# 検証方法のご紹介

# はじめに

作成したモデルの汎化性能を図るために、交差検証がよく使用されます。

（汎化性能＝未知の新たなデータに対してもうまく予測できる性能）

検証方法にはさまざまな種類があり、手元にあるデータのサンプル数や、
時系列データに対して時間軸を考慮した検証を行いたいかなど、状況によって
使い分ける必要があります。

本資料では、検証方法を4つご紹介します。RapidMinerで実行する際のオペレータ、
設定方法をご説明します。およそのサンプル数なども記載しておりますので、
状況に応じて検証方法を試してみてください。

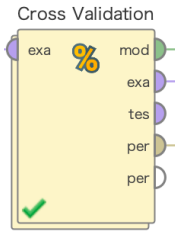
# 検証方法

# leave one out 交差検証

**データ数が少ない場合に用いられる**

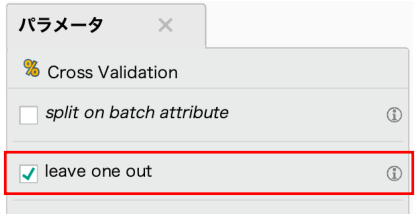
全データの中から1つだけ抜き出したデータをテスト用とし、残りの全てを学習用として利用する交差検証です。

**サンプル数：およそ30以下**




**Cross Validation オペレータ**

leave one outに チェックを入れて使用 します。




**leave one out cross-validation** (LOOCV)

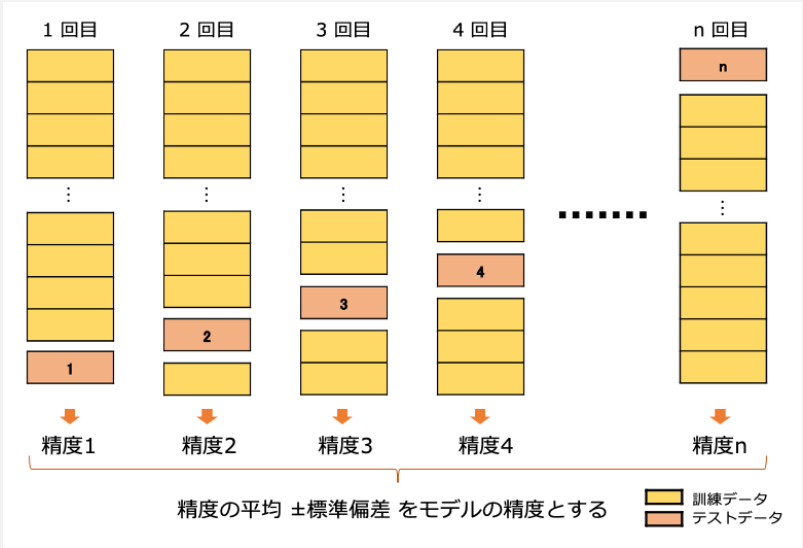
※k-fold交差検証(k分割検証)のkを標本サイズnにした場合と同じ

データ数がn個ある場合、データをn分割します。
その中から1つのデータをテスト用として取っておき、残り全てのデータ(n-1個) でモデルの作成を行い、取っておいた1つのテストデータで検証を行います。

これを全てのデータが1回ずつテストデータになるように、
データの数だけ(n回)繰り返します。

最後に、n回の学習で得られた全ての評価(精度)を平均したものを最終結果として算出します。

（評価結果 performanceを確認すると、テストデータが1サンプルだけなので、RMSEは実測値と予測値の誤差で計算できますが、決定係数や相関係数は計算することができず”0”になります。）



# k-fold 交差検証

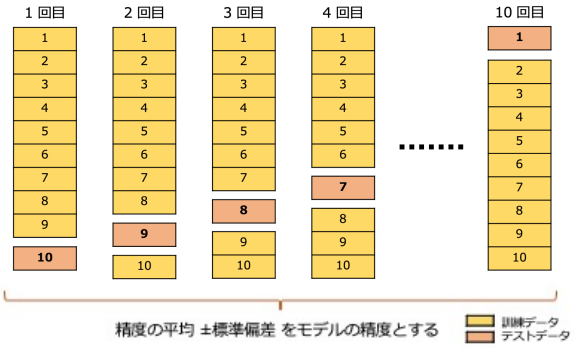
**データ数がある程度ある場合に用いられる**

データ全体をk個に分割し、そのうち1つをテスト用に、残りのk-1個を学習用として利用します。

テストデータと訓練データの組み合わせを変えてk回検証を繰り返します。

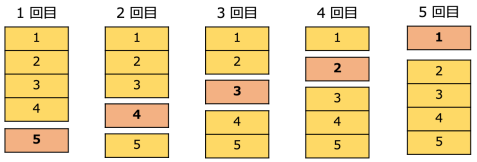
■ **10-fold 交差検証 (10分割)**

**サンプル数：およそ 100 から 30 くらい**



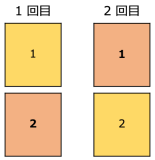

■ **5-fold 交差検証 (5分割)**

**サンプル数：およそ 1000 から 100 くらい**



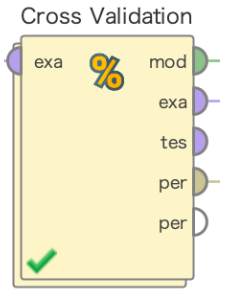

■ **2-fold 交差検証 (2分割)**

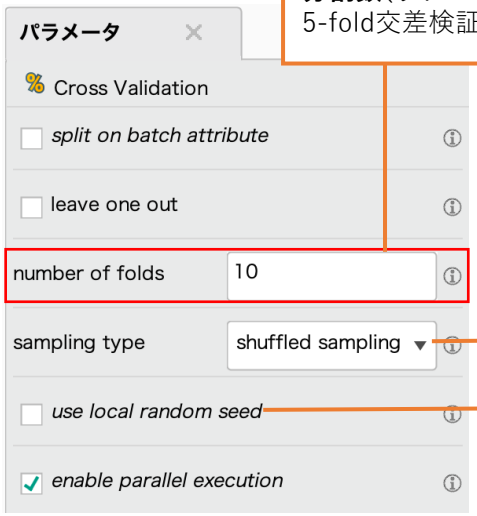
**サンプル数：およそ 1000 サンプル以上**



# k-fold 交差検証

## ▷オペレータのパラメータ設定

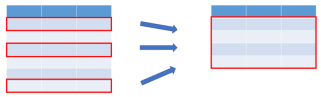


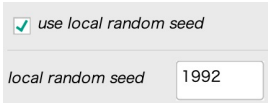

**分割数(サブセットの数)の指定**
5-fold交差検証を行いたい場合は ”5”に設定

**サンプリング方法の選択**

shuffled_sampling (シャッフルサンプリング)とは、
ExampleSet のランダムなサブセットを構築します。
ランダムに行が選択されます。



実行結果を合わせる場合はチェック
local random seedを同じ値にすると
ランダム性が同じサブセットが
生成されます。




＜他のサンプリングタイプ＞

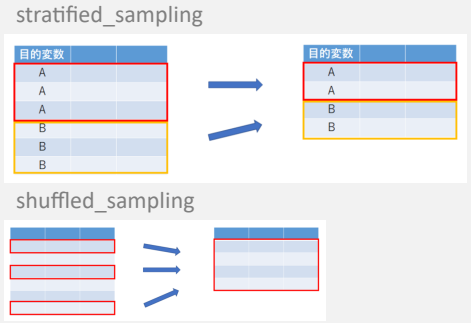
**automatic(自動モード) :**

分類問題の場合は、各サブセットにラベルの値のほぼ同じ割合が含まれる

ようにランダムなサブセットが構築されます（stratified_sampling ）。

目的変数が数値の場合は、自動でshuffled_samplingが行われます。

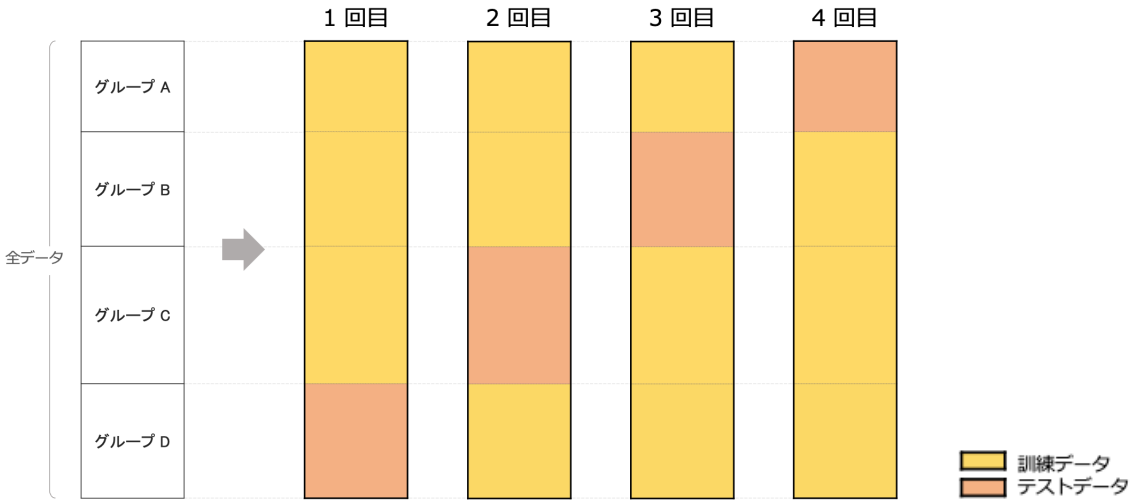
▼参照：サンプリング方法



# group k-fold 交差検証

**訓練データとテストデータの両方に同じグループが含まれないように**
**データを分ける場合に用いられる**

k-fold 交差検証では、サブセットの数を指定し同じサイズのk個のサブセットに分割されていました。group k-fold 交差検証では、任意のサブセット(グループ)で分割します。

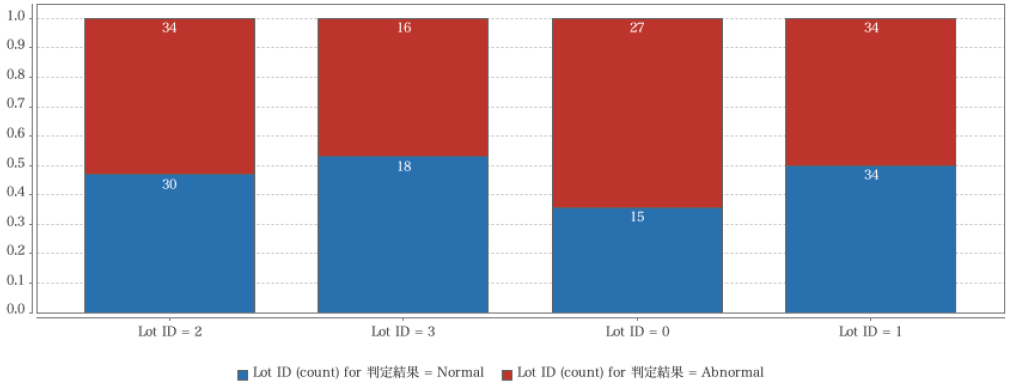



例えば、製造業の場合、ロット単位でデータが分割されるケースがよくあります。
訓練データとテストデータに同一のロットのデータが混在しないようにすることで、
時間的に早く生産された“Lot T0”のデータのみを使って、時間的に後で生産された
“Lot T1”の予測を行えるかどうかを判断することができるようになります。

# group k-fold 交差検証

## ▷ケース

4つのロットに分けて生産される208件の製品に対して、異常、正常が割り当てられたデータを取り上げます。異常、正常の分布は下図の通りであり、例えば、Lot ID＝1で製造された場合、34件の正常、34件の異常が存在しています。




データセットの例

| result | Lot ID | sensor_1 | sensor_2 | sensor_3 | s |
|---|---|---|---|---|---|
| Normal | 1 | 0.003 | 0.031 | 0.017 | 0 |
| Normal | 2 | 0.029 | 0.040 | 0.077 | 0 |
| Normal | 2 | 0.018 | 0.015 | 0.003 | 0 |
| Abnormal | 0 | 0.049 | 0.028 | 0.059 | 0 |
| Abnormal | 2 | 0.131 | 0.234 | 0.306 | 0 |
| Abnormal | 0 | 0.020 | 0.042 | 0.055 | 0 |

ExampleSet (208 行,2 特別属性,60 通常属性)

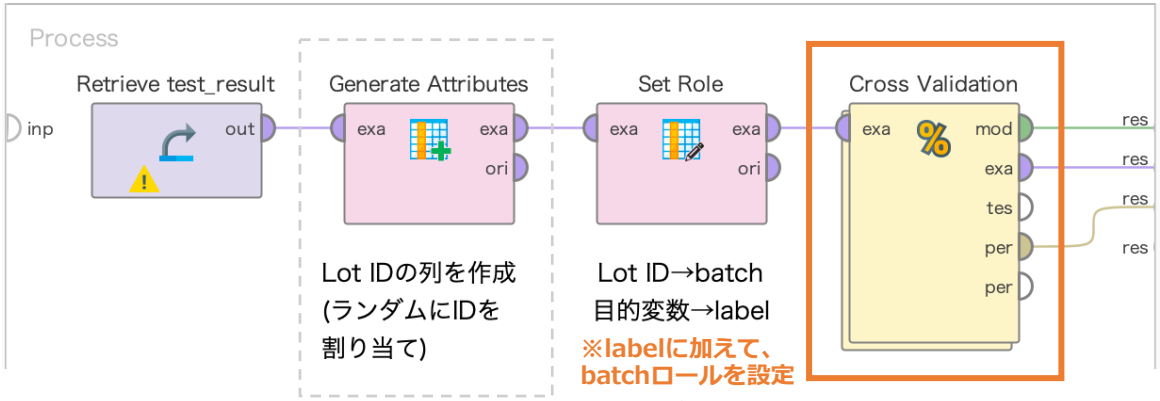
group k-fold 交差検証を行う場合、次ページのようなプロセスを構築します。

# group k-fold 交差検証

## ▷プロセス構築例

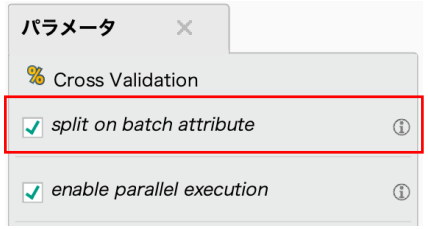
① labelに加えて、サブセットの分割単位となる属性にbatch ロールを付与します。
（今回のケースは、Lot IDにbatchロールを設定）

② 特別なロール「batch」を持つ属性を使用してグループ化し、データを分割します。

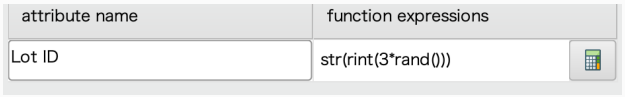



**Cross Validationオペレータ**

② split on batch attributeに
チェックを入れて使用します。




**※Lot IDが存在する場合はこちらの処理は不要です。**
プロセス構築例で使用したtest_resultのデータに
Lot IDの列がなかったため、IDをランダムに割り当てています。
IDをランダムに割り当てる場合は以下のような設定になります。




①


**分割単位となる属性にbatchロールを設定します。**そうすることで、
Lot IDが2,3,0,1ある場合、1回目のループではLot ID 2がテストデータとなり、
残りのLot ID 3,0,1が学習データとなります。2回目のループではLot ID 3がテスト
データとなり、残りのLot ID 2,0,1が学習データとなり、順次処理されます。

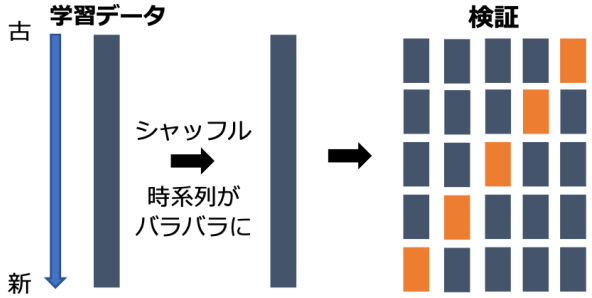
製造業において、ロット単位で管理されているケースがあるため、
新しいロットに対して一般化できるかどうか試したい時に使用できる可能性があります。

# Sliding Window Validation (時系列データに対する交差検証)

**時系列の前後関係を保持したまま、データを分割したい場合に用いられる**

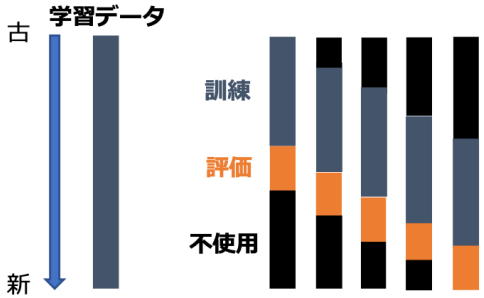
複数回検証するために交差検証がよく使用されるが、時系列データで交差検証を実施する場合は、時系列の前後関係を保持したままデータを分割する必要があります。

✓ 通常の交差検証の場合




- 未来の情報を含んだ学習データで学習し，過去のデータを含んだテストデータで検証
- 実際の運用とはかけ離れた検証になっている

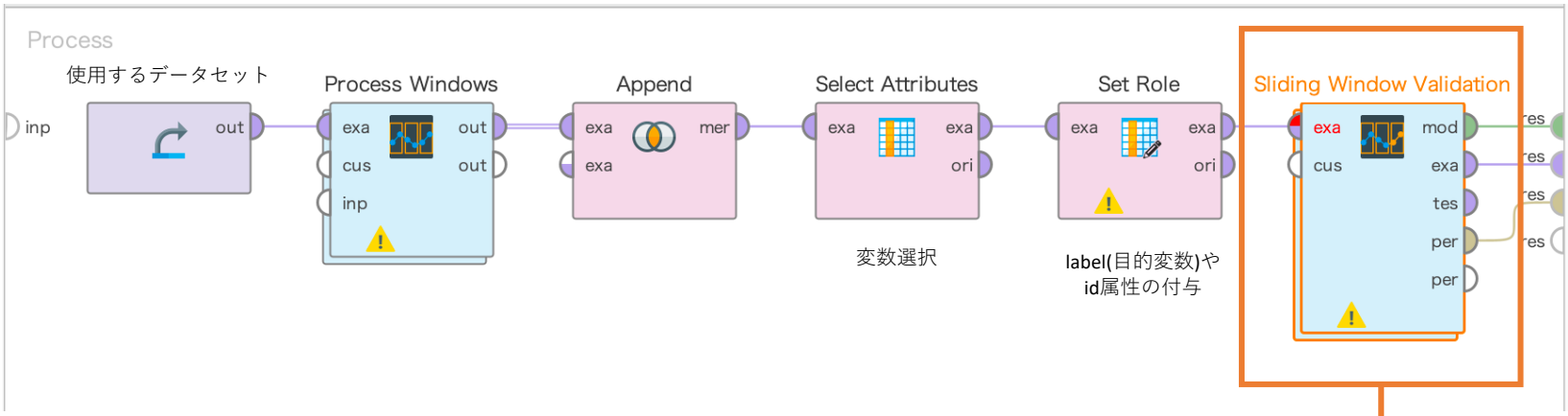
✓ Sliding Window Validationの場合

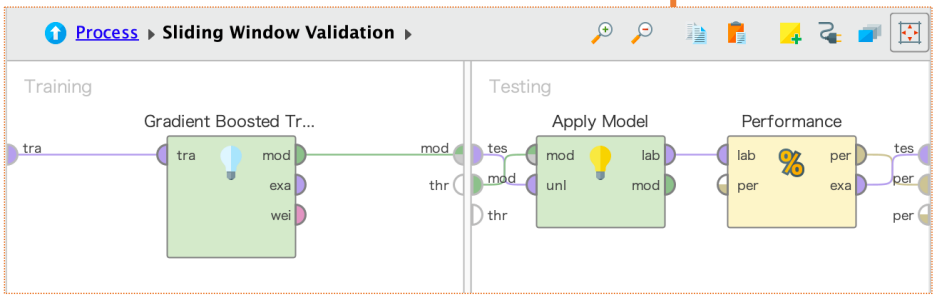



- 訓練データよりも評価データが未来になるようにデータを分割する
- 実際の運用と同様の設計

# Sliding Window Validation

▷プロセス構築例



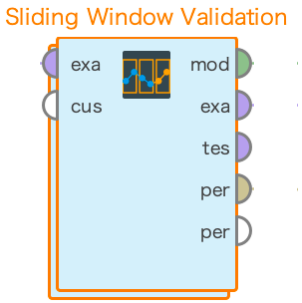



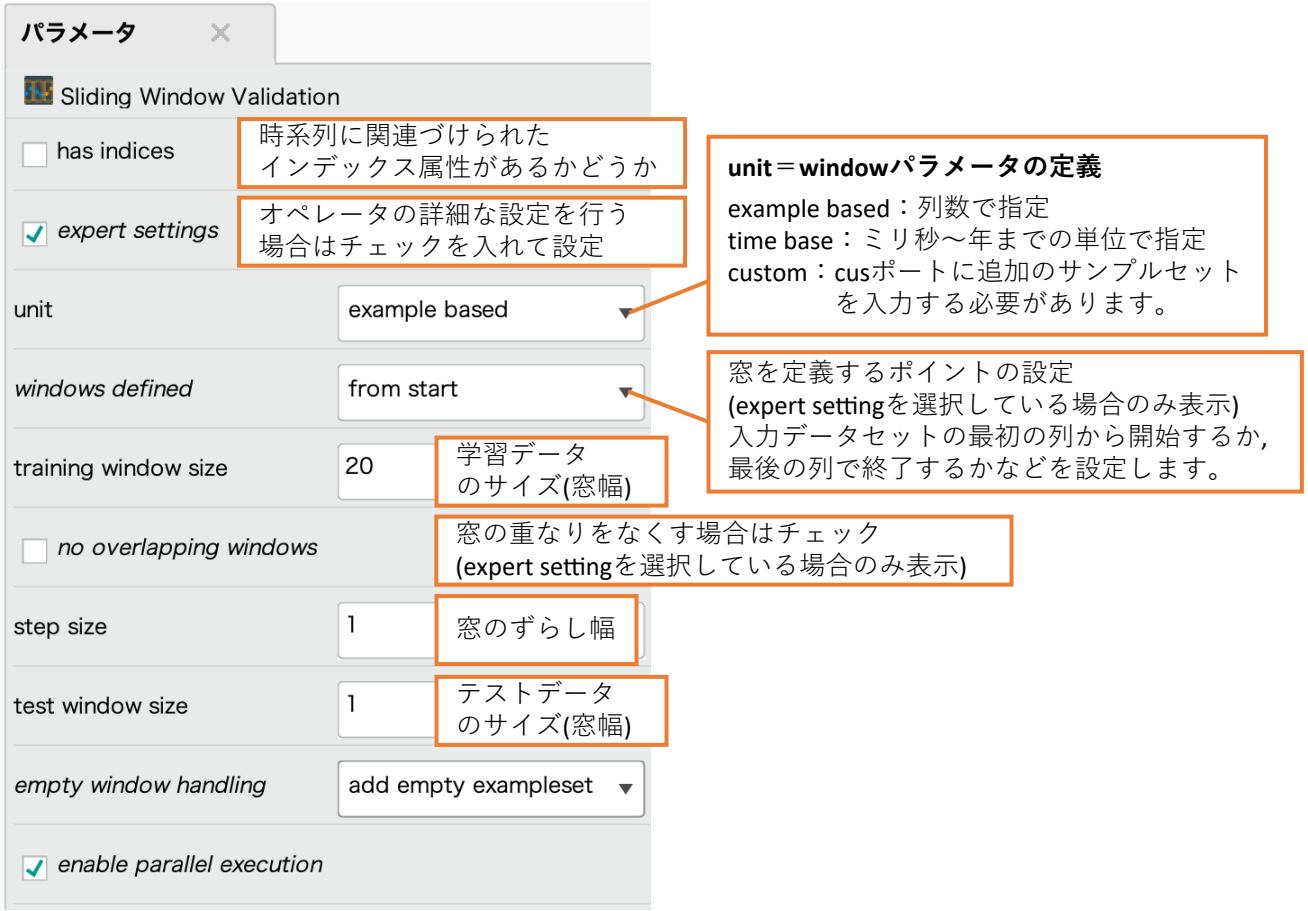

Time Seriesの**Sliding Window Validation オペレータ**を使用します。
（パラメータの設定については次ページに記載）

- Windowingや特徴量の作成を行ってから、時系列データに対して交差検証を行います。
- Forward Selectionを行う場合は、Cross Validationオペレータを使用する時と同様の構築方法で、Forward Selectionの中にSliding Window Validationを配置します。

# Sliding Window Validation

## ▷オペレータのパラメータ設定




例）1行= 7日間の集計、1ヶ月のデータを使って学習し、2週間のデータでテストする場合の設定
expert settings：チェック外す 、 unit：exmaple base 、 training window size：4 (1ヶ月)、 step size：1 、 test window size：2

# おわりに

4つの検証方法をご紹介しました。

手元にあるサンプル数が極端に少ない場合は、データを5分割するよりもできるだけ多くのデータを学習に使える方法を用いた方が良いと考えられます。データによって、分割方法を間違えると、テストデータや検証データに対する予測が正確に行われない可能性があります。状況に合わせて検証方法を使い分けていきましょう。

今回ご紹介した検証方法は、RapidMinerを使って簡単な設定で実行することができます。検証方法を変えて評価結果がどう変わるのかをぜひ試してみてください。