Topic 2
Data Preprocessing + Low-code AI tool

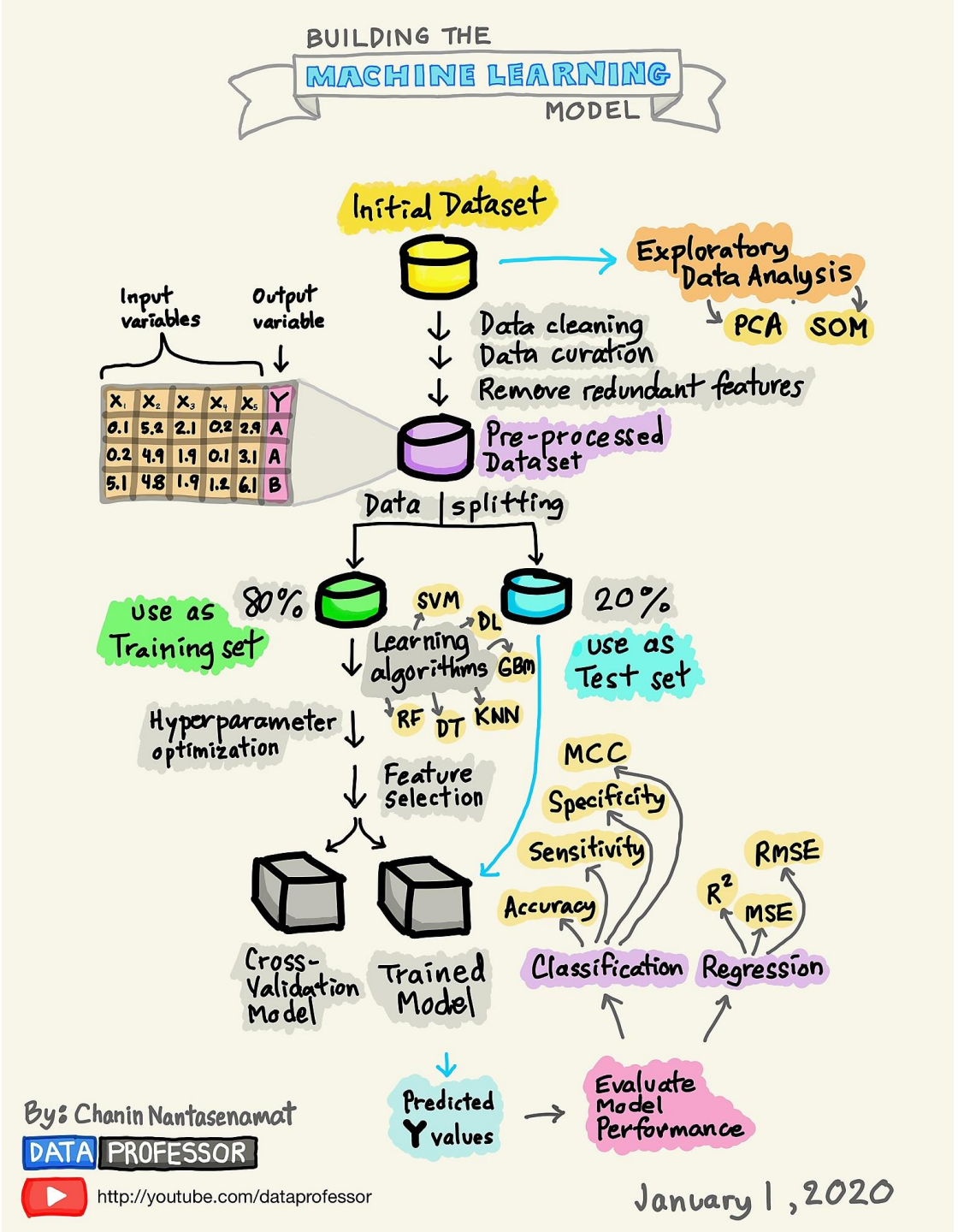
IT8302 APPLIED MACHINE LEARNING

Learning Outcomes

- Understand principles of data representation and feature engineering
  - Understanding representation of categorical variables
  - Explain binning, discretization
  - Understand feature selection
  - Understand how to utilize expert knowledge
- Use of Machine Learning Tools
  - Use Orange Data Mining to carry out simple machine learning training and predictions.

Machine Learning Process

Source: https://figshare.com/articles/figure/Building_The_Machine_Learning_Model/11492316



Machine Learning Process

- **The primary goal of the process is to identify a 'Model'**. The Model is the main thing that applications can submit requests to in order to gain insight on new data. A person working as the role of a Data Scientist performs the Machine Learning process and will ultimately decide on the right model to use.
- **The process starts with a question**; what are you trying to learn from your Machine Learning experiment? For example, in the case of recommendations, the question might be *"identify most commonly sold products for each product in the inventory"*

Machine Learning Process

- **The next step is to provide 'prepared data'**. Prepared data is one or more data sets that have been pre-processed (formatted, cleaned and sampled) in readiness to apply Machine Learning algorithms to. Preparing the data means that the data is in the best shape to draw scientific conclusions from and is not skewed in any way.
- Once you have your prepared data, you apply one or more Machine Learning algorithms to it with a view to producing a **Model**. This is an iterative process and you may loop around testing various algorithms until you have a Model that sufficiently answers your question.
- Once you have produced your chosen model, it will typically be exposed via some kind of API.

Choosing Learning Algorithms

Choosing a ML algorithm

- Every machine learning algorithm has its own style or ***inductive bias***. For a specific problem, several algorithms may be appropriate and one algorithm may be a better fit than others. But it's not always possible to know beforehand which is the best fit. In cases like these, several algorithms are listed together in the cheat sheet. An appropriate strategy would be to try one algorithm, and if the results are not yet satisfactory, try the others.

Choosing a ML algorithm

There are three main categories of machine learning: **supervised learning**, **unsupervised learning**, and **reinforcement learning**.

- In **supervised learning**, each data point is **labelled** or associated with a **category** or **value** of interest. An example of a categorical label is assigning an image as either a ‘cat’ or a ‘dog’. An example of a value label is the sale price associated with a used car. The goal of supervised learning is to study many labelled examples like these, and then to be able to make predictions about future data points. For example, identifying new photos with the correct animal or assigning accurate sale prices to other used cars. This is a popular and useful type of machine learning.

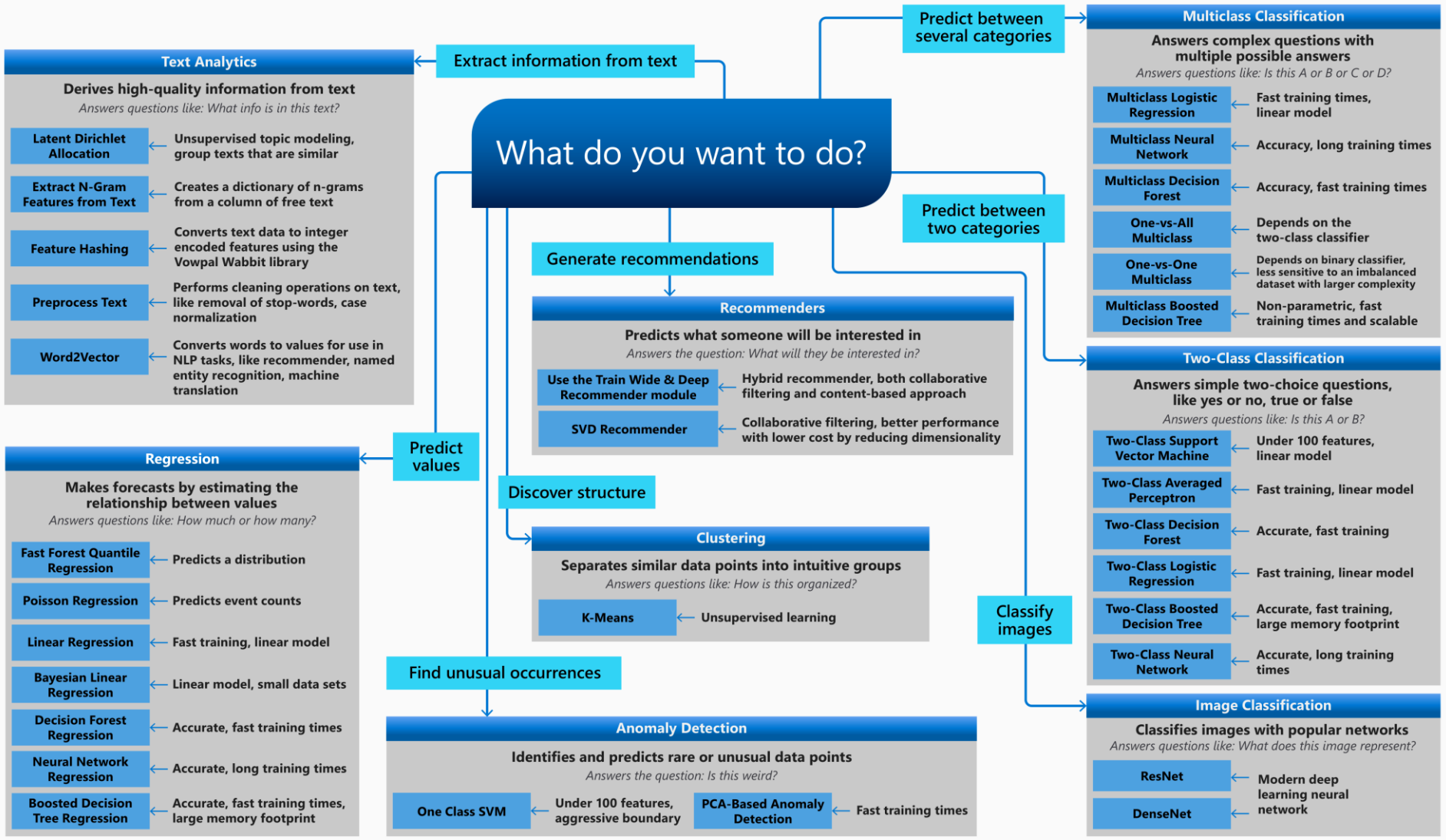
Choosing a ML algorithm

- In **unsupervised learning**, data points have no labels associated with them. Instead, the goal of an unsupervised learning algorithm is to organize the data in some way or to describe its structure. This can mean grouping it into clusters, as K-means does, or finding different ways of looking at complex data so that it appears simpler.
- In **reinforcement learning**, the algorithm gets to choose an action in response to each data point. It is a common approach in robotics, where the set of sensor readings at one point in time is a data point, and the algorithm must choose the robot's next action. It's also a natural fit for Internet of Things applications. The learning algorithm also receives a reward signal a short time later, indicating how good the decision was. Based on this, the algorithm modifies its strategy in order to achieve the highest reward.

Machine Learning Algorithm Cheat Sheet

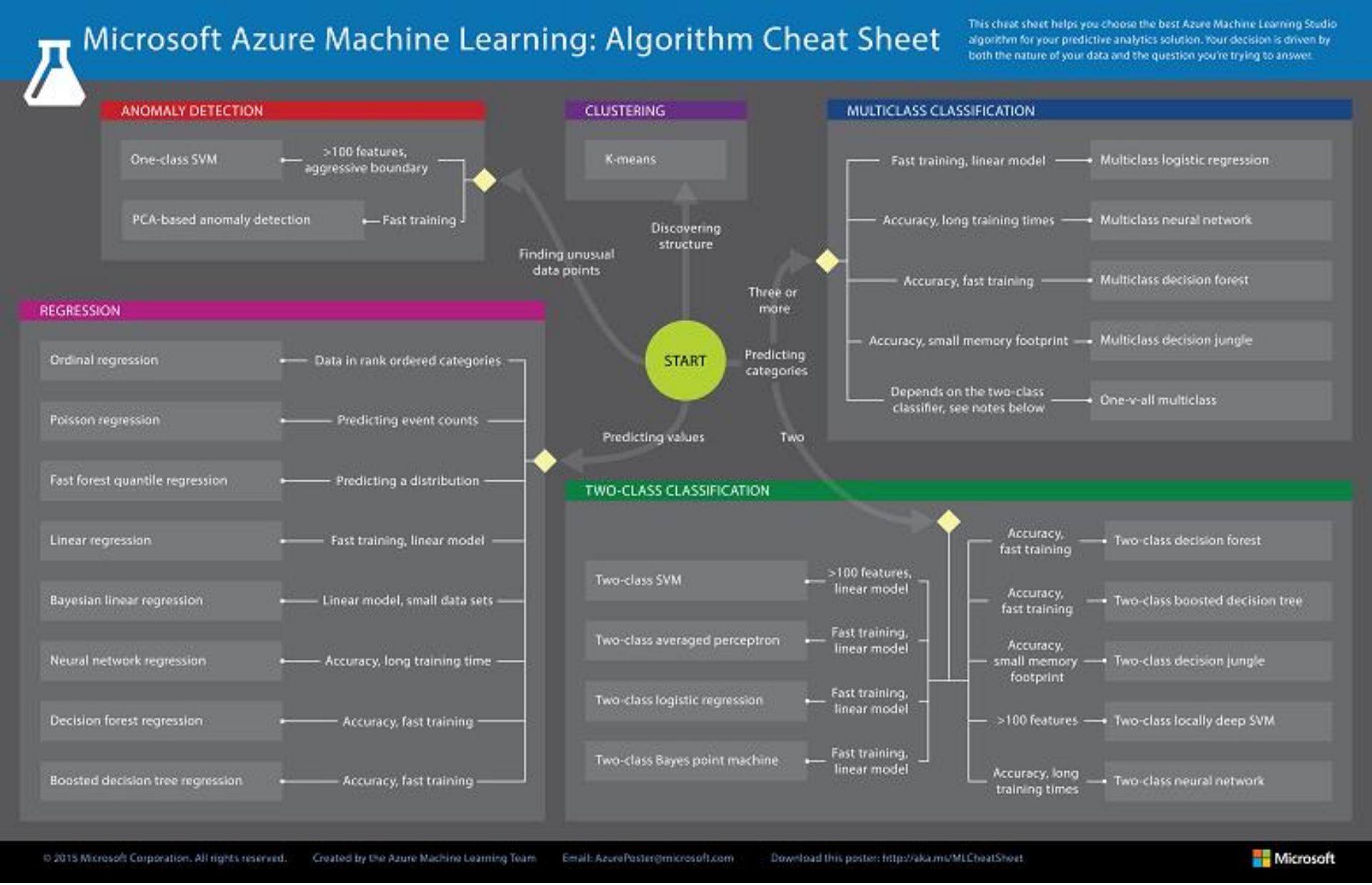
This cheat sheet helps you choose the best machine learning algorithm for your predictive analytics solution.
Your decision is driven by both the nature of your data and the goal you want to achieve with your data.

Microsoft Azure



 Share this poster: aka.ms/mlcheatsheet

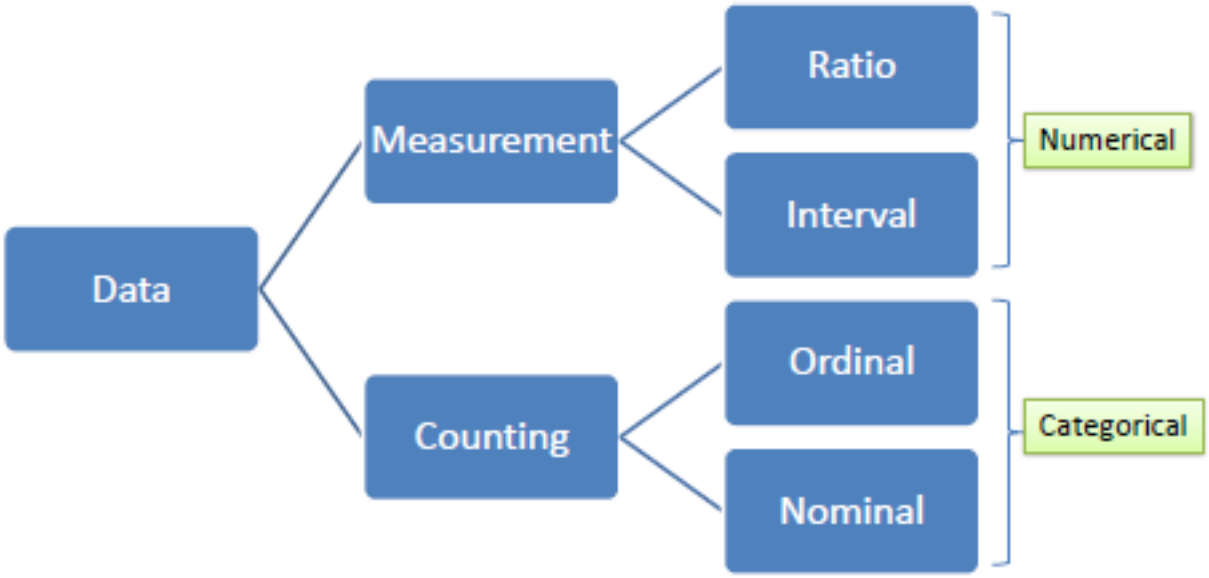
https://download.microsoft.com/download/3/5/b/35bb997f-a8c7-485d-8c56-19444dafd757/azure-machine-learning-algorithm-cheat-sheet-july-2021.pdf



This is for Azure ML Studio (Classic) and will be retired
download http://download.microsoft.com/download/A/6/1/A613E11E-8F9C-424A-B99D-65344785C288/microsoft-machine-learning-algorithm-cheat-sheet-v6.pdf

Understand Principles of Data Representation

Data Types



source http://www.saedsayad.com/data_preparation.htm

Data Types

Data is information typically the results of measurement (numerical) or counting (categorical). Variables serve as placeholders for data. There are two types of variables, numerical and categorical.

- A **numerical** or **continuous variable** is one that can accept any value within a finite or infinite interval (e.g., height, weight, temperature, blood glucose, ...). There are two types of numerical data, ***interval*** and ***ratio***. Data on an interval scale can be added and subtracted but cannot be meaningfully multiplied or divided because there is no true zero. For example, we cannot say that one day is twice as hot as another day. On the other hand, data on a ratio scale has true zero and can be added, subtracted, multiplied or divided (e.g., weight).
- A **categorical** or **discrete variable** is one that can accept two or more values (categories). There are two types of categorical data, ***nominal*** and ***ordinal***. Nominal data does not have an intrinsic ordering in the categories. For example, "gender" with two categories, male and female. In contrast, ordinal data does have an intrinsic ordering in the categories. For example, "level of energy" with three orderly categories (low, medium and high).

ML algorithms require the inputs variables to be numerical

- No problem for data that is already a real number or integer
- Categorical variables need to be converted or encoded into a number

Encoding categorical variables

- Use of one-hot encoding for categorical variables

Original values (Direction)	I0	I1	I2	I3
NORTH	1	0	0	0
EAST	0	1	0	0
SOUTH	0	0	1	0
WEST	0	0	0	1

One-hot encoding of variables = number of values
In the case 4 possible values = 4 (I0,I1,I2,I3 variables)
The original column is replaced by 4 new columns

Encoding categorical variables

- Use of dummy variables encoding for categorical variables

Original values (Direction)	I0	I1	I2
NORTH	1	0	0
EAST	0	1	0
SOUTH	0	0	1
WEST	0	0	0

Number of dummy/indicator variables = number of values – 1
In the case 4 possible values – 1 = 3 (I0,I1,I2 dummy variables)
The original column is replaced by 3 new columns

Why not just use a value encoding (e.g. LabelEncoder)?

The problem with value encoding or LabelEncoder is that is leads to undesirable side effects if the ML algorithm uses the numerical values. For example, Average of East and West = (1+3)/2 = 2 (South).

This is nonsense as the average of East and West has no meaning.

Direction	Value
NORTH	0
EAST	1
SOUTH	2
WEST	3

Discretization/Binning

Discretization (otherwise known as quantization or binning) provides a way to partition continuous features into discrete values.

Certain datasets with continuous features may benefit from discretization, because discretization can transform the dataset of continuous attributes to one with only nominal attributes.

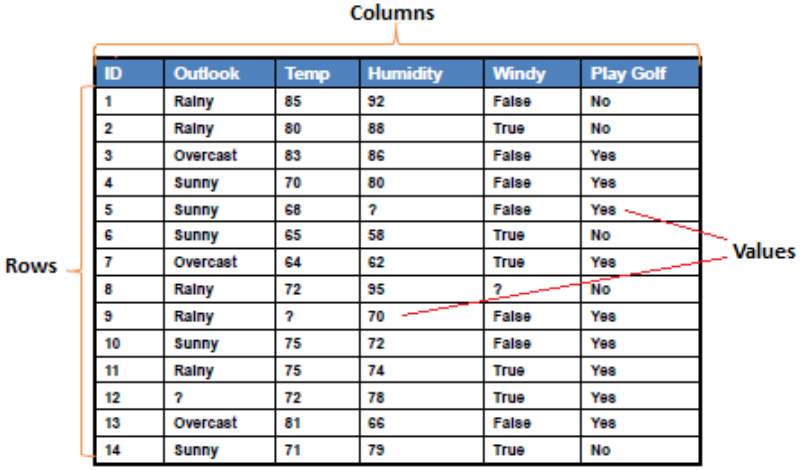
Time	TimeOfDay
00:00	0
07:30	1
08:00	1
12:00	2
13:00	2
14:00	2
17:00	2

We convert the 24H into 4 time period bins
00:00-06:00 is the bin 0 for early morning

Dataset

Dataset is a collection of data, usually presented in a tabular form. Each **column** represents a particular **variable** (**attribute**), and each **row** corresponds to a given **member** (**observation**) of the data.

In predictive modeling, **predictors** or **attributes** are the **input variables** and **target** or **class attribute** is the **output variable** whose value is determined by the values of the predictors and function of the predictive model.



ID	Outlook	Temp	Humidity	Windy	Play Golf
1	Rainy	85	92	False	No
2	Rainy	80	88	True	No
3	Overcast	83	86	False	Yes
4	Sunny	70	80	False	Yes
5	Sunny	68	?	False	Yes
6	Sunny	65	58	True	No
7	Overcast	64	62	True	Yes
8	Rainy	72	95	?	No
9	Rainy	?	70	False	Yes
10	Sunny	75	72	False	Yes
11	Rainy	75	74	True	Yes
12	?	72	78	True	Yes
13	Overcast	81	66	False	Yes
14	Sunny	71	79	True	No

Dataset

There are some alternatives terms for columns, rows and values.

- Columns, Fields, Attributes, Variables
- Rows, Records, Objects, Cases, Instances, Examples, Vectors
- Values, Data

Feature Engineering and Feature Selection

Feature Engineering

A **feature** is an **attribute** or **property** shared by all of the independent units on which analysis or prediction is to be done. Any attribute could be a feature, as long as it is useful to the model.

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work.

- **feature engineering**: This process attempts to create additional relevant features from the existing raw features in the data, and to increase the predictive power of the learning algorithm.
- **feature selection**: This process selects the key subset of original data features in an attempt to reduce the dimensionality of the training problem.

Normally **feature engineering** is applied first to generate additional features, and then the **feature selection** step is performed to eliminate irrelevant, redundant, or highly correlated features.

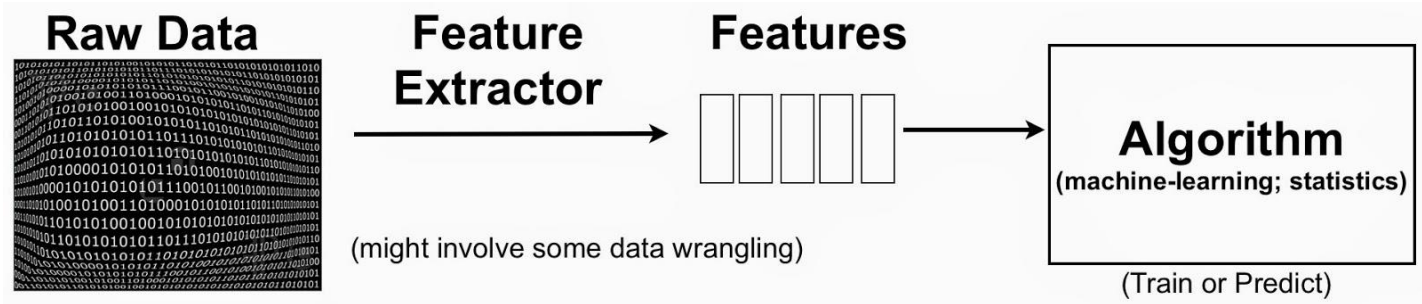
Feature engineering is fundamental to the application of machine learning, and is both difficult and expensive.

source https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/create-features

Feature Engineering

The **training data** used in machine learning can often be enhanced by **extraction of features** from the raw data collected. An example of an engineered feature in the context of learning how to classify the images of handwritten characters is creation of a bit density map constructed from the raw bit distribution data. This map can help locate the edges of the characters more efficiently than simply using the raw distribution directly.

What kind of features should be created to enhance the dataset when training a model? Engineered features that enhance the training provide information that better differentiates the patterns in the data. The new features are expected to provide additional information that is not clearly captured or easily apparent in the original or existing feature set. But **this process is something of an art**. Sound and productive decisions often require some domain expertise.



Feature Engineering

- Engineered and selected features increase the efficiency of the training process, which attempts to extract the key information contained in the data.
- They also improve the power of these models to classify the input data accurately and to predict outcomes of interest more robustly.
- Feature engineering and selection can also combine to make the learning more computationally tractable. It does so by enhancing and then reducing the number of features needed to calibrate or train a model. Mathematically speaking, the features selected to train the model are a minimal set of independent variables that explain the patterns in the data and then predict outcomes successfully.

Feature Selection vs Feature Extraction

In **feature selection** we try to find the best subset of the input feature set

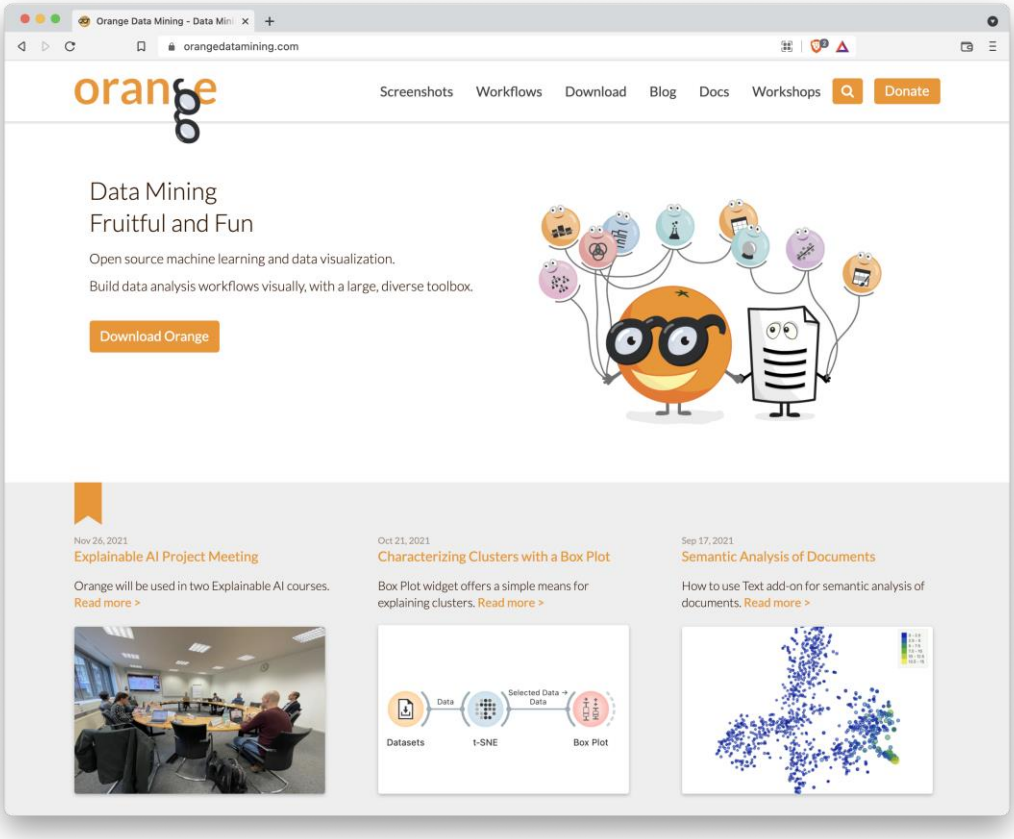
In **feature extraction** we create new features based on the transformation or combination of the original feature set

No Code / Low Code Machine Learning with Orange Data Mining

PRACTICAL EXERCISE

INTRODUCTION TO
ORANGE DATA MINING

https://www.orangedatamining.com


Orange Data Mining - Data Mini
orangedatamining.com
orange
Screenshots
Workflows
Download
Blog
Docs
Workshops
Donate
Data Mining
Fruitful and Fun
Open source machine learning and data visualization.
Build data analysis workflows visually, with a large, diverse toolbox.
Download Orange
Nov 26, 2021
Explainable AI Project Meeting
Orange will be used in two Explainable AI courses.
Read more >
Oct 21, 2021
Characterizing Clusters with a Box Plot
Box Plot widget offers a simple means for explaining clusters. Read more >
Data
Selected Data
Data
Datasets
t-SNE
Box Plot
Sep 17, 2021
Semantic Analysis of Documents
How to use Text add-on for semantic analysis of documents. Read more >

ORANGE DATA MINING FEATURES

- Designed to be an easy to use interface for applied machine learning
- Simple to use drag-and-drop interface
- Pre-built algorithms (no need to code anything)
- Sample data sets
- Based on Python (for more advanced users)
- Supports Windows, macOS and Linux
- For more information about what you can do: https://orangedatamining.com/

DOCUMENTATION

Documentation is available online

- https://orangedatamining.com/docs/

Orange Data Mining: Python Library Reference

- https://orange3.readthedocs.io/projects/orange-data-mining-library/en/latest/

WHAT CAN YOU DO WITH ORANGEDM?

- Supervised Learning (Classification)
  - Answers simple, two-choice questions, like yes-or-no, true-or-false.

    Is this tweet positive?

    Will this customer renew their service?

    Which of two coupons draws more customers

WHAT CAN YOU DO WITH ORANGEDM?

- Supervised Learning (Regression)
  - Forecast the future by estimating the relationship between variables.

    Estimate product demand

    Predict sales figures

    Determine equipment servicing priorities

WHAT CAN YOU DO WITH ORANGEDM?

- Unsupervised Learning (Clustering)
  - Separate similar data points into intuitive groups.

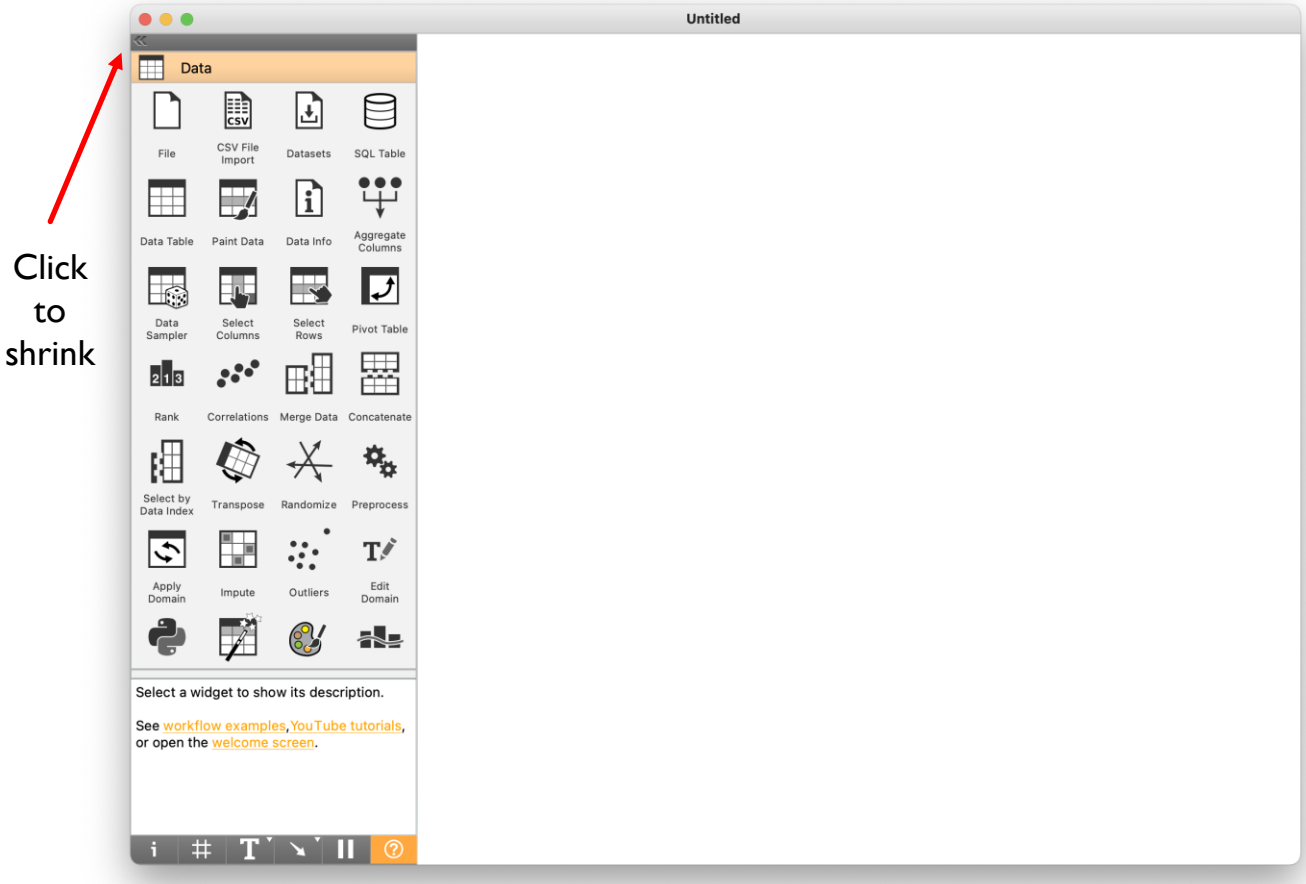
    Perform customer segmentation

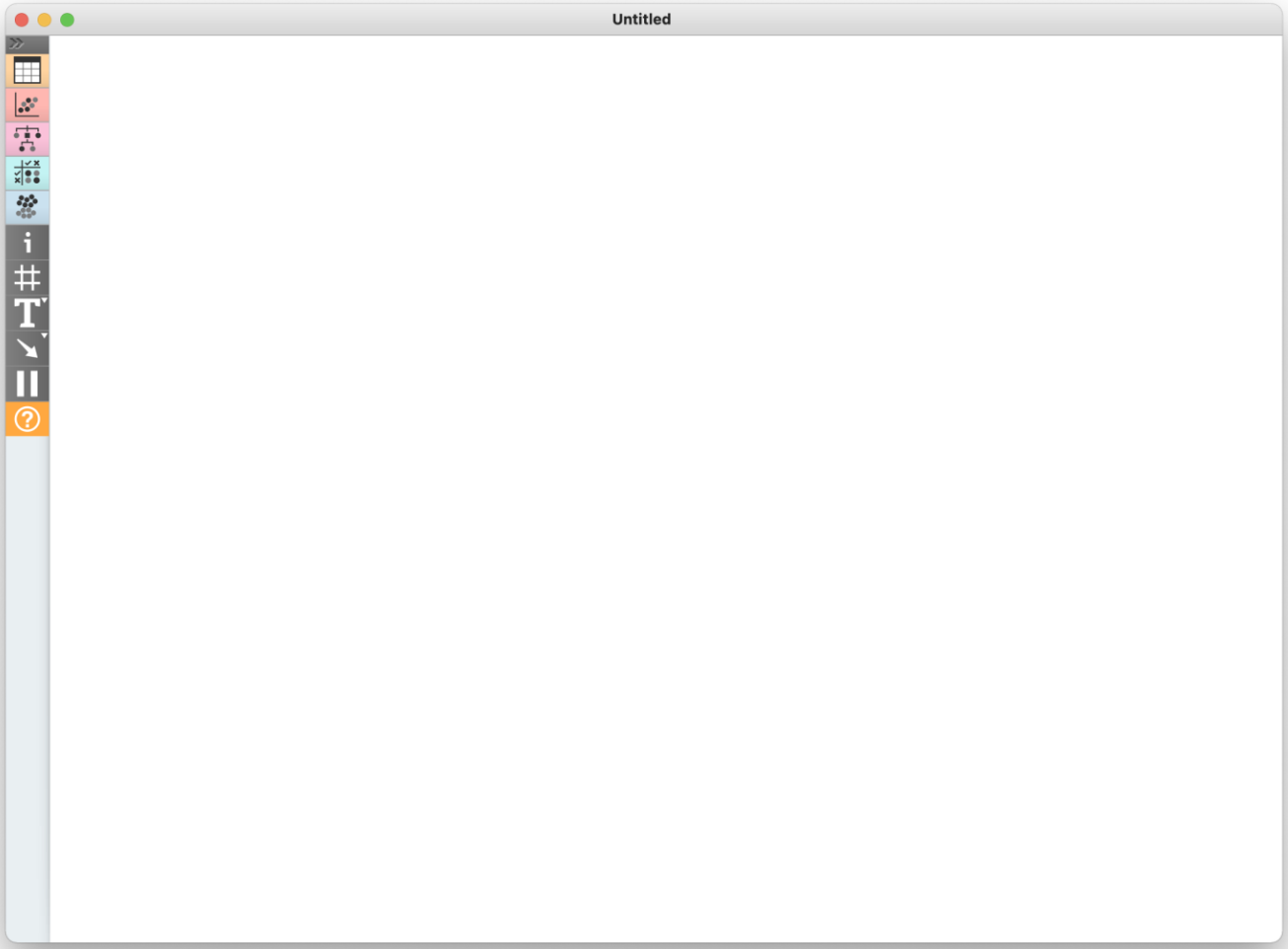
    Predict customer tastes

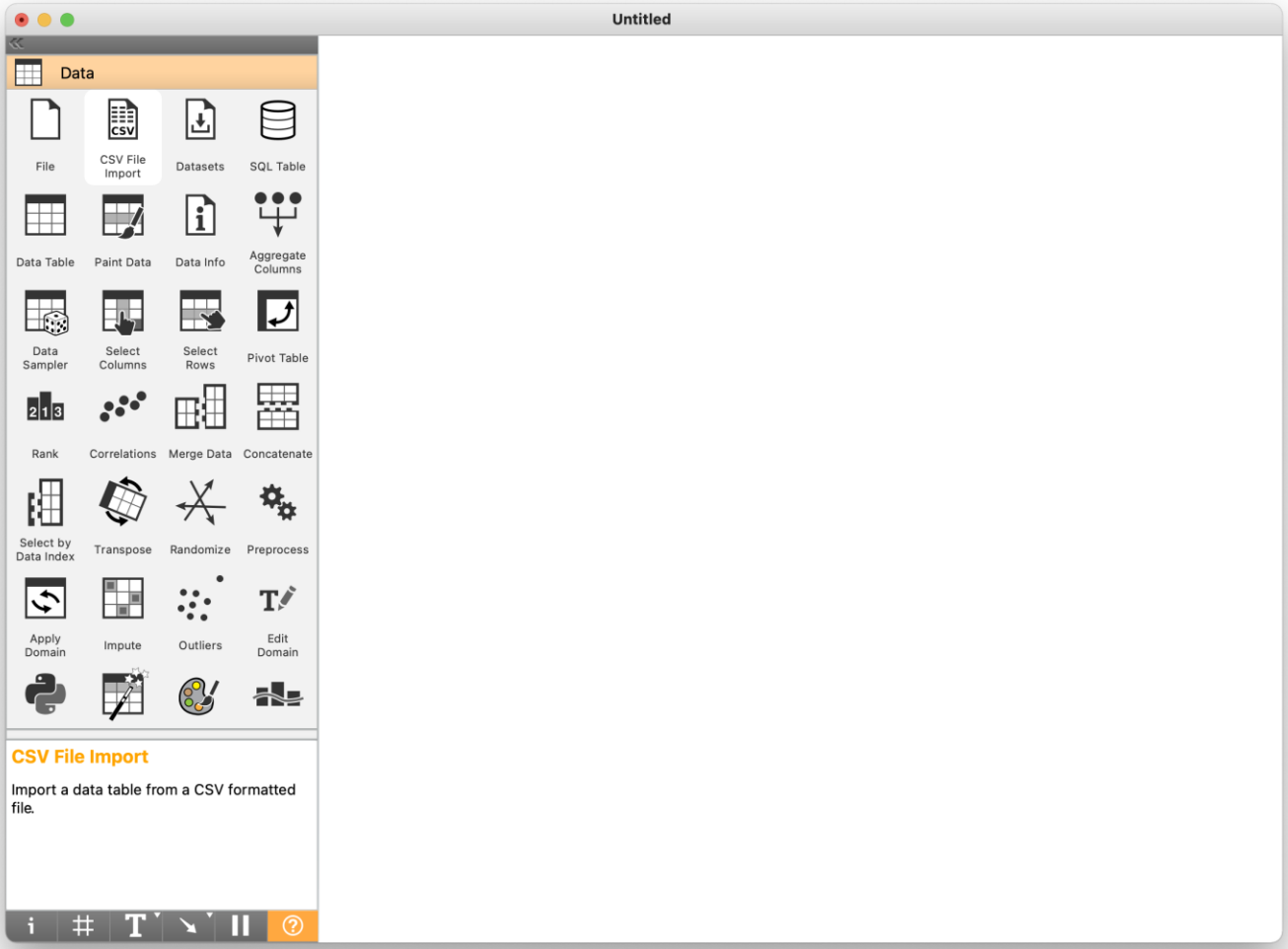
    Determine which products fail the same way

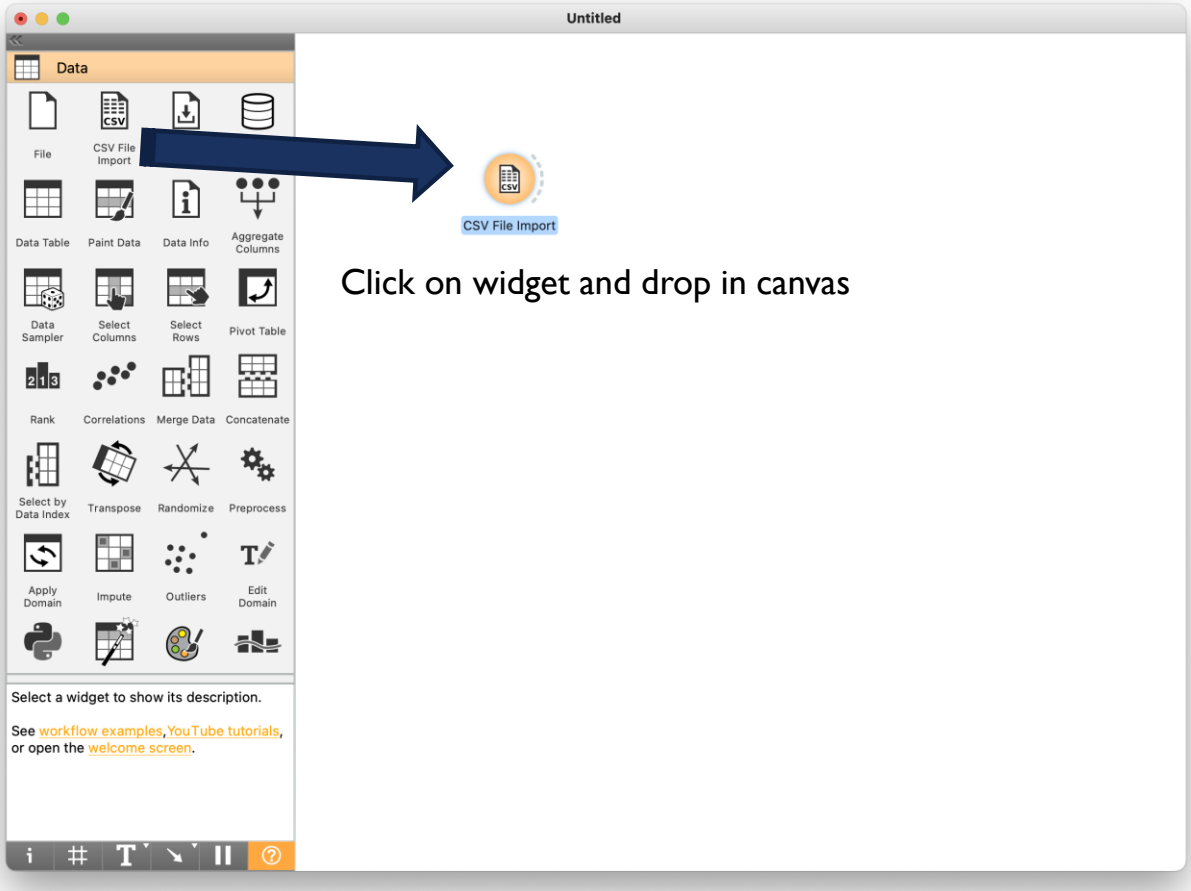
ACTIVITY

- Navigating around OrangeDM


Click
to
shrink
Untitled
Data
File
CSV File Import
Datasets
SQL Table
Data Table
Paint Data
Data Info
Aggregate Columns
Data Sampler
Select Columns
Select Rows
Pivot Table
Rank
Correlations
Merge Data
Concatenate
Select by Data Index
Transpose
Randomize
Preprocess
Apply Domain
Impute
Outliers
Edit Domain
Select a widget to show its description.
See workflow examples, YouTube tutorials, or open the welcome screen.


Untitled


Untitled
Data
File
CSV File Import
Datasets
SQL Table
Data Table
Paint Data
Data Info
Aggregate Columns
Data Sampler
Select Columns
Select Rows
Pivot Table
Rank
Correlations
Merge Data
Concatenate
Select by Data Index
Transpose
Randomize
Preprocess
Apply Domain
Impute
Outliers
Edit Domain
CSV File Import
Import a data table from a CSV formatted file.

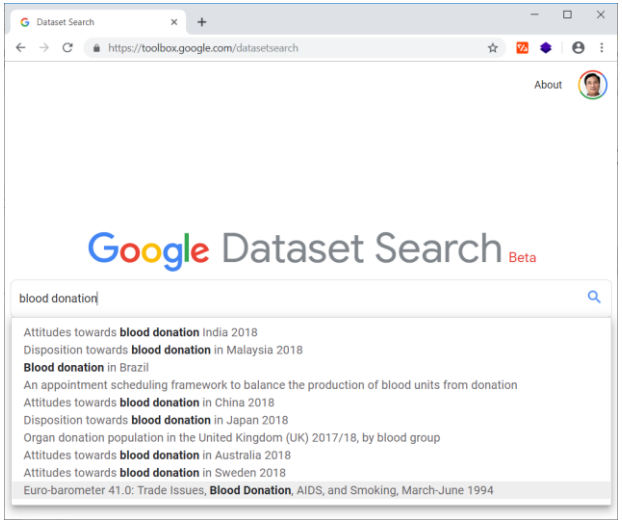


Click on widget and drop in canvas

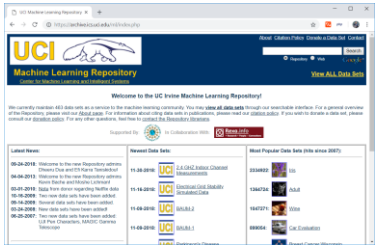
IMPORTING DATA

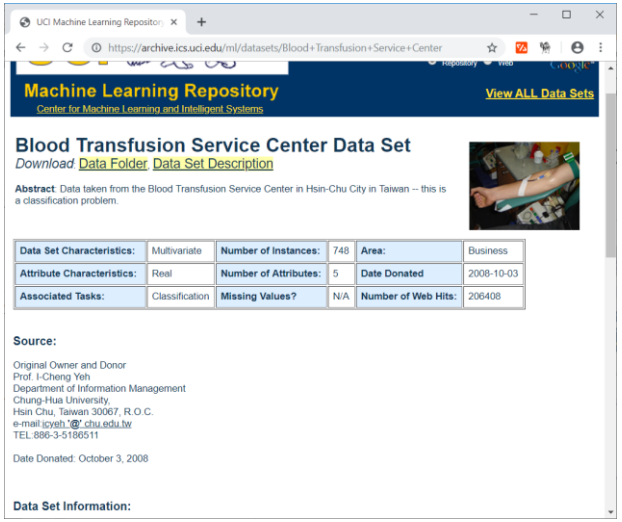
SEARCH FOR THE "BLOOD DONATION" DATASET

- Use Google Dataset Search Toolbox
- https://toolbox.google.com/datasetsearch



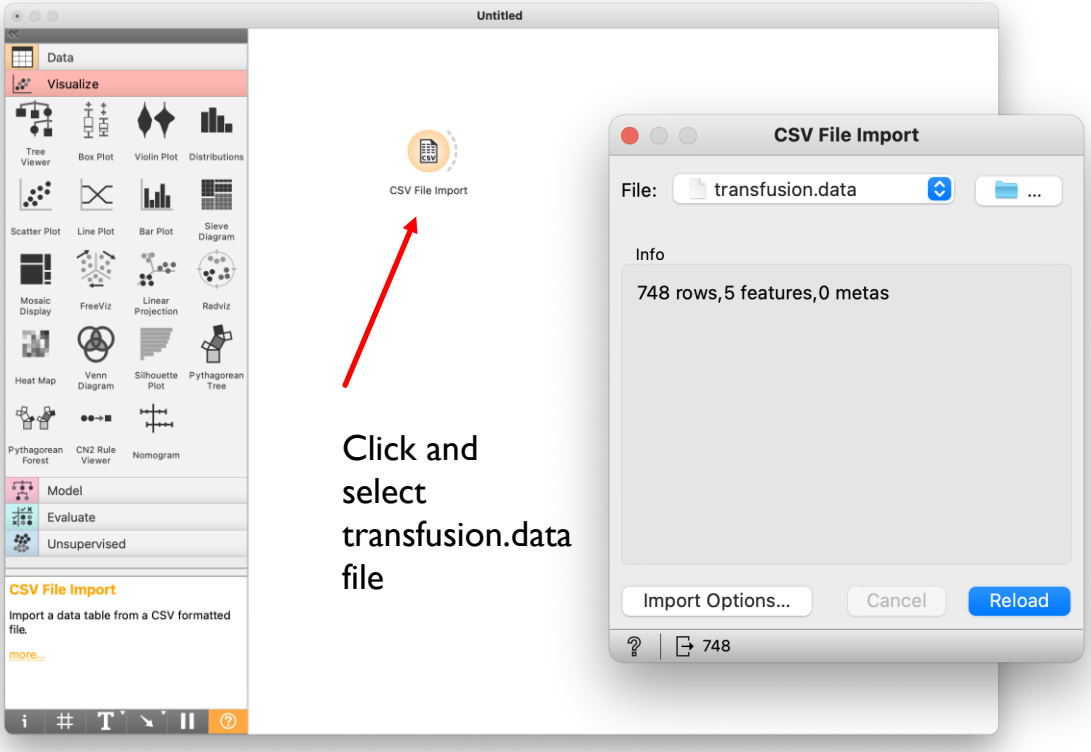
SEARCH FOR THE "BLOOD DONATION" DATASET

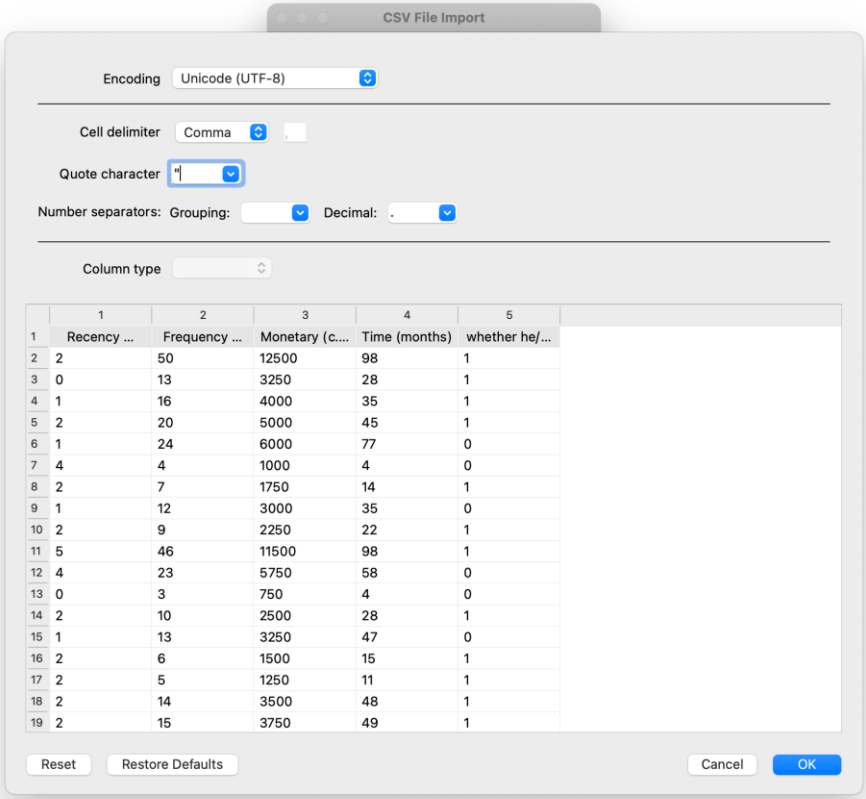




- Use UCI Machine Learning Repository
- https://archive.ics.uci.edu/ml/index.php

USE THE CSV FILE IMPORT

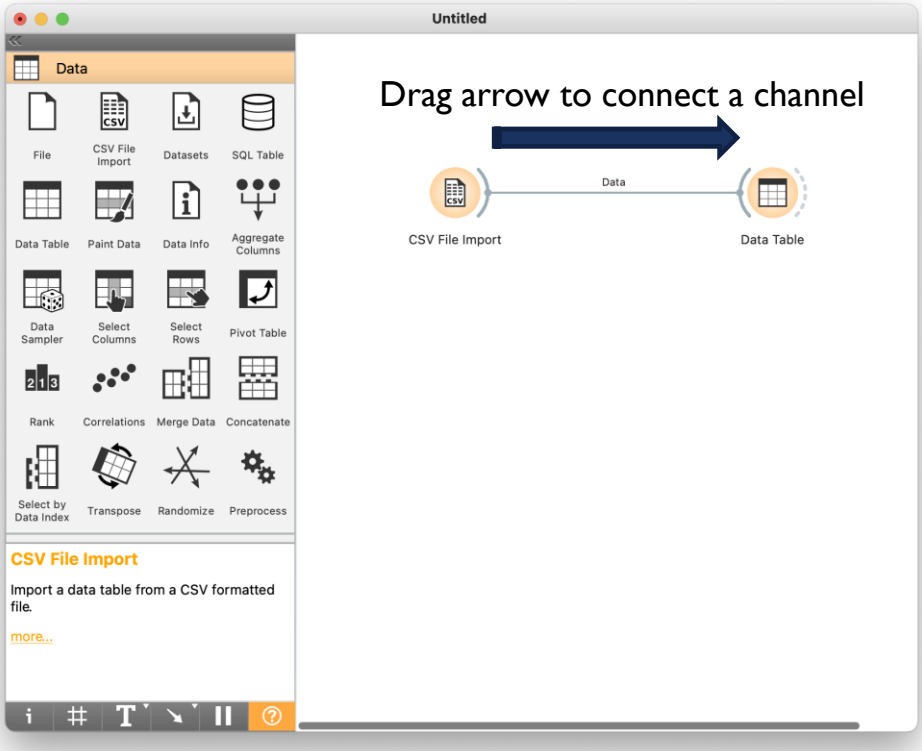




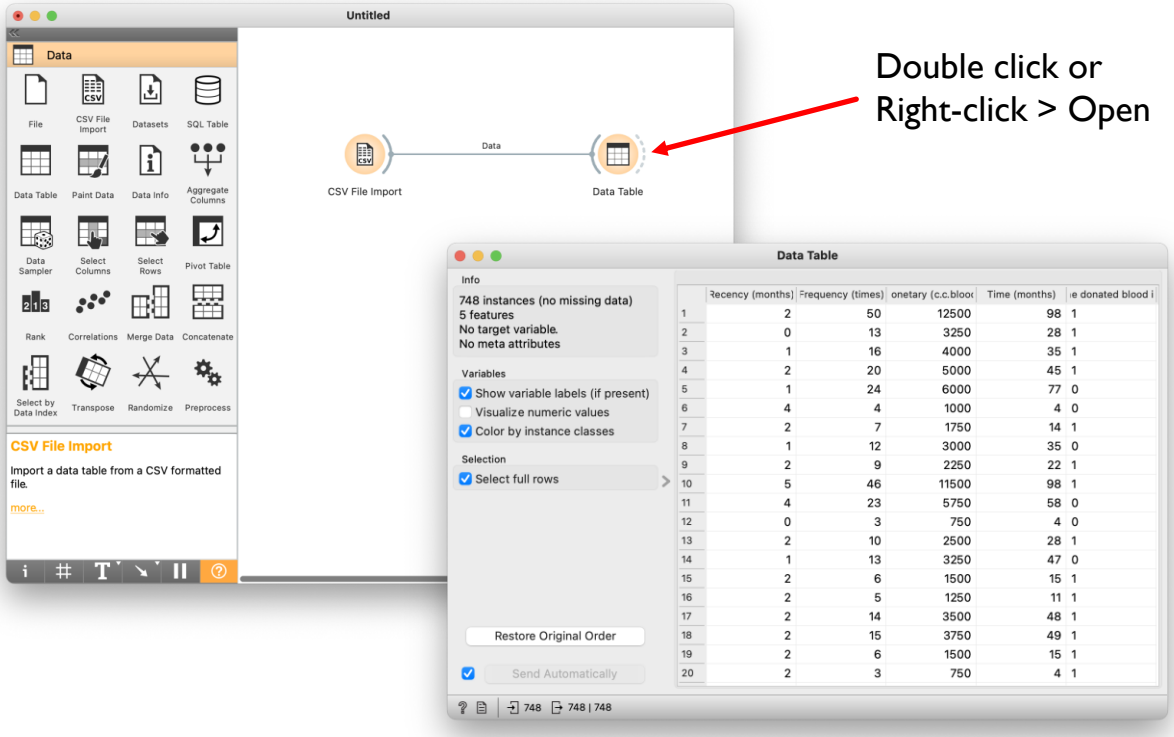
	1	2	3	4	5
1	Recency ...	Frequency ...	Monetary (c....	Time (months)	whether he/...
2	2	50	12500	98	1
3	0	13	3250	28	1
4	1	16	4000	35	1
5	2	20	5000	45	1
6	1	24	6000	77	0
7	4	4	1000	4	0
8	2	7	1750	14	1
9	1	12	3000	35	0
10	2	9	2250	22	1
11	5	46	11500	98	1
12	4	23	5750	58	0
13	0	3	750	4	0
14	2	10	2500	28	1
15	1	13	3250	47	0
16	2	6	1500	15	1
17	2	5	1250	11	1
18	2	14	3500	48	1
19	2	15	3750	49	1

DATA PREPARATION

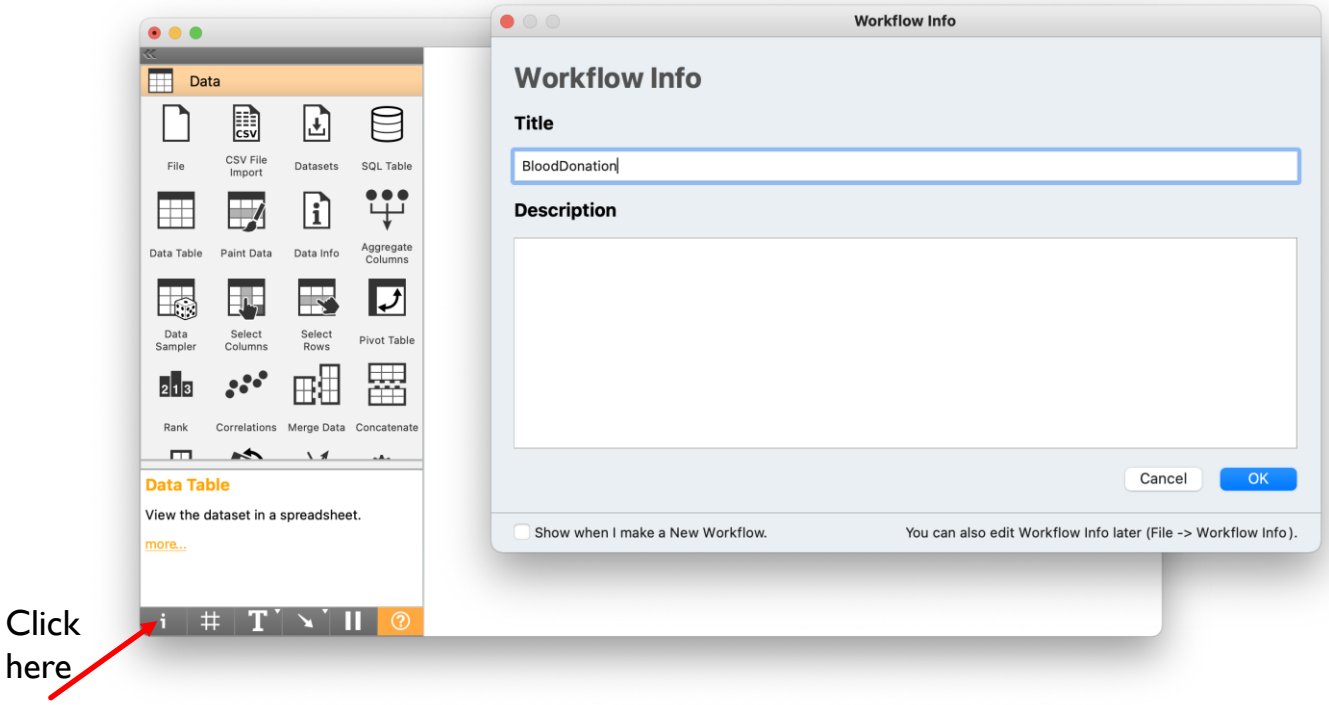
ADD A MODULE TO CHECK THE DATA

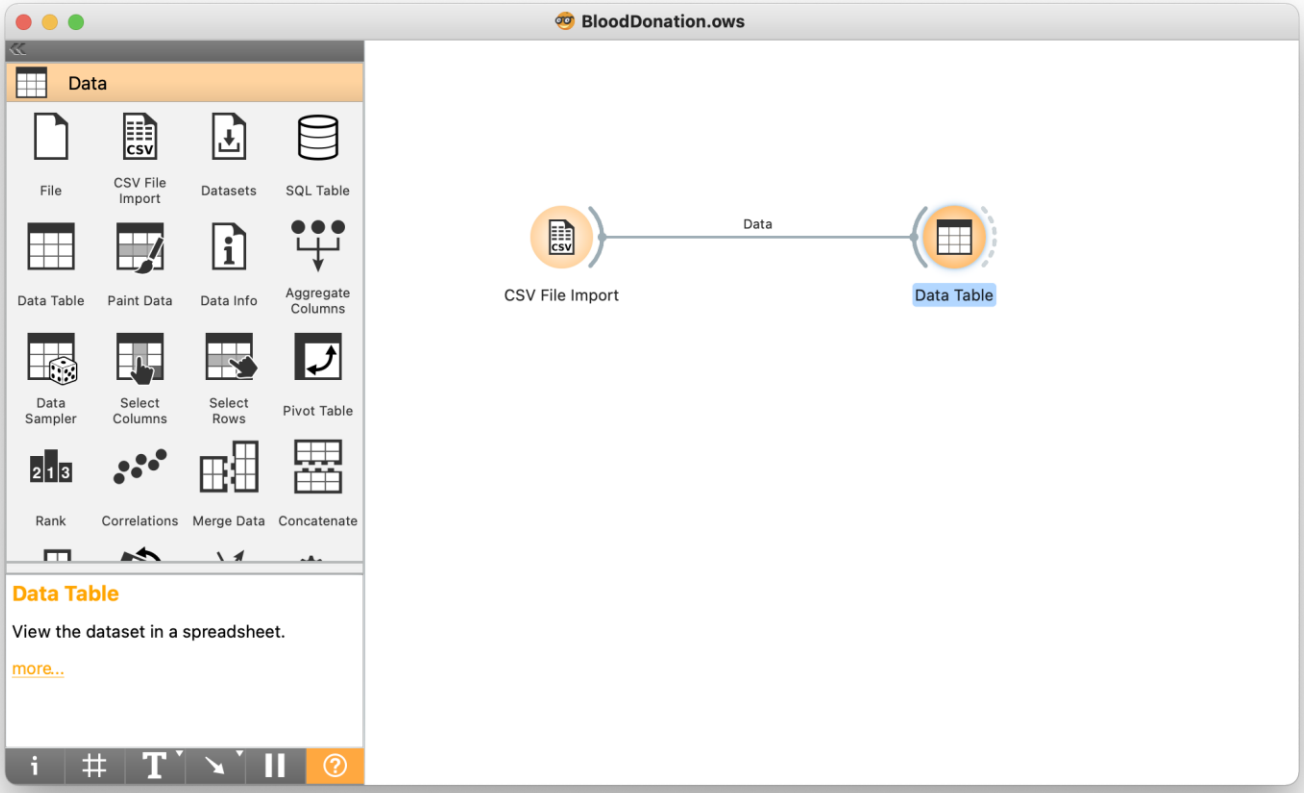


ADD A MODULE TO CHECK THE DATA

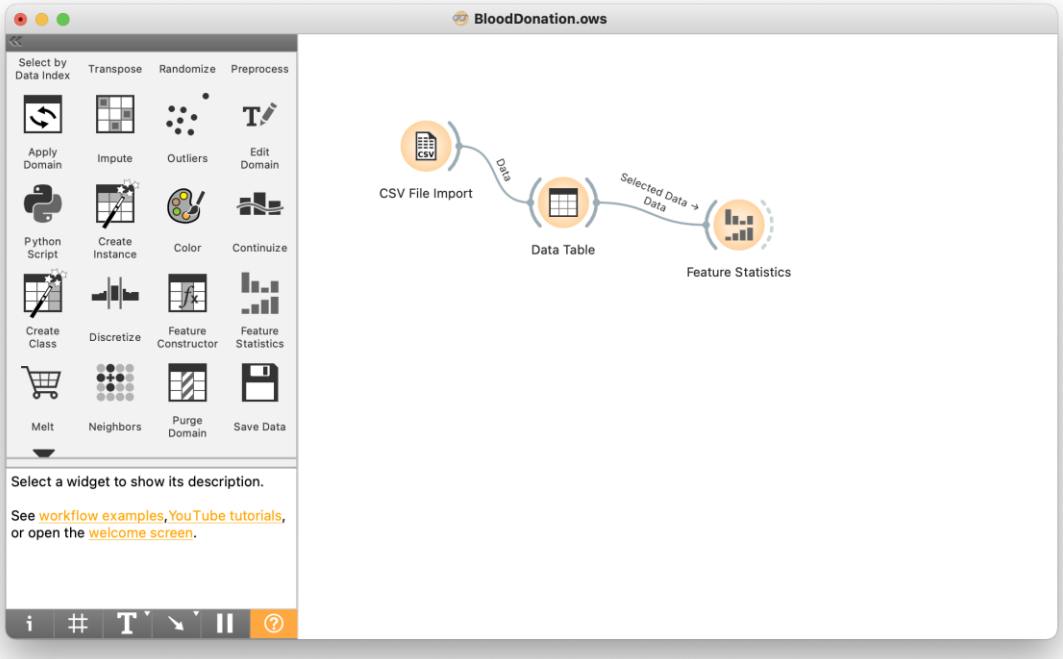


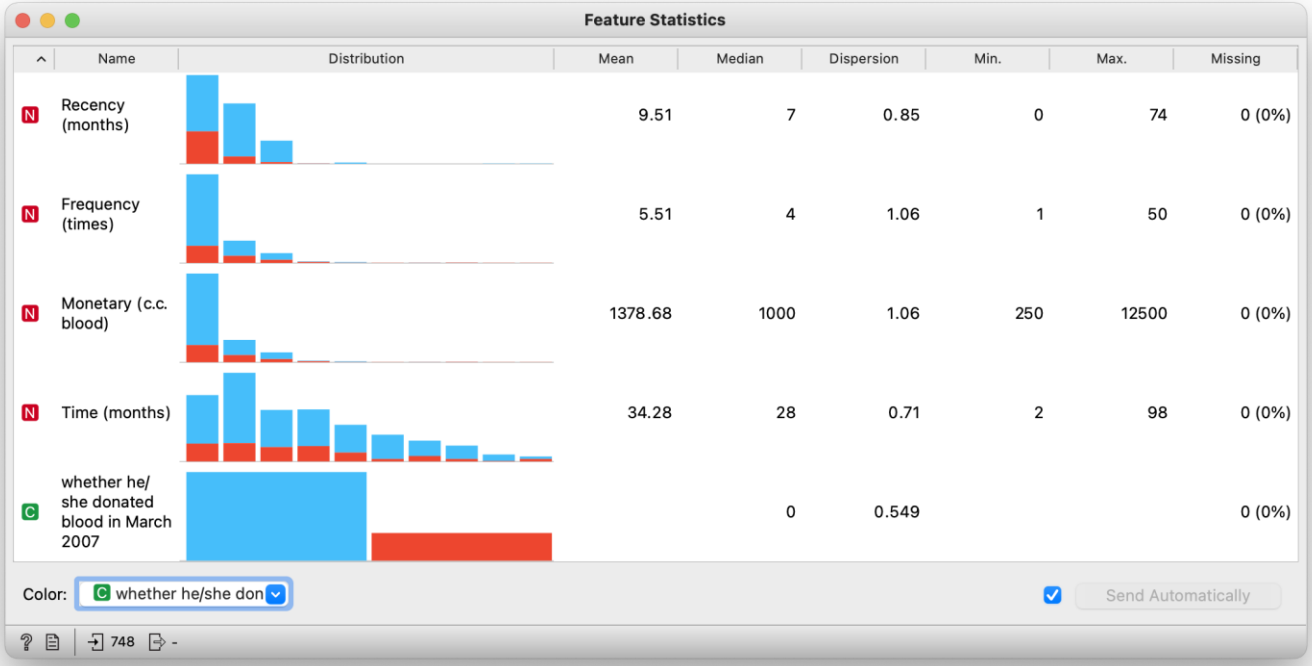
CHANGE WORKFLOW INFO TO BLOODTRANSFUSION. SAVE FILE.




BloodDonation.ows
Data
File
CSV File Import
Datasets
SQL Table
Data Table
Paint Data
Data Info
Aggregate Columns
Data Sampler
Select Columns
Select Rows
Pivot Table
Rank
Correlations
Merge Data
Concatenate
Data Table
View the dataset in a spreadsheet.
more...
Data
CSV File Import
Data Table

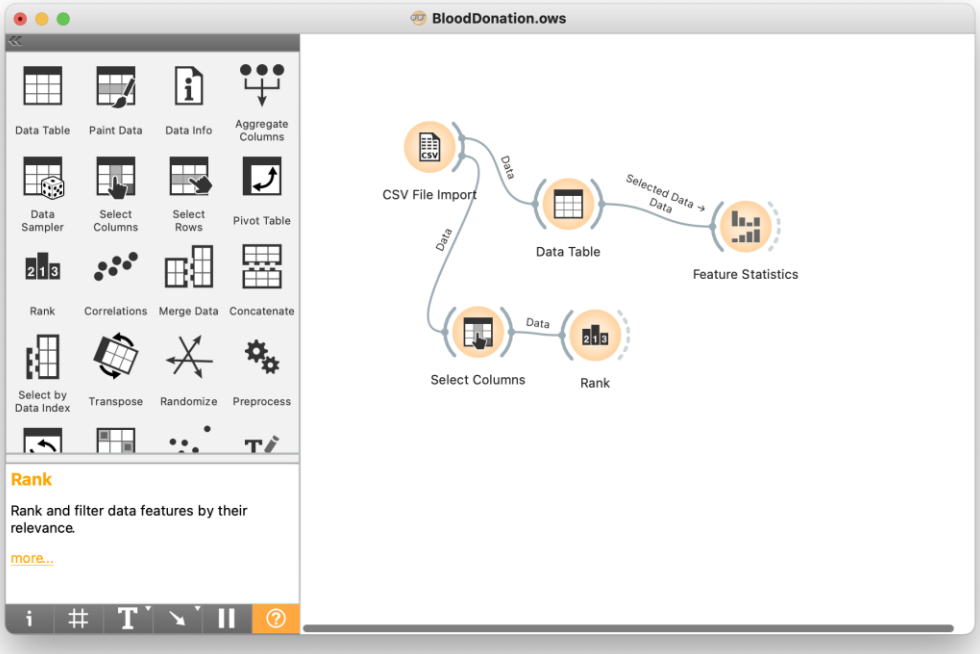
CHECK DATA QUALITY

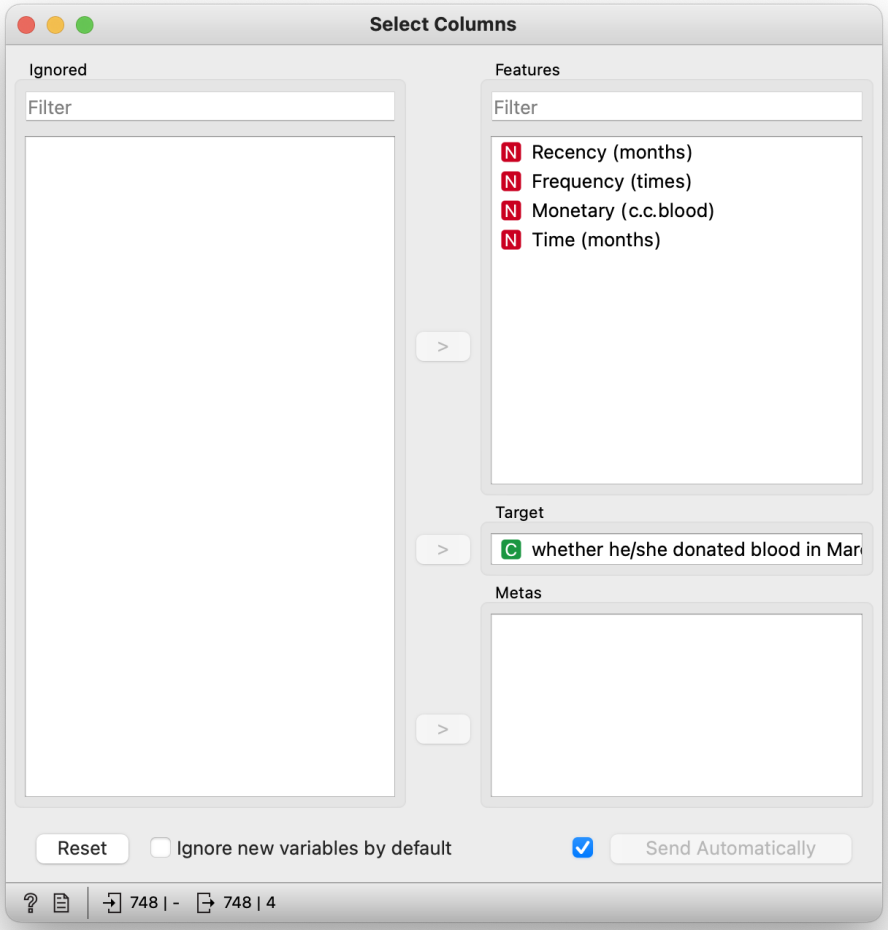


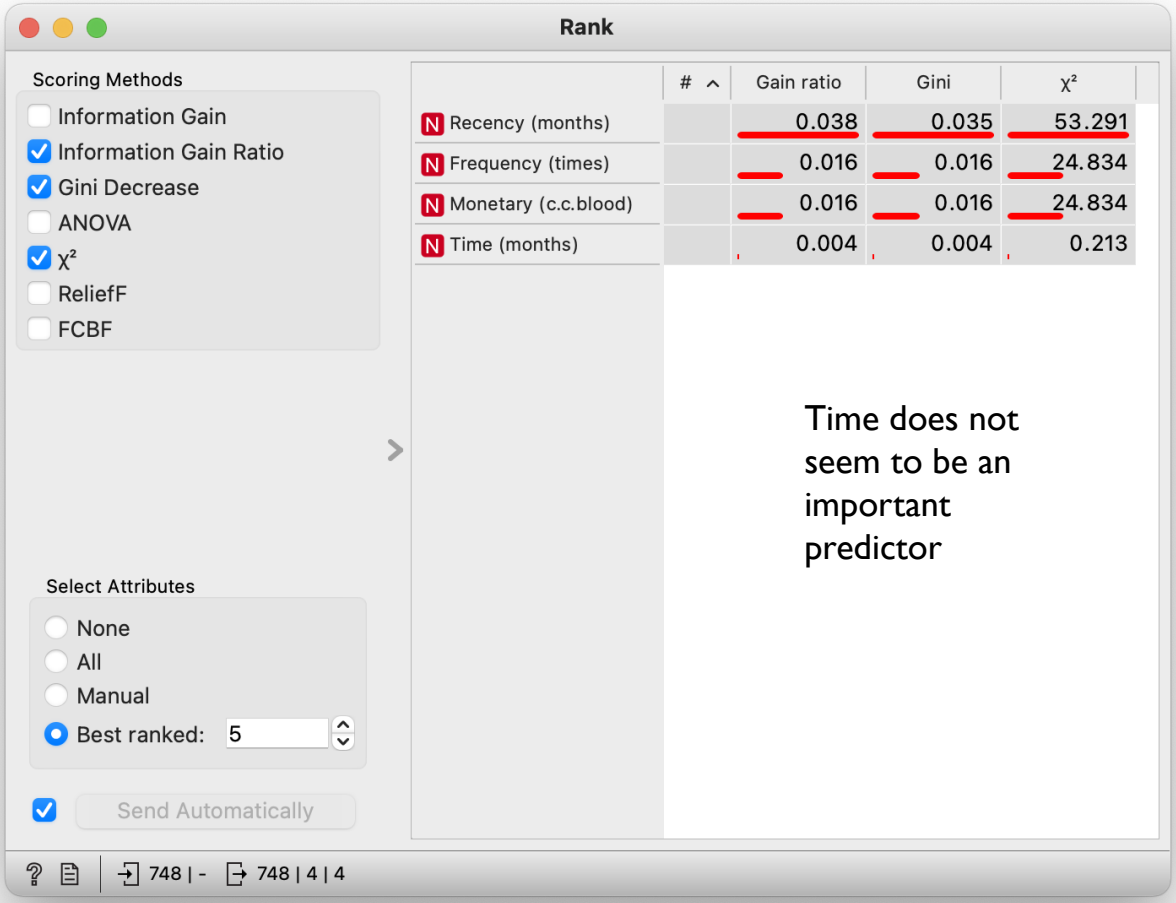

Feature Statistics
Name
Distribution
Mean
Median
Dispersion
Min.
Max.
Missing
Recency (months)
9.51
7
0.85
0
74
0 (0%)
Frequency (times)
5.51
4
1.06
1
50
0 (0%)
Monetary (c.c. blood)
1378.68
1000
1.06
250
12500
0 (0%)
Time (months)
34.28
28
0.71
2
98
0 (0%)
whether he/she donated blood in March 2007
0
0.549
0 (0%)
Color:
whether he/she don
Send Automatically
748

FEATURE SELECTION

FEATURE SELECTION

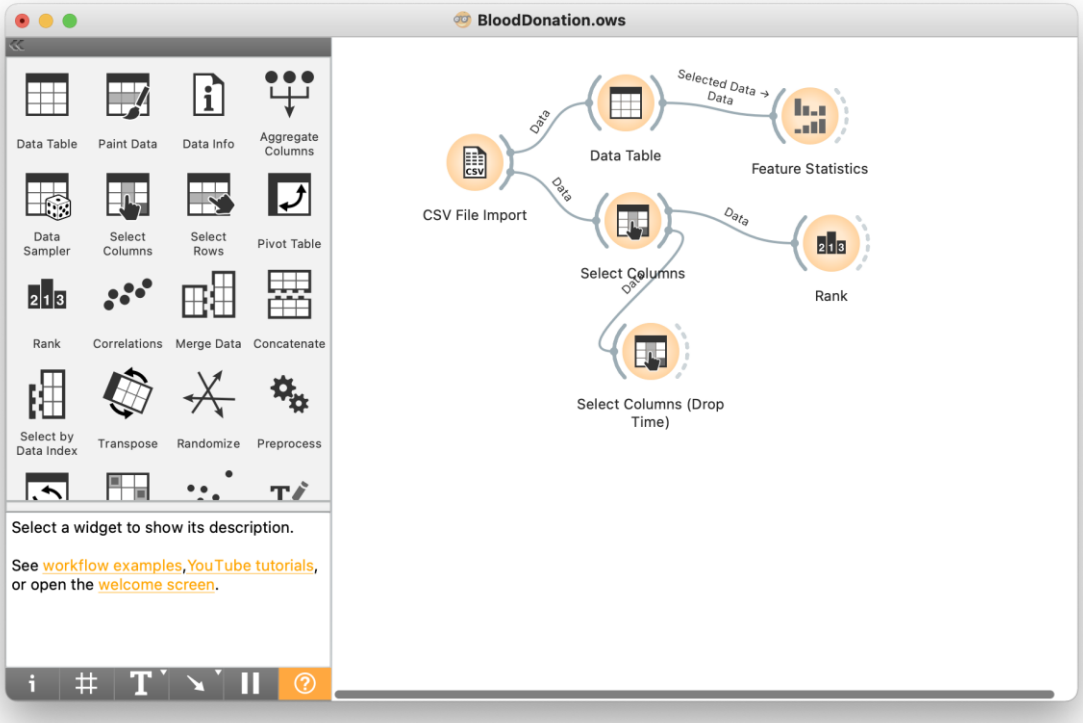


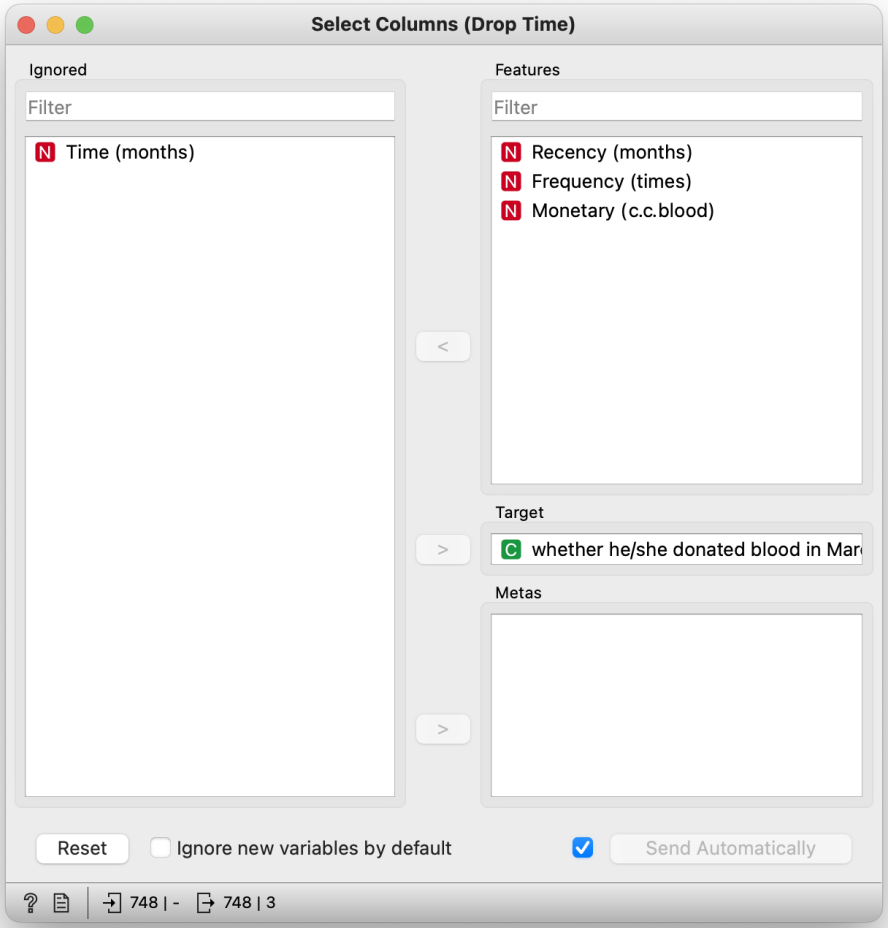

Select Columns
Ignored
Filter
Features
Filter
N Recency (months)
N Frequency (times)
N Monetary (c.c.blood)
N Time (months)
>
Target
>
C whether he/she donated blood in Mar
Metas
>
Reset
Ignore new variables by default
Send Automatically
748 | -
748 | 4


Rank
Scoring Methods
Information Gain
Information Gain Ratio
Gini Decrease
ANOVA
χ²
ReliefF
FCBF
Select Attributes
None
All
Manual
Best ranked: 5
Send Automatically
748 | -
748 | 4 | 4
Gain ratio Gini χ²
Recency (months) 0.038 0.035 53.291
Frequency (times) 0.016 0.016 24.834
Monetary (c.c.blood) 0.016 0.016 24.834
Time (months) 0.004 0.004 0.213
Time does not seem to be an important predictor

DROP TIME

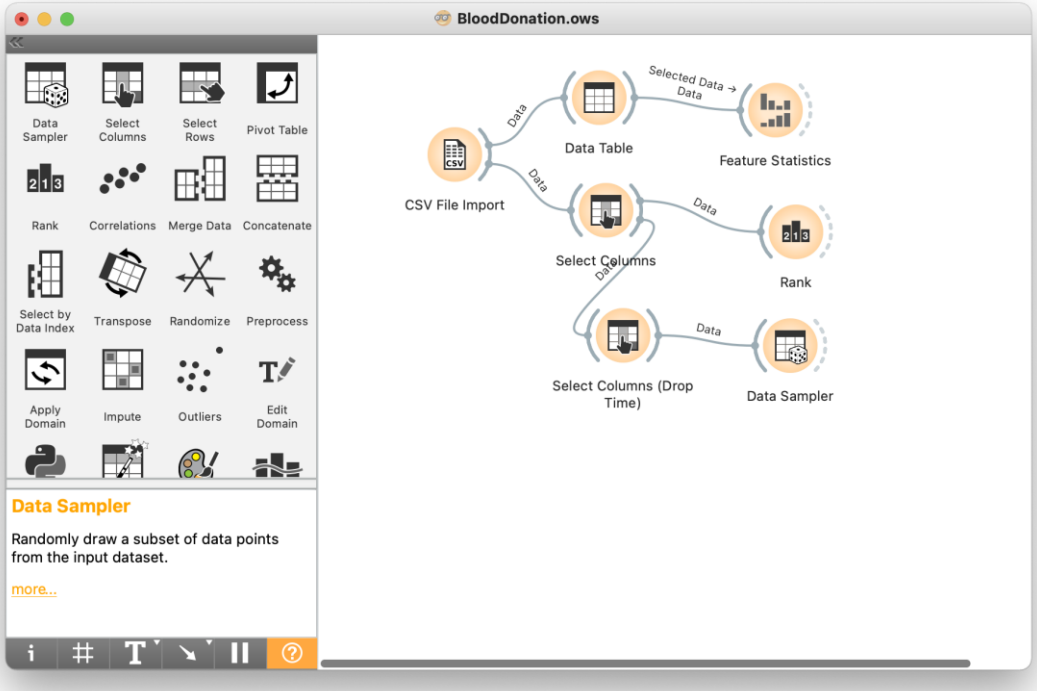
USE SECOND SELECT COLUMNS

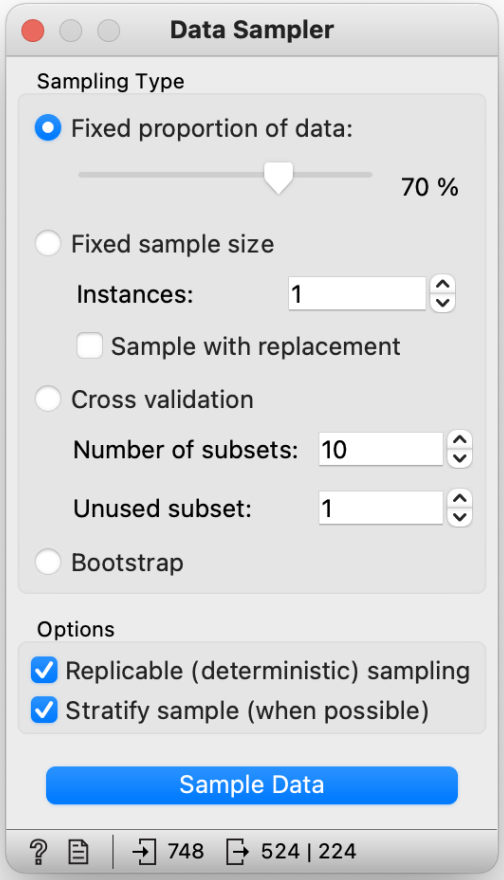



Select Columns (Drop Time)
Ignored
Filter
Time (months)
Features
Filter
Recency (months)
Frequency (times)
Monetary (c.c.blood)
<
Target
>
whether he/she donated blood in Mar
Metas
>
Reset
Ignore new variables by default
Send Automatically
748 | -
748 | 3

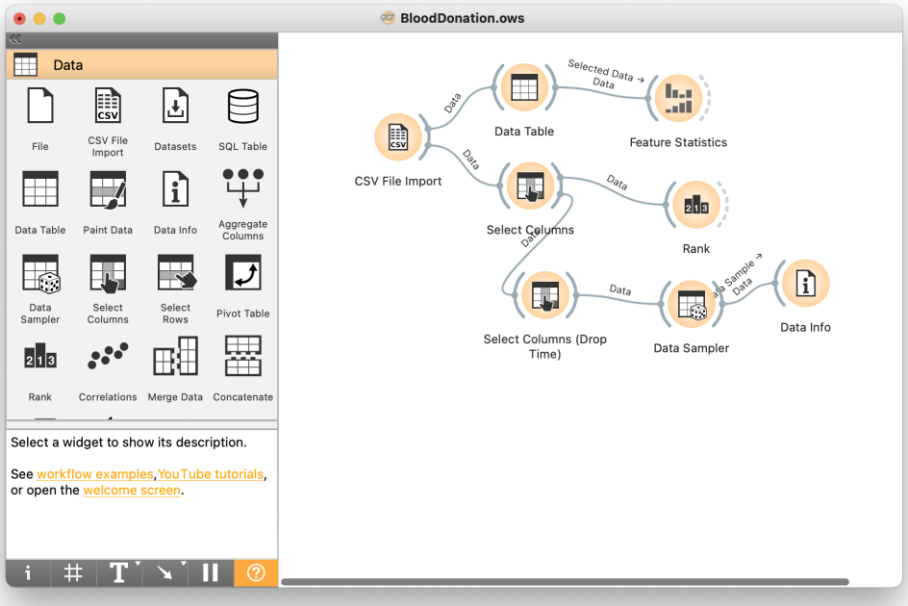
TRAIN TEST SPLIT

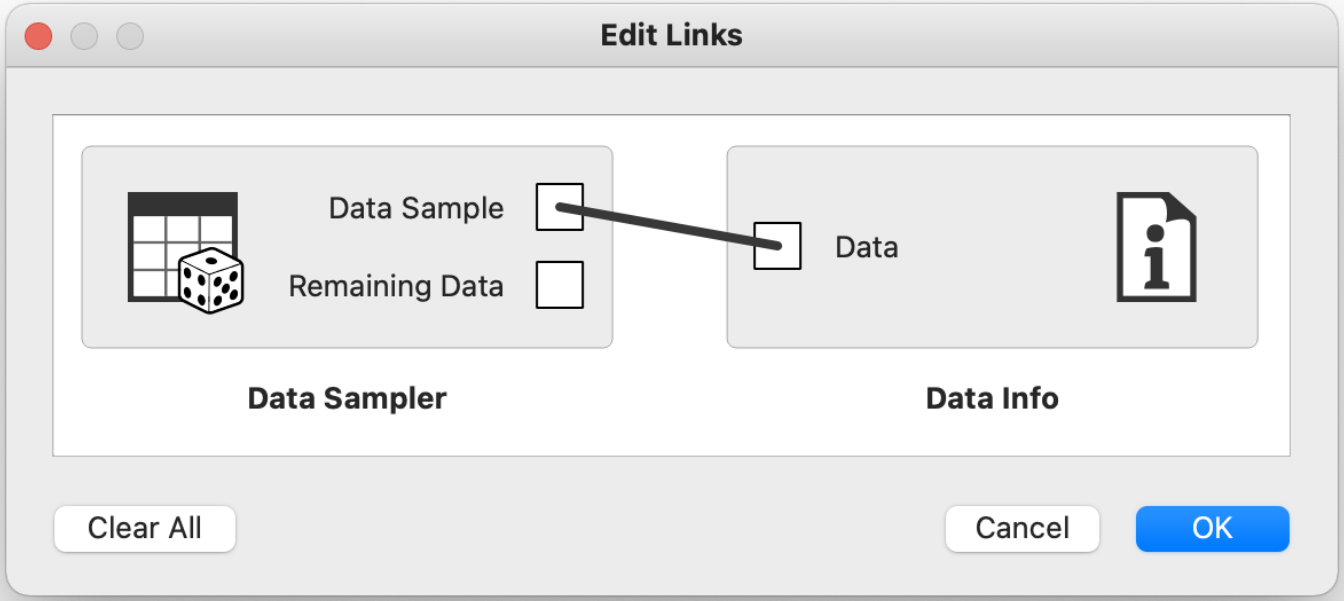
USE THE DATA SAMPLER

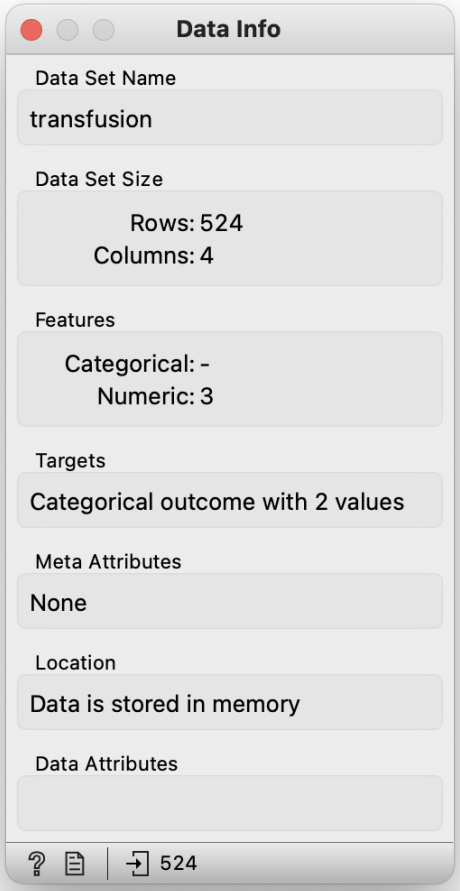



Data Sampler
Sampling Type
Fixed proportion of data:
70 %
Fixed sample size
Instances:
1
Sample with replacement
Cross validation
Number of subsets:
10
Unused subset:
1
Bootstrap
Options
Replicable (deterministic) sampling
Stratify sample (when possible)
Sample Data
748
524 | 224

USE DATA INFO TO CHECK THE DATA SAMPLING



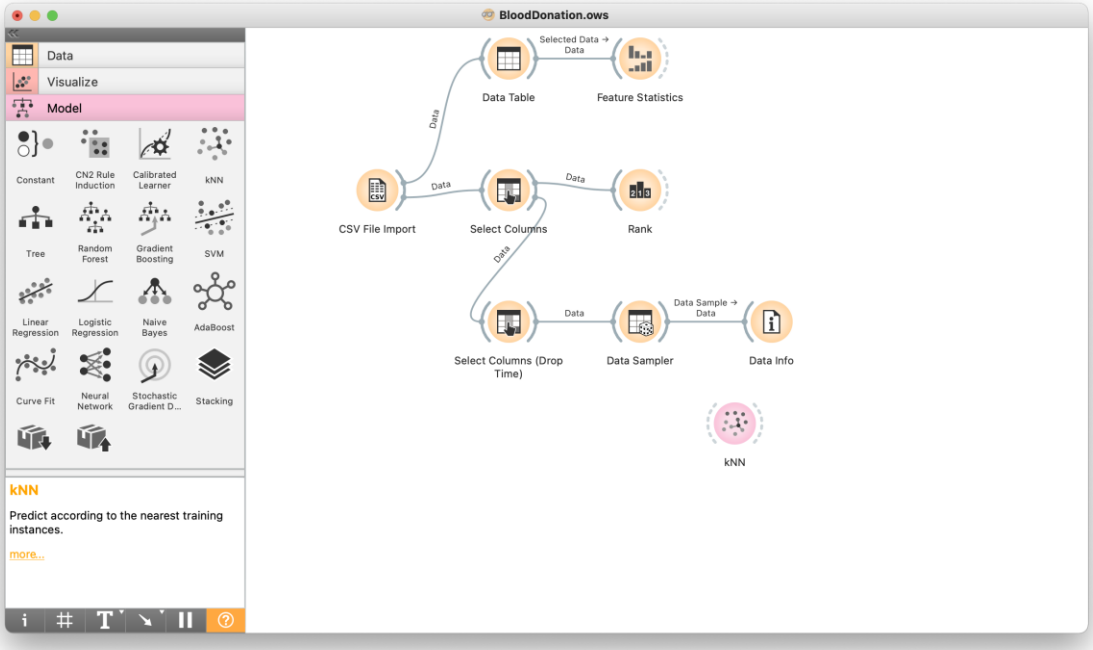

Edit Links
Data Sample
Remaining Data
Data
Data Sampler
Data Info
Clear All
Cancel
OK


Data Info
Data Set Name
transfusion
Data Set Size
Rows: 524
Columns: 4
Features
Categorical: -
Numeric: 3
Targets
Categorical outcome with 2 values
Meta Attributes
None
Location
Data is stored in memory
Data Attributes
524

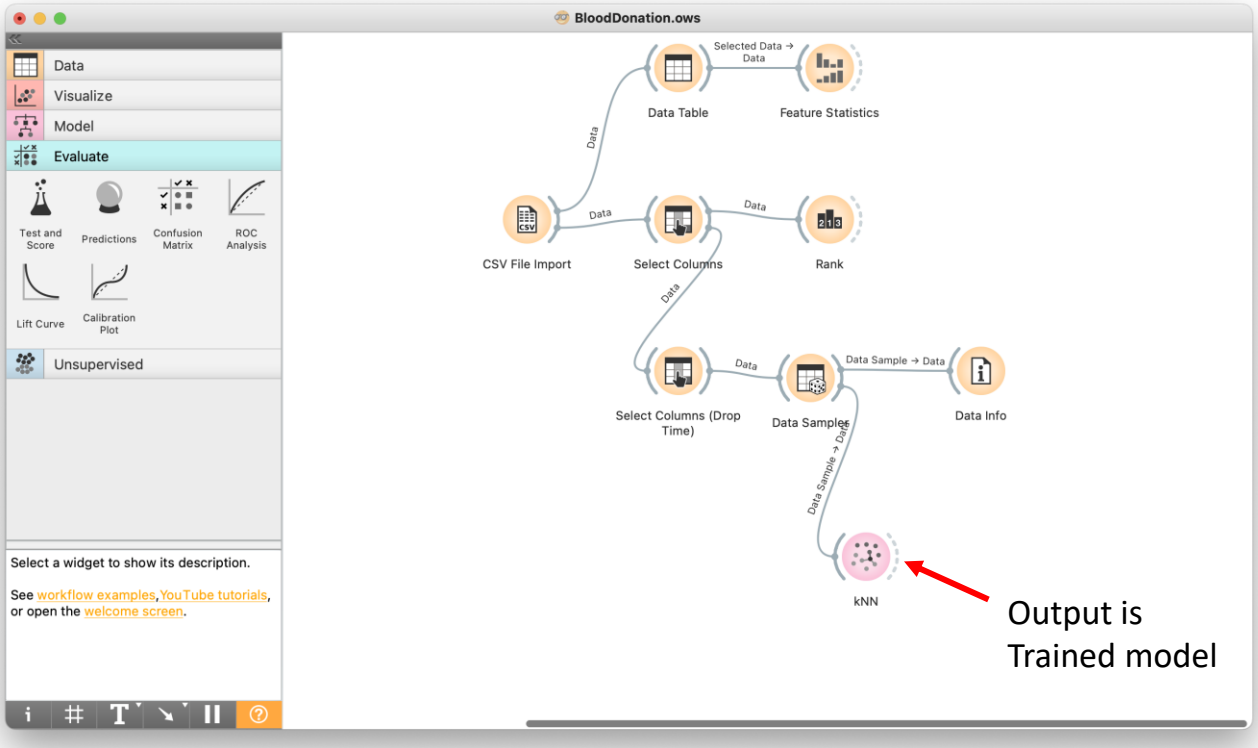
END OF DATA PREPARATION

Training and Scoring

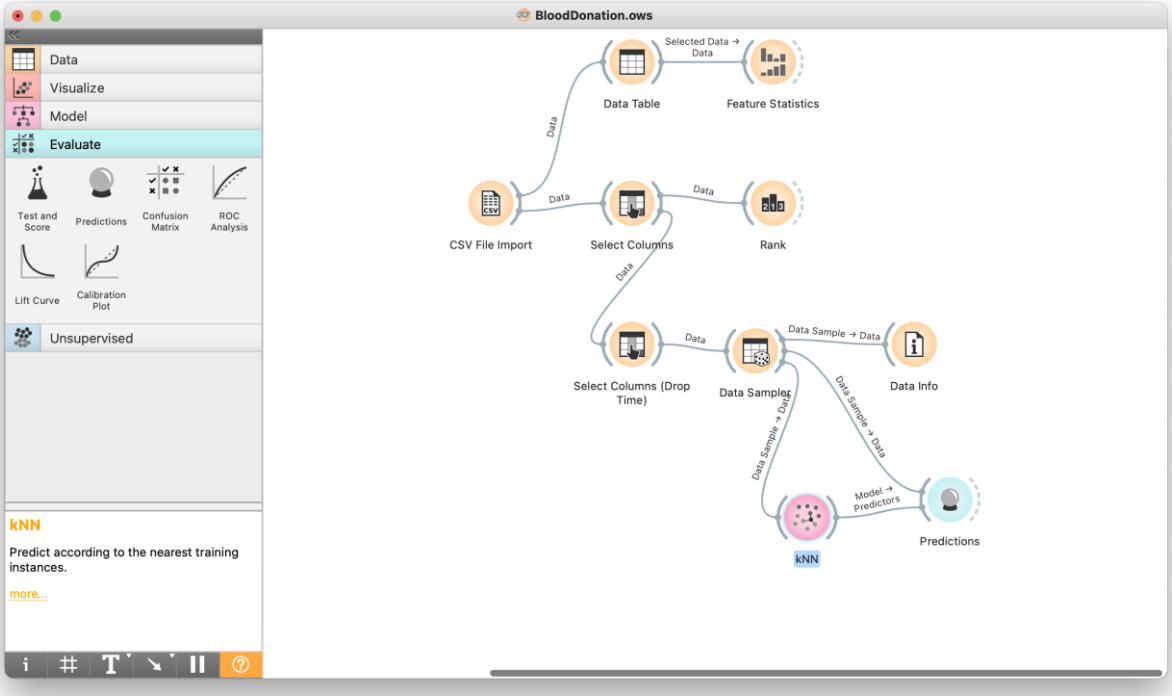
Use the KNN


BloodDonation.ows
Data
Visualize
Model
Constant
CN2 Rule Induction
Calibrated Learner
kNN
Tree
Random Forest
Gradient Boosting
SVM
Linear Regression
Logistic Regression
Naive Bayes
AdaBoost
Curve Fit
Neural Network
Stochastic Gradient D...
Stacking
kNN
Predict according to the nearest training instances.
more...
Data Table
Feature Statistics
CSV File Import
Select Columns
Rank
Select Columns (Drop Time)
Data Sampler
Data Info
kNN
Selected Data → Data
Data
Data Sample → Data

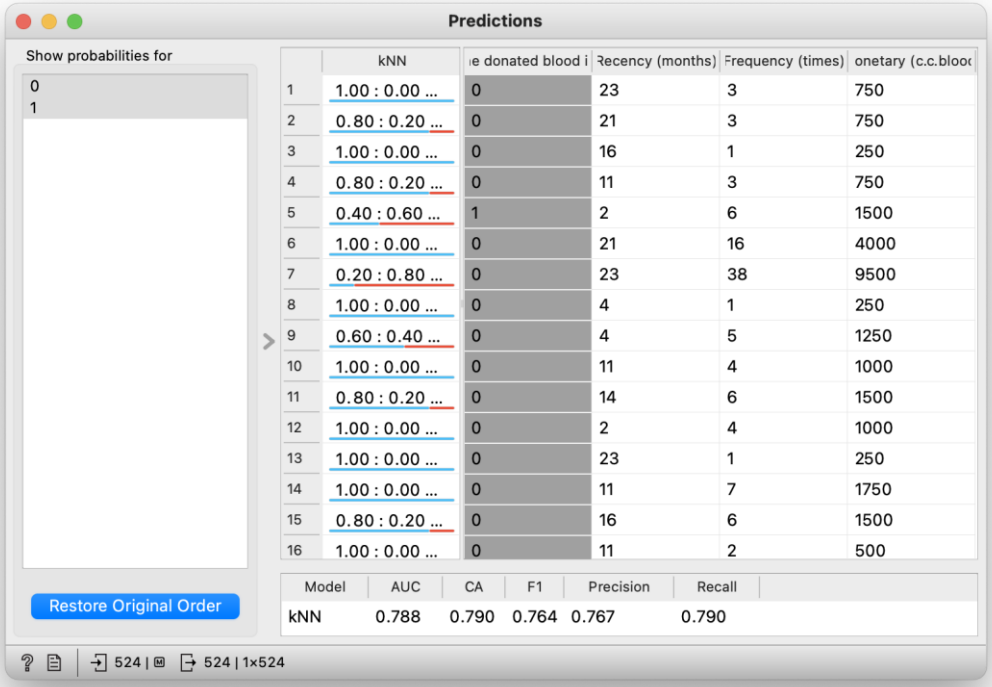
Connect the training data to KNN



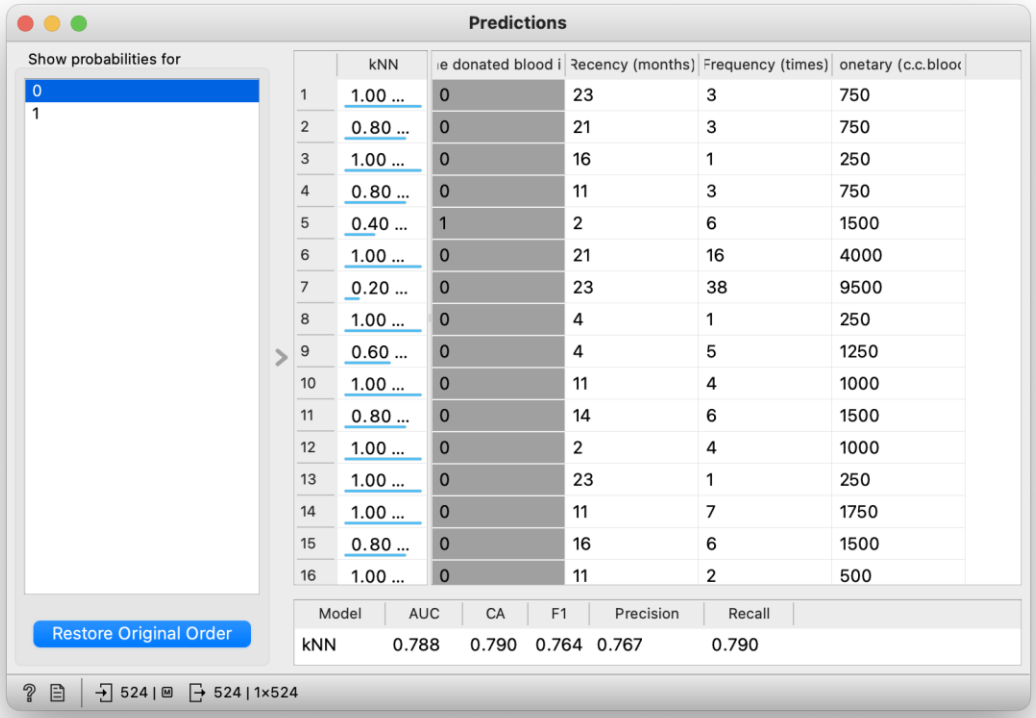
Use the Predictions Widget



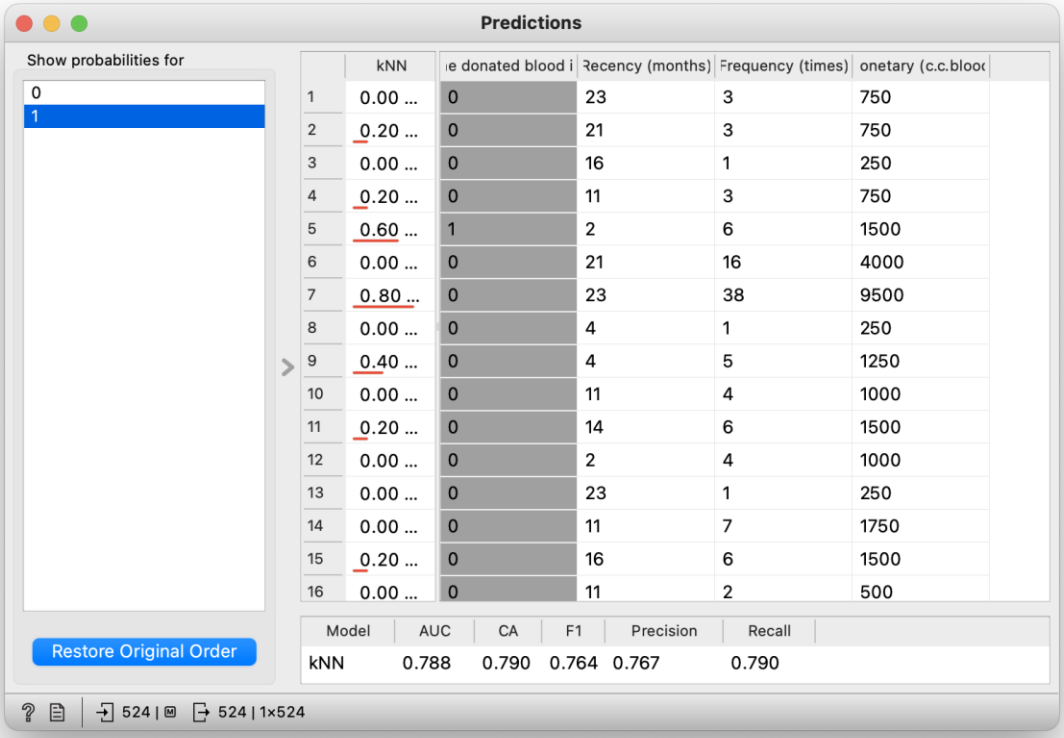
The Prediction uses the trained model from the KNN



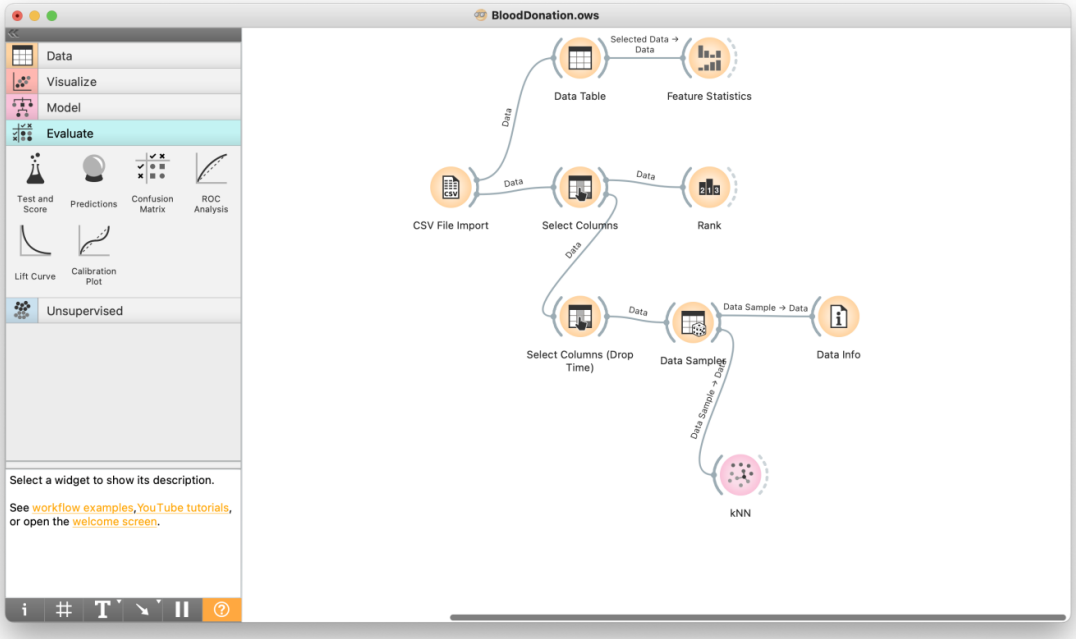
Predict 0


Predictions
Show probabilities for
0
1
Restore Original Order
kNN
e donated blood i
Recency (months)
Frequency (times)
onetary (c.c.blooc
1 1.00 ... 0 23 3 750
2 0.80 ... 0 21 3 750
3 1.00 ... 0 16 1 250
4 0.80 ... 0 11 3 750
5 0.40 ... 1 2 6 1500
6 1.00 ... 0 21 16 4000
7 0.20 ... 0 23 38 9500
8 1.00 ... 0 4 1 250
9 0.60 ... 0 4 5 1250
10 1.00 ... 0 11 4 1000
11 0.80 ... 0 14 6 1500
12 1.00 ... 0 2 4 1000
13 1.00 ... 0 23 1 250
14 1.00 ... 0 11 7 1750
15 0.80 ... 0 16 6 1500
16 1.00 ... 0 11 2 500
Model AUC CA F1 Precision Recall
kNN 0.788 0.790 0.764 0.767 0.790
524 | 524 | 1x524

Predict 1


Predictions
Show probabilities for
0
1
Restore Original Order
kNN
e donated blood i
Recency (months)
Frequency (times)
onetary (c.c.blooc
1 0.00 ... 0 23 3 750
2 0.20 ... 0 21 3 750
3 0.00 ... 0 16 1 250
4 0.20 ... 0 11 3 750
5 0.60 ... 1 2 6 1500
6 0.00 ... 0 21 16 4000
7 0.80 ... 0 23 38 9500
8 0.00 ... 0 4 1 250
9 0.40 ... 0 4 5 1250
10 0.00 ... 0 11 4 1000
11 0.20 ... 0 14 6 1500
12 0.00 ... 0 2 4 1000
13 0.00 ... 0 23 1 250
14 0.00 ... 0 11 7 1750
15 0.20 ... 0 16 6 1500
16 0.00 ... 0 11 2 500
Model AUC CA F1 Precision Recall
kNN 0.788 0.790 0.764 0.767 0.790
524 | 524 | 1×524

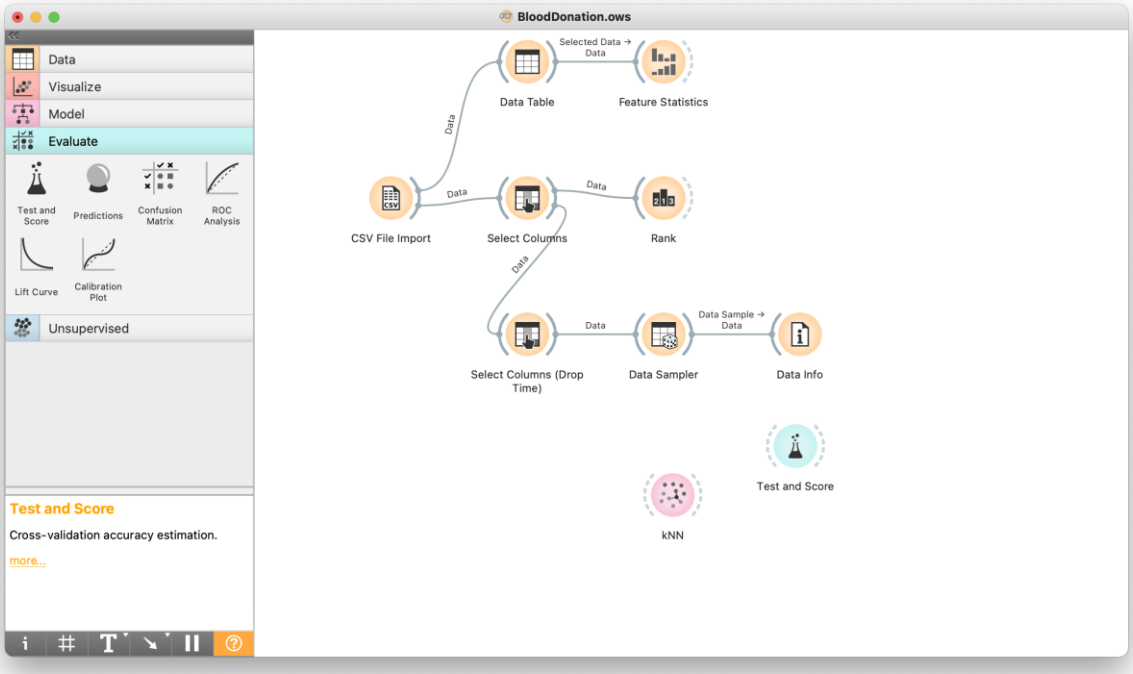
Remove the predictions/scorer

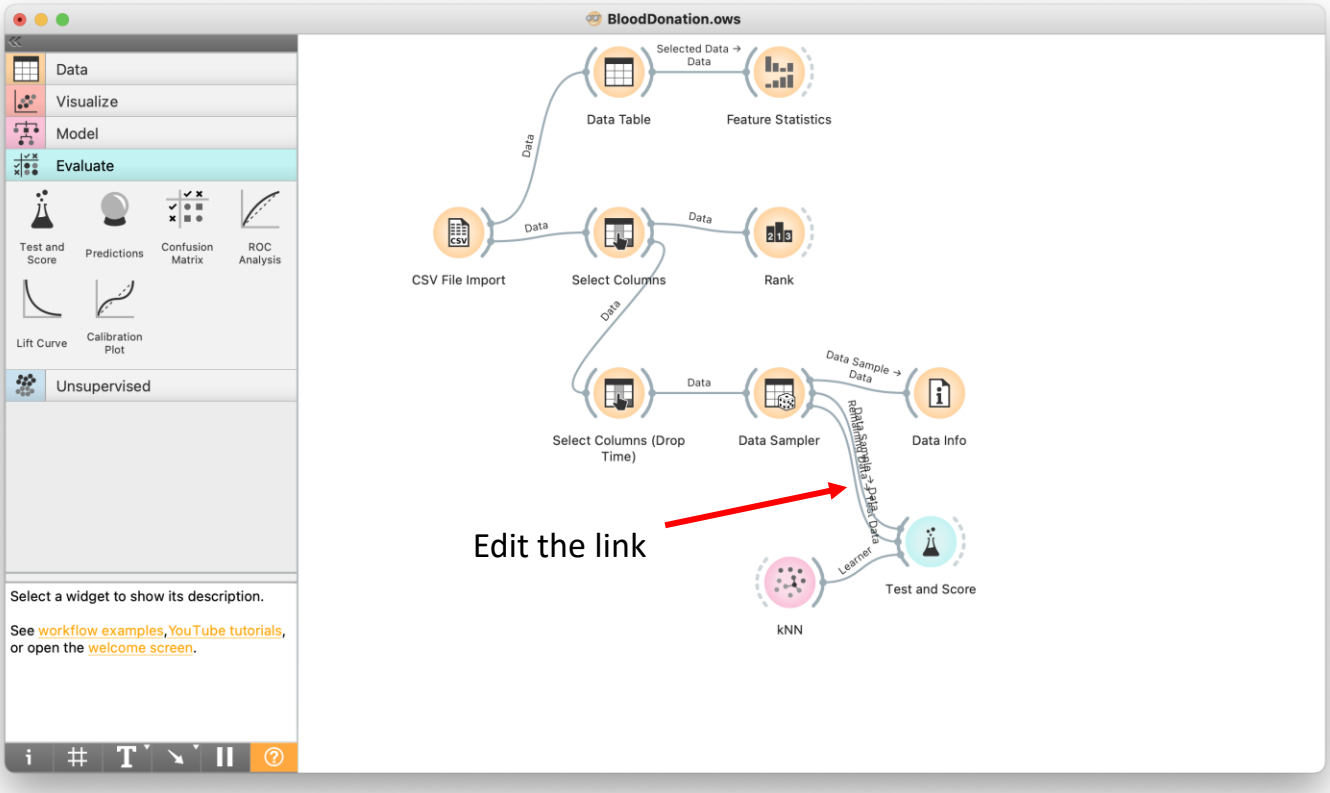


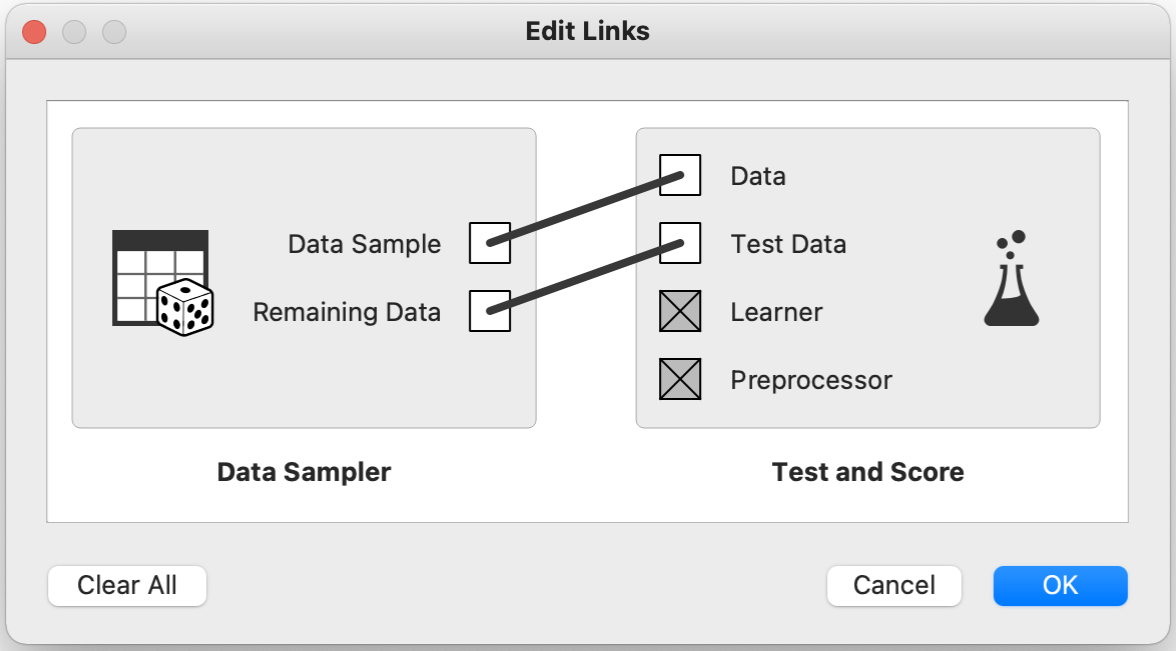
Evaluation

Connect them to the train and test set

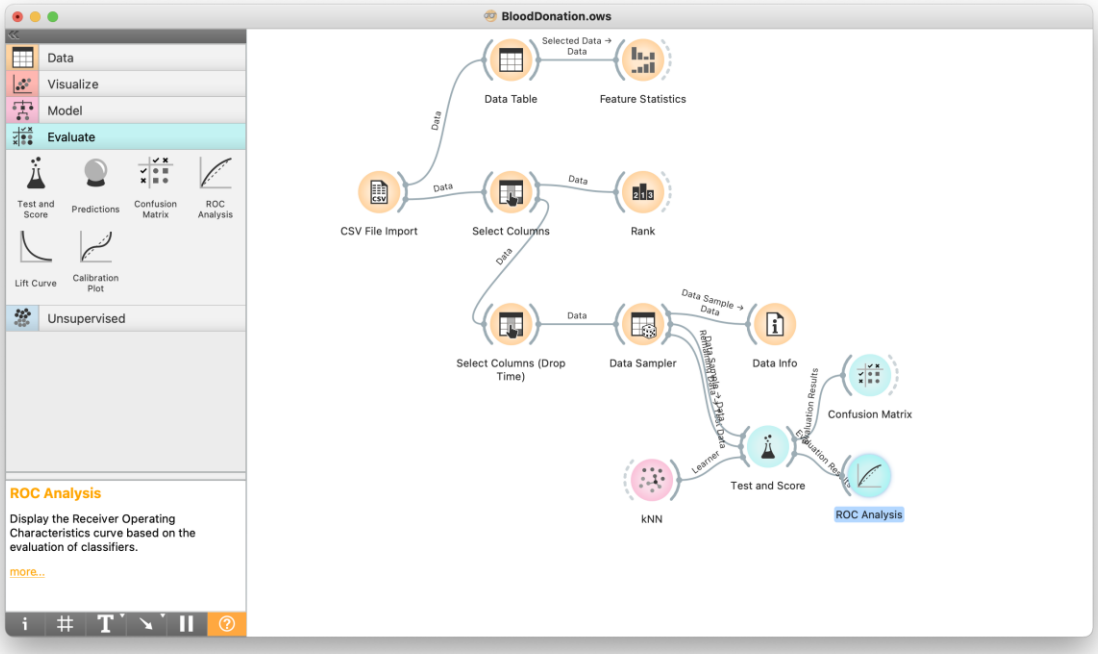
Use the Test and Score



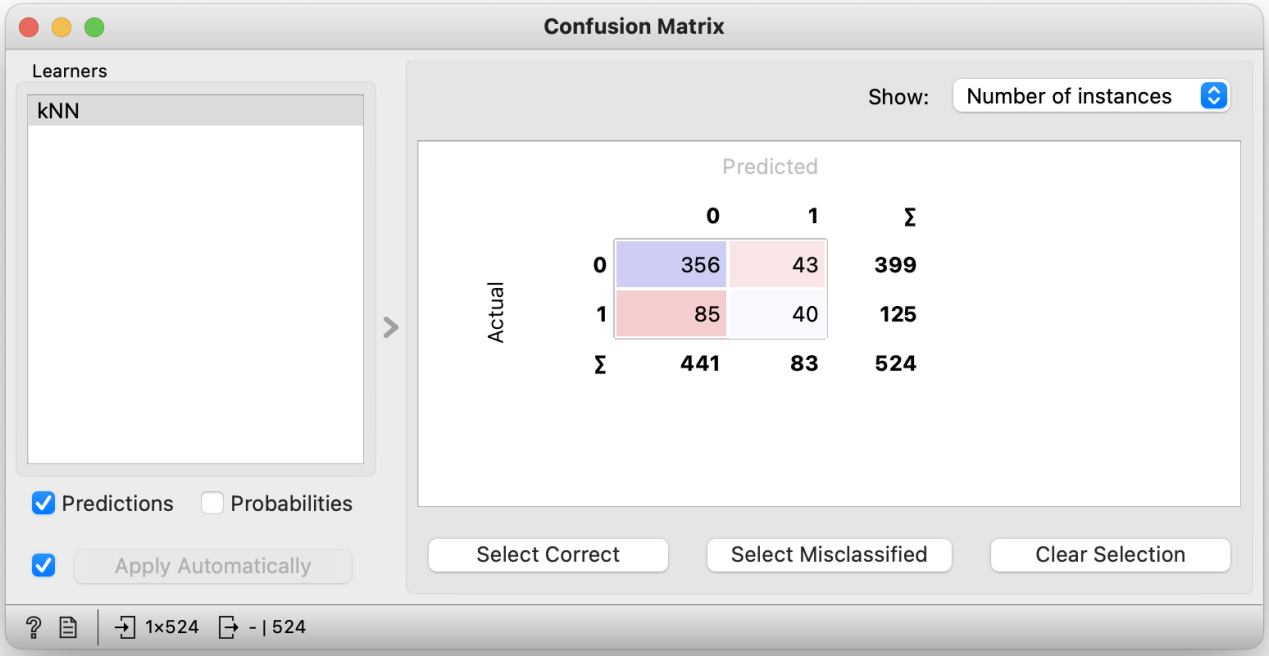

BloodDonation.ows
Data
Visualize
Model
Evaluate
Test and Score
Predictions
Confusion Matrix
ROC Analysis
Lift Curve
Calibration Plot
Unsupervised
Select a widget to show its description.
See workflow examples, YouTube tutorials, or open the welcome screen.
Selected Data → Data
Data Table
Feature Statistics
Data
CSV File Import
Select Columns
Rank
Select Columns (Drop Time)
Data Sampler
Data Info
Data Sample → Data
Remaining Data → Data
Learner
kNN
Test and Score
Edit the link


Edit Links
Data Sample
Remaining Data
Data
Test Data
Learner
Preprocessor
Data Sampler
Test and Score
Clear All
Cancel
OK

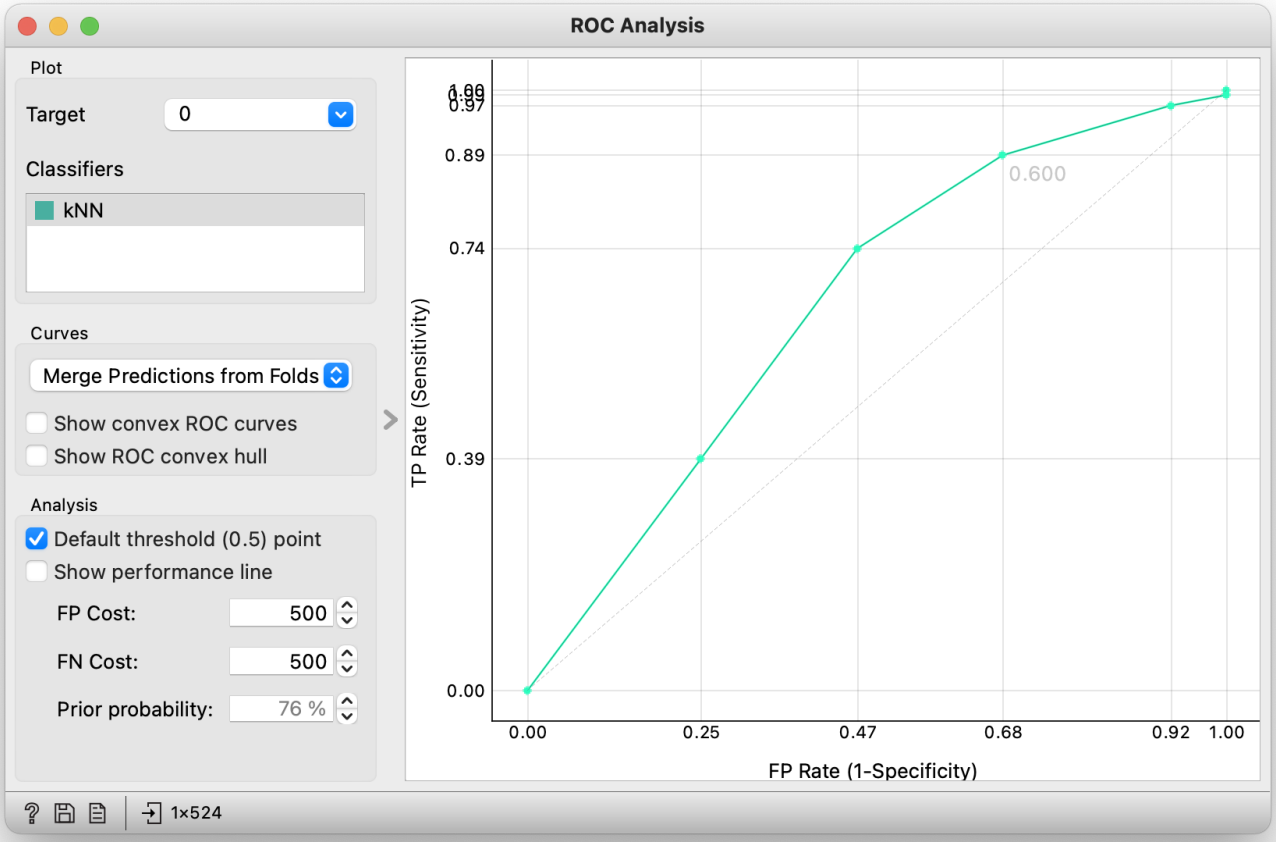
Use Confusion Matrix and ROC Analysis



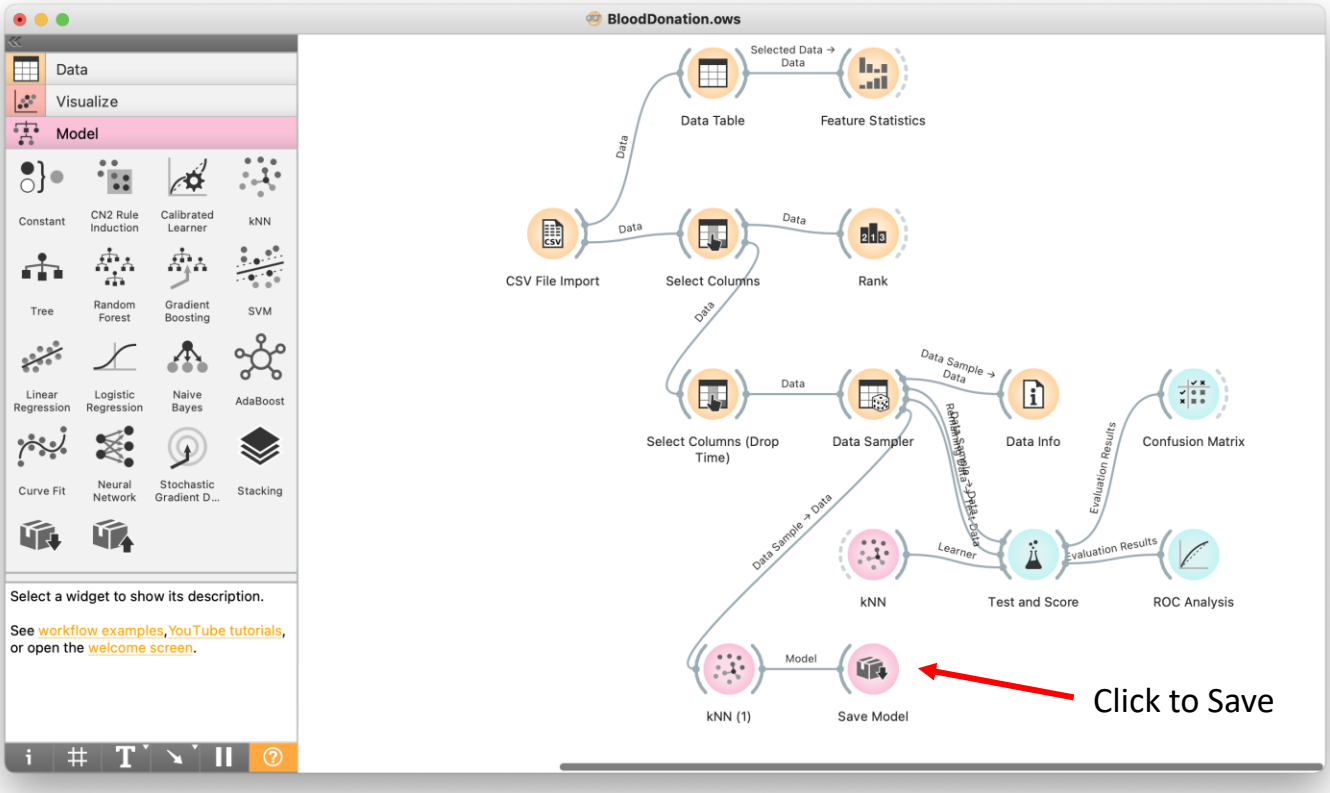
Confusion Matrix



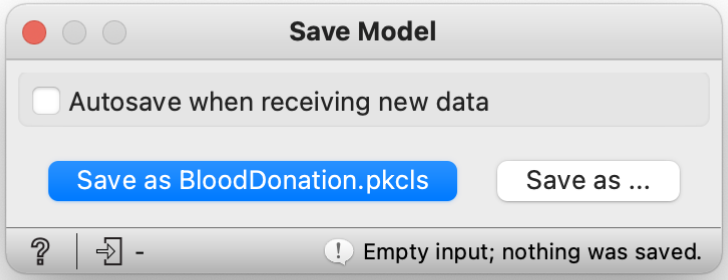
ROC


ROC Analysis
Plot
Target
0
Classifiers
kNN
Curves
Merge Predictions from Folds
Show convex ROC curves
Show ROC convex hull
Analysis
Default threshold (0.5) point
Show performance line
FP Cost:
500
FN Cost:
500
Prior probability:
76 %
1x524
TP Rate (Sensitivity)
1.00
0.97
0.89
0.74
0.39
0.00
0.600
0.00
0.25
0.47
0.68
0.92
1.00
FP Rate (1-Specificity)

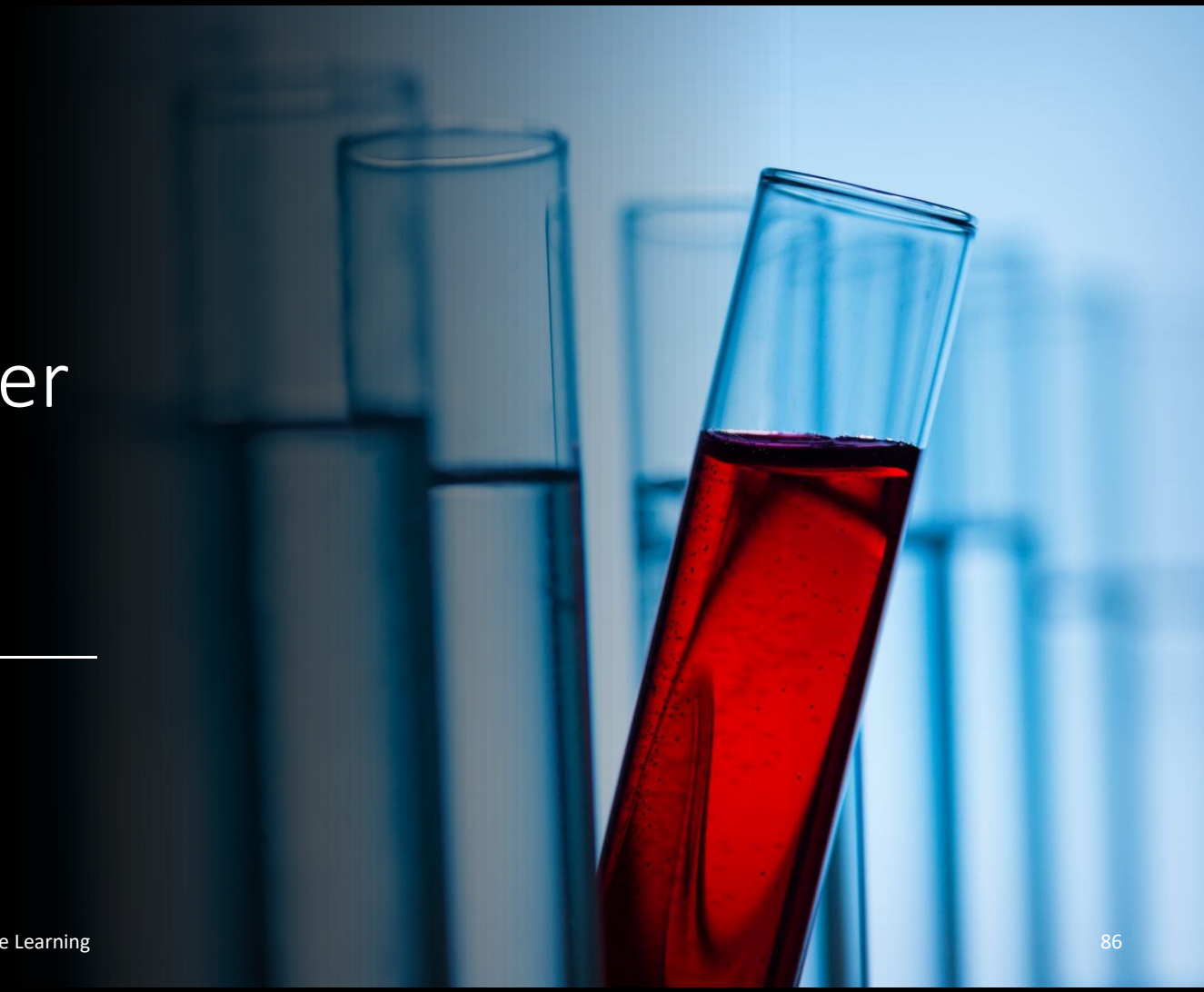
Deployment


BloodDonation.ows
Data
Visualize
Model
Constant
CN2 Rule Induction
Calibrated Learner
kNN
Tree
Random Forest
Gradient Boosting
SVM
Linear Regression
Logistic Regression
Naive Bayes
AdaBoost
Curve Fit
Neural Network
Stochastic Gradient D...
Stacking
Select a widget to show its description.
See workflow examples, YouTube tutorials, or open the welcome screen.
Selected Data → Data
Data Table
Feature Statistics
Data
CSV File Import
Select Columns
Rank
Select Columns (Drop Time)
Data Sampler
Data Info
Confusion Matrix
Data Sample → Data
Remaining Data → Data
Evaluation Results
Learner
kNN
Test and Score
ROC Analysis
Model
kNN (1)
Save Model
Click to Save

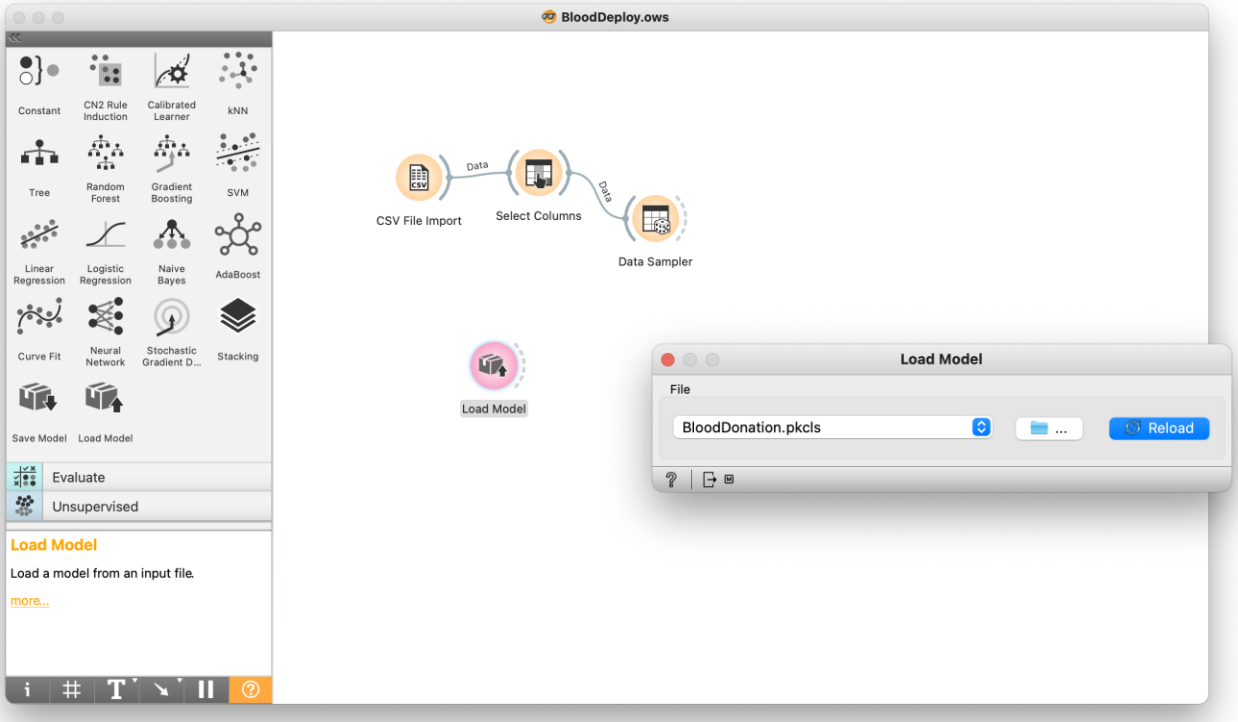
Save model


Save Model
Autosave when receiving new data
Save as BloodDonation.pkcls
Save as ...
-
Empty input; nothing was saved.

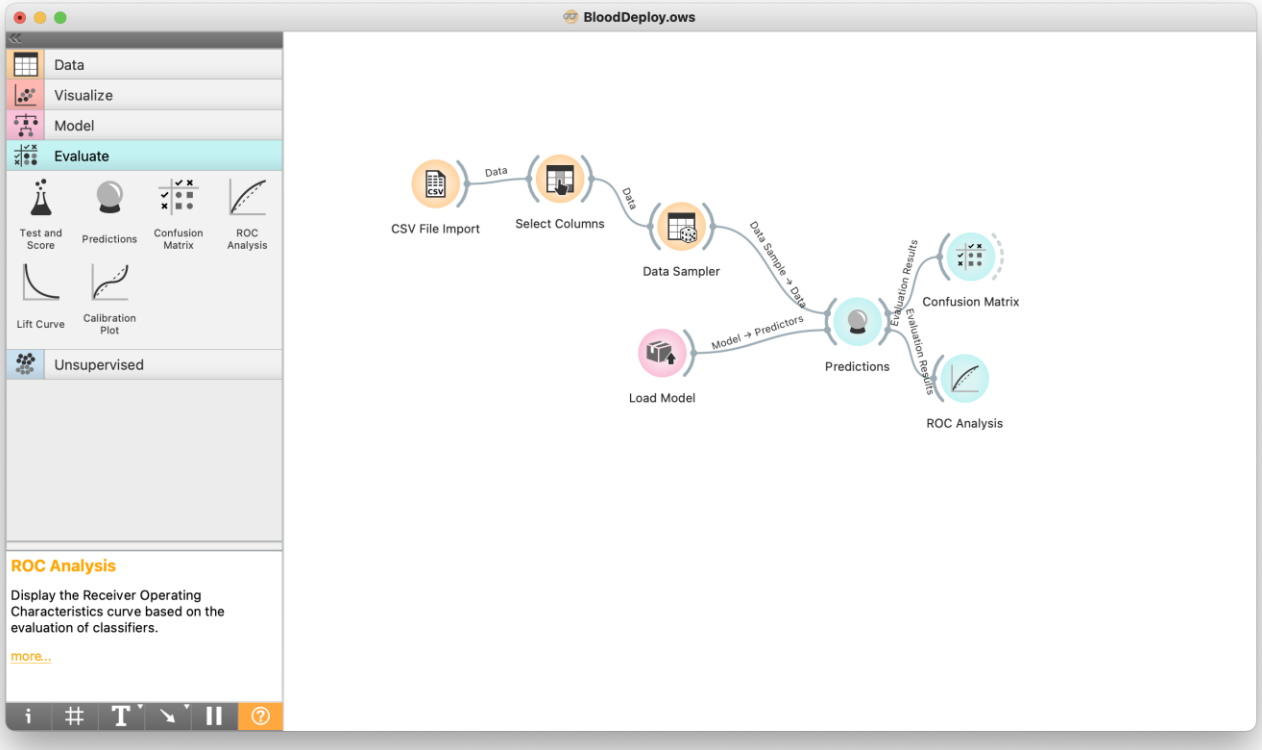
Create another BloodDeploy workflow

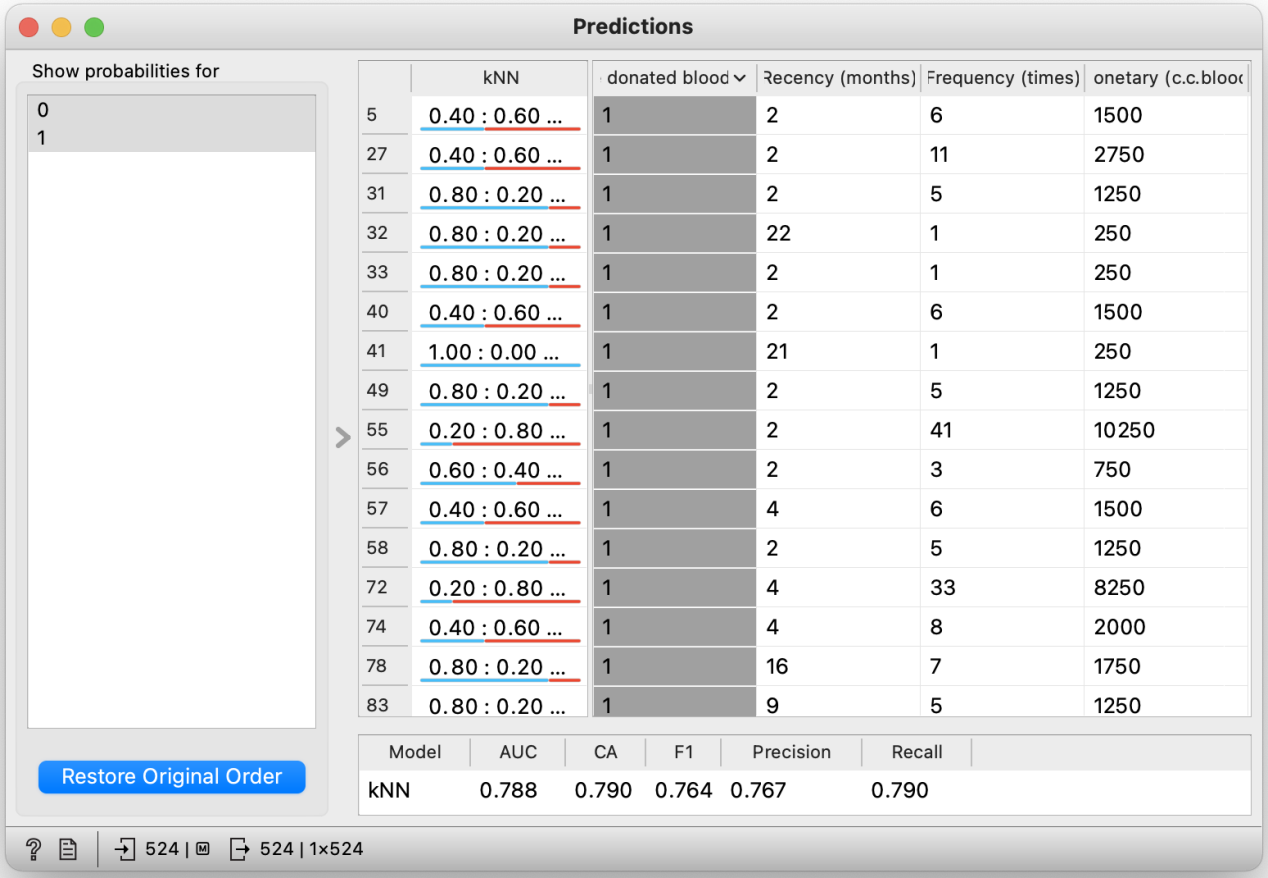


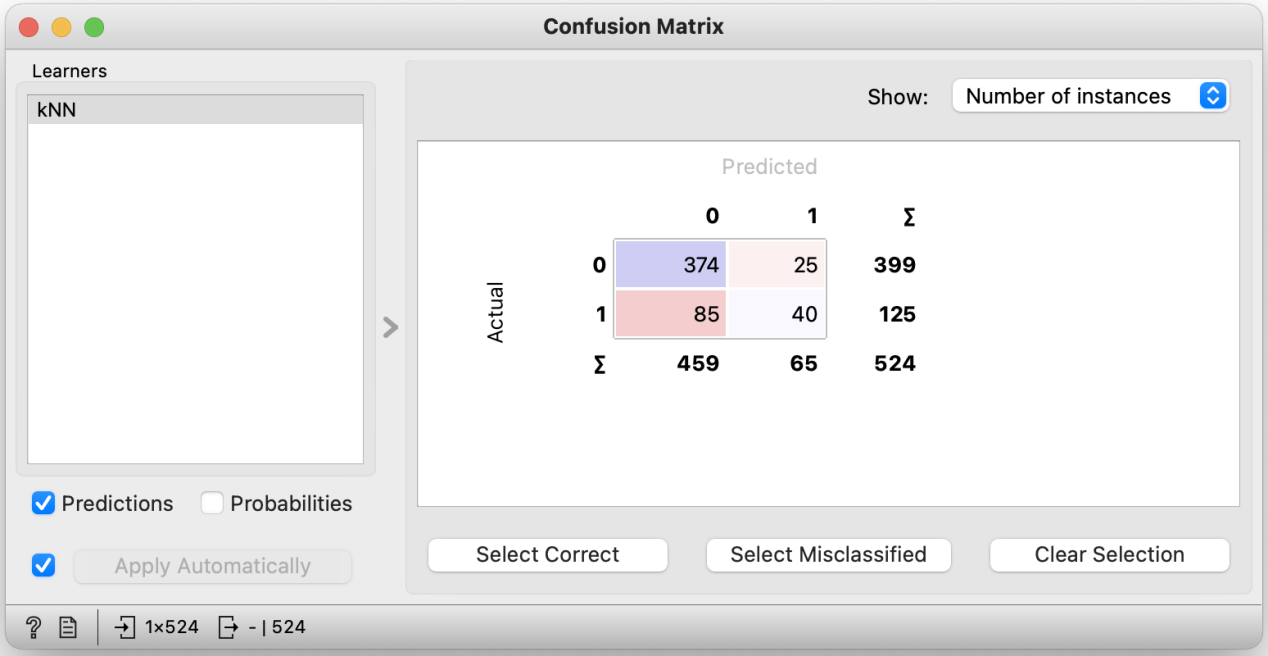
Use Load Model

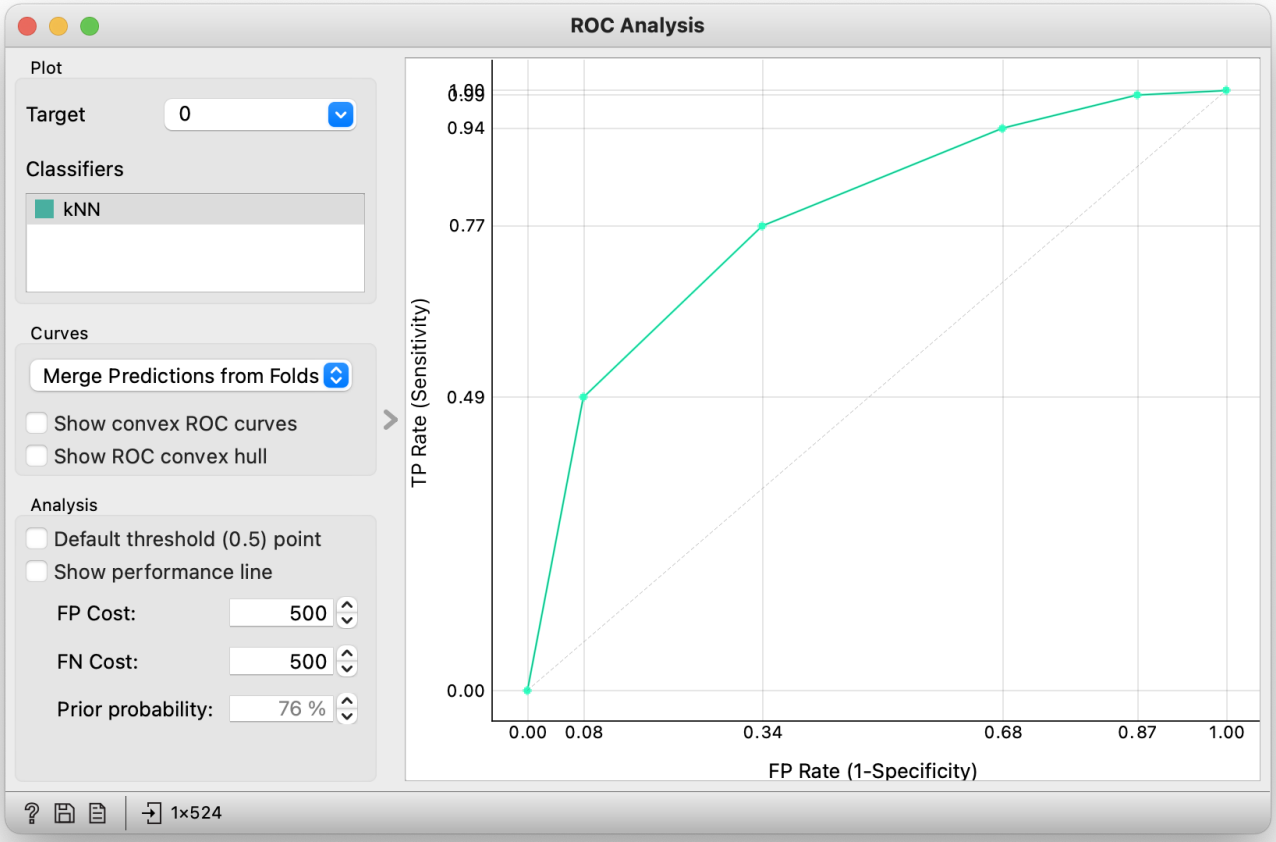


Use Predictions




Predictions
Show probabilities for
0
1
Restore Original Order
kNN
donated blood
Recency (months)
Frequency (times)
onetary (c.c.blooc
5 0.40 : 0.60 ... 1 2 6 1500
27 0.40 : 0.60 ... 1 2 11 2750
31 0.80 : 0.20 ... 1 2 5 1250
32 0.80 : 0.20 ... 1 22 1 250
33 0.80 : 0.20 ... 1 2 1 250
40 0.40 : 0.60 ... 1 2 6 1500
41 1.00 : 0.00 ... 1 21 1 250
49 0.80 : 0.20 ... 1 2 5 1250
55 0.20 : 0.80 ... 1 2 41 10250
56 0.60 : 0.40 ... 1 2 3 750
57 0.40 : 0.60 ... 1 4 6 1500
58 0.80 : 0.20 ... 1 2 5 1250
72 0.20 : 0.80 ... 1 4 33 8250
74 0.40 : 0.60 ... 1 4 8 2000
78 0.80 : 0.20 ... 1 16 7 1750
83 0.80 : 0.20 ... 1 9 5 1250
Model AUC CA F1 Precision Recall
kNN 0.788 0.790 0.764 0.767 0.790
524 | 524 | 1×524


Confusion Matrix
Learners
kNN
Show:
Number of instances
Predicted
0
1
Σ
Actual
0
374
25
399
1
85
40
125
Σ
459
65
524
Predictions
Probabilities
Apply Automatically
Select Correct
Select Misclassified
Clear Selection
1×524
- | 524


ROC Analysis
Plot
Target
0
Classifiers
kNN
Curves
Merge Predictions from Folds
Show convex ROC curves
Show ROC convex hull
Analysis
Default threshold (0.5) point
Show performance line
FP Cost:
500
FN Cost:
500
Prior probability:
76 %
1x524
TP Rate (Sensitivity)
1.00
0.99
0.94
0.77
0.49
0.00
0.00
0.08
0.34
0.68
0.87
1.00
FP Rate (1-Specificity)

Summary

We have learnt that:

- Orange Data Mining allows us to familiarize with the ML workflow and also quickly prototype some possible approaches for our modelling
- Data may need to be transformed/encoded before it can be fed to a machine algorithm
- Categorical variables can be encoded as dummy variables
- Sometimes it is useful to discretize numeric values
- Feature engineering is the part that needs real intelligence