SFIのMOCで学ぶ
機械学習の全体像

製造業の現場で実際に使える **AI・機械学習プラットフォーム** の仕組みを、画面の流れに沿って解説する実践型ガイドです。 難解な数式やプログラミングコードを排し、各画面に配置された機能・設定値が**どう連動するか**を直感的に理解することを目指します。

SEMINAR PURPOSE ／ 本セミナーの目的

「動かせるAI」を、現場の言葉で理解する

機械学習の難解な専門用語を排し、**データ定義から条件探索まで**の一連のシステムフローを直感的に押さえます。アプリ内の各画面の役割と、設定値の流れ方を整理しながら、今日明日からの現場改善に活かせる「使えるAI」の感覚を身につけていきましょう。

本日の学習内容 ／ AGENDA

1. **アプリの全体像とシステムフローの把握**
   Dataset → Training → Results → Predict → Process Optimization の **5ステップ** がどう繋がるかを俯瞰します。
2. **機械学習の基本概念をやさしく整理**
   データ準備の重要性、特徴量と目的変数、Train/Test 分割、評価指標の選び方を **製造業の具体例** で押さえます。
3. **ハンズオン：スクリーン印刷機の品質判定**
   約 2,800 件の実機データを使って、CSV取込から **条件最適化** までを一気通貫で体験します。
4. **自社データで応用するための次のアクション**
   明日から始める **3つの実践ステップ** を整理し、「使われるAI」を作るための最短ルートを描きます。

対象：機械学習に触れたことのない現場担当者・初学者の方。途中での質問はいつでも歓迎します。

CHAPTER

01

アプリの全体像と考え方

このアプリケーションが「何を解いて」「どう設計されているか」を、画面の流れと設定値の連動の観点から俯瞰します。機械学習の難解な専門用語は使わず、まずは"道具としての姿"を掴むことを目的とします。

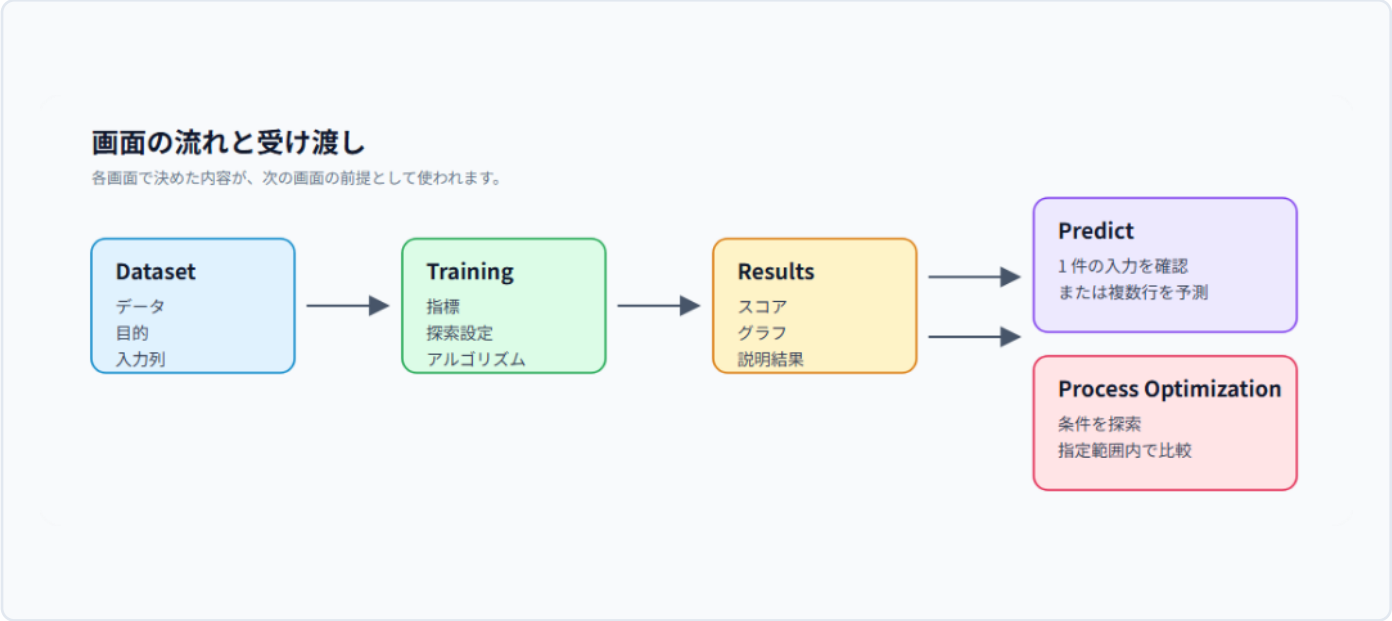
この章で扱う主なトピック

- システム全体の **5ステップ** フロー（Dataset → Training → Results → Predict → Optimization）
- このアプリ最大の特徴 — 設定値の **自動バインド** の仕組み
- **Dataset画面** の構成要素 ／ **Task Type** の種類と選び方
- 各画面の役割と、設定変更時の **ドミノ影響** の整理

02. アプリ全体のシステムフロー

このアプリケーションは、左側のナビゲーションに沿って **上から下へ、1方向に進む** よう設計されています。
各画面はバラバラに存在しているのではなく、データが順番に「定義 → 学習 → 評価 → 予測 → 最適化」と流れていく **一気通貫の設計思想** が特徴です。

> 💡 **初学者のポイント：**機械学習は「データの旅」です。この旅路を最初にイメージしておくと、後続のすべての画面操作がぐっと理解しやすくなります。



5つのステップ — それぞれの役割

STEP 1 **Dataset — データ取込**

CSVをアップロードし、**予測したい列（目的変数）** と **使用する列（特徴量）** を定義。タスクタイプ（分類・回帰など）もここで決定します。

STEP 2 **Training — 学習**

同じ前提条件の下で、**複数のアルゴリズムを並列に競争**させ、最も精度の高いモデルを自動探索します。

STEP 3 **Results — 評価**

出来上がったモデルの予測スコア、誤差傾向、各特徴量の **寄与度（SHAP重要度）** を可視化し、現場の妥当性を確認します。

STEP 4 **Predict — 予測実行**

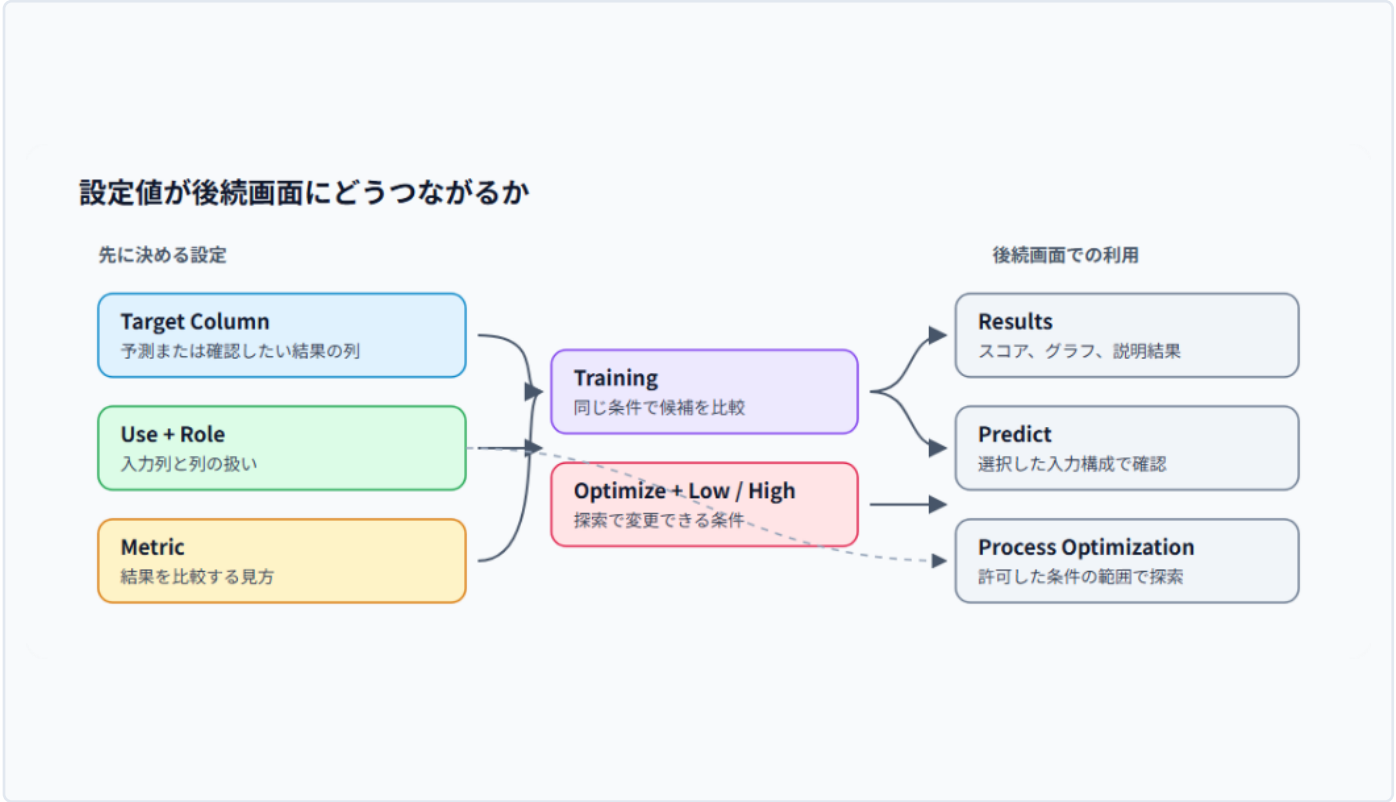
新しいデータに対してその場で **What-if 予測**。1件入力での感度分析と、CSVバッチでの一括処理を切り替えられます。

STEP 5 **Process Optimization — 条件の最適化（逆算）**

「**目標値（不良率最小化など）に近づくための入力条件はどれか？**」を逆算・自動探索。試作回数の削減や条件出しの効率化に直結する、本アプリ最大の差別化機能です。

03.このアプリの最大の特徴 — 設定値の自動バインド

Dataset で決めた前提が、**後続のすべての画面に自動で引き継がれる**。これが本アプリの設計思想です。



従来システムとの違い — Before / After

⚠ BEFORE ／ 従来のシステム

画面ごとに再設定が必要

同じ列を何度も指定し直し、設定ミスや前提のズレが頻発。
担当者の経験に依存し、再現性が低い。

✓ AFTER ／ このアプリ

一度決めれば自動で連動

Dataset の設定が後続全画面へ伝播。
前提のズレが起こらず、初学者でも安心して操作できる。

運用の鉄則 — 3つのポイント

RULE 01

Dataset が起点

後続画面の **Target / Use / Role** はすべて Dataset の指定をそのまま継承。

RULE 02

結果がおかしい時は戻る

異常な予測結果や重要度を見たら、
必ず Dataset に戻って前提を確認する。

RULE 03

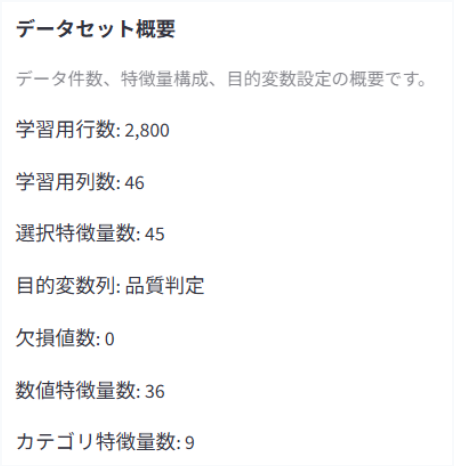
同一前提で全工程

Training・Predict・Optimization まで
同じデータ前提 で一貫処理されるため再現性が高い。

04.Dataset画面の全体像と画面構成

分析の出発点となるDataset画面。**3つの領域**でデータの取り込みと意味付けを行います。

CSVアップロード ／ ファイル選択とタスク種類の指定



データセット概要（行数・列・欠損）

目的変数プレビュー（クラス分布）

画面構成3領域の役割

1. **CSVアップロード ／ Task Type の選択**
   学習用 CSV を取り込み、**分類・回帰・異常検知**のいずれを行うかを最初に確定。以降の画面表示はこの選択で変化。
2. **データセット概要 ／ 行数・列・欠損値の自動チェック**
   取り込んだデータの**行数 × 列数**、欠損値、データ型をひと目で把握。データ準備の良し悪しがここで分かる。
3. **目的変数プレビュー ／ Target Column のクラス分布**
   選んだ目的変数の値の分布を確認。**クラスの偏り**が見えるので、Metric 選びの判断材料になる。

05. Task Type (予測タスク) の種類と選び方

解きたい課題に応じて最適なタスクタイプを選択します。
ここでの選択がTrainingやResultsの表示を切り替えます。

分類 (Classification)

教師あり学習

データの特徴をもとに、カテゴリ、ラベル、Yes/Noなどの「離散的なグループ」を予測します。

【製造業の具体例】 製品の外観画像やセンサー値から、良品・不良品を2択で自動的に判定する。

回帰 (Regression)

教師あり学習

過去データからパターンの傾向を学習し、売上や温度などの「連続した数値」を予測します。

【製造業の具体例】 気温や湿度、製造スピードから、最終製品の強度や硬度数値を予測する。

異常検知 (Anomaly Detection)

半教師あり / なし

「通常の正常な状態」のパターンをモデルに学習させ、そこから大きく外れた特異な値を検出します。

【製造業の具体例】 稼働中のモーター振動センサーから、故障前の「いつもと違う妙な動き」を察知する。

クラスタリング (Clustering)

教師なし学習

事前定義された正解ラベルを使わず、データ構造の類似性だけを分析し、似た者同士を自動でグループ化します。

【製造業の具体例】 設備エラーログを分析し、類似するトラブル傾向のグループに分類・要因分析する。

06. Training（学習） と Results（結果） の役割

モデル作成のための「複数条件比較（Training）」と、
構築されたAIの性能を可視化する「多角的確認（Results）」を解説します。

Training 同じデータ前提で、複数のAI候補を自動比較

試行回数(Trials)やモデル種類(Algorithms)を指定して、Optunaなどの効率的なアルゴリズムで最も優秀な組み合わせを自動探索します。

主要項目	説明（初学者向けの解釈）
Metric（評価指標）	「何を優先して賢さを判定するか」の評価基準（精度、誤差の小ささなど）。
Cross Validation（交差検証）	データの一部を「確認用」に隠しておき、未知のデータでも同じように予測できるか検証する設定。
Algorithms（アルゴリズム）	比較対象とするモデルの種類（決定木、ニューラルネットワーク、線形モデルなど）。

Results 完成したAIの「健康診断」と「影響要因」の見える化

作成されたモデルを評価します。1つの評価数値だけで判断せず、複数のグラフを組み合わせて予測の偏りを確認します。

可視化・指標	確認できること
Global Metrics（全体評価）	全体での平均的な「スコア（正解率や誤差の平均）」。 大まかな仕上がり確認。
Global Importance（需要度）	AIが予測を決定する上で、どの入力項目（Useに指定した列）を重視したか。
Local Explanation	特定のデータ「1行」を予測した時、どのプラス要因・マイナス要因が働いたか。

07. Predict （予測）と Process Optimization（プロセス最適化）

構築したAIを用いた「予測の実行（Predict）」と、
目標を達成する条件を逆算する「条件探索（Process Optimization）」です。

Predict 具体的な数値を入力して「AIの応答」を確認

出来上がったAIモデルに対して、手動入力または一括CSVデータをインポートすることで瞬時に結果予測が得られます。

項目・手法	説明（初学者向けの解釈）
Input Features	画面上のテキストボックスで、 1行ずつのパラメータを手動でシミュレーション入力します。
Batch CSV	数千〜数万件のデータを一気に処理したいとき、 CSVをインポートしてまとめて予測値を出力します。

Process Optimization 目標データに近づくための「入力条件」を自動逆算

「この結果（強度など）を得るにはどの設備パラメータを設定すべきか」をシミュレーションエンジンが
逆算・自動探索します。

設定項目	何に使うか？
Target Value/Class	「強度を◯◯(数値)にしたい」 「不良率を一番低くしたい」といった最適化のゴール。
Optimize (可変設定)	探索時に「変更（調整）しても良い項目」を指定（固定したい項目は除外）。
Low / High / Choices	設備の許容設定限界など、シミュレーションが探索してよい安全な範囲を指定。

08. 設定変更時の「ドミノ影響」

一度決めた設定値は、後続のすべての画面に影響を及ぼします。その影響範囲を一括で整理します。

最初に決めるパラメータ	後続画面に及ぼす「具体的な影響」
Target Column (予測対象の列)	Training で比較する際のスコアの意味、および Results で表示される予測値や誤差グラフのベースそのものを決定します。
Use (入力有無)	Training で学習する項目となり、Predict の手動入力フォーム、および Process Optimization で「調整できる項目」のベースに引き継がれます。
Metric (比較基準)	Results で複数モデルの中から「最も良いもの（Bestモデル）」を決めて並べる際の見方・ソート順に直接関わります。
Optimize (可変指定) (逆算パラメータ)	Process Optimization で目標数値に近づけるために、シミュレーションエンジンが調整（増減）を行ってよい許容項目となります。

💡 初学者へのアドバイス

「後続画面の数値がおかしい」と感じた場合は、いつでも一番最初の **Dataset** に戻って、**Target Column(目的変数)** や **Use(入力として使うか)** が正しく指定されているか見直してください。
前提が正しく揃っていることが、実用的なAIモデルを作る上での必須条件となります。

CHAPTER

02

機械学習の基本を理解する

アプリを使いこなすには、最低限の機械学習の用語と考え方を押さえる必要があります。
難しい数式は使わず、製造業の具体例で「何を準備し、何を評価するか」を整理します。

この章で扱う主なトピック

- **データ準備** の重要性 — Garbage In, Garbage Out
- **特徴量と目的変数** の区別 ／ **Use / Role** の正しい使い分け
- **Train / Test 分割** と **交差検証 (CV)** — 公平な評価の仕組み（理解の補足）
- **Metric**（評価指標）の選び方 — 分類と回帰で何を使うか

09.データ準備の重要性 — Garbage In, Garbage Out

機械学習は「魔法」ではなく、入力したデータの質がそのまま結果の質に直結します。準備段階の丁寧さが、AIの実用性を決定づけます。

やさしい言い換え：「ゴミを入れたら、ゴミしか出てこない」— 良い予測は良いデータから。

BAD CASE
汚れたデータ
欠損値だらけ／単位がバラバラ／古いデータ／記録ミスを含む

→

RESULT
使えないAI
現場に出すと予測が外れ、誰も使わなくなる

GOOD CASE
整ったデータ
単位統一／欠損補完／期間が揃っている／工程との対応が明確

→

RESULT
信頼できるAI
不良率予測などが現場の判断に活かせる

データ準備で必ず確認したい6つのチェックポイント

① 単位の統一

射出成形の「圧力」がMPaとbarで混在していないか。
単位がズレるとAIは別データと誤認します。

② 期間の揃え

学習データの期間と、現場で予測したい期間がかけ離れていないか。設備更新の前後は要注意。

③ 欠損値の扱い

空欄（NaN）をどう埋めるか。
前後の平均で補うのか、その行を捨てるのかを事前に決めます。

④ 外れ値の確認

明らかな記録ミス（温度1000℃など）が混ざっていないか。
これらはAIを大きく狂わせます。

⑤ ラベルの正しさ

「良品／不良品」の判定基準が、検査担当者によってブレていないか。教師データのブレは致命的。

⑥ データ件数

最低でも数百行、可能なら数千行以上。
特に「不良品」のサンプル数が少なすぎると検知できません。

💡 製造業の現場で多い落とし穴

切削加工の **歩留まり**（良品が取れる割合）を予測したいのに、刃物交換のタイミング情報が入っていない
— こうした「現場では当たり前」の変数が抜け落ちると、AIは原因を学べません。
データ準備の段階で、現場のベテランへのヒアリングが欠かせません。

10.特徴量（Feature）と目的変数（Target）

「何から予測するか（=特徴量）」と「何を予測するか（=目的変数）」を区別することが、機械学習の出発点です。

やさしい言い換え：特徴量＝「ヒントになる材料」／目的変数＝「予測したい答え」

FEATURE（特徴量）

ヒントになる材料

- 射出成形の**金型温度**
- 樹脂の**射出圧力**
- 計量時間 / 保圧時間
- 環境温度・湿度
- ロット番号・材料メーカー

→ 学習

TARGET（目的変数）

予測したい答え

- 製品の**引張強度**（回帰）
- 良品 / 不良品（分類）
- **不良率**（回帰：%）
- 検査工程での合否（分類）

製造業での具体例で押さえる「役割の違い」

業務シーン	Feature（材料）	Target（答え）
射出成形 寸法精度の予測	金型温度／射出速度／保圧時間／材料ロット	製品寸法 (mm) → 回帰タスク
切削加工 工具寿命の予測	回転数／送り速度／加工時間／振動量	工具交換が必要か (Yes/No) → 分類タスク
検査工程 不良判定の自動化	画像から抽出した色・形状特徴／センサー値	良品／不良品／要再検査 → 分類タスク（3クラス）
生産ライン 歩留まり予測	原料配合比／設備稼働時間／環境湿度	その日の不良率 (%) → 回帰タスク

⚠️ **ありがちな失敗：未来の値を特徴量に入れてしまう**

「不良率を予測したい」のに、特徴量に**検査結果**（不良が出た後にしか分からない情報）が混ざっているケース。これでは「答えを見て答えを当てる」状態になり、本番では全く役に立ちません。特徴量は**予測したい時点で必ず手に入る情報**だけを選びましょう。

11.Use / Role の使い分け

各列を「使う／使わない」(Use) と、「数値か / カテゴリか」(Role) を正しく指定することで、
AIの処理方法が大きく変わります。

やさしい言い換え：
Use＝「ヒントとして使うかどうかの○×」／Role＝「その列を “数字” として扱うか “種類名” として扱うか」

Role の2つの型 — ここを間違えると結果が変わります

N Numeric（数値）

大小関係に意味がある列

AIは「**数値の大きさそのもの**」を読み、「上がるほど○○になりやすい」といった連続的な傾向を学習します。

製造業の例 金型温度 (180〜220°C)
射出圧力 (50〜120 MPa)
刃物使用時間 (0〜500時間)
不良率 (0〜10%)

C Category（カテゴリ）

種類・名前を表す列

AIは「**名前の違い**」として扱い、大小ではなく「どのグループに属するか」で学習します。

製造業の例 材料メーカー名（A社／B社／C社）
設備号機（1号機／2号機／3号機）
シフト区分（昼／夜）
製品グレード（標準／高品位）

こんな時に間違えやすい — 紛らわしい列の判断

列の例	正しいRole	理由
号機番号 1, 2, 3...	Category	「1号機より3号機が大きい」わけではない。 数字に見えても “名前” として扱います。
製品ロットNo.	Category	ID番号は大小に意味がなく、グループ識別子。 Numericにすると誤った傾向を学びます。
温度設定値	Numeric	180 < 200 < 220 という大小に物理的意味がある。 連続的に変化する値です。
製品グレード A / B / C	Category	順序があると判断できる場合もあるが、まずはカテゴリとして扱うのが安全です。

💡 迷ったときの判断基準

その列を「**足し算・引き算しても意味があるか？**」を考えてみてください。
意味がある（温度、圧力など）→ Numeric。意味がない（号機番号、メーカー名）→ Category。

12.学習データと評価データ — Train / Test 分割

持っているデータの一部を「答え合わせ用」に隠しておくことで、
AIが **初めて見るデータ** でも正しく動くか確かめます。

やさしい言い換え：Train＝「教科書」、Test＝「テスト問題」。教科書でテスト問題を見せたら意味がない。

手持ちデータ100% の分け方（例：80 / 20）

TRAIN

教科書 — 学習に使う

AIはこのデータを何度も見て、特徴量と目的変数の関係（ルール）を学びます。

TEST

テスト問題 — 評価に使う

AIには絶対に見せず、学習が終わったあとに「初見の問題」として精度を測ります。

なぜ分けるのか — 過学習（オーバーフィッティング）を防ぐため

起きること	初学者向けのやさしい解釈
過学習 Overfitting	教科書を丸暗記しただけのAI。 **教科書(Train)**では満点だが、**テスト(Test)**では全く解けない状態。
汎化性能 Generalization	初見のデータにも対応できる「応用力」。 これを測るためにTestデータが必要。
分割の比率	一般的には **Train 80% / Test 20%** または **70% / 30%**。 データ件数が多いほどTestを増やせます。

製造業の現場での「分け方」の工夫

時系列でデータを分ける

過去6ヶ月をTrain、直近1ヶ月をTestに。
未来を予測する練習になるため、現場運用に近い検証ができます。

不良サンプルを偏らせない

不良品データが少ない場合、ランダム分割すると **Testに不良が1件も入らない** こともあります。
層化分割で偏りを防ぎます。

13.Metric（評価指標）の選び方

「AIの賢さ」を測るものさし。分類と回帰で全く違うものを使い、**目的に合わせて選ぶ**ことが重要。

分類タスク（Classification）の主なMetric

Accuracy
正解率

全体のうち、正しく当てた割合。

製造業での使いどころ 良品と不良品の数が**同じくらい**のとき。バランスが偏ると見かけ上高く出るので要注意。

Precision
適合率

「不良」と判定したうち、本当に不良だった割合。

製造業での使いどころ 誤って良品を「不良」と判定して**捨てたくない**場合に重視。

Recall
再現率

本物の不良のうち、何%を捕まえられたか。

製造業での使いどころ 不良の見逃しが**クレームや事故**につながる業界（医療部品など）で重視。

F1
F1スコア

Precision と Recall のバランスを取った総合点。

製造業での使いどころ 見逃しも誤検出も**両方避けたい**場合のバランス指標。
最も実務向き。

ROC-AUC
AUC値

確信度の順位付けがどれだけ上手いか（0.5〜1.0）。

製造業での使いどころ 不良品の**確信度ランキング**で上位から検査したいとき。

回帰タスク（Regression）の主なMetric

MAE
平均絶対誤差

予測値と実測値のズレの絶対値の平均。

製造業での使いどころ「平均で何mmズレるか」のような**直感的な誤差**を現場に説明したいとき。

RMSE
二乗平均平方根誤差

大きなズレを**強く罰する**誤差指標。

製造業での使いどころ 大外れ（極端な予測ミス）を**許容したくない**場合に有効。

R^2
決定係数

データのばらつきをどれだけ説明できたか（0〜1）。

製造業での使いどころ モデルが**そもそも傾向を捉えているか**の総合チェックに使う。

💡 製造業で迷ったときの選び方

分類タスクなら、まず**F1 Score**をデフォルトに。不良見逃しが致命的な業界では**Recall**を優先。
回帰タスクなら、現場説明には**MAE**（単位がそのまま残るので直感的）が最もおすすめです。

CHAPTER

03

ハンズオン：
スクリーン印刷機の品質判定

ここから実際にアプリを操作します。題材は「スクリーン印刷機の良品 / 不良品判定」。
約2,800件のセンサーログを使って、データ取込から条件最適化までを一気通貫で体験します。

この章で扱う主なトピック（5ステップ）

- **STEP 1 Dataset** — CSV取込、Target / Use / Role の指定、クラス不均衡への気付き
- **STEP 2 Training** — Metric=F1で複数モデルを自動比較 ／ Ensemble の効果
- **STEP 3 Results** — 全体指標・混同行列・ROC・スコア分布・SHAP重要度の読み解き
- **STEP 4 Predict** — 1件入力 / バッチCSV ／ schema_drift の運用ゲート
- **STEP 5 Process Optimization** — 不良率を最小にする設備条件の自動逆算

14.ハンズオン編をはじめます

ここからは実際にアプリを操作しながら、機械学習の一連の流れを体験します。
題材は「スクリーン印刷機の品質判定」です。

題材：スクリーン印刷機の良品 / 不良品判定

印刷機の各種センサー値・設備パラメータ（押し圧、スキージ速度、インク粘度、版間距離など）から、その印刷が **OK（良品）** か **NG（不良品）** かを予測する分類モデルを構築します。
最終的には、不良が出にくい設備条件を逆算する Process Optimization まで体験します。

使用するデータセット（2,800 件）

OK ／ 良品	NG ／ 不良品	TOTAL ／ 合計
2,380	420	2,800
全体の約85% 正常に印刷できた事例	全体の約15% かすれ・滲み等の不良	CSVファイル 1行＝1ショットの印刷

本日の流れ — 5ステップで一気通貫

STEP 01 Dataset データ取込 ▶ STEP 02 Training モデル学習 ▶ STEP 03 Results 結果評価 ▶ STEP 04 Predict 未知データ予測 ▶ STEP 05 Optimization 条件最適化

このハンズオンで身につけること

① CSVを読み込み、分類タスクを定義する

目的変数の選択、特徴量のRole確認、クラス不均衡への気付き方を体験します。

② 複数モデルを比較し最良を選ぶ

Algorithms × Trials の意味を理解し、アンサンブルの効果を実感します。

③ 結果を多面的に読み解く

混同行列・ROC・スコア分布・SHAP重要度を一通り確認します。

④ 予測と条件探索を実行する

Predictで未知ロットを判定し、Optimizationで「NGが減る設備条件」を逆算します。

💡 **進め方のコツ：**各画面で「**何を見るか**」と「**どう判断するか**」をセットで意識してください。後半の Predict / Optimization は前半の設定をすべて受け継ぎます。

STEP 01 ｜ 解説

15.Dataset でデータを読み込む

最初のステップは **Dataset** 画面。CSV をアップロードし、何を予測するか（目的変数）と、
どの列を入力に使うか（特徴量）を決めます。

この画面でやること

CSVファイルを読み込み、**目的変数（Target）**と**特徴量（Use / Role）**を指定する。タスクタイプ（分類）も同時に決定。

押さえるべき4つのポイント

1 CSVのアップロード

2,800行 × 数十列のCSVをそのままアップロード。
1行 = 1ショットの印刷データ。**列名は日本語でもOK**。

2 Task Type の選択

「良品 / 不良品」を当てる ⇒ **分類 (Classification)**。
回帰や異常検知ではない点に注意。

3 Target Column の指定

判定結果の列「**quality_label**」を指定。
OK / NG の2値ラベル。
ここを間違えると以降がすべて崩れる。

4 Use / Role の見直し

設備IDやロットNoは**Category**、温度・圧力・速度などは**Numeric**に。自動判定の結果を必ず確認。

このSTEPで気付きたいこと

⚠️ クラス不均衡に要注意

このデータセットは **OK 2,380件 / NG 420件 ≒ 約85:15** と偏りがあります。
「不良を見つける」のが目的なのに不良が少ないため、**単純な正解率では精度が高く見える罠** が起こります。
STEP3 の Results では、F1スコアやRecall を重視して評価します。

STEP 01 ｜ 実演

16.CSV をアップロードする

「Browse File」ボタンから、印刷機ログのCSV（2,800行）を選択 → 自動でプレビュー表示されます。

Dataset画面／ファイル選択 → プレビュー表示

ここを見る

CHECK 01

ファイル名と行数

「**screen_print_quality.csv**」が選択され、
自動で **2,800 rows × N cols** として認識されているか。

CHECK 02

プレビュー表示

先頭10行ほどがテーブルで表示される。
列名と値が想定通りか文字化けや列ズレがないか確認。

CHECK 03

列の自動判別

各列の型（Numeric / Category）が右側に表示される。
明らかに違う場合はこの段階で修正。

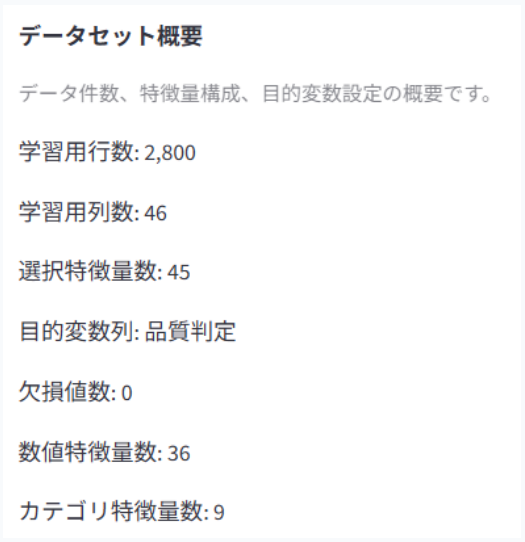
CHECK 04

エンコーディング

日本語列名がある場合、**UTF-8** で保存されたCSVを推奨。
Shift-JIS は文字化けの原因に。

17.データセット概要と目的変数のプレビュー

左：データセット全体のサマリ ／ 右：目的変数 quality_label の分布。**クラス不均衡**がこの時点で見えます。



データセット概要（行数・列数・型サマリ）

Target「quality_label」の分布プレビュー

読み方のポイント

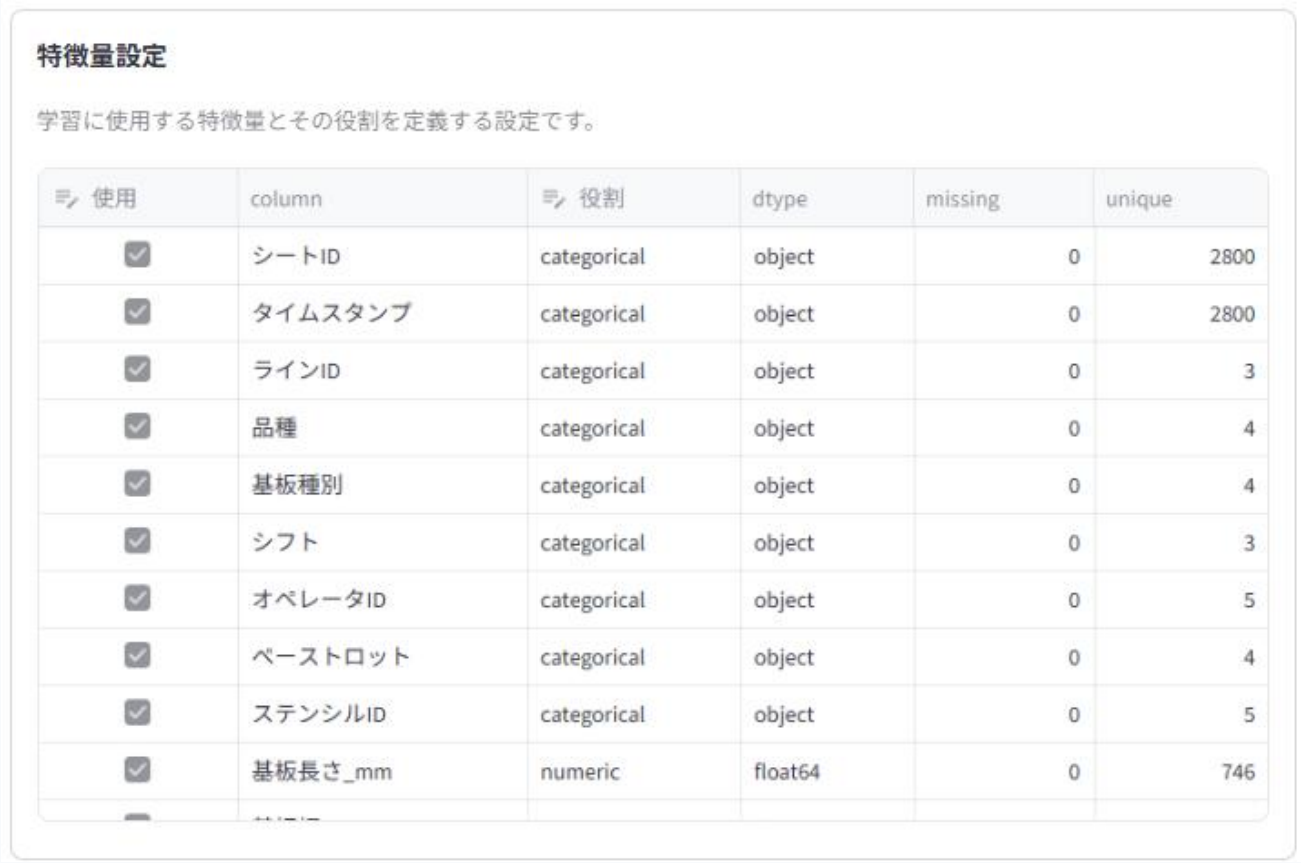
確認項目	この画面で見えるもの・判断すること
行数 / 列数	2,800 行 × 数十列。学習に十分なボリュームがあるか確認。
欠損値の有無	各列の missing 率をチェック。極端に欠損が多い列は要注意。
クラス比率	OK 2,380 ／ NG 420 ＝ 約 **85 : 15**。明らかに偏りあり。

> ⚠ **クラス不均衡に注意**
>
> NG（不良）が **15% しかない** ため、「全件OKと予測する」だけで正解率85%になってしまいます。
> **正解率(Accuracy)を信用してはいけない** のがこのデータの特徴。
> STEP2 では **Metric を F1 や Recall** に設定し、不良の見逃しを減らすモデルを優先します。

STEP 01 ｜ 実演

18.特徴量設定 — Use / Role の最終確認

各列を「学習に使うか（Use）」「数値かカテゴリか（Role）」で指定。最終確認のうえ次のSTEPへ。



特徴量設定

学習に使用する特徴量とその役割を定義する設定です。

使用	column	役割	dtype	missing	unique
☑	シートID	categorical	object	0	2800
☑	タイムスタンプ	categorical	object	0	2800
☑	ラインID	categorical	object	0	3
☑	品種	categorical	object	0	4
☑	基板種別	categorical	object	0	4
☑	シフト	categorical	object	0	3
☑	オペレータID	categorical	object	0	5
☑	ペーストロット	categorical	object	0	4
☑	ステンシルID	categorical	object	0	5
☑	基板長さ_mm	numeric	float64	0	746

特徴量設定 ／ 列ごとに Use（チェック）と Role（Numeric / Category）を指定

ここで判断すること

① 数値特徴量（Numeric）

押し圧、スキージ速度、インク粘度、版間距離、温度、湿度など。
大小に意味がある列は Numeric。

② カテゴリ特徴量（Category）

機台ID、ロットNo、シフト、インク銘柄など。
種類名・ID系は必ず Category に。

③ 除外する列

日時、検査者名、コメントなどは Use のチェックを外す。
リーク（答えが入っている列）にも注意。

④ Target は自動で除外

STEP1 で指定した **quality_label** は自動的に特徴量から除かれる。
同列が二重に使われない。

STEP 02 ｜ 解説

19.Training で複数モデルを自動比較

Datasetで定義した前提を受けて、**複数のアルゴリズム × 複数のハイパーパラメータ** を競争させ、
最良モデルを自動探索します。

この画面でやること

評価指標（**Metric**）と試行回数を決め、「学習開始」ボタンで複数モデルの**自動比較**を回します。

押さえるべき4つのポイント

1 Metric は F1 / Recall

クラス不均衡のため Accuracy はNG。**不良見逃しを減らしたい** なら Recall、バランス重視なら F1。
今回は F1 を選択。

2 Cross Validation = 5

K=5 の交差検証が標準。
データを5分割して順番に評価役を担当
⇒ **偶然の精度ブレを排除**。

3 Algorithms × Trials

5〜7種類のアルゴリズムを各 50〜100 trial で探索。
合計数百回の比較 から最良の組合せを自動選定。

4 アンサンブルの活用

単体の最良モデルだけでなく、複数モデルの予測を**平均する Ensemble** も自動生成。多くの場合さらに精度UP。

このSTEPで気付きたいこと

💡 **学習に時間はかかるが、人手は不要**

2,800行 × 数十列 × 数百試行で **10〜30分** 程度。実行中はバックグラウンドで進行するため、**他の作業と並行可能**。
完了後にRESULT画面で全モデルのスコアを一覧確認できます。

20.学習の設定画面

Metric・CV分割数・試行回数・対象アルゴリズムを指定して「Start Training」。

Training設定画面 ／ Metric・CV・Trials・Algorithmsの指定

推奨設定（このハンズオン）

Metric = F1 Score
不均衡データに強い指標。
Precision と Recall のバランスを取り、OK判定の数で精度が膨らむ罠を回避。

CV = 5-Fold
5分割の交差検証で **平均スコアと分散** を見る。
1回限りの当たり外れに惑わされない。

Trials = 50
各アルゴリズムで50通りのハイパーパラメータを試行。
10〜20分 で実用的な結果に到達。

Algorithms = 推奨セット
LightGBM / XGBoost / Random Forest / Logistic Regression など
5〜7種類。デフォルトで十分。

21.学習ステータスとアンサンブル結果

学習中はリアルタイムで進捗が表示され、完了するとアルゴリズム別スコア＋アンサンブルが並びます。

完了後のアンサンブル結果／単体モデルを上回るスコア

💡 アンサンブルが最良スコアを叩き出す

多くの場合、**Ensemble（複数モデルの平均）**が単体の最良モデルよりも高いF1スコアを記録します。
これは異なる視点のモデルが「間違いを補い合う」ためで、本番運用ではこちらを採用するのが定番です。

STEP 03 ｜ 解説

22.Results の読み解き方

Resultsは **6つのビュー** で構成。1つのスコアだけで判断せず、複数の観点から「このモデルは現場で使えるか」を多角的に検証します。

Resultsの6つのビュー ― 何を見るか

VIEW 01

全体指標
(Global Metrics)

F1・Precision・Recall・Accuracyなどの**全体スコア**を一覧。最初に見る場所。

VIEW 02

混同行列
(Confusion Matrix)

OK/NGを **正しく当てた数・間違えた数** を2×2マトリクスで可視化。

VIEW 03

クラス別指標
+ ROC曲線

OKとNGそれぞれの精度を分解表示。**不良の見逃し率**がここで判明。

VIEW 04

スコア分布
(Score Distribution)

確信度の分布。**判定しきい値**を変えたときの挙動を確認できる。

VIEW 05

全体の重要度
(Global Importance)

SHAP値による**特徴量ランキング**。「何が効いているか」が一目瞭然。

VIEW 06

個別の説明
(Local Explanation)

1行の予測について「**なぜそう判定したか**」をSHAPで分解。

読み解きの順序

① 全体スコアでざっくり感触
② 混同行列で「不良見逃しの数」を確認
③ クラス別指標とROCで深掘り
④ 重要度で「何が効いているか」を把握
⑤ 個別説明で代表ケースを確認
⑥ Predictへ。**1スコアだけで判断しない** のが鉄則。

23.全体指標 — まずは総合スコアを見る

F1・Precision・Recall・Accuracy・ROC-AUC の全体スコアを一覧。最初に見る場所。

Global Metrics ／ 各種スコアの一覧と推移グラフ

読み方のポイント

注目すべきは F1 と Recall

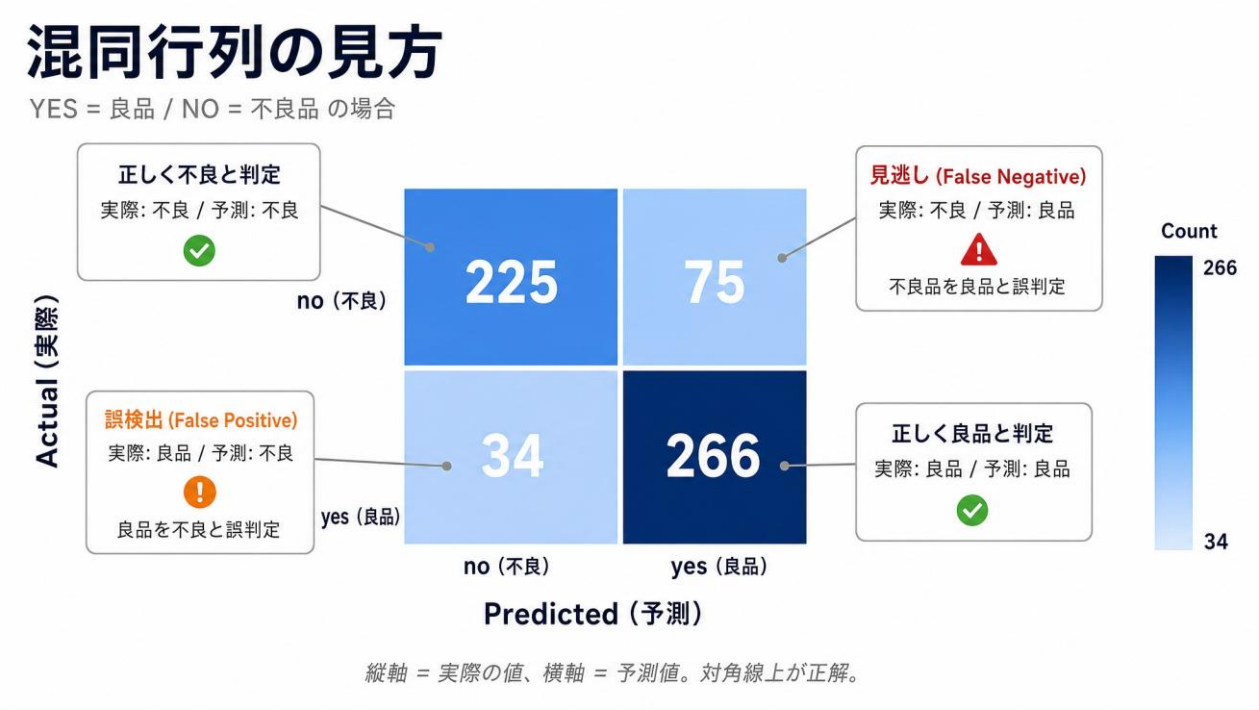
クラス不均衡のため **Accuracy は信用しない**。
F1 が 0.85 を超えれば実用域、Recall も併せて確認。

ROC-AUC ＝ 0.90 以上

確信度の順位付けの精度。
0.90 以上なら検査の優先順位付けに使える品質。

24.混同行列 — 正解と誤りを数で見る

2×2のマトリクスで「正しく当てた / 間違えた」件数を可視化。製造業では特に **FN（不良見逃し）** を最重視。



Confusion Matrix ／ 実測（縦）× 予測（横）の2×2行列

4つのマス目の意味と現場へのインパクト

TP ／ True Positive

NGを正しくNG判定

不良を不良と判定。**これが多いほど良い。**
検査スキップ削減につながる。

TN ／ True Negative

OKを正しくOK判定

良品を良品と判定。
無駄な再検査を減らす 効果。

FP ／ False Positive ⚠

OKをNGと誤判定

過剰検査・廃棄。**コスト増**になるが、見逃しよりは許容できる。

FN ／ False Negative 🚨

NGをOKと見逃し

最も重大な誤り。不良が市場に流出するため、絶対に減らしたい。

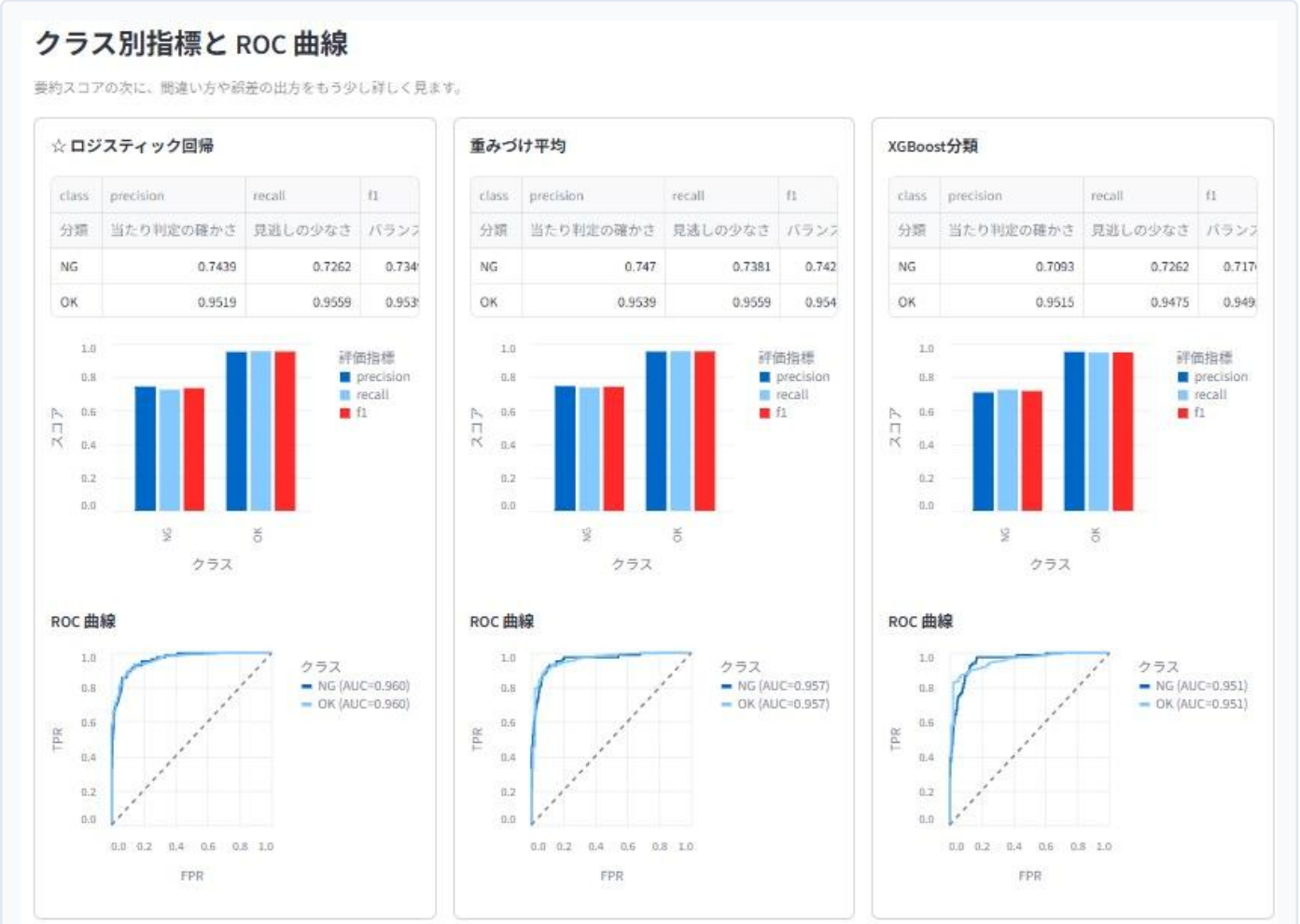
🚨 FN（不良見逃し）の数を最優先で確認

FN＝0 が理想ですが、Recall95% でも 100件中5件は見逃します。**FNが許容範囲か**を、品質保証ポリシーと照らして判断してください。
多すぎる場合は、しきい値を下げてRecallを優先する戦略を検討します。

STEP 03 ｜ 実演

25.クラス別指標と ROC 曲線

OK / NG それぞれの Precision・Recall を分解表示。ROC 曲線で**しきい値の柔軟性**を確認。



クラス別指標表＋ROC 曲線 ／ NG クラスの Recall が特に重要

読み方のポイント

① NG クラスの Recall を見る
「**本物の NG のうち何%を捕まえたか**」。0.85 以上が目標。

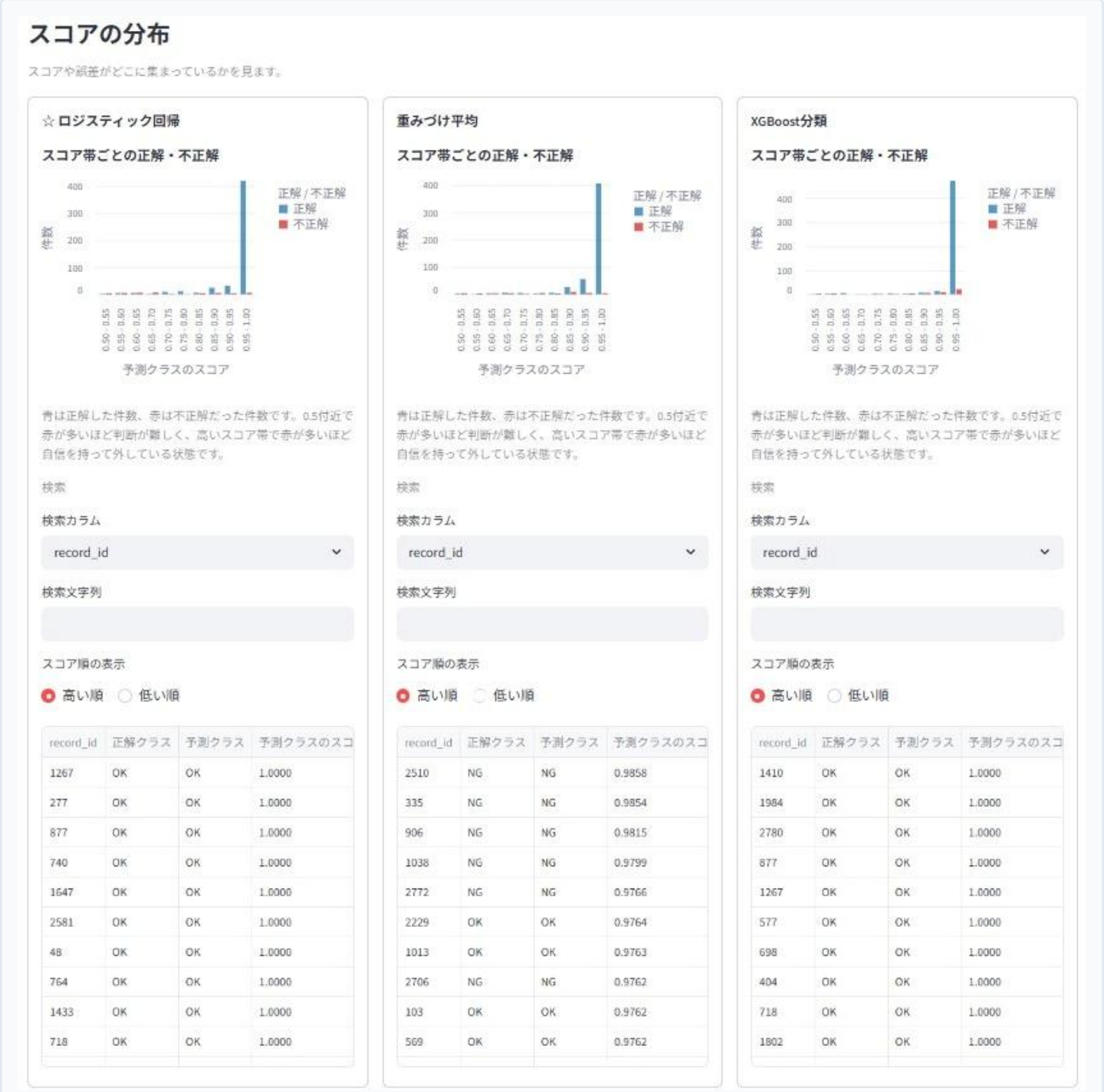
② NG クラスの Precision
「**NG 判定のうち本当に NG だった割合**」。低いと過剰検査が増える。

③ ROC 曲線の形
左上に張り付くほど良い。**AUC 0.90 以上**で実用域。

④ 動作点（しきい値）の選択
曲線上で **FPR と TPR のトレードオフ**を現場ポリシーで決定。

26.スコアの分布 — 確信度の様子を見る

各サンプルの予測スコア（確率）の分布。**2 つの山がくっきり分かれているか**を確認。



スコア分布（OK と NG の山が分離しているほど良いモデル）

何を見るか

山の分離度

OK と NG の山が **はっきり離れているほど良い**。
重なりが大きい＝判別が難しいサンプルが多い。

しきい値の選定

2 つの山の谷間に **しきい値を置く**のが基本。
Recall 重視なら左寄せ、Precision 重視なら右寄せ。

27.全体の重要度 — 何が効いているか（SHAP）

SHAP値で算出した特徴量ランキング。**改善活動の優先順位**がここで決まります。

Global Importance／全データを通した特徴量寄与度ランキング

使い方のポイント

① 上位3〜5項目に注目

上位の **「押し圧」「スキージ速度」「インク粘度」** が品質を支配。改善活動はここに集中する。

② ベテランの感覚と照合

現場ベテランが「これが効く」と言っていた特徴と**ランキングが一致するか**。違う場合は要再検討。

28.1行の予測理由 — なぜそう判定したか

特定の1サンプルについて、各特徴量が予測を「押し上げた / 引き下げた」内訳を分解表示。

Local Explanation ／ 1行の予測SHAP分解

どう読むか

① ベース予測値

全体の平均的な不良率（例：15%）からスタート。
これに各特徴量の影響が**加減算**される。

② プラス要因（赤）

不良率を**上げた**特徴。例：押し圧が異常に高い
⇒ 不良確率を +12% 押し上げ。

③ マイナス要因（青）

不良率を**下げた**特徴。
例：インク粘度が標準値内⇒ 不良確率を −5% 引き下げ。

④ 最終予測値

ベース ± 各要因 ＝ 最終の不良確率。
0.5を超えればNG判定。

💡 現場説明に最強

「このロットがNGになった理由」をベテランに**具体的数値で示せる**。AIへの納得感が一気に高まる場面。

💡 Global と Local を組み合わせる

「全体としては押し圧が効いている」を Global で把握
→「このロットでは特にこう」を Local で説明、
という二段構えが効果的。

STEP 04 ｜ 解説

29.Predict で未知データを判定

学習済みモデルに新しい印刷ロットの設備パラメータを入力 → **OK / NG 即時判定**。
1件入力と一括CSV入力の2モードがあります。

この画面でやること

まずモデルを**確定（Finalize）** → **条件を入力**（手動 or CSV）→ **予測結果を取得**（OK/NGとその確信度）。

押さえるべき4つのポイント

1 モデルの Finalize

Resultsで採用するモデル（多くはEnsemble）を**Finalize**。本番運用モデルとしてロックされる。

2 1件入力 ／ What-if 検証

「押し圧を 5N 上げたら？」など、**条件を変えてシミュレーション**。会議中の検討に最適。

3 CSVバッチで一括判定

当日の全ロットをまとめて評価。
日次品質レポートに組み込み可。

4 運用ゲートの確認

schema_drift（データ仕様の変化）を自動検出。
入力フォーマットが学習時と違うとアラート。

このSTEPの位置付け

📌 **「使えるAI」になる瞬間**

ここまでは過去データの検証。Predictで**未知データに対する判定**ができて初めて、AIは「ツール」として現場で使えるようになります。**運用ゲート（schema_drift警告）**はAIの信頼性を保つ重要な安全装置です。

30.Predict 画面 — Finalize → 入力 → 結果

3 ステップで予測完了。① **モデル確定** → ② **条件入力** → ③ **即時判定**。

採用するモデルを Finalize（本番モデルとして固定）

新ロットの設備パラメータを入力（温度・押し圧・スキージ角度…）

OK / NG 判定 + 確信度（スコア）

結果画面の読み方

予測クラスと**確信度（0〜1）**がワンセットで表示されます。

- 確信度 **0.9 以上**：自信あり、現場でそのまま採用可
- 確信度 **0.6〜0.8**：要注意、二重チェック推奨
- 確信度 **0.5 近辺**：判別困難、人手判断に委ねる

活用シーン例

朝礼確認
今日の条件で **NG確率** を1分確認

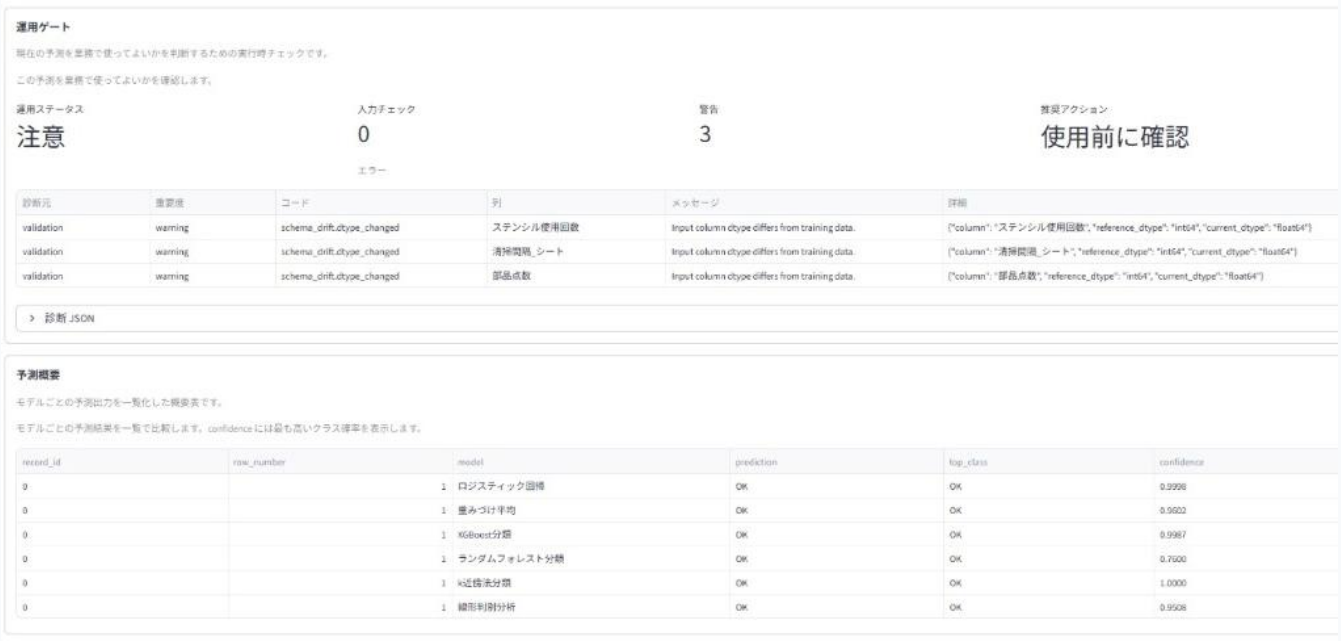
What-if
押し圧 5N 下げたら？を **即試行**

日次レポ
CSV一括 → **不良率予測**

新人教育
条件と結果を **体感学習**

31.運用ゲート — schema_drift 警告

入力データが学習時と異なる場合に自動アラート。**本番運用での品質保証**に欠かせない機能。



運用ゲート画面／入力データの仕様変化を自動検出→アラート表示

何を検出するか

検出項目	何を意味するか
列の不一致	学習時にあった列が入力CSVにない／逆に余分な列がある。**即時エラー**。
値範囲の逸脱	学習時の最大値を大きく超える値（例：押し圧が想定外）⇒ **警告**表示。
未知のカテゴリ	学習時になかった機台ID・インク銘柄など。**予測信頼度が低下**するため要確認。

🚨 **警告が出たら立ち止まる：**「動くから大丈夫」と無視せず、**原因を確認してから再学習を検討**するのが本番運用の鉄則。
データ品質の維持がAI品質を守ります。

STEP 05 ｜ 解説

32.Process Optimization で条件を逆算

「不良率を最小にする**設備条件**はどれか？」を AI に **自動探索**させます。Predictとは逆方向のシミュレーション。

この画面でやること

「目標値（NG率を最小化）」と「動かせる項目・範囲」を指定 → 探索エンジンが **数百〜数千通りの条件を自動試行** → 最適解を提示。

Predict と Process Optimization の違い

PREDICT（順方向）

条件 → 結果

設備パラメータを入力
→「この条件だとNG確率はX%」を予測。**1ケースずつ確認**。

⇄

OPTIMIZATION（逆方向）

目標 → 最適条件

「NG率を最小に」を目標
→ AIが**最適な設備条件を自動逆算**。What-ifを自動化。

設定すべき3つの軸

① 目標 (Target)

「NG率を最小化」「OK確率を90%以上」など、**探索のゴール**を1つ指定。

② 動かす項目 (Optimize)

調整可能な設備パラメータをON。
固定したい項目はOFF。

③ 範囲・候補 (Low/High/Choices)

各項目の**安全範囲**を指定。
広すぎると非現実な解が出る。

このSTEPの価値

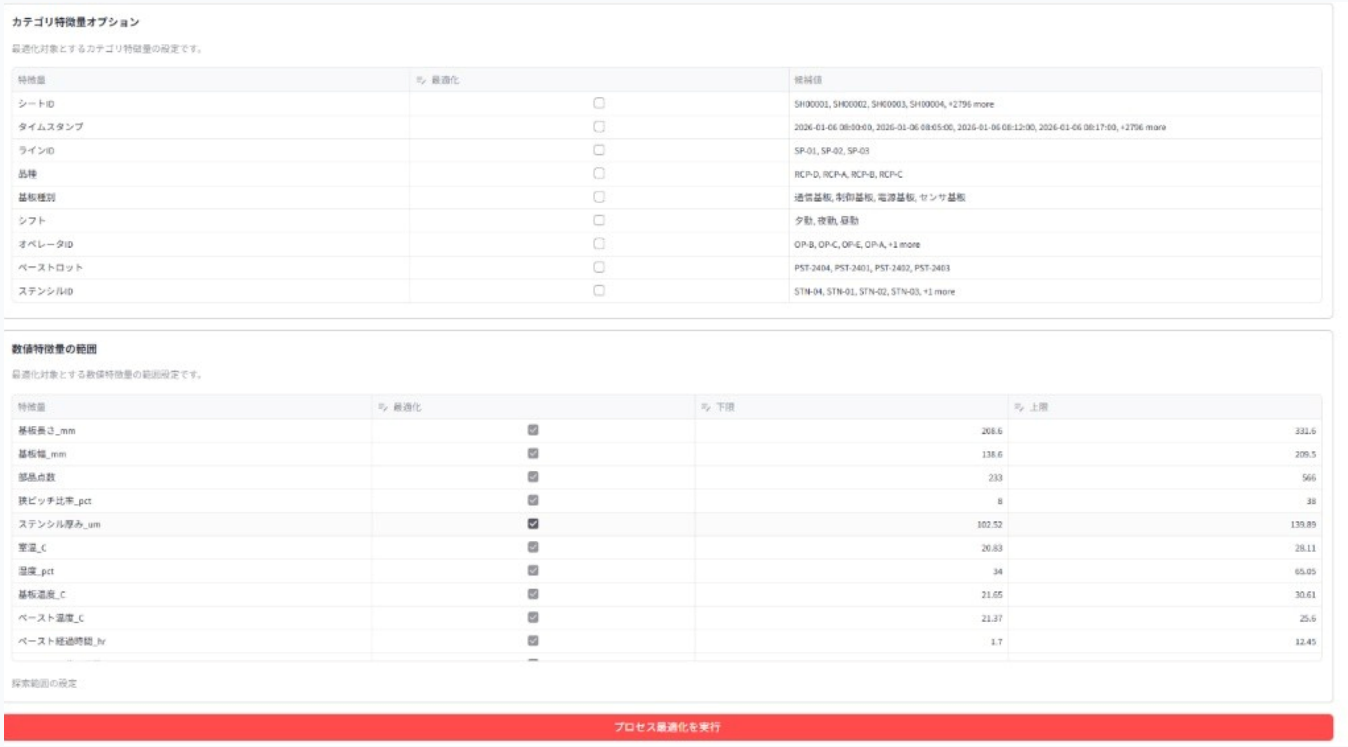
🎯 **「条件出し」を自動化して試作回数を激減**

従来は **ベテランの勘＋何百回の試作** で見つけていた最適条件を、AIが学習済みモデルを使って**数分で逆算**。
試作コストと時間を大幅削減します。
さらに、Edge / Rare Category Penalty を効かせれば「学習データから離れすぎない、現実的な提案」になります。
提案された条件は必ず現場のベテランと妥当性を確認した上で、本番ラインに反映してください。

33.探索条件の設定 — 目標と範囲を指定

上：探索の目的とアルゴリズム／下：各特徴量で「動かす範囲」を1つずつ設定。

探索の目的・試行数・アルゴリズムを設定



各特徴量で「動かしてよい範囲」または「選択肢」を指定

設定のコツ

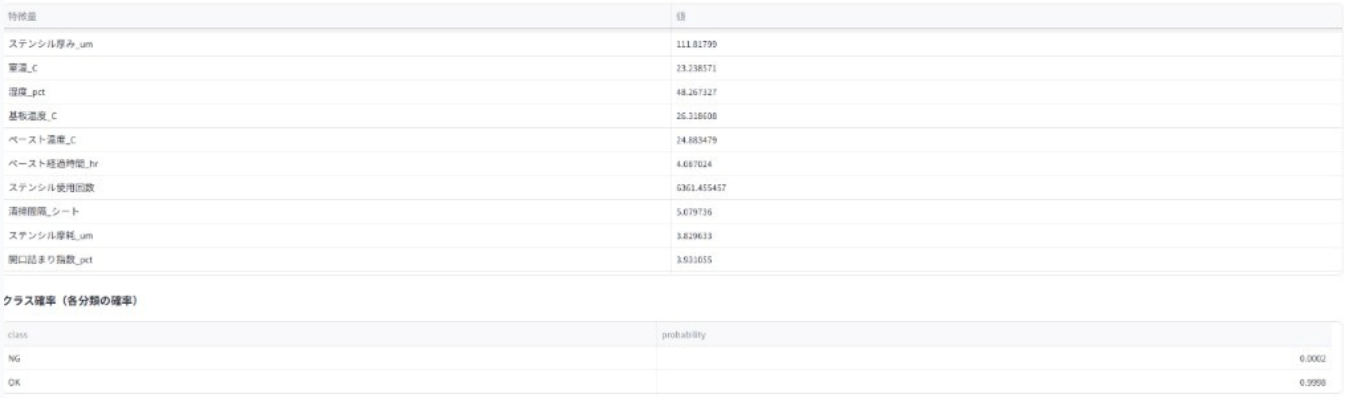
範囲は現実的に：押し圧5〜15Nなど、**設備の物理的許容内**に限定。広すぎると非現実な解。

固定項目を明確に：機台ID・材料銘柄など、**変えられない項目はOFF**。実運用に合った提案に。

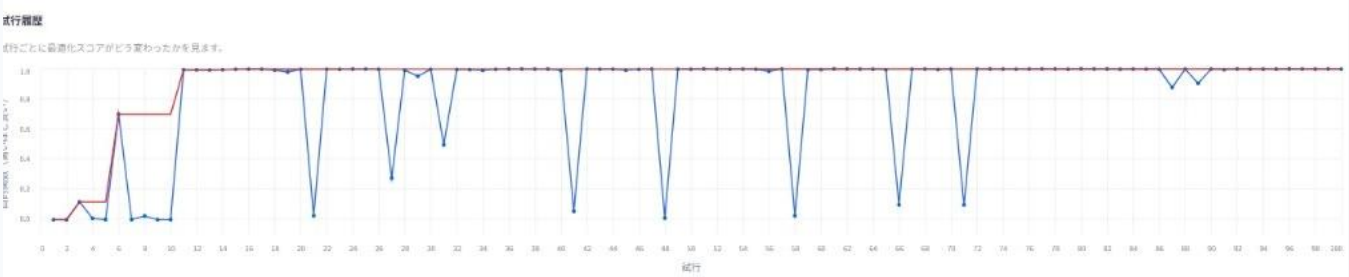
34.最適化結果 — 推奨条件と試行履歴

探索完了後の3つの結果ビュー。**サマリ → 推奨条件詳細 → 試行履歴** の順で読み解く。

最適化サマリ／Before-After のスコア比較



推奨される設備条件の詳細／各特徴量の最適値



試行履歴／どの条件をどの順序で試したか

結果の活かし方

提案された推奨条件をいきなり本番ラインに適用せず、まず **小ロット試作で実測値を確認**。
AIの予測精度と実機の挙動が一致するかを検証してから、本格採用へ。**ベテランとの妥当性レビュー**も必須です。

CHAPTER

04

まとめと次のアクション

ハンズオンで体験した5ステップを振り返り、自社データで実践するための具体的なアクションを整理します。「使えるAI」を作るための最短ルートを描きます。

この章で扱う主なトピック

本日体験した **5ステップ** の振り返り（Dataset → Optimization）

自社データで応用するために **明日から始める3つのアクション**

「使われるAI」を作るための **現場との関わり方**

セミナークロージングメッセージ

35.ハンズオン振り返り & 次のアクション

スクリーン印刷機の品質判定を題材に、Dataset から Process Optimization まで一気通貫で体験しました。

本日体験した5ステップの振り返り

STEP 01 Dataset ／ データ取込

CSVをアップロード → **Task Type = 分類**、**Target = quality_label** を指定
→ 各列のUse/Roleを確認。クラス不均衡（OK 85% / NG 15%）に気付いた。

STEP 02 Training ／ モデル学習

Metric = **F1**、CV = 5、Trials = 50で複数モデルを自動比較 → **Ensemble**が単体最良を上回るスコアを記録。

STEP 03 Results ／ 結果評価

6つのビュー（全体指標・混同行列・クラス別・分布・Global / Local Importance）で多角的に検証。
特に **FN（不良見逃し）** の数とSHAP重要度を重視。

STEP 04 Predict ／ 未知データ判定

モデルをFinalize → 1件入力でWhat-if検証、CSVバッチで日次判定。**schema_drift警告**で運用安全性を確保。

STEP 05 Process Optimization ／ 条件逆算

「NG率最小化」を目標に、押し圧・スキージ速度・粘度などの最適範囲を自動探索 → **推奨条件**を取得。試作回数の削減につながる。

自社データで応用するために

🎯 明日から始める3つのアクション

① **データを集める：**自社設備のセンサー値・品質ログを **1,000行以上** 用意。可能なら不良サンプルも **100件以上** 確保。

② **小さく試す：**まずは **1ライン・1製品** に絞って本ハンズオンと同じ手順を再現。「動くAI」を体感する。

③ **現場と一緒に評価：**SHAPの重要度ランキングをベテランに見せ、感覚と合うかをディスカッション。AIへの納得感を醸成。

THANK YOU

AIは「魔法」ではなく、現場とデータをつなぐ「道具」です。

本ハンズオンの5ステップが、皆さまの工場で「使われるAI」を作る最短ルートになることを願っています。


PieceMaker
PRODUCT by Vertys