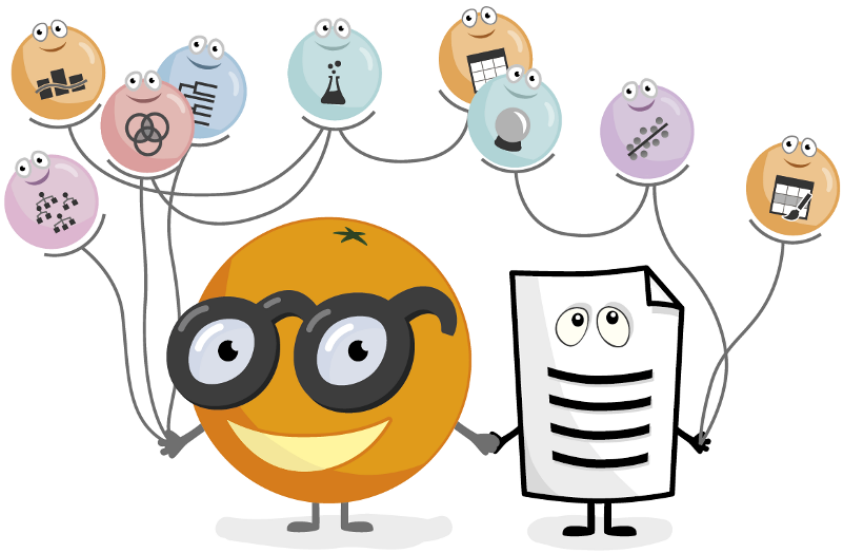
Orange: A Visual Programming Tool for Machine Learning and Data Analytics

A Guide to Mastering Orange for Research

First Edition



Olarik Surinta

Orange: A Visual Programming Tool for Machine Learning and Data Analytics

A Guide to Mastering Orange for Research

First Edition

Olarik Surinta
Department of Information Technology
Faculty of Informatics, Mahasarakham University
Maha Sarakham, Thailand

Orange: เครื่องมือสำหรับการโปรแกรมแบบวิชวล สำหรับการเรียนรู้เครื่องจักรและการวิเคราะห์ข้อมูล

คู่มือการเรียนรู้ Orange สำหรับการทำวิจัย

โอฬาริก สุรินต๊ะ

สาขาวิชาเทคโนโลยีสารสนเทศ
คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม
มหาสารคาม ประเทศไทย

คำนำ

ด้วยความก้าวหน้าทางเทคโนโลยีสารสนเทศ จึงทำให้มีข้อมูลเผยแพร่ผ่านทางอินเทอร์เน็ตอย่างรวดเร็ว เช่น ข้อมูลที่โพสต์ในเฟซบุ๊ค ทวิสเตอร์ หรือตามเว็บไซต์พันทิป ไม่ว่าจะเป็นข้อมูลที่เป็นข้อความหรือรูปภาพ ทำให้ผู้คนบางกลุ่มต้องการที่จะเข้าถึงข้อมูลเหล่านั้น เพื่อที่จะนำมาวิเคราะห์ และใช้ข้อมูลเหล่านั้นให้เกิดประโยชน์ เช่น นำมาช่วยในการวางแผนการตลาด และการเจาะกลุ่มเป้าหมาย จึงทำให้เกิดศาสตร์ใหม่ขึ้นมาเพื่อช่วยในการเรียนรู้ และบริหารจัดการข้อมูล เรียกว่า วิทยาการข้อมูล (Data Science) ซึ่งในอดีตเราจะรู้จักกับคำว่า "เหมืองข้อมูล (Data Mining)" และ "การเรียนรู้เครื่องจักร (Machine Learning)" โดยทั้งสองแขนงมุ่งเน้นที่อัลกอริทึมที่จะนำมาใช้เพื่อการเรียนรู้และการพยากรณ์ข้อมูล แต่อยู่บนพื้นฐานของการคำนวณด้วยวิธีเดียวกัน หากพูดถึงวิทยาการข้อมูลเปรียบได้กับการนำข้อมูลที่มีอยู่มาผ่านกระบวนการวิเคราะห์ทางธุรกิจ เพื่อให้ธุรกิจเกิดความได้เปรียบในการแข่งขัน

ปัจจุบันมีเครื่องมือมากมายที่สามารถนำมาใช้ในการวิเคราะห์ข้อมูล โดยโปรแกรม Orange มีลักษณะการทำงานแบบ Visualization ทำให้ผู้ใช้งานสามารถเรียกใช้งานได้อย่างสะดวกและรวดเร็ว โดยไม่จำเป็นต้องเขียนโค้ด อีกทั้งยังเป็นโปรแกรมที่สามารถดาวน์โหลดและใช้งานได้ฟรี ผู้เขียนจึงมีวัตถุประสงค์เพื่อเขียนหนังสือคู่มือการใช้โปรแกรม Orange เพื่อให้ผู้อ่านสามารถที่จะอ่านและปฏิบัติตามขั้นตอนได้อย่างถูกต้อง

โอฬาริก สุรินต๊ะ

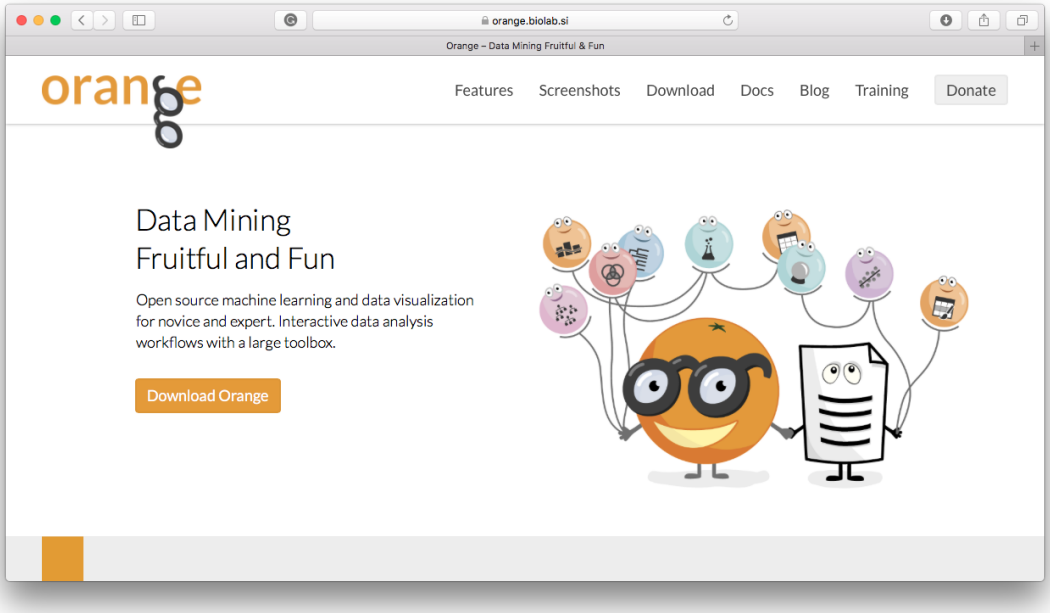
สารบัญ

สามารถดาวน์โหลดตัวอย่างโปรแกรม และหนังสือเล่มนี้ได้จากเว็บไซต์ github 

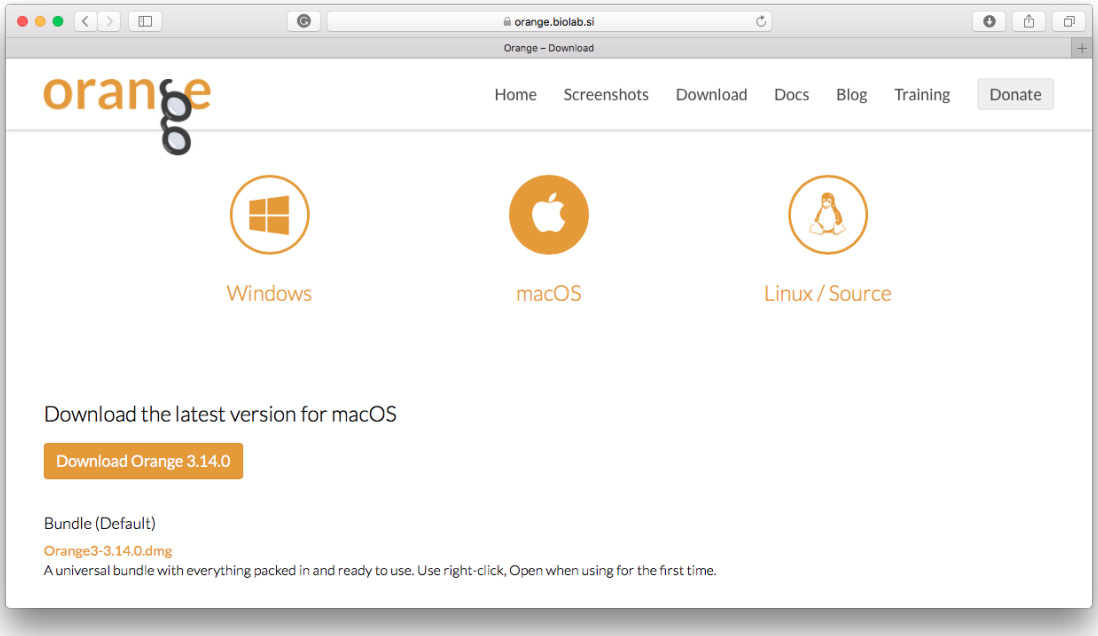
https://github.com/mrolarik/Orange-visual-programming

การติดตั้ง (Installation)

- สามารถดาวน์โหลด Orange ได้จากเว็บไซต์ https://orange.biolab.si

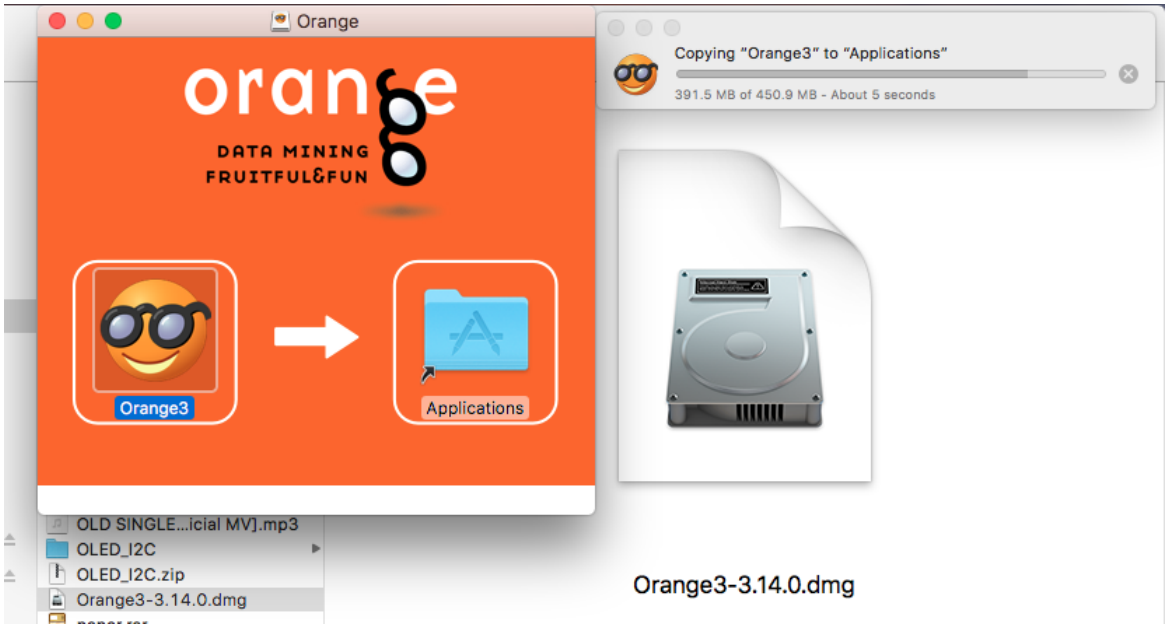


- คลิกที่เมนู **Download** หรือที่ปุ่ม **Download Orange**

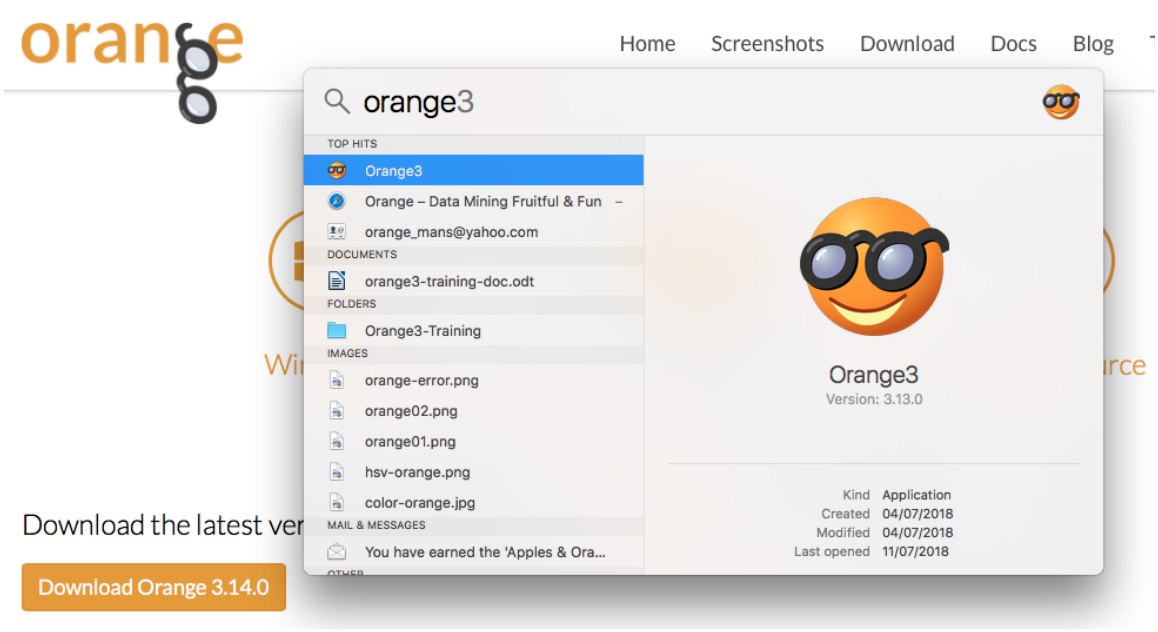


- สามารถเลือกดาวน์โหลดโปรแกรมได้ทั้งหมด 3 OS ประกอบด้วย Windows, macOS และ Linux / Source
- เมื่อเลือก OS เสร็จเรียบร้อย จากนั้นให้เลือกที่ **Download Orange** <version>

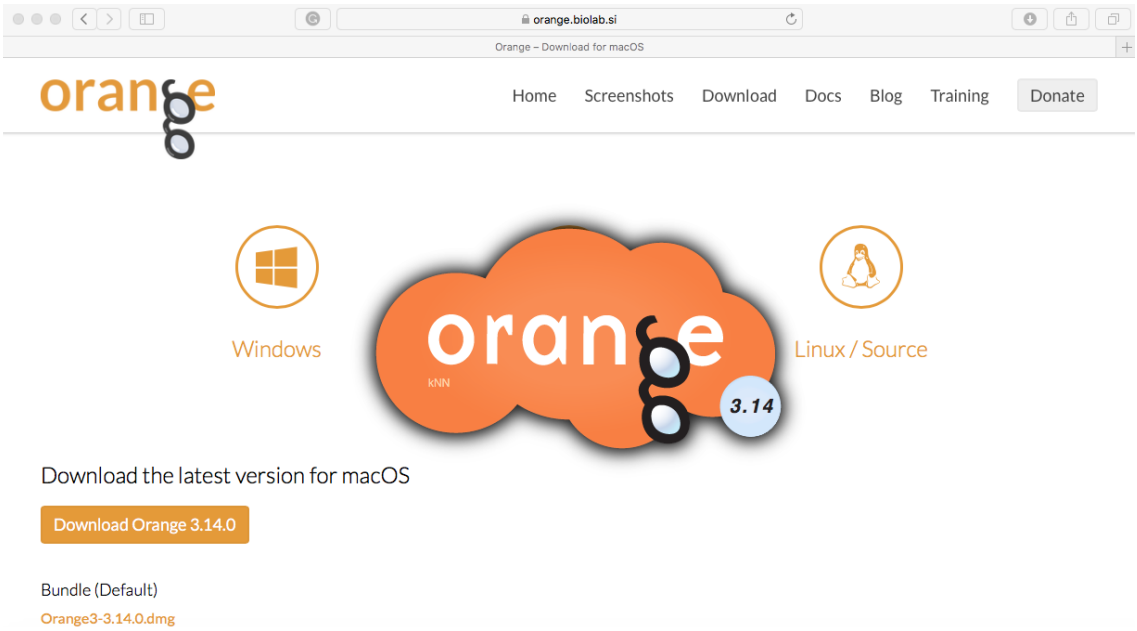
- จากนั้นไฟล์โปรแกรมที่ดาวน์โหลดจะถูกเก็บไว้ที่โฟลเดอร์ **Downloads**
- ตัวอย่างได้ดาวน์โหลดไฟล์ของ macOS โดยไฟล์ชื่อ **Orange3-3.140.dmg** โดยที่ 3.14.0 คือ version ของโปรแกรม orange ที่ได้ดาวน์โหลด



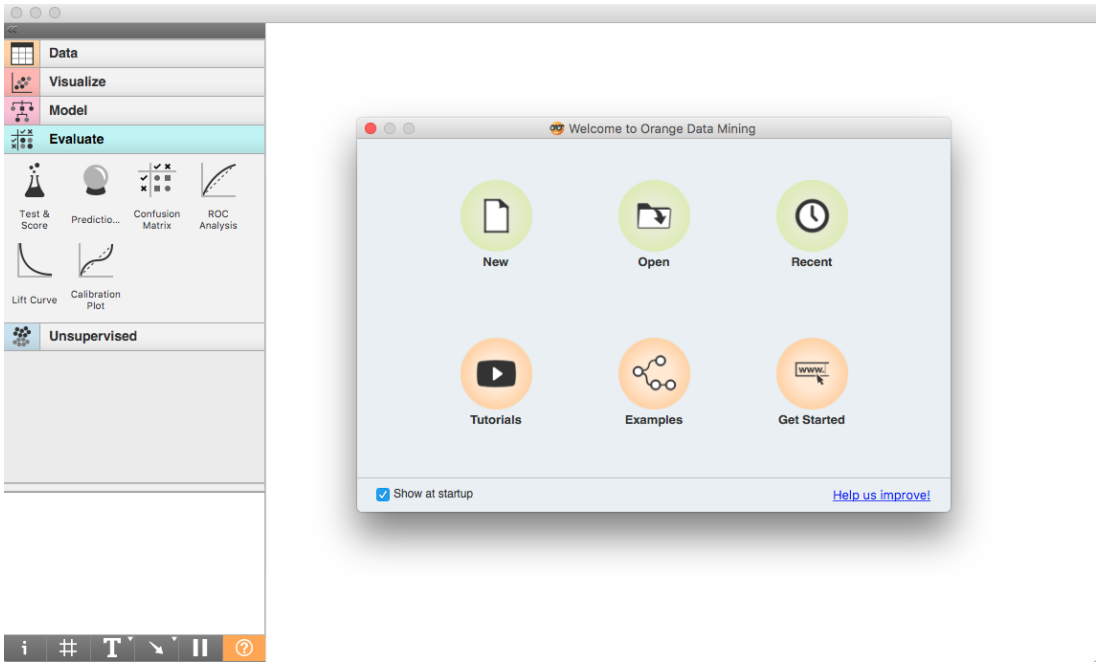
- จากนั้นให้ดับเบิ้ลคลิกที่ไฟล์ **Orange3-3.14.0.dmg** เพื่อติดตั้งโปรแกรม
- เมื่อติดตั้งเสร็จเรียบร้อยสามารถเปิดโปรแกรมขึ้นมาใช้งาน โดยโปรแกรมที่ติดตั้งเรียกว่า **Orange3**



- เมื่อคลิกเปิดโปรแกรมจะแสดงโลโก้ของโปรแกรม

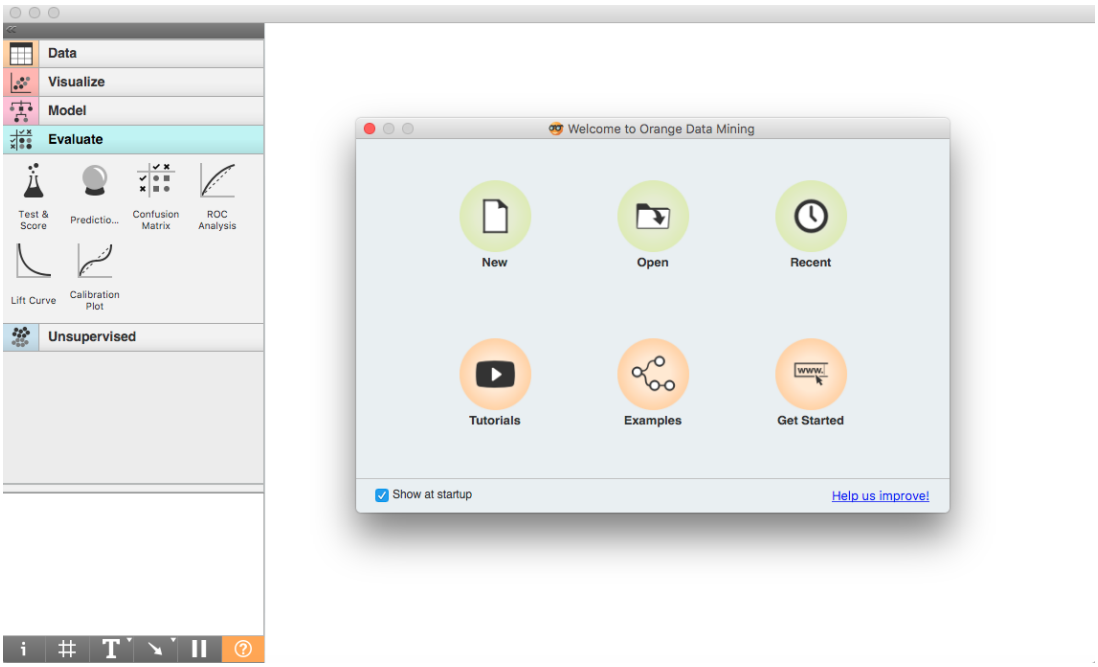


- เมื่อแสดงโลโก้ของโปรแกรมเสร็จเรียบร้อยจะเปิดโปรแกรม Orange เพื่อให้ใช้งาน



เริ่มต้นใช้งาน Orange (Getting Started with Orange)

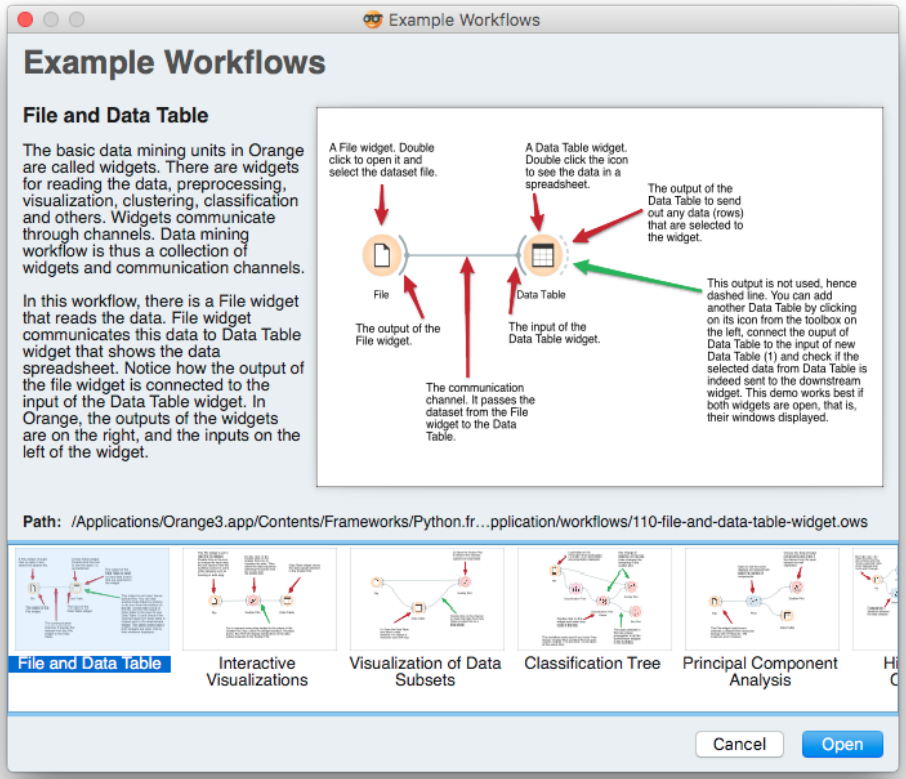
- เมื่อเรียกโปรแกรม Orange จะแสดงหน้าต่าง **Welcome to Orange Data Mining**

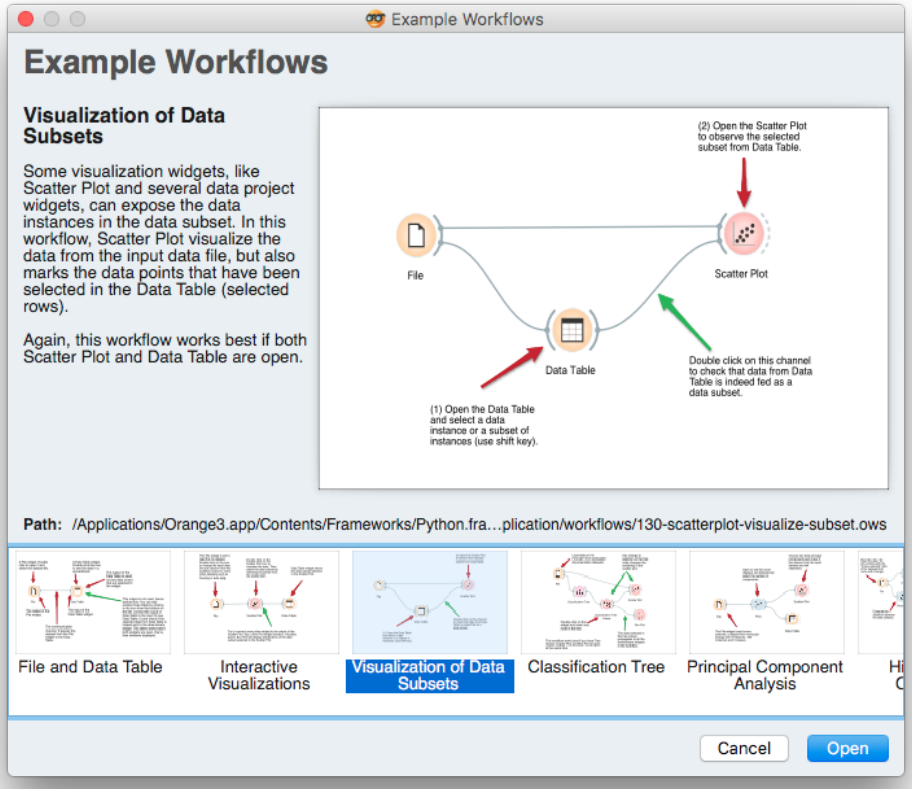


- จากหน้าต่าง Welcome to Orange Data Mining สามารถเลือกเมนูต่าง ๆ เช่น
  - สร้างไฟล์ใหม่ (New)
  - เปิดไฟล์เก่า (Open)
  - เปิดไฟล์ล่าสุด (Recent)
  - เปิดดูตัวอย่าง (Examples)

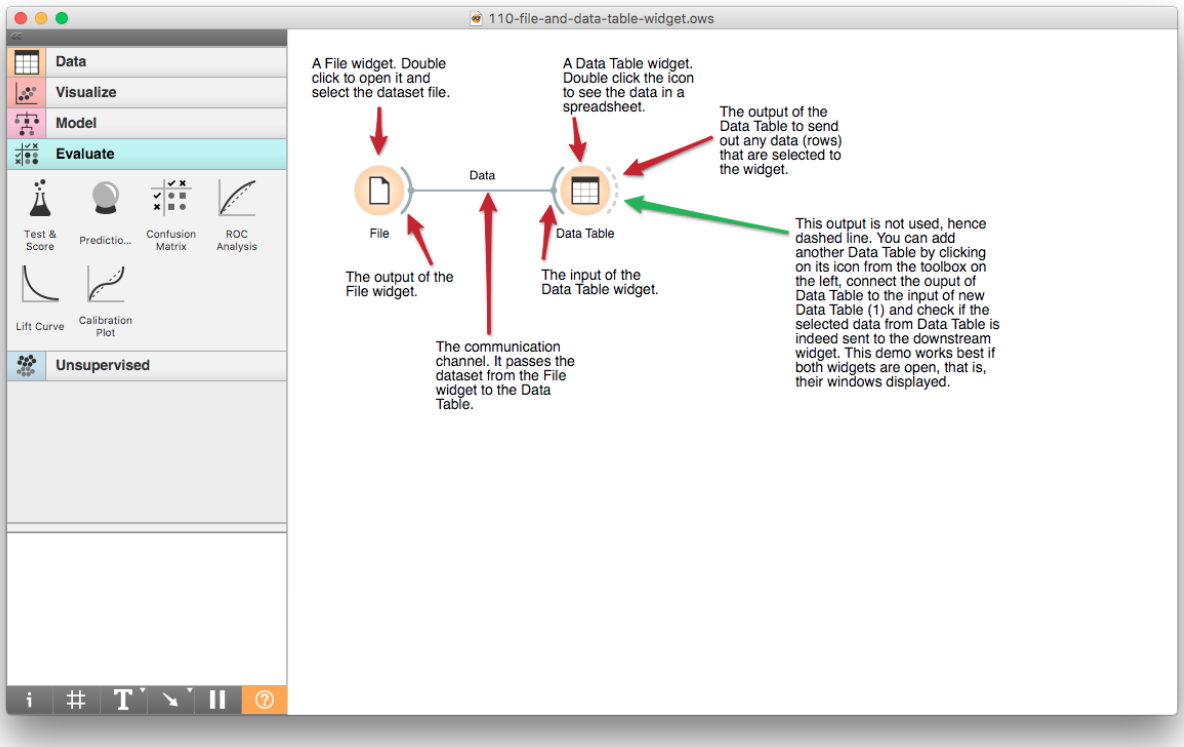
เปิดดูตัวอย่าง (Examples)

- ในกรณีที่คลิกที่ Examples จะปรากฏตัวอย่างในการทำงานของโปรแกรม Orange ดังตัวอย่างต่อไปนี้


Example Workflows
Example Workflows
File and Data Table
The basic data mining units in Orange are called widgets. There are widgets for reading the data, preprocessing, visualization, clustering, classification and others. Widgets communicate through channels. Data mining workflow is thus a collection of widgets and communication channels.
In this workflow, there is a File widget that reads the data. File widget communicates this data to Data Table widget that shows the data spreadsheet. Notice how the output of the file widget is connected to the input of the Data Table widget. In Orange, the outputs of the widgets are on the right, and the inputs on the left of the widget.
A File widget. Double click to open it and select the dataset file.
A Data Table widget. Double click the icon to see the data in a spreadsheet.
The output of the Data Table to send out any data (rows) that are selected to the widget.
File
Data Table
The output of the File widget.
The input of the Data Table widget.
This output is not used, hence dashed line. You can add another Data Table by clicking on its icon from the toolbox on the left, connect the ouput of Data Table to the input of new Data Table (1) and check if the selected data from Data Table is indeed sent to the downstream widget. This demo works best if both widgets are open, that is, their windows displayed.
The communication channel. It passes the dataset from the File widget to the Data Table.
Path: /Applications/Orange3.app/Contents/Frameworks/Python.fr…pplication/workflows/110-file-and-data-table-widget.ows
File and Data Table
Interactive Visualizations
Visualization of Data Subsets
Classification Tree
Principal Component Analysis
Cancel
Open

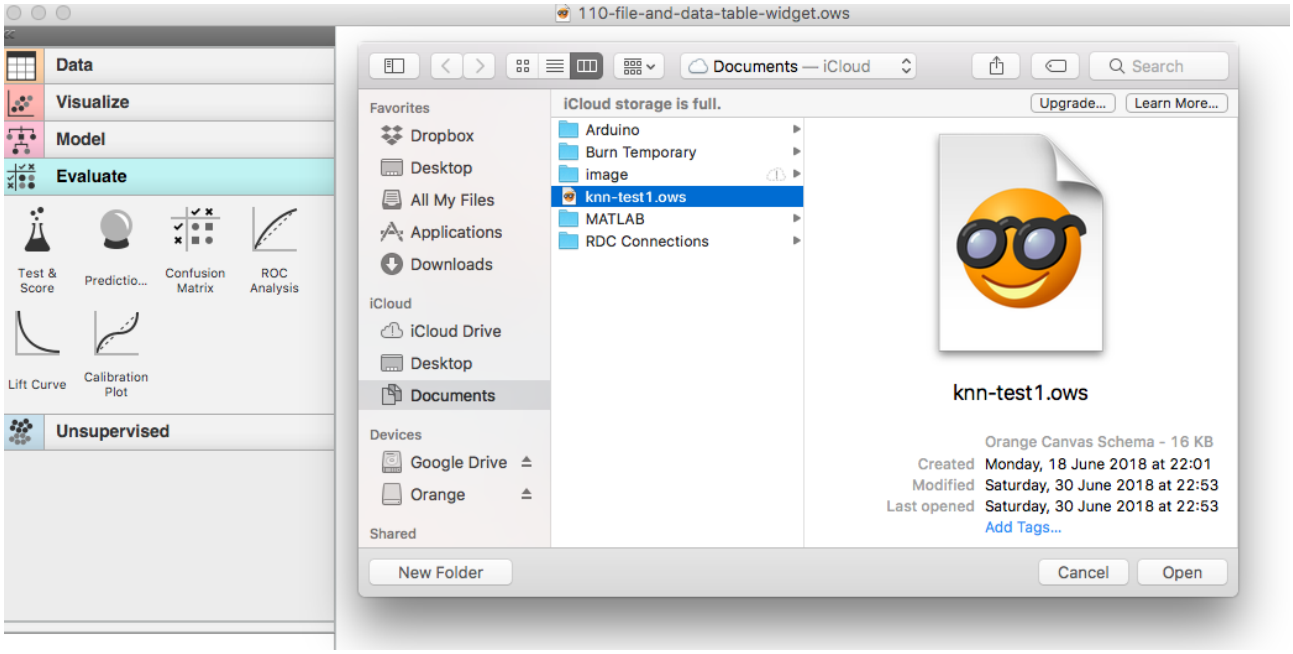

Example Workflows
Example Workflows
Visualization of Data Subsets
Some visualization widgets, like Scatter Plot and several data project widgets, can expose the data instances in the data subset. In this workflow, Scatter Plot visualize the data from the input data file, but also marks the data points that have been selected in the Data Table (selected rows).
Again, this workflow works best if both Scatter Plot and Data Table are open.
(2) Open the Scatter Plot to observe the selected subset from Data Table.
File
Scatter Plot
Data Table
Double click on this channel to check that data from Data Table is indeed fed as a data subset.
(1) Open the Data Table and select a data instance or a subset of instances (use shift key).
Path: /Applications/Orange3.app/Contents/Frameworks/Python.fra…plication/workflows/130-scatterplot-visualize-subset.ows
File and Data Table
Interactive Visualizations
Visualization of Data Subsets
Classification Tree
Principal Component Analysis
Cancel
Open

- เมื่อเลือกตัวอย่างที่ต้องการ ให้คลิกที่ปุ่ม **Open** เพื่อเปิดดูตัวอย่างนั้น

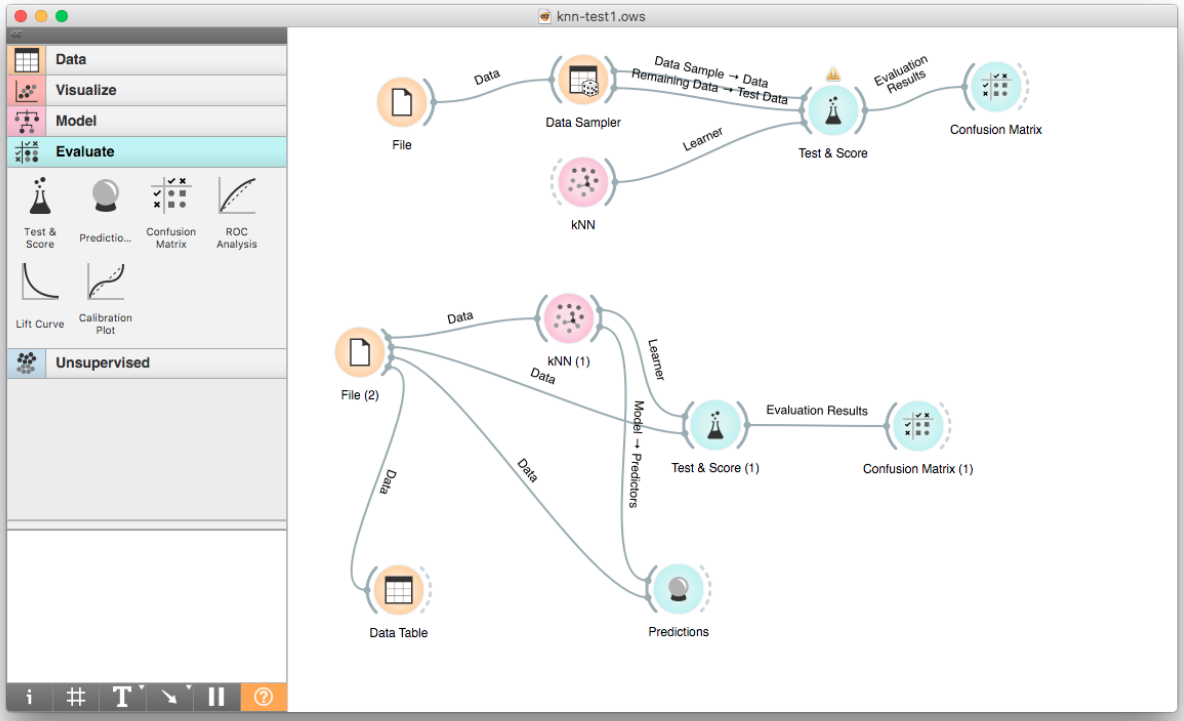


เปิดไฟล์เก่า (Open)

- หากต้องการเปิดไฟล์เดิมที่มีอยู่แล้วสามารถคลิกที่ **Open**

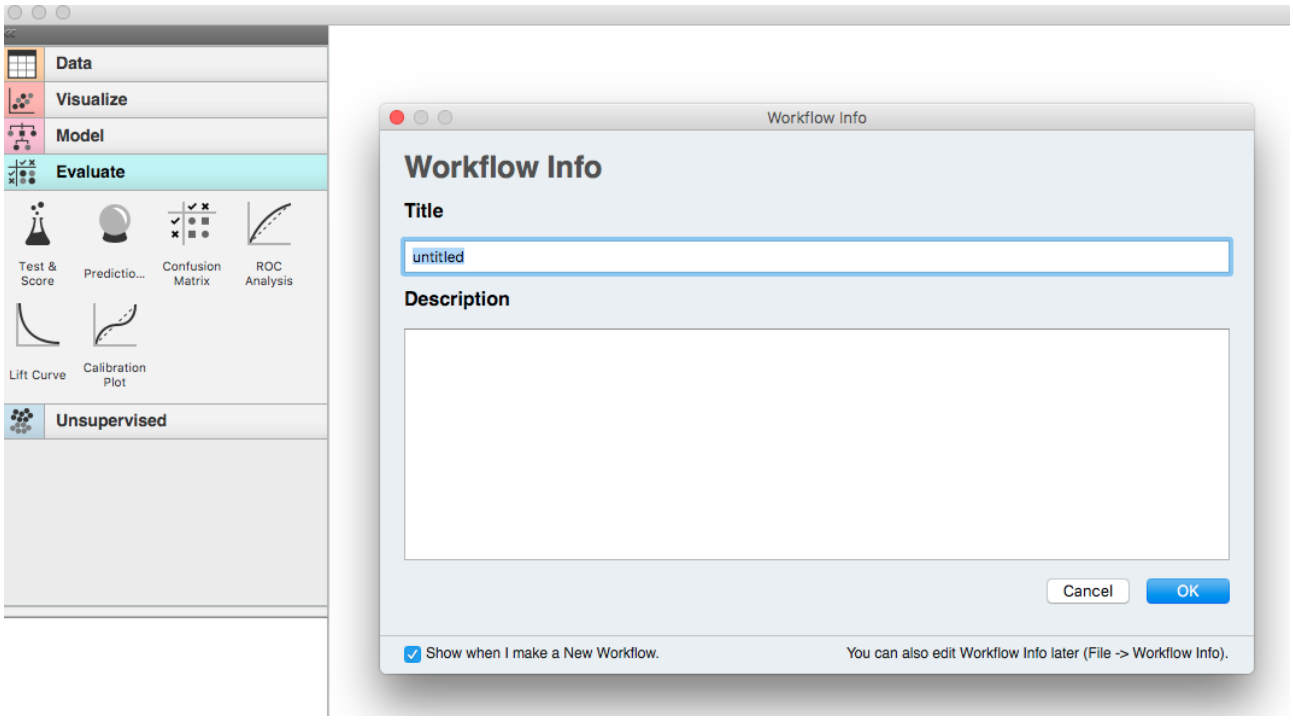


- เมื่อคลิกที่ปุ่ม Open โปรแกรมจะให้เลือกไฟล์ของโปรแกรม Orange ซึ่งไฟล์จะมีนามสกุล **.ows**
- จากนั้นคลิกที่ปุ่ม **Open** เพื่อเปิดไฟล์ที่ต้องการจะทำงาน

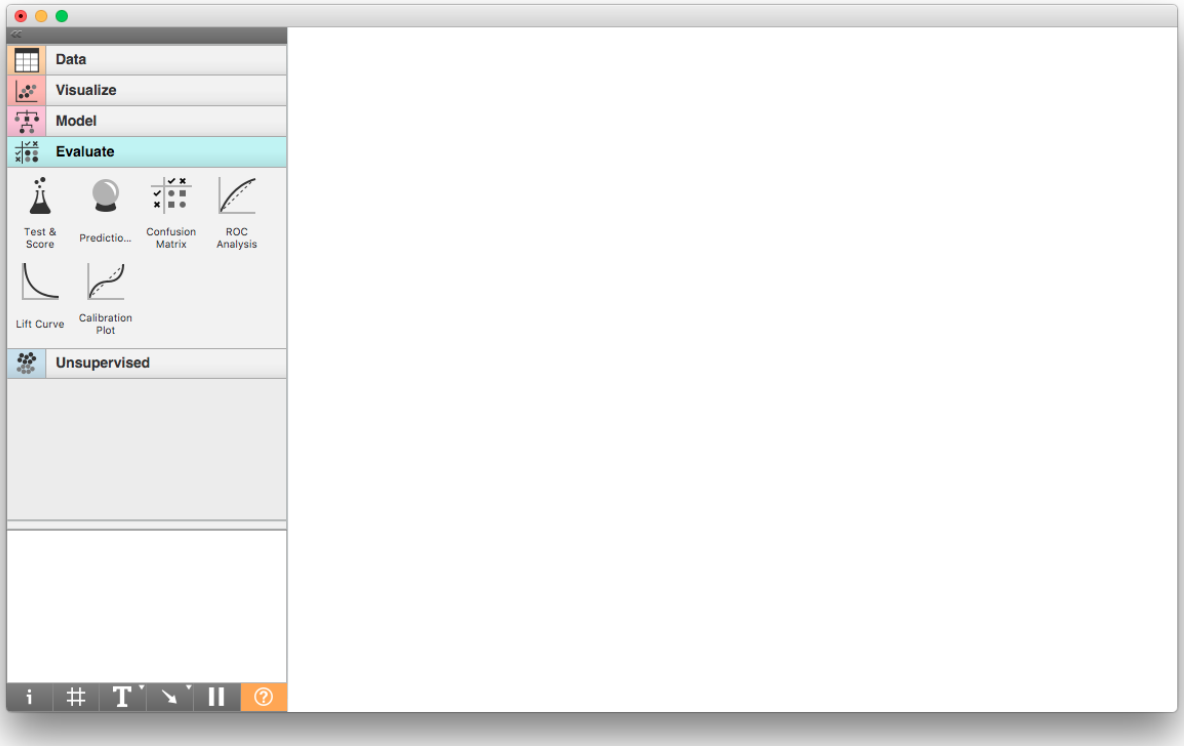


สร้างไฟล์ใหม่ (New)

- หากต้องการสร้างไฟล์ใหม่สามารถคลิกที่ **New**
- เมื่อคลิกที่ New จากนั้นจะปรากฏหน้าต่าง **Workflow Info** ให้พิมพ์รายละเอียดของโปรเจคที่ต้องการจะทำงาน ประกอบด้วย Title และ Description
  - Title - ชื่อของโปรเจค
  - Description – รายละเอียดของโปรเจค
- เมื่อพิมพ์รายละเอียดที่ต้องการให้คลิกที่ปุ่ม **OK** เพื่อเริ่มต้นการทำงาน

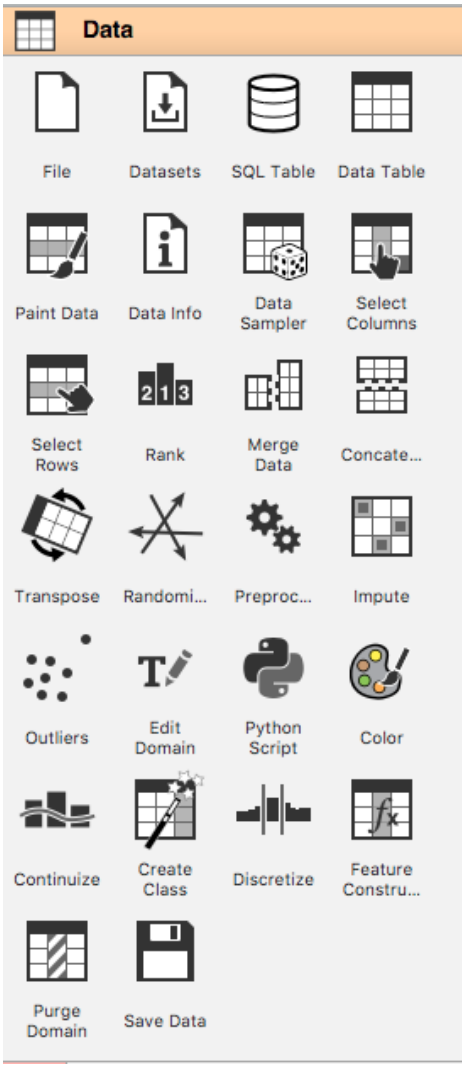


- เมื่อคลิกที่ **OK** จะปรากฏโปรแกรม Orange ที่พร้อมทำงาน

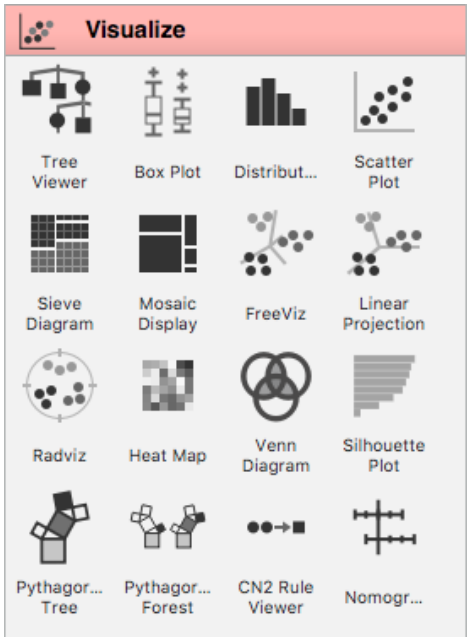


กล่องเครื่องมือ (Toolbox)

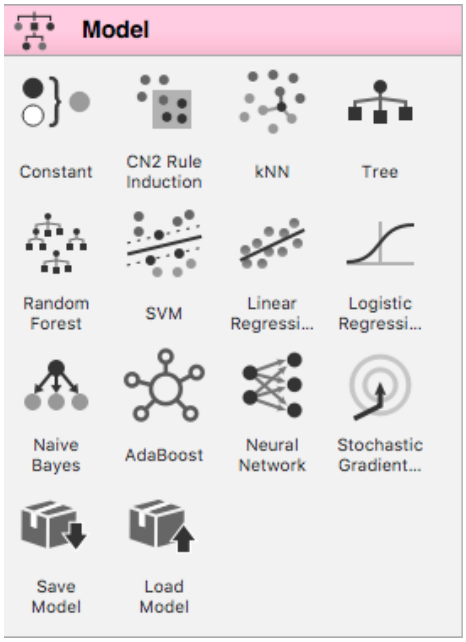
Data



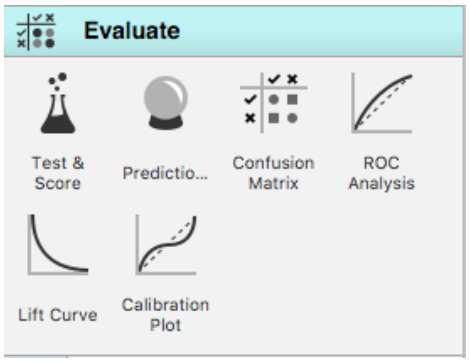
Visualize



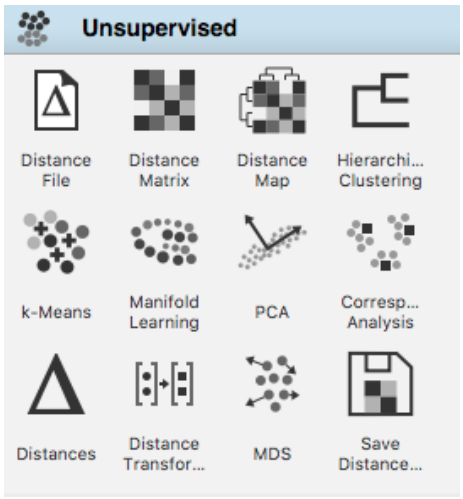
Model



Evaluate



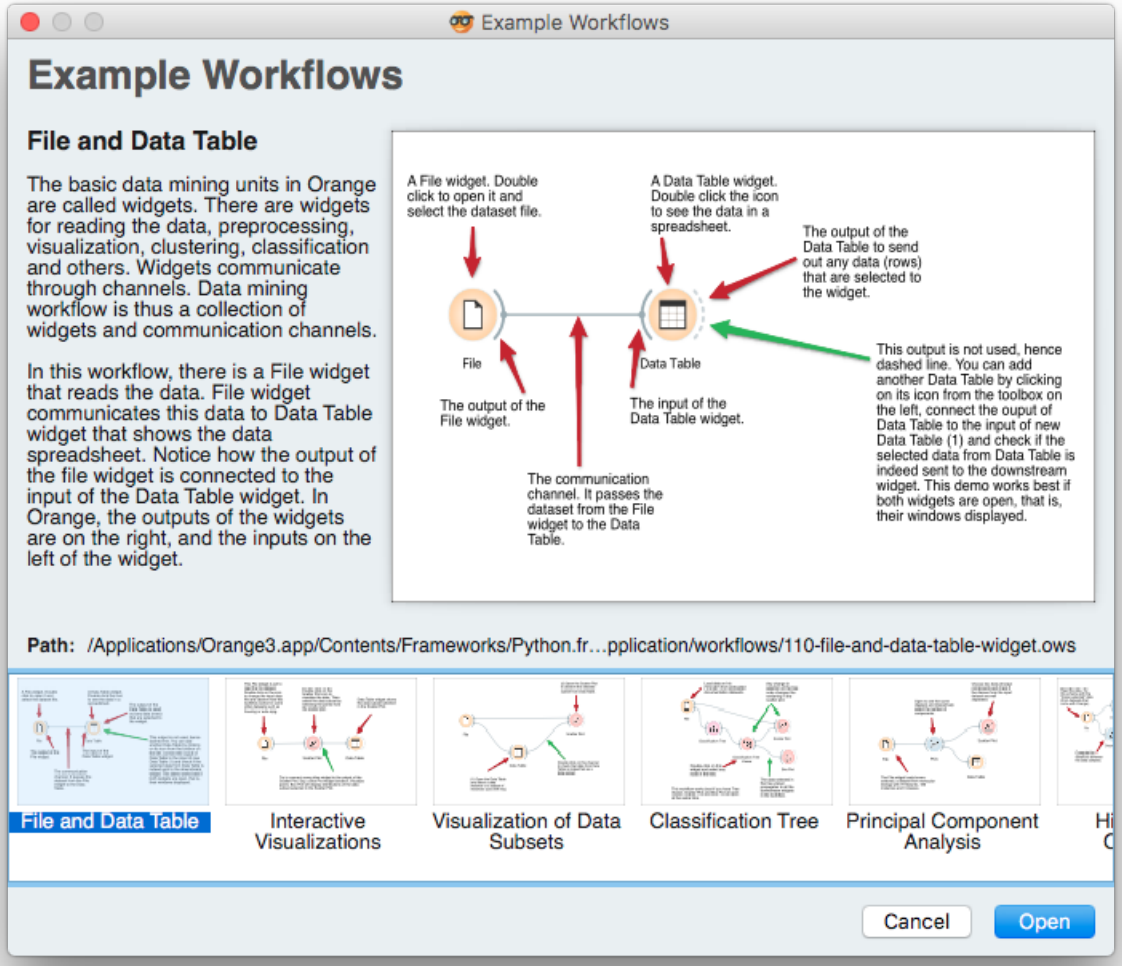
Unsupervised



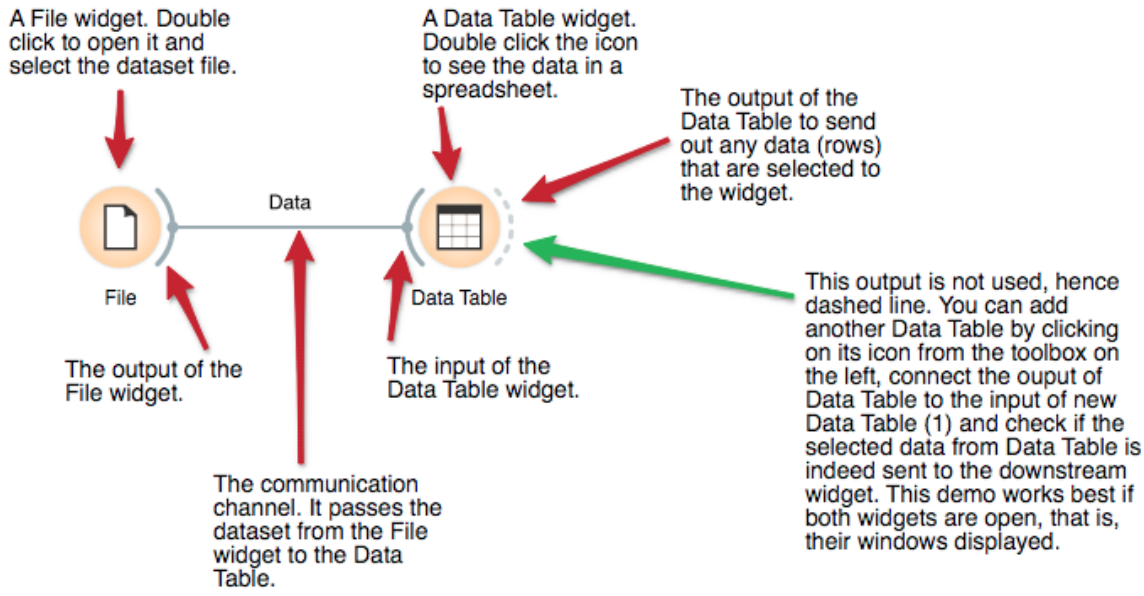
ไฟล์และตารางข้อมูล (File and Data Table)

ตัวอย่างไฟล์และตารางข้อมูล (Example of File and Data Table)

- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม



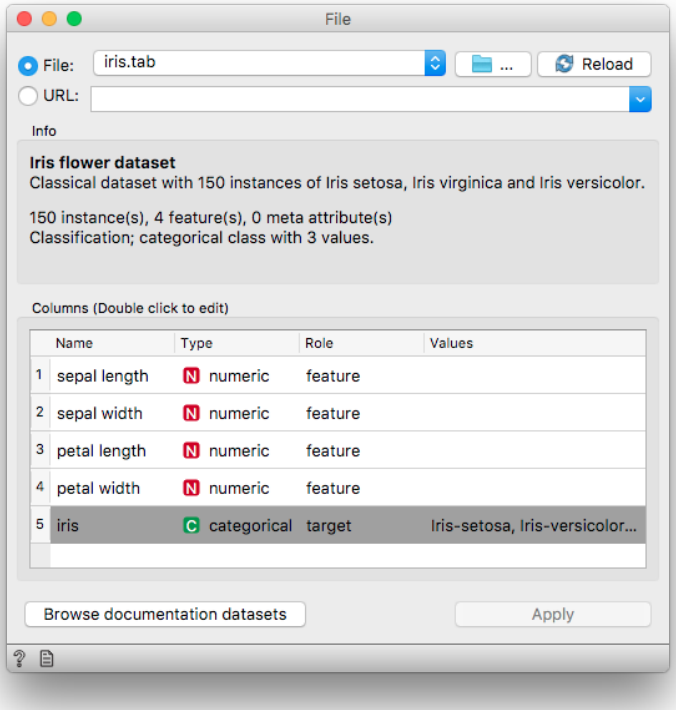
- จากนั้นให้เลือกที่ **File and Data Table** และกดที่ปุ่ม **Open**



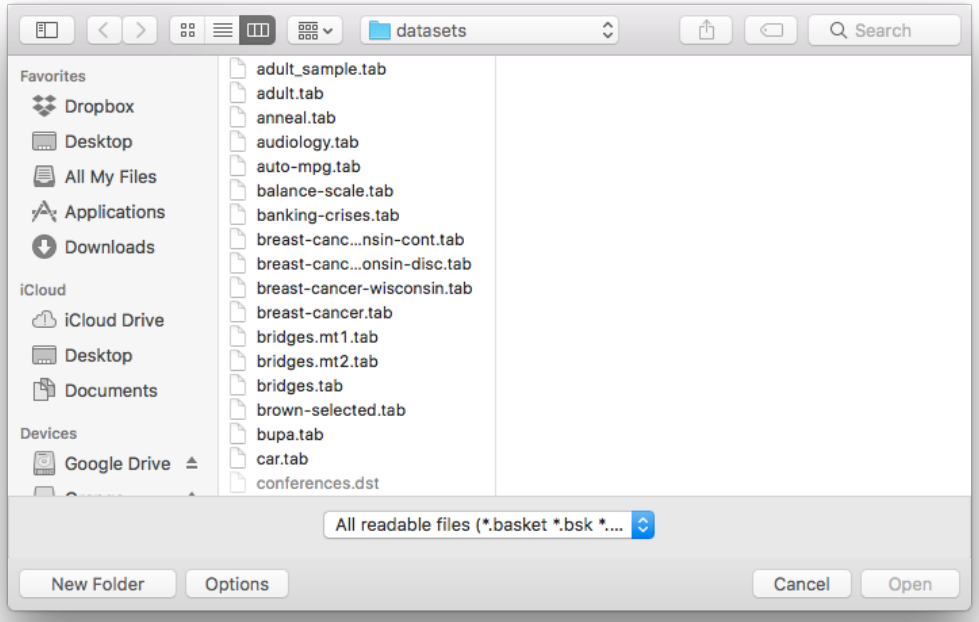
- เมื่อคลิกที่ Open จะปรากฏตัวอย่างการใช้งานไอคอน (Icon) หรือวิดเจท (Widget) File และ Data Table

จากตัวอย่างเริ่มต้นโดย

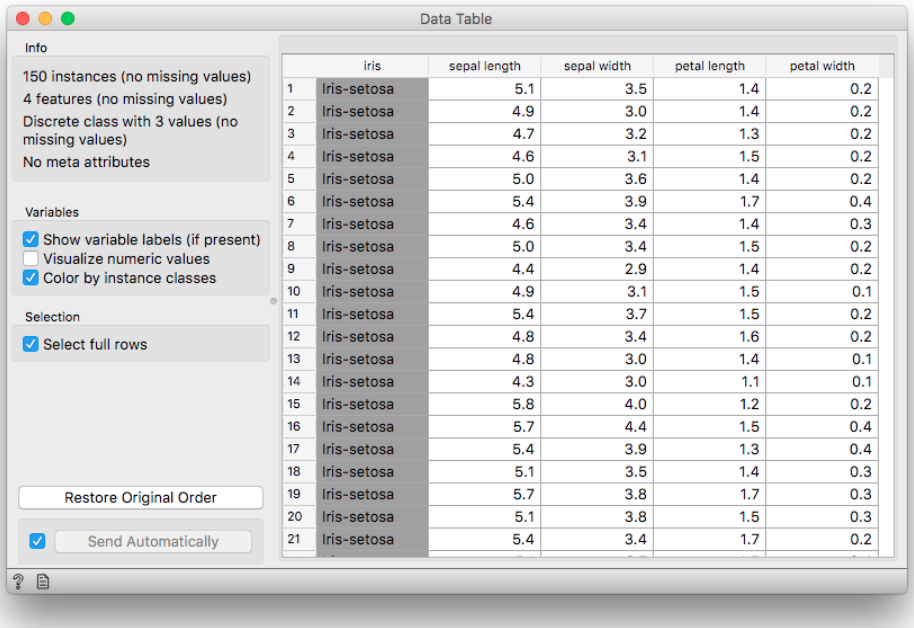
- ดับเบิ้ลคลิกที่ไอคอน **File** จากนั้นจะปรากฏหน้าต่างที่แสดงชุดข้อมูล (Dataset)
  - ในตัวอย่างนี้จะเป็นชุดข้อมูลที่ชื่อ iris ซึ่งเป็นชุดข้อมูลดอกไม้ที่ประกอบไปด้วย 4 attribute และ 3 class



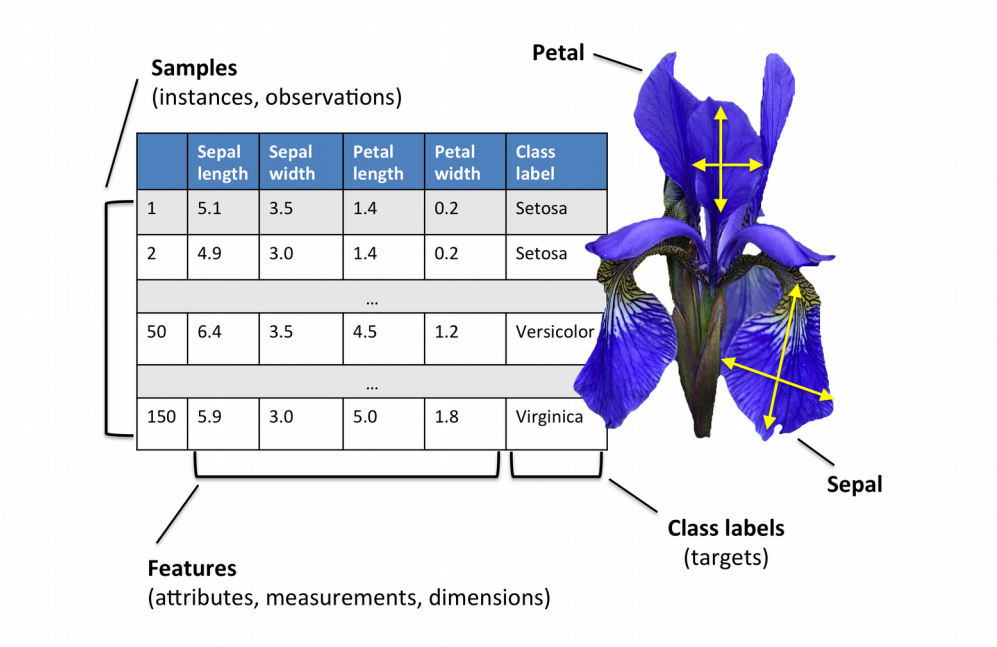
- หากต้องการเลือกข้อมูลชุดอื่นสามารถทำได้โดยคลิกที่รูปโฟลเดอร์
- จากนั้นจะปรากฏหน้าต่างให้เลือกข้อมูลชุดอื่น



- เมื่อเลือกชุดข้อมูลเสร็จเรียบร้อยให้คลิกที่ปุ่ม **Open**
- ในกรณีนี้ เลือกใช้ข้อมูลชุด iris
- จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน **Data Table** จะปรากฏหน้าต่างดังต่อไปนี้



- โดยหน้าต่างจะแสดงข้อมูลทั้งหมดของชุดข้อมูล iris ซึ่งประกอบด้วยข้อมูลทั้งสิ้น 150 ชุด
  - ข้อมูลแต่ละแถว (Instance) จะประกอบไปด้วย 4 attribute / feature คือ
    - Sepal length
    - Sepal width
    - Petal length
    - Petal width
  - ข้อมูลมีทั้งหมด 3 กลุ่ม (Class / Label) ประกอบด้วย
    - iris-setosa
    - iris-versicolor
    - iris-virginica

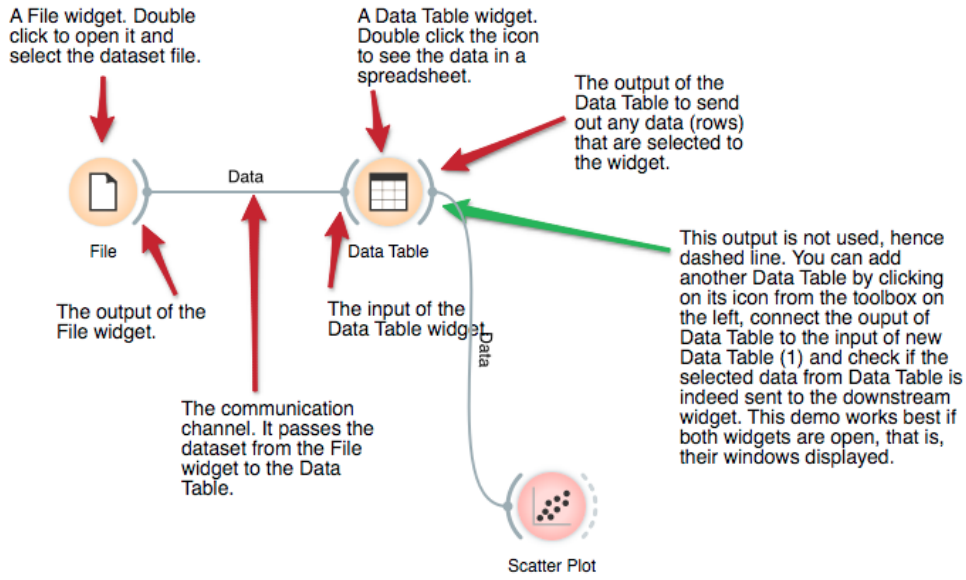


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

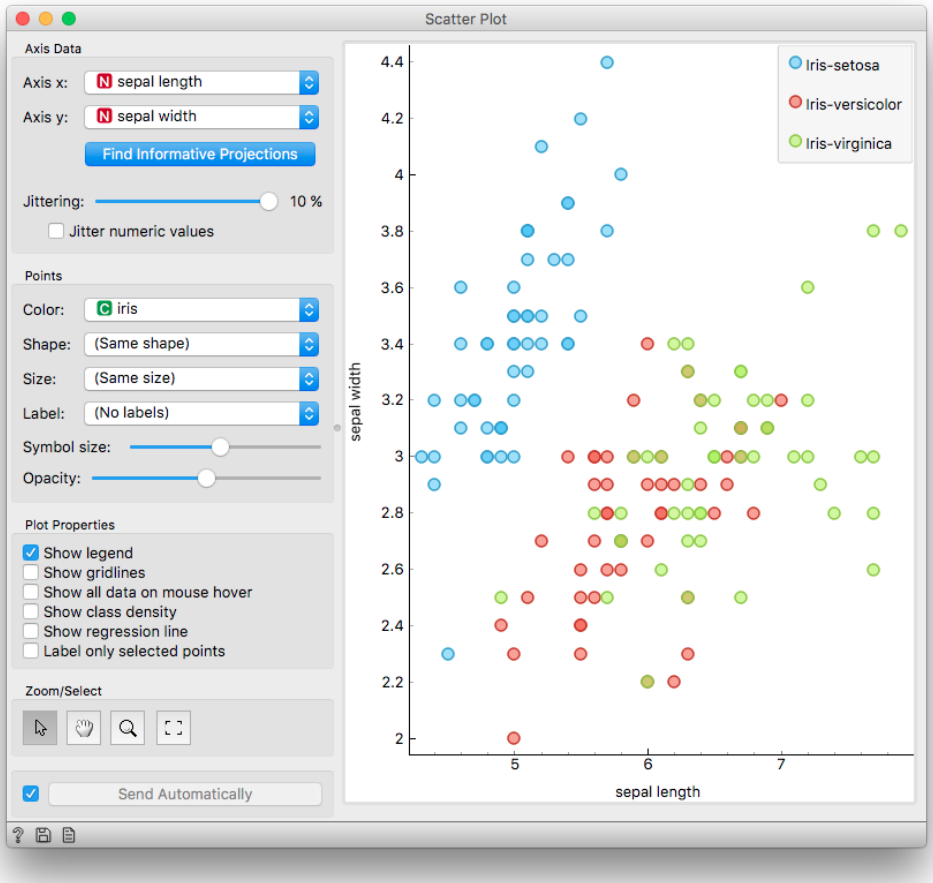
[ที่มา: https://rpubs.com/wjholst/322258]

เพิ่มไอคอน Visualize (Add Visualize Icon)

- จากนั้นทำการเพิ่มไอคอน Visualize เพื่อที่จะดูลักษณะของข้อมูลของ iris dataset
- ทำได้โดยเลือกที่แทป (Tab) **Visualize** และคลิกที่ไอคอน **Scatter Plot** จากนั้น Scatter Plot จะไปปรากฏใน Workflow ของโปรแกรม

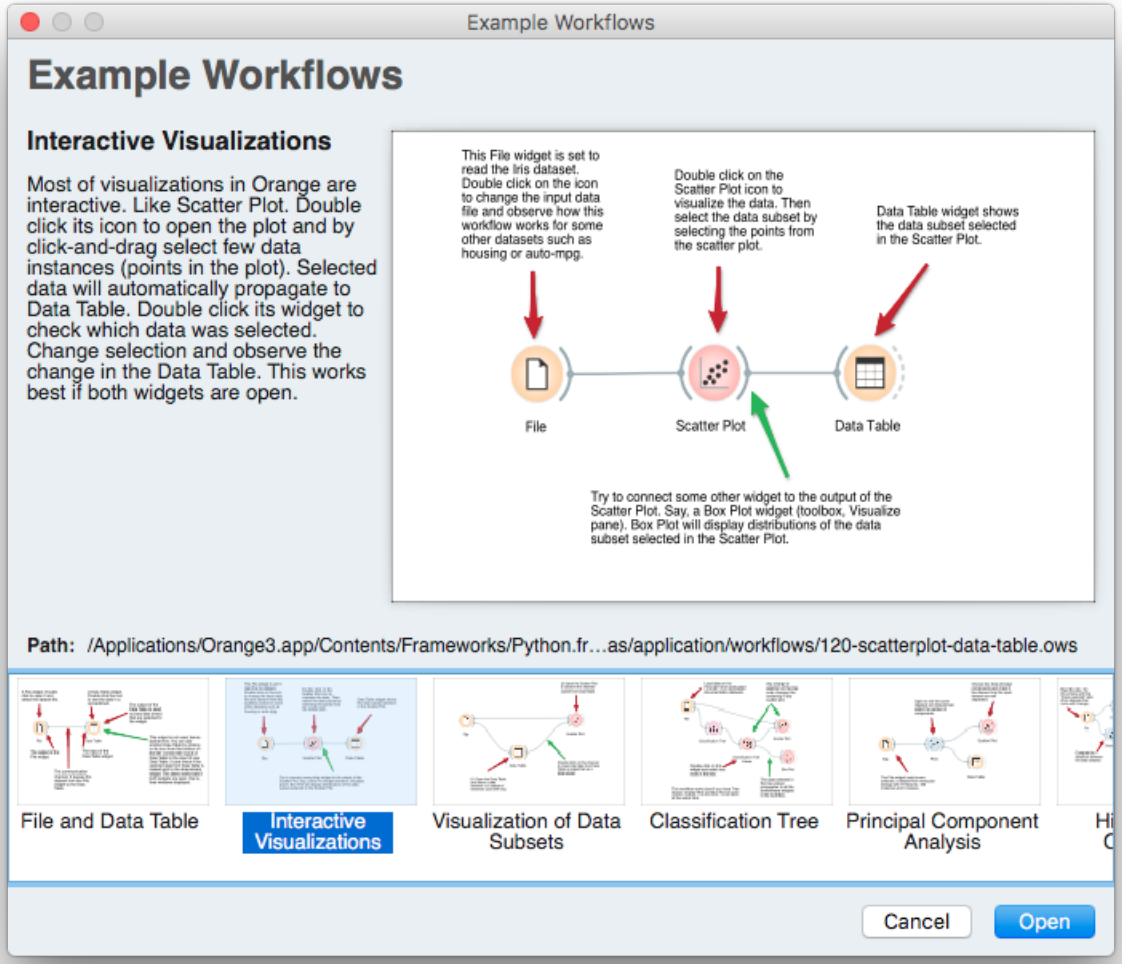


- ดับเบิ้ลคลิกที่ไอคอน Scatter Plot เพื่อดูกราฟ
  - โดยกราฟแสดงให้เห็นถึงการกระจายของข้อมูล

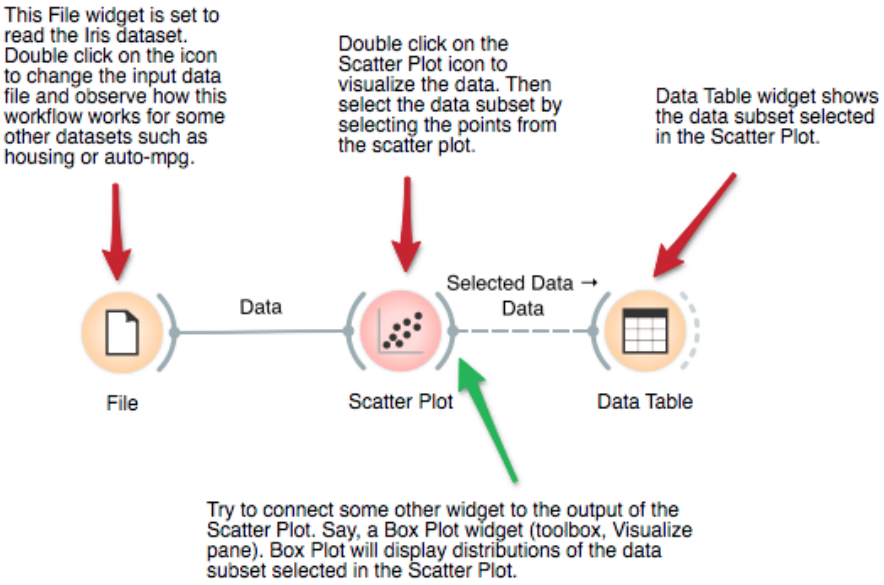


การแสดงข้อมูลเชิงโต้ตอบ (Interactive Visualizations)

- ในส่วนนี้เป็นการแสดงให้เห็นถึงวิธีการแสดงข้อมูลแบบ Visualization
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Interactive Visualizations** และคลิกที่ปุ่ม **Open**

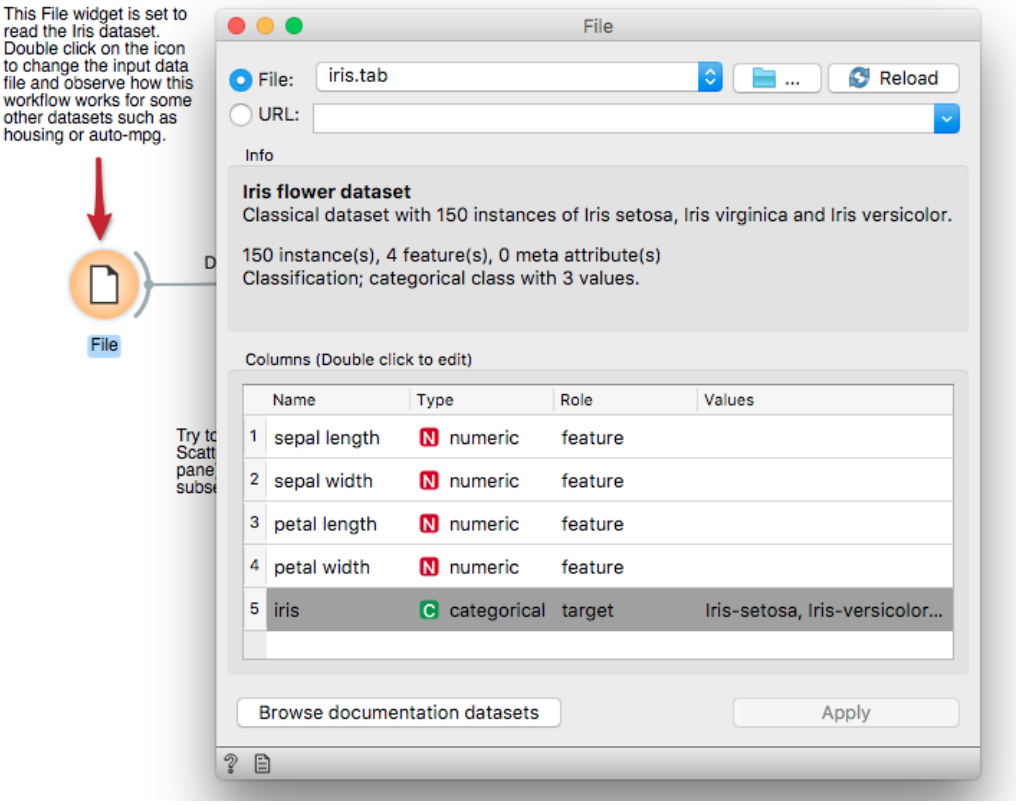


- เมื่อคลิกที่ Open จะปรากฏตัวอย่างการใช้งาน Interactive Visualizations ดังตัวอย่างต่อไปนี้

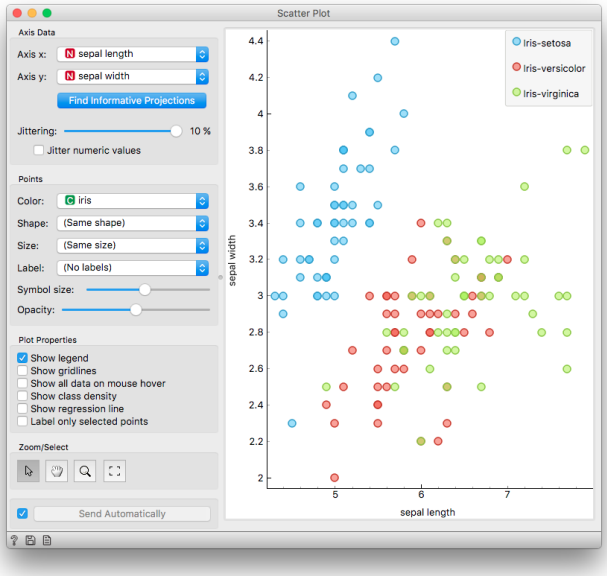


ขั้นตอนในการทำ Interactive Visualizations ทำได้ดังต่อไปนี้

- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูลที่ต้องการใช้งาน
  - ตัวอย่างกำหนดให้ใช้ชุดข้อมูล iris

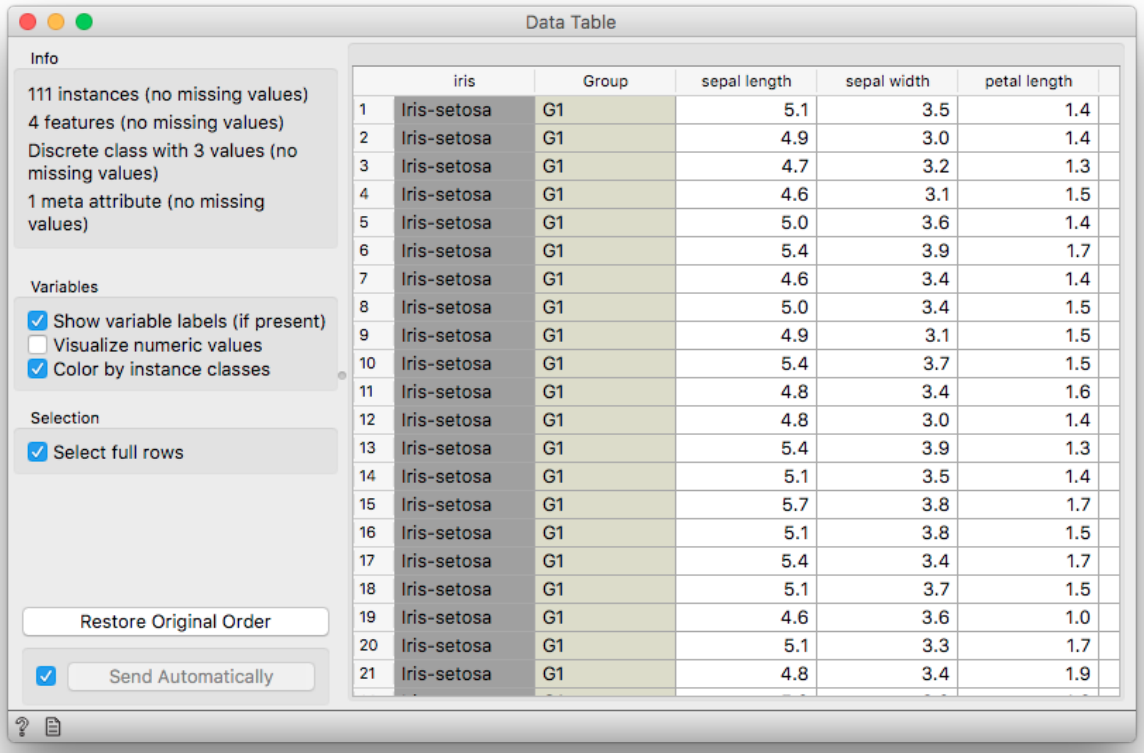


- ขั้นตอนต่อไปดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** โปรแกรมจะแสดงหน้าต่างเพื่อแสดงข้อมูลแบบ Visualization



- จากหน้าต่างข้างต้นสามารถ **คลิกข้อมูลแต่ละจุด** หรือ **ใช้เมาส์คลิกเพื่อเลือกข้อมูลแบบกลุ่ม**
  - ตัวอย่างแสดงให้เห็นถึงการเลือกข้อมูลบางส่วน จากนั้นให้คลิกเพื่อปิดหน้าต่าง

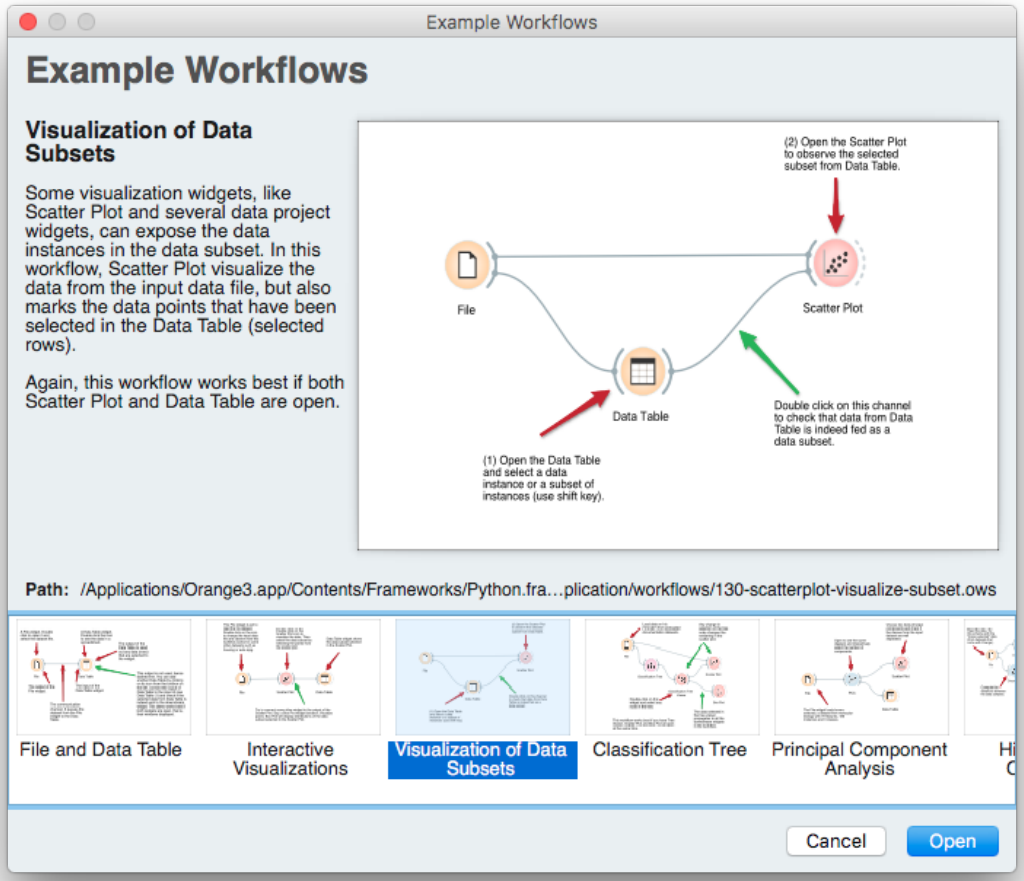
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Data Table**
  - โปรแกรมจะเปิดหน้าต่างเพื่อแสดงข้อมูลที่ได้เลือกไว้ในขั้นตอน Scatter Plot



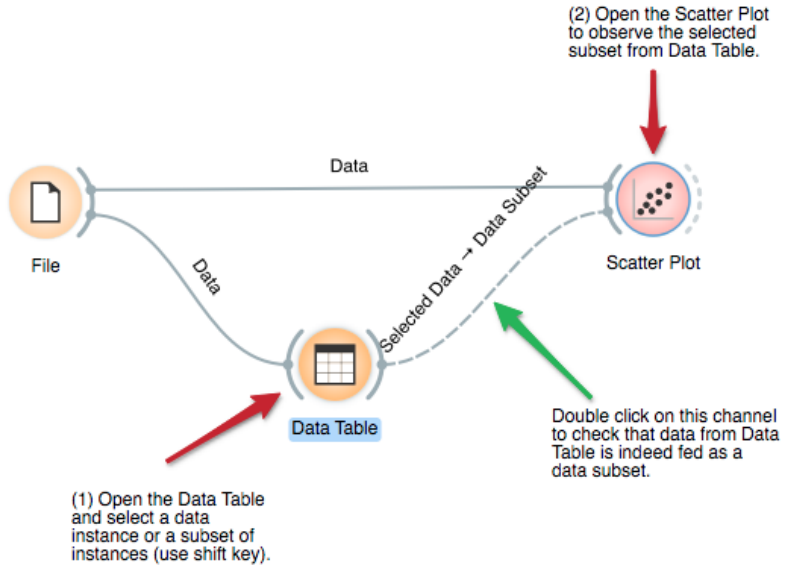
	iris	Group	sepal length	sepal width	petal length
1	Iris-setosa	G1	5.1	3.5	1.4
2	Iris-setosa	G1	4.9	3.0	1.4
3	Iris-setosa	G1	4.7	3.2	1.3
4	Iris-setosa	G1	4.6	3.1	1.5
5	Iris-setosa	G1	5.0	3.6	1.4
6	Iris-setosa	G1	5.4	3.9	1.7
7	Iris-setosa	G1	4.6	3.4	1.4
8	Iris-setosa	G1	5.0	3.4	1.5
9	Iris-setosa	G1	4.9	3.1	1.5
10	Iris-setosa	G1	5.4	3.7	1.5
11	Iris-setosa	G1	4.8	3.4	1.6
12	Iris-setosa	G1	4.8	3.0	1.4
13	Iris-setosa	G1	5.4	3.9	1.3
14	Iris-setosa	G1	5.1	3.5	1.4
15	Iris-setosa	G1	5.7	3.8	1.7
16	Iris-setosa	G1	5.1	3.8	1.5
17	Iris-setosa	G1	5.4	3.4	1.7
18	Iris-setosa	G1	5.1	3.7	1.5
19	Iris-setosa	G1	4.6	3.6	1.0
20	Iris-setosa	G1	5.1	3.3	1.7
21	Iris-setosa	G1	4.8	3.4	1.9

การแสดงข้อมูลย่อย (Visualizations of Data Subsets)

- ในส่วนนี้แสดงให้เห็นถึงการ Visualization ของชุดข้อมูลย่อย
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Visualization of Data Subsets** และคลิกที่ปุ่ม **Open**

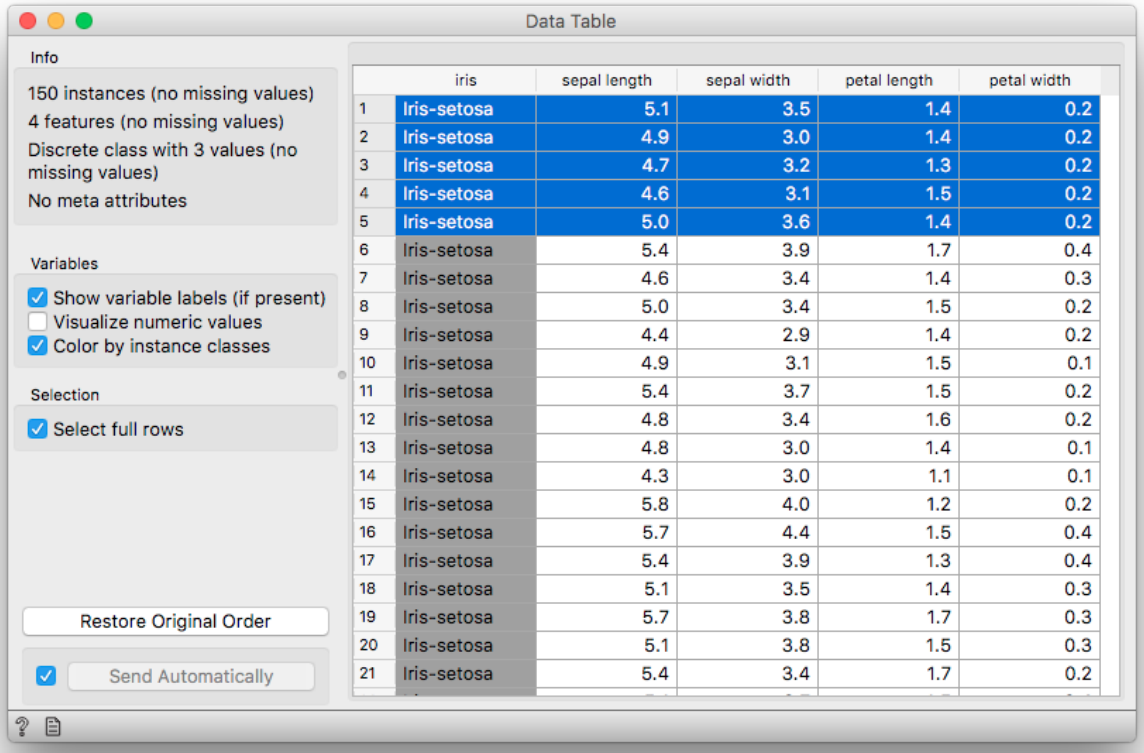


- เมื่อคลิกที่ **Open** จะปรากฏตัวอย่างการใช้งาน Visualization of Data Subsets ดังตัวอย่างต่อไปนี้



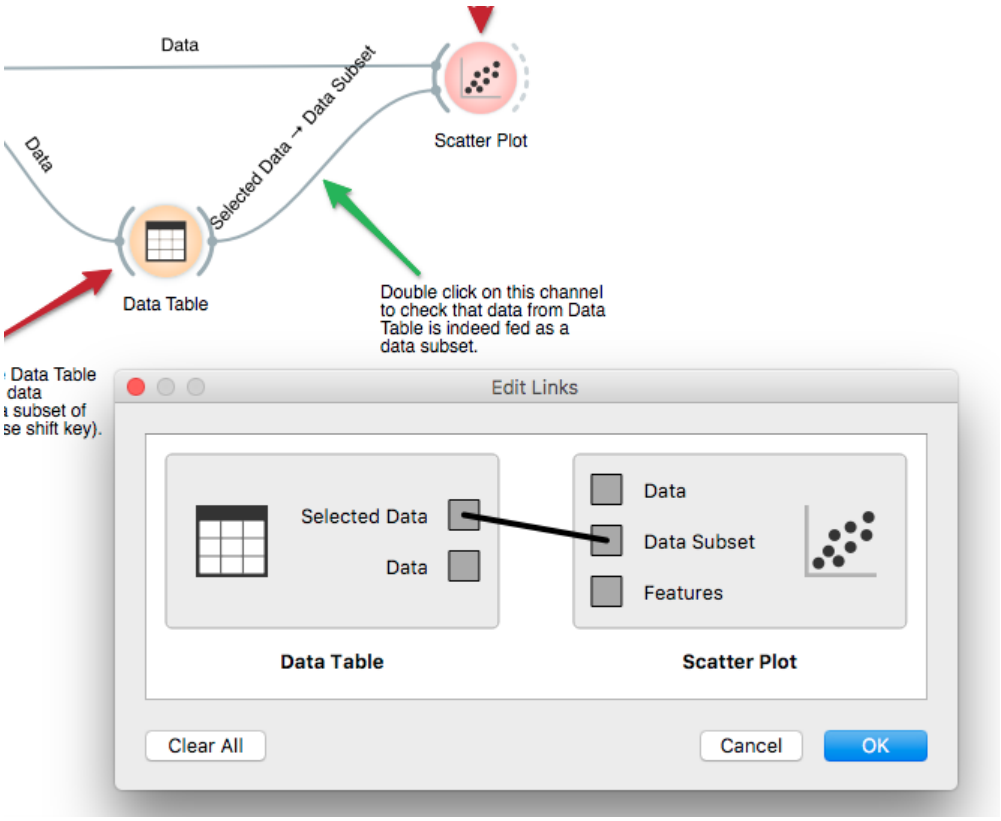
ขั้นตอนในการทำ Visualization of Data Subsets ทำได้ดังต่อไปนี้

- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูลที่ต้องการใช้งาน ในตัวอย่างใช้ข้อมูล iris
- ดับเบิ้ลคลิกที่ไอคอน **Data Table** และเลือกข้อมูลที่ต้องการนำไป Visualization



	iris	sepal length	sepal width	petal length	petal width
1	Iris-setosa	5.1	3.5	1.4	0.2
2	Iris-setosa	4.9	3.0	1.4	0.2
3	Iris-setosa	4.7	3.2	1.3	0.2
4	Iris-setosa	4.6	3.1	1.5	0.2
5	Iris-setosa	5.0	3.6	1.4	0.2
6	Iris-setosa	5.4	3.9	1.7	0.4
7	Iris-setosa	4.6	3.4	1.4	0.3
8	Iris-setosa	5.0	3.4	1.5	0.2
9	Iris-setosa	4.4	2.9	1.4	0.2
10	Iris-setosa	4.9	3.1	1.5	0.1
11	Iris-setosa	5.4	3.7	1.5	0.2
12	Iris-setosa	4.8	3.4	1.6	0.2
13	Iris-setosa	4.8	3.0	1.4	0.1
14	Iris-setosa	4.3	3.0	1.1	0.1
15	Iris-setosa	5.8	4.0	1.2	0.2
16	Iris-setosa	5.7	4.4	1.5	0.4
17	Iris-setosa	5.4	3.9	1.3	0.4
18	Iris-setosa	5.1	3.5	1.4	0.3
19	Iris-setosa	5.7	3.8	1.7	0.3
20	Iris-setosa	5.1	3.8	1.5	0.3
21	Iris-setosa	5.4	3.4	1.7	0.2

- ดับเบิ้ลคลิกที่ **เส้นเชื่อมต่อระหว่าง Data Table** และ **Scatter Plot**
  - ให้สังเกตการเชื่อม (Link) ข้อมูลระหว่าง Data Table และ Scatter Plot

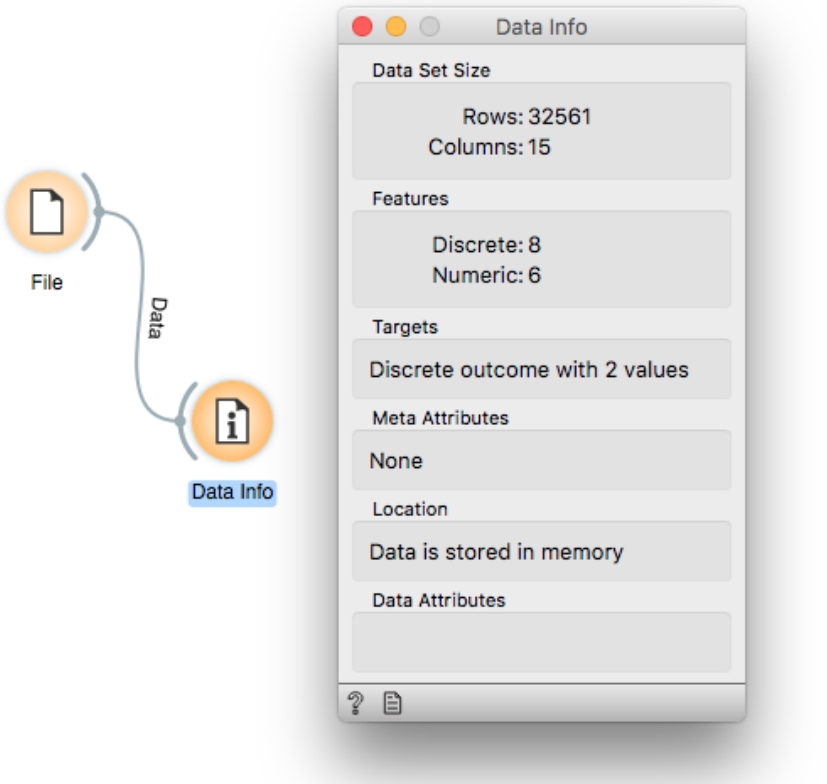


- สุดท้าย ดับเบิ้ลคลิกที่ **Scatter Plot** เพื่อดูข้อมูลชุดย่อยที่ได้เลือกไว้


Scatter Plot
Axis Data
Axis x: sepal length
Axis y: sepal width
Find Informative Projections
Jittering: 10 %
Jitter numeric values
Points
Color: iris
Shape: (Same shape)
Size: (Same size)
Label: (No labels)
Symbol size:
Opacity:
Plot Properties
Show legend
Show gridlines
Show all data on mouse hover
Show class density
Show regression line
Label only selected points
Zoom/Select
Send Automatically
Iris-setosa
Iris-versicolor
Iris-virginica
sepal width
sepal length
4.4
4.2
4
3.8
3.6
3.4
3.2
3
2.8
2.6
2.4
2.2
2
5
6
7

การแสดงรายละเอียดข้อมูล (Data Information)

- หากต้องการทราบรายละเอียด (Information) ของข้อมูลที่นำมาใช้งาน สามารถทำได้โดยเลือกที่ไอคอน **Data Info** โดยทำตาม workflow ดังต่อไปนี้

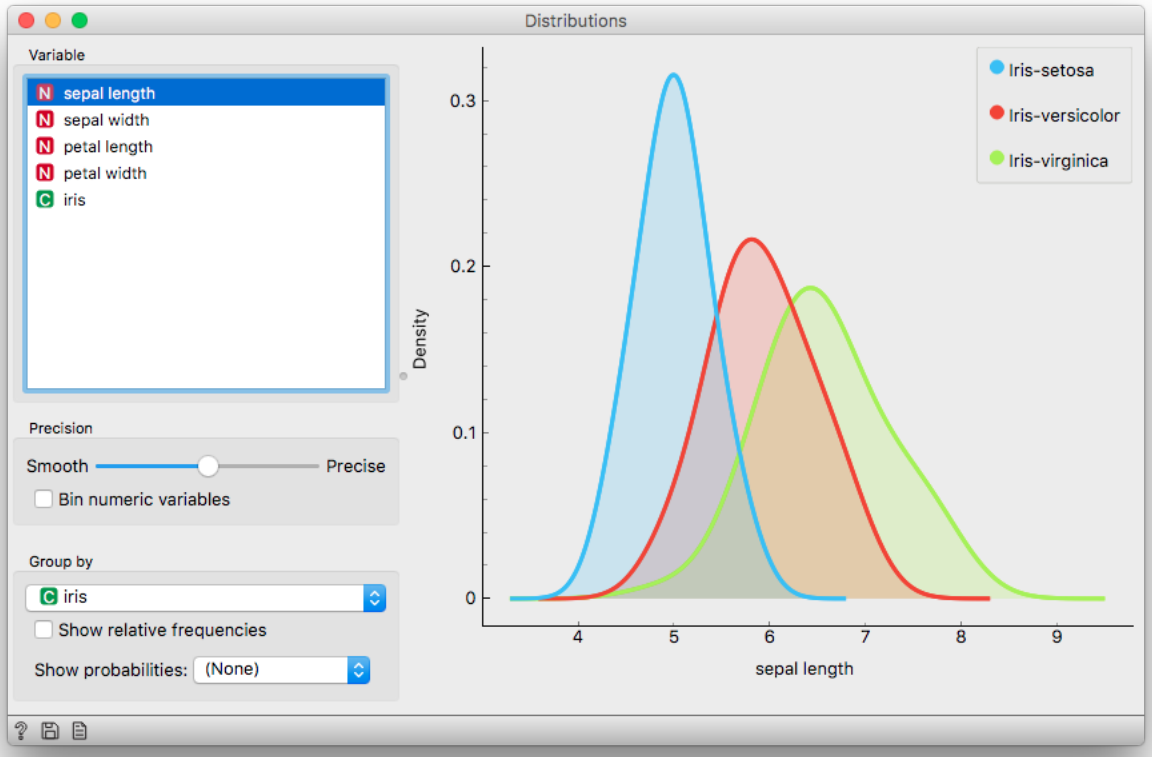


การแสดงการกระจายข้อมูล (Data Distributions)

- โปรแกรม Orange มีเครื่องมือ (Tool) ที่ช่วยให้ผู้ใช้งานสามารถดูการกระจายของข้อมูล เช่น จำนวนของข้อมูลในแต่ละ Class จำนวนของข้อมูลในแต่ละ Attribute เป็นต้น
- ผู้ใช้งานสามารถเห็นข้อมูลในลักษณะ Visualize ทำให้เข้าใจลักษณะของข้อมูลมากขึ้น

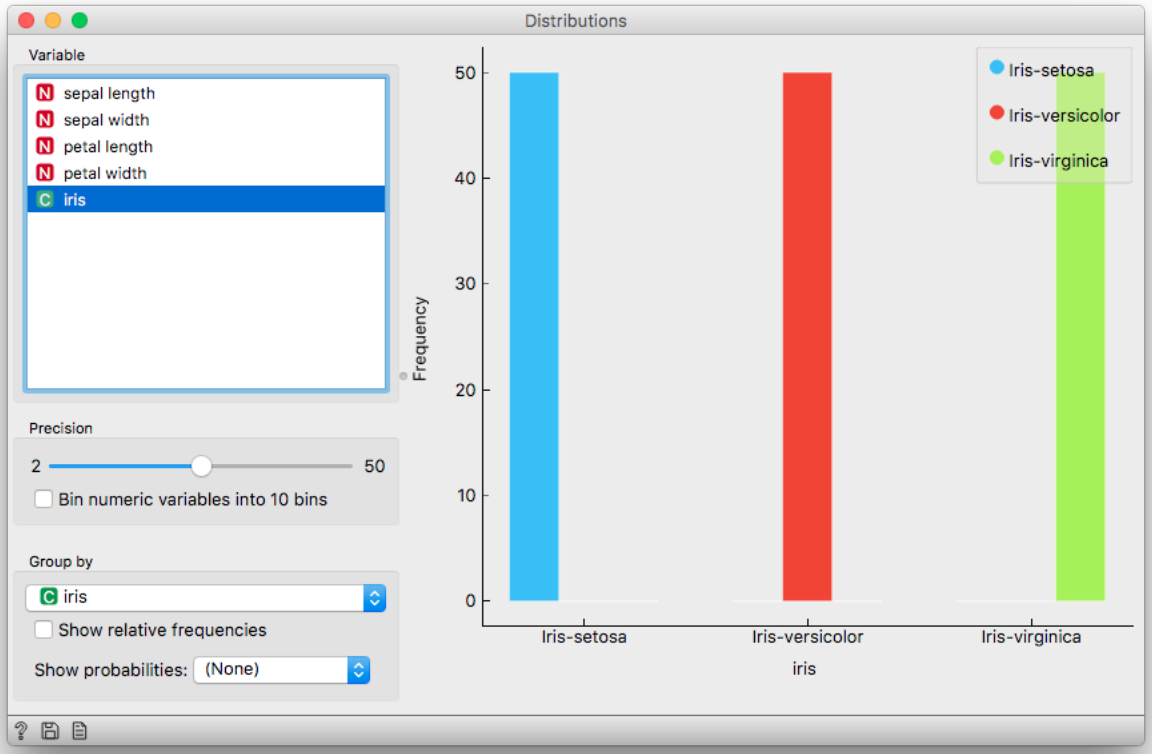
- การกระจายข้อมูล สามารถทำได้ดัง workflow ต่อไปนี้

- ในกรณีนี้ ข้อมูลที่นำมาใช้เพื่อแสดงการกระจายคือชุดข้อมูล iris
- ผู้ใช้งานสามารถดับเบิ้ลคลิกที่ไอคอน **File** และเลือกชุดข้อมูลที่ต้องการแสดงได้ตามต้องการ
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Distributions** จะปรากฏหน้าต่างดังตัวอย่างต่อไปนี้

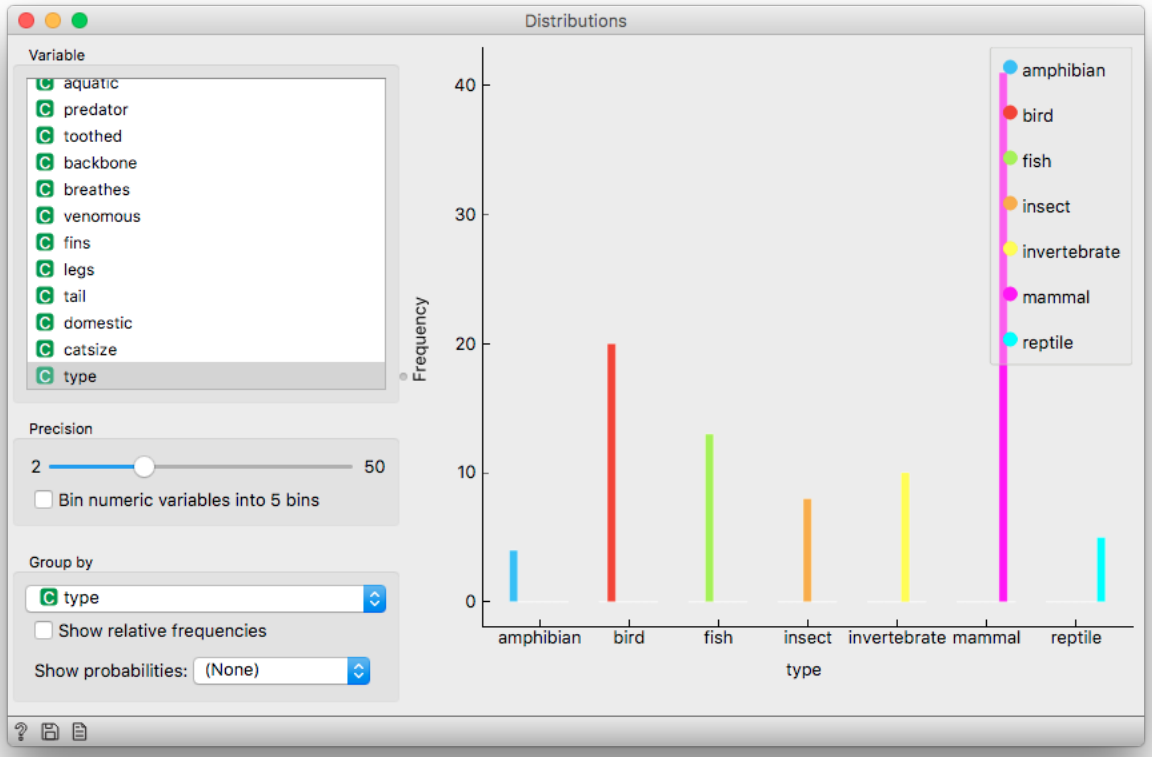


- ตัวอย่างข้างต้น แสดงให้เห็นถึงการกระจายข้อมูลของ Attribute ที่ชื่อ Sepal Length ทำให้รู้ถึงช่วงความกว้างของ Sepal (Sepal Length) ของดอกไม้ iris ในแต่ละสายพันธุ์ (Setosa, Versicolor และ Virginica) เป็นต้น

- ตัวอย่างต่อไปแสดงให้เห็นถึงจำนวนของสายพันธุ์ดอกไม้ iris ในแต่ละ Class โดยในชุดข้อมูล iris มีจำนวนดอกไม้ 150 ดอก แบ่งออกเป็น 3 Class และมี Class ละ 50 ดอก



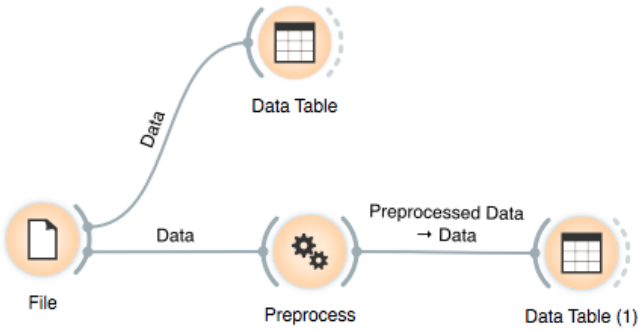
- ตัวอย่างต่อไปนี้ แสดงให้เห็นถึงการกระจายข้อมูลของสัตว์แต่ละประเภท



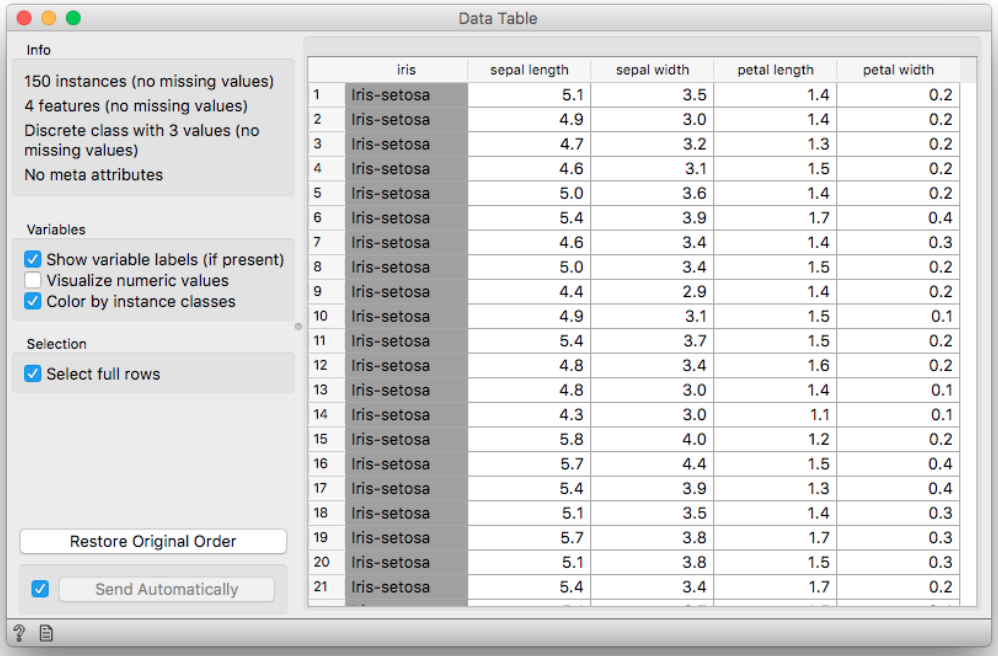
การประมวลผลข้อมูลเบื้องต้น (Data Preprocessing)

- การประมวลผลข้อมูลเบื้องต้น คือการเตรียมข้อมูลให้พร้อมสำหรับการเรียนรู้ เช่น
  - บางครั้งข้อมูลที่จะนำมาใช้มีข้อมูลบางส่วนที่สูญหายไป เรียกว่า Missing Value
  - ข้อมูลที่จะนำมาใช้ยังไม่ผ่านการ Normalize ให้อยู่ในช่วงข้อมูลเดียวกัน
  - ข้อมูลที่นำมาใช้มีจำนวน Feature/Attribute เยอะจนเกินไป อาจต้องใช้วิธีการ เช่น เลือก Feature ที่มีความสัมพันธ์ (Relevant) หรือใช้วิธี PCA เพื่อหา Component ที่เหมาะสมที่สุด

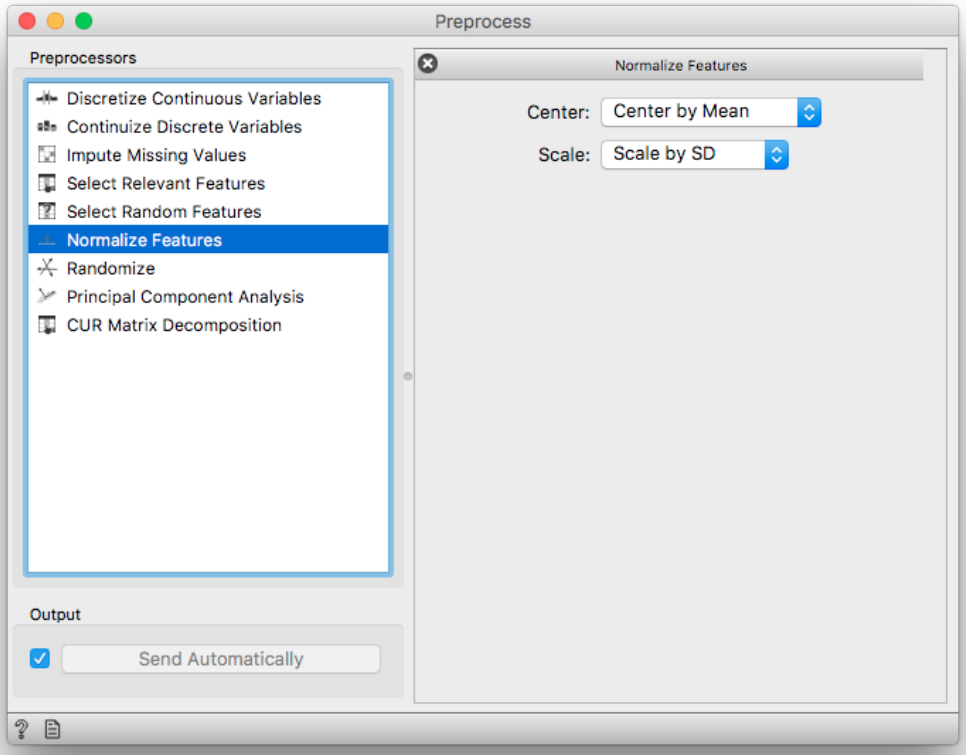
- ตัวอย่างต่อไปนี้ เป็นการประมวลผลข้อมูลเบื้องต้น โดยการ Normalize ข้อมูล Feature สามารถทำได้ดัง workflow ต่อไปนี้



- ตัวอย่างเลือกใช้ข้อมูล **iris** ในการทดสอบ สามารถทำได้โดยดับเบิ้ลคลิกที่ไอคอน File และเลือกที่ข้อมูล iris
- ดับเบิ้ลคลิกที่ไอคอน **Data Table** เพื่อดูชุดข้อมูล iris ก่อนการเปลี่ยนแปลง



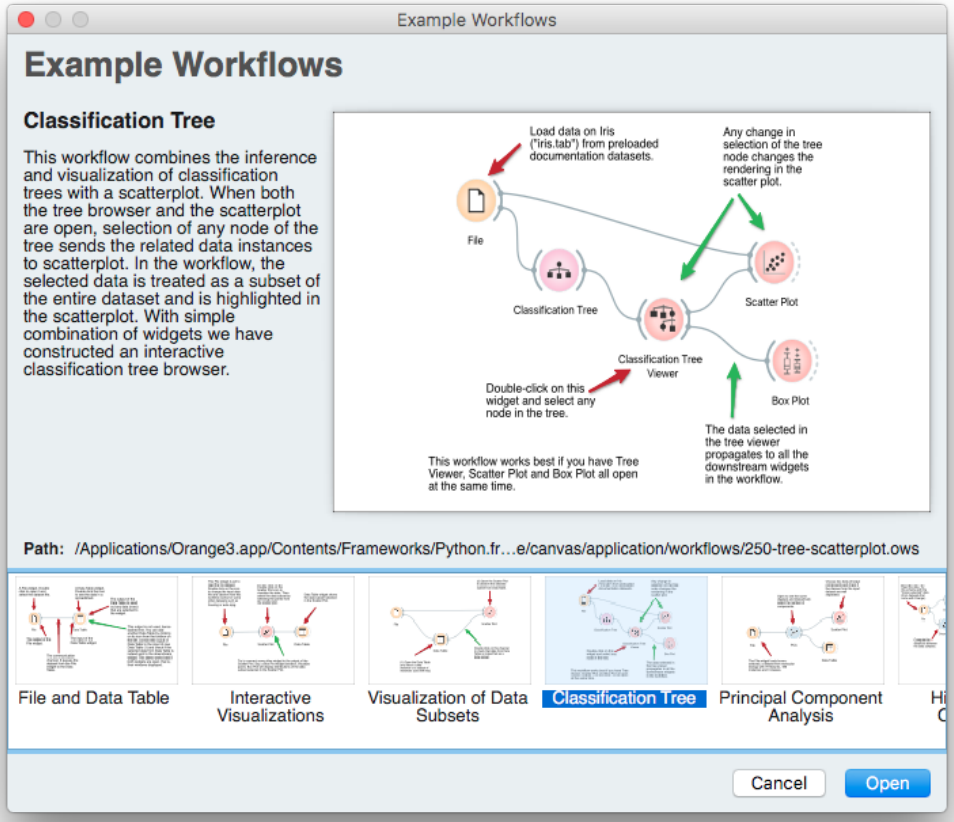
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Preprocess**



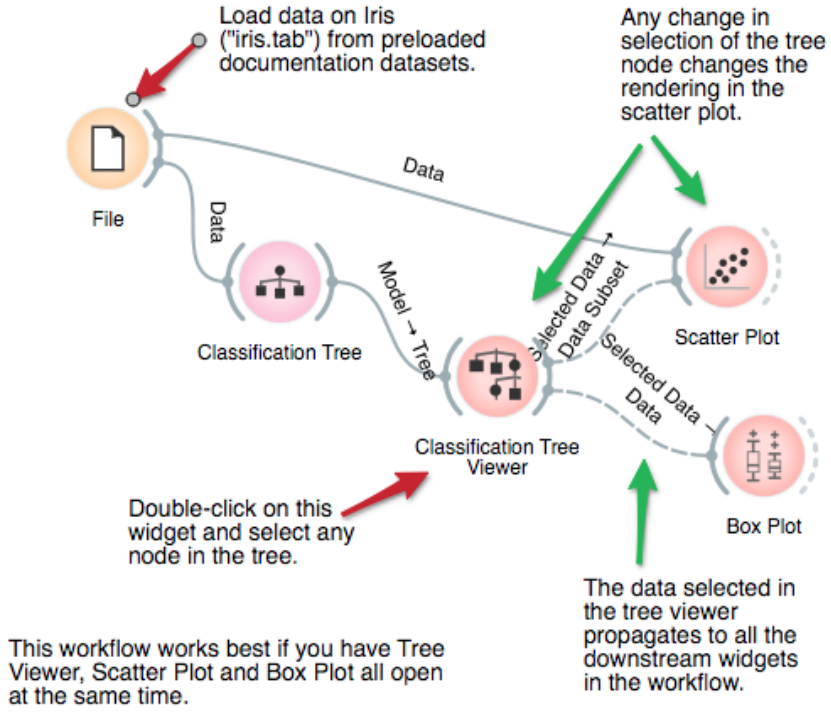
- จากตัวอย่างข้างต้น ได้เลือกใช้วิธีการ **Preprocess** ด้วยวิธีการ **Normalize** ข้อมูล
  - วิธีการเลือกคือใช้เมาส์คลิก วิธีการทางฝั่งซ้าย และนำมาวางหน้าต่างทางฝั่งขวามือ
- จากนั้นดับเบิ้ลคลิกที่ไอคอน Data Table (1) เพื่อดูการเปลี่ยนแปลงของข้อมูล

ต้นไม้จำแนก (Classification Tree)

- ในส่วนนี้แสดงให้เห็นถึงวิธีการจำแนก (Classification) ข้อมูลด้วยวิธีต้นไม้ (Tree)
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Classification Tree** และคลิกที่ปุ่ม **Open**

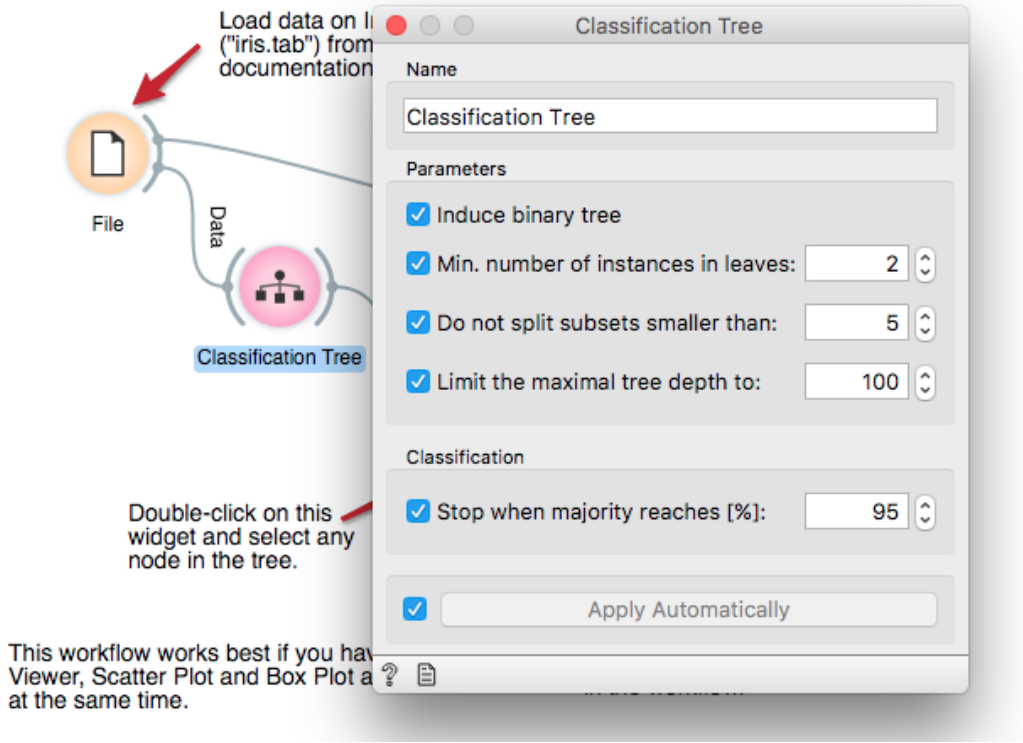


- เมื่อคลิกที่ **Open** จะปรากฏตัวอย่างการใช้งาน Classification Tree ดังตัวอย่างต่อไปนี้

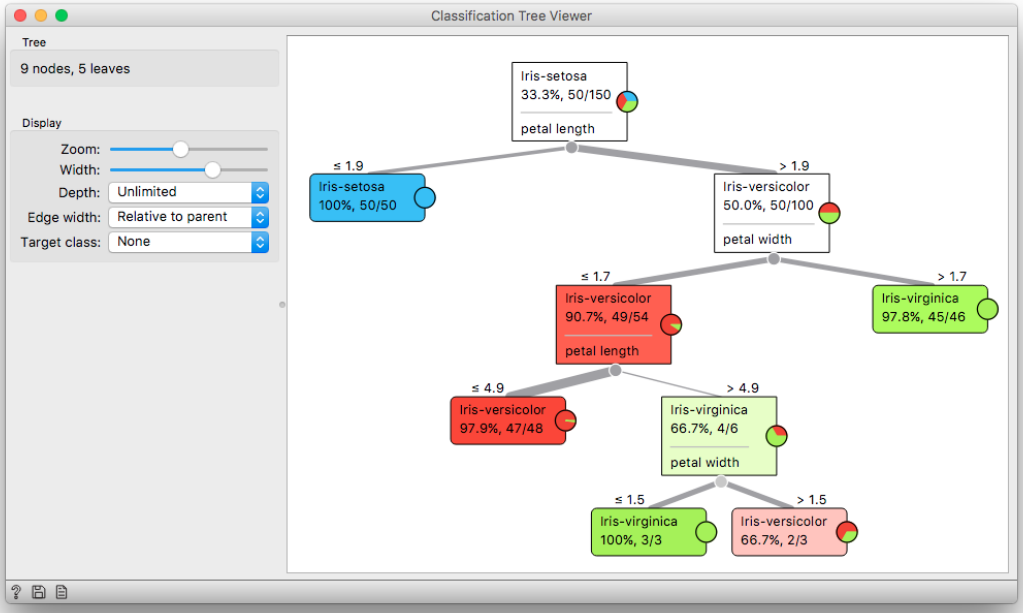


ขั้นตอนในการทำ Classification Tree ทำได้ดังต่อไปนี้

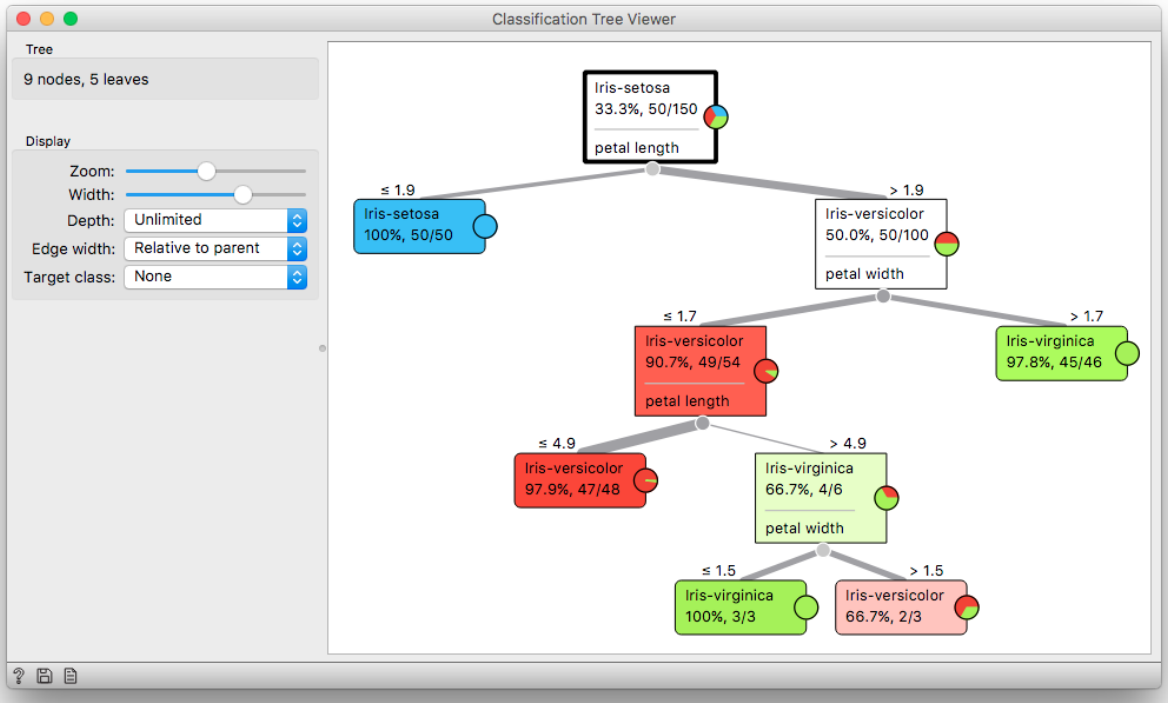
- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูลที่ต้องการใช้งาน ในตัวอย่างใช้ข้อมูล iris
- ดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูข้อมูลแบบ Visualize
- ดับเบิ้ลคลิกที่ไอคอน **Classification Tree** จะปรากฏหน้าต่าง Classification Tree เพื่อกำหนดค่า parameter



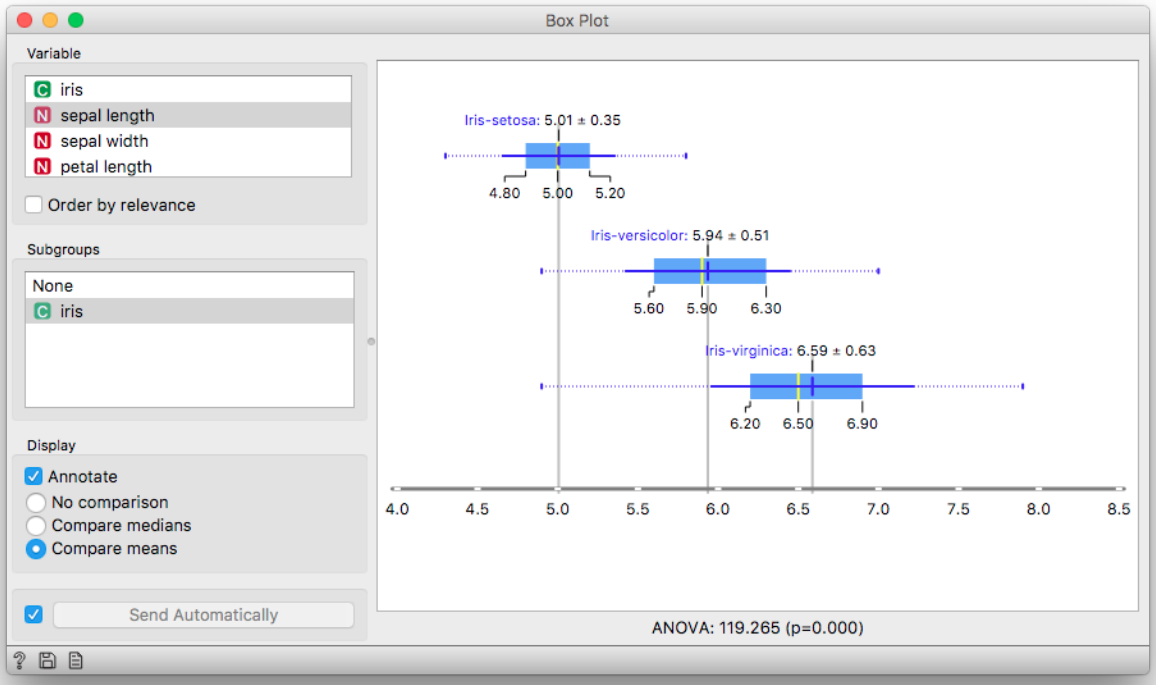
- ดับเบิ้ลคลิกที่ **Classification Tree Viewer** เพื่อดู Tree ที่คำนวณได้จากข้อมูลชุด iris



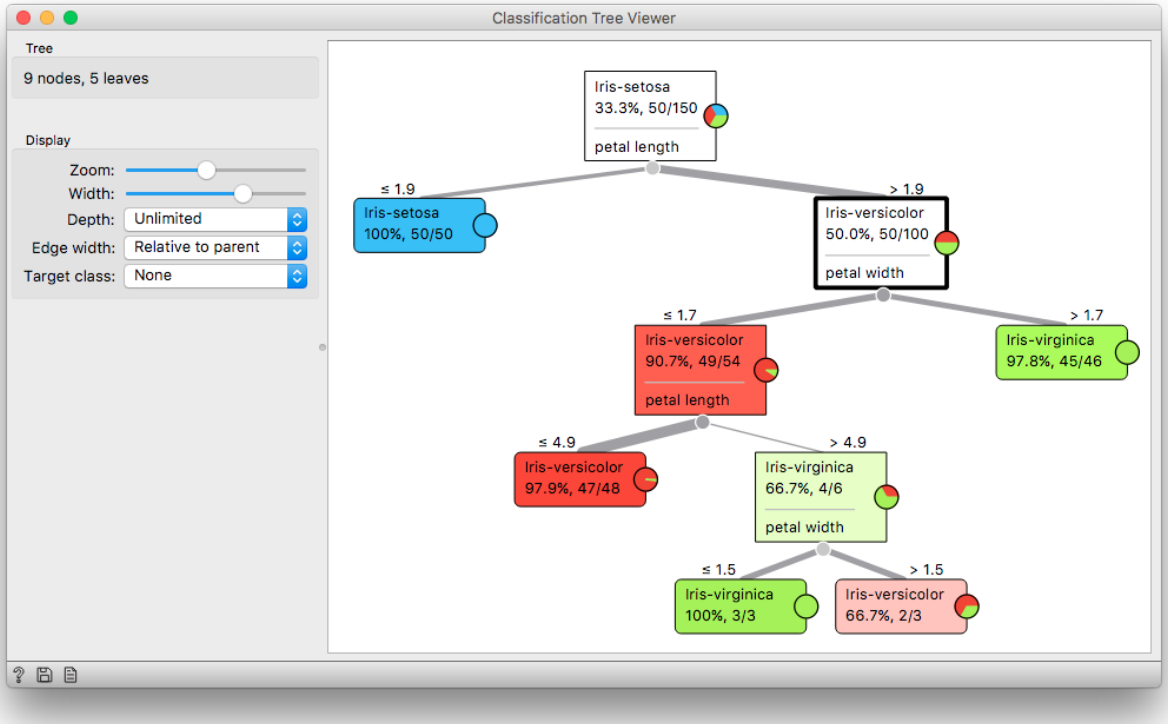
- จากหน้าต่าง **Classification Tree Viewer** ให้ทดลองเลือกโหนด (Node) จากตัวอย่างเลือกโหนดแรก



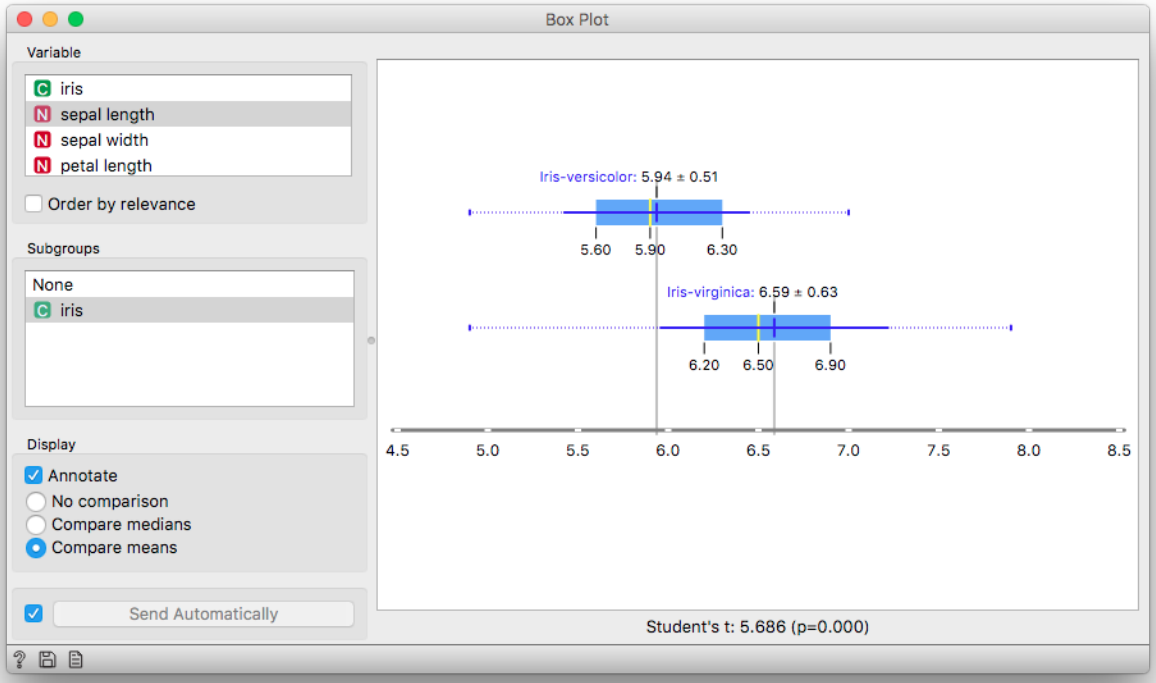
- จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน Box Plot จะแสดงหน้าจอดังต่อไปนี้



- ทดลองเลือก Node อื่น

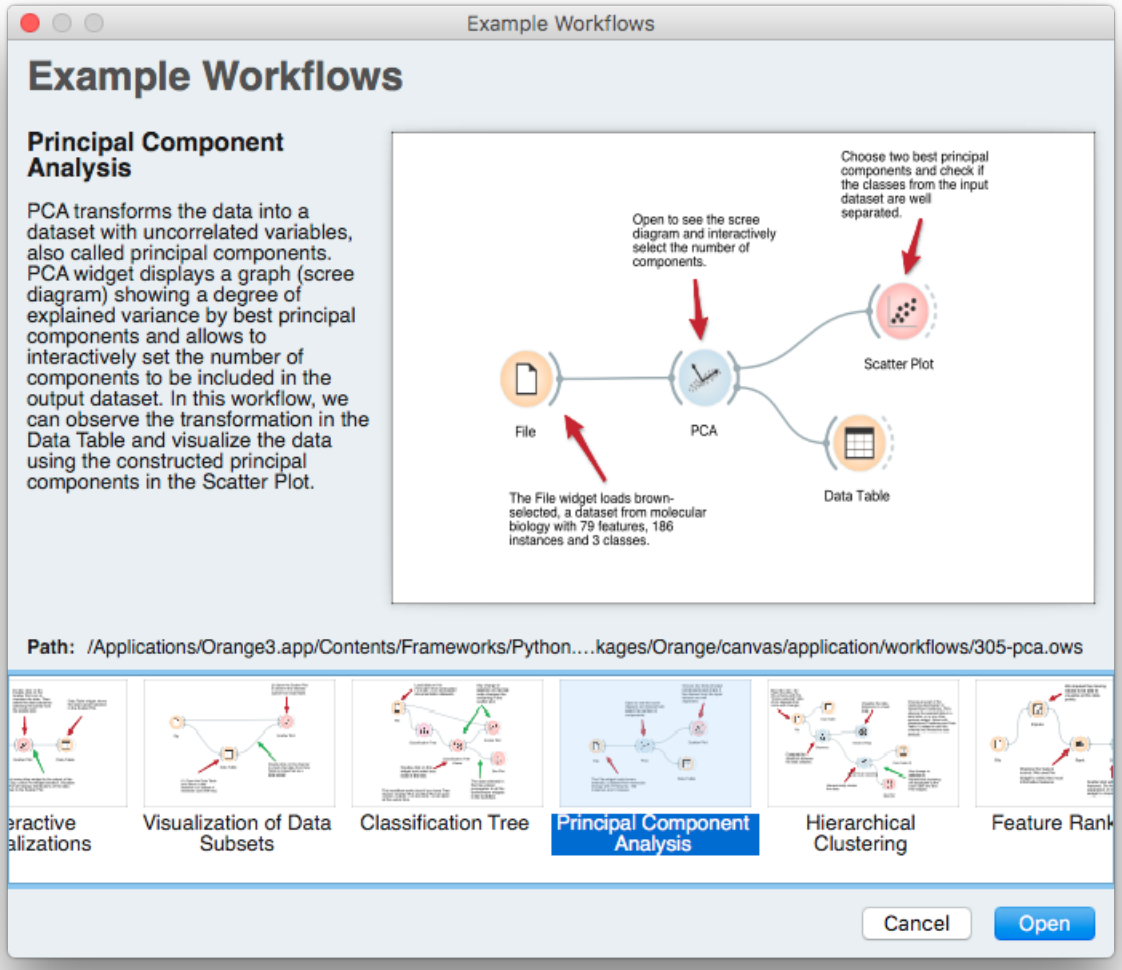


- จากนั้นเลือก Box Plot อีกครั้ง สังเกตผลลัพธ์ที่ได้จะมีความแตกต่างกัน



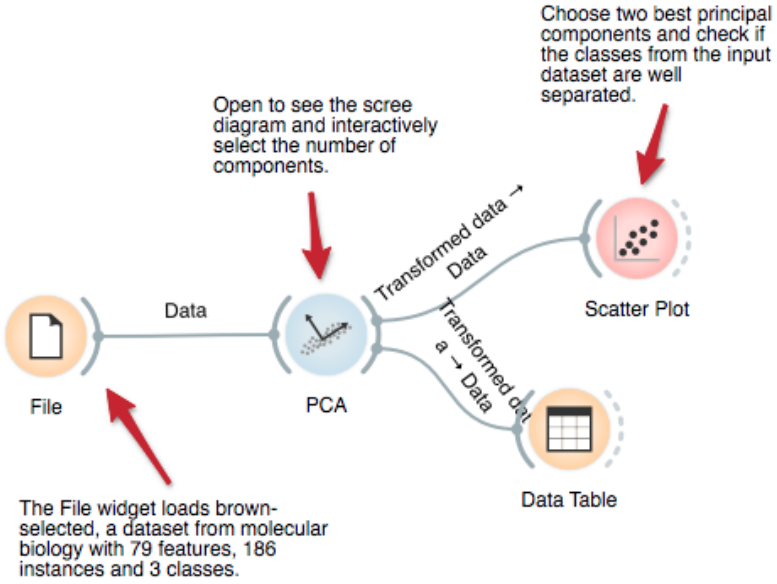
การวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA)

- วิธีการวิเคราะห์องค์ประกอบหลัก (Principal Component Analysis: PCA) เป็นวิธีในการวิเคราะห์ข้อมูลที่มีหลายตัวแปร เพื่อหาความสัมพันธ์ของตัวแปรเหล่านั้น วิธี PCA จึงนำมาเพื่อใช้ลดขนาดของเมตริกซ์ให้เล็กลง [1]
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Principal Component Analysis** และคลิกที่ปุ่ม **Open**



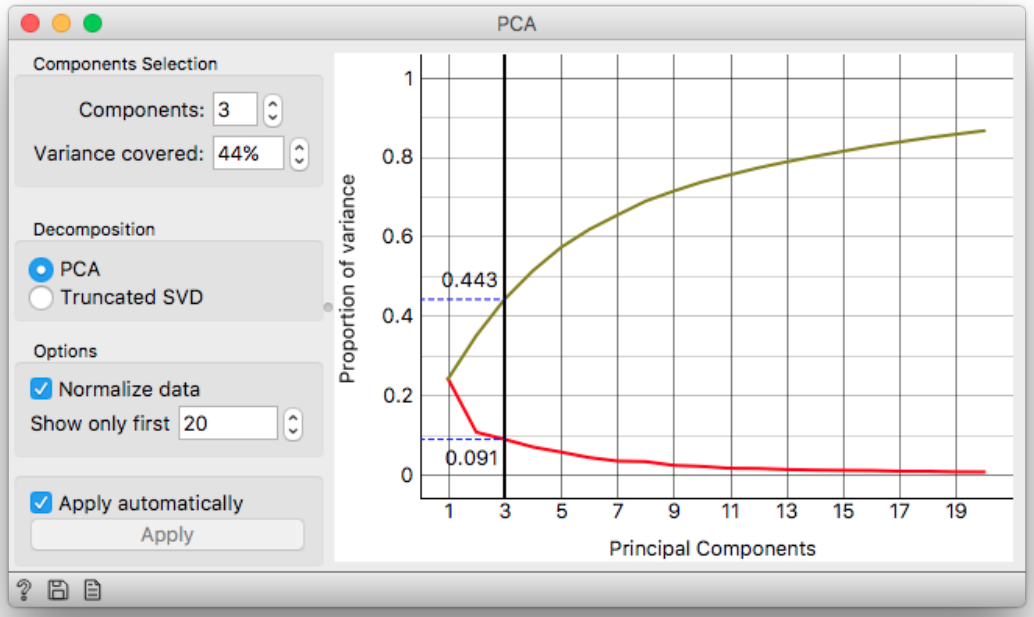
1 PCA: https://www.gotoknow.org/posts/566063

- เมื่อคลิกที่ **Open** จะปรากฏตัวอย่างการใช้งาน Principal Component Analysis ดังตัวอย่างต่อไปนี้

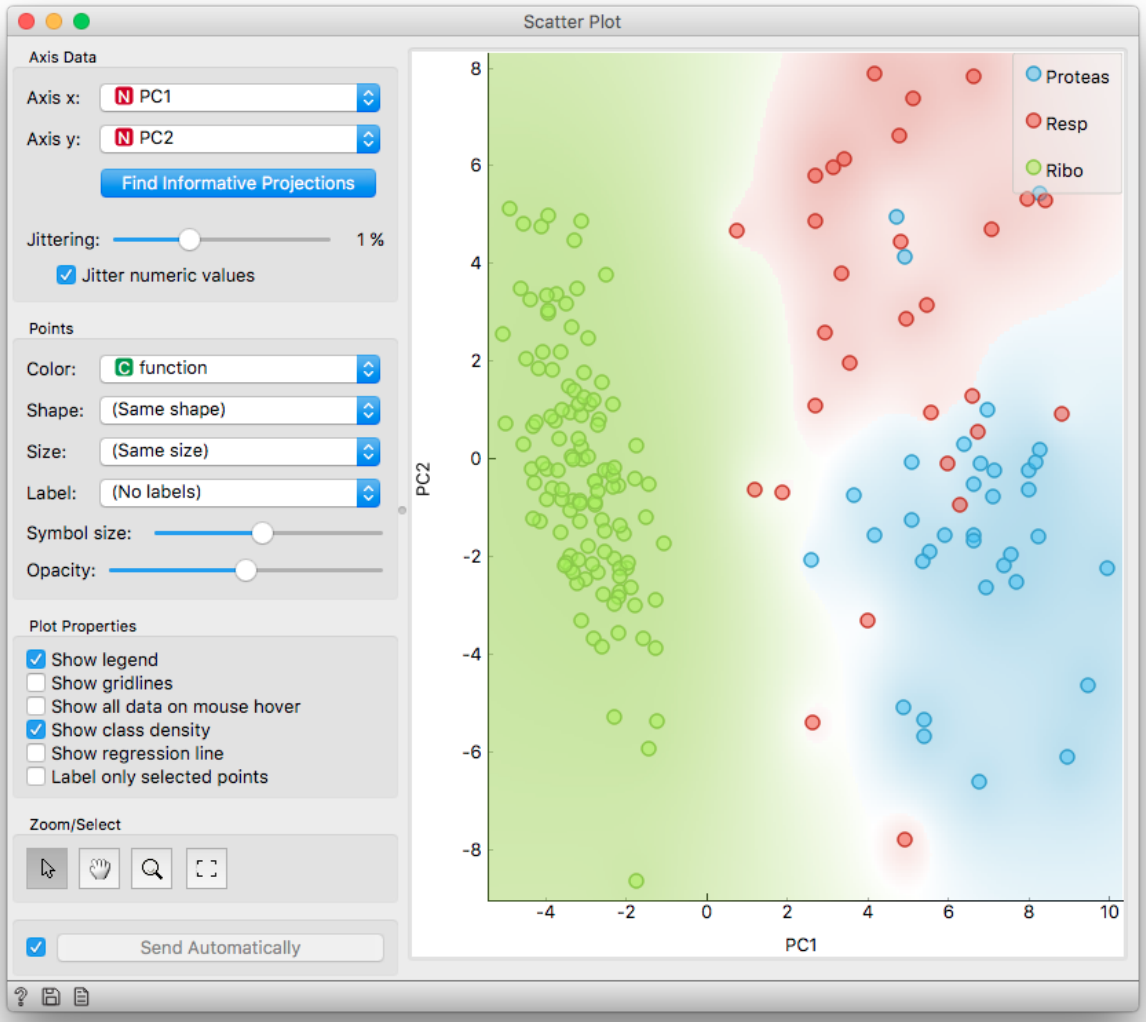


ขั้นตอนในการคำนวณ PCA ทำได้ดังต่อไปนี้

- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูลที่ต้องการใช้งาน
  - ตัวอย่างใช้ข้อมูล molecular biology ที่มีข้อมูลทั้งสิ้น 186 ชุด (Instance) แต่ละชุดมีจำนวน 79 Feature / Attribute โดยมีทั้งสิ้น 3 กลุ่ม (Class)
- ดับเบิ้ลคลิกที่ไอคอน **PCA** เพื่อแสดงการคำนวณของโปรแกรม

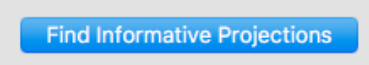


- ดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูค่า PCA จำนวน 2 ตัวแปรที่ดีที่สุด ทั้งนี้เนื่องจากใน Scatter Plot นั้นเป็นการแสดงข้อมูลแบบสองมิติ (2D)
  - ในกรณีนี้ ได้เลือกจำนวนของ Component ที่นำมาคำนวณจำนวน 3 ตัวแปร
  - โปรแกรมจะแสดง Principle Component (PC) ที่เหมาะสมที่จะนำมาแสดง จากตัวอย่างคือ PC1 และ PC2
  - ให้สังเกตที่กราฟ ข้อมูลทั้ง 3 class จะถูกจำแนกจากกันโดยใช้สีในการแบ่ง

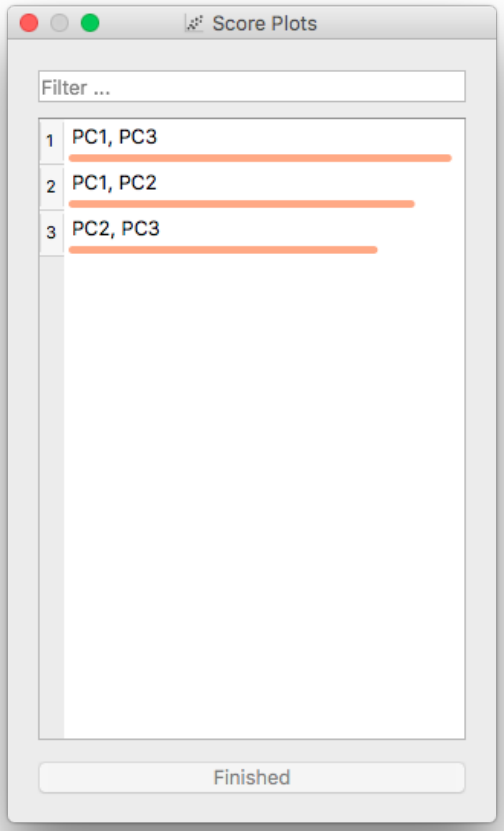


แสดง Principal Component ที่ดีที่สุด (Present Best Principal Components)

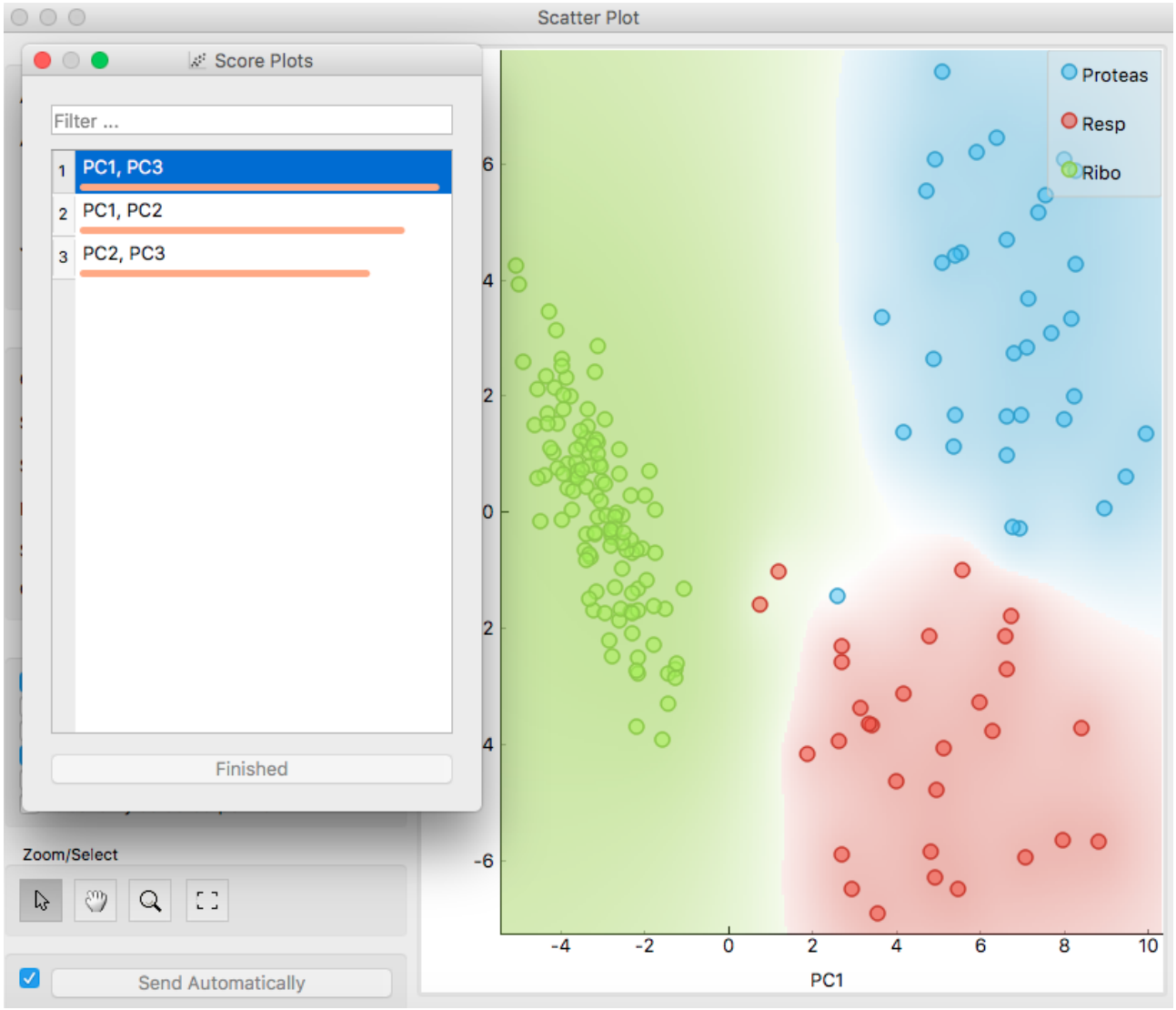
- เราสามารถเลือก Principal Component ที่ดีที่สุด (Best) ที่จะนำมาแสดงในกราฟ โดยคลิกที่ปุ่ม **Find Informative Projections**



- เมื่อคลิกที่ **Find Informative Projections** จะปรากฏหน้าต่างดังต่อไปนี้ จากนั้นให้คลิกที่ปุ่ม **Start** โปรแกรมจะคำนวณและหา PC จำนวน 2 ค่าที่เหมาะสมที่สุด (Best) มาแสดง

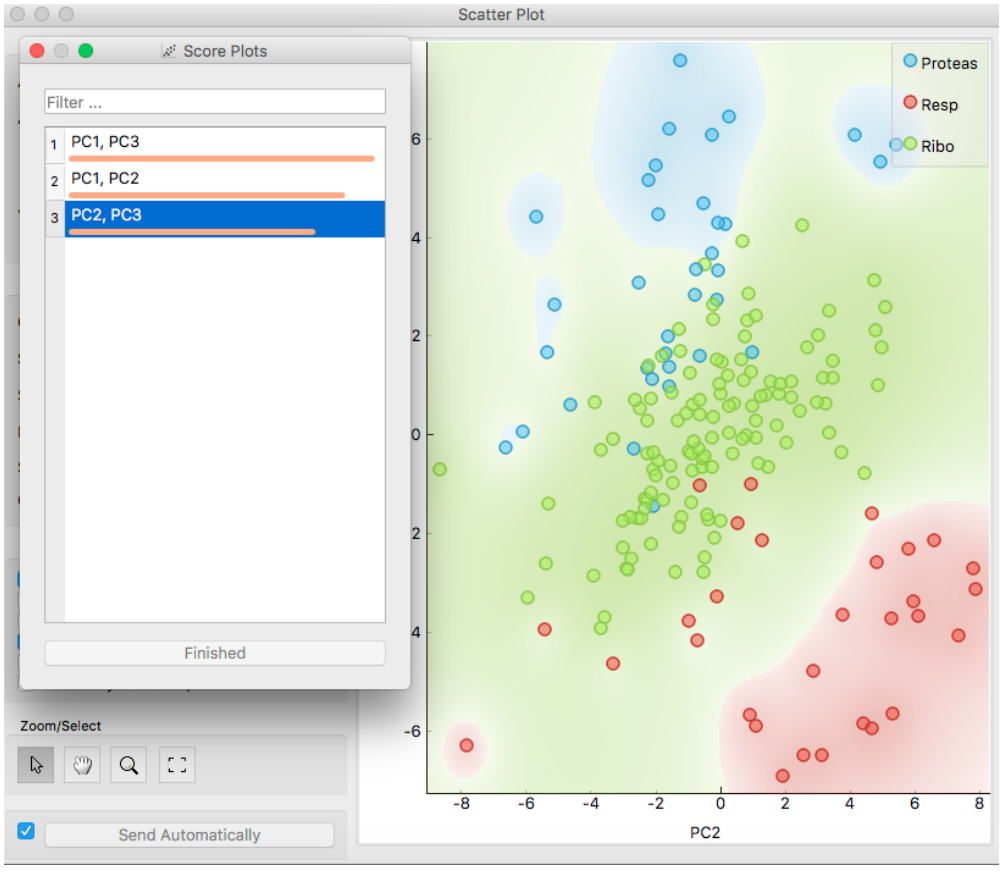


- จากตัวอย่างข้างต้น โปรแกรมจะคำนวณและแสดง PC ที่มีความเหมาะสมที่สุดจำนวน 2 ค่า โดยเรียงตามลำดับความเหมาะสมจากมากไปน้อย ดังนั้น เราสามารถเลือกค่าที่เหมาะสมที่สุดเพื่อตรวจสอบการจำแนกข้อมูล ดังตัวอย่างต่อไปนี้



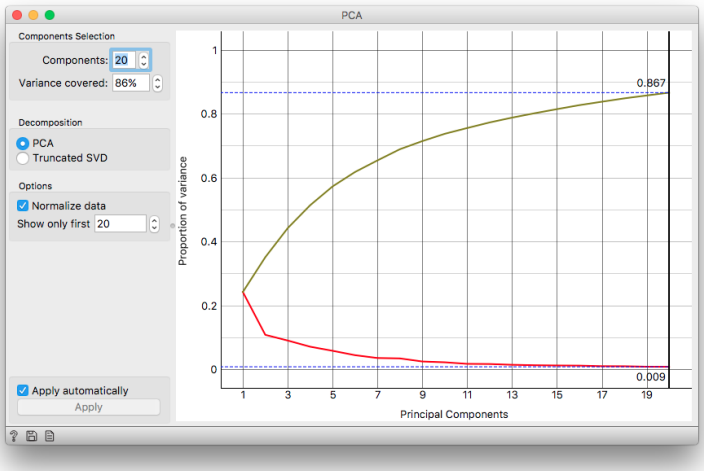
- จากตัวอย่างข้างต้นยังแสดงให้เห็นถึงความผิดพลาดในการจำแนกข้อมูล เช่น จุดสีแดงจำนวน 2 จุด และจุดสีฟ้าจำนวน 1 จุดที่จำแนกผิด


Scatter Plot
Score Plots
Filter ...
1 PC1, PC3
2 PC1, PC2
3 PC2, PC3
Finished
Zoom/Select
Send Automatically
Proteas
Resp
Ribo
PC1


Scatter Plot
Score Plots
Filter ...
1 PC1, PC3
2 PC1, PC2
3 PC2, PC3
Finished
Zoom/Select
Send Automatically
Proteas
Resp
Ribo
PC2

การเลือกจำนวน Principal Component ที่ใช้ในการคำนวณ (Selecting the Number of Principal Components)

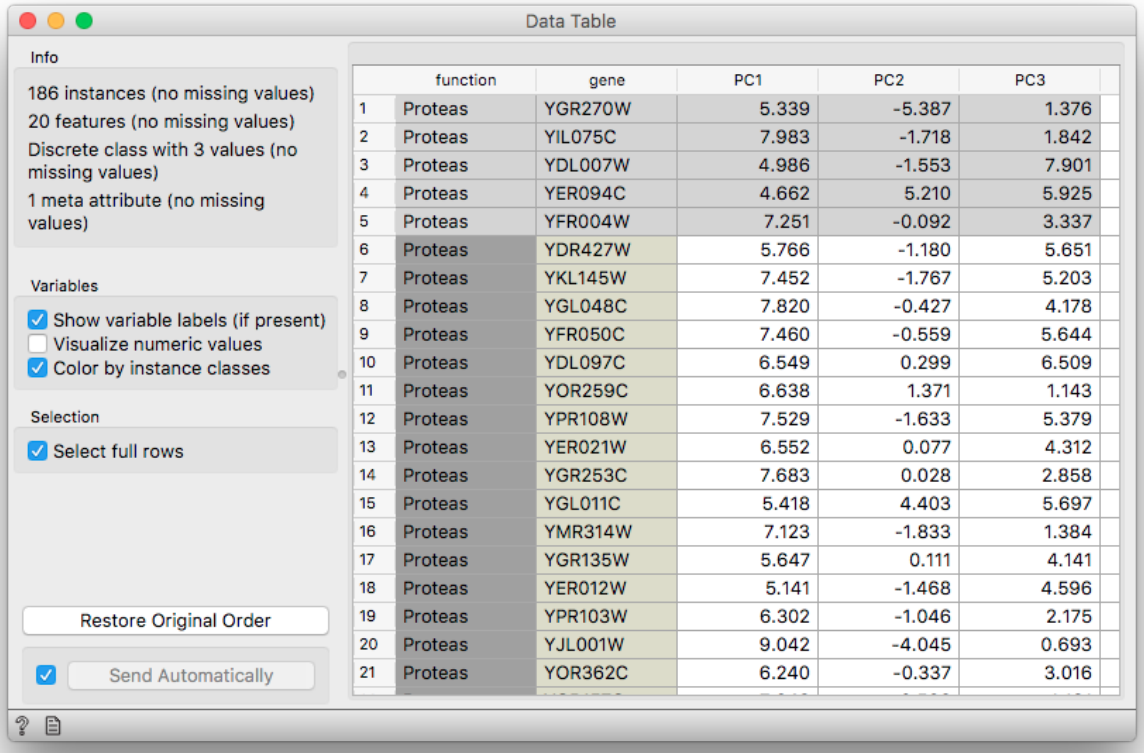
- ตัวอย่างต่อไปนี้แสดงให้เห็นถึงการนำ PC จำนวนทั้งสิ้น 20 Component มาคำนวณ



- ผลลัพธ์ที่ได้จากการคำนวณหาค่าความสัมพันธ์ระหว่าง PC แสดงดังตัวอย่างต่อไปนี้

การเรียกดูค่าของ Principal Component (Show Value of Principal Components)

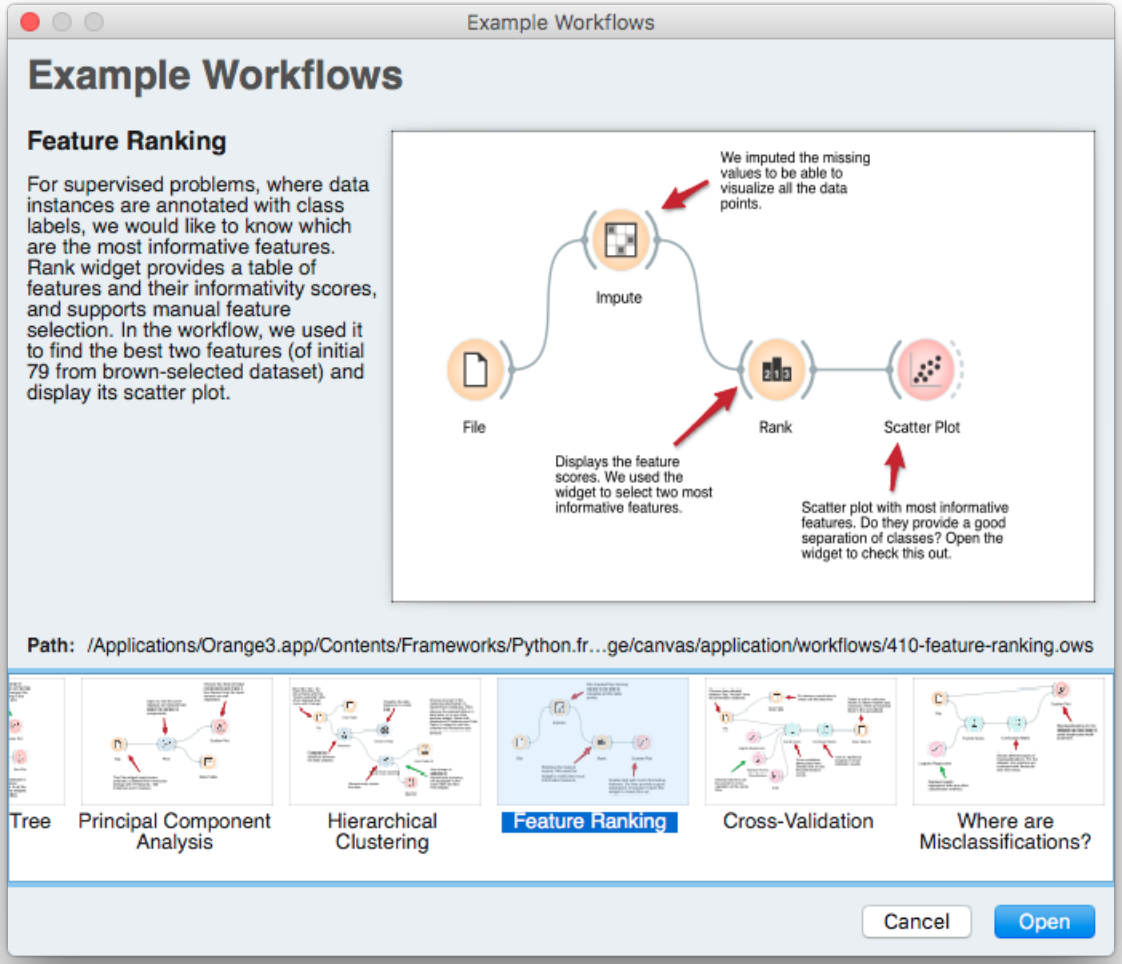
- ให้เลือกที่ไอคอน Data Table เพื่อดูค่า PC ที่ได้จากการคำนวณ
  - ในกรณีที่เลือก PC จำนวน 20 Component ตารางที่แสดงก็จะแสดงข้อมูล PC1 ถึง PC20
  - ในกรณีที่เลือก PC จำนวน 3 Component ตารางที่แสดงก็จะแสดงข้อมูล PC1 ถึง PC3
- ข้อมูล PC ทั้งหมดที่เลือกสามารถนำไปใช้ในการจำแนกข้อมูลด้วยวิธีอื่น ๆ เช่น K-Nearest Neighbor, Neural Network หรือ Support Vector Machine



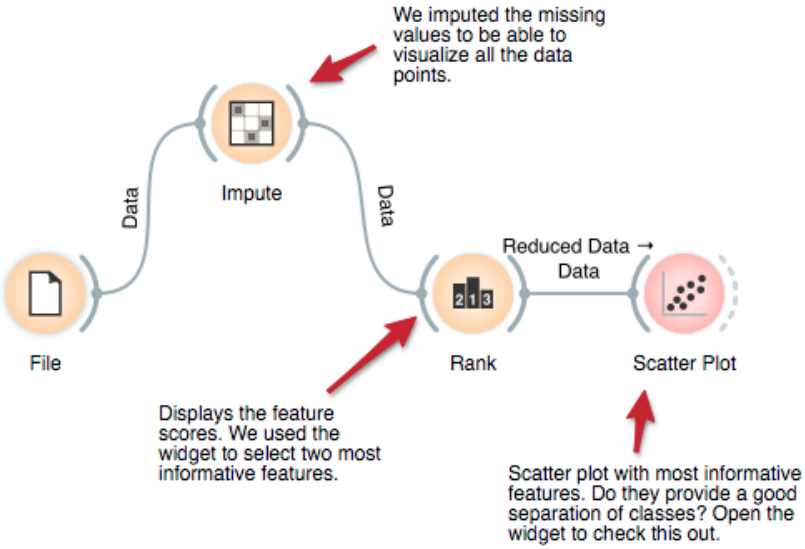
	function	gene	PC1	PC2	PC3
1	Proteas	YGR270W	5.339	-5.387	1.376
2	Proteas	YIL075C	7.983	-1.718	1.842
3	Proteas	YDL007W	4.986	-1.553	7.901
4	Proteas	YER094C	4.662	5.210	5.925
5	Proteas	YFR004W	7.251	-0.092	3.337
6	Proteas	YDR427W	5.766	-1.180	5.651
7	Proteas	YKL145W	7.452	-1.767	5.203
8	Proteas	YGL048C	7.820	-0.427	4.178
9	Proteas	YFR050C	7.460	-0.559	5.644
10	Proteas	YDL097C	6.549	0.299	6.509
11	Proteas	YOR259C	6.638	1.371	1.143
12	Proteas	YPR108W	7.529	-1.633	5.379
13	Proteas	YER021W	6.552	0.077	4.312
14	Proteas	YGR253C	7.683	0.028	2.858
15	Proteas	YGL011C	5.418	4.403	5.697
16	Proteas	YMR314W	7.123	-1.833	1.384
17	Proteas	YGR135W	5.647	0.111	4.141
18	Proteas	YER012W	5.141	-1.468	4.596
19	Proteas	YPR103W	6.302	-1.046	2.175
20	Proteas	YJL001W	9.042	-4.045	0.693
21	Proteas	YOR362C	6.240	-0.337	3.016

การเรียงตัวแปรตามลำดับความสำคัญ (Feature Ranking)

- ในส่วนนี้แสดงให้เห็นถึงวิธีการคำนวณเพื่อหา Feature หรือ Attribute ที่มีความสำคัญมากที่สุด (Rank) โดยคำนวณจากความสัมพันธ์ (Correlation) ของแต่ละ Feature
- ข้อมูลที่ใช้ในการทดสอบชื่อ Brown เป็นชุดข้อมูลยีน (Gene) ที่ประกอบด้วย 186 instance โดยแต่ละ instance จะมีทั้งสิ้น 79 feature และข้อมูลแบ่งออกเป็น 3 class
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Feature Ranking** และคลิกที่ปุ่ม **Open**



- เมื่อคลิกที่ Open จะปรากฏหน้าจอดังต่อไปนี้

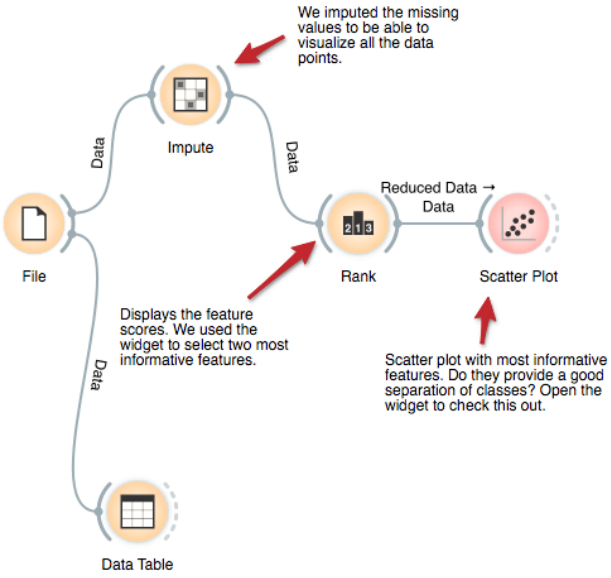


ขั้นตอนในการคำนวณ Feature Ranking ทำได้ดังต่อไปนี้

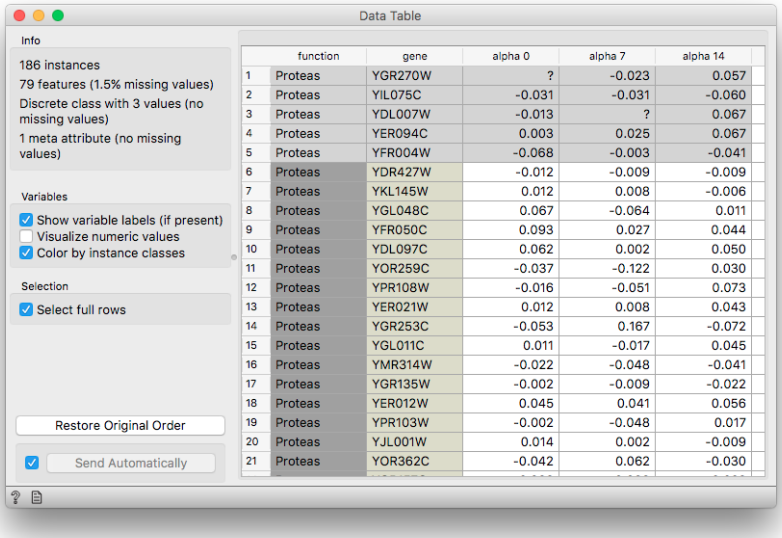
- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูลที่ต้องการใช้งาน ในกรณีนี้ใช้ข้อมูล Brown Selected โดยข้อมูลชุดนี้จะมีบางชุดที่ข้อมูลสูญหาย (Missing Value) ไป ดังนั้นจึงต้องคำนวณข้อมูลใหม่ เพื่อเติมลงไปในส่วนที่ข้อมูลสูญหาย

การเพิ่มไอคอน Data Table (Add Data Table Icon)

- ในกรณีที่ต้องการดูว่ามี instance ไหนบ้างที่ข้อมูลสูญหายไปสามารถเพิ่มไอคอน **Data Table** ลงไปใน workflow

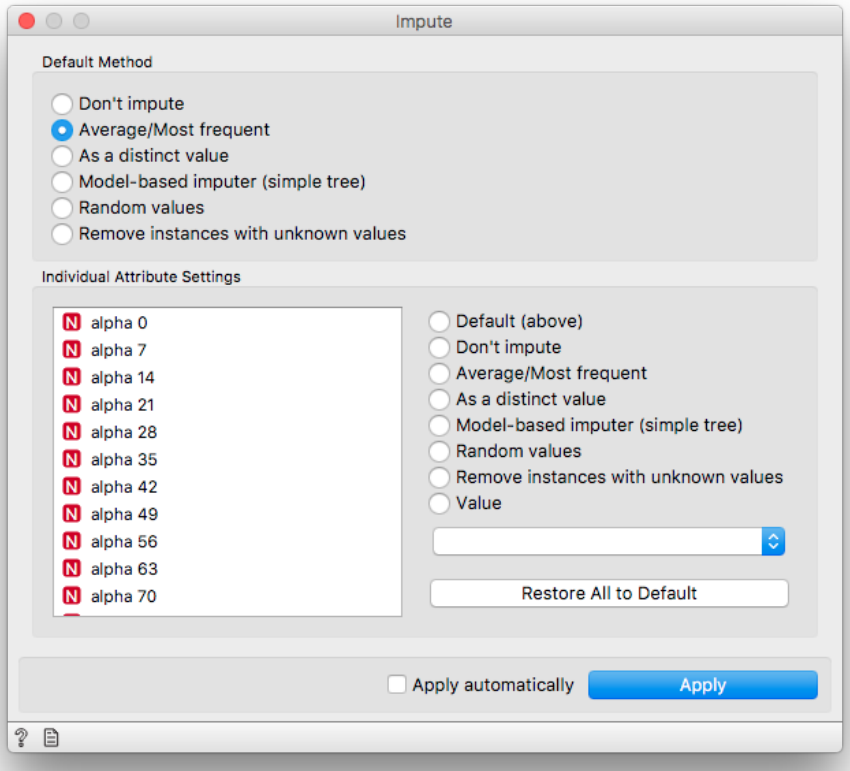


- คลิกที่ไอคอน **Data Table** เพื่อเปิดดูข้อมูล
  - จากตัวอย่างข้อมูลที่สูญหายไปจะแทนที่ด้วยเครื่องหมายคำถาม "?"
  - ในข้อมูลชุดนี้มีข้อมูลสูญหายไป 1.5%

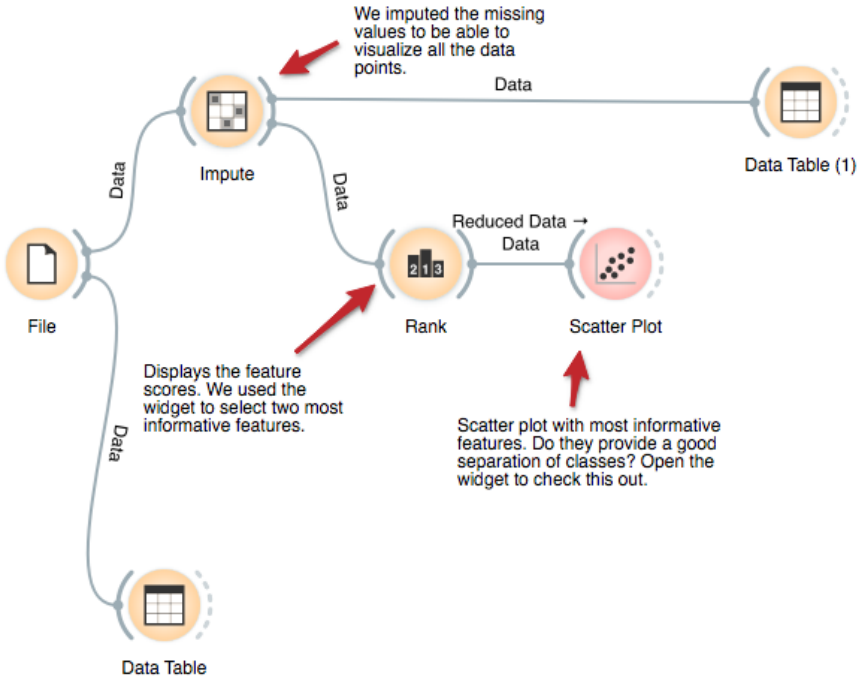


	function	gene	alpha 0	alpha 7	alpha 14
1	Proteas	YGR270W	?	-0.023	0.057
2	Proteas	YIL075C	-0.031	-0.031	-0.060
3	Proteas	YDL007W	-0.013	?	0.067
4	Proteas	YER094C	0.003	0.025	0.067
5	Proteas	YFR004W	-0.068	-0.003	-0.041
6	Proteas	YDR427W	-0.012	-0.009	-0.009
7	Proteas	YKL145W	0.012	0.008	-0.006
8	Proteas	YGL048C	0.067	-0.064	0.011
9	Proteas	YFR050C	0.093	0.027	0.044
10	Proteas	YDL097C	0.062	0.002	0.050
11	Proteas	YOR259C	-0.037	-0.122	0.030
12	Proteas	YPR108W	-0.016	-0.051	0.073
13	Proteas	YER021W	0.012	0.008	0.043
14	Proteas	YGR253C	-0.053	0.167	-0.072
15	Proteas	YGL011C	0.011	-0.017	0.045
16	Proteas	YMR314W	-0.022	-0.048	-0.041
17	Proteas	YGR135W	-0.002	-0.009	-0.022
18	Proteas	YER012W	0.045	0.041	0.056
19	Proteas	YPR103W	-0.002	-0.048	0.017
20	Proteas	YJL001W	0.014	0.002	-0.009
21	Proteas	YOR362C	-0.042	0.062	-0.030

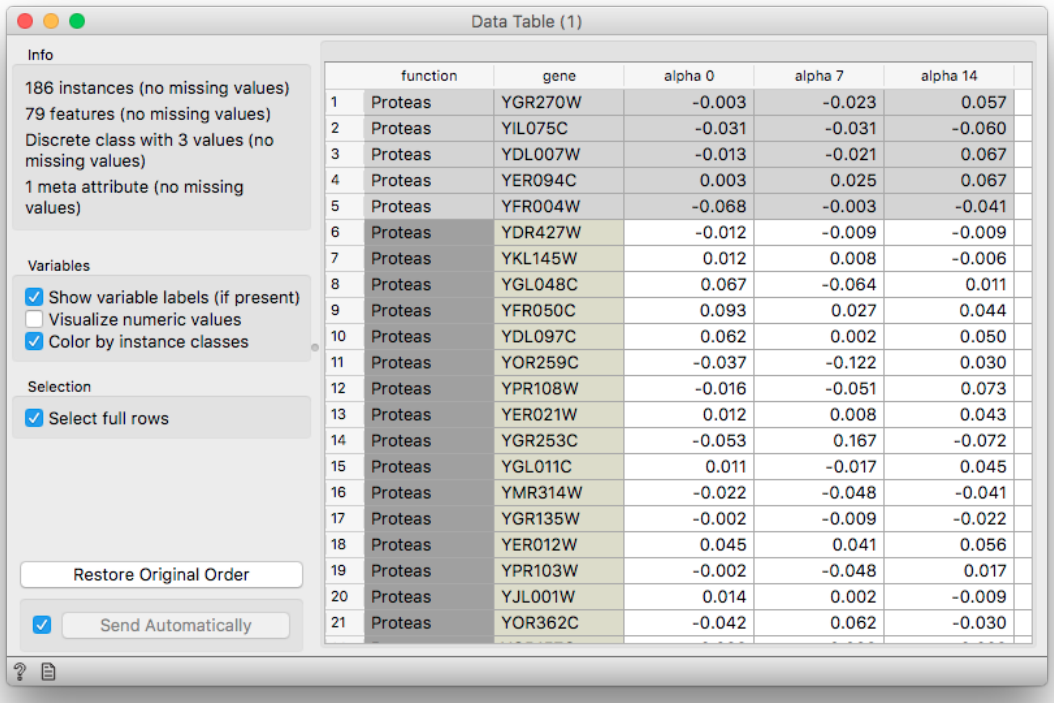
- ดับเบิ้ลคลิกที่ไอคอน **Impute** เพื่อแทนค่าข้อมูลที่สูญหายไป โดย Default Method คือวิธี Average/Most frequent และให้คลิกที่ปุ่ม **Apply** เพื่อคำนวณและแทนค่าข้อมูลที่สูญหาย



- หากต้องการที่จะแสดงข้อมูลให้เพิ่มไอคอน **Data Table** ลงไปใน workflow และลากเส้นเชื่อมต่อระหว่าง **Impute** และ **Data Table (1)**

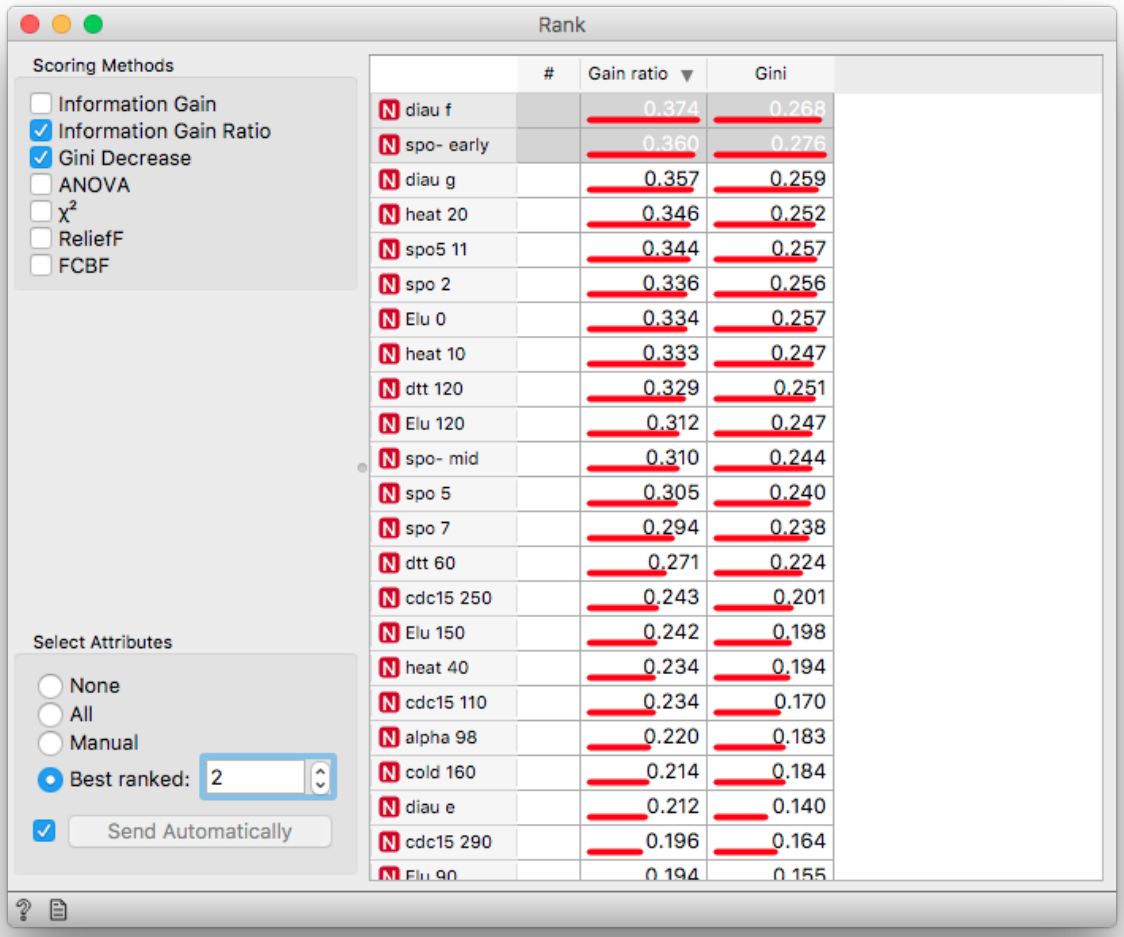


- ดับเบิ้ลคลิกที่ **Data Table (1)** ตัวที่เพิ่มลงไปใหม่ และสังเกตที่ข้อมูล ข้อมูลที่สูญหายจะถูกแทนที่ด้วยข้อมูลใหม่

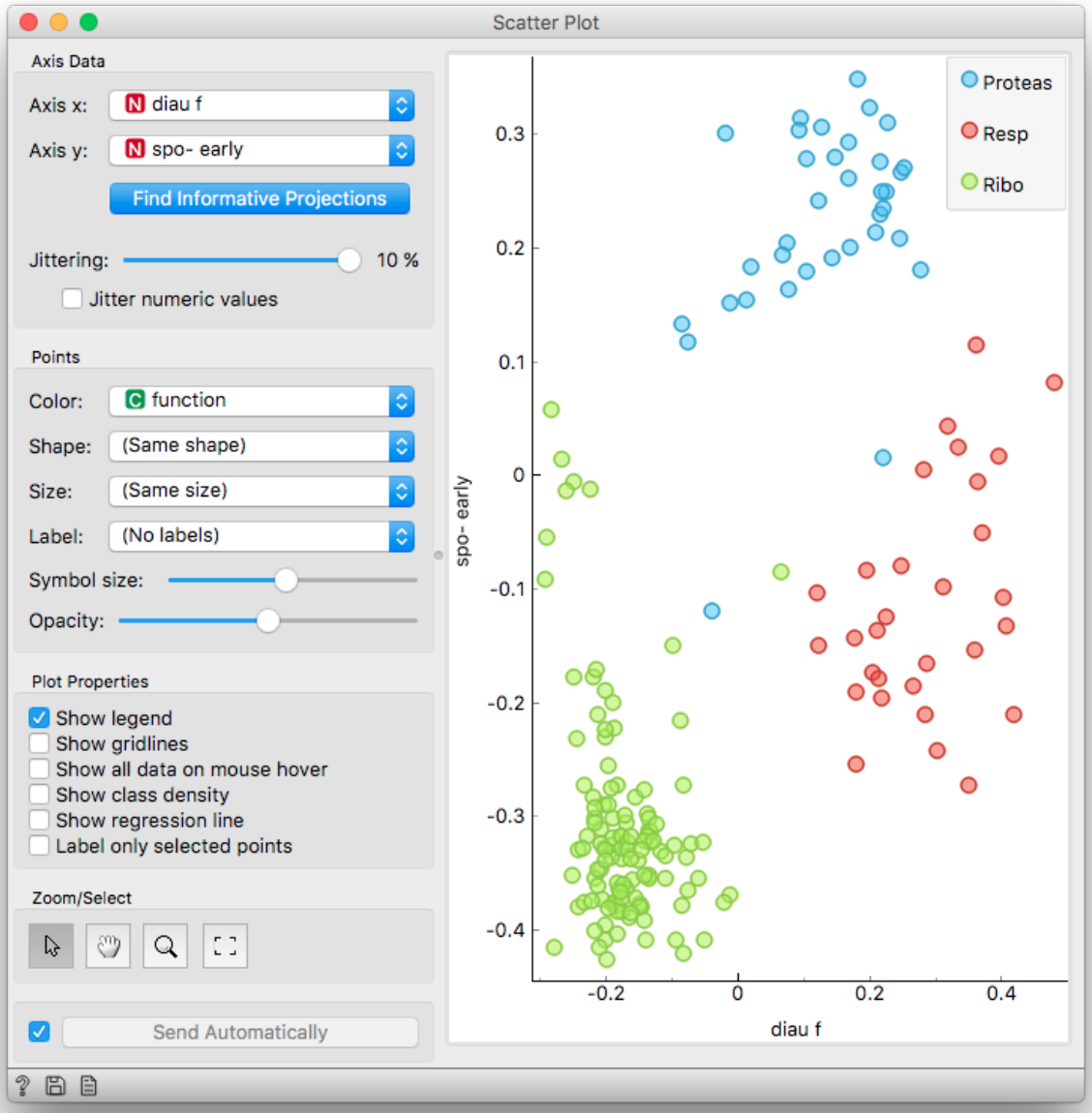


	function	gene	alpha 0	alpha 7	alpha 14
1	Proteas	YGR270W	-0.003	-0.023	0.057
2	Proteas	YIL075C	-0.031	-0.031	-0.060
3	Proteas	YDL007W	-0.013	-0.021	0.067
4	Proteas	YER094C	0.003	0.025	0.067
5	Proteas	YFR004W	-0.068	-0.003	-0.041
6	Proteas	YDR427W	-0.012	-0.009	-0.009
7	Proteas	YKL145W	0.012	0.008	-0.006
8	Proteas	YGL048C	0.067	-0.064	0.011
9	Proteas	YFR050C	0.093	0.027	0.044
10	Proteas	YDL097C	0.062	0.002	0.050
11	Proteas	YOR259C	-0.037	-0.122	0.030
12	Proteas	YPR108W	-0.016	-0.051	0.073
13	Proteas	YER021W	0.012	0.008	0.043
14	Proteas	YGR253C	-0.053	0.167	-0.072
15	Proteas	YGL011C	0.011	-0.017	0.045
16	Proteas	YMR314W	-0.022	-0.048	-0.041
17	Proteas	YGR135W	-0.002	-0.009	-0.022
18	Proteas	YER012W	0.045	0.041	0.056
19	Proteas	YPR103W	-0.002	-0.048	0.017
20	Proteas	YJL001W	0.014	0.002	-0.009
21	Proteas	YOR362C	-0.042	0.062	-0.030

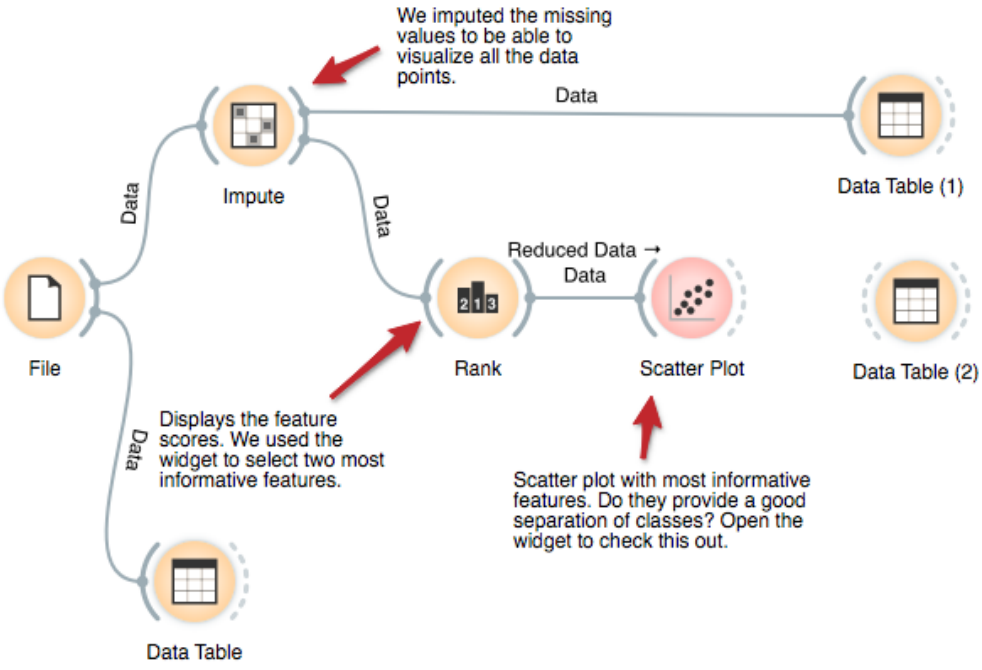
- จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน **Rank** เพื่อแสดงลำดับของ Feature ที่มี Score มากที่สุด โดยโปรแกรมจะแสดงจำนวน 2 ลำดับ
- โดย Feature ที่มี Score มากที่สุด 2 ลำดับแรกคือ diau f, spo-early



- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูการจำแนกข้อมูลโดยเป็นการนำ Feature ที่เลือกไว้ทั้ง 2 ตัว ประกอบด้วย diau f และ spo-early มาแสดงเป็น Visualize

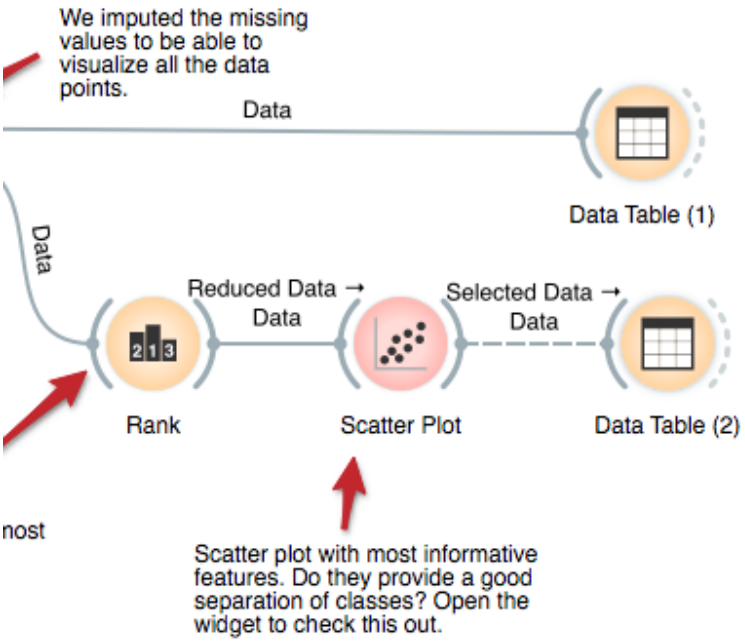


- หากต้องการนำ Feature ที่มี Score มากที่สุดไปใช้งานต่อในการจำแนกข้อมูลสามารถทำได้โดยเพิ่มไอคอม **Data Table** เข้าไปใน workflow
  - ตัวอย่างคือการเพิ่ม **Data Table (2)**

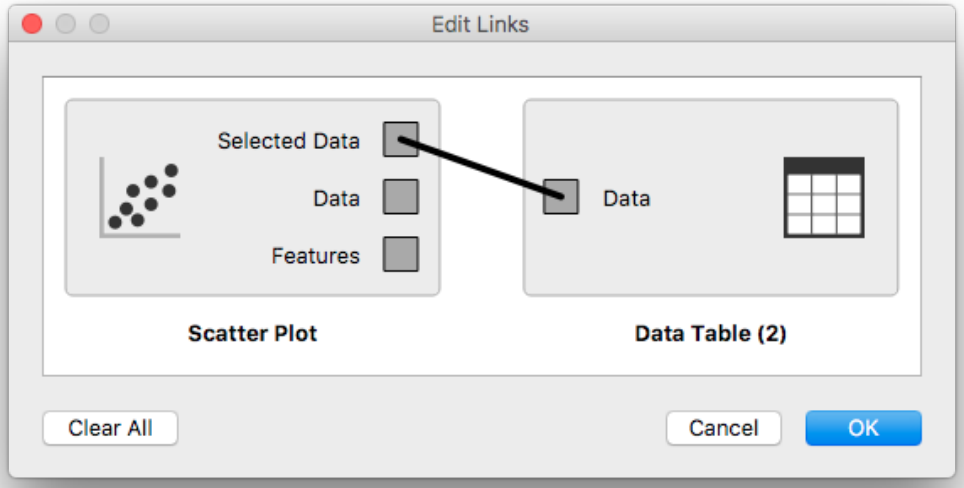


การสร้างการเชื่อมต่อเส้นระหว่างไอคอน (Create New Link to Icons)

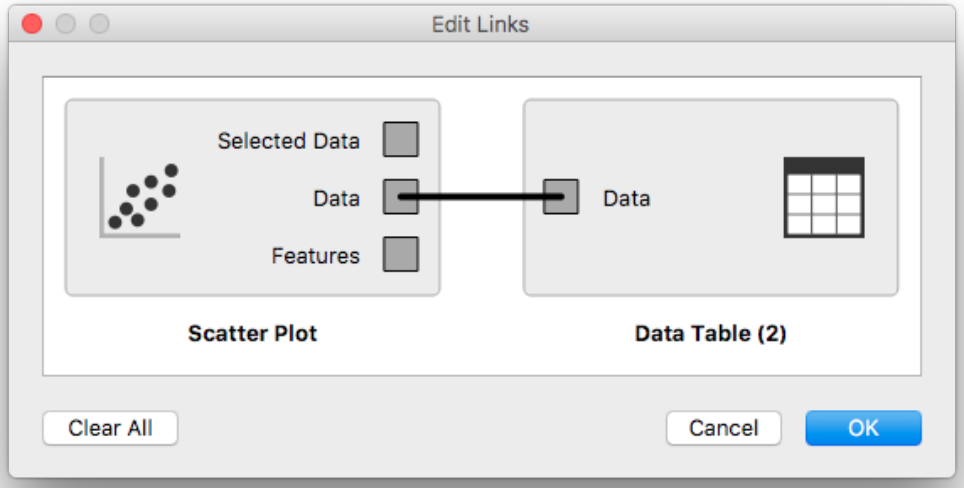
- ขั้นตอนถัดไปคือการลากเส้นเชื่อมต่อระหว่าง **Scatter Plot** และ **Data Table (2)**
  - เมื่อลากเส้นเชื่อมกันแล้วจะปรากฏ **เส้นประ** แสดงว่ายังเชื่อมข้อมูลไม่สำเร็จ



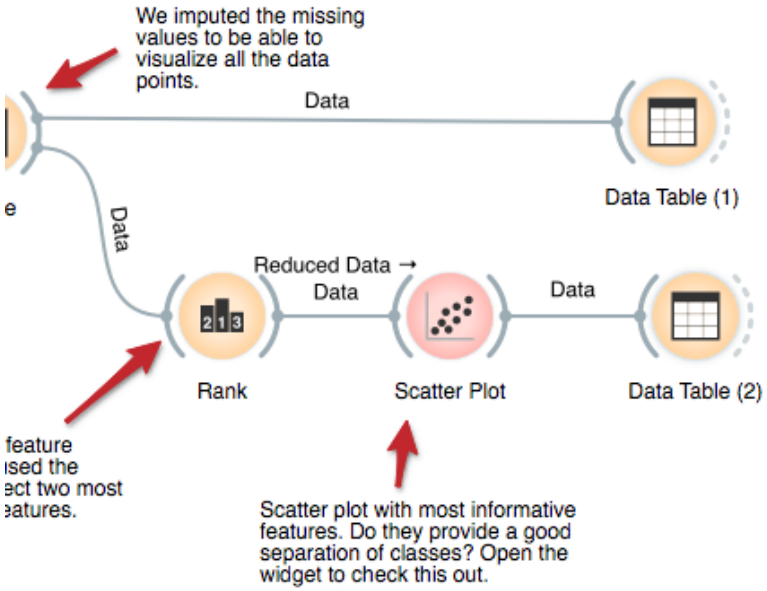
- ขั้นต่อไป ให้ดับเบิ้ลคลิกที่เส้นประ จะปรากฏดังหน้าต่างต่อไปนี้



- โดยจะเป็นการเชื่อมระหว่าง **Selected Data** และ **Data** ดังนั้น ให้ลบเส้นนี้ออกและเชื่อมใหม่เป็น **Data** และ **Data**



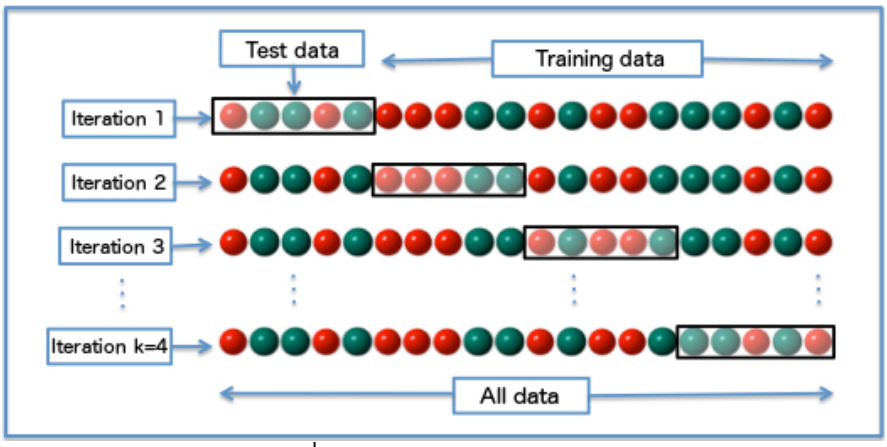
- เมื่อเลือกเสร็จแล้วให้กดปุ่ม **OK** เส้นประจะหายไปเปลี่ยนเป็น **เส้นทึบ**



- จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน **Data Table (2)** เพื่อเปิดดูข้อมูล
  - จากทำกำหนดในไอคอน Rank ได้เลือกไว้จำนวน 2 Feature ดังนั้นในตารางที่เปิดมาจึงประกอบไปด้วยข้อมูลเฉพาะ 2 Feature เท่านั้น ซึ่งก็คือ diau f และ spo-early

การแบ่งข้อมูลเพื่อทดสอบประสิทธิภาพของโมเดล (Cross-Validation)

- Cross-Validation คือการแบ่งข้อมูลออกเป็น 2 ชุด ประกอบด้วย ชุดเรียนรู้ (Training Set) และชุดทดสอบ (Test Set) เพื่อนำข้อมูลชุดเรียนรู้ไปใช้ในการสร้างโมเดล และใช้ข้อมูลชุดทดสอบเพื่อทดสอบประสิทธิภาพของโมเดลที่ได้สร้าง
- การทำ Cross-Validation ข้อมูลจะถูกนำมาแบ่งเพื่อทดสอบประสิทธิภาพของโมเดล โดยกำหนดให้ทดสอบจำนวน k-fold เช่น หากกำหนดให้ k=3 ก็จะทำการทดสอบจำนวน 3 ครั้ง และในแต่ละครั้งที่ทดสอบข้อมูลก็จะถูกแบ่งออกเป็น 3 ส่วนเท่ากัน จากนั้น
  - ข้อมูล K-1 ส่วน (3-1 = 2) จะถูกนำไปเป็น Training Set เพื่อสร้างโมเดล
  - ข้อมูล 1 ส่วนจะถูกนำไปเป็น Test Set เพื่อทดสอบประสิทธิภาพ

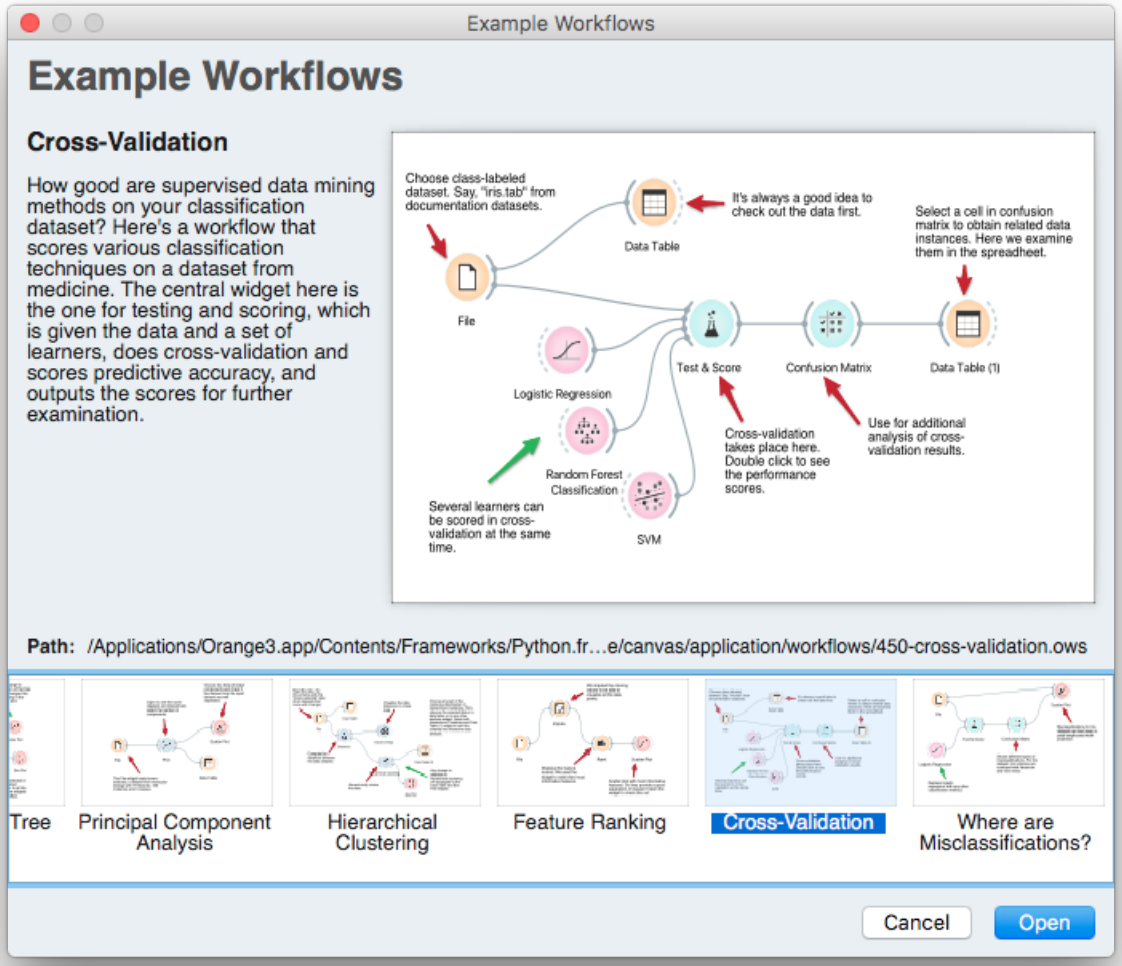


[ที่มาของรูป: wikipedia]

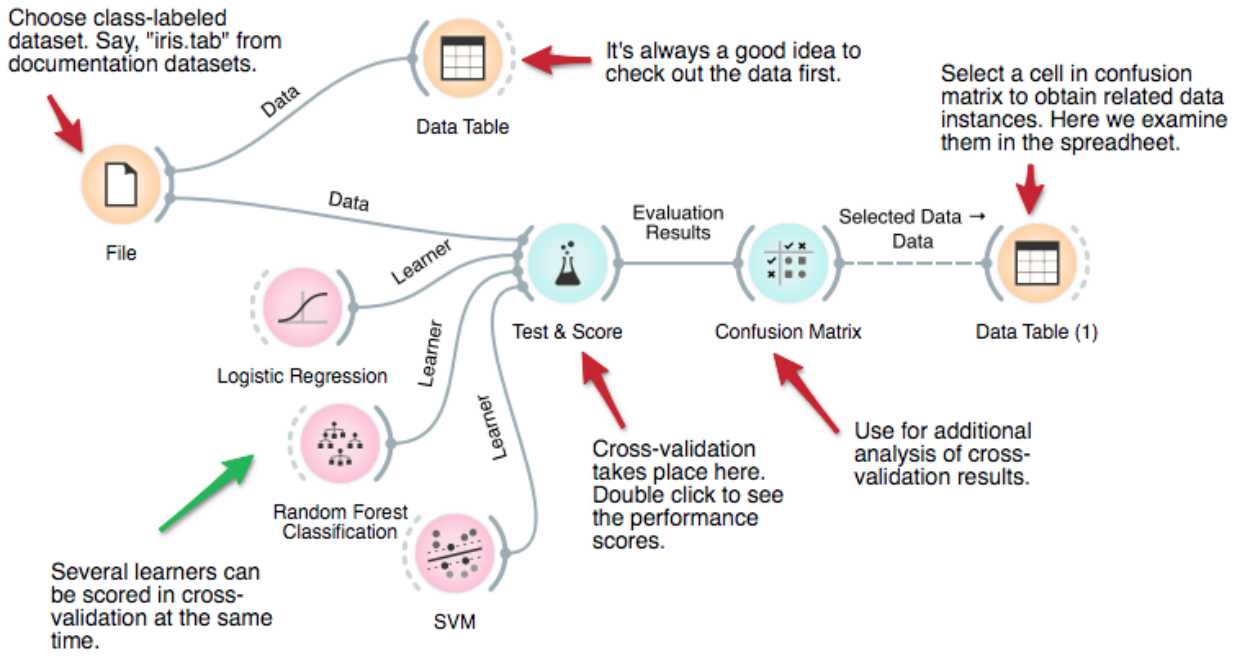
  - จากตัวอย่าง กำหนด k=4 ดังนั้นจึงต้องทดสอบประสิทธิภาพของโมเดลจำนวน 4 ครั้ง

ตัวอย่างการทำงานของ Cross-Validation สามารถทำได้โดย

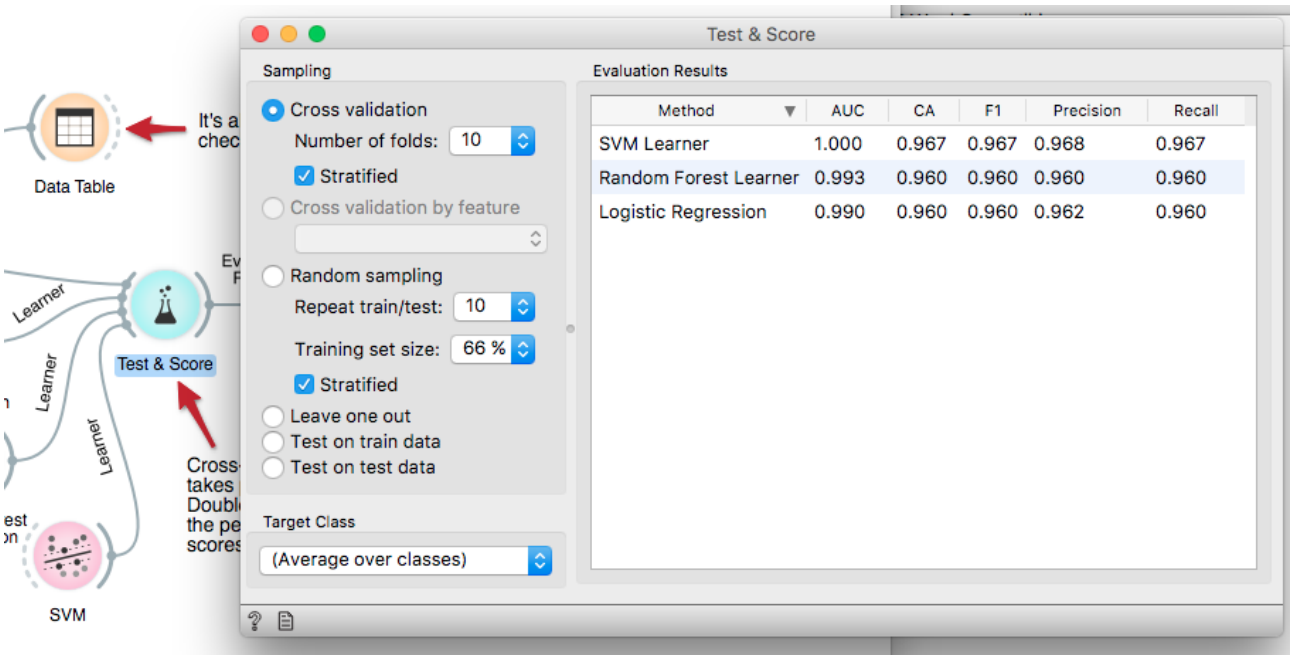
- คลิกที่เมนู **Help** > **Examples** โปรแกรม Orange จะเปิดตัวอย่างของการใช้โปรแกรม
- ให้คลิกเลือก **Cross-Validation** และคลิกที่ปุ่ม **Open**


Example Workflows
Example Workflows
Cross-Validation
How good are supervised data mining methods on your classification dataset? Here's a workflow that scores various classification techniques on a dataset from medicine. The central widget here is the one for testing and scoring, which is given the data and a set of learners, does cross-validation and scores predictive accuracy, and outputs the scores for further examination.
Choose class-labeled dataset. Say, "iris.tab" from documentation datasets.
It's always a good idea to check out the data first.
Data Table
Select a cell in confusion matrix to obtain related data instances. Here we examine them in the spreadheet.
File
Test & Score
Confusion Matrix
Data Table (1)
Logistic Regression
Random Forest
Classification
SVM
Several learners can be scored in cross-validation at the same time.
Cross-validation takes place here. Double click to see the performance scores.
Use for additional analysis of cross-validation results.
Path: /Applications/Orange3.app/Contents/Frameworks/Python.fr…e/canvas/application/workflows/450-cross-validation.ows
Tree
Principal Component Analysis
Hierarchical Clustering
Feature Ranking
Cross-Validation
Where are Misclassifications?
Cancel
Open

- เมื่อคลิกที่ **Open** โปรแกรมจะเปิด workflow ดังต่อไปนี้



  - จากรูปภาพตัวอย่างข้างต้น เป็นการทดสอบประสิทธิภาพของอัลกอริธึม หรือในโปรแกรม Orange จะเรียกว่า Learner จำนวนทั้งสิ้น 3 อัลกอริธึม ประกอบด้วย Logistic Regression, Random Forest Classification และ Support Vector Machine (SVM)
  - ส่วนของ Cross-Validation จะอยู่ในไอคอน **Test & Score**
  - ข้อมูลที่ใช้ในการทดสอบคือข้อมูลชุด iris
- ดับเบิ้ลคลิกที่ **Test & Score** จากนั้นจะปรากฏหน้าต่าง ดังตัวอย่างต่อไปนี้

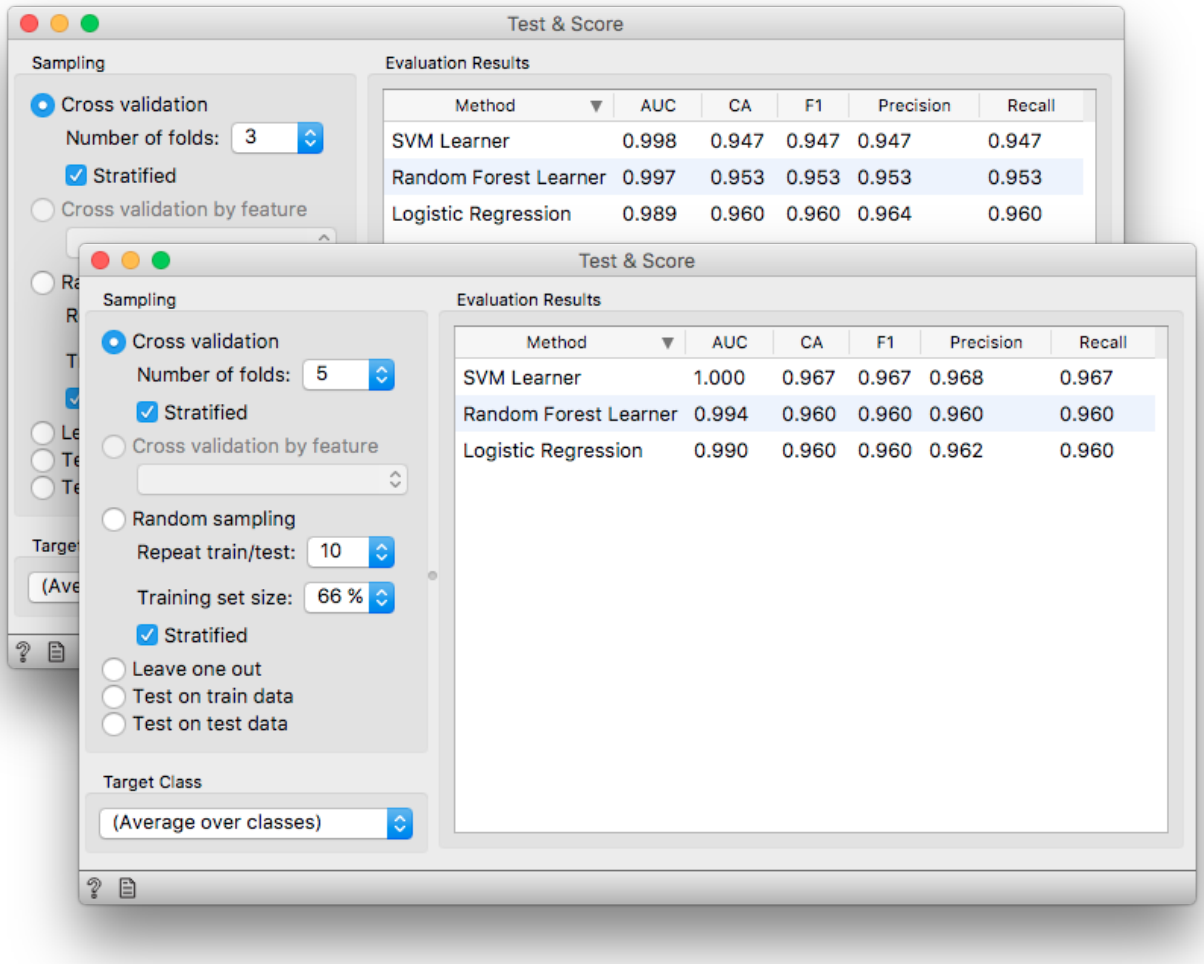


- ในส่วนของ **Sampling** สามารถกำหนด parameter ต่าง ๆ ดังนี้
  - Cross validation
  - Random sampling
  - Leave one out
  - Test on train data
  - Test on test data
- ในส่วนของ **Evaluation Results** ประกอบด้วยผลลัพธ์ทั้งสิ้น 5 วิธี ดังนี้
  - AUC - Area under receiver-operating curve
  - CA - Classification Accuracy is the proportion of correctly classified examples
  - F1 – Weighted harmonic mean of precision and recall
  - Precision – the proportion of true positives among instances classified as positive, e.g. the proportion of Iris virginica correctly identified as Iris virginica
  - Recall – the proportion of true positives among all positive instances in the data, e.g. the number of sick among all diagnosed as sick

Cross Validation

การแบ่งข้อมูลด้วยวิธี Cross Validation สามารถกำหนด parameter ได้ดังต่อไปนี้

- **Number of folds**
  - กำหนดจำนวนของชุดข้อมูล เช่น หากกำหนดให้เป็น 2 fold ดังนั้น ข้อมูลก็จะถูกแบ่งออกเป็น 2 ส่วน หากกำหนดเป็น 5 fold ข้อมูลก็จะถูกแบ่งเป็น 5 ส่วน และการทดสอบก็จะทดสอบจำนวน 5 รอบ (Iteration)
- **Stratified**
  - กำหนดให้การเลือกข้อมูลในแต่ละชุดมีการเลือกข้อมูลเท่า ๆ กัน เช่น เมื่อข้อมูลที่เลือกมาอยู่ใน Class 1 มีจำนวน 65 instance ดังนั้น ข้อมูลที่เลือกมาอยู่ใน Class 2 ต้องมีจำนวน 65 instance เช่นเดียวกัน



- จากตัวอย่างข้างต้น แสดงผลลัพธ์ของการเลือก Cross validation โดยกำหนดให้มีจำนวน 3 และ 5 fold ดังนั้น ให้สังเกตผลลัพธ์ของวิธี SVM
  - กำหนดให้เป็น 3 fold วิธี SVM จะมีความถูกต้อง (CA) 0.947
  - กำหนดให้เป็น 5 fold วิธี SVM จะมีความถูกต้อง (CA) 0.967 ซึ่งเพิ่มขึ้น
- โดยที่ 0.947 คือ 94.7%

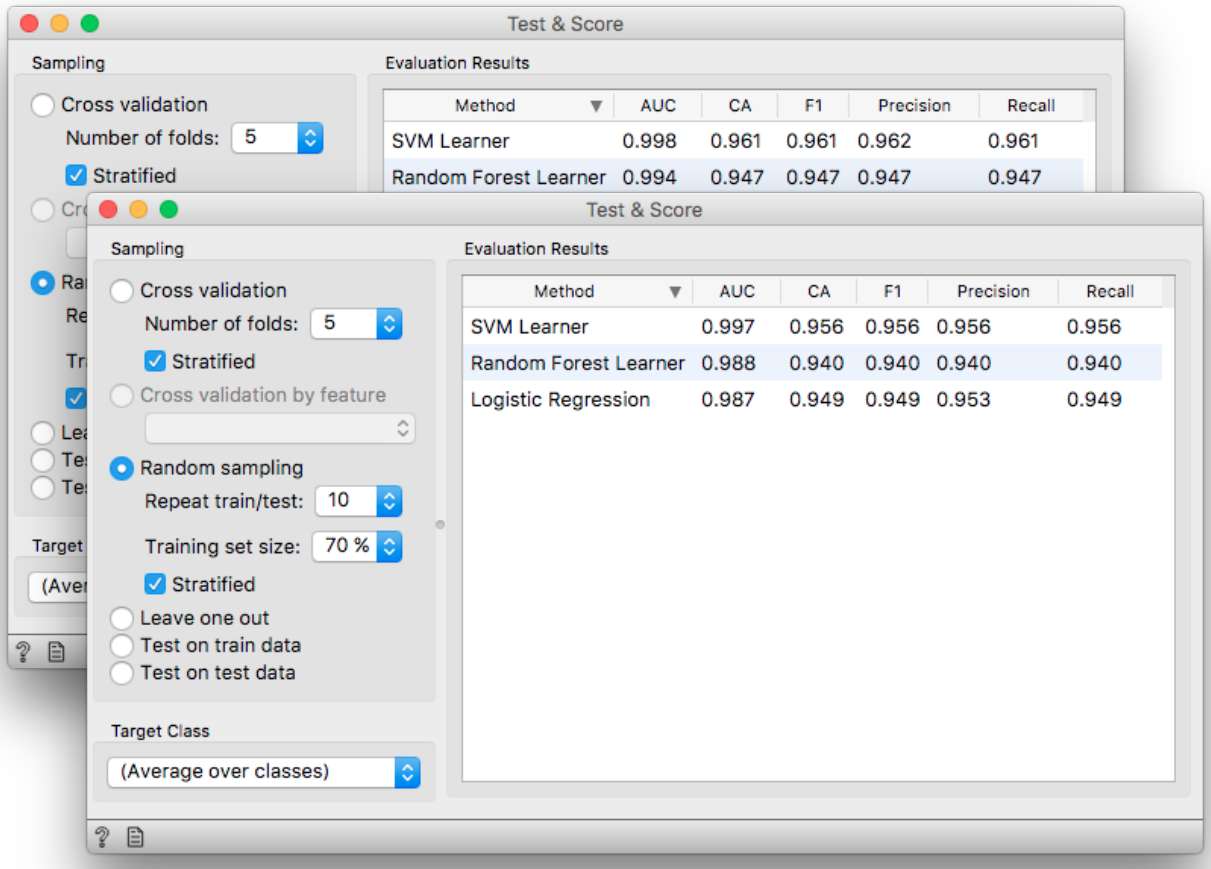
Random Sampling

- Random Sampling คือการสุ่มแบ่ง (Randomly Split) ข้อมูลเป็นสองส่วน (ข้อมูลชุดเรียนรู้ และข้อมูลชุดทดสอบ) เช่น แบ่งออกเป็น 70:30 หรือ 60:40 เป็นต้น โดยการทดสอบประสิทธิภาพจะขึ้นอยู่กับจำนวนที่ต้องการทดสอบ (Repeat Train/Test)

การแบ่งข้อมูลด้วยวิธี Random Sampling สามารถกำหนด parameter ได้ดังต่อไปนี้

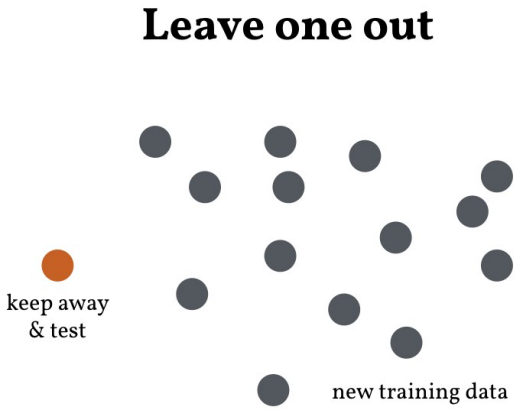
- **Repeat Train/Test**
  - กำหนดจำนวนของการทดสอบ เช่น หากกำหนดให้มีค่าเป็น 10 หมายถึงทำการทดสอบ 10 รอบ
- **Training set size**
  - กำหนดจำนวนขนาดของข้อมูลชุดเรียนรู้ เช่น หากกำหนดให้มีค่าเป็น 66% ดังนั้น หากมีข้อมูลทั้งสิ้น 100 instance ข้อมูลดังกล่าวจะถูกแบ่งให้เป็น Training set จำนวน 66 instance และที่เหลือ 34 instance จะแบ่งเป็น Test set
- **Stratified**
  - กำหนดให้การเลือกข้อมูลในแต่ละชุดมีการเลือกข้อมูลเท่า ๆ กัน เช่น เมื่อข้อมูลที่เลือกมาอยู่ใน Class 1 มีจำนวน 65 instance ดังนั้น ข้อมูลที่เลือกมาอยู่ใน Class 2 ต้องมีจำนวน 65 instance เช่นเดียวกัน

- ตัวอย่างต่อไปนี้ กำหนดให้ Training set size มีขนาด 66% และ 70% ปรากฏว่าวิธี SVM กำหนดให้ Training set มีขนาด 66% มีความถูกต้อง 0.956 ซึ่งมากกว่าการแบ่งแบบ 70% ที่มีความถูกต้อง 0.956



Leave One Out

- สำหรับวิธี Leave One Out คือการนำข้อมูลจำนวน 1 instance แยกออกมาเพื่อทำการ Test และข้อมูลที่เหลือใช้เป็นข้อมูลสำหรับ Train ดังนั้น หากดึงข้อมูลมาทดสอบ n instance ก็จะต้องทดสอบ n ครั้ง (Iteration) ซึ่งวิธีนี้จะมีความแม่นยำใกล้เคียงกับการนำไปใช้ในชีวิตจริง
- วิธีการ Leave One Out เป็นวิธีที่มีการทำงานที่ช้า เนื่องจากจะต้องทดสอบจำนวนรอบตามจำนวนของข้อมูล เช่น หากมีข้อมูล 10,000 instance ก็จะต้องทดสอบ 10,000 iteration นั่นหมายถึงจะต้องสร้างโมเดล และทดสอบ 10,000 ครั้ง

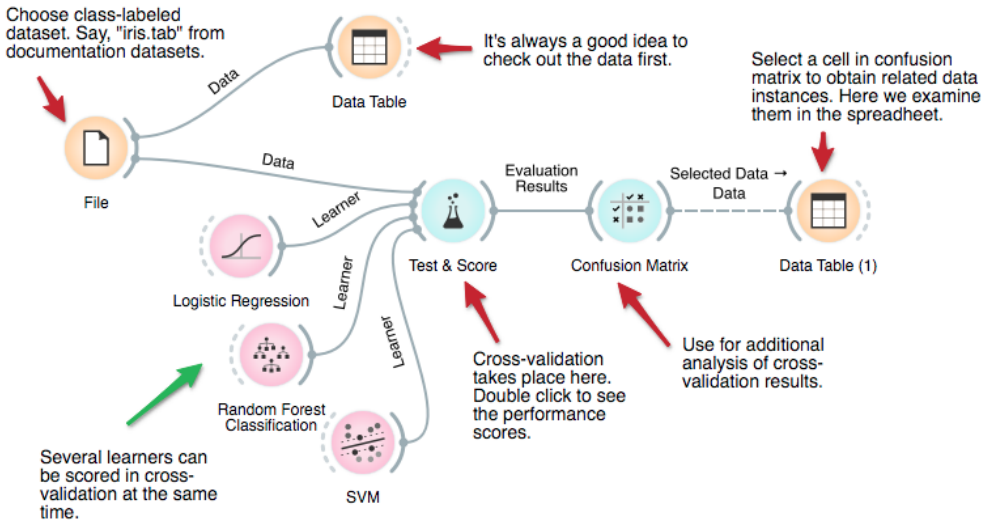


[ที่มา: https://goo.gl/PAQeiJ]

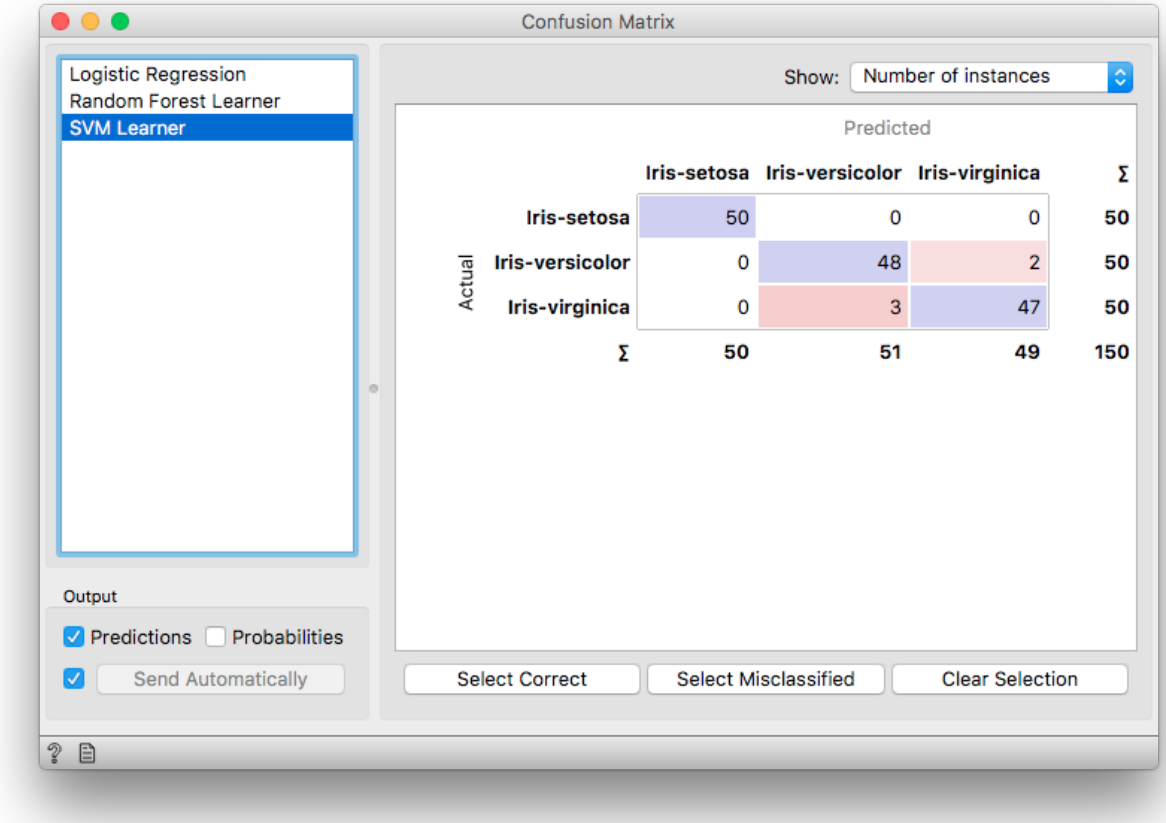
- ผลลัพธ์ที่ได้จาก Leave One Out ด้วยวิธี SVM มีความถูกต้อง 0.96 แสดงดังรูปภาพต่อไปนี้

Confusion Matrix

- เป็นวิธีที่เปรียบเทียบให้เห็นผลลัพธ์ของการพยากรณ์ (Prediction) และค่าที่แท้จริง (Actual) ของข้อมูล



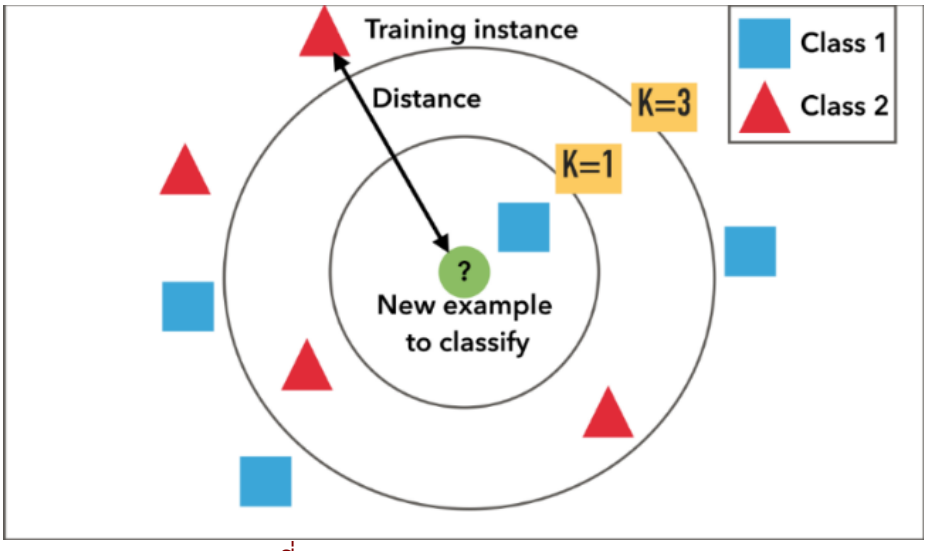
- จากรูปข้างต้นให้ดับเบิ้ลคลิกที่ไอคอน **Confusion Matrix**



- ข้อมูลชุด irtis มี class ทั้งสิ้น 3 class ประกอบด้วย setosa, versicolor และ virginica

- ข้อมูลในแนวแถว (Row) คือ actual class และในแนวคอลัมน์ (Column) คือการพยากรณ์ (predicted)
- จากการพยากรณ์ ปรากฏว่า
  - **แถวแรก** Actual class คือ **setosa** ผลจากการพยากรณ์ (Predicted) เป็น setosa จำนวน 50 แสดงว่าพยากรณ์ถูกต้องทั้งหมด
  - **แถวที่สอง** Actual class คือ **versicolor** ผลจากการพยากรณ์เป็น versicolor จำนวน 48 และพยากรณ์เป็น verginica จำนวน 2
  - **แถวที่สาม** Actual class คือ **verginica** ผลจากการพยากรณ์เป็น verginica จำนวน 47 และพยากรณ์เป็น versicolor จำนวน 3

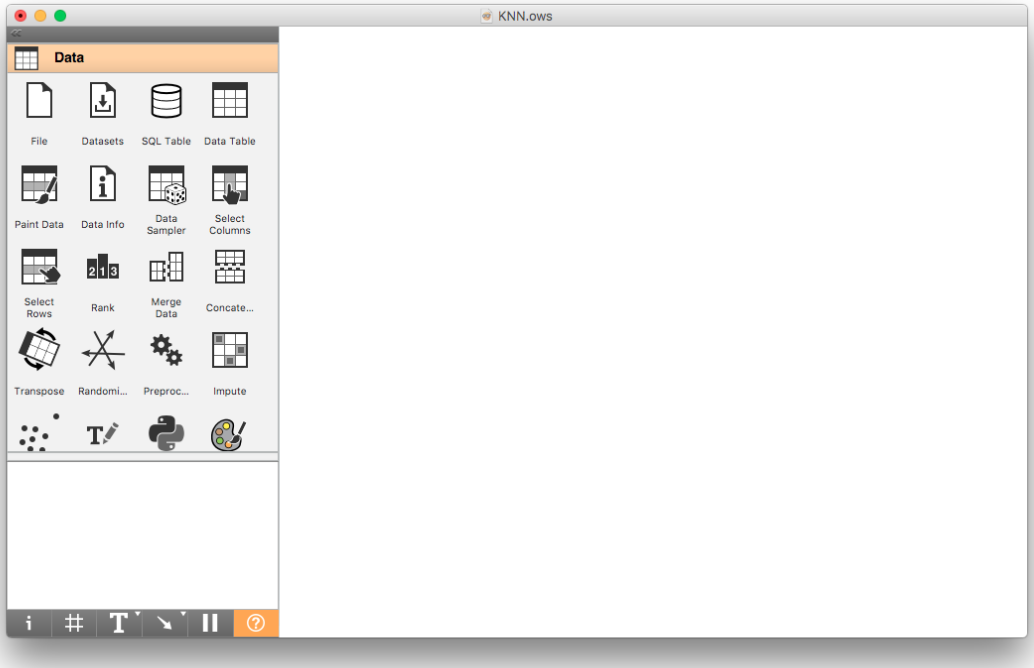
K-Nearest Neighbor (KNN)



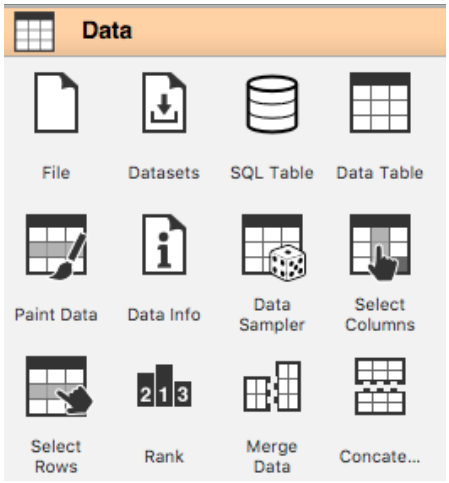
[ที่มา: https://goo.gl/1JR68b]

เริ่มต้นสร้างโมเดลของ KNN โดยสร้าง workflow ใหม่

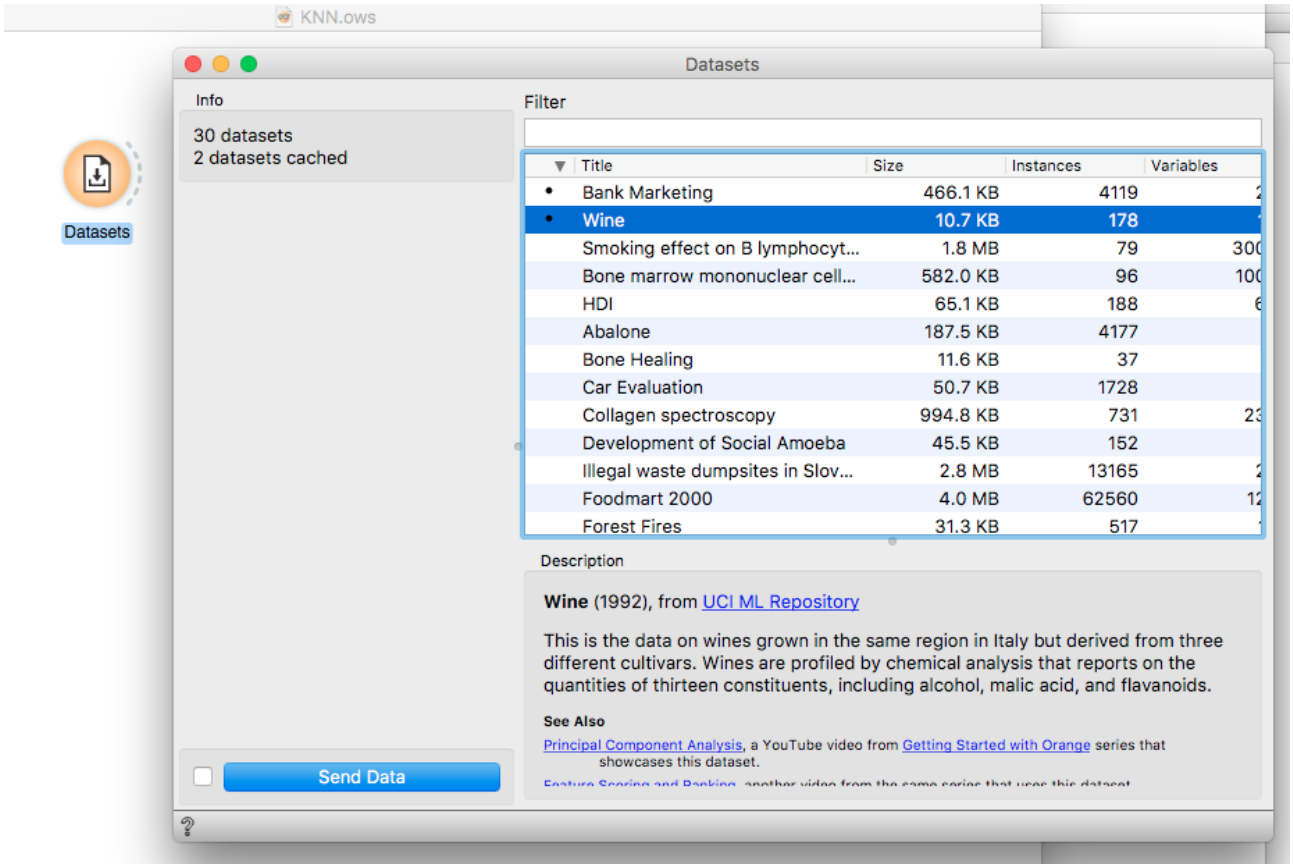
- ที่เมนูเลือก **File** > **New**
- จากนั้นกำหนดรายละเอียดของ workflow (Workflow Info) โดยพิมพ์ข้อมูลลงไปในช่อง **Title** และ **Description** ของ
- จากนั้นเลือกเมนู **File** > **Save** เพื่อบันทึก workflow โดย workflow ที่สร้างจะมีนามสกุล **.ows**



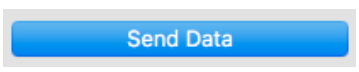
- จากนั้นเลือกไอคอน **Datasets** จากแท็ป Data เพื่อเพิ่มไอคอน Datasets ลงใน workflow



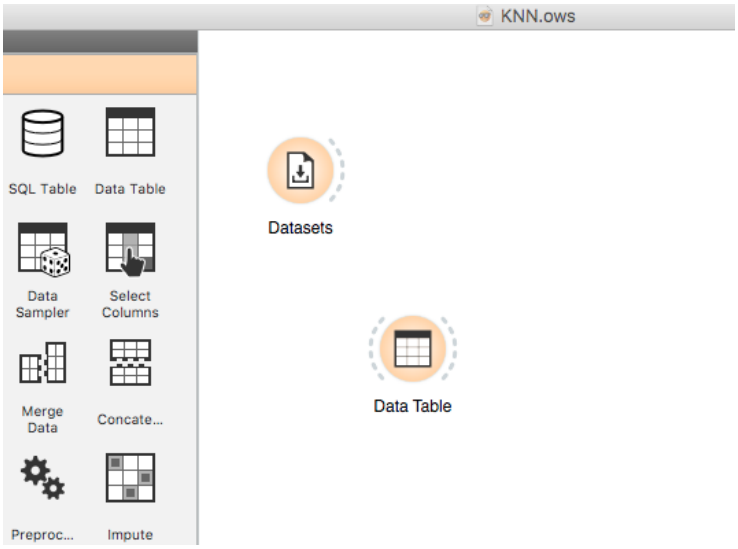
- ใน workflow ให้ดับเบิ้ลคลิกที่ไอคอน Datasets และเลือก dataset ที่ต้องการ
  - ในกรณีให้เลือก Dataset ที่ชื่อ **Wine**



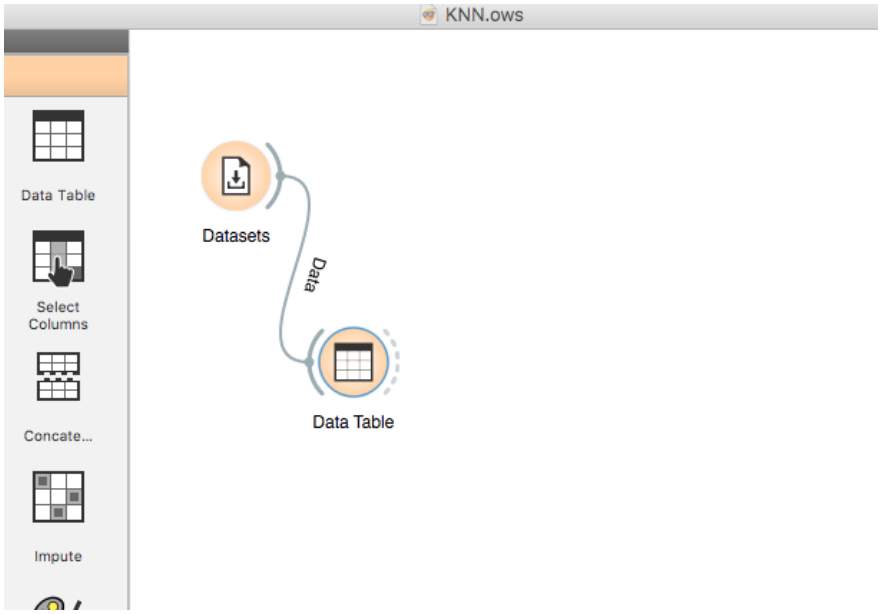
  - เมื่อเลือก Dataset ชื่อ Wine จากนั้นให้คลิกที่ปุ่ม **Send Data**



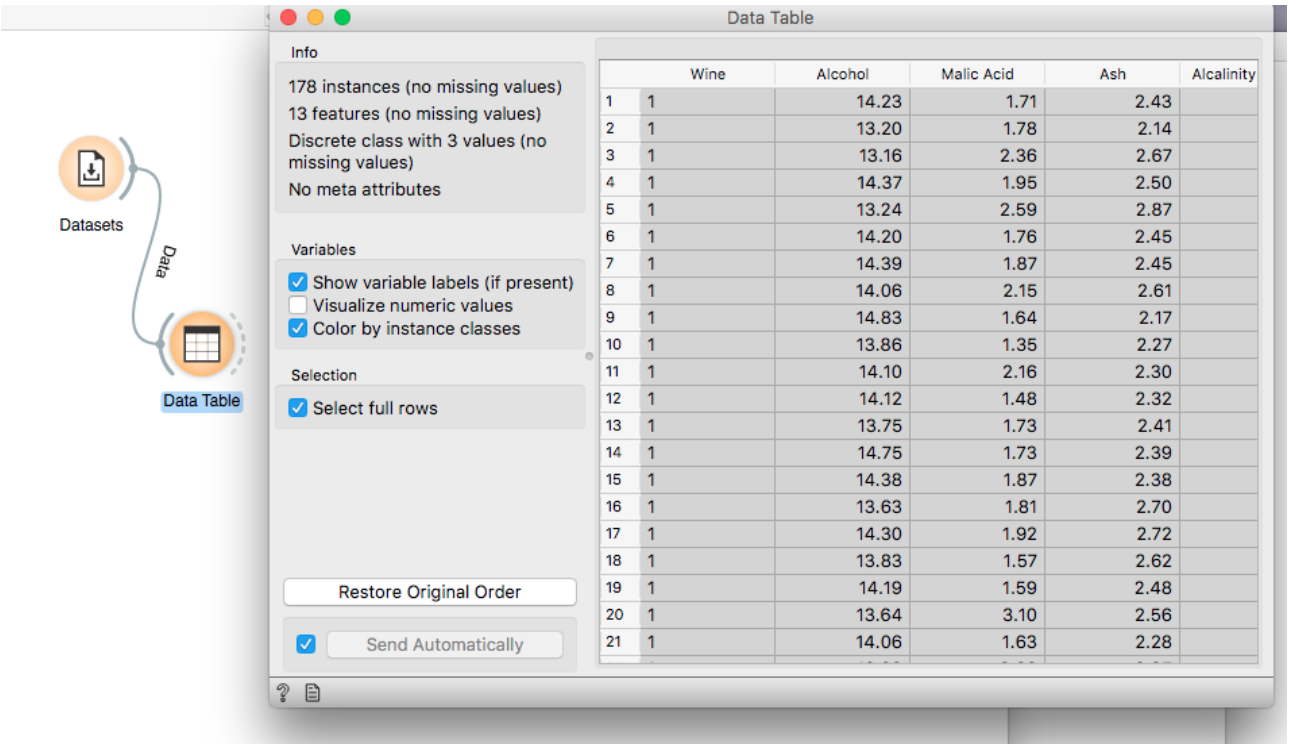
- จากนั้นเพิ่มไอคอน **Data Table** ลงใน workflow



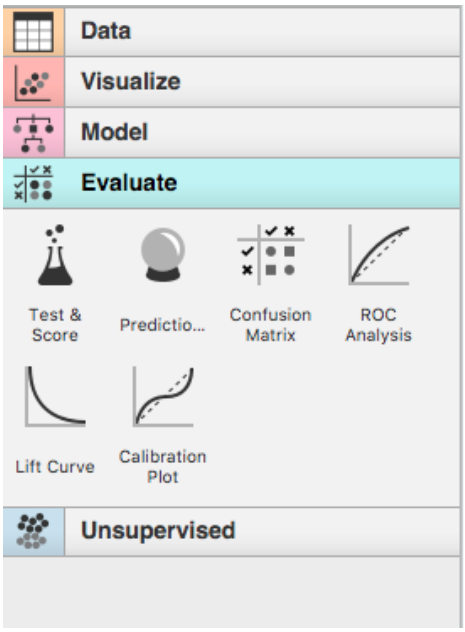
- ที่ไอคอน **Datasets** ให้คลิกเพื่อลากเส้นเชื่อมไปยังไอคอน **Data Table**

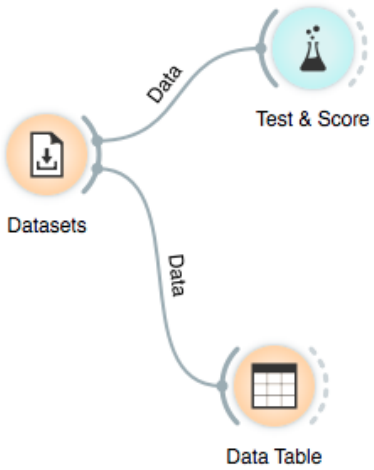


- หากต้องการที่จะดูข้อมูลในชุดข้อมูล Wine ให้ดับเบิ้ลคลิกที่ไอคอน **Data Table**

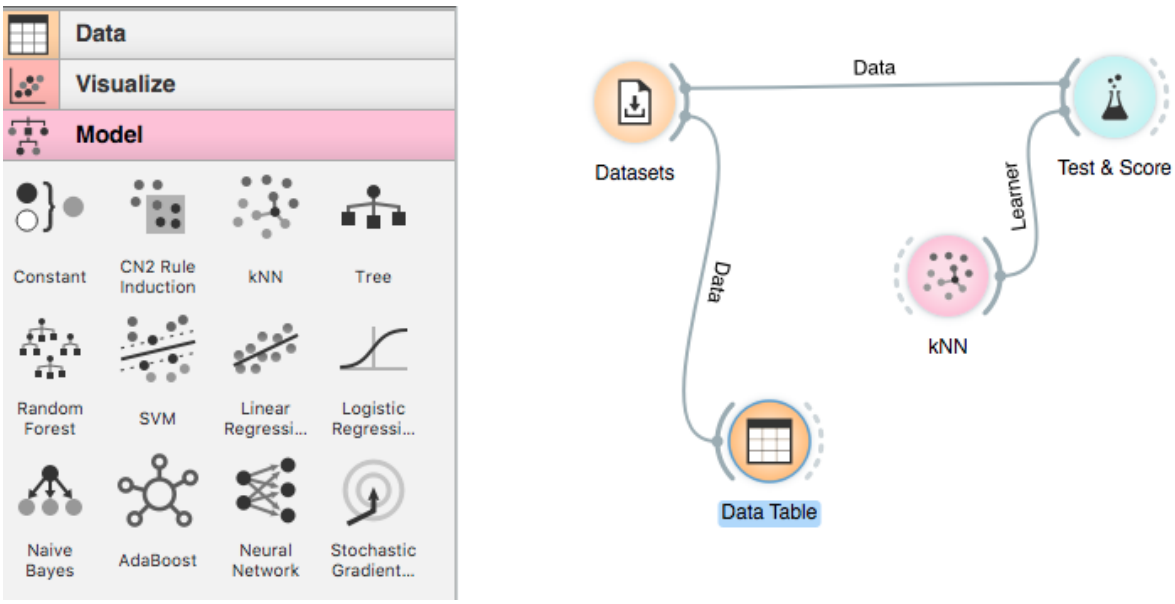


- จากนั้นที่แท็ป **Evaluate** ให้เพิ่มไอคอน **Test & Score** ลงไปใน workflow
- ลากเส้นเชื่อมระหว่างไอคอน **Datasets** และ **Test & Score**

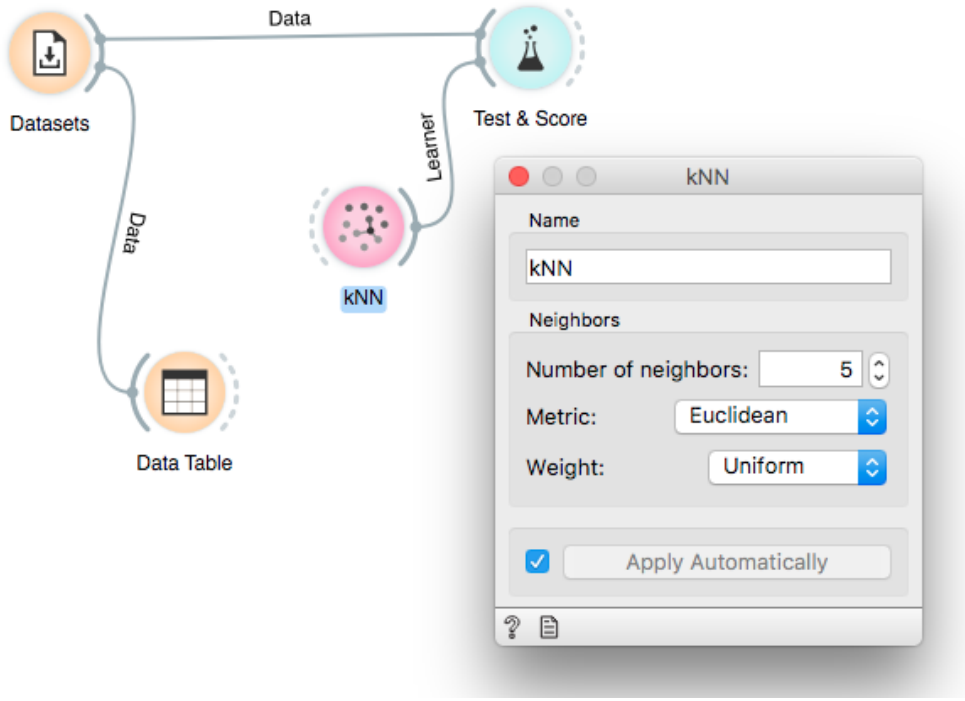




- ที่แท็ป Model ให้คลิกเพิ่มไอคอน KNN ลงไปใน workflow
- ลากเส้นเชื่อมระหว่างไอคอน KNN และ **Test & Score**

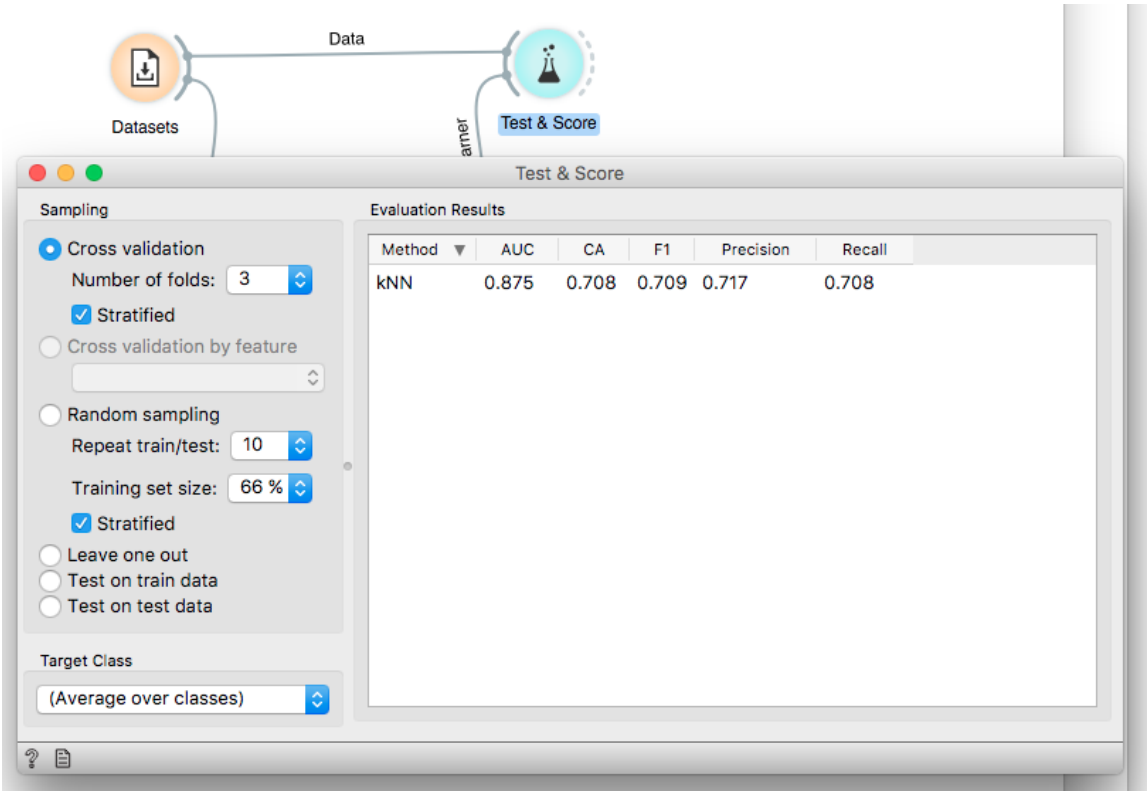


- สามารถกำหนด parameter ให้กับอัลกอริธึม KNN โดยดับเบิ้ลคลิกที่ไอคอน **KNN**

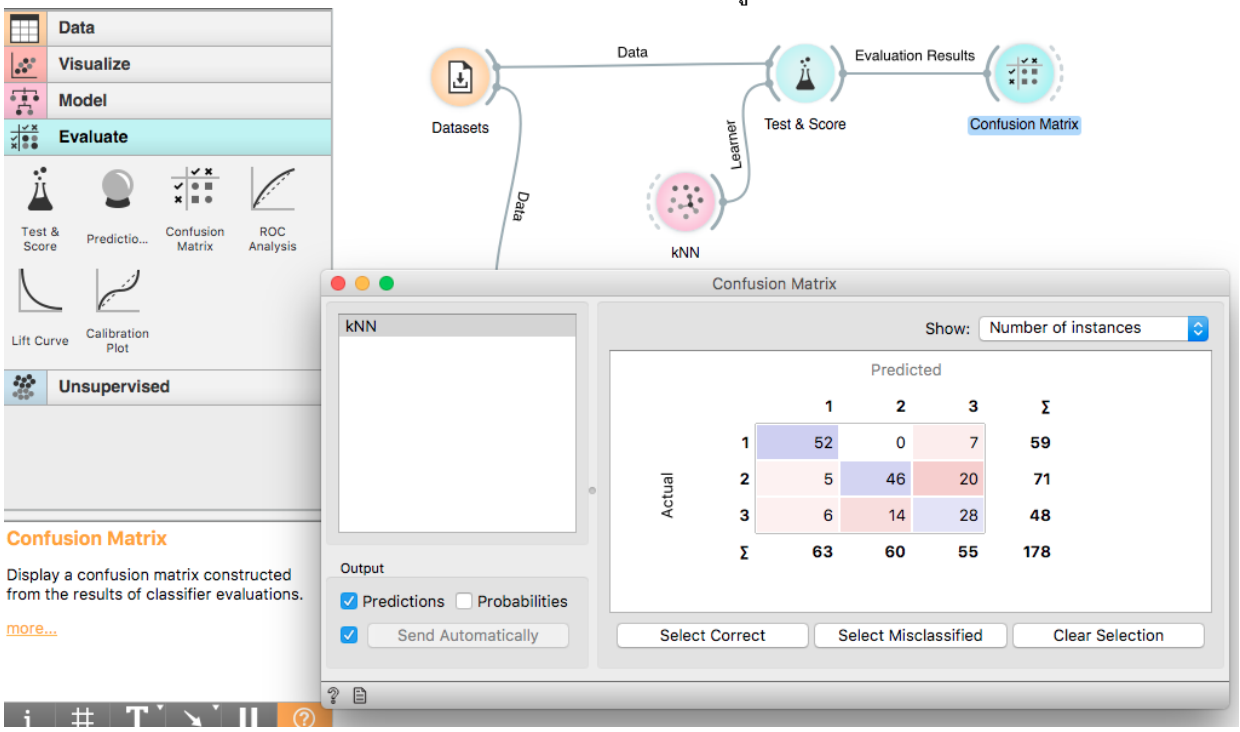


- Parameter ของ KNN ที่สามารถกำหนดได้ประกอบด้วย
  - Number of neighbors
  - Metric
  - Weight

- เมื่อกำหนด parameter ให้กับ KNN เรียบร้อย จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน **Test & Score** เพื่อดูผลลัพธ์ที่ได้จาก KNN
  - โดยสามารถกำหนดการทำ Cross-Validation ได้

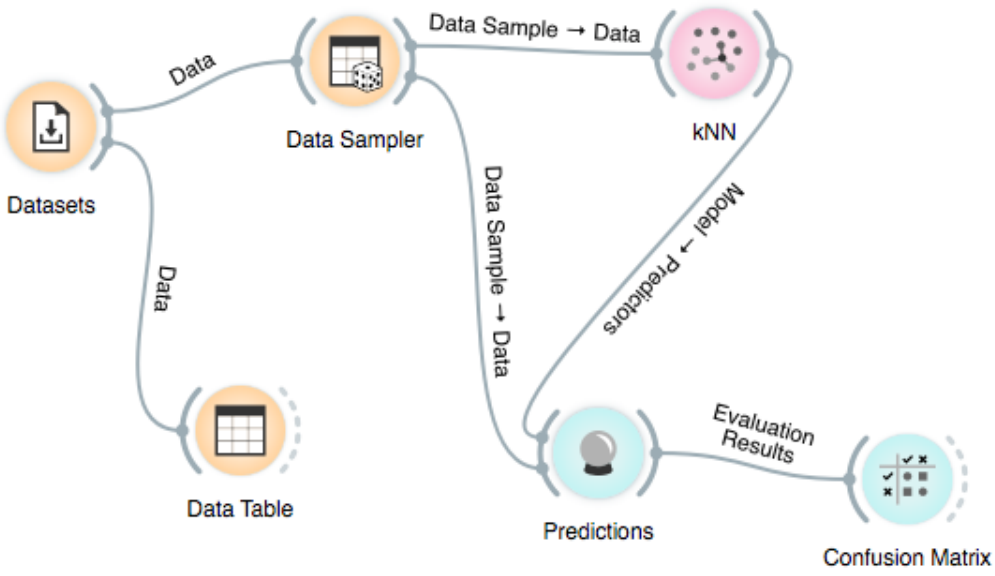


- จากนั้นเลือกที่แท็ป Evaluate และเลือกที่ไอคอน **Confusion Matrix** เพื่อเพิ่มลงไปใน workflow
- สามารถดับเบิ้ลคลิกที่ไอคอน Confusion Matrix เพื่อดูคำตอบที่ได้จากการพยากรณ์

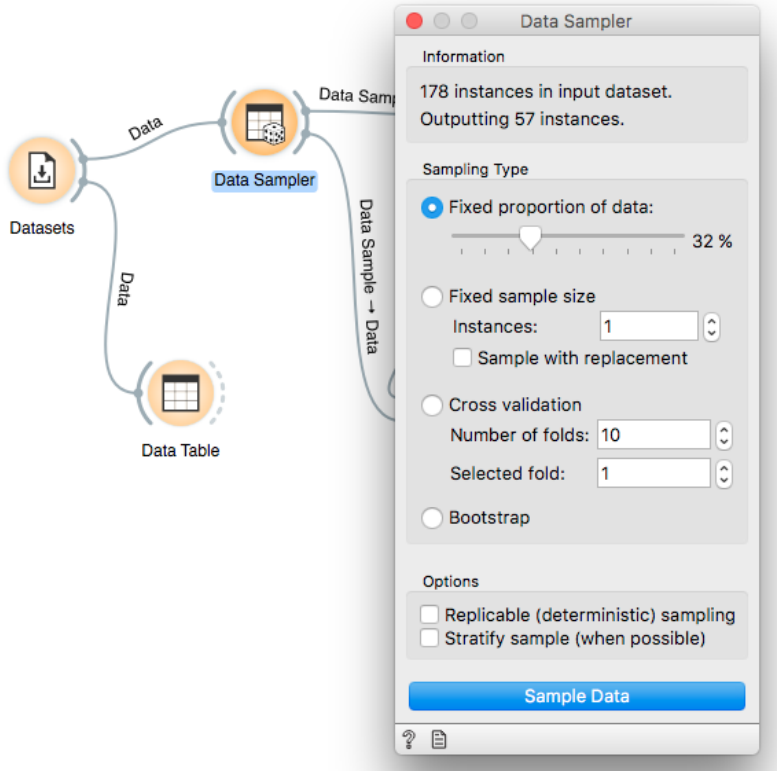


การพยากรณ์ด้วยวิธี KNN (KNN Prediction)

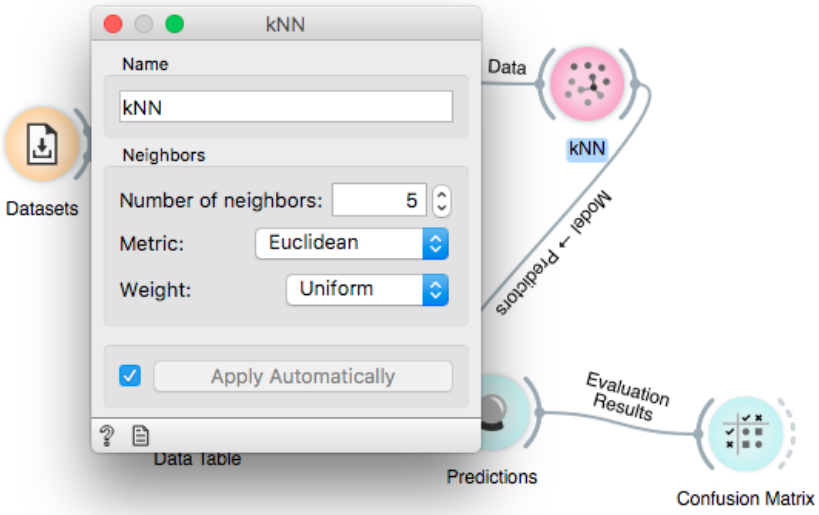
- หากต้องการดูผลการพยากรณ์ของข้อมูลในแต่ละ instance สามารถทำได้โดยใช้ไอคอน **Prediction**
- สร้าง workflow ดังตัวอย่างต่อไปนี้

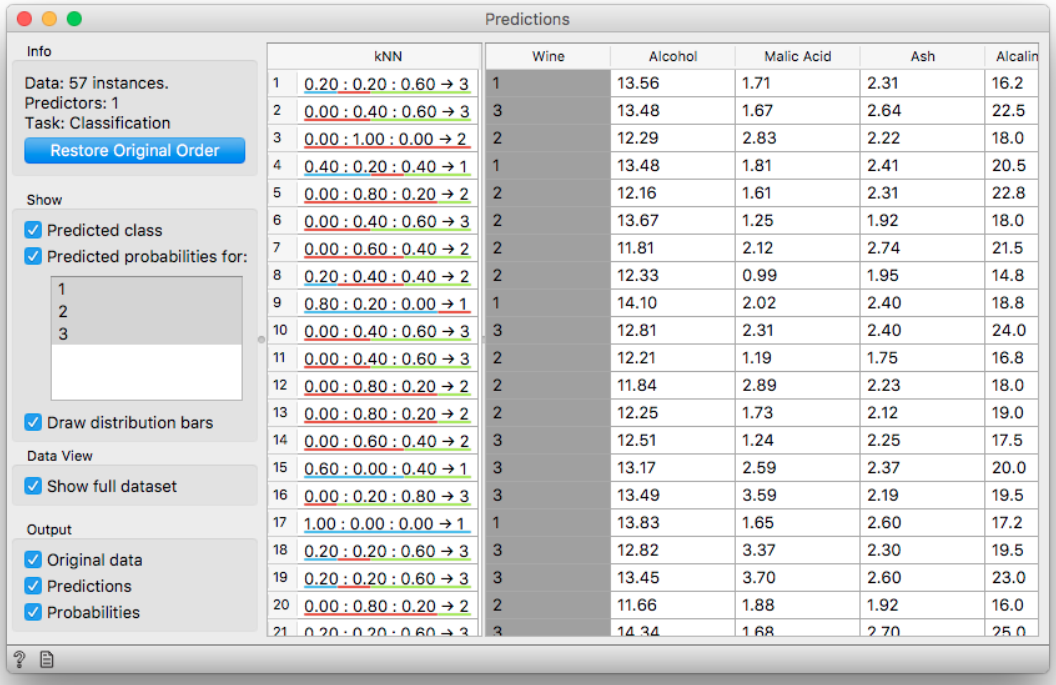


- ดับเบิ้ลคลิกที่ไอคอน **Datasets** และเลือกชุดข้อมูล **wine**
- ดับเบิ้ลคลิกที่ไอคอน **Data Sampler** และเลือกวิธี Sampling Type ที่ต้องการ
  - Fixed proportion of datasets
  - Fixed sample size
  - Cross validation
  - Bootstrap

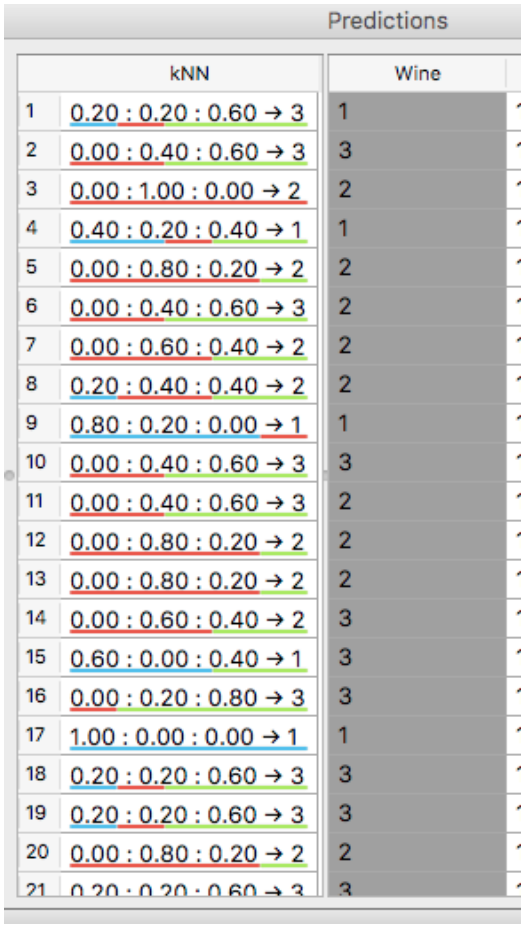


- จากตัวอย่างเลือก **Fixed proportion of data** และกำหนดขนาด 32%
  - หมายถึงเลือกข้อมูลมาจำนวน 32% ของข้อมูลทั้งหมด
- ดับเบิ้ลคลิกที่ไอคอน **KNN** จะปรากฏหน้าต่างดังต่อไปนี้ และโปรแกรมจะกำหนดค่ามาตรฐานของ parameter ดังนี้
  - Number of neighbors = 5
  - Metric = Euclidean
  - Weight = Uniform





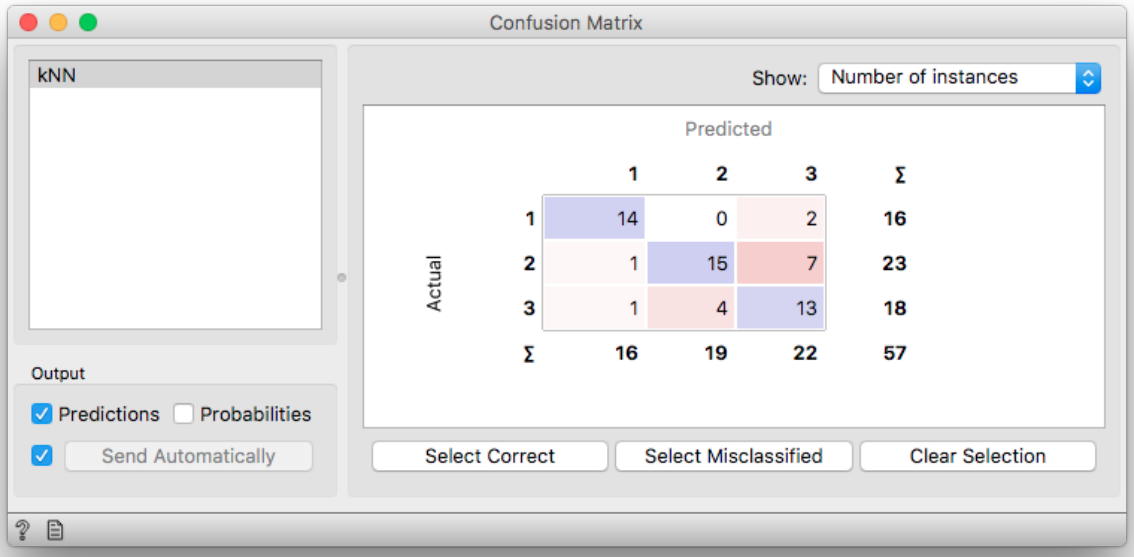
- ดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อดูผลลัพธ์จากการพยากรณ์
- จากรูปภาพข้างต้น แสดงให้เห็นถึงผลลัพธ์ของการพยากรณ์ของแต่ละ instance



- การพยากรณ์แสดงทาง column ทางฝั่งซ้าย
- Column ทางฝั่งขวา คือ Actual Class

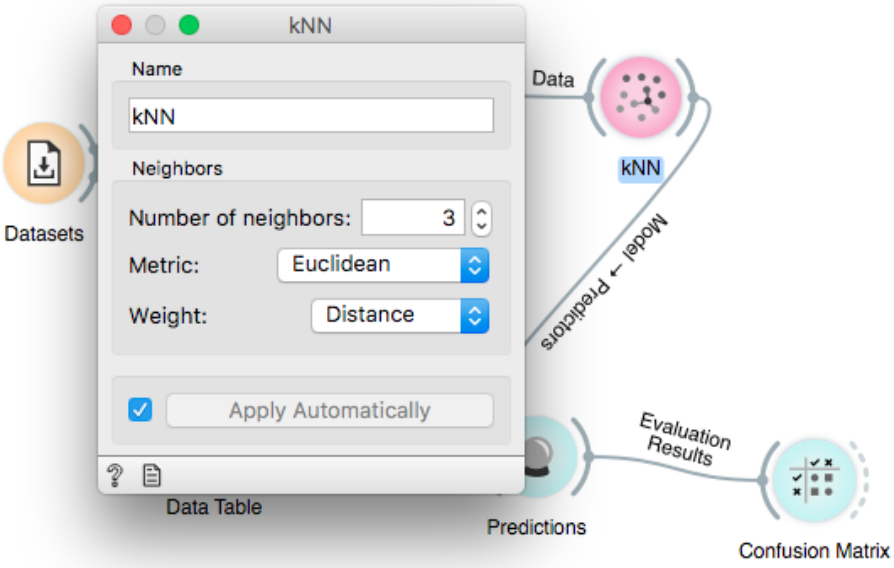
- จากตัวอย่าง 0.20 : 0.20 : 0.60 → 3 | Actual = 1
ความน่าจะเป็นที่จะเป็น Class 1 ที่ 0.20
ความน่าจะเป็นที่จะเป็น Class 2 ที่ 0.20
ความน่าจะเป็นที่จะเป็น Class 3 ที่ **0.60**
คำตอบที่ได้จึงเป็น **Class 3** ซึ่งเป็นคำตอบที่ **ผิด**

- ดับเบิ้ลคลิกที่ไอคอน **Confusion Matrix** เพื่อดูภาพรวมของการพยากรณ์
  - จากการพยากรณ์ มีการพยากรณ์ผิดพลาดจำนวน 15 instance (2+1+7+1+4)



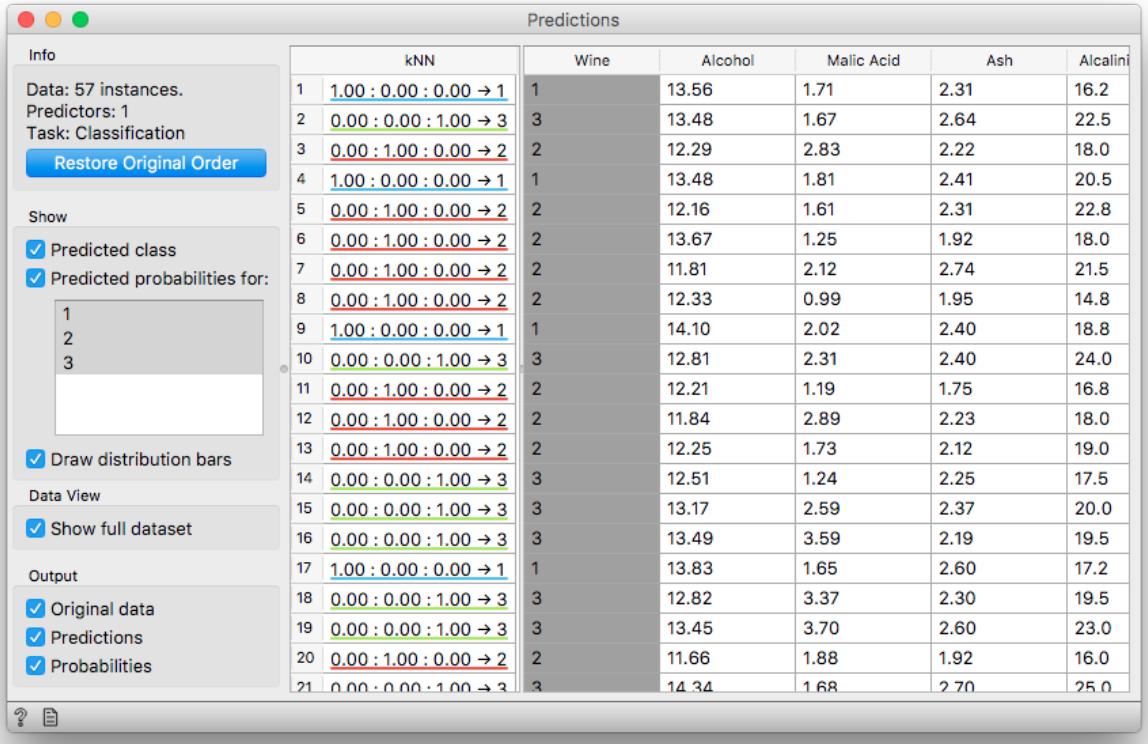
KNN Parameter Tuning (การปรับค่าพารามิเตอร์ของ KNN)

- หากต้องการให้ผลการการพยากรณ์มีความถูกต้องแม่นยำขึ้นในอัลกอริทึมของ KNN สามารถทำได้โดย ดับเบิ้ลคลิกที่ไอคอน **KNN**



- จากตัวอย่างทดลองปรับค่าพารามิเตอร์ โดยกำหนดให้
  - Number of neighbors = 3
  - Metric = Euclidean
  - Weight = Distance

- ดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อตรวจสอบผลการพยากรณ์

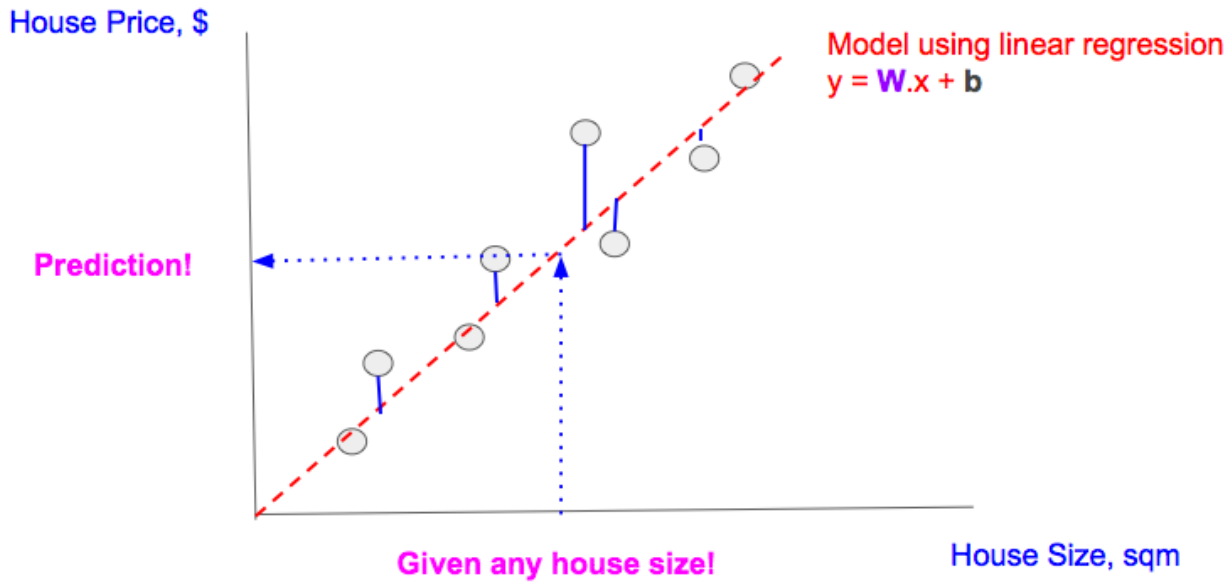


	kNN	Wine	Alcohol	Malic Acid	Ash	Alcalini
1	1.00 : 0.00 : 0.00 → 1	1	13.56	1.71	2.31	16.2
2	0.00 : 0.00 : 1.00 → 3	3	13.48	1.67	2.64	22.5
3	0.00 : 1.00 : 0.00 → 2	2	12.29	2.83	2.22	18.0
4	1.00 : 0.00 : 0.00 → 1	1	13.48	1.81	2.41	20.5
5	0.00 : 1.00 : 0.00 → 2	2	12.16	1.61	2.31	22.8
6	0.00 : 1.00 : 0.00 → 2	2	13.67	1.25	1.92	18.0
7	0.00 : 1.00 : 0.00 → 2	2	11.81	2.12	2.74	21.5
8	0.00 : 1.00 : 0.00 → 2	2	12.33	0.99	1.95	14.8
9	1.00 : 0.00 : 0.00 → 1	1	14.10	2.02	2.40	18.8
10	0.00 : 0.00 : 1.00 → 3	3	12.81	2.31	2.40	24.0
11	0.00 : 1.00 : 0.00 → 2	2	12.21	1.19	1.75	16.8
12	0.00 : 1.00 : 0.00 → 2	2	11.84	2.89	2.23	18.0
13	0.00 : 1.00 : 0.00 → 2	2	12.25	1.73	2.12	19.0
14	0.00 : 0.00 : 1.00 → 3	3	12.51	1.24	2.25	17.5
15	0.00 : 0.00 : 1.00 → 3	3	13.17	2.59	2.37	20.0
16	0.00 : 0.00 : 1.00 → 3	3	13.49	3.59	2.19	19.5
17	1.00 : 0.00 : 0.00 → 1	1	13.83	1.65	2.60	17.2
18	0.00 : 0.00 : 1.00 → 3	3	12.82	3.37	2.30	19.5
19	0.00 : 0.00 : 1.00 → 3	3	13.45	3.70	2.60	23.0
20	0.00 : 1.00 : 0.00 → 2	2	11.66	1.88	1.92	16.0
21	0.00 : 0.00 : 1.00 → 3	3	14.34	1.68	2.70	25.0

- ดับเบิ้ลคลิกที่ไอคอน **Confusion Matrix** เพื่อดูภาพรวมของการพยากรณ์
  - จากตัวอย่างแสดงให้เห็นว่า **การพยากรณ์ไม่มีความผิดพลาด**

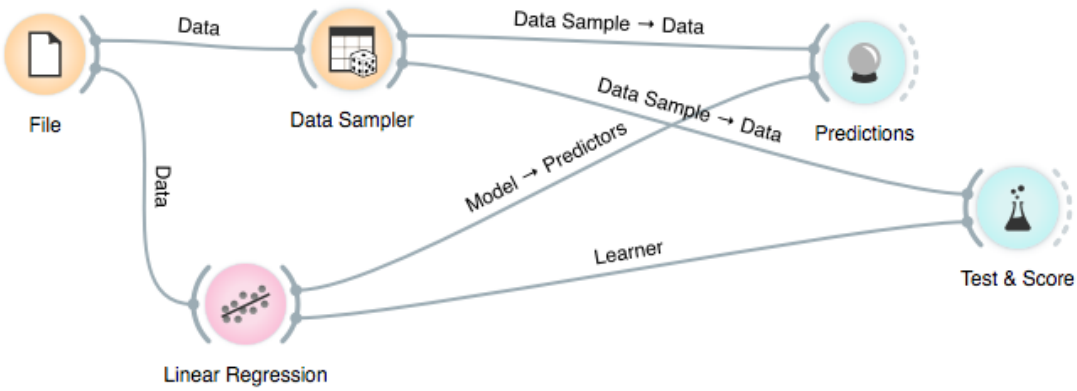
การวิเคราะห์การถดถอย (Linear Regression)

- การวิเคราะห์การถดถอย เป็นการศึกษาความสัมพันธ์ระหว่างตัวแปรตั้งแต่ 2 ตัวแปรขึ้นไป ซึ่งได้แก่ ตัวประมาณการ (Predictor, X) และตัวตอบสนอง (Response, y) โดยเป็นความสัมพันธ์แบบเชิงเส้น (Linear) [2]



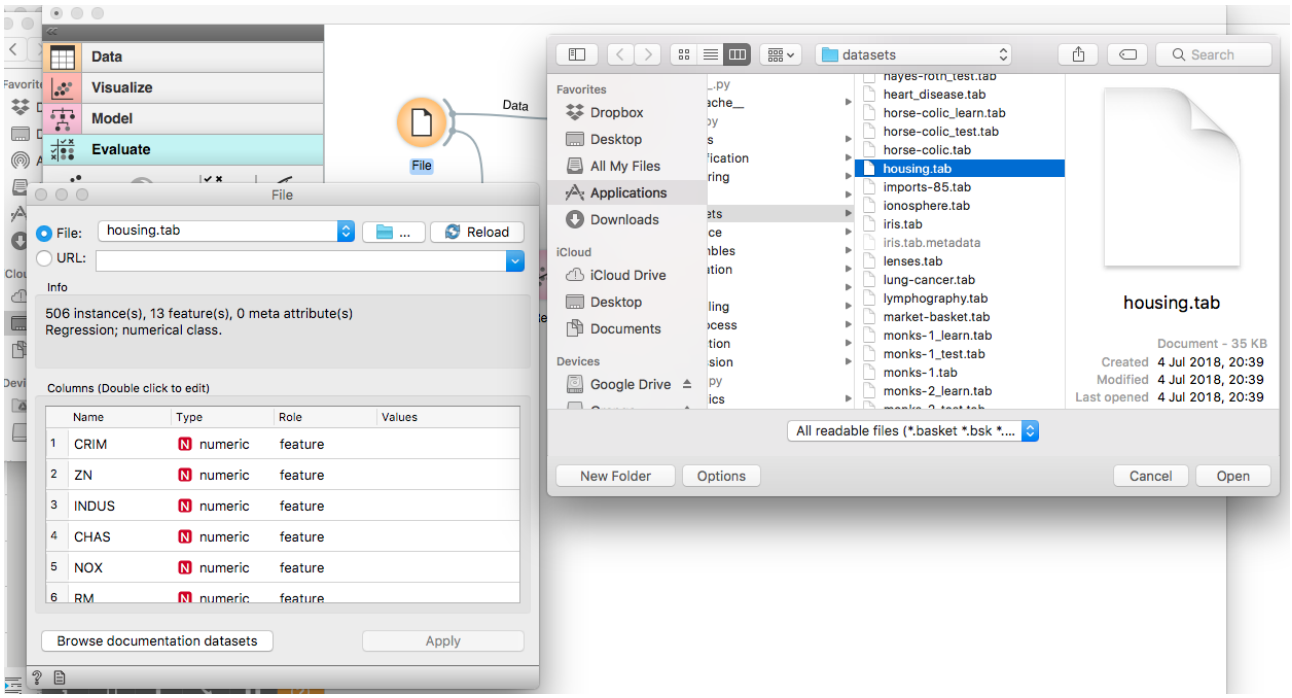
[ที่มา: https://goo.gl/eCvgqv]

- ในโปรแกรม Linear Regression สามารถทำได้ ดังตัวอย่างต่อไปนี้

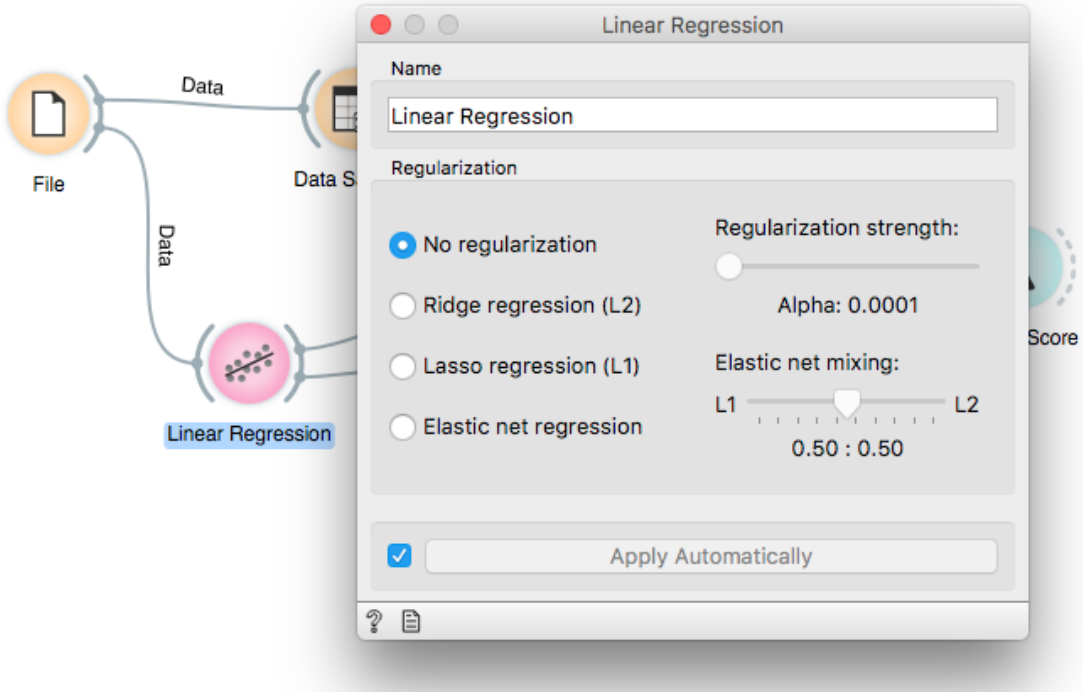


- ดับเบิ้ลคลิกที่ไอคอน **File** และเลือกชุดข้อมูล Housing
  - ข้อมูลชุดนี้ประกอบด้วยข้อมูลจำนวน 506 instance โดยมีทั้งสิ้น 13 feature ผลลัพธ์เป็นข้อมูลประเภทตัวเลข (Numerical Class) เหมาะสำหรับการนำมาทดสอบ Regression

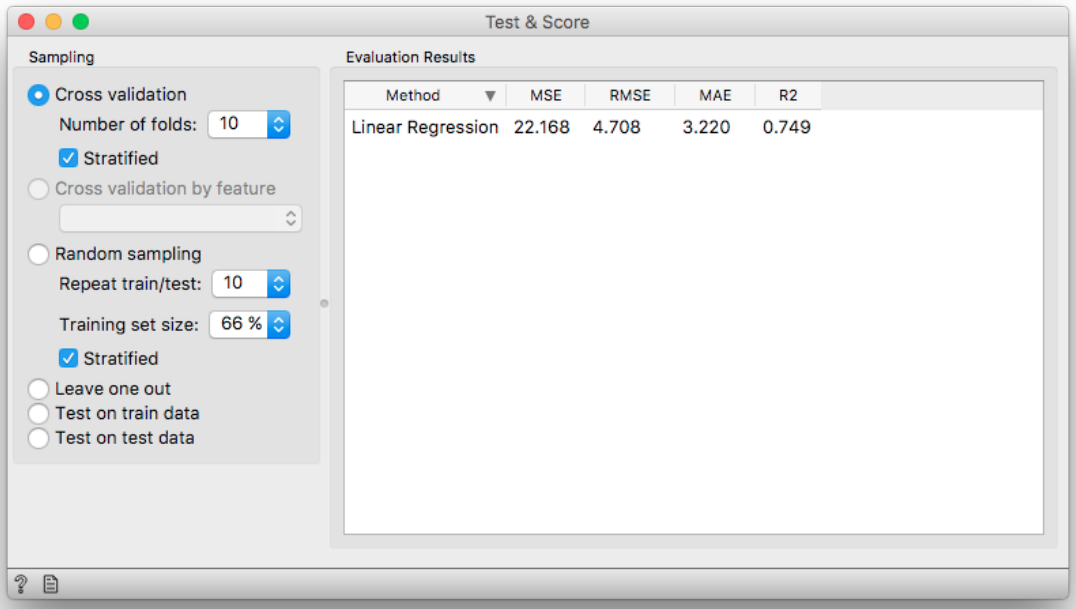
2 Linear Regression: https://goo.gl/w3pLND



- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Linear Regression** จะปรากฏหน้าต่างให้กำหนดค่า parameter



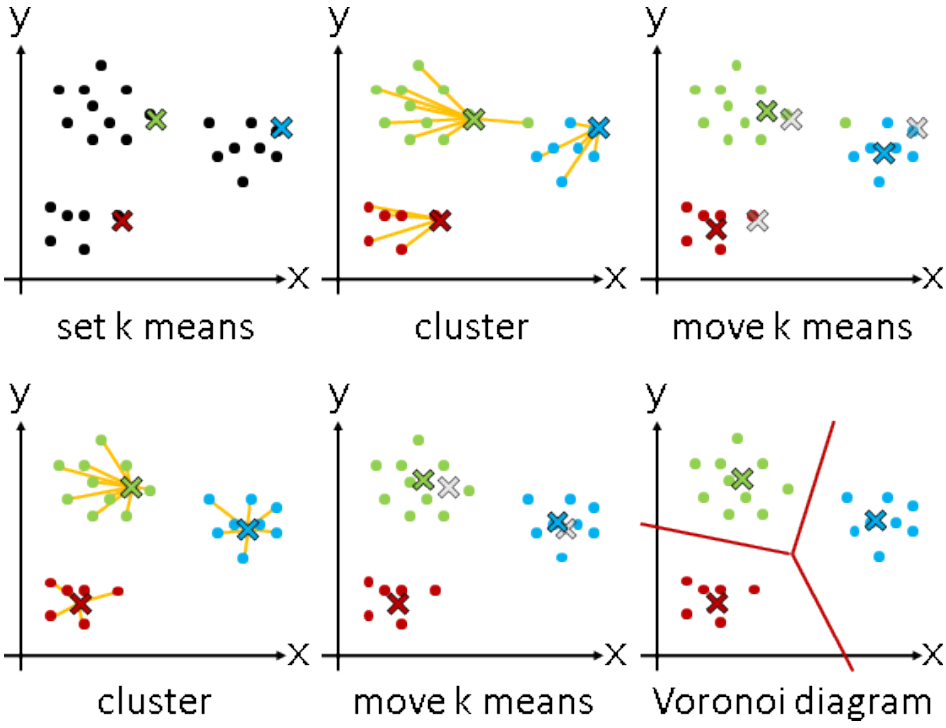
- ดับเบิ้ลคลิกที่ไอคอน **Data Sampler** เพื่อเลือกขนาดของข้อมูลที่จะทำการทดสอบ
- จากนั้น ดับเบิ้ลคลิกที่ไอคอน **Test & Score** เพื่อดูผลลัพธ์ที่ได้จาก Linear Regression วิธีที่ใช้วัดความถูกต้องของ Regression ประกอบด้วย Mean Square Error (MSE), Root MSE (RMSE), Mean Absolute Error (MAE) และ R2



- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อดูผลลัพธ์ที่ได้จากการพยากรณ์ โดยการพยากรณ์จะไม่ได้พยากรณ์ออกมาเป็น Class แต่จะพยากรณ์ออกมาเป็นตัวเลข ดังนั้น จึงจำเป็นต้องใช้ค่า Error ในการหาความถูกต้องของอัลกอริทึม

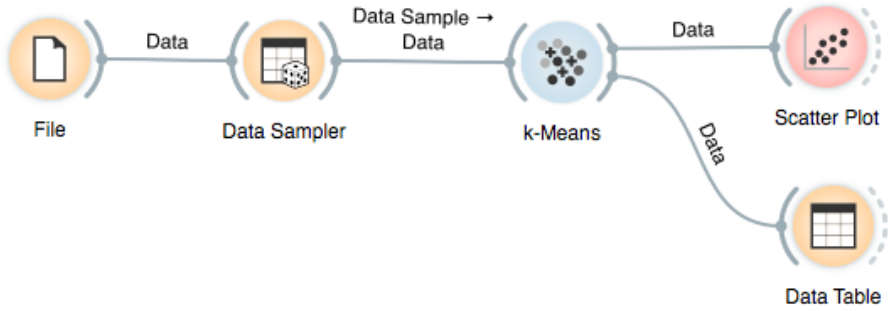
K-Means Clustering

- K-Means Clustering เป็นวิธีการในการแบ่งกลุ่มข้อมูล (Clustering) ซึ่งเป็นการเรียนรู้แบบไม่มีการเรียนการสอน (Unsupervised Learning)
- การเรียนรู้เริ่มต้นด้วยการกำหนดจำนวนกลุ่ม และจุดเริ่มต้นของกลุ่ม (Centroid) ให้กับชุดข้อมูลโดยการสุ่ม
- ข้อมูลแต่ละชุด (Instance) จะถูกนำไปคำนวณเพื่อหาค่าระยะห่าง (Distance Function) กับ Centroid ทั้งหมด และหากข้อมูลชุดนั้นมีค่าใกล้เคียงกับ Centroid ของกลุ่มไหนที่สุด ข้อมูลชุดนั้นจะถูกกำหนดให้เป็นสมาชิกของ Centroid กลุ่มนั้น
- ทำการเรียนรู้ไปเรื่อย ๆ จนกว่าสมาชิกของแต่ละ Centroid จะไม่มีการเปลี่ยนแปลง หรือตามจำนวนรอบ (Iteration) ที่ได้กำหนดไว้

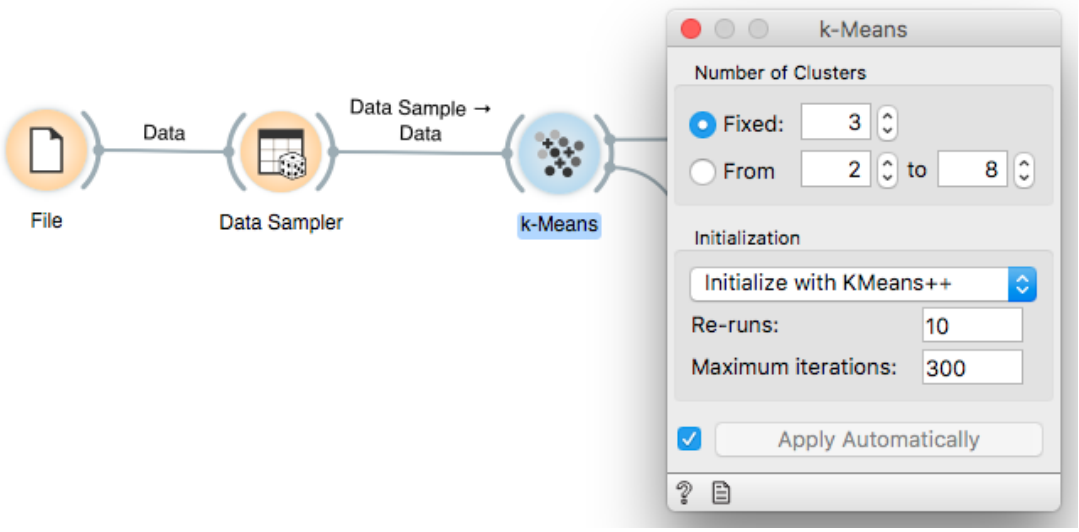


[ที่มา: https://goo.gl/QEmBVp]

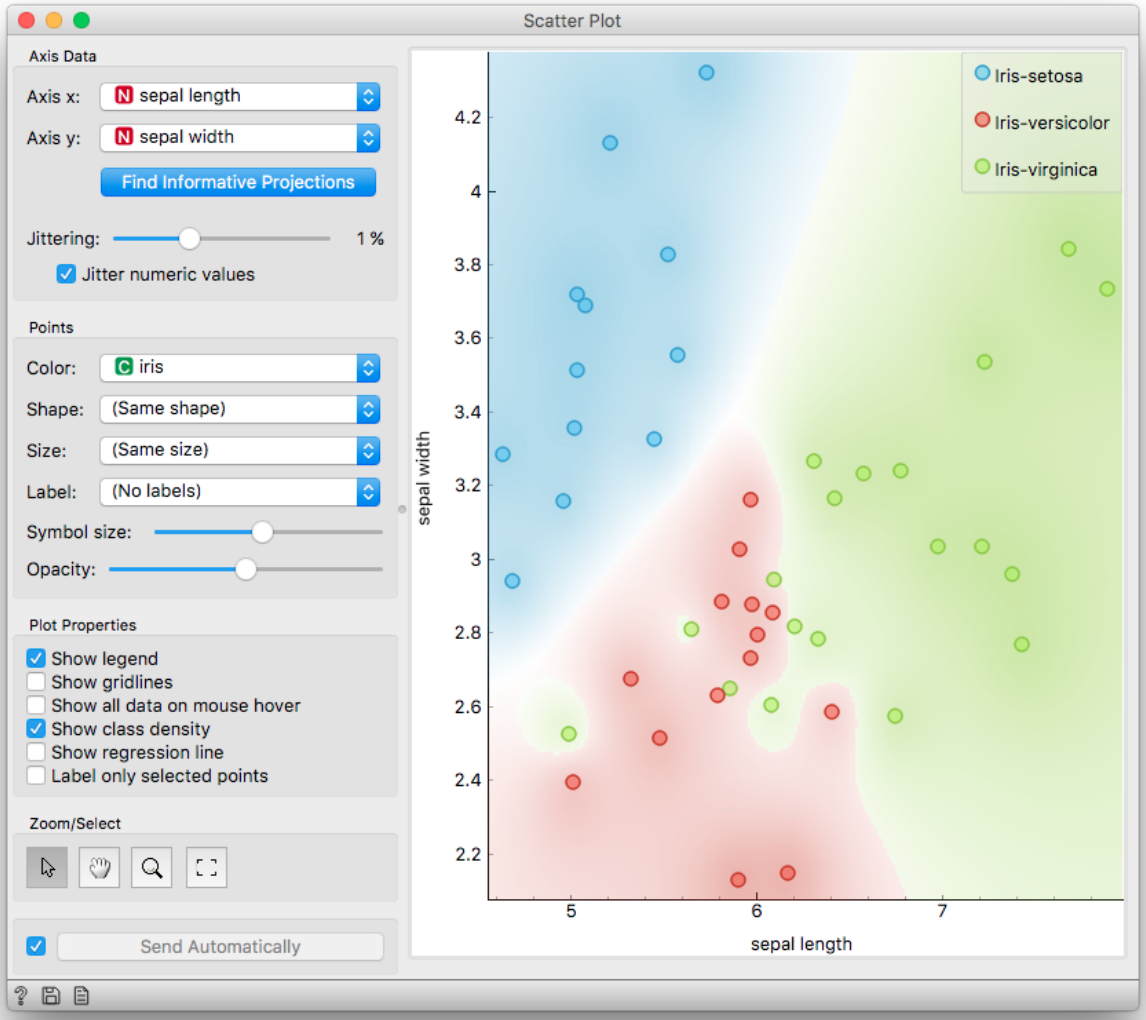
การทำงานของ K-Means สามารถสร้าง workflow ตามตัวอย่างดังต่อไปนี้



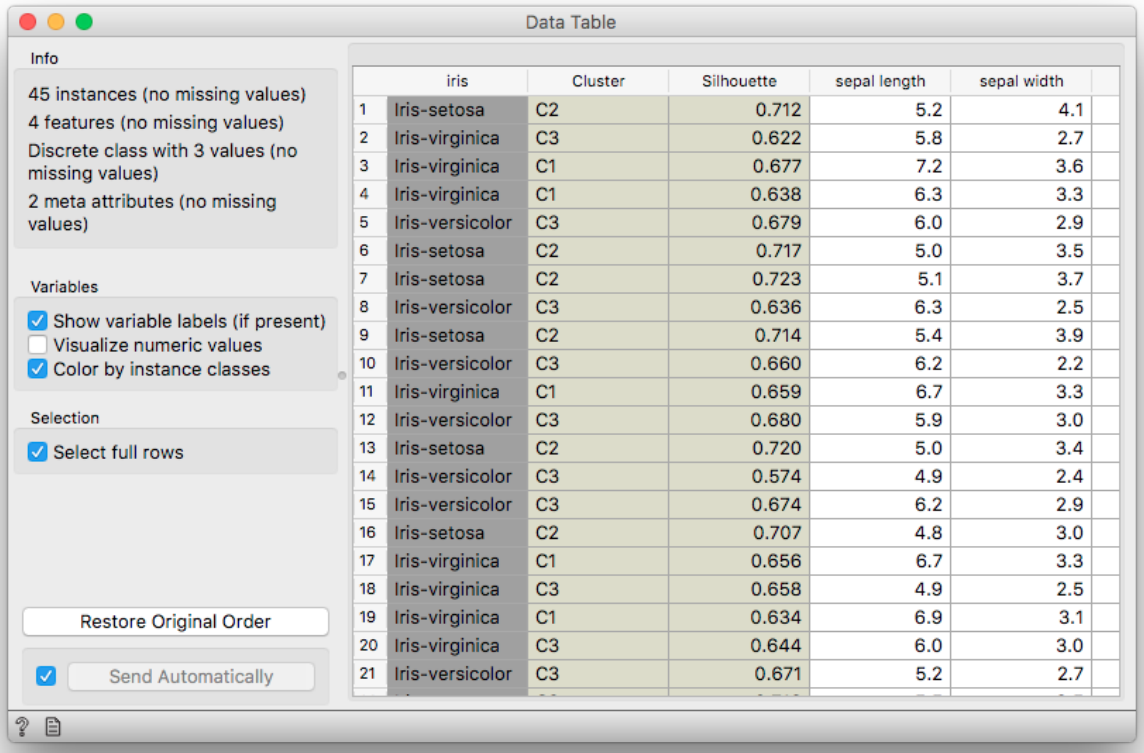
- เมื่อสร้าง workflow ดังตัวอย่างข้างต้น จากนั้นดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูล iris
- ดับเบิ้ลคลิกที่ไอคอน **Data Sampler** เพื่อกำหนดขนาดข้อมูลที่ใช้ในการเรียนรู้ และทดสอบ
- ดับเบิ้ลคลิกที่ไอคอน K-means เพื่อกำหนด parameter ดังต่อไปนี้
  - หากทราบจำนวนของกลุ่มที่แน่นอน สามารถระบุจำนวนกลุ่มที่ต้องการ ให้เลือกที่ **Fixed** ในกรณีนี้ได้เลือกที่ Fixed และกำหนดให้เป็น **3 กลุ่ม**
  - หากไม่ทราบจำนวนที่แน่นอนของกลุ่ม สามารถทำการทดสอบได้ว่าจำนวนกลุ่มเท่าไหร่ที่เหมาะสม สามารถเลือกได้จาก **From …. to …..** โดยกำหนดจำนวนกลุ่มที่ต้องการเช่น **From 2 to 8**
  - จำนวนรอบสูงสุดที่ใช้ในการเรียนรู้ **Maximum iterations** กำหนดไว้ 300 รอบ แต่ทั้งนี้หากข้อมูลไม่มีการเปลี่ยนแปลงโปรแกรมจะหยุดการทำงานก่อนครบ 300 รอบ



- ดับเบิ้ลคลิกที่ไอคอน Scatter Plot เพื่อดูการแบ่งข้อมูลแบบ Visualize



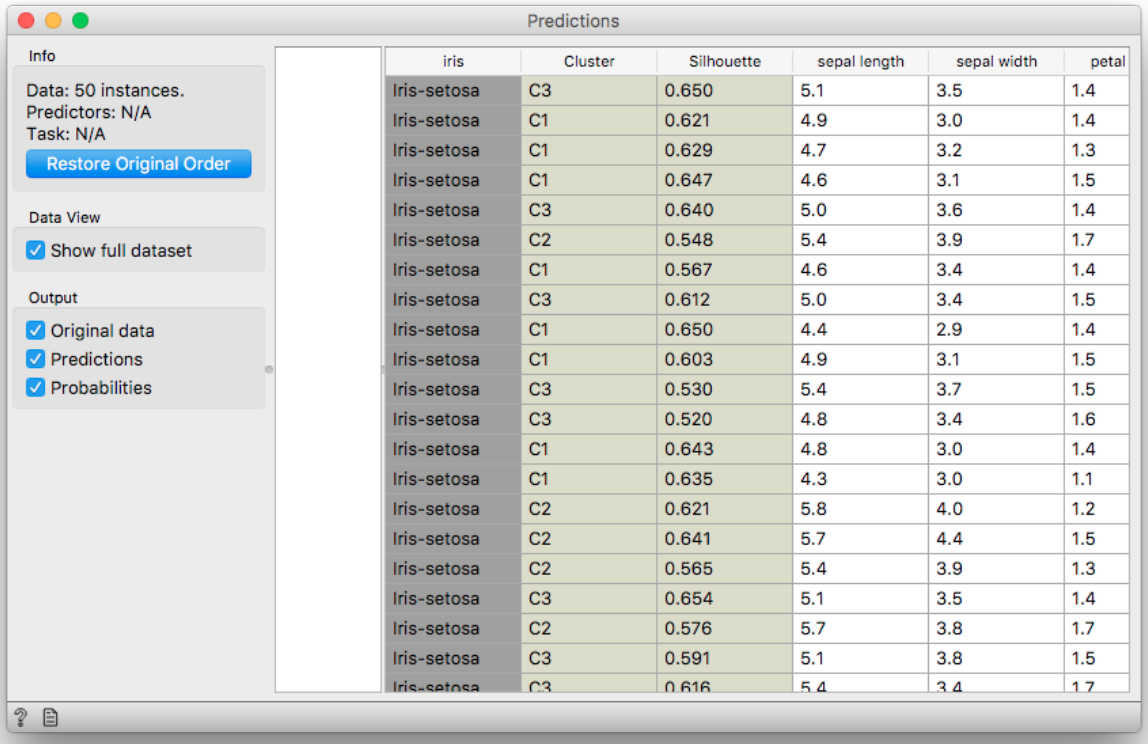
- ดับเบิ้ลคลิกที่ไอคอน **Data Table** เพื่อดูผลลัพธ์ที่ได้จากการพยากรณ์
  - ผลลัพธ์จากการพยากรณ์จะให้คำตอบเป็น C1, C2 และ C3 แทนคำตอบ



	iris	Cluster	Silhouette	sepal length	sepal width
1	Iris-setosa	C2	0.712	5.2	4.1
2	Iris-virginica	C3	0.622	5.8	2.7
3	Iris-virginica	C1	0.677	7.2	3.6
4	Iris-virginica	C1	0.638	6.3	3.3
5	Iris-versicolor	C3	0.679	6.0	2.9
6	Iris-setosa	C2	0.717	5.0	3.5
7	Iris-setosa	C2	0.723	5.1	3.7
8	Iris-versicolor	C3	0.636	6.3	2.5
9	Iris-setosa	C2	0.714	5.4	3.9
10	Iris-versicolor	C3	0.660	6.2	2.2
11	Iris-virginica	C1	0.659	6.7	3.3
12	Iris-versicolor	C3	0.680	5.9	3.0
13	Iris-setosa	C2	0.720	5.0	3.4
14	Iris-versicolor	C3	0.574	4.9	2.4
15	Iris-versicolor	C3	0.674	6.2	2.9
16	Iris-setosa	C2	0.707	4.8	3.0
17	Iris-virginica	C1	0.656	6.7	3.3
18	Iris-virginica	C3	0.658	4.9	2.5
19	Iris-virginica	C1	0.634	6.9	3.1
20	Iris-virginica	C3	0.644	6.0	3.0
21	Iris-versicolor	C3	0.671	5.2	2.7

การทำงานของ K-Means Clustering สามารถสร้าง workflow ที่แตกต่างกันออกไปได้ เช่น

- ในการกำหนดค่าต่าง ๆ กำหนดเหมือนกับตัวอย่าง K-Means Clustering ก่อนหน้านี้
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อดูผลลัพธ์ที่ได้จากการพยากรณ์
  - ผลลัพธ์จากการพยากรณ์จะให้คำตอบเป็น C1, C2 และ C3 แทนคำตอบ



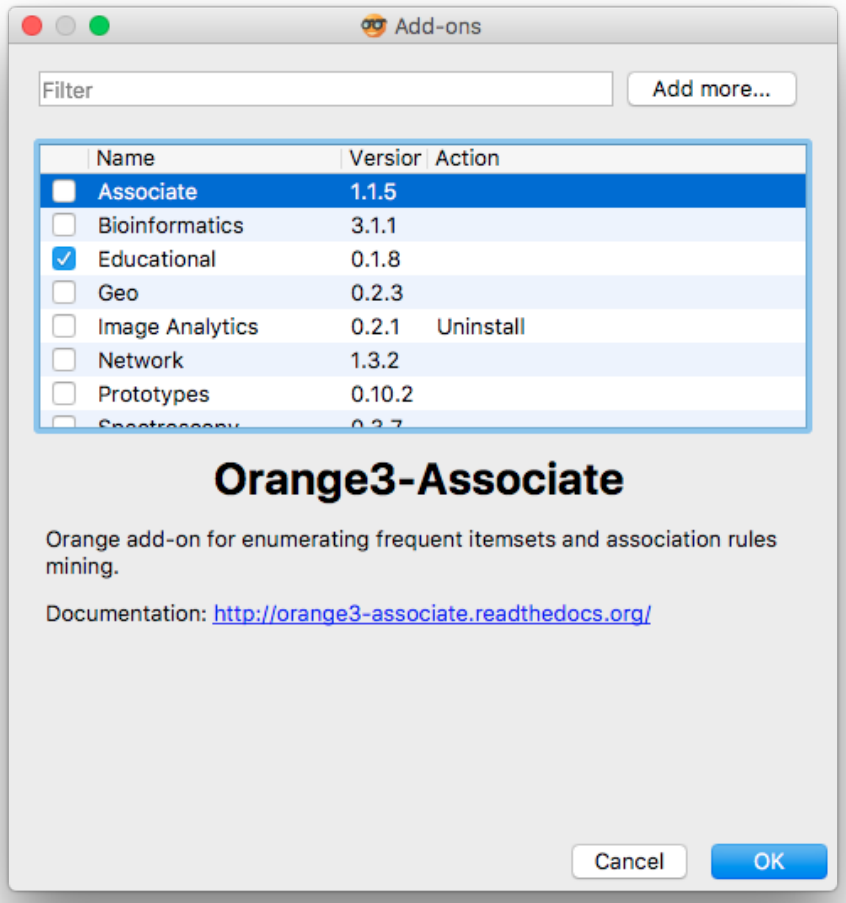
iris	Cluster	Silhouette	sepal length	sepal width	petal
Iris-setosa	C3	0.650	5.1	3.5	1.4
Iris-setosa	C1	0.621	4.9	3.0	1.4
Iris-setosa	C1	0.629	4.7	3.2	1.3
Iris-setosa	C1	0.647	4.6	3.1	1.5
Iris-setosa	C3	0.640	5.0	3.6	1.4
Iris-setosa	C2	0.548	5.4	3.9	1.7
Iris-setosa	C1	0.567	4.6	3.4	1.4
Iris-setosa	C3	0.612	5.0	3.4	1.5
Iris-setosa	C1	0.650	4.4	2.9	1.4
Iris-setosa	C1	0.603	4.9	3.1	1.5
Iris-setosa	C3	0.530	5.4	3.7	1.5
Iris-setosa	C3	0.520	4.8	3.4	1.6
Iris-setosa	C1	0.643	4.8	3.0	1.4
Iris-setosa	C1	0.635	4.3	3.0	1.1
Iris-setosa	C2	0.621	5.8	4.0	1.2
Iris-setosa	C2	0.641	5.7	4.4	1.5
Iris-setosa	C2	0.565	5.4	3.9	1.3
Iris-setosa	C3	0.654	5.1	3.5	1.4
Iris-setosa	C2	0.576	5.7	3.8	1.7
Iris-setosa	C3	0.591	5.1	3.8	1.5
Iris-setosa	C3	0.616	5.4	3.4	1.7

Interactive k-Means

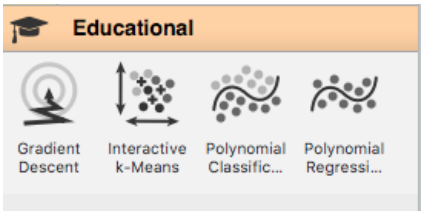
- การใช้งาน K-Means ในรูปแบบของการปฏิสัมพันธ์ (Interactive) จะต้องลงโปรแกรมเสริม (Add-ons…)

การติดตั้งโปรแกรม Add-on (Installing Add-on Program)

- ให้เลือกที่เมนู **Options** > **Add-ons** จากนั้นจะปรากฏหน้าต่าง ดังต่อไปนี้



- ให้คลิกเลือก **Educational** และกดที่ปุ่ม **OK** เพื่อทำการติดตั้ง
- เมื่อติดตั้งเสร็จเรียบร้อยให้ปิด และเปิดโปรแกรม Orange เพื่อทำให้โปรแกรมสามารถใช้ Add-on ที่ติดตั้งลงไปใหม่ได้



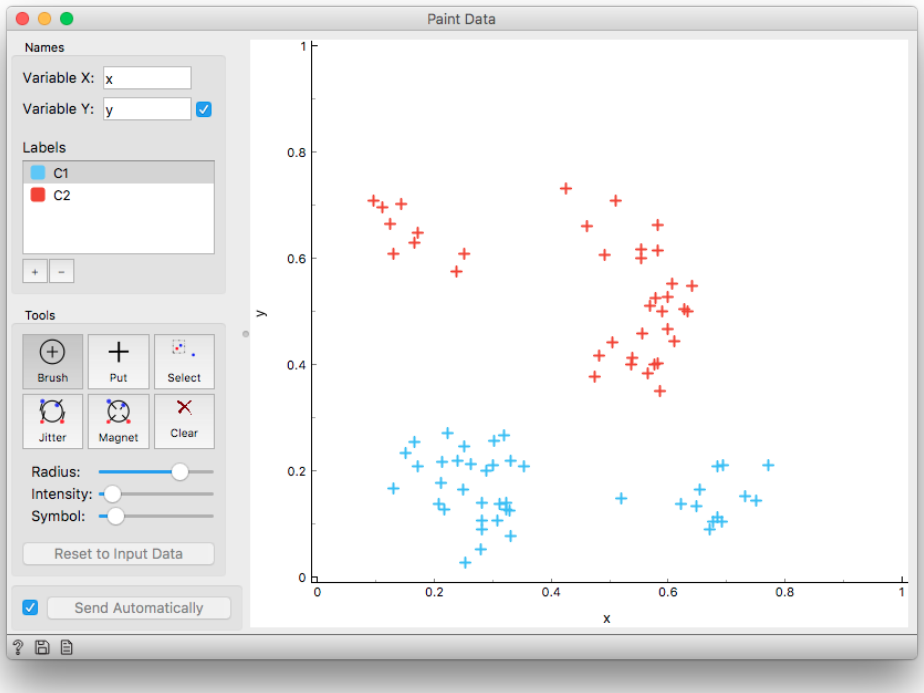
- หากติดตั้งเสร็จเรียบร้อยจะมีแท็ปใหม่ปรากฏ ชื่อว่า **Educational**

การทำ Interactive k-Means สามารถทำได้โดย

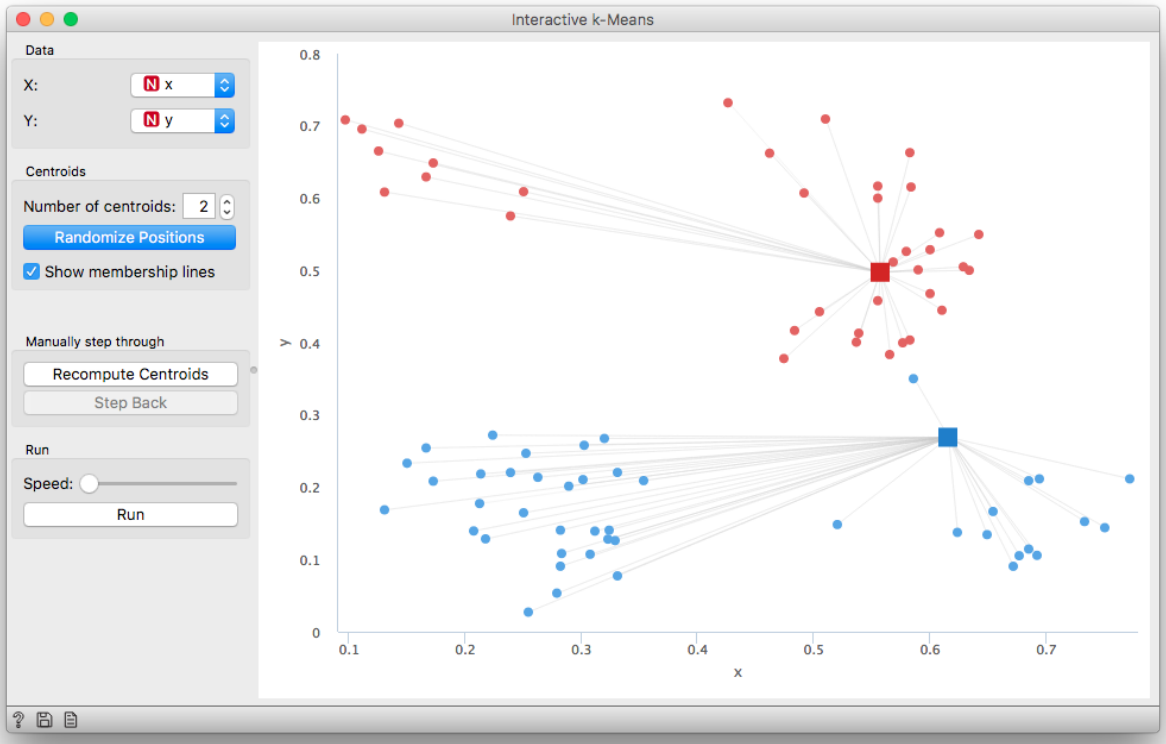
- เปิด workflow ขึ้นใหม่และเพิ่มไอคอนลงไป ดังต่อไปนี้
  - Paint Data
  - Interactive k-Means

การสร้างข้อมูลด้วยไอคอน Paint Data (Creating Data using Paint Data Icon)

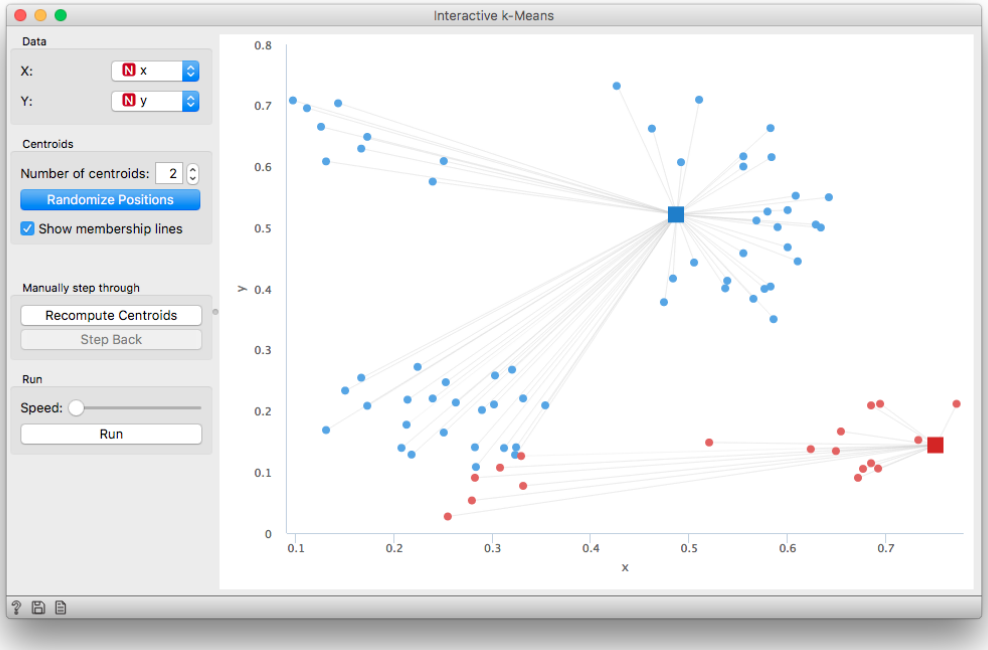
- จากนั้นดับเบิ้ลคลิกที่ **Paint Data** เพื่อสร้างข้อมูล ดังตัวอย่างต่อไปนี้



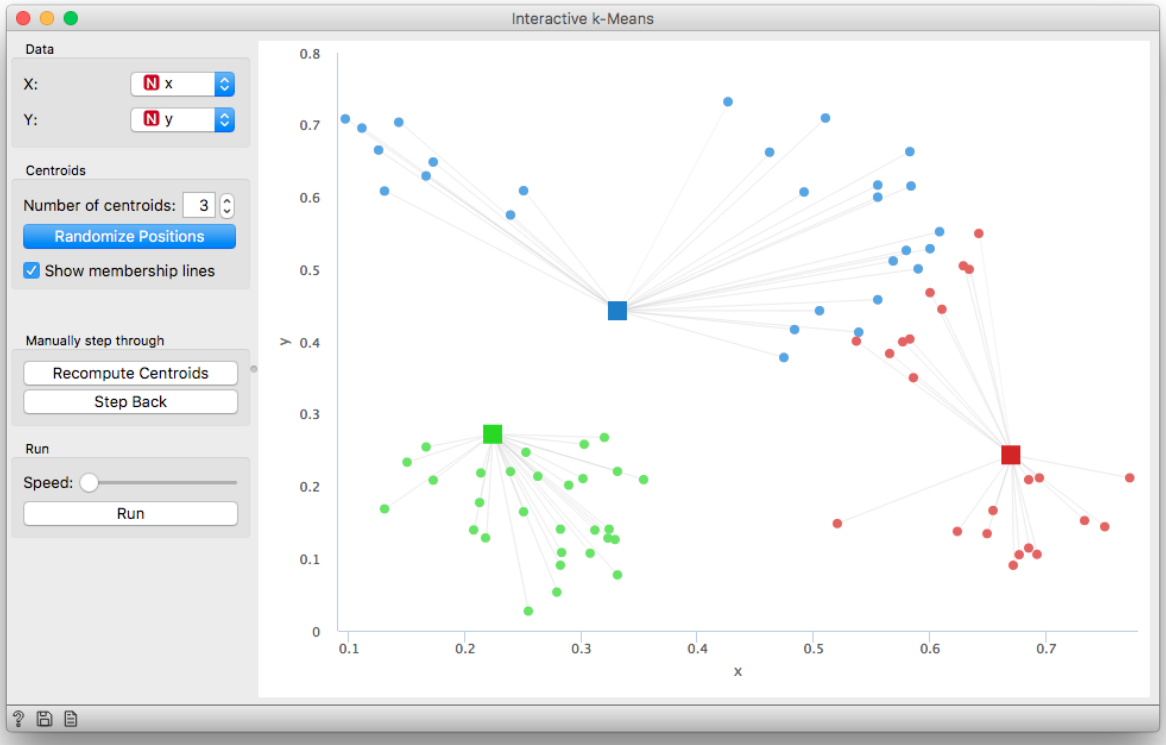
- ขั้นตอนต่อไป ดับเบิ้ลคลิกที่ไอคอน **Interactive k-Means** โปรแกรมจะแสดงการแบ่งกลุ่มข้อมูล ดังตัวอย่างต่อไปนี้



- จากตัวอย่างเราสามารถใช้เมาส์คลิกเลือกจุด Centroid ได้



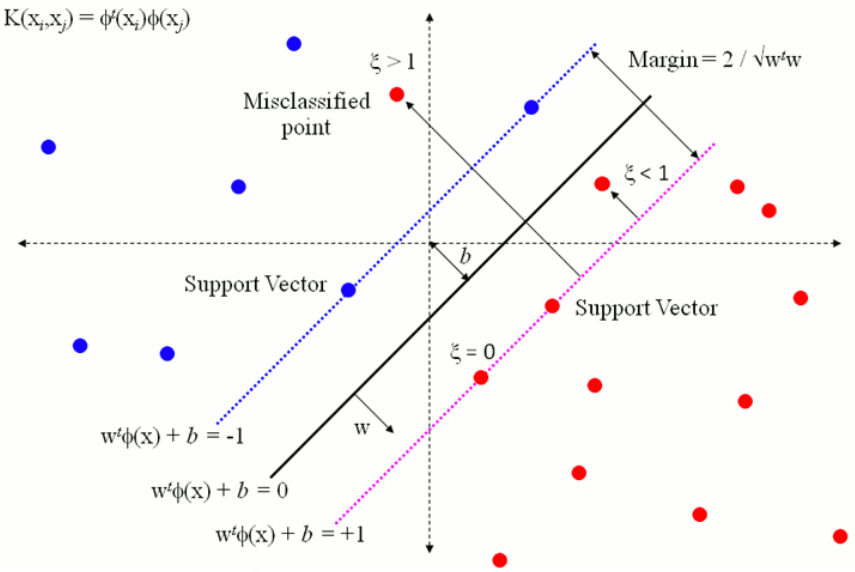
- อีกทั้งยังสามารถกำหนดจำนวนกลุ่มได้



- โปรแกรมจะแสดงการแบ่งกลุ่มแบบ Visualize เพื่อให้สามารถเข้าใจลักษณะของข้อมูลได้ง่ายขึ้น

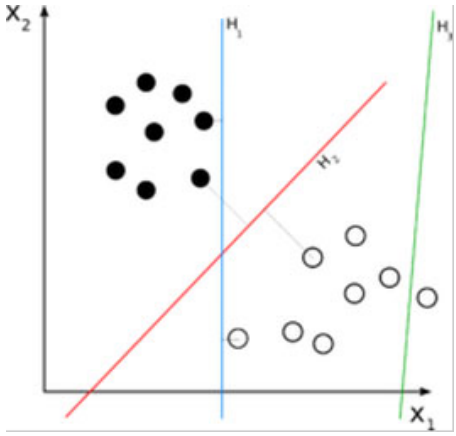
Support Vector Machine (SVM)

- Support Vector Machine (SVM) เป็นอัลกอริทึมที่ใช้ในการจัดหมวดหมู่ข้อมูล (Classification) แรกเริ่มอัลกอริทึม SVM ได้พัฒนาเพื่อการจัดหมวดหมู่ข้อมูลที่มีเพียงสองกลุ่มหรือ Binary Classification โดยใช้เส้นระนาบ (Hyperplane) เป็นตัวแบ่งข้อมูลสองกลุ่มออกจากกัน โดย Hyperplane นั้นจะเป็นเส้นตรง จึงเรียกว่า Linear Kernel แสดงดังภาพต่อไปนี้



[ที่มา: https://goo.gl/5XfMnY]

- จากรูปภาพข้างต้น เส้น hyperplane (เส้นทึบสีดำ) จะใช้เป็นเส้นแบ่งข้อมูล 2 กลุ่มออกจากกัน โดย hyperplane จะมี Margin (เส้นประสีน้ำเงิน) ประกบทั้งสองข้าง หาก Margin ที่ขนาดกว้าง (Maximum Margin) แสดงว่าข้อมูลที่นำมาจัดหมวดหมู่มีความแตกต่างกัน แต่หาก Margin แคบแสดงว่าข้อมูลที่นำมาจัดหมวดหมู่อาจมีความใกล้เคียงกันมาก โดยจุดที่เส้น Margin ลากผ่านคือจุด Support Vector

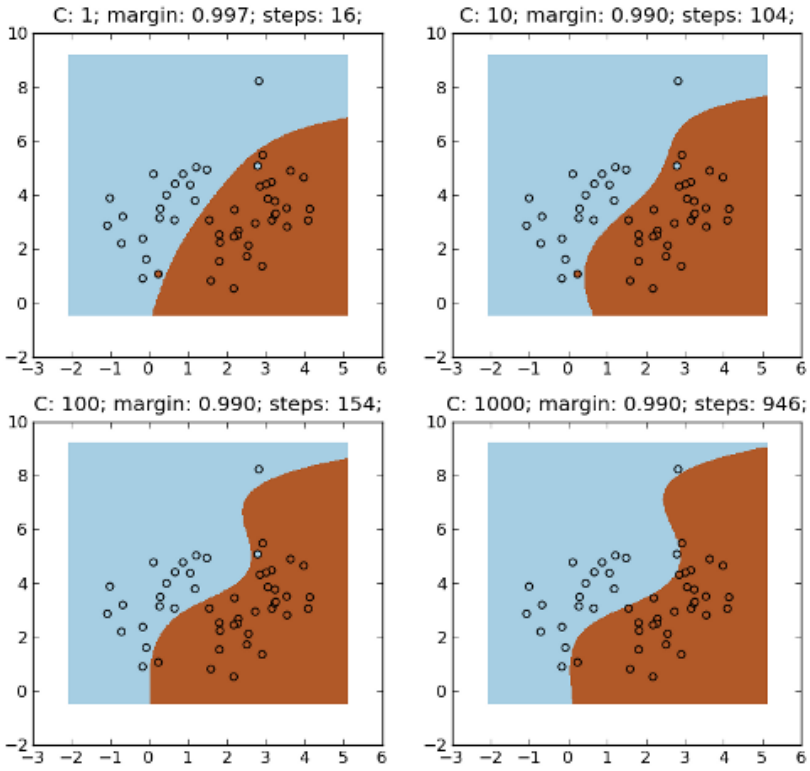


[ที่มา: https://goo.gl/JK4nKH]

- จากรูปภาพข้างต้นได้จำลองเส้น Hyperplane จำนวนสองเส้นคือ H1 และ H2 โดยทั้ง H1 และ H2 สามารถแบ่งข้อมูลออกเป็น 2 กลุ่มได้อย่างถูกต้อง
- แต่ทั้งนี้เส้น H1 มี Margin ที่แคบกว่าเส้น H2 ดังนั้น หากมีข้อมูลใหม่ปรากฏขึ้น ใกล้กับเส้น H1 อาจทำให้การจัดหมวดหมู่ผิดพลาดได้ ดังนั้น เส้น Hyperplane ที่ดีที่สุดคือเส้นที่มี Margin ที่กว้างที่สุด หรือ Maximum Margin

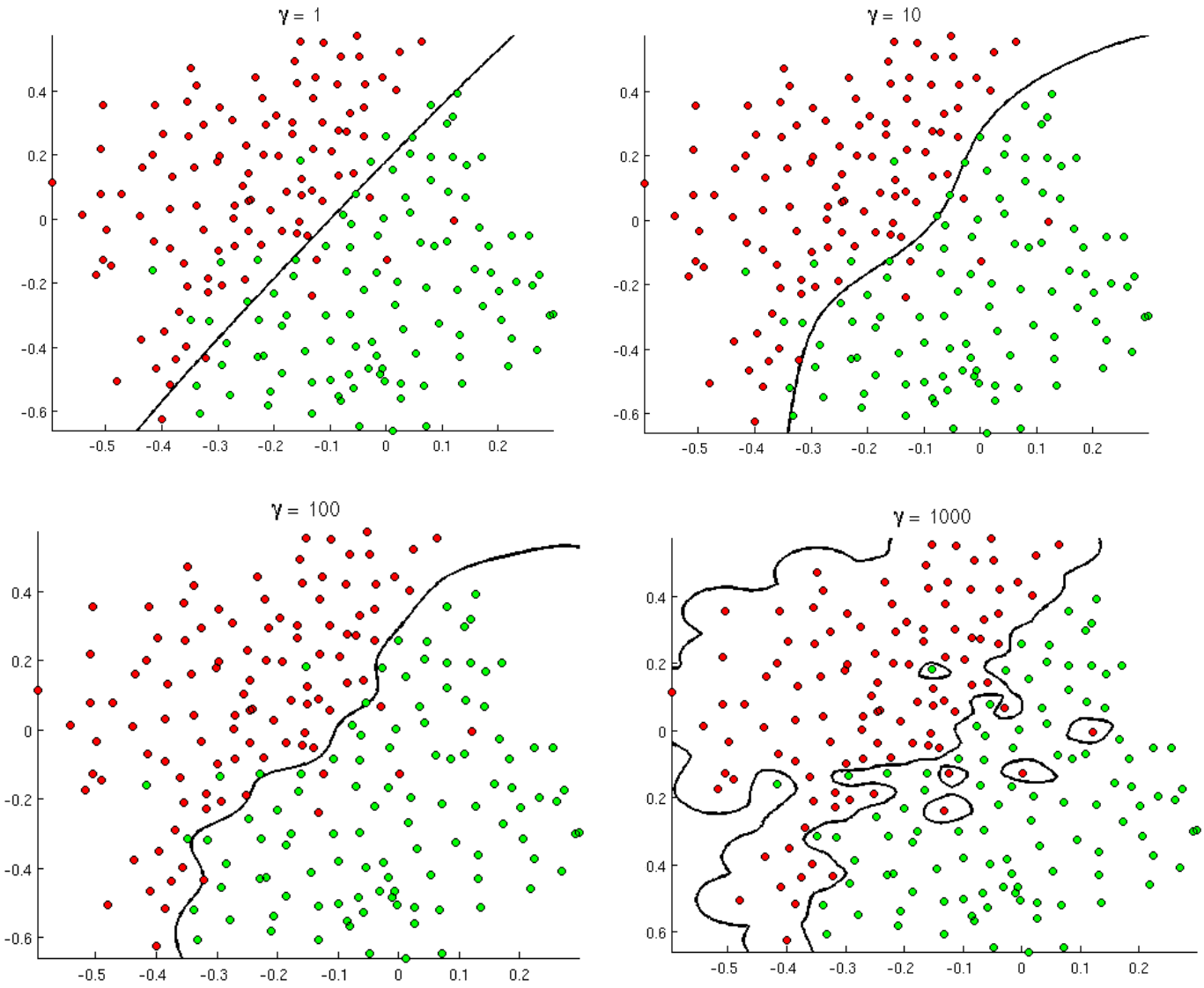
Kernel Function ที่ใช้ใน SVM (SVM Kernel Functions)

- โดยปกติ Kernel Function ที่ใช้ใน SVM คือ Liner Kernel แต่ Linear Kernel นี้เป็นสมการเส้นตรงจึงทำให้อาจเกิดข้อผิดพลาดได้หากข้อมูลมีความซับซ้อน จึงได้มีผู้คิดค้น Kernel Function เพื่อทำให้การจัดหมวดหมู่ข้อมูลทำได้ดีขึ้น
- Kernel Function ที่ใช้ใน SVM ประกอบด้วย
  - Linear Kernel
  - RBF Kernel
  - Polynomial Kernel



[ที่มา: http://mlpy.sourceforge.net/docs/3.4/svm.html]

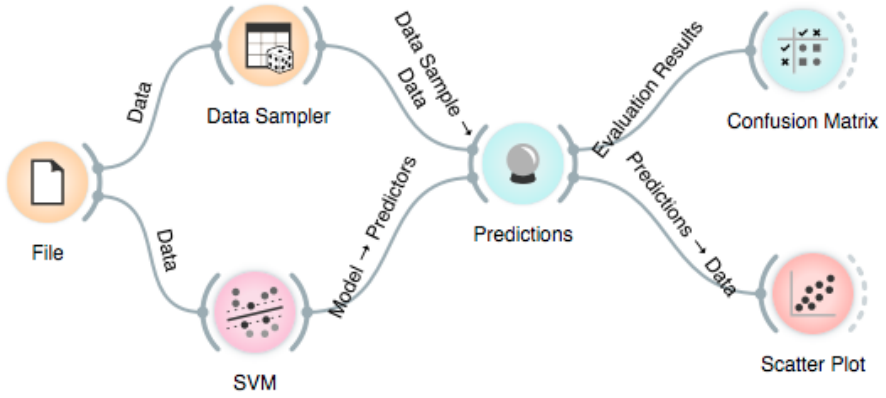
- รูปภาพข้างต้น แสดงให้เห็นถึงการปรับค่าพารามิเตอร์ C ใน RBF Kernel



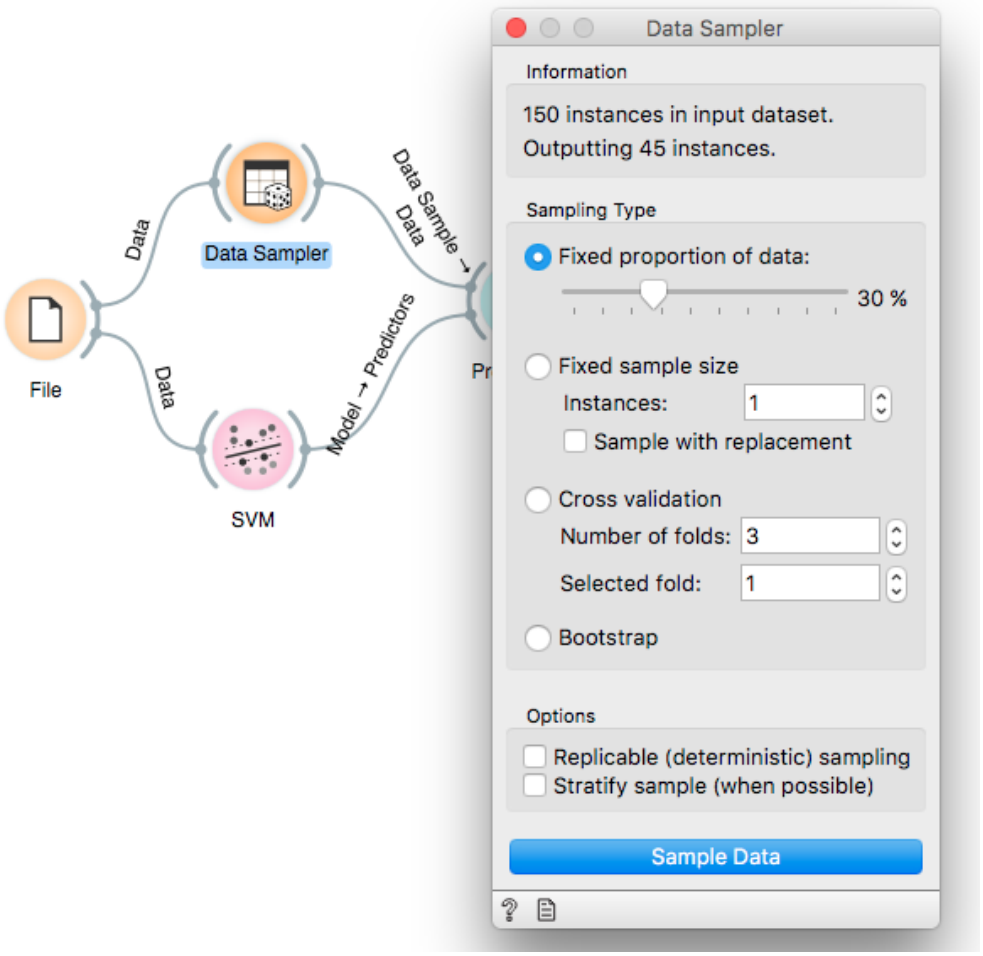
[ที่มา: https://goo.gl/QJSTL9]

- รูปภาพข้างต้น แสดงให้เห็นถึงการปรับค่า gamma เพื่อให้อัลกอริทึมสามารถจัดหมวดหมู่ข้อมูลได้ดีขึ้น แต่ทั้งนี้อาจทำให้เกิดการ Overfitting

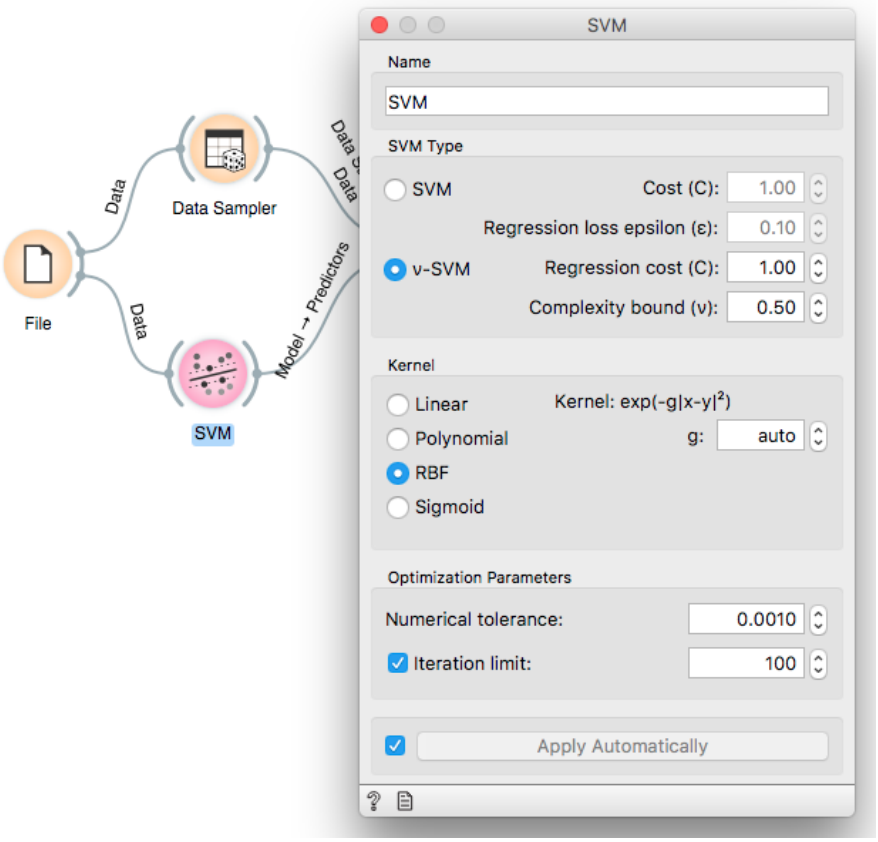
การทำงานของ SVM สามารถสร้าง workflow ตามตัวอย่างดังต่อไปนี้



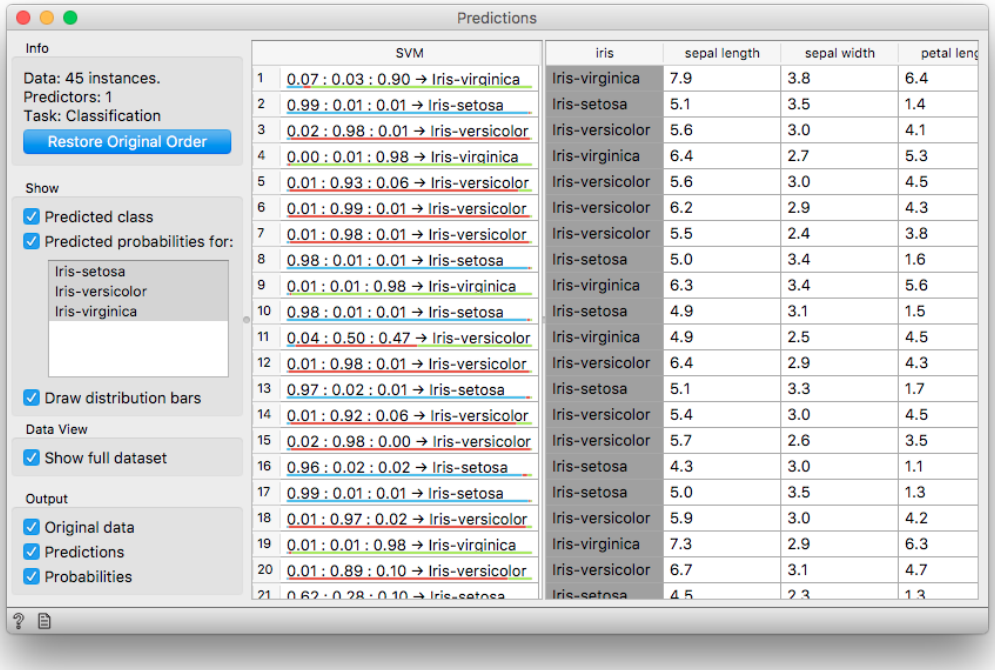
- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูล ในกรณีนี้ใช้ชุดข้อมูล **iris**
- ดับเบิ้ลคลิกที่ไอคอน **Data Sampler** เพื่อสุ่มเลือกข้อมูลที่ใช้ในการสร้างโมเดล และพยากรณ์



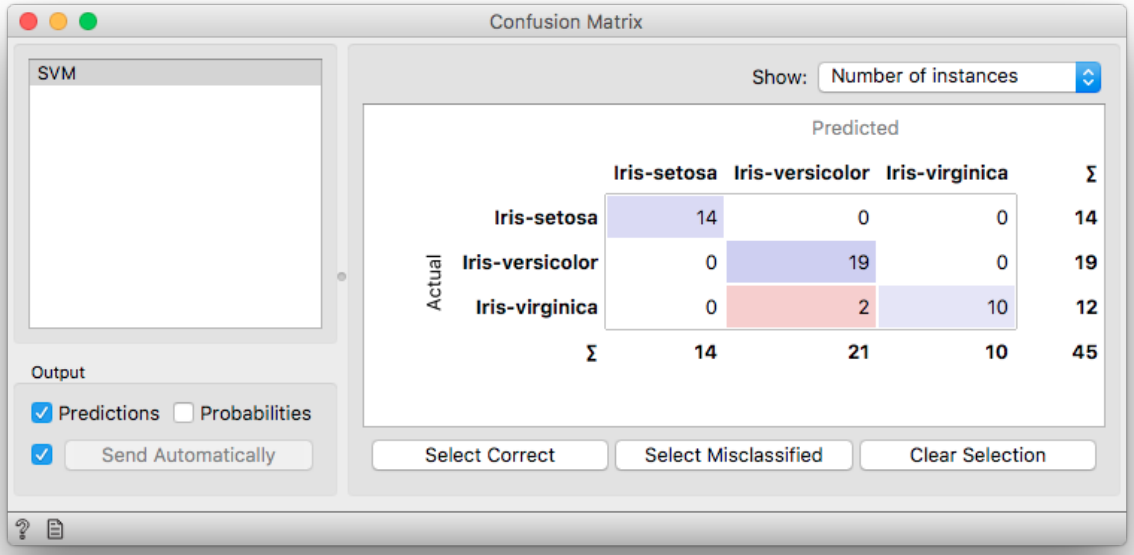
- ดับเบิ้ลคลิกที่ไอคอน **SVM** เพื่อปรับค่าพารามิเตอร์ที่ใช้ในการเรียนรู้

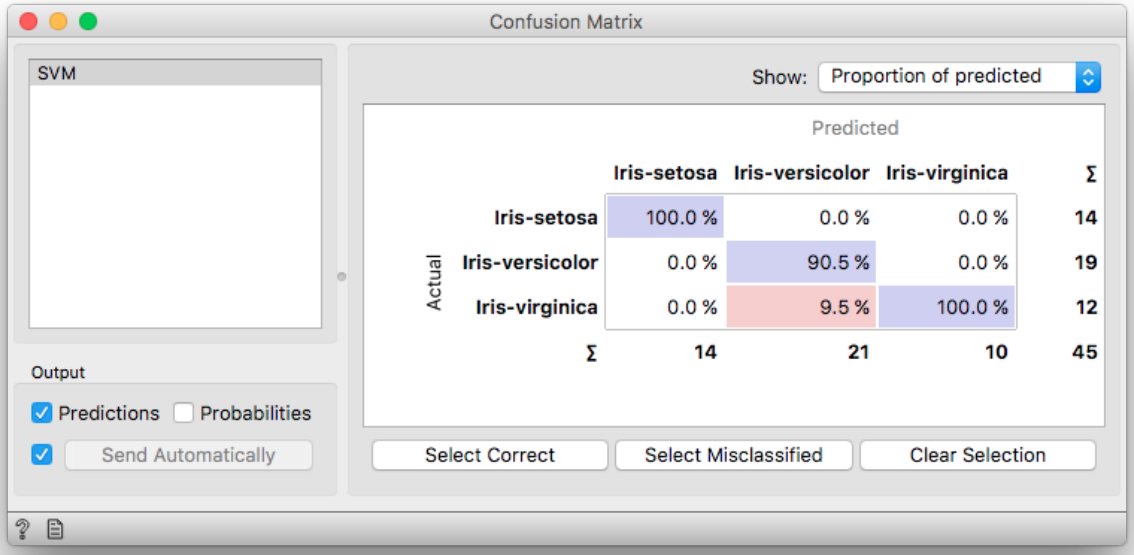


- ดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อดูผลลัพธ์ที่ได้จากการพยากรณ์ด้วยวิธี SVM



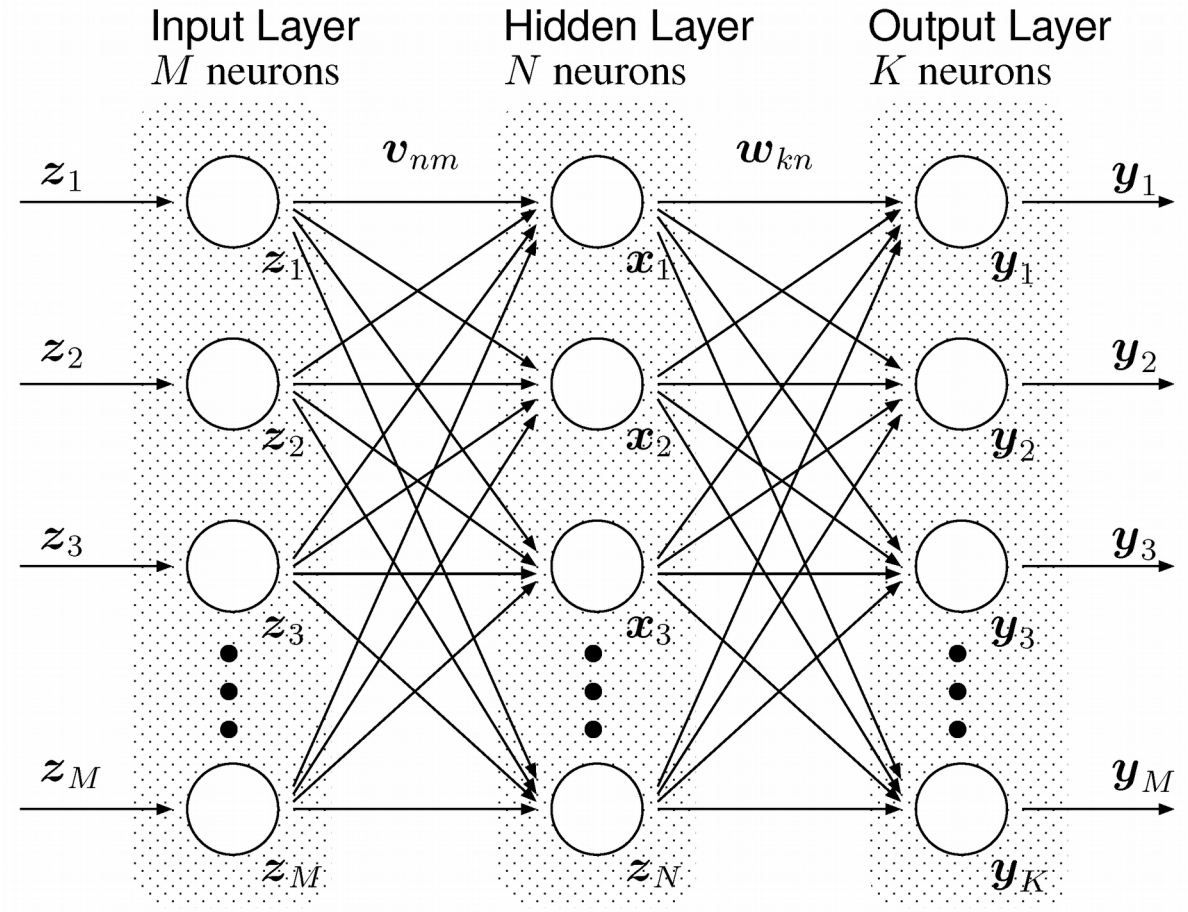
- ดับเบิ้ลคลิกที่ไอคอน **Confusion Matrix** เพื่อดูผลลัพธ์ โดยแสดงผลลัพธ์ตามกลุ่มข้อมูล





- ดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูข้อมูลการจัดหมวดหมู่ด้วย SVM แบบ Visualize

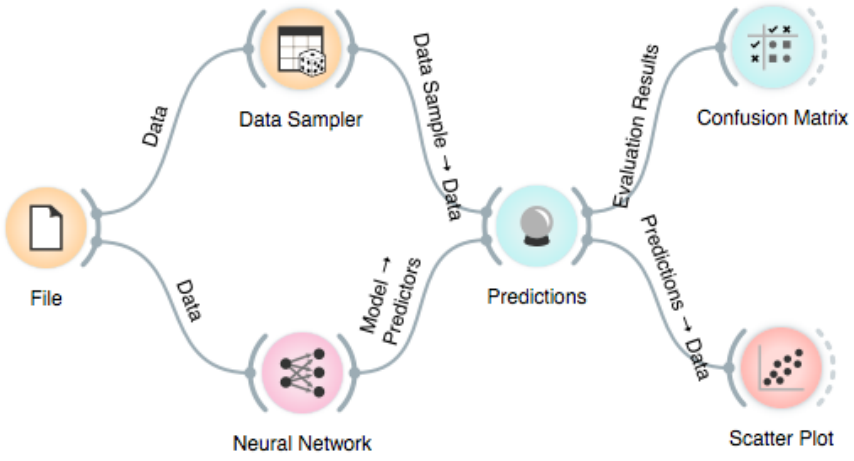
Neural Network



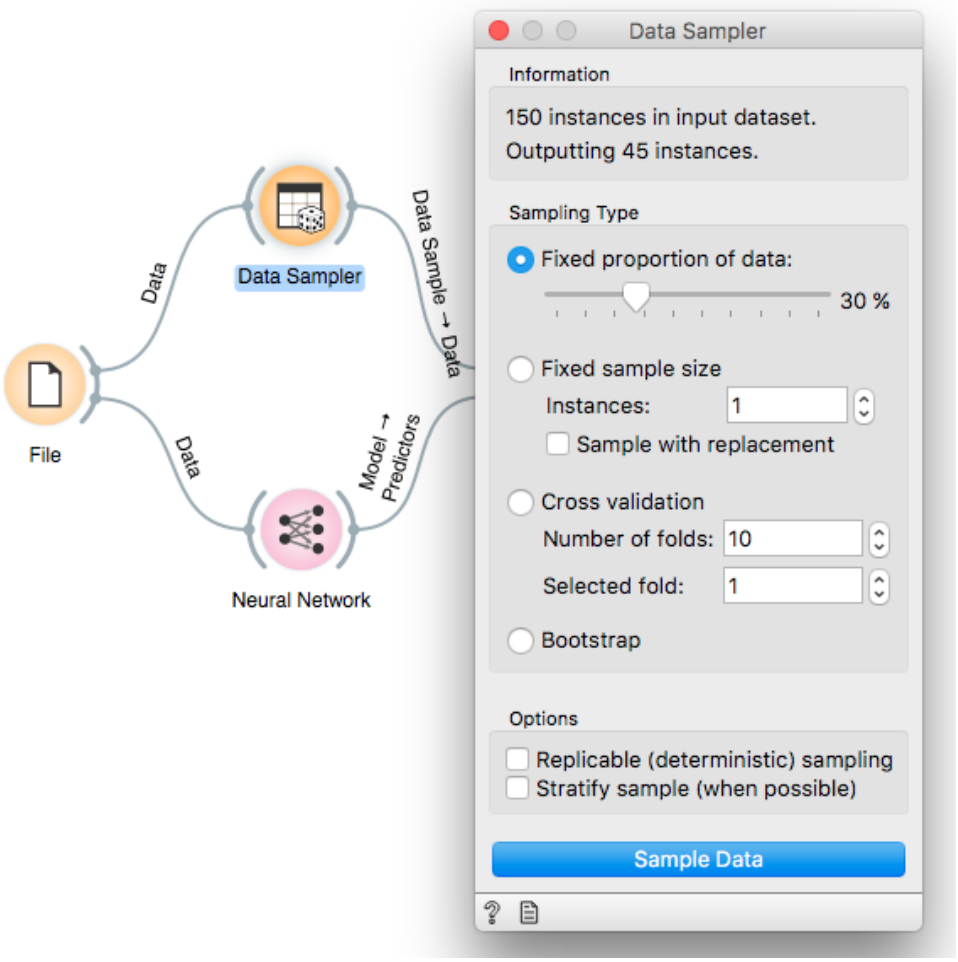
[ที่มา: http://www.mdpi.com/2078-2489/3/4/756]

- Neural Network เป็นวิธีในการจัดหมวดหมู่ข้อมูล ที่อยู่ในกลุ่มของ Supervised Learning ซึ่งหมายถึงการเรียนรู้จะต้องใช้ Label เพื่อช่วยในการเรียนรู้ โดย Neural Network ที่แสดงดังรูปข้างต้นเป็น Network แบบ Multi-Layer Perceptron ซึ่งจะต้องประกอบด้วย Input Layer, Hidden Layer และ Output Layer
- จำนวนของโหนดใน Input Layer จะขึ้นอยู่กับจำนวนของ Attribute หรือ Feature
- จำนวนของโหนดใน Output Layer ขึ้นอยู่กับจำนวนของ Class เช่น หากใช้ชุดข้อมูล iris ซึ่งมี 3 Class ดังนั้น Output Layer จะมีจำนวน 3 โหนด

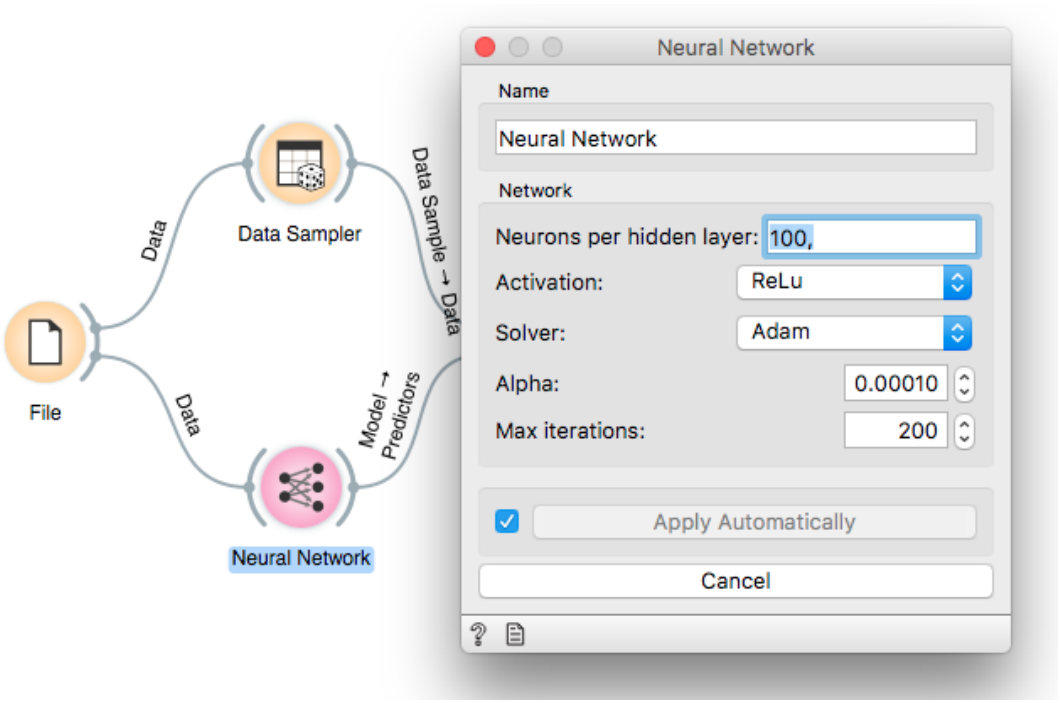
การทำงานของ Neural Network สามารถสร้าง workflow ตามตัวอย่างดังต่อไปนี้



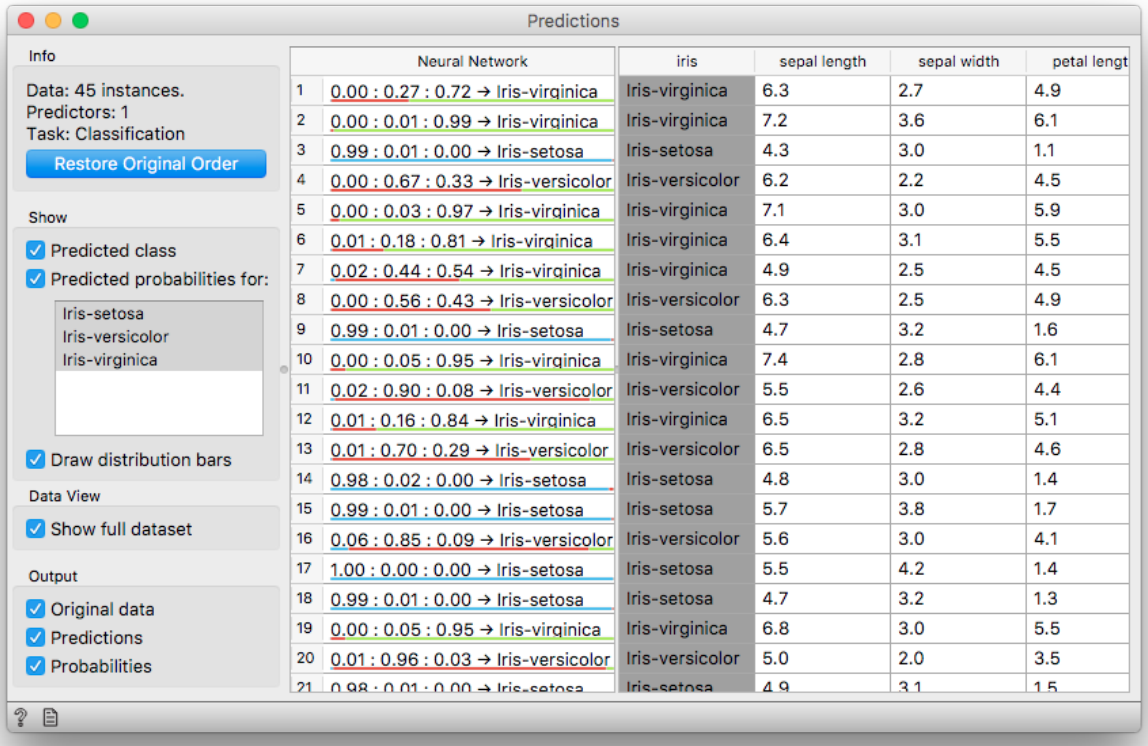
- ดับเบิ้ลคลิกที่ไอคอน **File** เพื่อเลือกชุดข้อมูล ในกรณีนี้ใช้ชุดข้อมูล **iris**
- ดับเบิ้ลคลิกที่ไอคอน **Data Sampler** เพื่อสุ่มเลือกข้อมูลที่ใช้ในการสร้างโมเดล และพยากรณ์



- ดับเบิ้ลคลิกที่ไอคอน **Neural Network** เพื่อปรับค่าพารามิเตอร์ที่ใช้ในการเรียนรู้

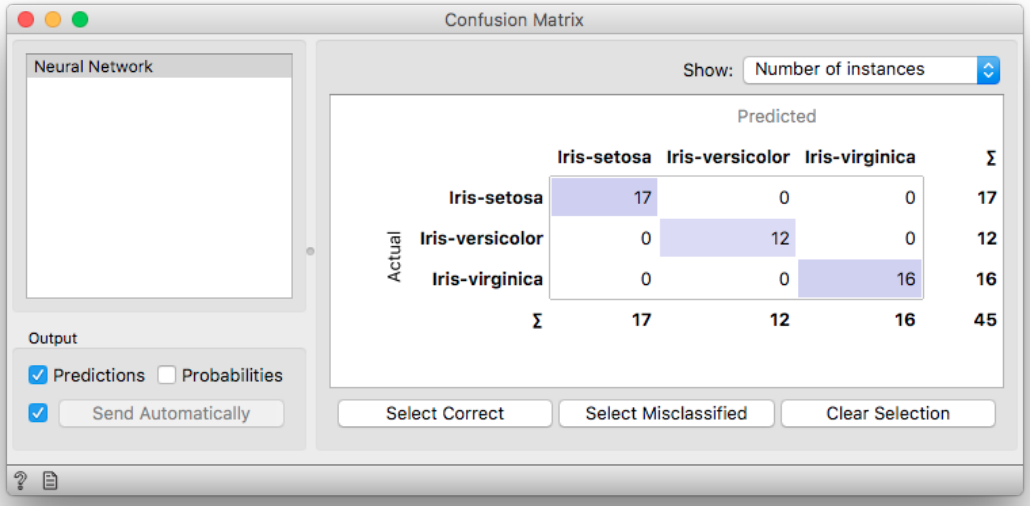


- ดับเบิ้ลคลิกที่ไอคอน **Predictions** เพื่อดูผลลัพธ์ที่ได้จากการพยากรณ์ด้วยวิธี Neural Network

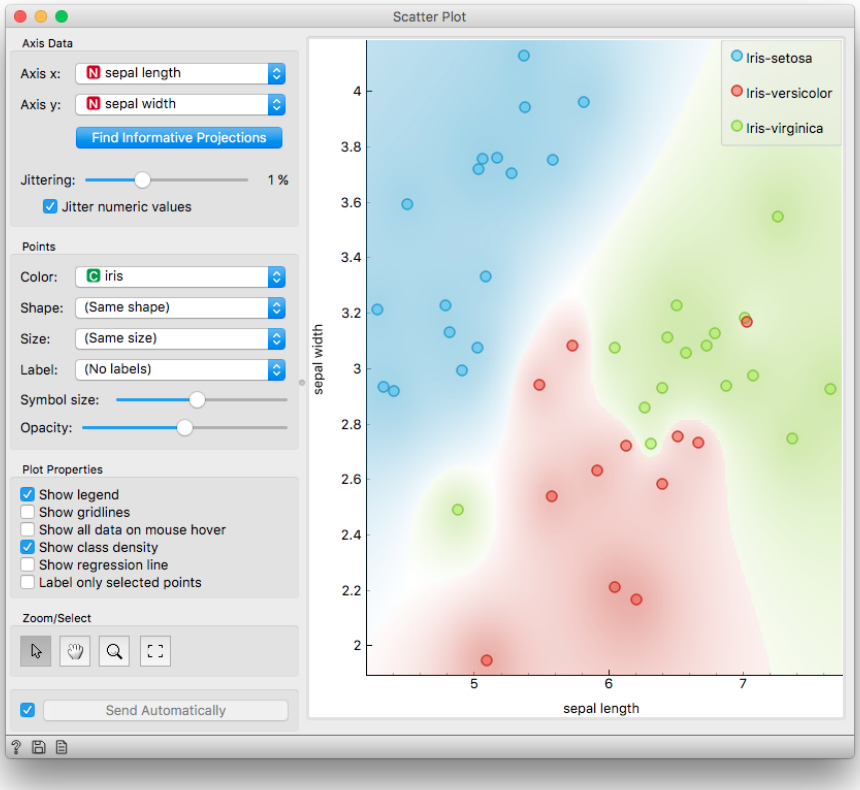


	Neural Network	iris	sepal length	sepal width	petal lengt
1	0.00 : 0.27 : 0.72 → Iris-virginica	Iris-virginica	6.3	2.7	4.9
2	0.00 : 0.01 : 0.99 → Iris-virginica	Iris-virginica	7.2	3.6	6.1
3	0.99 : 0.01 : 0.00 → Iris-setosa	Iris-setosa	4.3	3.0	1.1
4	0.00 : 0.67 : 0.33 → Iris-versicolor	Iris-versicolor	6.2	2.2	4.5
5	0.00 : 0.03 : 0.97 → Iris-virginica	Iris-virginica	7.1	3.0	5.9
6	0.01 : 0.18 : 0.81 → Iris-virginica	Iris-virginica	6.4	3.1	5.5
7	0.02 : 0.44 : 0.54 → Iris-virginica	Iris-virginica	4.9	2.5	4.5
8	0.00 : 0.56 : 0.43 → Iris-versicolor	Iris-versicolor	6.3	2.5	4.9
9	0.99 : 0.01 : 0.00 → Iris-setosa	Iris-setosa	4.7	3.2	1.6
10	0.00 : 0.05 : 0.95 → Iris-virginica	Iris-virginica	7.4	2.8	6.1
11	0.02 : 0.90 : 0.08 → Iris-versicolor	Iris-versicolor	5.5	2.6	4.4
12	0.01 : 0.16 : 0.84 → Iris-virginica	Iris-virginica	6.5	3.2	5.1
13	0.01 : 0.70 : 0.29 → Iris-versicolor	Iris-versicolor	6.5	2.8	4.6
14	0.98 : 0.02 : 0.00 → Iris-setosa	Iris-setosa	4.8	3.0	1.4
15	0.99 : 0.01 : 0.00 → Iris-setosa	Iris-setosa	5.7	3.8	1.7
16	0.06 : 0.85 : 0.09 → Iris-versicolor	Iris-versicolor	5.6	3.0	4.1
17	1.00 : 0.00 : 0.00 → Iris-setosa	Iris-setosa	5.5	4.2	1.4
18	0.99 : 0.01 : 0.00 → Iris-setosa	Iris-setosa	4.7	3.2	1.3
19	0.00 : 0.05 : 0.95 → Iris-virginica	Iris-virginica	6.8	3.0	5.5
20	0.01 : 0.96 : 0.03 → Iris-versicolor	Iris-versicolor	5.0	2.0	3.5
21	0.98 : 0.01 : 0.00 → Iris-setosa	Iris-setosa	4.9	3.1	1.5

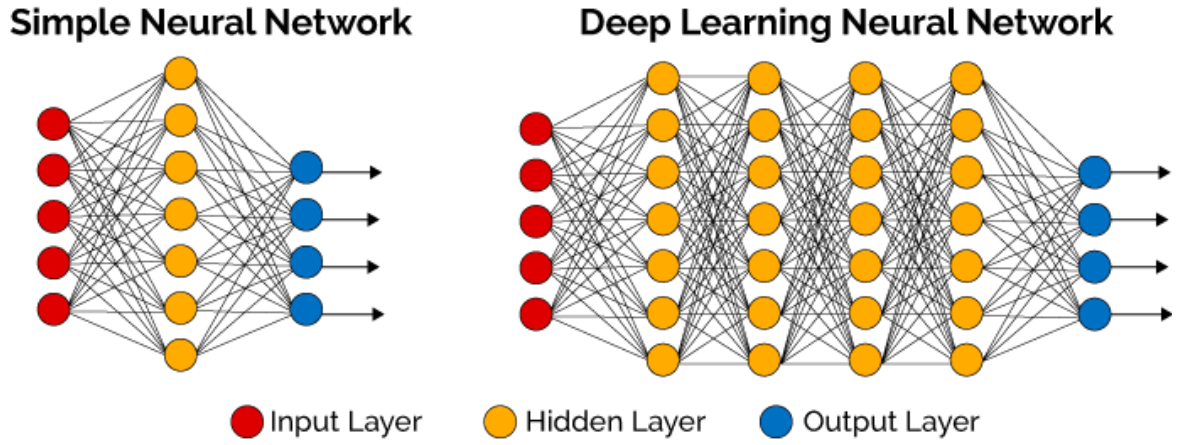
- ดับเบิ้ลคลิกที่ไอคอน **Confusion Matrix** เพื่อดูผลลัพธ์ โดยแสดงผลลัพธ์ตามกลุ่มข้อมูล



- ดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูข้อมูลการจัดหมวดหมู่ด้วย Neural Network แบบ Visualize



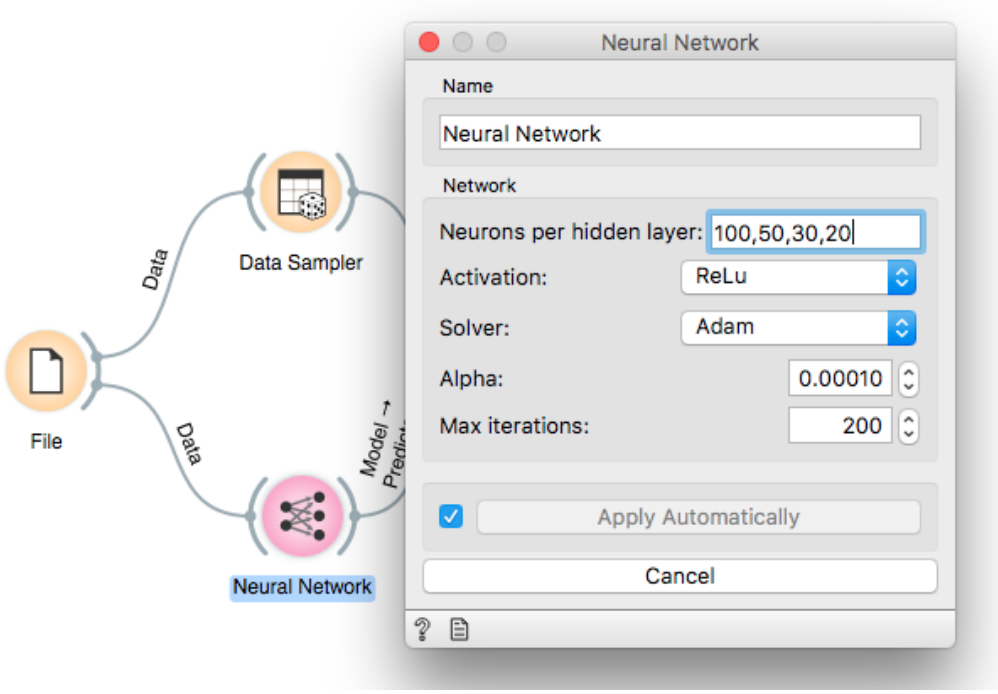
Deep Neural Network



[ที่มา: https://goo.gl/CqSmbq]

- ความแตกต่างระหว่าง Neural Network หรืออาจจะเรียกว่า Simple Neural Network และ Deep Neural Network ก็คือจำนวนชั้นของ Layer
- โดยปกติแล้ว Neural Network แบบ Multi-Layer Perceptron (MLP) จะประกอบด้วย Network จำนวน 3 ชั้นประกอบด้วย
  - ชั้นนำเข้า (Input Layer)
  - ชั้นซ่อน (Hidden Layer)
  - ชั้นแสดงผล (Output Layer)
- แต่ในส่วนของ Deep Neural Network จะทำการเพิ่มชั้น Hidden Layer ให้เพิ่มมากขึ้น ดูจากตัวอย่างข้างต้น (รูปด้านขวา) มีจำนวน Hidden Layer ทั้งสิ้น 4 ชั้น ซึ่งจะทำให้เพิ่มการคำนวณให้มากขึ้น ทำให้สามารถเรียนรู้ข้อมูลได้ดีขึ้น

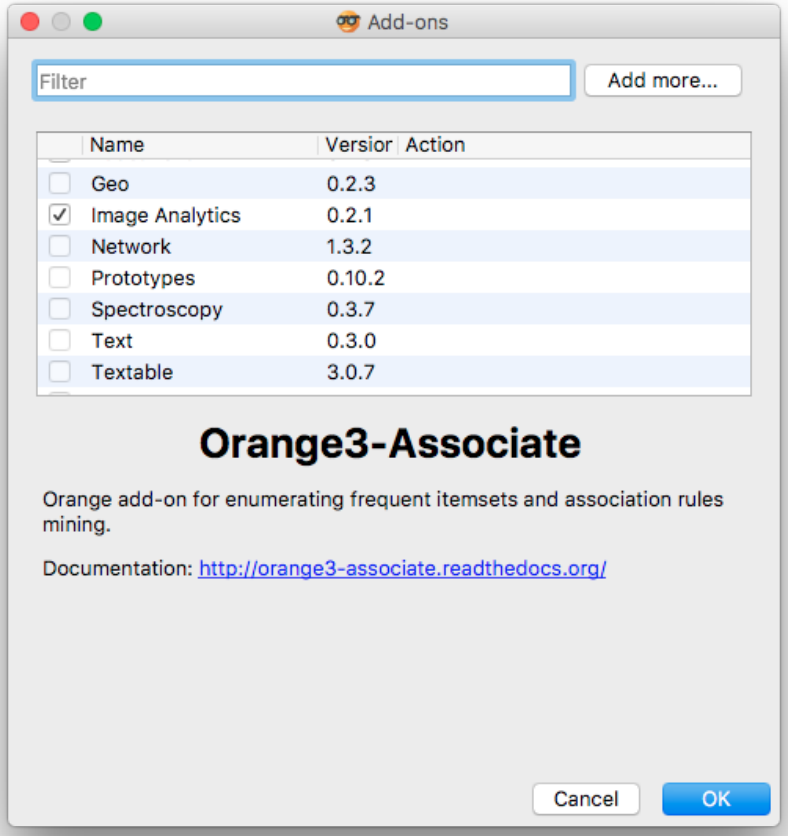
- การกำหนด Deep Neural Network ในโปรแกรม orange สามารถทำได้โดยดับเบิ้ลคลิกที่ไอคอน **Neural Network** จากนั้นเปลี่ยน parameter ในส่วนของ **Neurons per hidden layer** ดังตัวอย่างต่อไปนี้



- จากตัวอย่างข้างต้นได้กำหนดให้ **Neurons per hidden layer** มีค่าเป็น 100,50,30,20 ซึ่งหมายความว่า กำหนดให้มี Hidden Layer จำนวน 4 ชั้น F ดยแต่ละชั้นประกอบด้วย
  - Hidden Layer ชั้นที่ 1 มีจำนวน 100 Neurons
  - Hidden Layer ชั้นที่ 2 มีจำนวน 50 Neurons
  - Hidden Layer ชั้นที่ 3 มีจำนวน 30 Neurons
  - Hidden Layer ชั้นที่ 4 มีจำนวน 20 Neurons

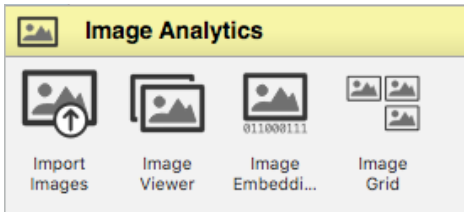
การวิเคราะห์รูปภาพ (Image Analytics)

- การใช้งานเครื่องมือของ Image Analytics จะต้องลงโปรแกรมเสริม (Add-ons…)
- ให้เลือกที่เมนู **Options** > **Add-ons** จากนั้นจะปรากฏหน้าต่าง ดังต่อไปนี้

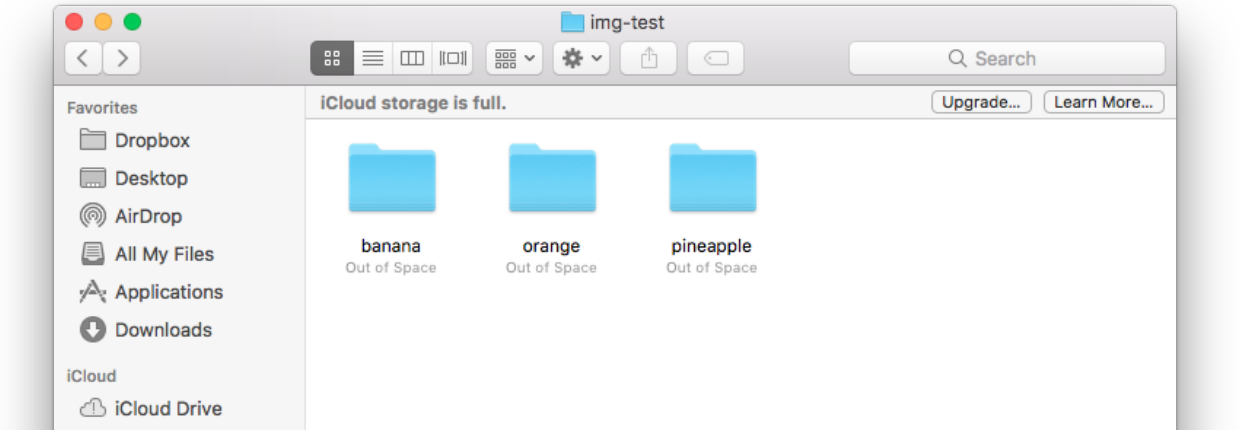


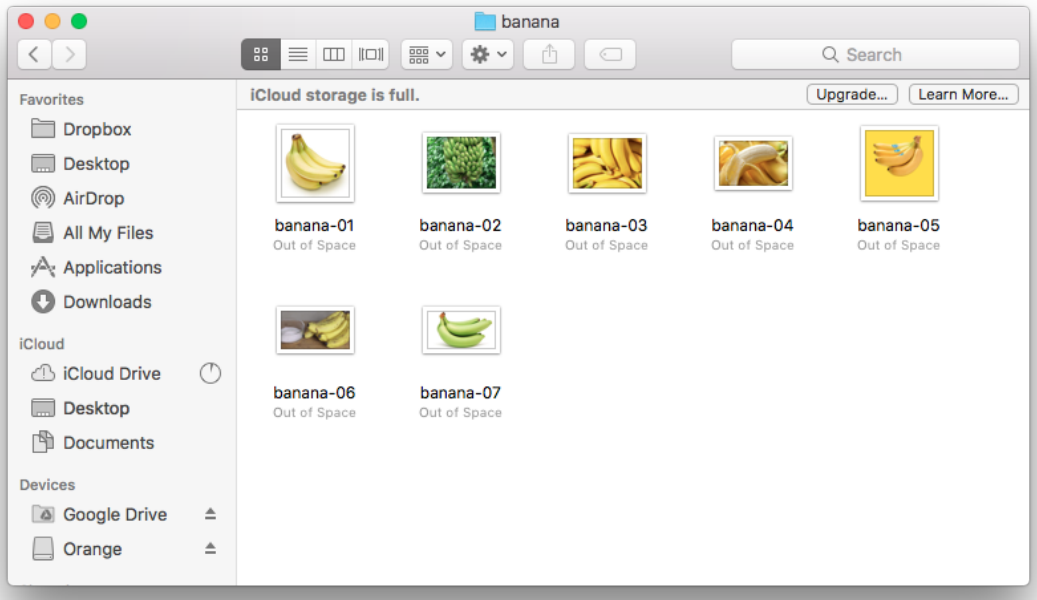
- จากนั้นคลิกเลือกที่ **Image Analytics** และคลิกที่ปุ่ม **OK**
- เมื่อติดตั้งเสร็จเรียบร้อยให้ ปิดและเปิดโปรแกรม orange อีกครั้ง โปรแกรม Image Analytics ถึงจะใช้งานได้

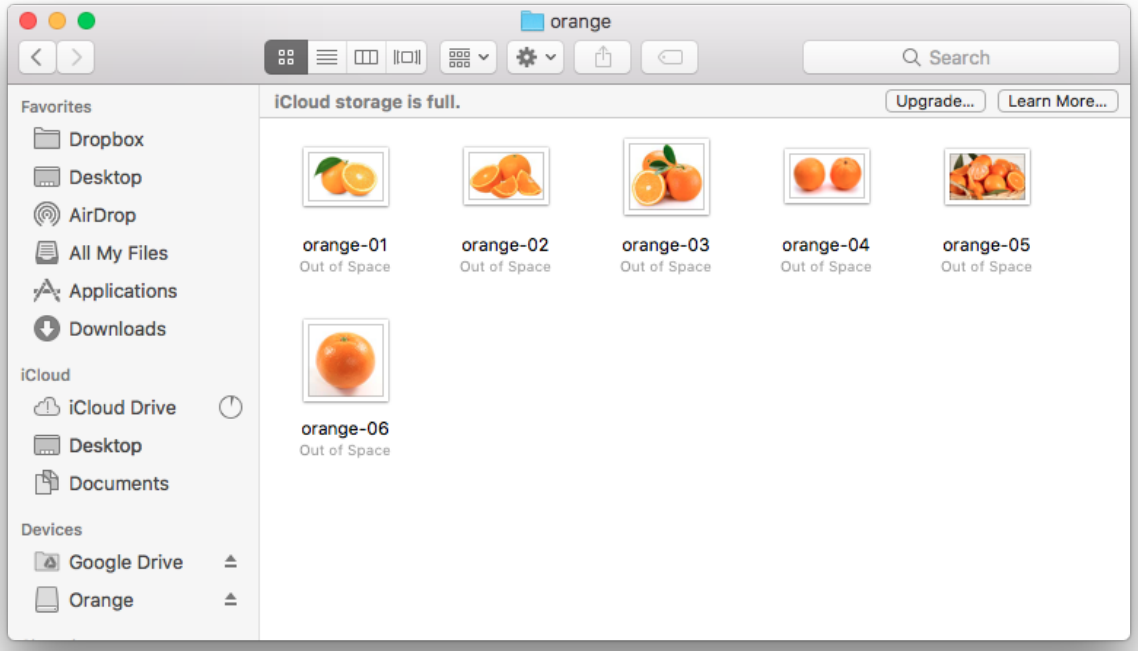
- เมื่อเปิดโปรแกรม orange โปรแกรมจะมีแท็ปของ Image Analytics เพิ่มเข้ามา ดังตัวอย่างต่อไปนี้

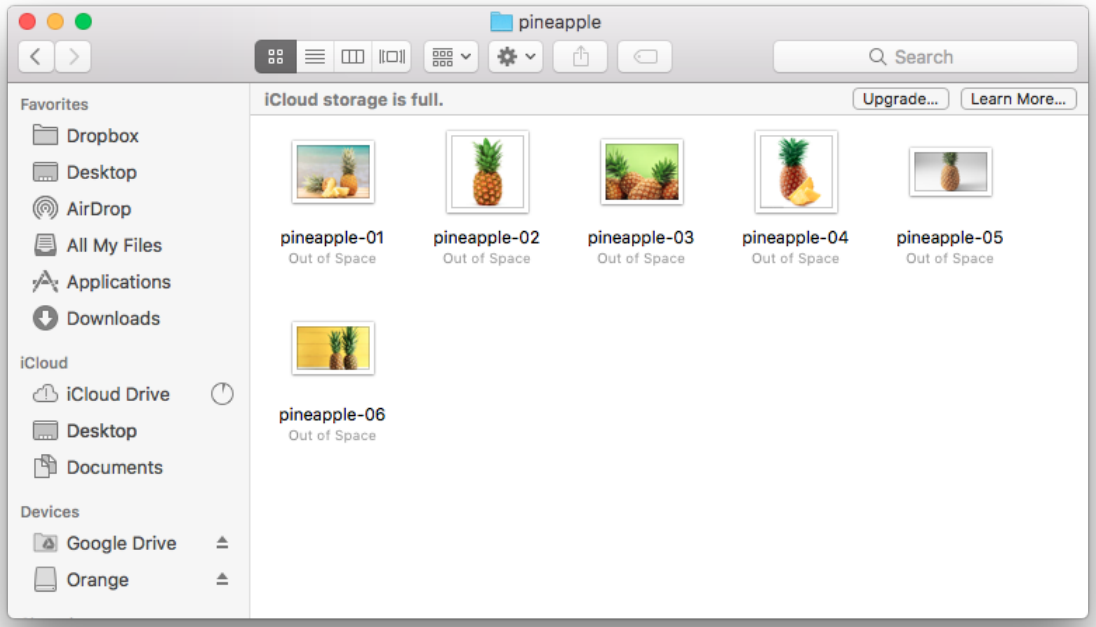


- ก่อนที่จะใช้งาน Image Analytics จะต้องเตรียมข้อมูลรูปภาพให้พร้อม ในกรณีนี้ได้เตรียมรูปภาพ โดยมีรูปภาพอยู่ 3 หมวด ประกอบด้วย banana, orange และ pineapple ซึ่งโปรแกรมจะใช้ชื่อของโฟลเดอร์เป็นชื่อของหมวดหมู่

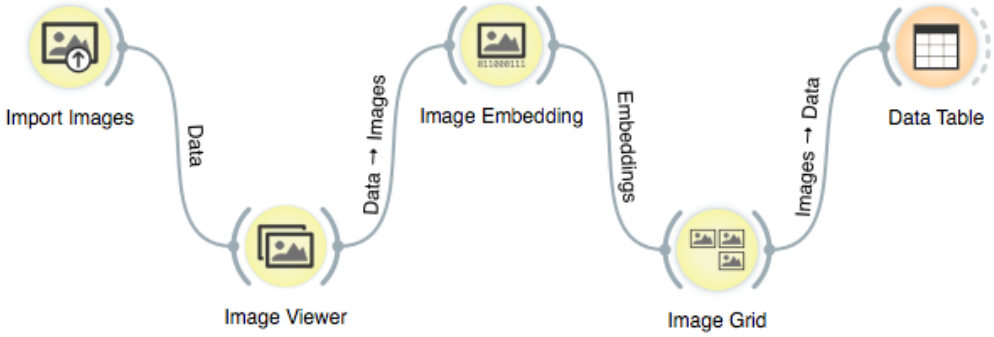




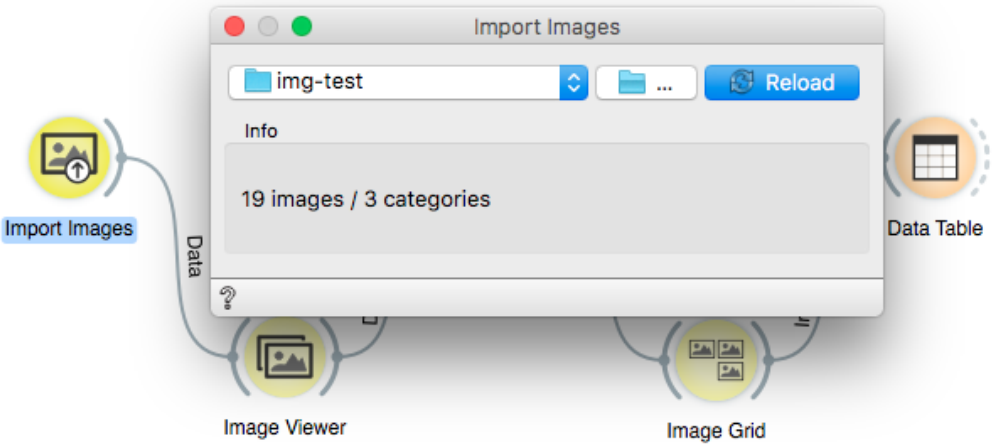

orange
Search
Favorites
Dropbox
Desktop
AirDrop
All My Files
Applications
Downloads
iCloud
iCloud Drive
Desktop
Documents
Devices
Google Drive
Orange
iCloud storage is full.
Upgrade...
Learn More...
orange-01
Out of Space
orange-02
Out of Space
orange-03
Out of Space
orange-04
Out of Space
orange-05
Out of Space
orange-06
Out of Space


pineapple
Search
Favorites
Dropbox
Desktop
AirDrop
All My Files
Applications
Downloads
iCloud
iCloud Drive
Desktop
Documents
Devices
Google Drive
Orange
iCloud storage is full.
Upgrade...
Learn More...
pineapple-01
Out of Space
pineapple-02
Out of Space
pineapple-03
Out of Space
pineapple-04
Out of Space
pineapple-05
Out of Space
pineapple-06
Out of Space

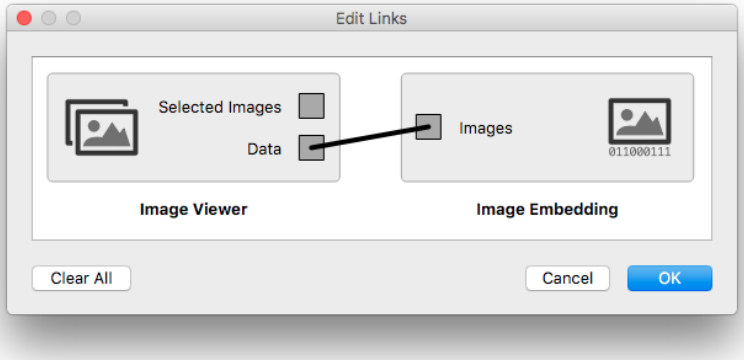
- จากนั้นสร้าง workflow ให้เหมือนดังตัวอย่างต่อไปนี้



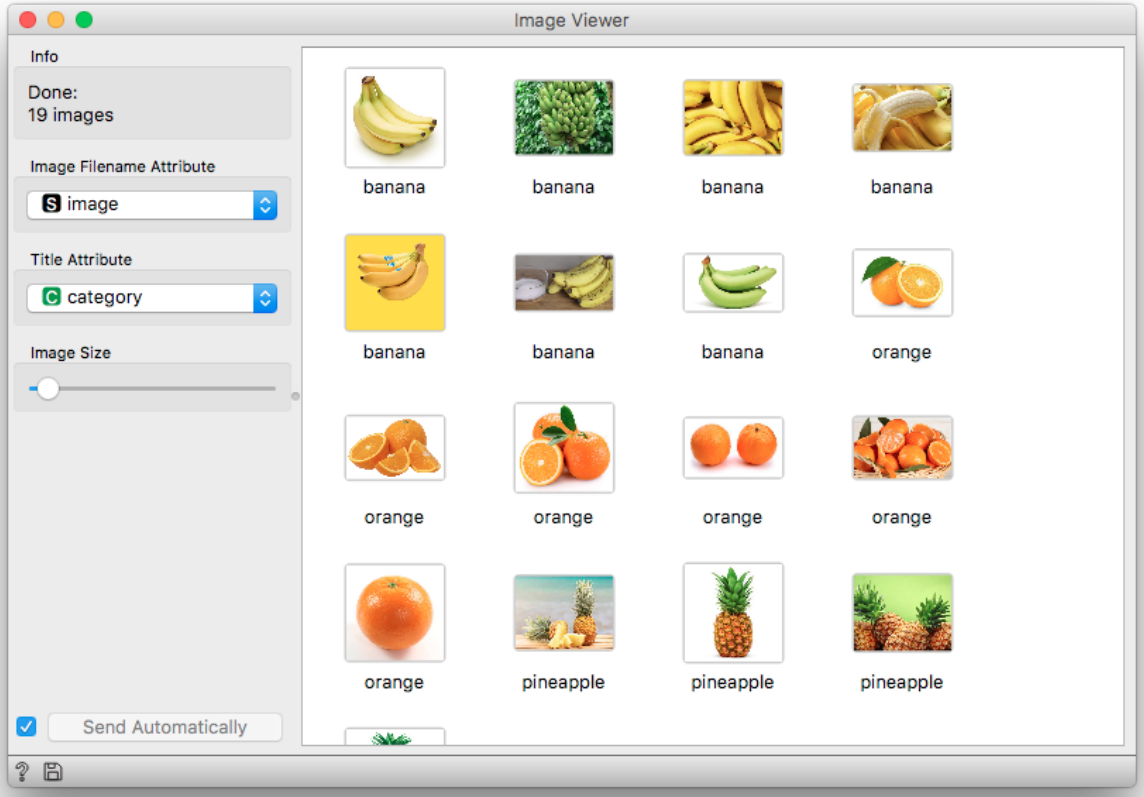
- เมื่อสร้าง workflow เสร็จเรียบร้อยให้ดับเบิ้ลคลิกที่ **Import Images** และเลือกโฟลเดอร์รูปภาพที่ได้เตรียมไว้



- จากนั้นให้ดับเบิ้ลคลิกที่เส้นเชื่อม (Link) ระหว่าง **Image Viewer** และ **Image Embedding**
  - เมื่อปรากฏหน้าต่างให้เลือกการเชื่อมต่อระหว่าง **Data** และ **Images**



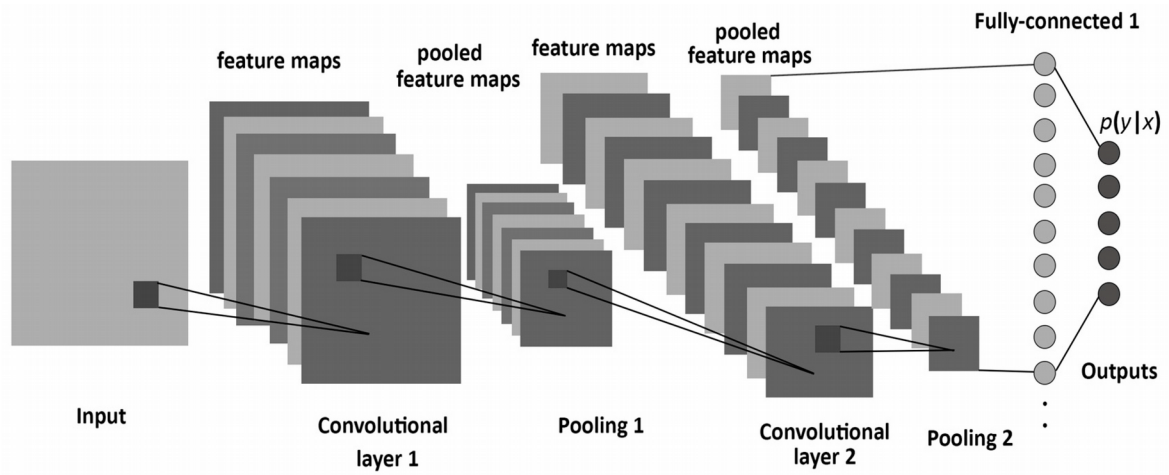
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Image Viewer** เพื่อเปิดดูรูปภาพที่ใช้ในการวิเคราะห์



- ขั้นตอนต่อไปให้ทำการดับเบิ้ลคลิกที่ไอคอน **Image Embedding** เพื่อเลือกโมเดลที่จะใช้ในการวิเคราะห์รูปภาพ

Deep Convolutional Neural Network (Deep CNN)

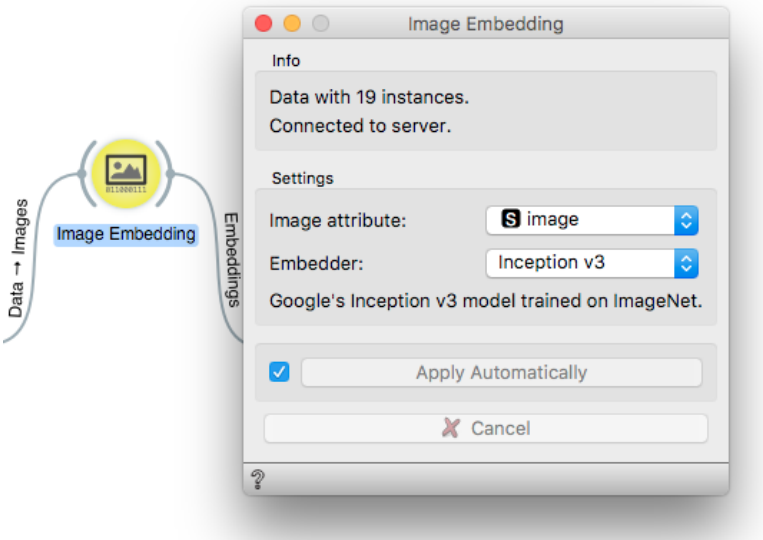
- ในขั้นตอนของ Image Embedding นั้นจะใช้โมเดลของ Deep CNN โดยมีโมเดลให้เลือกดังนี้
  - Inception v3
  - VGG-16
  - VGG-19
- โมเดลทั้ง 3 นั้นได้มาจากการเรียนรู้จากข้อมูล ImageNet ซึ่งใช้รูปภาพในการเรียนรู้เพื่อสร้างโมเดลมากกว่าล้านรูปภาพ



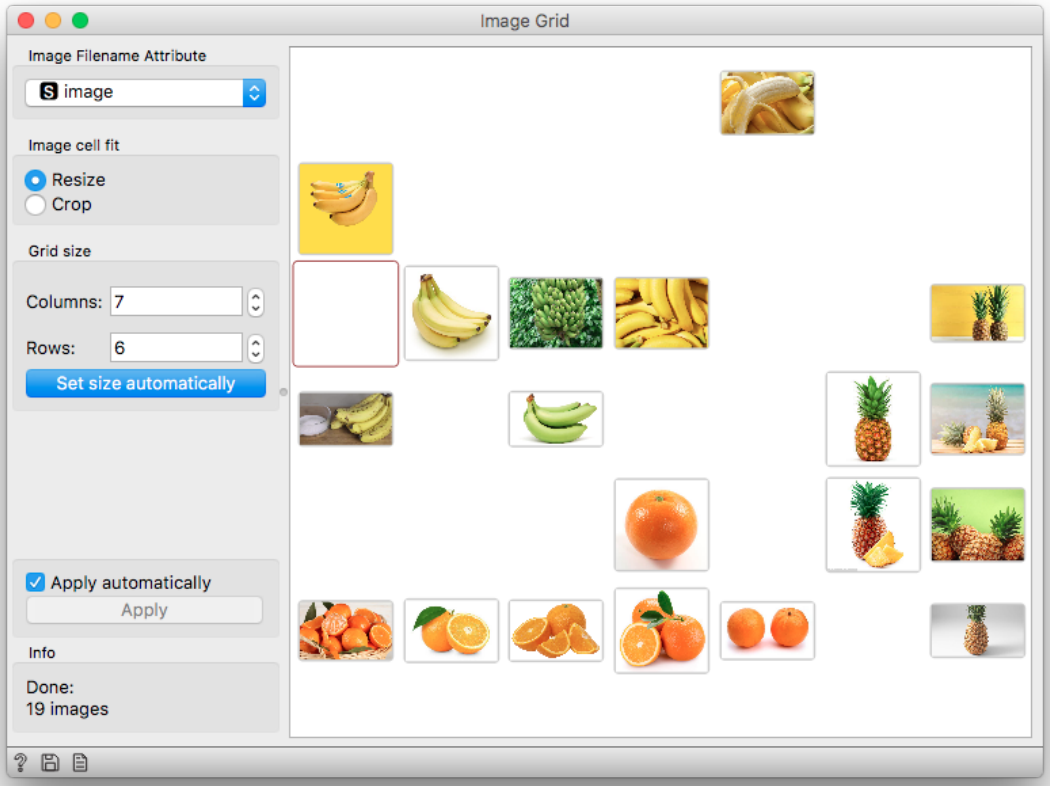
[ที่มา: http://www.mdpi.com/1099-4300/19/6/242]

Inception v3

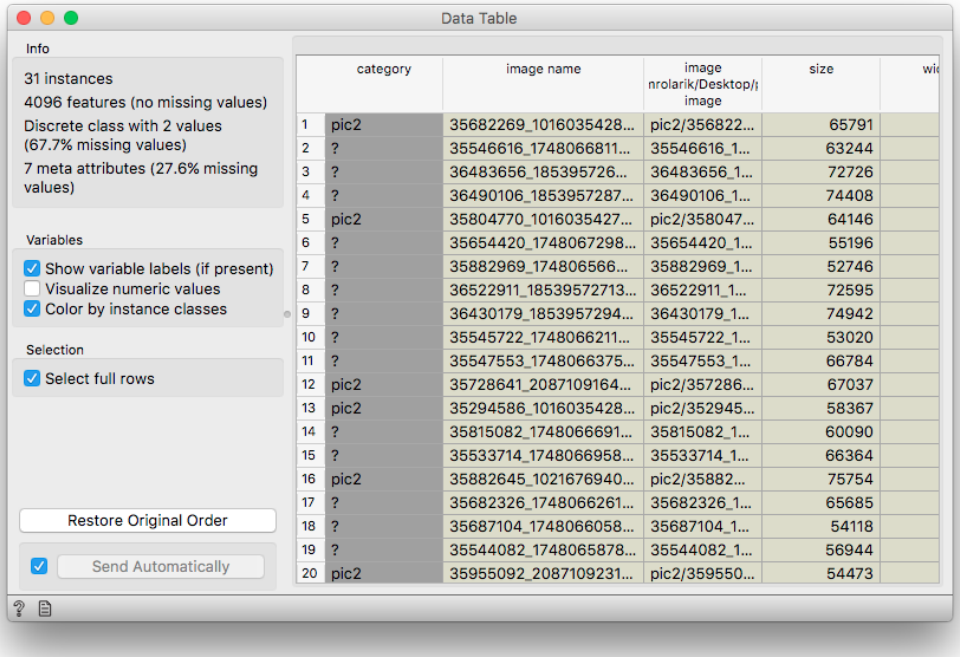
- ในกรณีนี้ เลือกใช้โมเดล **Inception v3** สำหรับการวิเคราะห์รูปภาพ
- ขั้นตอนในการ Analysis รูปภาพอาจใช้เวลาในการประมวลผลนาน



- หลังจากประมวลผลเสร็จเรียบร้อย ให้ดับเบิ้ลคลิกที่ไอคอน **Image Grid** เพื่อดูการจัดกลุ่มรูปภาพ

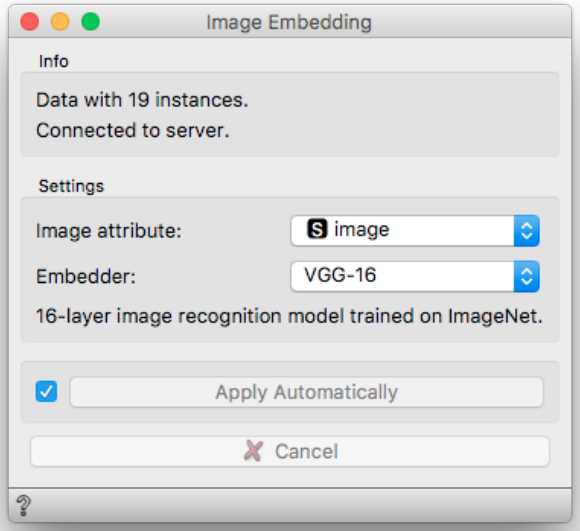


- หากต้องการดูข้อมูลที่ใช้ในการประมวลผลสามารถดับเบิ้ลคลิกได้ที่ไอคอน **Data Table**

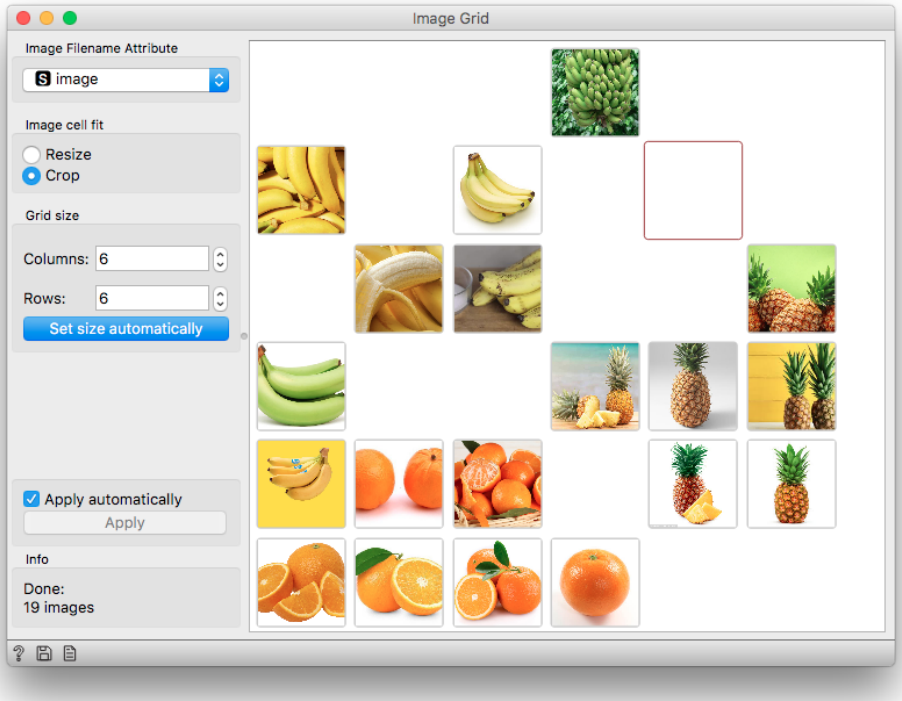


VGG-16

- ในกรณีที่ต้องการเปลี่ยนโมเดลเป็น VGG-16 สามารถทำได้โดยดับเบิ้ลคลิกที่ไอคอน **Image Embedding** และเปลี่ยน parameter ของ Embedder เป็น **VGG-16** จากนั้นรอให้โปรแกรมประมวลผล

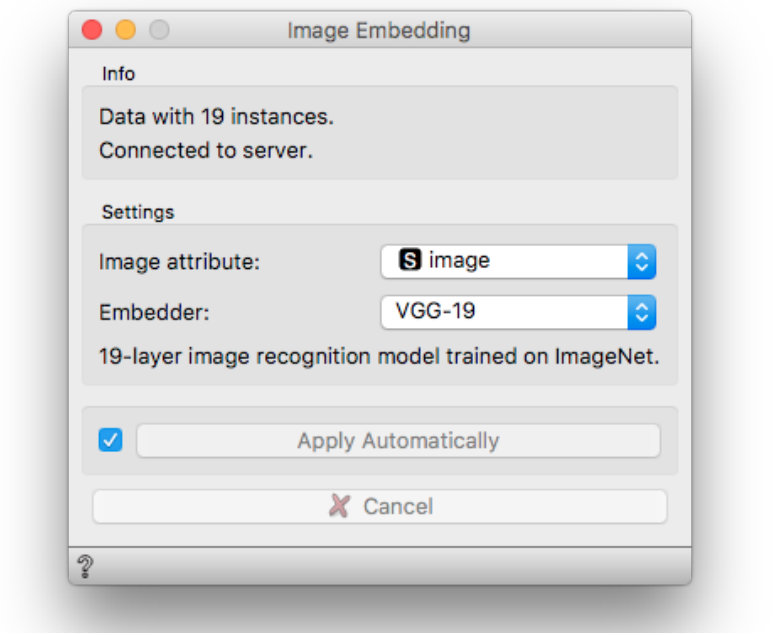


- เมื่อรอจนประมวลผลเสร็จให้ดับเบิ้ลคลิกที่ไอคอน **Image Grid** เพื่อดูผลลัพธ์



VGG-19

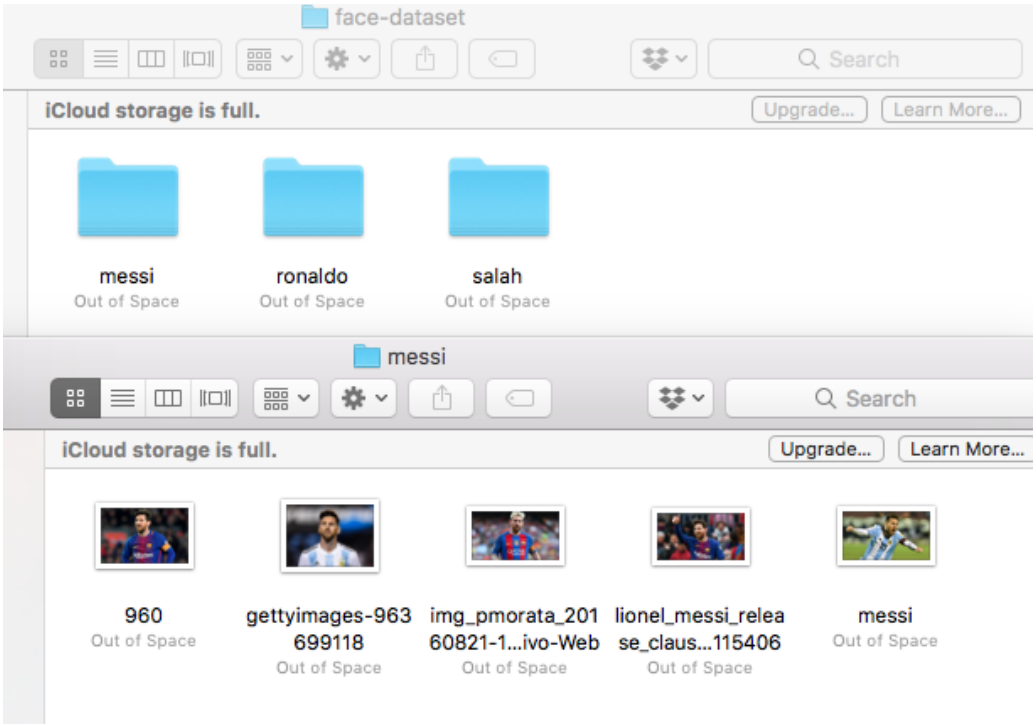
- ในกรณีที่ต้องการเปลี่ยนโมเดลเป็น VGG-19 สามารถทำได้โดยดับเบิ้ลคลิกที่ไอคอน **Image Embedding** และเปลี่ยน parameter ของ Embedder เป็น **VGG-19**



- เมื่อโปรแกรมประมวลผลเสร็จเรียบร้อยให้ดับเบิ้ลคลิกที่ไอคอน **Image Grid** เพื่อดูผลลัพธ์

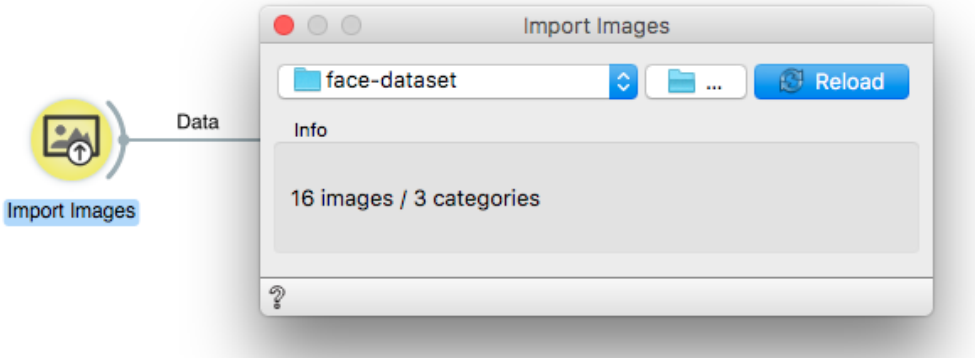
การจัดหมวดหมู่รูปภาพใบหน้า (Face Image Classification)

- โปรแกรม Orange สามารถจัดหมวดหมู่รูปภาพใบหน้า โดยโปรแกรมได้เลือกใช้โมเดล OpenFace ในการจัดหมวดหมู่รูปภาพใบหน้า ซึ่งโมเดลนี้ใช้ชุดข้อมูล FaceScrub และ CASIA-WEBFace ในการเรียนรู้เพื่อสร้างโมเดล
- ในกรณีนี้ ได้เลือกทดสอบโมเดลโดยใช้ข้อมูลนักฟุตบอลจำนวน 3 คน ได้แก่ Messi, Ronaldo และ Salah โดยแต่ละคนมีรูปภาพประมาณ 5-6 รูป
- รูปภาพนักฟุตบอลจะถูกจัดเก็บแยกออกเป็นโฟลเดอร์ ดังตัวอย่างต่อไปนี้

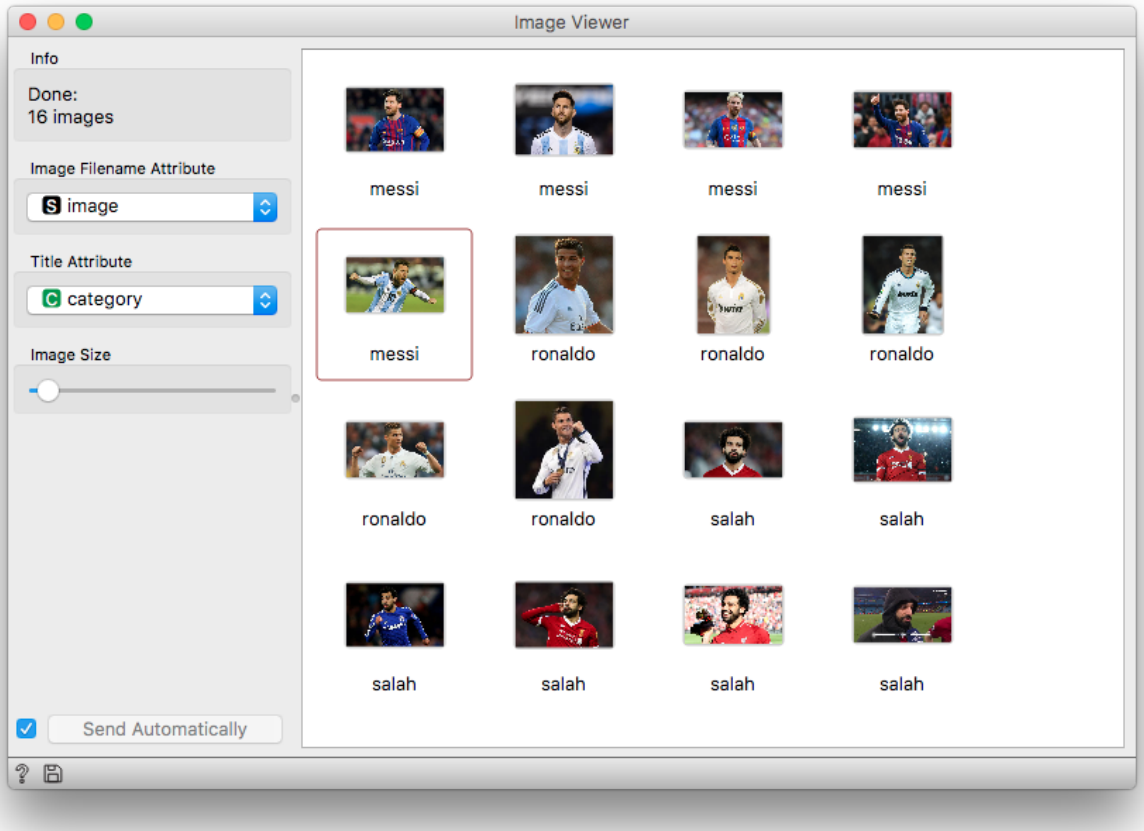


- เมื่อเตรียมข้อมูลรูปภาพบุคคลเสร็จเรียบร้อยให้สร้าง workflow ดังตัวอย่างต่อไปนี้

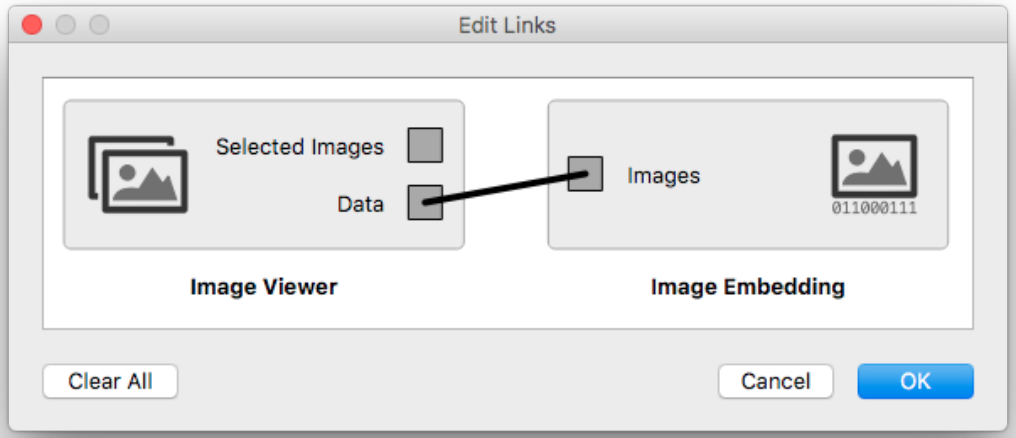
- จากนั้นดับเบิ้ลคลิกที่ไอคอน **Import Images** และเลือกโฟลเดอร์รูปภาพในหน้าที่ต้องการ



- เมื่อเลือกรูปภาพเสร็จเรียบร้อย จากนั้นดับเบิ้ลคลิกที่ไอคอน **Image Viewer** เพื่อดูรูปภาพที่ใช้ในการจัดหมวดหมู่

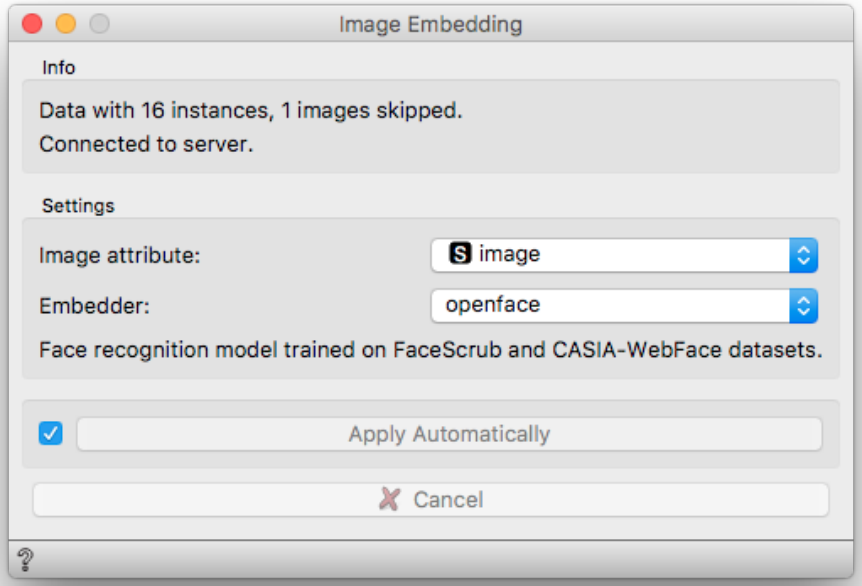


- จากนั้นดับเบิ้ลคลิกที่ Link ระหว่าง **Image Viewer** และ **Image Embedding** และเปลี่ยนการเชื่อมต่อให้เป็น **Data** และ **Images**

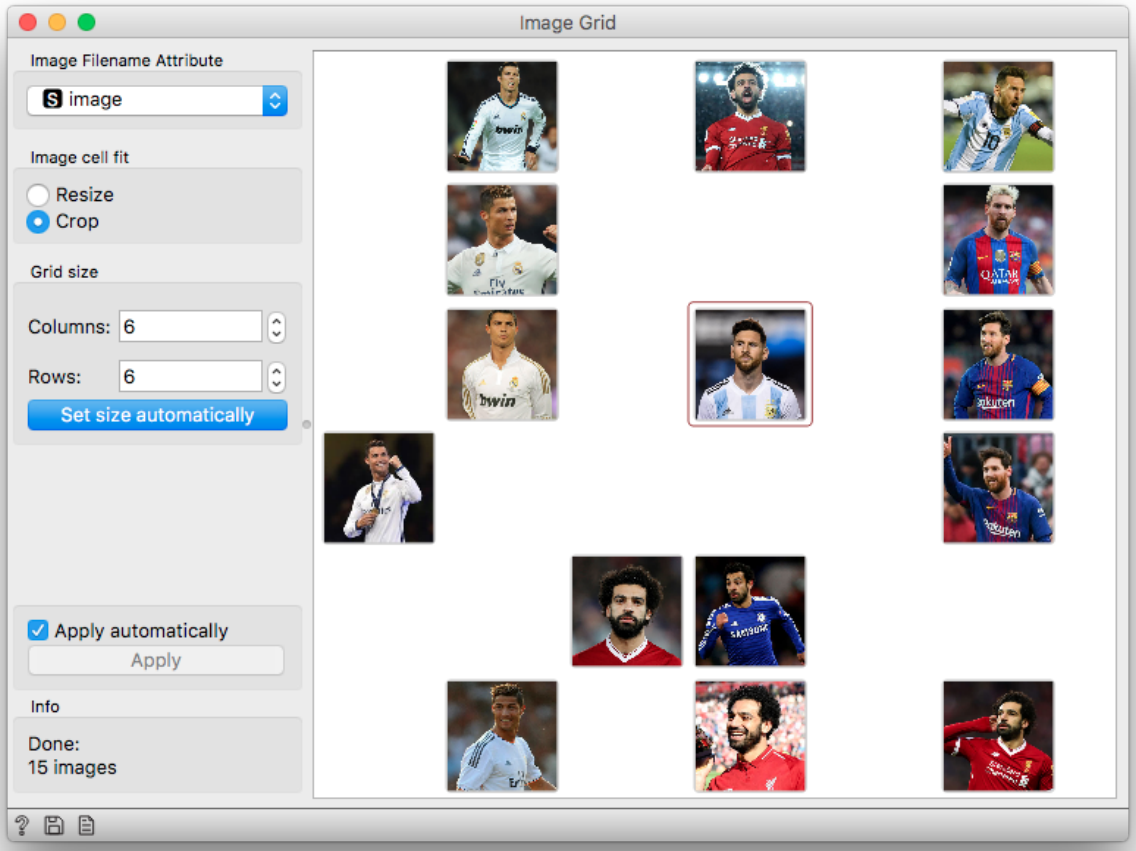


OpenFace

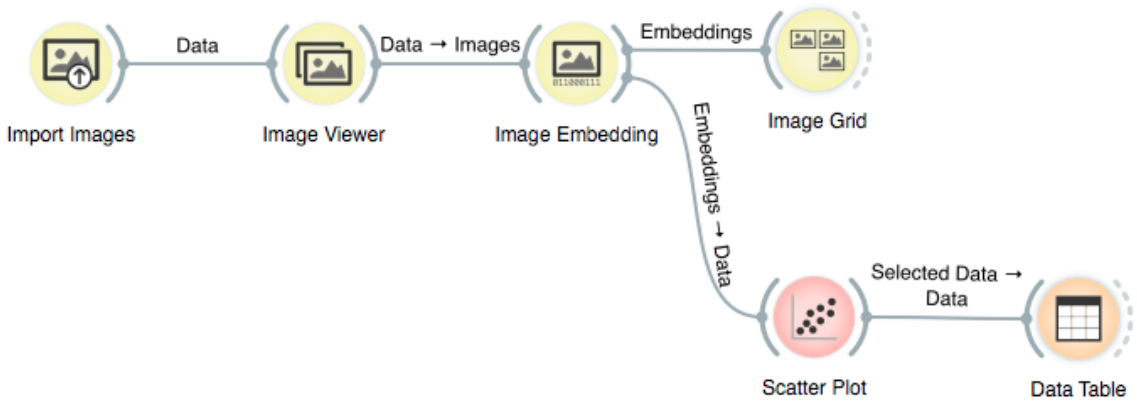
- สำหรับการจัดหมวดหมู่รูปภาพใบหน้าจะต้องใช้โมเดล **OpenFace**
- ดับเบิ้ลคลิกที่ไอคอน **Image Embedding** และเปลี่ยน **Embedder** ให้เป็น **openface**



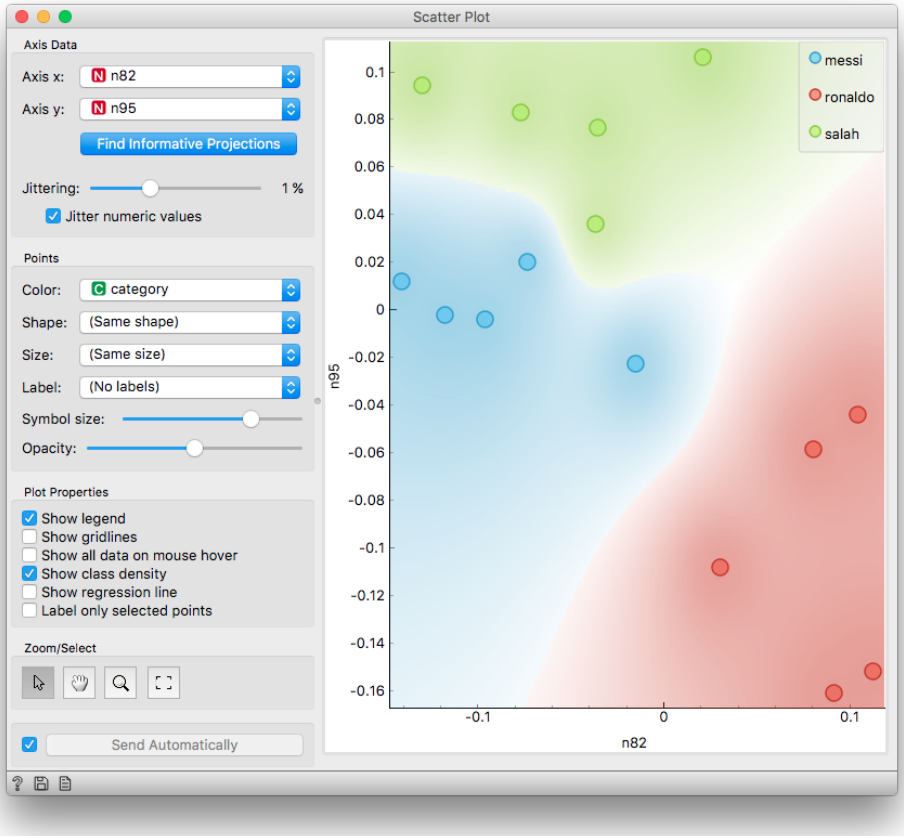
- ขั้นตอนสุดท้าย ให้ดับเบิ้ลคลิกที่ **Image grid** เพื่อดูผลลัพธ์ของการจัดหมวดหมู่รูปภาพใบหน้าบุคคล



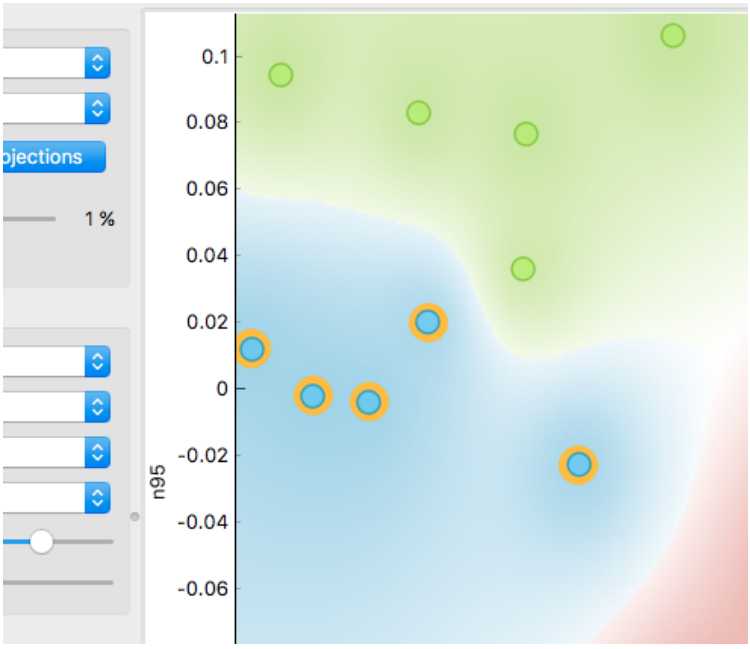
- หากต้องการตรวจสอบความถูกต้องของการจัดหมวดหมู่รูปภาพใบหน้า สามารถเพิ่มไอคอนลงไปใน workflow ดังตัวอย่างต่อไปนี้



- จากตัวอย่างได้เพิ่มไอคอน **Scatter Plot** และ **Data Table** เพื่อตรวจสอบการจัดหมวดหมู่ และดูข้อมูลว่าถูกต้องหรือไม่
- จากนั้นให้ดับเบิ้ลคลิกที่ไอคอน **Scatter Plot** เพื่อดูการจัดหมวดหมู่ของรูปภาพ



- หากต้องการตรวจสอบให้แน่ชัดว่า จุดวงกลมสีฟ้า คือรูปภาพของ Messi ใช่หรือไม่ สามารถทำได้โดยใช้เมาส์คลิกที่จุดวงกลมสีฟ้า



- สุดท้ายให้ดับเบิ้ลคลิกที่ไอคอน **Data Table** เพื่อตรวจสอบข้อมูล