

# Introduction to Rapidminer

**Amirhassan Monajemi 2023**



# ABOUT THE LECTURER

Dr. Amirhassan Monajemi (aka Monadjemi) is a Senior Lecturer in AI and Data Science at the School of Computing, the National University of Singapore. Before joining the NUS in 2019, he was with the Faculty of Computer Engineering, University of Isfahan, Iran, where he was serving as a professor of AI and Machine Learning.

Dr. Monajemi has taught diverse computer science courses for years, registered a few patents in the fields of AI, Machine Vision, and Signal Processing applications, published more than a hundred research papers in peer-reviewed, indexed journals, and supervised several related industrial projects in various scales.



# AGENDA

Day	Time	Modules	Module Topics
1	9-10:45	Introduction and Environment	<ol style="list-style-type: none"> <li>1. What is Rapidminer</li> <li>2. How to install it</li> <li>3. Applications</li> <li>4. Environment</li> </ol>
	11-13	Data, Preprocessing, and Visualization	<ol style="list-style-type: none"> <li>1. Data resources in RM</li> <li>2. Data Preprocessing</li> <li>3. Basic data visualization in RM</li> </ol>
<b>LUNCH</b>			
1	14-15:45	Clustering	<ol style="list-style-type: none"> <li>1. How to cluster data in RM</li> <li>2. Clustering example and practice</li> </ol>
	15:45-16:30	Classification and Function Estimation	<ol style="list-style-type: none"> <li>1. How to Estimate a function in RM</li> <li>2. How to classify data in RM</li> <li>3. Classification example and practice</li> </ol>
	16:30-17:30	Mini Project	

# Part 1

## Introduction to Machine Learning

# WHAT IS MACHINE LEARNING?

Machine learning (ML) is the study of computer algorithms that improve automatically through experience.

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks.

Supervised  
Learning

Reinforcement  
Learning

Unsupervised  
Learning

# WHAT IS MACHINE LEARNING?

A considerable portion of AI is Machine Learning.

We may design an intelligent machine to do something without the learning ability.

However, if that machine can ‘Learn’, it would be an instance of a learning machine.

Obviously, the second one has got some privileges:

- It can improve its performance.

- It can learn by examples.

- It can explore and get in rather new duties.

- It would be more like us!

# WHAT IS MACHINE LEARNING?

Examples:

Artificial Neural Networks  
and Deep Networks

Regression Alg.

Support Vector Machines

Learning Automatons

advanced businesses  
**intelligence**  
applied ai deep  
future octave  
**machine**  
data proof **artificial**  
programming analytics



# Practice 1

- What are the main differences between data mining and data analytics?
  - Look them up and write them down in a paragraph or two.

# Applications of Machine Learning

# MACHINE LEARNING APPLICATIONS

Virtual Personal Assistants

Predictions

Video Surveillance

Social Media Services

Email Spam and Malware Filtering

Online Customer Support

Search Engine Result Refining

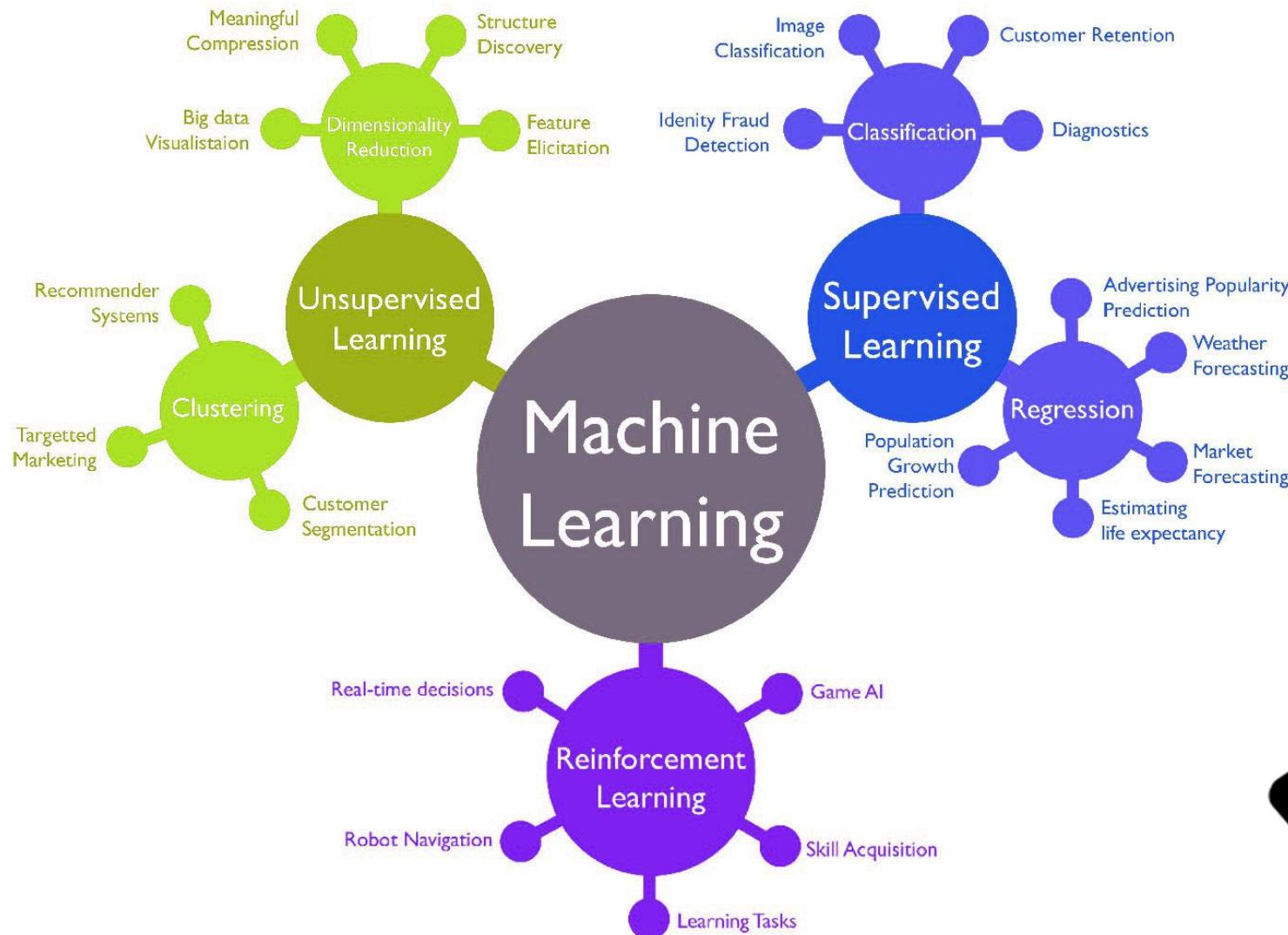
Product Recommendations

Natural Language Processing, Machine Translation, Abstraction

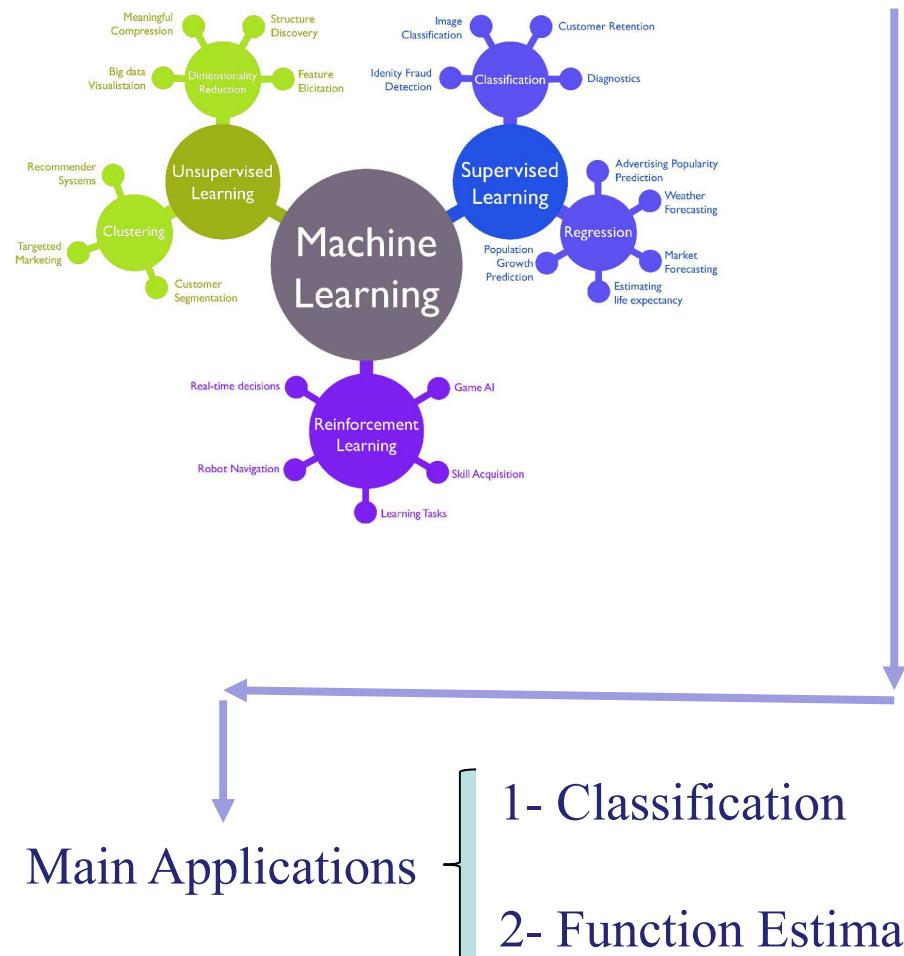
Driverless Vehicles



# MACHINE LEARNING APPROACHES



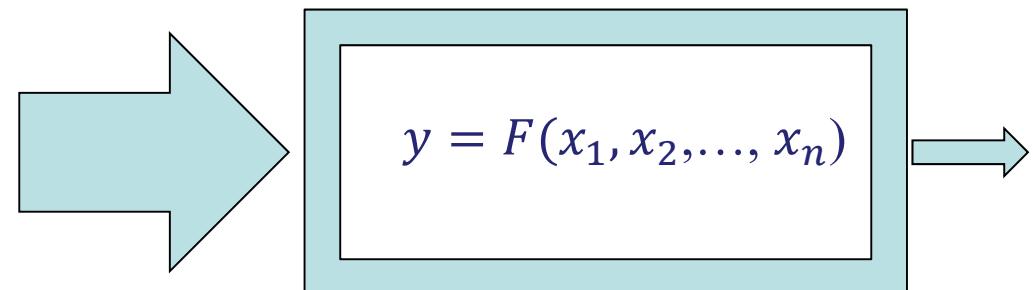
# Supervised Learning



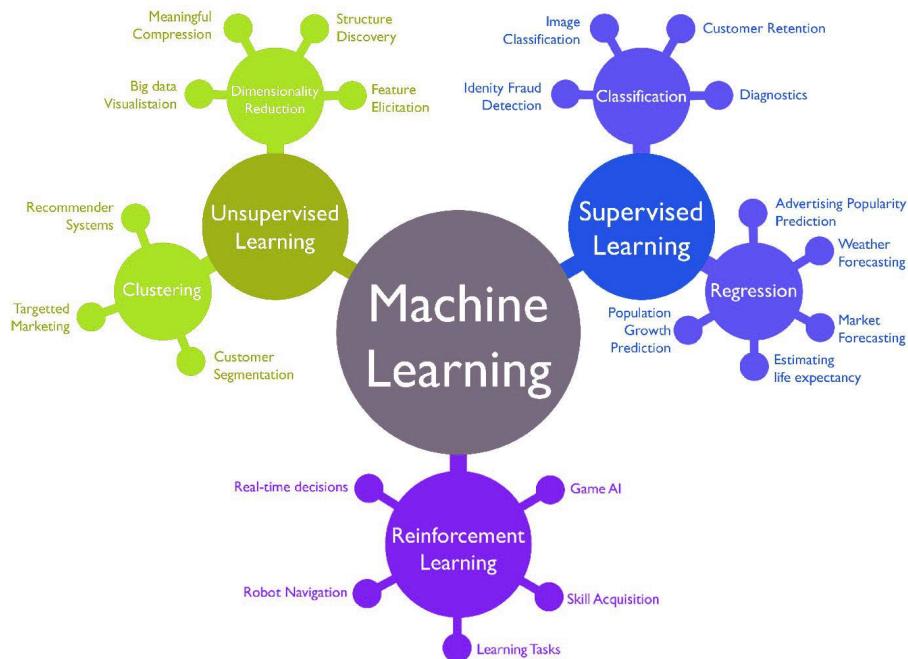
- Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs.
- It infers a function from labeled training data consisting of a set of training examples.
- A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples.

# Supervised Learning

- **To Classify ...**
  - Your suppliers into more-reliable and less-reliable classes
  - Celebrities into classic and modern era
  - Loan requests into approvable and risky
- **To Estimate ...**
  - Price of a used car based on its features (attributes, characteristics)

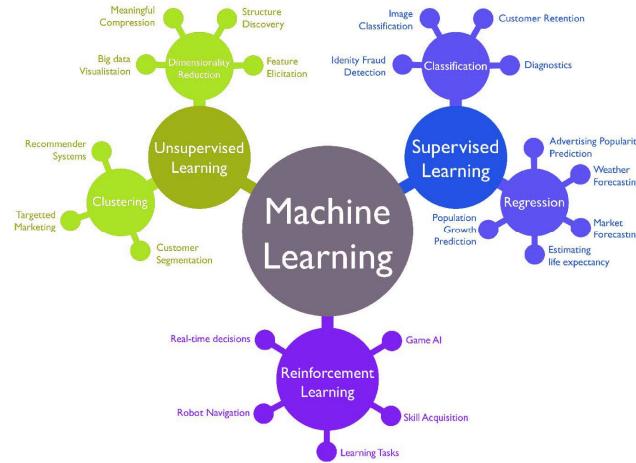


# Supervised Learning



- Applications:
  - Text categorization
  - Face Detection
  - Signature recognition
  - Customer discovery
  - Spam detection
  - Weather forecasting
  - Predicting prices
  - Stock price predictions

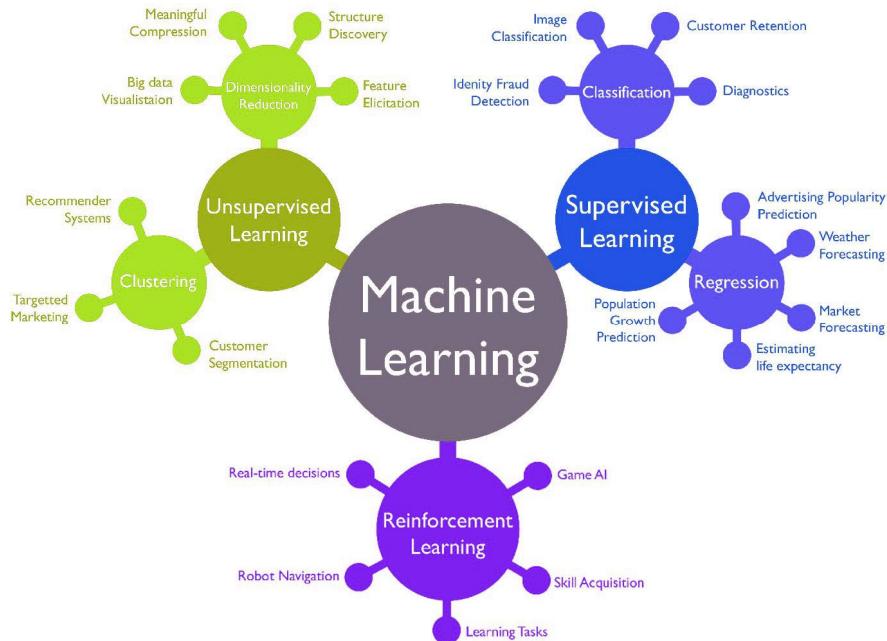
# Reinforcement Learning



- Reward/Punishment would be applied
- It's good for robot navigation and online decision making

- RL is an area of machine learning concerned with how intelligent agents ought to take actions in an environment in order to maximize the notion of cumulative reward.
- Reinforcement learning differs from supervised learning in not needing labelled input/output pairs be presented, and in not needing sub-optimal actions to be explicitly corrected. Instead the focus is on finding a balance between exploration.

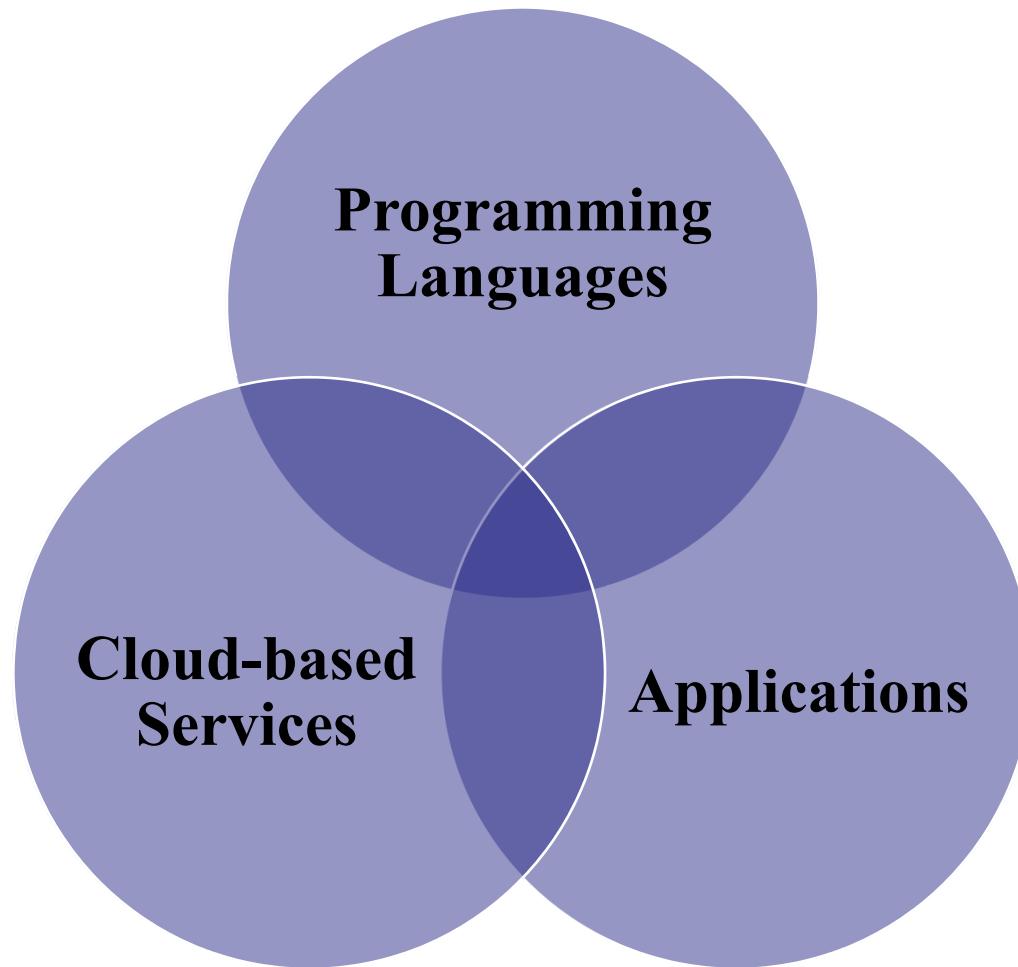
# Unsupervised Learning



- UL is a type of machine learning in which the algorithm is not provided with any pre-assigned labels or scores/outputs for the training data.
- So, UL algorithms must first self-discover any naturally occurring patterns in that training data set.
- Common examples include clustering, where the algorithm automatically groups its training examples into categories with similar features, and principal component analysis, where the algorithm finds ways to compress the training data set by identifying which features are most useful.

# Tools and Implementation

# Approaches



# Programming Languages

R

Octave/Matlab

Python

Anaconda

Tensorflow/Keras

PyTorch

SKLearn

# Applications

**Orange**

**Rapidminer**

**H2O.ai**

**Cnvrg.io**

**Spell**

# Cloud-Based Services

**Google Co-laboratory and Cloud AI platform**

**IBM Watson Machine Learning**

**Kaggle Kernel**

**Coclac**

**Microsoft Azure ML**

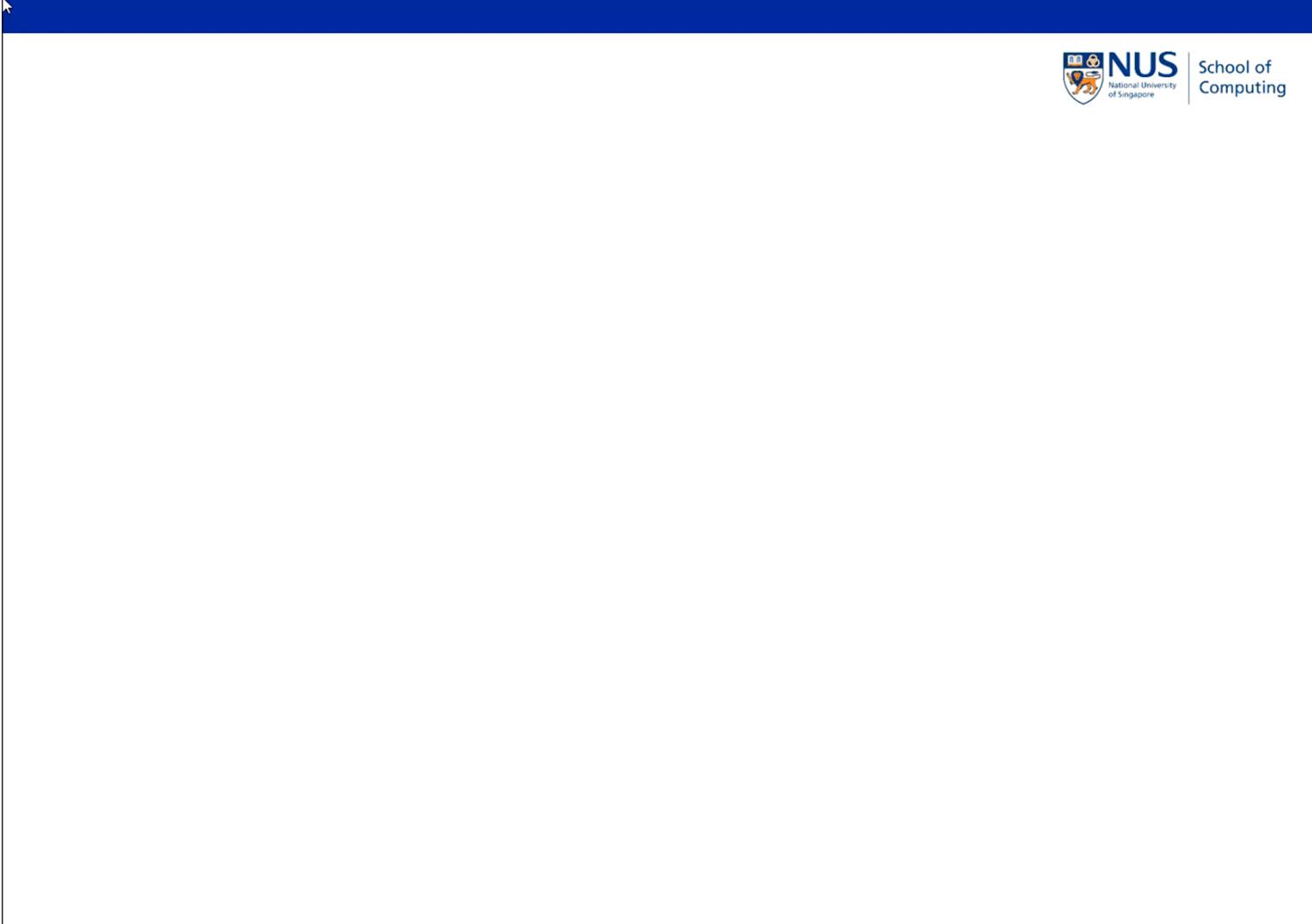
**Amazon AWS Machine Learning**

# UnSupervised Learning, Clustering

## ANIMATIONS



# UnSupervised Learning, Clustering



The slide features a large, empty white rectangular area in the center, likely intended for displaying content such as a video or a presentation slide. This central area is framed by a dark blue border at the top and bottom.

At the very bottom of the slide, there is a row of small, circular icons, each containing a different symbol, typical of a presentation navigation bar. These icons include symbols for back, forward, search, and other presentation controls.

# Part 3

Rapidminer

# What is RapidMiner?

# RAPIDMINER

## Depth for Data Scientists, Simplified for Everyone Else

Join the 40,000+ global organizations in every industry who use the RapidMiner data science platform to drive revenue, reduce costs, and avoid risk.

RapidMiner provides data mining and machine learning procedures including: data loading and transformation (ETL), data preprocessing and visualization, predictive analytics and statistical modeling, evaluation, and deployment.

RapidMiner provides a GUI to design and execute analytical workflows.



# RAPIDMINER

## The Hard Way

- › Slow to collaborate & transform
- › Acquire highly specialized expertise
- › Choose complexity or oversimplification
- › Opaque, fragmented processes
- › Commit to a specific ecosystem

## The RapidMiner Way

- › Transformational business impact
- › Upskill your organization
- › Depth for data scientists, simplified for everyone else
- › Easy to trust, tune & explain
- › Future-proof innovation, portability & extensibility



## RapidMiner Offers a Complete Path to Fully Automated Data Science



### Turbo Prep

Blend, wrangle, and cleanse data with intuitive data prep that's simple to use



### Auto Model

Create models in 5 clicks with automated machine learning and data science



### Model Ops

Model deployment and management made easy – for any model and any user



*Deliver impact with AI in three simple steps*



# RAPIDMINER

## Feedback



### FORRESTER®

RapidMiner is a Leader in The Forrester Wave: Multimodal Predictive Analytics & Machine Learning Solutions, Q3 2020

### Gartner

RapidMiner is a Visionary in the 2021 Gartner Magic Quadrant for Data Science and Machine Learning Platforms



RapidMiner is a June 2020 Gartner Peer Insights Customers' Choice for Data Science and Machine Learning Platforms for the third time in a row

### G<sup>2</sup> CROWD

RapidMiner is the Highest Rated, Easiest to Use Data Science and Machine Learning Platform and was named a Leader in G2's Spring 2021 Report.

# RAPIDMINER

## Best Machine Learning Software in 2021

Name	Platform	Algorithms or Features
<b>Scikit Learn</b>	Linux, Mac OS, Windows, a Python/ R package	Classification, Regression, Clustering Pre-processing, Model Selection Dimensionality reduction.
<b>PyTorch</b>	Linux, Mac OS, Windows, a Python/ R package	Autograd Module, Optim Module, ANN Module
<b>TensorFlow/ Keras</b>	Linux, Mac OS, Windows, a Python/ R package	Provides a library for dataflow programming, ANN, Deep Learning
<b>Shogun</b>	Windows, Linux, Mac OS	Regression, Classification, Clustering Support vector machines, Dimensionality reduction Online learning etc.
<b>Google Colab and Cloud AI Platform</b>	Cloud Service	Ready to run Python and R programs.
<b>Amazon Web Services (AWS)</b>	Cloud Service	Based on SageMaker got a web-based visual interface, plus engines. Pay-as-you-go pricing
<b>IBM Watson Machine Learning</b>	Cloud Service	Contains Auto AI, Machine Learning Cloud, Machine Learning Server, and Machine Learning on IBM Cloud Pak for Data
<b>MS Azure ML Studio</b>	Cloud Service	An end-to-end solution, Supports open-source tools and frameworks, including PyTorch and TensorFlow

# RAPIDMINER

## Best Machine Learning Software in 2021

Name	Platform	Algorithms or Features
<b>Weka</b>	Linux, Mac OS, Windows	Data preparation, Classification, Regression Clustering, Visualization, Association rules mining
<b>KNIME</b>	Linux, Mac OS, Windows	Can work with large data volume. Supports text mining & image mining through plugins
<b>Apache Mahout</b>	Cross-platform	Preprocessors, Regression, Clustering Recommenders, Distributed Linear Algebra.
<b>Rapid Miner</b>	Linux, Mac OS, Windows	Data loading & Transformation Data preprocessing & visualization. Regression, ANN, Deep Learning (simple) models. Speed is acceptable.
<b>MATLAB/ Octave</b>	Linux, Mac OS, Windows, A complete PL	Excellent performance in Modelling, Does not support other languages such as Python and R
<b>Python</b>	Linux, Mac OS, Windows, A complete PL	A solution from the scratch, sometimes implementation is lengthy and complicated.

# Installation

# HOW TO START?

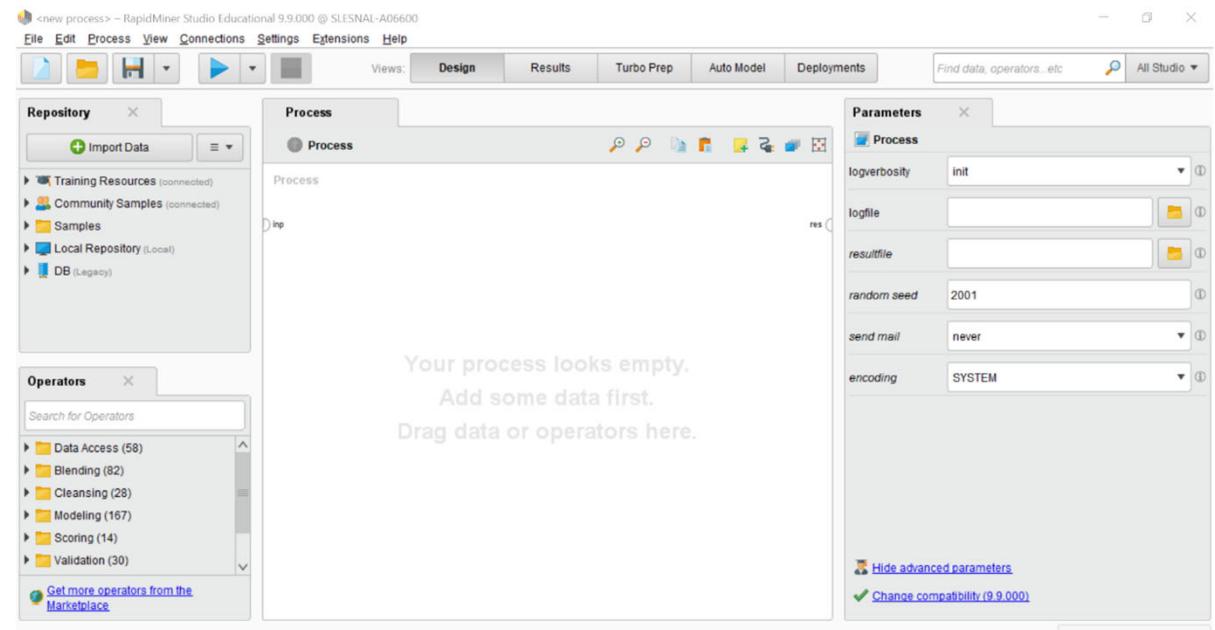
Install it

The basic **RapidMiner Studio** is free.

Run it !!!

Mind your OS

[www.rapidminer.com](http://www.rapidminer.com)



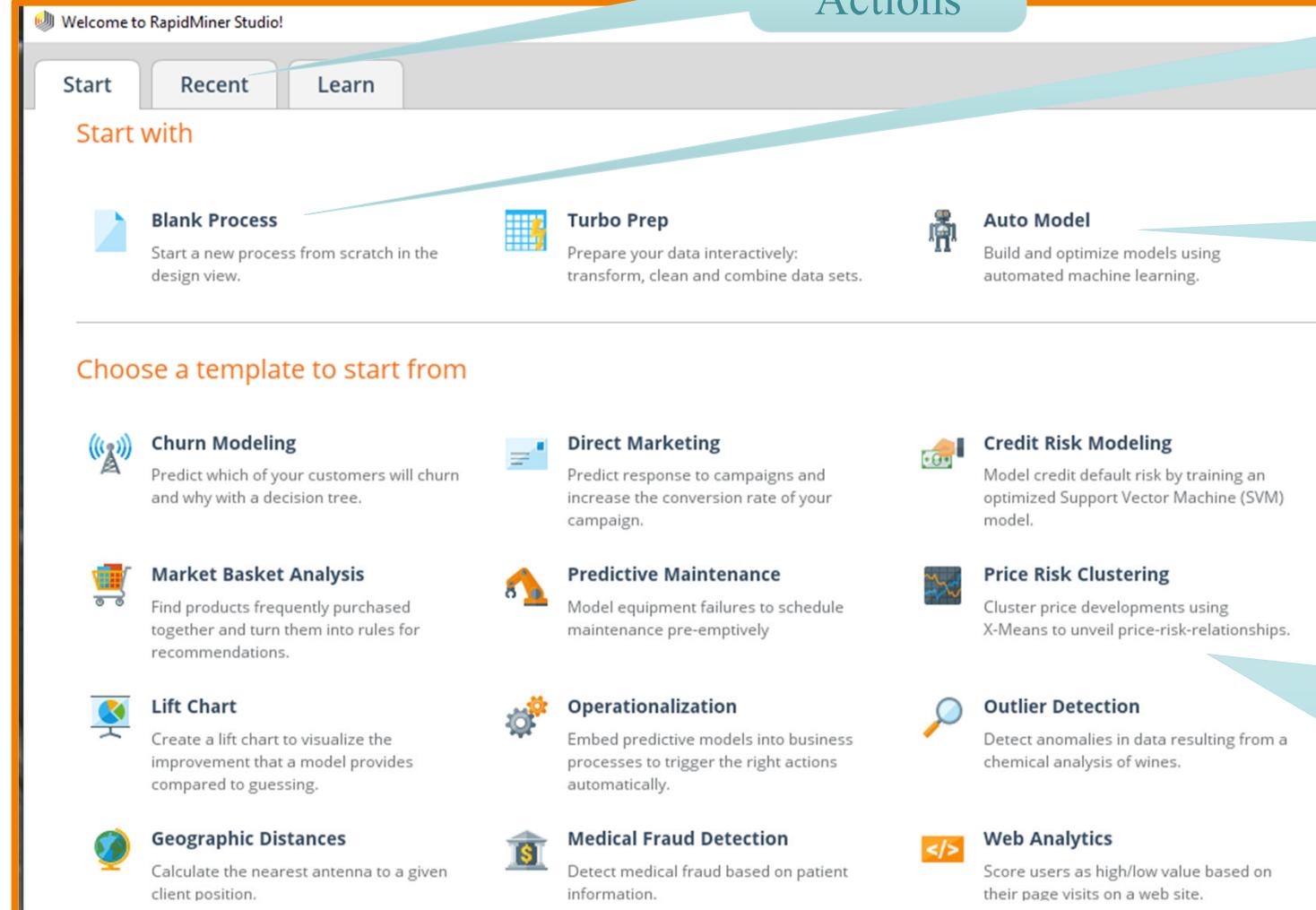
# HOW TO START?

**Recent Actions**

**Start with this**

**Advanced mode**

**Ready to use ML and DA Templates. We'll see these later**



Welcome to RapidMiner Studio!

Start Recent Learn

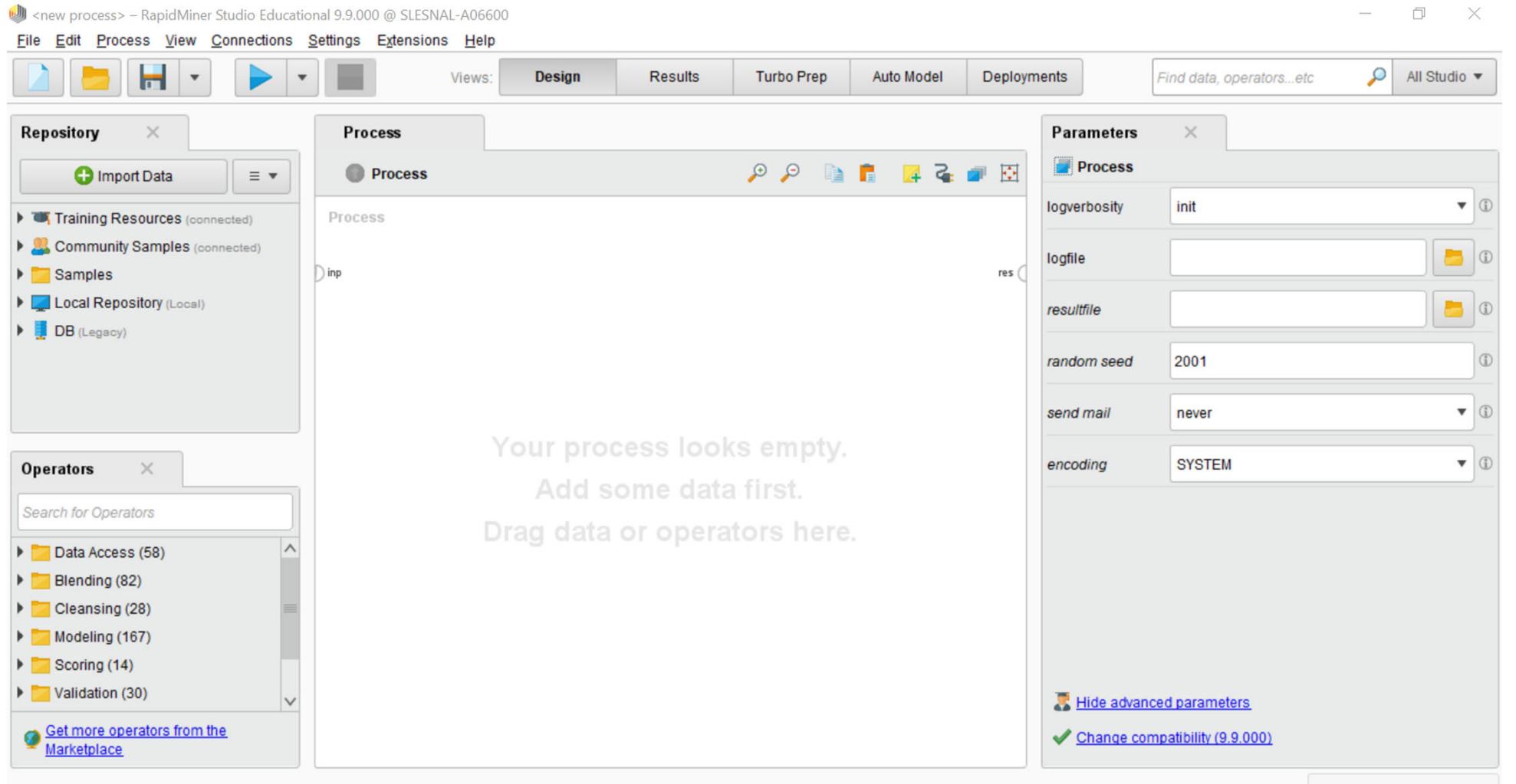
**Start with**

- Blank Process**: Start a new process from scratch in the design view.
- Turbo Prep**: Prepare your data interactively: transform, clean and combine data sets.
- Auto Model**: Build and optimize models using automated machine learning.

**Choose a template to start from**

<b>Churn Modeling</b> Predict which of your customers will churn and why with a decision tree.	<b>Direct Marketing</b> Predict response to campaigns and increase the conversion rate of your campaign.	<b>Credit Risk Modeling</b> Model credit default risk by training an optimized Support Vector Machine (SVM) model.
<b>Market Basket Analysis</b> Find products frequently purchased together and turn them into rules for recommendations.	<b>Predictive Maintenance</b> Model equipment failures to schedule maintenance pre-emptively.	<b>Price Risk Clustering</b> Cluster price developments using X-Means to unveil price-risk-relationships.
<b>Lift Chart</b> Create a lift chart to visualize the improvement that a model provides compared to guessing.	<b>Operationalization</b> Embed predictive models into business processes to trigger the right actions automatically.	<b>Outlier Detection</b> Detect anomalies in data resulting from a chemical analysis of wines.
<b>Geographic Distances</b> Calculate the nearest antenna to a given client position.	<b>Medical Fraud Detection</b> Detect medical fraud based on patient information.	<b>Web Analytics</b> Score users as high/low value based on their page visits on a web site.

# HOW TO START?

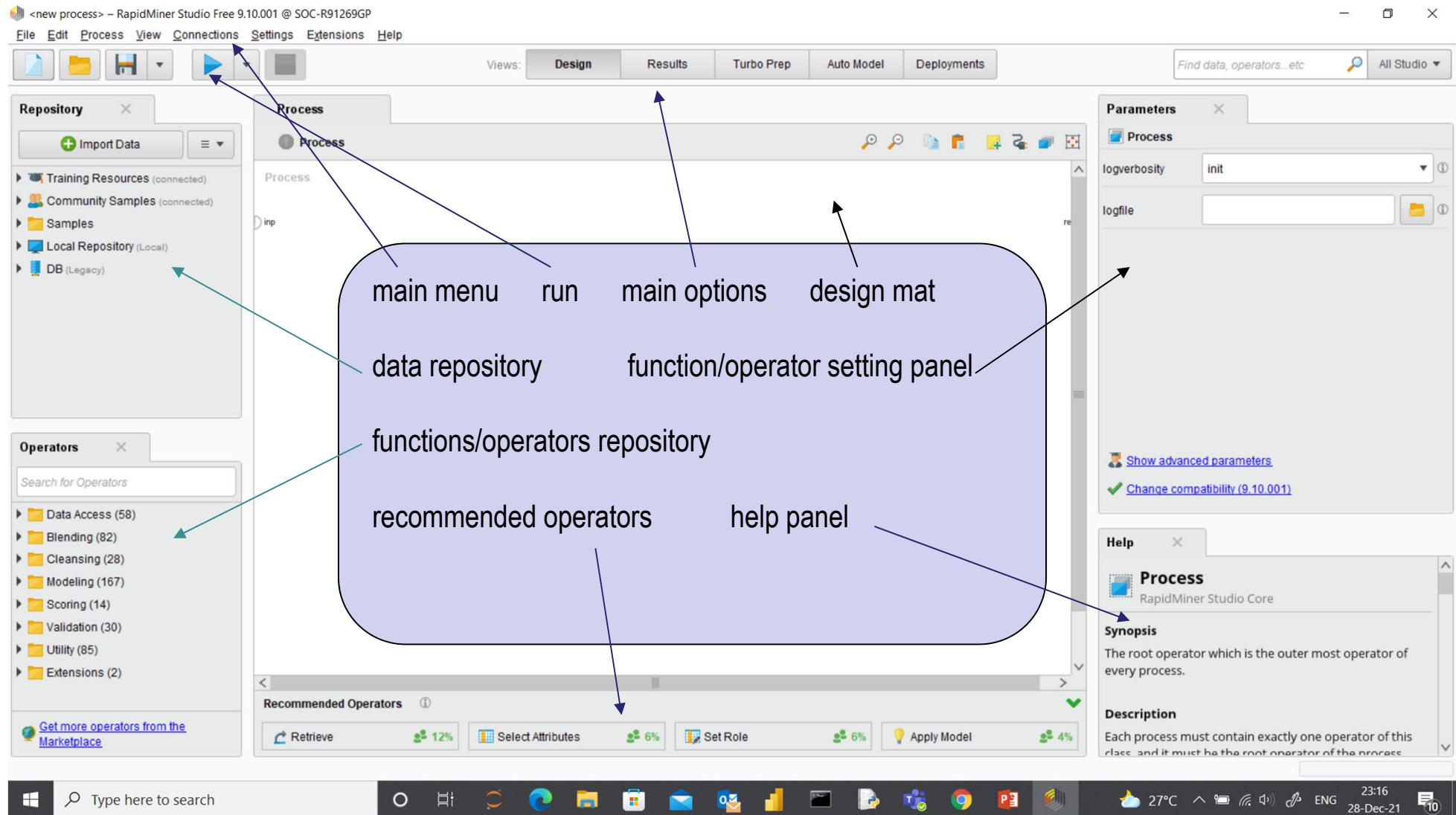


The screenshot shows the RapidMiner Studio interface. The top menu bar includes File, Edit, Process, View, Connections, Settings, Extensions, and Help. The toolbar below has icons for Import Data, Repository, Operators, and Process. The main workspace is titled "Process" and displays the message "Your process looks empty. Add some data first. Drag data or operators here." The left sidebar contains the "Repository" panel with sections for Training Resources, Community Samples, Samples, Local Repository, and DB; and the "Operators" panel with categories for Data Access, Blending, Cleansing, Modeling, Scoring, and Validation. The right sidebar contains the "Parameters" panel with settings for logverbosity (init), logfile, resultfile, random seed (2001), send mail (never), and encoding (SYSTEM). A status bar at the bottom indicates "Change compatibility (9.9.000)".

RAPIDMINER MAIN PAGE, WAIT TO SEE THAT.

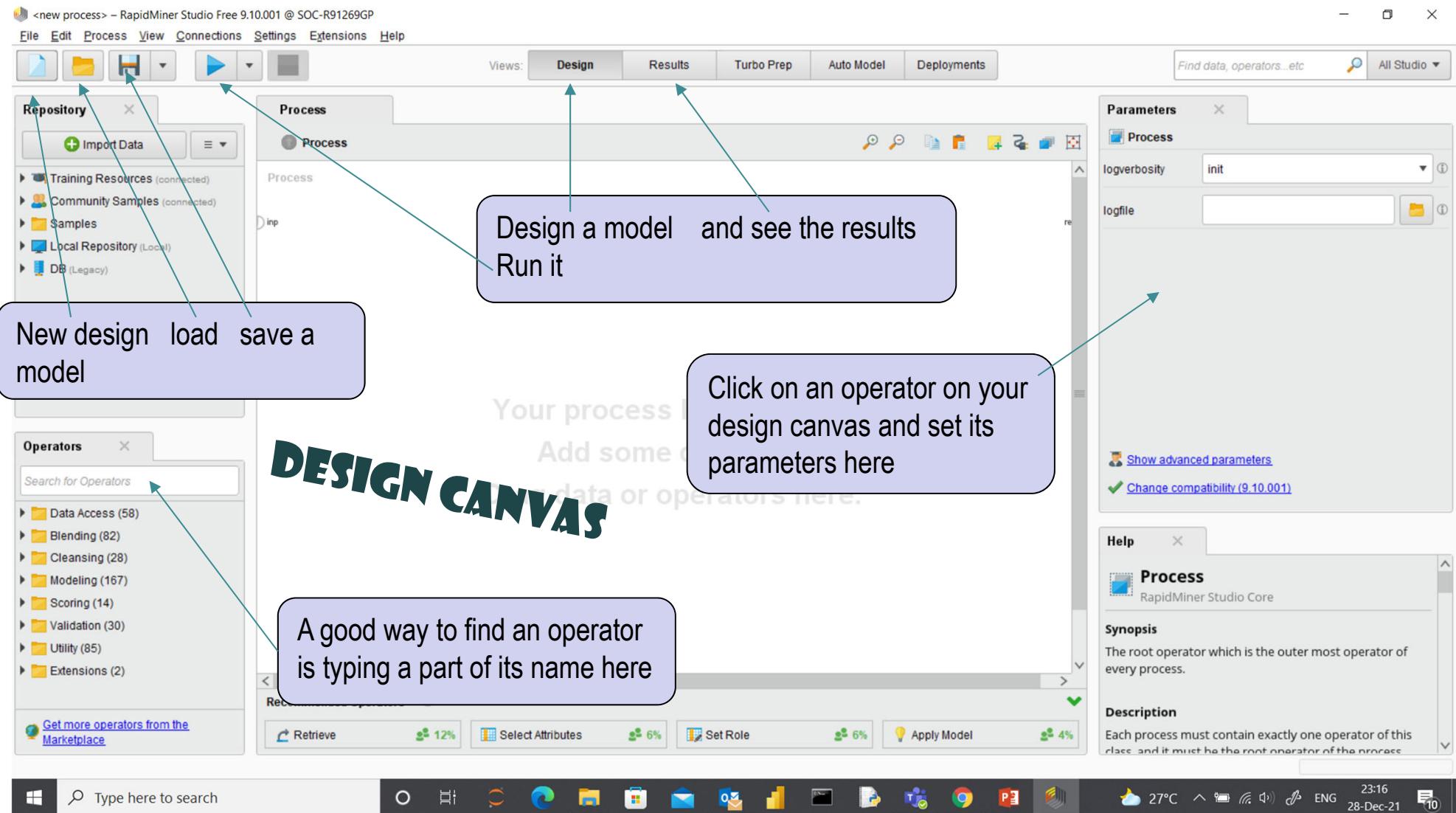
# Environment & Libraries

# RAPIDMINER MAIN PAGE



# Rapidminer Main page

**Design CANVAS**



New design load save a model

A good way to find an operator is typing a part of its name here

Design a model and see the results  
Run it

Click on an operator on your design canvas and set its parameters here

Design

Results

Turbo Prep

Auto Model

Deployments

Find data, operators...etc

All Studio

Process

Repository

Import Data

Training Resources (connected)

Community Samples (connected)

Samples

Local Repository (Local)

DB (Legacy)

Parameters

Process

logverbosity init

logfile

Show advanced parameters

Change compatibility (9.10.001)

Help

Process

RapidMiner Studio Core

Synopsis

The root operator which is the outer most operator of every process.

Description

Each process must contain exactly one operator of this class, and it must be the root operator of the process.

Type here to search

27°C ENG 23:16 28-Dec-21

# Libraries

<new process> - RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

**Repository**

- + Import Data
- Training Resources (connected)
- Community Samples (connected)
- Samples
- Local Repository (Local)
- DB (Legacy)

**Process**

Your process looks empty.

Start by adding data first.

Add operators here.

Rapidminer v.9 standard library

Click here for extra libraries

**Operators**

Search for Operators

- Data Access (58)
- Blending (82)
- Cleansing (28)
- Modeling (167)
- Scoring (14)
- Validation (30)
- Utility (85)
- Extensions (2)

Get more operators from the Marketplace

**Parameters**

Process

logverbosity: init

logfile:

Show advanced parameters

Change compatibility (9.10.001)

**Help**

**Process**

RapidMiner Studio Core

**Synopsis**

The root operator which is the outer most operator of every process.

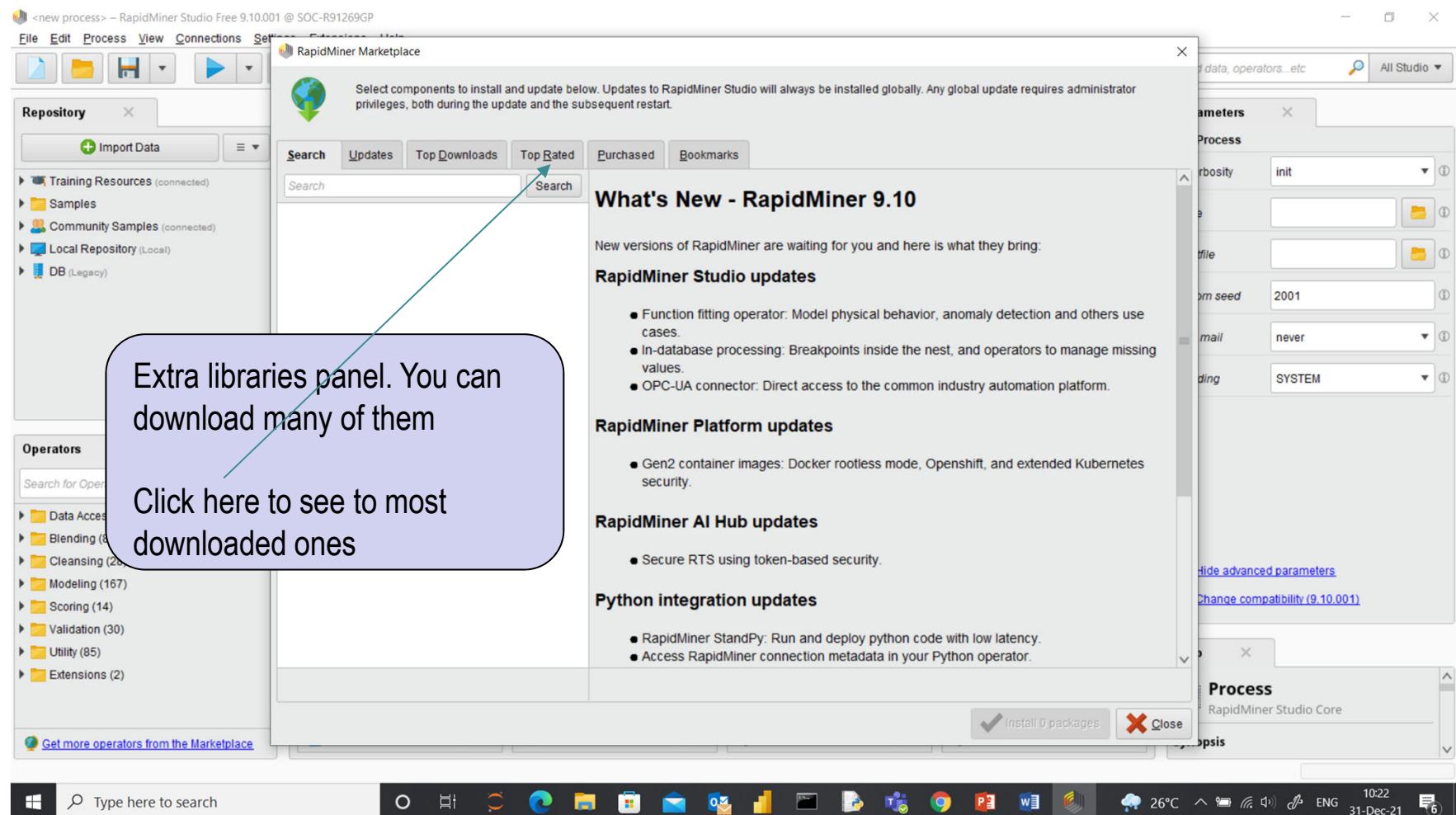
**Description**

Each process must contain exactly one operator of this class, and it must be the root operator of the process.

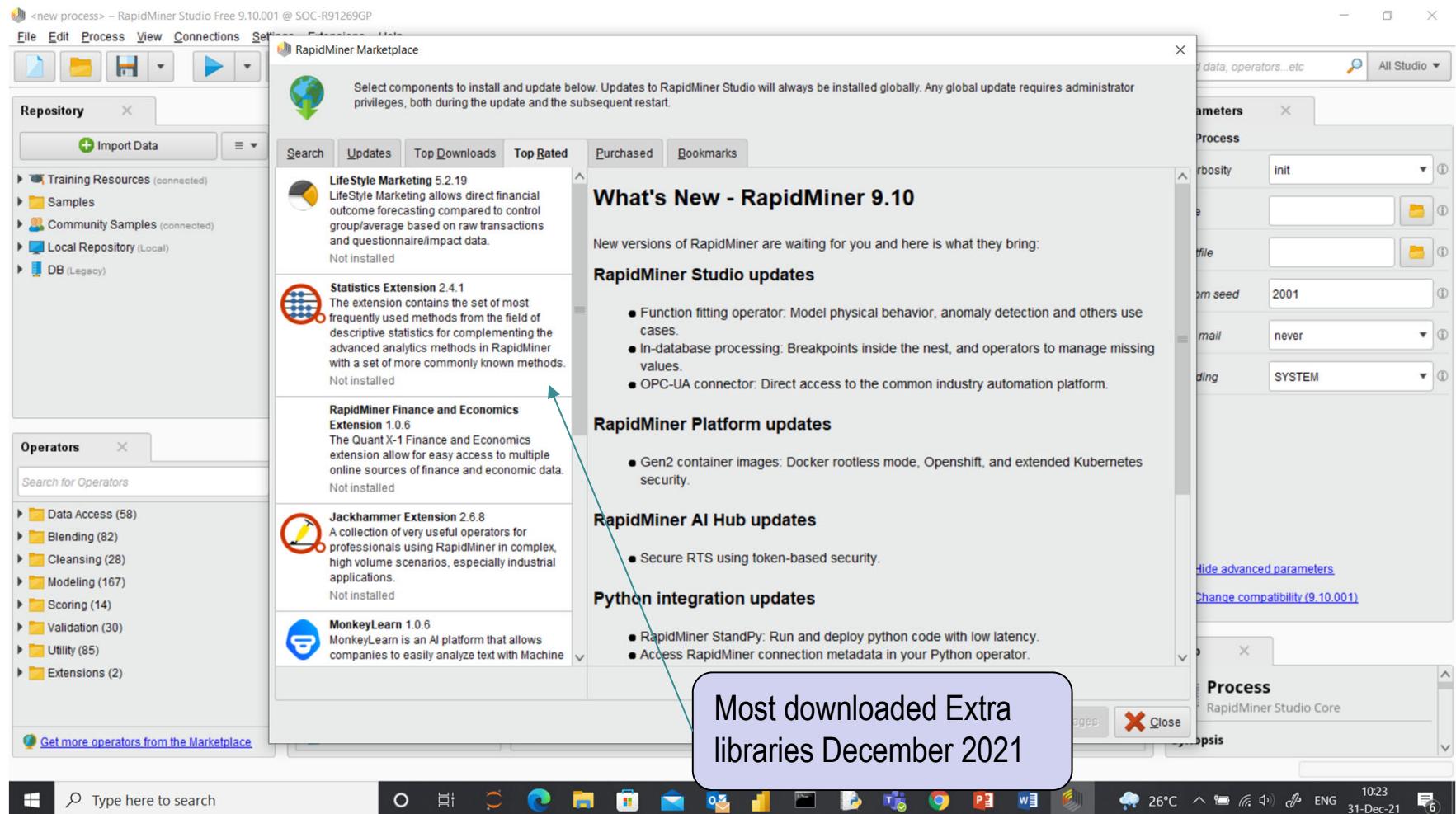
Type here to search

23:16 27°C ENG 28-Dec-21

# Libraries



# Libraries

A screenshot of the RapidMiner Studio Free 9.10.001 interface. The main window shows the 'RapidMiner Marketplace' with a list of available extensions. A callout bubble points to the 'Statistics Extension 2.4.1' entry. The extension details show it allows for descriptive statistics complementing advanced analytics. The 'What's New - RapidMiner 9.10' section highlights various updates like function fitting operators, in-database processing, and OPC-UA connector. The 'Process' tab on the right shows parameters for a process named 'init'. The taskbar at the bottom includes the Start button, a search bar, and various pinned application icons.

**Most downloaded Extra libraries December 2021**

- LifeStyle Marketing 5.2.19**  
LifeStyle Marketing allows direct financial outcome forecasting compared to control group/average based on raw transactions and questionnaire/impact data.  
Not installed
- Statistics Extension 2.4.1**  
The extension contains the set of most frequently used methods from the field of descriptive statistics for complementing the advanced analytics methods in RapidMiner with a set of more commonly known methods.  
Not installed
- RapidMiner Finance and Economics Extension 1.0.6**  
The Quant X-1 Finance and Economics extension allow for easy access to multiple online sources of finance and economic data.  
Not installed
- Jackhammer Extension 2.6.8**  
A collection of very useful operators for professionals using RapidMiner in complex, high volume scenarios, especially industrial applications.  
Not installed
- MonkeyLearn 1.0.6**  
MonkeyLearn is an AI platform that allows companies to easily analyze text with Machine Learning.  
Not installed

**What's New - RapidMiner 9.10**

New versions of RapidMiner are waiting for you and here is what they bring:

**RapidMiner Studio updates**

- Function fitting operator: Model physical behavior, anomaly detection and others use cases.
- In-database processing: Breakpoints inside the nest, and operators to manage missing values.
- OPC-UA connector: Direct access to the common industry automation platform.

**RapidMiner Platform updates**

- Gen2 container images: Docker rootless mode, Openshift, and extended Kubernetes security.

**RapidMiner AI Hub updates**

- Secure RTS using token-based security.

**Python integration updates**

- RapidMiner StandPy: Run and deploy python code with low latency.
- Access RapidMiner connection metadata in your Python operator.

## Practice 2, Later at Home

- Install Rapidminer
- Get familiar with it
- Try to test a few operators listed below
  - FP-Growth
  - Apply Model
  - Detect Outliers (Distances)
- Try to install the most popular extra library.

# Part 4

## Clustering: K-Means

# K-Means Clustering

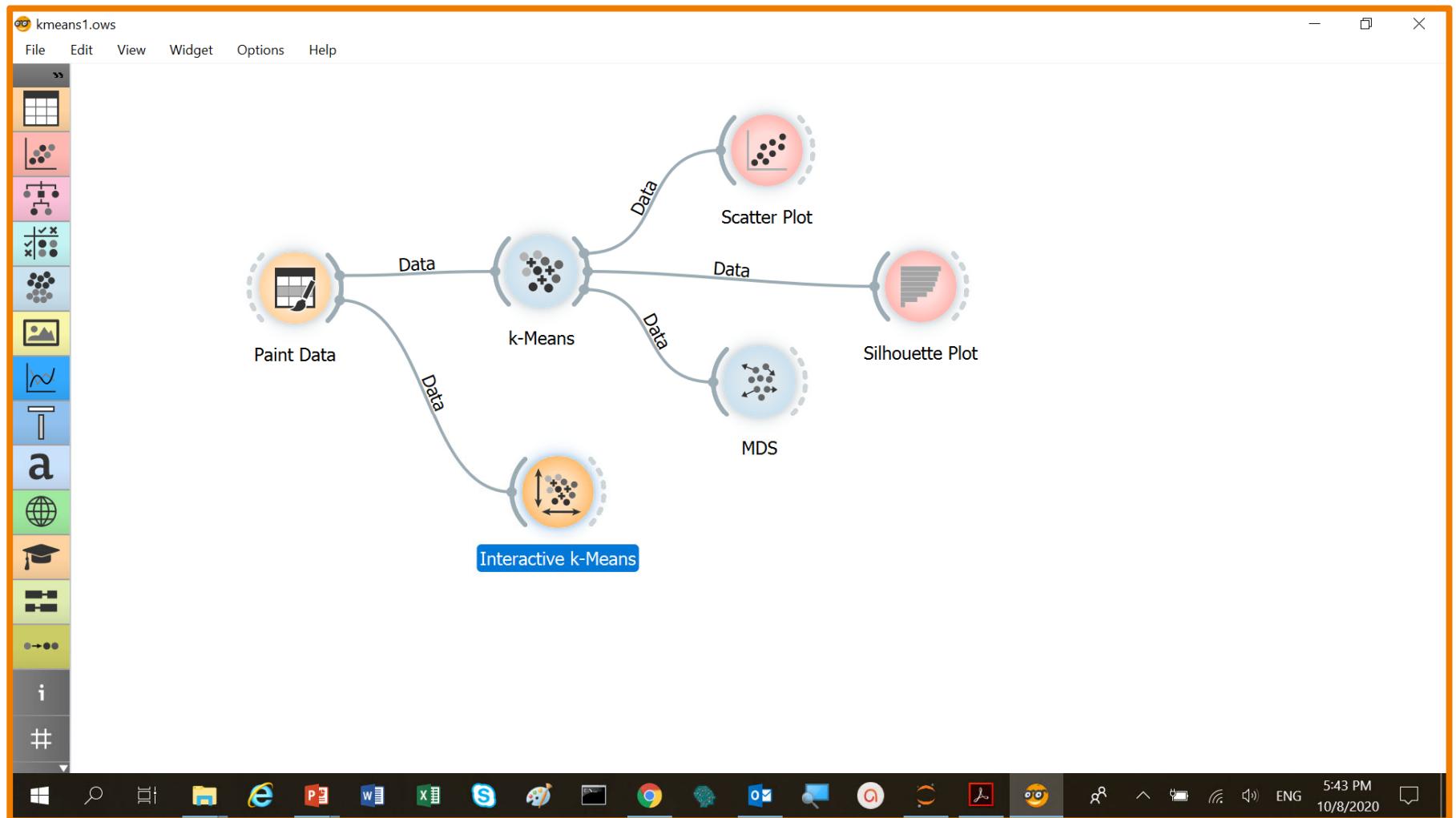
# K-MEANS CLUSTERING

- Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters).
- In simple words, the aim is to segregate groups with similar traits and assign them into clusters.

# K-MEANS CLUSTERING

- K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.
- Typically, unsupervised algorithms make inferences from datasets using only input vectors without referring to known, or labelled, outcomes.
- K-means algorithm identifies  $k$  number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.
- It will try to minimize intra-cluster distances, and maximize inter-cluster distances, using a MSD or MAD schemes.

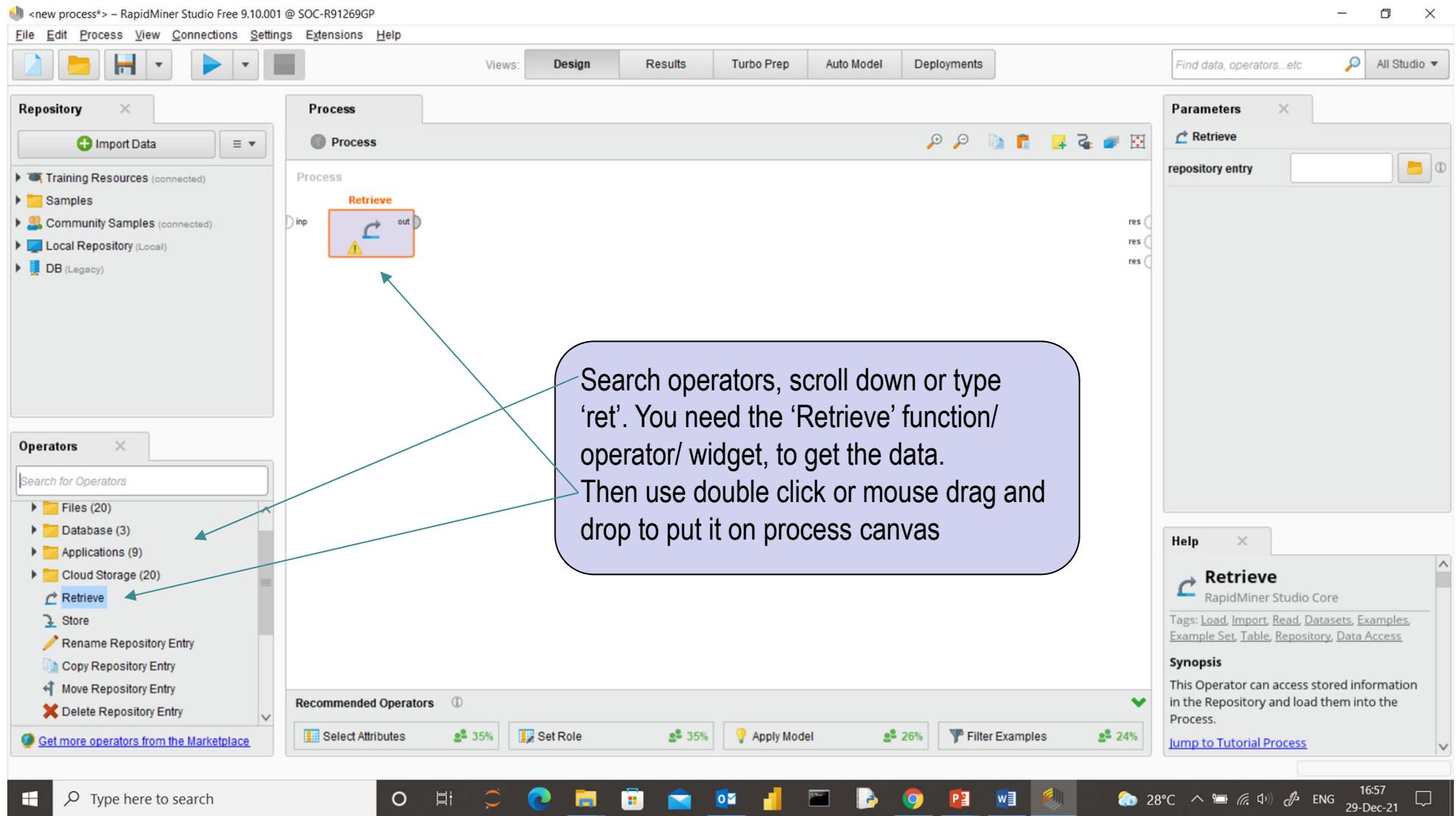
# Visualization of K-Means in Orange



# Example 1

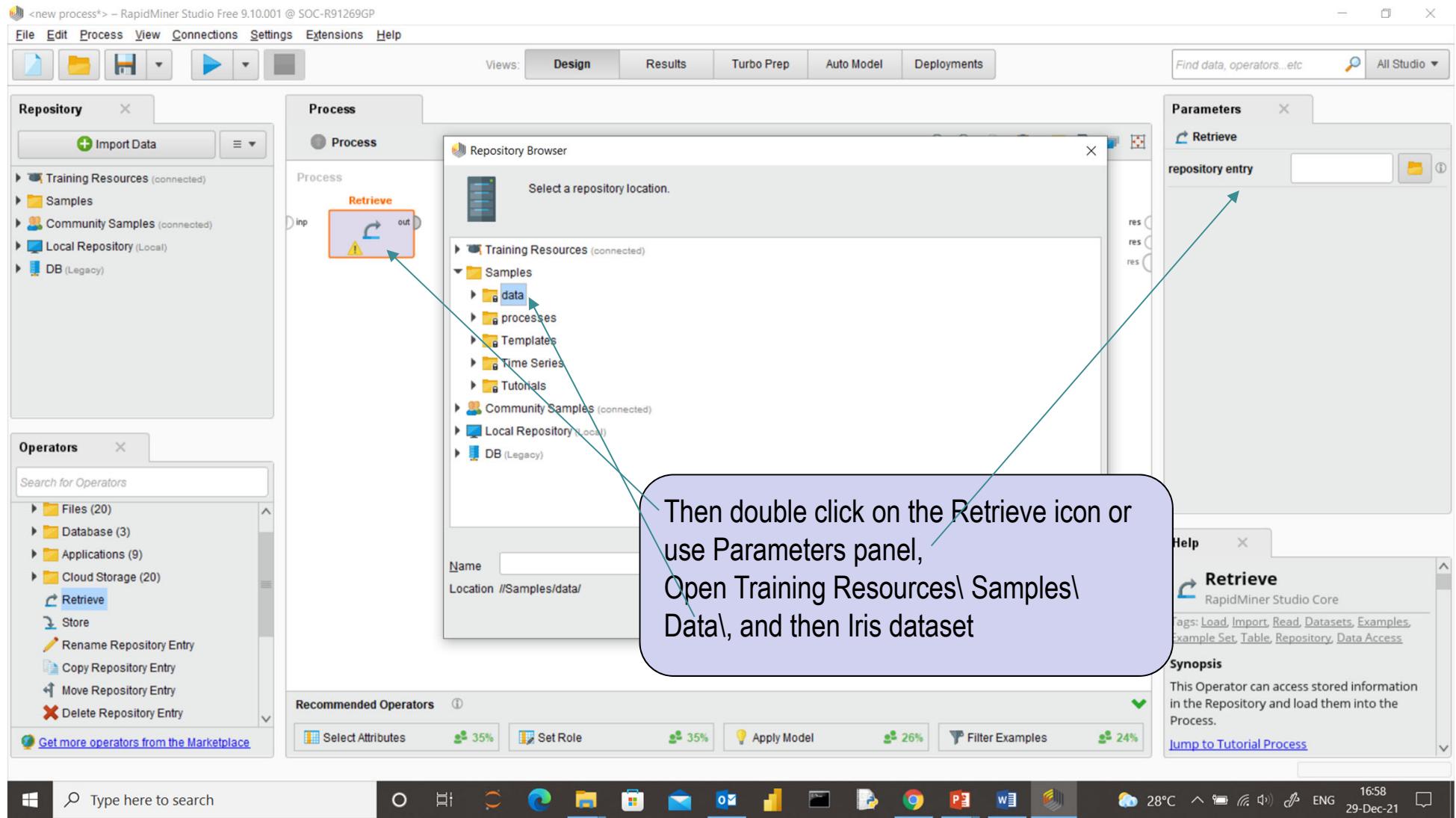
# Example 1: Bunch of Iris Flowers

Search operators, scroll down or type 'ret'. You need the 'Retrieve' function/ operator/ widget, to get the data.  
Then use double click or mouse drag and drop to put it on process canvas



The screenshot shows the RapidMiner Studio Free interface. On the left, the 'Operators' palette has 'Retrieve' selected. In the center, the 'Process' canvas shows a 'Retrieve' operator with 'inp' and 'out' ports. On the right, the 'Parameters' palette shows 'repository entry' selected. A callout box with the instructions is overlaid on the interface, pointing to the 'Operators' palette and the 'Retrieve' operator on the canvas.

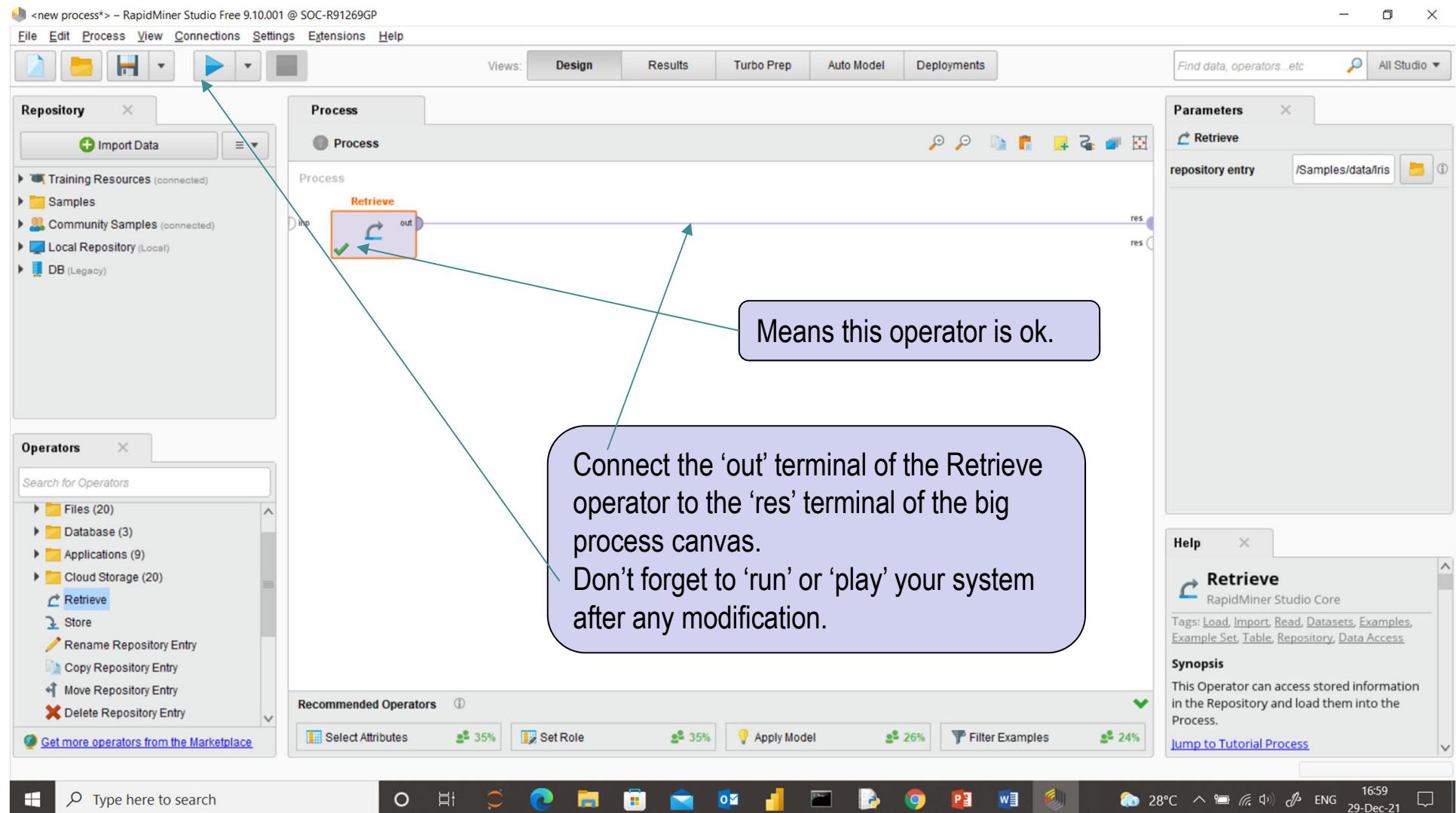
# Example 1: Bunch of Iris Flowers

Then double click on the Retrieve icon or use Parameters panel, Open Training Resources\ Samples\ Data\, and then Iris dataset

The screenshot shows the RapidMiner Studio interface with the following details:

- Repository Panel:** Shows 'Training Resources (connected)', 'Samples', 'Community Samples (connected)', 'Local Repository (Local)', and 'DB (Legacy)'.
- Process Panel:** Displays a process flow with an 'inp' port, a 'Retrieve' operator (highlighted with a red box), and an 'out' port.
- Operators Panel:** Shows a list of operators including 'Files (20)', 'Database (3)', 'Applications (9)', 'Cloud Storage (20)', 'Retrieve' (highlighted with a red box), 'Store', 'Rename Repository Entry', 'Copy Repository Entry', 'Move Repository Entry', and 'Delete Repository Entry'. A link to 'Get more operators from the Marketplace' is also present.
- Process Browser Window:** A modal window titled 'Repository Browser' with the instruction 'Select a repository location.' It shows a tree view of repository entries under 'Samples' (including 'data', 'processes', 'Templates', 'Time Series', 'Tutorials'), 'Community Samples (connected)', 'Local Repository (Local)', and 'DB (Legacy)'. A blue arrow points from the 'data' entry in the tree to the 'data' entry in the 'Location' field at the bottom of the window.
- Parameters Panel:** Shows a 'repository entry' field with three 'res' entries. A blue arrow points from the 'res' entry in the field to the 'res' entry in the 'repository entry' list on the right.
- Help Panel:** Provides a detailed description of the 'Retrieve' operator, including its synopsis: 'This Operator can access stored information in the Repository and load them into the Process.'
- System Tray:** Shows system status icons including battery level (28%), network signal, volume, and date/time (16:58, 29-Dec-21).

# Example 1: Bunch of Iris Flowers



Means this operator is ok.

Connect the 'out' terminal of the Retrieve operator to the 'res' terminal of the big process canvas.  
Don't forget to 'run' or 'play' your system after any modification.

<new process\*> - RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators... etc All Studio

Repository

Import Data

Training Resources (connected)  
Samples  
Community Samples (connected)  
Local Repository (Local)  
DB (Legacy)

Process

Process

Retrieve

inp out res res

Operators

Search for Operators

Files (20)  
Database (3)  
Applications (9)  
Cloud Storage (20)  
Retrieve  
Store  
Rename Repository Entry  
Copy Repository Entry  
Move Repository Entry  
Delete Repository Entry

Get more operators from the Marketplace

Parameters

repository entry /Samples/data/Iris

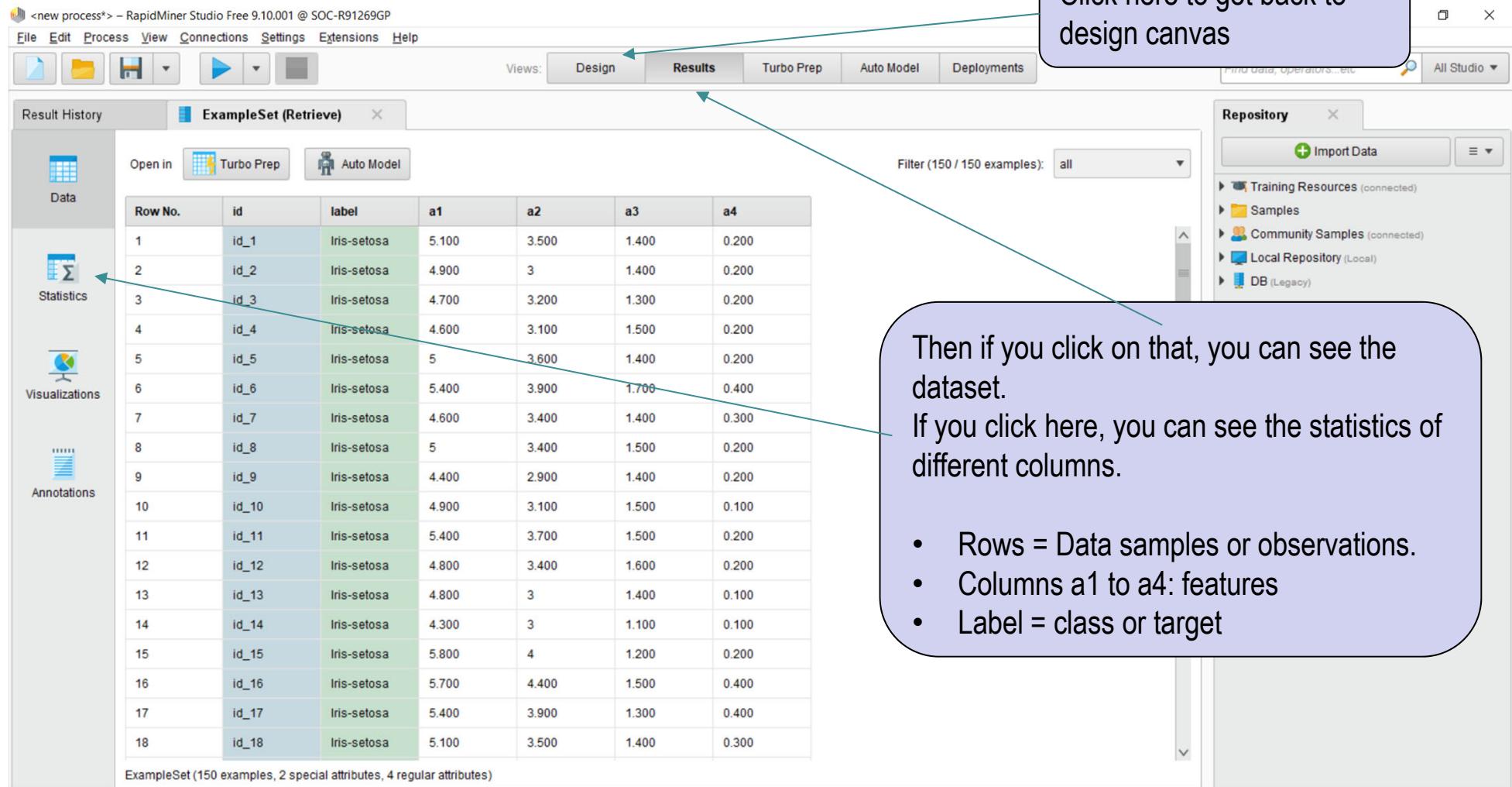
Help

**Retrieve**  
RapidMiner Studio Core  
Tags: Load, Import, Read, Datasets, Examples, Example Set, Table, Repository, Data Access  
Synopsis  
This Operator can access stored information in the Repository and load them into the Process.  
Jump to Tutorial Process

Type here to search

# Example 1: Bunch of Iris Flowers

Click here to get back to design canvas



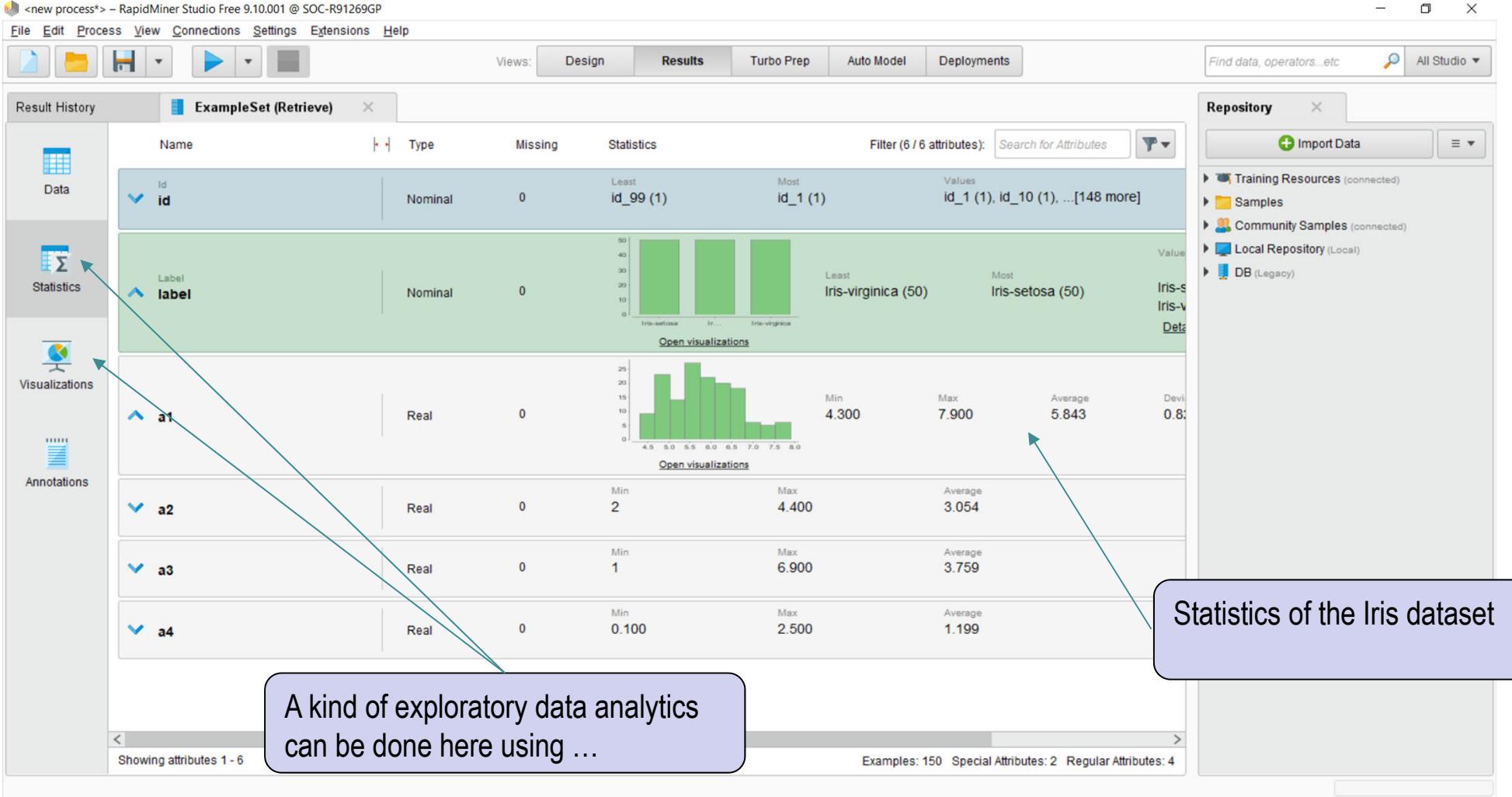
Row No.	id	label	a1	a2	a3	a4
1	id_1	Iris-setosa	5.100	3.500	1.400	0.200
2	id_2	Iris-setosa	4.900	3	1.400	0.200
3	id_3	Iris-setosa	4.700	3.200	1.300	0.200
4	id_4	Iris-setosa	4.600	3.100	1.500	0.200
5	id_5	Iris-setosa	5	3.600	1.400	0.200
6	id_6	Iris-setosa	5.400	3.900	1.700	0.400
7	id_7	Iris-setosa	4.600	3.400	1.400	0.300
8	id_8	Iris-setosa	5	3.400	1.500	0.200
9	id_9	Iris-setosa	4.400	2.900	1.400	0.200
10	id_10	Iris-setosa	4.900	3.100	1.500	0.100
11	id_11	Iris-setosa	5.400	3.700	1.500	0.200
12	id_12	Iris-setosa	4.800	3.400	1.600	0.200
13	id_13	Iris-setosa	4.800	3	1.400	0.100
14	id_14	Iris-setosa	4.300	3	1.100	0.100
15	id_15	Iris-setosa	5.800	4	1.200	0.200
16	id_16	Iris-setosa	5.700	4.400	1.500	0.400
17	id_17	Iris-setosa	5.400	3.900	1.300	0.400
18	id_18	Iris-setosa	5.100	3.500	1.400	0.300

ExampleSet (150 examples, 2 special attributes, 4 regular attributes)

Then if you click on that, you can see the dataset.  
If you click here, you can see the statistics of different columns.

- Rows = Data samples or observations.
- Columns a1 to a4: features
- Label = class or target

# Example 1: Bunch of Iris Flowers

A screenshot of the RapidMiner Studio interface. The main window displays the 'ExampleSet (Retrieve)' view, showing statistics for six attributes: 'id' (Nominal), 'label' (Nominal), 'a1' (Real), 'a2' (Real), 'a3' (Real), and 'a4' (Real). The 'label' attribute has three categories: 'Iris-setosa', 'Iris-versicolor', and 'Iris-virginica'. The 'a1' attribute has a histogram showing values from 4.300 to 7.900 with an average of 5.843. Arrows point from the sidebar categories 'Statistics' and 'Visualizations' to the respective sections of the table. A callout box labeled 'Statistics of the Iris dataset' points to the 'label' attribute section. Another callout box labeled 'A kind of exploratory data analytics can be done here using ...' points to the 'a1' attribute section.

**Result History**

**ExampleSet (Retrieve)**

**Data**

**Statistics**

**Visualizations**

**Annotations**

Name Type Missing Statistics Filter (6 / 6 attributes): Search for Attributes

Name	Type	Missing	Statistics	Filter (6 / 6 attributes):	Search for Attributes
<b>id</b>	Nominal	0	Least: id_99 (1)      Most: id_1 (1)      Values: id_1 (1), id_10 (1), ...[148 more]		
<b>label</b>	Nominal	0	Least: Iris-virginica (50)      Most: Iris-setosa (50)		
<b>a1</b>	Real	0	Min: 4.300      Max: 7.900      Average: 5.843      Deviation: 0.83	Open visualizations	
<b>a2</b>	Real	0	Min: 2      Max: 4.400      Average: 3.054		
<b>a3</b>	Real	0	Min: 1      Max: 6.900      Average: 3.759		
<b>a4</b>	Real	0	Min: 0.100      Max: 2.500      Average: 1.199		

Showing attributes 1 - 6

Examples: 150 Special Attributes: 2 Regular Attributes: 4

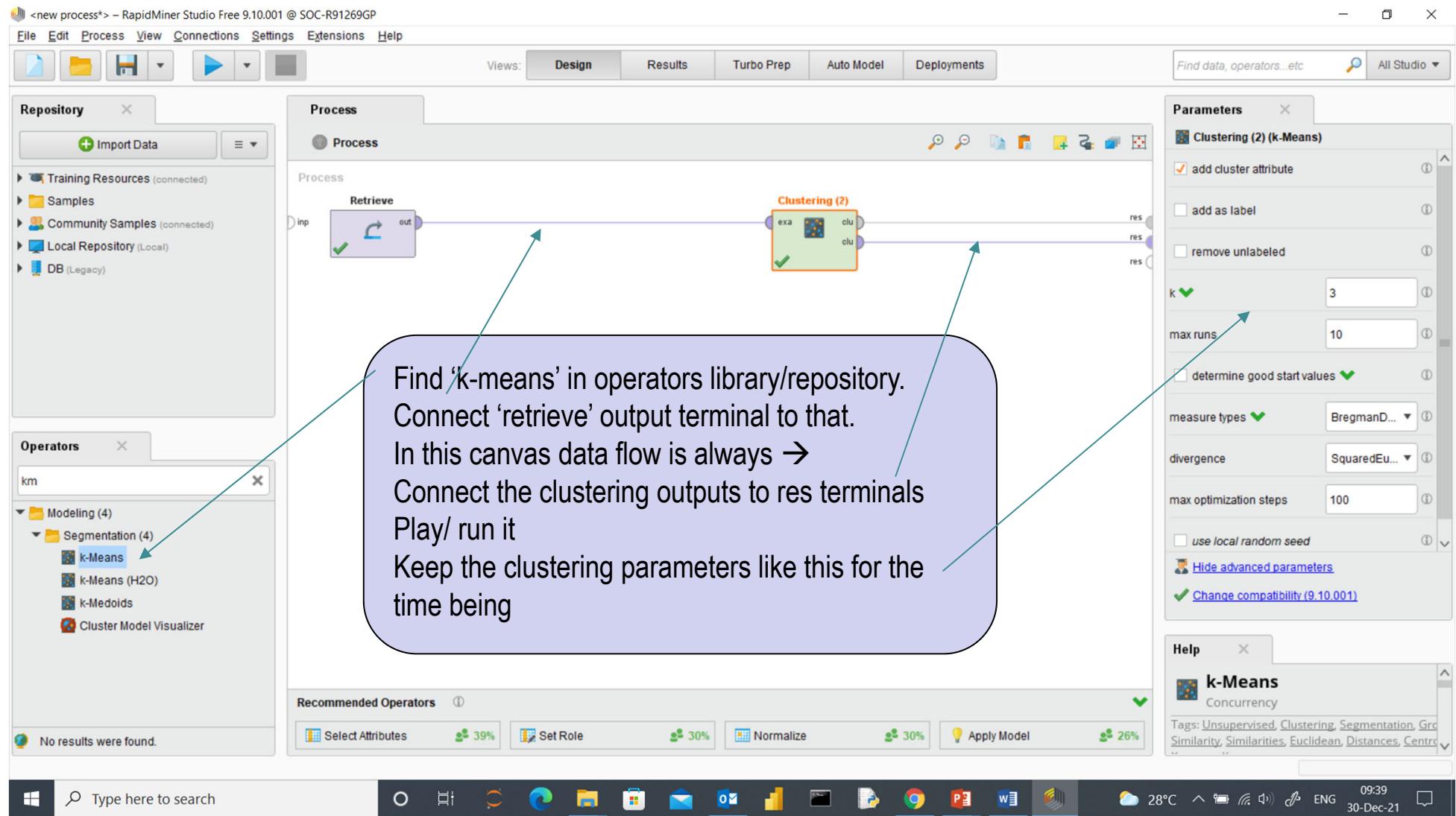
**Repository**

**Import Data**

- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

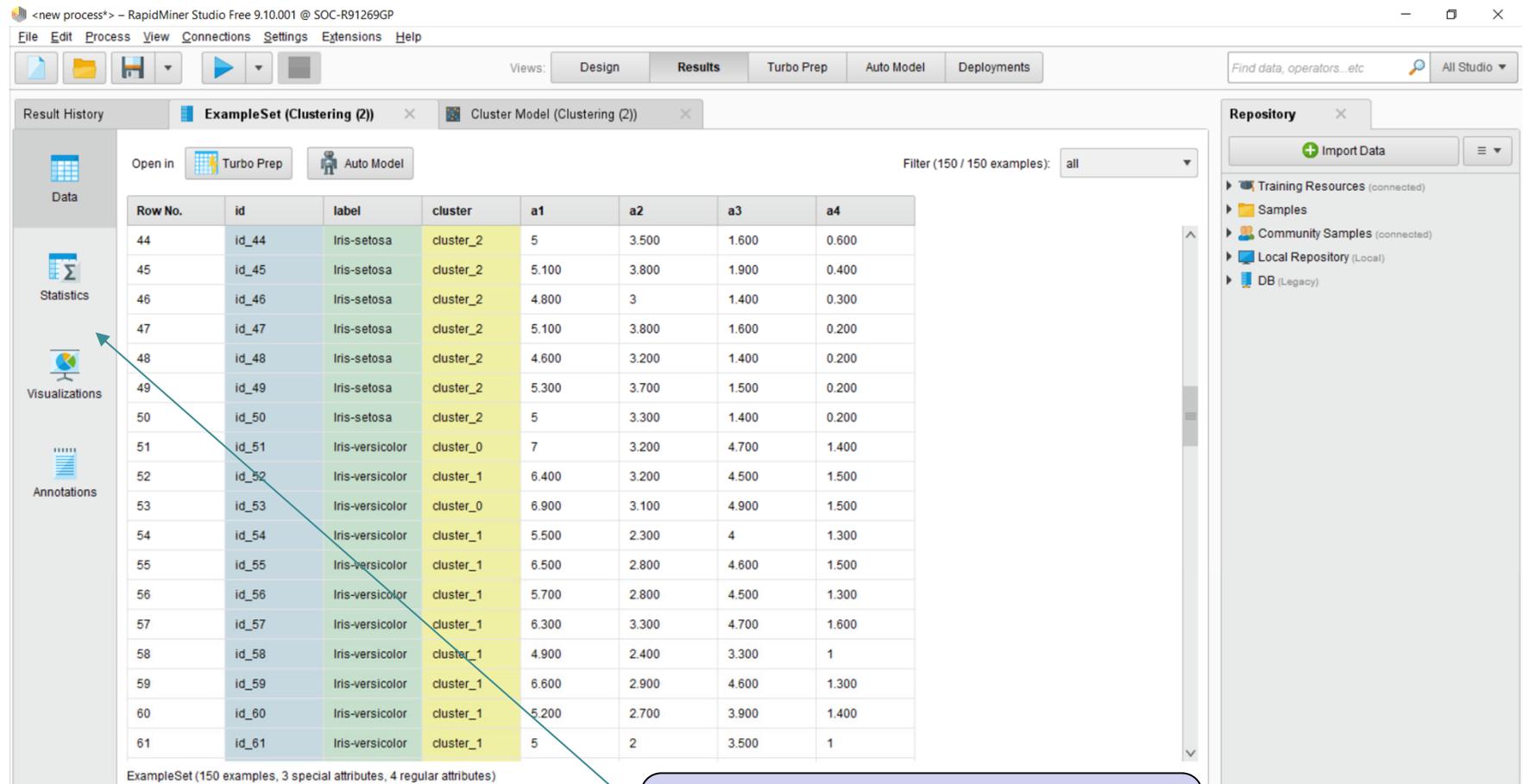
28°C 16:59 ENG 29-Dec-21

# Example 1: Clustering



Find 'k-means' in operators library/repository.  
Connect 'retrieve' output terminal to that.  
In this canvas data flow is always →  
Connect the clustering outputs to res terminals  
Play/ run it  
Keep the clustering parameters like this for the time being

# Example 1: Clustering Results

A screenshot of the RapidMiner Studio Free interface. The main window displays a table titled 'ExampleSet (Clustering (2))' containing 150 examples. The columns are Row No., id, label, cluster, a1, a2, a3, and a4. The 'cluster' column shows two distinct clusters: 'cluster\_2' (for Iris-setosa) and 'cluster\_1' (for Iris-versicolor). A blue arrow points from the 'Visualizations' icon in the left sidebar to the 'cluster' column header. The bottom status bar shows the text 'Clustering results, example set report Try them'.

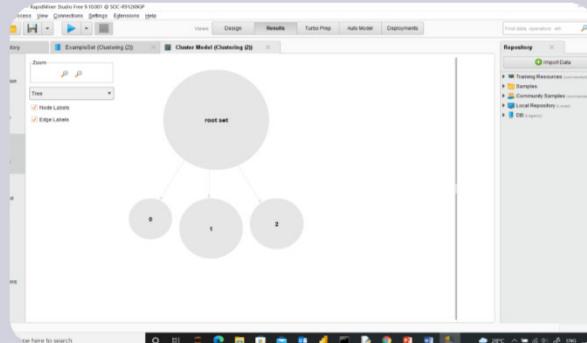
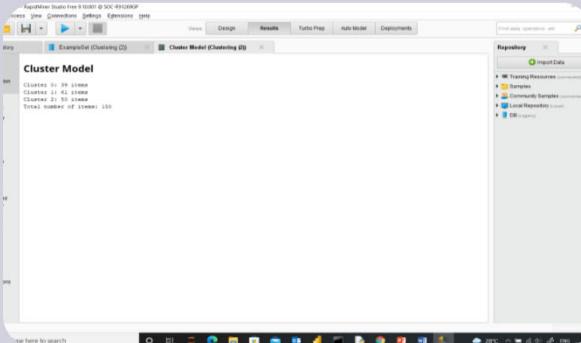
Row No.	id	label	cluster	a1	a2	a3	a4
44	id_44	Iris-setosa	cluster_2	5	3.500	1.600	0.600
45	id_45	Iris-setosa	cluster_2	5.100	3.800	1.900	0.400
46	id_46	Iris-setosa	cluster_2	4.800	3	1.400	0.300
47	id_47	Iris-setosa	cluster_2	5.100	3.800	1.600	0.200
48	id_48	Iris-setosa	cluster_2	4.600	3.200	1.400	0.200
49	id_49	Iris-setosa	cluster_2	5.300	3.700	1.500	0.200
50	id_50	Iris-setosa	cluster_2	5	3.300	1.400	0.200
51	id_51	Iris-versicolor	cluster_0	7	3.200	4.700	1.400
52	id_52	Iris-versicolor	cluster_1	6.400	3.200	4.500	1.500
53	id_53	Iris-versicolor	cluster_0	6.900	3.100	4.900	1.500
54	id_54	Iris-versicolor	cluster_1	5.500	2.300	4	1.300
55	id_55	Iris-versicolor	cluster_1	6.500	2.800	4.600	1.500
56	id_56	Iris-versicolor	cluster_1	5.700	2.800	4.500	1.300
57	id_57	Iris-versicolor	cluster_1	6.300	3.300	4.700	1.600
58	id_58	Iris-versicolor	cluster_1	4.900	2.400	3.300	1
59	id_59	Iris-versicolor	cluster_1	6.600	2.900	4.600	1.300
60	id_60	Iris-versicolor	cluster_1	5.200	2.700	3.900	1.400
61	id_61	Iris-versicolor	cluster_1	5	2	3.500	1

ExampleSet (150 examples, 3 special attributes, 4 regular attributes)

Clustering results, example set report  
Try them

# Example 1: Clustering Results

Clustering results, cluster model report



Attribute	cluster_0	cluster_1	cluster_2
a1	0.054	0.084	0.006
a2	3.077	2.741	0.416
a3	5.715	4.389	1.464
a4	2.054	1.434	0.244

Description:  
clustering  
overview

Graph:  
structure of  
clusters

Centroids  
Table: final  
coordination of  
centroids in this  
4d feature  
space

# Practice 3

- **Design a k-means clustering system**
  - Use Iris dataset
  - Use another optional dataset
  - Try different  $k=2, 4, \text{ and } 6$
- **Visualize the results**
- **Discuss the results**
- **Report the results**

# Improvement

# Example 1 with normalization

//Local Repository/processes/cluster1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

**Repository**

- Import Data

Training Resources (connected)

Samples

Community Samples (connected)

Local Repository (Local)

DB (Legacy)

**Process**

Process

Retrieve

Clustering (2)

Normalize

**Parameters**

Clustering (2) (k-Means)

add cluster attribute

add as label

remove unlabeled

K: 3

max runs: 10

determine good start values

measure types: BregmanD...

divergence: SquaredEu...

max optimization steps: 100

**Operators**

normali

Cleansing (3)

Normalization (3)

Normalize

De-Normalize

Scale by Weights

Modeling (1)

Time Series (1)

Transformation (1)

Normalize (Series)

No results were found.

Type here to search

**Parameters**

Normalize

create view

attribute filter type: all

invert selection

include special attributes

method: Z-transformation

**Help**

Filter Examples

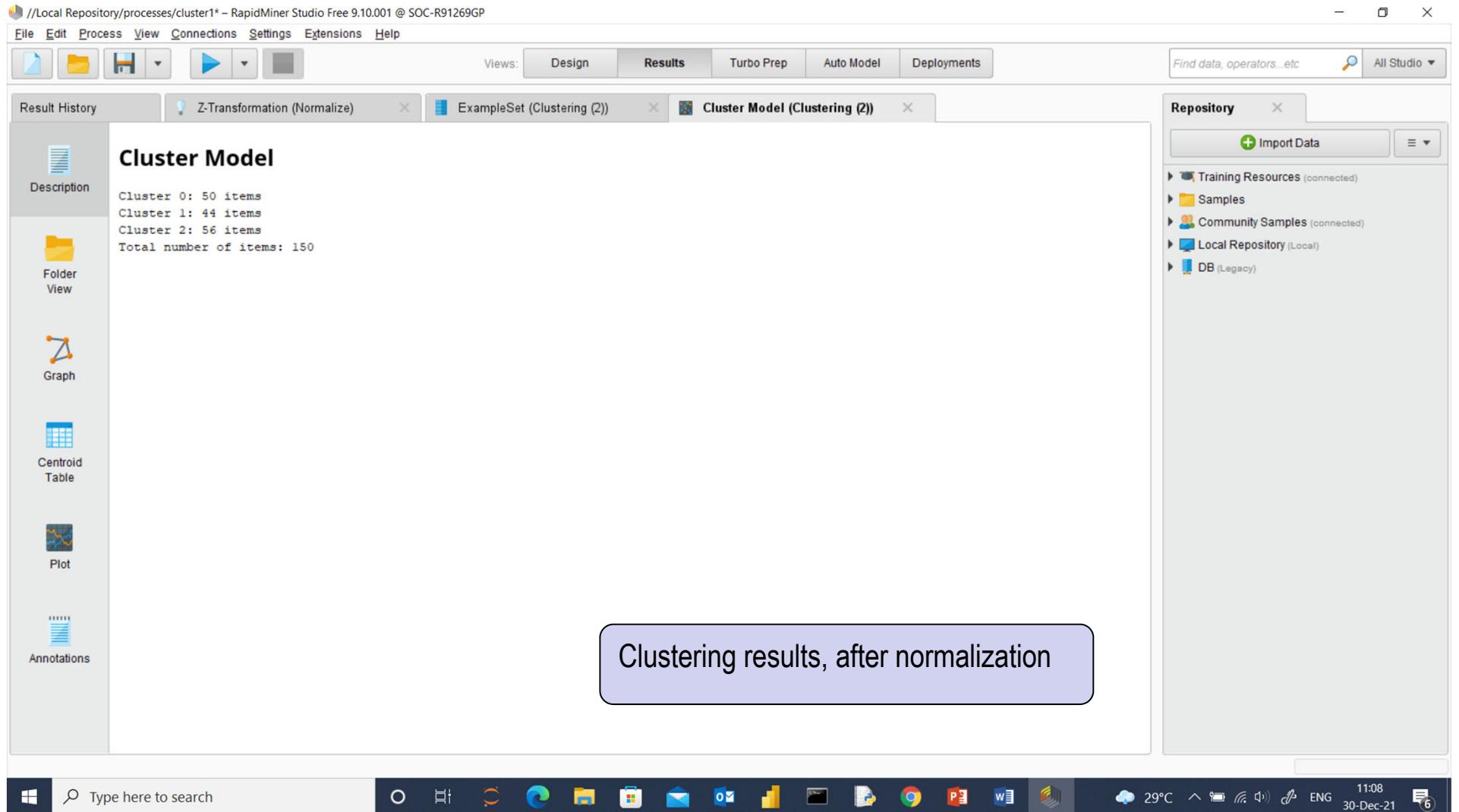
Tags Similar

30°C

23%

Find ‘Normalize’, put it between ‘retrieve’ and ‘clustering’  
Set the normalization method to Z-transformation  
This way you assume every variable’s distribution as Gaussian, and convert it to a standard Gaussian distribution with average=0, and variance=1.  
Check the clustering results

# Example 1 with normalization



The screenshot shows the RapidMiner Studio interface with the following details:

- Title Bar:** //Local Repository/processes/cluster1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP
- Menu Bar:** File Edit Process View Connections Settings Extensions Help
- Toolbar:** Includes icons for File, Folder, History, Play/Pause, and Stop.
- Views:** Design, Results, Turbo Prep, Auto Model, Deployments. The Results tab is selected.
- Search Bar:** Find data, operators...etc
- Result History:** Z-Transformation (Normalize), ExampleSet (Clustering (2)), Cluster Model (Clustering (2))
- Cluster Model Panel:**
  - Description:** Cluster 0: 50 items, Cluster 1: 44 items, Cluster 2: 56 items, Total number of items: 150.
  - Folder View:** Available options include Graph, Centroid Table, Plot, and Annotations.
- Repository Panel:** Training Resources (connected), Samples, Community Samples (connected), Local Repository (Local), DB (Legacy).
- Annotation:** A callout box highlights the text "Clustering results, after normalization".
- Taskbar:** Shows various application icons like File Explorer, Edge, Mail, etc.
- System Tray:** Displays battery level (29°C), signal strength, volume, ENG, date (30-Dec-21), and a notification icon with the number 6.

# Example 1, Improvement

//Local Repository/processes/cluster1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

**Repository**

- + Import Data
- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

**Process**

```

graph LR
    Retrieve[Retrieve] --> Normalize[Normalize]
    Normalize --> Clustering[Clustering 2]
    Clustering --> res1((res))
    Clustering --> res2((res))
    Clustering --> res3((res))
    Clustering --> res4((res))
  
```

**Operators**

- normali
- Cleansing (3)
  - Normalization (3)
    - Normalize
    - De-Normalize
    - Scale by Weights
- Modeling (1)
  - Time Series (1)
    - Transformation (1)
      - Normalize (Series)

No results were found.

**Parameters**

Clustering (2) (k-Means)

clu cluster attribute

k: 3

max runs: 30

determine good start values: checked

measure types: BregmanD...

divergence: SquaredEu...

max optimization steps: 300

use local random seed: unchecked

Hide advanced parameters

Change compatibility (9.10.001): checked

**Help**

k-Means

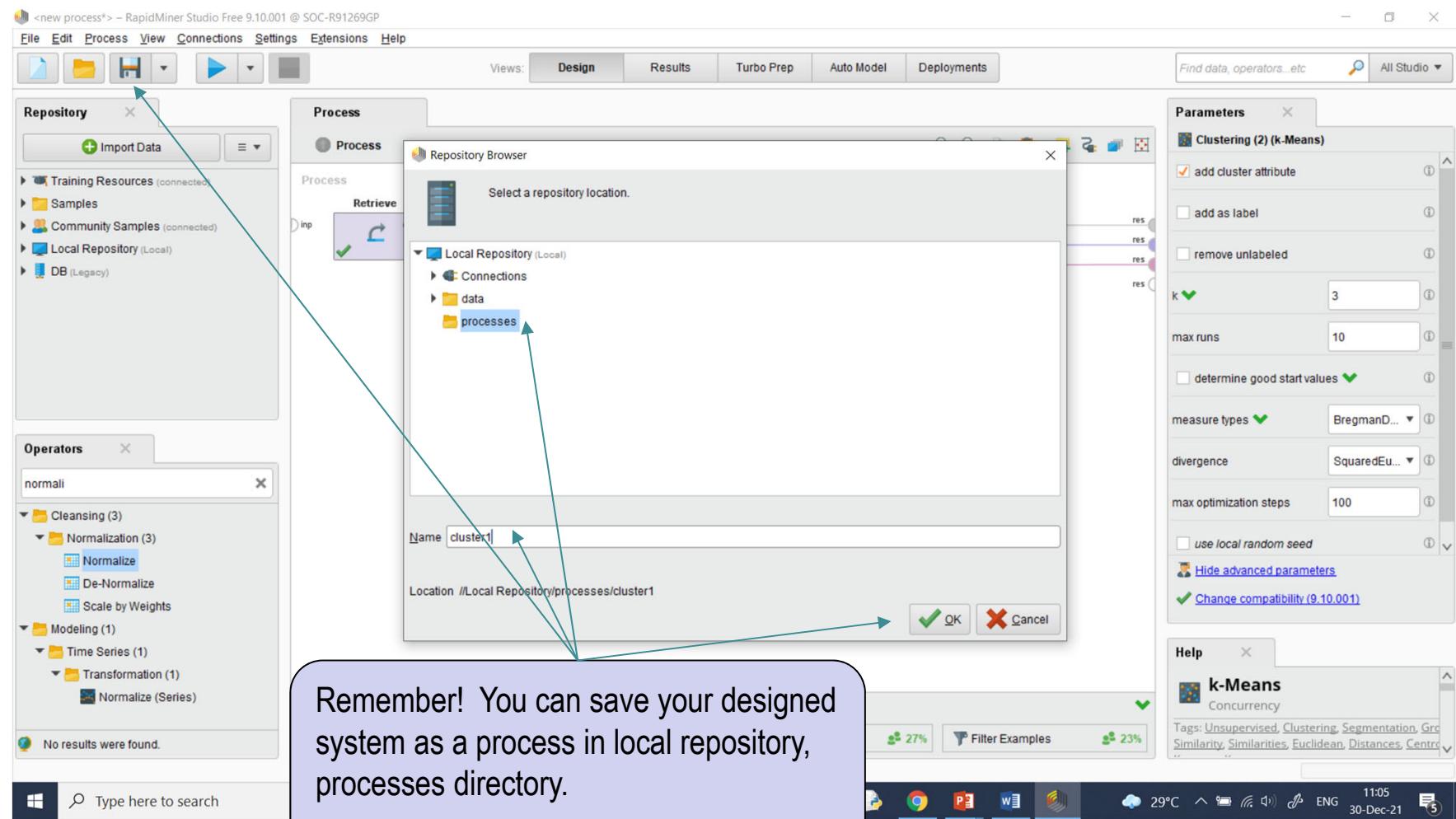
Concurrency

Tags: Unsupervised, Clustering, Segmentation, Grid, Similarity, Similarities, Euclidean, Distances, Centroids

Type here to search

29°C 11:07 ENG 30-Dec-21

# Save your System



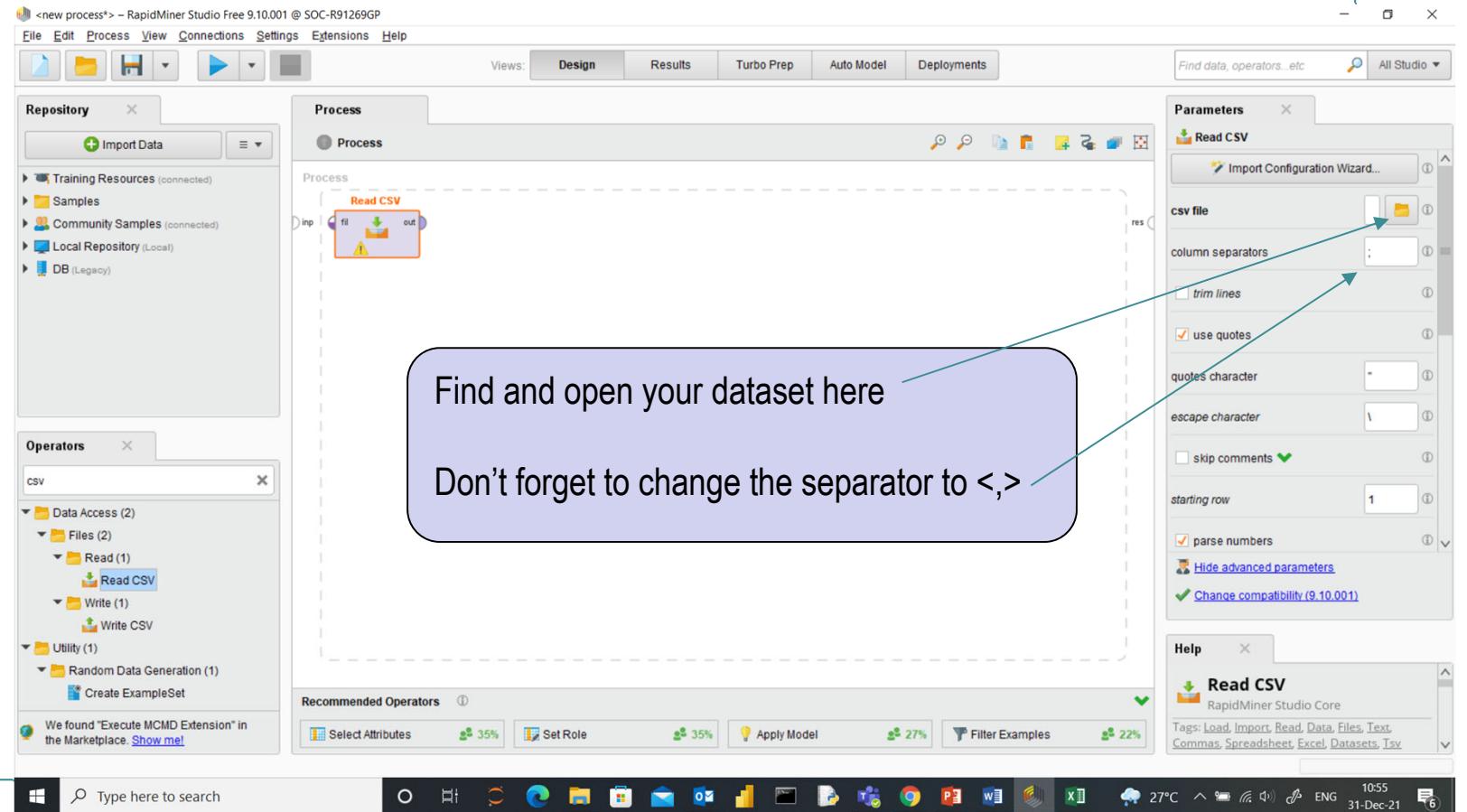
# Practice 4: Analyze the Titanic Disaster

- You can find `titanicpp.csv` data file in your `examples` directory
  - This is a comma separated values dataset contains some information about the Titanic passengers and crews
  - We have 2201

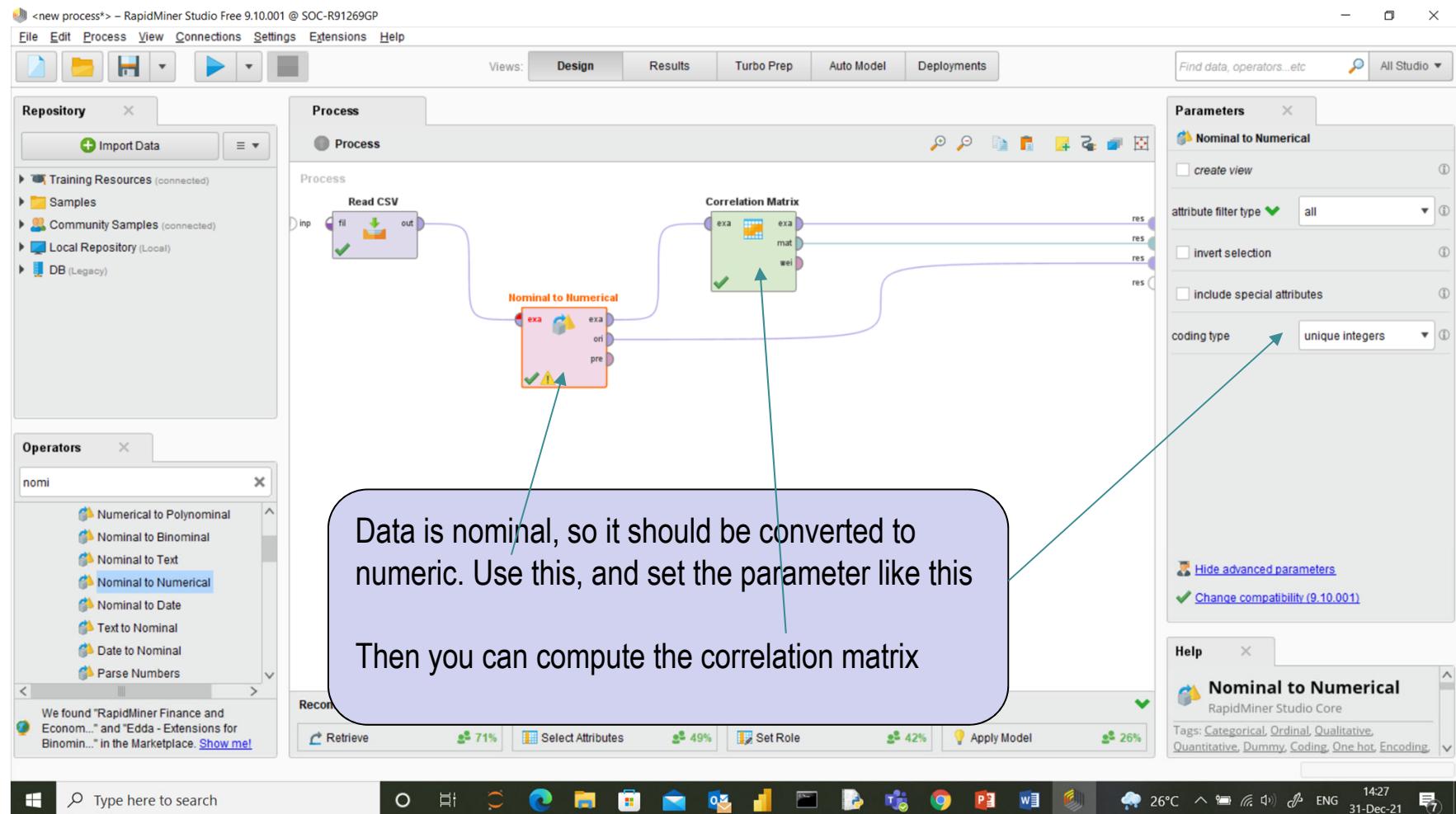
# Data samples of Crews and Passengers, and 4 Variables

# Practice 4 : Analyze the Titanic Disaster

- How to read the dataset? Find and use **Read CSV** operator.



# Practice 4 : Analyze the Titanic Disaster



Data is nominal, so it should be converted to numeric. Use this, and set the parameter like this

Then you can compute the correlation matrix

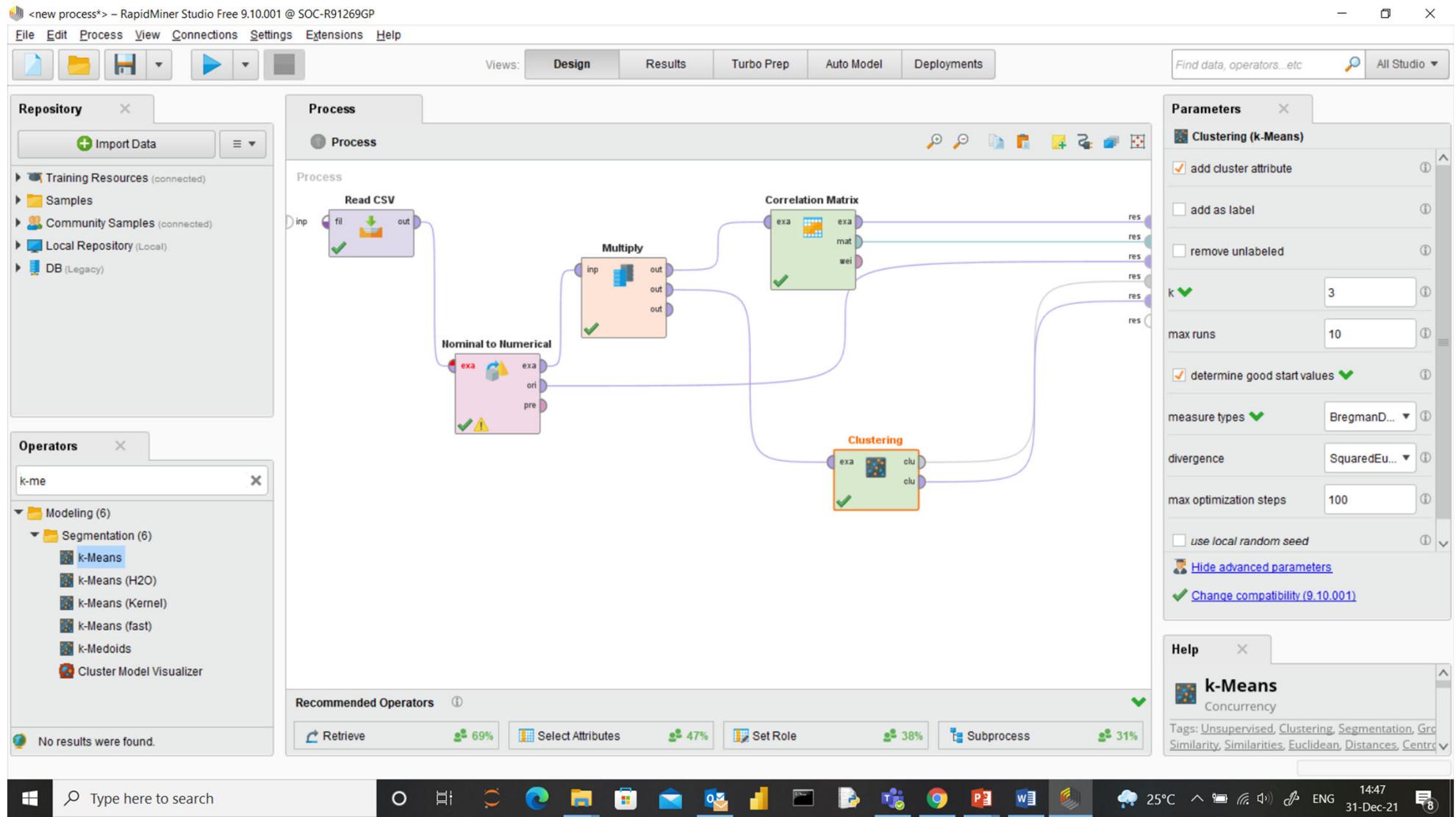
**Help**

**Nominal to Numerical**  
RapidMiner Studio Core  
Tags: Categorical, Ordinal, Qualitative, Quantitative, Dummy, Coding, One hot, Encoding

## Practice 4 : Analyze the Titanic Disaster

- Try to find any meaningful correlation between variables.
- Try variables statistics to see if you can learn anything from that report too.
- Next, you need to apply k-means clustering. Design something like the next page system.
- **Multiply** operator gives you more than one copy of a signal, here the output of **nominal-to-numeric** convertor.
- Try different  $k=2,3,4$ , then say what is the major point for each selected  $k$  in this clustering task, regarding the clustering outcomes.

# Practice 4 : Analyze the Titanic Disaster

A screenshot of the RapidMiner Studio Free 9.10.001 interface. The window title is "<new process\*> – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP". The menu bar includes File, Edit, Process, View, Connections, Settings, Extensions, and Help. The top right features a search bar "Find data, operators...etc" and a dropdown "All Studio". The main area is divided into several panels:

- Repository**: Shows Training Resources (connected), Samples, Community Samples (connected), Local Repository (Local), and DB (Legacy).
- Process**: Displays a flow diagram with nodes: Read CSV, Nominal to Numerical, Multiply, Correlation Matrix, and Clustering.
- Parameters**: Configurations for the Clustering (k-Means) operator, including "add cluster attribute" checked, k=3, max runs=10, divergence set to SquaredEu..., and other settings like "determine good start values" and "Change compatibility (9.10.001)" checked.
- Operators**: A tree view showing "k-me" under "Modeling (6)", which further branches into "Segmentation (6)" with "k-Means" selected. Other operators listed include k-Means (H2O), k-Means (Kernel), k-Means (fast), k-Medoids, and Cluster Model Visualizer.
- Recommended Operators**: A list of recommended operators: Retrieve (69%), Select Attributes (47%), Set Role (38%), and Subprocess (31%).
- Help**: Information about the k-Means operator, including its concurrency and tags: Unsupervised, Clustering, Segmentation, Gc, Similarity, Euclidean, Distances, Centre.

The bottom of the screen shows the Windows taskbar with various pinned icons and system status information: 25°C, ENG, 14:47, 31-Dec-21, and a notification icon with the number 8.

# What is the Hierarchical Clustering Method

# Hierarchical Clustering

- In Machine Learning, hierarchical is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types:
  - Agglomerative: This is a "bottom-up" approach: each observation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy.
  - Divisive: This is a "top-down" approach: all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy.
- In general, the merges and splits are determined in a greedy manner.
- The results of hierarchical clustering are usually presented in a dendrogram.

# Hierarchical Clustering

- This is a relatively computationally heavy and time consuming algorithm.
- To decide which clusters should be combined (for agglomerative), or where a cluster should be split (for divisive), a measure of dissimilarity between sets of observations is required, using an appropriate metric and a linkage.
  - Metric: A measure of distance between pairs of observations
  - Linkage criterion: Specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

# Hierarchical Clustering: Metric

- The choice of metric will influence the shape of the clusters
- Some commonly used metrics for hierarchical clustering are:

Names	Formula
Euclidean distance	$\ a - b\ _2 = \sqrt{\sum_i (a_i - b_i)^2}$
Squared Euclidean distance	$\ a - b\ _2^2 = \sum_i (a_i - b_i)^2$
Manhattan (or city block ) distance	$\ a - b\ _1 = \sum_i  a_i - b_i $
Maximum distance (or Chebyshev distance)	$\ a - b\ _\infty = \max_i  a_i - b_i $
Mahalanobis distance	$\sqrt{(a - b)^\top S^{-1} (a - b)}$ where $S$ is the Covariance matrix

- For text or other non-numeric data, metrics such as the Hamming distance or Levenshtein distance are often used.

# Hierarchical Clustering: LC

- The linkage criterion determines the distance between sets of observations as a function of the pairwise distances between observations.
- Some commonly used linkage criteria between two sets of observations A and B are:

Names	Formula
Maximum or complete-linkage clustering	$\max \{ d(a, b) : a \in A, b \in B \}.$
Minimum or single-linkage clustering	$\min \{ d(a, b) : a \in A, b \in B \}.$
Unweighted average linkage clustering (or UPGMA)	$\frac{1}{ A  \cdot  B } \sum_{a \in A} \sum_{b \in B} d(a, b).$
Weighted average linkage clustering (or WPGMA)	$d(i \cup j, k) = \frac{d(i, k) + d(j, k)}{2}.$
Centroid linkage clustering, or UPGMC	$\ c_s - c_t\ $ where $c_s$ and $c_t$ are the centroids of clusters $s$ and $t$ , respectively.
Minimum energy clustering	$\frac{2}{nm} \sum_{i,j=1}^{n,m} \ a_i - b_j\ _2 - \frac{1}{n^2} \sum_{i,j=1}^n \ a_i - a_j\ _2 - \frac{1}{m^2} \sum_{i,j=1}^m \ b_i - b_j\ _2$

# Basic Algorithm

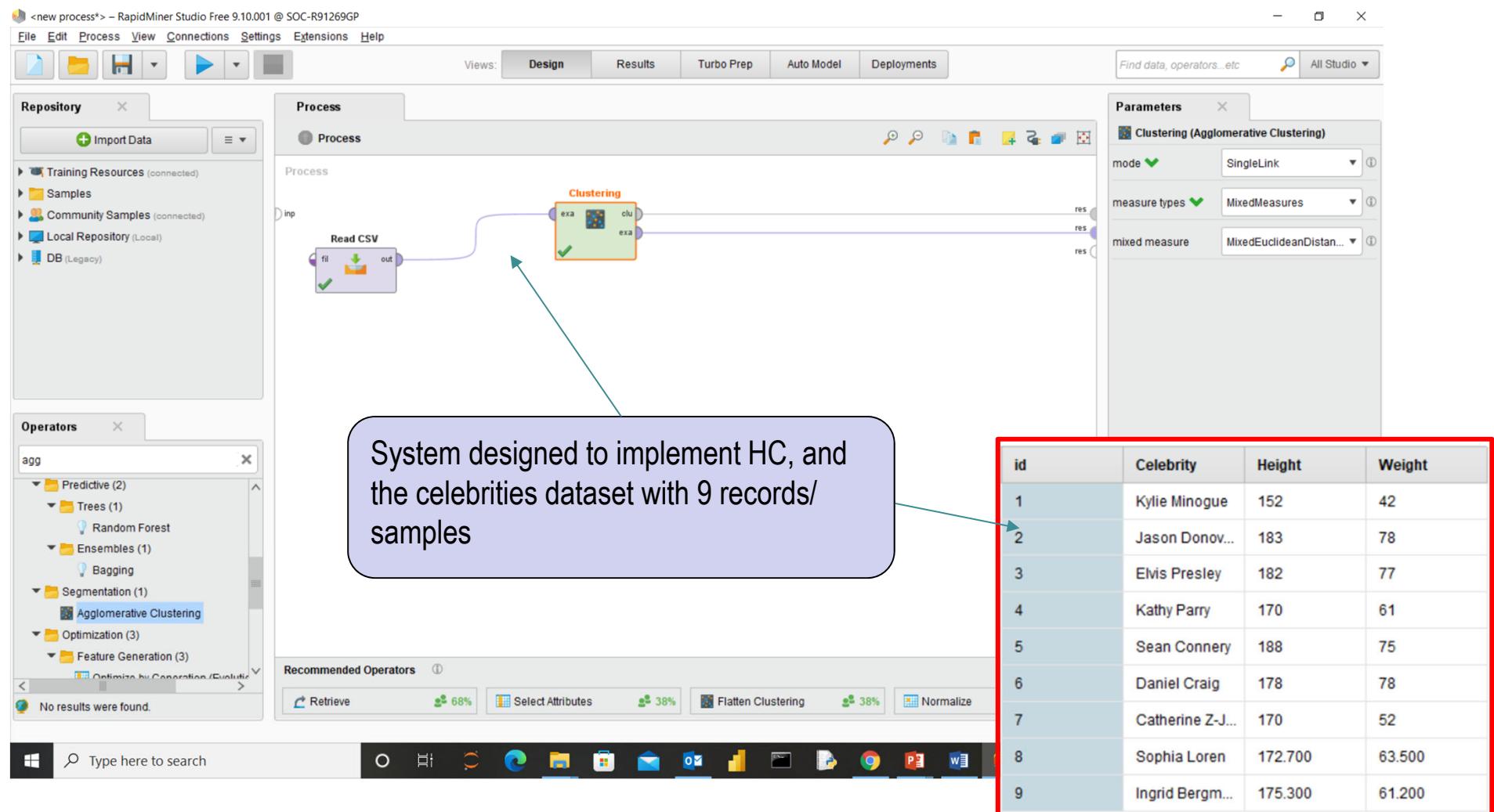
# Basic Algorithm



Seven animals, HC  
and agglomeration,  
and dendrogram

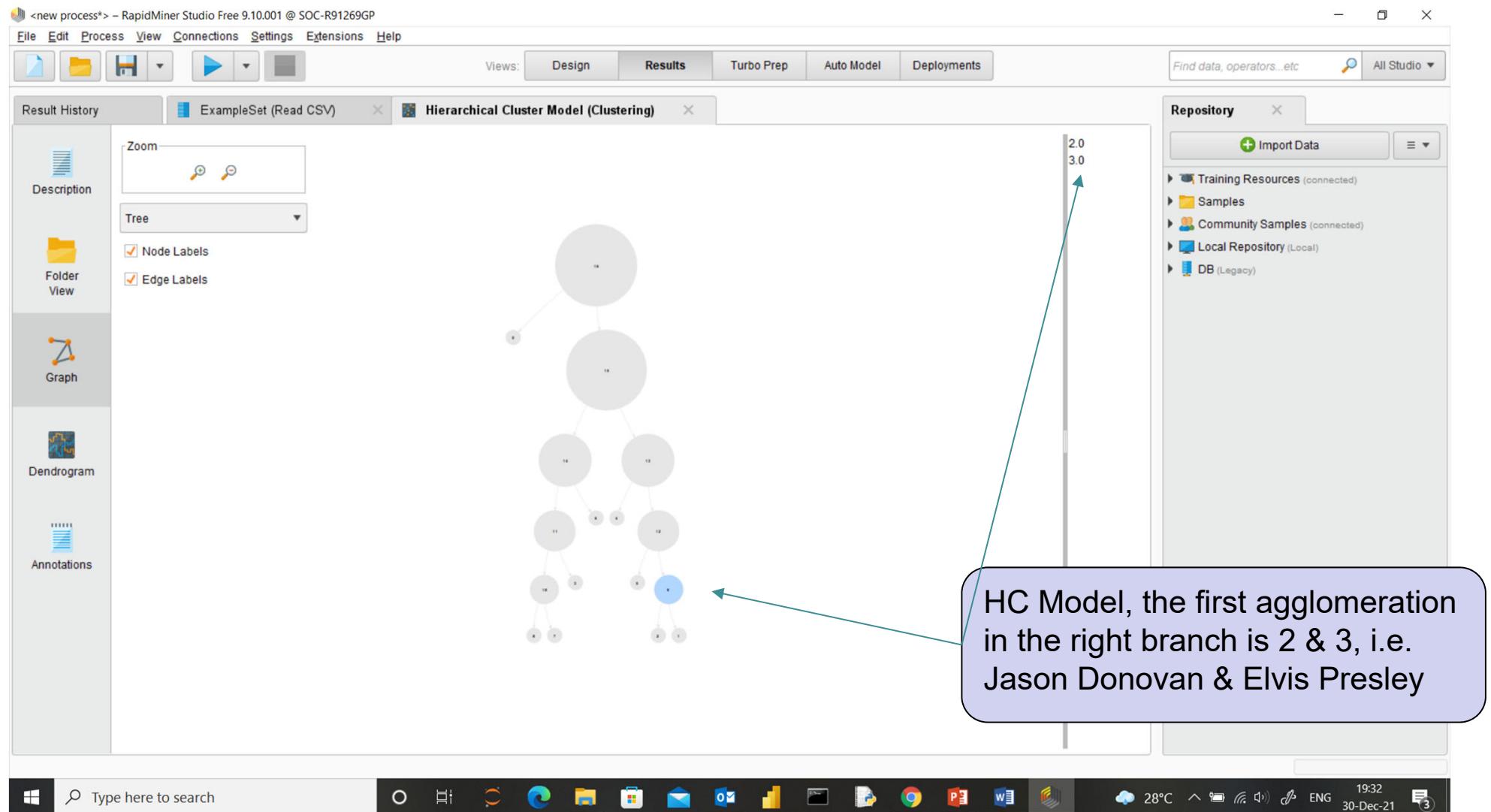
# Examples

# Example 1: Celebrities, Height, Weight

A screenshot of the RapidMiner Studio Free interface. The process tab shows a flow from a 'Read CSV' operator (fil) to a 'Clustering' operator (exa). The 'Clustering' operator has three output ports labeled 'exa', 'clu', and 'exa'. The 'Parameters' panel on the right is set for 'Agglomerative Clustering' with mode 'SingleLink', measure types 'MixedMeasures', and mixed measure 'MixedEuclideanDistance'. A callout box points to the 'Clustering' operator with the text: 'System designed to implement HC, and the celebrities dataset with 9 records/ samples'. To the right of the interface is a table of celebrity data:

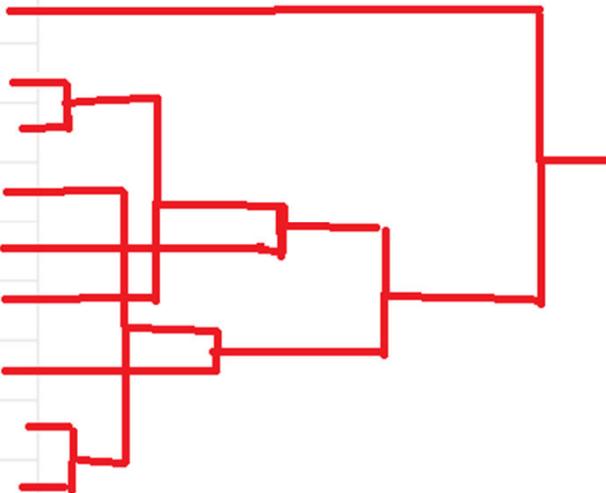
id	Celebrity	Height	Weight
1	Kylie Minogue	152	42
2	Jason Donov...	183	78
3	Elvis Presley	182	77
4	Kathy Parry	170	61
5	Sean Connery	188	75
6	Daniel Craig	178	78
7	Catherine Z-J...	170	52
8	Sophia Loren	172.700	63.500
9	Ingrid Bergm...	175.300	61.200

# Example 1: Celebrities, Height, Weight



# Example 1: Celebrities, Height, Weight

<b>id</b>	<b>Celebrity</b>	<b>Height</b>	<b>Weight</b>
1	Kylie Minogue	152	42
2	Jason Donov...	183	78
3	Elvis Presley	182	77
4	Kathy Parry	170	61
5	Sean Connery	188	75
6	Daniel Craig	178	78
7	Catherine Z-J...	170	52
8	Sophia Loren	172.700	63.500
9	Ingrid Bergm...	175.300	61.200



Dendrogram generated by HC  
Closest together, furthest to the  
other group.

# Mini Project

# Mini Project: Customer H Segmentation

- Dataset **Mall\_customers\_small.csv** contains data of 30 customers of a shopping mall. Again you can find it on [Examples](#) directory.
- Design a system to apply H clustering on that dataset.
- One of the variables is nominal, convert it to numeric if you think it's necessary.
- Find out how HC cluster the items in the first layers of clustering.
- Try different parameters and see the differences.

# Mini Project: Customer H Segmentation

- **Regarding the first agglomerations, which feature seems to be the least relevant/important one?**
- **Try to normalize the data, then re-answer the question above. Can you see any difference? Why?**

# Function Estimation Using Linear Regression

# Regression

- Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable and one or more independent variables.
- The two basic types of regression are simple linear regression and multiple linear regression, although there are non-linear regression methods for more complicated data and analysis.



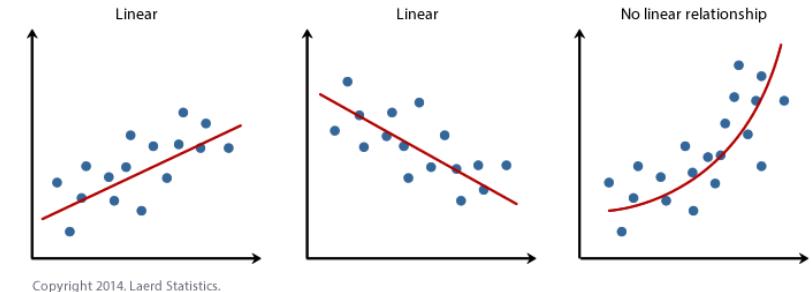
$$Y = f(x) = ax + b$$

# Regression

- Nonlinear regression is a form of regression analysis in which data is fit to a model and then expressed as a non-linear (curved) mathematical function.
- The goal of the model is to make the sum of the squares as small as possible.
- Training in regression needs a labeled dataset (inputs/output pairs), and an optimization algorithm.
- That algorithm either uses an iterative MSE minimization scheme, or an inverse matrix scheme.

$$y = ax_1^2 + bx_2^2 + cx_1 + dx_2 + e$$

$$y = ax_1^2 + cx_1 + e$$



# Linear Regression Example

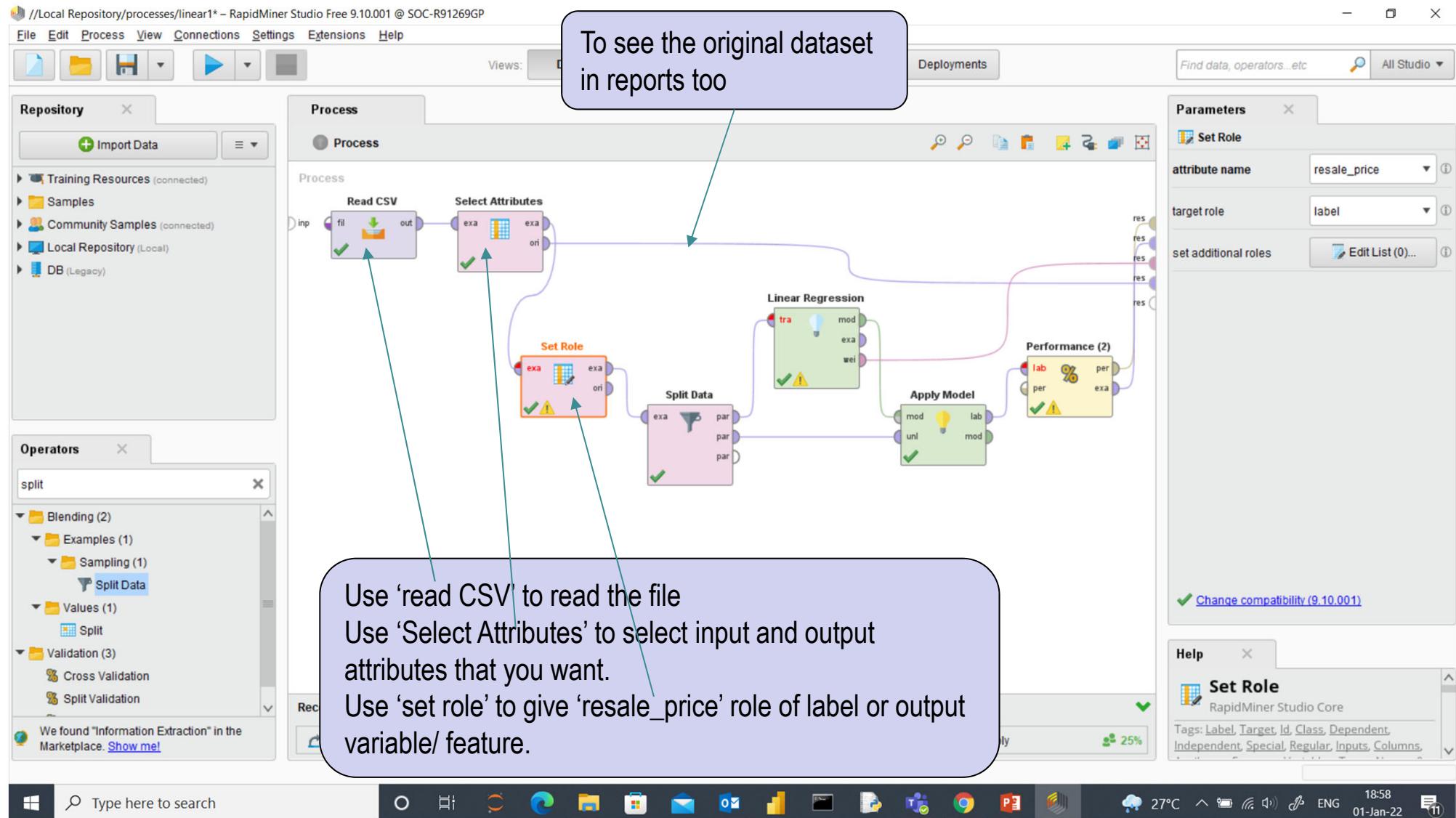
*Help your local real estate agent*

## Scenario:

- Data set: [resale-sample.csv](#), contains 2000 samples of HDBs data, Singapore, there are many features, and an output called, resale-price
- We'd like to use linear regression to estimate resale-price (S\$) from flat area (Sq M)
- We will expand this example to involving more features.



# HDBs price estimation using LR



# HDBs price estimation using LR

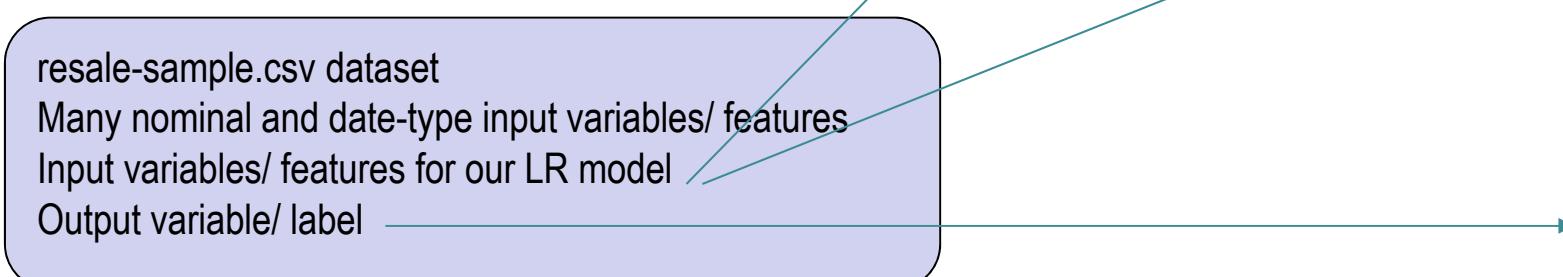
A	B	C	D	E	F	G	H	I	J	K	
1	id	month	town	flat_type	block	street_name	storey_range	floor_area_sqm	flat_model	lease_commence_date	resale_price
2	13547	2012-09	CHOA CHU	4 ROOM	119	TECK WHYE LA	04 TO 06	104	Model A	1989	400000
3	7200	2012-06	BUKIT MEF	3 ROOM	22	HAVELOCK RD	04 TO 06	64	Standard	1969	404000
4	78120	2016-05	JURONG W	EXECUTIVI	906	JURONG WEST	07 TO 09	141	Apartment	1989	422000
5	33138	2013-10	JURONG W	3 ROOM	510	JURONG WEST	13 TO 15	74	Model A	1984	375000
6	57905	2015-04	JURONG E/4	ROOM	232	JURONG EAST	07 TO 09	95	New Generatio	1982	385000
7	23096	2013-03	JURONG E/5	ROOM	284	TOH GUAN RD	07 TO 09	120	Improved	1998	655000
8	87705	2016-11	ANG MO KI	4 ROOM	326	ANG MO KIO A	10 TO 12	92	New Generatio	1977	590000
9	95211	2017-04	CENTRAL A	3 ROOM	668	CHANDER RD	10 TO 12	75	Model A	1984	375000
10	86039	2016-10	ANG MO KI	4 ROOM	428	ANG MO KIO A	10 TO 12	92	New Generatio	1978	490000
11	20521	2013-01	GEYLANG	4 ROOM	2	HAIG RD	04 TO 06	92	New Generatio	1976	543000
12	25891	2013-05	BUKIT PAN	4 ROOM	480	SEGAR RD	07 TO 09	94	Premium Apart	2002	435000
13	26437	2013-05	MARINE PA	3 ROOM	33	MARINE CRES	07 TO 09	65	Improved	1975	465000
14	38729	2014-03	HOUGANG	4 ROOM	695	HOUGANG ST	10 TO 12	104	Model A	1987	448000
15	85543	2016-09	SENGKANG	5 ROOM	412B	FERNVALE LIN	04 TO 06	114	Premium Apart	2004	460000
16	49662	2014-11	BEDOK	4 ROOM	103	BEDOK RESERV	04 TO 06	93	New Generatio	1985	405000
17	100216	2017-06	WOODLAN	EXECUTIVI	361	WOODLANDS	07 TO 09	145	Apartment	1996	638888
18	48223	2014-10	BUKIT BAT	4 ROOM	338	BT BATOK ST	301 TO 03	84	Simplified	1986	353000
19	27999	2013-06	SENGKANG	4 ROOM	319A	ANCHORVALE	01 TO 03	90	Model A	2002	454000
20	95100	2017-04	BUKIT MER	4 ROOM	5	DELTA AVE	04 TO 06	92	New Generatio	1985	638000

resale-sample.csv dataset

Many nominal and date-type input variables/ features

Input variables/ features for our LR model

Output variable/ label



# HDBs price estimation using LR

//Local Repository/processes/linear1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators... etc All Studio

**Repository**

- Import Data
- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

**Process**

```

graph LR
    ReadCSV[Read CSV] --> SelectAttributes[Select Attributes]
    SelectAttributes --> SetRole[Set Role]
    SetRole --> SplitData[Split Data]
    SplitData --> LinearRegression[Linear Regression]
    LinearRegression --> ApplyModel[Apply Model]
    ApplyModel --> Performance[Performance (2)]
    
```

**Operators**

- split
- Blending (2)
  - Examples (1)
  - Sampling (1)
    - Split Data
  - Values (1)
  - Split
- Validation (3)
  - Cross Validation
  - Split Validation

**Parameters**

Set Role

attribute name: resale\_price

target role: label

set additional roles: Edit List (0)

Change compatibility (9.10.001)

**Help**

**Set Role**  
RapidMiner Studio Core

Tags: Label, Target, Id, Class, Dependent, Independent, Special, Regular, Inputs, Columns

Type here to search

27°C 18:58 01-Jan-22

We need both training and testing/ validation data (sub) sets. Use 'split data' to separate them. E.g., 80% for training and 20% for testing (0.8 and 0.2). Then you will have them on the top to bottom output 'par' terminals. Set 'sampling type' to 'automatic'.

# HDBs price estimation using LR

//Local Repository/processes/linear1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators... etc All Studio

**Repository**

- Import Data
- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

**Process**

Process

```

graph LR
    ReadCSV[Read CSV] --> SelectAttributes[Select Attributes]
    SelectAttributes --> SetRole[Set Role]
    SetRole --> SplitData[Split Data]
    SplitData --> LinearRegression[Linear Regression]
    LinearRegression --> ApplyModel[Apply Model]
    ApplyModel --> Performance[Performance (2)]
    
```

**Operators**

split

- Blending (2)
  - Examples (1)
  - Sampling (1)
    - Split Data
  - Values (1)
    - Split
- Validation (3)
  - Cross Validation
  - Split Validation

We found "Information Extraction" in the Marketplace. [Show me!](#)

**Parameters**

Set Role

attribute name: resale\_price

target role: label

set additional roles: Edit List (0)

Change compatibility (9.10.001)

**Help**

Set Role

RapidMiner Studio Core

Tags: Label, Target, Id, Class, Dependent, Independent, Special, Regular, Inputs, Columns

Type here to search

27°C 18:58 01-Jan-22

# HDBs price estimation using LR

//Local Repository/processes/linear1\* – RapidMiner Studio Free 9.10.001 @ SOC-R91269GP

File Edit Process View Connections Settings Extensions Help

Views: Design Results Turbo Prep Auto Model Deployments

Find data, operators...etc All Studio

**Repository**

- + Import Data
- Training Resources (connected)
- Samples
- Community Samples (connected)
- Local Repository (Local)
- DB (Legacy)

**Process**

```

graph LR
    Read[Read CSV] --> Select[Select Attributes]
    Select --> Set[Set Role]
    Set --> Split[Split Data]
    Split --> Linear[Linear Regression]
    Linear --> Apply[Apply Model]
    Apply --> Performance[Performance]
    
```

**Operators**

- performa
- Validation (20)
  - Performance (18)
    - Performance (Classification)
    - Performance (Binomial Classification)
    - Performance (Regression)**
    - Performance (Costs)

We found "Model Management" in the Marketplace. [Show me!](#)

**Parameters**

**Performance (Performance (Regression))**

main criterion first

root mean squared error

absolute error

relative error

relative error lenient

relative error strict

normalized absolute error

root relative squared error

squared error

correlation

[Hide advanced parameters](#)

**Help**

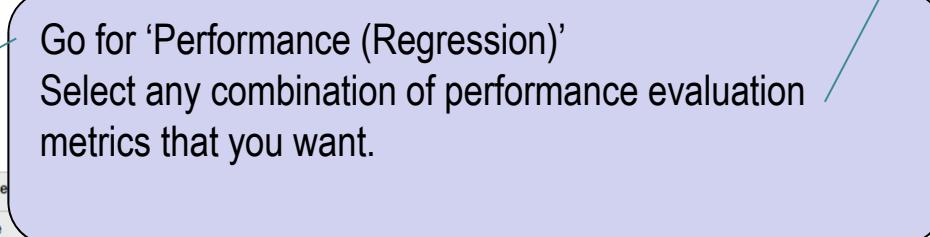
**Performance (Regression)**

RapidMiner Studio Core

Tags: RMSE, Errors, Absolute, Relative, Squared, Predictive

24°C ENG 09:21 02-Jan-22

Type here to search



# HDBs price estimation using LR, Results

The screenshot shows the KNIME Analytics Platform interface with two main windows open, both showing results for a Linear Regression model applied to HDB price estimation.

**Top Window (Results View):**

- Views:** Design, Results, Turbo Prep, Auto Model, Deployments
- Nodes:** ExampleSet (Read CSV), AttributeWeights (Linear Regression), ExampleSet (Apply Model), PerformanceVector (Performance)
- Performance Criterion:** root\_mean\_squared\_error
- Performance Value:** root\_mean\_squared\_error: 96877.614 +/- 0.000

**Bottom Window (Data View):**

- Views:** Design
- Nodes:** ExampleSet (Read CSV), AttributeWeights (Linear Regression), ExampleSet (Apply Model)
- Data Table:**

Row No.	resale_price	prediction(r...)	floor_area_...
1	454000	427301.459	90
2	350000	405496.547	84
3	390000	405496.547	84
4	390000	361886.722	72
5	346000	420033.155	88
6	270000	332813.505	64
7	415000	499984.500	110
8	399000	503618.653	111
9	277000	347350.113	68
10	660000	507252.805	112
11	435000	423667.307	89
12	540000	427301.459	90
- Buttons:** Open in Turbo Prep, Auto Model

## Practice 5

- Design a linear regressor to estimate the HDB prices.
- Use the **floor-area-sqm** as your input variable and **resale-price** as the label (output)
- Visualize the results and analyze them
- Add **lease-commence-date** as another input variable
- Compare the results with the former setting, can we make it better?

# Program Evaluation

# THE END ...

