

Original Research

Pharmacist-oriented evaluation of inpatient management of primary hypertension using a problem-based care framework and exploratory predictive modeling

Fayiz Adel Aldheisat  Heba A Khader , Luai Z. Hasoun , Zekrayat J.H. Medras, Muath Alsarafandi, Yazan Alsayed, Ahmad R. Alsayed*. 

Received (first version): 22 October 2025 Accepted: 31 December 2025 Published online: 30 Jul 2026

Abstract

Background: Primary hypertension is a major global risk factor for cardiovascular disease and premature mortality. Despite the availability of diverse pharmacological and non-pharmacological interventions, clinical practice often faces challenges. **Aim:** To evaluate whether integrating structured clinical frameworks, specifically the Medical Problem-Oriented Plan (MPOP), with machine learning (ML) techniques can facilitate a systematic, data-driven assessment of care effectiveness and safety using real-world clinical datasets. **Methods:** This study investigated datasets of inpatients (n = 267 screened, 96 eligible). Clinical data were structured according to the MPOP framework to identify and categorise medical problems. Logistic regression, Bayesian logistic regression with Bernoulli likelihood, and alternative models (negative binomial, zero-inflated models, and gradient boosting) were applied to assess effectiveness and safety. Model performance was evaluated using accuracy, Area Under the Receiver Operating Characteristic curve (AUROC), precision-recall, and cross-validation metrics. **Results:** The effectiveness domain revealed frequent treatment problems: 41% of cases demonstrated suboptimal therapy effectiveness, and 29% reported at least one safety concern. Logistic regression models showed modest discrimination (AUROC = 0.63; accuracy = 61%), limited by small sample size and the presence of zero-inflated variables. **Conclusion:** A structured MPOP framework for pharmacists is feasible for assessing inpatient hypertension care and identifying medication issues related to effectiveness and safety. Predictive modeling showed promise for exploration, but it requires larger datasets before it can be used in real-world practice.

Keywords: primary hypertension; medical problem-oriented plan; medication-related problems; machine learning; predictive modelling; personalised therapeutics.

Fayiz Adel Aldheisat. Department of Clinical Pharmacy and Therapeutics, Applied Science Private University (ASU), Amman 11931, Jordan.

Heba A Khader. Department of Clinical Pharmacy and Pharmacy Practice, Faculty of Pharmaceutical Sciences, The Hashemite University, Zarqa 13133, Jordan.

Luai Z. Hasoun. Biopharmaceutics and Clinical Pharmacy Department, Faculty of Pharmacy, Al-Ahliyya Amman University, Amman-19328, Jordan. l.hasoun@ammanu.edu.jo

Zekrayat J.H. Medras. College of Pharmacy, Amman Arab University, P.O. Box 2234, Amman 11953, Jordan.

Muath Alsarafandi. Faculty of Medicine, Islamic University of Gaza, P.O. Box 108, Gaza, State of Palestine. Malsarafandi02@gmail.com

Yazan Alsayed. Al-Rayhan Medical Center, Amman, Jordan.

Ahmad R. Alsayed*. Department of Clinical Pharmacy and Therapeutics, Applied Science Private University (ASU), Amman 11931, Jordan. a_alsayed@asu.edu.jo

development of primary hypertension is widely recognised as multifactorial, involving complex interactions among behavioural, genetic, and environmental factors^{2,4}.

A meta-analysis of more than a million adults revealed that a 20-mmHg reduction in systolic blood pressure considerably reduces cardiovascular risk, underscoring the significance of controlling systolic blood pressure. In particular, a decrease in the mortality risk from fatal stroke for men aged 40–49 is equivalent to a hazard ratio (HR) of 0.33, which indicates a 67% relative reduction⁴. Moreover, systolic blood pressure has been identified as a stronger predictor of cardiovascular events than diastolic blood pressure, even after accounting for factors such as smoking and age⁵. These findings highlight the need for improved hypertension management strategies, particularly those targeting systolic blood pressure, to reduce cardiovascular morbidity and mortality. Data-driven approaches, such as predictive modelling, may provide new insights into optimising such interventions.

The proper care of patients requires a proper assessment of the clinical plan. Thus, different medical frameworks could be applied, such as problem-oriented frameworks, for example, the medical problem-oriented plan (MPOP) and its associated tools^{6,7}. It is based on a problem-oriented approach.

A problem-oriented care framework improves chronic disease care strategies by systematically identifying and categorising drug-related and care strategy issues and linking them to a specific plan^{6–8}. Comparable structured pharmaceutical care models, such as Comprehensive Medication Management⁹, have consistently

INTRODUCTION

Hypertension is a chronic disease characterised by elevated systolic and/or diastolic blood pressure^{1,2}. Numerous demographics, behavioural, and clinical factors contribute to its onset and progression, including age, sex, lifestyle practices, and comorbid conditions such as chronic kidney disease³. Hypertension is classified as either primary (essential) or secondary, with primary hypertension accounting for approximately 95% of cases^{1–3}. The



resolved up to nearly half of identified drug therapy problems and generated considerable cost savings in chronic illness populations. By categorising patient issues into standardised categories, these approaches generate highquality datasets that are ideally suited for machine learning (ML). In turn, ML algorithms can influence this structure to predict outcomes and guide personalised treatment plans in hypertension care.

Even though there have been improvements in antihypertensive pharmacotherapy, structured assessments of pharmaceutical care for inpatients continue to be insufficient. In hospital settings, fragmented documentation and reactive medication adjustments may obscure unresolved medication-related problems. To improve clinical responsibility and continuity of care, we need structured assessment tools for pharmacists that can systematically identify problems related to indications, effectiveness, safety, and patient-related issues.

The systematic collection of clinical problems using frameworks such as MPOP not only improves care coordination but also enables advanced predictive modelling. ML can process these standardised datasets to link multifactor relationships influencing hypertension outcomes, from demographic and metabolic indicators to treatment adherence and lifestyle factors. Unlike traditional methods, these ML models adapt to data patterns, enhancing our understanding of blood pressure control and treatment response. In chronic disease management, such tools improve care strategies and optimise medication regimens^{10,11}. Their potential is maximised when trained on rich, structured datasets, ensuring that the outputs are clinically meaningful and aligned with known care strategies or when assessing them.

Logistic regression, the model of choice for this study, remains a standard tool for predicting hypertension outcomes. As a binary classification model, it estimates the likelihood of a desired outcome for everyone while maintaining interpretable results and allowing clinicians to investigate how specific variables influence the results¹². When combined with structured, problem-oriented clinical documentation, logistic regression models can support reproducible care-strategy assessments and be integrated into decisionsupport tools for routine hypertension management.

In this context, the MPOP framework is not merely a method for organizing data; it is also a structured approach to assessing pharmaceutical care that helps pharmacists systematically identify, categorize, and resolve medical problems. This supports pharmacists' optimization of antihypertensive therapy.

Statement of the Problem

Healthcare now generates high-volume, high-variety data that exceeds the capacity of manual review and conventional low-dimensional analyses to yield reliable, patient-level insight^{13,14}. Modern ML methods applied to electronic health records (EHR)

and related data have shown the ability to uncover complex, clinically meaningful patterns that enhance prediction and decision support¹⁵. Realising this potential requires rigorous data curation and validation to address missingness, bias, and heterogeneity; hence, the development and evaluation of robust, data-driven models for hypertension management are warranted¹⁵.

Hypertension care is moving toward individualised, risk-based decision-making that relies on extracting reliable signals from extensive, heterogeneous EHR data. Rigorous modelling can (i) reveal treatment-response heterogeneity, (ii) quantify benefit-harm trade-offs, and (iii) provide calibrated, decision-useful risk estimates beyond conventional low-dimensional analyses^{16,17}.

This study aimed to systematically evaluate medical care strategies for the management of primary hypertension across inpatients by integrating a problem-oriented framework with advanced ML techniques to improve treatment effectiveness, safety, and patient-centred outcomes.

The objectives of this research were to:

1. Apply a structured, pharmacist-focused MPOP framework to assess inpatient hypertension care.
2. Determine and classify issues pertaining to medications in the areas of indication, efficacy, and safety.
3. Examine the feasibility of using predictive modeling to generate hypotheses for clinical decision support.

METHODS

Study Design

This was a retrospective observational study based on the medical records of inpatients (Prince Hamzah Hospital) with primary hypertension during 2022 and 2023. The study was approved by the Institutional Review Board (IRB) of Prince Hamzah Hospital (6-11-2021-129).

Data Collection

Both hard and soft copies of files were considered. The inclusion Criteria for this study were adults aged ≥ 30 years with primary hypertension. Exclusion Criteria were cancer, secondary hypertension, pregnancy, and improper report exports (missing pharmacy/medications report, missing blood pressure readings, or missing diagnosis). Data were formatted into a Python Pandas dataframe for later assessment. The structured assessment reflects real-world clinical pharmacy documentation practices, although retrospective extraction limited the completeness of some pharmacist-relevant variables.



Assessment and Categorisation of MPOPs

All medical problems reported in the assessment step were pooled and then classified into the following categories: Indication, Effectiveness, Safety, Patient, and Miscellaneous, using the MPOP tool as detailed in previous research^{6-8,18}. Under the guidance of a senior clinical pharmacist, skilled clinical pharmacy staff carried out the MPOP coding and classification. Validated MPOP definitions served as a guide for classification decisions in order to ensure uniformity and compliance with standards for pharmaceutical care documentation. Figure 1 illustrates the workflow for transforming data into MPOP categories.

Data Preprocessing

Data was normalised into Pandas dataframes, and the MPOP domains were encoded using binary encoding with simple yes/no flags (1 for presence and 0 for absence).

Hard-copy inpatient general reports were reviewed manually to identify cases of primary hypertension, then entered into a Pandas dataframe via a Flask web application form. Blood pressure control was classified using AHA/ACC 2017². Thresholds from the last three in-hospital calendar days (or fewer for short stays) to limit the influence of early acute fluctuations due to infection.

Electronic reports (also general, not limited to hypertension) were provided read-only. Data collection was through automated Python scripts that queried the underlying XML structure and used regex/rule-based mapping to extract vitals, labs, medications (from the pharmacy section or medical orders), and diagnoses into the required format.

Files ready for analysis were stored as serialised Pandas objects (.pkl) for reproducible loading. All preprocessing steps were performed in Python; full details of the environment are in the Appendix.

Machine Learning

A results-oriented workflow was applied in this study, with data exploration in the initial stage for determining the model workflow. Simple summarisation and visualisation were researched to become familiar with data structures, and model specifications were formulated based on them. Early performance metrics were used to refine preprocessing operations, variable selection, and model parameters. Clinical interpretability and prediction accuracy were monitored throughout.

Datasets were examined using descriptive statistics (i.e., medians and counts), as well as plots of their distributions, to confirm data ranges and identify outliers. Normality testing was only performed when required to guide imputation decisions. Associations of candidate predictors with clinical outcomes were examined to confirm they were relevant to

MPOP framework domains.

The binary endpoints related to the MPOP domains were modelled using logistic regression classifiers. The categorical predictors were expressed in binary flag format as needed.

The preprocessing procedure was tailored to the needs of each dataset rather than applying a generalised pipeline. Imputation or deletion of missing data was done based on the importance of the variable, the extent of missingness, and clinical judgment. Ranking of clinical utility of features within MPOP categories came first, but correlations of features on an empirical basis, with outcomes, were also considered.

Models' performance has been evaluated using measures such as average precision, accuracy, recall, F1-score, and confusion matrices. Validation has been performed using k-fold cross-validation, along with a separate test set split from the beginning.

All analysis was in Python (version 3.13.7), using open-source libraries, statistical modelling, and Bayesian inference packages. The environment was managed via Anaconda, and modelling took place in IPython notebooks, while data entry for outpatients was done in the IPython Shell.

RESULTS

From a pharmaceutical care perspective, the analysis focused on the prevalence of medical problems and their distribution across effectiveness and safety domains.

The exported reports had 267 cases, 151 (56.55%) reports were provided in electronic format and 116 (43.45%) were provided in paper format. Of the 267 reports, 97 (36.33%) met the criteria and were included. From the included reports, 49 (50.5%) were from the paper format, and 48 (49.5%) from the electronic format.

The included reports (N=97), stratified by discharge blood-pressure status into Uncontrolled (n=50, 52%) and Controlled (n=43, 44%) groups, 4 (4%) reports lacked blood pressure readings but were included to preserve sample representativeness.

Table 1 presents baseline demographic characteristics and antihypertensive drug exposure by discharge blood-pressure status. Mean age did not differ between groups (61.66 ± 11.52 vs 62.21 ± 13.30 years; SMD = -0.04; $p = 0.833$). Sex distribution differed between groups (Cramér's V = 0.37; $p = 0.002$). No statistically significant differences were observed in class-level medication use: ACEI (24.0% vs 32.6%; $p = 0.493$), ARB (36.0% vs 20.9%; $p = 0.172$), thiazide (18.0% vs 11.6%; $p = 0.571$), dihydropyridine CCB (38.0% vs 34.9%; $p = 0.924$), and β -blocker (36.0% vs 27.9%; $p = 0.542$). Regimen complexity showed a non-significant trend toward greater multi-drug therapy in the Uncontrolled group (distribution across 0/1/2/ ≥ 3 agents: 13/11/17/9 vs 8/20/10/5; $\chi^2 p = 0.099$; Cramér's V = 0.26).



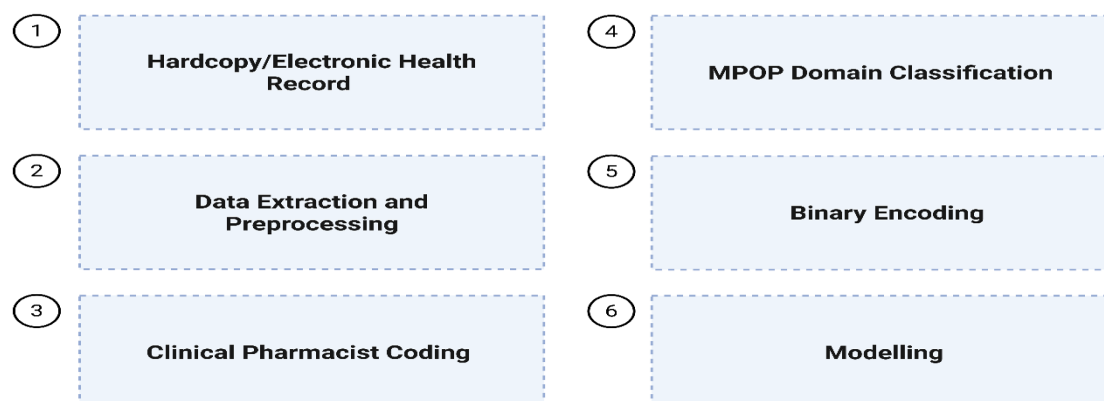


Figure 1.Workflow of data transformation into MPOP categories

Between-group comparisons use Welch's t-test or the Mann-Whitney U test for continuous data, and χ^2 or Fisher's exact tests for categorical data. Effect sizes are reported as Cohen's d (continuous) and ϕ /Cramér's V (categorical), with standardized mean difference (SMD) provided as a measure of baseline imbalance.

Respiratory complaints predominated, with shortness of breath the most common presenting feature. Chest pain and fatigue were also frequent, whereas headache, dizziness, and palpitations were less commonly documented; epistaxis and tinnitus were rare. The comorbidity profile was characterized by a high prevalence of diabetes mellitus, with more modest rates of chronic kidney disease and lower frequencies of heart failure and prior cerebrovascular disease (stroke/transient ischemic attack (TIA)). Figure 2 shows the top prevalence of comorbid conditions, and Figure 3 shows the top prevalence of presenting symptoms.

The largest out-of-interval burdens were observed for total protein (87.3% outside; 58.2% below; 29.1% above; n=55) and uric acid (83.6% outside; 69.1% below; 14.5% above; n=55). Among electrolytes/hematology, calcium was outside the reference range in 47.4% (predominantly lower; n=78), and hematocrit was outside the reference range in 46.9% (lower; n=96). WBC showed a right-shift with 36.1% above interval (40.2% outside overall; n=97). Chloride and sodium displayed bidirectional deviations (Cl 38.1% outside: 19.6% below/18.6% above; Na 35.1% outside: 23.7% below/11.3% above; both n=97). eGFR was <90 in 34.0% (n=97). Hepatic enzymes showed upper-tail elevations (ALT 26.0% above, n=96; AST 22.9% above, n=96). Potassium remained largely centered with 23.7% outside (9.3% below/14.4% above; n=97). Bars in Figure 4 are ordered by total outside-interval share (below+above), facilitating comparison of abnormality burden across assays.

Modelling Results

In the inpatient dataset (N = 97), MPOP codes (problems) were not observed (no problems in the report) in 33 (34%) records and were coded as 0 (no problem) for analysis. The other remaining records, the number of problems per case ranged from 1 to 4 (one problem: 46 cases (73.43% of cases had any problem); two: 11 (17.2% of cases had any problem); three: 6 (9.4% of cases had any problem); four: 1 (1.56% of cases had any problem)), with a median [IQR] of 1.0 [1.0]. 34 cases were reported as Data Insufficient Information (V-E-a) and/or Data Documentation Error (V-E-b); within strata this corresponded to 24/46 (52.17%) of cases that had one problem, 5/11 (45.45%) of cases that had two problems, 4/6 (66.67%) of cases that had three problems, and in the one case that had four problems, it had two codes for Data Insufficient Information and Data Documentation Error. For Indication domain problems, there was 1 indication error in the cases that had one-problem code (2.17%), 4 indication errors in cases that had two problems (36.36%), 2 indication errors in the cases that had three-problems (33.33%), and 0 indication errors in the cases that had four problems; Patient-related problems were not reported.

Effectiveness Model Results

Missing-value percentages were calculated as shown in Table 2. The threshold for missing values was set at 50% of the total values, and due to limitations in proper imputation and the high missing value percentage, they were dropped from the dataset.

After deleting all missing values and their related rows, and selecting only variables related to treatment effectiveness, only 64 cases remained suitable for modelling. MPOPs columns were encoded to fit the model's prediction needs, as they

Table 1. Baseline demographic characteristics and antihypertensive drug exposure by discharge blood pressure status					
Variable	Uncontrolled	Controlled	Effect Size	SMD	p-value
Age	61.66 ± 11.52	62.21 ± 13.30	Cohen d = -0.04	-0.04	0.833
Gender	Female: 23 (46.0%); Male: 27 (54.0%)	Female: 21 (48.8%); Male: 23 (51.2%)	Cramér V = 0.37	-	0.002
ACEI (any use)	Yes: 12 (24.0%), No: 38 (76.0%)	Yes: 14 (32.6%), No: 29 (67.4%)	Phi = 0.07	-0.19	0.493
ARB (any use)	Yes: 18 (36.0%), No: 32 (64.0%)	Yes: 9 (20.9%), No: 34 (79.1%)	Phi = 0.14	0.33	0.172
Thiazide (any use)	Yes: 9 (18.0%), No: 41 (82.0%)	Yes: 5 (11.6%), No: 38 (88.4%)	Phi = 0.06	0.18	0.571
DHP CCB (any use)	Yes: 19 (38.0%), No: 31 (62.0%)	Yes: 15 (34.9%), No: 28 (65.1%)	Phi = 0.01	-0.06	0.924
BB (any use)	Yes: 18 (36.0%), No: 32 (64.0%)	Yes: 12 (27.9%), No: 31 (72.1%)	Phi = 0.06	-0.17	0.542
Regimen (agents) distribution	0: 13; 1: 11; 2: 17; ≥3: 9	0: 8; 1: 20; 2: 10; ≥3: 5	Cramér V = 0.26	-	0.099

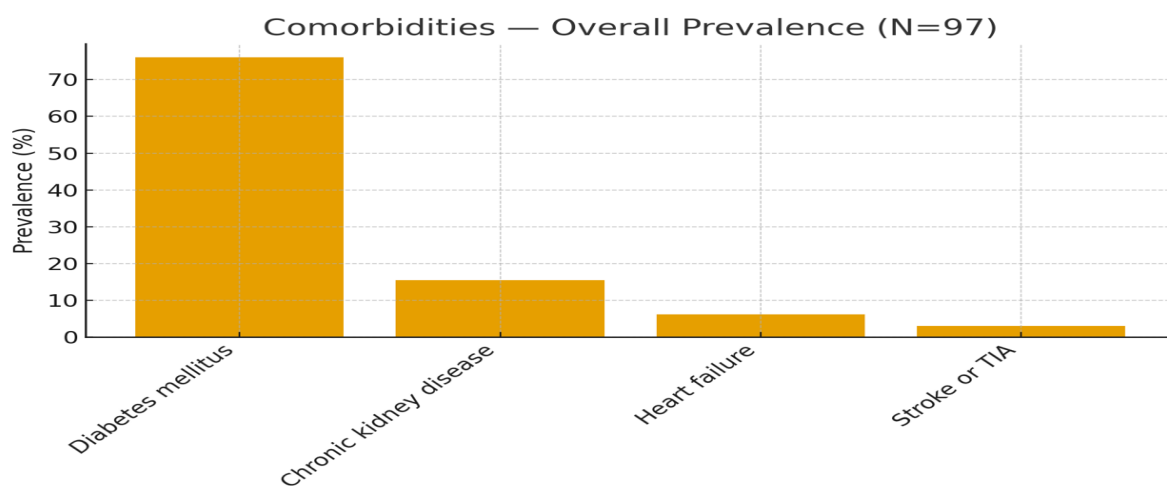


Figure 2. Inpatients data description: top prevalence of comorbid conditions

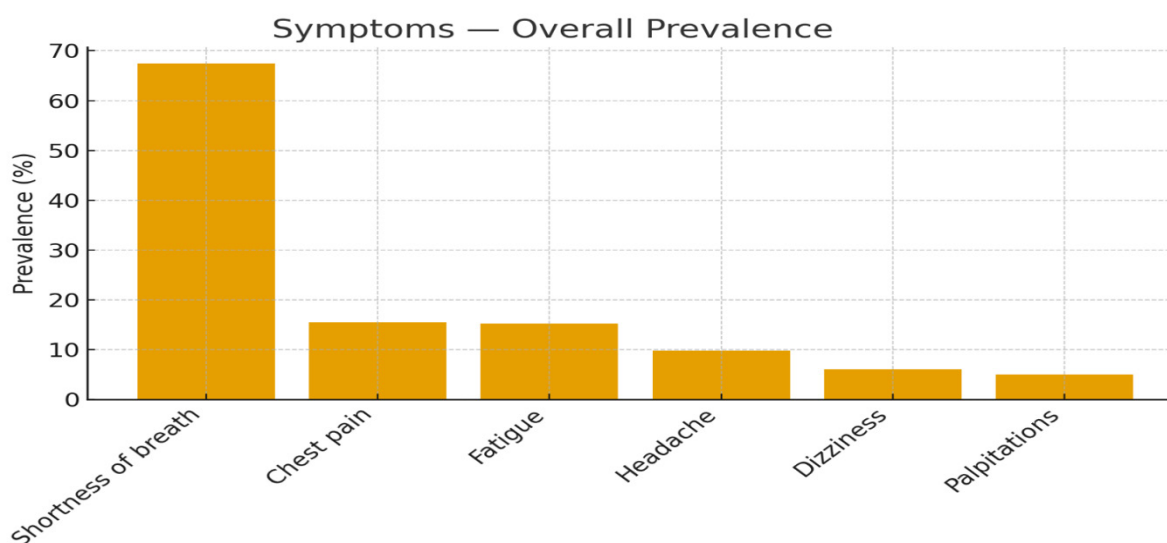


Figure 3. Inpatients data description: top prevalence of presenting symptoms

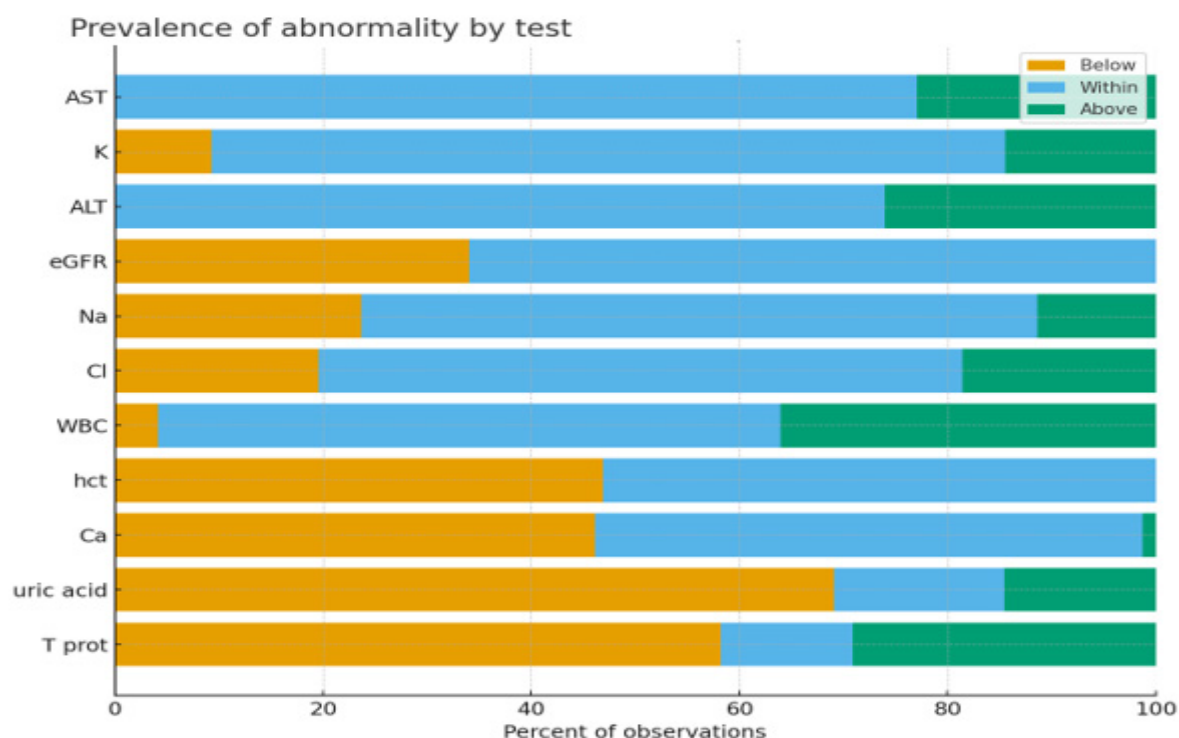


Figure 4. Inpatients data description: laboratory tests abnormality prevalence

contained MPOPs codes from other domains. To limit inflation in the column to 0, it was encoded only with the effectiveness domain, without its subcategories. If the case had any effectiveness problem, it was translated to the effectiveness domain general error, and this column was the y column. Logistic Regression model used with train-test splitting of 33 random seeds, and the size of the testing dataset was 25% of the original dataset.

After model training and testing, both the classification report and the confusion matrix were generated, as shown in Table 3 and Table 4, respectively.

The logistic model showed strong performance on the held-out subset (with accuracy, precision, recall, and F1 all equal 1.00; a confusion matrix showing 8/8 correct classifications per category; test size = 16). Apparent perfect performance in small test subsets likely reflects complete separation and small sample size rather than true clinical discrimination.

The multiclass testing was for three labels over 9 tests. Per-class results were class 0 - precision 0.00, recall 0.00, F1 0.00 (support = 1); class 1 - precision 0.00, recall 0.00, F1 0.00 (support = 2); class 2 - precision 0.80, recall 0.67, F1 0.73 (support = 6). Table 5 shows classification reports.

Safety Model Results

Distributions and missingness of variables were reviewed to reduce the noise from low-information fields before modeling. Symptom indicators were either mostly zero or largely missing,

situations where estimates in rare-outcome contexts can be unstable. While shortness of breath had relatively more positive entries, heterogeneity in recording and the desire for a consistent handling policy across symptom indicators led to the exclusion of all symptom variables from the safety model. Table 6 shows the symptoms' variables after deleting rows that were known from the previous model to have other missing values.

The resulting analytic dataset consisted of 36 columns, including 6 outcome indicators and 30 predictors. The number of positive (minority) cases was very low across all three safety outcomes, so the model did not correctly identify any in any split. As a result, overall accuracy only reflects how well the model predicts the majority class and does not show how well it detects the minority cases. Table 7 shows the safety model's results.

DISCUSSION

Of the inpatients in this cohort, almost 2 out of 5 patients required therapeutic optimization to achieve appropriate blood pressure control, and roughly 1 in 3 required improved safety monitoring, medication reviews, or treatment modifications. These results highlight how common medication-related issues are among hospitalized patients with primary hypertension and highlight the necessity of systematic, pharmacist-led evaluation to enhance treatment efficacy and safety results.

The modelling outputs in this study should be interpreted as exploratory and hypothesis-generating rather than diagnostic or decision-making tools.



Table 2. Percentage of missing data in the effectiveness model	
Variable name	Missing percentage
Aggravating Factors	97.94%
Alleviating Factors	97.94%
AV nicking	96.91%
Duration of HTN since diagnosis (years)	97.94%
Surgeries	86.60%
No. of previous hospitalization	87.63%
Vaccination	96.91%
Family History	92.78%
Smoking Status	89.69%
Alcohol	91.75%
Interference with work	98.97%
Interference with daily chores	88.66%
Impact on quality of life	88.66%
CO2	100.00%
BUN	100.00%
Mg	96.91%
Weight (kg)	92.78%
Height (m)	94.85%
BMI	94.85%
BMI Categories	94.85%
IBW	94.85%

Table 3. Effectiveness model's classification report				
Class	Precision	Recall	F1-score	Support
0	1	1	1	8
1	1	1	1	8
Accuracy			1	16
Macro avg	1	1	1	16
Weighted avg	1	1	1	16

Table 4. The effectiveness model's confusion matrix		
Actual \ Predicted	0	1
0	8	0
1	0	8

Table 5. Multiclass effectiveness model classification report				
Label	Precision	Recall	F1-score	Support
0	0	0	0	1
1	0	0	0	2
2	0.8	0.67	0.73	6

Table 6. Safety model's variables missingness				
Variable	Levels observed	Positives (1)	Negatives (0)	Notes / Action
Tinnitus	{0, NaN}	0	—	No 1's observed; removed
Epistaxis	{0, NaN}	0	—	No 1's observed; removed
Dizziness	{0,1, NaN}	5	77	Rare positives; removed
Headache	{0,1, NaN}	8	74	Rare positives; removed
Chest Pain	{0,1, NaN}	13	71	Rare positives; removed
Palpitations	{0,1, NaN}	4	76	Rare positives; removed
Fatigue	{0,1, NaN}	12	67	Rare positives; removed
Shortness of breath	{0,1, NaN}	62	30	Directionally mixed signal; removed for consistency across symptoms
Nausea / Vomiting	{0,1, NaN}	3	81	Rare positives; removed
Blurred Vision	{0,1, NaN}	1	82	Rare positives; removed

Table 7. Safety model's results				
Outcome	Confusion Matrix (TN FP / FN TP)	Accuracy	Macro F1	Positive class F1
mpop_iii_a*	[[13, 3], [3, 0]]	0.68421	0.40625	0
mpop_iii_d_a^	[[18, 1], [0, 0]]	0.94737	0.48649	0
mpop_iii_e_b*	[[17, 2], [0, 0]]	0.89474	0.47222	0
*: Safety Error code, the drug(s) need(s) to be replaced.				
^: Safety Error code, there is a drug-drug interaction.				
*: Safety Error code, there is a need for additional tests or further investigation.				

Among the included inpatients, 41% demonstrated suboptimal therapeutic effectiveness, suggesting that hospitalized patients often face complex comorbidities, acute clinical instability, and fragmented follow-up care that limit optimal blood pressure control. The modest predictive performance of the inpatient models (AUROC = 0.63) underscores the challenge of managing hypertension in acutely ill patients where polypharmacy, acute renal or cardiovascular complications, and incomplete documentation hinder systematic care evaluation. Clinically, this indicates that hospitalised patients require closer monitoring, individualised treatment adjustments, and proactive coordination between physicians, pharmacists, and nursing staff to ensure continuity of hypertension control during and after discharge.

Drug-related safety issues emerged as another significant dimension, with 29% of inpatients experiencing at least one adverse safety event. Clinically, this underscores that beyond lowering blood pressure, the choice and monitoring of therapy must balance efficacy with safety, particularly in populations with diabetes, renal impairment, or polypharmacy. The fit of zero-inflated negative binomial and gradient boosting models indicates that structured frameworks can better capture these clinically relevant risks. Still, more importantly, the findings highlight how safety surveillance should be embedded in daily practice. Clinicians should implement systematic medication reviews, encourage patients to report side effects, and use risk-stratification tools to personalise therapy. This aligns with the broader move toward precision therapeutics, where hypertension management is not only about reaching numeric targets but also about minimising harm, maximising adherence, and improving long-term cardiovascular outcomes¹⁹.

Effectiveness domain modelling proceeded in two parts. In the binary setting (logistic link; test $n = 16$), a perfect prediction for one class (8/8), implying (quasi-) complete separation. This specification follows population-level hypertension research that treats blood-pressure control as a binary endpoint and estimates multivariable (including mixed effects) logistic models. By contrast, the inpatient multiclass task (three codes; test $n = 9$) showed unstable class-wise recovery and limited discrimination: two rare categories (1 and 2 cases) were not recovered at all (recall = 0; $F1 = 0$), while only the predominant category (6 cases) was partially recovered (precision ≈ 0.80 ; recall ≈ 0.67 ; $F1 \approx 0.73$). Aggregate performance was low (micro-averages recall ≈ 0.44 ; micro- $F1 \approx 0.57$; macro- $F1 \approx 0.24$; weighted- $F1 \approx 0.48$), illustrating the expected divergence between instance-weighted micro averages and label-honest macro averages under severe imbalance. Given $n=16/n=9$, these are hypothesis-generating; not suitable for patient-level decisions. Results are best treated as exploratory signals pending larger cohorts, refined endpoints, and re-estimation with appropriate regularization and calibration assessment.

In the single-split test, two endpoints (mpop-III-D-a (Safety code, drug-drug interactions), mpop-III-E-b (Safety code, needs additional tests)) had no observed positives, and the third

(mpop-III-A) had three positives (treatment replacement), all missed at the pre-specified threshold (TN=13, FP=3, FN=3, TP=0). Under these conditions, balanced accuracy ≈ 0.41 ($=\frac{1}{2}$ ·specificity) and macro- $F1 \approx 0.41$ ($F1_{pos}=0$; $F1_{neg} \approx 0.81$), indicating that only the majority (no-event) class was effectively learned.

The pattern accords with a single-center, EMR-integrated fall-risk model evaluated on temporal hold-out, where sensitivity was highly threshold- and prevalence-dependent while specificity remained comparatively stable²⁰. However, given the present rarity and small test split, these estimates are unstable and non-actionable for patient-level decisions.

Safety Code III-A (treatment replacement) showed complete minority-class failure (0/3 detected), i.e., zero sensitivity with simultaneous FNs and FPs; although overall accuracy appeared moderate ($\sim 68\%$), macro-averaged performance was poor, indicating learning of the majority (no-event) class only. For III-D-a (drug-drug interactions) and III-E-b (needs repeated tests), test splits contained no positives (support=0; precision/recall undefined for the positive class), so high accuracies ($\sim 95\%$ and $\sim 89\%$) were non-informative and reflected trivial majority-class performance rather than true discrimination. Overall, severe class imbalance and event scarcity predominated, resulting in non-actionable outputs. These models should not guide patient-level safety decisions because results are hypothesis-generating²¹.

The analysis was constrained by limited positive events and small sample sizes over multiple endpoints. In these conditions, performance measures and parameter estimates are high-variance, and fold-to-fold variation becomes larger, making model or threshold selection seem erratic. Extremely small test splits (e.g., $n=10-20$) are especially susceptible to optimistic or stochastic point estimates (including seemingly perfect confusion tables) and broad uncertainty about the magnitude of the effects. The practical consequences are diminished power to detect small associations, increased vulnerability to data-driven specification decisions, and a tendency toward simple, regularized models to moderate overfitting.

Zero inflation in count outcomes also complicate estimation and evaluation. When the data abound in zeros but contribute little to separating the “structural zeros” from the “sampling zeros” and thus induce mixture models, identifiability and numerical stability are compromised, the argument goes. Aside from numerical stability of estimation, precision–recall–based summary measures, even when stable estimation dominates, rely overmuch on performance under severely imbalanced conditions; AUROC and accuracy do the same and are also sensitive to small variations in the relatively small positive class²². Net outcomes are blunted or misleading rate estimation in the count direction, inability to detect the inflation process, and conservative inference when shrinkage prior distributions are used to protect against false signals.

The inpatient source files held sparse clinical detail, and this generated risks of information bias. Limited granularity (e.g., omission or sporadic recording of symptoms, exposures, or laboratory testing) leads to nondifferential misclassification, which attenuates associations, or differential misclassification, which biases them. Furthermore, monitoring that depends on exposure (e.g., extensive laboratory follow-up for some drug classes) leads to detection asymmetries and collider structures, which prevent causal interpretation of the model's coefficients. Overall, these factors bias estimation toward weak or fluctuating signals, impede discrimination when the inpatient safety endpoints are scarce, and diminish generalizability beyond the recorded reporting workflow.

This study has various limitations. It was retrospective and single-centred, limiting external validity. Documentation inconsistencies in the records may have led to underdiagnosis of medical problems. These limitations reflect real-world challenges in the clinical pharmacy practice and highlight the need for prospective structured data capture.

Pharmacy Practice Implications

Improved medication reconciliation at admission and discharge is one way structured frameworks, such as MPOP, can enhance inpatient pharmaceutical care.

1. Enabling thorough antihypertensive dosage optimization

2. Identifying inadequacies in laboratory monitoring and drug-drug interactions
3. Strengthening physician-pharmacist cooperation
4. Ensuring the continuity of care planning once discharge

Integrating structured assessment into routine pharmacy workflows may reduce preventable medical problems among hospitalized patients with hypertension.

CONCLUSION & RECOMMENDATIONS

This study demonstrates that using a structured pharmacist-oriented MPOP framework to assess inpatient hypertension management is feasible. The necessity of a systematic pharmacist-led medication review in hospital settings was highlighted by a substantial proportion of patients who displayed medication-related safety and effectiveness issues.

Before being used, predictive modeling needs to be validated in larger, multi-center cohorts, but it has shown exploratory promise as a clinical decision-support adjunct. To improve inpatient management of hypertension, future studies should focus on combining prospective modeling frameworks with structured pharmaceutical care documentation.

References

1. Campbell NR, Burnens MP, Whelton PK, et al. 2021 World Health Organization guideline on pharmacological treatment of hypertension: Policy implications for the region of the Americas. *Lancet Reg Heal*. 2022;9.
2. Whelton PK, Carey RM, Aronow WS, et al. 2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA guideline for the prevention, detection, evaluation, and management of high blood pressure in adults: a report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. *J Am Coll Cardiol*. 2018;71(19):e127-e248.
3. James PA, Oparil S, Carter BL, et al. 2014 evidence-based guideline for the management of high blood pressure in adults: report from the panel members appointed to the Eighth Joint National Committee (JNC 8). *Jama*. 2014;311(5):507-520.
4. Sharma L, Sharma V. Effect of dash diet on the old age subjects suffering from hypertension.
5. Gu D, Kelly TN, Wu X, et al. Blood pressure and risk of cardiovascular disease in Chinese men and women. *Am J Hypertens*. 2008;21(3):265-272.
6. Alsayed AR. Illustrating how to use the validated alsayed_v1 tools to improve medical care: a particular reference to the global initiative for asthma 2022 recommendations. *Patient Prefer Adherence*. Published online 2023:1161-1179.
7. Alsayed AR, Al-Dulaimi A, Alnatour D, Awajan D, Alshammari B. Validation of an assessment, medical problem-oriented plan, and care plan tools for demonstrating the clinical pharmacist's activities. *Saudi Pharm J*. 2022;30(10):1464-1472.
8. Mansoor AR, Abed A, Alqudah A, Alsayed AR. Assessment of Medical Care Strategies for Primary Hypertension in Iraqi Adults: A Hospital-Based Problem-Oriented Plan. *Patient Prefer Adherence*. Published online 2025:1317-1335.
9. Chung TK, Jeon Y, Hong Y, Hong S, Moon JS, Lee H. Factors affecting the changes in antihypertensive medications in patients with hypertension. *Front Cardiovasc Med*. 2022;9:999548.
10. Nguyen HM, Anderson W, Chou SH, et al. Predictive Models for Sustained, Uncontrolled Hypertension and Hypertensive Crisis Based on Electronic Health Record Data: Algorithm Development and Validation. *JMIR Med Inform*. 2024;12:e58732.
11. Si F, Liu Q, Yu J. A prediction study on the occurrence risk of heart disease in older hypertensive patients based on machine learning. *BMC Geriatr*. 2025;25(1):27.
12. Chowdhury MZI, Naeem I, Quan H, et al. Prediction of hypertension using traditional regression and machine learning models: A systematic review and meta-analysis. *PloS One*. 2022;17(4):e0266334.
13. Beam AL, Kohane IS. Big data and machine learning in health care. *Jama*. 2018;319(13):1317-1318.
14. Rajkomar A, Dean J, Kohane I. Machine learning in medicine. *N Engl J Med*. 2019;380(14):1347-1358.



15. Topol EJ. High-performance medicine: the convergence of human and artificial intelligence. *Nat Med*. 2019;25(1):44-56.
16. Collins GS, Dhiman P, Ma J, et al. Evaluation of clinical prediction models (part 1): from development to external validation. *Bmj*. 2024;384.
17. Karmali KN, Lloyd-Jones DM. Global risk assessment to guide blood pressure management in cardiovascular disease prevention. *Hypertension*. 2017;69(3):e2-e9.
18. Alsayed AR, Hasoun L, Al-Dulaimi A, et al. Evaluation of the effectiveness of educational medical informatics tutorial on improving pharmacy students' knowledge and skills about the clinical problem-solving process. *Pharm Pract*. 2022;20(2):2652.
19. Munger MA. Polypharmacy and combination therapy in the management of hypertension in elderly patients with co-morbid diabetes mellitus. *Drugs Aging*. 2010;27(11):871-883.
20. Nishimoto M, Griffin KA, Wynne BM, Fujita T. Salt-sensitive hypertension and the kidney. *Hypertension*. 2024;81(6):1206-1217.
21. Fieggen J, Segal B, Walker EC, et al. Navigating Severe Class Imbalance in Population Cohort Data. In: *IEEE*; 2025:1-6.
22. Zhou Z, Li D, Huh D, Xie M, Mun E. A simulation study of the performance of statistical models for count outcomes with excessive zeros. *Stat Med*. 2024;43(24):4752-4767.

Appendix

Environment

Table 1. Operating System Environment.

Component	Version
OS	Arch Linux
Python	3.13.7 GCC 15.2.1
Interactive tools	JupyterLab 4.4.5 (inside PyCharm 2025.1.2), IPython 9.4
Package manager	Anaconda

Table 2. Python Libraries

Library	Version	Used for
pandas	2.2.3	Data ingestion, cleaning, joins, type handling, feature tables.
seaborn	0.13.2	Quick EDA plots (distributions, ECDF/violin) during exploration.
numpy	2.3.1	Vectorized Numerics, RNG seeds, array ops underpinning modeling.
matplotlib	3.10.3	Plotting needed graph
scikit-learn	1.7.0	Preprocessing (scalers/encoders), CV splits, metrics, simple baselines.
joblib	1.4.2	Parallelism and caching for CV and feature generation.
scipy	1.15.3	Statistical utilities, distributions, optimization helpers.
ArviZ	0.22.0	Bayesian diagnostics, posterior summaries, PPCs, credible intervals.
PyMC	5.23.0	Bayesian modeling (e.g., Bernoulli-logit, ZINB) and MCMC/NUTS sampling.
statsmodels	0.14.4	Classical GLM/NB fits and sanity-check estimates.
patsy	1.0.1	Formula/design-matrix construction (especially with statsmodels).
PyTensor	2.31.7	Computational graph / autodiff backend used by PyMC.
dill	0.3.9	Safe serialization of Python objects (traces, folds, artifacts).

6.2.2 Hardcopy Reports Handling

This appendix details the manual entry process, backend validation, derivations (eGFR, Body Mass Index (BMI), Ideal Body Weight (IBW)), and persistence that can be used in inpatient paper report conversion to an electronic dataset.

No identifying details were recorded: No national identity numbers, no unique identifiers, no names were stored in the computer dataset. Key record: Each report had a clear ID (handwritten and utilized the computer key within the electronic record) in such a way that the original record could be accessed directly whenever needed without keeping the patient's personal information within the dataset.

A simple Flask-based web form used to input data manually from the reports then the data was directly mapped to study variables previously defined. Client-side validation: UI limited the input by type (numeric, date, categorical).

Submission process: On submit, the form serialized the data into JavaScript object notation (JSON) and sent a POST request to the backend. Where JSON is a data format used to handle the data. Figure 6.1 shows a screenshot for the form.

Figure 1. Screenshot for the form used to enter the data.

On the backend, incoming JSON was parsed into a Python dictionary with predefined keys. Validation pass:

- Missing fields check: if any data were missing, the backend process would reject the post request with an error message of the missing data.
- Type enforcement (e.g., integers/floats for vitals and labs).

After validation, the backend sorted and added derived fields:

- eGFR (from serum creatinine, age, sex; CKD-EPI family equations).
- Body Mass Index (BMI) (kg/m²) calculated from height and weight.
- Ideal Body Weight (IBW) (Devine/Hamwi convention).

The following code shows eGFR, BMI & IBW functions code in python.

```
1.      # eGFR function to calculate the eGFR from Sr. Cr.
2.      # @param scr: Serum creatinine value (mg/dl)
3.      # @param age: The patient age
4.      # @param male: sex of the patient, True if male, False if female
5.      # This function is valid for cases older than 18 which granted in all cases.
6.      def eGFR( scr: float, age: int | float, male: bool ) -> float:
7.          if male: a: float = 0.9
8.          else: a: float = 0.7
9.          if male and scr <= 0.9: b: float = -0.302
```

```
10.     elif not male and scr <= 0.7: b: float = -0.241
11.     else: b: float = -1.2
12.     if male: c: float = 1.0
13.     else: c: float = 1.012
14.     return 142 * pow(scr/a,b) * pow(0.9938,age) * c
15.     # Ideal Body Weight Function
16.     def IBW( male: bool, height: float ) -> float:
17.         if male: base: float = 50.5
18.         else: base: float = 45.5
19.         ht: float = height * 39.37 - 60 # convey from meters to inch and delete 60 inches then add 2.3 kgs to the base for each inch
20.         return round( base + ht * 2.3, 2 )
21.     # Body Mass Index
22.     # Return tuple of two items, first item BMI and the second category [float,str]
23.     def BMI( height: float, weight: float ) -> tuple[float,str]:
24.         bmi: float = weight / pow(height,2)
25.         cat_bmi: str = ""
26.         if bmi < 16.0: cat_bmi = 'Severe Thinness'
27.         elif bmi < 17.0 : cat_bmi = 'Moderate Thinness'
28.         elif bmi < 19.0 : cat_bmi = 'Mild Thinness'
29.         elif bmi < 25.0 : cat_bmi = 'Normal'
30.         elif bmi < 30.0 : cat_bmi = 'Overweight'
31.         elif bmi < 35.0 : cat_bmi = 'Obese Class I'
32.         elif bmi < 40.0 : cat_bmi = 'Obese Class II'
33.         else: cat_bmi = 'Obese Class III'
34.         Return (bmi,cat_bmi)
```

Primary store: Stored records to a Pandas DataFrame and Pickle (.pkl) format store, representing the final hardcopy dataset. Primary retention of records: each row maintained the individual record ID so traceability back to the original hardcopy (without storing the personal IDs) could be achieved.

Inpatients - Effectiveness Model

All needed libs are loaded earlier in the used notebook.

```
1. X_v = g1_df_encode_v2.drop( columns=['mpop'] )
2. Y_v = g1_df_encode_v2[ 'mpop' ]
3. X_v_train, X_v_test, Y_v_train, Y_v_test = train_test_split(X_v, Y_v, test_size=0.25, random_state=33)
4. model_v = LogisticRegression(max_iter=1000)
5. model_v.fit(X_v_train, Y_v_train)
6. Y_v_pred = model_v.predict(X_v_test)
7. print(classification_report(Y_v_test, Y_v_pred))
```

Comments:

- This code fits a **logistic** model for a **binary** endpoint after splitting the data for training and testing.
- The MPOP column was recoded to be 1 if it got any effectiveness error or 0 in none, as an attempt to generalize the MPOP effectiveness group.

```
1. X = g1_df_encode.drop( columns=['mpop_ii_a','mpop_ii_b','mpop_ii_c_a'] )
2. Y = g1_df_encode[ ['mpop_ii_a','mpop_ii_b','mpop_ii_c_a'] ]
3. X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.25, random_state=33)
4. model = MultiOutputClassifier(LogisticRegression(max_iter=1000))
5. model.fit(X_train, Y_train)
6. Y_pred = model.predict(X_test)
7. print(classification_report(Y_test, Y_pred))
```

Comment: This part did not generalize the mpop codes and took each one separately.

Inpatients – Safety Model

All needed libs are loaded earlier in the used notebook.

```
1. for code in ['mpop_iii_a','mpop_iii_d_a','mpop_iii_e_b']:  
2.     print(f"Results for {code}:")  
3.     X_g1_final_df.drop(columns=['mpop_v_e_a','mpop_v_e_b','mpop_iii_a','mpop_iii_b','mpop_iii_d_a','mpop_iii_e_b'])  
4.     Y = g1_final_df[code]  
5.     X_train, X_test, Y_train, Y_test = train_test_split( X, Y, test_size=0.2, random_state=33)  
6.     pipeline = Pipeline([('scale', StandardScaler()),  
7.         ('select', SelectKBest(score_func=mutual_info_classif, k=20)),  
8.         ('clf', LogisticRegression(solver='lbfgs', penalty='l2', max_iter=1000, class_weight='balanced')) ])  
9.     pipeline.fit(X_train, Y_train)  
10.    Y_pred = pipeline.predict(X_test)  
11.    print("Confusion Matrix:\n", confusion_matrix(Y_test, Y_pred))  
12.    print("\nClassification Report:\n", classification_report(Y_test, Y_pred,digits=2))  
13.    print("="*70)
```

Comments:

- **Lines 1–2:** Loops over three target columns ('mpop_iii_a', 'mpop_iii_d_a', 'mpop_iii_e_b') and prints a header indicating which target's results are being produced.
- **Lines 3–4:** Constructs the feature matrix X by dropping specified label columns from g1_final_df; sets the current target Y to the loop-selected column.
- **Lines 5–7:** Splits X and Y into training and test sets with an 80/20 split and a fixed random seed (random_state=33) for reproducibility.
- **Lines 9–13:** Builds a scikit-learn Pipeline: StandardScaler() standardizes each feature using the training data, SelectKBest(score_func=mutual_info_classif, k=20) selects the 20 features with highest mutual information with the target, LogisticRegression(solver='lbfgs', penalty='l2', max_iter=1000, class_weight='balanced') fits a logistic classifier with L2 regularization; class_weight='balanced' weights classes inversely to their frequency.
- **Line 14:** Fits the entire pipeline on X_train, Y_train; scaling and feature selection are learned on the training data and applied within the pipeline.
- **Line 16:** Generates test-set predictions Y_pred by applying the trained pipeline to X_test.
- **Lines 18–19:** Prints the confusion matrix comparing Y_test and Y_pred, then prints a classification report (precision, recall, F1, support) with numeric formatting set to two decimal places.

```

1. for target_code in ['mpop_iii_a','mpop_iii_d_a','mpop_iii_e_b']:
2.     print(f"Working on {target_code}:")
3.     X = g1_final_df.drop(columns=[
4.         'mpop_v_e_a','mpop_v_e_b','mpop_iii_a','mpop_iii_b','mpop_iii_d_a','mpop_iii_e_b'])
5.     y = g1_final_df[target_code].astype(int)
6.     minority = y.value_counts().min()
7.     if minority < 2:
8.         raise ValueError("Each class needs ≥2 samples for stratified CV.")
9.     n_splits = max(2, min(5, minority))
10.    cv = RepeatedStratifiedKFold(n_splits=n_splits, n_repeats=n_5, random_state=33)
11.    k = min(20, X.shape[1])
12.    pipe = Pipeline([
13.        ('impute', SimpleImputer(strategy='median')),
14.        ('scale', StandardScaler()),
15.        ('select', SelectKBest(score_func=mutual_info_classif, k=k)),
16.        ('clf', LogisticRegression(
17.            solver='lbfgs', penalty='l2', max_iter=2000, class_weight='balanced', random_state=rng)) ])
18.    n = len(y)
19.    oof_prob_sum = np.zeros(n, dtype=float)
20.    oof_count = np.zeros(n, dtype=int)
21.    for tr, te in cv.split(X, y):
22.        pipe.fit(X.iloc[tr], y.iloc[tr])
23.        p = pipe.predict_proba(X.iloc[te])[0, 1]
24.        oof_prob_sum[te] += p
25.        oof_count[te] += 1
26.    if (oof_count == 0).any():
27.        missing = np.where(oof_count == 0)[0]
28.        raise RuntimeError(f"Some indices never evaluated OOF: {missing[:10]} ...")

```

```

29. oof_prob = oof_prob_sum / oof_count

30. oof_pred = (oof_prob >= 0.5).astype(int)

31. y_true = y.values

32. print("Confusion Matrix (OOF, averaged repeats):\n", confusion_matrix(y_true, oof_pred))

33. print("\nClassification Report (OOF, averaged repeats):\n",

34. classification_report(y_true, oof_pred, digits=5))

35. print(f"Balanced Accuracy (OOF): {balanced_accuracy_score(y_true, oof_pred):.4f}")

36. if len(np.unique(y_true)) == 2:

37. print(f"ROC AUC (OOF): {roc_auc_score(y_true, oof_prob):.4f}")

38. print(f"Average Precision (PR AUC, OOF): {average_precision_score(y_true, oof_prob):.4f}")

39. print("="*72)

```

Comments:

- **Lines 1–2:** Iterates over three binary targets ('mpop_iii_a', 'mpop_iii_d_a', 'mpop_iii_e_b') and prints a header indicating which target is being processed.
- **Lines 3–6:** Builds feature matrix X by dropping several target/label columns from g1_final_df, and sets the current label y to the selected target_code cast to integers.
- **Lines 8–10:** Computes the minority class sample size (minority = y.value_counts().min()) and raises an error if any class has fewer than 2 samples, which is required for stratified cross-validation to form at least two folds.
- **Line 12:** Determines the number of CV folds n_splits as the minority count capped at 5 and with a minimum of 2 (max(2, min(5, minority))).
- **Line 13:** Creates a RepeatedStratifiedKFold object with n_splits folds and n_5 repeats (external variable), preserving class proportions in each split; uses random_state=33 for reproducibility.
- **Line 14:** Sets k to the smaller of 20 or the number of available features, to control how many features will be selected.
- **Lines 15–21:** Defines a scikit-learn Pipeline with four sequential steps:
SimpleImputer(strategy='median') fills missing values;
StandardScaler() standardizes features;
SelectKBest(mutual_info_classif, k=k) keeps the top-k features by mutual information with y;
LogisticRegression(..., class_weight='balanced') trains a classifier with L2 penalty and LBFGS optimizer, using class weighting to compensate for class imbalance; max_iter=2000 and random_state=rng are set.
- **Lines 23–25:** Initializes arrays to accumulate out-of-fold predictions: oof_prob_sum (float sums of predicted probabilities) and oof_count (how many times each index has been predicted).
- **Lines 27–31:** For each stratified train/test split produced by cv.split(X, y), fits the Pipeline on the training subset and obtains class-1 probabilities p for the test subset; adds these probabilities to oof_prob_sum[te] and increments oof_count[te].

- **Lines 32–34:** After all repeats/folds, checks that every observation received at least one out-of-fold prediction; if any index has zero predictions, raises an error listing the first few missing indices.
- **Line 36:** Computes the averaged out-of-fold probability for each observation as `oof_prob = oof_prob_sum / oof_count`.
- **Line 37:** Converts probabilities to hard class labels with a 0.5 threshold: `oof_pred` is 1 where `oof_prob ≥ 0.5`, otherwise 0.
- **Lines 39–43:** Extracts ground-truth labels (`y_true = y.values`) and prints evaluation summaries based on the out-of-fold predictions: confusion matrix, a detailed classification report (precision/recall/F1 per class), and balanced accuracy.
- **Lines 46–48:** If the target is binary (`len(np.unique(y_true)) == 2`), prints threshold-free metrics computed from the out-of-fold probabilities: ROC AUC and Average Precision (PR AUC).