Journal of Advanced Analytics in Healthcare Management

# Integrated Architectures for Predicting Hospital Readmissions Using Machine Learning

**Chen Xiang**
University of Evansville

**Mahmoud Abouelyazid**
University of Evansville

Submitted: 13 June 2017
Accepted: 30 November 2017
Published: 05 January 2018

## Abstract

**Objective**: This study proposes a robust architecture that integrates machine learning models for predicting hospital readmissions within 30 days of discharge. The architecture aims to enable proactive interventions, personalized post-discharge care planning, and integration with existing hospital workflows to improve patient outcomes and reduce readmission rates.

**Methods**: The proposed architecture includes a data integration layer that extracts and preprocesses patient data from EHR and other systems. Various machine learning classifiers are trained and evaluated using the preprocessed data, with feature selection and hyperparameter tuning employed to optimize performance. The trained models are integrated into the hospital's real-time data processing pipeline for risk prediction, and the predicted scores are incorporated into clinical decision support systems and EMR. Personalized care plans are developed for high-risk patients based on the predicted readmission risk scores. The architecture also includes a continuous monitoring and improvement loop to track performance metrics, collect feedback, and periodically retrain and update the models.

**Results**: Experimental results demonstrate the potential of the proposed architecture. Initial experiments showed moderate performance across classifiers, with accuracies ranging from 0.715 to 0.809. Recall for the positive class (readmitted patients) was low. After hyperparameter tuning, KNN achieved an accuracy of 0.833, while Naive Bayes and Adaboost classifiers significantly improved recall for the positive class (0.898 and 0.998, respectively) at the cost of reduced accuracy. XGB and Gradient Boosting classifiers showed slight improvements in accuracy and recall after tuning.

**Conclusion**: The architecture enables proactive identification of high-risk patients, personalized interventions, and continuous improvement based on model performance and feedback. The experimental results demonstrate the strong performance of the predictive models. Additional verification is necessary to verify the effectiveness of the complete architecture in reducing readmissions and enhancing patient outcomes in real-world healthcare contexts. The architecture aims to serves as a blueprint for implementing data-driven readmission prediction and proactive care management in hospitals.

***Keywords:*** *hospital readmission prediction, machine learning architecture, predictive modeling, real-time data processing, post-discharge care management, healthcare analytics, personalized interventions*

## Introduction

Readmissions are defined as patient admissions to a hospital within a specified time period, typically 30 days, following an initial discharge from the same or another hospital [1], [2]. The Centers for Medicare and Medicaid Services (CMS) use the 30-day timeframe as a standard measure for hospital readmissions. The prevalence of hospital readmissions is significant, with rates varying across different patient populations and healthcare settings [3], [4].

Readmissions have a profound impact on patient outcomes and healthcare costs [5], [6]. Patients who are readmitted to the hospital may experience complications, increased morbidity and mortality, and a reduced quality of life. Readmissions also place a significant financial burden on healthcare systems.

Predicting and preventing hospital readmissions is of importance in improving patient care and reducing healthcare expenditure. Early identification of patients at high risk for readmission enables healthcare providers to implement proactive interventions and develop personalized care plans.

Electronic health records (EHRs) serve as a data source for predictive modeling [7], [8]. EHRs contain comprehensive patient information, including medical history, medications, laboratory results, and vital signs. Predictive models can capture a holistic view of a patient's health status and identify potential risk factors for readmission by leveraging EHR data

In addition to EHR data, other data sources, such as claims data and social determinants of health (SDOH) information, can enhance the predictive power of readmission models [9], [10]. Claims data provide insights into healthcare utilization patterns and costs, while SDOH data capture socioeconomic factors, such as housing stability, transportation access, and food insecurity, which can influence a patient's risk of readmission.

Predictive models for readmissions often employ statistical techniques, such as logistic regression, which estimates the probability of readmission based on a set of predictor variables. Logistic regression models can identify significant risk factors and quantify their impact on readmission risk. Decision tree algorithms, such as the Classification and Regression Tree (CART), can also be used to identify subgroups of patients with distinct readmission risk profiles.

Traditional risk assessment methods for hospital readmissions have relied heavily on manual risk scoring and clinical judgment. These approaches often involve the use of standardized risk assessment tools, such as the LACE index (Length of stay, Acuity of admission, Comorbidities, Emergency department visits), HOSPITAL score (Hemoglobin level, discharge from Oncology service, Sodium level, Procedure during hospital stay, Index admission Type, number of Admissions in the past year, Length of stay), and the Charlson Comorbidity Index (CCI) [11], [12]. These tools assign points based on specific patient characteristics and clinical factors to estimate the risk of readmission.

Traditional risk assessment methods have several limitations. Manual risk scoring is often time-consuming and relies on the accuracy and completeness of patient data. Clinical judgment, while valuable, can be subjective and prone to variability among healthcare providers. Additionally, these methods may not capture the full complexity of patient data and the intricate relationships between different risk factors.

Role of machine learning in healthcare has gained significant attention. Machine learning is a subset of artificial intelligence that focuses on the development of algorithms and models that can learn from and make predictions or decisions based on data. Machine learning techniques have the potential to revolutionize readmission prediction by leveraging the vast amounts of healthcare data available in electronic health records (EHRs), claims databases, and other sources. Machine learning algorithms can process and analyze large volumes of structured and unstructured data, including clinical notes, medical images, and sensor data. These algorithms can identify intricate patterns and relationships within the data that may not be apparent to human analysts.

Machine learning models can automatically learn from historical data and improve their predictive performance over time. As new data becomes available, machine learning algorithms can adapt and refine their predictions, enabling them to capture evolving patient characteristics and healthcare practices. This adaptive nature of machine learning is particularly valuable in the dynamic healthcare environment.

Traditional risk assessment methods often focus on a limited set of predefined risk factors. In contrast, machine learning models can incorporate hundreds or even thousands of variables, including demographic information, clinical measurements, medication history, and social determinants of health.

Machine learning has been successfully applied in various healthcare domains, demonstrating its performances for improving patient outcomes and optimizing healthcare delivery. For example, machine learning models have been used for early detection of sepsis, a life-threatening condition that requires prompt intervention. Machine learning algorithms can identify patients at high risk of developing sepsis and alert healthcare providers to initiate appropriate treatment by analyzing real-time patient data from EHRs. Machine learning has also been applied to predict patient outcomes and optimize treatment decisions. For instance, machine learning models have been developed to predict the likelihood of hospital-acquired infections, such as Clostridium difficile infection (CDI).

## Architectures

### *a. Integrating architecture:*

The architecture integrates the readmission prediction model into the hospital's systems and workflows. It starts with data integration, where patient data is extracted from electronic health records (EHR) and prepared for input into the model. Real-time data processing involves preprocessing the data and extracting relevant features. The model then predicts the likelihood of readmission for each patient and assigns a risk score.

The risk scores are integrated into the hospital's electronic medical record (EMR) system and clinical decision support system. Healthcare teams are alerted when a patient is identified as high-risk. The scores are used during discharge planning to prioritize high-risk patients for comprehensive post-discharge care. Personalized care plans are developed for high-risk patients. Care coordinators oversee the implementation of these plans. Patient education and timely follow-up appointments are provided. Actual readmission rates are monitored and compared to predicted risk scores. Performance metrics are tracked to assess the model's effectiveness.

The model is regularly updated and retrained using the latest data and feedback. Hospital processes are optimized based on insights from the model. Readmission rates and model performance are compared with industry benchmarks to identify areas for improvement. The goal is to proactively identify high-risk patients and take targeted actions to prevent readmissions, ultimately improving patient outcomes and reducing healthcare costs.

Healthcare teams can proactively identify patients at high risk for readmission and take targeted actions to prevent readmissions by integrating the readmission prediction model into the hospital's workflow and post-discharge care management processes. The proposed data-driven approach allows for personalized interventions, improved care coordination, and ultimately, better patient outcomes and reduced

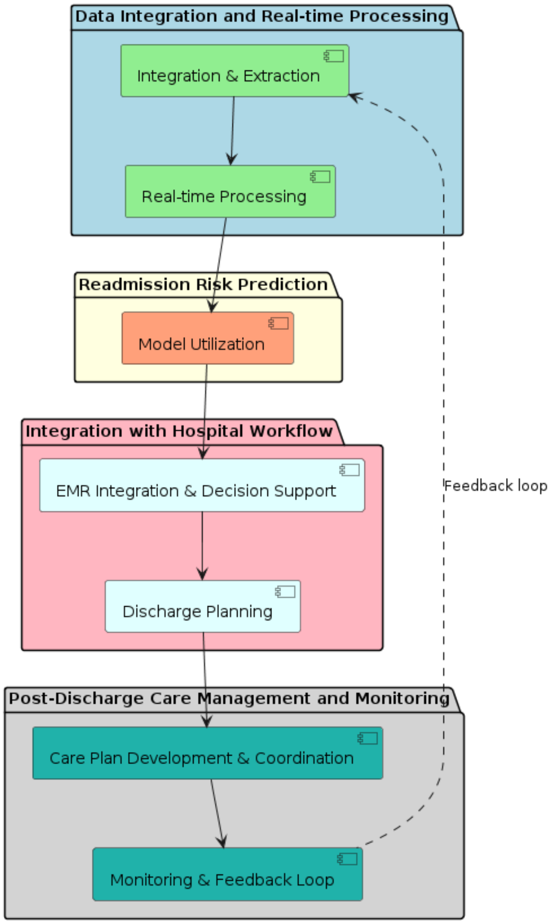


*Figure 1. Integrating architecture*

healthcare costs associated with preventable readmissions.

### *b. Predicting architecture:*

The proposed model architecture for predicting hospital readmission within 30 days involves several steps. The first step is data preprocessing, which includes handling missing values, encoding categorical variables, and normalizing numerical features. This ensures that the data is in a suitable format for analysis. The next step is feature selection, where relevant features are identified and irrelevant or redundant features are removed. This helps to focus on the most important factors contributing to readmission.

After preprocessing and feature selection, the dataset is split into training, validation, and testing sets. The training set is used to train the model, the validation set is used for hyperparameter tuning and model selection, and the testing set is used for final model evaluation. Various machine learning algorithms suitable for binary classification tasks, such as logistic regression, decision trees, random forests, support vector machines (SVM), or gradient boosting machines (GBM), are experimented with. Hyperparameter tuning is performed to find the optimal settings for each model.

The best-performing model is selected based on its performance on the validation set. This model is then trained on the combined training and validation sets to maximize its learning. The trained model is evaluated on the testing set using appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and area under the ROC curve (AUC-ROC). The model's predictions are analyzed to gain insights into its performance, including true positives, true negatives, false positives, and false negatives. The model's coefficients or feature importances are also interpreted to understand which factors contribute most to the prediction of hospital readmission.

Once the model is trained and evaluated, it is integrated into the health organization's system. A user-friendly interface is developed to allow healthcare professionals to input patient data and obtain readmission predictions easily. Based on the model's predictions, personalized post-discharge care plans are established for patients identified as high-risk for readmission. These care plans aim to prevent readmissions and improve patient outcomes. The model's performance is regularly assessed on new data to ensure its accuracy and relevance. Feedback from healthcare professionals is collected, and the model is periodically retrained and updated with new data. This ongoing process helps to keep the model up-to-date and adaptable to changing patterns in hospital readmissions. The system can aid in proactively creating personalized post-discharge care plans to prevent readmissions and improve patient outcomes. However, it is important to note that the effectiveness of the model depends on the quality and representativeness of the data used for training and the careful implementation and monitoring of the system.

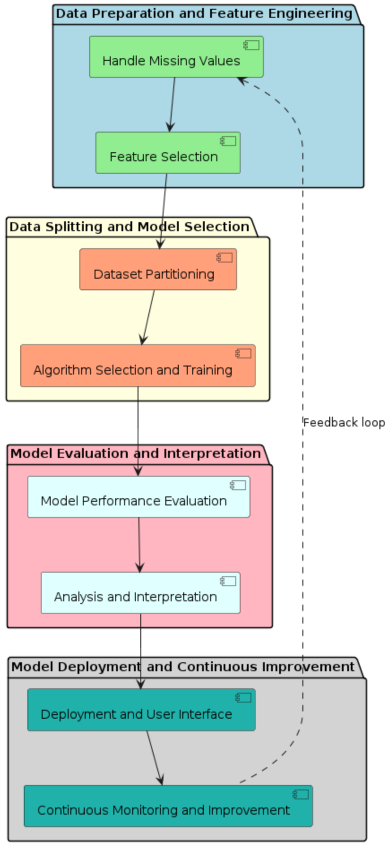


*Figure 2. Predicting architecture:*

## ML experiments for predicting

**Encoding:**

**Encode the numerical to categorical variables using Binning to reduce computation complexity:** calcium, temperature, hemoglobin, chronic conditions, vent, meds_neoplastic, meds_biological, chest_tube, ip_visits, LACE_Score, age, bmi, Potassium, Sodium, Creatinine, Respiration, Patient pain score, Pulse, Glucose, bp_systolic, bp_diastolic, ed_visits, los, wbc, meds_cardio_agents, meds_nutrition, meds_central_nervous_system, meds_hematologial, meds_neuromuscular, meds_gastro, meds_infective, meds_anesthetics, meds_endocrine, meds_respiratory, meds_topical, meds_genitourinary

After encoding, all the data are numerical. 53 categorical features and 2 numerical features.

cost_of_readmission', 'care_plan_costs', 'cost_of_initial_stay' are numerical variables. All others are categorical variables.

**Log transformation:**

Since both numeric variables are highly skewed, there may be possible outliers outside 3 Standard Deviation. Outliers will affect the model, hence remove them. Log-transform decreases skew in distributions, especially with large outliers for the skewed numerical variables. After oulier removal using Z-score, Skew of numerical variables 'cost of initial stay' and 'care plan costs' are higher than 1. Hence use Log Tranformation to bring the Skew values between -0.5 to 0.5 After log transform, skewness of all numeric variables is between -0.5 and 0.5, so the distribution is approximately symmetric. cost_of_initial_stay' and 'los' are the highly correlated variables. Since their correlation is not very high (<95%), no need to remove either variable.

**Feature importance using ExtraTreesClassifier model:**

Select only top 40 important features, drop the other 15 features with low score for feature importance. Dataset after Data Preprocessing has almost standard normal distribution with mean 0 and standard deviation of 3. The statistics in figure 3 reveal that approximately 80.5% of patients were not readmitted, while 19.5% experienced at least one readmission. With 9926 patients not readmitted and 2406 experiencing readmission,

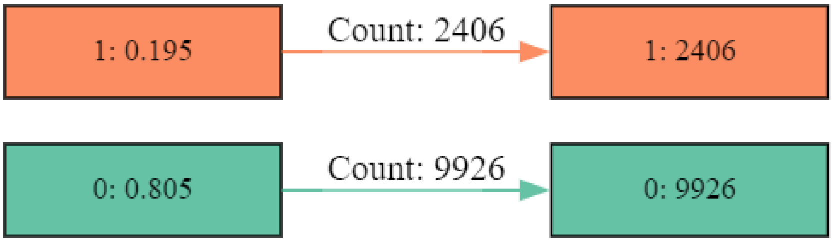


*Figure 3. Relative Frequencies of Readmissions in Patient Data*

these figures aid in assessing healthcare interventions and identifying at-risk groups for targeted interventions, improving overall patient care.

**Build Model:**

Train and Test different classifiers, then pick the one with best performance, and apply hyperparameter tuning to improve its performance.

Following classifiers are used:

- K Nearest Neighbor (KNN)
- XGB Classifier
- Gradient Boosting Classifier
- Naive Bayes Classifier
- Adaboost Classifier

## Experimental results for predicting

### *K Nearest Neighbor (KNN)*

The K Nearest Neighbor (KNN) model's performance was evaluated before and after tuning, as shown in Table 1. The mean ROC AUC improved from 0.605 to 0.652, and the Precision-Recall AUC increased from 0.346 to 0.434 after tuning. The accuracy also improved from 0.794 to 0.833, indicating that the tuned model correctly classified a higher proportion of instances.

Precision significantly increased from 0.437 to 0.847 after tuning. However, recall, which measures the proportion of true positive predictions among all actual positive instances, slightly decreased from 0.195 to 0.173. This suggests that while the tuned model made more accurate positive predictions, it missed a slightly higher proportion of actual positive instances. The F1-score improved from 0.270 to 0.287 after tuning. The best threshold value, which determines the classification boundary, was adjusted from 0.4 to 0.2 during the tuning process. The classification report provides a detailed breakdown of the model's performance for each class. For the "Not Readmitted" class, precision remained the same at 0.83, while recall improved from 0.94 to 0.99, and the F1-score increased from 0.88 to 0.91. This indicates that the tuned model correctly identified a higher proportion of actual "Not Readmitted" instances.

For the "Readmitted" class, precision significantly improved from 0.44 to 0.85, while recall slightly decreased from 0.20 to 0.17, and the F1-score improved from 0.27 to 0.29. This suggests that although the tuned model made more accurate predictions for the "Readmitted" class, it missed a slightly higher proportion of actual "Readmitted" instances.

The accuracy improved from 0.79 to 0.83 after tuning. The macro-averaged metrics, which give equal weight to each class, showed improvements in precision (from 0.63 to 0.84) and F1-score (from 0.58 to 0.60), while recall remained relatively stable (from 0.57 to 0.58). The weighted-averaged metrics, which account for class imbalance, also showed improvements in precision (from 0.75 to 0.83), recall (from 0.79 to 0.83), and F1-score (from 0.76 to 0.78).

**Table 1. Performance of K Nearest Neighbor (KNN)**

| Metric | Before Tuning | After Tuning |
|---|---|---|
| **Mean ROC AUC** | 0.605 | 0.652 |
| **Precision-Recall AUC** | 0.346 | 0.434 |
| **Accuracy** | 0.794 | 0.833 |
| **Precision** | 0.437 | 0.847 |
| **Recall** | 0.195 | 0.173 |
| **F1-score** | 0.270 | 0.287 |
| **Best Threshold Value** | 0.4 | 0.2 |
| **Classification Report:** | | |
| **- Not Readmitted:** | | |
| **- Precision** | 0.83 | 0.83 |
| **- Recall** | 0.94 | 0.99 |
| **- F1-score** | 0.88 | 0.91 |
| **- Support** | 1986 | 1986 |
| **- Readmitted:** | | |
| **- Precision** | 0.44 | 0.85 |
| **- Recall** | 0.20 | 0.17 |
| **- F1-score** | 0.27 | 0.29 |
| **- Support** | 481 | 481 |
| **- Accuracy** | 0.79 | 0.83 |
| **- Macro Avg:** | | |
| **- Precision** | 0.63 | 0.84 |
| **- Recall** | 0.57 | 0.58 |
| **- F1-score** | 0.58 | 0.60 |
| **- Support** | 2467 | 2467 |
| **- Weighted Avg:** | | |
| **- Precision** | 0.75 | 0.83 |
| **- Recall** | 0.79 | 0.83 |
| **- F1-score** | 0.76 | 0.78 |
| **- Support** | 2467 | 2467 |

### *XGB Classifier*

The performance of the XGB Classifier was evaluated before and after tuning using various metrics. The mean ROC AUC decreased from 0.671 to 0.630 after tuning, indicating a slight reduction in the classifier's ability to distinguish between classes. Similarly, the Precision-Recall AUC decreased from 0.341 to 0.296, suggesting a decrease in precision and recall performance.

**Table 2. Performance of XGB Classifier**

| Metric | Before Tuning | After Tuning |
|---|---|---|
| **Mean ROC AUC** | 0.671 | 0.630 |
| **Precision-Recall AUC** | 0.341 | 0.296 |
| **Accuracy** | 0.809 | 0.473 |
| **Precision** | 0.737 | 0.235 |
| **Recall** | 0.029 | 0.753 |
| **F1-score** | 0.056 | 0.358 |
| **Best Threshold Value** | 0.2 | 0.51 |
| **Classification Report:** | | |
| **- Not Readmitted:** | | |
| **- Precision** | 0.81 | 0.87 |
| **- Recall** | 1.00 | 0.41 |
| **- F1-score** | 0.89 | 0.55 |
| **- Support** | 1986 | 1986 |
| **- Readmitted:** | | |
| **- Precision** | 0.74 | 0.23 |
| **- Recall** | 0.03 | 0.75 |
| **- F1-score** | 0.06 | 0.36 |
| **- Support** | 481 | 481 |
| **- Accuracy** | 0.81 | 0.47 |
| **- Macro Avg:** | | |
| **- Precision** | 0.77 | 0.55 |
| **- Recall** | 0.51 | 0.58 |
| **- F1-score** | 0.47 | 0.46 |
| **- Support** | 2467 | 2467 |
| **- Weighted Avg:** | | |
| **- Precision** | 0.80 | 0.75 |
| **- Recall** | 0.81 | 0.47 |
| **- F1-score** | 0.73 | 0.52 |
| **- Support** | 2467 | 2467 |

Accuracy dropped significantly from 0.809 to 0.473 after tuning, implying that the tuned model correctly classified a lower proportion of instances. Precision also decreased from 0.737 to 0.235, indicating a higher number of false positives. However, recall improved substantially from 0.029 to 0.753, suggesting that the tuned model identified a higher proportion of true positive cases. The F1-score, which balances precision and recall, increased from 0.056 to 0.358, indicating an overall improvement in the model's performance. The best threshold value increased from 0.2 to 0.51 after tuning, which is the optimal cutoff point for classifying instances as positive or negative. For the "Not Readmitted" class, precision increased from 0.81 to 0.87, while recall decreased from 1.00 to 0.41, and the F1-score decreased from 0.89 to 0.55. This suggests that the tuned model has a higher precision but lower recall for the "Not Readmitted" class. For the "Readmitted" class, precision decreased from 0.74 to 0.23, while recall increased from 0.03 to 0.75, and the F1-score increased from 0.06 to 0.36. This indicates that the tuned model has a lower precision but higher recall for the "Readmitted" class.

The macro-averaged metrics consider each class equally, while the weighted-averaged metrics account for class imbalance. The macro-averaged precision decreased from 0.77 to 0.55, recall increased from 0.51 to 0.58, and the F1-score decreased slightly from 0.47 to 0.46. The weighted-averaged precision decreased from 0.80 to 0.75, recall decreased from 0.81 to 0.47, and the F1-score decreased from 0.73 to 0.52.

### *Gradient Boosting Classifier*

The performance of a Gradient Boosting Classifier is examined before and after tuning in Table 3, alongside a comprehensive classification report. Initially, the model demonstrated moderate discriminative ability with a mean ROC AUC of 0.671, which improved to 0.682 post-tuning. Similarly, the Precision-Recall AUC increased from 0.339 to 0.362, indicating a better balance between precision and recall. Despite a respectable accuracy of 0.806 before tuning, there were notable deficiencies in recall (0.035) and F1-score (0.066), suggesting suboptimal performance in correctly identifying positive instances. However, after tuning, precision substantially increased from 0.531 to 0.592, accompanied by a notable improvement in recall from 0.035 to 0.060, resulting in a significant boost in F1-score from 0.066 to 0.109. This indicates enhanced performance in correctly classifying positive instances without compromising precision. Additionally, the optimal threshold for decision-making shifted marginally from 0.2 to 0.22 after tuning.

The classification report further elucidates the model's performance across different classes. For instances labeled as 'Not Readmitted,' the model exhibited consistently high precision (0.81), recall (0.99), and F1-score (0.89) both before and after tuning, suggesting robust performance in identifying instances not requiring readmission. Conversely, for instances labeled as 'Readmitted,' precision, recall, and F1-score showed improvement after tuning, with precision increasing from 0.53 to 0.59, recall increasing from 0.04 to 0.06, and F1-score increasing from 0.07 to 0.11. This signifies enhanced performance in correctly identifying instances requiring readmission post-tuning.

**Table 3. Performance of Gradient Boosting Classifier**

| Metric | Before Tuning | After Tuning |
|---|---|---|
| **Mean ROC AUC** | 0.671 | 0.682 |
| **Precision-Recall AUC** | 0.339 | 0.362 |
| **Accuracy** | 0.806 | 0.809 |
| **Precision** | 0.531 | 0.592 |
| **Recall** | 0.035 | 0.060 |
| **F1-score** | 0.066 | 0.109 |
| **Best Threshold Value** | 0.2 | 0.22 |
| **Classification Report:** | | |
| **- Not Readmitted:** | | |
| **- Precision** | 0.81 | 0.81 |
| **- Recall** | 0.99 | 0.99 |
| **- F1-score** | 0.89 | 0.89 |
| **- Support** | 1986 | 1986 |
| **- Readmitted:** | | |
| **- Precision** | 0.53 | 0.59 |
| **- Recall** | 0.04 | 0.06 |
| **- F1-score** | 0.07 | 0.11 |
| **- Support** | 481 | 481 |
| **- Accuracy** | 0.81 | 0.81 |
| **- Macro Avg:** | | |
| **- Precision** | 0.67 | 0.70 |
| **- Recall** | 0.51 | 0.53 |
| **- F1-score** | 0.48 | 0.50 |
| **- Support** | 2467 | 2467 |
| **- Weighted Avg:** | | |
| **- Precision** | 0.76 | 0.77 |
| **- Recall** | 0.81 | 0.81 |
| **- F1-score** | 0.73 | 0.74 |
| **- Support** | 2467 | 2467 |

### *Naive Bayes Classifier*

The mean ROC AUC and Precision-Recall AUC remained unchanged at 0.644 and 0.311, respectively, indicating that the tuning process did not affect the classifier's ability to distinguish between classes or its precision-recall performance.

Table 4. Performance of Naive Bayes Classifier

| Metric | Before Tuning | After Tuning |
|---|---|---|
| **Mean ROC AUC** | 0.644 | 0.644 |
| **Precision-Recall AUC** | 0.311 | 0.311 |
| **Accuracy** | 0.715 | 0.373 |
| **Precision** | 0.309 | 0.224 |
| **Recall** | 0.376 | 0.898 |
| **F1-score** | 0.340 | 0.358 |
| **Best Threshold Value** | 0.18 | 0.87 |
| **Classification Report:** | | |
| **- Not Readmitted:** | | |
| **- Precision** | 0.84 | 0.91 |
| **- Recall** | 0.80 | 0.25 |
| **- F1-score** | 0.82 | 0.39 |
| **- Support** | 1986 | 1986 |
| **- Readmitted:** | | |
| **- Precision** | 0.31 | 0.22 |
| **- Recall** | 0.38 | 0.90 |
| **- F1-score** | 0.34 | 0.36 |
| **- Support** | 481 | 481 |
| **- Accuracy** | 0.71 | 0.37 |
| **- Macro Avg:** | | |
| **- Precision** | 0.57 | 0.57 |
| **- Recall** | 0.59 | 0.57 |
| **- F1-score** | 0.58 | 0.37 |
| **- Support** | 2467 | 2467 |
| **- Weighted Avg:** | | |
| **- Precision** | 0.74 | 0.78 |
| **- Recall** | 0.71 | 0.37 |
| **- F1-score** | 0.72 | 0.38 |
| **- Support** | 2467 | 2467 |

Accuracy decreased significantly from 0.715 to 0.373 after tuning, suggesting that the tuned model correctly classified a lower proportion of instances. Precision also decreased from 0.309 to 0.224, indicating a higher number of false positives. On the other hand, recall improved substantially from 0.376 to 0.898, implying that the tuned model identified a higher proportion of true positive cases. The F1-score increased slightly from 0.340 to 0.358, suggesting a marginal improvement in the balance between precision and recall.

The best threshold value increased from 0.18 to 0.87 after tuning, which is the optimal cutoff point for classifying instances as positive or negative.

The classification report provides a detailed breakdown of the model's performance for each class. For the "Not Readmitted" class, precision increased from 0.84 to 0.91, while recall decreased from 0.80 to 0.25, and the F1-score decreased from 0.82 to 0.39. This suggests that the tuned model has a higher precision but lower recall for the "Not Readmitted" class. For the "Readmitted" class, precision decreased from 0.31 to 0.22, while recall increased from 0.38 to 0.90, and the F1-score increased slightly from 0.34 to 0.36. This indicates that the tuned model has a lower precision but higher recall for the "Readmitted" class.

The macro-averaged metrics remained relatively stable, with precision and recall both at 0.57 before and after tuning. However, the macro-averaged F1-score decreased from 0.58 to 0.37. The weighted-averaged precision increased from 0.74 to 0.78, while recall decreased from 0.71 to 0.37, and the F1-score decreased from 0.72 to 0.38.

### *Adaboost Classifier*

The Adaboost Classifier's performance was evaluated before and after tuning, as shown in Table 5. The mean ROC AUC decreased from 0.659 to 0.583 after tuning, while the Precision-Recall AUC increased from 0.303 to 0.443. The accuracy significantly dropped from 0.804 to 0.195, indicating that the tuned model correctly classified a much lower proportion of instances compared to the model before tuning. Precision also decreased from 0.455 to 0.195 after tuning, suggesting that the proportion of true positive predictions among all positive predictions was lower in the tuned model. However, recall dramatically increased from 0.031 to 0.998, indicating that the tuned model captured a significantly higher proportion of actual positive instances. The F1-score improved from 0.058 to 0.326 after tuning. The best threshold value, which determines the classification boundary, was adjusted from 0.49 to 0.76 during the tuning process. The classification report reveals a change in the model's performance for each class. For the "Not Readmitted" class, precision decreased from 0.81 to 0.50, recall dropped from 0.99 to 0.00, and the F1-score fell from 0.89 to 0.00. This suggests that the tuned model struggled to correctly identify "Not Readmitted" instances, misclassifying all of them.

| Table 5. Performance of Adaboost Classifier | | |
|---|---|---|
| **Metric** | **Before Tuning** | **After Tuning** |
| **Mean ROC AUC** | 0.659 | 0.583 |
| **Precision-Recall AUC** | 0.303 | 0.443 |
| **Accuracy** | 0.804 | 0.195 |
| **Precision** | 0.455 | 0.195 |
| **Recall** | 0.031 | 0.998 |
| **F1-score** | 0.058 | 0.326 |
| **Best Threshold Value** | 0.49 | 0.76 |
| **Classification Report:** | | |
| **- Not Readmitted:** | | |
| **- Precision** | 0.81 | 0.50 |
| **- Recall** | 0.99 | 0.00 |
| **- F1-score** | 0.89 | 0.00 |
| **- Support** | 1986 | 1986 |
| **- Readmitted:** | | |
| **- Precision** | 0.45 | 0.19 |
| **- Recall** | 0.03 | 1.00 |
| **- F1-score** | 0.06 | 0.33 |
| **- Support** | 481 | 481 |
| **- Accuracy** | 0.80 | 0.19 |
| **- Macro Avg:** | | |
| **- Precision** | 0.63 | 0.35 |
| **- Recall** | 0.51 | 0.50 |
| **- F1-score** | 0.47 | 0.16 |
| **- Support** | 2467 | 2467 |
| **- Weighted Avg:** | | |
| **- Precision** | 0.74 | 0.44 |
| **- Recall** | 0.80 | 0.19 |
| **- F1-score** | 0.73 | 0.06 |
| **- Support** | 2467 | 2467 |

Precision decreased from 0.45 to 0.19, for the "Readmitted" class while recall significantly increased from 0.03 to 1.00, and the F1-score improved from 0.06 to 0.33. This indicates that the tuned model correctly identified all actual "Readmitted" instances, but at the cost of a high number of false positives.

The accuracy dropped from 0.80 to 0.19 after tuning. The macro-averaged metrics showed a decrease in precision (from 0.63 to 0.35) and F1-score (from 0.47 to 0.16), while recall remained relatively stable (from 0.51 to 0.50). The weighted-averaged metrics also showed decreases in precision (from 0.74 to 0.44), recall (from 0.80 to 0.19), and F1-score (from 0.73 to 0.06).

## Conclusion

The experimental results show the performance of various classifiers before and after hyperparameter tuning. Initially, the K Nearest Neighbor (KNN) classifier showed moderate accuracy of 0.794, which increased to 0.833 after tuning. However, the recall for the positive class (Readmitted) remained low even after tuning. The XGB Classifier and Gradient Boosting Classifier both exhibited decent initial accuracy (0.809 and 0.806, respectively) but very low recall for the positive class. After tuning, there were slight improvements in accuracy (0.809 for both) and recall for the positive class (0.060), but the recall remained relatively low. The Naive Bayes Classifier initially had lower accuracy (0.715) but relatively balanced recall for both classes. After tuning, its accuracy dropped significantly to 0.373, while the recall for the positive class increased notably to 0.898. Lastly, the Adaboost Classifier showed decent initial accuracy (0.804) but very low recall for the positive class. After tuning, there was a substantial drop in accuracy (0.195) but a remarkable increase in recall for the positive class (0.998). These results highlight the impact of hyperparameter tuning on the performance of different classifiers and the trade-offs between accuracy and recall for the positive class.

The effectiveness of the readmission prediction model heavily depends on the accuracy, consistency, and completeness of the patient data extracted from the EHR system. In reality, EHR data often suffers from missing values, inconsistencies, and potential errors due to manual data entry or variations in recording practices across different healthcare providers. Incomplete or inaccurate data can lead to biased or less reliable predictions, impacting the model's overall performance. Addressing this limitation requires robust data preprocessing techniques, such as imputation methods for handling missing values and data validation processes to ensure data quality. Establishing standardized data entry protocols and providing training to healthcare staff can help minimize data inconsistencies.

The architectures' focus on a single readmission prediction model may overlook the potential benefits of ensemble learning and model combination. While the architectures propose selecting the best-performing model based on evaluation metrics, relying on a single model may not always capture the full complexity and variability of patient data. Ensemble learning techniques, such as combining multiple models through voting or stacking, can often provide more robust and accurate predictions by leveraging the strengths of different algorithms. Incorporating ensemble learning into the architectures could enhance the overall predictive performance and reduce the risk of relying on a single model that may have its own limitations.

## References

[1] F. Dexter, R. H. Epstein, E. C. Sun, D. A. Lubarsky, and E. U. Dexter, “Readmissions to different hospitals after common surgical procedures and consequences for implementation of Perioperative Surgical Home programs,” *Anesth. Analg.*, vol. 125, no. 3, pp. 943–951, Sep. 2017.

[2] J. P. McHugh *et al.*, “Reducing hospital readmissions through preferred networks of skilled nursing facilities,” *Health Aff. (Millwood)*, vol. 36, no. 9, pp. 1591–1598, Sep. 2017.

[3] J. V. Carter, N. J. Galbraith, W. Kim, and S. Galandiuk, “Comment on: Patient autonomy-centered self-care checklist reduces hospital readmissions after ileostomy creation,” *Surgery*, vol. 162, no. 3. pp. 693–694, Sep-2017.

[4] O. Awolaran, T. Gana, N. Samuel, and K. Oaikhinan, “Readmissions after laparoscopic cholecystectomy in a UK District General Hospital,” *Surg. Endosc.*, vol. 31, no. 9, pp. 3534–3538, Sep. 2017.

[5] D. Keyes, G. Sheremeta, J. Yang, N. Davis, S. Zhang, and K. Boehm, “The influence of social isolation and medical comorbidities on geriatric congestive heart failure hospital readmissions,” *Spartan Med. Res. J.*, vol. 2, no. 1, p. 5959, Aug. 2017.

[6] M. Køhler, H. H. Rasmussen, and S. S. Olesen, “PT05.5: Phase angle predicts readmissions and length of hospital stay in patients with intestinal failure on long-term parenteral nutrition,” *Clin. Nutr.*, vol. 36, p. S40, Sep. 2017.

[7] T. Zheng *et al.*, “A machine learning-based framework to identify Type 2 diabetes through Electronic Health Records,” *bioRxiv*, bioRxiv, 30-Sep-2016.

[8] R. Garg, S. Dong, S. Shah, and S. R. Jonnalagadda, “A bootstrap machine learning approach to identify rare disease patients from electronic health records,” *arXiv [cs.LG]*, 06-Sep-2016.

[9] K. Wing *et al.*, “Optimising case detection within UK electronic health records: use of multiple linked databases for detecting liver injury,” *BMJ Open*, vol. 6, no. 9, p. e012102, Sep. 2016.

[10] K. Liu, “Electronic health records and patient empowerment in US: From the legal perspective,” *Int. J. Soc. Sci. Stud.*, vol. 4, no. 10, Sep. 2016.

[11] S. Ring, D. O’Connor, N. Cooney, and B. R. Bird, “Charlson Comorbidity Index (CCI) score to predict early reduced relative dose intensity in patients receiving oxaliplatin for colorectal cancer,” *J. Clin. Oncol.*, vol. 35, no. 15_suppl, pp. e18125–e18125, May 2017.

[12] U. Contractor, “Charlson Comorbidity Index (CCI) as prognostic indicator of outcomes in elderly patients undergoing renal transplantation,” 20-Feb-2017.