Hes·so VALAIS WALLIS
Institute of Information Systems

How to Detect soft falls on ANDROID devices

some signal processing with KNIME

Prof. Dr. Dominique Genoud
Institute of Information Systems
Dominique.Genoud@hevs.ch
Techno-Pôle 3 – CH-3960 Sierre
Switzerland

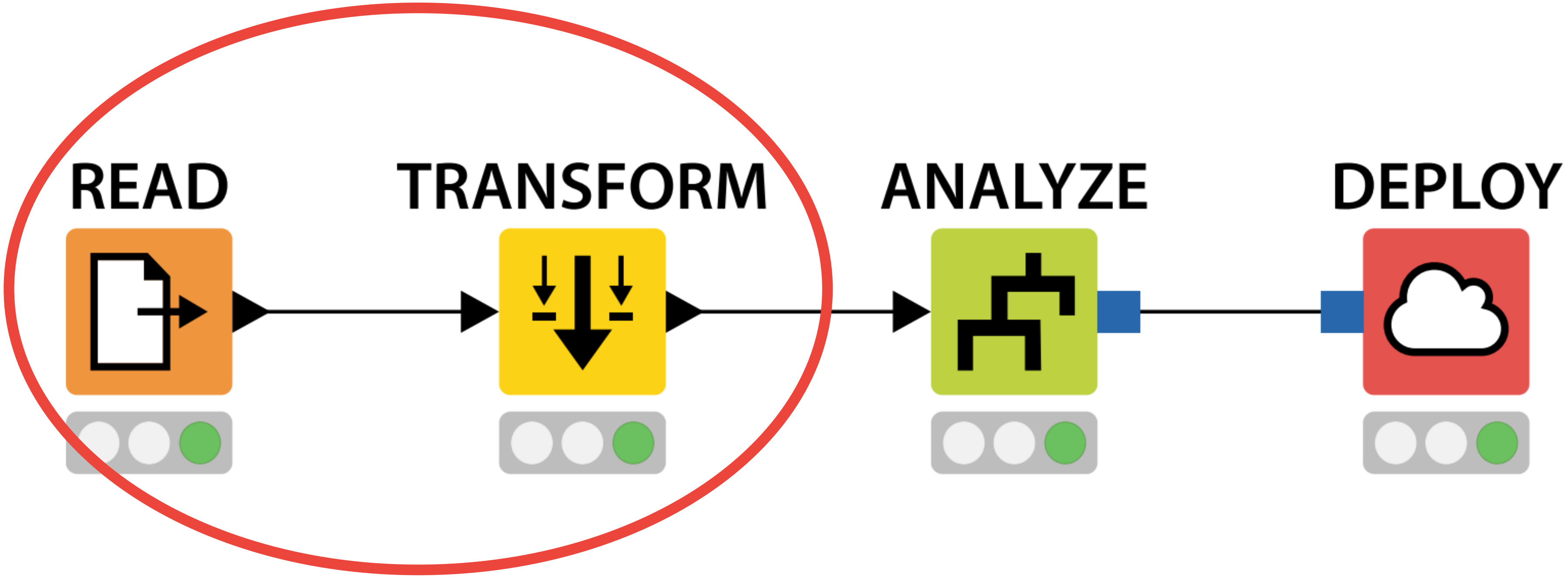
Vincent Cuendet master HES-SO,
Lausanne, Switzerland

Julien Torrent FST,
Neuchâtel,Switzerland

datastory

KNIME Spring Summit 2016 – Berlin

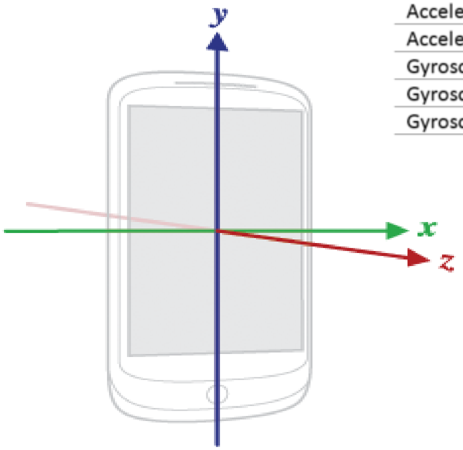
Processing flow



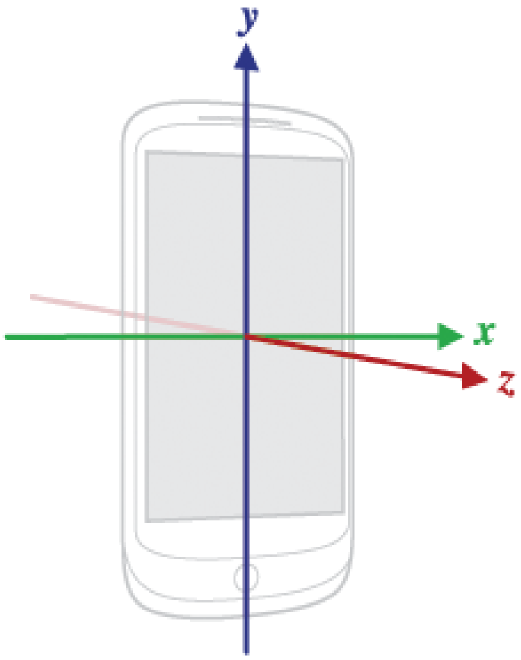
Connected android watches

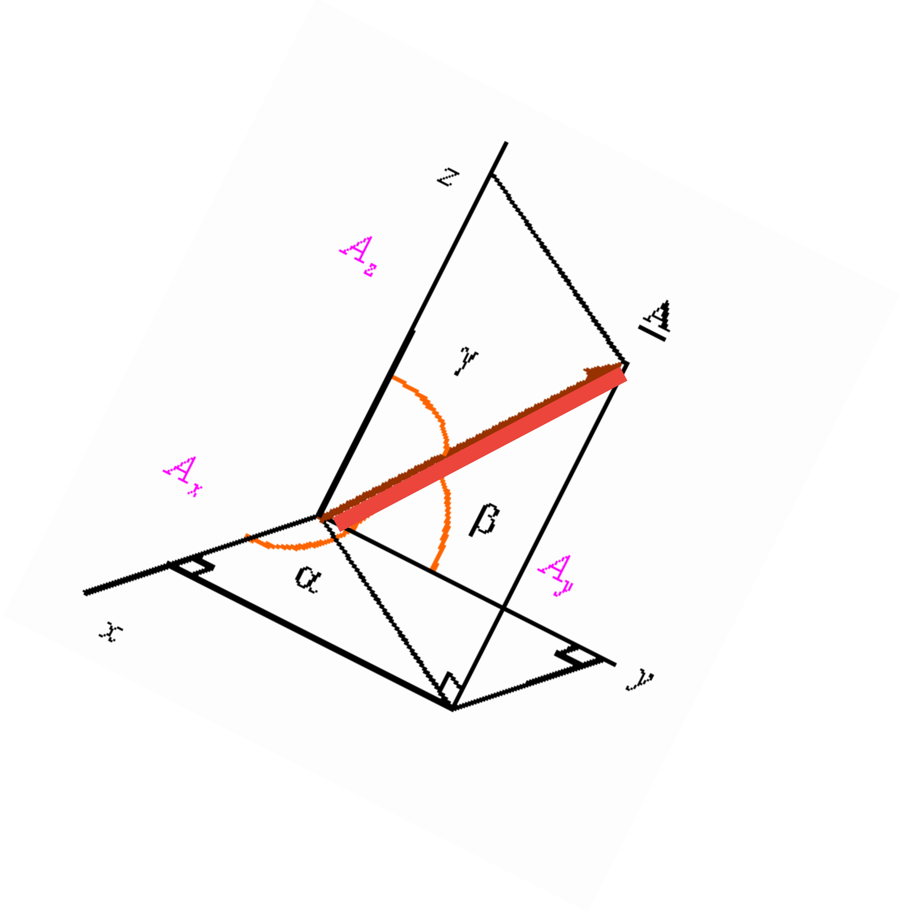
Moto 360 and LG-G watches

Name	Format	Possible values	Type
Fall type	Alphanumeric, 2 symbols	B1, B2, B3, M1, M2, M3, NO, FR	Category
Subject	Alphabetic, 2 characters	AA, BA, BO, FA, GU, KU, LA ; NI, PI, TO, TR, UN, VA	Category
Age	Numeric, 2 digits	[22, 93]	Continuous
Sex	Alphabetic, 1 character	F, M	Category
Auxiliary mean	Alphanumeric, 2 symbols	00, CA, DR, DS	Category
Linear acceleration, X axis.	Numeric	[-34.86532974243164, 33.24461364746094]	Continuous
Linear acceleration, Y axis.	Numeric	[-44.05729675292969, 42.206565856933594]	Continuous
Linear acceleration, Z axis.	Numeric	[-29.68331527709961, 35.54317855834961]	Continuous
Acceleration, X axis	Numeric	[0,0]	Continuous
Acceleration, Y axis	Numeric	[-39.08319854736328, 39.369998931884766]	Continuous
Acceleration, X axis	Numeric	[-38.91899871826172, 39.82899856567383]	Continuous
Gyroscope, X axis	Numeric	[-38.91109848022461, 39.54209899902344]	Continuous
Gyroscope, Y axis	Numeric	[-39.08319854736328, 39.369998931884766]	Continuous
Gyroscope, Z axis	Numeric	[-13.002599716186523, 14.506699562072754]	Continuous

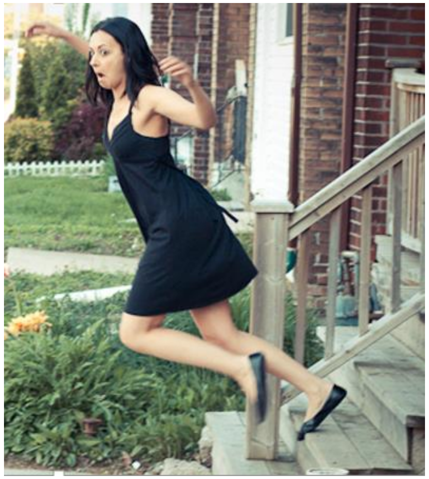


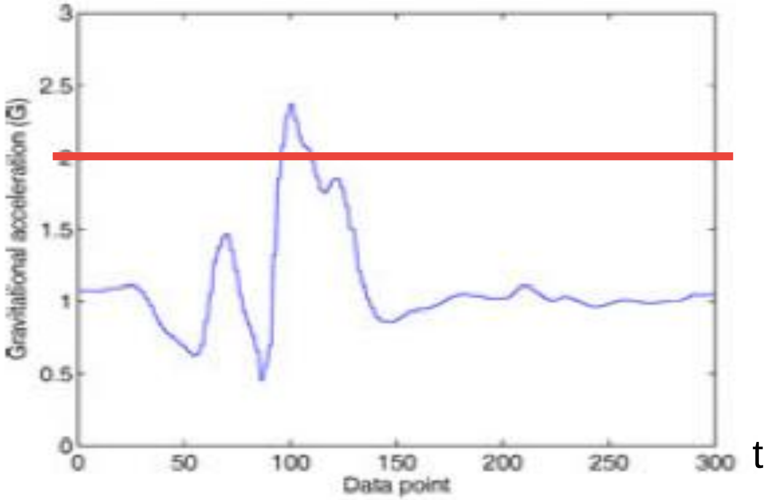
Use the magnitude of 3D vector





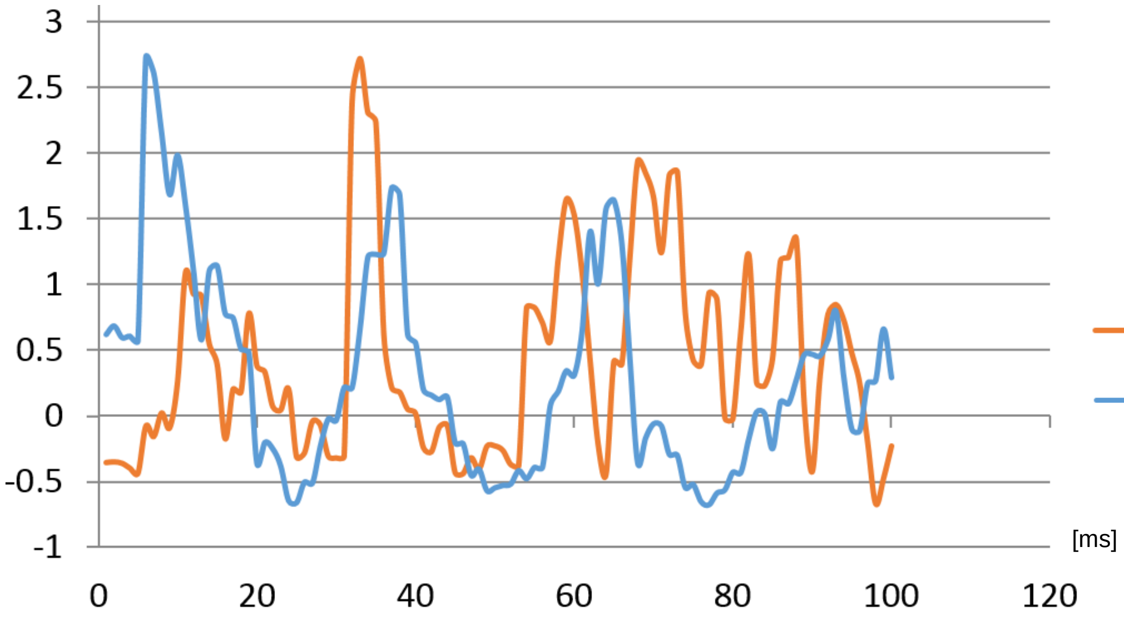
Falls





About 98% detection

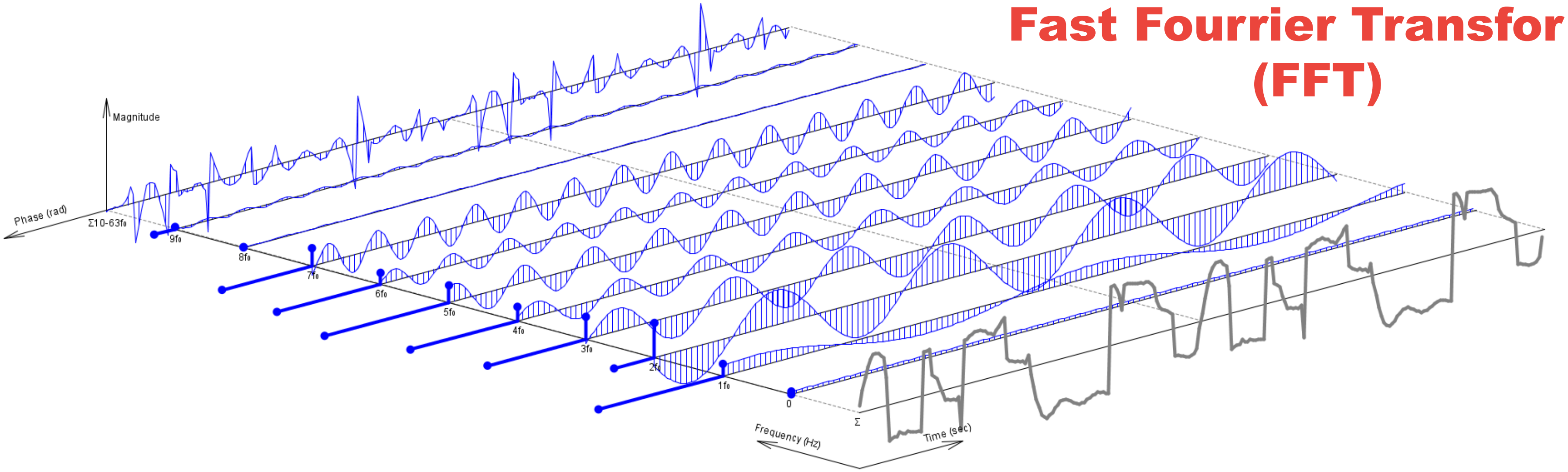
Soft falls



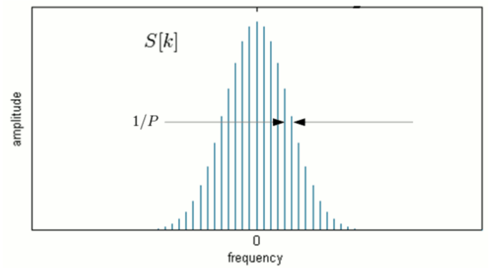
1% detection with thresholds

Signal processing to find patterns

- Decomposition of a temporal signal:



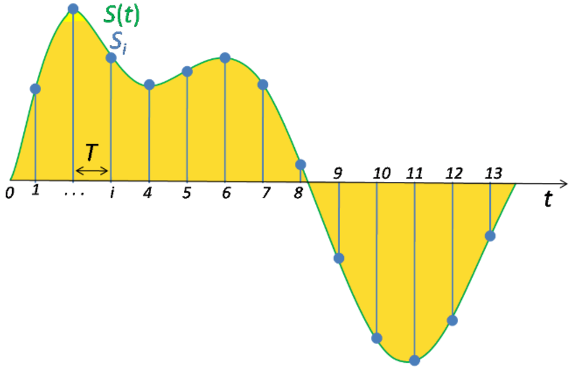
Fast Fourrier Transform (FFT)



In the digital word :

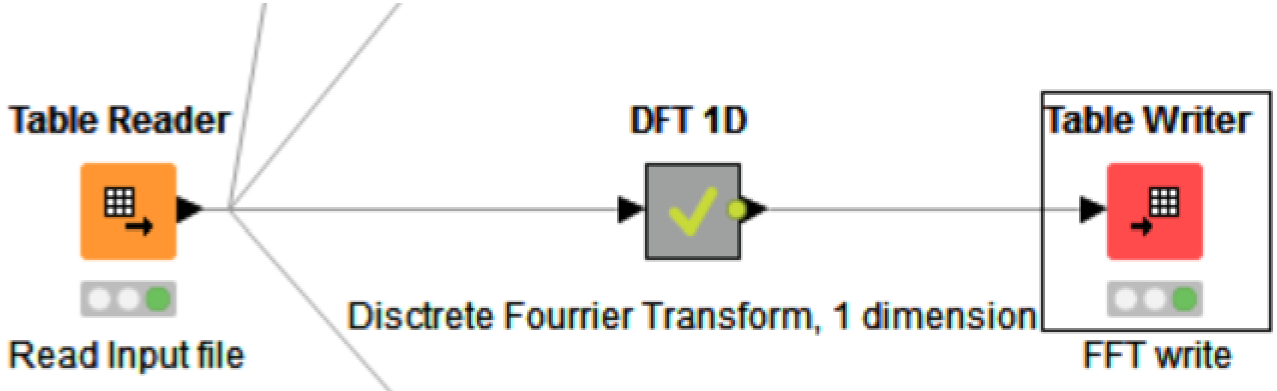
Discrete Fourrier Transform (DFT)

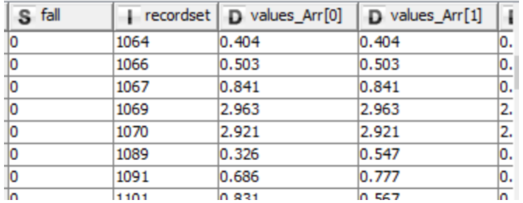
Knime to perform DFT



Reading the samples, and computing the magnitude vector

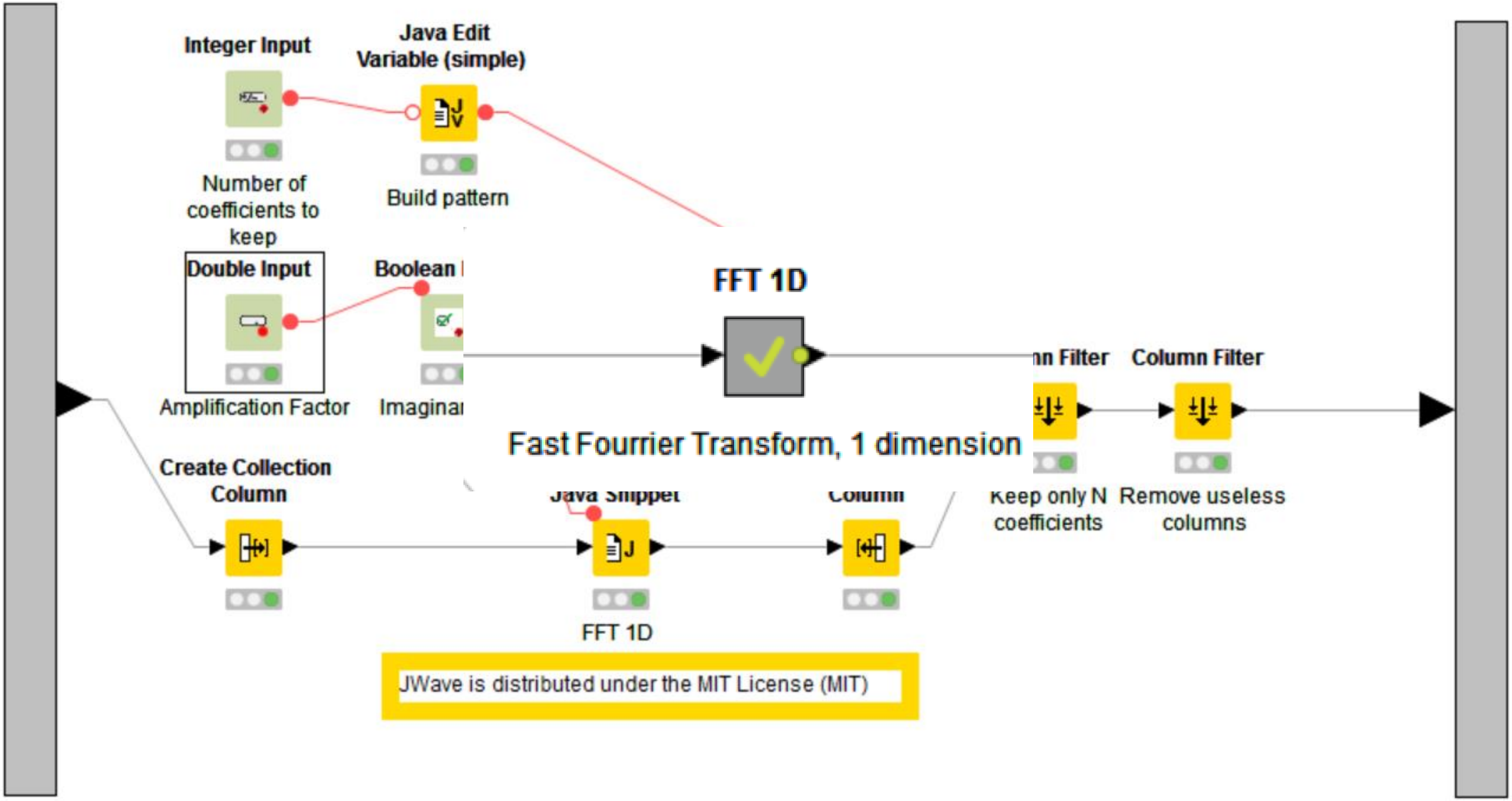
T = 20 ms (50Hz) sampling rate in a window of 128 samples (2.56 seconds)





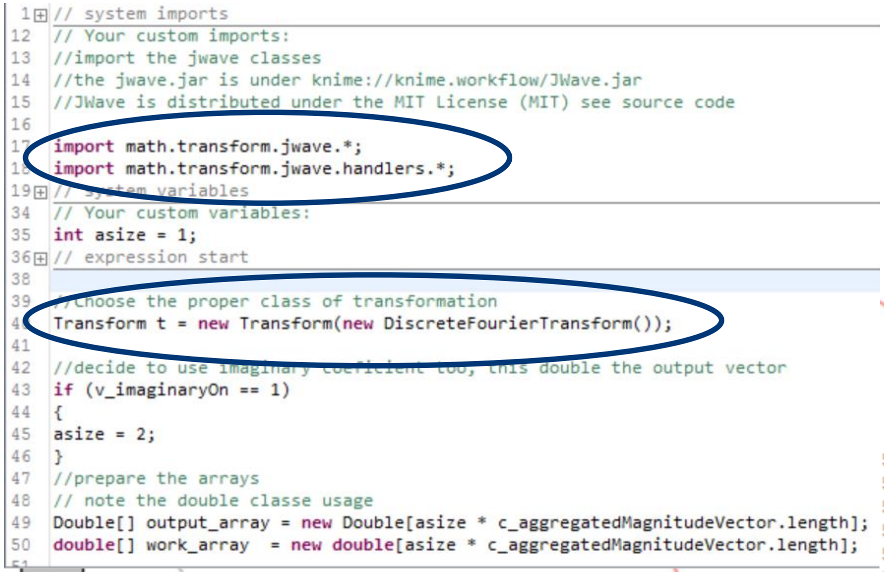
S fall	I recordset	D values_Arr[0]	D values_Arr[1]	I
0	1064	0.404	0.404	0.
0	1066	0.503	0.503	0.
0	1067	0.841	0.841	0.
0	1069	2.963	2.963	2.
0	1070	2.921	2.921	2.
0	1089	0.326	0.547	0.
0	1091	0.686	0.777	0.
0	1101	0.831	0.567	0.

Knime to perform DFT

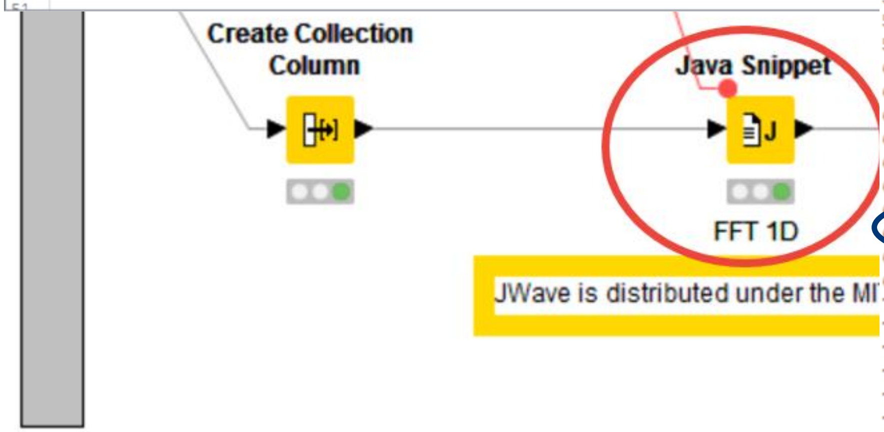


JWave is distributed under the MIT License (MIT)

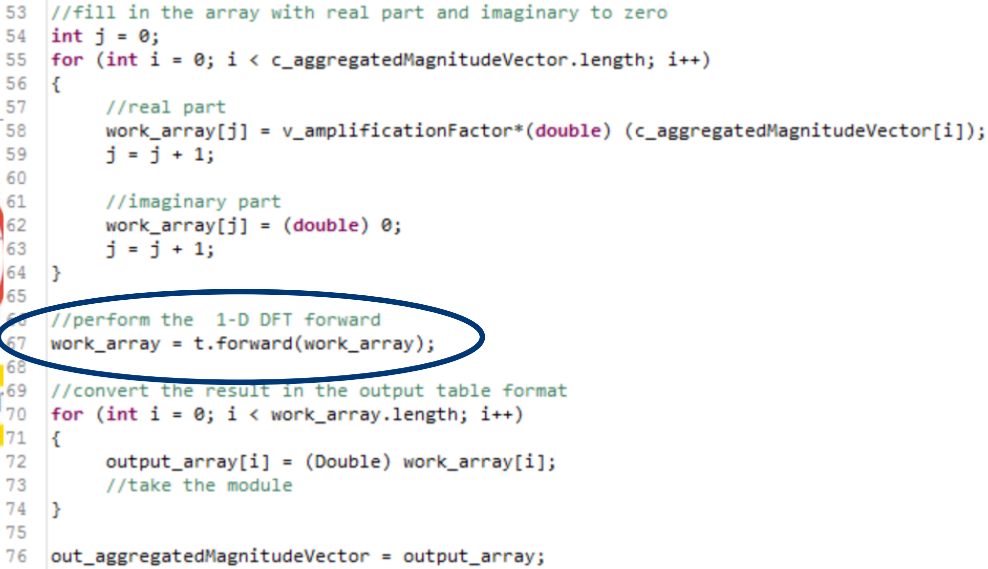
Knime to perform DFT



```
// system imports
// Your custom imports:
//import the jwave classes
//the jwave.jar is under knime://knime.workflow/JWave.jar
//JWave is distributed under the MIT License (MIT) see source code

import math.transform.jwave.*;
import math.transform.jwave.handlers.*;
// system variables
// Your custom variables:
int asize = 1;
// expression start

//choose the proper class of transformation
Transform t = new Transform(new DiscreteFourierTransform());

//decide to use imaginary coeficient too, this double the output vector
if (v_imaginaryOn == 1)
{
asize = 2;
}
//prepare the arrays
// note the double classe usage
Double[] output_array = new Double[asize * c_aggregatedMagnitudeVector.length];
double[] work_array  = new double[asize * c_aggregatedMagnitudeVector.length];
```



JWave is distributed under the MI



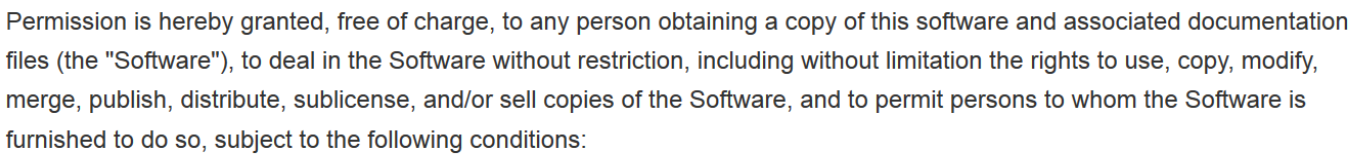
```
//fill in the array with real part and imaginary to zero
int j = 0;
for (int i = 0; i < c_aggregatedMagnitudeVector.length; i++)
{
     //real part
     work_array[j] = v_amplificationFactor*(double) (c_aggregatedMagnitudeVector[i]);
     j = j + 1;

     //imaginary part
     work_array[j] = (double) 0;
     j = j + 1;
}

//perform the  1-D DFT forward
work_array = t.forward(work_array);

//convert the result in the output table format
for (int i = 0; i < work_array.length; i++)
{
     output_array[i] = (Double) work_array[i];
     //take the module
}

out_aggregatedMagnitudeVector = output_array;
```

KNIME Spring Summit 2016 – Berlin

Institute of Information Systems

Knime to perform DFT

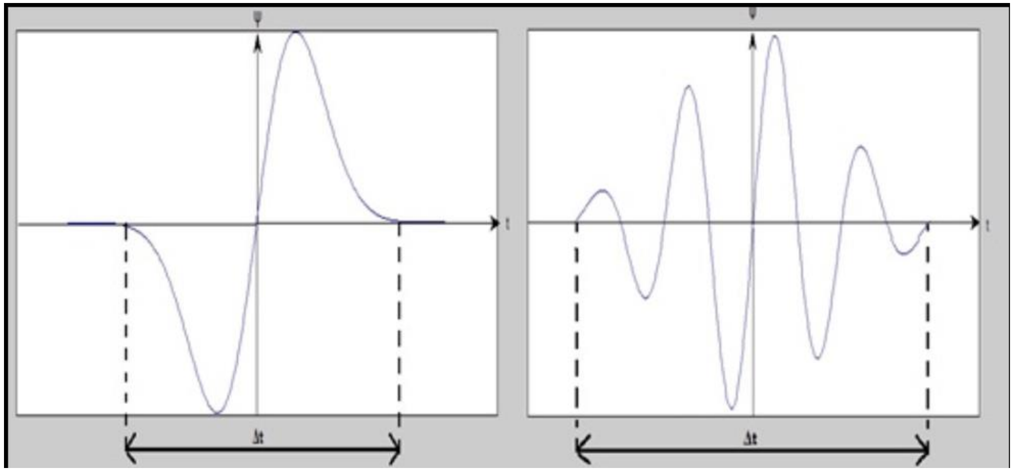
- Jwave.jar:
  - open source library developed by C. Scheiblich (MIT)
- The source code and examples are available there:
  - https://github.com/cscheiblich/JWave
- License free :

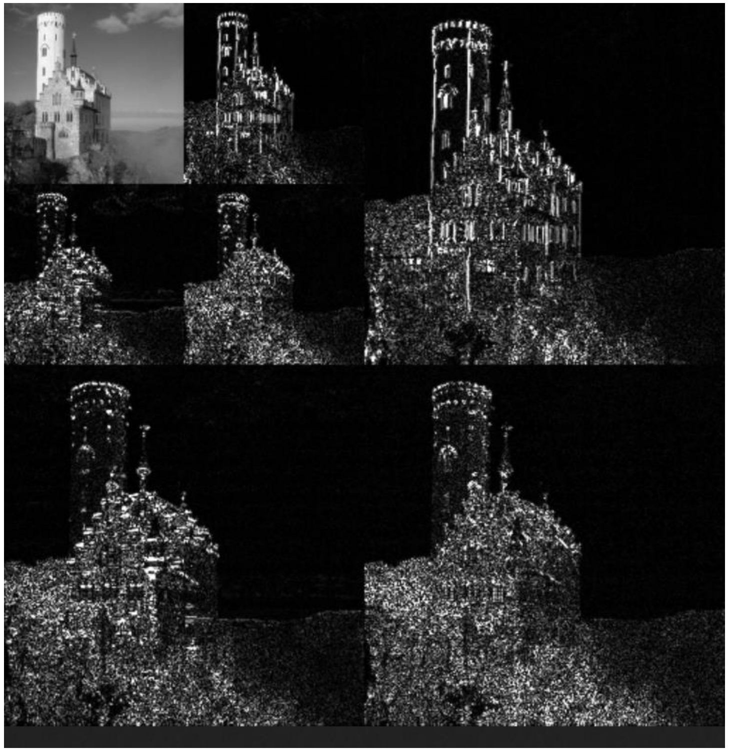


Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

Other polular pattern detection

- Wavelet transform : https://en.wikipedia.org/wiki/Wavelet_transform

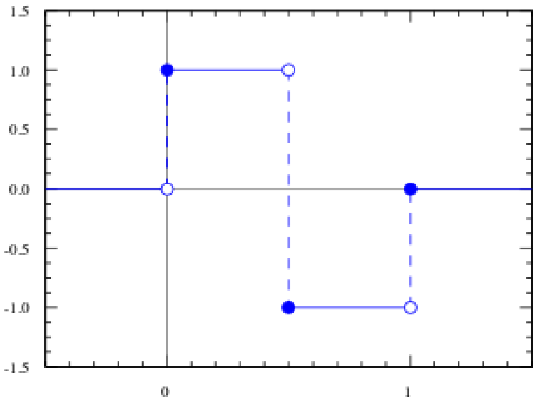




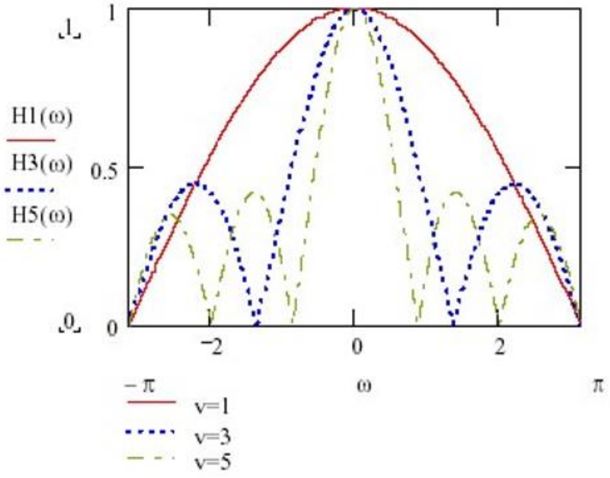
Knime to perform Wavelets

- Implemented in the jwave.jar library:

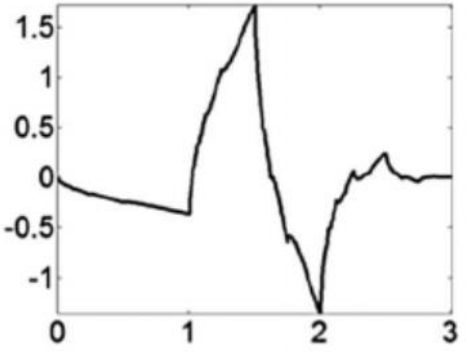
Haare



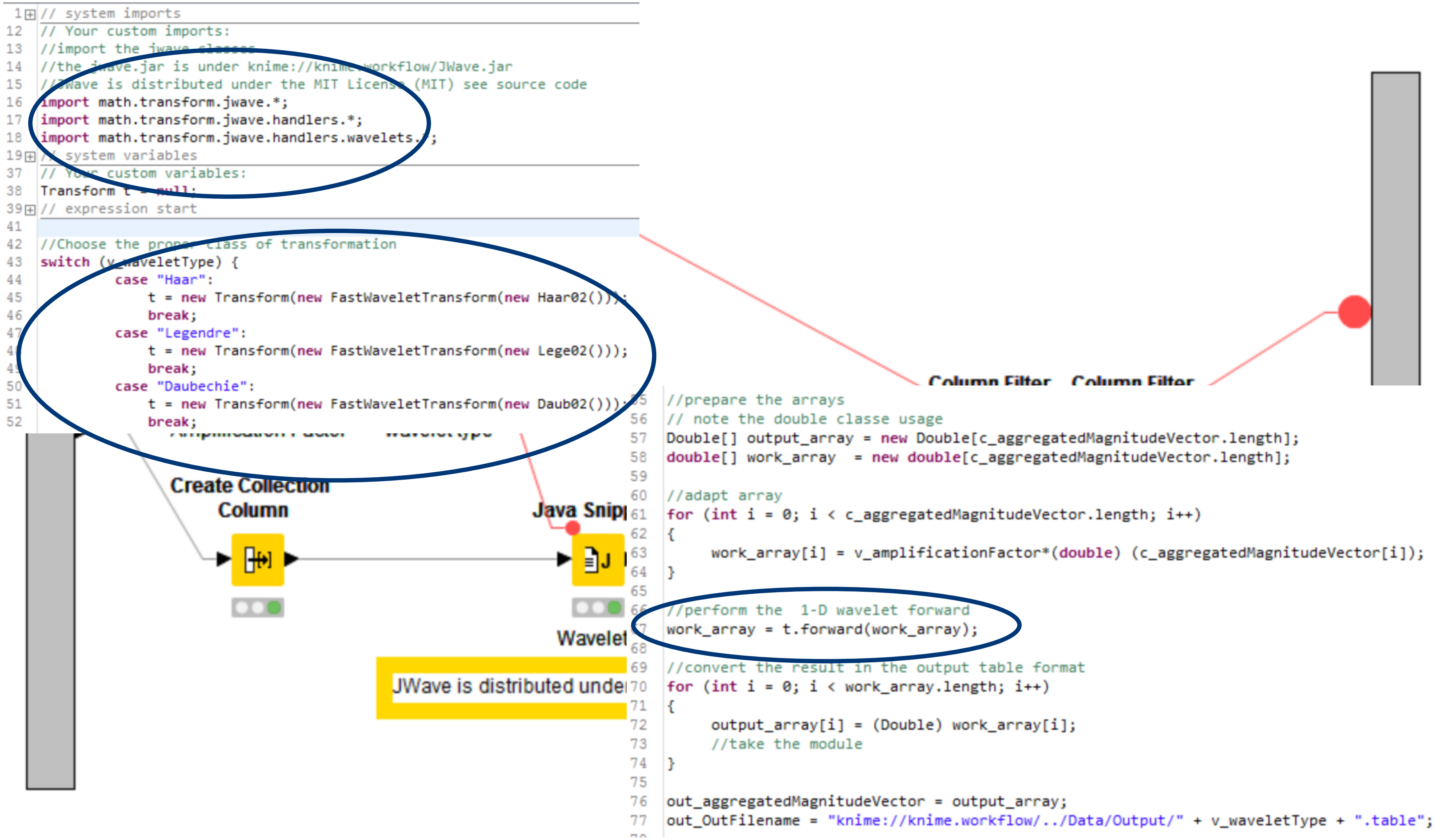
Legendre



Daubechie (db2)



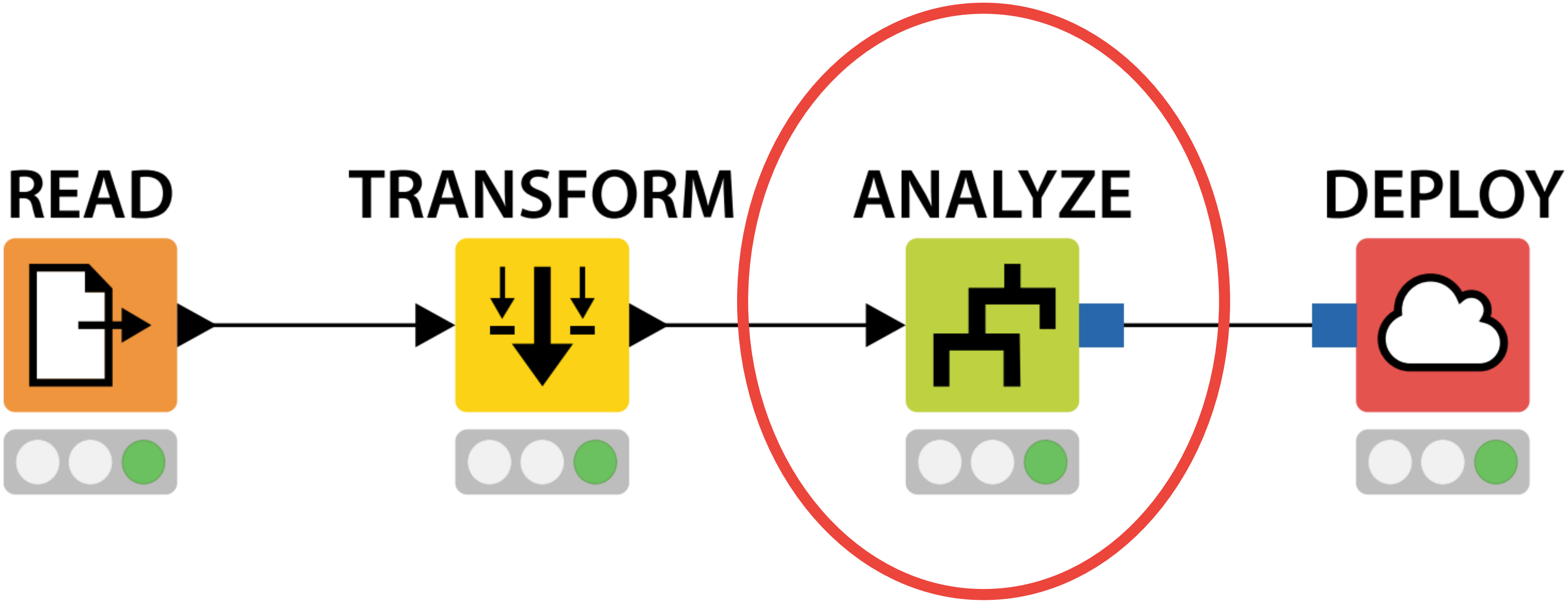
Knime to perform Wavelets



```
// system imports
// Your custom imports:
//import the jwave classes
//the jwave.jar is under knime://knime.workflow/JWave.jar
//JWave is distributed under the MIT License (MIT) see source code
import math.transform.jwave.*;
import math.transform.jwave.handlers.*;
import math.transform.jwave.handlers.wavelets.*;
// system variables
// Your custom variables:
Transform t = null;
// expression start

//Choose the proper class of transformation
switch (v_waveletType) {
        case "Haar":
            t = new Transform(new FastWaveletTransform(new Haar02()));
            break;
        case "Legendre":
            t = new Transform(new FastWaveletTransform(new Lege02()));
            break;
        case "Daubechie":
            t = new Transform(new FastWaveletTransform(new Daub02()));
            break;
```

```
//prepare the arrays
// note the double classe usage
Double[] output_array = new Double[c_aggregatedMagnitudeVector.length];
double[] work_array  = new double[c_aggregatedMagnitudeVector.length];

//adapt array
for (int i = 0; i < c_aggregatedMagnitudeVector.length; i++)
{
     work_array[i] = v_amplificationFactor*(double) (c_aggregatedMagnitudeVector[i]);
}

//perform the  1-D wavelet forward
work_array = t.forward(work_array);

//convert the result in the output table format
for (int i = 0; i < work_array.length; i++)
{
     output_array[i] = (Double) work_array[i];
     //take the module
}

out_aggregatedMagnitudeVector = output_array;
out_OutFilename = "knime://knime.workflow/../Data/Output/" + v_waveletType + ".table";
```

Processing flow



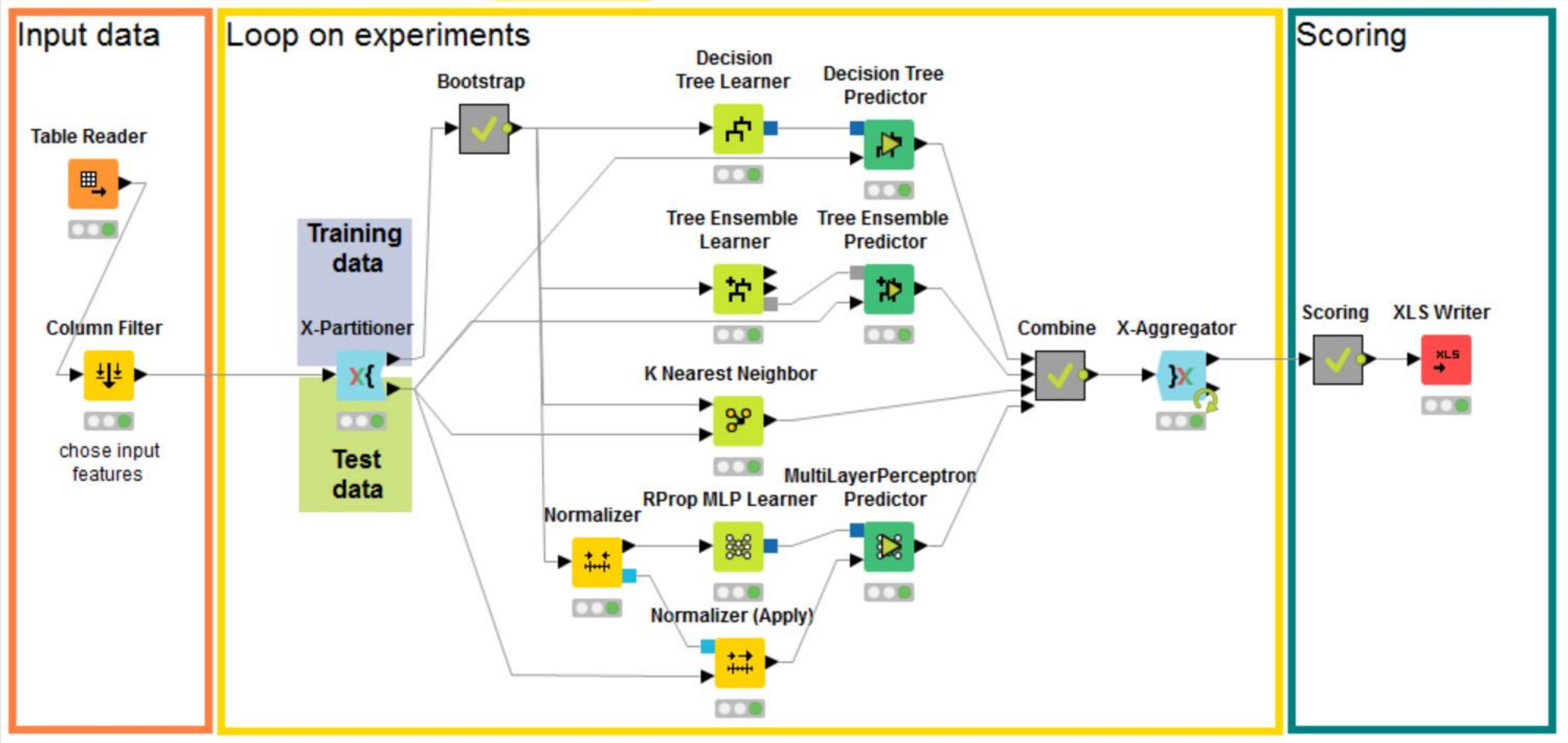
READ

TRANSFORM

ANALYZE

DEPLOY

Classification experiments



Results

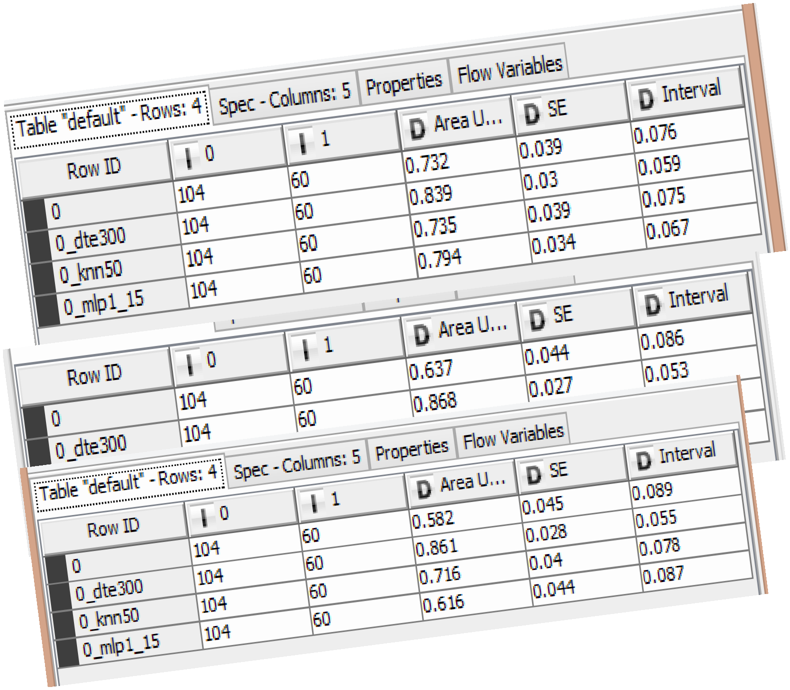


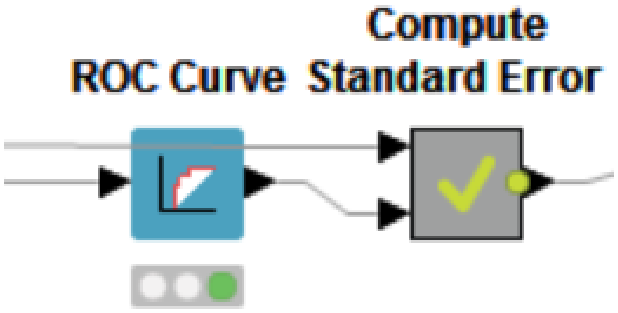
Table "default" - Rows: 4 | Spec - Columns: 5 | Properties | Flow Variables

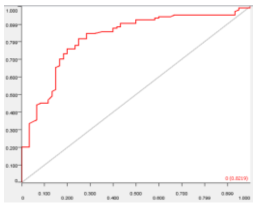
Row ID	I 0	I 1	D Area U...	D SE	D Interval
0	104	60	0.732	0.039	0.076
0_dte300	104	60	0.839	0.03	0.059
0_knn50	104	60	0.735	0.039	0.075
0_mlp1_15	104	60	0.794	0.034	0.067

Row ID	I 0	I 1	D Area U...	D SE	D Interval
0	104	60	0.637	0.044	0.086
0_dte300	104	60	0.868	0.027	0.053

Table "default" - Rows: 4 | Spec - Columns: 5 | Properties | Flow Variables

Row ID	I 0	I 1	D Area U...	D SE	D Interval
0	104	60	0.582	0.045	0.089
0_dte300	104	60	0.861	0.028	0.055
0_knn50	104	60	0.716	0.04	0.078
0_mlp1_15	104	60	0.616	0.044	0.087



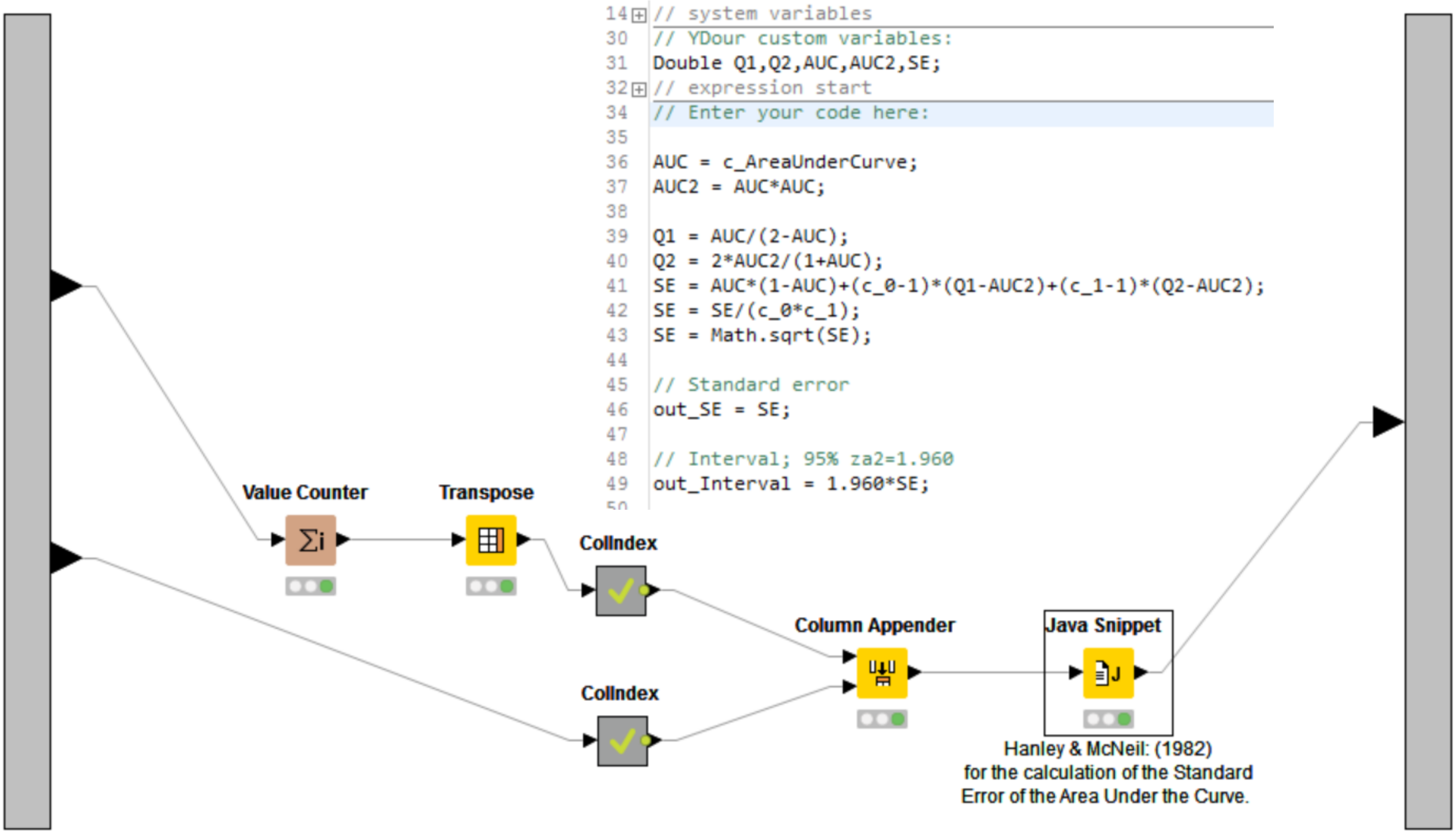


Preprocessing	Predictor	AUC	precision
FFT 8 coefficients	Decision Tree Ensemble	0.84	±0.03
	Mulltilayer Perceptron	0.79	±0.03
Wavelets Haare	Decision Tree Ensemble	0.87	±0.03
	K Nearest Neibourghs	0.75	±0.04
Daubechie 128 coefficients	Decision Tree Ensemble	0.86	±0.03
	K Nearest Neibourghs	0.73	±0.04

Full dataset: **500 soft falls and 1500 normal activities**

Stratified Subset of 20% of the original data

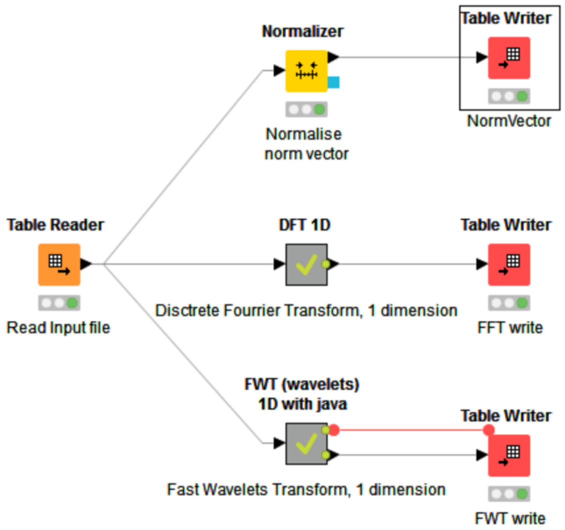
Scoring and precision of AUC

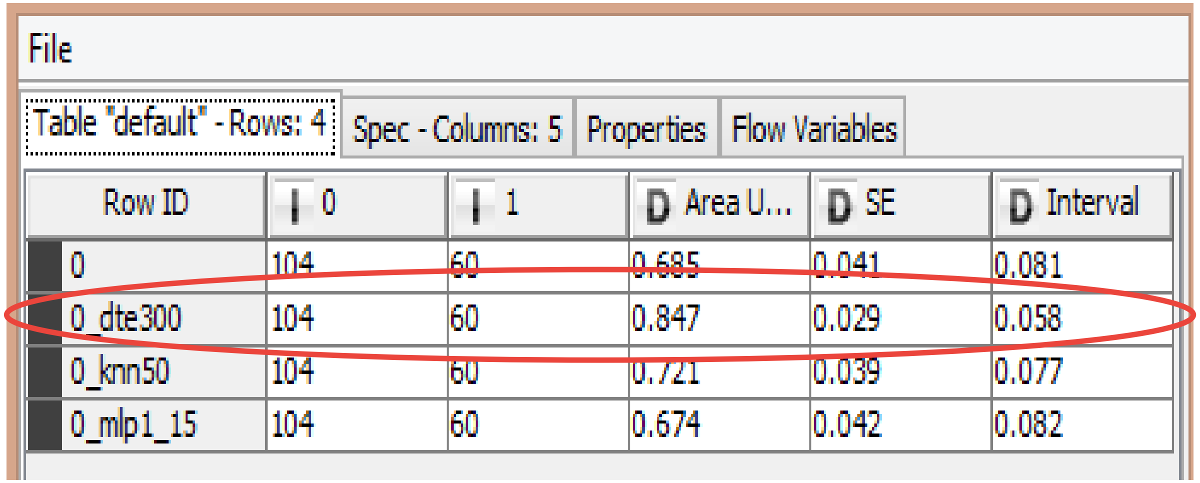


```
// system variables
// YDour custom variables:
Double Q1,Q2,AUC,AUC2,SE;
// expression start
// Enter your code here:

AUC = c_AreaUnderCurve;
AUC2 = AUC*AUC;

Q1 = AUC/(2-AUC);
Q2 = 2*AUC2/(1+AUC);
SE = AUC*(1-AUC)+(c_0-1)*(Q1-AUC2)+(c_1-1)*(Q2-AUC2);
SE = SE/(c_0*c_1);
SE = Math.sqrt(SE);

// Standard error
out_SE = SE;

// Interval; 95% za2=1.960
out_Interval = 1.960*SE;
```

Hanley & McNeil: (1982)
for the calculation of the Standard
Error of the Area Under the Curve.

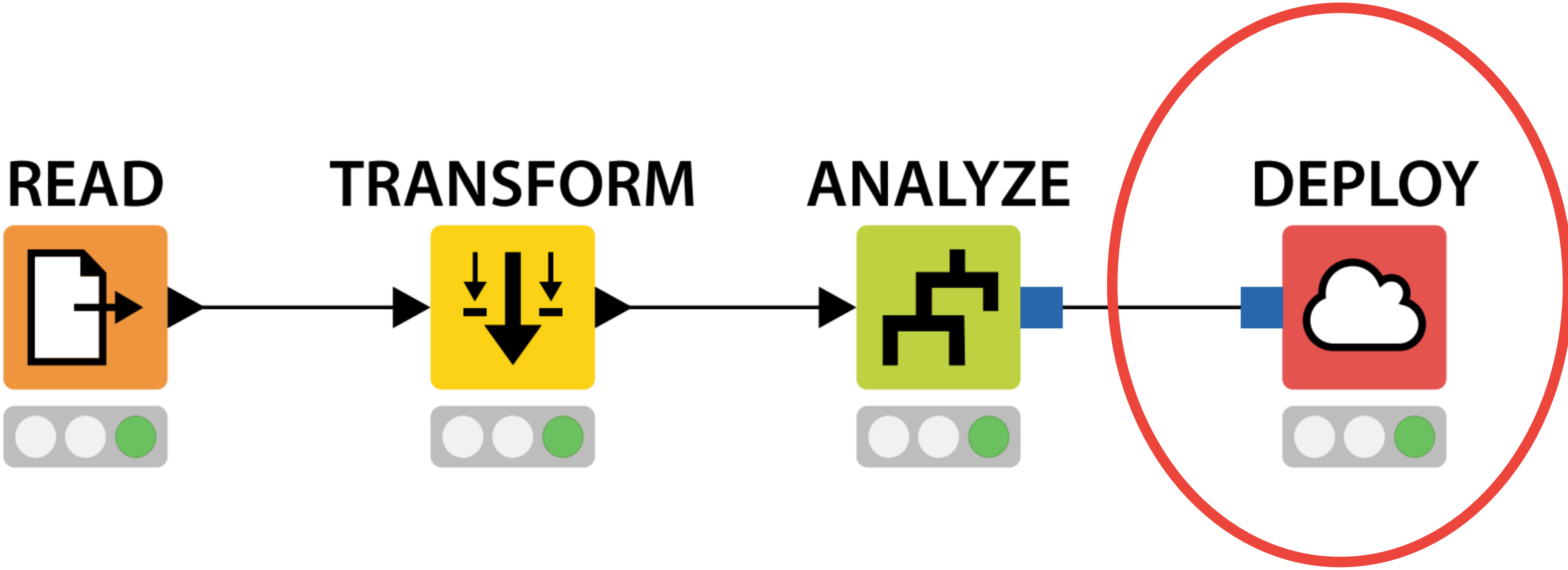
How good are the Decision Tree Ensemble





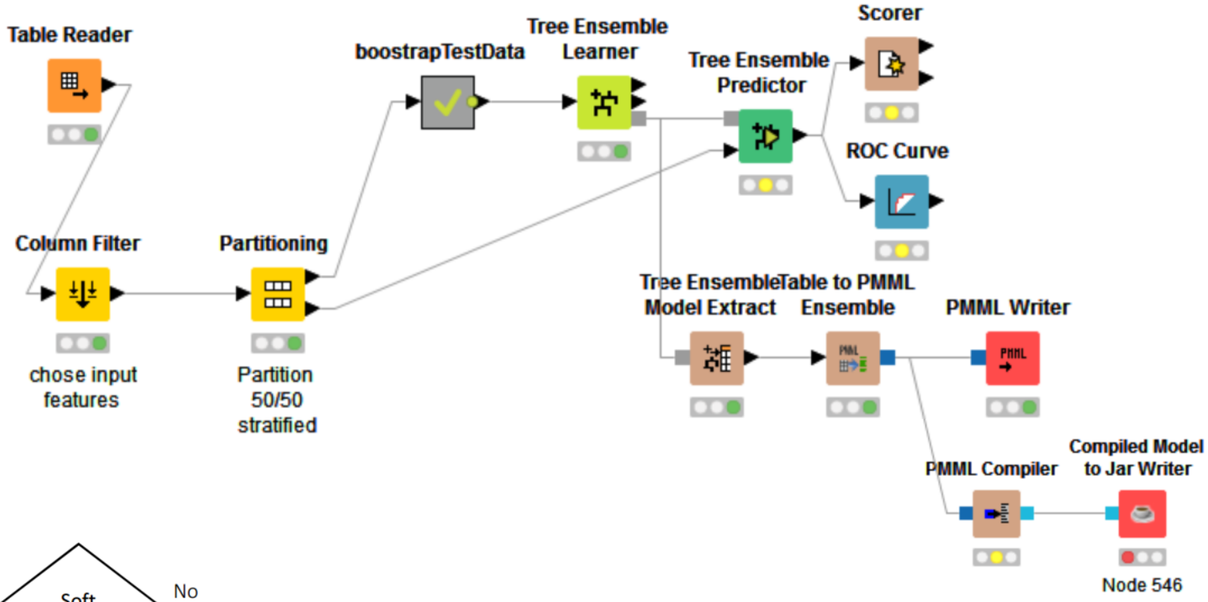
Row ID	I 0	I 1	D Area U...	D SE	D Interval
0	104	60	0.685	0.041	0.081
0_dte300	104	60	0.847	0.029	0.058
0_knn50	104	60	0.721	0.039	0.077
0_mlp1_15	104	60	0.674	0.042	0.082

Processing flow

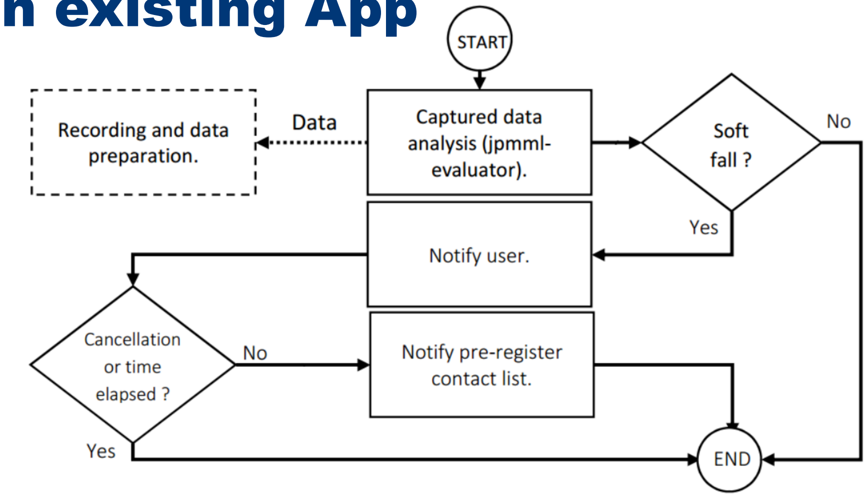


Implementation on Android

Jpmml - open source library



On existing App



START

Captured data analysis (jpmml-evaluator).

Data

Recording and data preparation.

Soft fall ?

No

Yes

Notify user.

Cancellation or time elapsed ?

No

Notify pre-register contact list.

Yes

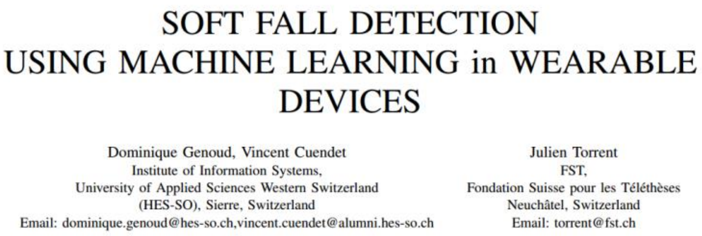
END

Enhancements

- Pmml to java compiler
- Wrapped nodes
- Optimize the size of the pmml in memory

References

- The example workflows will be available
- Paper on soft falls



SOFT FALL DETECTION
USING MACHINE LEARNING in WEARABLE
DEVICES

Dominique Genoud, Vincent Cuendet
Institute of Information Systems,
University of Applied Sciences Western Switzerland
(HES-SO), Sierre, Switzerland
Email: dominique.genoud@hes-so.ch,vincent.cuendet@alumni.hes-so.ch

Julien Torrent
FST,
Fondation Suisse pour les Téléthèses
Neuchâtel, Switzerland
Email: torrent@fst.ch

The 30th IEEE International Conference on
Advanced Information Networking and Applications (AINA-2016)
Le Régent Congress Centre, Crans-Montana, Switzerland,
March 23-25, 2016

Questions?

DOGBERT CONSULTS
MY DATA-MINING SOFTWARE HAS FOUND ANOTHER MESSAGE FROM GOD.
scottadams@aol.com
www.dilbert.com
IT SAYS YOU'VE BEEN STEALING LUNCHES FROM THE REFRIGERATOR IN THE BREAK ROOM.
© 1999 United Feature Syndicate, Inc.
THEN IT SAYS, "HA HA, THAT WASN'T PUDDING!"