22	Marlene	11	1978-06-29 16:25:24
12	71962	nSYVL7v	23	Annamarie	12	1996-09-22 16:25:24
13	45355	Hvt8FvKk	25	Magdalena	13	1965-02-18 16:25:24
14	59736	UVy8PnW	26	Waltraud	14	1980-09-20 16:25:24
15	44466	PmeOEulj	28	Maria	15	1960-09-29 16:25:24
16	62985	IOppPg8t	29	Barbara	16	1949-05-19 16:25:24
17	48487	FQONr9Qj	30	Katrin	17	1971-12-16 16:25:24
18	51008	ZBaMGyed	32	Barbara	18	1929-05-28 16:25:24
19	72496	vkg98Roy	35	Katrin	19	1943-01-23 16:25:24
20	38676	w1RMi0Ye	37	Susanne	20	1996-01-19 16:25:24
21	65683	67pT3SG7	38	Rosemarie	21	1964-09-07 16:25:24
22	57636	bxwqlFrx	39	Hannelore	22	1991-01-07 16:25:24
23	58302	dtW2nTUg	41	Annemarie	23	1948-08-29 16:25:24
24	80454	Q5Hs8l2N	42	Josephine	24	1991-10-24 16:25:24

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.

Process

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'."/>

Import Data

Retrieves (JeffreyMager)

Routes (JeffreyMager)

Generation (JeffreyMager)

Feature Generation (JeffreyMager)

Generate Aggregation (JeffreyMager)

Hotel App Generate Aggregation (JeffreyMager - v1, 12/13/11)

Hotel App Generate Aggregation Solution (JeffreyMager - v1, 12/13/11)

Generate Attributes (JeffreyMager)

Loop Attributes (JeffreyMager)

Hotel App Loop Attributes (JeffreyMager - v1, 12/13/11)

Hotel App Loop Attributes Solution (JeffreyMager - v1, 12/13/11)

Names & Roles (JeffreyMager)

Rename by Replacing (JeffreyMager)

Hotel App Rename by Replacing (JeffreyMager - v1, 12/13/11)

Hotel App Rename by Replacing Solution (JeffreyMager - v1, 12/13/11)

Control (2)

(2)

Loop Attributes

Loop Attribute Subsets

(1)

(1)

**find the loop attributes operator
bring that in to the process**

Process

Process

Retrieves (JeffreyMager)

Retrieve Transaction... Data Prep Feature Generation Loop Attributes

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):

You may find the following lessons helpful:
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

Result List (0) Regex Options

Inline Text Search

Text:

Result preview:

Item Shortcuts: CustomerId, ChurnIndicator, PostalCode, FirstName, Geschlecht, AverageTransactionValue, MostRecentTransactionDate, TotalTransactionValue, Age, TotalCount

Matched Items: countPaymentMethod_credit, countPaymentMethod_checkout, countPaymentMethod_cash

Parameters

Loop Attributes

attribute filter type: regular_expression regular_expression

regular expression: count+ count+ 4

use except expression:

invert selection:

include special attributes:

attribute name macro: loop_attribute loop_attribute

reuse results:

enable parallel execution:

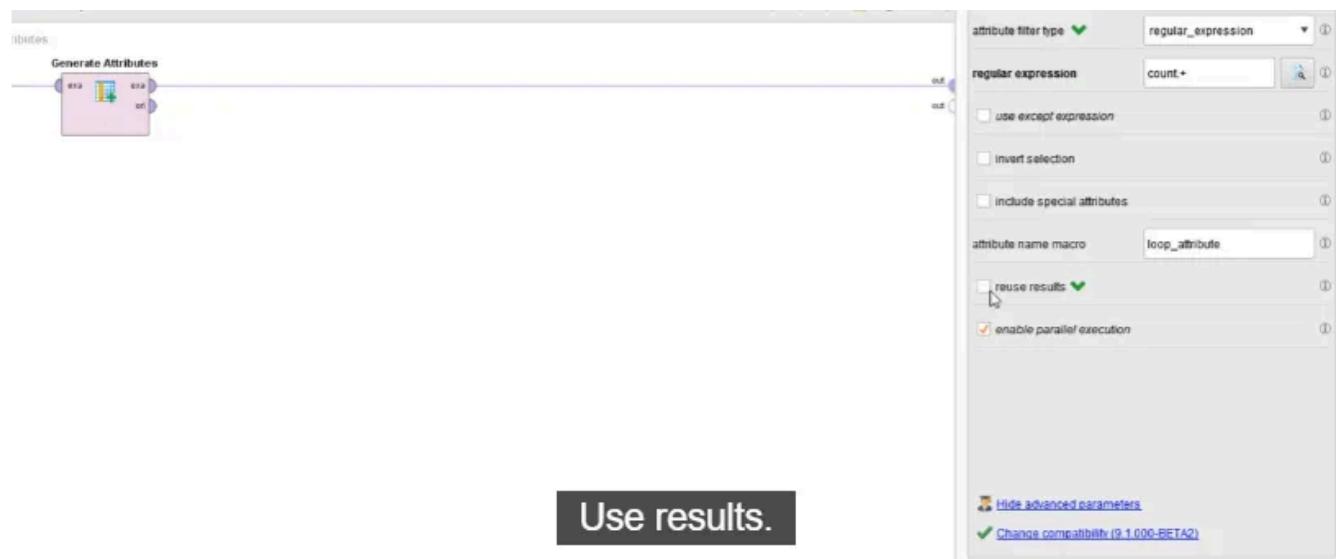
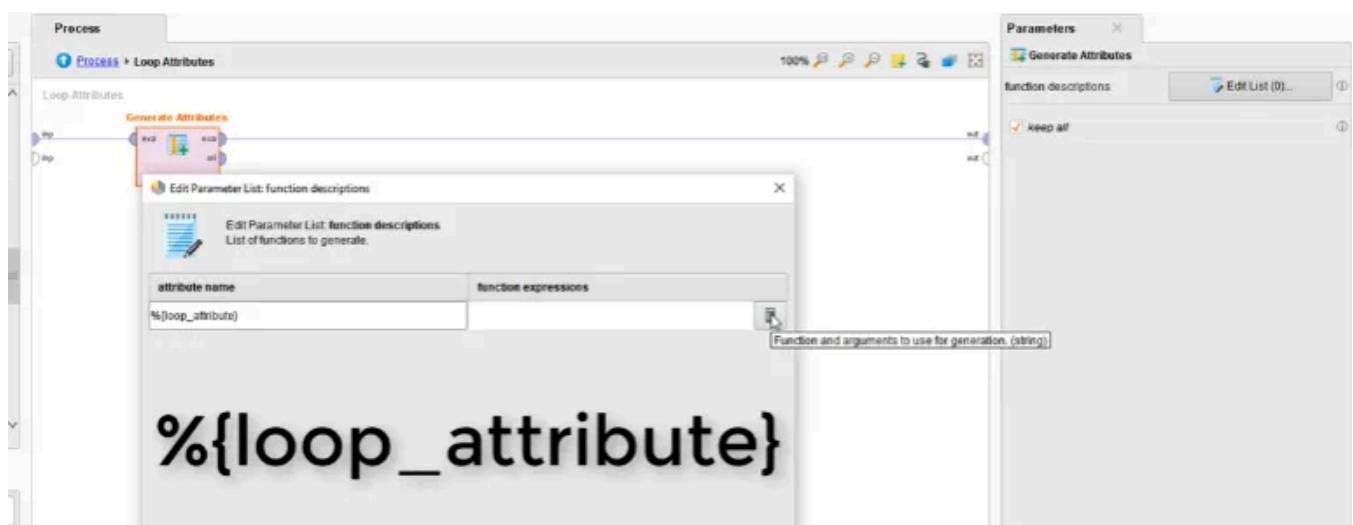
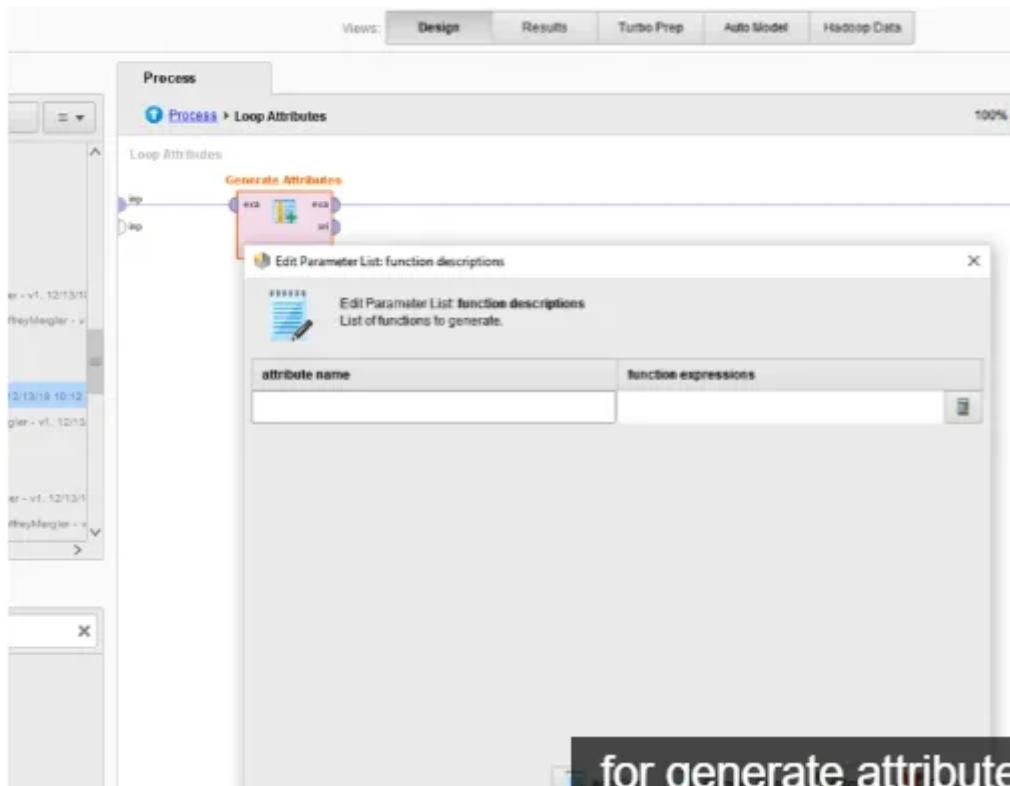
Hide advanced parameters:

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

Find data, operators, etc. All Studio

Filter (591 / 591 examples): all ▾

#Transa...	FirstName	Geschlecht	Age	PostalCode	MostRecent...	TotalCount	countPaymentMethod_cre...	countPaymentMethod_cheque	countPayme...
1070	Eva	w	30.657	8	215370322764	5	0.800	0.200	0
1282	Wolfram	m	44.324	3	205118500822	1	1	0	0
438	Maximiliane	w	26.382	7	212851065634	1	1	0	0
1803	Stephan	m	53.145	7	236440110774	1	1	0	0
1451	Bernhard	m	41.594	3	178478869308	8	1	0	0
388	Alfred	m	27.455	2	181677441725	2	1	0	0
504	Sebastian	m	32.304	4	233986713341	3	0	0.333	0.667
1247	Christian	m	26.958	5	239778253800	10	0.100	0.100	0.800
70	Rosina	w	54.460	5	246672768488	5	1	0	0
455	Adolf	m	50.021	5	184079910601	5	1	0	0
1905	Eustachius	m	50.656	3	247705410155	8	0.875	0.125	0
1099	Genevieve	w	70.704	6	24496540967	2	0	0	1
1989	Marlene	w	40.458	4	2576582	1	0	0	0
1948	Annemarie	w	22.223	7	2313608	1	0	0	0
1613	Magdalena	w	53.816	4	261486451061	2	0	0	0
1775	Kaspar	m	45.501	2	217204345802	2	0	0	0
1927	Barbara	w	69.570	6	231325882437	7	0.857	0.143	0

Also see that, in
fact, if we sum

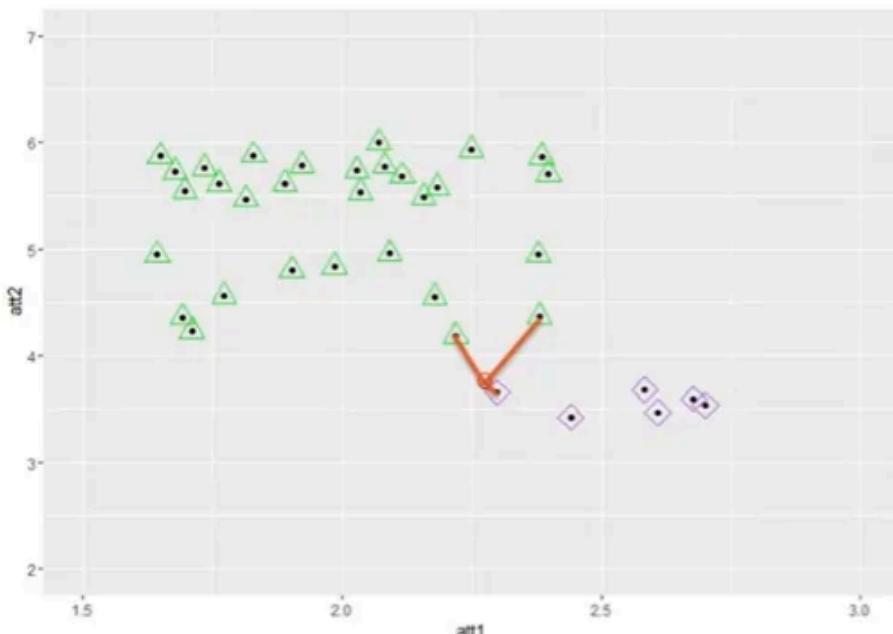
Intro to Sampling

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.

Objectives

- Revisit k-NN
- Down-sampling & Up-sampling with k-NN
- Generate Weights by Stratification

Class sample size



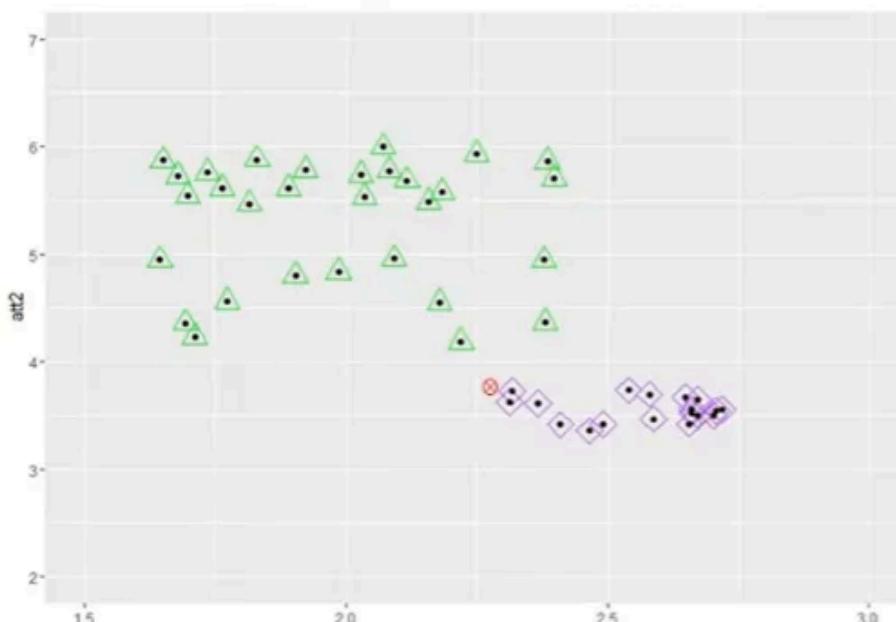
BUT

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



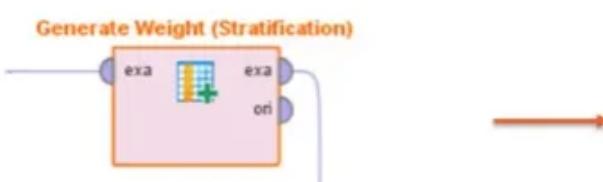
Sample numbers should be balanced!

Up-sampling (adding points + jitter)



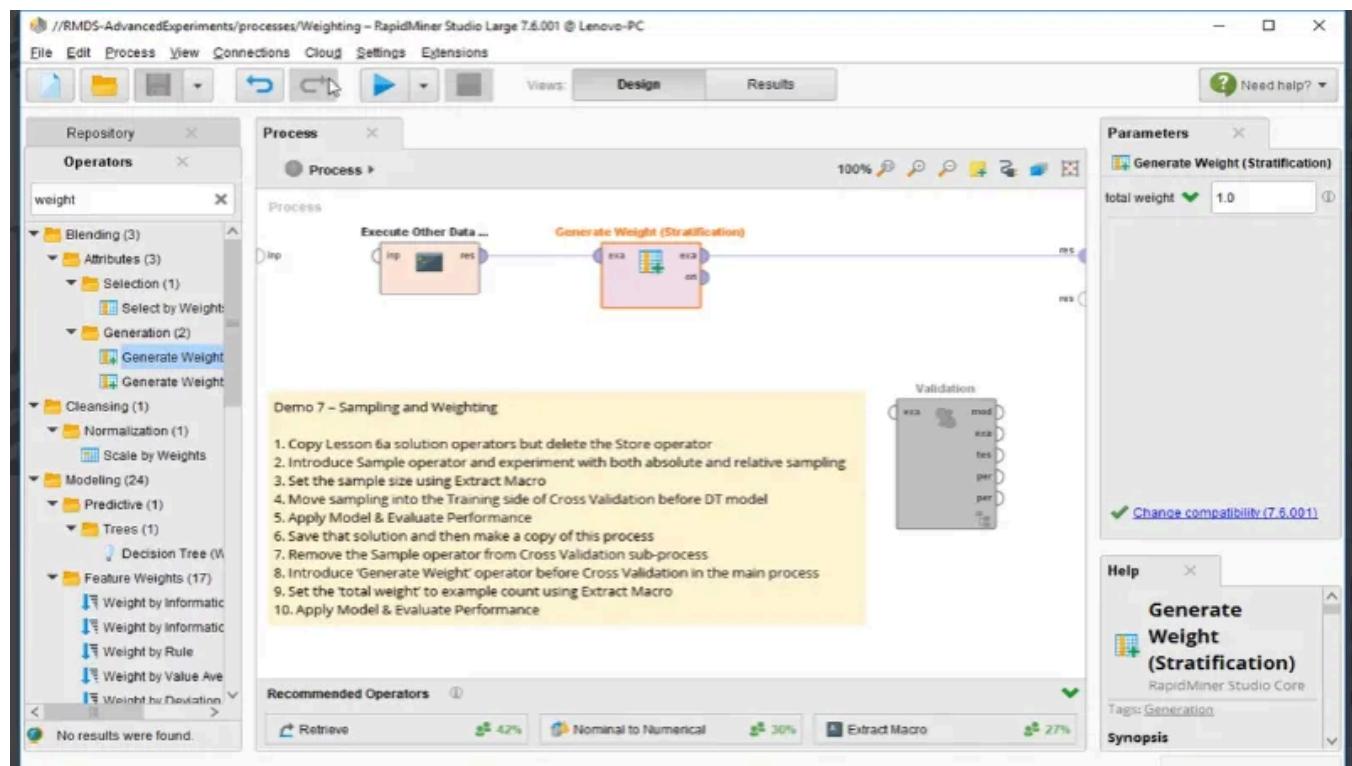
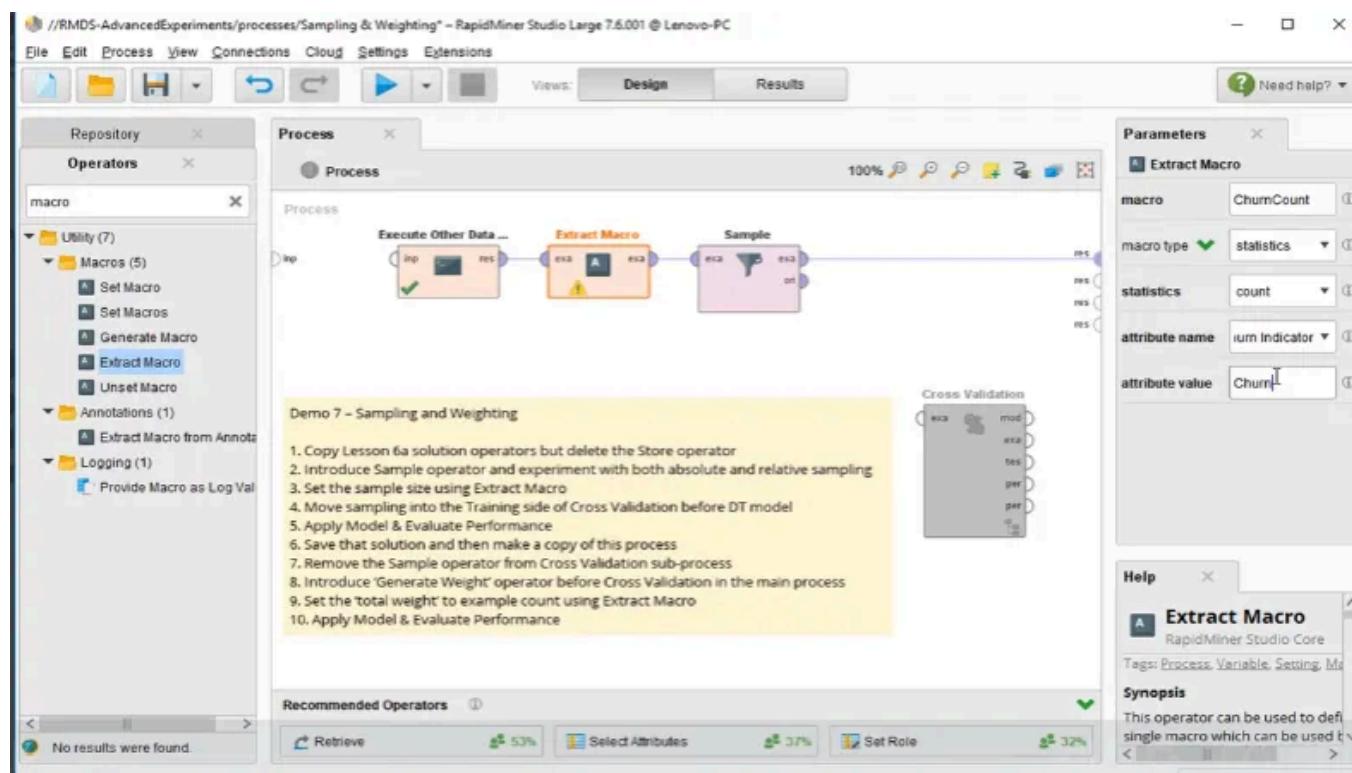
- There are more purple points
- As a whole they gained importance
- Classification near the green cluster is still accurate

Generate Weight (Stratification)



Sampling and Weighting

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



File Edit Process View Connections Cloud Settings Extensions

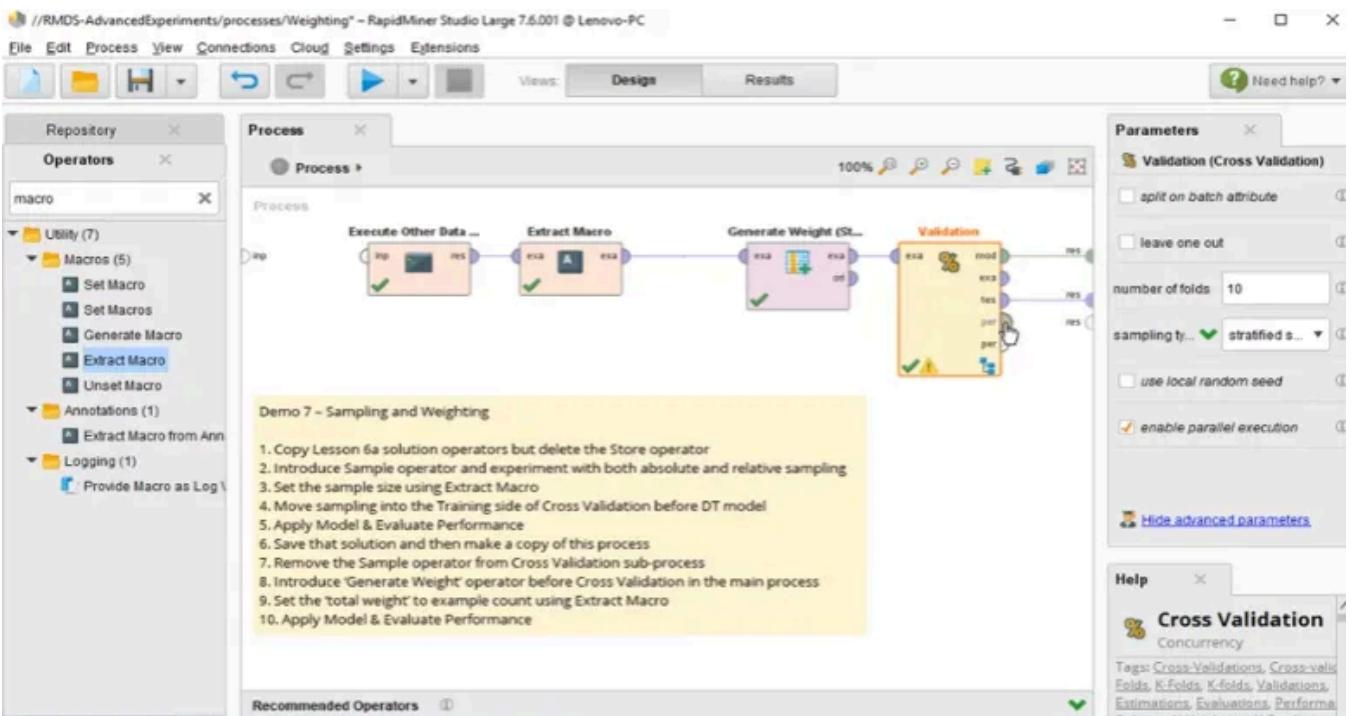
Views: Design Results

Result History ExampleSet (Generate Weight (Stratification))

Repository

The screenshot shows the RapidMiner Studio interface. On the left, there's a vertical toolbar with icons for Data, Statistics, Charts, Advanced Charts, and Annotations. The main area displays a table titled 'ExampleSet (591 examples, 3 special attributes, 10 regular attributes)'. The table has columns: Row No., CustomerId, Churn Indica..., weight, AverageTra..., TotalTransa..., Gender, Age, and Posta. The repository on the right shows a tree structure under 'RMDS-Advanced (Training)'.

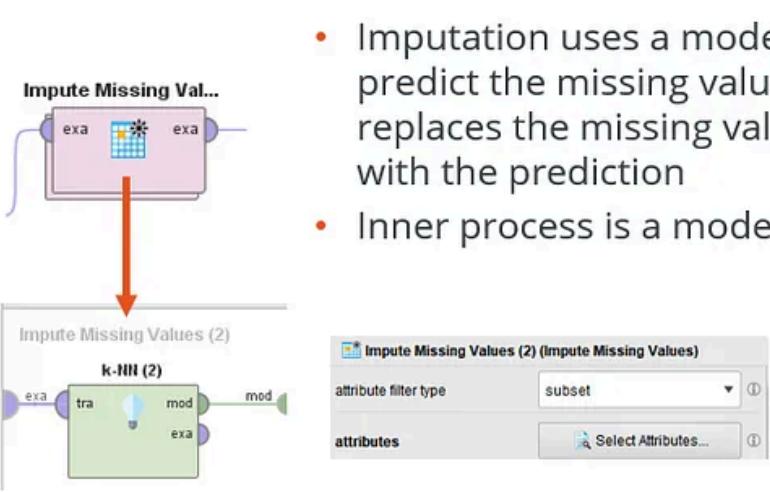
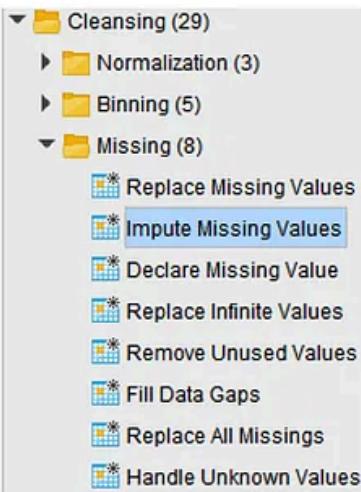
Row No.	CustomerId	Churn Indica...	weight	AverageTra...	TotalTransa...	Gender	Age	Posta
1	3	Loyal	0.001	23.814	119.070	w	29.575	8
2	5	Loyal	0.001	432.282	432.282	m	43.231	3
3	7	Loyal	0.001	111.438	111.438	w	25.290	7
4	8	Loyal	0.001	512.803	512.803	m	52.053	7
5	10	Loyal	0.001	14.056	112.451	m	40.502	3
6	11	Loyal	0.001	365.694	731.388	m	26.363	2
7	13	Loyal	0.001	121.168	363.504	m	31.212	4
8	15	Loyal	0.001	21.625	216.247	m	25.876	5
9	16	Churn	0.005	0.500	2.500	w	53.367	5
10	18	Loyal	0.001	87.091	435.455	m	48.929	5
11	19	Loyal	0.001	16.113	128.905	m	49.564	3
12	20	Churn	0.005	244.050	488.099	w	69.611	6
13	22	Loyal	0.001	38.998	233.989	w	39.366	4
14	23	Loyal	0.001	432.948	432.948	w	21.131	7
15	25	Churn	0.005	54.807	109.613	w	52.724	4
16	27	Churn	0.005	75.682	529.775	m	44.409	2
17	29	Churn	0.005	44.418	310.927	w	68.478	6



Replace Missing Values

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

Imputation



- Imputation uses a model to predict the missing value, and replaces the missing value with the prediction
- Inner process is a model



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

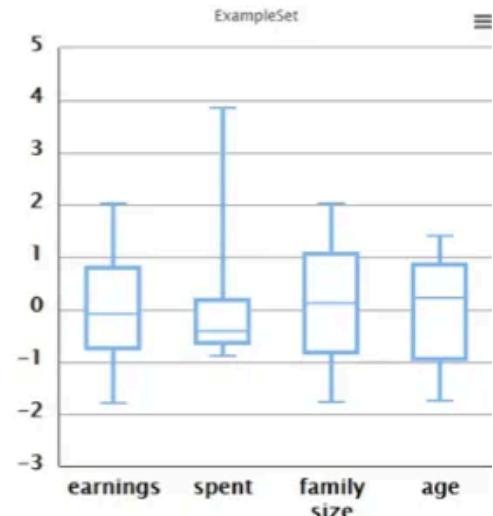
Learn about normalizing your data.

Normalization (Before)

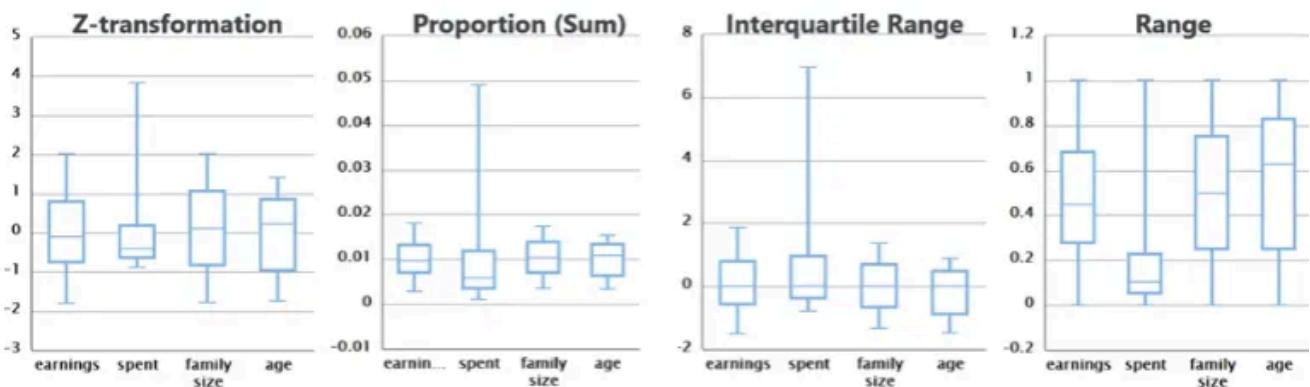
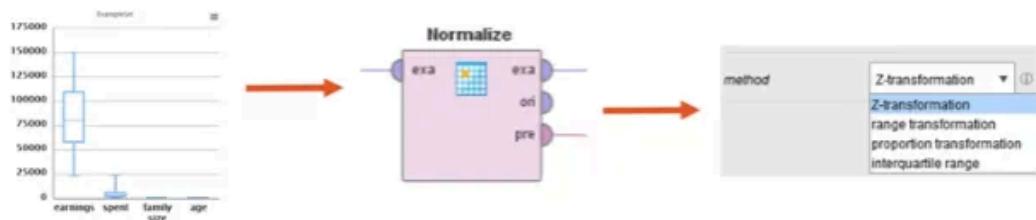


Normalization (After)

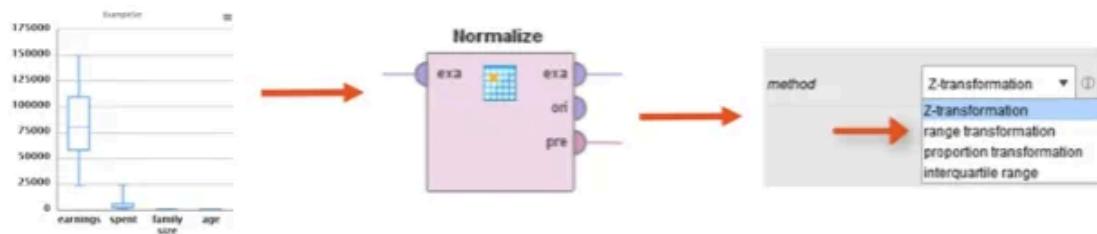
spent Number	earnings Number	family size Number	age Number
0.016	0.583	0.114	0.986
-0.397	-1.507	0.114	-0.645
-0.638	1.071	0.114	1.393
0.411	1.451	0.114	0.695
-0.691	0.387	-0.833	1.219
...



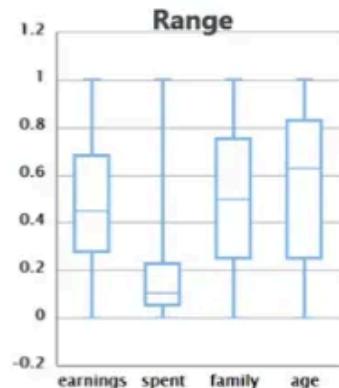
Normalization Methods



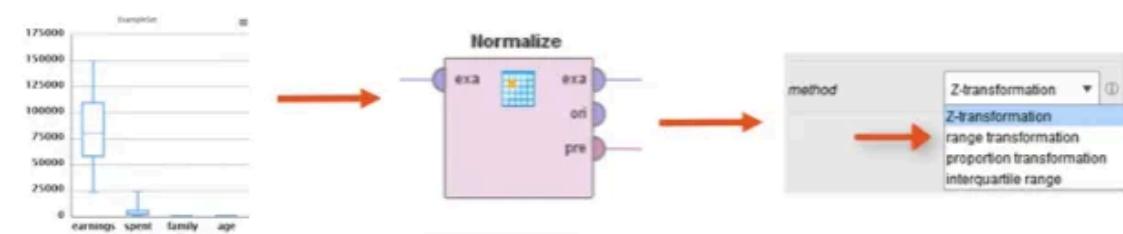
Normalization Methods



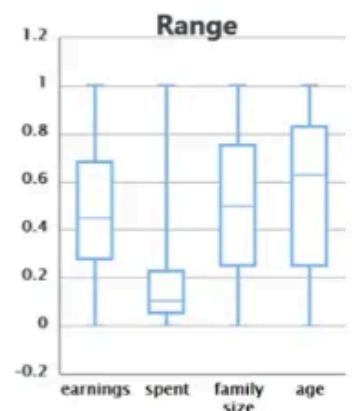
Set the max value to 1
Set the min value to 0
Other values are place in proportion to min and max



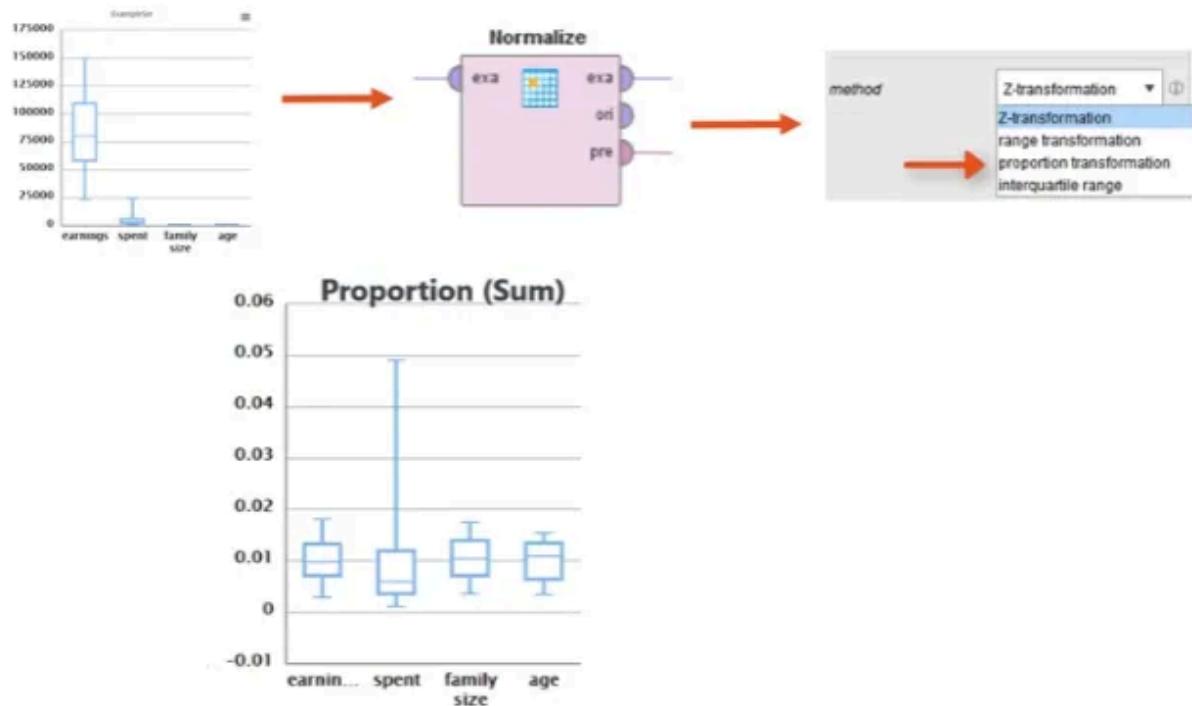
Normalization Methods



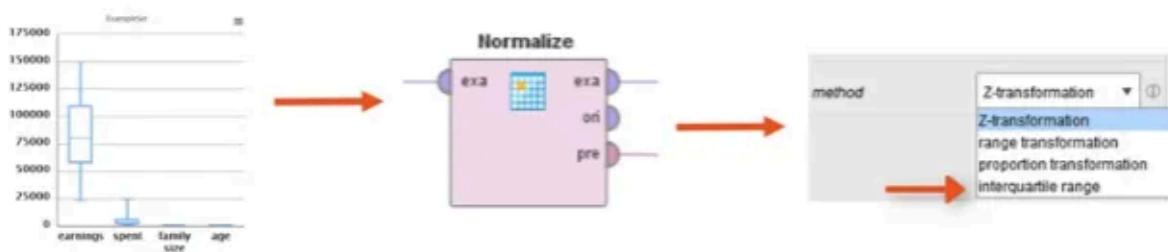
33006 → 0
96003 → ~0
118760 → ~0
?,000,000,000 → 1



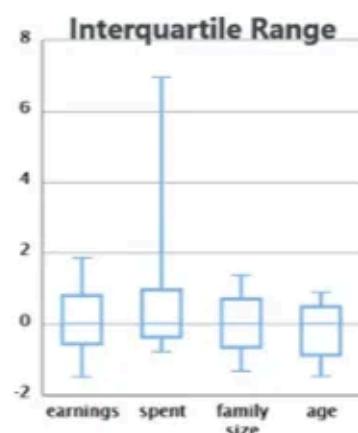
Normalization Methods



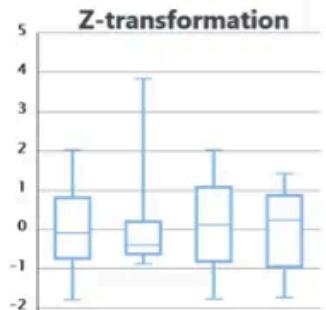
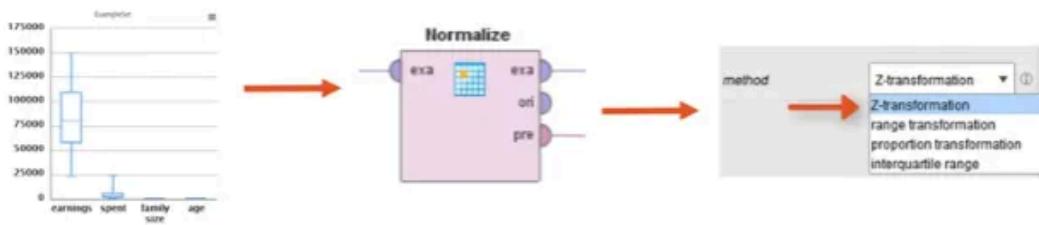
Normalization Methods



First quartile gets
-1 and Third
quartile gets +1

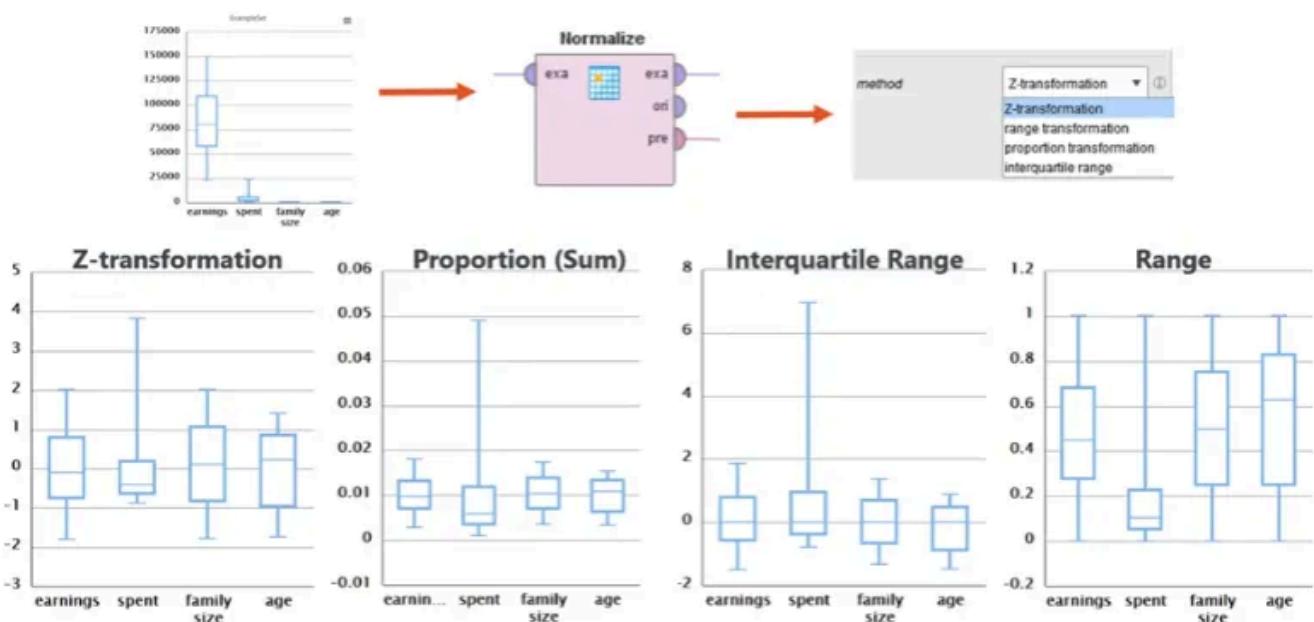


Normalization Methods



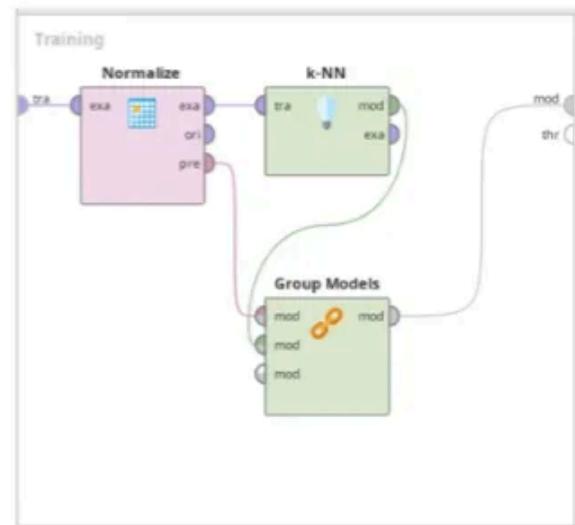
Subtract mean and divide by standard deviation

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.



Repository: Shows a tree view of available operators and solutions. Under 'Normalize', several 'APS Failure Normalize' and 'Hotel App Normalizing' solutions are listed. Other categories include 'Missing', 'Outliers', 'PCA', 'Text Processing', 'United Data Prep Challenge', 'Model', 'Optimize', 'Compare Models', and 'Feature Selection'.

Process: Displays a process flow titled 'PROCESS'. It starts with an 'ETL' subprocess (with 'in', 'out', and 'out' nodes) which feeds into a 'Validation' subprocess (with 'tra' and 'mod' nodes). The 'mod' output from the validation is highlighted in blue.

Operators: Shows a list of operators under the 'SpVal' tab. The 'Validation' category is expanded, showing 'Split Validation' (selected), 'Extensions', 'Radoop', and 'Validation' (including 'Split Validation (Radoop)').

Notes: You may find the lesson [here](#).

1. Review the ETL subprocess
2. Perform Cross Validation with k-NN
3. Introduce Normalize before k-NN and Group Models before applying
4. Review results
5. Consider changing the k-NN k value

Repository

- Missing
- Normalize
 - APS Failure Normalize (v1 ~ 4 kB)
 - APS Failure Normalize Solution (v1 ~ 5 kB)
 - Hotel App Normalizing (v1)
 - Hotel App Normalizing Solution (v1)
 - Telco Normalization (v1 ~ 6 kB)
 - Telco Normalization Solution (v1 ~ 11 kB)
- Outliers
- PCA
- Text Processing
- Unified Data Prep Challenge
- Model
 - Optimize
 - Compare Models
 - Feature Selection

Process

Process > Validation

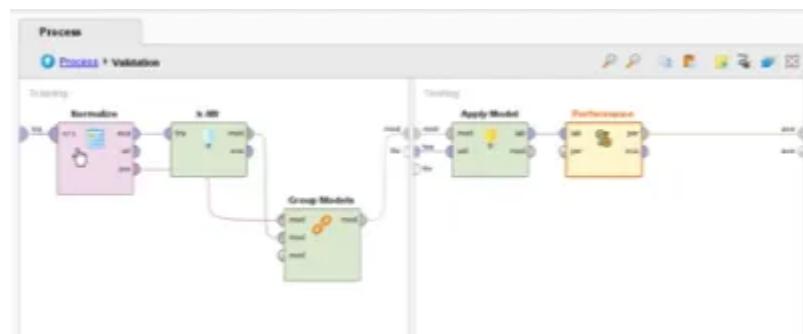
Training

Testing

Operators

Normalize

- Cleansing (2)
- Normalization (2)
 - Normalize
 - De-Normalize



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。

Views: Design Results Turbo Prep Auto Model Deployments Hadoop Data

Result History GroupedModel (Group Models) PerformanceVector (Performance) ExampleSet (Select Attributes)

Z-Transformation

Normalize 2 attributes to mean 0 and variance 1.
Using
LastTransaction --> mean: 26.76730190571715, variance: 230.98595787362083
Age --> mean: 45.66098294884654, variance: 354.18816489629523

Process

Process

You may find the lesson [here](#).

1. Review the ETL subprocess
2. Perform Cross Validation with k-NN
3. Introduce Normalize before k-NN and Group Models before applying
4. Review results
5. Consider changing the k-NN k-value

Parameters

- Normalize (2) (Normalize)
 - create view
 - attribute filter type all
 - invert selection
 - include special attributes
- method Z-transformation

Result History GroupedModel (Group Models) PerformanceVector (Performance) ExampleSet (Select Attributes)

Performance

Criterion	accuracy	precision	recall	AUC (optimistic)	AUC	AUC (pessimistic)
accuracy	74.92%					
precision						
recall						
AUC (optimistic)						
AUC						
AUC (pessimistic)						

Description

Annotations

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.

Rapidminer

Machine Learning

Algorithms

Data Science

Feature Engineering



Follow

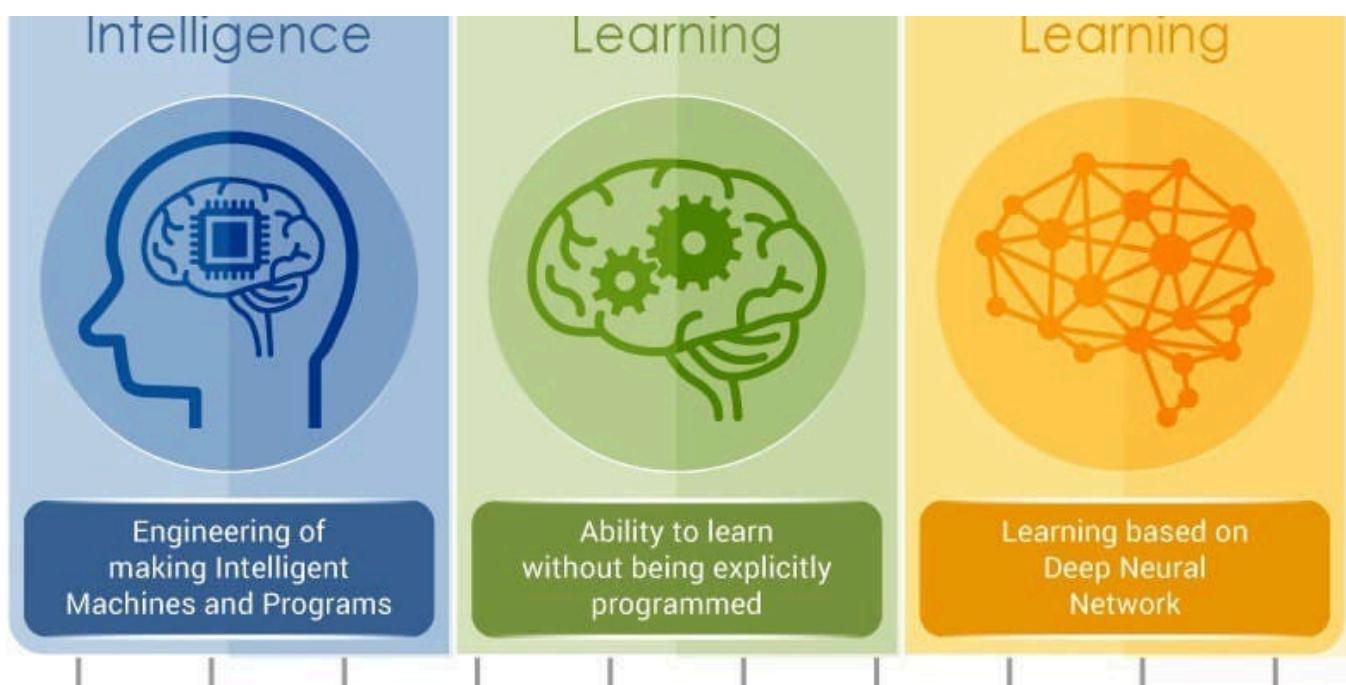


Written by KevinLuo

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

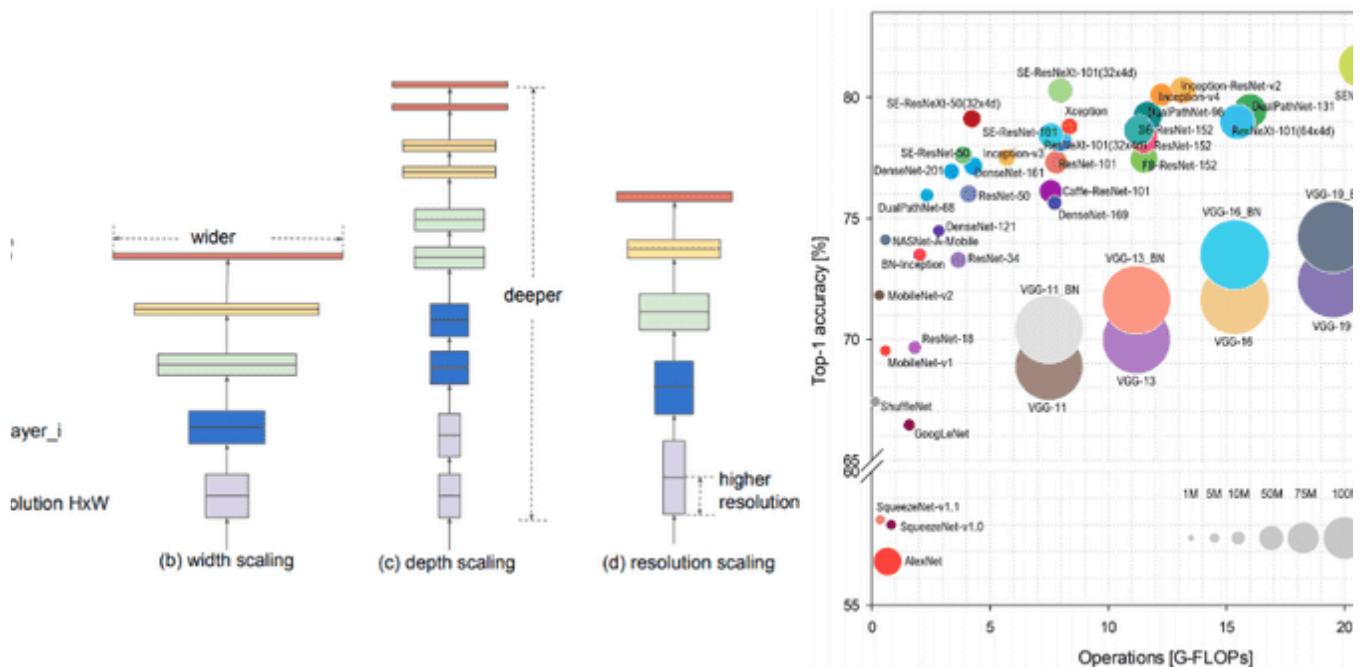
Top 10 您應該要學會的深度學習演算法 (Fundamental Review Series)

首先，我們來簡單回想一下...

10 min read · Dec 1, 2021

12





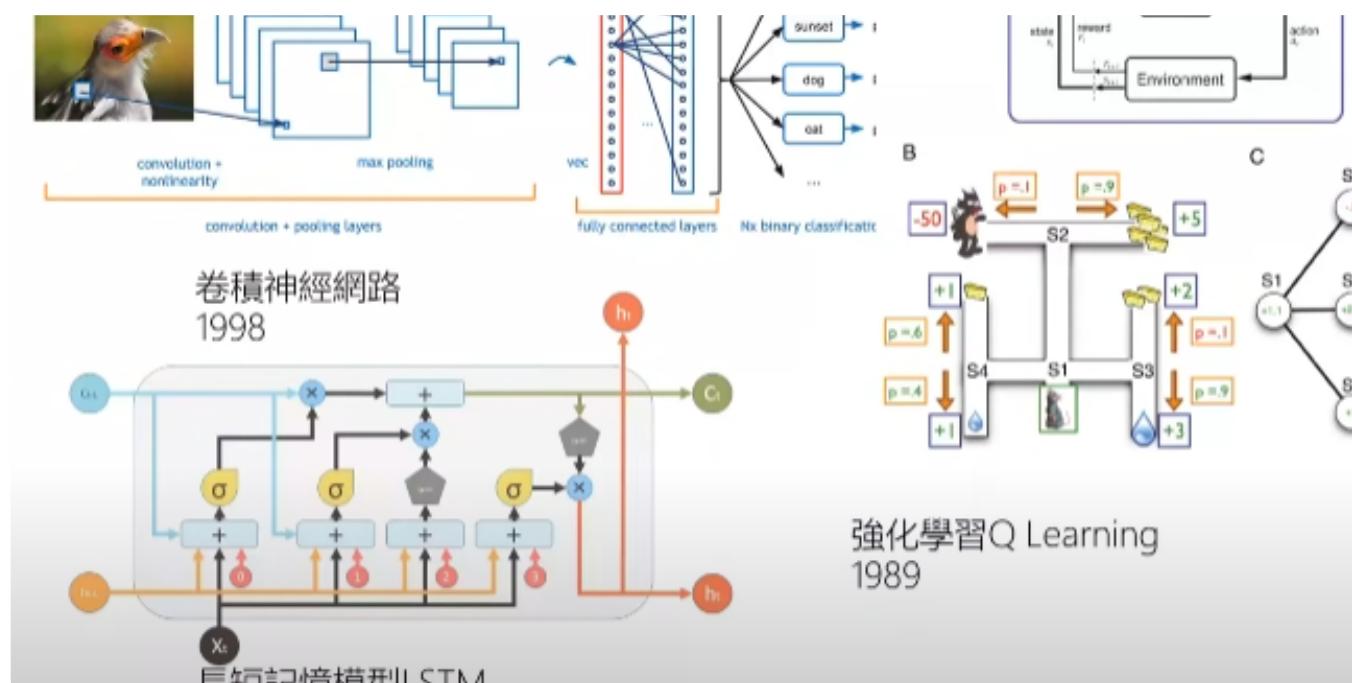
KevinLuo

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



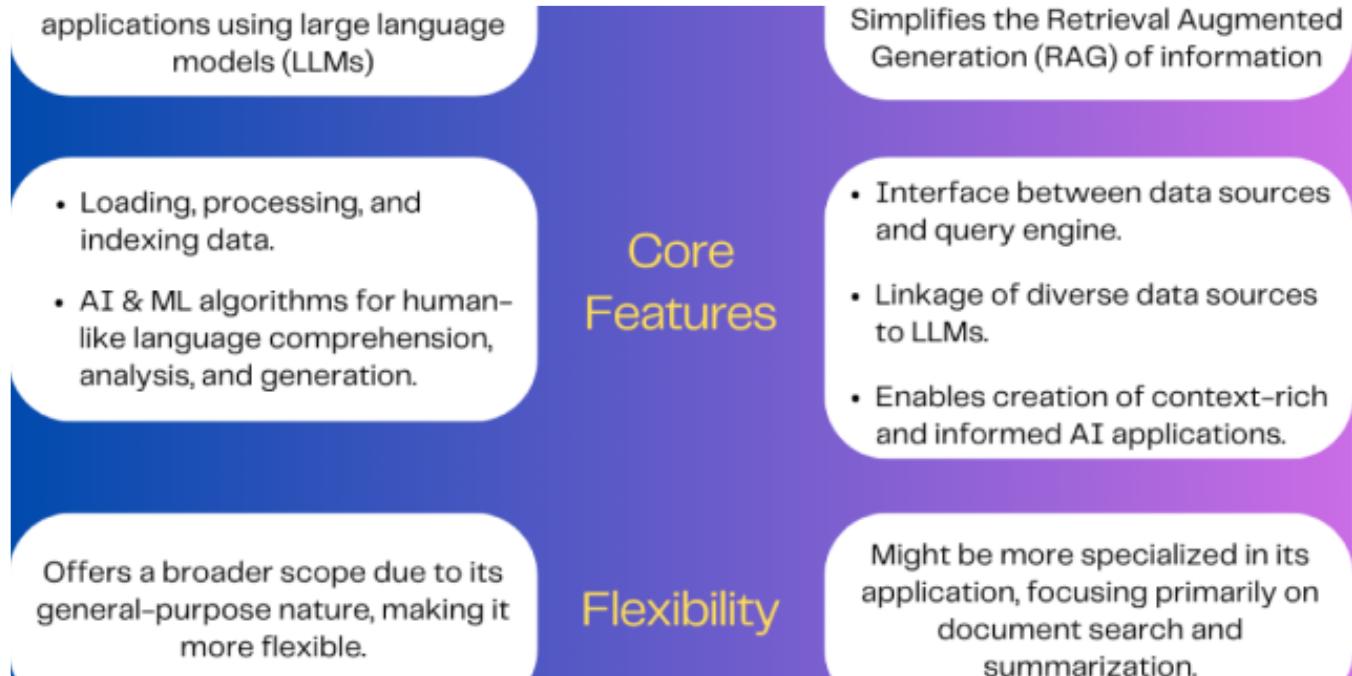
KevinLuo

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

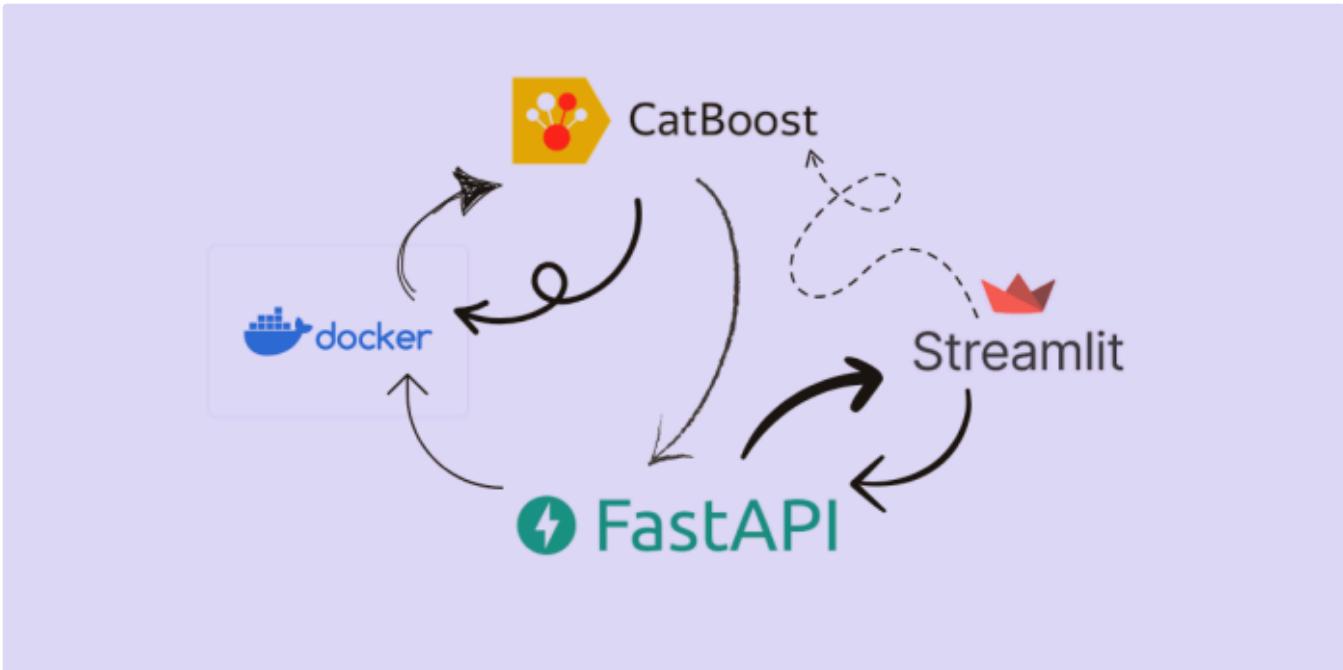
3 min read · Mar 10, 2024

53



See all from KevinLuo

Recommended from Medium



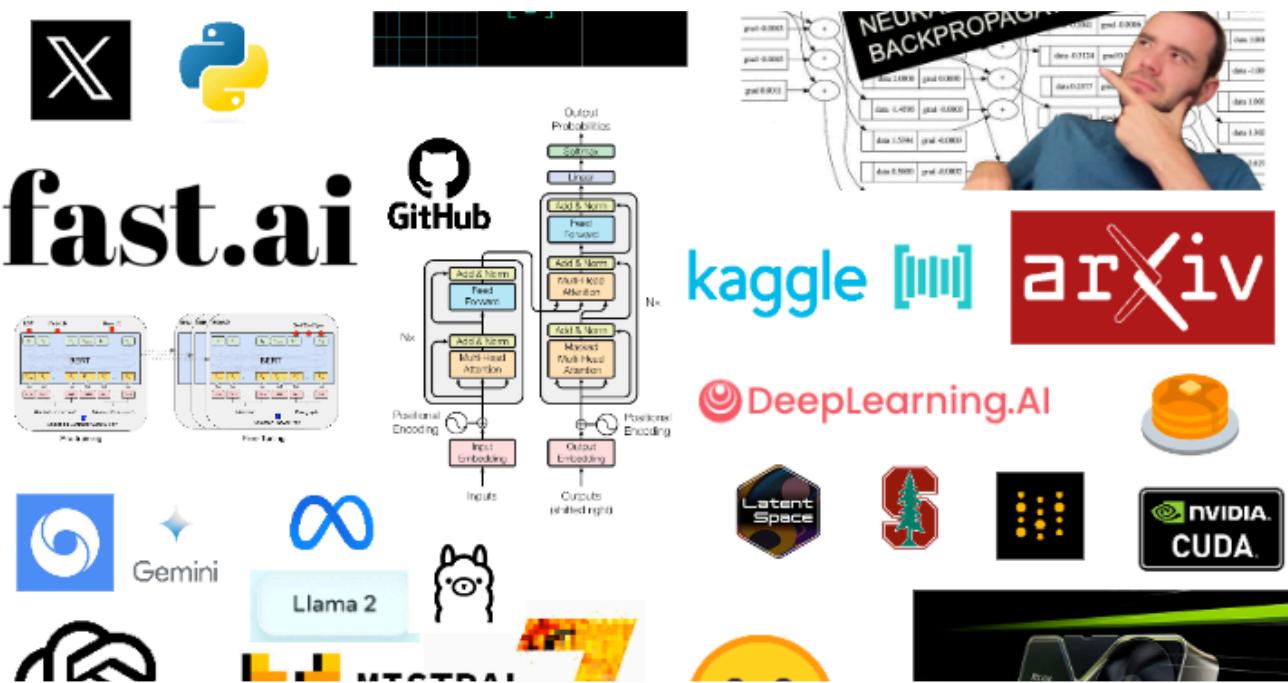
 Ramazan Olmez

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

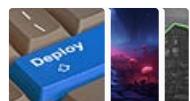
11 min read · Mar 11, 2024

11.1K

113



Lists



Predictive Modeling w/ Python

20 stories · 1141 saves



Practical Guides to Machine Learning

10 stories · 1373 saves



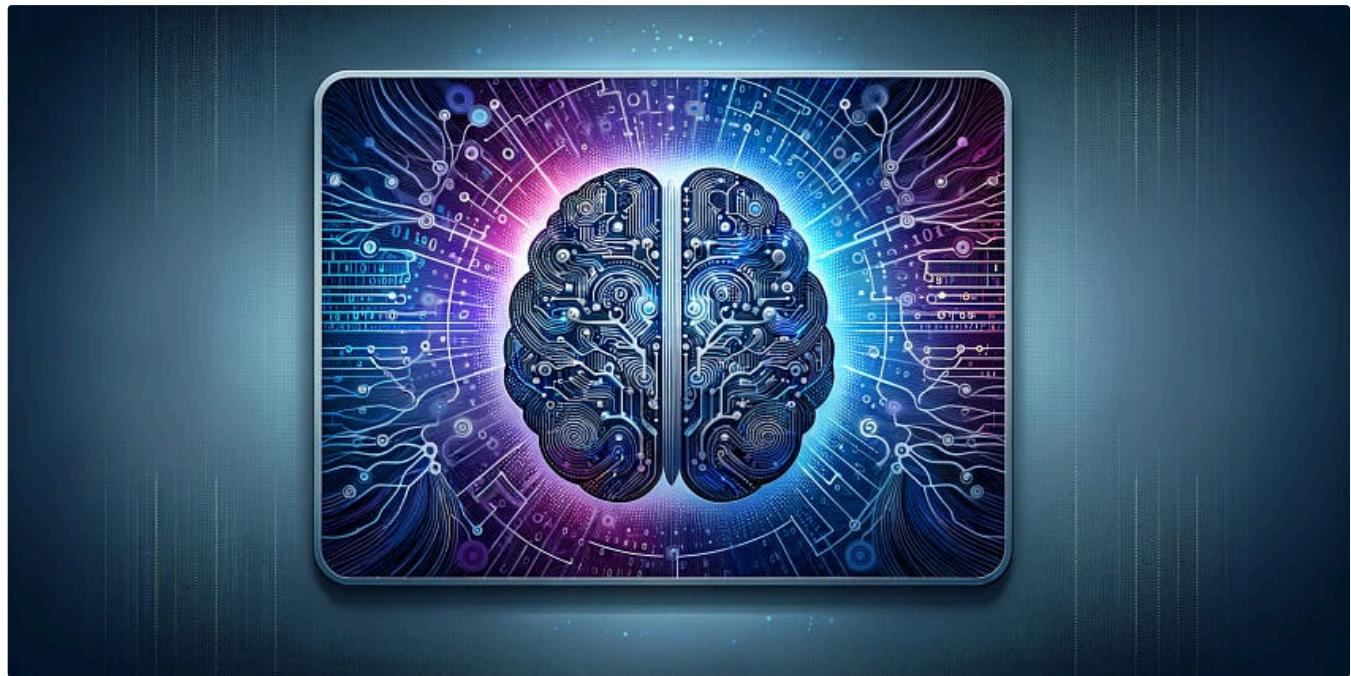
Natural Language Processing

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



```
*__, a, b, *__ = [1, 2, 3, 4, 5, 6]
print(__, __)
```

What does this print?

- A) Syntax error
- B) [1] [4, 5, 6]
- C) [1, 2] [5, 6]
- D) [1, 2, 3] [6]
- E) <generator object <genexpr> at 0x1003847c0>



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