Predicting Patient Length of Stay (LOS) for Efficient Hospital Management

Your Name

2025

Abstract

A statistically rigorous framework to forecast hospital Length of Stay (LOS) at admission; includes GLM (Poisson/NB/Zero-Inflated), tree-ensemble ML (RF/GBM/XGBoost), calibration, explainability (SHAP), and deployment scaffolding.

1 Problem Formulation

Length of Stay (LOS) is the number of calendar days between admission and discharge. Statistically, LOS is a non-negative, discrete random variable $Y \in \{0, 1, 2, \dots\}$. Empirically, LOS exhibits right skewness and overdispersion ($\mathrm{Var}(Y) > \mathbb{E}[Y]$). Ignoring these properties leads to inefficient estimates and biased predictions, which cascade into staffing, bed turnover, and supply chain misallocation.

2 Objectives

2.1 Statistical Characterization of LOS Predictors

Modeling the count nature and dispersion. We start with Poisson regression, assuming equidispersion, and extend to Negative Binomial (NB) when overdispersion is present; Zero-Inflated models handle excess short/zero stays. LASSO/Elastic Net shrink high-dimensional effects; SHAP provides model-agnostic interpretability.

Formulas

- Poisson GLM:

$$Y_i \sim \mathrm{Poisson}(\mu_i), \qquad \log(\mu_i) = \beta_0 + \sum_{j=1}^{p} \beta_j X_{ij}, \quad \mathrm{Var}(Y_i|X_i) = \mu_i.$$

- Negative Binomial:

$$Y_i \sim \mathrm{NB}(\mu_i, \alpha), \qquad \mathrm{Var}(Y_i|X_i) = \mu_i + \alpha\mu_i^2,\ \alpha > 0.$$

- Zero-Inflated Poisson (ZIP) mixture:

$$\Pr(Y_i = 0) = \pi + (1-\pi)e^{-\mu_i}, \quad \Pr(Y_i = y) = (1-\pi)\frac{e^{-\mu_i}\mu_i^y}{y!},\ y > 0.$$

- LASSO (illustrative squared-loss form):

$$\hat{\beta}^{\mathrm{LASSO}} = \arg\min_{\beta} \left\{ \frac{1}{2n} \sum_{i=1}^{n} (y_i - X_i\beta)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

- SHAP additive decomposition:

$$\hat{y}_i = \phi_0 + \sum_{j=1}^{p} \phi_{ij}.$$

2.2 Development and Validation of Predictive ML Models

Capturing nonlinearities and interactions. We train tree-based ensembles (RF, GBM, XGBoost) and validate with k-fold Grouped CV by patientid to prevent leakage. Evaluate with RMSE/MAE/R^2; optionally discretize LOS and assess AUROC/F1. Include calibration diagnostics.

Formulas

- Random Forest:

$$\hat{f}_{\mathrm{RF}}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x).$$

- Gradient Boosting (additive model):

$$\hat{f}_M(x) = \sum_{m=1}^{M} \gamma_m h_m(x).$$

- XGBoost (stage-wise objective at round t):

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l\Big(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\Big) + \Omega(f_t).$$

- Metrics:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}, \quad \mathrm{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|, \quad R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}.$$

2.3 Derivation of Operational Insights and Policy Implications

From prediction to action. We compute Average Marginal Effects (AMEs), Partial Dependence, and SHAP interaction effects to quantify operational levers (e.g., staff, bed capacity). Scenario analyses simulate expected LOS changes under counterfactual resource allocations. These are associational unless paired with causal designs.

Formulas

- Average Marginal Effect for feature j:

$$\mathrm{AME}_j = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial \mathbb{E}[Y_i | X_i]}{\partial X_{ij}}.$$

2.4 Deployment Readiness and Integration

Operationalizing predictions. Best models are serialized (joblib/pickle) and exposed via a predict(X_json) API with explainability (global & local SHAP). We attach prediction intervals via quantile-aware methods (e.g., quantile forests or conformal prediction).

Formula

- Predictive interval coverage:

$$\Pr\big(L(x^*) \le Y(x^*) \le U(x^*)\big) \approx 1 - \alpha.$$

3 Repository Guide

```
.
  data/                      # (optional) raw/processed data placeholders
  notebooks/
    01. Statistical characterization of LOS predictors.ipynb
    02. Development and validation of predictive machine learning models.ipynb
    03. Derivation of operational insights and policy implications.ipynb
    04. Deployment readiness and integration.ipynb
  src/
    features.py              # encoding, leakage guards, transformers
    train.py                 # training loop + GroupKFold by patientid
    evaluate.py              # metrics, calibration, fairness slices
    infer.py                 # predict(X_json), SHAP summaries
  requirements.txt
  README.md
```

3.1 Quickstart

```
# 1) Create & activate environment
python -m venv .venv
source .venv/bin/activate  # Windows: .venv\Scripts\activate

# 2) Install deps
pip install -r requirements.txt

# 3) Run training (example)
python src/train.py --data data/hospital_admissions.csv --target "Stay (in
    days)" --group patientid

# 4) Evaluate & produce reports
python src/evaluate.py --data data/hospital_admissions.csv --artifacts
    artifacts/

# 5) Inference example
python src/infer.py --json_payload examples/admission_request.json
```

Minimum requirements.txt (suggested)

```
pandas
numpy
scikit-learn
xgboost
lightgbm
catboost
matplotlib
shap
scipy
joblib
```

4 Dataset Snapshot

- Rows: 500,000; Columns: 15; No missing values detected.
- Target: Stay (in days); Range: 3–51; mean $\approx$ 12.38; median = 9.

- Key predictors: severity, admission type, department, age band, insurance, staffing, ward, doctor.

Tip: Use GroupKFold by patientid to avoid information leakage across multiple admissions for the same patient.

5 Governance & Monitoring

- Bias/fairness: Report subgroup MAE/RMSE across sex, age, insurance ($|\Delta\text{MAE}| \leq 10\%$ recommended).
- Calibration: Reliability curves; periodic recalibration under casemix shifts.
- Model card: Data provenance, validation, limitations, and retraining policy.