



PWS Cup 2025

本戦ガイドブック

第 1.1 版



2025 年 9 月 23 日

PWS2025 実行委員会

PWS Cup 2025 WG

まえがき

本書は、PWS Cup 2025 の参加者を対象として、本戦の手順とルール（制約条件）を記したものである。ただし本書に記した本戦の手順またはルールに問題点や不明点が発覚した際は、事務局の判断に基づき、本戦の手順やルールを変更・追加する場合がある。本戦の手順やルールが変更・追加となった場合は、本書を更新するとともに、速やかに参加者にその旨を連絡する。

本戦に先立ち 2025 年 8 月 20 日から 9 月 15 日にかけて予備戦が行われたが、本戦の手順・ルールは、予備戦の手順・ルールとは一部異なるため注意。なお予備戦からの変更点は第 6 章に記載している。

本書が更新された際（例：第 1 版→第 1.1 版）は、更新箇所を別色で記載する。第 1.1 版の更新箇所は緑字としている。

目次

まえがき	1
第 1 章 はじめに	4
1.1 スケジュール	4
1.2 参加条件.....	4
1.3 チームの制約事項	4
1.4 発表会・表彰式.....	5
1.5 競技用オンラインシステム Codabench	5
第 2 章 本コンテストの概要	7
2.1 全体の流れ.....	7
2.2 加工対象のデータ	7
2.3 本戦加工フェーズ	9
2.4 本戦攻撃フェーズ	9
2.5 有用性の得点	10
2.6 匿名性の得点	12
2.7 攻撃力の得点	12
第 3 章 本戦加工フェーズの手順・ルール	13
3.1 手順.....	13
第 4 章 本戦攻撃フェーズの手順・ルール	17
4.1 手順.....	17
4.2 ルール（制約条件）	18
第 5 章 採点ルール	19
第 6 章 予備戦からの変更点.....	20
付録 A Synthea について	21

A.1 概要	21
A.2 インストールと合成患者データ生成	21
A.3 出力データ	21
付録 B スクリプト、サンプルコード	23

第1章 はじめに

PWS Cup は、情報処理学会 コンピュータセキュリティ研究会 PWS 組織委員会主催で 2015 年から毎年開催している、個人データの安全な活用を目的に匿名化技術（加工技術）と攻撃技術を競う対戦型コンテストである。今年で 11 回目を迎える PWS Cup 2025（以下、本コンテスト）では、架空の患者データを用いて、患者のプライバシーを守りつつ誤差の少ない医療分析ができる匿名化データおよび機械学習モデルの作成を目的とする。

1.1 スケジュール

本戦以降のスケジュールは以下の通り（時刻は日本時間）。

- 2025 年 9 月 19 日(金)9:00～10 月 3 日(金)9:00：本戦加工フェーズ
- 2025 年 10 月 8 日(水)9:00～10 月 21 日(火)9:00：本戦攻撃フェーズ
- 2025 年 10 月 29 日(水)9:00～17:30：発表会・表彰式（岡山コンベンションセンター）

1.2 参加条件

本コンテストに参加するためには、チームを作り、チーム代表者が 2025 年 9 月 15 日(月)までに [参加申込ページ](#) から参加申込を行う必要がある。チームは 1 名から最大 5 名とする。ただし学生チームの場合は責任者 1 名と指導者 1 名をチームに追加できる。最大人数を超えなければチームメンバーを後から追加してもよいが、別チームへの変更は認めない。チームメンバーを追加する場合は速やかに [PWS Cup 2025 事務局](#)（以下、事務局）に連絡すること。

チーム代表者は、本コンテストの発表会・表彰式が行われるコンピュータセキュリティシンポジウム 2025（CSS2025）に参加登録しなければならない。止むを得ず参加登録できない場合は、別のチームメンバーが代替してもよいが、その場合は速やかに事務局に連絡すること。

1.3 チームの制約事項

同一の組織や研究室での複数チームの参加は認められ、指導者は複数チームに所属してもよいが、指導者以外が複数チームに所属することは認められない。複数のチームが、他のチームにとって不公平にならない範囲で本コンテストに関する情報交換や勉強会等を実施することは認められる。不公平と見なされる例としては、自チームのみが知る配布データの共有、チーム固有の匿名化手法や機械学習モデル作成手法の共有（論文等で公知の手法の共有は OK）、匿名化データや機械学習モデルの共有、有用性結果や攻撃サンプルコードの実行結果の共有、攻撃手法の共有（論文等

で公知の手法の共有は OK)、攻撃データや攻撃結果の共有が挙げられる。不公平と疑われる行為が発覚した場合は、事務局が判断し、不公平な行為と判断した場合は当該チームを失格とする場合があるので注意すること。

1.4 発表会・表彰式

各参加チームは、1.1 節に記載の発表会・表彰式にて、本コンテストで実施した内容についてショートプレゼン（5 分程度の予定）およびポスター発表（8 チーム同時で 45 分程度の予定）を行う。発表時間帯や発表方法等の詳細は、チーム代表者にメールで通知し、[PWS Cup 2025 ホームページ](#)にも掲載する。

表彰式では、以下の受賞チームを発表し、表彰を行う。

- 総合 1 位～5 位
 - 匿名性の得点＋有用性の得点が高かったチーム（得点については第 4 章を参照）
 - 予備戦の得点と本戦の得点を 1 対 9 の割合として合計（詳細は第 4 章を参照）
- ベストアタック賞
 - 攻撃力の得点が最も高かったチーム（詳細は第 4 章を参照）
- ベストプレゼン賞
 - ショートプレゼンおよびポスター発表が最も優れていたチーム
 - 別途公表される複数の審査員により決定
- ベストデータサイエンティスト賞
 - 実際に今回作成した匿名化データを使って有用な分析手法を提案したチーム
 - 分析手法の独創性や実用性、匿名化データを使った分析の有用性等を総合的に評価
 - ショートプレゼン、ポスター発表で提案（盛り込むかどうかは任意）
 - 別途公表される複数の審査員により決定

上記各賞を受賞したチームには賞状が授与される。また総合 1～3 位、ベストアタック賞、ベストプレゼン賞を受賞したチームには、副賞として岡山に関する記念品が贈呈される。

1.5 競技用オンラインシステム Codabench

本コンテストでは競技用のオンラインシステムとして [Codabench](#) を用いる。チーム代表者は [Codabench の本コンテスト用サイト](#)（以下、本サイト）にアクセスし、“Get Started”の”How to Participate”タブの記載に従い本サイトの利用申請・手続きを行う。手続きが完了すると、加工フェーズや攻撃フェーズにおけるデータの提出、リーダーボード（各チームの得点等が表示）や FAQ

の閲覧が可能となる。なおチーム代表者以外は、データの不正提出防止のため、利用申請しても許諾されない。したがってデータの提出はチーム代表者が行い、リーダーボードや FAQ の情報は適宜チーム代表者が他のチームメンバーに共有していただきたい。

本サイトから提出するデータは、システムの仕様で zip ファイルに限定される。zip ファイル以外はエラーとなり、フォルダーを zip ファイルにした場合もエラーとなるので注意すること。zip ファイルの中身やファイル名は 3.1 節、4.1 節で説明する。

本サイトからのデータ提出回数の上限は、本戦加工フェーズが 1 日 3 回（日本時間の 9:00 にリセット）、本戦攻撃フェーズがトータルで 8 回である。

本サイトからデータを提出すると、

”Submitting”、”Submitted”、”Preparing”、”Running”、”Scoring”、”Finished”

の順にステータスが変更する（ブラウザをリロードすると現在のステータスが表示される）。

”Finished”は正常終了であり、提出回数が 1 回増え、提出データの得点等が表示され、リーダーボードへの登録が可能となる。提出データの不備等、問題が生じた場合は”Failed”となり異常終了する。システム設定により、10 分後までに”Finished”にならなかった場合も通常は”Failed”になる。

Codabench は他のコンテストでも用いられており、負荷状況等によって動作が不安定になることがある。特に提出データに不備が無くても”Failed”になったり、10 分経っても”Finished”以前のステータスのまま変わらないことがある。止むを得ず”Cancel Submission”ボタンを押すと、提出回数が 1 回増えてしまうことがあるので、”Cancel Submission”ボタンの使用は避けること。問題が解消しない場合は事務局に連絡していただきたい。

第2章 本コンテストの概要

2.1 全体の流れ

本コンテストでは、全ての参加チームは「加工フェーズ」（匿名化フェーズ）と「攻撃フェーズ」の両方に参加する。加工フェーズでは、出題者から渡された（架空の）患者データから、匿名化データと機械学習モデルを作成して期限内に本サイトから提出する。攻撃フェーズでは、他チームの匿名化データと機械学習モデルを攻撃（メンバーシップ推定攻撃¹）して攻撃結果データを提出する。加工フェーズと攻撃フェーズの処理イメージをつかむための[ハンズオン](#)を公開しているので、適宜参照されたい。

出題者は後述する各チームの「有用性」「匿名性」「攻撃力」の得点や順位等の結果を発表する。参加チームは、有用性と匿名性の得点の合計の高さ、および攻撃力の得点の高さを競う。全体の流れのイメージは図1のようになる。

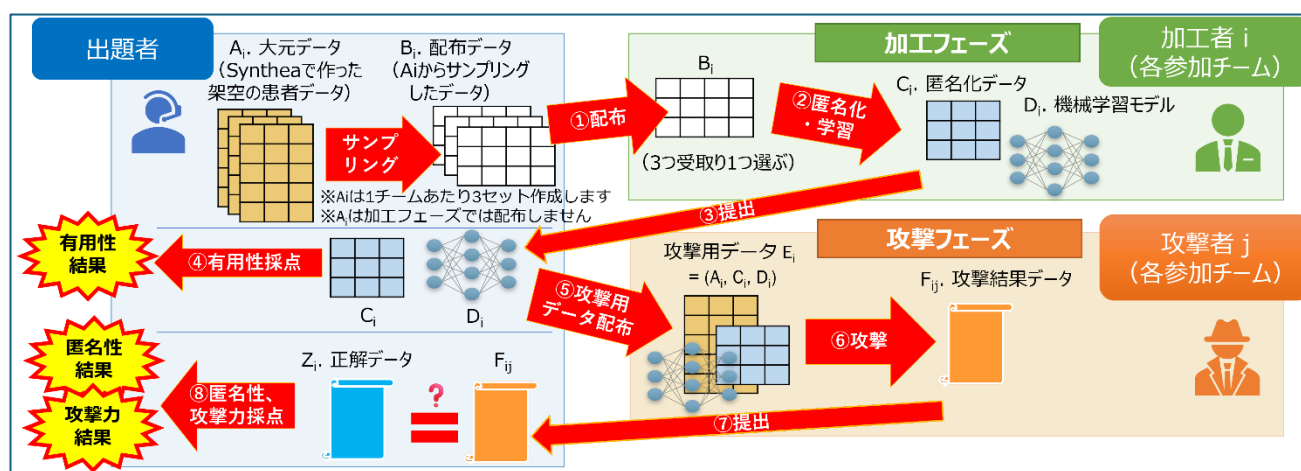


図1 PWS Cup 2025 の全体の流れのイメージ

2.2 加工対象のデータ

本コンテストにおいて出題者（＝事務局）は、架空の患者データを生成するソフトウェア [Synthea](#)、および独自のデータ加工プログラム [unified_synthea.py](#) を用いて、CSV 形式の 10 万人分の患者データファイル（図1の A_i に相当）を作成する。当該ファイルはヘッダー行を含む 100,001

¹ メンバーシップ推定攻撃：匿名化データや機械学習モデル等の加工データに対し、ある個人のデータが当該加工データの作成に使われたかどうかを推定する行為。推定が難しければ、当該加工データの個人特定も難しいと考えられることから、個人特定のリスク評価に用いられる。

行 18 列の 1 患者 1 行の CSV ファイルとなる。本戦の患者データファイルの形式を表 1 に示す (columns_range.json で定義されている)。

列番号	属性名	種別	値域	小数桁
1	GENDER	カテゴリ	F/M	—
2	AGE	整数	[2,110]	0
3	RACE	カテゴリ	asian/black/Hawaiian/native/other/white	—
4	ETHNICITY	カテゴリ	hispanic/nonhispanic	—
5	encounter_count	整数	[4,1211]	0
6	num_procedures	整数	[0,2442]	0
7	num_medications	整数	[0,7195]	0
8	num_immunizations	整数	[0,40]	0
9	num_allergies	整数	[0,17]	0
10	num_devices	整数	[0,160]	0
11	asthma_flag	整数	[0,1]	0
12	stroke_flag	整数	[0,1]	0
13	obesity_flag	整数	[0,1]	0
14	depression_flag	整数	[0,1]	0
15	mean_systolic_bp	実数	[37.22, 37.23, …, 172.84, 172.85]	2
16	mean_diastolic_bp	実数	[1.78, 1.79, …, 126.16, 126.17]	2
17	mean_bmi	実数	[2.97, 2.98, …, 73.57, 73.58]	2
18	mean_weight	実数	[4.55, 4.56, …, 196.6, 196.7]	2

表 1 本戦の患者データファイルの形式

本戦の患者データファイルは、24 チーム×3 個のファイル AAI_k.csv (i はチーム番号 01~24、k は 1,2,3) となる。各ファイルはそれぞれ異なる地域設定で患者データを作成しているため、分布が異なるデータとなっている。なお全てのファイルについて重複行は存在しない（出題者が確認し、存在した場合は当該ファイルを作成し直す）。

次に出題者は、AAI_k.csv のヘッダー行以外の 10,000 行のデータをランダムに抽出し、ヘッダー行を加えた 10,001 行 18 列の CSV ファイル BBI_k.csv を作成する。出題者は AAI_k.csv の何行目のデータが抽出されたかを示すファイル ZZI_k.csv を作成しておく。ZZI_k.csv は 100,000 行 1 列のデータであり、AAI_k.csv の $\ell + 1$ 行目のデータが BBI_k.csv に存在すれば、ZZI_k.csv の ℓ 行目の値

は”1”、含まれなければ”0”となる。ZZi_k.csv にはヘッダー行は無いことに注意。

2.3 本戦加工フェーズ

本戦加工フェーズでは、事務局が開始日時後速やかに各チーム i の代表者に(BBi_1.csv, BBi_2.csv, BBi_3.csv)のダウンロードリンクをメールで通知する。リンク先ページにアクセスすると、ダウンロード用パスワードがチーム代表者にメールされるので、当該パスワードを用いて(BBi_1.csv, BBi_2.csv, BBi_3.csv)をダウンロードする。

加工者（参加チーム） i は、BBi_1.csv, BBi_2.csv, BBi_3.csv の何れかを選び（加工者が自由に選んでよい。選んだファイルを BBi_s.csv (s は 1,2,3 の何れか) とする)、匿名化データファイル CCI_s.csv および機械学習モデル（2.5 節で説明する「XGBoost を用いた脳卒中リスク予測モデル」）データファイル DDi_s.json を作成して本サイトから提出する。提出回数は 1 日 3 回までとなる（日本時間の 9:00 にリセットされる）。CCI_s.csv および DDi_s.json の作成や提出方法については第 3 章を参照。

ある提出時の CCI_s.csv および DDi_s.json の s の値は一致させる必要がある。すなわち、同一の BBi_s.csv から CCI_s.csv および DDi_s.json を作成する。ただし毎回同一の s とする必要は無く、例えばある提出時は CCI_1.csv, DDi_1.json で、別の提出時は CCI_2.csv, DDi_2.json としてもよい。加工フェーズ終了時刻にリーダーボードに登録されているデータ ID に紐づいたデータが最終版の提出データとして採用される。

本戦加工フェーズでは、本サイトからデータを提出して”Finished”になると、本サイトに有用性の得点（2.5 節参照）が表示される。この得点を参考に、最終版の提出データを選んでリーダーボードに登録する。加工フェーズ終了時刻まではリーダーボードに登録するデータ ID を変更できる。したがって他チームの状況を見ながら戦略的にデータ ID をリーダーボードに登録することも可能である。

2.4 本戦攻撃フェーズ

本戦攻撃フェーズでは、事務局が開始日時後速やかに全チームの攻撃用データ AAi.csv, CCI.csv, DDi.json (i はチーム番号 01~24) のダウンロードリンクを [PWS Cup 2025 ホームページ](#)に掲載するので、攻撃者（参加チーム）は当該データをダウンロードする（事務局は各チームが提出した CCI_s.csv, DDi_s.json およびそれらに対応した AAi_s.csv をそれぞれ CCI.csv, DDi.json, AAi.csv にリネームする）。ただし未提出のチームがあった場合は、当該チームのファイルは存在せず、事務局がチーム代表者にその旨を周知する。

攻撃者 j (j は自チーム番号) は、未提出のチームを除く全チームの匿名化データ C*Ci*.csv、機械学習モデル DD*i*.json、および大元データ AA*i*.csv を用いて、メンバーシップ推定攻撃を行い、攻撃結果ファイル FF*j*.csv を作成して本サイトから提出する。提出回数はトータルで 8 回までとなる。攻撃結果ファイルを提出すると、本サイトに攻撃結果の得点が表示される。詳細は第 4 章を参照。

2.5 有用性の得点

本コンテストでは、有用性の評価項目として下記に示す「項目 1：基本統計等」、「項目 2：喘息リスク因子の分析」「項目 3：年齢群別にみる医療利用の分布差解析」、「項目 4：XGBoost を用いた脳卒中リスク予測」を用いる。

- 項目 1：[基本統計等](#)（詳細はリンク先ページ参照）
 - 数値列の統計（min-max 正規化後）：平均、標準偏差、四分位数
 - 数値×数値の相関行列（Pearson）
 - 各カテゴリ列の集計値の比率（ratio）
 - カテゴリ×数値の要約統計（min-max 正規化後）：カテゴリの値毎の数値列の平均、標準偏差、四分位数（Group By）
 - ✧ AGE を [0-17, 18-44, 45-64, 65-74, 75+] の固定ビンで再生成してカテゴリ列 AGE_GROUP を新たに作成
 - カテゴリ×カテゴリのクロス集計の値の比率（ratio）
- 項目 2：[喘息リスク因子の分析](#)（詳細はリンク先ページ参照）
 - 二値目的変数 asthma_flag に対してロジスティック回帰を適用し、係数や信頼区間由来の指標を 0～1 に正規化して出力
 - 出力（何れも 0～1 の実数）
 - ✧ AUC
 - ✧ ロジスティック回帰の回帰係数（各回帰係数の差をロジスティック関数で正規化：
 $(\text{ロジスティック関数}(\text{差}) - 0.5) \times 2$ ）
 - ✧ p 値
 - ✧ オッズ比 OR を $OR/(1+OR)$ に変換
 - ✧ 95%信頼区間の下限値 CI_low を $CI_low/(1+CI_low)$ に変換
 - ✧ 95%信頼区間の上限値 CI_high を $CI_high/(1+CI_high)$ に変換
 - ✧ 多重共線性の強さを示す VIF を $1 - 1/\max(VIF, 1)$ に変換
- 項目 3：[年齢群別にみる医療利用の分布差解析](#)（詳細はリンク先ページ参照）
 - 年齢を AGE_GROUP で群分けし、各医療指標（encounter_count, num_medications など）について Kruskal-Wallis 検定を行い、統計量を 0～1 に正規化して出力

- 出力（0～1 に正規化）
 - ✧ H_{norm} （Kruskal-Wallis 検定の統計量 H の規格化指標）
 - ✧ p 値を $-\log_{10}(p)$ に変換し数値安定化
 - ✧ 効果量：epsilon2, eta2_kw, rank_eta2
 - ✧ 群間ペアの優越確率に基づく $A_{\text{pair_avg}}$ と差の非対称性 $A_{\text{pair_sym}}$
- 項目 4：[XGBoost を用いた脳卒中リスク予測](#)（詳細はリンク先ページ参照）
 - XGBoost を用いて二値目的変数 obesity_flag を予測する機械学習モデルのデータファイル DDi.json を入力
 - 出題者が用意している、BBi_k.csv のヘッダー行以外からランダムに 5,000 行、および AAi_k.csv のヘッダー行と BBi_k.csv のデータ以外からランダムに 5,000 行抽出したテストデータファイル XXi_k.csv を入力（XXi_k.csv は Codabench に格納されているため、参加者はアクセスできず、本サイトからデータ提出したときのみ利用され、DDi.json を用いた予測結果のみが得られる）
 - 出力：テストデータファイル XXi_k.csv の予測正解数

有用性の得点は、以下の 4 つの得点の合計点となる（100 点満点。小数点第 3 位を四捨五入）。

- （項目 1 の得点：40 点満点）BBi_s.csv と CCI_s.csv をそれぞれ項目 1 の実行プログラムに入力し、各出力値の差分を求め、当該差分の最大値 stats_diff_max に対して、

$$\text{Max}(0, 1 - \text{stats_diff_max}) \times 40$$

※stats_diff_max が 1 以上のときに得点がマイナスにならないよう、Codabench で得点を計算する際は上記のように Max 関数を適用

- （項目 2 の得点：20 点満点）BBi_s.csv と CCI_s.csv をそれぞれ項目 2 の実行プログラムに入力し、各出力値の差分を求め、当該差分の最大値 LR_diff_max に対して、

$$(1 - \text{LR_diff_max}) \times 20$$

- （項目 3 の得点：20 点満点）BBi_s.csv と CCI_s.csv をそれぞれ項目 3 の実行プログラムに入力し、各出力値の差分を求め、当該差分の最大値 KW_diff_max に対して、

$$(1 - \text{KW_diff_max}) \times 20$$

- （項目 4 の得点：20 点満点）DDi_s.json と XXi_s.csv を項目 4 の実行プログラムに入力し、出力される正解数 XGBT_match に対して、

$$\text{XGBT_match} / 10000 \times 20$$

2.6 匿名性の得点

加工者（参加チーム） i の匿名性の得点は、提出データ $CCi_s.csv$, $DDi_s.json$ および大元データ $AAi_s.csv$ を用いて各攻撃者が $BBi_s.csv$ のメンバーシップ推定攻撃を行った正解数の最大値 MIA_match_max および減点 $Penalty_value$ に対して、

$$\underline{\text{Max}(0, (10000 - MIA_match_max) / 100 - Penalty_value)}$$

と定義される。ここで $Penalty_value$ は、 $BBi_s.csv$ の ℓ 行目のデータのメンバーシップ推定攻撃に成功した攻撃者の数を $L(\ell)$ とし、 $L(\ell)$ の最大値を L としたとき、 $Penalty_value = L - 1$ と定義される。これは、全ての患者のプライバシーを守るための本コンテストの特徴であり、この特徴により、各加工者は攻撃されやすい患者データが無いよう加工することを誘導している。

2.7 攻撃力の得点

本戦の攻撃力の得点は、総合得点（有用性の得点 + 匿名性の得点）が高い 上位 10 チーム に対するメンバーシップ推定攻撃成功数の合計値を 100 で割った値（0～1000 点）と定義される。自チームが上位 10 チームに入っている場合は、自チームに対するメンバーシップ推定攻撃成功数を、他チームによる自チームに対するメンバーシップ推定攻撃成功数の最大値で置き換える。

第3章 本戦加工フェーズの手順・ルール

3.1 手順

加工者（参加チーム）i は以下を行う。

1. 事務局から(BBi_1.csv, BBi_2.csv, BBi_3.csv)を受け取る ※2.3 節に具体的な手順を記載
2. ファイル id.txt を作成し、チーム番号 i (01~24) を書き込む
3. 期限（10月3日(金)9:00(JST)）まで以下を行う
 - (ア)匿名化データファイル CCI_s.csv および機械学習モデルデータファイル DDi_s.json を作成する（s は 1,2,3 の何れか）
 - ✧ 匿名化データファイルを作成するサンプルコードとして [ano.py](#) が、機械学習モデルデータファイルを作成するスクリプトとして [xgbt_train.py](#) が公開されている
 - BBi_s.csv や CCI_s.csv を xgbt_train.py に入力し DDi_s.json を作成できる
 - (イ)ファイル index.txt を作成し、s (1,2,3 の何れか) を書き込む
 - (ウ)CCI_s.csv, DDi_s.json, id.txt, index.txt を zip ファイルにまとめ（zip ファイルのファイル名は任意。例えば submission.zip。フォルダーを含めないこと）、本サイトから提出する（図2参照）
 - ✧ 提出は1日3回まで（毎日、日本時間の 9:00 にリセット）
 - ✧ 提出した zip ファイルが正常に処理されると、図3のようにステータスが”Finished”となり、正常終了して得点が表示される
 - ✧ 正常終了した結果の行をクリックすると（一番左にある ID をデータ ID と呼ぶ）、図4のように”DOWNLOADS”タブにある”Output from scoring step”をダウンロードでき、有用性の得点の詳細を確認できる
 - ✧ 図5のように”LOGS”タブの Scoring Logs のログ表示からも確認できる
 - ✧ 異常終了した場合は、ステータスによっては、”LOGS”タブの Scoring Logs の”stderr”タブにエラー内容が記載される
 - (エ)正常終了したデータ ID を選び、リーダーボードに登録する
 - ✧ 登録方法は、図3の”Add to Leaderboard”の吹き出しが出ているアイコンをクリックすればよい
 - ✧ 期限内であれば何回でも変更できるが、期限時点に登録したデータが最終提出データとして採用されるので注意

Get Started
Phases
My Submissions
Results
Forum
?

予備戦：加工フェーズ
予備戦：攻撃フェーズ
本戦：加工フェーズ
本戦：攻撃フェーズ

?

Number of submissions used for the day
1 out of 5


Number of total submissions used
1 out of 100

Submission upload

Metadata or Fact Sheet

コメント:


Submit as: ?
Yourself



このフィールドをクリックし、提出するzipファイルを選択

図2 本サイトでのデータ提出画面

Submit as: ?
Yourself



Search...
Status






ID # ▾	File name	Date	Status	Score	Detailed Results	
375379	submission_team11_1.zip	2025-09-21 21:58	Finished	36.97		Add to Leaderboard   
375374	submission_14.zip	2025-09-21 21:49	Failed ?			

図3 正常終了イメージ

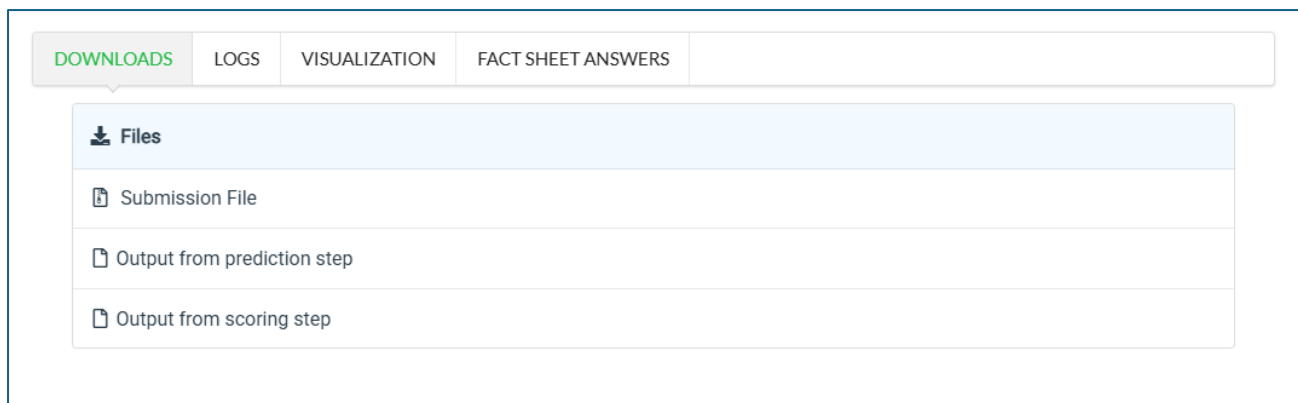


図 4 有用性の得点詳細データのダウンロード画面

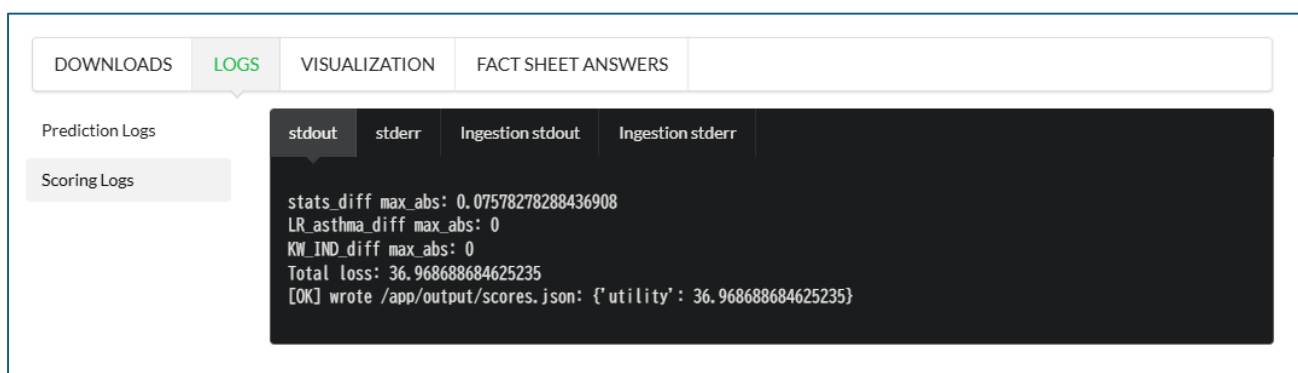


図 5 有用性の得点詳細データの標準出力画面

3.2 ルール（制約条件）

提出データの匿名化データファイル CCI_s.csv および機械学習モデルデータファイル DDi_s.json は以下を満たさなければいけない。

- 匿名化データファイル CCI_s.csv
 - 正常に有用性評価項目 1,2,3 の得点が出力されること
 - ✧ 付録 B に記載の stats_diff.py, LR_asthma_diff.py, KW_IND_diff.py が正常に処理されること
 - BBI_s.csv の形式と同様、ヘッダー行を含む 100,001 行 18 列の CSV ファイルであること
 - ✧ 列番号も BBI_s.csv と同様に、表 1 に従うこと
 - 各データの値域は [columns_range.json](#) で定義された範囲とすること

✧ 表 1 の記載と等しい

✧ 付録 B に記載の `pws_data_format.py` で形式チェックできる（付録 B に記載の `eval_all.py` で、`CCi_s.csv` が制約条件を満たしているかどうかチェックしている）

- 機械学習モデルデータファイル `DDi_s.json`

- 正常に有用性評価項目 4 の得点が出力されること

- ✧ 付録 B に記載の `xgbt_pred.py` が正常に処理されること

- 付録 B に記載の `xgbt_train.py` の出力と形式が等しいこと

- 付録 B に記載の `validate_model_json.py` が正常に処理されること

第 4 章 本戦攻撃フェーズの手順・ルール

4.1 手順

攻撃者（参加チーム） j は以下を行う。

1. 攻撃用データをダウンロードする ※2.4 節に具体的な手順を記載

- ✧ $AAi.csv$, $CCi.csv$, $DDi.json$ (i は 01~24)

2. 期限（10 月 21 日(火)9:00(JST)）まで以下を行う

(ア) 攻撃結果ファイル $FFj.csv$ を作成する

- ✧ $FFj.csv$ は 100,000 行 24 列のデータとする（ヘッダー行なし）

- ✧ i 列目はチーム i の攻撃結果、すなわち i 列目の k 行目は $AAi.csv$ の $k+1$ 行目がメンバーシップ（ $BBi.csv$ に含まれる）と推定すれば”1”を、そうでなければ”0”を記入

- ✧ j 列目は空欄または”0”で埋める

- ✧ $AAi.csv$, $CCi.csv$, $DDi.json$ が未提出のチームが存在した場合は、そのチームの列も空欄または”0”で埋める

- 未提出のチームが存在した場合は事務局がその旨を周知する

- ✧ i 列目のデータを作成するサンプル攻撃コードとして、付録 B の $attack_Ci.py$, $attack_Di.py$, $attack_example.py$ がある

(イ) $FFj.csv$, $id.txt$ を zip ファイルにまとめ（zip ファイルのファイル名は任意。例えば $submission.zip$ 。フォルダーを含めないこと）、本サイトから提出する

- ✧ 提出は期限内までトータルで 8 回まで

- ✧ 3.1 節の説明同様、得点やエラー内容を確認できる

(ウ) 正常終了したデータ ID を選び、リーダーボードに登録する

- ✧ 登録方法は 3.1 節の説明同様

4.2 ルール（制約条件）

提出データの攻撃結果ファイル FFj.csv は以下を満たさなければいけない。

- 100,000 行 24 列のデータであること
- j 列および事務局が指定した列（未提出のチーム番号）は空欄または”0”で埋めること
- j 列および事務局が指定した列以外の列のデータは”1”が 10,000 個、”0”が 90,000 個であること

第 5 章 採点ルール

本コンテストでは各チームの「有用性」「匿名性」「攻撃力」の得点を競う。各得点の採点ルールは 2.5～2.7 節で説明した通りである。最終的には、予備戦の得点 $\times 0.1$ + 本戦の得点 $\times 0.9$ とし、1.4 節で説明した各賞を授与する。ただし予備戦の攻撃力の得点は、上位 5 チーム（本戦では上位 10 チーム）からの得点としているため、点数を 2 倍する。

第 6 章 予備戦からの変更点

予備戦とは異なる点は以下の通り：

- 本戦加工フェーズで本サイトからデータを提出すると、有用性の評価項目 1,2,3,4 の得点および合計得点を確認でき、リーダーボードに登録するとこれらの得点が表示される（予備戦では評価項目 1 のみ）
- 加工データ CCI_k.csv, DDi_k.json の形式をチェックする機能を強化（CCI_k.csv の形式をチェックする [pws_data_format.py](#) を追加し、本サイトからデータ提出した際も自動チェックが起動し、不備があるとエラーを返す）
- 本戦攻撃フェーズで本サイトからデータを提出できる回数は 8 回まで（予備戦では 4 回まで）
- 攻撃力の得点になる総合得点上位チーム数を 10 チームとする（予備戦では 5 チーム）
- 予備戦データと混ざらないようにするため、予備戦で用いていたファイル名に含まれるアルファベット A, B, C, D, F, X, Z を本戦ではそれぞれ AA, BB, CC, DD, FF, XX, ZZ とする
 - 例：AA01_1.csv, BB02_2.csv, CC03_3.csv, DD04_1.json, FF05.csv, XX06_2.csv, ZZ.csv

付録 A Synthea について

A.1 概要

Synthea は、MITRE Corporation が公開提供している、SyntheticMass と呼ばれる合成患者データ群を生成するツールである。SyntheticMass は、実際の患者データ群から得られる疾患や治療の状態遷移モデルや、地域ごとの人口分布、年齢分布、死亡率、医療費用等の統計に基づき生成され、架空の患者データではあるがリアリティーが高く、医療分析等の用途として各種データが公開されている。Synthea は csv 形式を含む様々なフォーマットの合成患者データを出力できる。

A.2 インストールと合成患者データ生成

synthea の [README](#) にインストール方法と合成患者データ生成方法の記載があるので、詳細はこちらを参照されたい。インストールには Java JDK 11 以降が必要。Synthea リポジトリをクローンし、テストスイートをビルドする場合は、次の 3 つのコマンドを順に実行する。

```
git clone https://github.com/synthetichealth/synthea.git
cd synthea
./gradlew build check test
```

マサチューセッツ州の 100 人分の合成患者データを生成して csv 形式で出力する場合は、

```
./run_synthea -p 100 Massachusetts --exporter.format=csv
```

と実行する。なお内部では乱数を用いた処理を行っているため、同じコマンドを繰り返し実行しても毎回異なる結果となる。同一の結果を得たい場合は乱数シードを固定するオプション `-s` を利用する。例：`./run_synthea -s 1234 -p 100 Massachusetts --exporter.format=csv`

A.3 出力データ

Synthea が出力する各種 csv ファイルは表 2 のとおり。

項番	ファイル名	説明
1	allergies.csv	患者のアレルギーデータ
2	careplans.csv	患者ケア計画データ（目標含む）
3	claims.csv	患者の請求データ

4	claims_transactions.csv	請求ごとの明細項目あたりの取引データ
5	conditions.csv	患者の状態または診断データ
6	devices.csv	患者が装着するデバイスのデータ
7	encounters.csv	患者の診察データ
8	imaging_studies.csv	患者の画像メタデータ
9	immunizations.csv	患者の予防接種データ
10	medications.csv	患者の投薬データ
11	observations.csv	バイタルサインや検査レポートのデータ
12	organizations.csv	病院を含むデータ提供機関のデータ
13	patients.csv	患者の人口統計データ
14	payer_transitions.csv	支払者移行データ（健康保険の変更など）
15	payers.csv	支払者組織のデータ
16	procedures.csv	手術を含む患者の処置データ
17	providers.csv	患者ケアを提供する医療従事者データ
18	supplies.csv	医療サービス提供に用いられる資材データ

表 2 Synthea が出力する CSV ファイル一覧

付録 B スクリプト、サンプルコード

本コンテストでは以下のスクリプト、サンプルコードを作成して [Github に公開](#)している。なお引数の記<…>は必須、[…]はオプションを意味する。

- analysis/KW_IND.py
 - 有用性の評価項目 3 の得点を標準出力する
 - 引数：<CCi.csv> ※他にもテスト用のオプションがいくつかある
- analysis/LR_asthma.py
 - 有用性の評価項目 2 の得点を標準出力する
 - 引数：<CCi.csv> ※他にもテスト用のオプションがいくつかある
- analysis/stats.py
 - 有用性の評価項目 1 の得点を標準出力する
 - 引数：<CCi.csv>
- analysis/validate_model_json.py
 - 機械学習モデルデータファイル DDi.json の形式チェックを行い、結果を標準出力する
 - 引数：<DDi.json>
- analysis/xgbt_pred.py
 - 機械学習モデルを記した json ファイルとテスト用 CSV ファイルを読み込み、予測結果をファイル出力する
 - 引数：<DDi.json> <--test-csv XXi.csv> <FFj.csv> [--target TARGET] ※他にもテスト用のオプションがいくつかある
- analysis/xgbt_train.py
 - CSV ファイルを読み込み、XGBoost を用いた予測モデル（機械学習モデル）を生成してデータファイル（json ファイル）を出力する
 - 引数：<BBi.csv（または CCi.csv 等）> <--model-json DDi.json> [--target TARGET] ※他にもテスト用のオプションがいくつかある
- anonymization/ano.py
 - 匿名化データを作成するサンプルコード
 - 引数：<BBi.csv> <CCi.csv> [--seed SEED]
- anonymization/gen_Di.py
 - 機械学習モデルデータを作成するサンプルコード
 - 引数：<BBi.csv（または CCi.csv 等）> <DDi.json>
- anonymization/randomshuffle_rows.py
 - 匿名化の補助的手段として、入力 of CSV ファイルの行データをランダムに置き換える
 - 引数：<CCi.csv> <CCi2.csv>

- attack/attack_Ci.py
 - AAI.csv, CCI.csv を入力として、メンバーシップ推定攻撃結果をファイル出力するサンプルコード
 - 引数：<AAi.csv> <CCi.csv> [-o output.csv]
- attack/attack_Di.py
 - AAI.csv, DDi.json を入力として、メンバーシップ推定攻撃結果をファイル出力するサンプルコード
 - 引数：<AAi.csv> <DDi.json> [-o output.csv]
- attack/attack_example.py
 - attack_Ci.py と attack_Di.py の結果を使って攻撃するサンプルコード
 - 引数：<AAi.csv> <CCi.csv> <DDi.json> [-o output.csv]
- attack/make_attack_submission.py
 - attack_Di.py を使って全チームを攻撃し、提出可能な zip ファイルを作るサンプルコード
 - 引数：<i (自チーム ID)> <input directory> <output directory>
- attack/mia.py
 - attack_Ci.py に継承
- evaluation/KW_IND_diff.py
 - KW_IND.py への 2 入力の結果の差分を標準出力する
 - 引数：<BBI.csv> <CCI.csv> ※他にもテスト用のオプションがいくつかある
- evaluation/LR_asthma_diff.py
 - LR_asthma.py への 2 入力の結果の差分を標準出力する
 - 引数：<BBI.csv> <CCI.csv> ※他にもテスト用のオプションがいくつかある
- evaluation/check_ans.py
 - 匿名性の得点算出用コード
 - 引数：<ZZi.csv> <FFij.csv> ※他にもテスト用のオプションがいくつかある
- evaluation/eval_all.py
 - 有用性の評価項目 1,2,3 の得点をまとめて標準出力する
 - 引数：<BBI.csv> <CCI.csv> [-d (ログ出力)]
- evaluation/gen_ans.py
 - AAI.csv, BBI.csv を入力し、正解データファイル ZZi.csv を出力
 - 引数：<AAi.csv> <BBI.csv> <ZZi.csv>
- evaluation/stats_diff.py
 - stats.py への 2 入力の結果の差分を標準出力する

- 引数：<BBi.csv> <CCi.csv>
- util/check_and_fix_csv.py
 - （予備戦で使用。本戦では使用予定なし）
- util/check_csv.py
 - pws_data_format.py に継承
- util/check_duplicates.py
 - CSV ファイルを入力し、重複行がないかチェックする
 - 引数：<AAi.csv>
- util/columns_range_json.py
 - CSV ファイルを入力し、各列の値域を json ファイルとして出力する
 - 引数：<AAi.csv> [-o columns_range.json]
- util/pws_data_format.py
 - CCi.csv を入力し、形式チェック結果を標準出力する
 - 引数：<CCi.csv>
- util/random_sampling.py
 - CSV ファイルを入力し、ランダムに指定行数を抽出した CSV ファイルを出力
 - 引数：<AAi.csv> <BBi.csv> [-n N (default: 10000)] [--seed SEED]
- util/rev_csv.py
 - （予備戦で使用。本戦では使用予定なし）
- util/unified_synthea.py
 - Synthea で生成した 18 個の CSV ファイルから AAi_k.csv を生成する
 - 引数：<output.csv>