SFI
VERTYS Inc.

CLASSIFICATION ALGORITHM GUIDE

SFI
分類分析アルゴリズム
解説資料

～ 現場で使える15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

15

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。
現場の判定課題に、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、良品/不良品の判定、不良モードの分類、設備の正常/異常判別など、
「カテゴリを当てる」場面が日常的にあります。
SFIには15種類の分類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、
「最新のAIをまず試す」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、
ご自身の判定課題に対して「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

「クラス不均衡」「説明性が必要」「外れ値が多い」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

品質会議や監査の場で「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

「精度数値」に振り回されない

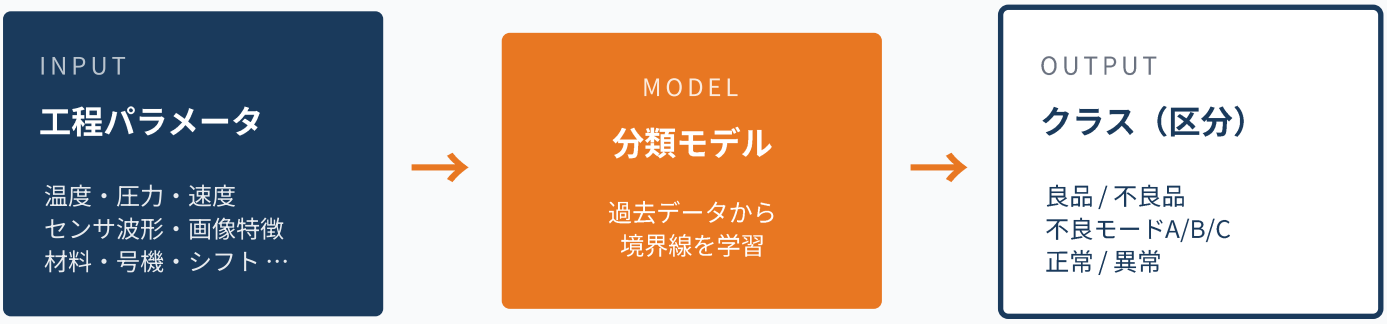
分類では「正答率99%」が誤解を招く場面が多い。
混同行列・再現率・適合率の意味を踏まえた判断ができる。

02 WHAT IS CLASSIFICATION

分類分析とは何か

測定値や過去データから、データが「どのカテゴリ」に属するかを判定する仕組み

CONCEPT



分類と回帰の違い

分類 Classification（本資料の対象）

カテゴリ（区分）を当てる
→ 良品 / 不良品
→ 不良モードA / B / C
→ 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

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

製造現場における分類の代表例

- 良品 / 不良品判定
- 不良モードの分類
- A級 / B級 / C級判定
- 設備の正常 / 異常判別
- 異常兆候の早期検知
- 製品の取引先別分類
- 工程の安定 / 不安定判定
- 故障モード推定
- 受入検査の合否判定

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**	本資料の目的と分類分析の基礎
P.05-06	**全体マップ**	15アルゴリズムを3グループに分けて俯瞰
P.07-21	**アルゴリズム詳細**	推奨順1〜15位、1アルゴリズム1ページ
P.22-23	**比較表**	7観点 ×15アルゴリズムの◯△×評価
P.24-26	**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.27-28	**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.29-30	**よくある誤解と注意点**	「正答率99%」が誤解を招く理由など
P.31	**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

木/アンサンブル系 — 精度と汎用性が高い

線形系 — 説明性が高く高速

近傍/確率系 — 直感的・少データ対応

04 ALGORITHM MAP

15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
分類モデル
15種

GROUP 01
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、推奨スコア上位を独占する主力モデル群。

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

GROUP 02
線形系
Linear Family / 5種

直線（または曲線）でクラスの境界を引く。
説明性が高く、高速で、ベースラインとして必須のモデル群。

- ◆ 線形判別分析（LDA）
- ◆ 二次判別分析（QDA）
- ◆ リッジ分類
- ◆ ロジスティック回帰
- ◆ 線形SVM

GROUP 03
近傍 / 確率系
Neighbor & Probabilistic / 2種

「過去の似た事例」を参照、
あるいは確率モデルで判定。
少データ・前処理少・直感的判定の場面で活躍。

- ◆ k近傍法 (k-NN)
- ◆ ガウシアン ナイーブベイズ

数は少ないですが、
木でも線形でも扱いにくい場面で
独自の強みを発揮します。

0 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

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判定する。製造業データで推奨スコア上位を独占。

▸ 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)、不均衡データ

▸ 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 02

線形系

Linear Family

特徴量の組合せでクラス境界を「直線（や曲線）」で引く。説明性が高く、本番運用の堅牢さでベースラインに最適。

▸ 得意なこと

説明性、ベースライン作り、確率値出力（ロジスティック）、本番運用の軽量実装、判別の見える化（LDA/QDA）

▸ 苦手なこと

複雑な非線形関係、特徴量間の強い相互作用、カテゴリ変数のそのままの扱い

GROUP 03

近傍 / 確率系

Neighbor & Probabilistic

「似た過去事例の多数決」「確率モデルでの判定」など、直感的な発想に最も近いアプローチ。前処理が少なく試作向き。

▸ 得意なこと

少データ、ベテラン判断の模倣(k-NN)、超高速分類(GNB)、説明のしやすさ

▸ 苦手なこと

大規模データ（k-NN）、特徴量間の独立性が崩れる場面（GNB）、高次元データ

POINT
推奨スコアが高いのは木系ですが、「絶対の正解」ではありません。
説明性や運用条件次第で線形系・確率系が正解になる場面も多々あります。

GROUP 01 ・ 木 / アンサンブル系

ランダムフォレスト分類

Random Forest Classification

01

RANK 1 / 15

ひと言で言うと

複数の決定木を組み合わせる推奨スコア No.1。
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、
多数決で結論を出す。
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、その多数決で予測。
「まずこれで様子見」が定石。

現場ユースケース

■ 不良品 / 良品判定

工程センサ・寸法測定値・材料情報から、
ロット毎の良品/不良品を判定。
特徴量重要度で「どのパラメータが効くか」が見える。

■ 不良モードの分類

「気泡」「バリ」「変色」など複数の不良モードを
多クラス分類。設備別の不良傾向把握にも使える。

▸ 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（実測 3.9秒）

▸ 短所・注意点

- 個々の判定ロジックは見えない（ブラックボックス性）
- 学習範囲外の予測には対応不可
- クラス不均衡では工夫が要る
- 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

▸ 推奨スコア1位の理由
「予測精度」と「要因ランキング」の両方が出るのが分類の現場で最強の武器。改善活動の起点になります。

GROUP 01 ・ 木 / アンサンブル系

CatBoost分類

CatBoost Classification

02

RANK 2 / 15

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（推奨スコア0.99）。

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定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

■ 多品種少量生産の不良品判定

「材料メーカー」「号機番号」「金型ID」「シフト」「品番」などカテゴリ情報が大量にある工程で、良品/不良品を高精度に判定。

■ 受入検査の合否判定

取引先・材料グレード・ロット番号など、文字情報を主軸にした合否判定。

▸ 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**（実測 56秒、最遅クラス）
- 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
- 「なぜそう判定したか」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 01 ・ 木 / アンサンブル系

XGBoost分類

eXtreme Gradient Boosting Classification

03

RANK 3 / 15

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（推奨スコア0.90）。

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。
CatBoostよりも実用的に速く（実測 13.6秒）、
LightGBMより挙動が読みやすい。

現場ユースケース

■ 量産ラインの本番判定モデル

高精度かつ安定動作が求められる本番ラインで、
各製品の良品/不良品判定や、不良モード分類を実行。

■ 異常検知（多クラス）

設備の正常／予兆異常／故障の3クラス判別など、
不均衡データでも安定動作。

▸ 長所

- 製造業データで**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 ・ 木 / アンサンブル系

LightGBM分類

Light Gradient Boosting Machine Classification

04

RANK 4 / 15

ひと言で言うと

数百万行のIoTセンサデータでも高速に判定。
大規模製造データの定番（推奨スコア0.67、実測 8.3秒）。

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

■ リアルタイム異常検知

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に正常/異常を判定。

■ 大量画像特徴量からの分類

外観検査の画像特徴量が数百次元になる場合の、
不良モード多クラス分類。

▸ 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 ・ 木 / アンサンブル系

エクストラツリー分類

Extra Trees Classification

05

RANK 5 / 15

ひと言で言うと

ランダム性をあえて強くした森。
学習が極めて速く（実測 1.9秒）、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、その多数決をとる。
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、それでも多数決で精度が出ます。

現場ユースケース

■ 組立工程の合否判定

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判定。

■ ノイズの多い検査データ

表面測定値や振動データなど、ノイズの多い特徴量に対して過学習せず安定した分類モデルを構築。

▸ 長所

- 学習が速い（ランダムフォレストの約2倍速）
- 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 03 ・ 近傍 / 確率系

k近傍法分類

k-Nearest Neighbors Classification (k-NN)

06

RANK 6 / 15

ひと言で言うと

「似た過去事例k件の多数決」で判定する、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「このサンプルは3年前のあのロットと、去年のこのロットと似ている。
両方とも不良だったから、たぶんこれも不良」と判断するのとほぼ同じ。

ポイント：
「学習」をほぼせず、判定のときに似たデータk件を毎回探して多数決を取る方式。
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

■ 類似ロット参照による合否判定

過去の類似条件ロット（材料・温度・湿度が近い）の判定結果から、新ロットの合否を予測。

■ 異常パターンの類似度判定

過去の異常事例データベースを基に、「今のセンサ波形は過去の○○異常に似ている」を提示。

▸ 長所

- 直感的で**現場説明が容易**「過去の似たロット」
- 事前学習が不要（実測 学習時間 0秒）
- 非線形な境界も自然に表現
- 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
- クラス不均衡だと多数派に引きずられる

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 02 ・ 線形系

線形判別分析（LDA）

Linear Discriminant Analysis

07

RANK 7 / 15

ひと言で言うと

クラスの分かれやすさを見る線形判別。最速クラスの分類モデル（速度寄与1位）。

仕組みのイメージ

たとえ話：
良品と不良品の特徴量を散布図にプロットし、
「2つの集団が最もきれいに分かれて見える方向」に
直線を引きます。その線で区切ったらどちらに入るかで判定。

ポイント：
クラス間のばらつきは大きく、クラス内のばらつきは
小さくなる方向を統計的に求める手法。
判別の根拠が「軸」として明確に見えるのが強み。

現場ユースケース

■ 多変量検査値の合否判定

引張強度・伸び・硬さなど複数の検査値から、
合格/不合格を判定。「どの特徴量で分かれているか」が
可視化できる。

■ 不良モード判別の初期分析

不良サンプル群がいくつかの不良モードに分けられるか、
「分離しやすさ」を統計的に評価。

▸ 長所

- **超高速**（速度寄与1.0、最速クラス）
- 判別軸の可視化で解釈性が高い
- 少データでも安定動作
- 多クラス分類にも自然に拡張

▸ 短所・注意点

- **境界が直線**に限られる
- クラスごとに正規分布の仮定が必要
- 外れ値に弱い
- 特徴量間の相関が強いと不安定

こんな時に選ぶ

◆ **「2つのクラスが何で分かれているか」**を統計的に説明したいとき
◆ 学習時間が極めて短く済むベースラインが欲しいとき
◆ クラスが正規分布に近い、整った検査データの初期分析

▸ **立ち位置** 古典的だが今でも有用。「説明可能性」が必須の場面で根強い人気がある統計の伝統技。

GROUP 02 ・ 線形系

二次判別分析（QDA）

Quadratic Discriminant Analysis

08

RANK 8 / 15

ひと言で言うと

LDAを曲線の境界（二次曲線）にも対応させた判別。
クラスごとに散らばり方が違うときに有効。

仕組みのイメージ

たとえ話：
良品は中央付近にコンパクトに集まり、
不良品は外側に幅広く散らばっている、
というように**クラスごとに散らばり方が違う**場合、
LDAの直線ではうまく区切れません。
QDAは曲線で囲うように区切ります。

ポイント：
LDAの「全クラス共通の散らばり」という仮定を緩め、
クラスごとに別の散らばりを許す形に拡張したもの。

現場ユースケース

■ ばらつきが違うクラスの判別

良品は精度よく管理されているがばらつき小、
不良品はモードによって大きくばらつく場合の判別。

■ 設備の状態判別

正常時はパラメータが狭い範囲に収まり、
異常時は広く散らばる状況での正常/異常の二値判別。

▸ 長所

- クラスごとの**ばらつきの違いを表現できる**
- 非線形（二次曲線）の境界が引ける
- 高速（速度寄与0.79）
- 確率的な判定結果が得られる

▸ 短所・注意点

- クラスごとに正規分布の仮定が必要
- 少データだとパラメータ推定が不安定
- 特徴量数が多いと計算量が増える
- 外れ値に弱い

こんな時に選ぶ

◆ **良品と不良品で「ばらつき具合」が明らかに違う**とき
◆ LDAで境界が直線だと精度が不足するとき
◆ 統計的に説明可能で、それでいて非線形にも対応したいとき

▸ **立ち位置** LDAとセットで試すのが定石。精度がLDA以上で、解釈性が木モデル以上、という独自のポジション。

GROUP 02 ・ 線形系

リッジ分類

Ridge Classification (L2 Regularization)

09

RANK 9 / 15

ひと言で言うと

過学習を抑えやすい線形分類。相関の強い特徴量があっても係数が暴れない安定モデル。

仕組みのイメージ

たとえ話：
普通の線形分類だと、温度・圧力・速度のように
連動するパラメータがあると係数が極端な値になりがち。
リッジは「係数を大きくしすぎないでね」と制約を加えることで、安定した判別線を引きます。

ポイント：
リッジ回帰の分類版。
判定の確率値は出ませんが、その分シンプルで速い。

現場ユースケース

■ 工程パラメータが相関するとき

シリンダ温度・金型温度・射出圧力・保圧など、
互いに相関するパラメータからの良品/不良品判定。

■ 軽量ベースライン

最初に必ず一回試す、
シンプルで堅牢なベースライン分類モデル。

▸ 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
- 確率値は出力されない（必要ならロジスティック回帰）
- 正則化の強さ（α）の調整が必要
- 多クラス分類は工夫が要る

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）
◆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
◆ロジスティック回帰より確率値が要らず、速度を重視したいとき

▸ **製造業との相性** 工程パラメータは大抵相関しています。
「素朴な線形分類」よりリッジ分類を標準ベースラインにする価値は大きい。

GROUP 02 ・ 線形系

ロジスティック回帰

Logistic Regression

10

RANK 10 / 15

ひと言で言うと

シンプルで解釈しやすい線形分類。
「不良である確率は78%」のように確率値が出るのが強み。

仕組みのイメージ

たとえ話：
線形回帰で得た数値を、0〜1の確率値に変換する
「ロジスティック関数（S字曲線）」を通します。
「不良率0.78 → 不良と判定」のように、
判定根拠を確率で示せます。

ポイント：
名前に「回帰」と入っていますが、
れっきとした分類モデル。係数の意味が明確で、
医療・金融など説明責任の重い分野でも長年使われている
王道モデルです。

現場ユースケース

■ 確率付きの不良予測

「このロットが不良になる確率は78%です」と
確率値付きで提示。アラート閾値の現場調整がしやすい。

■ 現場説明・監査資料

「温度が1度上がると不良確率が3%増える」など、
係数で説明できるため監査対応にも強い。

▸ 長所

- **確率値が出力**される（閾値調整可能）
- 係数で説明できる（最高の解釈性）
- 計算が瞬時、本番実装が極めて軽い
- 多クラス分類にも自然に拡張可能

▸ 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**
- 外れ値に弱い
- クラス不均衡では予測確率がずれやすい

こんな時に選ぶ

◆ **「不良の確率値」**を出してアラート閾値を現場で調整したいとき
◆ 判定根拠を係数で説明する必要があるとき（監査対応など）
◆ ベースライン分類モデルとして必ず一度走らせる

▸ **批評的視点** 「最新AI」を持ち出す前に、まずロジスティック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 02 ・ 線形系

線形SVM分類

Linear Support Vector Machine

11

RANK 11 / 15

ひと言で言うと

クラス間のマージン（余白）を最大化する線形分類。判定の「迷いにくさ」を最適化する。

仕組みのイメージ

たとえ話：
良品と不良品の散布図に線を引くとき、
「**どちらの集団にも一番余裕がある場所**」に線を引きます。
境界近くのサンプル（サポートベクター）から最大限離れた位置を選ぶイメージ。

ポイント：
判定境界の「ロバスト性」を最大化するため、
新しいデータに対しても外しにくい。
高次元データに強い。

現場ユースケース

■ 高次元データの分類

画像から抽出した特徴量や、スペクトルデータなど
特徴量数が100以上ある場合の二値判別。

■ 境界近くの判定が重要な場面

合否判定の閾値付近で「迷う」サンプルを最小化したいとき、
SVMのマージン最大化が威力を発揮。

▸ 長所

- **高次元データに強い**
- 境界のロバスト性が高い
- 過学習しにくい
- カーネルSVMに拡張可能（非線形対応）

▸ 短所・注意点

- 確率値が出ない（要追加処理）
- 大規模データで学習が遅くなる
- パラメータ調整（C値）の影響が大きい
- 多クラス分類は工夫が要る

こんな時に選ぶ

◆ **特徴量数 ≫ サンプル数**（画像特徴・スペクトルなど）
◆ 判定境界のロバスト性を最優先したいとき
◆ ロジスティック回帰より境界が「安定」している分類器が欲しいとき

▸ **立ち位置** かつての「最強分類器」。今は木系に主役を譲ったが、高次元データでは依然として強力な選択肢。

GROUP 01 ・ 木 / アンサンブル系

決定木分類

Decision Tree Classification

12

RANK 12 / 15

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」と現場に渡せるルールを抽出できる。

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「不良」「良品」の
ラベルが書いてある。

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、
「なぜそう判定したか」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」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

「なぜこの条件で不良判定したか」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（推奨スコア0.09）
- 少しのデータ変化で木構造が大きく変わる
- クラス不均衡に弱い

こんな時に選ぶ

◆ **「なぜそう判定したか」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 ・ 木 / アンサンブル系

勾配ブースティング分類

Gradient Boosting Classification (GBC)

13

RANK 13 / 15

ひと言で言うと

誤差を補いながら積み上げる木モデル。原理は強力だが、実装としてはXGBoost/LightGBMに譲るスタンダード版。

仕組みのイメージ

たとえ話：
最初の検査員が「これは不良」と判定する。
次の検査員は「最初の人が間違えたサンプル」に集中して再判定する。
これを100〜1000回繰り返して全体の精度を上げます。

ポイント：
非常に強力な原理ですが、scikit-learnの標準実装は**速度が遅い**（実測 145秒・最遅）。
実務ではXGBoost/LightGBM/CatBoostを使うのが定石です。

現場ユースケース

■ XGBoost等の比較対象

勾配ブースティングの「素」の挙動を確認し、XGBoost/LightGBM/CatBoostとの差を理解するベンチマーク。

■ 教育・原理理解

ブースティングの原理を学ぶ際の標準実装。
本番運用ではあまり選ばれません。

▸ 長所

- 原理がシンプルで挙動が読みやすい
- 中規模データで安定動作
- 過学習しにくい（弱学習器の組み合わせ）
- 非線形・相互作用を捉えられる

▸ 短所・注意点

- **学習時間が極端に長い**（145秒・最遅）
- 並列計算がしづらい
- カテゴリ変数の前処理が必要
- 近年は出番が少ない（XGBoost等に移行）

こんな時に選ぶ

◆ **ブースティング系の挙動を素の状態で比較**したいとき
◆ XGBoost/LightGBMが何らかの理由で使えない環境
◆ 速度より「ベースラインの統一性」を優先したいベンチマーク

▸ 実務での立ち位置

「勾配ブースティング」は概念。XGBoost/LightGBM/CatBoostがその実用版。本番運用では後者を選ぶのが定石。

GROUP 03 ・ 近傍 / 確率系

ガウシアンナイーブベイズ

Gaussian Naive Bayes (GNB)

14

RANK 14 / 15

ひと言で言うと

特徴量同士の独立性を仮定した超高速分類。
仮定が大胆な代わりに、実装が極めて軽い確率分類器。

仕組みのイメージ

たとえ話：
「温度・圧力・速度はそれぞれ独立に効く」と**大胆に仮定**して、各特徴量から不良確率を計算し、掛け算でまとめます。
実際には特徴量は相関しますが、それでも意外と動きます。

ポイント：

仮定が雑な分、計算は超高速・実装は数行で済みます。
スパムフィルタなどテキスト分類で広く使われている手法。

現場ユースケース

■ ベースラインの最速比較

他のアルゴリズムの精度を評価する際の
「下限」として走らせる。
これに勝てないモデルは採用に値しない。

■ エッジ機器での簡易判定

PLCや簡易マイコンに組み込む超軽量分類器として、
ルールベースに代わる選択肢。

▸ 長所

- **超高速**（速度寄与1.0、最速クラス）
- 確率値が出力される
- 少データでも動く
- 多クラス分類に自然に対応

▸ 短所・注意点

- **独立性の仮定**が現実とずれる場面が多い
- 精度はあまり期待できない（推奨スコア0）
- 特徴量が正規分布前提（ガウシアン版の場合）
- 確率値の絶対値は信頼しにくい

こんな時に選ぶ

◆ **速度最優先**のエッジ機器・PLC組込み
◆ 他モデルの精度評価の「下限ベースライン」として
◆ 数行で実装が必要な簡易プロトタイプ

▸ **立ち位置** 製造業の本番運用ではあまり選ばれませんが、「これに勝てないモデルはダメ」という基準線として価値があります。

GROUP 01 ・ 木 / アンサンブル系

AdaBoost分類

Adaptive Boosting Classification

15

RANK 15 / 15

ひと言で言うと

弱いモデルを順に強化する古典的ブースティング。
歴史的に重要だが、本番運用では後継モデルに譲るレガシー。

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
エッジ機器上での分類。

■ 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（実測 0.37秒）
- 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（推奨スコア0）
- 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）

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。

05 COMPARISON TABLE (1/2)

アルゴリズム比較表

7つの観点から15アルゴリズムを ○ △ × で評価（推奨順）

順位	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
1	ランダムフォレスト	○	○	△	○	△	○	○
2	CatBoost	○	○	△	○	×	○	○
3	XGBoost	○	○	△	○	○	○	○
4	LightGBM	○	○	△	○	○	○	○
5	エクストラツリー	○	○	△	○	○	○	○
6	k近傍法 (k-NN)	×	×	△	△	×	△	×
7	線形判別分析 (LDA)	○	△	○	△	○	×	×
8	二次判別分析 (QDA)	○	△	○	△	○	×	×
9	リッジ分類	○	△	○	△	○	×	×
10	ロジスティック回帰	○	△	○	△	○	×	×
11	線形SVM	△	○	△	△	○	×	×
12	決定木	△	△	○	×	○	△	○
13	勾配ブースティング	×	○	△	△	×	○	△
14	ガウシアンナイーブベイズ	○	△	○	×	○	×	×
15	AdaBoost	△	△	△	×	○	×	△

凡例

○ 得意・対応可　△ 条件付き・標準的　× 不得意・要工夫

▸ **順位 = SFIの推奨スコア（精度寄与・速度寄与の総合評価）**

順位はあくまで「データ全体での推奨度合い」を示すもの。**課題によっては7位以下が正解になる場面も多数あります。**
「説明性が必須」「速度最優先」「少データ」など、現場固有の条件次第で正解は変わります。

▸ **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

数万件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。
木系・線形SVMが強く、k-NNやGNBは弱い。

▸ 説明性

「なぜそう判定したか」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

分類精度の総合評価。
あくまで「他観点が同等なら」の比較軸。
推奨上位5モデル（木系）が圧倒的。

▸ 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。
LDA / GNB / 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。
木系アンサンブルが強い。
線形系は弱い。

▸ 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

ロジスティック回帰 / リッジ分類 / LDA

本命モデル

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

特殊用途

k-NN(類似検索) / QDA(クラス分散違い) / 決定木(ルール抽出)

▸ **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）

▶ LDA / ロジスティック回帰 / k-NN
複雑モデルは過学習リスク大。
線形系・k-NNが安定動作。

02 クラスが不均衡（良品99% / 不良品1%）

▶ ランダムフォレスト / XGBoost / LightGBM
クラス重みや評価指標（F1・再現率）の調整で対応。
線形系は要工夫。

03 センサ異常値・外れ値が混じる

▶ ランダムフォレスト / XGBoost
木系アンサンブルは外れ値の影響を受けにくい。
線形系は外れ値除去が前提。

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。
リアルタイム判定も視野。

06 特徴量数 ≫ サンプル数（画像特徴・スペクトル）

▶ 線形SVM / ロジスティック回帰
高次元データではSVMが安定。
木系は過学習リスクが高まる。

07 良品と不良品で「ばらつき具合」が違う

▶ 二次判別分析 (QDA)
クラスごとに異なる分散を仮定。
LDAより柔軟な境界を引ける。

08 工程パラメータが相関している

▶ リッジ分類 / ロジスティック回帰
多重共線性に強い線形系。
係数が暴れず安定した判別線を引ける。

▸ 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 「なぜそう判定したか」を現場説明する必要

▶ **決定木 / ロジスティック回帰 / LDA**
決定木は判定ルールを抽出。
ロジスティックは係数で説明。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）

▶ **ロジスティック回帰 / リッジ分類 / LDA**
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 「不良である確率値」を出したい

▶ **ロジスティック回帰 / LDA / QDA**
確率値が出るのでアラート閾値の現場調整が可能。
木系も別途確率出力可。

13 高精度が最優先（最終判定モデル）

▶ **ランダムフォレスト / XGBoost / LightGBM / CatBoost**
必ず推奨上位5モデルを横並び比較。
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 / XGBoost / ロジスティック回帰**
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 過去の類似サンプルから判定したい

▶ **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

▶ **ロジスティック回帰 / LDA / 決定木 / GNB**
係数式・ルール式・確率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（不良品判定）

温度・圧力・速度・保圧・材料・金型・号機・シフト

ベースライン：ロジスティック回帰（係数で要因解釈、確率値出力）
本命：CatBoost / ランダムフォレスト（カテゴリ多数 ＋ 要因可視化）
説明用：決定木で「圧力≧X かつ 温度≦Y なら不良」のルールを抽出

⚡ 溶接（合否・不良モード分類）

電流・電圧・速度・板厚・シールドガス・継手

ベースライン：QDA（クラスごとに分散が違うことが多い）
本命：ランダムフォレスト / XGBoost（非線形 ＋ 多クラス不良モード判別）
類似検索：k-NN（過去の類似条件サンプルから判定）

🛠️ 加工（正常/異常判別・予知保全）

送り速度・主軸回転・工具摩耗・振動センサ

ベースライン：リッジ分類 / LDA
本命：LightGBM（センサ大量データを高速処理 ＋ リアルタイム判定）
高次元データ：線形SVM（振動波形特徴量が数百次元の場合）

🔩 組立（合否判定）

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：ロジスティック回帰で「効く要因」を係数で確認

🔍 外観検査・品質管理

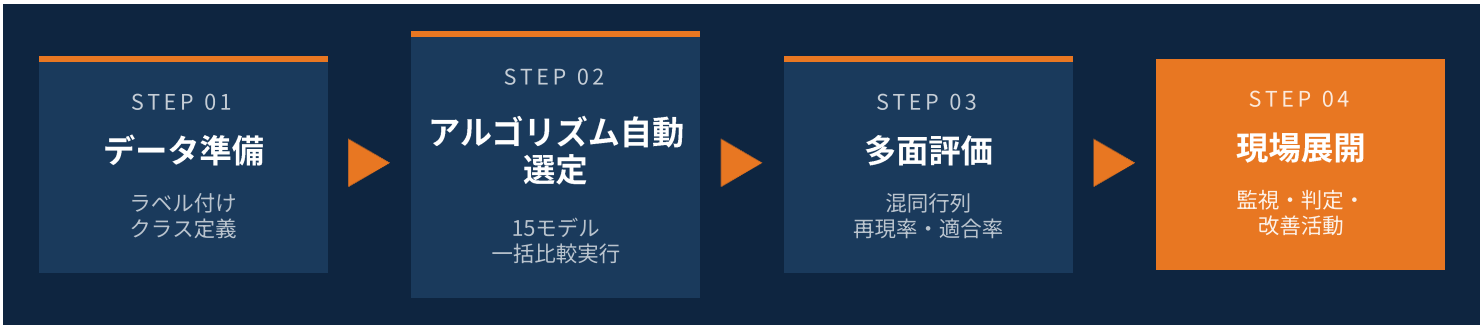
画像特徴量・寸法測定値・色情報・パターン認識

本命：ランダムフォレスト → XGBoost（精度 ＋ 多クラス不良モード分類）
高次元画像特徴：線形SVM

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

◆ **クラス定義の明確化**（何を「不良」と呼ぶか）
◆ ラベル付けの一貫性チェック
◆ クラス不均衡の確認（良品/不良品比率）
◆ 現場知見の「特徴量化」
◆ 異常ラベルの誤り検出

▸ **最重要：**分類で最も品質を決めるのは「ラベルの正しさ」。
誤ラベルが10%あれば、どんなモデルでも精度は90%が上限です。

STEP 02

アルゴリズム自動選定

◆ **15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
◆ 推奨スコアで上位3〜5モデルを自動抽出
◆ ベースライン（ロジスティック）との差を可視化
◆ 速度・メモリも合わせて比較

▸ **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

多面評価（精度数値以外も）

◆ **混同行列**（取りこぼし vs 過検知）
◆ 再現率（不良の見落とし率）
◆ 適合率（過検知による生産性低下）
◆ 説明性・速度・現場感覚との一致
◆ コスト見合いの判定閾値調整

▸ **注意：**「正答率99%」だけで選ぶと、
製造業では致命的な見落としが起きます。次ページ参照。

STEP 04

現場展開

◆ 監視ダッシュボードへの組込み
◆ 判定閾値設定（現場合意のうえで）
◆ 現場説明・OJT
◆ **判定ハズレ事例の収集**
◆ 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、15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

ロジスティック回帰で十分なケースが、15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、ロジスティック回帰の説明性を「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 → ロジスティック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 精度99%だけで選ぶ → 実は不良の取りこぼしが致命的な数値だった

▸ 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)

分類分析でよくある誤解

「正答率99%」が現場を破滅させる場面がある

✕ 「正答率99%＝良いモデル」

▸ **なぜ問題か：**
不良率1%の工程で「全部良品と判定するモデル」を作れば、正答率は自動的に99%になります。
しかし、それは**不良を1個も検出していない**役立たずのモデルです。
▸ **本質：**
分類では**混同行列・再現率・適合率・F1スコア**を必ず併せて見ます。
「正答率」だけで判断するのは、製造業では致命的な間違いです。

✕ 「クラス不均衡はそのまま学習すればよい」

▸ **なぜ問題か：**
良品99% / 不良品1%のデータをそのまま学習させると、ほとんどのアルゴリズムは「全部良品」と判定する方向に最適化されます。
不良の見落としが最大化される、最悪の挙動です。
▸ **本質：**
クラス重み調整、SMOTE等のオーバーサンプリング、F1スコア最大化など、
不均衡データ専用の対策を打ったうえで学習させる必要があります。

✕ 「最新のAIモデル＝最良」

▸ **なぜ問題か：**
製造業の分類課題は、ロジスティック回帰やLDAで十分なケースが少なくありません。
本資料の推奨スコアでも、線形系は上位5位までの精度には及ばないものの、
説明性・速度・実装容易性で圧倒します。
▸ **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。

✕ 「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が不良と判定したから廃棄しました」では、顧客監査・行政対応で破綻します。
▸ **本質：**
説明性の高いモデル（ロジスティック回帰・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：判定閾値の問題／時系列特有の落とし穴／因果と相関の取り違え

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

「判定閾値はデフォルト（0.5）でよい」

×

▸ **なぜ問題か：**
「不良確率0.5以上を不良と判定」はデフォルト値であって、現場で正しいとは限りません。
不良の見落としコストが大きいなら閾値を下げる、過検知コストが大きいなら閾値を上げる、
という**現場経済合理性に基づいた調整**が必要です。
▸ **本質：**
ROC曲線・PR曲線を見ながら、現場と合意のうえで閾値を決めるのが正しいプロセスです。

「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、
不良モードの新規発生などで、判定境界は常に動きます。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、
混同行列の継続モニタリングを運用設計に組み込むことが必須です。

「ラベル付けは適当でも大量にあればOK」

×

▸ **なぜ問題か：**
「良品/不良品」のラベルが現場担当者間でブレていると、モデルは「ブレた判定基準」を
忠実に学んでしまい、いくらアルゴリズムを変えても精度は頭打ちです。
▸ **本質：**
分類精度の上限はラベル品質で決まります。
少量でも**「ラベル付けマニュアル」と「不一致時の合議」**がある方が、大量の雑なラベルより上です。

「不良と判定された＝原因が分かった」

×

▸ **なぜ問題か：**
分類モデルは「不良である」を当てる仕組みであって、
「なぜ不良なのか」を答える仕組みではありません。
特徴量重要度や係数を「原因」と取り違えると、改善打ち手を間違えます。
▸ **本質：**
分類モデルが示すのは**「相関」であって「因果」ではない。**
現場知見との突き合わせと、実験計画（DOE）による因果検証が不可欠です。

▸ **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。

15種類の分類アルゴリズムは、どれも「正解」ではなく「特性の違う道具」です。
現場の判定課題・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 正答率だけで選ばない

混同行列・再現率・適合率・F1。
クラス不均衡時は特に、複数指標で判断します。

② 迷ったら全部試して比較

SFIは「15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

分類モデルは「相関」であって「因果」ではない。
改善打ち手は現場知見との突合せが必須。

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 分類分析アルゴリズム解説資料
Vertys Inc. | 2026