SFI
VERTYS Inc.

REGRESSION ALGORITHM GUIDE

SFI
回帰分析アルゴリズム
解説資料

～ 現場で使える19のモデル、その本質と選び方 ～
製造現場の生産技術・製造技術・製品開発担当者向け

19

ALGORITHMS

PUBLISHED BY

Vertys Inc. | SFI Product Team
Manufacturing Data Analysis Platform

PieceMaker

01 INTRODUCTION

本資料の目的

アルゴリズム名を「覚える」ための資料ではありません。
現場の課題に、どのアルゴリズムを当てるかが判断できるようになるための資料です。

製造現場では、品質予測・歩留まり改善・予知保全など、数値を予測したい場面が無数にあります。
SFIには19種類の回帰アルゴリズムが搭載されていますが、それぞれに得意不得意があり、
「とりあえず最新のAIを使えばよい」という考え方は、本質的ではありません。

本資料は、統計や機械学習の専門家ではない、**生産技術・製造技術・製品開発の担当者**の皆様が、
ご自身の現場課題に対して「どのアルゴリズムが筋がよいか」を素早く見極められるよう、
たとえ話と図解を中心に整理したものです。

本資料が目指す3つの状態

01

課題からモデルを逆引きできる

「外れ値が多い」「カテゴリ変数が多い」など、現場の状況から候補アルゴリズムを2〜3個に絞れる。

02

長所と短所を言葉で説明できる

上司や品質会議で「なぜこのモデルを選んだか」を、数式なしで筋道立てて説明できる。

03

バズワードに流されない

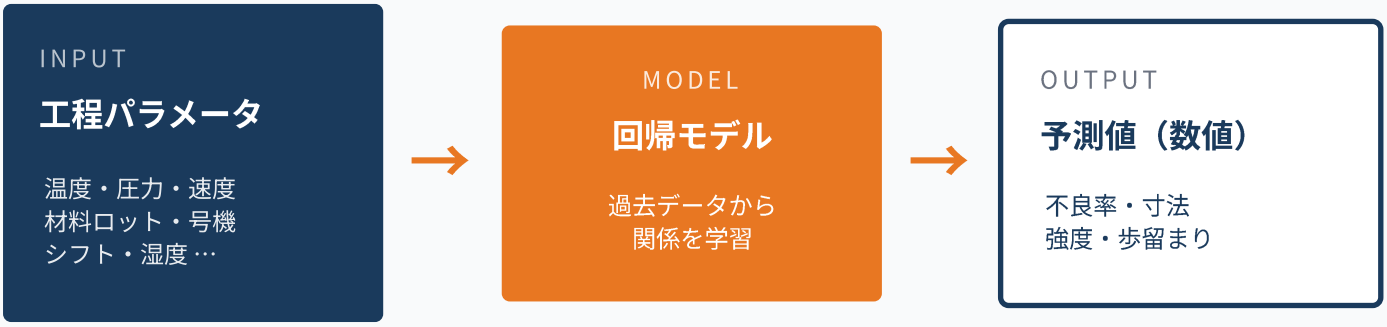
「最新AI＝最良」ではない。
線形回帰で十分なケースを見抜き、本当に効く打ち手を選べる。

02 WHAT IS REGRESSION

回帰分析とは何か

測定値や過去データから、まだ測っていない値を予測する仕組み

CONCEPT



分類と回帰の違い

分類 Classification

カテゴリ（区分）を当てる
→ 良品 / 不良品
→ A級 / B級 / C級
→ 異常あり / 異常なし

例：外観検査画像から「不良品」を判定する

回帰 Regression（本資料の対象）

連続した数値を当てる
→ 寸法 12.34 mm
→ 不良率 3.2 %
→ 引張強度 480 MPa

例：温度・圧力から成形品の収縮量を予測する

製造現場における回帰の代表例

- 不良率の予測
- 製品寸法の予測
- 歩留まりの予測
- サイクルタイム予測
- 設備寿命の予測
- 引張強度の予測
- エネルギー消費量予測
- 工程能力 Cpk 予測
- コスト予測

03 HOW TO READ

本資料の読み方

まず早見表、次に詳細。逆引きで使うのがおすすめです。

RECOMMENDED FLOW — 推奨の読み方

STEP 01
早見表で当たりをつける

→

STEP 02
候補2〜3個の詳細ページを読む

→

STEP 03
SFIで全部試して比較

本資料の構成

ページ	セクション	内容
P.02-04	**はじめに**	本資料の目的と回帰分析の基礎
P.05-06	**全体マップ**	19アルゴリズムを3グループに分けて俯瞰
P.07-25	**アルゴリズム詳細**	1アルゴリズム1ページ、現場ユースケース付き
P.26-27	**比較表**	7観点 ×19アルゴリズムの◯△×評価
P.28-30	**早見表（逆引き）**	現場のシチュエーションから推奨アルゴリズム
P.31-32	**SFI活用の流れ**	データ準備から現場展開までの実務フロー
P.33-34	**よくある誤解と注意点**	「精度＝良いモデル」ではない、など
P.35	**まとめ**	アルゴリズムは道具、選び方は現場知見が決める

アルゴリズム詳細ページの色分け凡例

線形系
— 説明性が高くベースラインに適する

木/アンサンブル系
— 精度と汎用性が高い

近傍/カーネル系
— 非線形・少データに対応

04 ALGORITHM MAP

19アルゴリズム全体マップ

3つのグループに分けて俯瞰してみましょう

SFI
回帰モデル
19種

GROUP 01
線形系
Linear Family / 9種

入力と出力の関係を「直線（や直線の組合せ）」で捉える。
説明性が高くベースラインに最適。

- ◆ 線形回帰
- ◆ リッジ回帰
- ◆ ラッソ回帰
- ◆ エラスティックネット
- ◆ ラッソLARS
- ◆ 最小角回帰(LARS)
- ◆ ベイズリッジ
- ◆ 直交マッチング追跡(OMP)
- ◆ Huber回帰

GROUP 02
木・アンサンブル系
Tree & Ensemble / 8種

「もし○○なら××」という分岐ルールを大量に積み上げる。
非線形・カテゴリ変数に強く、精度も高い。

- ◆ 決定木
- ◆ ランダムフォレスト
- ◆ エクストラツリー
- ◆ 勾配ブースティング
- ◆ AdaBoost
- ◆ XGBoost
- ◆ LightGBM
- ◆ CatBoost

GROUP 03
近傍 / カーネル系
Neighbor & Kernel / 2種

「過去の似た事例」を参照したり、空間を曲げて非線形関係を扱う。
少〜中規模データ向け。

- ◆ k近傍法 (k-NN)
- ◆ カーネルリッジ

数は少ないですが、
線形でも木でも扱いにくい問題に独自の強みを発揮します。

04 GROUP CHARACTERISTICS

3グループの特徴をひと言で

それぞれの「向き」と「不向き」を整理します

GROUP 01

線形系

Linear Family

「Y = a × 温度 + b × 圧力 + c」のように、入力と出力の関係を直線で表現するモデル群。

▸ 得意なこと

説明性、ベースライン作り、
相関の強い特徴量の制御、
外れ値対応(Huber)、変数選択(Lasso系)

▸ 苦手なこと

複雑な非線形関係
（曲線・しきい値挙動）、
特徴量間の相互作用が強い問題

GROUP 02

木/アンサンブル系

Tree & Ensemble

「温度が180以上なら… 圧力が80以下なら…」という分岐ルールを大量に積み上げて予測する。

▸ 得意なこと

非線形・しきい値挙動、カテゴリ変数、
特徴量重要度の可視化、
大規模データ(LightGBM/XGBoost)

▸ 苦手なこと

学習データの範囲外への外挿、
極端に少ないデータ、
純粋な線形関係の効率的表現

GROUP 03

近傍 / カーネル系

Neighbor & Kernel

「似た過去事例の平均をとる」「空間を曲げて非線形を扱う」など、直感的・数学的なアプローチ。

▸ 得意なこと

非線形関係、少データでも動く(k-NN)、
滑らかな関係のモデル化(カーネル)、
ベテラン的判断の模倣

▸ 苦手なこと

大規模データ（計算コスト爆発）、
特徴量重要度の解釈、
高次元（特徴量が数十個以上）

POINT どのグループが「絶対に良い」ということはありません。
データの性質と、現場で求められる説明性のレベルで決まります。

GROUP 02 ・ 木 / アンサンブル系

01

CatBoost回帰

CatBoost Regression

ひと言で言うと

カテゴリ変数（材料種別・号機・ロット）の扱いがそのままで上手い、勾配ブースティングの進化版。

仕組みのイメージ

たとえ話：
新人がベテランの後ろで作業を見て、
ベテランが間違えた部分だけを次の新人が修正していく。
これを何百回も繰り返して全体の判断精度を上げていきます。

CatBoostの特徴：
「号機A」「材料メーカーX」のような
文字情報（カテゴリ変数）を数値に変換する前処理を内部で自動的にやってくれるため、
現場担当者の手間が劇的に減ります。

現場ユースケース

■ 射出成形の不良率予測

「材料メーカー」「号機番号」「金型ID」「シフト」など、
カテゴリ情報が10種類以上ある工程で、
ロット毎の不良率を予測。

■ 多品種少量生産の歩留まり予測

品番・取引先・装置といったカテゴリ情報が
大量にある加工現場で、品番ごとの歩留まりを予測。

▸ 長所

- カテゴリ変数を**前処理なしで扱える**（One-Hot不要）
- 過学習しにくい設計（順序付きブースティング）
- 少ない調整でそこそこの精度が出やすい
- 欠損値もそのまま投入できる

▸ 短所・注意点

- 学習に時間がかかる（LightGBMより遅め）
- カテゴリ変数が少ないと、他の木モデルと差が出にくい
- ハイパーパラメータが多く、本格チューニングは煩雑
- 「なぜそう予測したか」は要SHAP等の追加解析

こんな時に選ぶ

◆工程データに**カテゴリ変数（号機/材料/シフト/品番）が5種類以上**あるとき
◆ロット間ばらつきの要因に「どの号機か」「どのメーカー材料か」が効いていそうなとき
◆前処理に時間をかけたくない、まず「カテゴリそのまま」で精度を見たいとき

▸ **現場視点での注意** カテゴリ変数の質（記入ゆれ、表記ゆれ）が悪いと、CatBoostでも精度は出ません。
マスタ整備が大前提です。

GROUP 02 ・ 木 / アンサンブル系

02

勾配ブースティング回帰

Gradient Boosting Regression (GBR)

ひと言で言うと

誤差を一段ずつ補正する小さな木を積み上げる、汎用性が高くまず試したい王道モデル。

仕組みのイメージ

たとえ話：
最初の検査員が「この製品は不良率3%」と言う。
次の検査員は「最初の人は0.5%甘く見ている」と修正する。
3人目は「2人目は0.1%厳しすぎる」
とさらに修正…これを100〜1000回繰り返します。

ポイント：
一人ひとり（弱学習器）は浅い決定木ですが、
「前の人の間違いを修正する」形で積み重ねることで、
全体として非常に高精度になります。

現場ユースケース

■ 溶接強度の予測

電流・電圧・速度・板厚といった連続値パラメータから、
溶接強度を非線形に予測。
線形回帰では捉えきれない相互作用を学習。

■ 加工品の寸法予測

工具摩耗・送り速度・主軸回転数から仕上がり寸法を予測。
「ある閾値を超えると急に精度が落ちる」
という非線形挙動も捉えられます。

▸ 長所

- 非線形・相互作用を自動で捉える
- 多くの製造データで安定して高精度
- 特徴量重要度が出せる（要因の見える化）
- 外れ値の影響を比較的受けにくい

▸ 短所・注意点

- 学習時間が長め（XGBoost/LightGBMより遅い）
- 木の本数や深さの調整で精度が変わる
- 外挿（学習範囲外）には弱い
- カテゴリ変数は事前にエンコード必要

こんな時に選ぶ

◆ **非線形な関係**が予想され、線形回帰では精度が頭打ちのとき
◆ データ数が数百〜数万行で、計算時間に多少余裕があるとき
◆ XGBoost/LightGBMを使う前の**「標準的な比較対象」**として

▸ **立ち位置** 「勾配ブースティング」は概念。XGBoost / LightGBM / CatBoost はその高速・高機能版です。
まずはこれで様子を見るのが王道。

GROUP 02 ・ 木 / アンサンブル系

03

エクストラツリー回帰

Extra Trees Regression (Extremely Randomized Trees)

ひと言で言うと

ランダム性をあえて強くした森。学習が速く、過学習しにくい安定モデル。

仕組みのイメージ

たとえ話：
50人の検査員にそれぞれ違う条件・違う観点で判定させ、その平均をとる。
ランダムフォレストよりさらに「観点をランダムに選ぶ」ようにしたのがエクストラツリーです。

ポイント：
各木の分岐点を最適探索せず**ランダムに決める**ため計算が速く、それでも多数決の平均で精度が出ます。

現場ユースケース

■ 組立工程のサイクルタイム予測

作業者・部品種別・治具状態など多数の特徴量がある中で、ばらつきの大きいデータでもブレない予測を行う。

■ 加工条件と表面粗さの関係予測

ノイズの多い表面測定値に対して、過学習せず安定した予測モデルを構築。

▸ 長所

- 学習が速い（ランダムフォレストより高速）
- 過学習に強い（ランダム性が高いため）
- 外れ値の影響を受けにくい
- 並列処理しやすく実装が安定

▸ 短所・注意点

- ブースティング系よりピーク精度はやや劣る
- 木の本数が少ないと精度が安定しない
- 結果のブレ（実行毎の揺らぎ）に注意
- 解釈はランダムフォレストと同様、要追加解析

こんな時に選ぶ

◆ **ノイズが多く、過学習が心配**なデータ（センサ計測値が多い工程など）
◆ ランダムフォレストで学習が遅すぎるとき、高速化したい場合
◆ アンサンブル系の「2番目の選択肢」として比較に使うとき

▸ **現場視点** ランダムフォレストと「兄弟関係」。両方試して、精度と速度のバランスが良い方を採用するのが実務的です。

GROUP 02 ・ 木 / アンサンブル系

04

ランダムフォレスト回帰

Random Forest Regression

ひと言で言うと

アンサンブル学習の王道。
安定した精度と、「どの工程パラメータが効くか」が見える定番モデル。

仕組みのイメージ

たとえ話：
100人の検査員にそれぞれ違うサンプル（バラついた学習データ）を見せて判定させ、その平均を取る。
一人の偏りは消え、全体として安定した判定になります。

ポイント：
多数の決定木をデータ・特徴量ともランダムサンプリングで作り、その平均で予測。
シンプルだが頑健で、**「まずこれで様子見」が定石。**

現場ユースケース

■ 歩留まり要因の特定

特徴量重要度を見ることで「温度の影響が最大、次にロット差、その次に湿度」という**要因ランキング**を現場に提示できる。

■ 設備別の不良率予測

号機ごとのデータでも安定して動き、装置間比較の基礎モデルとして使える。

▸ 長所

- 特徴量重要度で**「効く要因」が見える**
- 調整が少なくても安定した精度
- 外れ値・欠損にそこそこ強い
- 並列計算で実用速度が出る

▸ 短所・注意点

- ピーク精度はブースティング系に劣る
- 個々の予測ロジックは見えない（ブラックボックス）
- 外挿は不可（学習範囲外の予測は苦手）
- 木の本数が多いとメモリ消費が大きい

こんな時に選ぶ

◆ **「何が効くか」を可視化したい**とき（要因解析が主目的）
◆ チューニングに時間をかけず、まず動くモデルが欲しいとき
◆ 安定運用が必要で、ブースティング系の挙動が読みにくいと感じたとき

▸ **現場での価値** 「予測精度」だけでなく「**要因ランキング**」が出るのが実務上の最大の武器。改善活動の起点になります。

GROUP 01 ・ 線形系

05

ベイズリッジ回帰

Bayesian Ridge Regression

ひと言で言うと

予測値だけでなく「予測の不確かさ（信頼区間）」まで出せる、少データに強い線形回帰。

仕組みのイメージ

たとえ話：
ベテランが「この条件なら寸法は12.3mmです。**ただし±0.05mmの幅で見ておいてください**」と答えるイメージ。確信度も一緒に提示します。

ポイント：
リッジ回帰に確率論（ベイズ統計）を組み合わせ、「データが少ない部分は予測幅を広めに、データが多い部分は狭く」と自動調整します。

現場ユースケース

■ 試作段階での寸法予測

実験データが30件しかない試作段階で、「予測値 ± 信頼区間」を提示。設計判断のリスク評価に使える。

■ 高価な試験の代替予測

破壊試験や長時間試験など、データが集めにくい指標を、少ないサンプルから信頼区間付きで予測。

▸ 長所

- **予測の不確かさ（信頼区間）**が得られる
- 少データでも安定動作（過学習しにくい）
- 正則化の強さを自動推定（手動調整が少ない）
- 解釈性が高い（線形係数で説明できる）

▸ 短所・注意点

- 非線形な関係には弱い（線形モデルの限界）
- 大規模データではメリットが薄れる
- 木系より絶対的な予測精度は低めになりがち
- 信頼区間の解釈には統計の素養が必要

こんな時に選ぶ

◆ データ数が**100件以下と少なく、試作段階・量産前評価**のとき
◆ 「予測値はこれくらい、ただし幅はこれくらい」とリスク込みで報告したいとき
◆ 設計判断・投資判断など、予測の確信度が意思決定に関わる場面

▸ **本質的価値** 「点予測」より「**分布で示す予測**」のほうが、現場の意思決定には誠実です。少データ時代の必修モデル。

GROUP 02 ・ 木 / アンサンブル系

06

LightGBM回帰

Light Gradient Boosting Machine

ひと言で言うと

数百万行のIoTセンサデータでも高速に動く、大規模製造データの定番。

仕組みのイメージ

たとえ話：
勾配ブースティングと同じ「誤差を順に補正」する考え方ですが、Microsoft社が「**どうすれば速く・省メモリで・大量データを処理できるか**」を徹底的に追求した実装版です。

ポイント：
木の成長方法を工夫（葉単位成長／ヒストグラムベース分割）し、大規模データで桁違いに高速。
それでいて精度はXGBoost並み。

現場ユースケース

■ 加工ラインの寸法・サイクルタイム予測

数秒間隔でセンサ値が記録される加工現場で、
数百万行のデータからリアルタイム寸法を予測。

■ 予知保全（設備寿命予測）

振動・電流・温度などのIoTセンサ大量データから、
軸受や工具の残寿命を高速予測。

▸ 長所

- **大規模データで圧倒的に高速**
- カテゴリ変数を直接扱える
- 欠損値もそのまま投入可
- 精度・速度のバランスが非常に良い

▸ 短所・注意点

- 少データ（〜数千行）では過学習しやすい
- ハイパーパラメータが多く本格調整は煩雑
- 葉単位成長は外れ値に過敏なことがある
- 解釈性は要SHAPなど追加解析

こんな時に選ぶ

◆ **IoTセンサデータが大量（数十万〜数百万行）**あるとき
◆ 学習・推論ともに速度が重要なリアルタイム用途
◆ XGBoostで遅すぎる場面、本番運用での実装の手軽さが欲しいとき

▸ **製造業との相性** PLC・センサからの時系列データと最も相性が良い。「データはあるが処理が追いつかない」現場の救世主。

GROUP 01 ・ 線形系

07

直交マッチング追跡回帰

Orthogonal Matching Pursuit (OMP)

ひと言で言うと

「効く特徴量を、決めた個数だけ」選んでくれる、スパース志向の線形回帰。

仕組みのイメージ

たとえ話：
「不良の原因を5つだけ挙げて」と指示すると、
現場で最も寄与の大きい順に5つを抜き出してくれる。

ポイント：
残った誤差に最も寄与する特徴量を1つずつ順に
追加していく方式。**「使う特徴量の数」を
最初に指定できる**のが、ラッソとの違いです。

現場ユースケース

■ 工程パラメータの絞り込み

50個の工程パラメータから「不良率に効く5個だけ」を取り出し、現場のチェックリストを作る。

■ センサ選定

多数のセンサのうち「品質予測に本当に必要な3個」を選び、設備のIoT化コストを抑える。

▸ 長所

- 使う特徴量数を**明示的に指定できる**
- 計算が速い（変数選択が逐次的）
- 説明性が高い（少数の特徴量で説明）
- センサ削減・モデル軽量化に向く

▸ 短所・注意点

- 非線形関係は捉えられない
- 選ぶ個数の指定が難しい（試行錯誤）
- 相関の強い特徴量から1つを選ぶときに不安定
- ラッソほど正則化の理論が整っていない

こんな時に選ぶ

◆ **「上位N個の効く要因だけが知りたい」**とき
◆ センサ・計測点の絞り込みで投資判断に使いたいとき
◆ ラッソより明示的に変数数をコントロールしたいとき

▸ **使いどころ** 「データを集めすぎたが何が本当に効くか分からない」現場でこそ価値を発揮する変数選択モデル。

GROUP 02 ・ 木 / アンサンブル系

08

XGBoost回帰

eXtreme Gradient Boosting

ひと言で言うと

高速・高精度・安定運用。世界中のデータ分析コンペで定番の、勾配ブースティング実装。

仕組みのイメージ

たとえ話：
勾配ブースティングと考え方は同じですが、
「正則化（過学習防止）」「並列計算」「欠損値の扱い」
をすべて**エンジニアリング的に磨き上げた**実装。

ポイント：
長年の実運用実績があり、本番システムへの組み込み
でも安定。LightGBMより少し遅いものの、
その分挙動が読みやすい。

現場ユースケース

■ 量産品の品質予測（本番運用）

高精度かつ安定動作が求められる本番ラインで、
各ロットの品質指標（強度・密度等）を予測。

■ OEE改善のための要因解析

停止時間・速度低下・不良率からなるOEEを多変量で
予測し、改善施策の効果見積もりに使う。

▸ 長所

- 製造業データで**最も外しにくい高精度**
- 正則化機能で過学習を抑えやすい
- 欠損値の自動処理が優秀
- 本番運用での安定実績が豊富

▸ 短所・注意点

- LightGBMより学習が遅い（大規模では明確に劣る）
- ハイパーパラメータが多く調整が必要
- ブラックボックス性（要SHAP/解釈ツール）
- 少データでは過学習リスクあり

こんな時に選ぶ

◆ **「精度を最優先」**の本番運用モデル
◆ データ規模が中規模（数千〜数十万行）
◆ 実績豊富で安心して使える「本命」モデルが欲しいとき

▸ **実務での立ち位置** XGBoost / LightGBM / CatBoost の三兄弟は必ず横並びで比較すべき。データ次第で勝者が入れ替わります。

GROUP 01 ・ 線形系

09

リッジ回帰

Ridge Regression (L2 Regularization)

ひと言で言うと

過学習を抑える線形回帰。温度・圧力など相関の強い工程パラメータを一緒に扱える。

仕組みのイメージ

たとえ話：
線形回帰だと、温度と圧力が連動しているとき
「温度＝+10、圧力＝-9」のような**極端で不安定な係数**に
なることがあります。
リッジは「係数を大きくしすぎないでね」という制約を
加えます。

ポイント：
係数の二乗和に罰則をかけることで、
相関のある特徴量同士で**バランスよく係数を分配**します。

現場ユースケース

■ 射出成形パラメータからの品質予測

シリンダ温度・金型温度・射出圧力・保圧など、
互いに相関する10〜20個のパラメータから品質値を予測。

■ 線形回帰で係数が暴れたとき

多重共線性が疑われる時の「修正版線形回帰」として使う。

▸ 長所

- **相関の強い特徴量**に強い
- 過学習を抑制（少データでも安定）
- 係数で説明できる（高い解釈性）
- 計算が高速、本番実装が容易

▸ 短所・注意点

- 非線形関係は捉えられない
- 変数選択はしない（不要な特徴量も残す）
- 正則化の強さ（α）の調整が必要
- 外れ値に弱い（Huber推奨）

こんな時に選ぶ

◆工程パラメータが**互いに相関している**（温度・圧力・速度など）
◆線形回帰で「係数の符号がおかしい」と感じたとき
◆ベースラインを少し賢くしたいとき、本番運用に堅く出したいとき

▸ **製造業との相性** 工程パラメータは大抵相関しています。線形回帰よりリッジを「標準のベースライン」にする価値は大きい。

GROUP 02 ・ 木 / アンサンブル系

10

AdaBoost回帰

Adaptive Boosting Regression

ひと言で言うと

「外したデータ」に重みを付け直して、次の弱学習器が集中対応する積み上げモデル。

仕組みのイメージ

たとえ話：
1人目の検査員が外した不良サンプルに、
2人目は「赤マーク」を付けて重点的に見る。
3人目は2人目が外したものに集中…と、
難しいサンプルへの注力を順送りする仕組み。

ポイント：勾配ブースティング登場以前の、ブースティングの古典的形態。今は出番が減りましたが、軽量で原理が分かりやすいモデルです。

現場ユースケース

■ 軽量モデルが必要な場面

数千行までのデータで、機材リソースに制限のあるエッジ機器上での予測。

■ ベンチマーク

勾配ブースティング系と比較する際のコントロール群。AdaBoostで充分な精度が出るなら、複雑なモデルは不要、という判断ができる。

▸ 長所

- 原理がシンプルで挙動が読みやすい
- 小〜中規模データで安定動作
- 計算が比較的軽量
- 過学習しにくい（弱学習器の組み合わせ）

▸ 短所・注意点

- **外れ値に弱い**（外したサンプルを過度に重視）
- 勾配ブースティング系より精度は劣る
- 大規模データで遅い
- 近年は出番が少ない（後継モデルに移行が進む）

こんな時に選ぶ

◆ **ブースティング系の比較対象**として横並びで評価したいとき
◆ データがクリーンで、外れ値の影響が少ない時
◆ シンプルな実装が望ましい教育・研究用途

▸ **立ち位置** 現代の主力はXGBoost/LightGBM/CatBoost。AdaBoostは「歴史的に重要、比較用」と割り切るのが現実的です。

GROUP 01 ・ 線形系

11

Huber回帰

Huber Regression (Robust Regression)

ひと言で言うと

外れ値に強い線形回帰。センサノイズや異常値が混じる現場データに最適。

仕組みのイメージ

たとえ話：
普通の線形回帰は「外れ値1点」に大きく引きずられて線が傾きます。
Huber回帰は「ちょっと外れた点は普通に扱うが**大きく外れた点はそこそこに扱う**」と賢く割り切ります。

ポイント：
誤差の大きさによって損失関数を切り替えるロバスト回帰。
外れ値の自動検出より、現場の運用に近い発想です。

現場ユースケース

■ 予知保全（振動センサデータ）

振動センサに突発スパイクが混じる現場で、
設備の残寿命を予測。
スパイクに引きずられず安定した予測線が引ける。

■ 温度・電流の異常混入データ

PLCログに通信エラーで異常値が混じるとき、
外れ値除去の前処理を省略してそのまま回帰できる。

▸ 長所

- **外れ値・突発ノイズに頑健**
- 線形係数で説明性が高い
- 計算が高速で安定
- 前処理の手間（外れ値除去）が減る

▸ 短所・注意点

- 非線形関係は捉えられない
- 外れ値が「半数近く」あると効果が薄れる
- 切り替え閾値（ε）の調整が必要
- 外れ値の「中身」を捨ててしまうリスク

こんな時に選ぶ

◆ **センサノイズ・通信エラーで外れ値が混じる**データ
◆ 外れ値除去ロジックを手前で組むのが現実的でないとき
◆ 線形回帰の結果が「数点の外れ値で大きく変わる」場合

▸ **注意点** 「外れ値は無視」してよいかは現場判断。本当はその外れ値こそが「異常検知すべき情報」のケースもあります。

GROUP 01 ・ 線形系

12

ラッソ回帰

Lasso Regression (L1 Regularization)

ひと言で言うと

不要な特徴量の係数をゼロにして自動的に「捨てる」、説明性の高い線形回帰。

仕組みのイメージ

たとえ話：
「30個ある工程パラメータのうち、本当に効くものだけ残して、あとは無視していい」と指示すれば、
ラッソが自動で**いらない係数を0にする**。

ポイント：
係数の絶対値の合計に罰則をかけるので、
不要な特徴量の係数が正確にゼロになる。
リッジは「小さくする」、
ラッソは「ゼロにする」と覚えると分かりやすい。

現場ユースケース

■ 工程パラメータの絞り込み

30個以上の工程パラメータから、
本当に品質に効く5〜6個を自動抽出。
現場のチェックポイント設計に直結。

■ 設備の状態監視

多数のセンサ信号のうち、不良予兆に効く項目だけを残し、
監視ダッシュボードを簡素化。

▸ 長所

- **自動で変数選択**してくれる
- 説明性が非常に高い（残った係数だけ説明）
- 計算が高速
- 過学習を抑制

▸ 短所・注意点

- 相関の強い特徴量からは1つだけ残す傾向
- 正則化の強さ（α）の調整が必要
- 非線形関係は扱えない
- 「捨てた変数」が本当に不要かは要検証

こんな時に選ぶ

◆ **特徴量が多すぎて何が効くか分からない**とき
◆ 「効く要因」を絞り込んで現場に展開したいとき
◆ 変数選択と予測モデル作成を同時にやりたいとき

▸ **現場での効用** 「残った5変数だけで現場説明」が可能。改善活動と監視項目の絞り込みに極めて実用的。

GROUP 01 ・ 線形系

13

ラッソLARS回帰

Lasso LARS (Least Angle Regression with Lasso)

ひと言で言うと

ラッソを「変数選択の経路」を効率的に求める形に変換した、軽量版ラッソ。

仕組みのイメージ

たとえ話：
ラッソが「全変数を見て徐々に係数を絞っていく」のに対し、LARSベースは「**最も効く変数から1つずつ追加していく**」。同じ結果に到達しますが、経路が違うため計算が速いことがあります。

ポイント：
特に「変数の数＞サンプル数」の状況や、変数選択の途中経過（係数パス）を見たいときに有用です。

現場ユースケース

■ 高次元データの変数選択

分光分析・スペクトル波形などで特徴量数が数百〜数千ある場合の変数絞り込み。

■ 効く要因の段階的可視化

正則化の強さを変えると、どの変数が先に残るかが順に見える。要因の重要度ランキングと相性が良い。

▸ 長所

- 変数選択の**経路（係数パス）**が見える
- 高次元データで効率的
- ラッソと同等の解釈性
- 変数追加の順序が要因重要度として読める

▸ 短所・注意点

- ノイズの強いデータには弱い
- 大規模データでは通常のラッソより遅いことも
- 解釈に統計的素養が必要
- 相関の強い特徴量で挙動が不安定になることがある

こんな時に選ぶ

◆ **特徴量数がサンプル数より多い**（分光・画像特徴など）
◆ 変数選択の「経路」を見ながら正則化強度を決めたいとき
◆ ラッソの結果が安定しない時の代替候補

▸ **立ち位置** 通常はラッソで十分。「特徴量数が多すぎる」「途中経過を見たい」という特殊状況での選択肢。

GROUP 03 ・ 近傍 / カーネル系

14

カーネルリッジ回帰

Kernel Ridge Regression

ひと言で言うと

リッジ回帰を非線形にも対応させた、滑らかな曲線関係を扱える線形系の拡張版。

仕組みのイメージ

たとえ話：
「直線では引けない関係」を、空間をぐにゃっと曲げて
直線が引けるようにしてから引く。
曲げ方を決めるのが「**カーネル関数**」と呼ばれる仕掛けです。

ポイント：
溶接の電流とビード強度のように「単調ではない、
ある範囲だけ急激に上がる」関係も自然に表現できます。

現場ユースケース

■ 溶接条件と強度の非線形関係

電流・速度・板厚と引張強度の関係は明確に非線形。
カーネルリッジで滑らかな予測曲面を作る。

■ 化学・熱処理条件の最適化

温度プロファイルと製品物性の関係など、
しきい値・ピークがある非線形関係を
中規模データでモデル化。

▸ 長所

- 非線形関係を**滑らかに**表現できる
- 少〜中規模データで高精度
- リッジの正則化で過学習を抑制
- 外挿性が木モデルより良い

▸ 短所・注意点

- **大規模データに弱い**（計算コストが急増）
- カーネル種・パラメータの選択が必要
- 解釈性が低い（係数で説明できない）
- 特徴量が多すぎると不安定

こんな時に選ぶ

◆ データ数が**数十〜数千**と中規模で、明確な非線形関係があるとき
◆ 木モデルより滑らかな予測曲面が欲しいとき
◆ 化学・材料・物理プロセスなど「物理的にも滑らかな関係」が想定されるとき

▸ **使いどころ** 「現象として滑らかなはず」という物理的直感があるとき、木モデルより自然な予測が得られます。

GROUP 02 ・ 木 / アンサンブル系

15

決定木回帰

Decision Tree Regression

ひと言で言うと

ルールが完全に追える単一の木。「圧力≧X かつ 温度≦Y なら不良率Z%」と説明できる。

仕組みのイメージ

たとえ話：
ベテラン工員の判断手順を「フローチャート」にしたもの。
「温度は180度以上か？ → はい、なら圧力は…」と
分岐していき、最後の葉に予測値が書いてある。

ポイント：
ランダムフォレストやXGBoostは
「これを大量に組み合わせた」もの。
決定木1本では精度は劣りますが、
「なぜそう予測したか」が完全に見えるのが最大の武器です。

現場ユースケース

■ 不良判定ルールの抽出

「圧力80以上かつ温度185以下なら不良率3倍」のような
現場に渡せる判定ルールを抽出。

■ 監査対応・QC会議資料

「なぜこの条件で警告を出したか」を
一枚の図で説明できる。
製造責任の説明資料に最適。

▸ 長所

- **完全な説明性**（ルールが追える）
- 非線形・しきい値挙動を表現できる
- 前処理不要（スケーリング不要）
- カテゴリ変数も扱える

▸ 短所・注意点

- **過学習しやすい**（深くすると暗記）
- 単独では精度が出にくい
- 少しのデータ変化で木構造が大きく変わる
- 外挿不可

こんな時に選ぶ

◆ **「なぜそう予測したか」を現場説明する必要**があるとき
◆ 不良判定ルールを抽出して現場に展開したいとき
◆ ランダムフォレストの「中身を1本だけ見たい」とき

▸ **現場での価値** 精度より「現場が納得して動いてくれるか」が大事な場面で、決定木の説明性は他では代替できない強みです。

GROUP 01 ・ 線形系

16

線形回帰

Linear Regression (Ordinary Least Squares)

ひと言で言うと

最も基本。まず「ベースライン」として走らせ、他モデルの精度比較の基準にする。

仕組みのイメージ

たとえ話：
散らばった点の真ん中を通る直線（または平面）を引く。
「Y ＝ 温度×0.5 ＋ 圧力×0.3 ＋ 10」のような
最も素朴な数式。

ポイント：
古典的ですが、係数の意味が明確で説明しやすく、
計算も瞬時。
何より重要なのは「**線形回帰で十分な精度が出るなら、複雑なモデルは要らない**」という判断ができること。

現場ユースケース

■ プロジェクト最初のベースライン

新しい予測課題で「まず一本」走らせる定番。
他モデルの精度がこれを上回るかでROIを判断。

■ 単純な物理関係の確認

「電流と発熱量」のような**物理的に線形な関係**では、これで十分な精度が出る。

▸ 長所

- **最高の説明性**（係数の意味が明確）
- 計算が瞬時、本番実装が極めて軽い
- 調整パラメータが事実上ない
- 統計検定の道具が豊富

▸ 短所・注意点

- 非線形関係は表現できない
- 相関の強い特徴量で**係数が暴れる**（多重共線性）
- 外れ値に弱い
- 特徴量数 > サンプル数では使えない

こんな時に選ぶ

◆ **何より「最初のベースライン」**として、必ず一回は走らせる
◆ 物理的に線形な関係であることが分かっているとき
◆ 結果を上司・現場に「最もシンプルに説明したい」とき

▸ **批評的視点** 「最新AI」を持ち出す前に、まず線形回帰でどこまで行けるか確認するのが、本来あるべき技術判断です。

GROUP 03 ・ 近傍 / カーネル系

17

k近傍法回帰

k-Nearest Neighbors Regression (k-NN)

ひと言で言うと

「似た過去事例k件の平均」で予測する、ベテラン工員の経験則に最も近い直感モデル。

仕組みのイメージ

たとえ話：
ベテラン工員が「この条件は3年前のあのロットと、去年のこのロットと似ている。平均して歩留まりは88%くらいだろう」と判断するのとほぼ同じ。

ポイント：
「学習」をほぼせず、予測のときに似たデータを毎回探す方式。**蓄積データ＝モデルそのもの**という、現場感覚に最も近い設計。

現場ユースケース

■ 類似ロットからの歩留まり予測

過去の類似条件ロット（材料・温度・湿度が近い）の歩留まり平均から、新ロットを予測。

■ 試作見積もり

新試作の条件に最も近い過去試作の結果を引っ張ってきて「だいたいこれくらい」を即答。

▸ 長所

- 直感的で**現場説明が容易**「過去の似たロット」
- 事前学習がほぼ不要
- 非線形関係も自然に表現
- 少データでも動く

▸ 短所・注意点

- **大規模データで遅い**（毎回距離計算）
- 特徴量数が多いと精度が落ちる（次元の呪い）
- スケーリングの前処理が必須
- 外挿不可（過去事例の範囲内のみ）

こんな時に選ぶ

◆ 「過去の似たロットを参考に」が現場の自然な発想と一致するとき
◆ データ数が数千件以内、特徴量も10〜20個程度のとき
◆ 現場説明のしやすさを最優先にしたいとき

▸ **製造業との相性** 「ベテランの判断」を模倣する最古典手法。現場感覚と一致するため、運用上の納得感が高い。

GROUP 01 ・ 線形系

18

エラスティックネット回帰

Elastic Net Regression (L1 + L2)

ひと言で言うと

リッジ（相関に強い）とラッソ（変数選択）の「いいとこ取り」。バランス型の線形回帰。

仕組みのイメージ

たとえ話：
「不要な変数は捨てて欲しい（ラッソ）」
「でも相関の強い変数のうち1つだけ残されると困る（リッジ）」という現場ニーズに、両方の罰則をブレンドして応えたモデル。

ポイント：
ブレンド比率（L1とL2の混合）を調整可能。
製造業データの「**相関が強い変数群がいくつもある**」状況で安定動作します。

現場ユースケース

■ 工程パラメータグループ単位の分析

「温度系3変数」「圧力系4変数」のように相関したグループから、まとまった単位で要因を選びたいとき。

■ ラッソが不安定な時の代替

ラッソで「実行ごとに残る変数が変わる」場合、エラスティックネットで安定化を図る。

▸ 長所

- 相関の強い**変数群を一緒に残せる**
- 変数選択と安定性のバランスが良い
- 過学習を抑制
- 解釈性が高い

▸ 短所・注意点

- 調整パラメータが2つ（α と L1比率）
- 非線形関係は扱えない
- 純粋なラッソ・リッジより複雑
- 外れ値には弱い

こんな時に選ぶ

◆ **変数選択もしたいが、相関グループは一緒に残したい**とき
◆ ラッソの結果が実行毎にばらつく場合
◆ リッジとラッソの中間的な動作を期待するとき

▸ **立ち位置** 線形系3兄弟（リッジ・ラッソ・エラスティックネット）は必ず横並びで比較するのが定石です。

GROUP 01 ・ 線形系

19

最小角回帰 (LARS)

Least Angle Regression

ひと言で言うと

「効く順」に特徴量を1つずつ追加していく、変数選択に強い線形回帰。

仕組みのイメージ

たとえ話：
「不良率に最も効く特徴量は温度。次に効くのは材料。
さらに次は…」と**優先度順に変数を1つずつ取り入れていく。**
途中経過が「重要度ランキング」として読める。

ポイント：
OMPやラッソLARSと考え方は近いですが、
こちらは「変数追加の角度」を理論的に最小にする手順を
取り、変数選択の理論的基礎として位置付けられています。

現場ユースケース

■ 要因の優先度可視化

工程パラメータを「効く順」に並べたい改善活動の初期分析。

■ 高次元データの変数選択

分光・スペクトルデータなど特徴量数が多い時、
変数選択の経路を理解しながら絞り込みたい場合。

▸ 長所

- **変数追加の順序＝重要度ランキング**
- 高次元データで効率的
- ラッソとほぼ同等の結果を高速に得られる
- 理論的基盤が明確

▸ 短所・注意点

- ノイズが強いと変数選択が不安定
- 非線形関係は扱えない
- 外れ値に弱い
- 相関の強い変数で挙動が複雑

こんな時に選ぶ

◆ **「要因の優先順位」**を出して現場提示したいとき
◆ 高次元データの変数選択を理論的に説明したいとき
◆ ラッソLARSと比較しながら最適モデルを選びたいとき

▸ **使いどころ** 「効く要因の順番」が知りたい改善活動と、最も相性の良い線形モデルの1つです。

05 COMPARISON TABLE (1/2)

アルゴリズム比較表

7つの観点から19アルゴリズムを ○ △ × で評価

アルゴリズム	データ量(大規模)	特徴量数(多い)	説明性	精度	速度	外れ値耐性	カテゴリ変数対応
線形回帰	○	×	○	×	○	×	×
リッジ回帰	○	△	○	△	○	×	×
ラッソ回帰	○	○	○	△	○	×	×
エラスティックネット	○	○	○	△	○	×	×
ラッソLARS	△	○	○	△	○	×	×
最小角回帰 (LARS)	△	○	○	△	○	×	×
ベイズリッジ	△	△	○	△	○	×	×
直交マッチング追跡 (OMP)	△	○	○	△	○	×	×
Huber回帰	○	△	○	△	○	○	×
決定木	△	△	○	×	○	△	○
ランダムフォレスト	○	○	△	○	△	○	○
エクストラツリー	○	○	△	○	○	○	○
勾配ブースティング	△	○	△	○	△	○	△
AdaBoost	△	△	△	△	△	×	△
XGBoost	○	○	△	○	○	○	○
LightGBM	○	○	△	○	○	○	○
CatBoost	○	○	△	○	△	○	○
k近傍法 (k-NN)	×	×	△	△	×	△	×
カーネルリッジ	×	△	×	○	△	×	×

凡例

○ 得意・対応可　△ 条件付き・標準的　× 不得意・要工夫

▸ **読み方のコツ** 「○が多い＝最良」ではありません。**課題で重視すべき列だけ**を見て候補を絞ってください。

05 COMPARISON TABLE (2/2)

7つの観点 — どこを見るべきか

比較軸の意味と、現場でどう判断材料にするか

▸ データ量

数万行を超える大規模データに耐えるか。
IoT・センサデータが多い現場なら最重要。
LightGBM / XGBoost が強い。

▸ 特徴量数

パラメータが数十〜数百ある場合に動くか。
ラッソ系・木系が強く、k-NNやカーネル系は弱い。

▸ 説明性

「なぜそう予測したか」が言葉で説明できるか。
製造業の品質説明責任で最重要。**線形系・決定木**が強い。

▸ 精度

予測値が真値にどれだけ近づくか。
あくまで「他観点が同等なら」の比較軸。
勾配ブースティング三兄弟が強い。

▸ 速度

学習・推論の速さ。
リアルタイム監視や頻繁な再学習が要件なら重要。
線形系・LightGBM が速い。

▸ 外れ値耐性

突発ノイズ・センサ異常値が混ざってもブレないか。
Huber・木系アンサンブルが強い。

▸ カテゴリ変数対応

号機・材料・シフトなど文字情報を直接扱えるか。
CatBoost・LightGBM が圧倒的に強い。

▸ 重視軸の決め方

①現場で説明が必要 → 説明性
②データが膨大 → データ量・速度
③ノイズだらけ → 外れ値耐性
④マスタが多い → カテゴリ対応
⑤精度が利益に直結 → 精度

標準的な比較セット例（実務で「まずこの3〜5個」）

ベースライン

線形回帰 / リッジ / ラッソ

本命モデル

ランダムフォレスト / XGBoost / LightGBM / CatBoost

特殊用途

ベイズリッジ(少データ) / Huber(外れ値) / k-NN(類似検索)

▸ **SFI活用** 上記の組合せをSFIで「一括実行→精度比較」するのが最効率です。

06 QUICK REFERENCE (1/3)

こんな時はこのアルゴリズム①

データの性質から逆引きする

01 データ量が少ない（〜100件、試作段階）

▶ ベイズリッジ / リッジ / k-NN
ベイズリッジなら信頼区間も出せる。
複雑モデルは過学習リスク大。

02 特徴量が多すぎて何が効くか分からない

▶ ラッソ / ラッソLARS / OMP
自動で不要変数を捨て、効く5〜6個に絞れる。
改善活動の起点に。

03 センサ異常値・外れ値が混じる

▶ Huber / ランダムフォレスト
Huberは線形の頑健版。
RFはアンサンブル平均で外れ値を希釈。

04 カテゴリ変数（号機/材料/シフト）が多い

▶ CatBoost / LightGBM
One-Hot不要、文字情報をそのまま投入できる。
前処理工数が劇減。

05 大規模IoTデータ（数百万行）

▶ LightGBM / XGBoost
LightGBMが学習速度で頭一つ抜ける。
リアルタイム用途も視野。

06 相関の強い特徴量が多い（温度・圧力・速度）

▶ リッジ / エラスティックネット
係数が暴れず、相関変数群をバランスよく扱える。

07 特徴量数 > サンプル数（分光・スペクトル等）

▶ ラッソLARS / 最小角回帰 (LARS) / OMP
高次元データでの変数選択が本来の用途。
線形回帰は使えない。

08 非線形な関係（しきい値・ピーク挙動）

▶ カーネルリッジ / 勾配ブースティング系
溶接条件×強度のような滑らかな非線形はカーネル、
複雑な分岐は木系。

▸ 次ページ ▸ 「目的・運用条件から選ぶ」「精度と説明性のトレードオフ」も整理しています。

06 QUICK REFERENCE (2/3)

こんな時はこのアルゴリズム ②

目的と運用条件から逆引きする

09 「なぜそう予測したか」を現場説明する必要

▶ 決定木 / 線形回帰 / ラッソ
決定木は判定ルールを抽出、
ラッソは効く変数だけ残す。監査対応にも。

10 とりあえずベースラインが欲しい（初回の調査）

▶ 線形回帰 / リッジ
最初に必ず走らせる定番。
これを超えるか否かが他モデル採用の判断基準。

11 安定運用しつつ精度も欲しい（本番展開）

▶ ランダムフォレスト / XGBoost / LightGBM
長期間の実績、ハイパラ調整しやすさ、
安定動作のバランスが良い。

12 予測の不確かさ（信頼区間）も知りたい

▶ ベイズリッジ
「点予測 ± 範囲」を出せる。
設計判断・リスク評価で重宝する。

13 高精度が最優先（最終判定モデル）

▶ XGBoost / LightGBM / CatBoost / 勾配ブースティング
必ず三兄弟を横並び比較。
データ次第で順位が変わる。

14 要因の重要度を可視化したい

▶ ランダムフォレスト / 勾配ブースティング系 / ラッソ
特徴量重要度（or係数）が明確に出る。
改善活動の優先順位付けに。

15 過去の類似ロットから歩留まり予測したい

▶ k近傍法 (k-NN)
ベテラン工員の「あの時と似ている」発想を、
最も素直に再現するモデル。

16 エッジ機器・PLCに組み込む軽量モデル

▶ 線形回帰 / リッジ / ラッソ / 決定木
係数式 or ルール式で実装が極めて軽い。
本番運用のメンテも容易。

▸ 重要
どの推奨も「絶対」ではありません。SFIで2〜3候補を横並びで比較し、現場で最も納得感のあるものを選ぶのが定石です。

06 QUICK REFERENCE (3/3)

工程別・課題別レシピ集

「うちの工程ではどれから試すか」の現場別レシピ

⚙️ 射出成形 温度・圧力・速度・保圧・材料・金型・号機・シフト

ベースライン：リッジ回帰（温度・圧力・速度が相関するため）
本命：CatBoost（材料メーカー・号機・金型IDなどカテゴリ多数）
説明用：ランダムフォレストで特徴量重要度を確認、現場展開は決定木でルール化

⚡ 溶接 電流・電圧・速度・板厚・シールドガス・継手

ベースライン：線形回帰（物理関係の初期確認）
本命：カーネルリッジ（電流×強度は非線形が明確）
代替：勾配ブースティング系（非線形と相互作用を捕捉）

🛠️ 加工（切削・研削） 送り速度・主軸回転・工具摩耗・振動センサ

ベースライン：リッジ / ラッソ
本命：LightGBM（センサ大量データを高速処理）
外れ値あり：Huber回帰（突発スパイクが混じるとき）

🔩 組立 作業者・治具・部品ロット・トルク・サイクルタイム

本命：エクストラツリー / CatBoost（作業者・治具がカテゴリ）
類似ロット参照：k-NN（過去の類似条件をそのまま参照）
改善初期：ラッソで効く要因を5〜6個に絞る

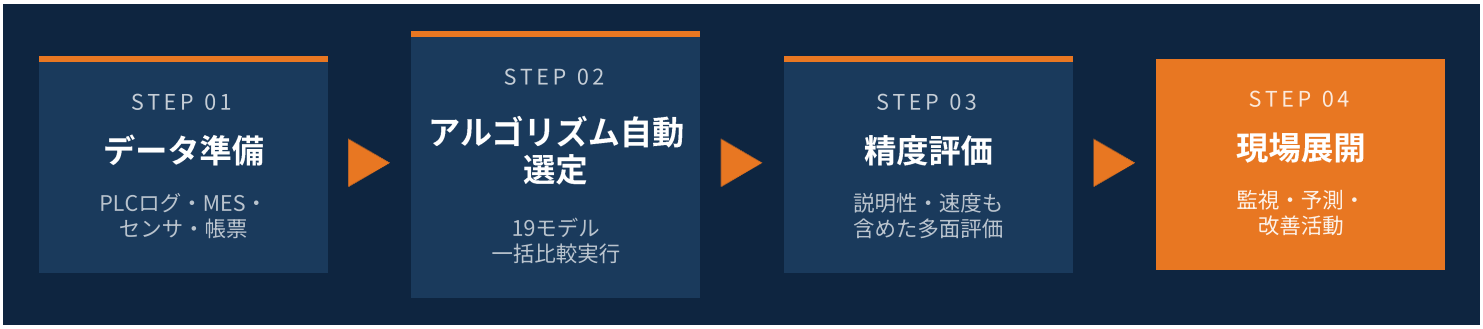
🔍 品質管理 / OEE改善 不良率・歩留まり・停止時間・速度・サイクル

本命：ランダムフォレスト → XGBoost（要因解析＋本番予測）
少データ試作：ベイズリッジ（信頼区間付き）

07 SFI WORKFLOW (1/2)

SFI活用の流れ

データ準備から現場展開まで



STEP 01

データ準備

◆ PLC・MES・帳票・センサからの統合
◆ 時刻同期・単位統一
◆ 異常値の確認（除去ではなく確認）
◆ 欠損の発生パターン把握
◆ **現場知見の「特徴量化」**

▸ **最重要：**精度の半分以上はここで決まる。
「現場の人が経験的に効くと思っている要因」を漏れなく数値化する。

STEP 02

アルゴリズム自動選定

◆ **19アルゴリズムを一括実行**
◆ クロスバリデーションで公平比較
◆ R^2、RMSE、MAEなど複数指標で評価
◆ 上位3〜5モデルを自動抽出
◆ ベースラインとの差を可視化

▸ **SFIの本質的価値：**「迷ったら全部試せる」。
1モデルだけ選ぶより、横並び比較の意思決定の方が誠実です。

STEP 03

精度評価（多面的に）

◆ 精度 — 当然見るが、それだけではない
◆ 説明性 — 現場・上司に説明できるか
◆ 速度 — 本番運用に耐えるか
◆ 安定性 — 再学習でブレないか
◆ **「現場感覚」と一致するか**

▸ **注意：**精度数値だけで選ぶと、
現場で使われないモデルになります。

STEP 04

現場展開

◆ 監視ダッシュボードへの組込み
◆ アラート閾値設定
◆ 現場説明・OJT
◆ **予測ハズレ事例の収集**
◆ 定期的な再学習サイクルの設計

▸ **本質：**モデルは「作って終わり」ではなく、
現場の暗黙知を吸い上げ続ける仕組み。

▸ **次ページ** SFIが「迷ったら全部試せる」プラットフォームである価値を、さらに掘り下げます。

07 SFI WORKFLOW (2/2)

なぜ「迷ったら全部試せる」が価値か

SFIの本質的な存在意義

▸ KEY MESSAGE

最初から「正解のアルゴリズム」を当てるのは
経験豊富なデータサイエンティストでも難しい。
ならば、19個まとめて走らせて比較すればよい。

01

「最新AI＝偉い」を相対化できる

線形回帰で十分なケースが、
19比較すれば即座に分かる。
複雑モデルに振り回されない
判断基盤。

02

「精度 vs 説明性」を同時に検討できる

XGBoostの精度と、ラッソの説明性を
「並べて」見れる。
意思決定者と現場の双方を説得できる。

03

試行錯誤の属人化を防ぐ

「あの担当者しか分からない
選定理由」を排除。
再現性と引継ぎ可能な意思決定
プロセスを残せる。

▸ ANTI-PATTERN —— SFIを使わない場合の典型的な失敗

① 流行に乗ってXGBoostだけ試す → 線形回帰で同等精度だったことに気付かない
② ベンダーの推奨で1モデル採用 → 別モデルの方が現場説明しやすかった
③ 担当者の好みでk-NNを使う → 大規模化で破綻、他案への切替コストが膨大に

▸ SFIの存在意義

アルゴリズムを「選ぶ」のではなく、「並べて比較する」。
その上で、現場知見と突き合わせて意思決定する。これがSFIの設計思想です。

08 COMMON MISUNDERSTANDINGS (1/2)

よくある誤解と注意点

バズワードの裏側で見落とされがちな本質

「精度が高い＝良いモデル」

✕

▸ **なぜ問題か：**テストデータに対する精度99%でも、本番の新ロットで30%、というのは過学習の典型。さらに、「なぜ予測したか」が説明できなければ品質管理の文脈では使えません。
▸ **本質：**精度・説明性・運用性・現場感覚との一致、の**4軸での評価**が必要です。

「最新のAIモデル＝最良」

✕

▸ **なぜ問題か：**製造業の予測課題は、線形回帰やリッジ回帰で十分なケースが少なくありません。にもかかわらず「DXだから最新AI」となると、運用コスト・説明責任・属人化リスクが膨れ上がります。
▸ **本質：**「**最も単純で目的を達成できるモデル**」を選ぶのがエンジニアリングの規律です。

「データさえあれば予測できる」

✕

▸ **なぜ問題か：**ゴミデータからは何も生まれません。「どの特徴量を作るか」「異常値をどう扱うか」「現場知見をどう数値化するか」という**人間の判断**が、精度の大半を決めます。
▸ **本質：**アルゴリズム選定の前に、**特徴量設計と現場知見の言語化**に時間を割くべきです。

「ブラックボックスでも結果が出ればOK」

✕

▸ **なぜ問題か：**製造業は**品質保証・トレーサビリティ・PL法**のもと、説明責任が問われる業界です。「AIが言ったから」で不良品流出は通用しません。客先監査・行政対応で破綻します。
▸ **本質：**説明性の高いモデル（線形系・決定木）を、最低限のサブモデルとして併存させるのが現実解です。

▸ **続き** 次ページ：特徴量設計の重要性／時系列特有の落とし穴／運用フェーズの誤解

08 COMMON MISUNDERSTANDINGS (2/2)

運用フェーズで起きる誤解

モデルは作って終わりではなく、運用が本番

「特徴量は多ければ多いほど良い」

✕

▸ **なぜ問題か：**無関係な特徴量はノイズになり、特に少データでは精度を下げます。
また、運用コスト（データ取得・前処理）が膨らみ、運用負荷を圧迫します。
▸ **本質：ラッソ・OMP・LARSなどで「効く5〜10個」に絞る**のが現場運用のコツです。

「一度作ったモデルは長く使える」

✕

▸ **なぜ問題か：**材料ロットの変更、設備の経年劣化、季節・湿度の変化、
オペレータの交替などで、現場のデータ分布は常に動きます。古いモデルは静かに精度を落とします。
▸ **本質：**定期的な**再学習サイクル**（月次・四半期）と、**精度モニタリング**を運用設計に組み込むことが必須です。

「時系列データに普通の回帰を使えばOK」

✕

▸ **なぜ問題か：**時系列データを単純な回帰で扱うと、**「未来のデータで過去を予測する」**というリークが起きやすく、テスト精度が異常に高く出る誤った成功体験を生みます。
▸ **本質：**時系列はラグ特徴量の設計・時系列クロスバリデーションが必須。
「いつのデータでいつを予測しているか」を常に意識します。

「相関＝因果」

✕

▸ **なぜ問題か：**「気温と不良率に強い相関」が出ても、原因は「気温」ではなく
「気温が高い日は湿度も高い」「夏は人員配置が変わる」など別要因かもしれません。
回帰の係数を「原因」と取り違えると、改善打ち手を間違えます。
▸ **本質：**回帰モデルが示すのは**「関係性」であって「因果」ではない。**
現場知見との突き合わせが不可欠です。

▸ **道具はあくまで道具です。使い手の規律と現場感覚こそが、本質的な品質を作ります。**

09 SUMMARY

まとめ

アルゴリズムは**道具**。
選び方は**現場知見**が決める。

19種類の回帰アルゴリズムは、どれも「正解」ではなく「特性の違う道具」です。
現場の課題・データの性質・運用条件・説明責任を踏まえて、**適切な道具を、適切な使い方で**選ぶこと。それが製造業データ解析の本質です。

① バズワードに流されない

「最新AI＝偉い」を相対化。
線形回帰で十分なら、それが最良の選択です。

② 迷ったら全部試して比較

SFIは「19モデル一括実行・横並び比較」のためのプラットフォーム。
これがSFIの存在意義です。

③ 現場知見と必ず突き合わせる

回帰モデルは「関係性」であって「因果」ではない。
現場感覚と合うかを最後の検証軸に。

CONTACT

お問い合わせ

SFI導入・運用・データ活用についてのご相談は、Vertys（ヴァーティス）まで。
製造業のための、本質的に機能するデータ解析を。

S SFI
VERTYS Inc.

SFI 回帰分析アルゴリズム解説資料
Vertys Inc. | 2026