# KNIMEを使った材料探索 基本操作(2)

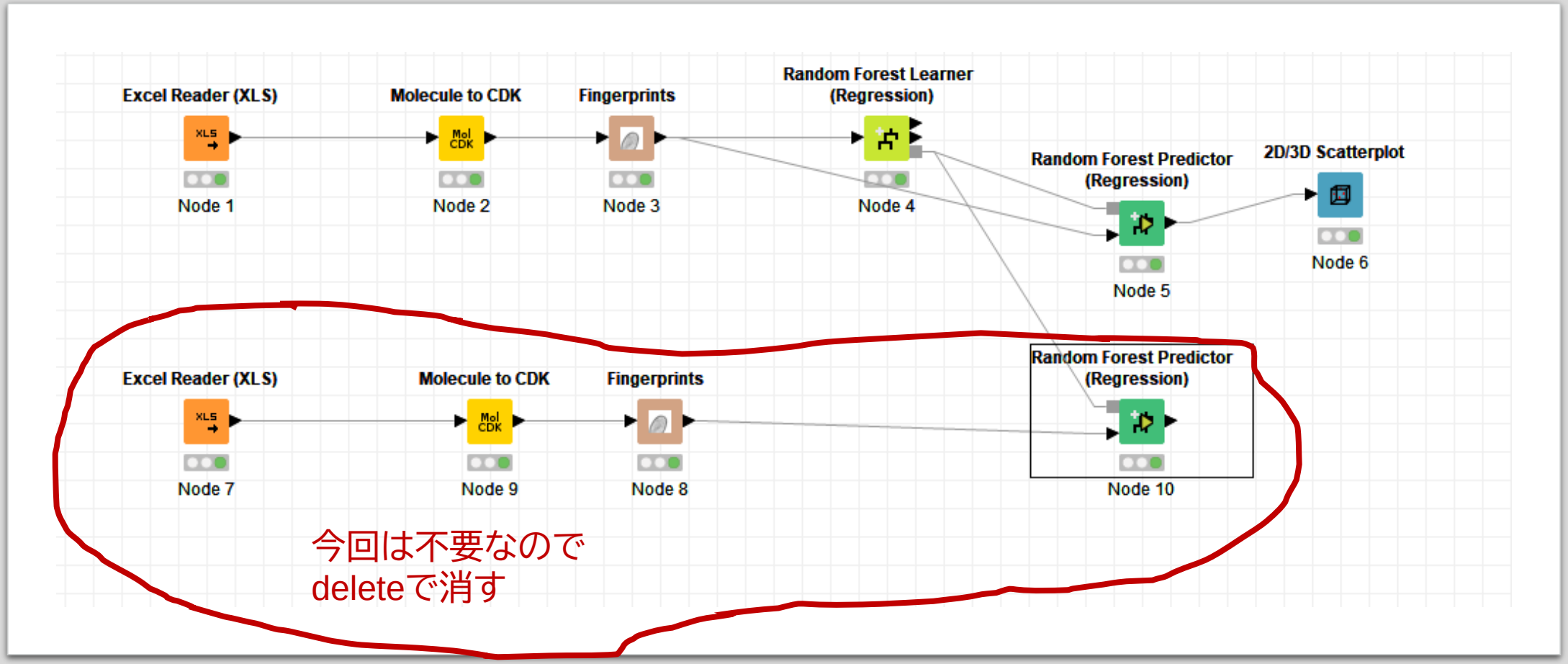
早稲田大学 応用化学科
講師(任期付) 畠山 歓
https://github.com/KanHatakeyama/
satokan@toki.早稲田.jp

# 今回扱う内容

- 訓練、検証データセットの作成
- 結果の定量評価
- 正規化
- 各種モデルの利用

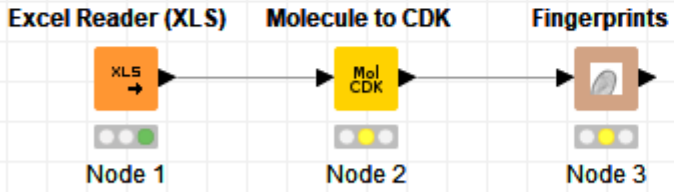
# 訓練・検証データセットの作成

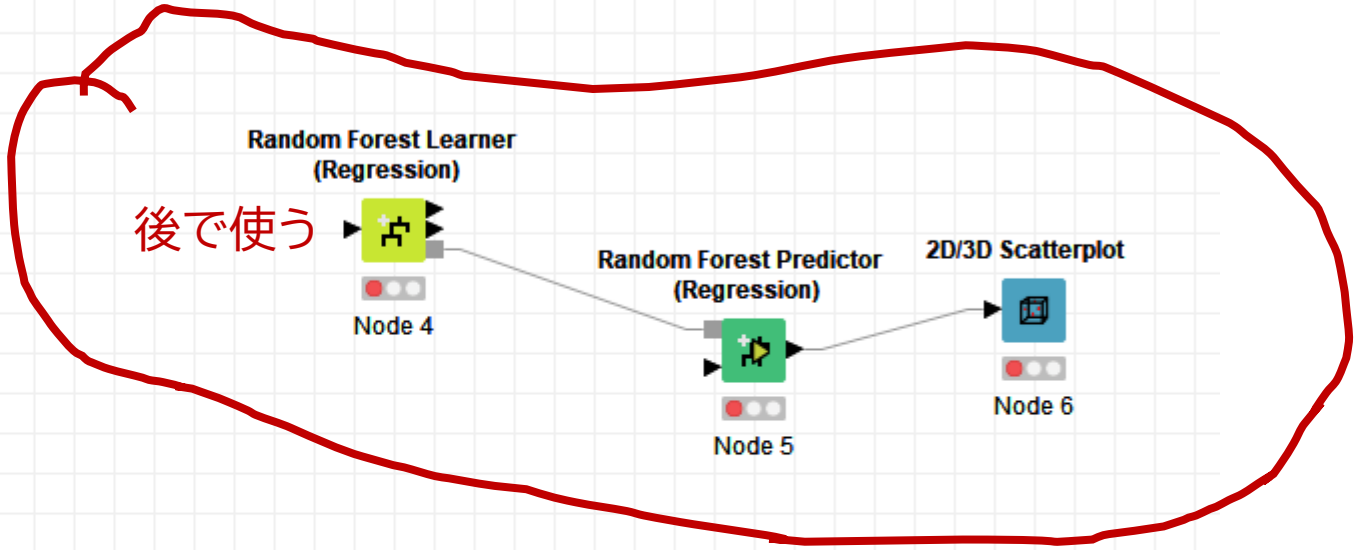
- 何故分けるのか?
  - Googleや書籍などで調べて下さい
  - 試験 検証 データ 分割 理由


Excel Reader (XLS)
Molecule to CDK
Fingerprints
Random Forest Learner (Regression)
Random Forest Predictor (Regression)
2D/3D Scatterplot
Node 1
Node 2
Node 3
Node 4
Node 5
Node 6
Excel Reader (XLS)
Molecule to CDK
Fingerprints
Random Forest Predictor (Regression)
Node 7
Node 9
Node 8
Node 10
今回は不要なので
deleteで消す

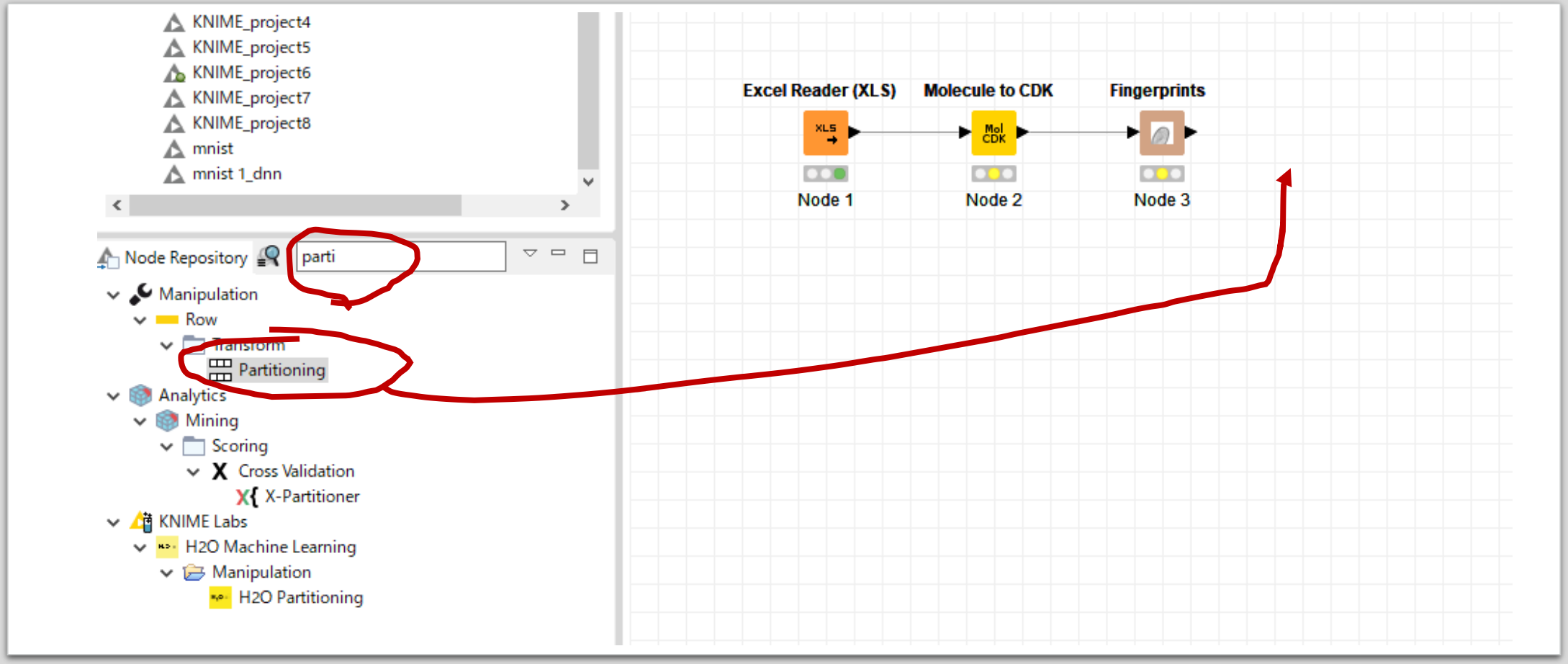
# 前回の画面

Deleteでノード間の結合は切れます



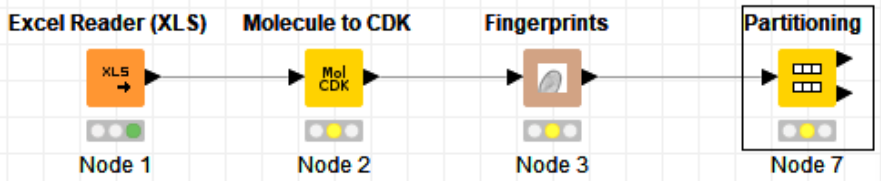



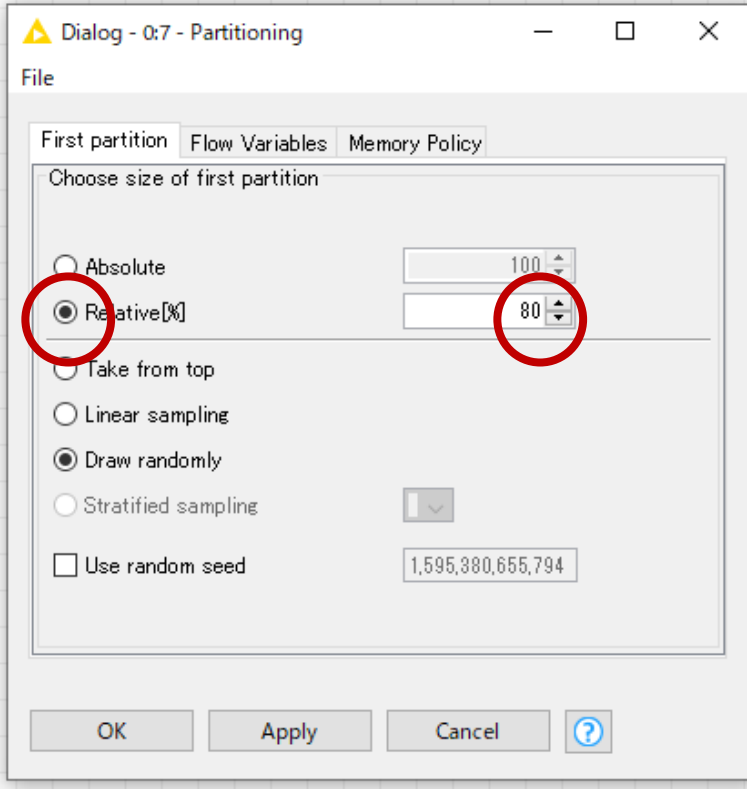

# ノードをつなぎ替える



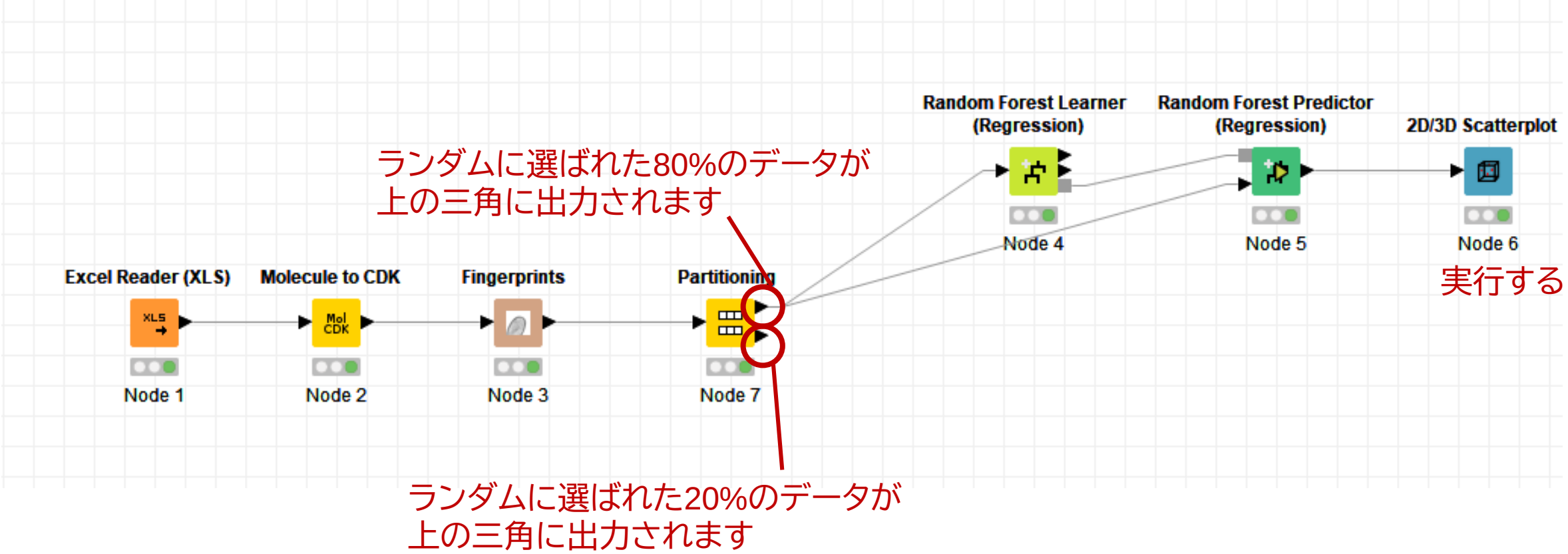

# Partitioningを設置

データの80%を訓練
20%を検証用に
ランダム抽出する、という指示

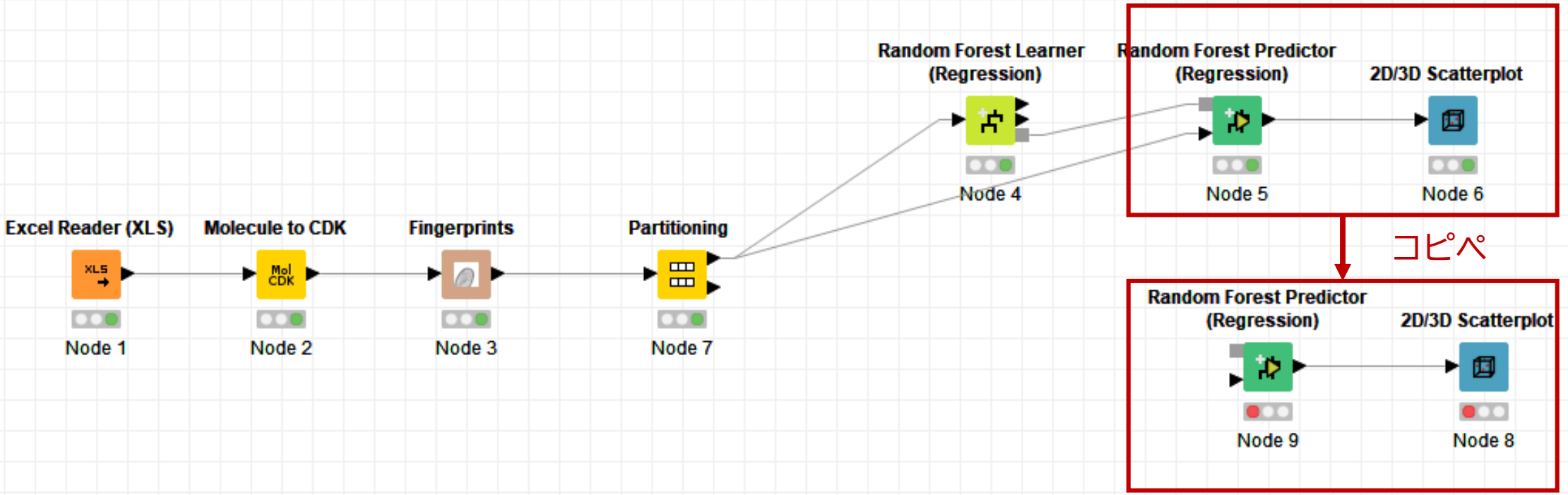


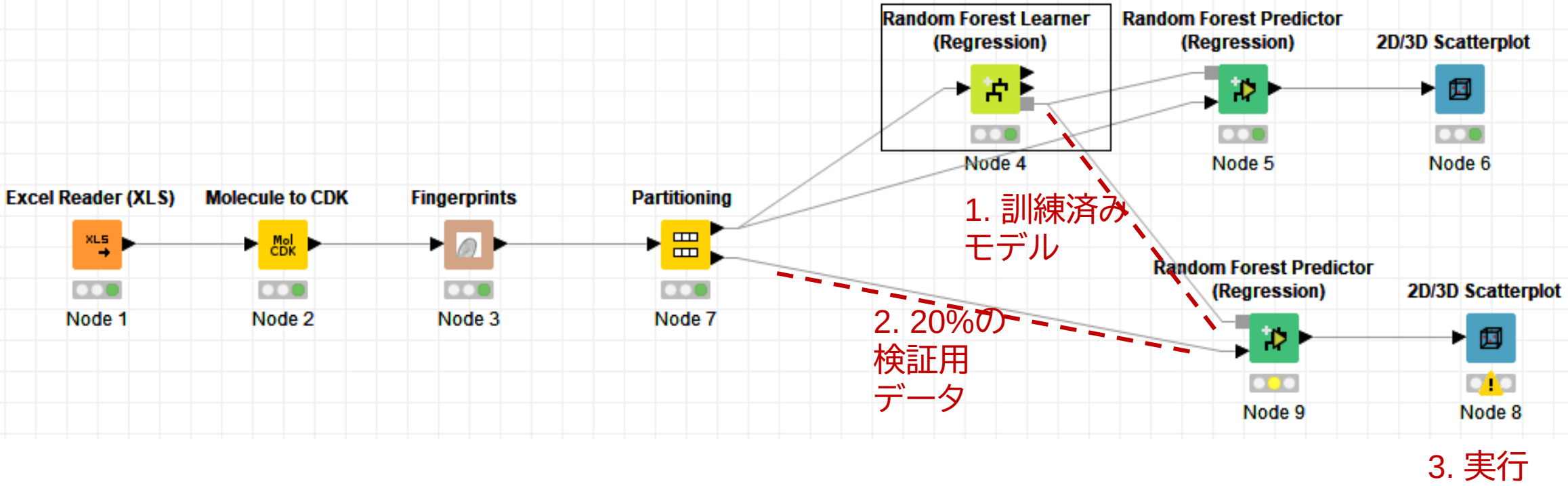

# Partitioningを設定


ランダムに選ばれた80%のデータが
上の三角に出力されます
Random Forest Learner
(Regression)
Random Forest Predictor
(Regression)
2D/3D Scatterplot
Excel Reader (XLS)
Molecule to CDK
Fingerprints
Partitioning
Node 4
Node 5
Node 6
実行する
Node 1
Node 2
Node 3
Node 7
ランダムに選ばれた20%のデータが
上の三角に出力されます

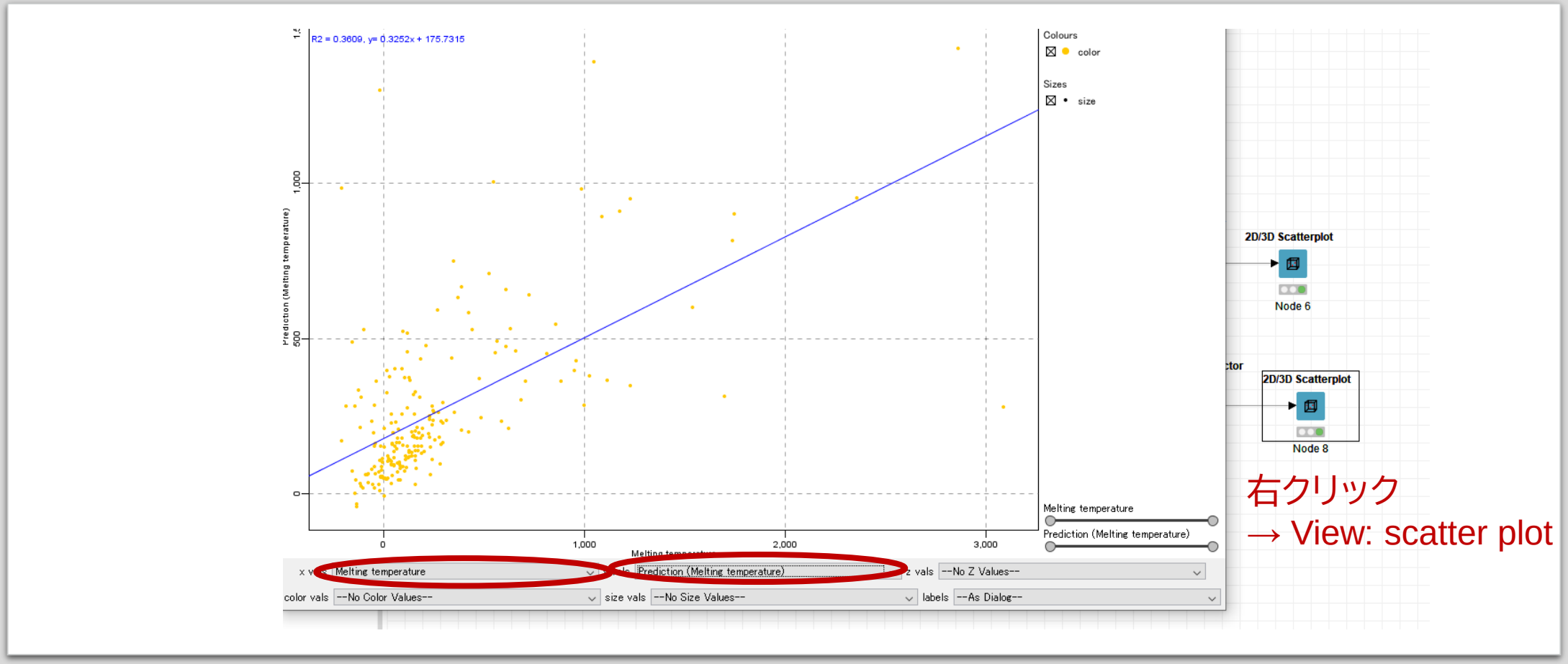
# 訓練データを学習




# 検証用Predictorを準備


Random Forest Learner
(Regression)
Random Forest Predictor
(Regression)
2D/3D Scatterplot
Node 4
Node 5
Node 6
Excel Reader (XLS)
Molecule to CDK
Fingerprints
Partitioning
Node 1
Node 2
Node 3
Node 7
1. 訓練済み
モデル
Random Forest Predictor
(Regression)
2D/3D Scatterplot
2. 20%の
検証用
データ
Node 9
Node 8
3. 実行

# 繋ぐ

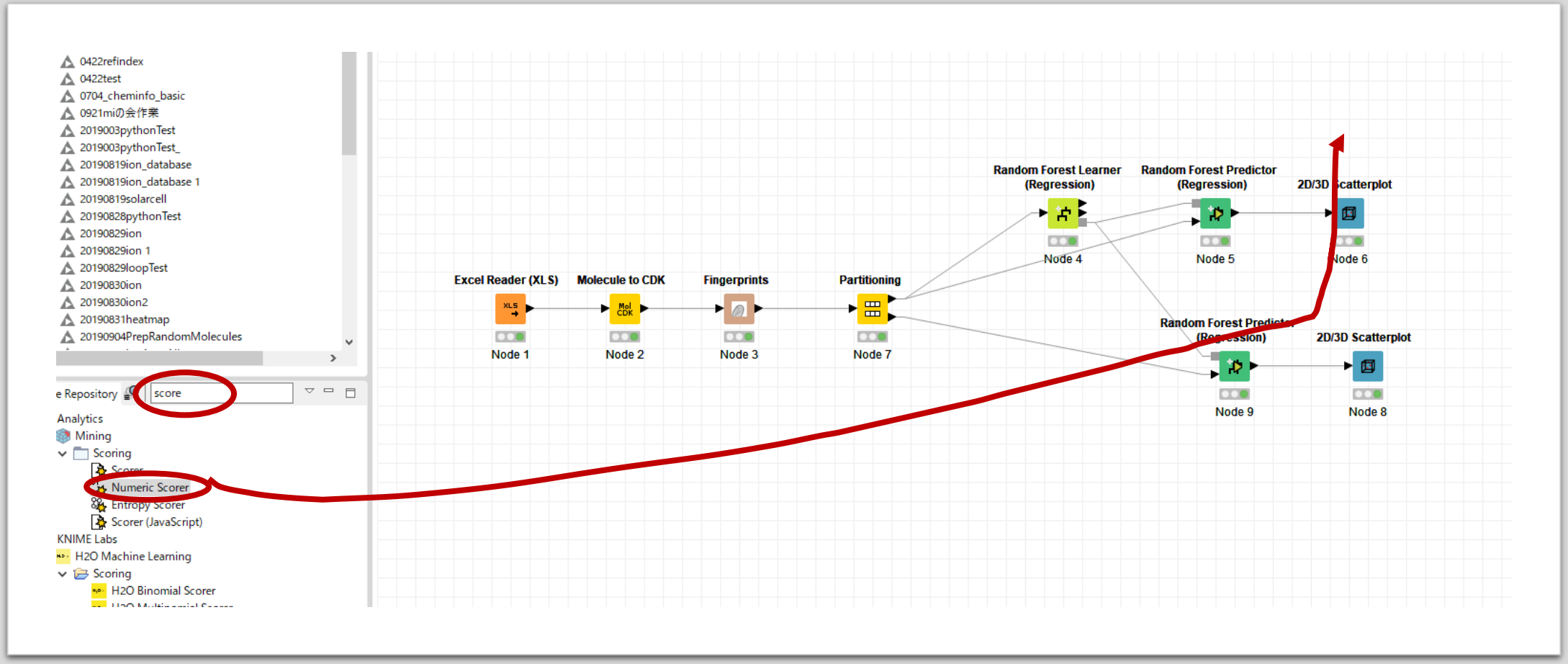

R2 = 0.3609, y= 0.3252x + 175.7315
Prediction (Melting temperature)
1,000
500
0
0
1,000
2,000
3,000
Colours
color
Sizes
size
Melting temperature
Prediction (Melting temperature)
x vals
Melting temperature
Prediction (Melting temperature)
z vals
--No Z Values--
color vals
--No Color Values--
size vals
--No Size Values--
labels
--As Dialog--
2D/3D Scatterplot
Node 6
2D/3D Scatterplot
Node 8
右クリック
→ View: scatter plot

# 結果確認

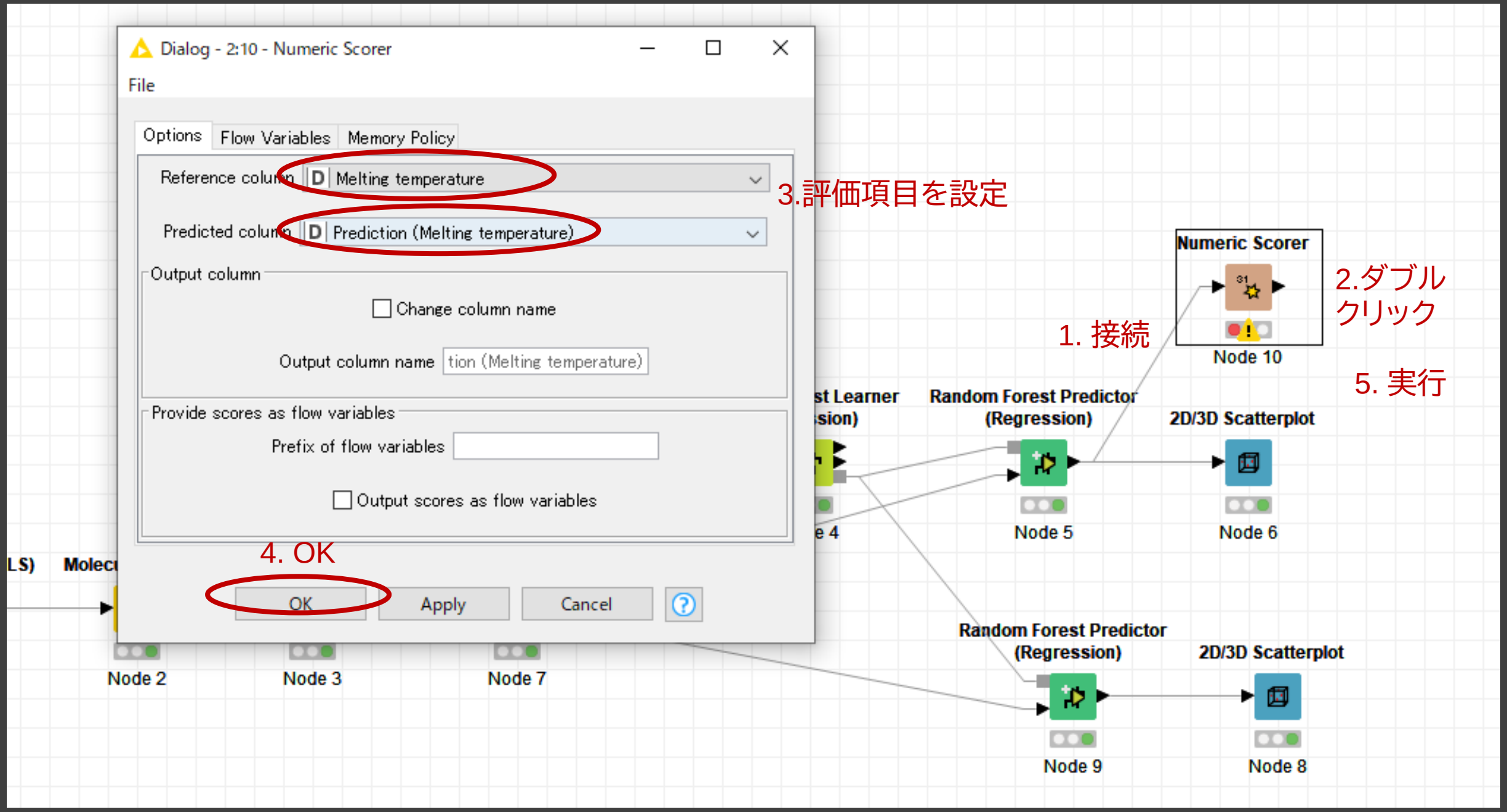
# 結果の定量評価

R2, MAE, MSE, …

意味は他所で調べて下さい




# Numeric Scoreerを配置


Dialog - 2:10 - Numeric Scorer
File
Options
Flow Variables
Memory Policy
Reference column
Melting temperature
Predicted column
Prediction (Melting temperature)
Output column
Change column name
Output column name
tion (Melting temperature)
Provide scores as flow variables
Prefix of flow variables
Output scores as flow variables
OK
Apply
Cancel
3.評価項目を設定
4. OK
1. 接続
2.ダブル
クリック
5. 実行
Numeric Scorer
Node 10
Random Forest Predictor
(Regression)
2D/3D Scatterplot
Node 5
Node 6
Random Forest Predictor
(Regression)
2D/3D Scatterplot
Node 9
Node 8
Node 2
Node 3
Node 7

Statistics - 2:10 - Numeric Scorer

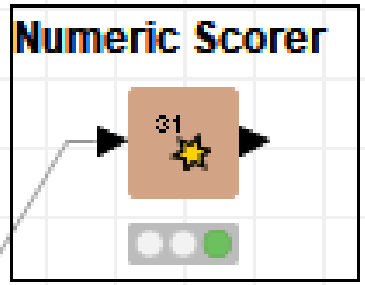
File Hilite Navigation View

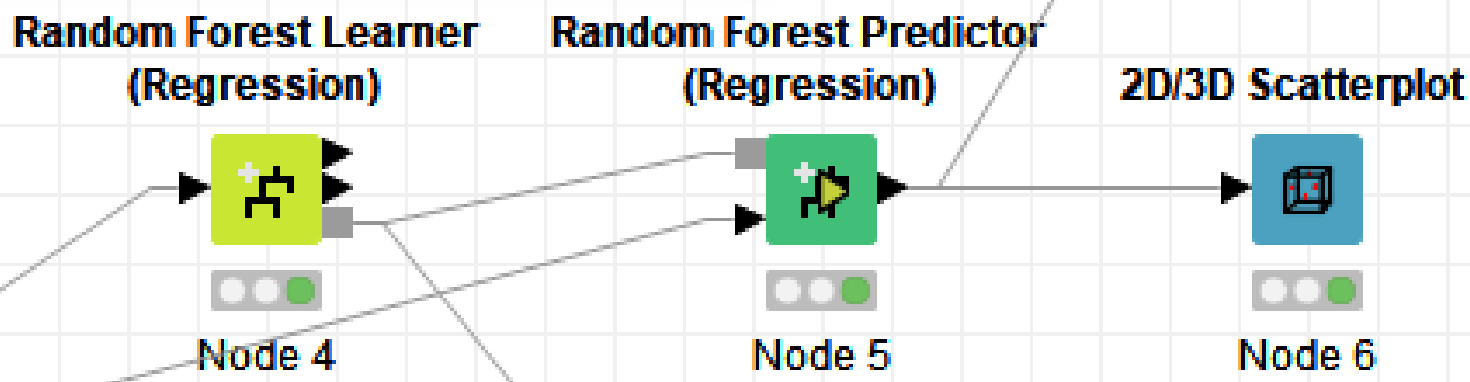
Table "Scores" - Rows: 5 | Spec - Column: 1 | Properties | Flow Variables

| Row ID | D Predicti... |
|---|---|
| R^2 | 0.801 |
| mean absolute error | 111.926 |
| mean squared error | 43,819.422 |
| root mean squared deviation | 209.331 |
| mean signed difference | -4.919 |

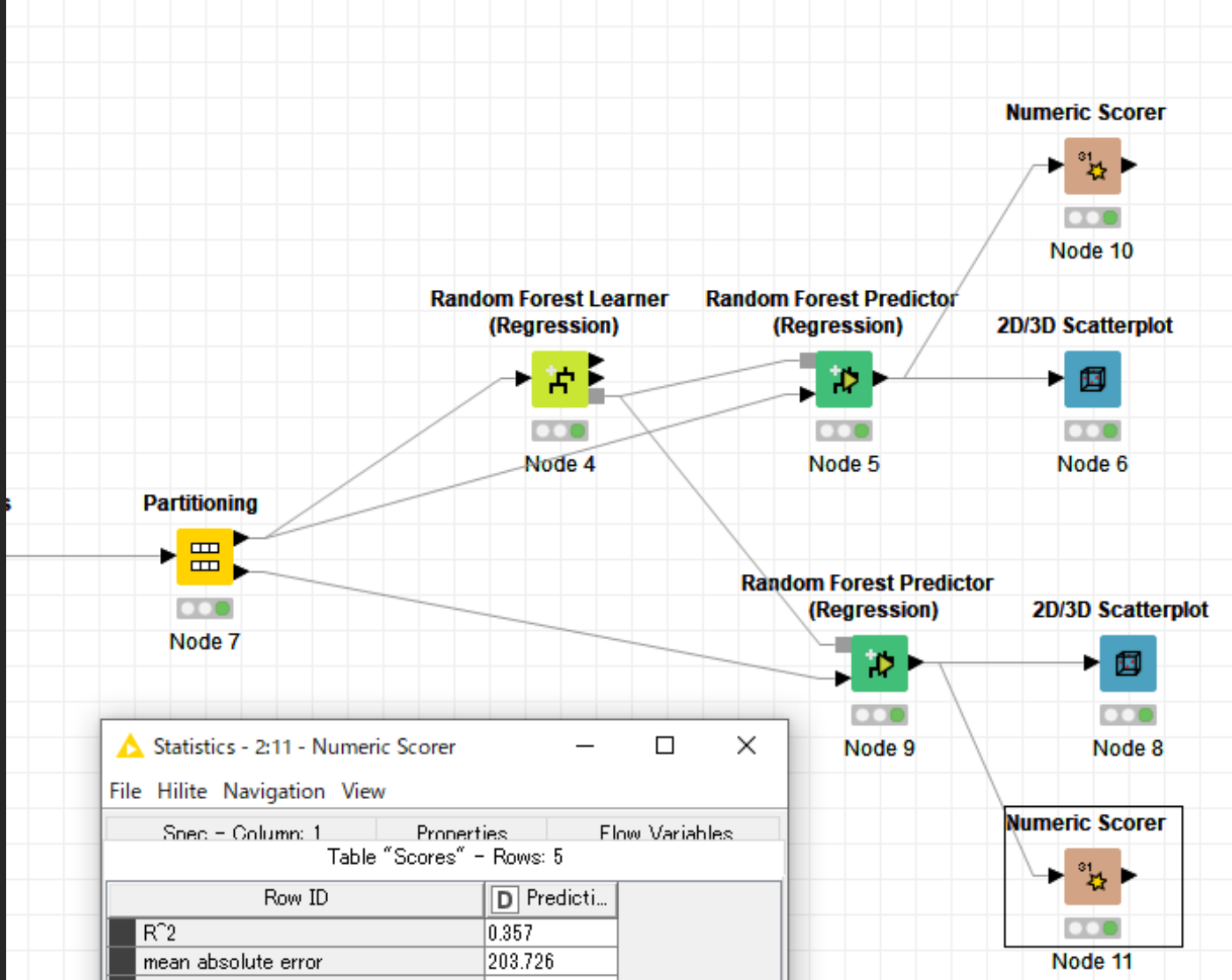
結果が出てきました!

右クリック
→Statistics



# Numeric scorerをコピペし、検証データの結果も見る




Statistics - 2:11 - Numeric Scorer

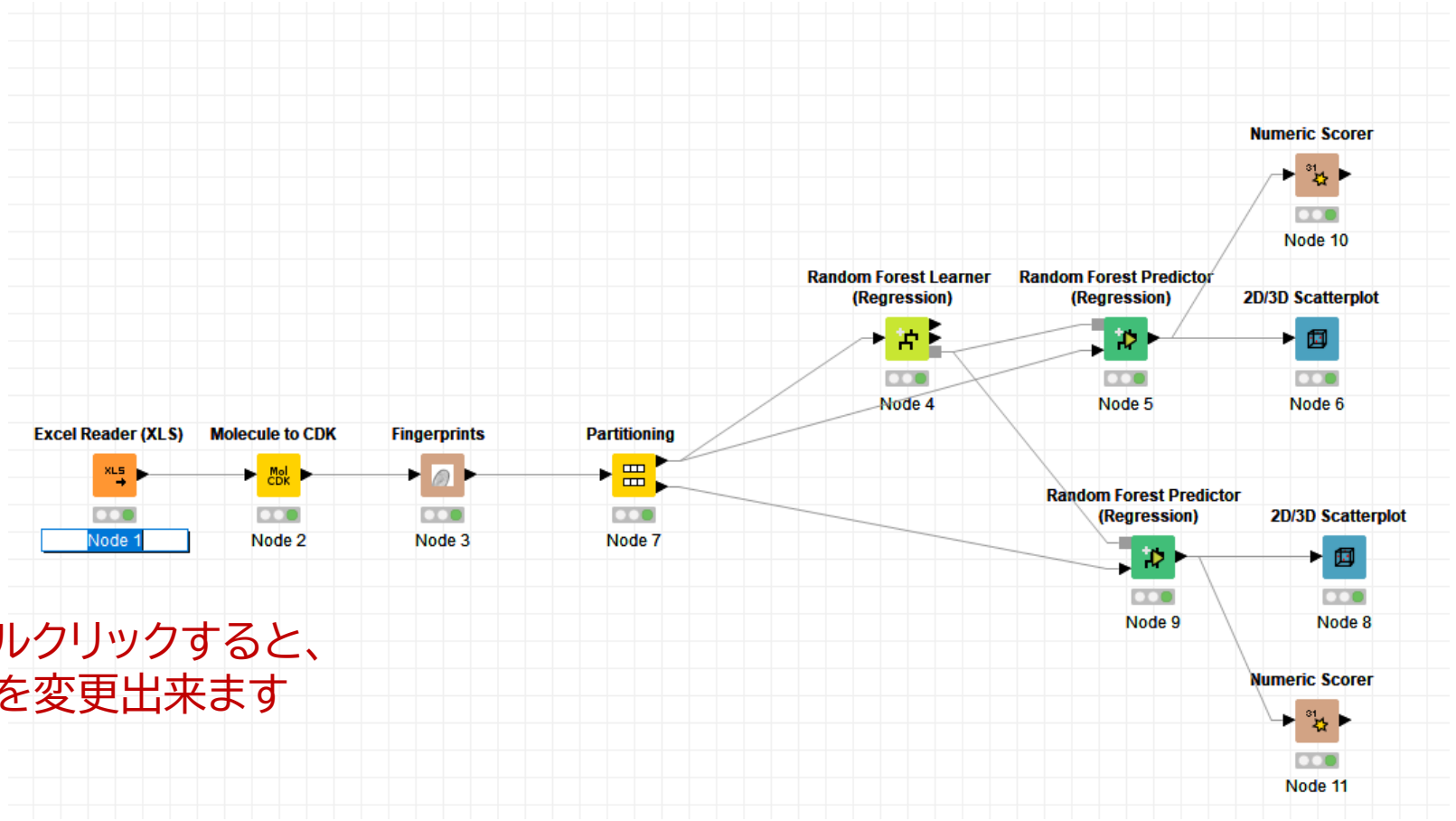
File Hilite Navigation View

Spec - Column: 1 Properties Flow Variables

Table "Scores" - Rows: 5

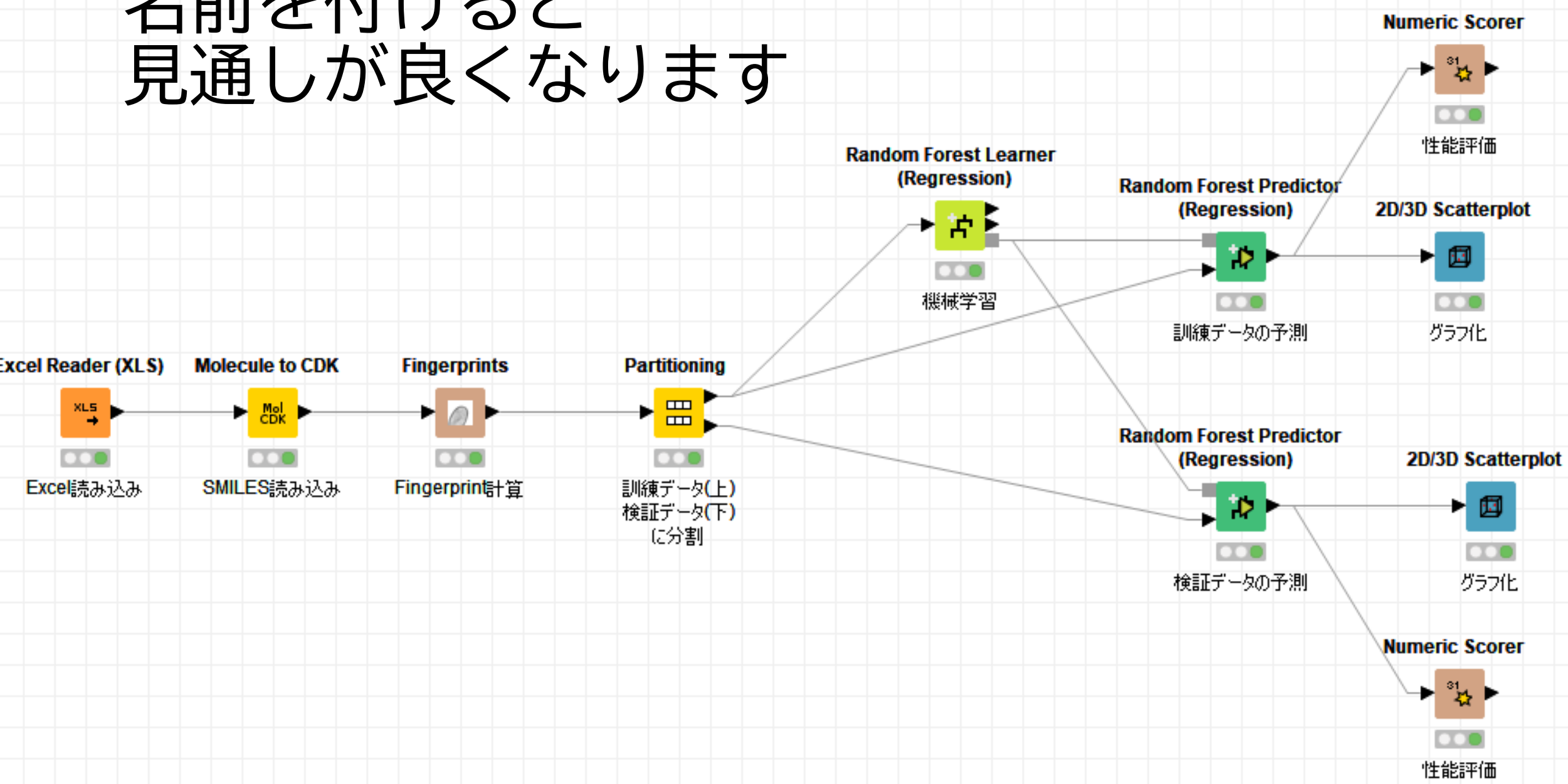
| Row ID | D Predicti... |
|---|---|
| R^2 | 0.357 |
| mean absolute error | 203.726 |
| mean squared error | 138,910.585 |
| root mean squared deviation | 372.707 |
| mean signed difference | 1.734 |

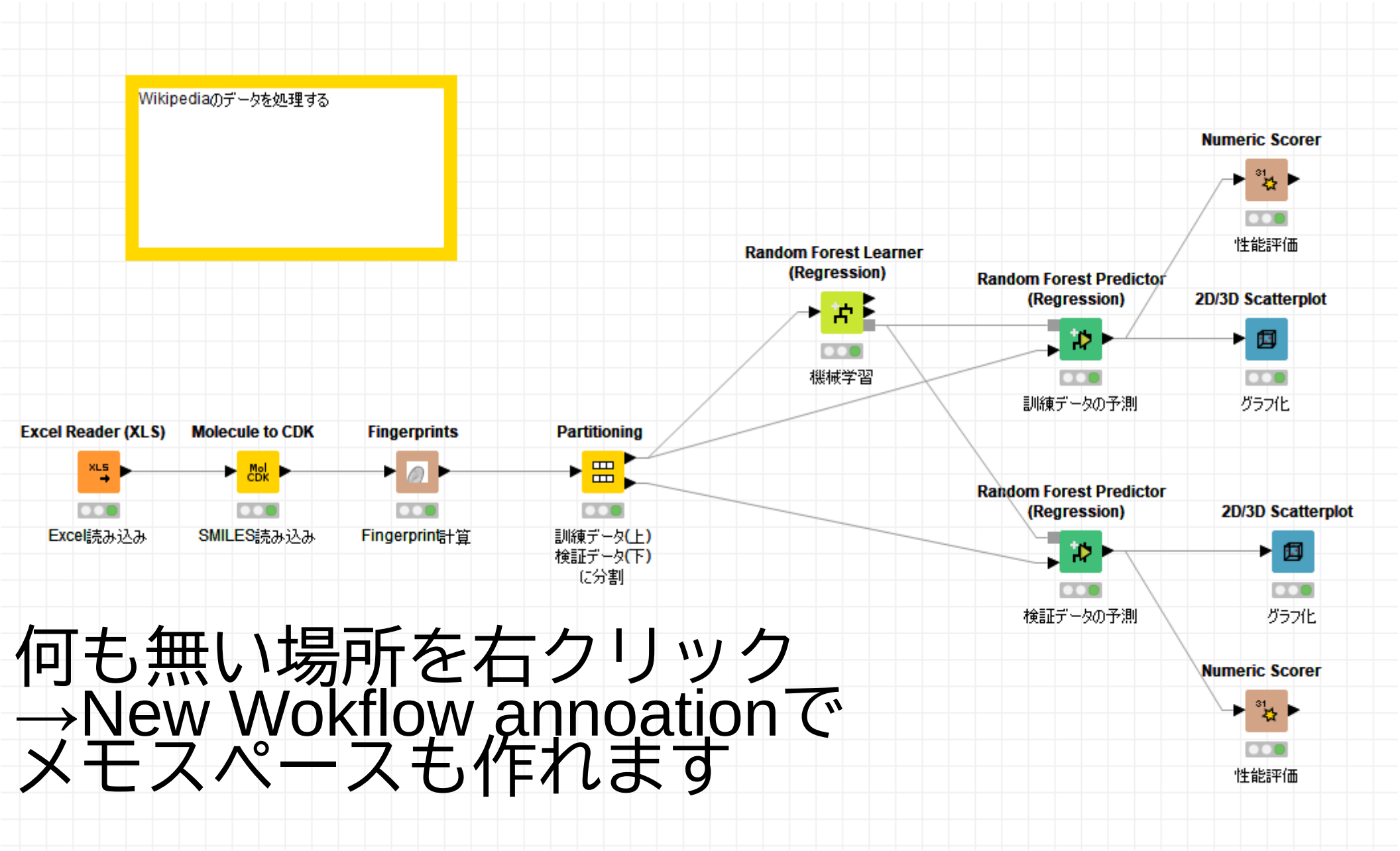
# 全体図




ダブルクリックすると、
名前を変更出来ます

# 名前を付けると
# 見通しが良くなります

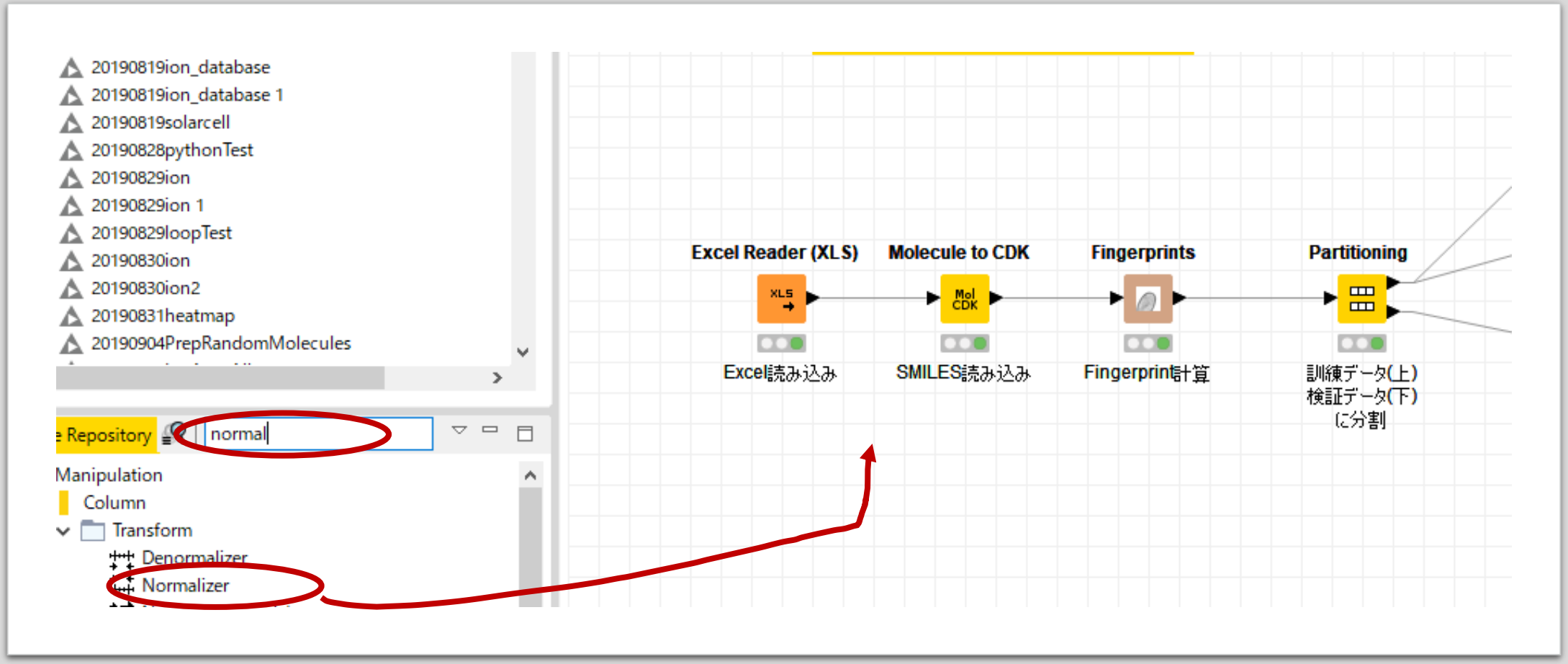




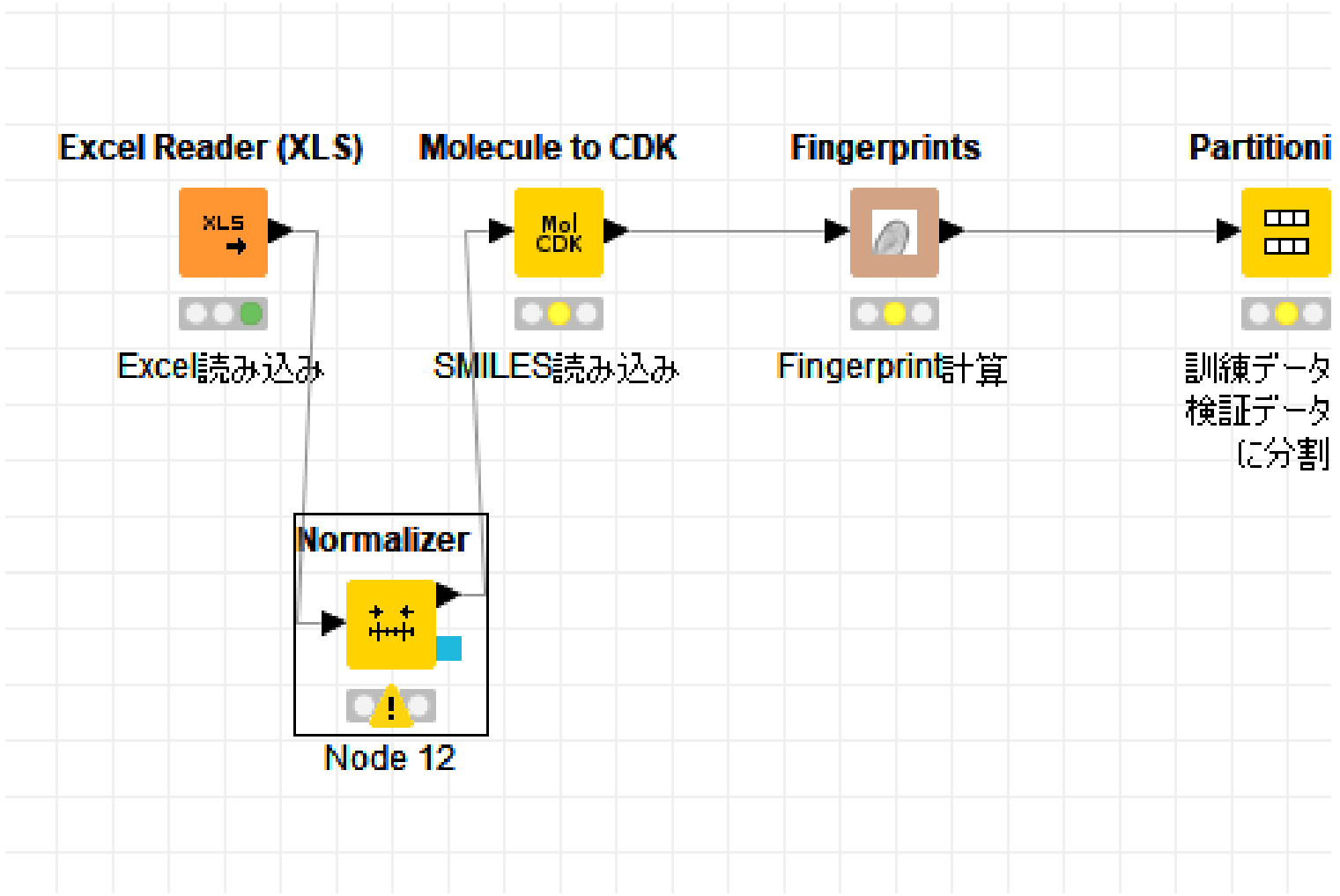

何も無い場所を右クリック
→New Wokflow annoationで
メモスペースも作れます

# 正規化

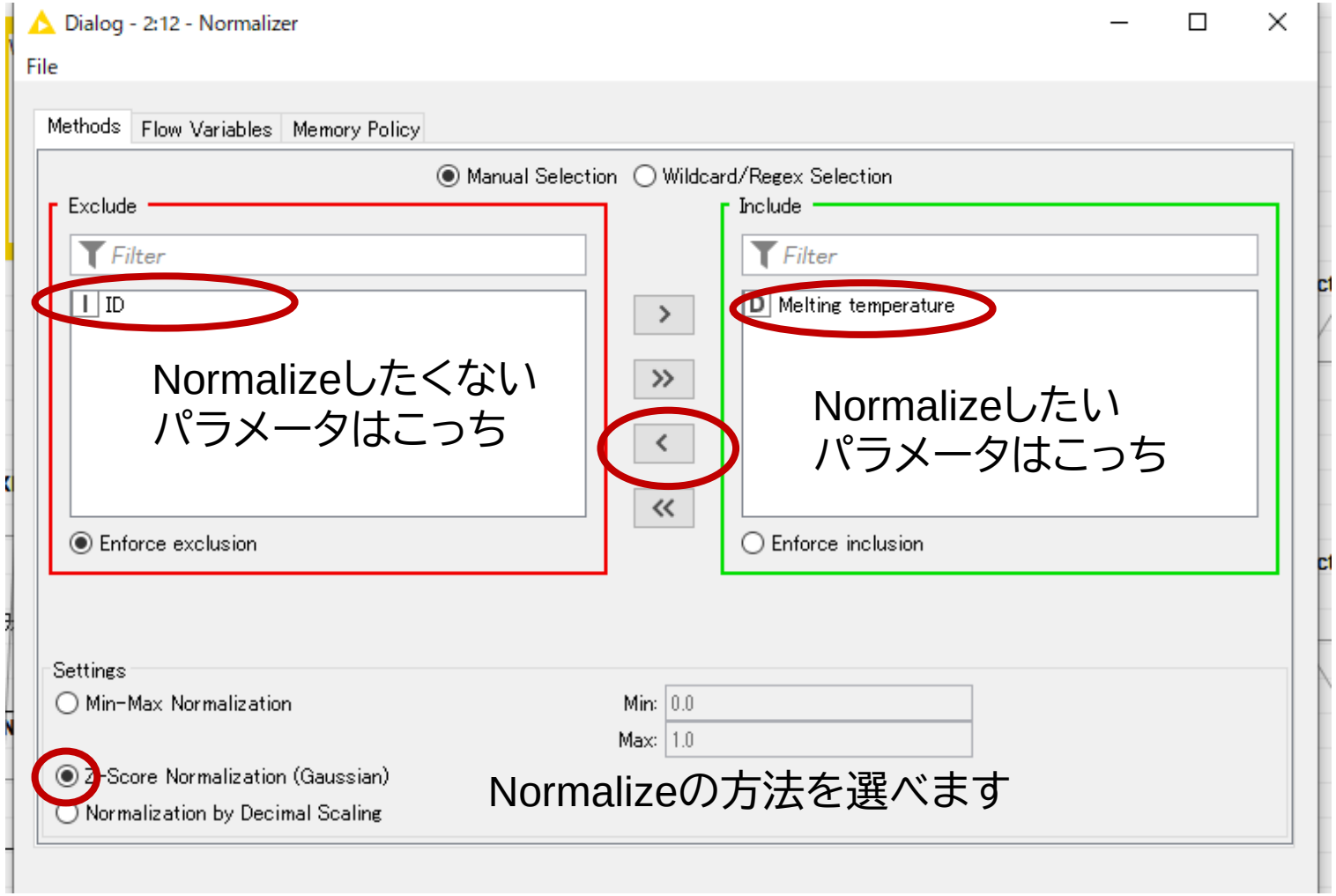
いわゆる標準得点 (z-score)を計算します
操作の理由は他所で調べて下さい




# Normalizerの設置


Excel Reader (XLS)
Molecule to CDK
Fingerprints
Partitioni
XLS
Mol
CDK
Excel読み込み
SMILES読み込み
Fingerprint計算
訓練データ
検証データ
に分割
Normalizer
Node 12

# ノードのつなぎ替え

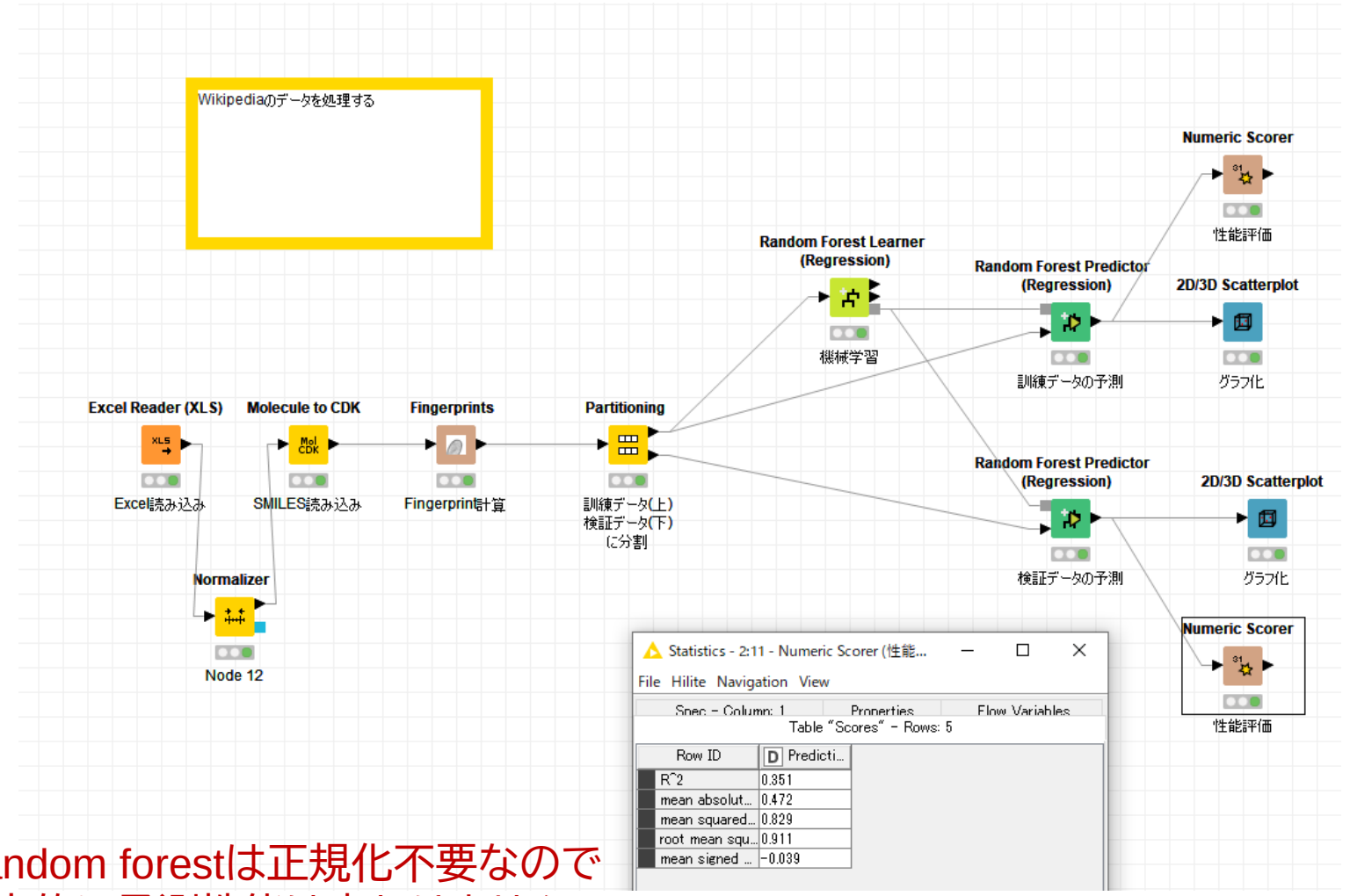



# Normalizer 設定

| Row ID | I ID | S SMILES | D Melting ... |
|---|---|---|---|
| Row0 | 1 | [Cu]=S | 0.518 |
| Row1 | 2 | c1cc2ccc3cc... | -0.299 |
| Row2 | 3 | O1[Fe]2O[F... | 2.736 |
| Row3 | 4 | O=C1NC(=O... | -0.026 |
| Row4 | 5 | P#[Y] | -0.12 |
| Row5 | 6 | C1=CC=C(C... | 0.07 |
| Row6 | 7 | ClC(Cl)C(=O... | -0.34 |
| Row7 | 8 | FC(F)F | -0.88 |
| Row8 | 9 | O=[N+]([O-]... | -0.318 |
| Row9 | 10 | CCC[C@@H]... | -0.365 |
| Row10 | 11 | c1ccc2c(c1)... | -0.613 |
| Row11 | 12 | O=C(O)[C@... | 0.059 |
| Row12 | 13 | F[Co](F)F | 1.43 |
| Row13 | 14 | [Cs+].[I-] | 0.8 |
| Row14 | 15 | C1CCC(CC1... | -0.54 |
| Row15 | 16 | O=C2c3c(O[... | -0.013 |
| Row16 | 17 | BrC(F)(F)F | -0.907 |

正規化
されました

Normalizer
右クリック
→
Normalized
table

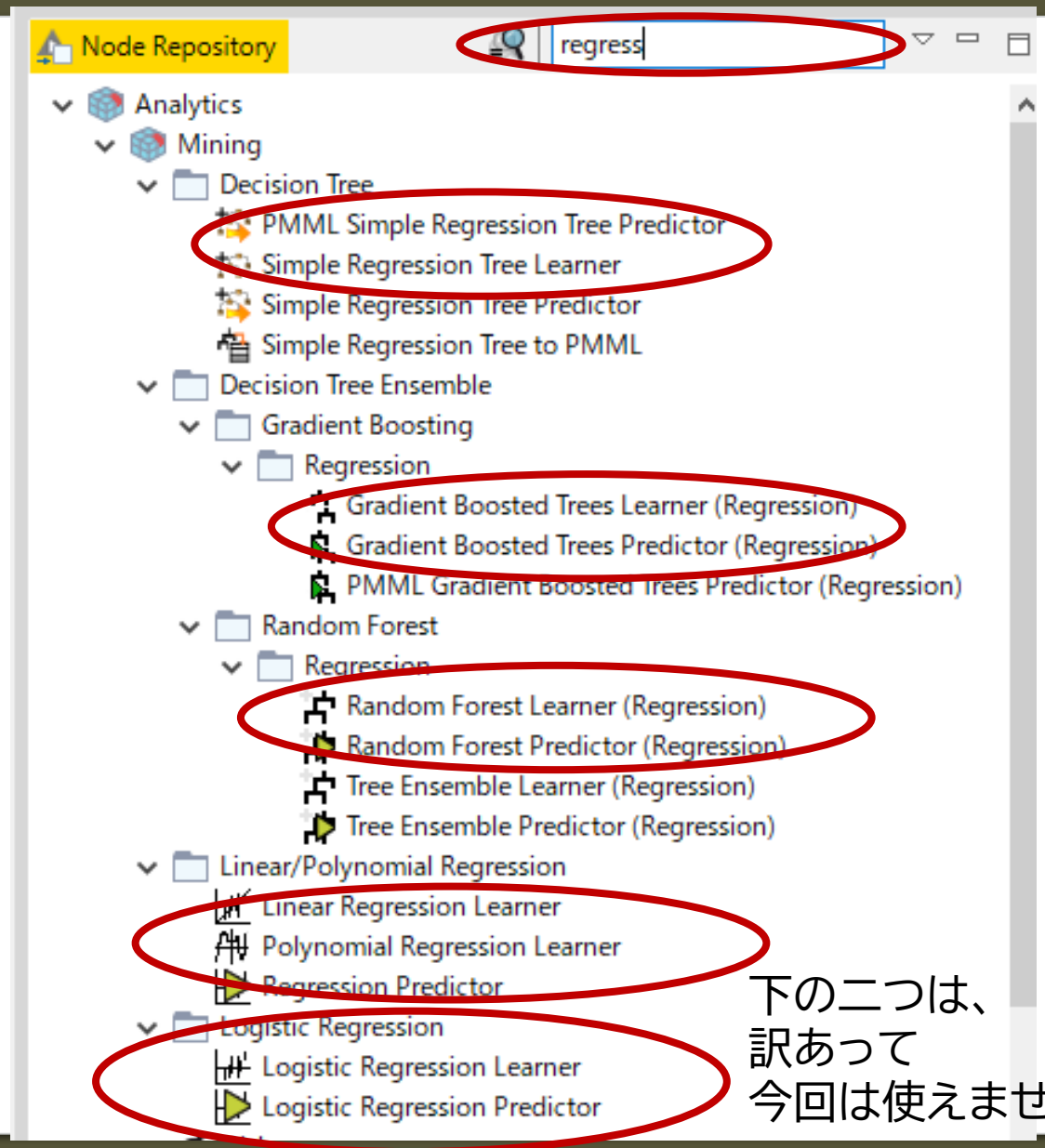



全て実行

Random forestは正規化不要なので
基本的に予測性能は変わりません

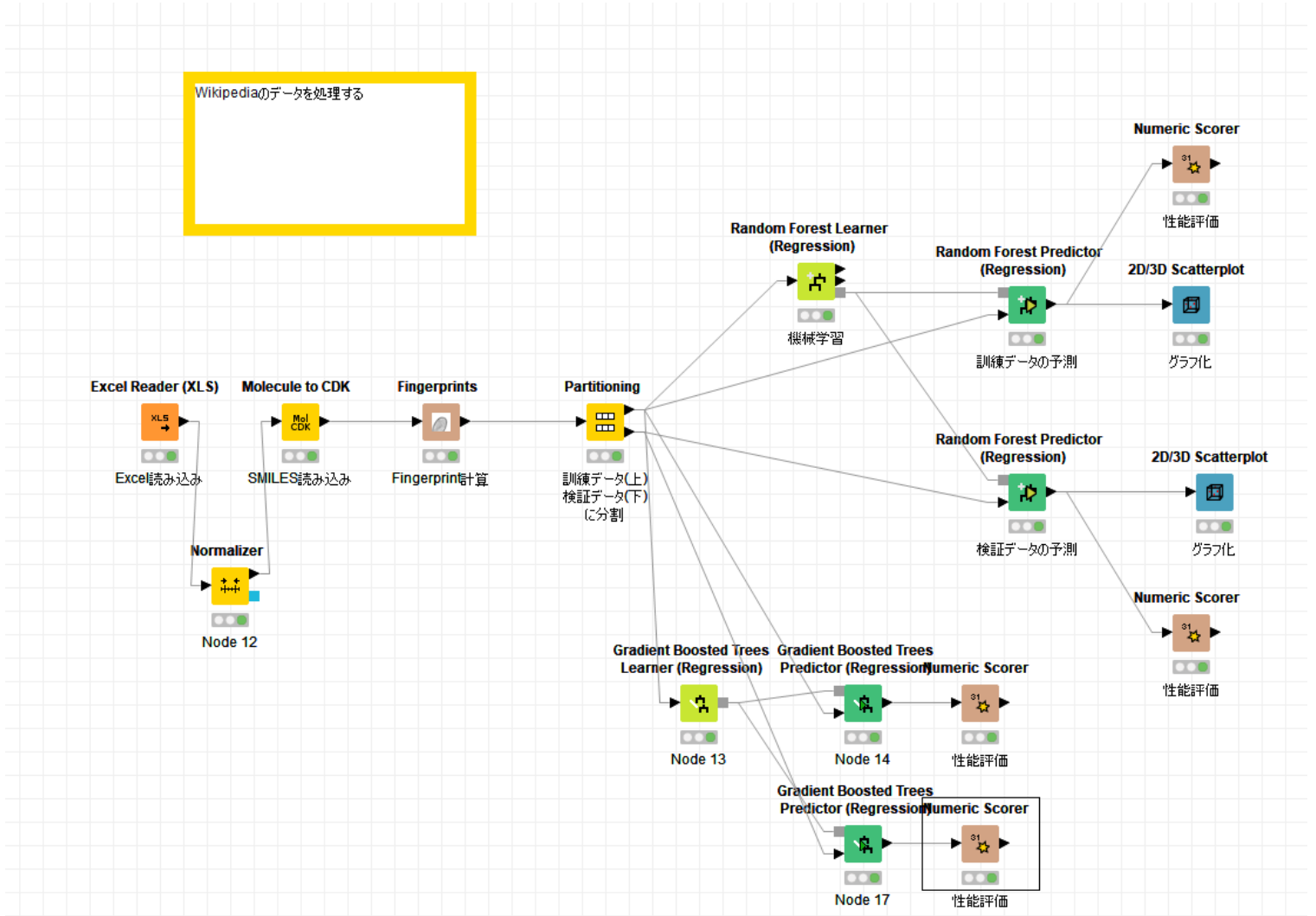
# 各種モデルの利用

実はKNIMEの対応モデルは標準では
あまり多くありません

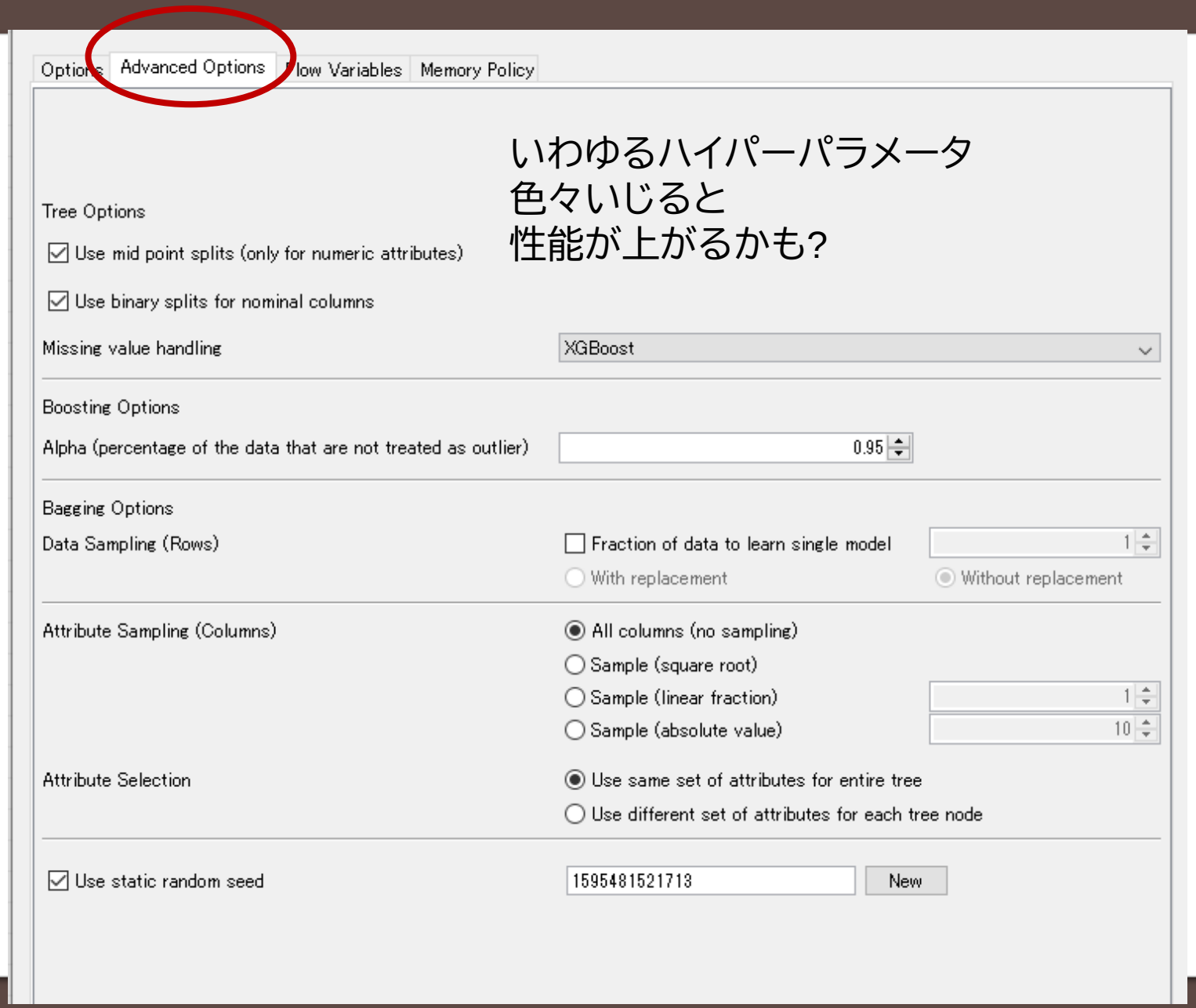



# 利用可能な回帰モデル

下の二つは、
訳あって
今回は使えません(Fingerprint)




# Gradient boostingを追加してみた例




# Gradient boostingのAdvanced options

# 今後のTODO

- Fingerprint以外の方法
- 欠損データの処理
- 多彩なモデルの利用