

Detailed predictive modeling of hospital readmission risk using a class-balanced machine learning approach

G. Siva Shanmugam [†]

*Department of Computer Engineering
School of Computer Science & Engineering
RV University
Bangalore 560059
Karnataka
India*

Akarsh Kunwar ^{*}

M. Anbarasi [§]

Ojas Nahta [‡]

Vrinda Kumar [@]

Dhruv Jain [#]

*Department of Computer Engineering
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore 632014
Tamil Nadu
India*

Abstract

Forecasting 30-day hospital readmission is a major concern in healthcare because of its financial cost and the missed opportunity to act early on high-risk patients. We highlight the accuracy trap inherent in naive modeling of highly imbalanced data: a typical SMOTE-based Random Forest baseline reached a misleading 91% accuracy yet identified almost no high-risk patients (recall $\approx$ 0.0%). To address this, we built a comparative methodology that moves from cost-sensitive Random Forest classifiers to an optimized final model: XGBoost with class-balancing weights and post-hoc threshold tuning (lowered to 0.4396). The final

[†] E-mail: gsivashanmugam@rvu.edu.in

^{*} E-mail: akarsh.kunwar@gmail.com (Corresponding Author)

[§] E-mail: manbarasi@vit.ac.in

[‡] E-mail: ojasnahta2004@gmail.com

[@] E-mail: vrindakr0504@gmail.com

[#] E-mail: todhruvjain@gmail.com

model achieved a readmission recall of 69.97%, far above the baseline, offering a viable basis for clinical decision support in which identifying at-risk patients is the priority.

Subject Classification (2020): 68T05, 68T07, 68T20.

Keywords: Hospital readmission, Class imbalance, F1-score optimization, Random forest, Predictive modeling, Machine learning interpretability, Healthcare AI.

1. Introduction

Thirty-day hospital readmission after discharge is a critical and costly problem. Preventable readmissions cost the U.S. healthcare system billions of dollars every year [6], [7], and hospitals face financial penalties under programs such as the Hospital Readmissions Reduction Program (HRRP) [6]. Readmission also reflects possible shortcomings in discharge planning and follow-up, and for patients it is linked to worse health, higher morbidity and reduced quality of life [9]. The problem is especially severe for chronic conditions such as diabetes, which require continuous care outside institutions. Identifying high-risk diabetic patients before discharge allows scarce resources, such as nurse follow-up calls and care-management programs, to be targeted at those who need them most.

Machine learning (ML) [8] can uncover complex patterns in electronic health records (EHR), but a sharp statistical obstacle remains: extreme class imbalance [4]. In the UCI Diabetes 130-US Hospitals dataset [14] used here, patients readmitted within 30 days form only about 9% of the population. A model can then reach a high accuracy (here 91%) simply by always predicting the majority class ("Not Readmitted"). Such a model is statistically accurate but clinically useless, since it misses the patients it was built to find.

The key contribution of this paper is a rigorous comparative case study showing how this trap can be avoided through three modeling pipelines: (i) **a naive baseline** (Random Forest with SMOTE [11]), which attained 91% accuracy but only 3 of 1,222 true readmissions (recall $\approx 0.0\%$); (ii) **a tuned class-weighted Random Forest** [3], optimized with GridSearchCV for recall, which achieved a recall of 48.61% (594 cases); and (iii) **an optimized XGBoost model** [10] with balanced *scale_pos_weight* and threshold adjustment ($0.50 \rightarrow 0.4396$), which achieved the highest recall of 69.97% (855 cases), confirming the value of post-hoc threshold tuning over traditional balancing methods.

2. Related Work

Jencks, Williams, and Coleman [7] showed that rehospitalizations among Medicare patients are common, costly and often preventable. Kansagara et al. [6] reviewed readmission risk models and found that most perform poorly because they rely on limited administrative data and omit clinical and social factors. For diabetes, Strack et al. [5] demonstrated the importance of disease-specific clinical indicators. ML methods, together with scalable big-data infrastructures [19] and clustering techniques for uncertain data [20], have increasingly been adopted to overcome these limits; Huang et al. [8] found that ML generally outperforms logistic regression because it captures non-linear interactions in EHR data, and Shang et al. [9] applied ML classifiers to diabetic cohorts, stressing the need for careful model selection.

The central obstacle is class imbalance. He and Garcia [4] explain why standard learners are biased toward the majority class, giving high accuracy and low sensitivity. Chawla et al. [11] proposed SMOTE to synthesize minority examples, made available in practice through the imbalanced-learn toolbox [13]. For the classifier, Random Forests [3] are a robust baseline that limits overfitting and tolerates noise, and have proved effective on other imbalanced problems [21], [22]; XGBoost [10] is a scalable boosting system well suited to sparse healthcare data. Both are implemented with Scikit-learn [12]. Finally, the black-box nature of such models hinders clinical adoption, which SHAP [2] addresses by explaining individual predictions.

3. Methodology and Data Pipeline

3.1 Data Source and Preprocessing

The UCI Diabetes 130-US Hospitals dataset [14] contains 101,766 encounters with 50 clinical attributes. The binary target (30-day readmission) is strongly imbalanced, at roughly 9:1 (Figure 1). Only each patient's first encounter was retained, and encounters ending in death or hospice discharge were excluded because these patients cannot be readmitted, leaving 67,113 records. Features with severe missingness (weight, about 97%) were dropped, whereas payer code (about 40% missing) was imputed as "Unknown" (Table 1). A scikit-learn [12] *ColumnTransformer* standardized the 11 numerical variables (*StandardScaler*) and one-hot encoded the 30 categorical ones (*handle_unknown='ignore'*).

The data were split in a stratified manner into training (80%, n = 53,690) and test (20%, n = 13,423) sets, each containing 9.1% readmissions.

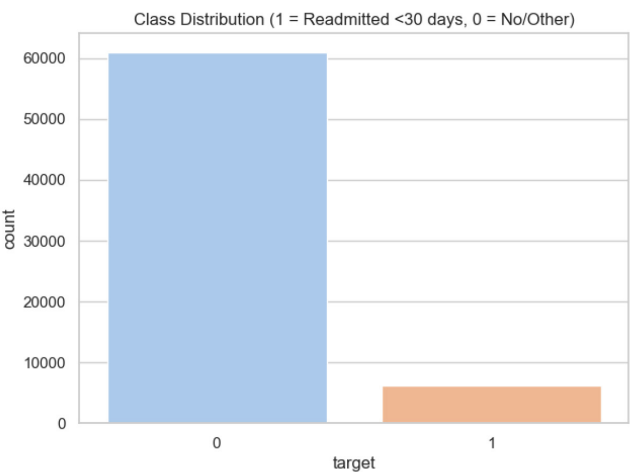


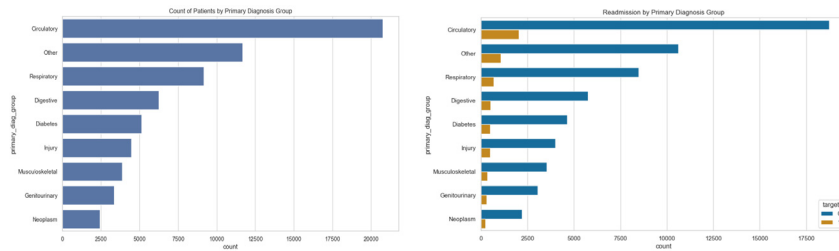
Figure 1
Severe class imbalance in the target variable

Table 1
Key data characteristics and preprocessing decisions

Features	% Missing	Handling strategy
Weight, max_glu_serum, A1Cresult, medical_specialty	> 50%	Dropped due to sparsity
Payer code	39.6%	Imputed as “Unknown”
Diagnosis codes (diag_1–3)	< 2%	Mapped to ICD-9 groups

3.2 Feature Engineering

The target was defined as readmission within 30 days (“<30” = 1; “NO” or “>30” = 0). To handle the high cardinality of *diag_1* (over 700 ICD-9 codes), a new feature, *primary_diag_group*, mapped the codes into nine clinically meaningful categories (e.g., Circulatory, Respiratory, Diabetes, Neoplasm). The distribution of patients and readmissions across these groups is shown in Figure 2.



(a) Patients by primary diagnosis group. (b) Readmissions by primary diagnosis group.

Figure 2

Distribution of patients and readmissions by primary diagnosis group.

3.3 Model Selection and Training Setup

The experiment had three phases, from a naive baseline to cost-sensitive boosting.

Baseline (naive approach): A Random Forest [3] ($n_estimators = 100$, $random_state = 42$) was combined with SMOTE [11], which creates synthetic minority examples by interpolating between a minority sample and its k nearest neighbors. An *ImbPipeline* [13] applied SMOTE only to the training folds, so that validation and test sets kept their original skewed distribution and data leakage was avoided. Despite this, the model minimized global error: since 90.9% of samples are negative, predicting “No Readmission” for everyone yields 91% accuracy. Synthetic oversampling therefore failed to push the decision boundary into the minority region of the large, sparse feature space.

Random Forest with intrinsic balancing: Instead of altering the data, we altered the cost function by using $class_weight = balanced_subsample$, which recomputes weights on each tree’s bootstrap sample and so corrects imbalance tree by tree. A 3-fold GridSearchCV with recall as the scoring metric (instead of accuracy) explored $max_depth \in \{10, 20\}$, $n_estimators \in \{100, 150\}$ and $min_samples_leaf \in \{5, 10\}$ (16 candidates, 48 fits). The best configuration was $n_estimators = 150$, $max_depth = 10$ and $min_samples_leaf = 10$ (validation recall 0.4673).

XGBoost with threshold tuning: XGBoost [10] builds trees sequentially, each correcting the residual errors of the ensemble. Imbalance was handled with $scale_pos_weight = \text{Count}(\text{Negative}) / \text{Count}(\text{Positive}) \approx 9.98$, so that a false negative is penalized about ten times more than a false positive during gradient updates. A second GridSearchCV (3 folds, 24 candidates, 72 fits) maximized F1-score over $learning_rate \in \{0.01, 0.1\}$,

$max_depth \in \{3, 5, 7\}$, $subsample \in \{0.8, 1.0\}$ and $n_estimators$. The best model used $learning_rate = 0.1$, $max_depth = 3$, $subsample = 0.8$ and $n_estimators = 200$, suggesting that the readmission signal lies in strong, high-level feature interactions rather than deep, complex patterns.

Post-hoc threshold optimization: The default threshold of 0.50 is rarely optimal for healthcare screening, where a missed readmission is far costlier than a false alarm. We therefore analyzed the precision-recall curve of the predicted probabilities and lowered the cut-off to 0.4396, the value at which recall reaches approximately 70%. This turns the model output into a “clinical knob”: administrators can trade precision (tolerating more false alarms) for a guaranteed safety net of high-risk patients.

4. Experimental Results

4.1 Quantitative Performance Comparison

All results refer to the held-out test set ($n = 13,423$; 1,222 readmissions) and are summarized in Table 2. The **naive baseline** reached a falsely high accuracy of 91% but identified only 3 of the 1,222 readmissions (recall $\approx 0.0\%$, precision 0.10), empirically confirming the accuracy trap. The **tuned Random Forest** lowered accuracy to 67% but raised recall to 48.61% (594 cases), showing that intrinsic cost-sensitive learning is superior to synthetic oversampling on this dataset. The **final XGBoost model** (threshold 0.4396) detected 855 of 1,222 readmissions (recall 69.97%) with the highest AUC-ROC (0.6385). This comes at the cost of more false positives (6,328), lowering accuracy to 50% and precision to 0.12.

Table 2
Comparison of evaluation metrics across methods (test set)

Model strategy	Acc.	Prec.	Recall	True pos.	AUC-ROC
Naive (RF + SMOTE)	0.91	0.10	0.00	3	0.6068
Class-weighted RF	0.67	0.14	0.49	594	0.6358
XGBoost (0.50)	0.65	0.14	0.54	659	0.6385
XGBoost (0.4396)	0.50	0.12	0.70	855	0.6385

Acc. = accuracy; Prec. = precision of the readmitted class; Recall = readmission recall (sensitivity).

4.2 Visual Confirmation of Model Lift

The naive model’s confusion matrix is dominated by false negatives (1,219 missed cases), making it clinically useless. At the default threshold, the tuned XGBoost model (Figure 3 (a)) finds 659 readmissions (recall 53.9%). Lowering the threshold to 0.4396 shifts the matrix toward true positives (855 detected, 367 missed) at the cost of 6,328 false positives, which is an acceptable price in a screening setting where leaving a high-risk patient unflagged is the costlier error. Figure 3 (b) shows the precision-recall trade-off behind this choice.

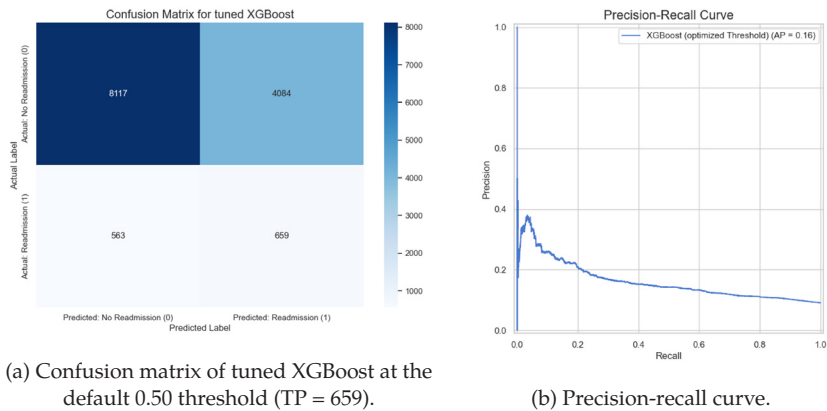


Figure 3
Confusion matrix and precision-recall trade-off of the XGBoost model

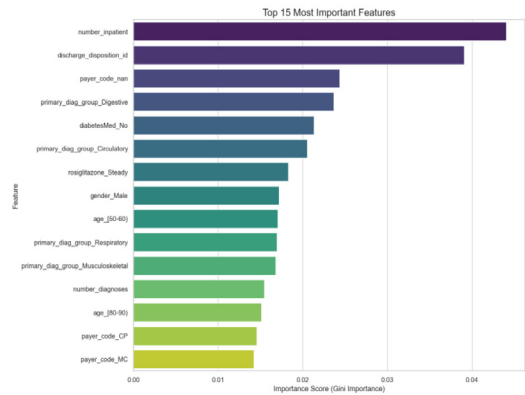


Figure 4
Top 15 features ranked by importance

4.3 Feature Importance

Figure 4 shows the 15 most important features. The strongest predictors are prior inpatient visits (*number_inpatient*) and discharge disposition, followed by payer information, primary diagnosis group (digestive, circulatory, respiratory, musculoskeletal), diabetes medication status, age bands and the number of diagnoses. These are clinically plausible markers of prior utilization, disease burden and care transitions.

5. Discussion and Clinical Implications

Conventional modeling that maximizes overall accuracy is unsuitable for this task: the SMOTE baseline exceeded 90% accuracy yet had essentially zero recall. Combining class-weighted XGBoost with F1-oriented tuning raised recall to about 54%, and post-hoc threshold adjustment lifted it to about 70%.

Errors are not equally costly in healthcare. A false negative leads to adverse outcomes, financial cost and institutional penalties, whereas a false positive triggers an inexpensive preventive step such as a nurse call or outpatient visit. Given this asymmetry, the implementation prioritizes recall, accepting lower precision so that most truly vulnerable patients receive transitional care. The model should thus be seen as an automated triage tool, not a diagnostic one, that focuses scarce resources on the most urgent cases.

6. Limitations and future work

The model relies on historical EHR data prone to coding errors and omissions, lacks social determinants of health (income, housing stability, health literacy), and achieves a modest AUC-ROC (0.64) with low precision (0.12). Future work should integrate multimodal data, including secure semantic-interoperability frameworks for healthcare IoT [15–18], apply NLP to unstructured physician notes, and add SHAP-based explanations [2]. A prospective shadow-mode trial against standard discharge procedures is needed to measure the actual reduction in readmissions and the payback period.

7. Conclusion

Readmission prediction is a highly imbalanced problem that must balance statistical and clinical needs. This study showed that naive

resampling such as SMOTE can leave the model unable to detect minority-class cases in noisy clinical data. Instead, an XGBoost framework that penalizes minority-class errors, combined with manual threshold tuning, detected about 70% of at-risk patients and gives hospital administrators a tunable, proactive intervention tool that supports predictive triage rather than reactive care. The implementation, including the preprocessing pipeline and trained models, is available in the project repository [1].

Acknowledgment

The authors thank Dr. Anbarasi M. and Dr. Siva Shanmugam G. for their guidance and feedback, the School of Computer Science and Engineering, Vellore Institute of Technology, for a supportive research environment, and the authors of the UCI Diabetes 130-US Hospitals dataset [14] for making the data publicly available.

References

- [1] A. Kunwar, "Readmission Predictor," *GitHub*. [Online]. Available: <https://github.com/akarshkunwar/ReadmissionPredictor> (2026).
- [2] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, pp. 4765–4774 (2017).
- [3] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32 (2001).
- [4] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284 (2009).
- [5] B. Strack, J. P. DeShazo, E. A. Chrischilles, and B. Leifson, "Impact of HbA1c measurement on hospital readmission rates for patients with diabetes," *BioMed Res. Int.*, vol. 2014, Art. no. 781670 (2014).
- [6] D. Kansagara, H. Englander, A. Salanitro, D. Kagen, C. Theobald, M. Freeman, and S. Kripalani, "Risk prediction models for hospital readmission: A systematic review," *JAMA*, vol. 306, no. 15, pp. 1688–1698 (2011).

- [7] S. F. Jencks, M. V. Williams, and E. A. Coleman, "Rehospitalizations among patients in the Medicare fee-for-service program," *N. Engl. J. Med.*, vol. 360, no. 14, pp. 1418–1428 (2009).
- [8] Y. Huang, A. Talwar, S. Chatterjee, and R. R. Aparasu, "Application of machine learning in predicting hospital readmissions: A scoping review of the literature," *BMC Med. Res. Methodol.*, vol. 21, Art. no. 96 (2021).
- [9] Y. Shang, K. Jiang, L. Wang, Z. Zhang, S. Zhou, Y. Liu, J. Dong, and H. Wu, "The 30-days hospital readmission risk in diabetic patients: Predictive modeling with machine learning classifiers," *BMC Med. Inform. Decis. Mak.*, vol. 21, no. Suppl. 2, Art. no. 57 (2021).
- [10] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.* pp. 785–794 (2016).
- [11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357 (2002).
- [12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830 (2011).
- [13] G. Lemaître, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning," *J. Mach. Learn. Res.*, vol. 18, no. 17, pp. 1–5 (2017).
- [14] B. Strack, J. P. DeShazo, C. Gennings, J. L. Olmo, S. Ventura, K. J. Cios, and J. N. Clore, "Impact of HbA1c measurement on hospital readmission rates: Analysis of 70,000 clinical database patient records," *BioMed Res. Int.*, vol. 2014, pp. 1–11 (2014).
- [15] S. Purushothaman, G. S. Shanmugam, and S. Nagarajan, "Achieving seamless semantic interoperability and enhancing text embedding in Healthcare IoT: A deep learning approach with survey," *SN Comput. Sci.*, vol. 5, p. 99 (2024).

- [16] P. Sony, G. Siva Shanmugam, and N. Sureshkumar, "Enhancement of semantic interoperability in healthcare systems using IFClIoT architecture," *KSII Trans. Internet Inf. Syst.*, vol. 18, no. 4, pp. 881–902 (2024).
- [17] P. Sony, G. Siva Shanmugam, and N. Sureshkumar, "A precise IFClIoT platform for the achievement of semantic interoperability in healthcare data systems," in *Proc. Int. Conf. Artif. Intell. Softw. Eng. (ICAISE)*, Kochi, India, pp. 29–34 (2023).
- [18] P. Sony, G. S. Shanmugam, and S. Nagarajan, "An intuitionistic fuzzy-based intelligent system for semantic interoperability and privacy preservation in healthcare systems," *Soft Computing*, (2023), doi: 10.1007/s00500-023-08972-6.
- [19] M. Anbarasi and M. A. Saleemdurai, "Social networks based disease analysis using Hadoop," *Int. J. Pharm. Technol.*, vol. 8, no. 2, pp. 4130–4141 (2016).
- [20] U. Tiwari, M. Anbarasi, and M. A. Saleemdurai, "A rough set based kernel possibilistic C-means algorithm for medical data," *Int. J. Pharm. Technol.*, vol. 8, no. 2, pp. 16596–16604 (2016).
- [21] S. Gupta and A. Chug, "Software maintainability prediction using an enhanced random forest algorithm," *J. Discrete Math. Sci. Cryptogr.*, vol. 23, no. 2, pp. 565–580 (2020).
- [22] A. Gupta, M. C. Lohani, and M. Manchanda, "Financial fraud detection using naive Bayes algorithm in highly imbalance data set," *J. Discrete Math. Sci. Cryptogr.*, vol. 24, no. 5, pp. 1559–1572 (2021).

Received June, 2026