# COMPASデータセットを用いた再犯予測モデルの精度向上

工学部 電子情報学科
学籍番号：08D23091
辻 孝弥

July 10, 2025

## 1 はじめに

本課題では，COMPAS データセットを用いて再犯予測モデルの精度向上を目的とした。XGBoost と MLP の 2 種類のモデルを構築・改善し，さらに両者をスタッキングする手法を適用した。

## 2 実験方法

### 2.1 XGBoost モデルの導入とハイパーパラメータ調整

使用モデル: XGBClassifier

#### 2.1.1 Optuna による事前探索と二段階学習

Optuna を用いて最適なハイパーパラメータを事前に調査し，その結果を `best_params` に設定して一次学習を実施した。

続いて一次学習済みモデルで派生特徴量を生成し，クラス不均衡補正などを加えた新たなパラメータセット `new_params` で再度全データを用いた二次学習を行った。

#### 2.1.2 主なハイパーパラメータ設定

- 学習率 (`learning_rate`)：0.05 → 0.08
- 木の深さ (`max_depth`)：3
- サブサンプリング率 (`subsample`)：0.90
- 特徴量サンプリング率 (`colsample_bytree`)：0.78
- 最小子ノード重量 (`min_child_weight`)：5
- 正則化パラメータ (`gamma`)：0.50
- L1 正則化 (`reg_alpha`)：$5.4 \times 10^{-6}$
- L2 正則化 (`reg_lambda`)：5.88

- 木数 (n_estimators)：200
- 早期停止 (early_stopping_rounds)：10
- クラス不均衡補正 (scale_pos_weight)：訓練データのクラス比に基づき自動設定

## 2.2 特徴量エンジニアリング

### 2.2.1 不要特徴量の削減

重要度上位 80%を残し，下位 20%を削除（priors_count・age は保護）

### 2.2.2 ChatGPT による特徴量選択

全ての CSV カラムを逐次試すのは非効率なため，ChatGPT を用いて平均的に効果が見込める無難な派生特徴量案を取得し，それを見ながら実装した。

### 2.2.3 主な派生特徴量

- priors_per_year（前科数／ (年齢＋ 1)）
- sum_priors_and_age（前科数＋年齢）
- age_squared（年齢 $^2$）
- log_priors_p1（log(前科数＋ 1)）
- age_times_priors（年齢×前科数）
- total_juv_cnt（少年期犯罪合計）
- juv_ratio（少年期犯罪合計／ (前科数＋ 1)）
- log_len_stay（拘束期間の対数化）

### 2.2.4 dob（生年月日）の扱い

- 初期には再犯率と無関係と判断し除外したが，除外時の Accuracy が 0.688→0.679 に低下
- 最終的には dob を含める実装とし，Accuracy を 0.699→0.710 へ改善

## 2.3 MLP モデルの構造・学習戦略改善

### 2.3.1 ネットワーク構造

隠れ層：256 → 128，ReLU ＋ BatchNorm ＋ Dropout(0.3/0.2)

### 2.3.2 学習戦略

- Optimizer：Adam(lr=1e-3, weight_decay=1e-5)
- Scheduler：CosineAnnealingLR
- 損失関数：クラス重み付き CrossEntropy
- EarlyStopping：patience=15

### 2.4 モデルスタッキング

- メタ学習器：LogisticRegression (L2, C=1.0)
- 入力特徴：XGBoost/MLP の検証データ予測確率
- 閾値決定：Youden's J による最適閾値

## 3 結果

| モデル | Accuracy |
|---|---|
| ① MLP 単体 | 0.675 |
| ② XGBoost 単体＋特徴量変更 | 0.688 |
| ③ ②モデル（dob 除外） | 0.679 |
| ④ 今回実装モデル（dob 除外） | 0.699 |
| ⑤ 完全実装モデル（dob 含む） | 0.710 |

### 3.1 ⑤の詳細ログ

#### 3.1.1 混同行列

```
[[1591  421]
 [ 627  968]]
```

#### 3.1.2 分類レポート

```
              precision    recall  f1-score   support

           0      0.717     0.791     0.752      2012
           1      0.697     0.607     0.649      1595

    accuracy                         0.710      3607
   macro avg      0.707     0.699     0.701      3607
weighted avg      0.708     0.710     0.706      3607
```

Accuracy: 0.710
ROC-AUC : 0.767
LogLoss : 0.576

## 4 考察

### 4.1 Optuna によるハイパラ最適化

一次学習で得られたパラメータをもとに，二次学習時にはクラス不均衡補正などを加えた `new_params` を適用し，性能向上に寄与した。

### 4.2 ChatGPT 活用の特徴量設計

無難で効果の期待できる特徴量案を迅速に取得でき，実装工数を大幅に削減できた。

### 4.3 dob 除外の効果検証

当初「再犯率に関係がない」と判断して dob を除外したが，Accuracy が 0.688→0.679 に低下した。

dob を含めることで 0.699→0.710 へ改善し，生年月日情報が有用であることを確認した。

### 4.4 スタッキング効果

異なるモデルの補完性により，最終的に Accuracy:0.710 ／ ROC–AUC:0.767 ／ LogLoss:0.576 を達成した。

## 5 結論

Optuna による二段階ハイパラ最適化と ChatGPT 提案の特徴量エンジニアリングを組み合わせ，実装に忠実に dob 情報の有効性を再検証した結果，再犯予測タスクにおいて高い汎化性能を実現できた。今後はさらに異なる情報源やモデル統合手法を探索し，性能向上を図る余地がある。

## 参考文献


- スタッキングの実装と効果について
  https://potesara-tips.com/ensemble-stacking/#toc13