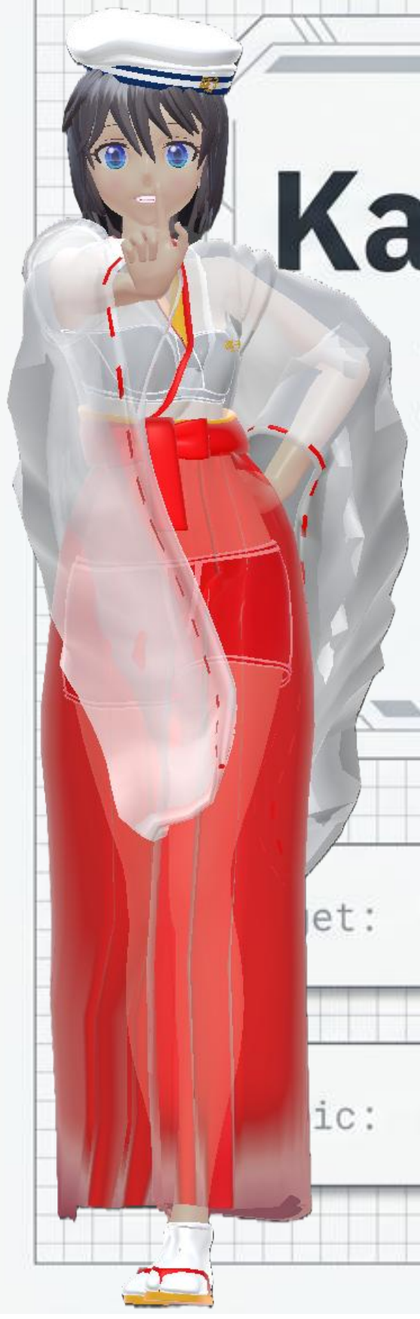


Kaggle S4E8 毒キノコ二値分類：：Top 15%到達への戦略解剖

外部データの統合からMCC閾値最適化、OOFアンサンブルまでの完全軌跡

et: Binary Prediction of Poisonous Mushrooms

Final Score: Private LB 0.98497 / Public 0.98515

ic: Matthews Correlation Coefficient (MCC)

Position: Top 15% (Top 1%まで残り+0.00007)

The Alpha Matrix：勝利を決定づけたコア戦術

S-Tier （最重要・ブレイクスルー）

UCI Secondary Datasetの結合

カラム完全一致の外部データを学習へ直接投入。

MCC特化型 閾値ベクトル化最適化

0.5固定を廃し、累積和を用いたO(n log n)での全探索。

OOFベース・アンサンブル

交差検証の予測値に基づく重み付き線形ブレンド。

A-Tier （高インパクト）

AutoGluon (v2)の導入

AutoMLによる予測多様性の確保（3時間ラン）。

ノイズの構造化

稀少カテゴリを__noise__へ、欠損を__missing__へ統合。

B-Tier （補助的改善）

TabNetの追加

非GBM系アーキテクチャ（NN）の注入。

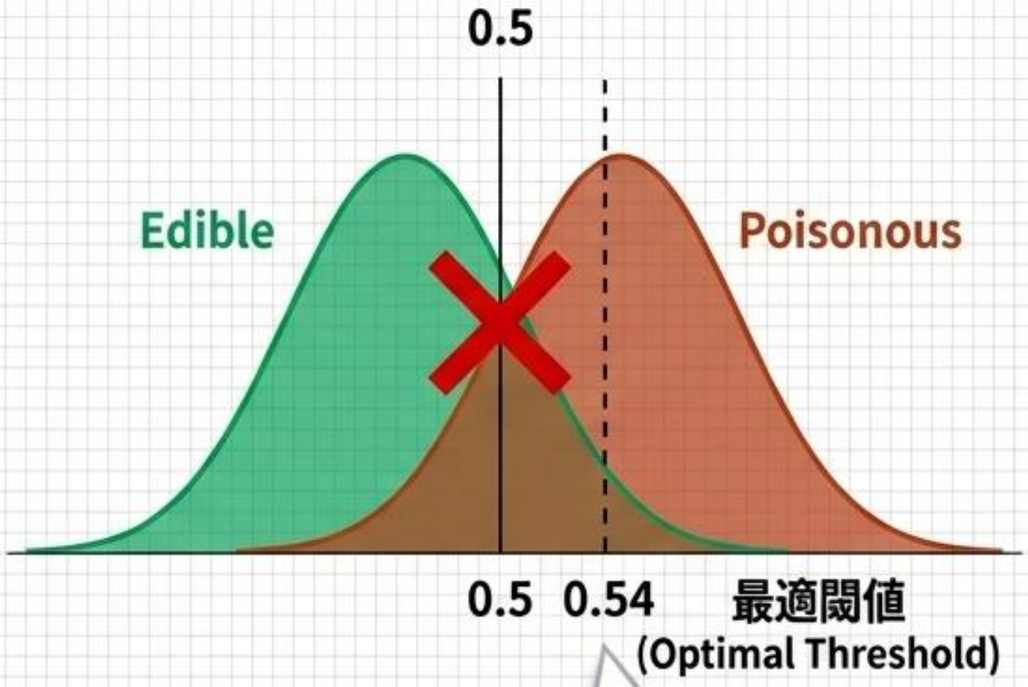
Primary Dataの活用

173種のルール表に基づくベイジアン特徴量の生成。

評価指標の罠：なぜ「閾値0.5固定」が命取りになるのか

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

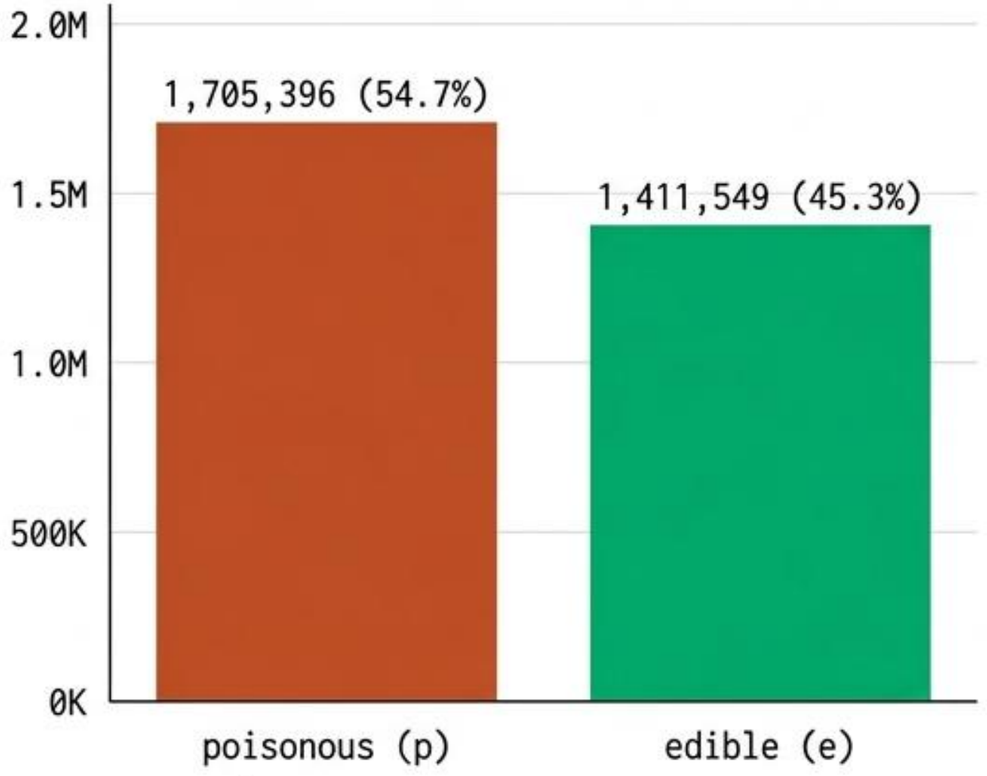
クラス不均衡に頑健（値域 -1 ～ +1）。
しかし「確率」ではなく「二値（0/1）」に対して計算される。



MCC最大化ポイントは分布によって変動する。
固定値0.5では真の性能を引き出せない。

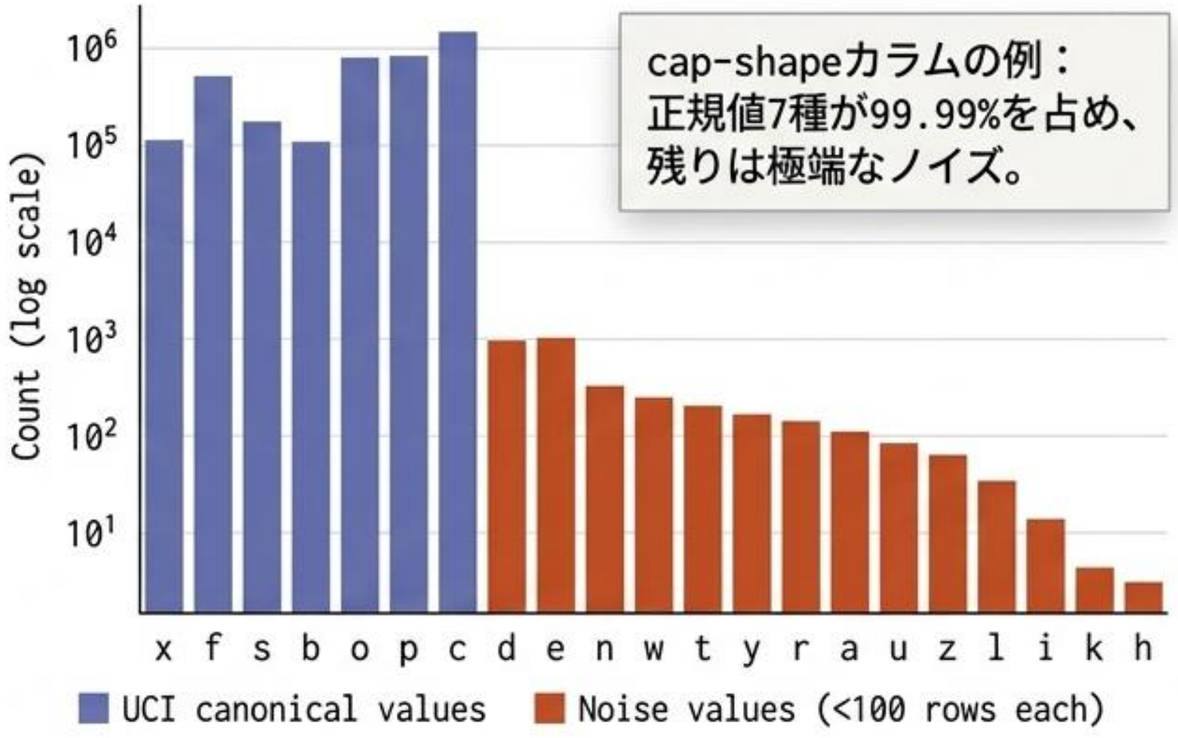
データランドスケープ：均衡なクラスと、隠されたノイズ

train class distribution (nearly balanced)



インサイト: Poisonous (54.7%) / Edible (45.3%)。クラス分布はほぼ均衡（対策不要）。

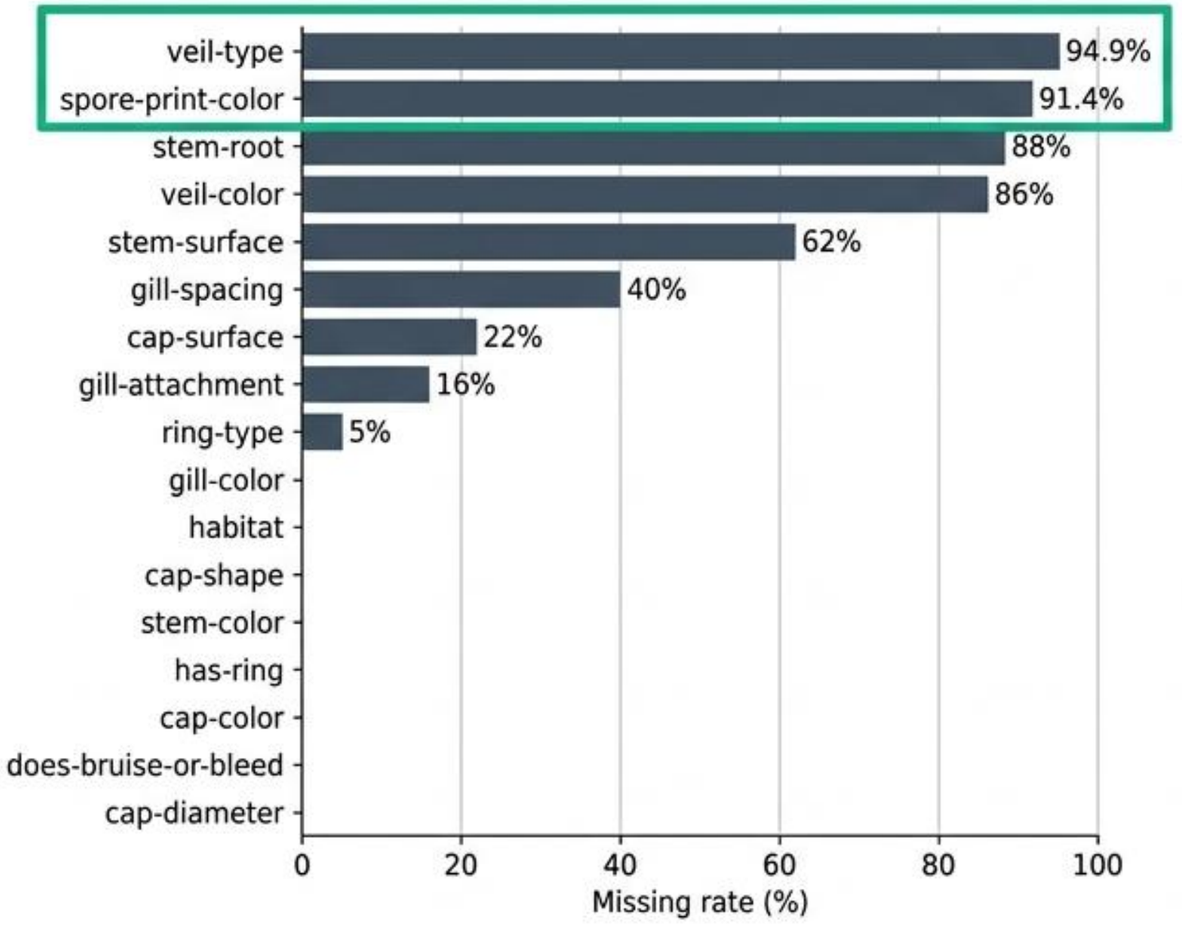
cap-shape values: 7 canonical values vs top 15 noise values (+52 more, ~1 row each)



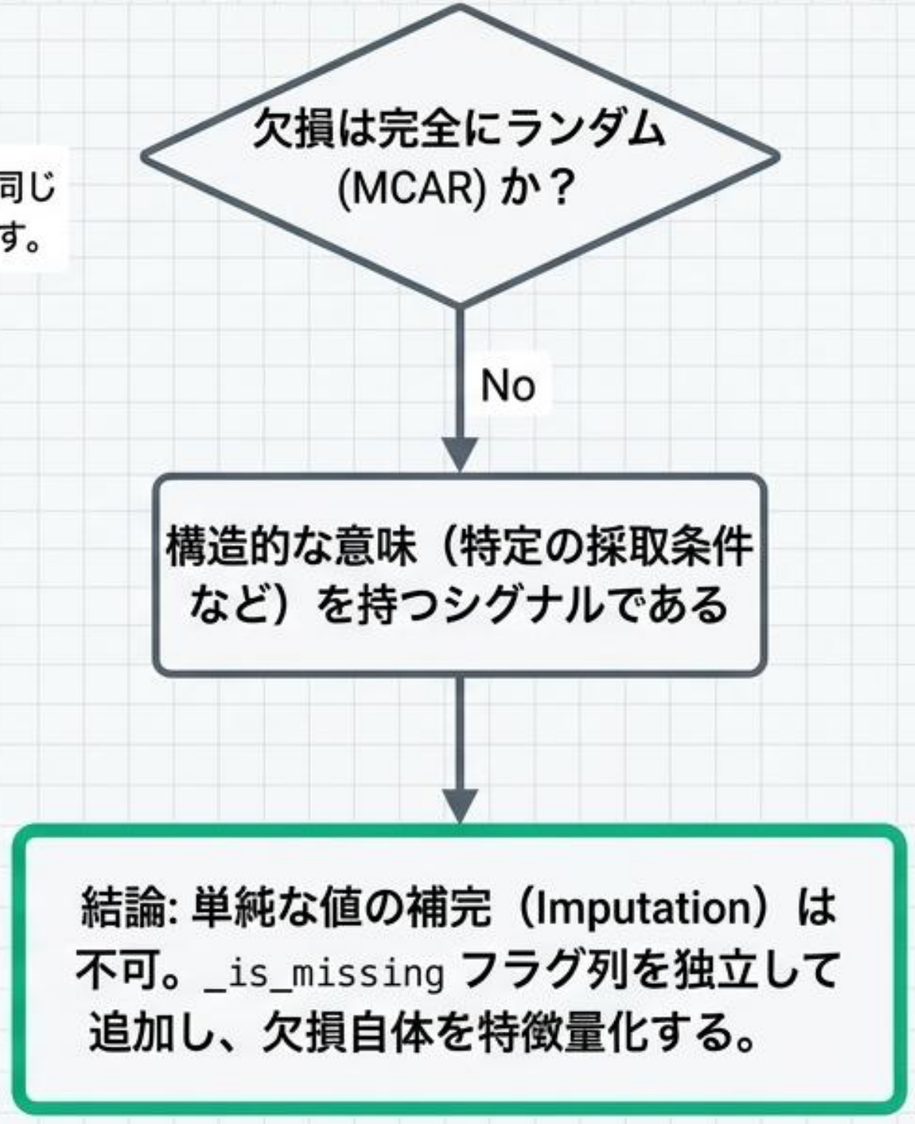
cap-shapeカラムの例：
正規値7種が99.99%を占め、
残りは極端なノイズ。

アクション: 正規値以外はすべて __noise__ として統合するクリーニングが必須。

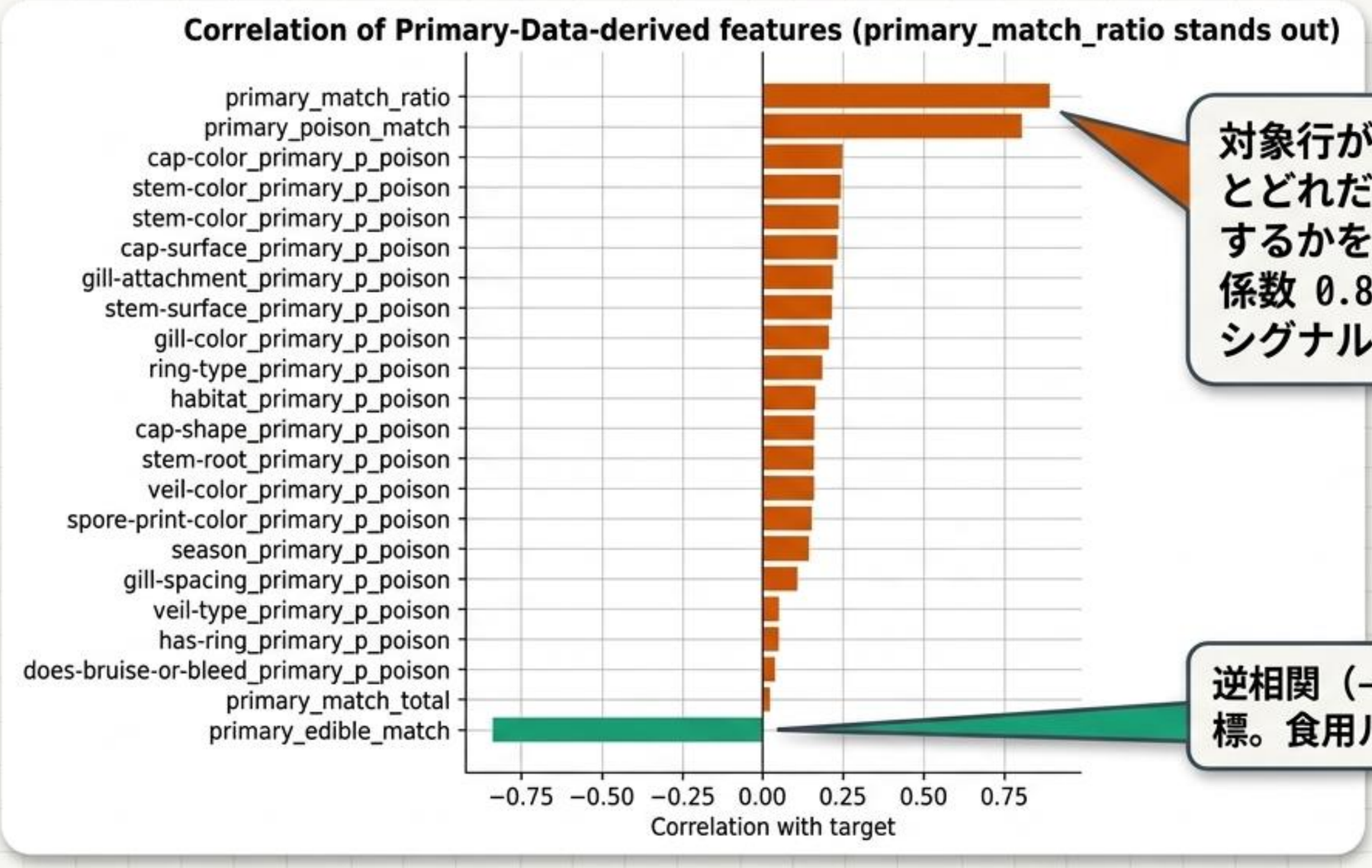
「欠損」という名の構造的シグナル



trainとUCIで全く同じ欠損パターンを示す。



ゲームチェンジャー：Primary Dataによる「毒性」の数値化



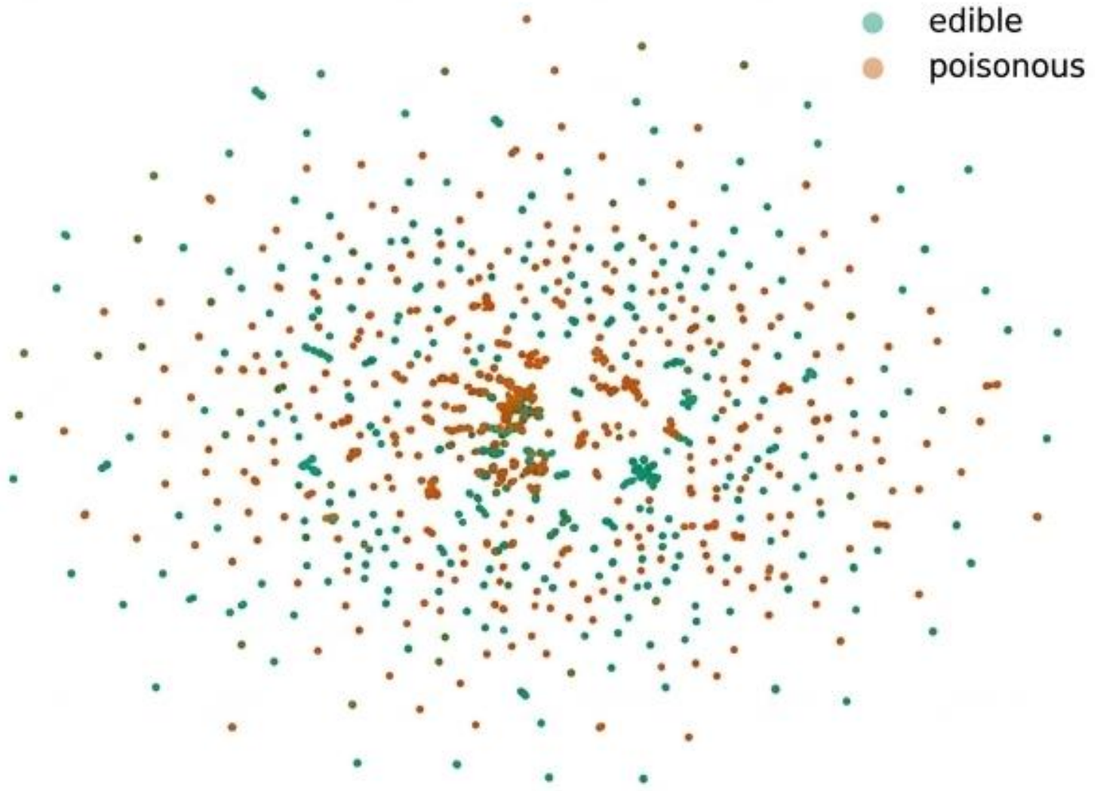
対象行が173種のルール表とどれだけ値レベルで一致するかを集計。単体で相関係数 0.8896 という破壊的シグナルを検出。

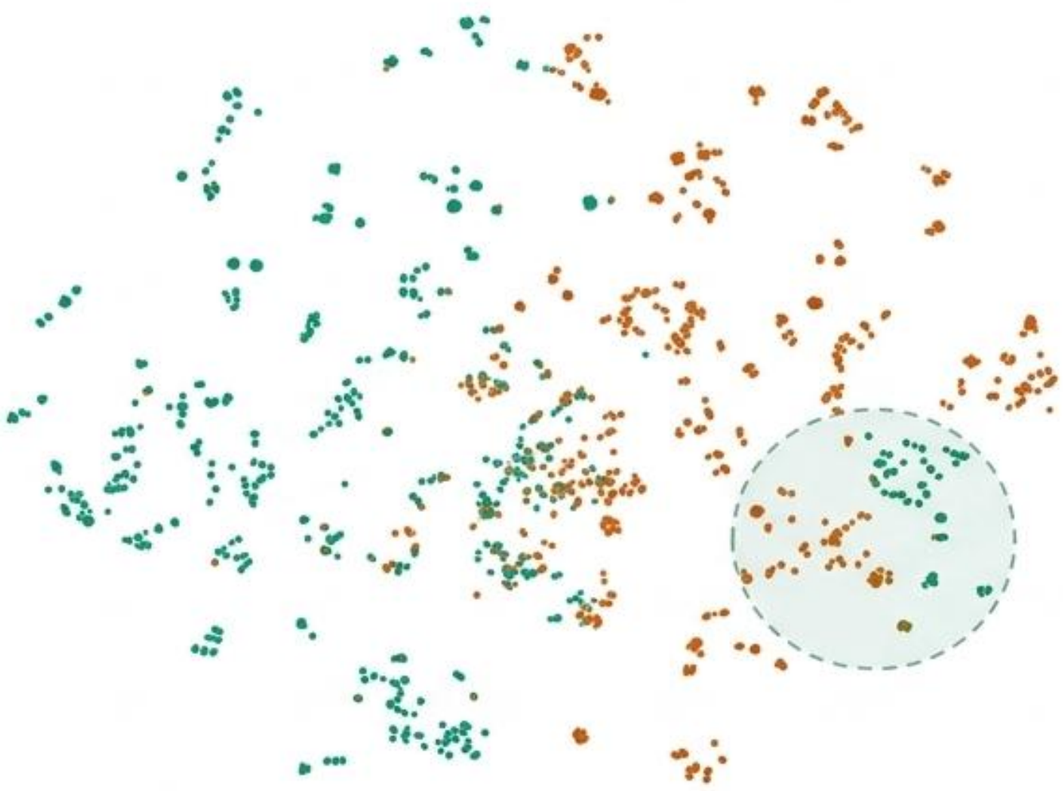
逆相関（-0.82）の強力な指標。食用ルールとの一致率。

実装：P(poisonous|value) を列ごとのBayesian特徴量として生成し、強力な予測基盤を構築。

次元削減が暴く真の構造：t-SNE vs UMAP

前提条件: 全215次元ではなく、Targetと強く相関する Primary特徴量21次元のみ を使用

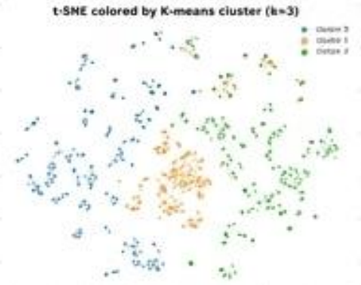


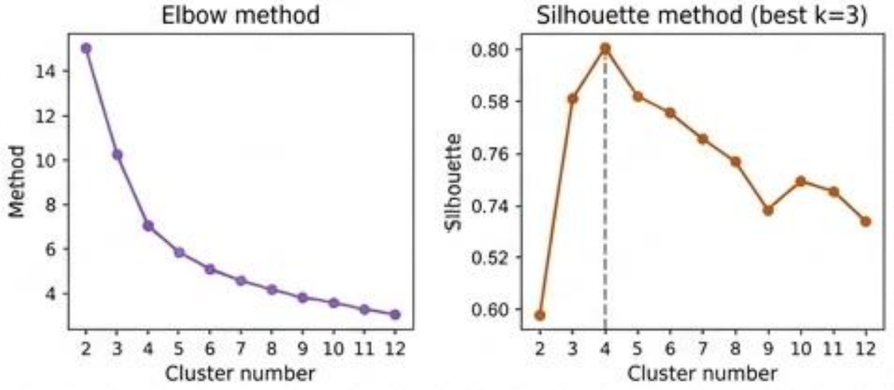


UMAP: 大域構造の保持。EdibleとPoisonousが中心部で混在している。

t-SNE: 局所構造の保持。明確なクラスタ（島）としてEdibleとPoisonousが分離して出現。

診断マトリックス：K-means (k=3) が導き出した3つの世界





Cluster 0 (36.1%)

primary_match_ratio: 0.000
Poisonous率: 0.6%

【確信的食用】

Cluster 1 (44.4%)

primary_match_ratio: 0.999
Poisonous率: 99.6%

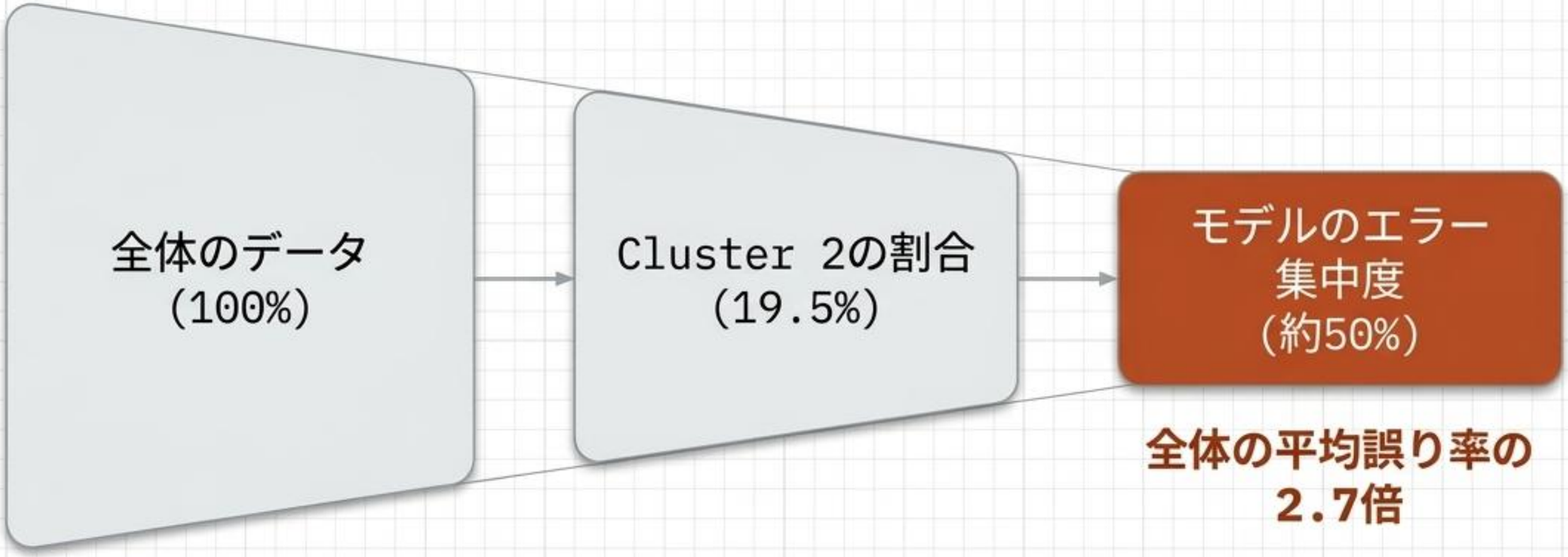
【確信的毒】

Cluster 2 (19.5%)

primary_match_total: 0.000
Poisonous率: 52.8%

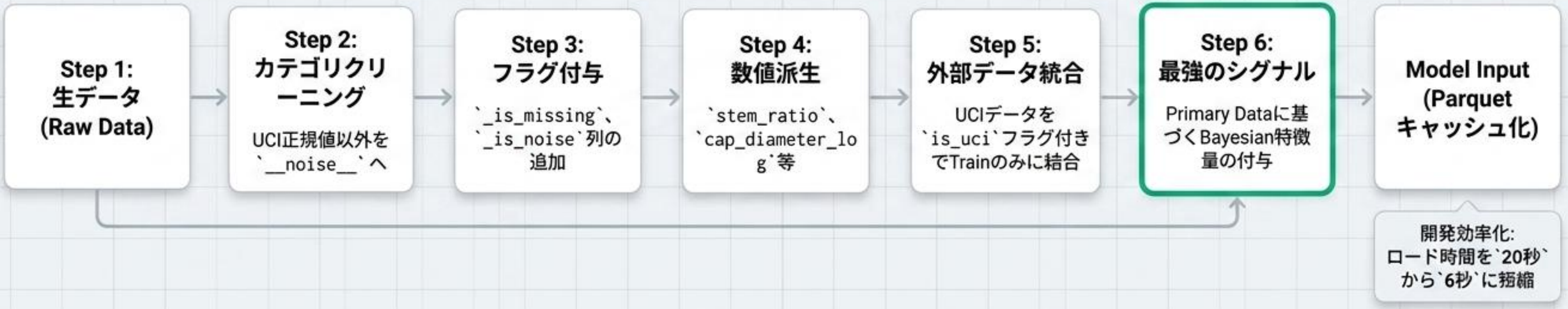
【未知の領域】
Primary Dataが一切カバーしていない謎のデータ群

ボトルネックの正体：「未知の19.5%」に潜む限界

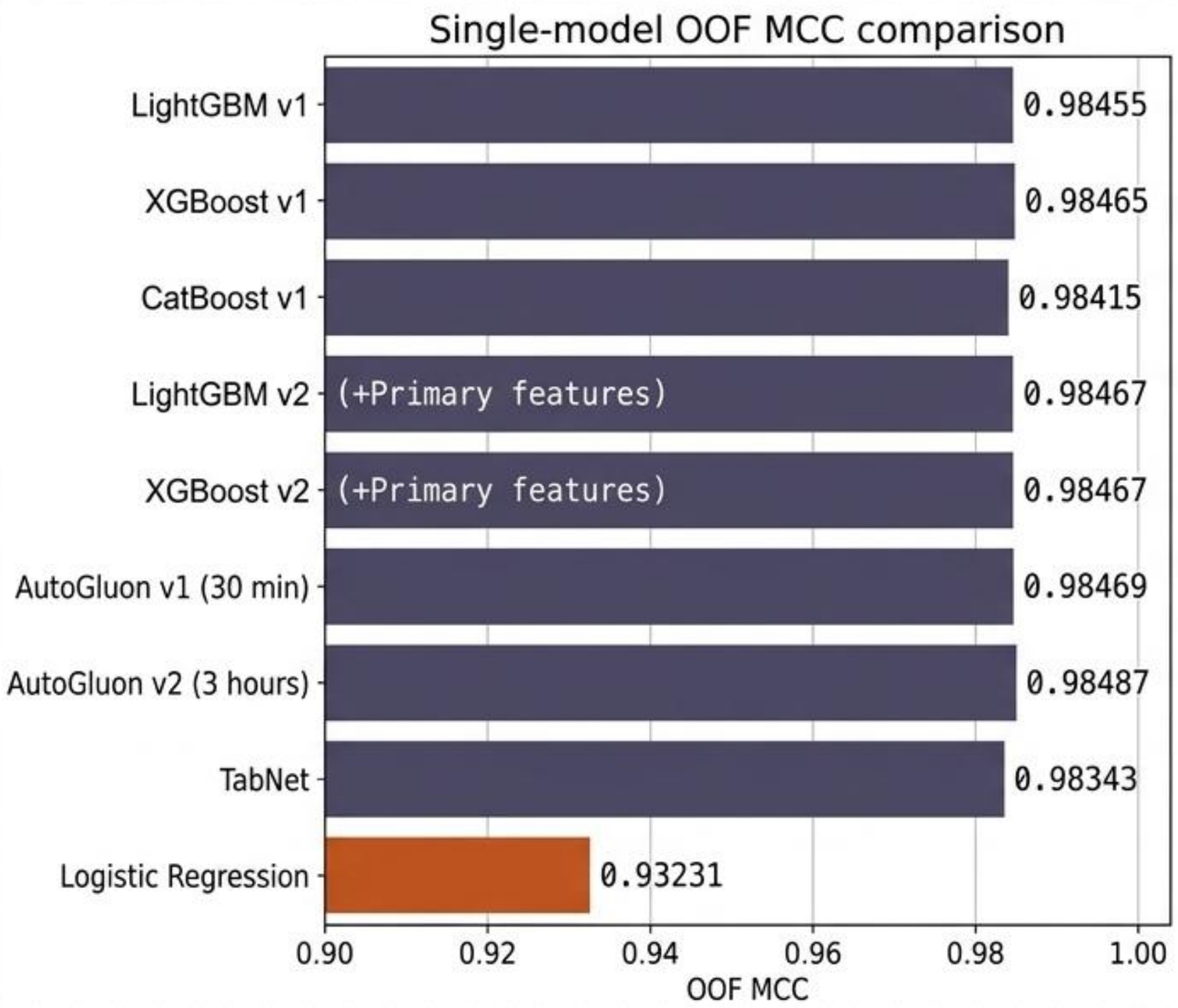


検証と結論: 複合欠損度（critical_missing_count）を追加学習してもOOF MCCは0.98466で停滞。 これは特徴量エンジニアリングの不足ではなく、データセット自体が持つ『構造的な限界（未知の値の組み合わせ）』であると結論付けた。

特徴量エンジニアリング・パイプライン



単一モデル・バトル：OOF MCCとハードウェアの相性



LightGBM / XGBoost v2 (0.98467)

最高性能。ただしLightGBMのGPUツリー学習器は低カーディナリティ特徴量でクラッシュしたため、CPUモードで回避。

CatBoost (0.98415)

GBM系で最も低スコアで学習時間も最長（96分）。ブレンド候補から除外。

Logistic Regression (0.93231)

全特徴量のOne-HotではGBM系に惨敗。EDAでの「全次元では分離不能」という仮説を裏付ける結果。

アンサンブル戦略：異なるアーキテクチャによる多様性の注入

GBM系（ベースライン）

LightGBM & XGBoost（強力なツリーベースアンサンブル）

AutoGluon v2（AutoML枠）

3時間ラン。内部構成: LightGBM XT (65%) + Random Forest Gini (30%) + RF Entropy (5%)。「予測の安定化」に寄与。

TabNet（NN枠）

PyTorchベース（OOF 0.98343）。決定木とは根本的に異なる学習メカニズムによる多様性のスパイスとして機能。

重み付き線形ブレンド

交差検証(OOF)の予測値に基づく最適化された統合

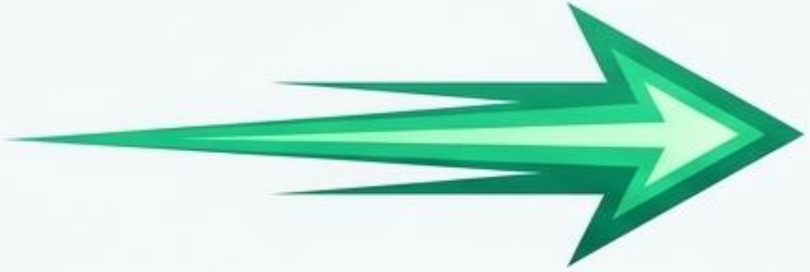
O(n log n)の魔法：MCC閾値のベクトル化最適化

Before (Grid Search)

手法: 991段階を sklearn に全探索ループで渡す。

計算量: $O(n\_steps \times n)$

結果: 66通りのブレンド探索に **数十分** を要する。



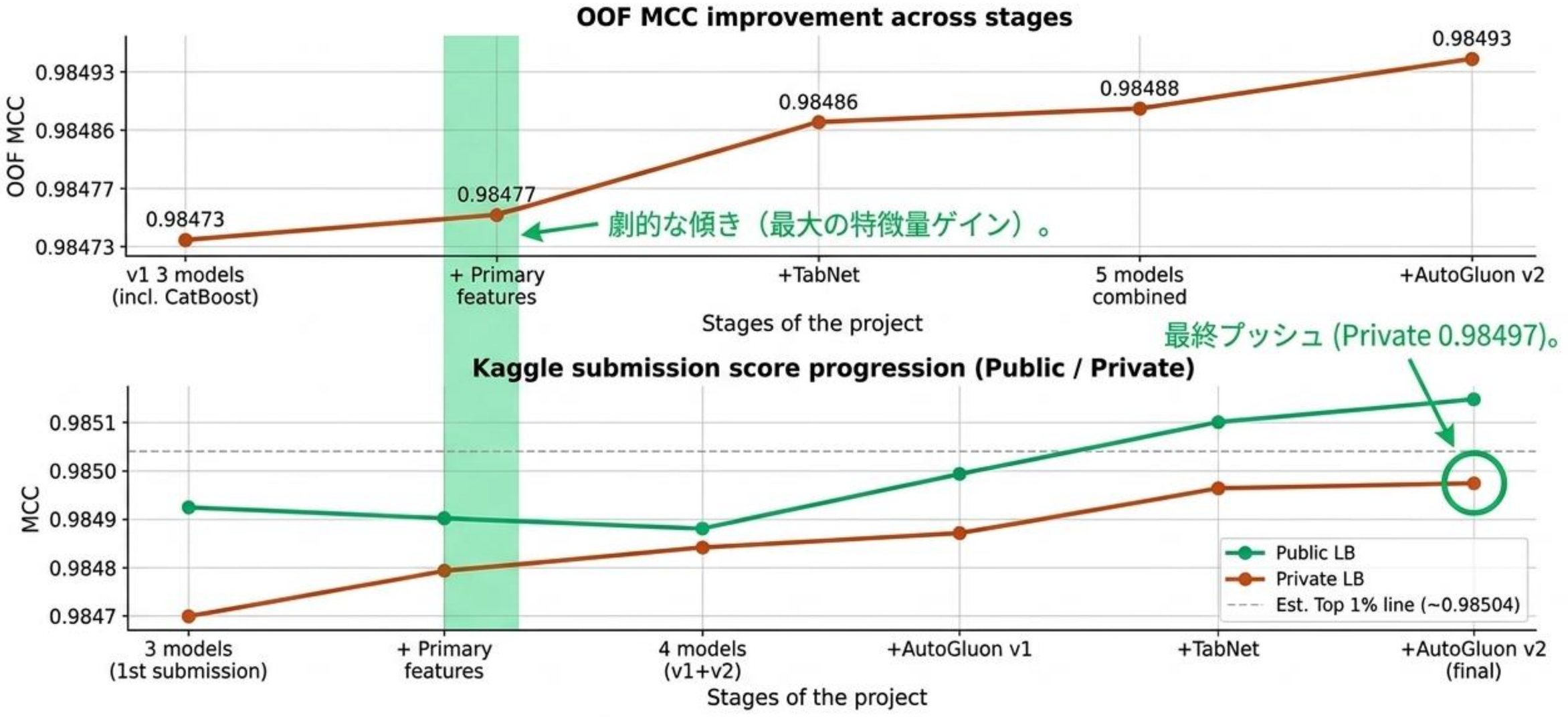
After (Vectorized Cumulative Sum)

手法: 予測確率を降順ソートし、累積和 (cumsum) でTP/FP/FN/TNを一括計算。

計算量: $O(n \log n)$ （ソート1回のみ）

結果: 離散化誤差のない厳密解を **36.3秒** で算算出。イテレーションを劇的に高速化。

成長の軌跡：Top 15%到達へのマイルストーン



注記: MCCの4〜5桁目の差はサンプリングノイズの範囲内。過度なPublic LBへの適合（オーバーフィット）を避け、OOFとモデル多様性を信じた結果。

結論と次なる課題（Lessons Learned & Next Steps）

Technical Pitfalls（学び）

- conda run の標準出力バッファリング問題 (Python直接呼び出しで回避）。
- CatBoost GPU実行時のcallbacks 不全（ファイル監視への切り替え）。

Strategic Win（勝因）

- 評価指標(MCC)の数学的性質をハックしたベクトル化実装。
- 次元削減とK-meansによる「Primary Dataのシグナル」の視覚的証明。

Next Steps（課題）

- Cluster 2（謎の19.5%）を判別する新たな外部シグナルの探索。
- ブレンド重み探索への非線形オプティマイザ（Optuna）の導入。

The ultimate battle is always isolating the signal from the noise.

（究極の戦いは、常にノイズからシグナルを分離することにある）