データを一から分析してみよう(Part 2)

Kaggle で公開されているアイオワ州住宅価格のデータセットを使って、EDA（探索的データ解析）からモデル作成までのデータ分析プロセスを行なっていきます。Part1 は、データの読み込みと正しいデータ型へ修正するところまで行いました。Part2 では、データを確認しながら欠損値や外れ値の処理を行い、モデルを作成していきます。Kaggle のコンペティションに実際に参加し、作成したモデルの結果（順位）を確認するまでを掲載しています。

※本資料は、macOS、RapidMiner Studio Version9.10 を使用しています。

使用データ：Ames Housing dataset

アメリカ合衆国アイオワ州エイムズ都市の住宅に関するデータセット

https://www.kaggle.com/c/house-prices-advanced-regression-techniques/overview

外装素材、地下室の高さ、ガレージのサイズなど、79 の属性（説明変数）

各属性（説明変数）の特徴は data_description.txt を参照

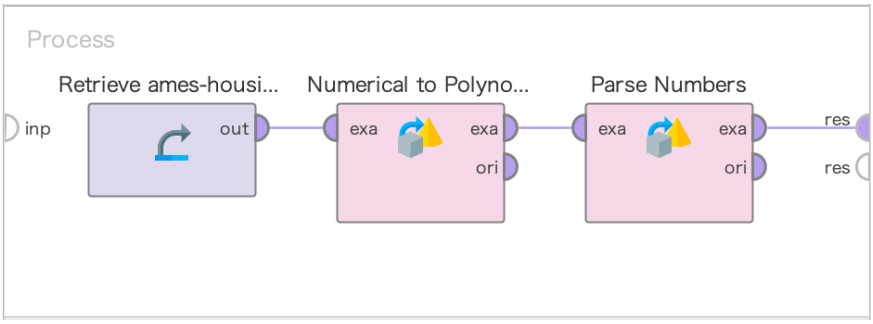
分析の目的：各住宅の販売価格を予測すること

<Part2 で行うプロセス>

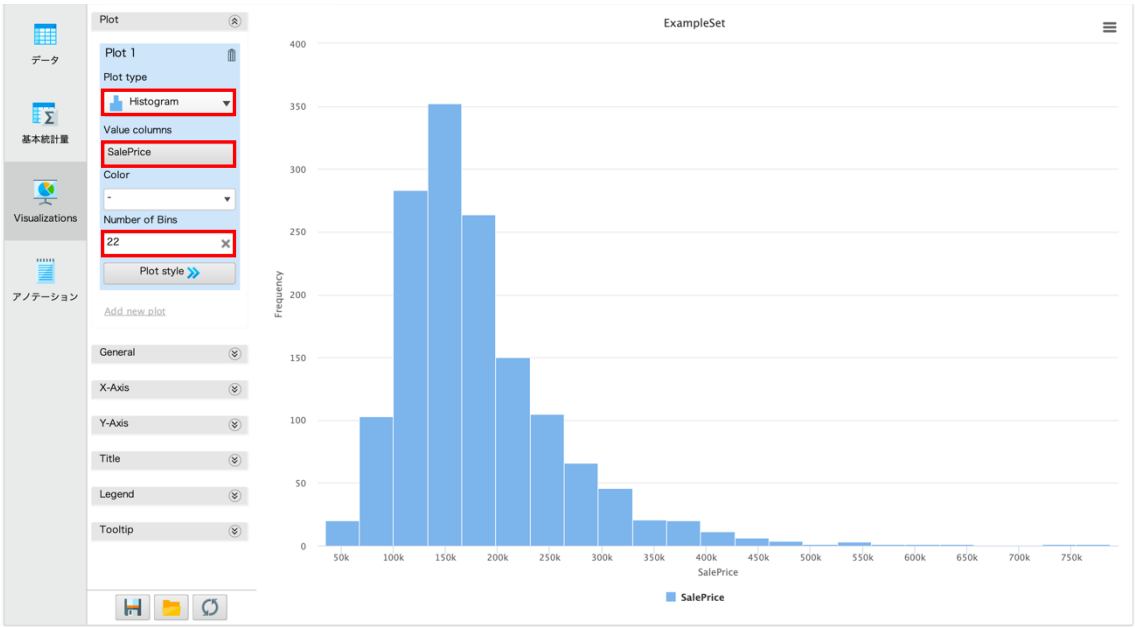
■ データの理解
■ 欠損値の処理
■ 目的変数との相関
■ 説明変数同士の相関
■ 外れ値の処理
■ モデル作成・適用
■ Kaggle のコンペティションへ参加

データの理解

Part1 で作成した以下のプロセスを実行し、データの特徴や欠損値の有無などを確認します。



▽　目的変数(SalePrice)のヒストグラムから、データの特徴を把握します

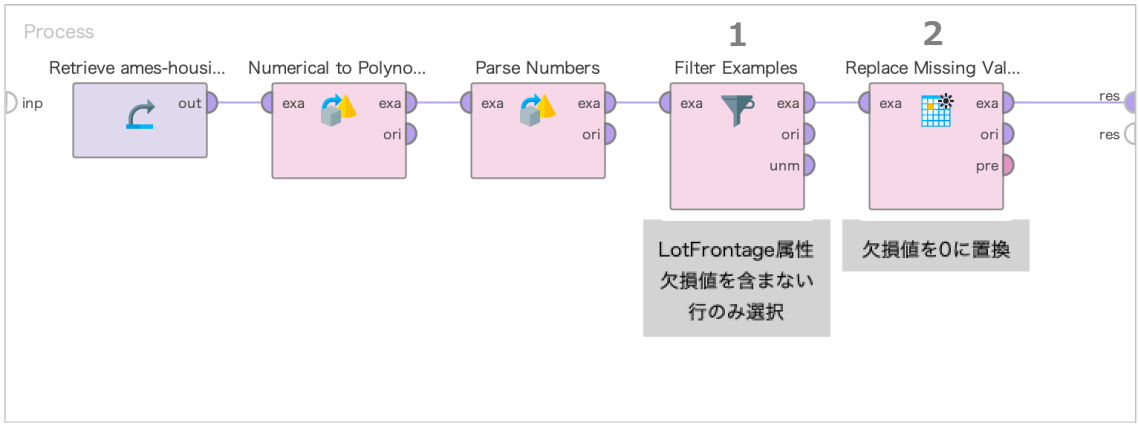


データの約 60%が 100,000 ドル～200,000 ドルを占めています。

▽　欠損値の有無を確認します

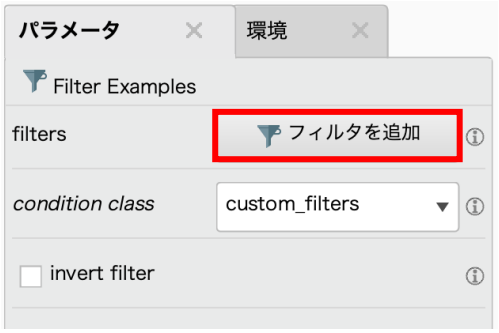
3 つの属性に欠損値がありました。次のステップで欠損値の処理を行います。
LotFrontage：敷地に接続する道路の直線距離
GarageYrBlt：ガレージが建てられた年
MasVnrArea：石積みベニヤの面積（平方フィート）

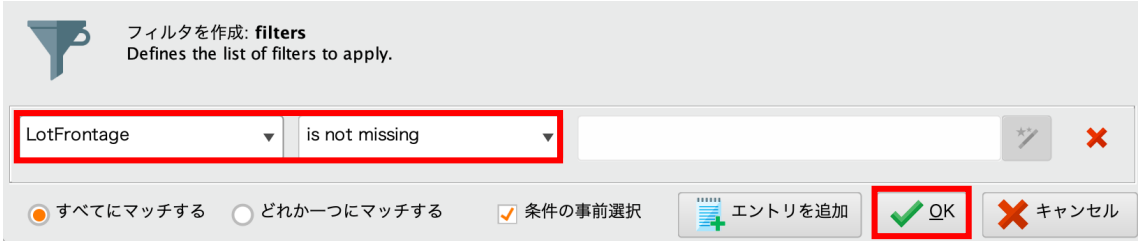
欠損値の処理



Step1.　LotFrontage 属性は、欠損値が多いので、Filter Examples オペレータを使用して欠損値を含まない行のみを出力するように設定します。

＜パラメータの設定＞

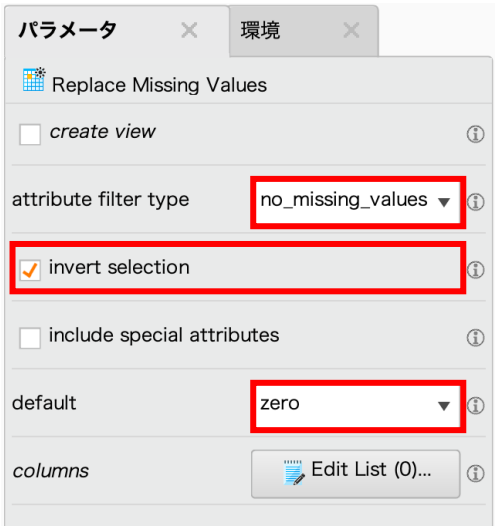




Step2.　GarageYrBlt 属性は、GarageType 属性(ガレージの場所)を確認すると None(ガレージなし)が 74 あり、ガレージ自体が存在しないため欠損していたことがわかりました。MasVnrArea 属性も、MasVnrType 属性(ベニヤタイプ)を確認すると None(ベニヤなし)が 6 あり、ベニア自体が存在しませんでした。

→Replace Missing Values オペレータを使用し、欠損値を 0 に置き換えます。

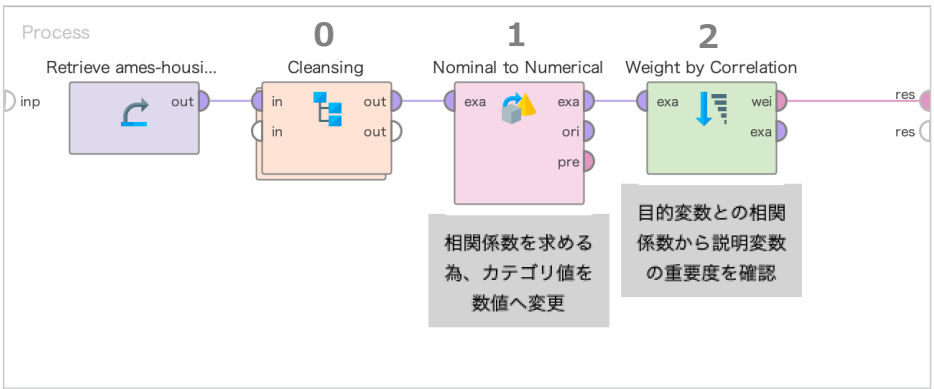
＜パラメータの設定＞



欠損している値を zero に置き換えたいので、invert selection にチェックを入れます。
設定後プロセスを実行すると、欠損値が全て 0 に変換されていることが確認できます。

目的変数との相関

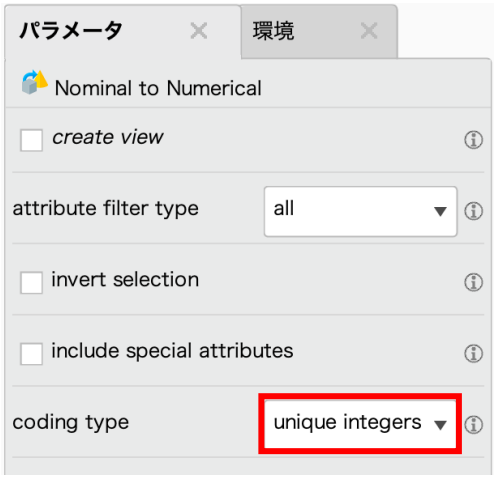
属性数が 79 と多いので、目的変数と各属性の相関係数を求め、説明変数の重要度を確認します。



Step0.　欠損値処理までのプロセスをサブプロセスにまとめました。

Step1.　相関係数を計算するには、全て数値型に変更してから行う必要があります。
Nominal to Numerical オペレータ使用し、カテゴリ型から数値型へ変更します。

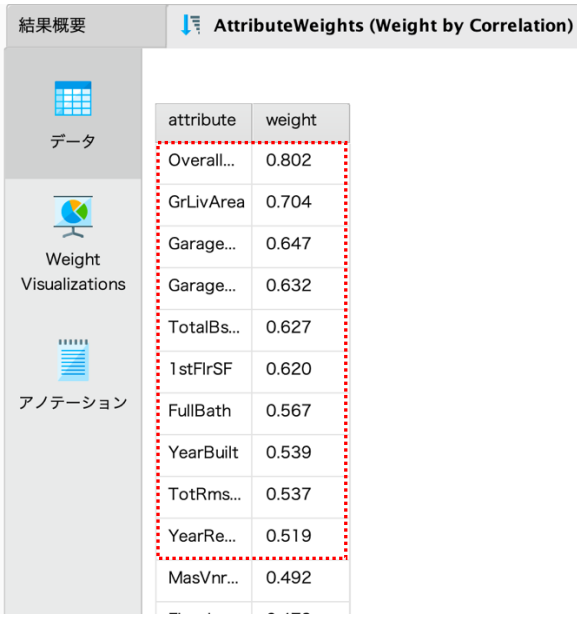
＜パラメータの設定＞



coding type は unique integers を指定することで、カテゴリ属性の値が等しくランク付けされた実数値の属性に変更されます。

Step2.　Weight by Correlation オペレータを使用し、目的変数との相関係数から説明変数の重要度を確認します。

＜パラメータの設定＞



sort direction は descending を指定し、降順にソートされるように設定します。
プロセスを実行すると、目的変数と各属性との相関係数を降順で確認することができます。

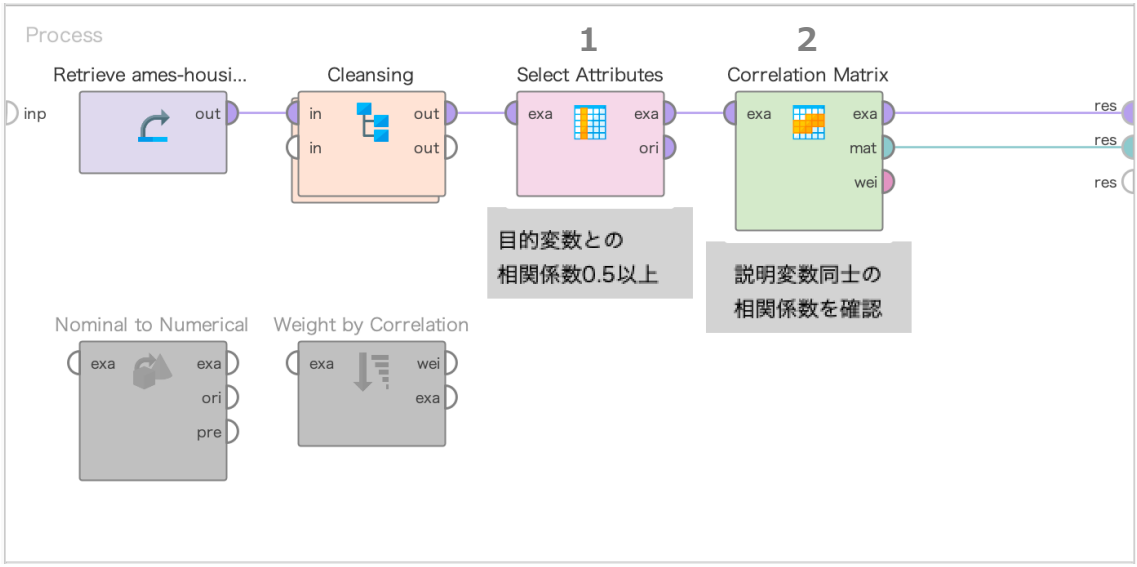
目的変数と相関が高い属性を説明変数とします。今回、相関係数が 0.5 以上であった 10 の属性を選択して分析を進めていきます。(変数の選択は次のステップで行います)

説明変数：OverallQual、GrLivArea、GarageCars、GarageArea、TotalBsmtSF
1stFlrSF、FullBath、YearBuilt、TotRmsAbvGrd、YearRemodAdd

ここで注意したいのが、「多重共線性」という問題です。多重共線性とは、説明変数同士に強い相関があるときに起こる状態のことで、分析結果の信頼性が低下してしまう可能性があります。説明変数の選択は、多重共線性を考慮して行う必要があります。

説明変数同士の相関

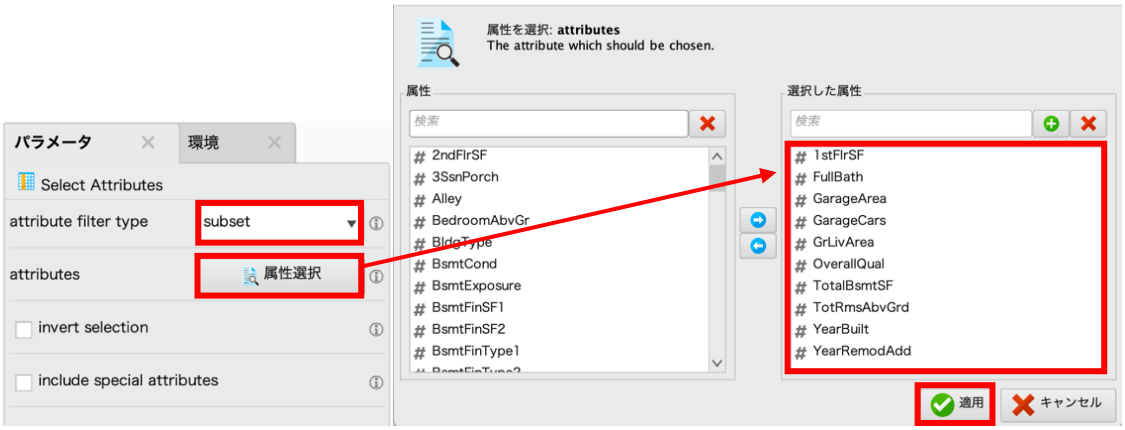
多重共線性を防ぐために、まず説明変数同士に強い相関があるものが存在するかを確認し、強い相関が存在する場合はいずれか一方を削除します。



（Nominal to Numerical と Weight by Correlation は属性の重要度を確認するために使用したので、無効化しました）

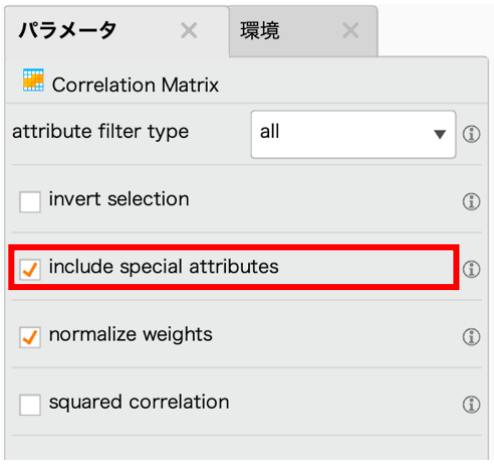
Step1.　Select Attributes オペレータを使用し、目的変数との相関係数が 0.5 以上であった 10 の属性を説明変数として選択します。

＜パラメータの設定＞



Step2.　Correlation Matrix オペレータを使用して相関行列を生成し、説明変数同士に強い相関があるものが存在するかを確認します。

＜パラメータの設定＞



include special attributes にチェックを入れて実行すると、特別属性である label 属性と id 属性を含む相関行列が以下のように生成されます。

Attributes	SalePrice	OverallQual	YearBuilt	YearRemodAdd	TotalBsmtSF	1stFlrSF	GrLivArea	FullBath	TotRmsAbvGrd	GarageCars	GarageArea	Id
OverallQual	0.802	1	0.588	0.561	0.570	0.521	0.608	0.564	0.445	0.613	0.584	-0.039
YearBuilt	0.539	0.588	1	0.597	0.421	0.330	0.208	0.483	0.111	0.543	0.498	-0.012
YearRemodAdd	0.519	0.561	0.597	1	0.315	0.283	0.295	0.456	0.196	0.426	0.389	-0.024
TotalBsmtSF	0.627	0.570	0.421	0.315	1	0.833	0.478	0.340	0.302	0.458	0.512	-0.019
1stFlrSF	0.620	0.521	0.330	0.283	0.833	1	0.573	0.390	0.418	0.464	0.511	-0.005
GrLivArea	0.704	0.608	0.208	0.295	0.478	0.573	1	0.620	0.829	0.474	0.474	-0.003
FullBath	0.567	0.564	0.483	0.456	0.340	0.390	0.620	1	0.548	0.471	0.418	0.010
TotRmsAbvGrd	0.537	0.445	0.111	0.196	0.302	0.418	0.829	0.548	1	0.378	0.350	0.020
GarageCars	0.647	0.613	0.543	0.426	0.458	0.464	0.474	0.471	0.378	1	0.890	0.003
GarageArea	0.632	0.584	0.498	0.389	0.512	0.511	0.474	0.418	0.350	0.890	1	-0.012
SalePrice	1	0.802	0.539	0.519	0.627	0.620	0.704	0.567	0.537	0.647	0.632	-0.037
Id	-0.037	-0.039	-0.012	-0.024	-0.019	-0.005	-0.003	0.010	0.020	0.003	-0.012	1

今回、説明変数同士の相関が 0.8 以上の組み合わせをピックアップし、目的変数との相関係数が低い方を削除します。SalePrice(目的変数)列は、各説明変数と目的変数との相関係数を表すため、ドラッグ&ドロップで Attributes 列のすぐ右隣に移動させました。

説明変数同士の相関が 0.8 以上であった組み合わせは 3 つありました。

・GarageCars と GarageArea …… 0.890
・TotalBsmtSF と 1stFlrSF …… 0.833
・GrLivArea と TotRmsAbvGrd … 0.829

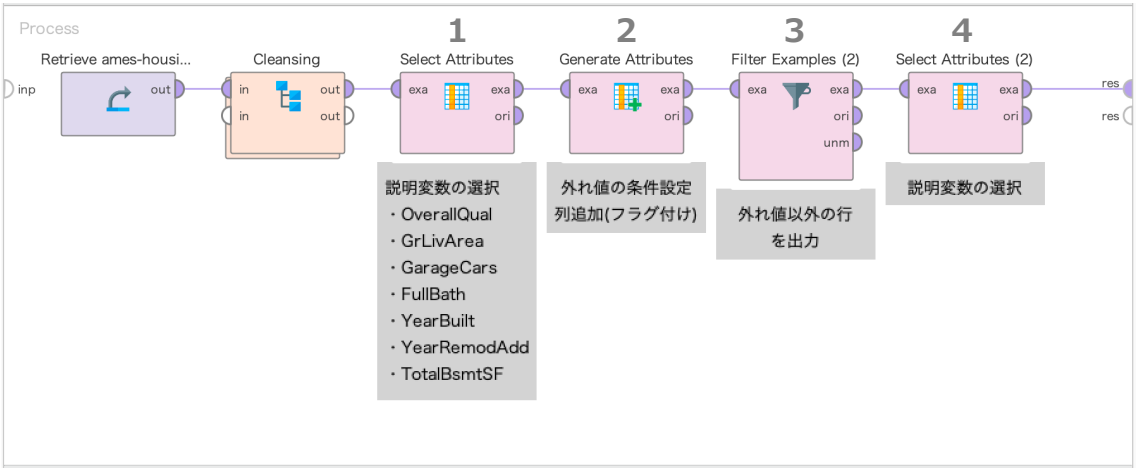
目的変数との相関係数を確認した結果、GarageArea、1stFlrSF、TotRmsAbvGrd を削除します。

多重共線性を考慮し、以下の 7 つを説明変数とします。

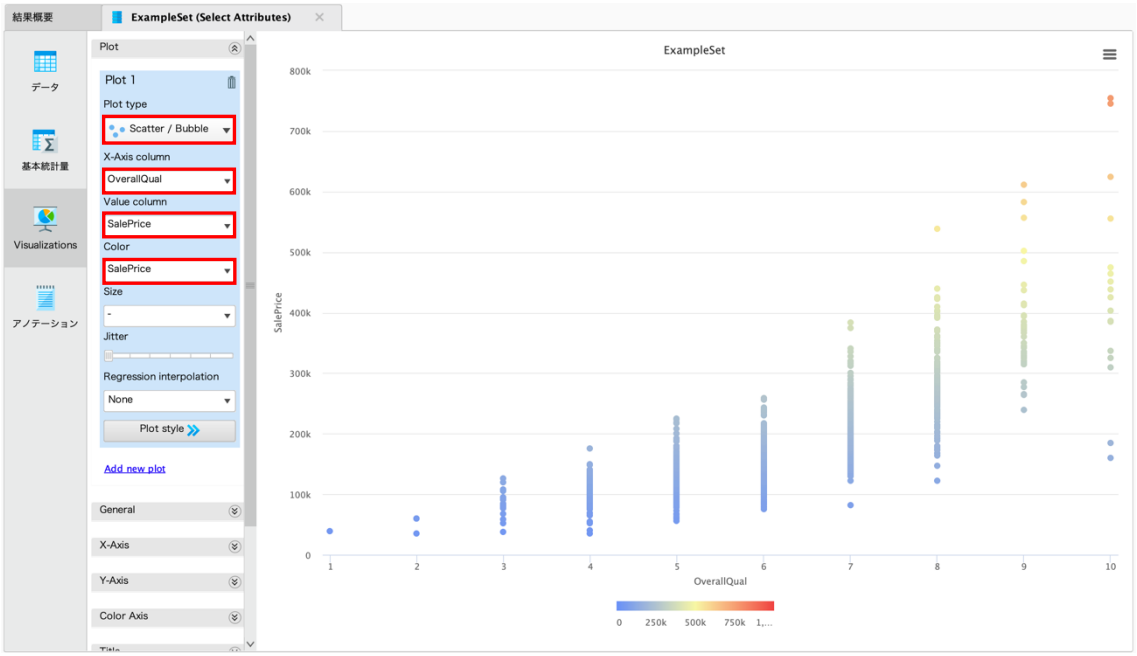
説明変数：OverallQual、GrLivArea、GarageCars、TotalBsmtSF、FullBath
YearBuilt、YearRemodAdd（変数の選択は次のステップで行います。）

外れ値の処理

モデルの作成をする前に、データの分布をみて他の値から大きく外れた値（外れ値）の有無を確認し、外れ値があった場合は取り除きます。

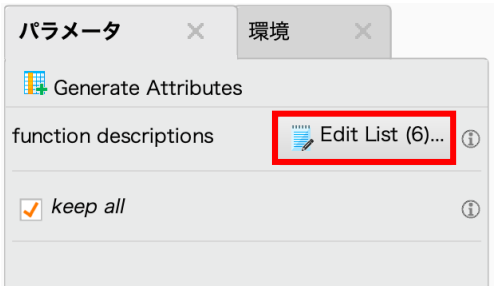


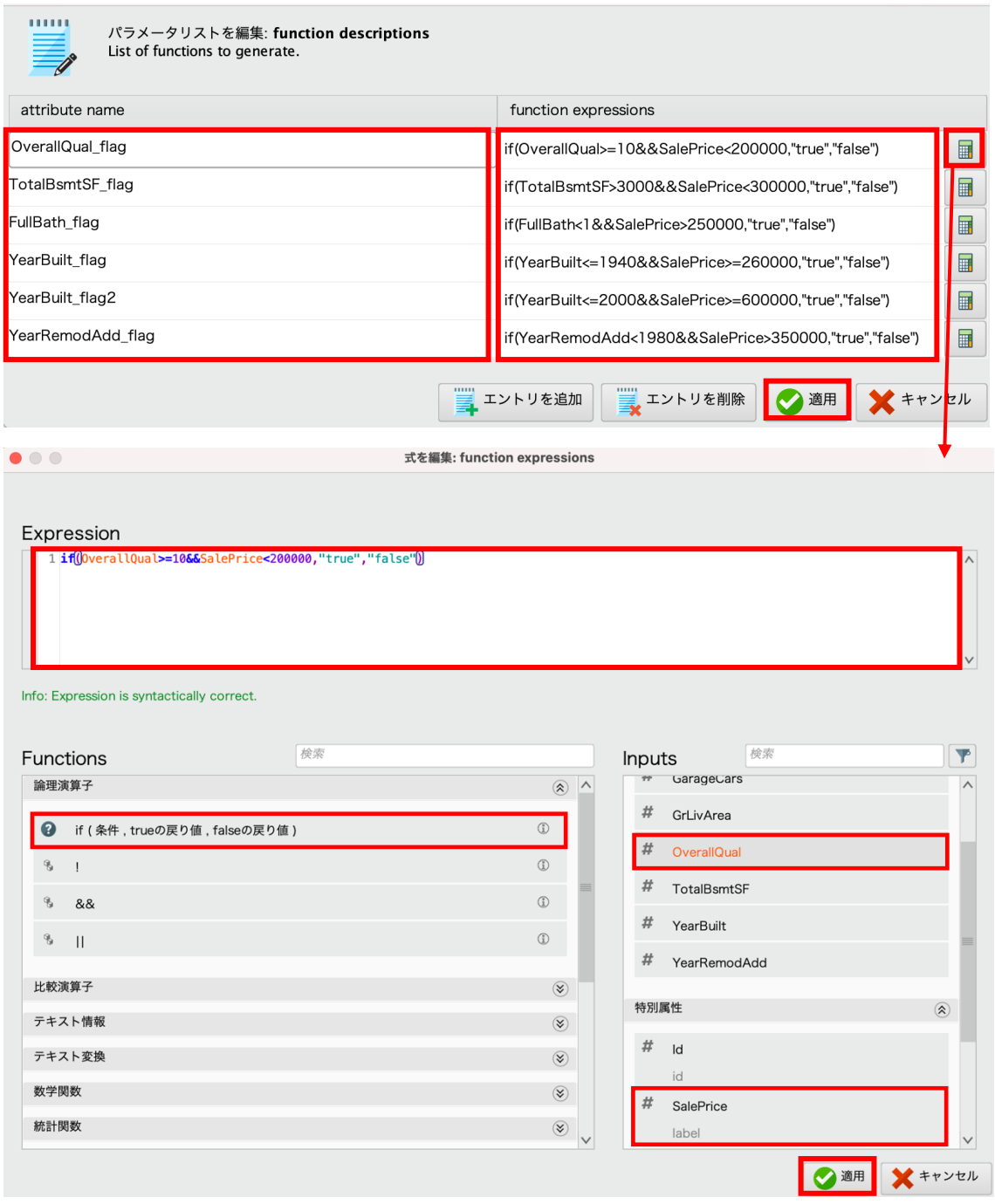
Step1.　Select Attributes オペレータを使用し、7 つの説明変数を選択します。（パラメータ設定は p.6 参照）各説明変数の散布図をみて外れ値の有無を確認します。
“X-Axis column”を変更すると、各説明変数のデータ分布が確認できます。



外れ値の有無を確認した結果、以下を外れ値として処理することにします。
OverallQual　……評価が 10 以上で、住宅価格が 200,000 ドルより下は削除
TotalBsmtSF ……地下室総面積が 3000 より上で、住宅価格 300,000 ドルより下は削除
FullBath ……評価が 1 より下で、住宅価格が 250,000 ドルより上は削除
YearBuilt……1940 年以前に建てられた物件で、住宅価格が 260,000 ドル以上は削除
　　2000 年以前に建てられた物件で、住宅価格が 60,0000 ドル以上は削除
YearRemodAdd…1980 年より前のリフォーム物件で、住宅価格が 35,0000 より下は削除

Step2.　Generate Attributes オペレータを使用して、Step1 で決めた外れ値の条件を指定し、
　　外れ値のフラグ付けした列を作成します。
＜パラメータの設定＞





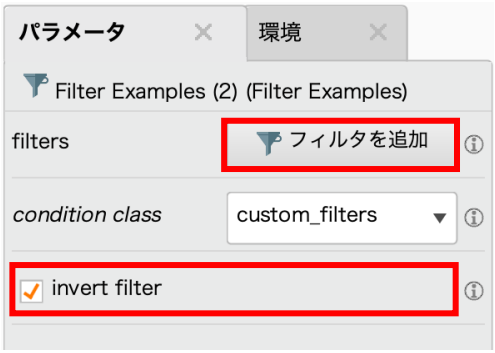
“属性名＋_flag”という新しい属性(列)を作成し、数式には外れ値とする条件を入れます。数式は直接入力または電卓マークから設定できます。適用しプロセスを実行すると、データセットには、新しい属性(列)が以下のように追加されます。

geCars	OverallQual_flag	TotalBsmtSF_flag	FullBath_flag	YearBuilt_flag	YearBuilt_flag2	YearRemodAdd_flag
	false	false	false	false	false	false
	false	false	false	true	false	false
	false	false	false	false	false	false

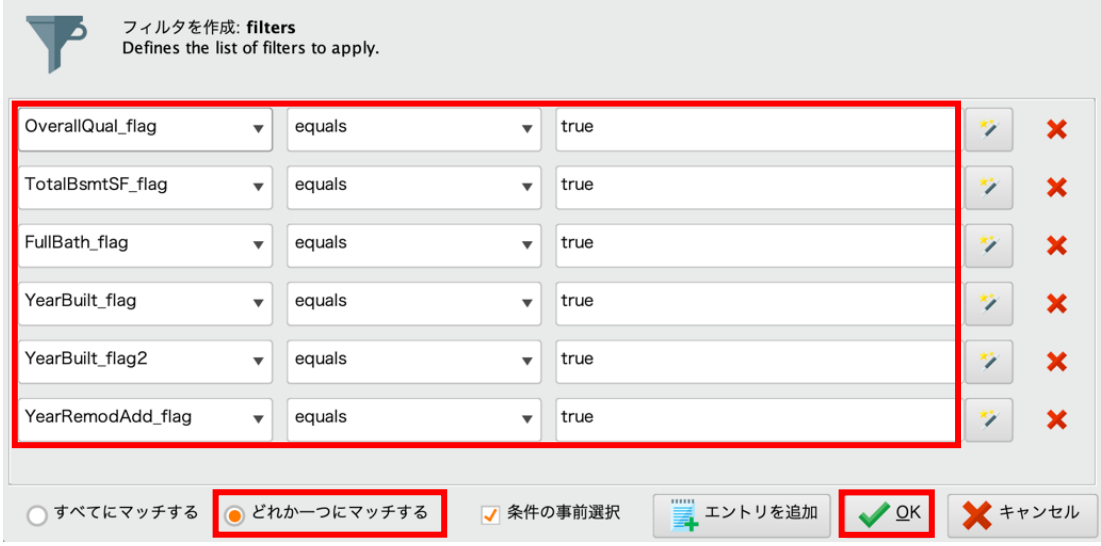
外れ値のある行には”true”、それ以外の行は”false”と表示されます。

Step3.　Filter Examples オペレータを使用し、外れ値以外の列を出力します。

＜パラメータの設定＞



invert filter をクリックすると、フィルタを追加 から設定した条件が反転されます。



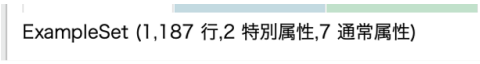
外れ値のフラグ付けした属性(列)に、外れ値”true”が一つでもあればその行は出力しないように設定します。

Step4.　Select Attributes オペレータを使用して、説明変数を再選択します。

Step2 で外れ値のフラグ付けした属性(列)を作成して属性が増えたので、Step1 と同様に説明変数を 7 つ選択します。

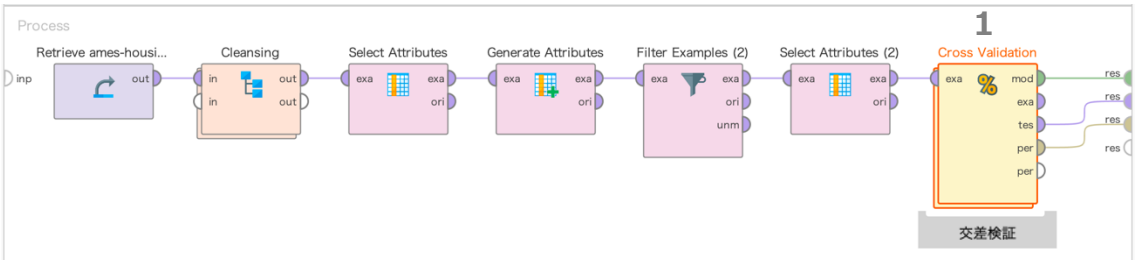
説明変数：OverallQual、GrLivArea、GarageCars、TotalBsmtSF、FullBath
YearBuilt、YearRemodAdd

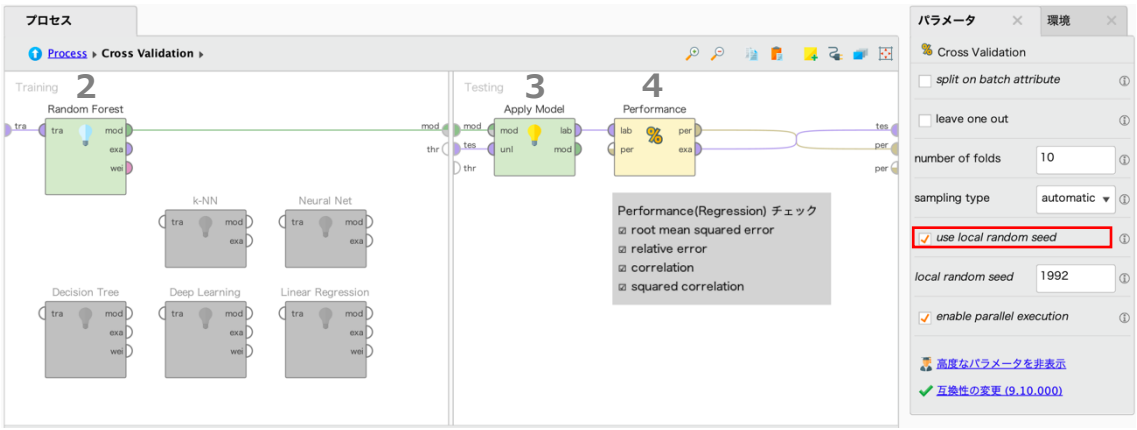
ここまでのプロセスを実行すると、外れ値の存在する行を削除したことによって、データが 1,201 行から 1,187 行になります。



モデル作成・適用

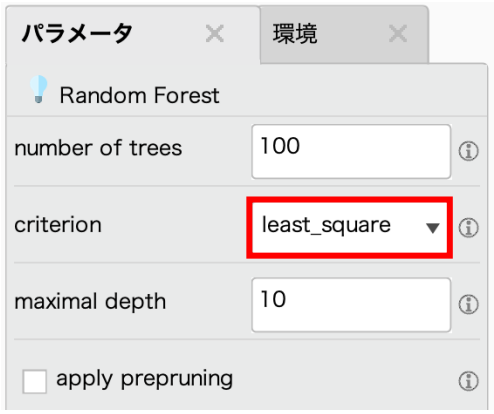
データの準備ができたので、いくつかの機械学習アルゴリズムを試して交差検証を行い、今回扱うデータと相性が良いのはどれかを確認します。





Step1.　Cross Validation オペレータを配置し、交差検証を行います。
（実行結果を揃えるため、use local random seed にチェックを入れます）

Step2.　住宅価格の予測なので、回帰の機械学習アルゴリズムを選択しモデルを作成します。
＜パラメータの設定　例＞

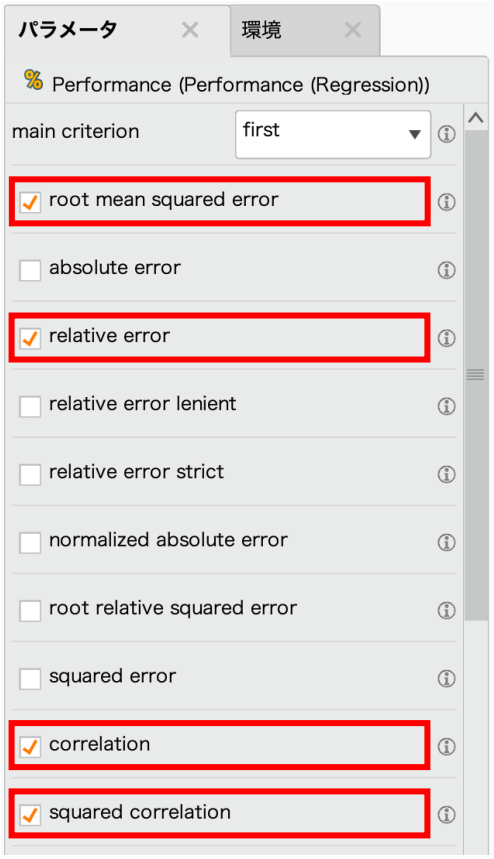


回帰を用いるので、criterion は least_square を指定します。（Decision Tree の場合も同様）
今回、パラメータはデフォルト値のままにします。

Step3.　Apply model オペレータで、交差検証用に作成したモデルにテスト用データを適用します。

Step4.　Performance(Regression)オペレータを使用し、検証用に作成したモデルが未知データに対してどのくらいの精度があるかを、4 つの評価指標で確認します。6 つのアルゴリズムを試し、結果を比較します。

＜パラメータの設定＞



実行結果を下記表にまとめました。(パラメータ：least_square 指定以外はデフォルト設定)

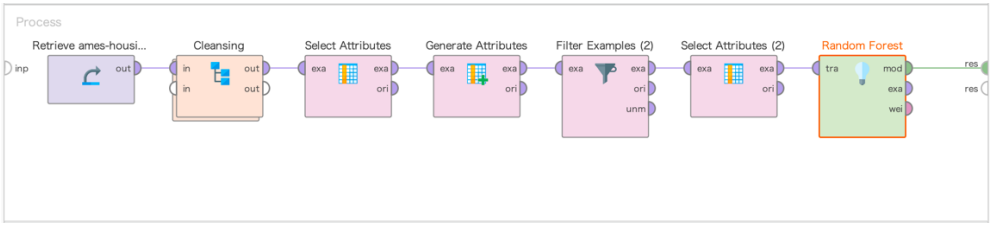
機械学習アルゴリズム	root mean squared error	relative error	correlation	squared correlation
Random Forest（回帰）	27055.052 +/- 2395.754	11.50% +/- 1.34%	0.937 +/- 0.018	0.878 +/- 0.034
Deep Learning	28153.327 +/- 2262.050	12.64% +/- 1.26%	0.936 +/- 0.016	0.876 +/- 0.029
Neural Net	28623.182 +/- 3020.979	13.25% +/- 2.73%	0.938 +/- 0.016	0.880 +/- 0.030
Liner Regression	33036.439 +/- 4293.221	14.82% +/- 1.15%	0.906 +/- 0.015	0.821 +/- 0.027
Decision Tree（回帰）	33461.141 +/- 2613.566	14.18% +/- 1.45%	0.905 +/- 0.020	0.819 +/- 0.036
k-NN	39293.207 +/- 5267.998	15.82% +/- 1.70%	0.866 +/- 0.024	0.751 +/- 0.043

※バージョンや環境などにより、実行結果に多少誤差がある場合があります

この中では、ランダムフォレストがデータと最も相性が良いことが分かりました。

▽　モデル作成

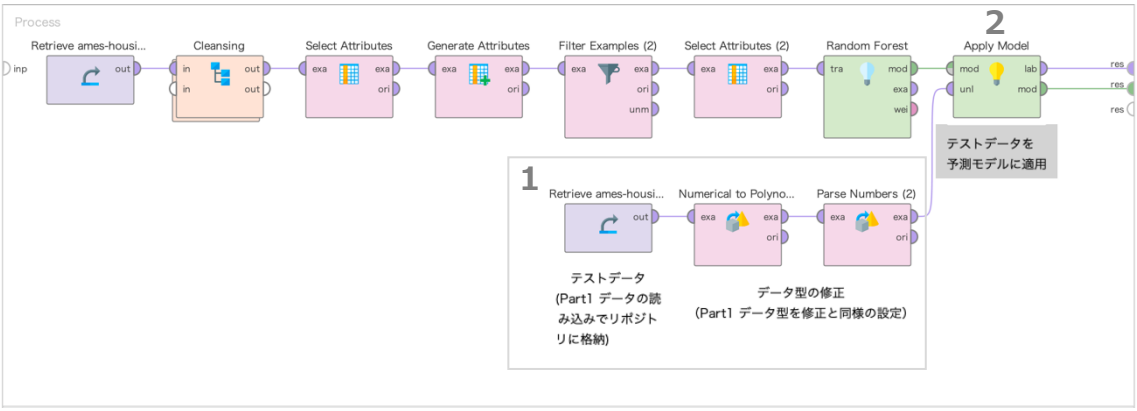
交差検証の結果から、ランダムフォレストを用いてモデルを作成します。



交差検証の Trainig フェーズに配置したものと同じオペレータ（Random Forest）をプロセス画面に配置します。これで、予測モデルを作成できました。

▽　モデル適用

生成した予測モデルを使って、実際に各住宅の販売価格を予測します。



Step1.　予測したいデータ（テストデータ）をモデルに適用させるための準備をします。Part1 で ID 属性を定義しリポジトリに格納しておいたテストデータを配置し、訓練データと同様の方法で正しいデータ型に修正します。

Step2.　Apply Model オペレータを使用し、テストデータを予測モデルに適用させます。

プロセスを実行すると、以下のように prediction(SalePrice)の列が追加され、テストデータに対する住宅価格の予測値が出力されます。

結果概要　Random Forest Model (Random Forest)　ExampleSet (Apply Model)

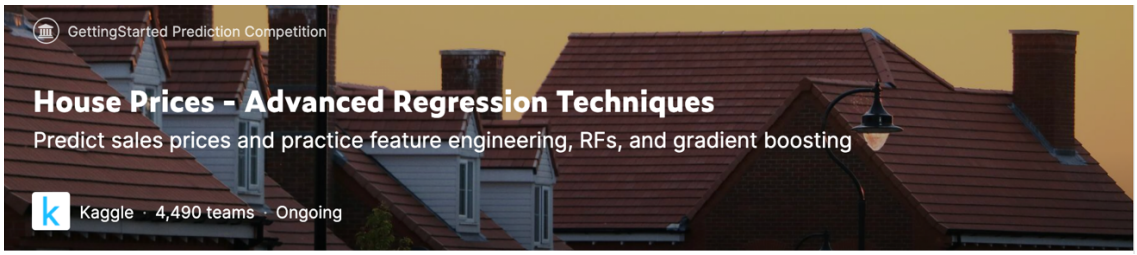
開く　Turbo Prep　Auto Model　フィルタ (1,459 / 1,459 行): all

データ　基本統計量

Row No.	Id	prediction(SalePrice)	LotFrontage	GarageYrBlt	MSSubClass	MSZoning	LotArea	Street	Alley
1	1461	119679.437	80	1961	20	RH	11622	Pave	NA
2	1462	149361.923	81	1958	20	RL	14267	Pave	NA
3	1463	177780.634	74	1997	60	RL	13830	Pave	NA
4	1464	182618.892	78	1998	60	RL	9978	Pave	NA
5	1465	210578.093	43	1992	120	RL	5005	Pave	NA

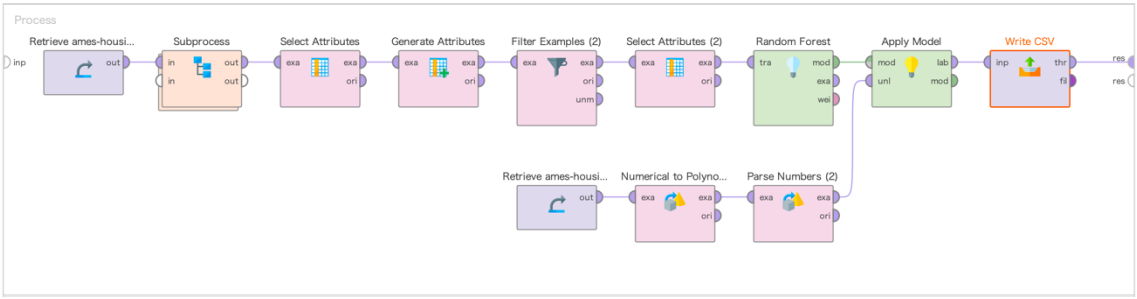
この予測結果を Kaggle に提出し、コンペティションに実際に参加してみます。

Kaggle のコンペティションへ参加

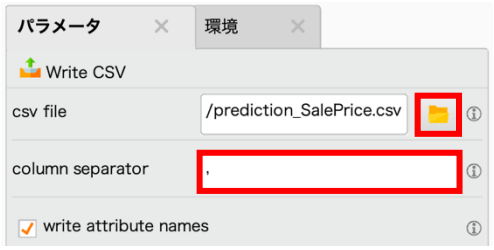


Kaggle は、世界中のデータサイエンティストたちが集まり、構築したモデルの精度を競うコンペティションです。予測結果を CSV ファイルに書き出して提出すると、順位が表示されます。(Kaggle に無料アカウント登録を行うと参加できます)

▽ Write CSV オペレータを使用し、予測結果を CSV ファイルに書き出します



<パラメータの設定>

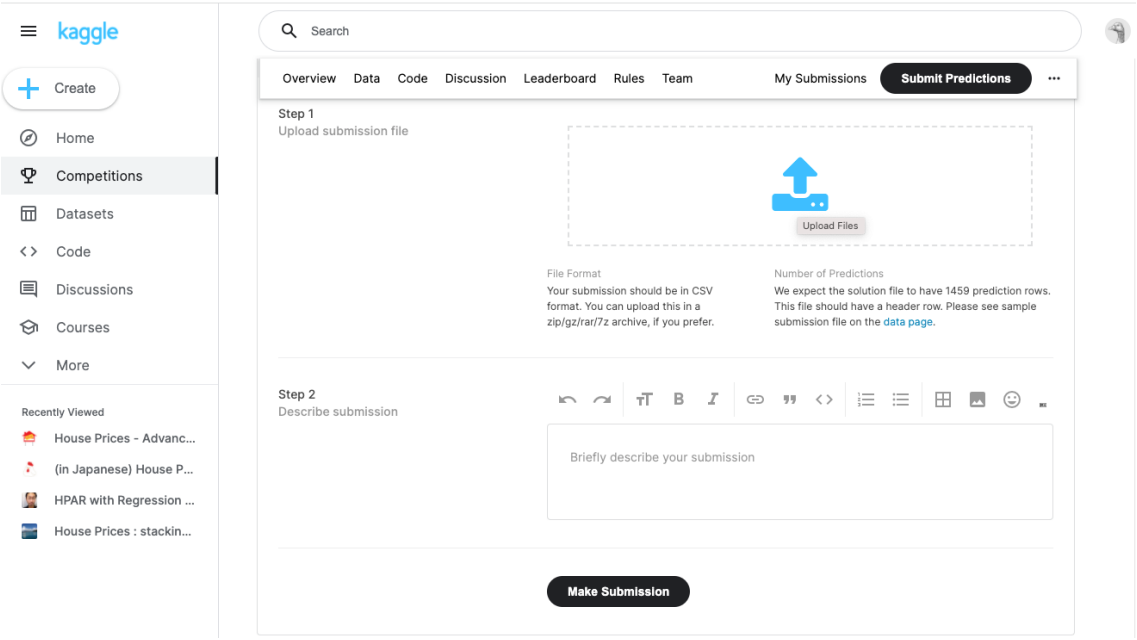


フォルダのアイコンから保存場所を指定し、column separator はカンマ「,」に変更します。
プロセスを実行後、パラメータで指定した保存場所から CSV ファイルを開きます。
id と prediction(SalePreice)以外の列を削除し、prediction(SalePreice)の列名は SalePrice に変更します。

	A	B	C	D
1	Id	SalePrice		
2	1461	119679		
3	1462	149361		
4	1463	177780		

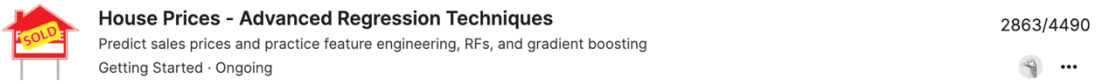
これで、コンペティションに参加する準備が完了しました。

▽ **予測結果 CSV ファイルを提出します**



Upload submission file から CSV ファイルをアップロードし、Make Submission をクリックすると提出完了です。順位表やスコアは、Leaderboard から確認できます。または、右上のアイコン(Your Profile>Competitions)からも順位が確認できます。

結果は、スコア 0.16167 で 4490 チーム中 2863 位の順位でした。



終わりに

欠損値や外れ値など確認しながらデータ分析を行なっていきました。機械学習プロジェクトにおいて約 8 割がデータ前処理（データ準備）に費やされていると言われており、各変数の特徴や変数同士の関係性などを把握する上でも重要なプロセスになります。

今回初めて Kaggle のコンペティションに参加しました。データ分析の勉強にもなり、一からプロセスを考えていく楽しさを感じながら進めていくことができました。皆さんも参加されてみてはいかがでしょうか。

次回は、今回作成した予測モデルの精度を上げていきたいと思います。