Health Data Science (Python) Final Project Report

ICU In-hospital Mortality Prediction Using First-24-Hour Features

Jintao Li

January 18, 2026

Abstract

This project builds a reproducible baseline pipeline for in-hospital mortality prediction using structured ICU features from the first 24 hours after admission. Using the course-provided dataset `icu_first24hours.csv`, we define `HOSPITAL_EXPIRE_FLAG` as the binary target (0: survived, 1: died). The analysis follows a standard health data science workflow: data loading and quality inspection, missingness analysis, lightweight preprocessing, exploratory visualization, baseline model construction, and model evaluation. After preprocessing, the feature matrix has shape $X = (13258, 250)$ and the positive class rate is 5.9%, indicating severe class imbalance. We train two baseline models (balanced Logistic Regression and balanced Random Forest) and evaluate them using Accuracy, ROC-AUC, and PR-AUC (Average Precision), accompanied by ROC and Precision–Recall curves. Results show that balanced Logistic Regression achieves ACC=0.862, ROC-AUC=0.820, PR-AUC=0.248 and identifies a portion of positive cases under the default threshold, while balanced Random Forest achieves higher ROC-AUC/PR-AUC but tends to predict nearly all samples as negative at threshold 0.5, illustrating the "inflated accuracy" phenomenon under imbalance. The report emphasizes reproducibility and highlights limitations and future work directions (threshold optimization, calibration, and feature engineering) for clinical decision support.

Contents

1 Introduction 4

2 Dataset Description 5

2.1 Data Source 5

2.2 Task Definition 5

2.3 Target Distribution and Class Imbalance 5

3 Data Quality and Preprocessing 6

3.1 Missingness Overview 6

3.2 Preprocessing Strategy 6

3.3 Rationale 6

4 Exploratory Data Analysis 7

4.1 Overview 7

4.2 Group-wise Boxplot Comparison 7

4.3 Notes on Interpretation 7

5 Modeling Methodology 8

5.1 Baseline Models 8

5.2 Imbalance Handling 8

5.3 Why Not More Models? 8

6 Evaluation Metrics 9

6.1 Metrics Under Class Imbalance 9

6.2 Confusion Matrix 9

7 Results 10

7.1 Overall Performance 10

7.2 Confusion Matrices (Key Observations) 10

7.3 ROC and Precision–Recall Curves 11

8 Result Analysis 12

8.1 Why Accuracy Can Be Misleading 12

8.2 ROC-AUC vs PR-AUC 12
8.3 Model Selection Considerations 12

9 Limitations and Future Work 13

9.1 Limitations 13
9.2 Future Work 13

10 Conclusion 14

A Reproducibility Notes 15

A.1 Project Files 15
A.2 How to Run 15
A.3 Key Output Used in This Report 15

1 Introduction

Predicting patient outcomes in Intensive Care Units (ICUs) is a clinically important task for resource allocation, early intervention, and risk communication. With increasing digitization in healthcare, large-scale structured data derived from clinical monitoring and laboratory measurements provide an opportunity to build data-driven risk models.

However, ICU prediction problems often come with practical challenges: (1) missing values caused by incomplete testing or documentation, (2) heterogeneous feature distributions across patients, and (3) severe class imbalance because mortality is relatively rare compared to survival. These issues can cause misleading performance estimates if evaluation relies only on aggregate metrics such as accuracy.

This project aims to deliver a complete and reproducible baseline workflow for ICU mortality prediction using first-24-hour structured features. The focus is not on maximizing model performance but on producing a clear, course-aligned pipeline and reporting interpretable results. The work is organized into dataset description, preprocessing and exploratory analysis, baseline modeling, evaluation, and a discussion of limitations and improvement directions.

Contributions. The main contributions are:

- A reproducible data workflow for loading, inspecting, and preprocessing ICU structured data;
- A set of essential exploratory visualizations (target distribution, missingness overview, group-wise boxplots);
- Baseline model training using two standard algorithms under imbalance-aware settings;
- Evaluation using ROC/PR metrics and a discussion of why accuracy can be misleading under class imbalance.

2 Dataset Description

2.1 Data Source

The dataset `icu_first24hours.csv` is provided by the course materials. It contains structured clinical features computed from the first 24 hours after ICU admission. The dataset includes a binary target variable `HOSPITAL_EXPIRE_FLAG` indicating in-hospital mortality.

2.2 Task Definition

We formulate a binary classification task:

- Input: first-24-hour structured features for an ICU admission;
- Output: predicted probability of in-hospital death;
- Target: `HOSPITAL_EXPIRE_FLAG` (1: death, 0: survival).

After preprocessing (removing missing labels and filtering high-missingness features), the dataset statistics are:

- Feature matrix shape: $X = (13258, 250)$
- Positive class rate: 0.059 (5.9%)

2.3 Target Distribution and Class Imbalance

The target distribution is highly imbalanced (Fig. 1). In such settings, a naïve classifier that predicts all samples as negative can achieve high accuracy, but has no clinical usefulness. Therefore, we report ROC-AUC and PR-AUC in addition to accuracy, and include PR curves to reflect minority-class performance.

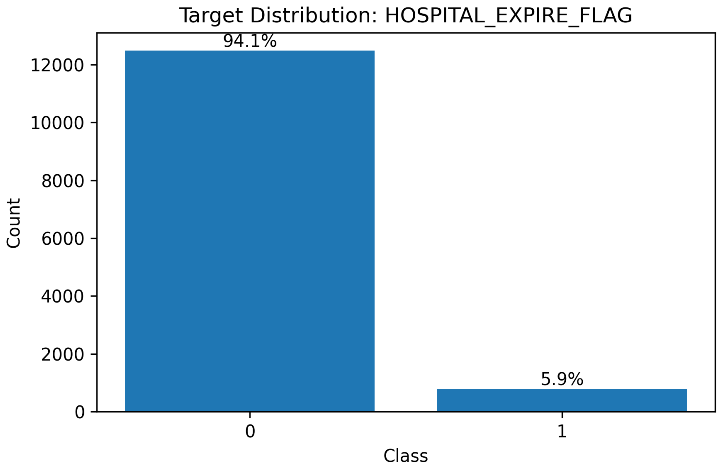

Figure 1: Distribution of `HOSPITAL_EXPIRE_FLAG`. The positive class rate is only 5.9%, indicating severe class imbalance.

3 Data Quality and Preprocessing

3.1 Missingness Overview

Missing values are common in clinical datasets. To provide a quick inspection, we compute the missing ratio per feature and visualize the top-10 features with the highest missingness (Fig. 2). This helps identify which variables may be unreliable and guides preprocessing decisions.

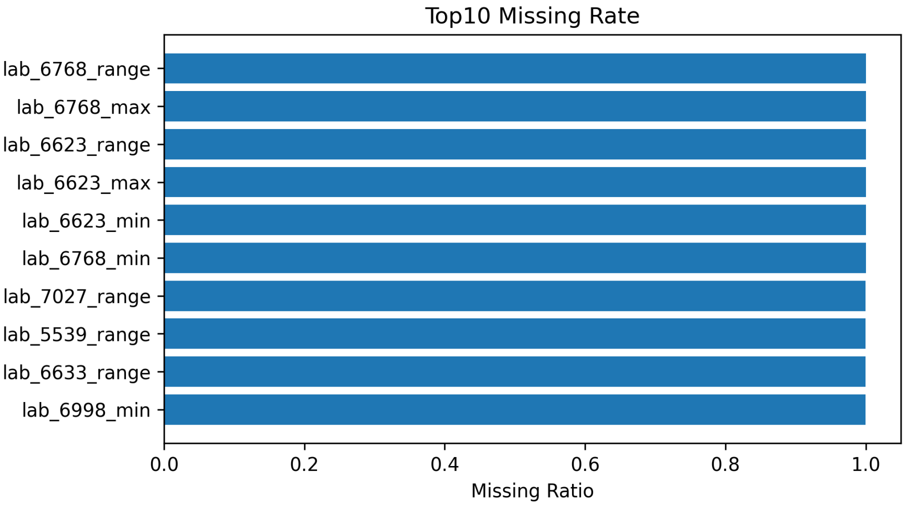

Figure 2: Top-10 features with the highest missing value ratios.

3.2 Preprocessing Strategy

Given time constraints and the baseline nature of this project, we apply lightweight preprocessing designed to be robust and reproducible:

- **Label cleaning**: remove rows with missing `HOSPITAL_EXPIRE_FLAG`.
- **Feature filtering**: remove features whose missing ratio exceeds a threshold (implemented in the pipeline).
- **Imputation**: median imputation for numerical variables (robust to outliers).
- **Train/test split**: stratified split (70/30) to preserve class ratio.

3.3 Rationale

Median imputation is a common baseline approach in medical data preprocessing because it is stable under skewed distributions and outliers. Stratified splitting avoids accidental shifts in class ratio between train and test sets, which could otherwise distort evaluation.

4 Exploratory Data Analysis

4.1 Overview

Exploratory data analysis (EDA) is used to understand feature distributions and potential differences between outcome groups. Due to the limited time budget, EDA is focused on a small set of essential visualizations:

- Target distribution (Fig. 1);
- Missingness overview (Fig. 2);
- Group-wise distribution comparisons via boxplots for selected laboratory features (Fig. 3).

4.2 Group-wise Boxplot Comparison

Fig. 3 shows boxplots of selected laboratory features split by outcome. While this analysis is not a statistical test, it provides an intuitive view of distribution shifts and potential predictive signals. In clinical datasets, even small distribution shifts can be meaningful when aggregated across many features.

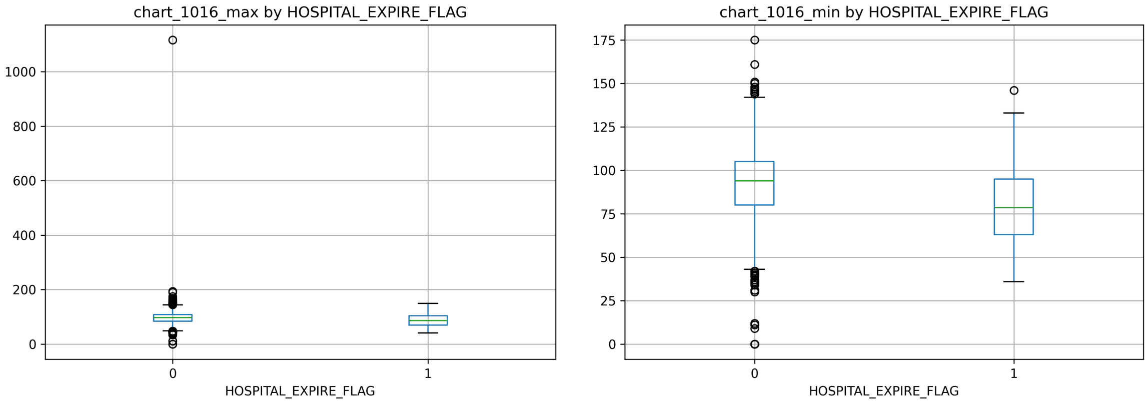

Figure 3: Boxplots of selected laboratory features by outcome: maximum values (left) and minimum values (right).

4.3 Notes on Interpretation

Boxplots should be interpreted cautiously:

- Differences may reflect confounding factors (e.g., severity of illness, measurement frequency).
- Missingness and imputation can affect apparent distributions.
- Statistical testing is not performed here to keep the pipeline minimal and reproducible.

5 Modeling Methodology

5.1 Baseline Models

To build baselines and facilitate comparison, we train two models:

- **Balanced Logistic Regression (LogReg_balanced)**: an interpretable linear baseline commonly used in clinical prediction.
- **Balanced Random Forest (RF_balanced)**: a nonlinear ensemble baseline capturing feature interactions.

5.2 Imbalance Handling

Class imbalance is addressed using imbalance-aware settings:

- Logistic Regression uses `class_weight=balanced`;
- Random Forest uses balanced sampling / class-weighted training.

These settings shift decision boundaries toward the minority class, though threshold selection remains important.

5.3 Why Not More Models?

The objective is to demonstrate a complete and reproducible workflow with limited time. Adding more models (e.g., SVM, gradient boosting) would increase tuning and reporting complexity. The chosen pair (LogReg + RF) is sufficient to illustrate the common trade-off between interpretability and nonlinear modeling.

6 Evaluation Metrics

6.1 Metrics Under Class Imbalance

We evaluate models using:

- **Accuracy (ACC)**: overall correctness, but can be misleading under imbalance.
- **ROC-AUC**: threshold-independent ranking quality.
- **PR-AUC (Average Precision, AP)**: more informative when positives are rare, emphasizing precision-recall trade-off.

6.2 Confusion Matrix

In addition to threshold-independent curves, we also inspect confusion matrices at the default threshold 0.5 to understand the error profile and whether positive cases are detected.

7 Results

7.1 Overall Performance

The main test-set results from the executed pipeline are summarized in Table 1.

Table 1: Test-set performance of baseline models (from pipeline output).

Model	ACC	ROC-AUC	PR-AUC (AP)
LogReg_balanced	0.862	0.820	0.248
RF_balanced	0.941	0.845	0.266

7.2 Confusion Matrices (Key Observations)

For **LogReg_balanced**, the confusion matrix at threshold 0.5 is:

$$\begin{bmatrix} 3302 & 442 \\ 105 & 129 \end{bmatrix}$$

This model detects a portion of positive cases (TP=129) under the default threshold.

For **RF_balanced**, the confusion matrix at threshold 0.5 is:

$$\begin{bmatrix} 3744 & 0 \\ 234 & 0 \end{bmatrix}$$

Although accuracy is high, the model predicts almost all samples as negative at threshold 0.5 (TP=0), illustrating the inflated accuracy phenomenon under severe class imbalance.

7.3 ROC and Precision–Recall Curves

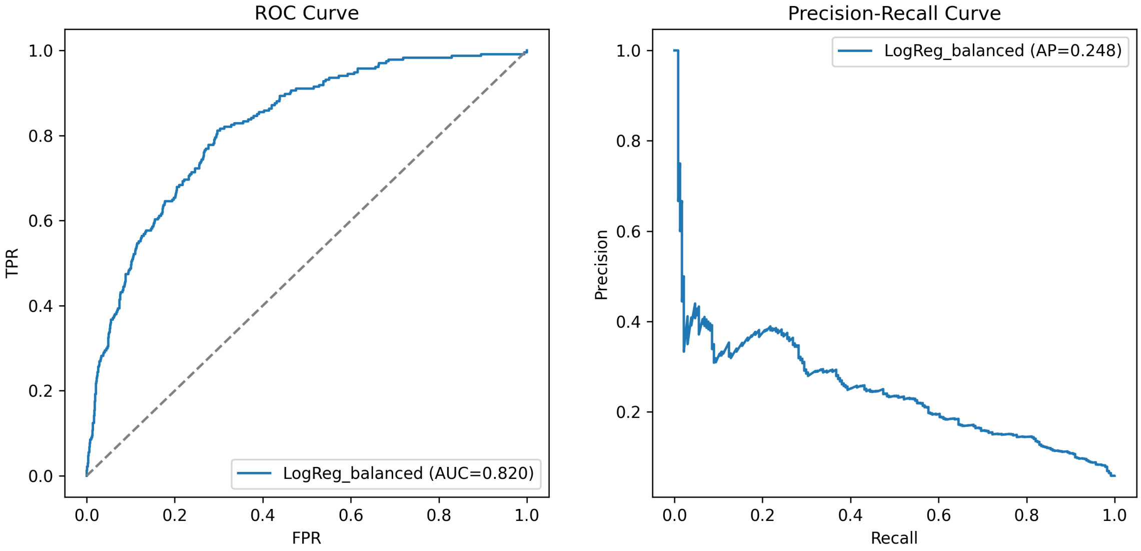

Figure 4: ROC curve (left) and PR curve (right) for LogReg_balanced.

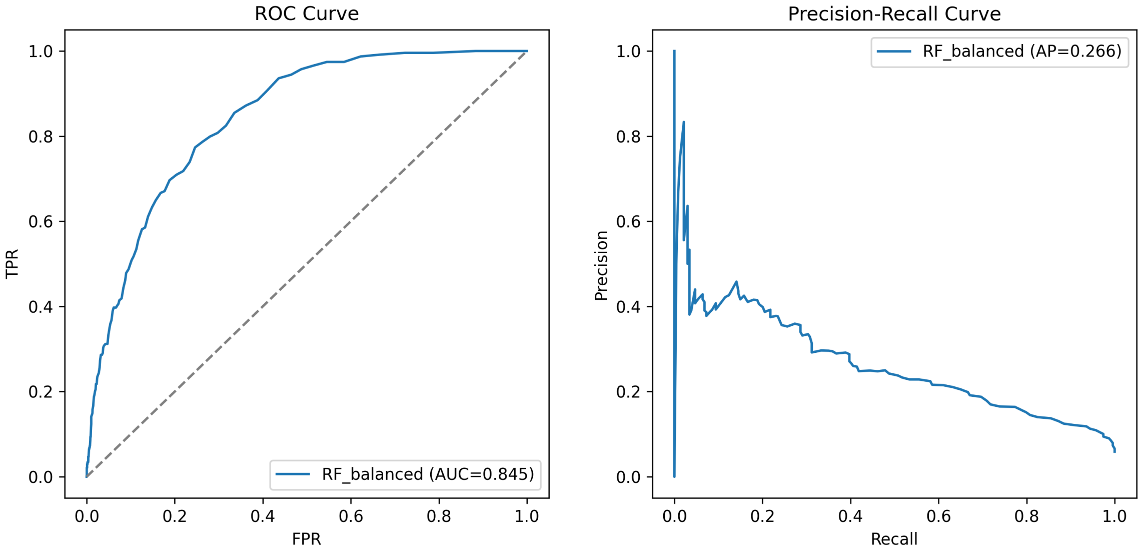

Figure 5: ROC curve (left) and PR curve (right) for RF_balanced.

8 Result Analysis

8.1 Why Accuracy Can Be Misleading

With a positive rate of only 5.9%, a classifier that predicts all samples as negative can still achieve an accuracy close to 94%. This is why accuracy should not be used as the sole metric for clinical risk prediction.

In our results, the Random Forest baseline achieved higher ACC (0.941) and slightly higher ROC-AUC/PR-AUC than Logistic Regression. However, its confusion matrix indicates no true positives at the default threshold 0.5. This suggests that, while the model may rank cases reasonably (reflected by AUC metrics), its default decision threshold is not appropriate for detecting positives in this imbalanced setting.

8.2 ROC-AUC vs PR-AUC

ROC-AUC measures ranking quality across thresholds but can look optimistic when negative examples dominate. PR-AUC focuses on the minority class and is often more informative in imbalanced tasks. In this project, PR-AUC values (0.248 and 0.266) are substantially above the random baseline (approximately 0.059), indicating meaningful predictive signal despite the difficulty of the task.

8.3 Model Selection Considerations

For clinical use, the preferred baseline is not necessarily the one with higher accuracy. A clinically useful model should detect high-risk patients (positives) with acceptable trade-offs between false positives and false negatives. Logistic Regression provides interpretability and detects positives under the default threshold. Random Forest may require threshold tuning or calibration to improve sensitivity.

9 Limitations and Future Work

9.1 Limitations

- **Threshold selection not optimized**: Confusion matrices were computed at a default threshold of 0.5. Optimizing thresholds (e.g., Youden's J, cost-sensitive thresholds) may improve positive detection.
- **No probability calibration**: Techniques such as Platt scaling or isotonic regression may improve reliability of predicted probabilities.
- **Minimal feature engineering**: Only baseline preprocessing was applied; domain-informed feature transformations may improve performance.
- **Single dataset evaluation**: No external validation was performed; generalization across settings remains unknown.

9.2 Future Work

- Incorporate advanced imbalance handling (e.g., SMOTE, focal loss-style objectives, cost-sensitive learning).
- Evaluate additional models such as gradient boosting (XGBoost/LightGBM) and compare calibration quality.
- Perform systematic hyperparameter tuning and cross-validation to assess stability.
- Add interpretability analysis (e.g., feature importance or SHAP) for clinical transparency.

10 Conclusion

This project delivered a complete and reproducible baseline pipeline for ICU in-hospital mortality prediction using first-24-hour structured features. The dataset is highly imbalanced (5.9% positives), motivating the use of PR-AUC and careful interpretation of accuracy. Balanced Logistic Regression achieved ACC=0.862, ROC-AUC=0.820, PR-AUC=0.248 and detected positive cases under the default threshold. Balanced Random Forest achieved slightly higher ROC-AUC/PR-AUC but failed to detect positives at threshold 0.5, highlighting the need for threshold tuning and calibration. Overall, the results demonstrate meaningful predictive signal and provide a practical baseline for further improvement.

A Reproducibility Notes

A.1 Project Files

A minimal project structure is as follows:

```
project_root/
  icu_first24hours.csv
  main.py
  figures/
    fig_target_bar.png
    fig_missing_top10.png
    fig_box_chart_1_016_max.png
    fig_box_chart_1_016_min.png
    fig_roc_LogReg_balanced.png
    fig_pr_LogReg_balanced.png
    fig_roc_RF_balanced.png
    fig_pr_RF_balanced.png
  outputs/
    summary.txt
    model_metrics.csv
  report.tex
```

A.2 How to Run

Run the pipeline locally (VS Code or terminal):

```
python main.py
```

The script generates figures in `figures/` and summary tables in `outputs/`.

A.3 Key Output Used in This Report

The following outputs were used directly in the report:

- $X = (13258, 250)$, positive rate = 0.059
- LogReg_balanced: ACC=0.862, ROC-AUC=0.820, PR-AUC=0.248
- RF_balanced: ACC=0.941, ROC-AUC=0.845, PR-AUC=0.266