

RapidMiner9
Learn Tutorials
日本語ガイドブック

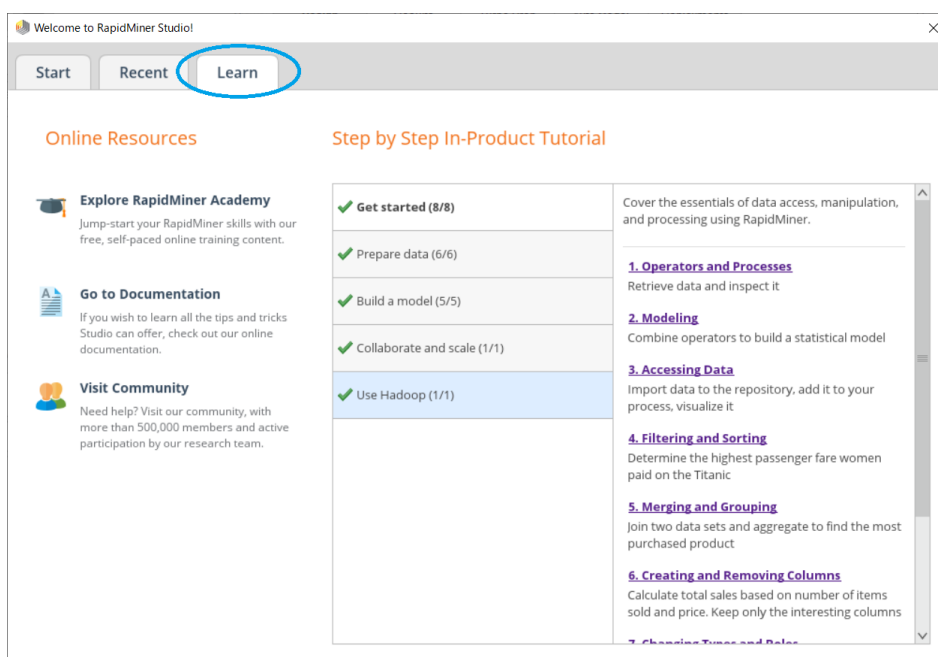
株式会社 KSK アナリティクス

目次

1. LEARN Tutorial の始め方	2
2. Get started コース	7
2.1 Get started コース概要	7
2.2 1. Operators and Processes	7
2.3 2. Modeling.....	13
2.4 3. Accessing Data	17
2.5 4. Filtering and Sorting	20
2.6 5. Merging and Grouping	25
2.7 6. Creating and Removing Columns.....	32
2.8 7. Changing Types and Roles.....	37
2.9 8. More Modeling	40
3. Prepare data コース	45
3.1 Handle Missing Values	45
3.2 Normalization and Outlier Detection	52
3.3 Pivoting and Renaming.....	58
3.4 Macros and Sampling	63
3.6 Writing Data.....	76
4. Build a model コース	79
4.1 Modeling.....	79
4.2 Scoring	83
4.3 Test Sprints and Validation.....	86
4.4 Cross Validation	90
4.5 Visual Model Comparison	96

1. LEARN Tutorial の始め方

RapidMiner9 をより本格的にお使い頂けるように、RapidMiner では簡単な LEARN Tutorial という練習コースをご用意しております。本資料は LEARN Tutorial の翻訳資料でありますので、実際にお手元を動かしながら RapidMiner を学んでいただくことが可能です。

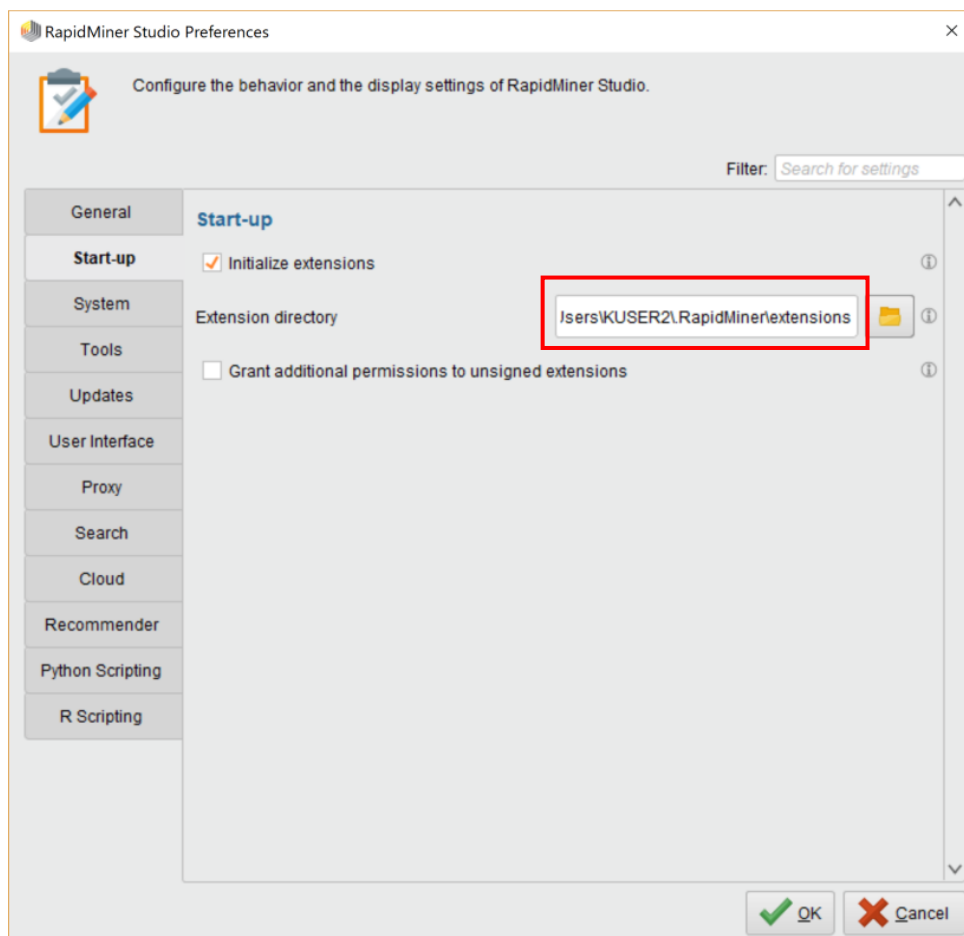


Tutorials の始め方は簡単です。RapidMiner9 を起動しますと、上のようなスタート画面が表示されます。その中の LEARN をクリックしますと”Get started”, ”Prepare data”, ”Build a model”が選択できますので、それぞれのコースを進めることができます。

RapidMiner9 では、下記手順に従って英語表示を日本語に切り替えることができます。
jar ファイル(rapidminer-extension-language-pack-beta-0.9.0-all.jar)を使用して RapidMiner Studio を日本語化します。

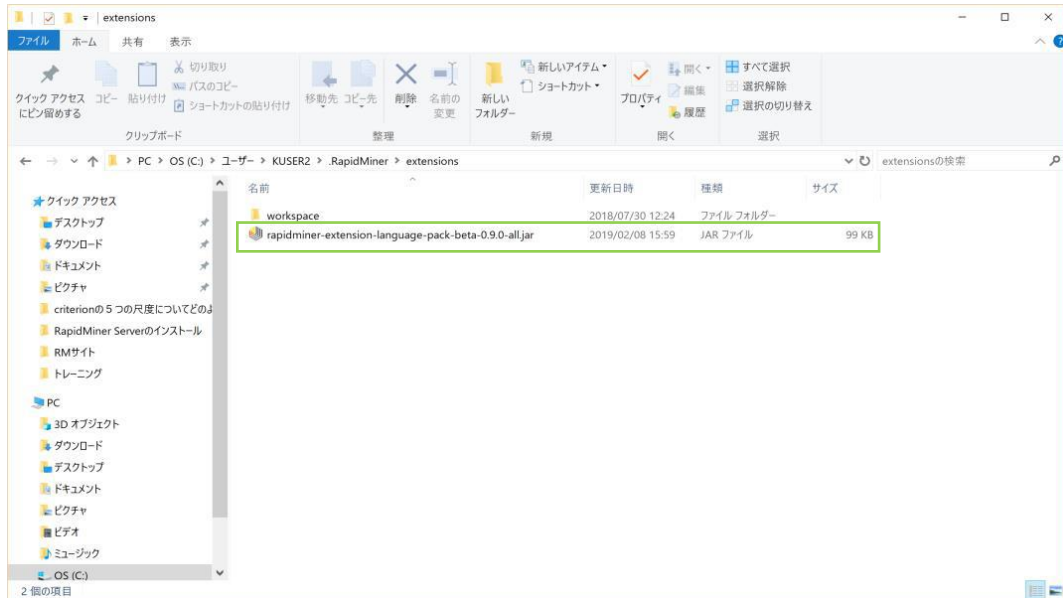
英語表示から日本語表示への切り替え手順

1. jar ファイルを配置する場所を確認します。RapidMiner の Preference を開き、Start-up タブの【Extension directory】に設定されているパスを確認してください。(デフォルトでは “C:\%Users%\<ユーザー名>\.RapidMiner\extensions” になっています)

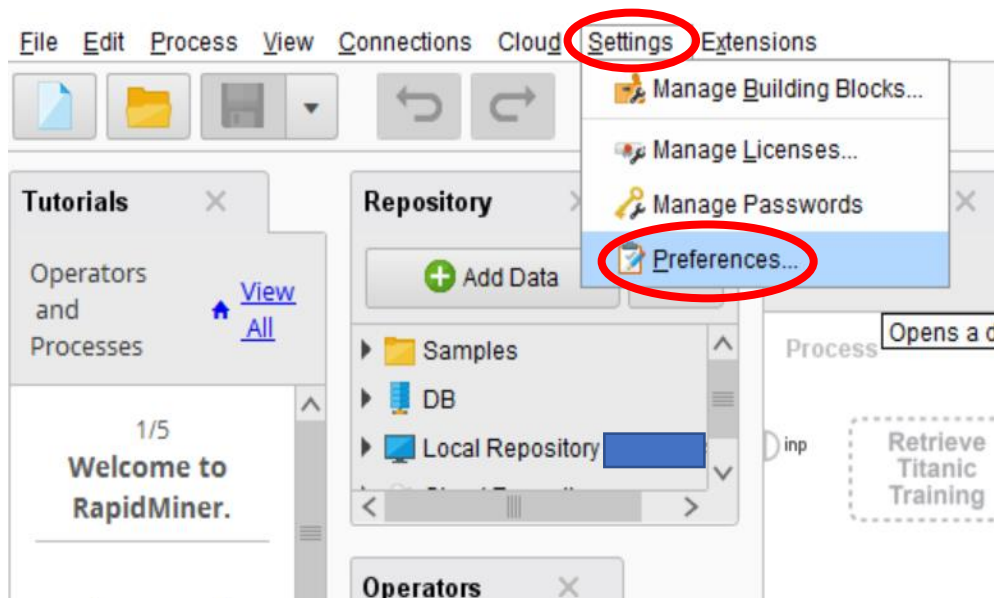


- 1 で確認した場所に jar ファイル(rapidminer-extensionlanguage-pack-beta-0.9.0-all.jar)を配置します。

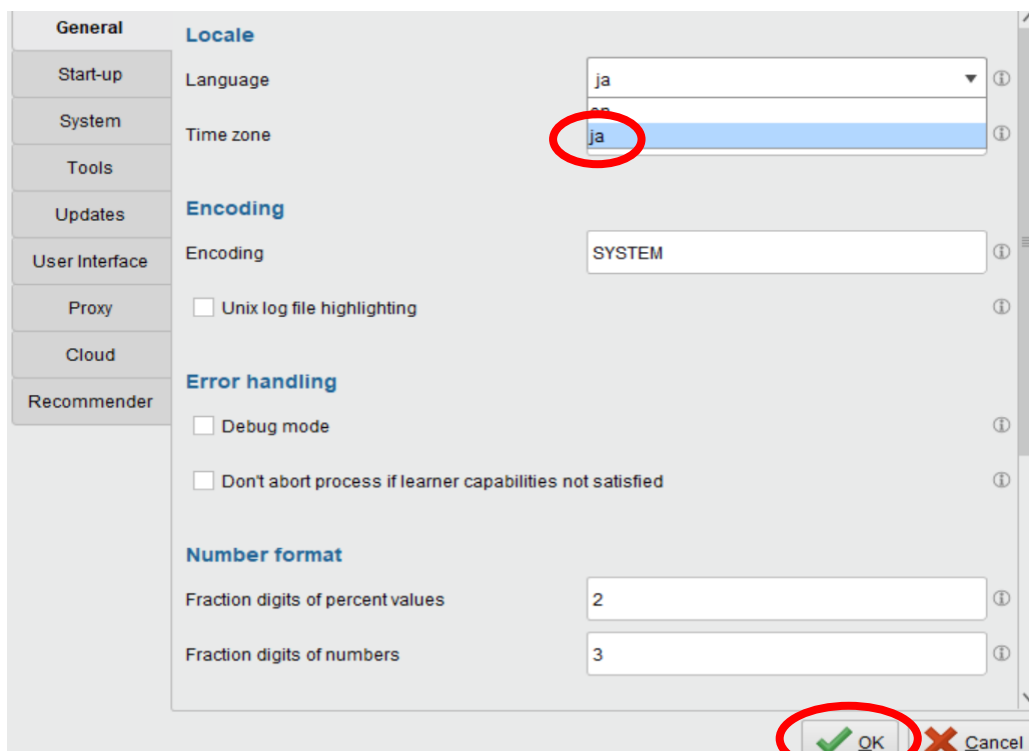
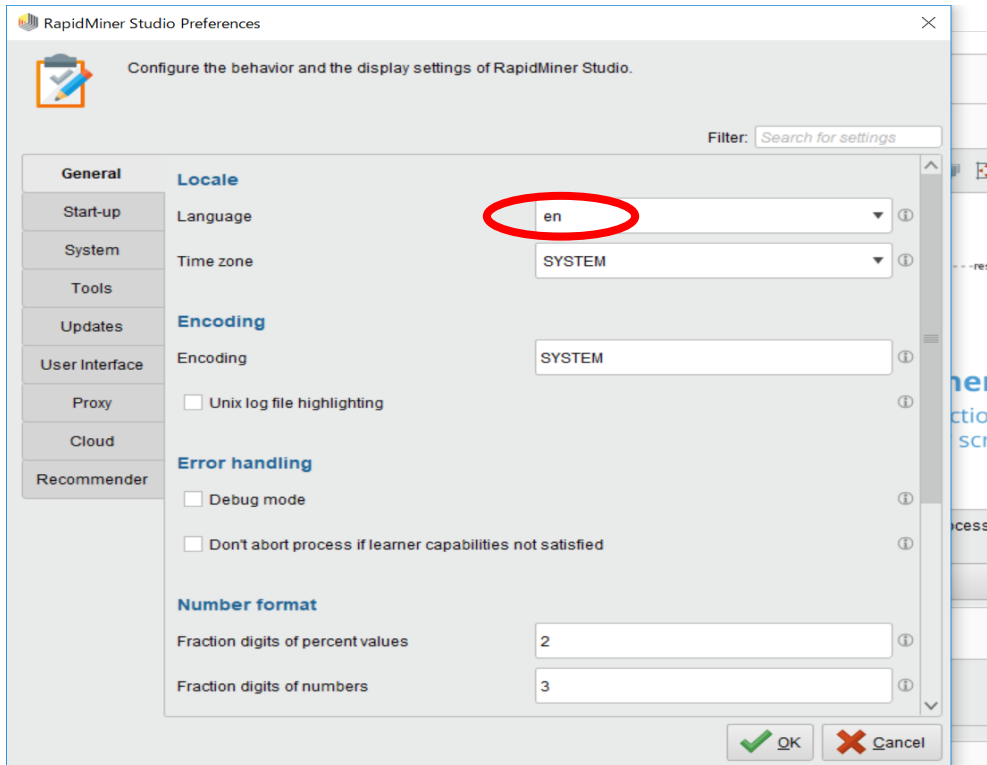
配置後に Rapidminer を再起動してください。



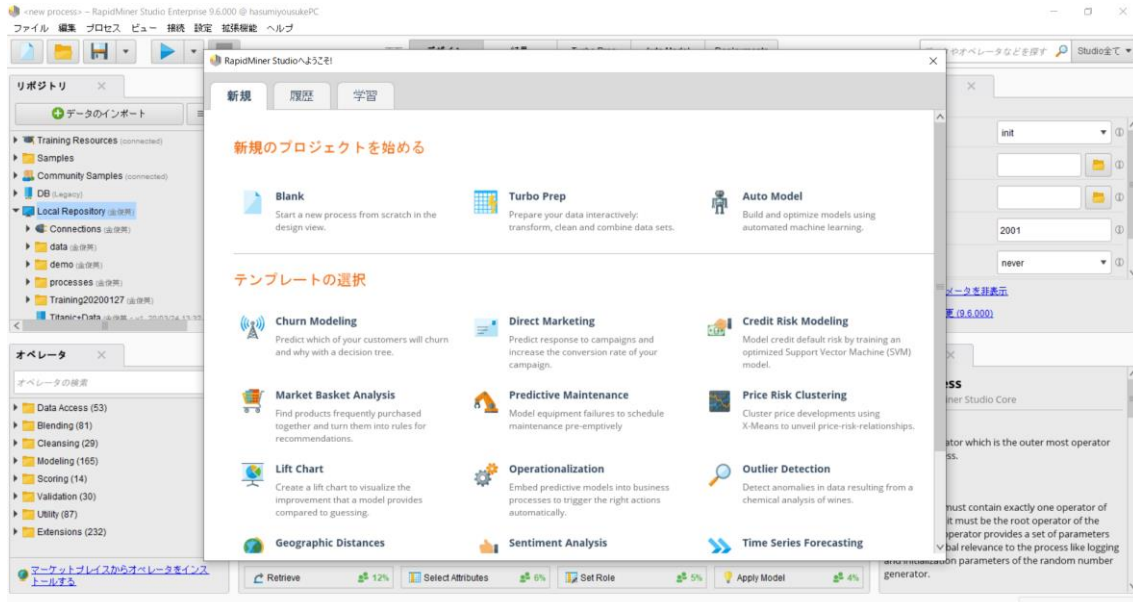
3. Rapidminer のメニューバーから、Settings>Preferences...を選択し、Preferences (設定) 画面を開く。



4. 開いた画面の General タブの中に Locale>Language がありますので、右の選択バーを押し、en(英語)→ja(日本語)へ変更し、OK を押します。



5. すぐに日本語へ切り替わらないため、プログラムを閉じ、再起動させます。
6. 再起動をすると、日本語への切り替えが完了します。



2. Get started コース

2.1 Get started コース概要

Get started コースを開始するには Tutorial のメニューから Get started を選択します。Get started コースには8つのトレーニングが用意されており、簡単に概要を説明します。

1. Operators and Processes
データを取り込み、内容を確認します。
2. Modeling
分析モデルを構築し実行します。
3. Accessing Data
リポジトリにデータを取り込み、可視化させます。
4. Filtering and Sorting
タイタニック号での高生存者を分類します。
5. Merging and Grouping
二つのデータセットを統合して集計し、最も購入された商品を見つけます。
6. Creating and Removing Columns
重要な列のみを残して、販売した商品と価格の総売上を計算します。
7. Changing Types and Roles
項目の形式変更が必要なデータを用意して、予測モデルの為の分析ターゲットを定義します
8. More Modeling
タイタニック号の悲劇での生存者を性別、船室等級、家族構成から分析します。

2.2 1. Operators and Processes

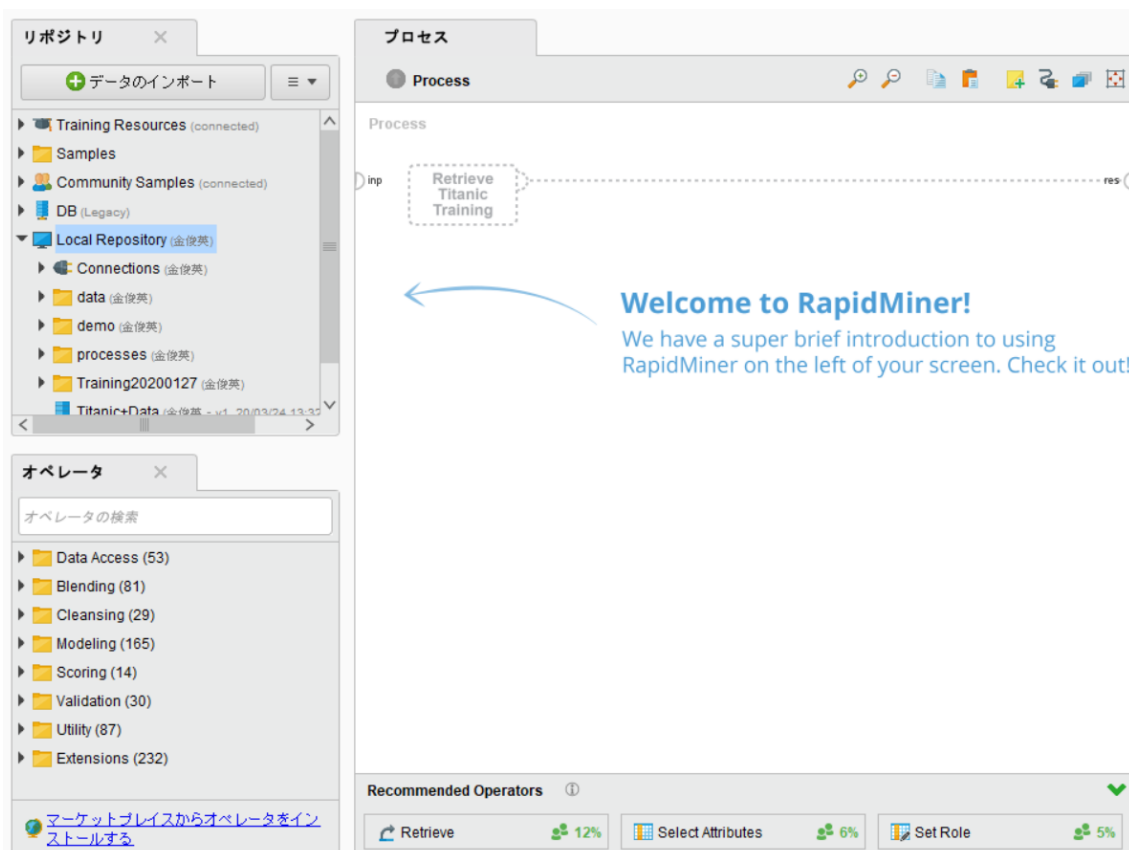
□ステップ 1/5

RapidMiner へようこそ！

RapidMiner は多くの機能を有しており、これから数分間、RapidMiner Studio を使い、データアクセスやデータ変換、統計モデルの構築といった基本的なデータサイエンスのテクニックを学びます。使用するのはタイタニック号の乗客データです。

EXPLANATION(説明)

各チュートリアルは、RapidMiner Studio の基本を学ぶため、いくつかの手順に分かれています。各手順では、実行すべき具体的なアクションと、そのアクションが重要である理由の説明も行います。プロセスパネルの点線プレビューに注目してください。これは、チュートリアルで作成するプロセスを示しています。



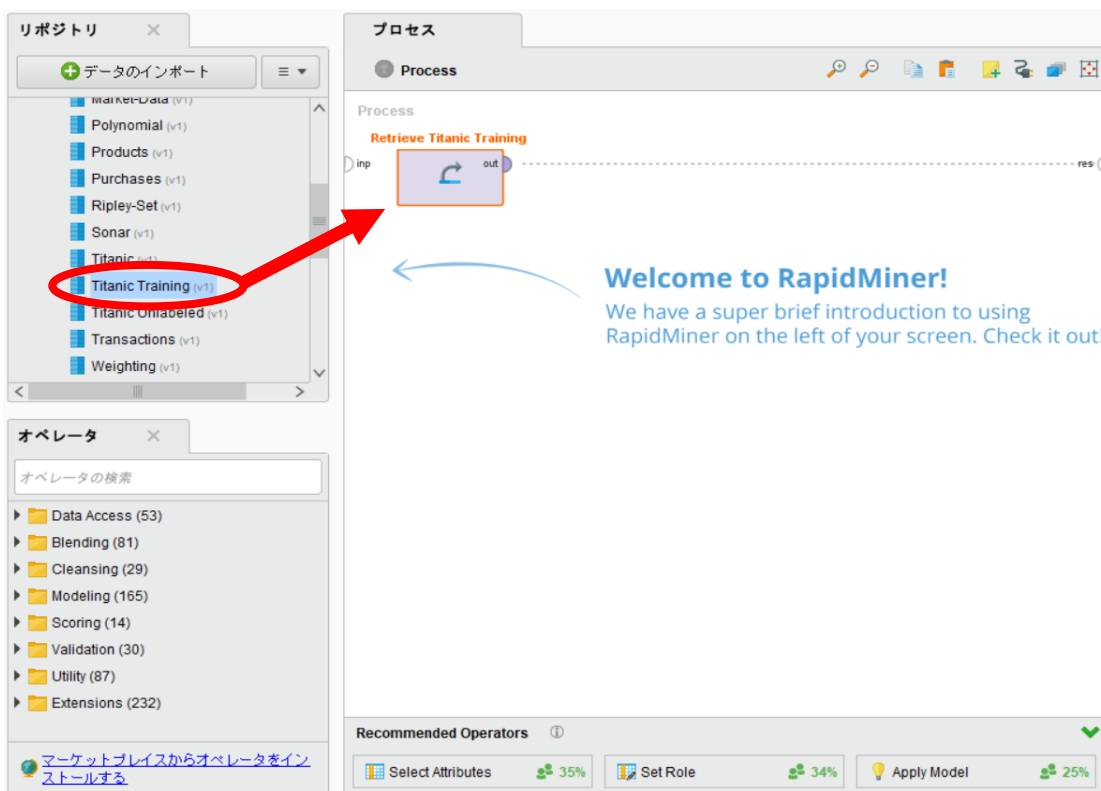
□ステップ 2/5

データの取り込み

それではタイタニック号の乗船者データを取り込んでみましょう。

ACTIVITY(アクティビティ)

1. 左側にあるリポジトリの欄を参照します
2. Samples フォルダの中の data フォルダを開きます
3. “Titanic Training”のデータセットをプロセスエリアにドラッグ&ドロップします。



EXPLANATION(説明)

よくできました！ RapidMiner に初めてのオペレータ、すなわち Retrieve オペレータを追加しました。オペレータはアクションの実行、この場合はリポジトリからデータを取り込みます。では、これで何が出来るのか見てみましょう！

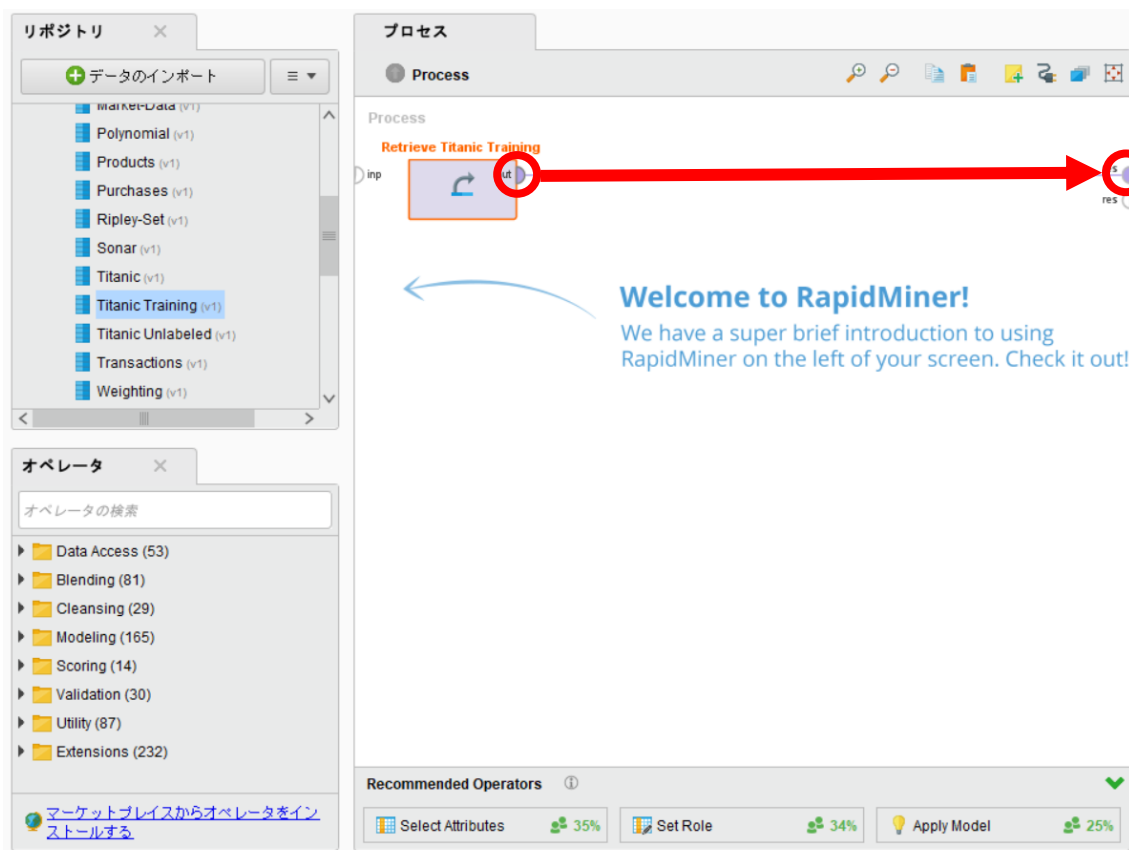
□ステップ 3/5

はじめてのプロセス構築

RapidMiner ではオペレータを加え、接続することでプロセスを作成します。オペレータはポートによって相互に接続されます。

ACTIVITY(アクティビティ)

1. Retrieve Titanic Training の出力ポート ("out") をプロセスパネルの右側にある結果ポート ("res") に接続します。
2. ポート間の線をドラッグするか、最初に片方のポートをクリックしてからもう片方のポートをクリックして接続します。



EXPLANATION(説明)

素晴らしい！ RapidMiner Studio で初めてプロセスを構築しました。これで Retrieve オペレータの出力結果を見ることが出来るようになりました。オペレータの結果を見たいときは、res ポートに接続されているか確認してください。

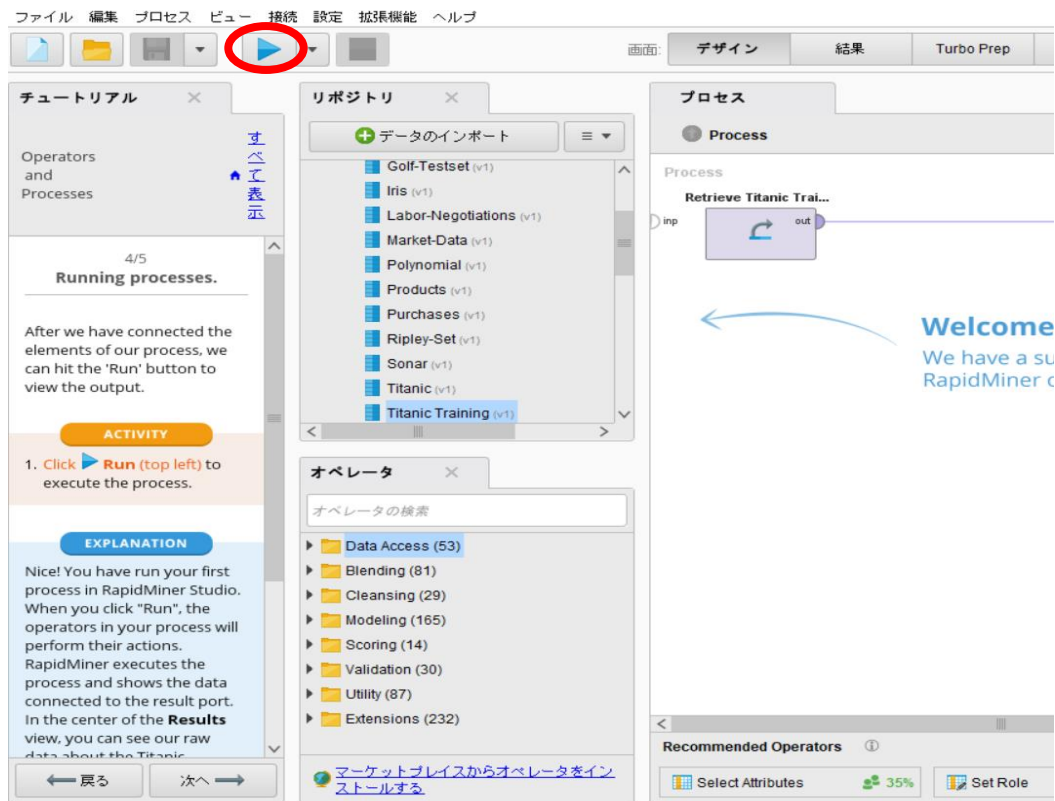
ロステップ 4/5

プロセスの実行

プロセスの接続が完了したので、実行ボタンを押して結果を出力することができます。

ACTIVITY(アクティビティ)

実行ボタン(画面上部の青い矢印)をクリックして、プロセスを実行することができます。



EXPLANATION(説明)

いいですね！ RapidMiner Studio で最初のプロセスを実行することが出来ました。実行ボタンをクリックするとオペレータの内容が実行されます。RapidMiner でプロセスを実行すると"result"ポートに接続されたデータが表示されます。結果ビューの中央にはタイタニック号乗船者の家族構成や年齢などの生データが表示されています。基本統計量(Statistics)タブをクリックすると統計の概要や有益な情報が表示されます。例えば今回のデータだとタイタニック号の生存者は 349 人になっています。(次のページの画像参照)

異なる結果になっていた場合は、正しいデータセット(Titanic Training)を使っているか確認してください。

属性名	データ型	欠損値	フィルタ (7/7 属性):	属性の検索
<input checked="" type="checkbox"/> Survived	Binominal	0	<input checked="" type="checkbox"/> Yes (349)	最小値 No (567)
<input checked="" type="checkbox"/> Age	Real	0	最小頻度値 0.166700000	最大値 80
<input checked="" type="checkbox"/> Passenger Class	Polynomial	0	最小頻度値 Second (184)	最大値 Third (491)
<input checked="" type="checkbox"/> Sex	Binominal	0	最小頻度値 Female (322)	最大値 Male (594)
<input checked="" type="checkbox"/> No of Siblings or Spouses on B...	Integer	0	最小値 0	最大値 8
<input checked="" type="checkbox"/> No of Parents or Children on B...	Integer	0	最小値 0	最大値 9
<input checked="" type="checkbox"/> Passenger Fare	Numeric	0	最小値 0	最大値 512.329200000

1 - 7 属性を表示中 行: 916 特別属性: 1 普通属性: 6

□ステップ 5/5

おめでとう、半分まで来ました！

よく出来ました！最初のチュートリアルは完了です。内容を簡単に振り返ってみましょう。

- ・オペレータはそれぞれのアクションを実行します。
- ・オペレータ同士を接続することで、プロセスを構築することができます。
- ・"res"ポートに接続すると、オペレータの実行結果を見ることができます。
- ・プロセスを実行するとすべてのオペレータが実行され、結果が表示されます。

EXPLANATION(説明)

さらに RapidMiner をマスターするには、次のチュートリアルに進みましょう！

2.3 2. Modeling

□ステップ 1/5

さあデータ分析を行きましょう！

前のチュートリアル最後では、データセットを取り込んでプロセスを実行しました。その結果タイタニック号事件の生存者は349人であることが分かりました。それでは生存者に何か共通点があるのかを発見するプロセスを構築してみましょう。

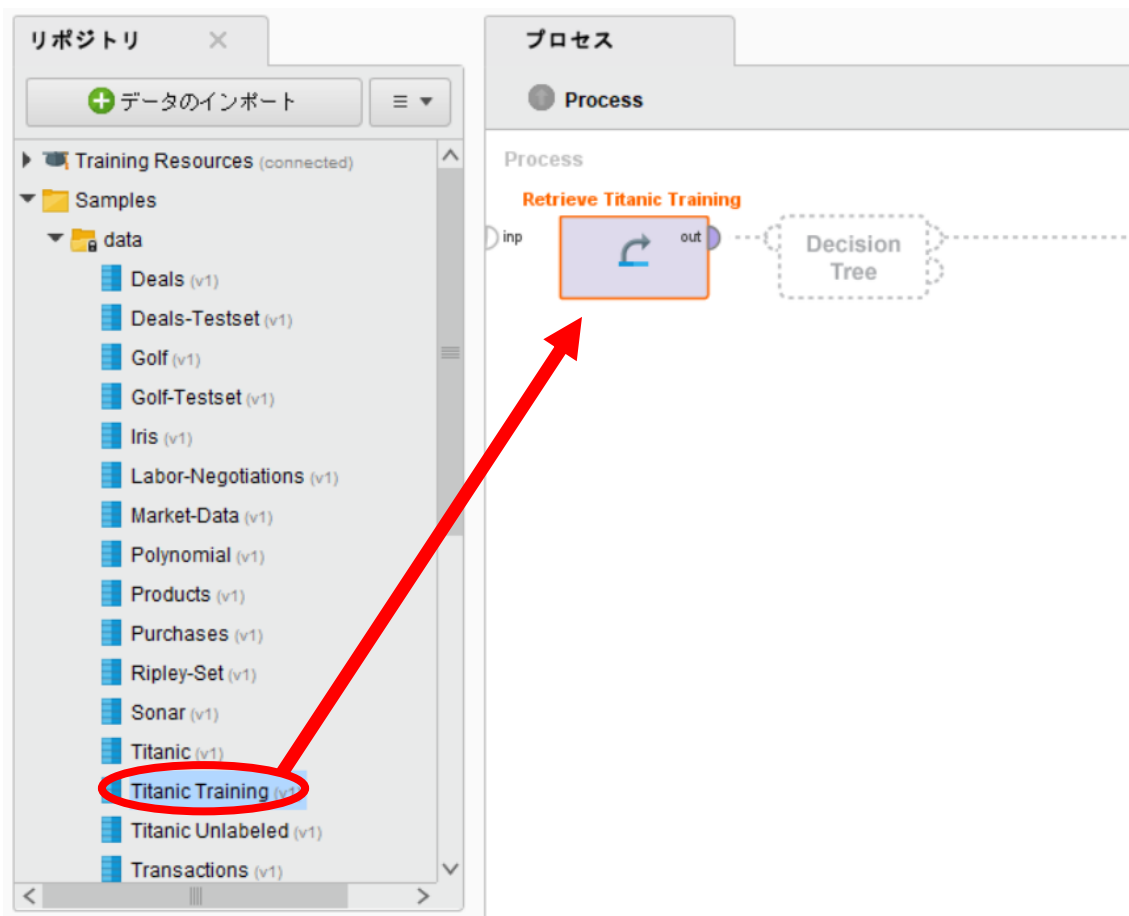
□ステップ 2/5

タイタニック号のデータ取り込み

タイタニック号事件のデータを読み込みましょう。

ACTIVITY(アクティビティ)

リポジトリの Samples>data から”Titanic Training”データをプロセスパネルにドラッグします。



EXPLANATION(説明)

プロセスの最初の操作を定義しました。ここからこのデータを使い、機械学習モデルを構築することが、いかに簡単なことか学んでいきましょう。

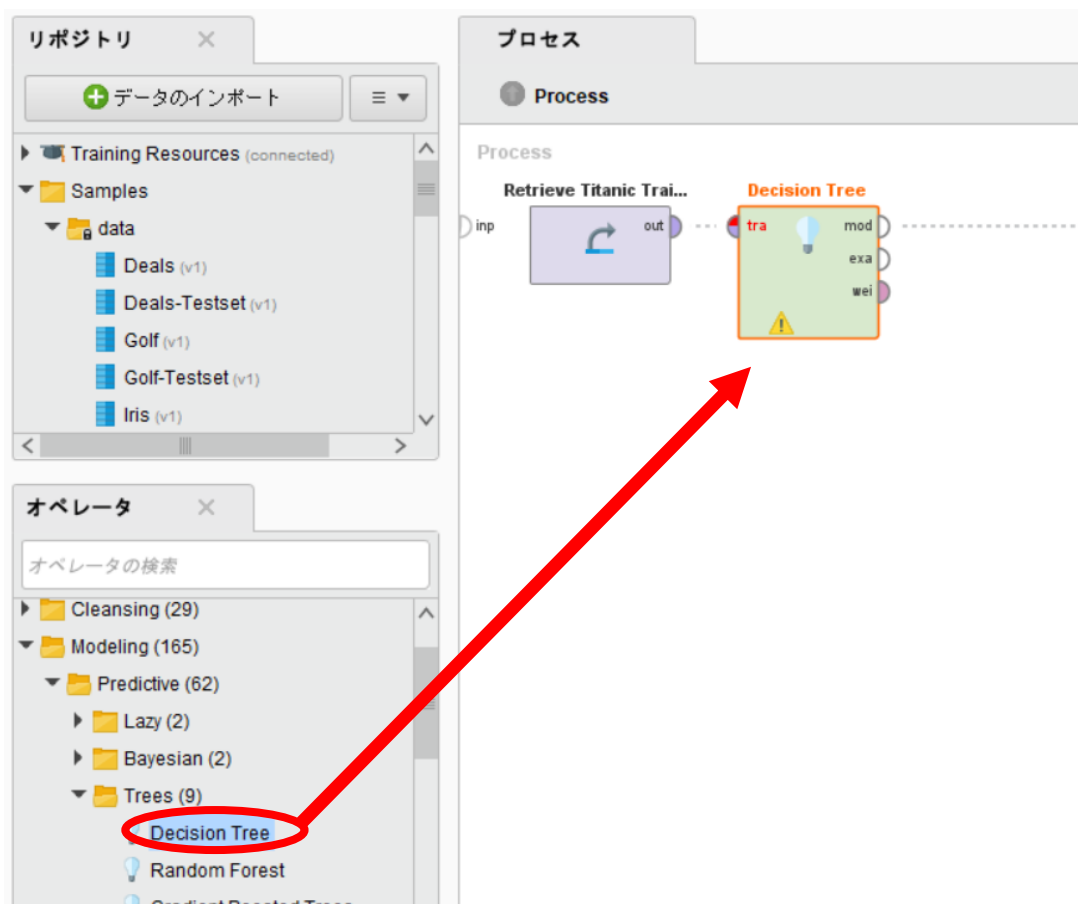
ロステップ 3/5

決定木モデルの構築

決定木モデルは、データの中に隠れたパターンを発見するのに使われる統計モデリング手法です。まずは作ってみましょう。

ACTIVITY(アクティビティ)

1. 左側のオペレータ欄に注目します。
2. フォルダを Modeling>Predictive>Trees の順番で開きます。(または検索欄で"tree"と入力します)
3. "Decision Tree" オペレータをプロセスの"Retrieve Titanic Training" オペレータの右側にドラッグ&ドロップします。



EXPLANATION(説明)

オペレータをプロセスに配置しましたが、実行する前にお互いを接続し、出力として表示するものを定義する必要があります。

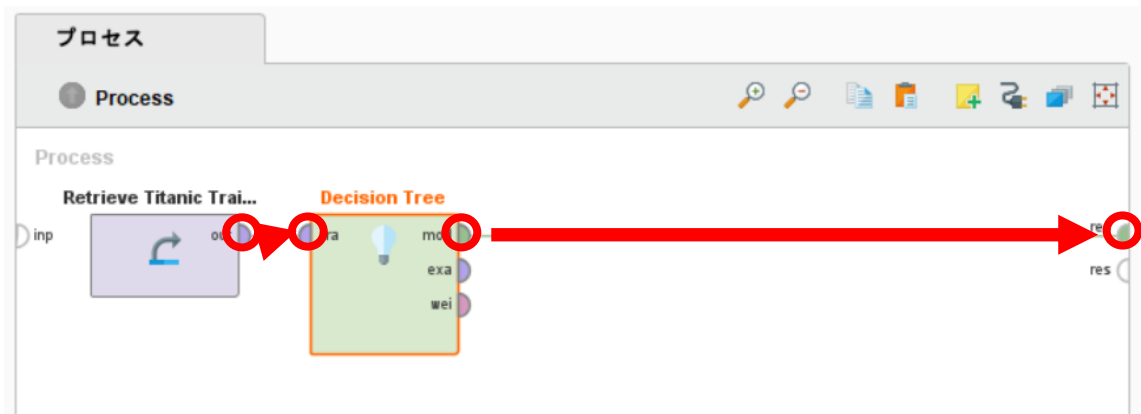
□ステップ 4/5

接続と実行

“Decision Tree”オペレータは“Titanic Training”データを元に決定木モデルを構築してくれます。実行するためにはそれぞれのオペレータを接続する必要があります。

ACTIVITY(アクティビティ)

1. Retrieve Titanic Training の出力ポート (“out”) と Decision Tree の入力ポート (“tra”は “training ”の意味です) を接続します。ポートとポートのクリックか、ドラッグでも出来ます。



2. Decision Tree の最初の出力("mod")をプロセスパネルの右側にある結果ポート("res")に接続します。
3. 実行ボタンを押し、プロセスを実行します。

ファイル 編集 プロセス ビュー 接続 設定 拡張機能 ヘルプ



EXPLANATION(説明)

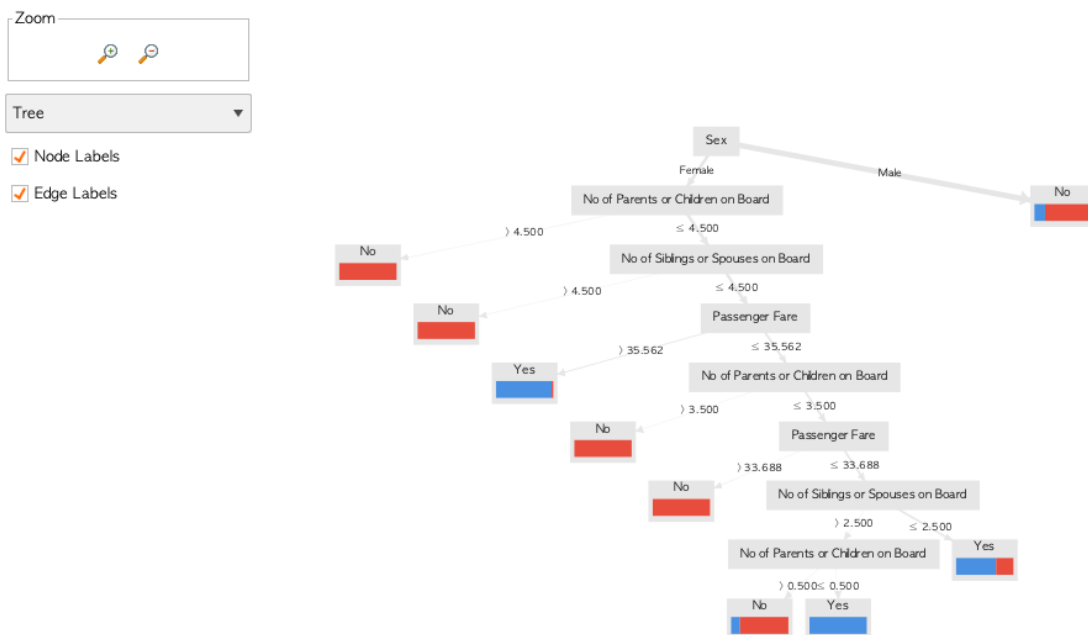
よくできました。これでプロセスは完了し、生存者と犠牲者、それぞれの大半に共通していたものを説明する決定木モデルを提供できます。オペレータポートとディビジョンツリーの説明をもっと読みたい場合は、チュートリアル画面で各項目をクリックしてください。

□ステップ 5/5

このステップのまとめ

おめでとうございます！初めての機械学習のモデルを構築する事ができました。きっと難しくなかったと思います。ここではデータの取り込み、機械学習モデルの構築、プロセスの実行を行いました。

※バージョンによって、作成されたモデルは多少異なることがあります。


EXPLANATION(説明)

決定木モデルの結果から、生存の分かれ目は偶然ではなかった事が明確になりました。実際、家族が少なく、少なくとも高価なチケットを持っている女性客は本当に幸運だったことが分かります。

このチュートリアルで紹介しているのは、RapidMiner で出来る事の氷山の一角に過ぎません。RapidMiner をより深く学び続けるには、次のチュートリアルにさっそく進みましょう。

RapidMiner へようこそ！

2.4 3. Accessing Data

□ステップ 1/6

RapidMiner へのデータ取り込み

データを RapidMiner に取り込むことは分析業務の最初のタスクとしてよく行われます。このチュートリアルでは、あるデータを RapidMiner の"リポジトリ"と呼ばれる中心ストレージに取り込む方法を学びます。引き続きタイタニック号事件のデータを扱っていきませんが、今回は Excel ファイルの状態から取り込んでみたいと思います。

EXPLANATION(説明)

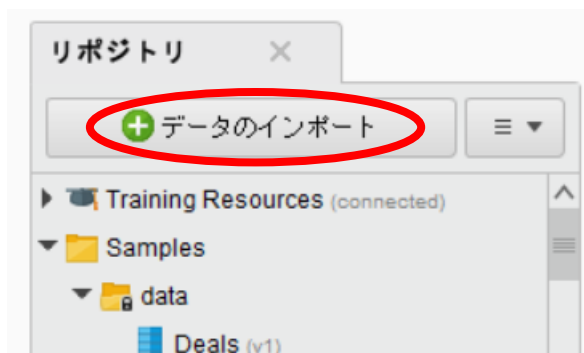
ここでは、データの取り込みから始まる分析プロセス構築の中で重要なステップを説明していきます。この後のチュートリアルではデータの準備とモデリングについて説明します。また、これまでに見たステップのいくつかについても、より詳細に説明します。

□ステップ 2/6

データのダウンロード

ACTIVITY(アクティビティ)

1. リンクから Excel ファイルをダウンロードします。
2. RapidMiner にダウンロードしたファイルをインポートするには、リポジトリの[データのインポート](Import Data)をクリックし、表示されるウィザードに従います。



3. インポート設定が完了すると、リポジトリの"Local Repository"にタイタニック号のデータが格納されます。

EXPLANATION(説明)

リポジトリパネルはデフォルトでは左上隅に配置されており、データやプロセス、結果などが保存される場所です。特に XLS や CSV といったファイルを扱う場合には常に一旦リポジトリにデータを取り込む必要があります。RapidMiner のリポジトリにはデータと共にメタデータも保存されますので、分析プロセスの設計が大幅に簡略化されます。

□ステップ 3/6

プロセスへのデータ追加

ACTIVITY(アクティビティ)

1. 上のデザインタブをクリックするとプロセスパネルに戻ります。
2. リポジトリから先ほど取り込んだタイタニック号のデータをプロセスにドラッグします。

EXPLANATION(説明)

リポジトリからプロセスにデータをドラッグすると、"data-loading"オペレータに変換されます(この場合は"Retrieve Titanic"です)。データはプロセスを実行するかオペレータの出力ポートからデータを流さない限り、実際にはデータは読み込まれません。

□ステップ 4/6

接続を完了し、結果を見る

ACTIVITY(アクティビティ)

1. "Retrieve Titanic"の出力ポートと、プロセスパネル右側の"res"という結果ポートとを接続します。
2. ポート間の線をドラッグして接続するか、最初に片方のポートをクリックしてからもう片方のポートをクリックして接続します。

EXPLANATION(説明)

プロセスの実行後、右側の結果ポートに接続されているデータのみ見ることができます。もし一つも結果ポートにプロセスを接続していない場合には、実行ボタンを押しても結果は表示されません。

□ステップ 5/6

プロセスの実行

ACTIVITY(アクティビティ)

実行ボタン(画面上部バーの青い矢印)をクリックしてプロセスを実行しましょう。

EXPLANATION(説明)

一度実行すれば、自動的に結果ビューに切り替わります。覚えているでしょうか？この結果はプロセスパネル右側の結果ポートに接続したプロセスの情報が出力されています。プロセスパネルに戻りたい場合には、上部のデザインタブをクリックします。

□ステップ 6/6

ステップの総括 - おめでとうございます！

初めてのデータ取り込みはこれで完了です。今後、各チュートリアルはいくつか追加質問がついています。スキルをさらに向上するために次の課題に答えてみましょう！

Challenge(追加質問)

- ・ 出力されたデータの中を見てください。どうやって欠損値を見つければ良いでしょうか。
- ・ 基本統計量タブはデータの要約を示しています。ファーストクラスに乗った人は何人でしたか？またタイタニック号事件で助からなかった人は何人ですか？
- ・ 何種類かのチャートで見てください。どうやって興味深いパターンを見つけられ良いでしょうか。

2.5 4. Filtering and Sorting

□ステップ 1/5

最も高い運賃を払った女性は、いくら払ったか？

先ほどのチュートリアルでは、データやモデル、プロセスなどの保存場所である RapidMiner のリポジトリへとデータを取り込む方法を学びました。このチュートリアルでは、タイタニック号のデータをフィルタリングして、女性乗客のみ見えるようにします。そして、女性の中で一番高い運賃を見つけるために簡単な並び替えを行います。男性に対して同じ問題を考える場合にも利用できます。運賃に違いがあると思いますか？

□ステップ 2/5

ワークフローへのデータ取り込み

ACTIVITY(アクティビティ)

リポジトリの Samples フォルダから、“Titanic”のデータをプロセスにドラッグします。

EXPLANATION(説明)

- ・ RapidMiner では行を *examples* と呼び、テーブルを *example sets* と呼んでいます。RapidMiner を通してこの単語を使うので、今すぐにでも覚える価値は十分あります。
- ・ 女性の中で一番高い運賃を見つける方法は幾つかあります。このチュートリアルではテーブルから男性データを取り除く、つまり *example sets* から *examples* をフィルターで取り除いていきます。

□ステップ 3/5

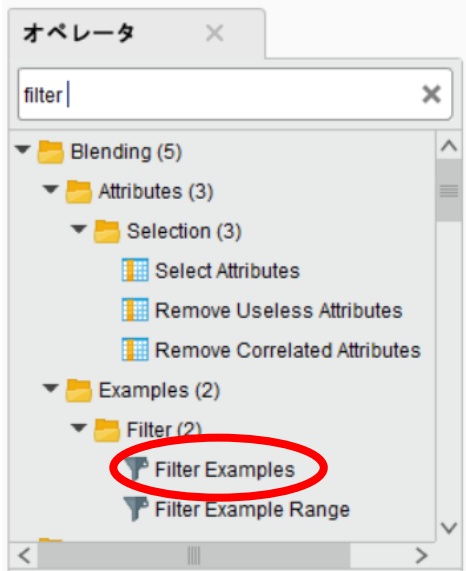
フィルタリングのセットアップ

EXPLANATION(説明)

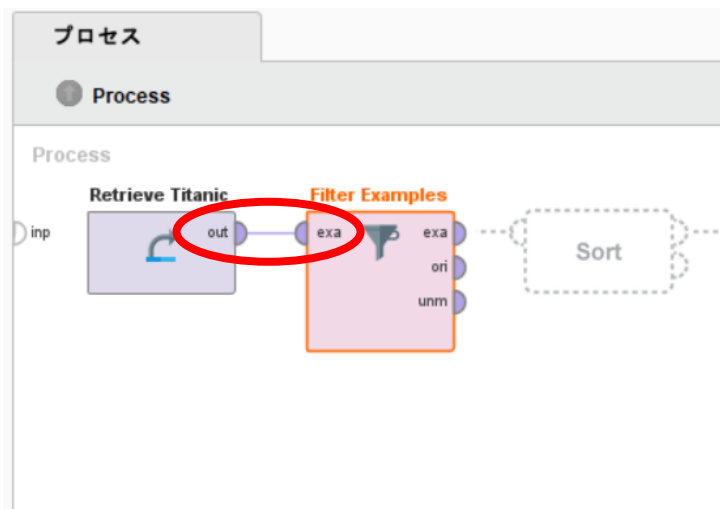
今回の場合は女性という、定義されたフィルター条件を満たす *examples*(行)のみを *examples sets*(テーブル)の中に残します。他のすべての行は削除します。

ACTIVITY(アクティビティ)

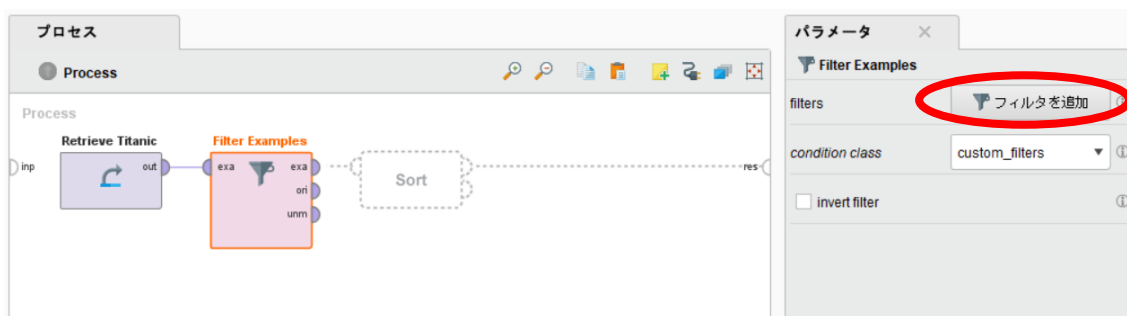
1. オペレータパネル上部の検索ボックスを利用して”Filter Examples”オペレータを探します。そして探してきたオペレータをプロセスパネルに配置します。



2. “Retrieve Titanic”のアウトポートポートと”Filter Examples”のインポートポートを接続します。



3. “Filter Example”オペレータをクリックし、フィルターの内容を定義するために、パラメータパネルの[フィルタを追加]をクリックします。



4. 左のボックスで Sex を選択し、真ん中のボックスはイコール(=)、そして最後のボックスには"Female"と入力を行います。入力する代わりに、隣の魔法の杖ボタンをクリックしてリストを表示させリストから Female を選ぶことも出来ます。



EXPLANATION(説明)

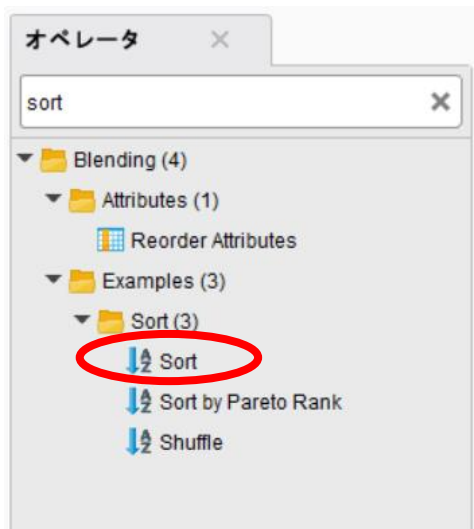
プロセスにオペレータを追加した時はすぐに接続するようにしてください。オペレータ同士の間にはデータが流れていますので、オペレータの接続設定はパラメータにも影響します。たとえば、データソースに繋がっていない状態では"Filter"オペレータが性別の列を見つけることはできません。

□ステップ 4/5

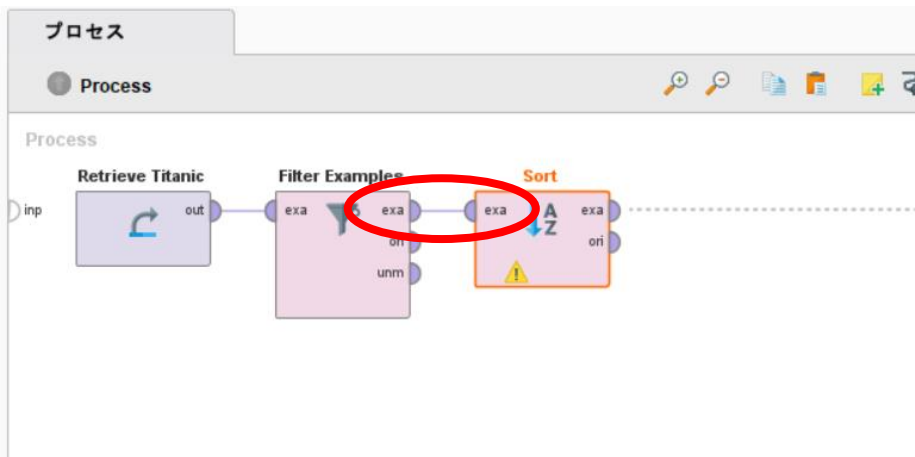
並び替え、一行目に最も高い運賃を表示させる

ACTIVITY(アクティビティ)

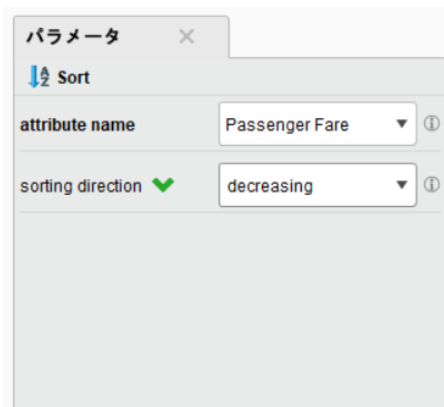
1. "Sort"オペレータを検索し、プロセスエリアにドラッグします。



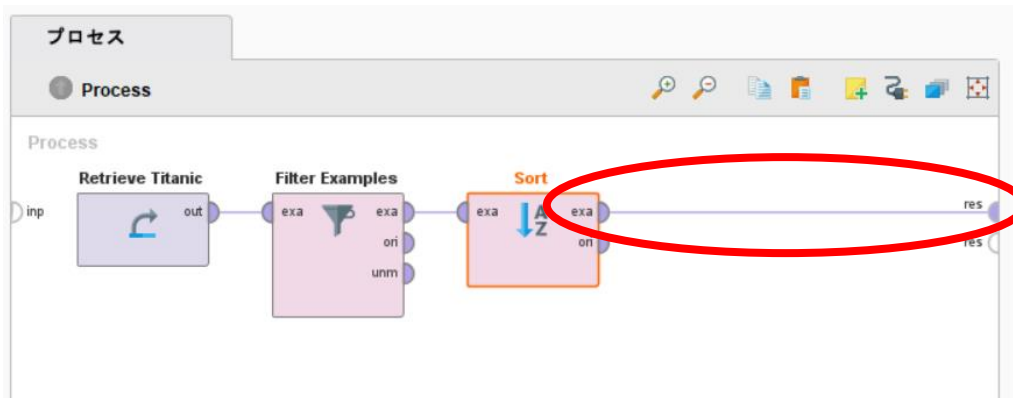
2. “Filter”オペレータの出力ポートと、“Sort”オペレータの入力ポートを接続します。



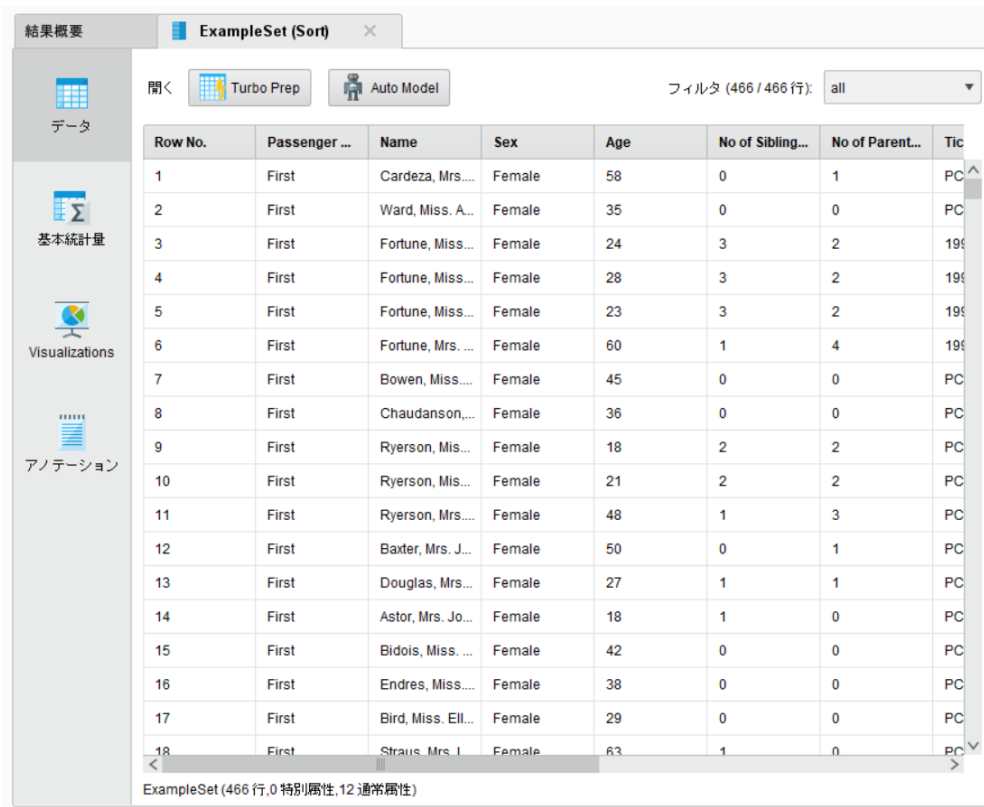
3. “Sort”オペレータをクリックし、パラメータパネルに注目します。
4. “attribute name”カラムをクリックし、“Passenger Fare”を選択します。
5. “Sorting Selection”カラムをクリックして、“decreasing”を選択します。



6. “Sort”パラメータの出力ポートと、プロセスパネルの”res”ポートとを接続します。



7. 結果ボタンをクリックしてプロセスを実行し、結果を見ます。



The screenshot shows the 'ExampleSet (Sort)' results viewer. It displays a table with 18 rows of passenger data. The columns are: Row No., Passenger..., Name, Sex, Age, No of Sibling..., No of Parent..., and Ticket No. (Tic). The table is filtered to show 466 rows.

Row No.	Passenger ...	Name	Sex	Age	No of Sibling...	No of Parent...	Tic
1	First	Cardeza, Mrs....	Female	58	0	1	PC
2	First	Ward, Miss. A...	Female	35	0	0	PC
3	First	Fortune, Miss...	Female	24	3	2	199
4	First	Fortune, Miss...	Female	28	3	2	199
5	First	Fortune, Miss...	Female	23	3	2	199
6	First	Fortune, Mrs. ...	Female	60	1	4	199
7	First	Bowen, Miss....	Female	45	0	0	PC
8	First	Chaudanson,...	Female	36	0	0	PC
9	First	Ryerson, Mis...	Female	18	2	2	PC
10	First	Ryerson, Mis...	Female	21	2	2	PC
11	First	Ryerson, Mrs....	Female	48	1	3	PC
12	First	Baxter, Mrs. J...	Female	50	0	1	PC
13	First	Douglas, Mrs...	Female	27	1	1	PC
14	First	Astor, Mrs. Jo...	Female	18	1	0	PC
15	First	Bidois, Miss. ...	Female	42	0	0	PC
16	First	Endres, Miss....	Female	38	0	0	PC
17	First	Bird, Miss. Ell...	Female	29	0	0	PC
18	First	Straus, Mrs. J...	Female	63	1	0	PC

ExampleSet (466 行, 0 特別属性, 12 通常属性)

EXPLANATION(説明)

ほとんどのオペレータは、オペレータの動作内容を定義する設定があります。オペレータをクリックし選択した後、プロセス画面の右側にあるパラメータパネル上で見ることができます。

□ステップ 5/5

ステップの総括 - おめでとうございます！

これでタイタニック号の女性乗客の中で、一番高い運賃がわかりました！

” Passenger Fare ” の列で一番先頭の値がそうです。

Challenge(追加質問)

男性の中で最も高い運賃を表示するように変更できますか？それは女性の時の金額と異なりますか？

2.6 5. Merging and Grouping

□ステップ 1/6

二つの新しいデータとの出会い

タイタニック号からは一旦離れて、データ準備でよく使われる方法、特にマージやグループ化について勉強しましょう。組織が販売した商品のデータセットと、取引（どの顧客がどの商品を購入したかの情報）のデータセットの二つのデータセットを扱います。いずれも製品を購入した顧客に関する情報になります。これらのデータを統合した後は、最も頻繁に購入された商品やロイヤルカスタマーは誰かという質問に答えられるようになります。さあ、始めましょう。

□ステップ 2/6

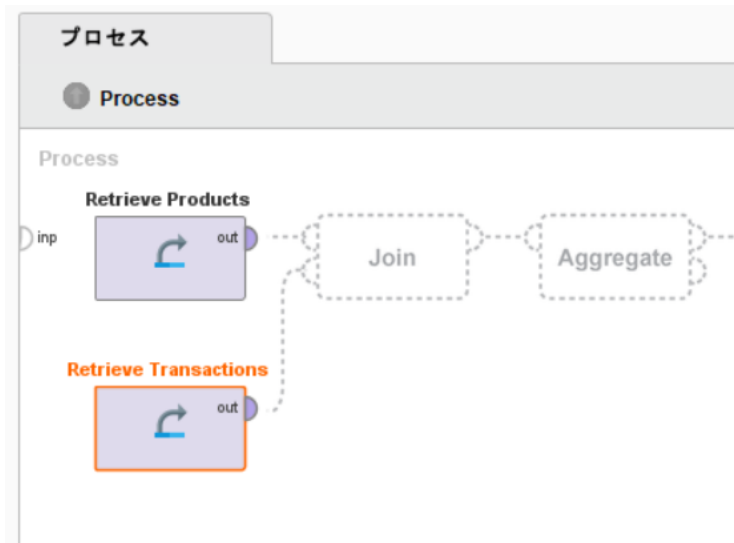
データを取得し、ワークフローに落とし込む

ACTIVITY(アクティビティ)

1. リポジトリパネルの Samples フォルダを展開します。次に data フォルダを展開し”Products”と”Transactions”のデータセットを見つけます。
2. この二つのデータセットをプロセスパネルにドラッグ&ドロップします。

EXPLANATION(説明)

RapidMiner では、この二つのデータ取得オペレータの内容は、プロセスが実行されるまでデータが読み込まれないことに注意して下さい。

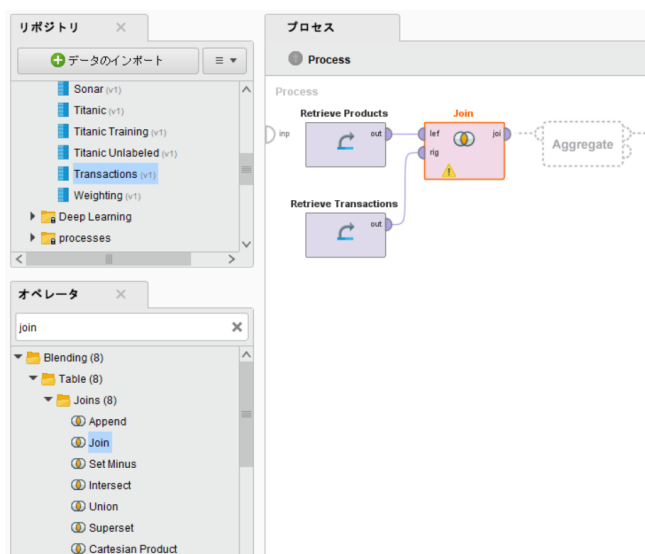


□ステップ 3/6

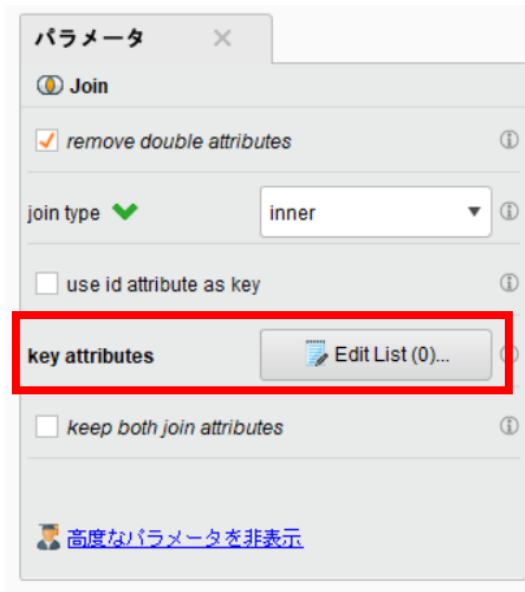
データの統合

ACTIVITY(アクティビティ)

1. オペレータパネルの検索ボックスから”Join”オペレータを検索し、プロセスパネルにドラッグします。
2. “Product”オペレータの out ポートと”join”オペレータの入力ポート(left/right どちらでも可)と接続します。
3. “Transactions”のオペレータを”join”オペレータのもう一つの入力オペレータに接続します。



4. “join”オペレータをクリックします。パラメータパネルの”key attribute”を見つけます。



5. “Edit List”をクリックします。” Product ID” を” left key” と” right key” の両方に設定し、apply をクリックし適用します。



EXPLANATION(説明)

- ・各オペレータのパラメータ設定を変更する前に、そのオペレータが接続されていることを確認して下さい。オペレータはどのデータが使えるのかが分からないので、接続しない限り”Product ID”を使用することはできません。

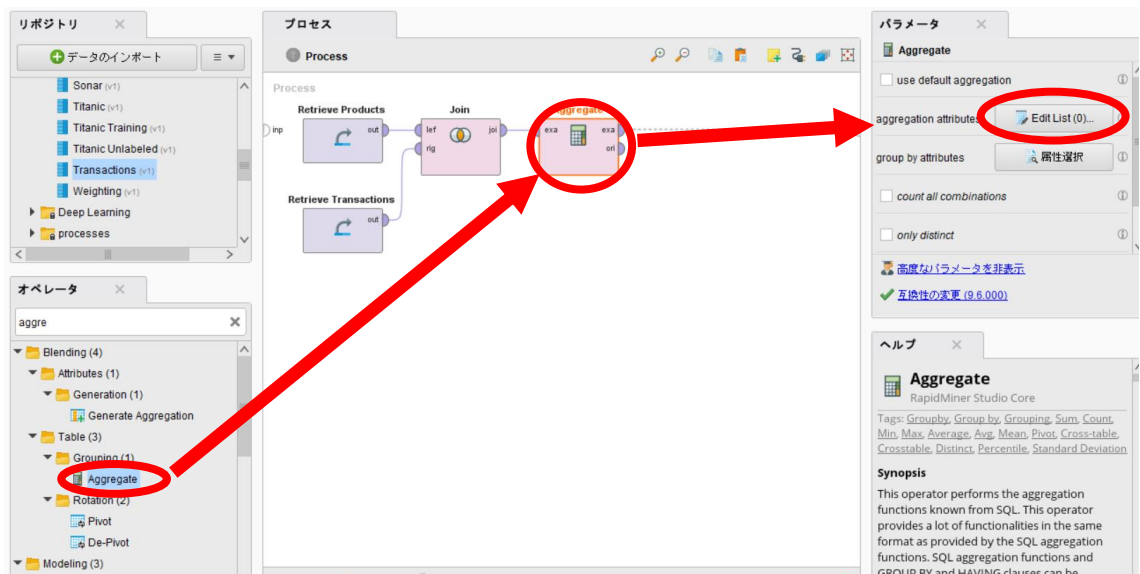
- ・ "Join"の結果、各取引とその商品の詳細テーブルが完成します。統合のキー属性として設定した2つの ID 列は、二つの元テーブルの行をそれぞれマッピングします。

□ステップ 4/6

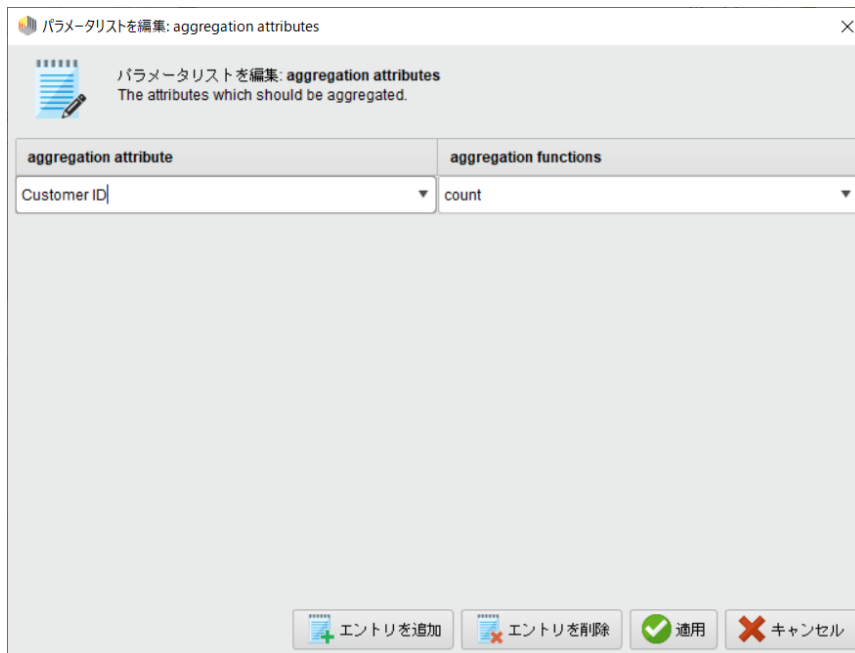
データをまとめて、商品の購入数を数える。

ACTIVITY(アクティビティ)

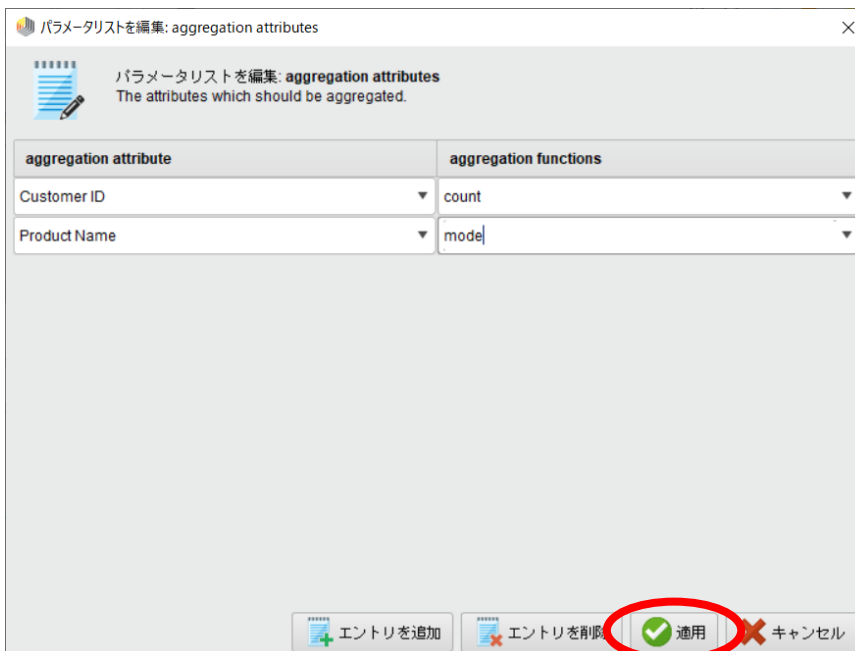
1. "Aggregate"オペレータを検索し、プロセスにドラッグします。それを"join"オペレータの"join"ポートと接続します。
2. "Aggregate"オペレータをクリックし、パラメータ設定を以下の通りにします。
3. aggregation attributes 横の"Edit List"をクリックします。



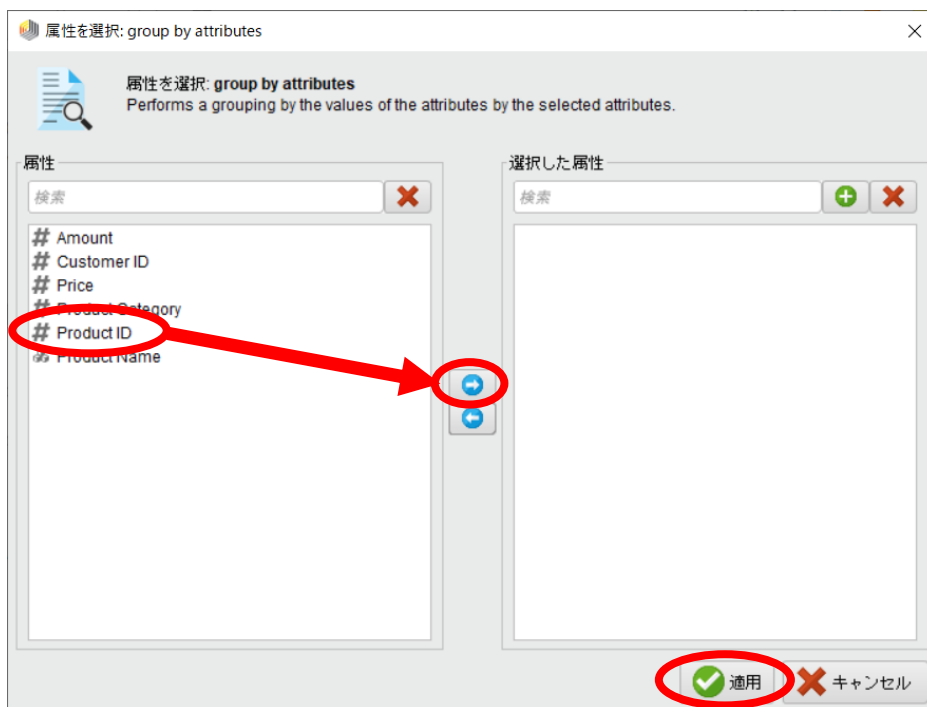
4. 左側のボックスは”Customer ID”を選び、右側の関数ボックスは”count”を設定します。



5. 同じ画面のまま、次は「エントリを追加」をクリックし、”Product Name”を左側に関数(function)を”mode”に設定。「適用」をクリックします。



6. ”group by attributes”横の「属性選択」をクリック。そして、”Product ID”を選択し、右側のボックスに移します。出来たら「適用」をクリックします。



EXPLANATION(説明)

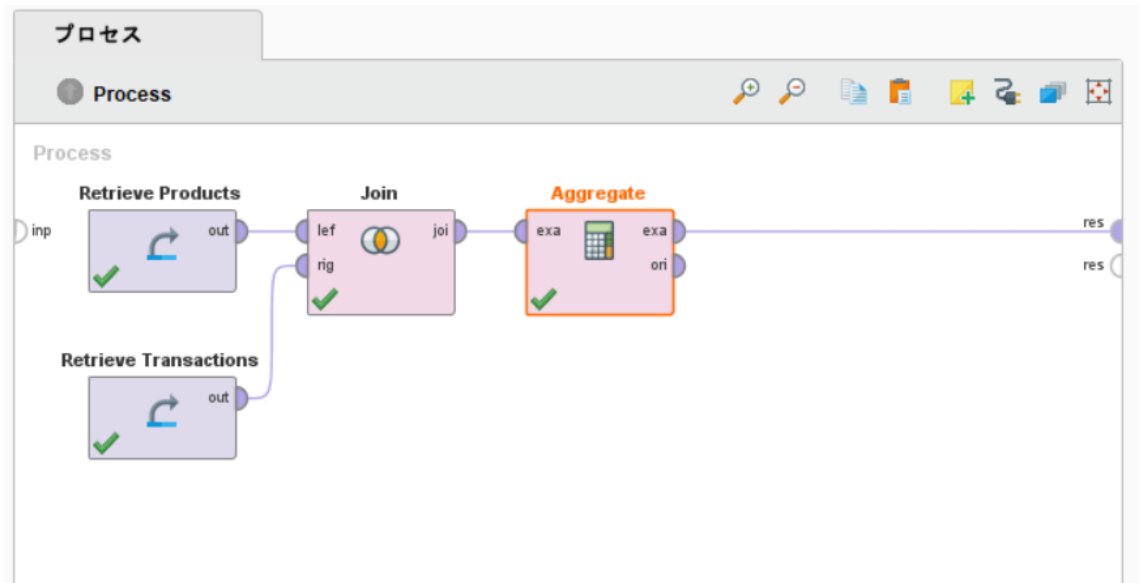
“Aggregate”オペレータは、いわゆるデータベース言語の“group-by-function”と同じ役割を果たします。”join”、“filters”の次に“aggregate”機能は、データブレンディングするために最も重要なオペレータの一つです。この場合”product”によってデータをグループ化し、各商品の購入数を数え、”product name”により商品が説明されます。結果は”product ID”、“product name”、商品の購入者数という属性を持つ全商品の表が完成します。

□ステップ 5/6

プロセスの実行

ACTIVITY(アクティビティ)

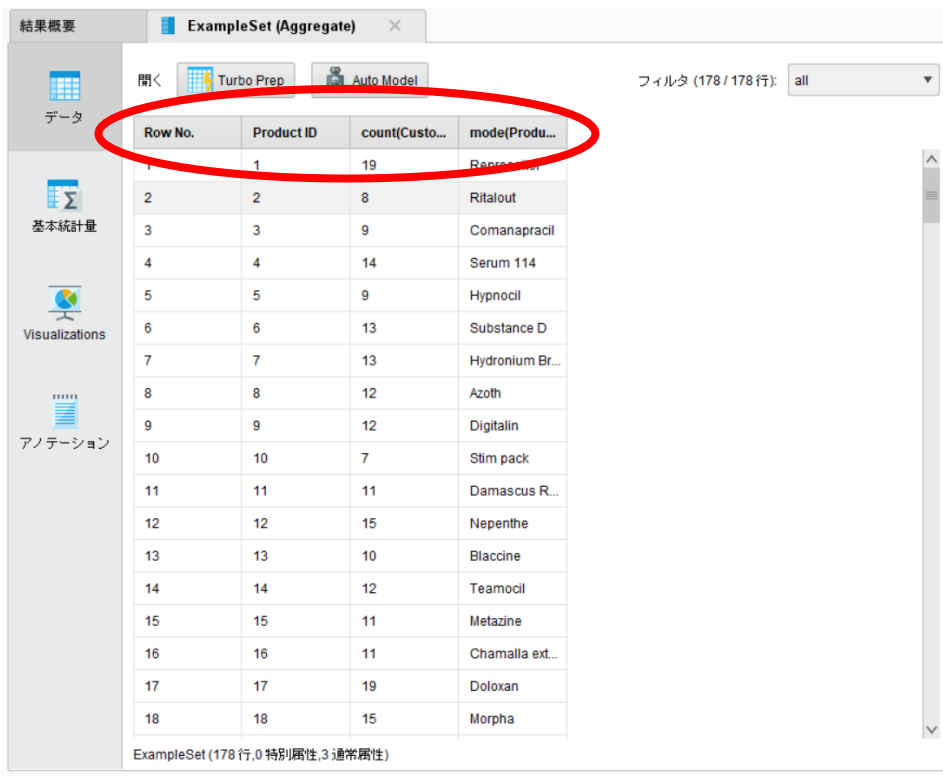
1. “Aggregate”オペレータと右側の”res”ポートを接続します。



2. 実行ボタンをクリックします。

EXPLANATION(説明)

結果ビューでは、列ヘッダーをクリックして、データを昇順、または降順に並び替えられます。



Row No.	Product ID	count(Custo...	mode(Produ...
1	1	19	Repro...
2	2	8	Ritalout
3	3	9	Comanapracil
4	4	14	Serum 114
5	5	9	Hypnocil
6	6	13	Substance D
7	7	13	Hydronium Br...
8	8	12	Azoth
9	9	12	Digitalin
10	10	7	Slim pack
11	11	11	Damascus R...
12	12	15	Nepenthe
13	13	10	Blaaccine
14	14	12	Teamocil
15	15	11	Metazine
16	16	11	Chamalla ext...
17	17	19	Doloxan
18	18	15	Morpha

ExampleSet (178 行, 0 特別属性, 3 通常属性)

□ステップ 6/6

ステップのまとめ - おめでとうございます！

RapidMiner でデータをブレンドする最初のステップが終わりました！次のチュートリアルに進む前に以下の質問について考えてみてください。

Challenge(追加質問)

- ・最もよく売れている商品はなんですか？また5回しか売れなかった商品はなんですか？
- ・基本統計量(Statistics)タブをクリックして、取引の平均回数は何回であったかを把握しましょう。またこのタブで値の分布をグラフで見ることができますか？
- ・Count 関数は各商品の取引回数を数えています、それぞれの商品は複数回取引されることもあります。各製品の合計が集計されるように"Aggregate"パラメータを変更できますか？そしてどの製品が65回以上販売されているのでしょうか？

2.7 6. Creating and Removing Columns

□ステップ 1/5

属性の検証

RapidMiner 上で最初の予測モデルを構築する準備がほとんど整っています。しかしまず初めに、データセットをより学習に適した形に変換する重要な二つの操作に取り組む必要があります。このプロセスの始めは前回取り組んだ物と同じなので、今まで学んだことを実践する絶好の機会です。その後、新しいデータ列、すなわち属性を作成し、不使用・不要な列を削除します。

□ステップ 2/5

製品の詳細をトランザクションに追加する

ACTIVITY(アクティビティ)

1. リポジトリから"Products"と"Transactions"のデータセットをプロセスにドラッグします。
2. "join"オペレータを検索してプロセスにドラッグします。
3. すべてのオペレータを接続します。
4. "Join"パラメータを調整して使用する列を指定します。key attributes の"edit List" をクリックして"Product ID" を左右両方のボックスのキー属性に設定します。



EXPLANATION(説明)

出力されるデータセットには、各取引における製品の詳細を含んだすべての取引が表示されています。

□ステップ 3/5

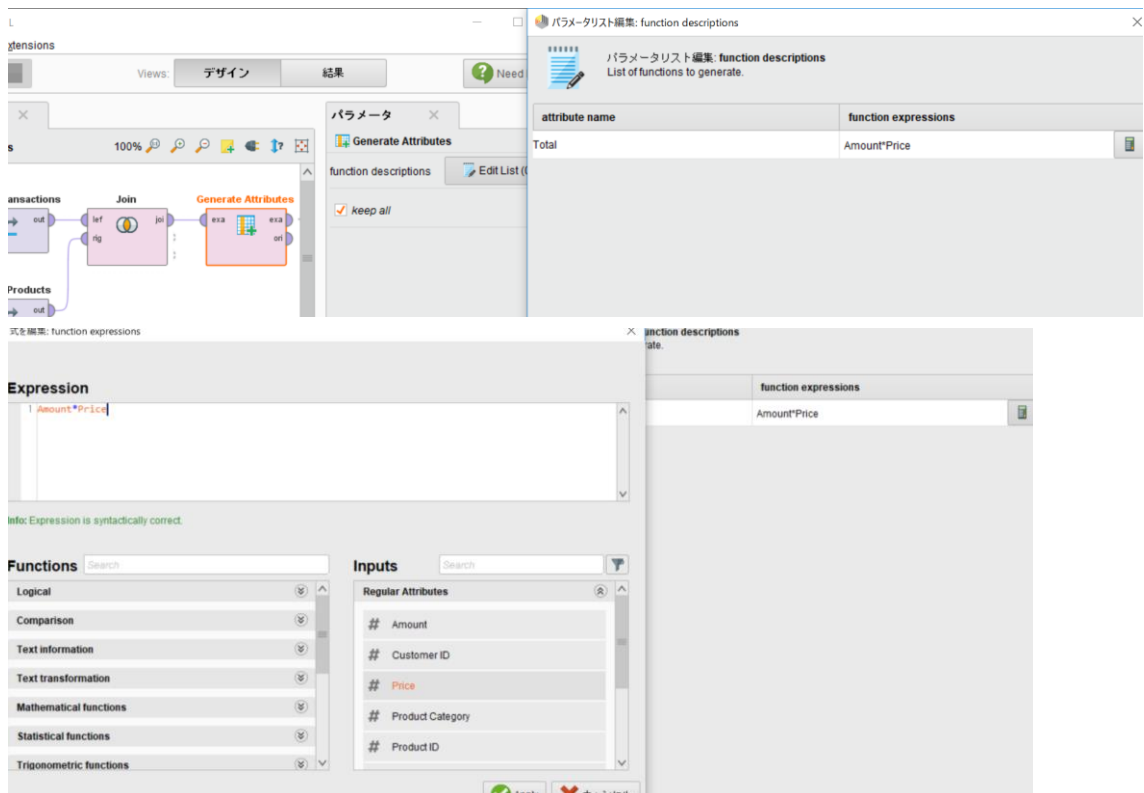
新しい属性の定義

EXPLANATION(説明)

属性(attribute)という単語は RapidMiner 用語で列の事を指します。機械学習において、データセットのそれぞれの行は特定の状況における具体例を表し、属性(列)は状況を記述する特性を表します。

ACTIVITY(アクティビティ)

1. “Generate Attribute”オペレータを追加します。
2. “join”オペレータと接続します。
3. “Generate Attribute”のパラメータにある“Edit List”をクリックして、新しい属性(項目)を追加します。ダイアログがポップアップ表示されるでしょう。
4. ダイアログボックスの左側”Attribute name”に”Total”と名前を入力します。
5. 右側の”function expression”に”Amount * Price”と入力します。



EXPLANATION(説明)

ダイアログに表示される電卓マークのボタンをクリックすることで、式の編集エディタ (Expression Editor)を使用する事ができます。試してみてください。テキストフィールドに入力するよりも簡単に式を作る事ができます。

□ステップ 4/5

不要な属性を取り除く

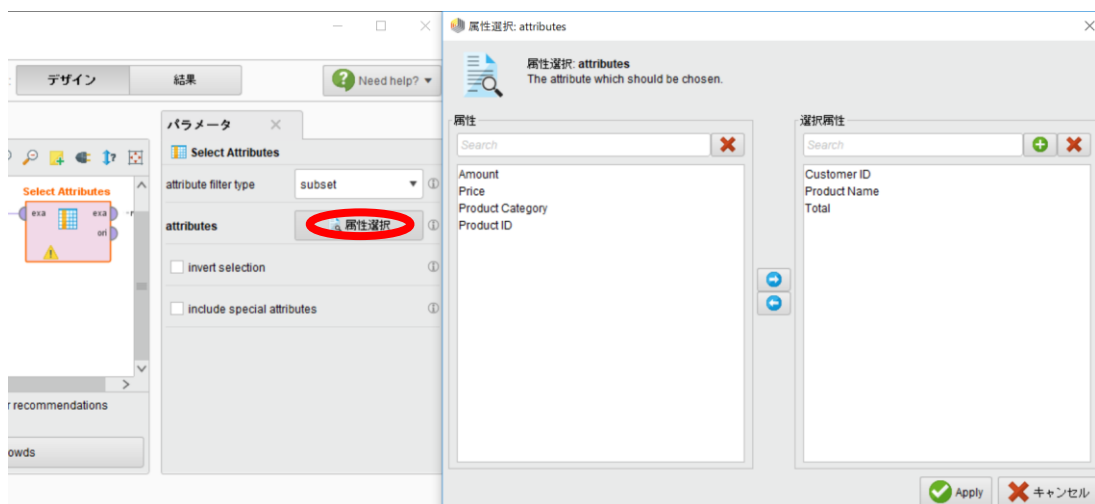
EXPLANATION(説明)

現在の結果のデータセットには、各取引において支払われた合計額(すなわち売れた数とその価格をかけたもの)が含まれています。

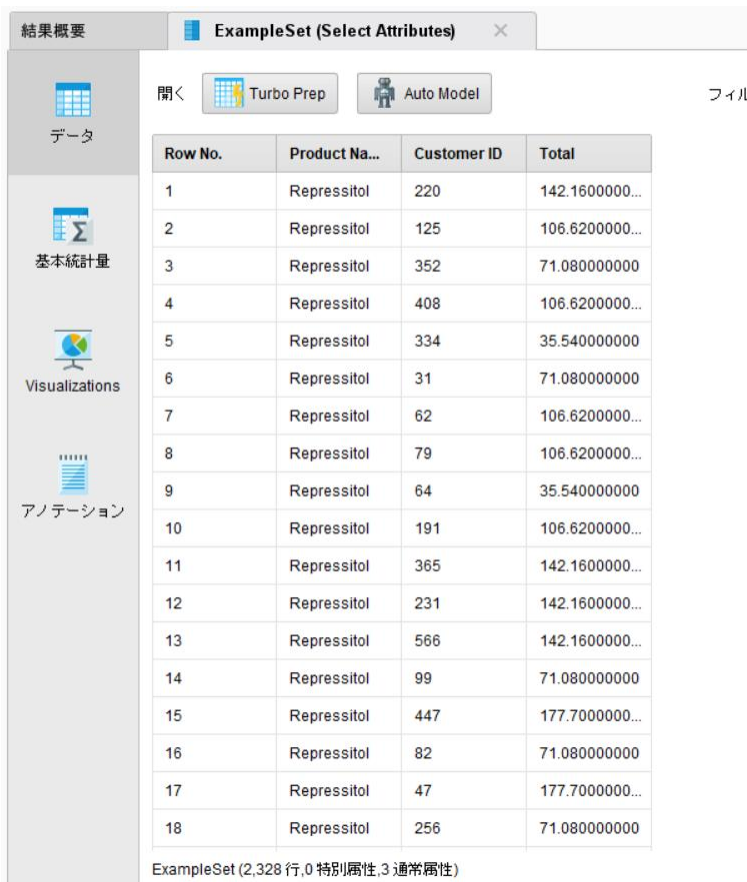
ACTIVITY(アクティビティ)

1. “Select Attributes”オペレータを追加して接続します。パラメータで次の変更を行います。

2. “attribute filter type”を”subset”にします。この機能は、あなたが指定した属性(列)のみにオペレータを適用するということを意味しています。ここでは、データから保持する列のサブセットを選択し、残りの列はすべて削除されます。
3. “属性選択(Select Attributes)”をクリックします。
4. 表示されるダイアログで、“Customer ID”, “Product Name”, “Total”を選択します。もしリストに何も表示されない場合はオペレータ同士が接続されているか確認してみてください。



5. 画面上部の青い矢印ボタンを押してプロセスを実行します。



Row No.	Product Na...	Customer ID	Total
1	Repressitol	220	142.1600000...
2	Repressitol	125	106.6200000...
3	Repressitol	352	71.080000000
4	Repressitol	408	106.6200000...
5	Repressitol	334	35.540000000
6	Repressitol	31	71.080000000
7	Repressitol	62	106.6200000...
8	Repressitol	79	106.6200000...
9	Repressitol	64	35.540000000
10	Repressitol	191	106.6200000...
11	Repressitol	365	142.1600000...
12	Repressitol	231	142.1600000...
13	Repressitol	566	142.1600000...
14	Repressitol	99	71.080000000
15	Repressitol	447	177.7000000...
16	Repressitol	82	71.080000000
17	Repressitol	47	177.7000000...
18	Repressitol	256	71.080000000

ExampleSet (2,328 行, 0 特別属性, 3 通常属性)

EXPLANATION(説明)

最後に res ポートへ接続する事を忘れないで下さい。結果は、各顧客が購入したそれぞれの商品へ支払った金額を示しています。”Select Attributes” で選択されていない属性は削除されています。

□ステップ 5/5

ステップのまとめ - おめでとうございます！

データブレンディングの達人になりつつあります！これまで”Join”、”Aggregate”、”Filter”、”Sort”、”Generate Attributes”、”Select Attributes”などデータの前処理に関して重要なオペレータを幾つか見て来ました。RapidMiner にはより多くのオペレータが存在しますが、この6つがもっともよく使用されます。

Challenge(追加質問)

- ・ 結果ビューから単一商品に最も多く支払った顧客の"customer ID"を見つけられますか？それはいくらですか？列ヘッダーをクリックするとデータを並び替える事が可能です。
- ・ 上記と同じ質問に、代わりにオペレータを使って答えられますか？
- ・ 新しい Total 属性について分布グラフはどのような形になりますか？基本統計量タブ (Statistics)での閲覧や、実際にグラフを作ってみてください。
- ・ 各取引の量(amount)の二乗を示す新しい列"Squared"を計算するようにプロセスを変更できますか？またそのプロセスを実行した後に、この新しい属性を表示させるためには他にどのような変更が必要ですか？

2.8 7. Changing Types and Roles

□ステップ 1/5

どれを予測するか

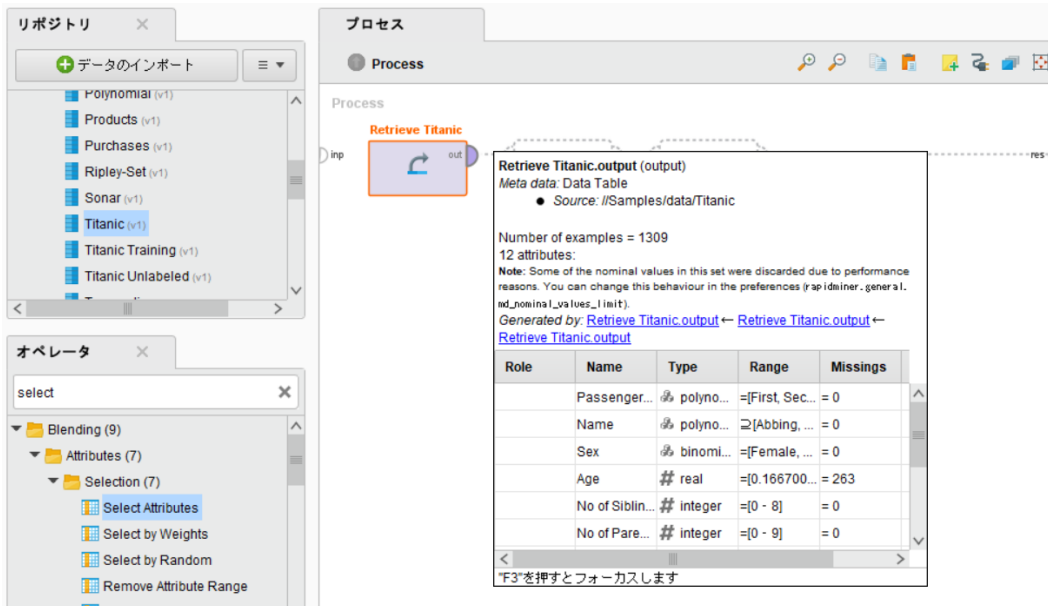
予測モデルの構築について学ぶために、先ほどのタイタニック号のデータに戻ってみましょう。このデータはややシンプルで準備もそれほど必要ではありませんが、予測したい項目を指定する必要があります。

□ステップ 2/5

タイタニックデータの詳細

ACTIVITY(アクティビティ)

1. "Titanic"データをプロセスにドラッグします。
2. "Retrieve"オペレータの out ポートの上にマウスをしばし置くと、ポップアップが表示されタイタニック号のデータセットに関するメタデータが表示されます。これらは基本統計量(Statistics)タブから知り得る情報の一部です。



Retrieve Titanic.output (output)
 Meta data: Data Table
 ● Source: //Samples/data/Titanic

Number of examples = 1309
 12 attributes:
 Note: Some of the nominal values in this set were discarded due to performance reasons. You can change this behaviour in the preferences (rapidminer_general.id_nominal_values_limit).
 Generated by: Retrieve Titanic.output ← Retrieve Titanic.output ← Retrieve Titanic.output

Role	Name	Type	Range	Missings
	Passenger...	polyno...	= [First, Sec...	= 0
	Name	polyno...	= [Abbing, ...	= 0
	Sex	binomi...	= [Female, ...	= 0
	Age	real	= [0.166700...	= 263
	No of Sibilin...	integer	= [0 - 8]	= 0
	No of Pare...	integer	= [0 - 9]	= 0

F3を押すとフォーカスします

3. 下部の表中の”Role”列と”type”列に気をつけて下さい。

EXPLANATION(説明)

- ・各属性は型(type)を持っています (例、nominal(項目型)もしくは numerical(数値型))
- ・属性に付与されている役割(role)は、機械学習オペレータがその項目をどう使うのか説明しています。ロールが付与されていない属性 (説明変数(regular attribute)とも呼ばれます) は学習の入力データとして使用される一方で、id ロールが付与されている属性は観測データの一意的識別子としてのみ使用されるため、モデルアルゴリズムによって無視されます。

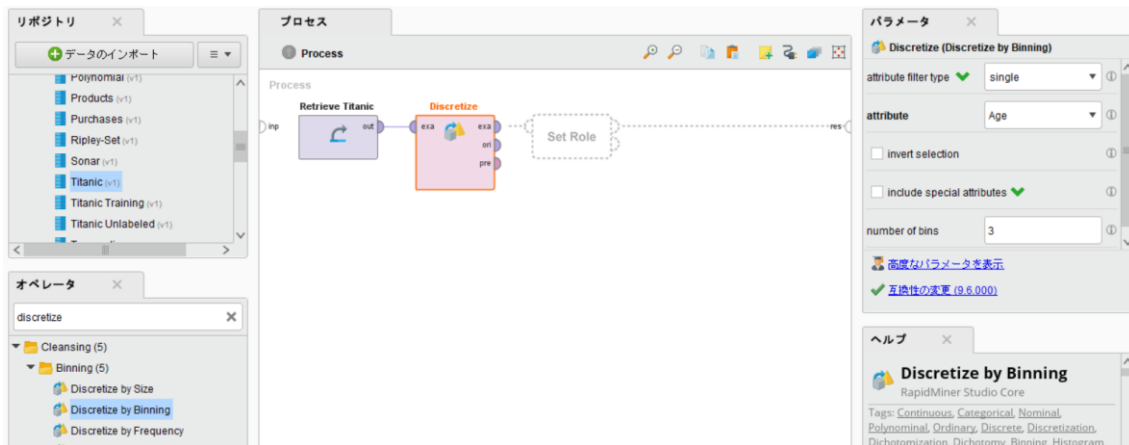
ロステップ 3/5

属性の型と役割の定義

ACTIVITY(アクティビティ)

1. “Discretize by Binning”オペレータを追加して接続します。
 - ・パラメータパネルで”attribute filter”の設定を single に設定します(すなわち、属性のうち一つだけに作用するようにします)。
 - ・”attribute”を”Age”に設定します。

- ・ “number of bins”を 3 に設定します。(高度なパラメータを表示(show advanced parameter)をクリックして下さい)



EXPLANATION(説明)

“Binning”は数値型から多項目型（三つ以上の値を持つ“nominal”型）に変換する一般的な手法です。ここでは値の範囲をカバーする三つのトレイ(‘bins’)を作成します。次にオペレータは数値型の値を、値が属するトレイの名前へと置き換えます。このテクニックは、データをより簡単なグループに分類して評価するのによく役立ちます。例えば、「x より大きい = 興味あり、x より小さい= 興味がない」というようになります。このようなグループは、計算やグラフに使用したり、モデル学習のためのラベルとしても使用できます。

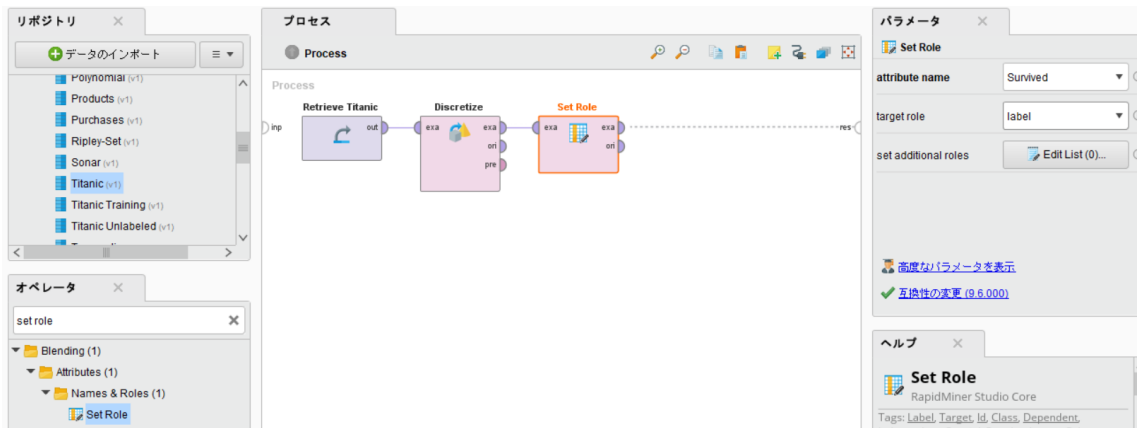
□ステップ 4/5

予測する列の定義

RapidMiner では予測する項目をラベルと呼びます。これは目的変数(Target)やクラスと呼ばれることもあります。

ACTIVITY(アクティビティ)

1. “Set Role”オペレータを追加して接続します。
2. パラメータパネルで、“attribute name”を”Survived”に、“target role”を”label”に変更します。



3. プロセスを実行し、結果を見てみましょう。

EXPLANATION(説明)

結果ビューの基本統計量タブを見てください。'Survived'のロールがLabel（目的変数）に変更されていることに注意して下さい。'Age'に関しても、新しい型をもち、数値は新しい値に置き換わっています。

□ステップ 5/5

ステップのまとめ - おめでとうございます！

識別 ID や重みといったように、属性のロールを設定することは多くの事に便利ですが、最も多く使われるのは目的変数(label)の定義、すなわち他の属性に基づいてどの属性を予測するのか定義するという事です。

Challenge(追加質問)

- ・ 結果ビューで"Age"列をもう一度見てください。作成した3つの区切りの境界線の値は何ですか。
- ・ "Age"と"Passenger Fare"を5つの範囲に分けるプロセスに変更してください。
- ・ 'Set Role'オペレータでは、ID と label 以外にどのようなロールがありますか？

2.9 8. More Modeling

□ステップ 1/5

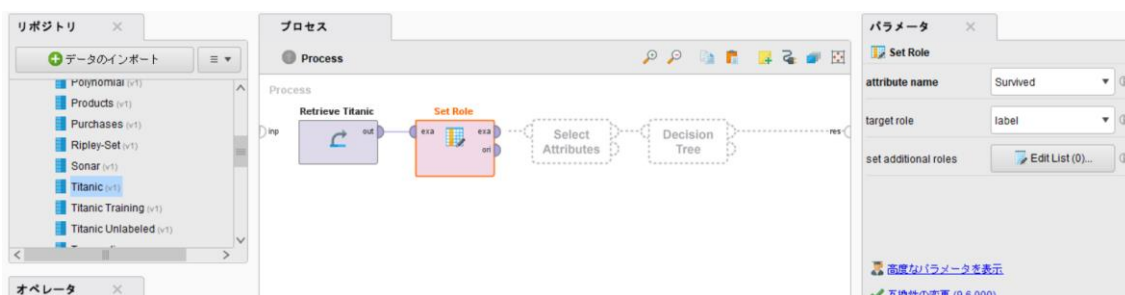
生存予測モデルの構築

これまで説明してきたオペレータを使うことで、予測モデルを構築するためのデータセットに対してブレンドや前処理を行う事ができます。このチュートリアルでは、最も広く使われている機械学習手法の一つである決定木(Decision Tree)を使ってタイタニック号事件の生存者の予測を行います。もちろん沈没後の今、私たちに出来る事はありませんが、同様の状況をこのモデルで予測する事はできます。あなたが家族で旅行していたとして本当に3等客室のチケットを買うべきでしょうか？モデルが教えてくれます！

□ステップ 2/5

タイタニック号のデータを取得する

1. タイタニック号のデータをプロセスにドラッグします。
2. “Set Role”を追加して接続します。そして前のチュートリアルの様に”attribute name”を”Survived”に、“target role”を”label”に変更します。



EXPLANATION(説明)

目的変数(label)の属性が予測する対象である事に注意して下さい。ラベルを設定することが重要なのは、決定木アルゴリズムの様に目的変数の項目が分かっている既存のデータ(トレーニングセット)を使用して、隠れたパターンを見つける機械学習の手法があるからです。そして、そのパターンから予測モデルを作成し、未知の新しいデータ(テストセット)に適用します。

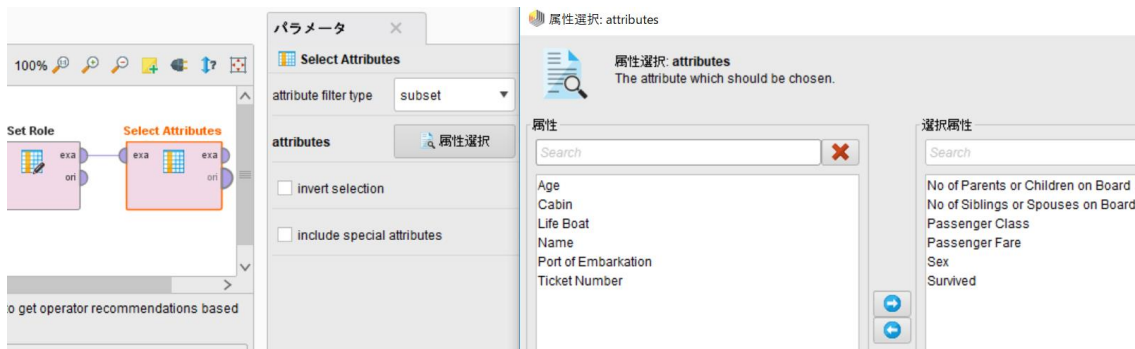
□ステップ 3/5

不要な属性を取り除く

ACTIVITY(アクティビティ)

1. プロセスに“Select Attributes”オペレータを追加し、接続します。

2. “attribute filter type”を”subset”に設定し、続けて属性選択(”Select Attributes”)をクリックします。
3. 表示されたダイアログで、”Survived”, ”Sex”, ”Passenger Class”, ”Passenger Fare”, ”No of Parent or Children on the Board”, ”No of Siblings or spouses on the Board”を[選択した属性]に移動させます。



EXPLANATION(説明)

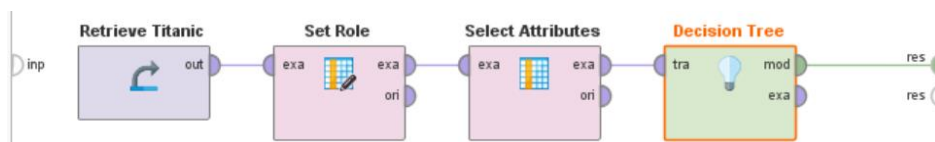
ライフポートに乗った乗客は生存する可能性が高いので、ライフポートを削除しました（選びませんでした）。この情報を追加すると、実質的にはこの情報だけに頼った意味のないモデルになってしまいます。本当の問題は、そもそも誰がライフポートを使用したかです。名前とチケット番号は ID の別の言い方に過ぎませんので、同様に削除します。

ロステップ 4/5

決定木モデルの構築

ACTIVITY(アクティビティ)

- ・ “Decision Tree”オペレータをプロセスにドラッグし、input ポートと res ポートとを接続します。

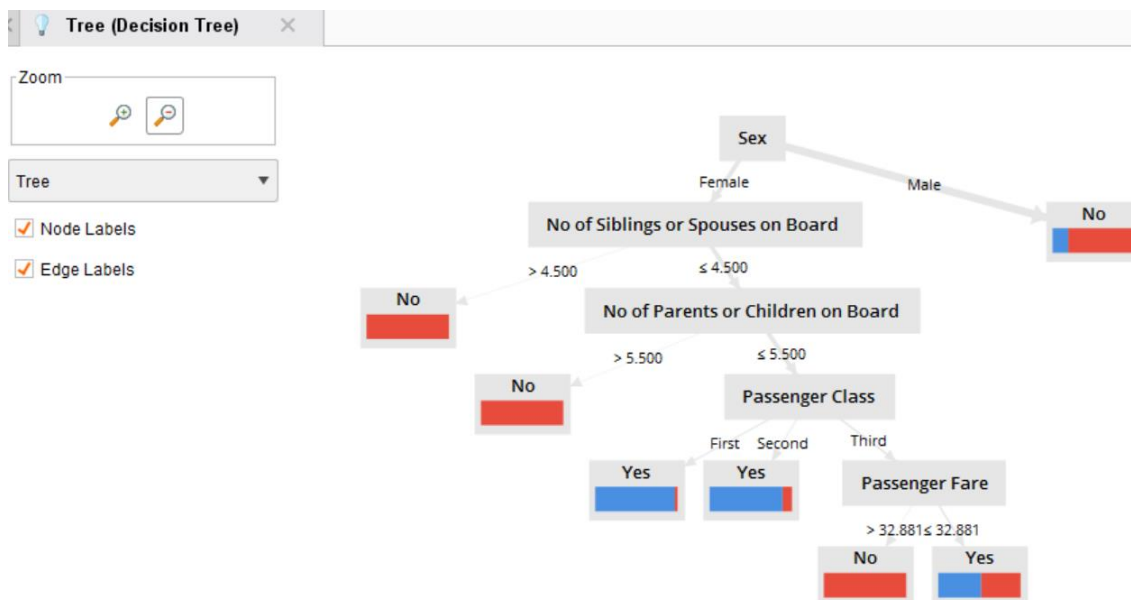


EXPLANATION(説明)

モデルの接続は緑色、データの接続は青色になることを確認して下さい。これは正しいポート接続が行えているのかを確認する簡単な方法になります。

ACTIVITY(アクティビティ)

1. プロセスを実行します。
 2. 決定木モデルの内容を確認します
- ※バージョンによって、作成されたモデルは多少異なります。



EXPLANATION(説明)

女性にとっては“family size”が“passenger class”より重要であるということは興味深い事です。男性にはこのパターンは検出されませんでした。女性や子供が優先されるので、一般的に男性が生き残る可能性は低くなります。

□ステップ 5/5

ステップのまとめ - おめでとうございます！

これで最初のチュートリアルはすべて終わりです！最も一般的なデータ前処理の方法とRapidMiner上で初めての予測モデルを構築しました。興奮してきますね！

Challenge(追加質問)

- ・ 作成した決定木の深さを制限する、すなわち複雑さを軽減させる方法が分かりますか？
なぜこれは良いアイデアだと言えますか？
- ・ 決定木モデルの深さを4に制限できますか？パラメータパネルで調整が可能です。

- ・ プロセスを再実行し、制限したツリーを見てください。深さは4に設定されています。ツリーの各色のバーの幅は、このバケツに入ってくる乗客の数を表しています。生存者の中で最大のグループであり、それゆえに生存の可能性が最も高いのはどんな人かわかりますか？
- ・ このグループの生存率はおおよそどのくらいの確率であったかと言えますか？男性の生存確率と比較してどの様なことが言えますか？

3. Prepare data コース

3.1 Handle Missing Values

□ステップ 1/6

より高度なデータ処理の学習

あなたはアナリストとしてデータの前処理に時間の大半を費やすことになるでしょう。データの処理には一般的に二つの方法があり、それはブレンディングとクレンジングです。今回と次回のチュートリアルではデータクレンジングを行うにあたり大切な操作について説明します。

EXPLANATION(説明)

ブレンディングはある状態から別のデータセットへと変換を行うことや、複数のデータセットを統合する事などです。クレンジングは、より精度の高いモデルを実現するようデータを改善する事を意味します。

タイタニック号のデータをもう一度見てみましょう。データの中にクエスチョンマークで表された欠損値があった事を覚えているかと思います。基本統計量ビューでは各属性の欠損値の数も確認する事が出来ます。欠損値はデータの前処理やモデル操作の際に大きな障壁となります。このチュートリアルでは、欠損値の除去や置き換えをするために最も一般的な方法を学習します。

EXPLANATION(説明)

ここからは二つ目のチュートリアルのコースです。先に進む前に、前のコースを終えているか確認して下さい。

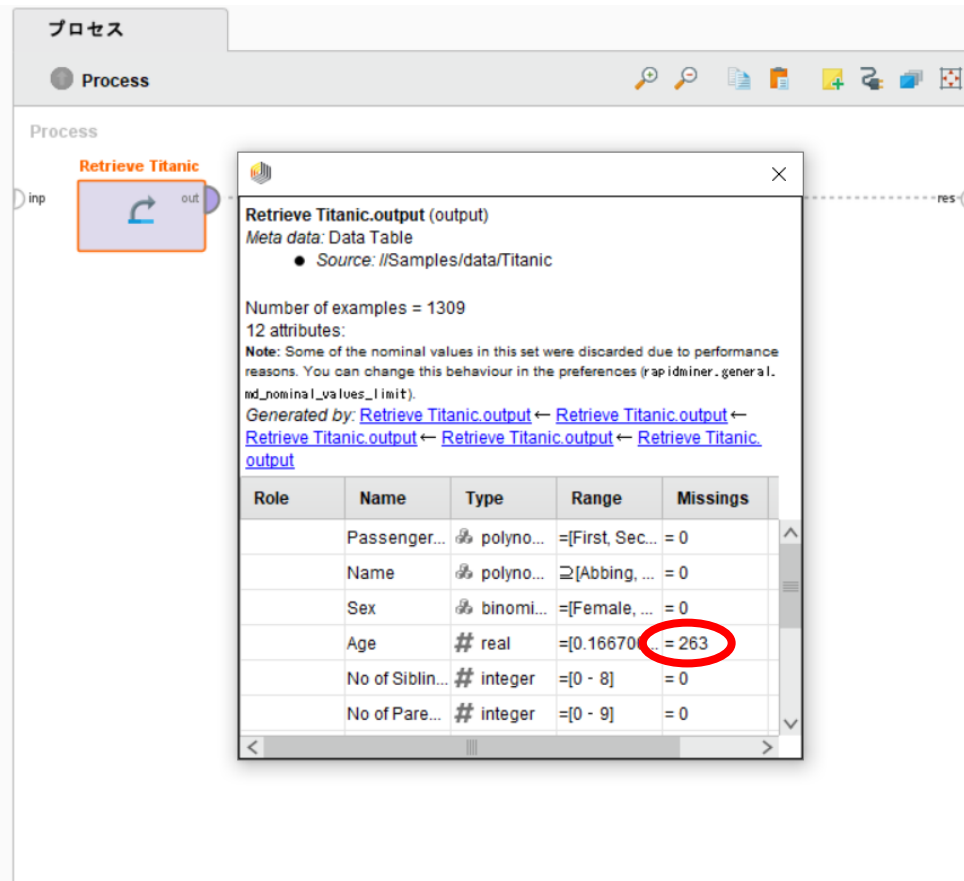
□ステップ 2/6

データの準備

ACTIVITY(アクティビティ)

1. "Titanic"データをプロセスにドラッグします。
2. out ポートにポインタを配置し、メタデータを表示するツールチップを待ちます。

3. ツールチップが表示されたら F3 を押します。ウィンドウが固定され、すべての列に関する情報をスクロールして表示させる事ができます。
4. 欠損値がある列をチェックしましょう。



Retrieve Titanic.output (output)
 Meta data: Data Table
 ● Source: //Samples/data/Titanic

Number of examples = 1309
 12 attributes:
 Note: Some of the nominal values in this set were discarded due to performance reasons. You can change this behaviour in the preferences (rapidminer.general.nd_nominal_values_limit).
 Generated by: Retrieve Titanic.output ← Retrieve Titanic.output ← Retrieve Titanic.output ← Retrieve Titanic.output ← Retrieve Titanic.output

Role	Name	Type	Range	Missings
	Passenger...	polyno...	={First, Sec...	= 0
	Name	polyno...	={Abbing, ...	= 0
	Sex	binomi...	={Female, ...	= 0
	Age	# real	={0.16670	= 263
	No of Sibli...	# integer	={0 - 8]	= 0
	No of Pare...	# integer	={0 - 9]	= 0

EXPLANATION(説明)

欠損値は”Age”, ”Passenger Fare”, ”Cabin”, ”Port of Embarkation”, ”Life Boat”の5つの属性に含まれています。その中でも”Cabin”と”Life Boat”には最も欠損値が多く含まれているので、最初に対処しましょう。

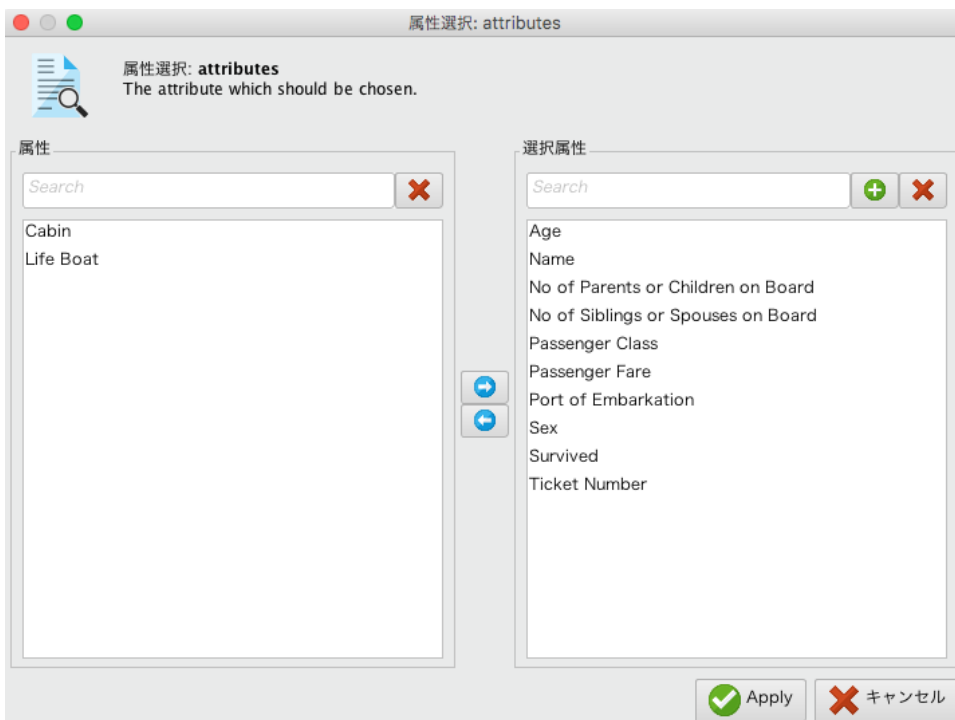
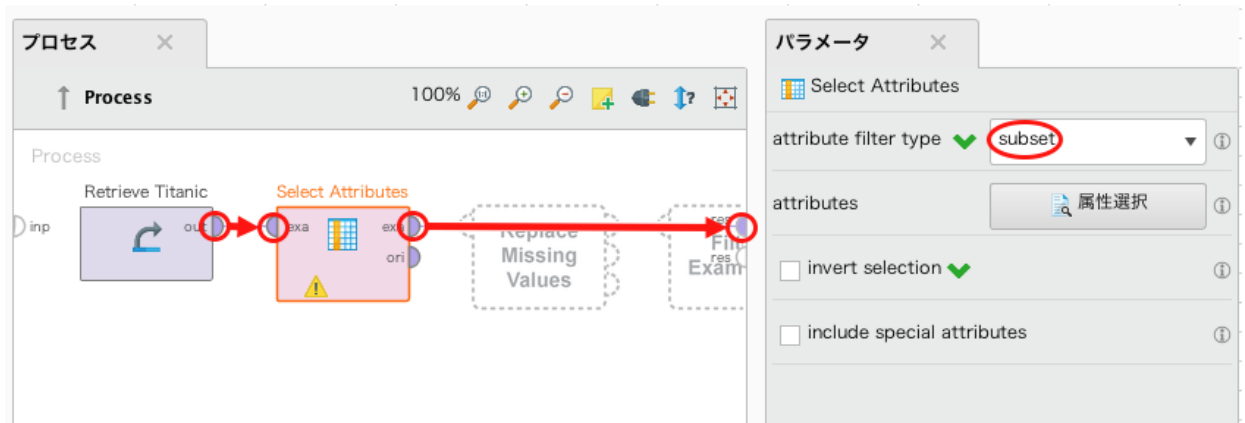
□ステップ 3/6

欠損値の多い属性を削除する

ACTIVITY(アクティビティ)

1. ”Select Attribute”オペレータを追加します。
2. ”Select Attribute”と”Retrieve”オペレータを接続して、右側の”res”ポートに接続します。

3. パラメータの"attribute filter type"を"subset"に変更し、"Select attributes"から"Cabin"と"Life Boat"以外のものをすべて選択します。これはこの二つの列が削除される事を意味します。



4. プロセスを実行します
5. 基本統計量タブをクリックし、欠損値のある列がまだ残っている事を確認します。

属性名	データ型	欠損値	フィルタ (10 / 10 属性):	属性の検索
Passenger Class	Polynomial	0	最小頻度値 Second (277)	最大値 Third (70)
Name	Polynomial	0	最小頻度値 van Melk [...] lemon (1)	最大値 Connolly,
Sex	Binominal	0	最小頻度値 Female (466)	最大値 Male (84)
Age	Real	263	最小値 0.166700000	最大値 80
No of Siblings or Spouses on B...	Integer	0	最小値 0	最大値 8
No of Parents or Children on B...	Integer	0	最小値 0	最大値 9
Ticket Number	Polynomial	0	最小頻度値 W/C 14208 (1)	最大値 CA. 2343
Passenger Fare	Numeric	1	最小値 0	最大値 512.3292
Port of Embarkation	Polynomial	2	最小頻度値 Queenstown (123)	最大値 Southamp

1 - 10 属性を表示中 行: 1,309 特別属性: 0 普通属性: 10

EXPLANATION(説明)

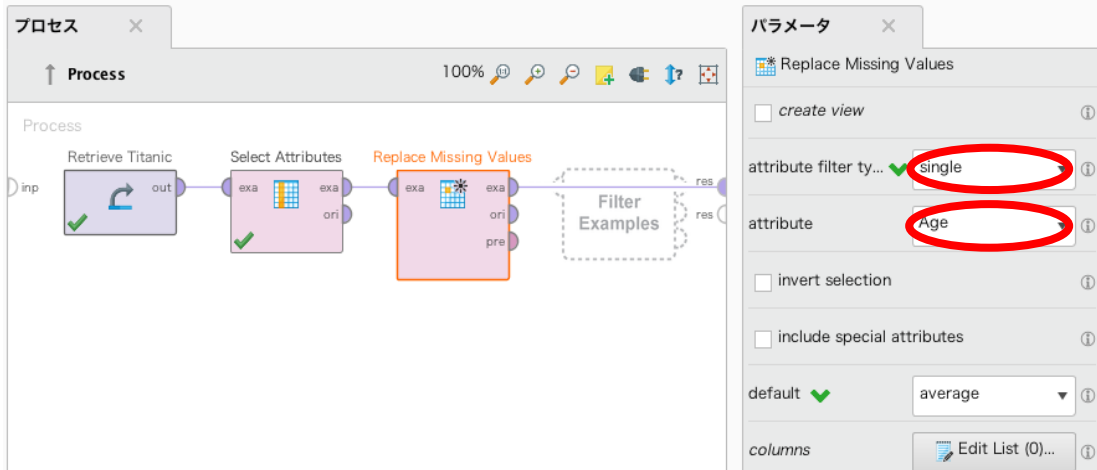
列のほとんどの値が欠損していて、残りの値にもおそらく有益な情報が残っていないと判断したので"Cabin"列を削除しました。そして同様の理由及び目的変数との相関が強いので、"Life Boat"列も削除しました。"Age"列も同様にかかなりの欠損値を持っていますが、次は別の方法を用いてこれを処理していきたいと思います。

□ステップ 4/6

欠損値の置き換え

1. "Replace Missing Values"を検索し、プロセスに追加します。追加するオペレータはオペレータとポート間のライン上にドロップできます（オペレータをドロップする前に、接続が強調される場所まで移動させます）。これによりオペレータを再接続する手間が省けます。

- このオペレータのパラメータ欄で、"attribute filter type"の"single"を使用し"attribute"を"Age"に設定します。



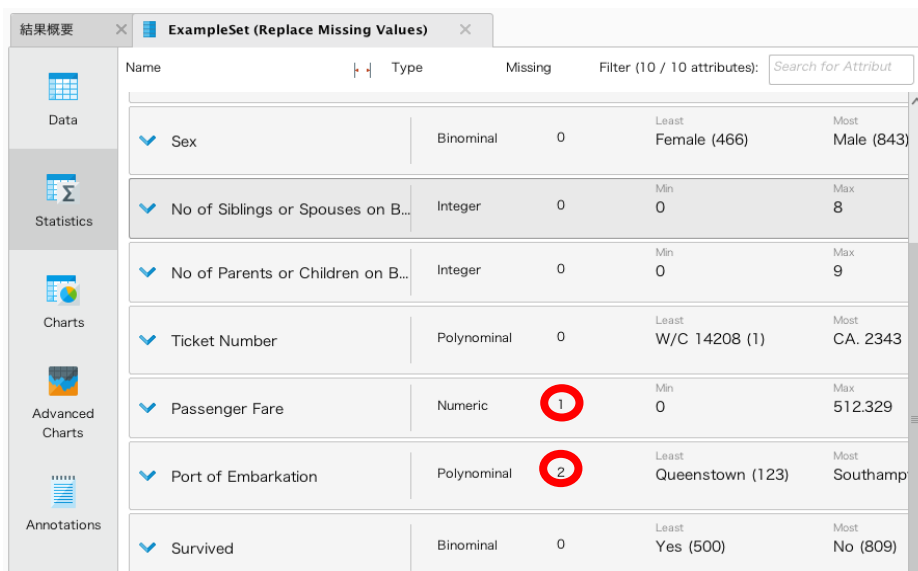
The screenshot shows a workflow in the 'プロセス' (Process) window. The workflow consists of four operators: 'Retrieve Titanic', 'Select Attributes', 'Replace Missing Values', and 'Filter Examples'. The 'Replace Missing Values' operator is highlighted in orange. To the right, the 'パラメータ' (Parameters) window for the 'Replace Missing Values' operator is open. The 'attribute filter type' parameter is set to 'single' and the 'attribute' parameter is set to 'Age'. Both 'single' and 'Age' are circled in red in the original image.

- プロセスを再度実行します。
-

EXPLANATION(説明)

・接続ラインにオペレータをドロップすることで、自分で接続を作成する手作業を省くことができます。この方法が最初から上手く行かなくても心配しないでください。その場合は手で再接続を行ってください。

・基本統計量タブを見ると、欠損値がある列がいくつか残っていることがわかります。このプロセスを実行すると、"Age"列の欠損値は"Age"の平均値に置き換わります。これは欠損値が多い場合における一般的な処理方法です。これで残りの欠損値は僅かになるので、安全にデータセットのフィルタリングを行う事ができます。



The screenshot shows the '結果概要' (Summary) window for the 'ExampleSet (Replace Missing Values)' operator. The 'Statistics' tab is selected, showing a table of statistics for various attributes. The 'Missing' column shows the number of missing values for each attribute. The values '1' and '2' in the 'Missing' column for 'Passenger Fare' and 'Port of Embarkation' respectively are circled in red.

Name	Type	Missing	Least	Most
Sex	Binominal	0	Least Female (466)	Most Male (843)
No of Siblings or Spouses on B...	Integer	0	Min 0	Max 8
No of Parents or Children on B...	Integer	0	Min 0	Max 9
Ticket Number	Polynomial	0	Least W/C 14208 (1)	Most CA. 2343
Passenger Fare	Numeric	1	Min 0	Max 512.329
Port of Embarkation	Polynomial	2	Least Queenstown (123)	Most Southamp
Survived	Binominal	0	Least Yes (500)	Most No (809)

□ステップ 5/6

欠損値のある行を削除する

EXPLANATION(説明)

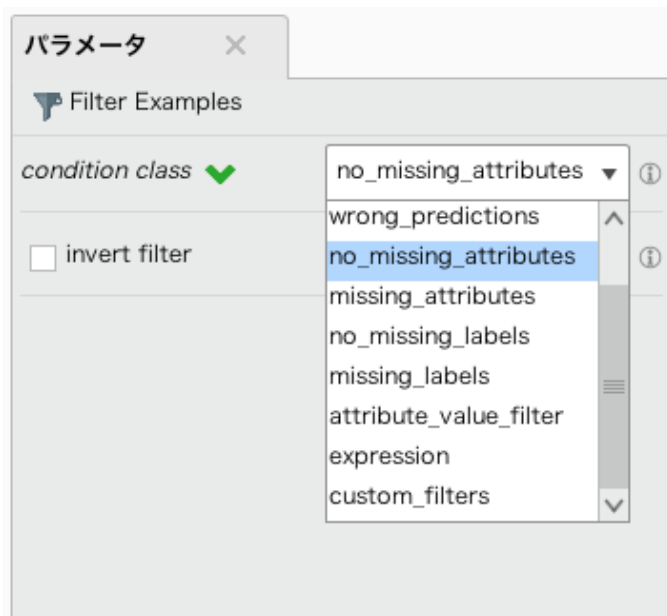
以前にも”Filter Example”オペレータを使用しましたが、今回は詳細設定をして欠損値を含んでいる行の削除を行います。

ACTIVITY(アクティビティ)

1. “Filter Examples”オペレータを検索し、接続ライン上に再びドロップします。失敗した場合は、もちろん手動で繋げても構いません。
2. パラメータ欄下部にある、高度なパラメータを表示/非表示するリンクを見てください。オペレータにあるすべてのパラメータを表示するには、「高度なパラメータを表示」(“Show advanced parameters”)をクリックします。



3. 新しいパラメータが表示されたら、“condition class”を“no_missing_attributes”に設定します。



4. プロセスを再度実行します。

□ステップ 6/6

ステップのまとめ - おめでとうございます！

欠損値のある列を削除する、欠損値を他の値に置き換える、欠損値のある行を削除するという3つの異なるアプローチを使用して欠損値処理を実施しました。次のチュートリアルに進む前に、以下の問題について考えてみてください。

Challenge(追加質問)

- ・基本統計量タブをチェックして下さい。まだ欠損値のある列は存在しますか？
- ・Age 列の欠損値を削除してしまうことは、欠損値を置き換えることよりも良いアイデアとは言えないのはなぜですか。
- ・”Select Attribute”オペレータを右クリックし、「オペレータの有効化」(“Enable Operator”)のチェックを外してオペレータを無効にします。そして”Replace Missing Value”オペレータも無効にします。プロセスを実行するとどうなると思いますか。試してみてください。
- ・データセットには現在いくつの行が残っていますか？

3.2 Normalization and Outlier Detection

□ステップ 1/7

データ内の異常や外れ値の特定

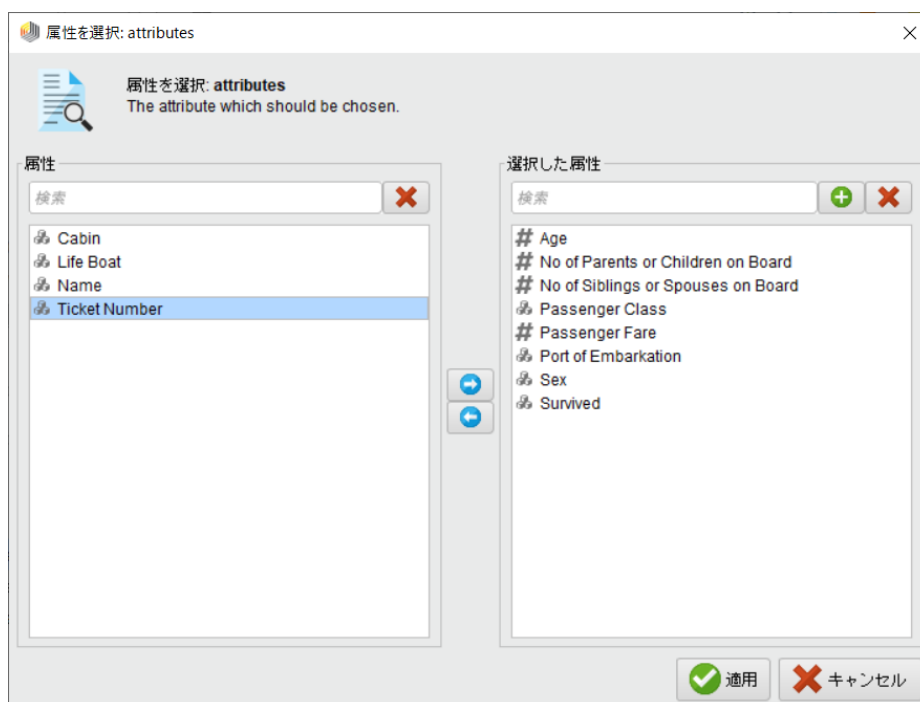
データクレンジングのもう一つの重要なステップは、特異なケースを識別しそれらをデータセットから削除することです。状況によっては外れ値自体に最も関心を寄せる場合もあります(例えばクレジットカードの不正取引を検出する場合など)。しかし多くの場合において、外れ値は不正確な測定による結果であり、データセットから削除する必要があります。このチュートリアルでは、この作業を主に行っていきます。

□ステップ 2/7

データの準備

ACTIVITY(アクティビティ)

1. "Titanic"データをプロセスに追加します。
2. "Select Attributes"オペレータを追加し、接続します。
3. パラメータ欄の"attribute filter type"を"subset"に変更し、"attributes"の属性選択をクリックします。そして"Cabin", "Life Boat", "Name", "Ticket Number"以外を選択し、これらの列を削除します。



EXPLANATION(説明)

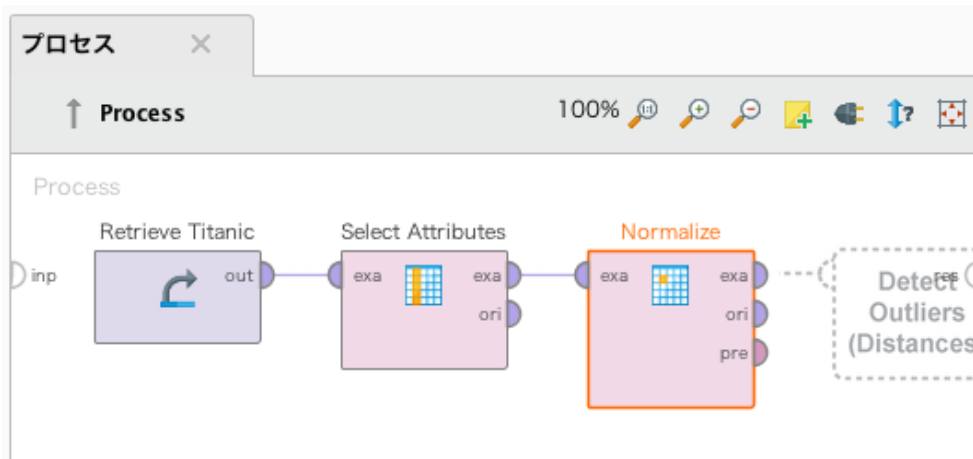
結果は、外れ値の検出に貢献すると思われる項目のみのデータセットになります。データ間のユークリッド距離を計算し、他のデータから最も遠い位置にある点を外れ値としてマークする距離ベースの外れ値検出アルゴリズムを用います。ユークリッド距離は属性ごとに2つのデータ間の距離を使用します。次のような場合について考えてみましょう。属性によって値の範囲が違う場合(ある属性の範囲が0から5で、別の属性の範囲が1から1000という場合など)、距離にどのような影響を与えるのでしょうか。大きな値を持つ属性は、小さい値を持つ属性より、影響度は大きくなります。このため、すべての属性が同じような値の範囲内で収まっている事を確認して下さい。この変換を正規化といいます。

□ステップ 3/7

変数の値を正規化する

ACTIVITY(アクティビティ)

“Normalize”のオペレータを追加し、接続します。


EXPLANATION(説明)

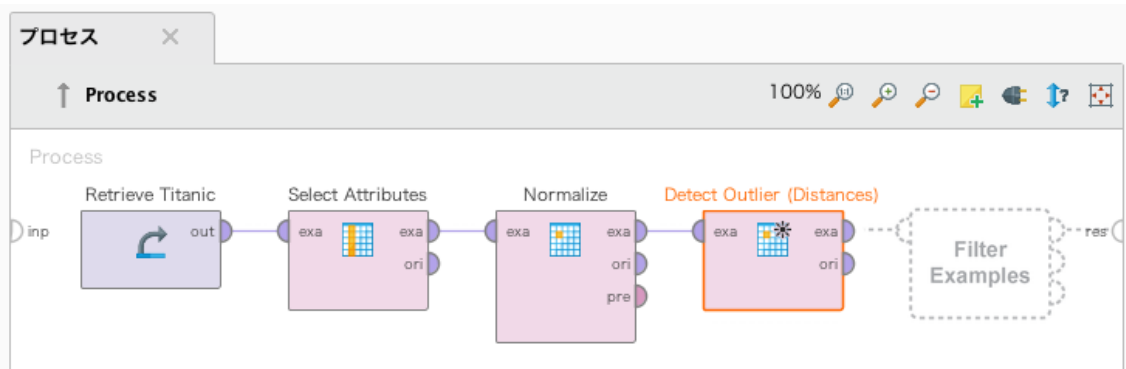
一般的に、外れ値検出や K-means クラスタリングの様な距離に基づくアルゴリズムを適用する前には、常にデータを正規化する必要があります。既定のパラメータを使用して、“Normalize”オペレータは各属性の平均値が0、標準偏差が1の値となるZ変換（標準化とも呼ばれます）を実行します。つまり標準化を行い同じ範囲にしたすべての属性は、互いに比較を行うことができます。

□ステップ 4/7

外れ値検出

ACTIVITY(アクティビティ)

“Detect Outlier(Distances)”オペレータを検索し、プロセスに追加、接続します



EXPLANATION(説明)

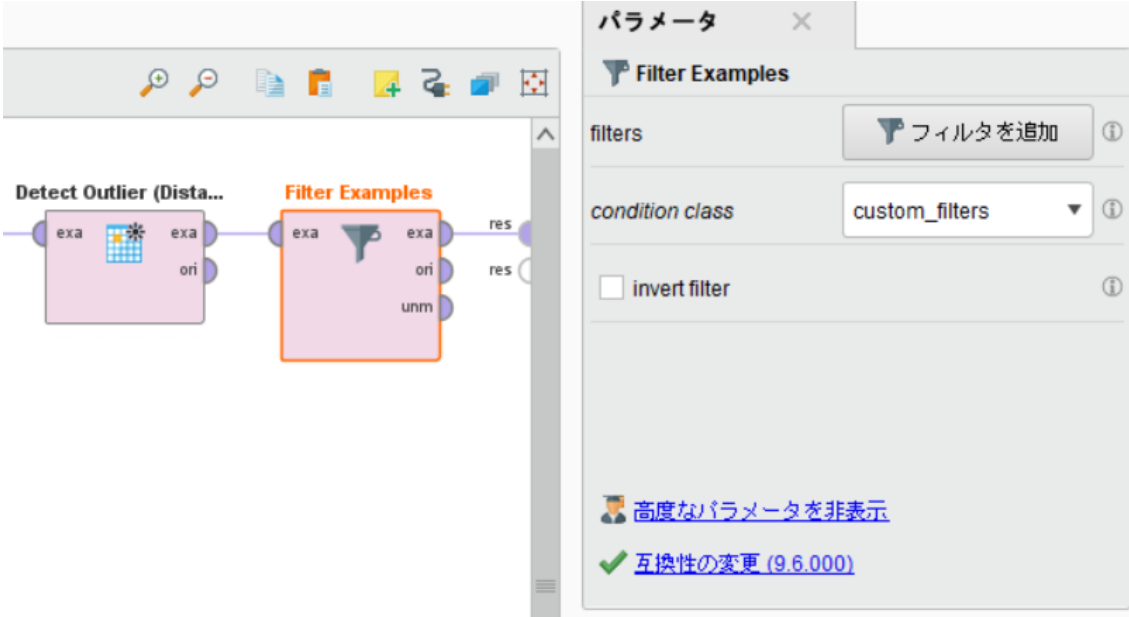
このオペレータは他の全ての値から最も遠い 10 のレコードを認識し、それらを外れ値にします。そして外れ値という新しい列が作られ、True である場合は外れ値、False である場合はそうでないことを示します。

□ステップ 5/7

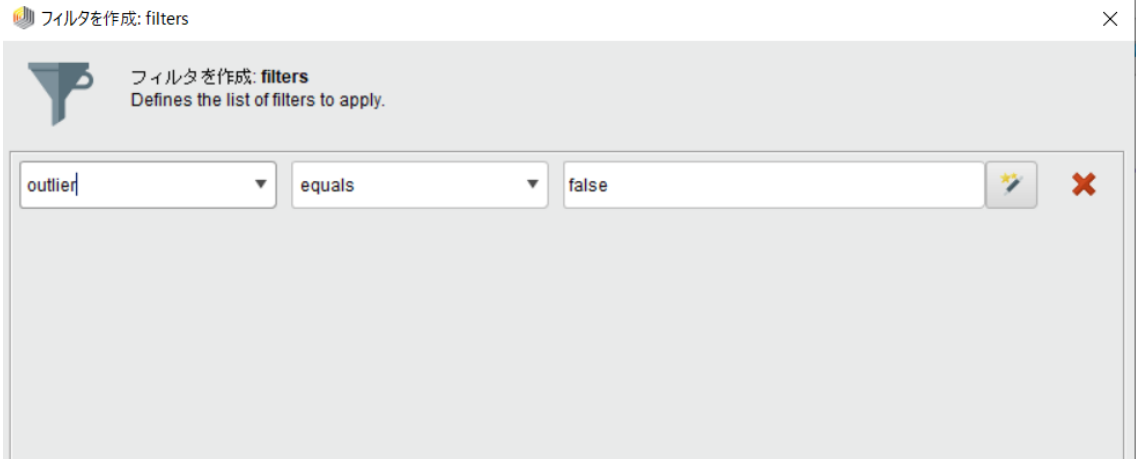
外れ値を除外する

ACTIVITY(アクティビティ)

1. プロセスに“Filter Example”オペレータを追加し、右側の res ポートに接続します。
2. パラメータ設定で“filter”の「フィルタを追加」(“Add Filter”)をクリックし、開いたウィザード上でパラメータの値として“outlier”, “equals”, “false”を設定しフィルターを追加します。



The screenshot shows the RapidMiner workflow editor. On the left, a process flow is visible: 'Detect Outlier (Distal...)' followed by 'Filter Examples'. The 'Filter Examples' process is highlighted with an orange border. On the right, the 'パラメータ' (Parameters) panel for 'Filter Examples' is open. It shows a 'filters' dropdown menu with a 'フィルタを追加' (Add Filter) button. Below it, the 'condition class' is set to 'custom_filters'. There is an unchecked 'invert filter' checkbox. At the bottom of the panel, there are two links: '高度なパラメータを非表示' (Hide advanced parameters) and '互換性の変更 (9.6.000)' (Change compatibility (9.6.000)).



The second screenshot shows the 'フィルタを作成: filters' (Create Filter: filters) dialog box. It has a title bar with a close button. The main area contains a funnel icon and the text 'フィルタを作成: filters' and 'Defines the list of filters to apply.' Below this, there is a configuration row with three fields: a dropdown menu showing 'outlier', a dropdown menu showing 'equals', and a text input field containing 'false'. To the right of these fields are two icons: a star and a red 'X'.

3. プロセスを実行します。

EXPLANATION(説明)

プロセスの実行にしばらく時間がかかる可能性があります、終わったら自動的に結果ビューに切り替えられます。1299 の行のデータセットが作成され、10 の外れ値が正常に削除された事を確認して下さい。

結果概要 ExampleSet (Filter Examples) ×

開く Turbo Prep Auto Model フィルタ (1,299 / 1,299 行): all

Row No.	outlier	Age	No of Sibling...	No of Parent...	Passenger F...	Passenger ...	Se:
1	false	-0.061132586	-0.478903729	-0.444829492	3.439849339	First	Fer
2	false	-2.009535166	0.481103899	1.865812593	2.284728814	First	Ma
3	false	-1.934376459	0.481103899	1.865812593	2.284728814	First	Fer
4	false	0.008246817	0.481103899	1.865812593	2.284728814	First	Ma
5	false	-0.338650197	0.481103899	1.865812593	2.284728814	First	Fer
6	false	1.257076065	-0.478903729	-0.444829492	-0.130325596	First	Ma
7	false	2.297767106	0.481103899	-0.444829492	0.862905137	First	Fer
8	false	0.632661441	-0.478903729	-0.444829492	-0.643283153	First	Ma
9	false	1.603973079	1.441111526	-0.444829492	0.351317399	First	Fer
10	false	2.852802327	-0.478903729	-0.444829492	0.313159540	First	Ma
11	false	1.187696662	0.481103899	-0.444829492	3.752598885	First	Ma
12	false	-0.824306016	0.481103899	-0.444829492	3.752598885	First	Fer
13	false	-0.408029600	-0.478903729	-0.444829492	0.695623012	First	Fer
14	false	-0.269270794	-0.478903729	-0.444829492	0.880133169	First	Fer
15	false	3.477216952	-0.478903729	-0.444829492	-0.063670094	First	Ma
16	false	?	-0.478903729	-0.444829492	-0.142400868	First	Ma
17	false	-0.408029600	-0.478903729	0.710491550	4.138926445	First	Ma
18	false	1.395834871	-0.478903729	0.710491550	4.138926445	First	Fer

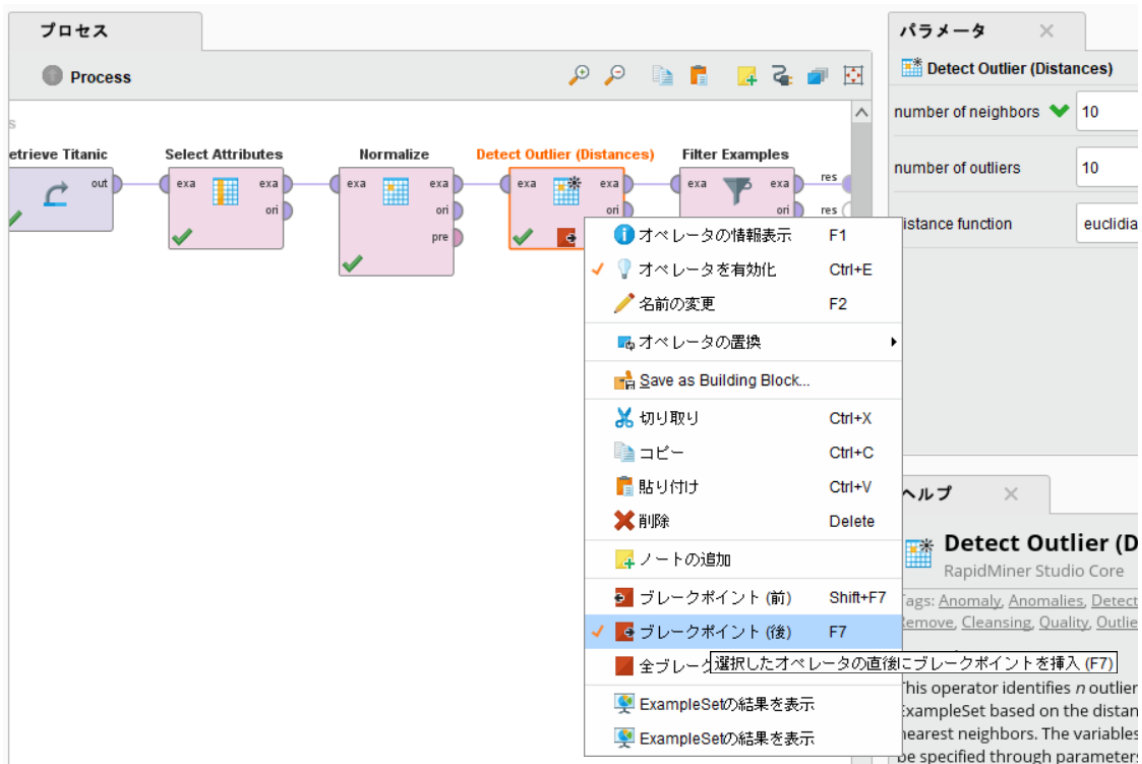
ExampleSet (1,299 行, 1 特別属性, 8 通常属性)

□ステップ 6/7

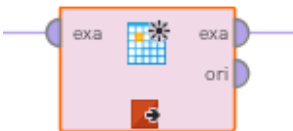
ブレークポイントを使用し、中間結果を表示する

ACTIVITY(アクティビティ)

1. デザインビューに戻ります。
2. “Detect Outlier”オペレータを右クリックし、メニューの[ブレークポイント(後)]を選択します。ブレークポイントを追加した後にオペレータ下部に小さなアイコンが表示されることを確認して下さい。



Detect Outlier (Distances)



3. プロセスを再度実行します。

EXPLANATION(説明)

プロセスがブレークポイントで一時停止して、その時点での結果が表示されます。この様にブレークポイントは、プロセスのデバッグに便利なツールです。1309 行のすべてのデータは、この時点ではデータセットに残っています。もう一度実行ボタンを押すことで処理を続行し、最終的な結果を参照することが出来ます。

□ステップ 717

ステップのまとめ - おめでとうございます！

あなたはデータから 10 の外れ値を発見しそれを取り除くことに成功しました！このクレンジングによってモデル品質を向上することが出来ます。例の通り、以下の課題についても考え、次のチュートリアルに進んでください。

Challenge(追加質問)

- ・外れ値を 10 ではなく 20 個検出するためには、プロセスをどのように変更したらいいでしょうか。
- ・外れ値を削除せず、外れ値だけを表示させるためにはプロセスをどのように変更したらいいでしょうか。
- ・"Detect Outlier(Distances)"オペレータと"Detect Outlier(LOF)"オペレータを取り替えて、実行前にこのオペレータにブレークポイント(後)を追加した。以前との違いはなんでしょうか。
- ・最も大きい外れ値だけを残すように、フィルターを変更するにはどのようにすればいいでしょうか？

3.3 Pivoting and Renaming

□ステップ 1/5

データを集計してピボットする

このチュートリアルでは、もう一つの一般的なデータブレンディング手法であるデータのピボットについて学習します。BI ツールや Excel でピボットの内容については慣れ親しんでいるかも知れませんが、ピボットとは縦長形式のデータ(一つの列と大量の行)から、横長形式のデータ(複数の列と一つの行)へと回転させることです。この変換は機械学習のデータを準備するステップで、二つ以上の次元で情報を集約するのに特に便利です。機械学習モデルでは、データを広いテーブル形式で保存する必要があるため、実際のモデリングを開始する前に、この前処理ステップに頻繁に遭遇することになります。

□ステップ 2/5

データのピボット

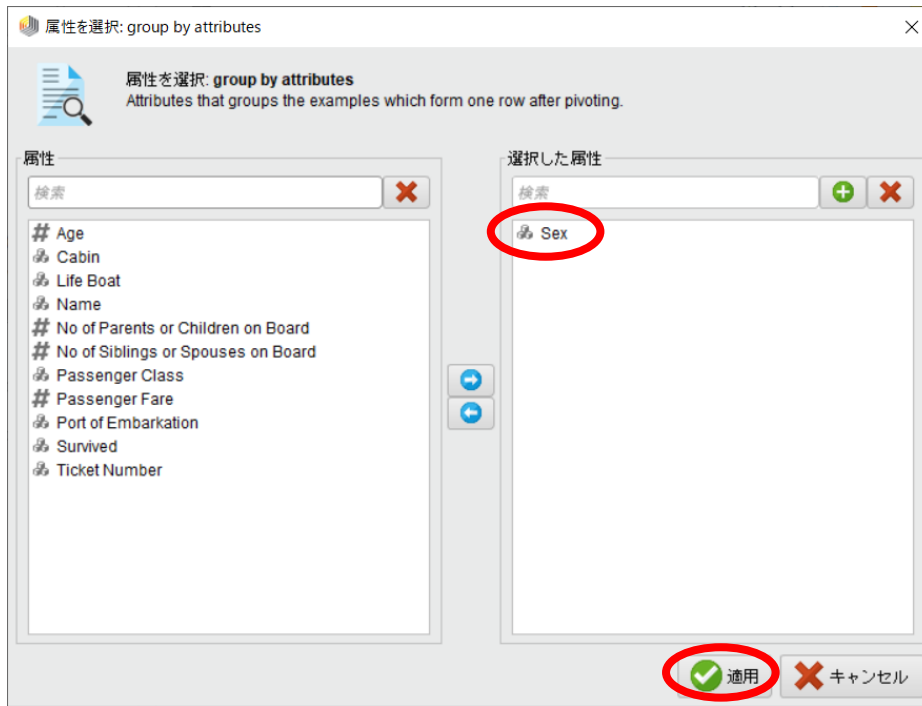
EXPLANATION(説明)

各クラスに何人の乗客がいたのか、男女別に分けた表を作ってみましょう。

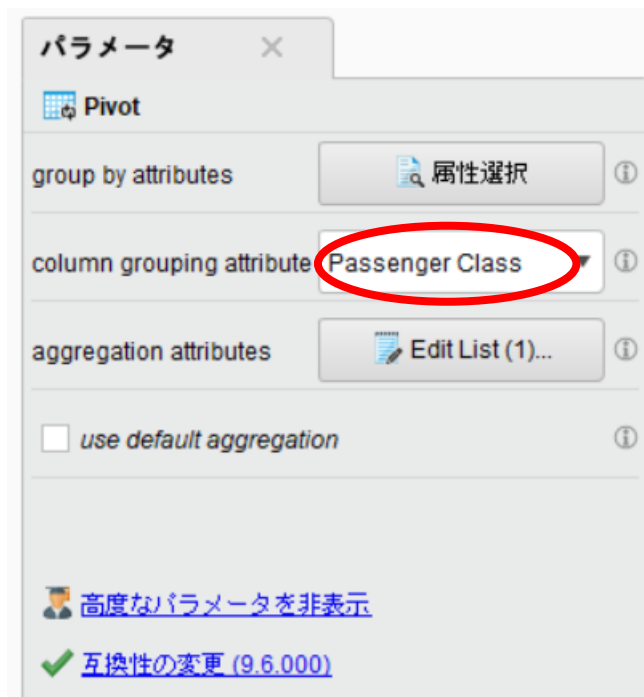
ACTIVITY(アクティビティ)

1. タイタニック号のデータをプロセスにドラッグしてきます。
2. "Pivot"オペレータを追加して、タイタニック号のデータと接続します。

- パラメータ欄で”group by attribute”の[属性選択(Select attributes)]をクリックし”Sex”を選択します。

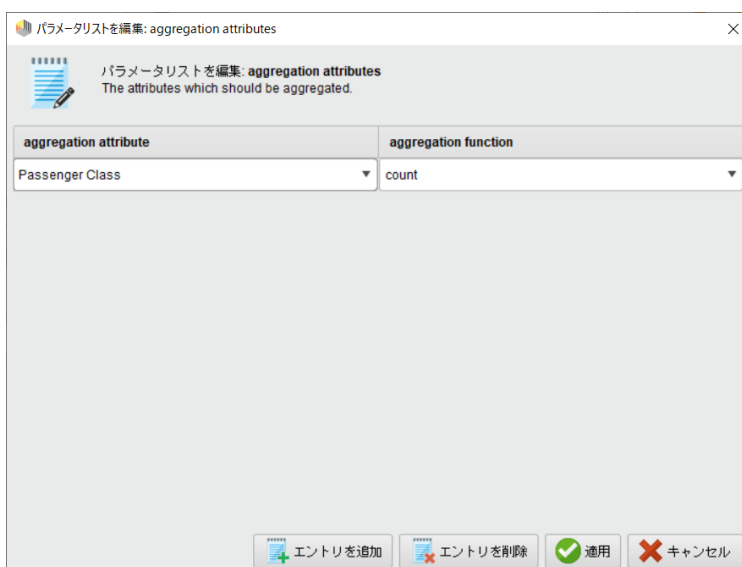


- パラメータ欄に戻り”column grouping attribute”に”Passenger Class”と設定します。



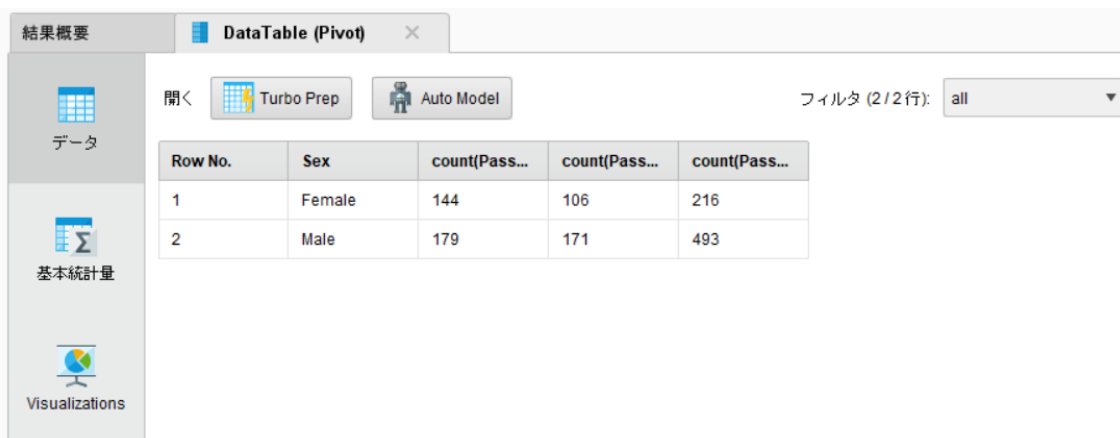
- (”Edit list”をクリックして)”aggregation attribute”に”Passenger Class”、”function”を”count”

に設定します。



EXPLANATION(説明)

結果として得られるデータ表は、4つの列と2つの行があります。各行は、Sex列 (“group by attributes”のパラメータです) の値を表します。column grouping attribute (Passenger Class)の三つの異なる値が、三つの新しい列になります。表の値は、行にあるグループの組み合わせ、つまり性別ごとの列にあるグループの組み合わせ、この場合は乗客クラスのカウントを表しています。例えば、ファーストクラスを予約した女性は144名です。



Row No.	Sex	count(Pass...)	count(Pass...)	count(Pass...)
1	Female	144	106	216
2	Male	179	171	493

□ステップ 3/5

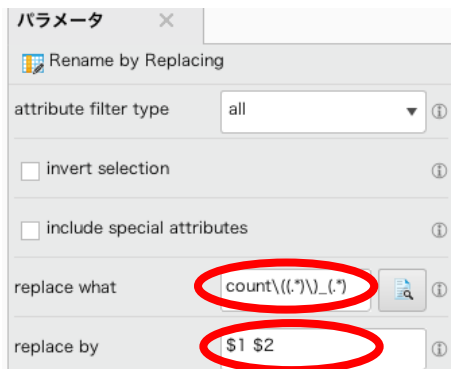
正規表現による属性の名前変更

EXPLANATION(説明)

作成された新しい列の名前は読みやすいものではありません。そこで”Rename”オペレータを使用して、”Passenger Class First”の様なより良い名前へと手動で変更する事ができます。Rename オペレータは名前を変更する属性が少ない場合に取れる方法ですが、ここでは一度に何百ものラベルの名前を変更する事ができる高度なアプローチを学びます。

ACTIVITY(アクティビティ)

1. “Rename by Replacing”オペレータを検索し、プロセスへの追加と”Pivot”とも接続を行います。
2. そして右側の”res”ポートにも接続します。
3. “replace what”のパラメータ欄に「count\((.*)\)_(.*)」をコピーして貼り付けます。すべての記号が正しく入力されているか確認してください。
4. “replace by”のパラメータ欄に「\$1 \$2」を貼り付けます。



EXPLANATION(説明)

あなたは既に正規表現(先ほど名前の変更に使用した見慣れないパラメータの名前です)に精通しているかも知れません。正規表現は強力なツールで RapidMiner の所々で使う機会があると思います。replace what に使用した式は count(and)_の中にある文言と、アンダースコアの後の文言を探しています。これら二つの要素は丸括弧によって指定されています。丸括弧を使用する度に、置換で参照するキャプチャグループを定義しています。ここでは丸括弧は特別な意味を持っているので、もとの属性名の括弧にはバックスラッシュを用いて引用しなければなりません。最後に、replace by パラメータでドル記号とグループ番号を指定し、キャプチャグループを使用します。\$1 は一つ目のグループの内容を表し、常

に"Passenger Class"を意味します。\$2 は二つ目のグループの内容を表しており、"First", "Second", "Third"という三つの異なるクラスになります。

□ステップ 5/6

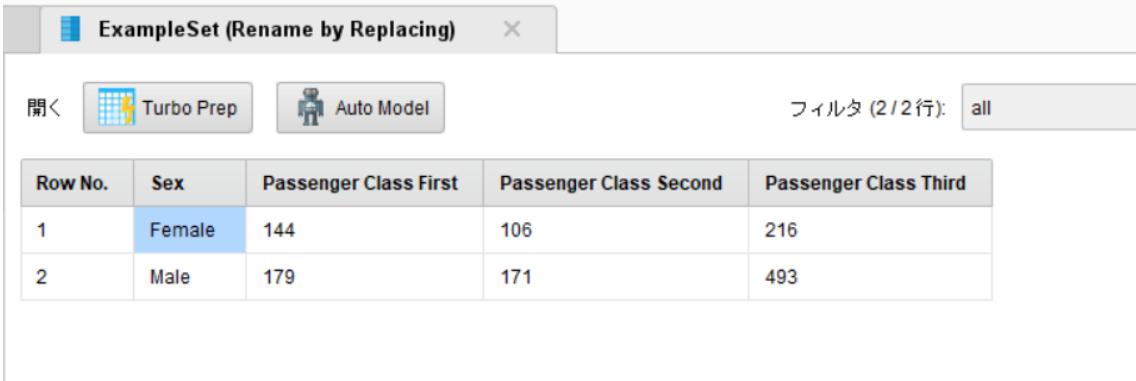
プロセスの実行

ACTIVITY(アクティビティ)

プロセスを実行します。

EXPLANATION(説明)

データセットは"Passenger Class First"の様な列名を持っているはずですが。



Row No.	Sex	Passenger Class First	Passenger Class Second	Passenger Class Third
1	Female	144	106	216
2	Male	179	171	493

□ステップ 6/6

ステップのまとめ - おめでとうございます！

横長のテーブル形式にデータセットを集約することができました。ピボットは少し設定が難しいかもしれませんが。ただ、group by attributes パラメータで、グループごとに1つの行をもつ行を作成し、column grouping attribute パラメータの値は新しい列を定義することを覚えておいてください。

Challenge(追加質問)

- ・ 列名が"First Passenger Class", "Second Passenger Class", "Third Passenger Class"となる様にプロセスを変更する事ができますか？
- ・ 同様に"First Class", "Second Class", "Third Class"という形に変更できますか？

- ・ピボットを変更し、性別を新しい列に、乗客クラスで3つのグループを定義するようにします。いくつかの行と列を得られるでしょうか？
- ・列名を変更し、性別名のみを新しいピボットの列名として使用できますか？
- ・"Rename by Replacing"オペレータを削除し、"Pivot"から"column grouping attribute"を削除します。"group by attributes"に Sex と Passenger Class を設定し、"aggregation attribute"で Passenger Class を count させます。プロセスを実行し、結果を確認します。最初に"Pivot"で得た結果とどのように異なりますか？

EXPLANATION(説明)

Challenge の 2、3、5 が上手くいかなくても気を落とさないで下さい。初めは正規表現は難しいですが、間違いなく学習する価値があります。書籍やオンライン上のリソースを活用しましょう。

3.4 Macros and Sampling

□ステップ 1/6

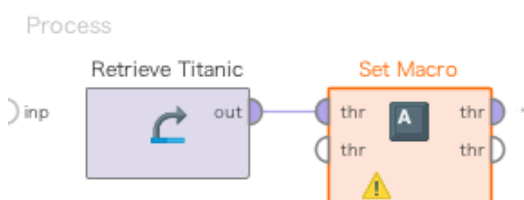
プロセス変数としてのマクロ

このチュートリアルでは、マクロというパワフルな RapidMiner の概念について学びます。マクロはプロセス内部で動的に値を格納し、読み込めるという点で変数と似ています。今回は、マクロを使用しデータセットのサイズを 50%に削減して、新しいデータサイズを計算しましょう。

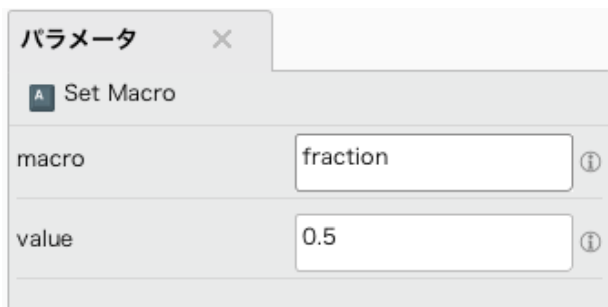
□ステップ 2/6

データの準備

1. "Titanic"のサンプルデータをプロセスにドラッグします。
2. "Set Macro"オペレータを検索し、プロセスに追加します。
3. "Titanic"と"Set Macro"オペレータをそれぞれ接続します。



4. “Set Macro”をクリックし、パラメータ欄で以下の変更を行います。
5. “macro”のパラメータ欄に“fraction”と入力します。
6. “value”のパラメータ欄に“0.5”と入力します。



EXPLANATION(説明)

マクロは任意の英数字を格納する事ができます。各マクロは名前と値をそれぞれもちます。マクロの設定に“Set Macro”を使用する方法もありますが、別の方法についても学習していきましょう。

□ステップ 3/6

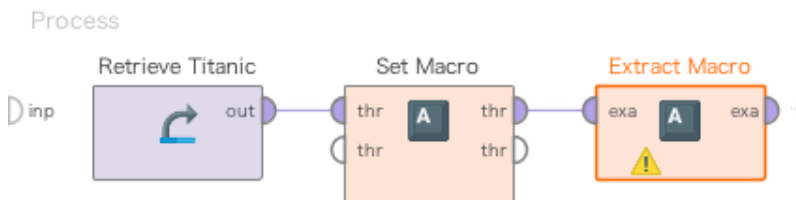
データからマクロの抽出

EXPLANATION(説明)

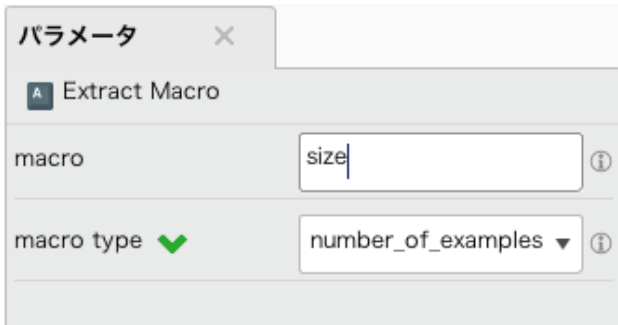
“Set Macro”を使用すると、手動で値を指定しマクロを作成できます。同様にデータセットの行数などプロセスの別の部分から値をとってマクロを作成することもできます。

ACTIVITY(アクティビティ)

1. “Extract Macro”オペレータをプロセスに追加し、接続します。



2. パラメータの“macro”欄に“size”と入力します。
3. “macro type”を“number_of_examples”に設定します。



ロステップ 4/6

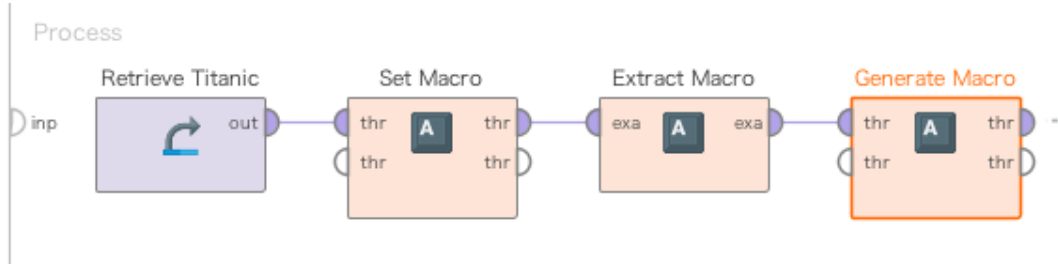
新しいマクロを計算する

EXPLANATION(説明)

以前に”Generate Attributes”オペレータを使用して新しい列を作ったのと同じ方法で、任意の計算式を用いてマクロを作成します。ここで計算する新しい行セットは”Set Macro”で以前に定義したものと、”Extract Macro”の古いものとの分数を掛け合わせたもので、最後に結果を整数にします。

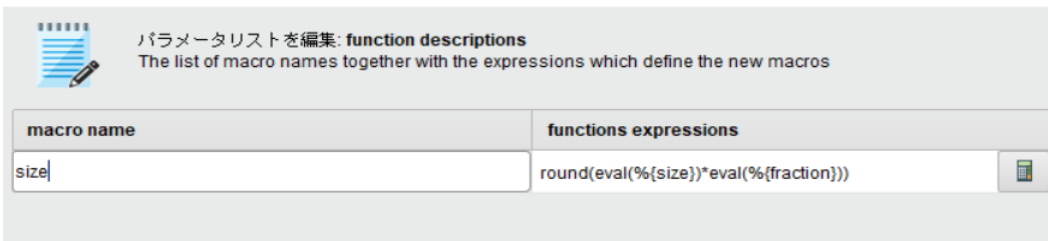
ACTIVITY(アクティビティ)

1. “Generate Macro”パラメータを検索し、プロセスにドラッグします。
2. 前のオペレータと接続します。



3. パラメータ欄で、”function descriptions”の”Edit List”をクリックします。
4. “macro name”に”new size”と、”function expression”に
`round(eval(%{size})*eval(%{fraction}))`とそれぞれ入力を行います。

パラメータリストを編集: function descriptions



EXPLANATION(説明)

マクロを定義すると"macro name"といった様にマクロ名に任意のテキストを使用する事ができます。その他のオペレータのパラメータ欄でマクロを用いる場合、"%{macro name}"という形式を使用して下さい。

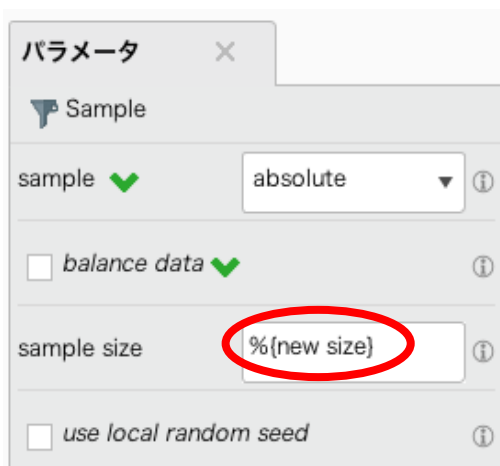
eval 関数は記述した内容を RapidMiner の構文解析にかけ、マクロの値を評価します。今回の場合、二つのマクロを数値に変換し、乗算で使用できるようにします（二つのテキストでは掛け合わせられません）。関数式の一部としてマクロを使いたい場合にはしばしば eval 関数を使用する必要がありますが、他のオペレータのパラメータとしてマクロを使用する場合には必要ありません。

□ステップ 5/6

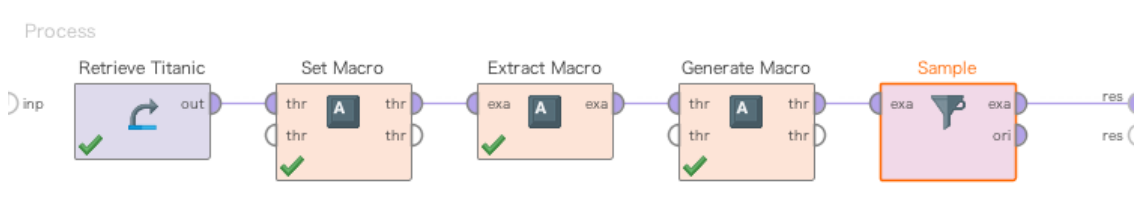
データのサンプリング

ACTIVITY(アクティビティ)

1. プロセスに"Sample"オペレータを追加して、接続します。
2. パラメータ欄の"sample size"を"%{new size}"に設定します。



3. オペレータを右側の res ポートに接続します。



4. プロセスを実行します。

元データのデータ数

ExampleSet (1,309 行,0 特別属性,12 通常属性)

プロセス実行後のデータ数

ExampleSet (655 行,0 特別属性,12 通常属性)

EXPLANATION(説明)

パラメータ設定のマクロの値にアクセスするために“%”形式を使う必要があることに再度気をつけて下さい。

□ステップ 6/6

ステップのまとめ - おめでとうございます！

マクロニンジャになりつつありますよ。マクロは例えば割合でサンプリングを行いたい場合など、プロセスの初期段階に主要パラメータを定義するものとして使用され、簡単に変更や再利用ができます。

Challenge(追加質問)

- ・元のサイズの30%と80%のサンプルデータを構築しようと計画しました。何を変更する必要がありますか？結果のデータサイズはいくつでしょうか？
- ・サンプルリポジトリから“Titanic”を“Iris”のデータセットに変更してみてください。他にも変更する必要があるものがありますか？そのまま実行しても大丈夫でしょうか？
- ・“Sample”オペレータのパラメータを見てください。マクロでの計算なしに50%のサンプルデータを作成する設定はどのように行いますか？“fraction”マクロの設定は維持したまま、“Sample”オペレータ内の割合設定に直接使用するようにプロセスを変更してください。

3.5 Looping, Branching, and Appending

□ステップ 1/10

サブプロセスを繰り返し実行する

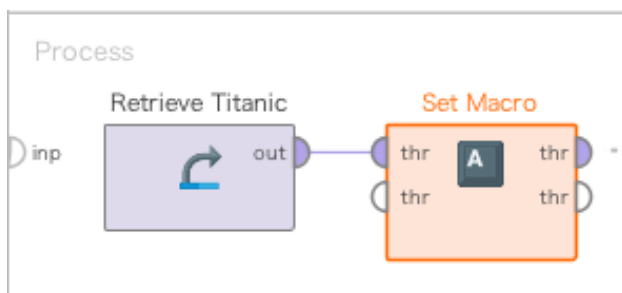
ループや分岐文はすべてのプログラミング言語で使われている有名な概念です。RapidMiner は行や属性、指定した値などをループ処理させるオペレータを数十種類提供しています。このチュートリアルでは、“Titanic”データで使っている3つの異なる“Passenger Class”をループ処理させるプロセスを構築し、行の数が、事前に指定した数字を上回っているかを検証します。上回っている場合はその数字と同じ数サンプリングし、それ以下だとそのまま保持するという方法をとります。このような処理は、例えば限られた非常に大きなクラスが少数のケースを支配している場合に、クラスのバランスをとるために使用することができます。

□ステップ 2/10

データとマクロの準備

ACTIVITY(アクティビティ)

1. “Titanic”のデータをプロセスにドラッグします。
2. “Set Macro”オペレータを追加して、接続します。



3. パラメータ欄で“macro”を“max size”に、“value”を 400 と入力します。



The screenshot shows a dialog box titled 'パラメータ' (Parameters) for the 'Set Macro' operator. It contains two input fields: 'macro' with the value 'max size' and 'value' with the value '400'. Each field has an information icon (i) to its right.

EXPLANATION(説明)

ここでは3つの乗船クラスそれぞれに最大 400 行を振り分けます。プロセスの開始時に定義したマクロは、後から容易に設定を変更することが出来ます。このことはプロセスで複数の値を扱うときにとりわけ役に立ちます。

ロステップ 3/10

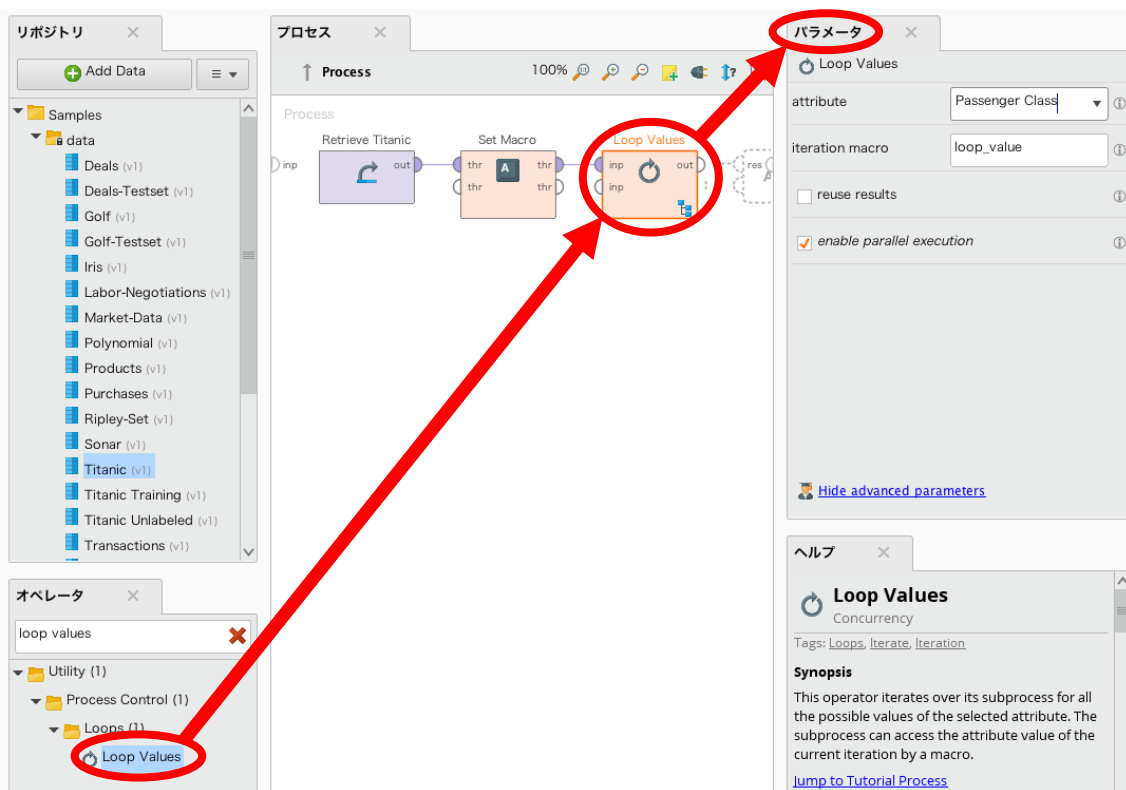
属性値に対してループの実行

EXPLANATION(説明)

3つの乗船クラスをループして各クラスの条件を確かめましょう。この場合は”max size”よりも少ない条件にするべきです。

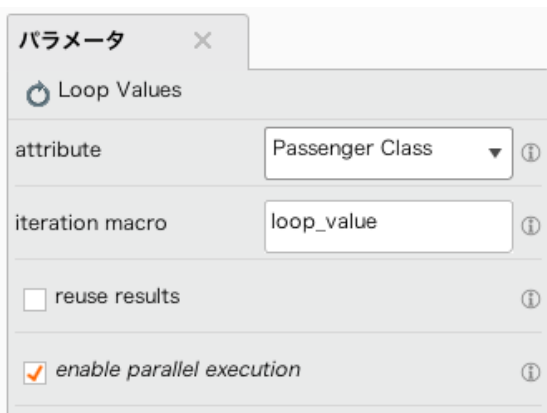
ACTIVITY(アクティビティ)

1. “Loop Values”パラメータを検索し、プロセスにドラッグします。



The screenshot shows the RapidMiner interface with several panels. On the left, the 'リポジトリ' (Repository) panel shows a list of data samples, with 'Titanic (v1)' selected. Below it, the 'オペレータ' (Operators) panel shows a search for 'loop values' and the 'Loop Values' operator highlighted. In the center, the 'プロセス' (Process) panel shows a workflow with 'Retrieve Titanic' and 'Set Macro' operators, and a 'Loop Values' operator being dragged into the process. On the right, the 'パラメータ' (Parameters) panel shows the configuration for the 'Loop Values' operator, with 'Passenger Class' selected for the attribute and 'loop_value' for the iteration macro. A red arrow points from the 'Loop Values' operator in the operators panel to its instance in the process flow, and another red arrow points from the 'Loop Values' operator in the process flow to its parameters panel.

2. そのパラメータで”attribute”を”Passenger Class”に設定します。



□ステップ 4/10

ループの内側

EXPLANATION(説明)

“Loop Values”オペレータが二重になっていることに気が付きましたか。この形は、このオペレータは内部に他のオペレータを内蔵できることを意味しています。ダブルクリックでオペレータの内部に入れます。

ACTIVITY(アクティビティ)

ダブルクリックで” Loop Values”オペレータの内部に入ります。

EXPLANATION(説明)

“loop”の内部では、それぞれの旅客クラスに応じたループを実行するサブプロセスを定義することができます。オペレータ 内部のプロセスを参照している時、左上隅でプロセスの名前を確認することができます。プロセスパネルの上部には同様に、メインプロセスに戻るためのリンクも存在します。

□ステップ 5/10

現在のクラスの列のみ保持

EXPLANATION(説明)

階層化されたオペレータによって、非常に強力なプロセスを組むことができます。このシス

テムは後にモデルの検証やパラメータ設定の最適を実施する時に多く用いることになりま
す。今のところは、“loop”の内側で起こることを定義してみましょう。

ACTIVITY(アクティビティ)

1. “loop”オペレータをダブルクリックして、プロセスの内部に入ります。
2. “loop”オペレータの内部で以下の作業を実施していきます。
3. “Filter Example”オペレータを検索し、追加します。
4. 左端のポートと、オペレータのポートとを接続します。これでデータセットをオペレー
タへ流すことができます。
5. “Filter Examples”のパラメータ設定で、“filters”の「フィルタを追加」をクリック
し、“Passenger Class”, “equals”, “%{loop_value}”を設定します。



EXPLANATION(説明)

今マクロ設定した“loop_value”はオペレータ内部で使用できる、“Loop Values”オペレータに
よって定義されたマクロです。それぞれのループで、マクロがループの現在の属性値
(attribute value)に設定されます。“Filter Examples”などでもマクロを同じような方法で利
用できます。その場合、元々のデータセットの内、その時点で使用しているループの属性値
を持つ行のみを残します。

ロステップ 6/10

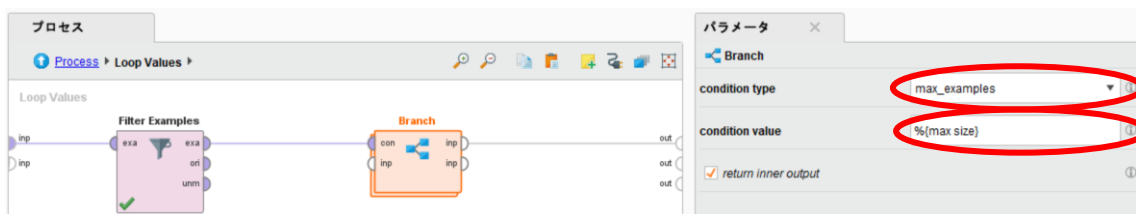
条件に基づくプロセスの分岐

EXPLANATION(説明)

本来のタスクを覚えていますか？乗客数が 400 以下である等級のサブグループを維持する一方で、乗客数が 400 を超えるサブグループをサンプリングすることです。そのためには分岐について学ぶ必要があります。

ACTIVITY(アクティビティ)

1. “Loop Values”オペレータ内部のままで、“Branch”オペレータをプロセスに追加します。
2. “Filter Examples”オペレータの exa ポートと、“Branch”の con ポートを接続します。
3. “Branch”オペレータ右側の inp ポートと、サブプロセスの out ポートを接続します。これでループを実行した結果をメインプロセスに流すことができます。
4. “Branch”オペレータをクリックしてパラメータを設定します。“condition Type”を”max_examples”に設定し、“condition value”に”%{max size}”と入力します。これは冒頭で定義したマクロになります。



□ステップ 7/10

“Branch”の内部

EXPLANATION(説明)

あなたは“Branch”オペレータも“Loop Values”と同様にサブプロセスを持っていることに気が付きましたか？ここでは“if – else”のロジックに従い二つのサブプロセスを構築します。それは“Branch”オペレータでの条件を満たした場合は最初の“Then”サブプロセスを実行し、そうでない場合は二つ目の“Else”サブプロセスを実行します。

ACTIVITY(アクティビティ)

“Branch”オペレータをダブルクリックして、オペレータ内部に入ります。

□ステップ 8/10

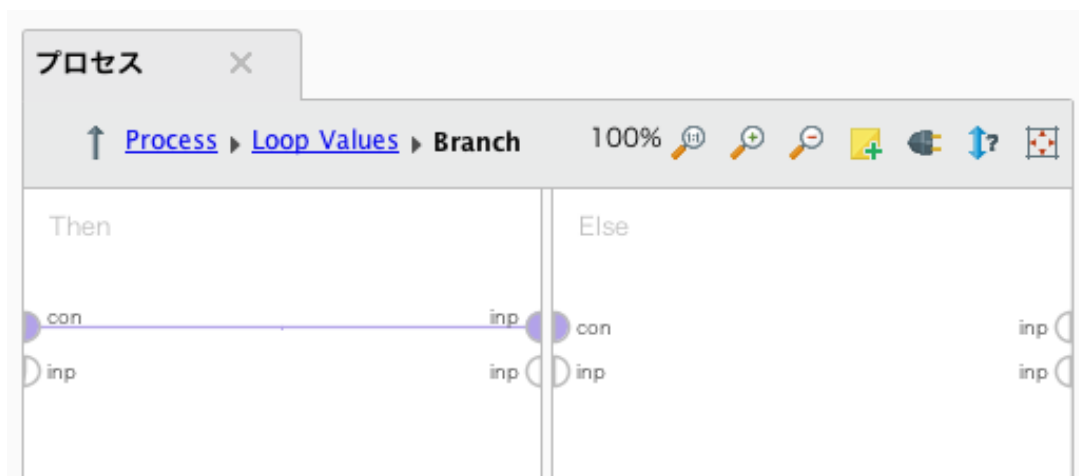
両方の場合の動作を設定

EXPLANATION(説明)

“Then”と“Else”という2つのサブプロセスを参照して下さい。現在のサブセットが指定した最大値よりも少ない場合は完全なデータ量を維持し、それ以外は最大値に合わせてダウンサイズ化します。

ACTIVITY(アクティビティ)

左側の“Then”サブプロセスで、右側のポートと左側(中央の)ポートを接続します。ここでは特定のオペレータを使用しません。

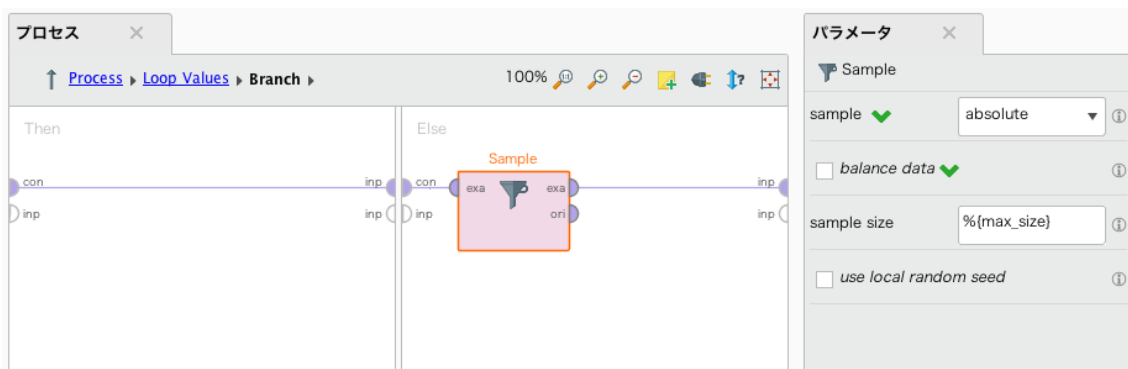


EXPLANATION(説明)

そのままの形でデータを残しておきます。つまりこの場合はデータに一切手を加えずにサブプロセスの結果として出力します。

ACTIVITY(アクティビティ)

1. 右側の“Else”サブプロセスに“Sample”オペレータを追加し、左右の inp ポートとオペレータをそれぞれ接続します。
2. “Sample”のパラメータで“sample_size”の欄を“%{max_size}”と入力します。


EXPLANATION(説明)

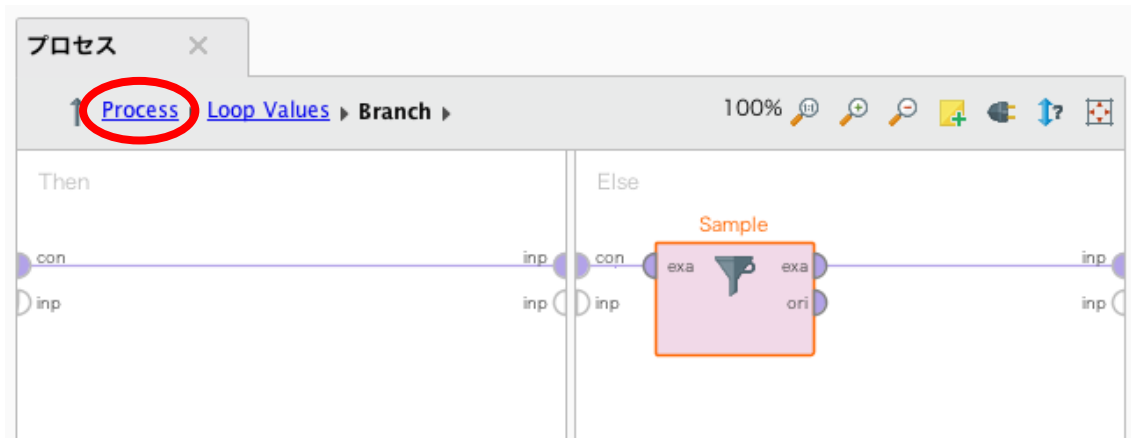
もし現在行われているループで乗客クラスのサンプルが大きすぎる場合は、目的に沿う大きさにダウンサイズ化してから、このサブプロセスの結果を出力します。

□ステップ 9/10

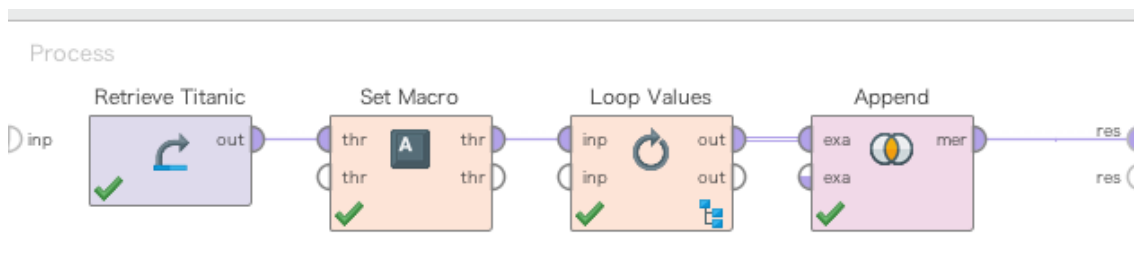
3つの結果を追加する。

ACTIVITY(アクティビティ)

1. プロセスパネルの上部のリンクから、プロセスに戻ります。



2. “Append”オペレータをプロセスに追加します。
3. “Loop Values”の out ポートと接続し、右側の res ポートとも接続します。



EXPLANATION(説明)

“Loop Values”と“Append”との間の二重線に気をつけてください。これは“Loop Values”のアウトプットが複数あることを示しています。ループは三つの乗船クラスそれぞれに実行されます。“Append” オペレータを追加した理由は、それぞれ単一のセットをもう一度一つのセットにまとめるためです。

ACTIVITY(アクティビティ)

プロセスを実行します。

EXPLANATION(説明)

データセットの合計サイズが削減され、数が 400 を超える乗船クラスがないことを確認してください（基本統計量タブを参照してください）。もしそうでない場合は原因を見つけるために、今までのチュートリアルの手順をもう一度確認してみてください。

▼ Passenger Class	Polynomial	0	Least Second (277)	Most Third (400)
-------------------	------------	---	-----------------------	---------------------

□ステップ 10/10

ステップのまとめ - おめでとうございます！

これまで作成した中で最も複雑なプロセスでした。このプロセスはマクロや、ループ、サブプロセスによるネスト化などで構成されています。これらの考え方と RapidMiner の数百ものデータ変換オペレータを組み合わせる事で、データの前処理に関する問題を解決する非常に強力なプロセスを構築する事ができます。そして一行ものコードを記述する必要はありません。

Challenge(追加質問)

- ・ "Loop Values" にブレークポイント(後)を配置しプロセスを再実行します。結果の左側にあるセクターに注目して下さい。ここから異なる乗船クラスのビューへと切り替える事ができます。Append で追加したデータを確認するにはプロセスを追加実行します。
- ・ "Loop" 内部の "Filter" にブレークポイント(後)を試してみてください。プロセスを実行する時、各ループでフィルタリングされた結果が表示されます。中間の実行結果を確認した後、また実行を継続する事ができ、次のループが実行されます。必要がなくなればすべてのブレークポイントを削除してください。
- ・ 各乗船クラスの最大値が 200 行になるようにプロセスを変更してみてください。
- ・ 3つの乗船クラスの行数を計算し、最小の行数は何行かわかりますか？また、すべての乗船クラスが同じ数になるように、最小数まで他の乗船クラスのダウンサンプリングを行ってみてください。

3.6 Writing Data

□ステップ 1/4

データのエクスポート

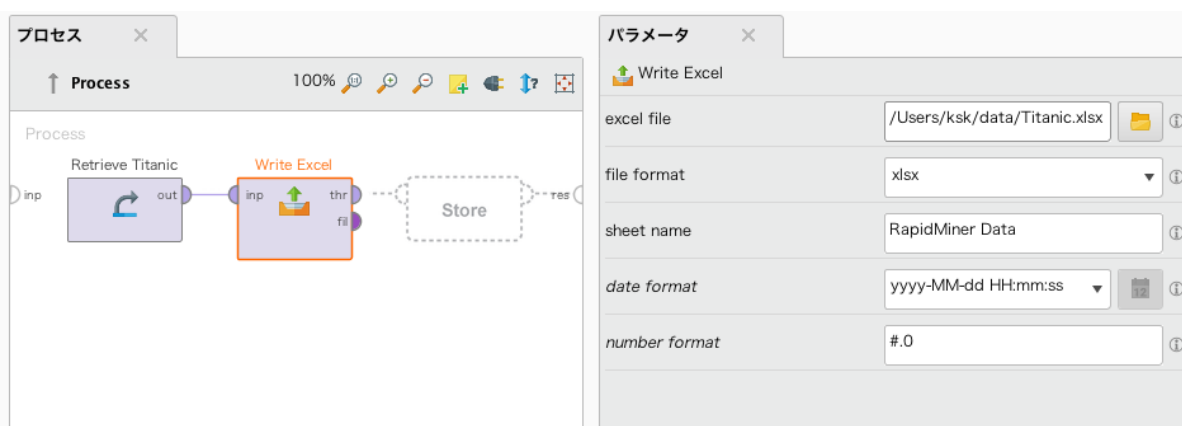
チュートリアルでは全般的に RapidMiner 上でのデータ操作に焦点を当てていますが、RapidMiner からデータを転送するにはどうしたらいいのでしょうか？コラボレーションや、高度な分析、データの冗長化などを実行するためにデータのエクスポートは必要不可欠な作業です。このチュートリアルでは、RapidMiner から簡単にデータをエクスポートする方

法をお見せしましょう。

ロステップ 2/4

ACTIVITY(アクティビティ)

1. “Titanic”のデータをプロセスにドラッグします。
2. “Write Excel”オペレータを検索し、プロセスにそれをドラッグします。
3. オペレータ同士を接続します。
4. “Write Excel”オペレータのパラメータで、エクセルファイルの場所を指定します。指定した場所に書き込める権限があることを確認します。



EXPLANATION(説明)

・エクセルファイルのパスを任意の場所へ指定できます。オペレータはタイタニックデータを取り出して、完全なデータセットをファイルに書き込むことができます。他にも異なるファイル形式を扱うオペレータが数十種類あり、データベースへ書き込むことも可能です。加えてCRMやERPといったビジネスアプリケーションにもデータを格納する事が可能です。この連携機能は非常に重要です。予測モデルに則ってビジネスアクションを行うことを、私たちはモデル運用と呼びます。これがモデルから価値を創造する一番の方法です。

・より多くのオペレータが必要になった際は、オペレータ欄下部の“Get More Operators”をクリックします。そうすると RapidMinerMarketplace にアクセスする事ができ、そこには数百もの追加オペレータが提供されています。

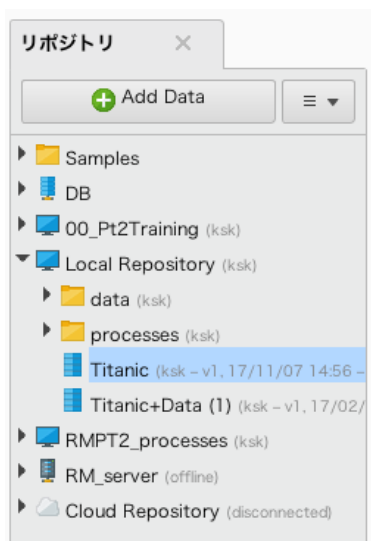
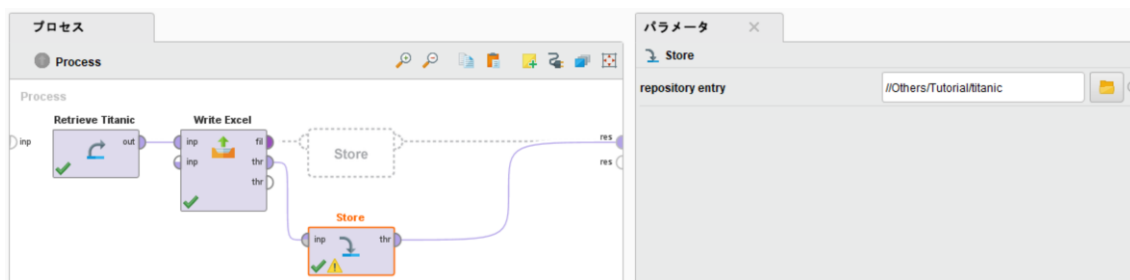
□ステップ 3/4

EXPLANATION(説明)

RapidMiner はリポジトリにメタデータを蓄積するので、出来る限りリポジトリやデータベースの使用をお勧めします。メタデータはプロセスの構築などに使われます。例えば、RapidMiner は選択ダイアログで別の属性名を表示することも出来ます。インポートする前に、リポジトリに蓄積したデータを確認する事ができます。オペレータを使って、リポジトリにデータ、モデル、他の結果を蓄積する方法を身につけていきましょう。

ACTIVITY(アクティビティ)

1. プロセスに”Store”オペレータを追加し、接続します。
2. パラメータ欄の”repository entry”を編集し、あなたのローカルリポジトリの場所を選択します。
3. 右側の”res”ポートに接続し、”Store”のデータを出力します。
4. プロセスを実行します。



□ステップ 4/4

ステップのまとめ - おめでとうございます！

これで、先ほど書いた Excel ファイルと、新しいリポジトリのエントリも確認してみてください。これで二つ目のコースは終了です。これで RapidMiner を使ってすべてのデータ処理タスクを解決する事が出来るようになりました。もちろん、時々解決方法を模索する事が困難な場合もあるかと思いますが、諦めないでください。何度も繰り返し、ツールボックスにある何百ものオペレータの使い方に慣れることで、問題への対処方法はより上手になります。サポートが必要な場合は、私たちのサポートページを是非参考にして下さい。

Challenge(追加質問)

1. “Write Excel”オペレータの代わりに”Write CSV”オペレータに置き換えてみてください。
2. ところで最後の”res”ポートに接続する必要はあるのでしょうか？この接続を削除しもう一度実行すると結果はどうなりますか？接続は線を選択した状態で Delete キーをクリックするか、右クリックしてメニューから削除を選ぶ、Alt キーを押しながら線をクリックすると削除することができます。

4. Build a model コース

4.1 Modeling

□ステップ 1/4

予測モデルの構築

ここまでは、最も重要であるデータの取り扱い方を学び、それを柔軟に組み合わせてプロセスを作る方法を見てきました。さて、これから予測モデルを構築しましょう。予測モデルを使用し、新規データの各レコードへの予測値を追加します。そして、新しい状況でモデルがどれくらいよく機能するかを検証しましょう。

EXPLANATION(説明)

予測モデリング(Predictive Modeling)とは大規模なデータセットから規則性を探しだし、その規則性から将来の状況の予測を作成するといった機械学習技術の集合体です。これらの予測は分類型(これを分類学習(classification learning)と呼びます)か、あるいは数値型(これを回帰学習(regression learning)と呼びます)になります。これらのモデルのタイプは結果を

導く基本的なプロセスを理解するのに非常に適しています。

このチュートリアルでは、決定木モデル、ルールセット、ベイズモデルという三つの分類モデルをタイタニック号のデータを用いて作成します。これらのモデルを探索することで、事故についてより多くのことを知り、誰が最も生存の可能性が高かったのかを確かめましょう。

EXPLANATION(説明)

これは RapidMiner のチュートリアルの 3 段階目です。これまでのチュートリアルで学習した概念を再び使っていきますので、それらを完全に理解しているかを確認することができます。楽しんでいきましょう。

□ステップ 2/4

“Titanic Training”データを取り込む

ACTIVITY(アクティビティ)

1. “Titanic Training”データを Samples リポジトリからプロセスヘドラッグする

EXPLANATION(説明)

“Titanic Training”データセットはモデルを学習する用にすでに準備されています。このデータセットは欠損値がなく、目的変数(label)はあらかじめ定義されています。目的変数(label)は予測したい属性(この場合は survived です)であることを覚えておいてください。この機械学習手法にはあらかじめラベルをもった学習データが必要です。このことから、このような手法を教師あり学習(supervised learning)と呼びます。

□ステップ 3/4

3つの異なるモデルの構築

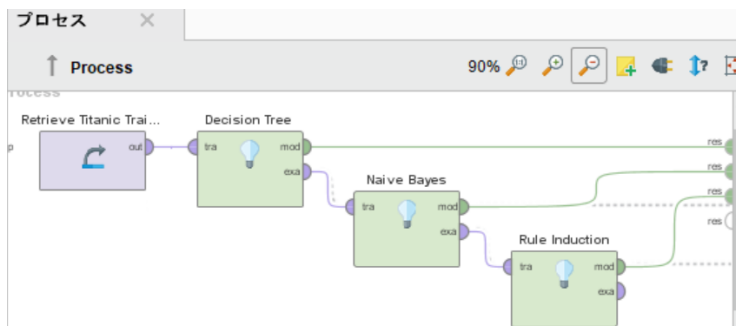
ACTIVITY(アクティビティ)

1. “Decision Tree”オペレータをプロセスにドラッグし、Retrieve Titanic Training の”out”ポートに接続する。
2. “Naive Bayes”オペレータをプロセスにドラッグし、そのインポートポートと”Decision Tree”オペレータの”exa”ポートを接続する。
3. “Rule Induction”オペレータをプロセスにドラッグし、そのインポートポートと”Naive

Bayes”オペレータの”exa”ポートを接続する。

4. 3つのモデリングオペレータの”mod”ポートをプロセス右側の結果ポート”res”に接続する。

その後青色の三角マーク(実行ボタン)を押してプロセスを実行する。



5. 3つの異なるモデルを詳しく見ましょう。(次ページ例参照)

概要 × RuleModel (Rule Induction) ×

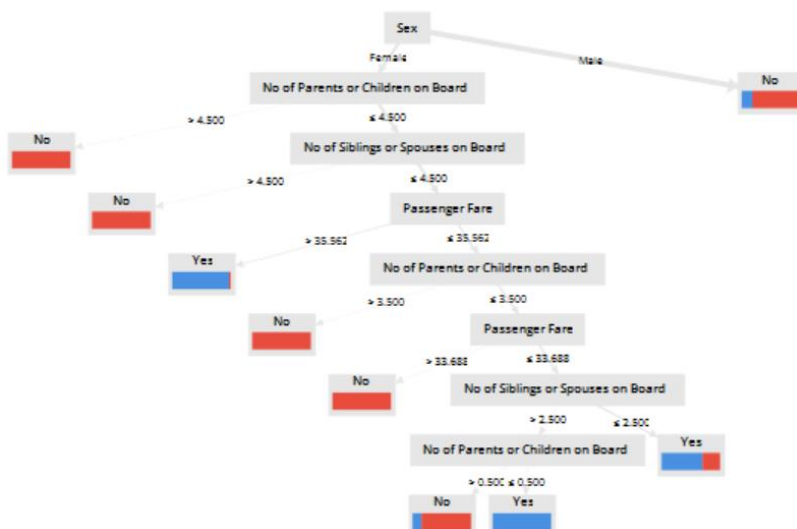
RuleModel

```

if Sex = Male and Passenger Fare ≤ 26.269 then No (57 / 367)
if Sex = Female and Passenger Class = First then Yes (37 / 4)
if Sex = Male and Passenger Fare > 31.137 then No (33 / 90)
if Passenger Class = Second and Age ≤ 28.500 then Yes (36 / 4)
if Passenger Fare ≤ 24.808 and Passenger Fare > 15.373 and Age > 29.441 then Yes (18 / 3)
if Passenger Fare ≤ 14.281 then Yes (68 / 40)
if Passenger Class = Third and Passenger Fare > 23.746 then No (1 / 23)
if Passenger Class = Second and Passenger Fare > 30.375 then Yes (4 / 0)
if No of Parents or Children on Board ≤ 0.500 and Age ≤ 30.441 and Passenger Fare ≤ 28.710 and Age >
if Age ≤ 54 then Yes (33 / 22)
if Age ≤ 71 then No (0 / 6)
else Yes (0 / 0)
    
```


correct: 750 out of 915 training examples.

SimpleDistribution (Naive Bayes) Tree (Decision Tree)




結果概要 RuleModel (Rule Induction) Simp

結果概要
RuleModel (Rule Induction)
Simp



Description



Charts

SimpleDistribution

Distribution model for label attribute Survived

Class Yes (0.381)
6 distributions

Class No (0.619)
6 distributions

EXPLANATION(説明)

決定木モデルは、女性にとって乗船クラスよりも家族構成の大きさの方が重要であることを明白に示しています。この行動パターンは男性には見られません。一般的に男性は生存確率が低いです(女性と子供が優先されるからです)。このことを最も手早く確認する方法がナイーブベイズモデルの可視化グラフの中にあります。一般的には正確性の高いモデルの型ではないのですが、ルールセットは読みやすいフォーマットであり、モデルを解釈するのに役立ちます。

□ステップ 4/4

RapidMiner で異なるモデルを作成することがいかに簡単であるかということがお分かりいただけただけかと思います。次のチュートリアルではスコアリング、すなわち新規データの結果を予測する方法を学びます。それに加え、モデルの予測がどれほど正確であるのかを測る方法も考察します。次のチュートリアルに進む前に、以下のチャレンジにも取り組んでみてください。

CHALLENGE(追加質問)

1. ナイーブベイズモデルのアウトプットである正規分布モデル(Simple Distribution model)のグラフを見てください。様々な属性を選択し、グラフを詳しく見てください。どうしてその中のいくつかは棒グラフで、またいくつかは線グラフなのですか？
2. 生存者と死亡者の間で最も顕著な違いが見られる属性はどの属性だと言えますか？
3. この発見と決定木モデルにおける最も違いが顕著な属性、ルールセットにおける最も違いが顕著な属性を比較してください。それらは同じですか？このことから何がわかりますか？

4.2 Scoring

□ステップ 1/5

予測モデルを構築する。

前回のチュートリアルでは予測モデルがどのようにしてデータについての見解を示すことができるのかを実証しました。ただデータを見るだけで、生存に関する性別と家族構成の大きさの影響を確認することはできないでしょう。このチュートリアルでは、この見解を利用して将来の結果を予測する方法について考察します。より詳しくいうと、ナイーブベイズモデルを使用して、それぞれの乗客の"Survived"クラス(yes/no)を予測し、それぞれの信頼性を確認します。

EXPLANATION(説明)

新規データへの予測にモデルを使用することがスコアリング(Scoring)と呼ばれます。

□ステップ 2/5

モデルを訓練する

ACTIVITY(アクティビティ)

1. “Titanic Training”データをリポジトリの Samples からプロセスヘドラッグする
2. “Naïve Bayes”オペレータを追加して”Titanic Training”オペレータと接続してください。

EXPLANATION(説明)

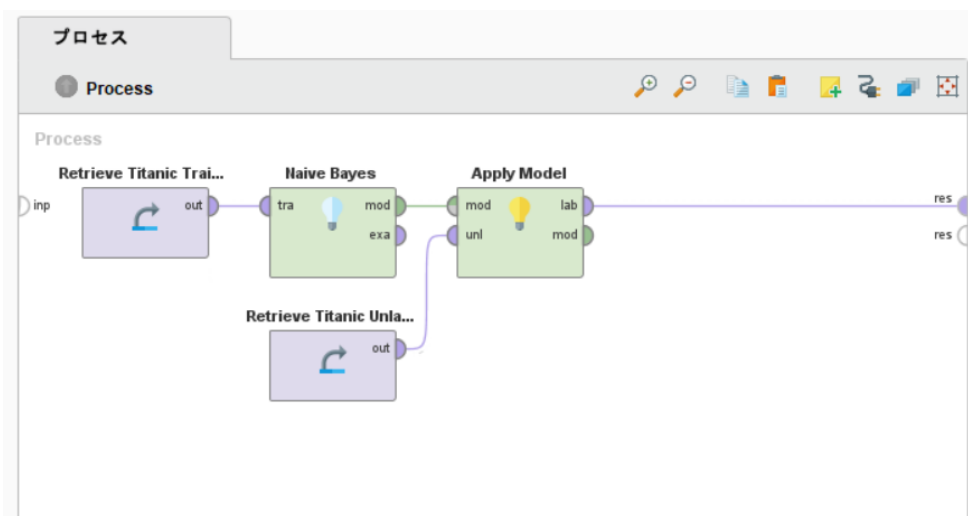
ここまでのプロセスは、以前確認した通り、単純にナイーブベイズモデルを構築しただけです。次は Apply Model と呼ばれる、新規でラベルなしのデータセットについて予測を作成するオペレータを使用する必要があります。

□ステップ 3/5

モデルを適用する

ACTIVITY(アクティビティ)

1. “Apply Model”オペレータを検索し、プロセスにドラッグしてください。
2. リポジトリの Samples から”Titanic Unlabeled”データをプロセスにドラッグします。
3. Naive Bayes オペレータのアウトポートポート”mod”(緑色)を Apply Model オペレータの緑色のインポートポートに接続してください。
4. ラベルなしのデータを Apply Model オペレータのインポートポート”unl”(青色)に接続してください。
- 5.最後に、Apply Model オペレータのアウトポートポート(青色)をプロセスパネル右側の結果ポート”res”に接続してください。



EXPLANATION(説明)

Apply Model オペレータの"unl"ポートと"lab"ポートが何を意味するのか不思議に思っていることでしょう。Apply Model オペレータはラベルなしデータ (unlabeled data) を取得し、"mod"ポートに接続したモデルを適用します。そして予測がモデルによって作成され、ラベル(label)付きのデータが出力されます。

EXPLANATION(説明)

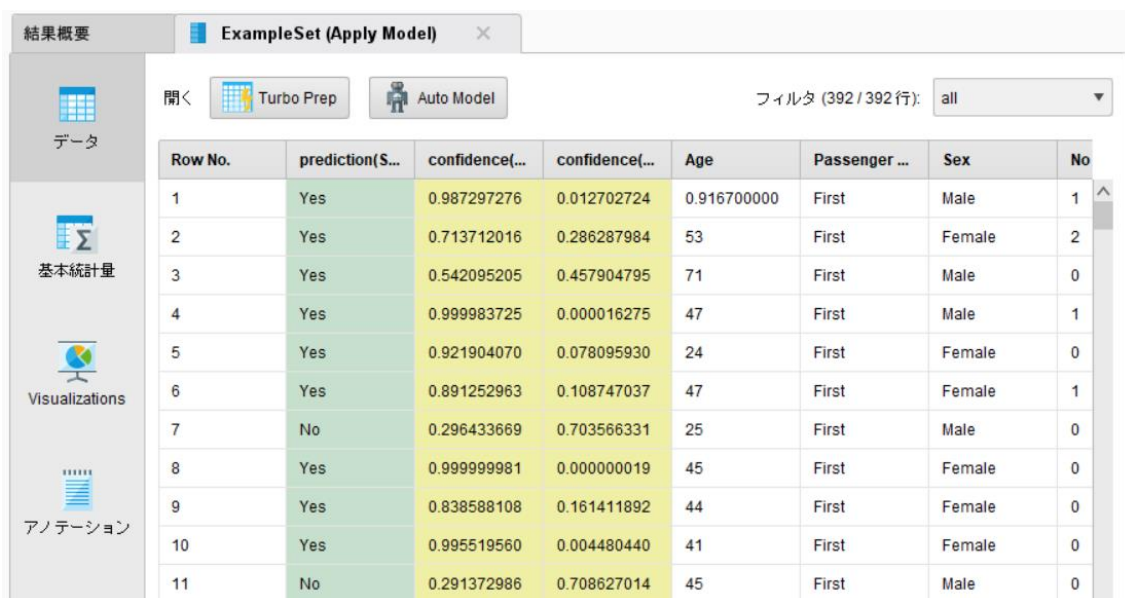
より高度な RapidMiner のオペレータを知るにつれて、オペレータのポートの機能がより重要になってきます。もしポートの機能に興味があるのなら、ポートの上にマウスポインタをしばらくおいてフルネームを確認しましょう。

ロステップ 4/5

プロセスを実行し、結果を確認する

ACTIVITY(アクティビティ)

1. プロセスを実行してください。
2. 結果を確認していきましょう。



Row No.	prediction(S...	confidence(...	confidence(...	Age	Passenger ...	Sex	No
1	Yes	0.987297276	0.012702724	0.916700000	First	Male	1
2	Yes	0.713712016	0.286287984	53	First	Female	2
3	Yes	0.542095205	0.457904795	71	First	Male	0
4	Yes	0.999983725	0.000016275	47	First	Male	1
5	Yes	0.921904070	0.078095930	24	First	Female	0
6	Yes	0.891252963	0.108747037	47	First	Female	1
7	No	0.296433669	0.703566331	25	First	Male	0
8	Yes	0.999999981	0.000000019	45	First	Female	0
9	Yes	0.838588108	0.161411892	44	First	Female	0
10	Yes	0.995519560	0.004480440	41	First	Female	0
11	No	0.291372986	0.708627014	45	First	Male	0

EXPLANATION(説明)

結果は元々のラベルなしデータに、"Survived"の予測されたクラス(yes/no)の列と異なる 2

つのクラス(yes/no)の信頼性(confidences)について 2 列を追加したものです。例えば 1 行目のデータを見ると、予測は"yes"が約 98.7%の信頼性を持っていて、"no"の信頼性は約 1.3%です。

□ステップ 5/5

ステップのまとめ - おめでとうございます！

素晴らしい。予測モデルを使用してデータをスコアリングすることは、Apply Model オペレータで簡単に行うことができます。ラベルなしデータのフォーマットが学習データのフォーマットと同じであることを確認することを忘れないでください。同じ属性を使用し、そしてできるならば同じ範囲の値を使用してください。大きなデータの変化にも対応できる頑健性のあるモデルがある一方で、容易に機能しなくなるモデルもあります。

CHALLENGE(追加質問)

1. 生存確率が最も高いケースを示すようにデータを並べ替えてください。生存確率が最も高い 10 人のうち何%が女性ですか？
2. Naive Bayes オペレータを Decision Tree オペレータに置き換えてください。これは予測の信頼性にどのような影響を与えますか？なぜそうなるのか想像できますか？

4.3 Test Sprints and Validation

□ステップ 1/5

学習データとテストデータに分割する

予測モデルを構築した後に尋ねるべき最も重要な質問は、「このモデルはどのくらいよく機能するのか？」です。いまだかつて出会ったことがないようなシナリオに対して、作成したモデルがよく機能するかどうかをどのようにして答えることができるでしょうか？これを正確に行う方法はいつだって同じです。それはラベル付きのデータのうちいくつかを取っておき、モデル構築に使用しない方法です。このデータはまだラベル付きなので、予測と実際の結果を比較することができます。そしてどのくらいモデルが正確であったかを計算することもできます。このチュートリアルでは、どうやってこの検証を実行することができるのかをみていきましょう。

EXPLANATION(説明)

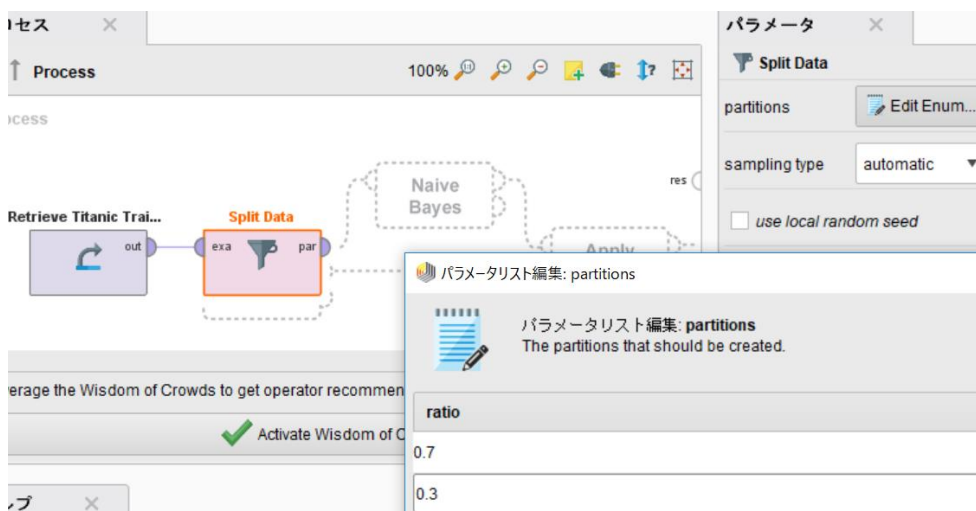
学習データについても、どれくらいの頻度でモデルが正確な予想をしたのかを計算することができると主張する人がいますが、この「学習エラー」を使用することはよくない考えだと思います。これをすると、モデルが単純にデータを記憶し、新しいケースに対する発見を生み出す方法を学習しなくなります。新規データに対しては機能せず、学習データについては100%正確であるモデルに何の価値があるのでしょうか？

□ステップ 2/5

ラベル付きデータを2つに分割する

ACTIVITY(アクティビティ)

1. Titanic Training データを Samples リポジトリからプロセスヘドラッグする
2. Split Data オペレータをプロセスに追加して Titanic Training データに接続する
3. Split Data オペレータのパラメータ内にある分割(partitions)を見つけ、Edit Enumeration... をクリックしてください
4. Add Entry を2回クリックし、1つ目のテキストボックスに0.7を、2つ目のテキストボックスに0.3を入力してください



EXPLANATION(説明)

Split Data オペレータはデータセットを設定した割合で分割します。この場合では一方は70%、もう一方は30%で2つに分割します。両方のデータセットはともにラベル付きのまま

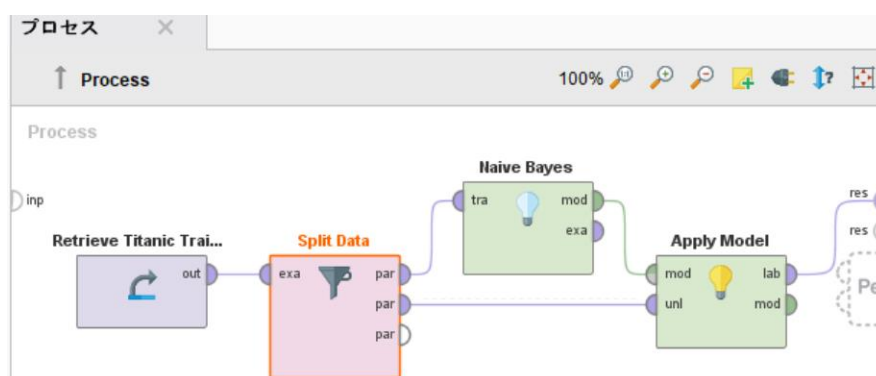
ます。70%の方はモデルを構築する学習データになります。残りの 30%はモデルの予測と比較することのできるテストデータになります。この学習データとテストデータの 70/30 の比率は実際はかなりよく使われており、有効な値です。

□ステップ 3/5

モデルの学習と適用

ACTIVITY(アクティビティ)

1. Naive Bayes オペレータをプロセスに追加して、Split Data オペレータの最初の出力ポートに接続してください
2. Apply Model オペレータをプロセスに追加してください
3. Naive Bayes オペレータの緑色のモデルポート(mod)から Apply Model オペレータの入力側のモデルポートに接続してください
4. Split Data オペレータの 2 つ目のアウトポートポートを Apply Model オペレータの"uni"ポートに接続する



EXPLANATION(説明)

この時点でプロセスを実行すると、スコアリングについてのチュートリアルで出てきたものと似た結果が得られます。30%のテストデータに、生存予測の列と"yes"と"no"の 2 つのクラスに対するそれぞれの信頼性の列が追加されています。しかし今回の場合、テストデータは実際の値をもつラベル列も持っているので、ただラベルと予測を比べるだけで正確度 (accuracy) を計算することができます。

□ステップ 4/5

モデルの正確度を計算する

ACTIVITY(アクティビティ)

1. Performance オペレータをプロセスに追加してください。
2. Performance オペレータの"lab"入力ポートを Apply Model オペレータの"lab"出力ポートに接続してください
3. Performance オペレータの両方の出力ポートを右側の結果ポートに接続してください
4. プロセスを実行し、結果を詳しく調べてください

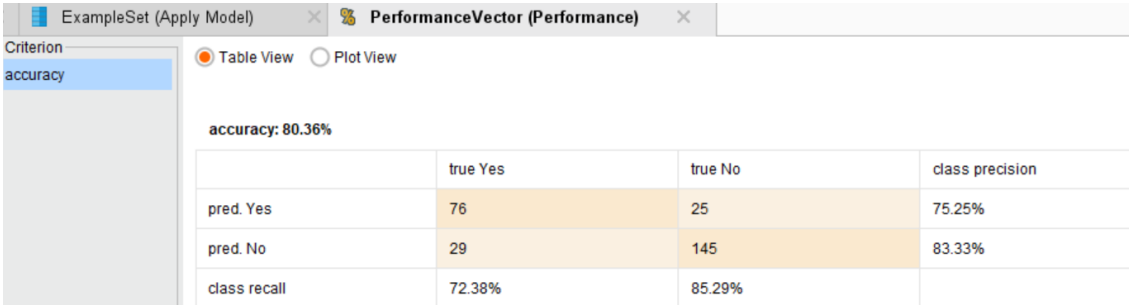


Table View

accuracy: 80.36%

	true Yes	true No	class precision
pred. Yes	76	25	75.25%
pred. No	29	145	83.33%
class recall	72.38%	85.29%	

EXPLANATION(説明)

最初に確認できる結果は目的変数と予測を含んだテストデータです。2つ目の結果はテストデータにおけるモデルのパフォーマンスです。画面の左側にある"critereion"で様々なパフォーマンスの測定値を選択することができます。正確度(accuracy)は 80.36%で、全体的に見てモデルがどれくらい正確であるかがわかります。混同行列(confusion matrix)は様々なエラーを示します。例えば、実際は"yes"なのに"no"と予測されたものが29ケースありました。正確度(accuracy)とは、左上と右下という対角線上の数字の和をすべての数の和で割ったものです！この対角線の数字が大きければ大きいほど、モデルのパフォーマンスは良くなります。

□ステップ 5/5

ステップのまとめ - おめでとうございます！

素晴らしいです。あなたはラベルありのデータを学習データとテストデータに分割できました。そしてモデルがどれくらいよく機能するのかを計算する Performance オペレータを使用しました。下の追加質問に取り組んでみてください。

CHALLENGE(追加質問)

- ・テストデータのデータビューに戻ってください。右上にフィルタがあります。フィルタを

wrong prediction に設定して、データそのものがどのように変化するかだけでなく、選択ボックスの横の数字の変化にも注目してください。テストデータは合計で何行ありましたか？そして間違った分類は何行ありましたか？

- ・今度は混同行列に注目してください。間違った分類は合計何行ありますか？この数字は上で見たものと合致しますか？
- ・ Split Data オペレータにブレイクポイント(後)を設定し、もう一度プロセスを実行してください。幾つのデータセットが得られますか？それらのサイズは何ですか？
- ・ Performance オペレータを Performance(Classification)に置き換えてください。パラメータ内の正確度(accuracy)、分類誤差(classification error)、平均平方二乗誤差(root mean squared error)を選択してください。分類誤差は何になるか予想してください。再実行してこれを検証してください。

4.4 Cross Validation

□ステップ 1/6

正解度のより良い測定

先ほどのチュートリアルではデータをトレーニングセットとテストセットに分割して、モデルの正解度の測定値を作成することがいかに簡単かを見ました。しかし、トレーニングセットとテストセットの間に大きな差があったとしたらどうでしょうか？現在のパフォーマンス測定では、「簡単」または「難しい」テストセットに基づいている可能性があります。このチュートリアルでは、各データが学習データにもテストデータにも同じ頻度で使用されるように交差検証(Cross Validation)と呼ばれるテクニックを紹介し、この問題を回避します。

EXPLANATION(説明)

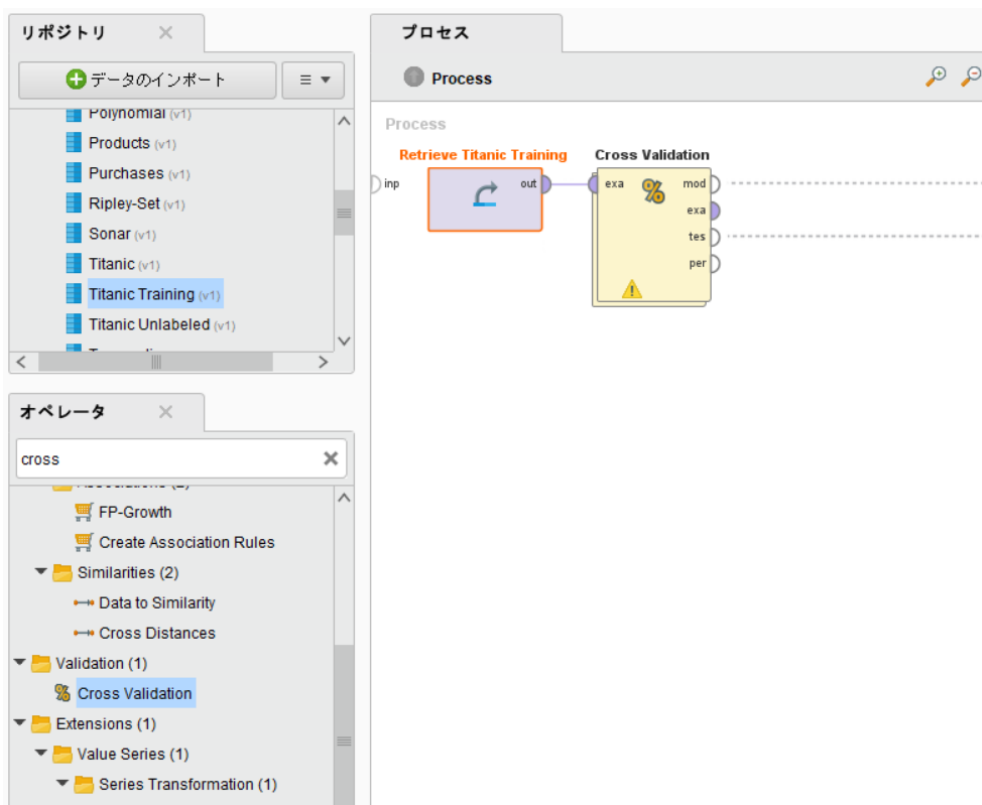
交差検証では、データセットを等分に分割し、すべての部分をローテーションで回しながら、常に一つをテスト用に、他をモデルの学習用に使用します。最後に、全テストの正確度の平均値が結果として得られます。これはモデルの正確度を計算するための素晴らしい方法であり、計算する時間をかけられる場合はいつでも標準的な測定手法とすべきです。

□ステップ 2/6

データを読み込み、検証にかけます

ACTIVITY(アクティビティ)

1. "Titanic training"データをプロセスにドラッグします。
2. "Cross Validation"オペレータを加え、接続します。



EXPLANATION(説明)

“Cross Validation”は、交差検証を実行するオペレータの名前で、ラベル付きのデータセットが必要です。デフォルトでは、データを10個に分割するため、10-fold cross validationとも呼ばれます。もちろん、パラメータ画面で分割数を変更することができます。

□ステップ 3/6

モデルの訓練と適用

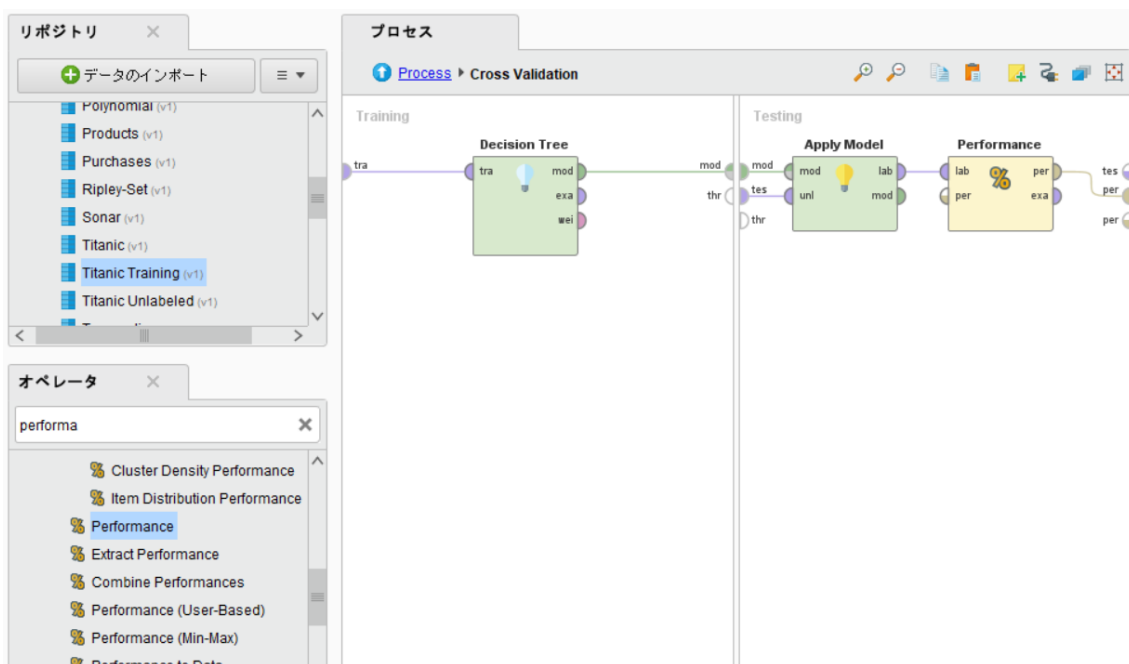
EXPLANATION(説明)

“Cross Validation”の右下の小さなアイコンに注意してください。これはこのオペレータに層があることを意味していることを思い出してください。実際に“Cross Validation”には二つの

サブプロセスがあり、一つはモデルの訓練、もう一つはモデルの検証のためにあります。ダブルクリックして、サブプロセスの中を見てみましょう。

ACTIVITY(アクティビティ)

1. "Cross Validation"をダブルクリックします。プロセスパネルに"Training"と"Testing"という二つのサブプロセスが現れます。
2. "Decision Tree"を"Training"サブプロセスに追加します。
3. 左側の"tra"ポートとオペレータの"tra"ポートとを、オペレータの"mod"ポートと右側の"mod"ポートとを接続します。
4. "Testing"サブプロセスに"Apply Model"を追加します。
5. 左側の"mod"と"tes"ポートを"Apply Model"に接続します。
6. "Testing"に"Performance"を追加します。
7. "Apply Model"と"Performance"をつなぎ、オペレータと右側の"per"ポートを接続します。



EXPLANATION(説明)

右側の "per"ポートに送信される全てのパフォーマンスは、平均化され、"Cross Validation"オペレータの "per"出力ポートに流されます。また、"Cross Validation"の内部の動きを完全に制御できることにも注意してください。これにより、正規化や特徴選択のような前処理の効果を考慮することができ、モデルのパフォーマンスに大きな影響を与えることができま

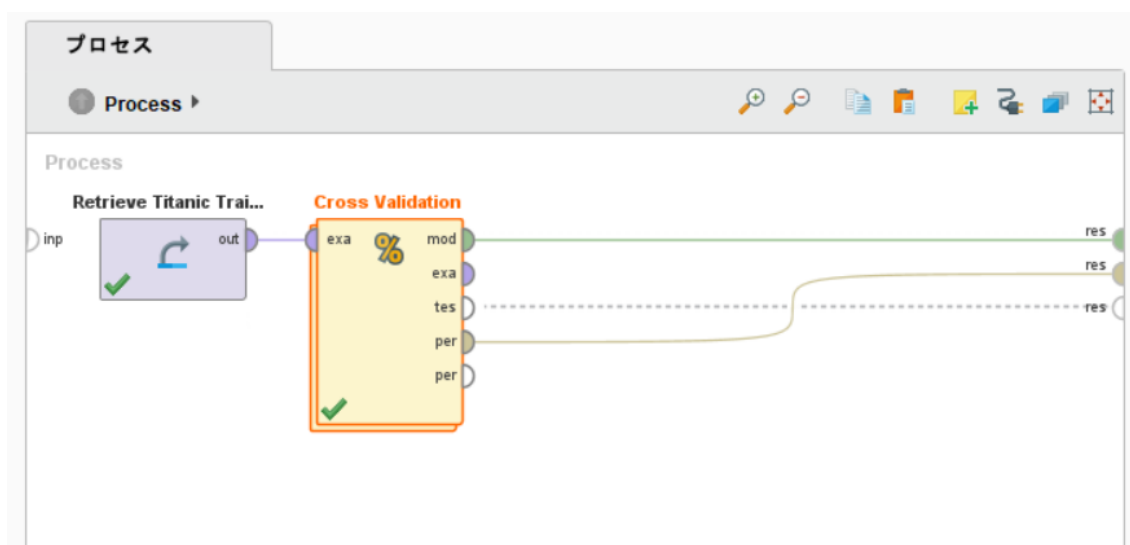
す。

□ステップ 4/6

結果ポートと接続し、プロセスの実行

ACTIVITY(アクティビティ)

1. プロセスパネル左上部の上矢印か"Process"をクリックし、メインプロセスに戻ります。
2. 緑の"mod"と黄の"per"を右側の"res"ポートにつなぎます。
3. プロセスを実行し、結果を確認します。



EXPLANATION(説明)

- ・ 正確度には、交差検証のパフォーマンスの標準偏差を示す数値 (" +/- "の後です) が追加されていることに注意してください。標準偏差はモデルがどれだけ頑健なのかを教えてください。標準偏差が小さければ小さいほど、モデル性能のテストデータへの依存度は低くなります。

- ・ 出力されるモデルは、全データセットで学習したモデルであり、交差検証内のモデルの一つではありません。都合の良いものが出ていますが、交差検証はあくまでもモデルの正確度を推定するためのものであって、最良のものを構築するためのものではないことを覚えておいてください。一般的には、可能な限り多いデータでモデルを構築する必要があります。


□ステップ 5/6

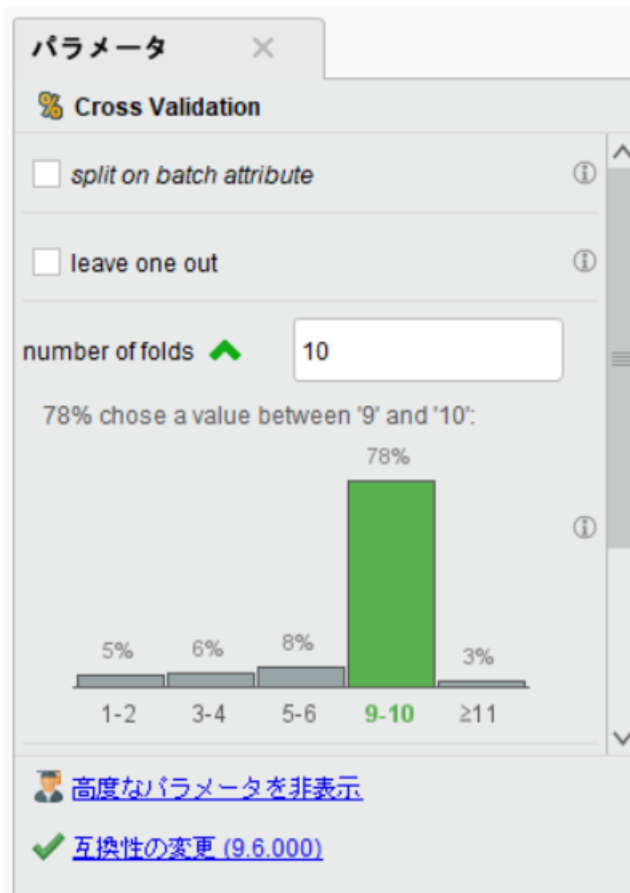
群衆の知恵を活用

EXPLANATION(説明)


このステップは、インターネットに接続しており、RapidMiner コミュニティのユーザーとして登録している場合にのみ有効です。群衆の知恵をフルに活用するために、今すぐ実行することを検討してみてください。また、プロセスパネルの下部にある機能を有効にする必要があります。

ACTIVITY(アクティビティ)

1. "Cross Validation"オペレータをクリックして選択してください。
2. パラメータパネルを見てください。いくつかの項目に緑のチャレット  があることに気が付きましたか。このチャレットのついたパラメータは、RapidMiner のユーザーによって頻繁に変更されています。
3. number of folds の横にあるチャレットをクリックします。小さなチャートが表示され、何人のユーザーがこのパラメータを変更したか、またそのユーザーが変更したパラメータの内容が表示されます。




パラメータ

 Cross Validation


split on batch attribute


leave one out

number of folds  10

78% chose a value between '9' and '10':

number of folds	Percentage
1-2	5%
3-4	6%
5-6	8%
9-10	78%
≥11	3%

 [高度なパラメータを非表示](#)

 [互換性の変更 \(9.6.000\)](#)

4. 交差検証の分割数のもう 1 つの典型的な値は 9 または 10 であることがわかります。バーをクリックして、これらの値のいずれかを自動的に選択するか、または手動で値を入力します。
5. プロセスを再実行します。

EXPLANATION(説明)

RapidMiner に緑のキャレットが表示されたら、いつでも「群衆の知恵」にアクセスすることができます。RapidMiner は、何十万人もの他のユーザーの習慣に基づいて、プロセスを最適化するための推奨事項を提供します。また、プロセスパネルの下部には、プロセスに追加するための提案されたオペレータが表示されます。ぜひお試しください。

□ステップ 6/6

ステップのまとめ - おめでとうございます！

教師あり機械学習モデルを適切に検証する方法を学びました。テストにすべてのデータを使用したという事実は、交差検証を一回の分割検証よりもはるかに強力にします。また、”Cross Validation”オペレータのモジュール設計により、前処理オペレータを内部に入れ子にすることができるため、より優れた測定が可能になります。しかし残念ながら、これにはリソースコストがかかります。10 倍の交差検証を行う場合、10 個のモデルをトレーニングする必要があります。これは、プロセスに余分な数値計算を追加し、必要な実行時間を増加させる可能性があります。

CHALLENGE(追加質問)

- ・ 交差検証は、データにラベルがない教師なし学習でも機能するのでしょうか？
- ・ 2 倍の交差検証が出来るようにプロセスを変更します。”Cross Validation”内にブレイクポイントを使用して、トレーニングセットとテストセットの現在のサイズを確認します。どのようなものになるのでしょうか？
- ・ 最適なモデルはどれでしょうか？決定木、ナイーブベイズ、ルール、ニューラルネットのどれが最適でしょうか？すべてのモデルを試してみて、どれが最も正確度が高いかを調べてみましょう。

4.5 Visual Model Comparison

□ステップ 1/5

視覚的パフォーマンス指標としての ROC 曲線

受信者動作特性 (ROC) 曲線は、バイナリ機械学習モデルがどれだけうまく機能するかを示します。これは、モデルのさまざまな信頼度のしきい値への偽陽性率 (FPR) に対する真の陽性率 (TPR) を示しています (詳細はリンク)。その結果、モデルが単なる推測であれば直線的な対角線となり、モデルが良くなればなるほど左上に向かって曲線がどんどん移動していきます。このチュートリアルでは、複数のモデルの ROC 曲線を作成する方法と、それらを視覚的に比較して、どれが一番良いのかを素早く識別する方法を紹介します。

EXPLANATION(説明)

ROC 曲線は、モデルの性能を可視化するためのよく知られた方法です。その背後にある数学を知らなくても心配しないでください。ただ、より良いモデルのカーブは左上に移動することを覚えておいてください。完璧なモデルは、まっすぐ上に (垂直に) 行き、右に (水平に) 直進する線を生成します。

□ステップ 2/5

データ読込

ACTIVITY(アクティビティ)

1. "Titanic Training"データをプロセスにドラッグします。
2. "Compare ROCs"オペレータをプロセスに追加し、接続します。

EXPLANATION(説明)

"Compare ROCs"は、もう 1 つのネストオペレータです。このオペレータの中に複数のモデルを配置して、各モデルの ROC 曲線を 1 つのグラフに作成することができます。

□ステップ 3/5

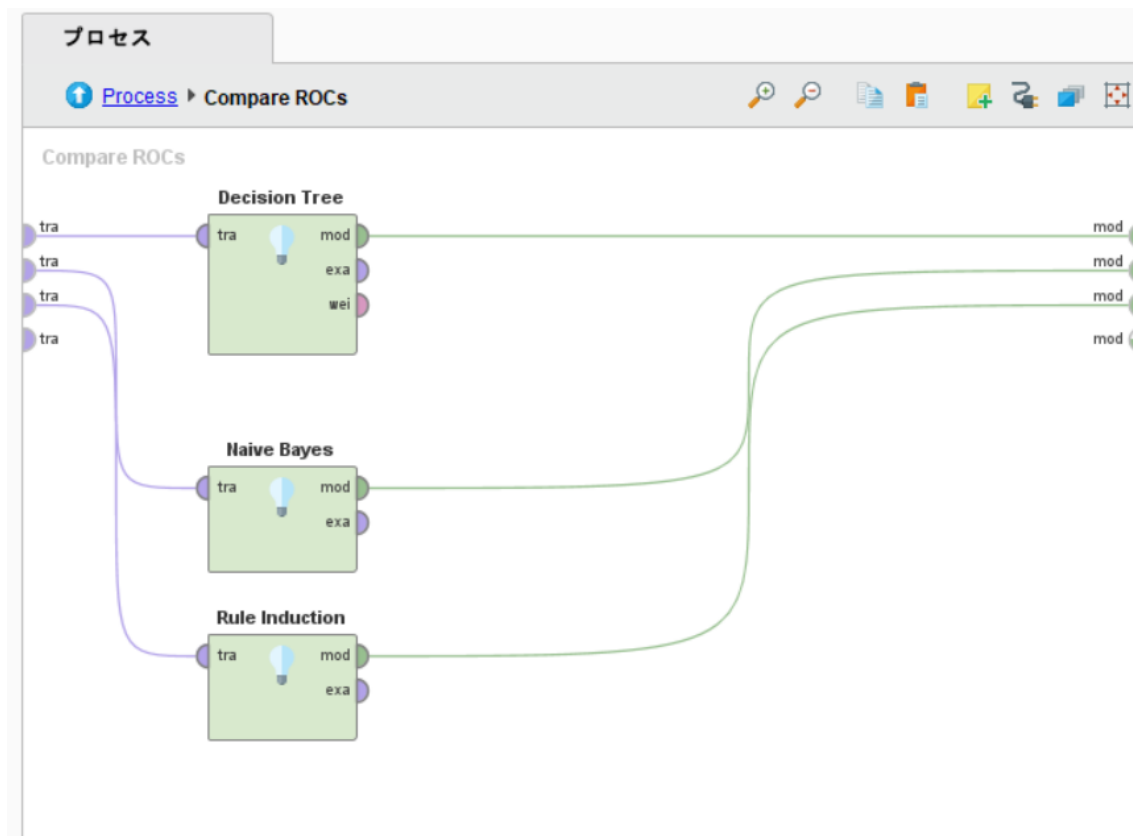
モデルの訓練と適用

ACTIVITY(アクティビティ)

1. "Compare ROCs"をダブルクリックします。"Compare ROCs"というサブプロセスが表示されます。
2. サブプロセスに決定木とナイーブベイズ、ルールインダクションのオペレータを追加し

ます。

3. 各学習オペレータの入力ポートと左側の"tra"ポートを接続します。一つ接続する度に新しいポートが現れます。
4. 学習オペレータの"mod"ポートを右側の"mod"ポート接続します。



EXPLANATION(説明)

“Compare ROCs”は実際には入力データへ設定されたモデルを用いて交差検証を行います。オペレータのパラメータで分割数を変更することもできます。分割が多いほど時間がかかることを覚えておいてください。このパラメータの値としては、一般的に 3~10 回の分割が良いでしょう。

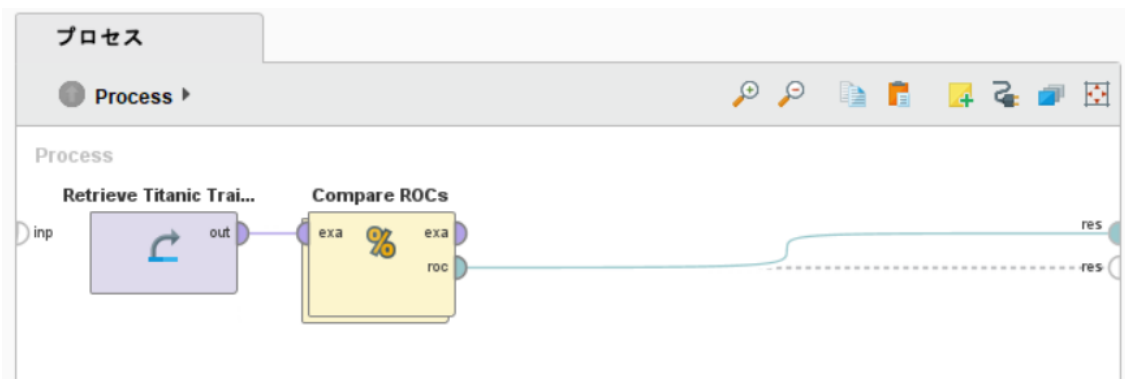
ロステップ 4/5

結果ポートと接続し、プロセスの実行

ACTIVITY(アクティビティ)

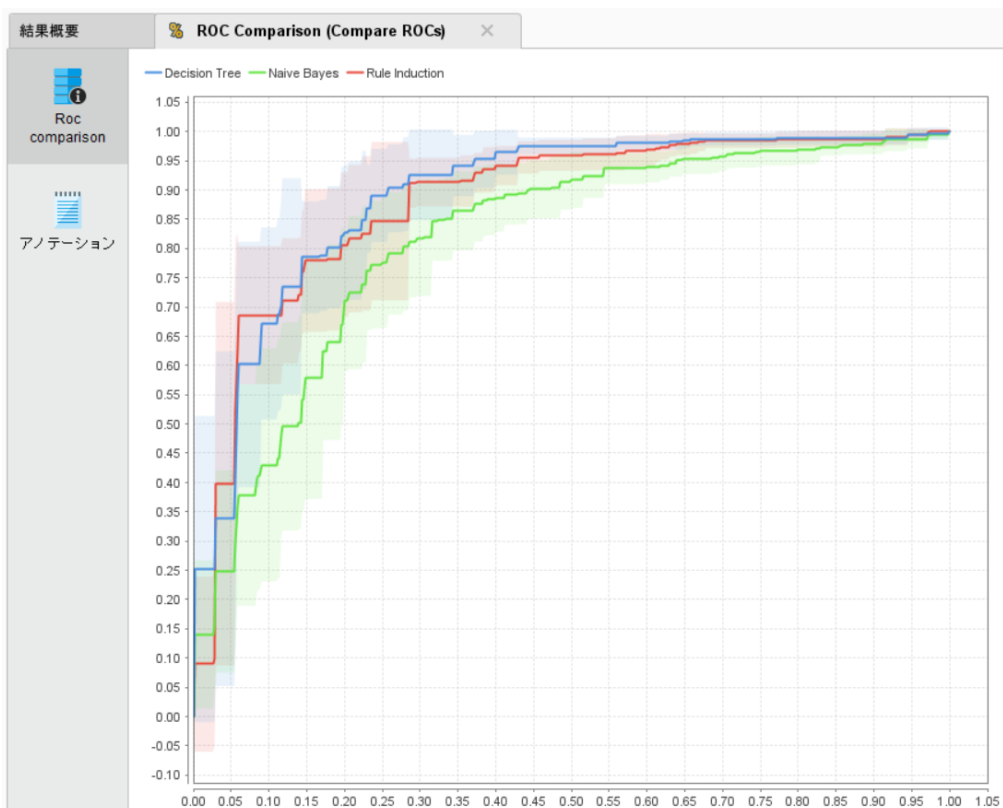
1. プロセスパネル左上部の上矢印が“Process”をクリックし、メインプロセスに戻ります。
2. “Compare ROCs”の“roc”ポートと右側の“res”ポートを接続します。

3. プロセスを実行し、結果を確認しましょう。



EXPLANATION(説明)

チャートを見ると、3種とも左上に向かってカーブしているのですが、当て推量よりも効果が高いことがわかります。この場合、ナイーブベイズは左上隅から最も離れており、このケースでは最悪の性能を発揮します。しかし、モデルの性能は、異なるデータセットでは全く違うかもしれません。



□ステップ 5/5

ステップのまとめ - おめでとうございます！

このチュートリアルは終了しました。モデルを構築するための最も重要なテクニック、モデルを使って新しいテーブル(example sets)をスコア化すること、そしてそれらの精度を検証することを紹介しました。スコアリングは非常に簡単で、すべてのモデルタイプに同じオペレータ"Apply Model"を使用します。モデルを検証するための複数の方法を見てきましたが、交差検証は、実行時間の点では多少コストがかかりますが、予測精度の点では最高の測定を提供することを覚えておいてください。

CHALLENGE(追加質問)

- ・ number of folds を 10 個から 5 個に変更して、処理を再実行します。これにより、品質的にモデルの順番は変わりますか？
- ・ 各 ROC 曲線の周りの透明域の意味は何だと思いますか？
- ・ 曲線の視覚的な形を 1 つの値に変換する良い方法は、曲線下面積(AUC)のサイズを計算することです。この値も RapidMiner のいくつかの場所で見ることができます。完全な分類器の曲線下面積 (AUC) とは何でしょうか？ランダムな推測を行っているだけの分類器の AUC は何ですか？

以上

いかがでしたか？

日本語版チュートリアルの解説はここまでです。

独学でチュートリアルを実施しても消化不良の方は、同テキストを用いた講師付きの無償ハンズオンをぜひご受講下さい。さらに、発展的でより実務的でより詳しい内容をご希望の際は、トレーニング（3日間）をぜひご受講下さい。

詳細は[こちら](#)