**.3Machine Learning Models for Hospital Readmission Prediction**

**3.1Introduction**

This study presents a concise evaluation of various machine learning methods for predicting hospital readmissions. We compare baseline models, ensemble methods, and neural networks—both with and without resampling techniques—to address class imbalance and enhance model robustness. Accurate prediction of whether a patient will be readmitted within 30 days, beyond 30 days, or not at all is crucial for optimizing resource allocation and improving patient care.

3.2**Data Description**

* **Source:** DataPreprocessing.csv
* **Key Features:**
  + **Demographics:** age, gender, race (influence on readmission risk).
  + **Clinical Metrics:** num\_lab\_procedures, num\_medications (indicators of treatment intensity).
  + **Visit History:** number\_outpatient, number\_emergency (reflect frequent care needs).
  + **Diagnostics:** diag\_1–diag\_3, number\_diagnoses (severity and complexity).
  + **Laboratory:** max\_glu\_serum, A1Cresult (control of chronic conditions).
  + **Medication:** change, diabetesMed (treatment adjustments).
  + **Target:** readmitted (<30, >30, NO).

**4. Feature & Parameter Rationale**

* **Feature Selection:** Chosen for documented correlation with readmission in literature (e.g., frequent ED visits signal instability).
* **Class Weighting:** class\_weight='balanced' ensures minority class (<30-day readmissions) is emphasized in loss function.
* **SMOTETomek:** Combines oversampling and cleaning to generate realistic minority samples and remove noise.
* **Hyperparameters:** Grid search ranges focused on tree depth (control overfitting) and number of estimators (trade-off between bias and variance).

**4. Data Preprocessing**

* **4.1 Split Data:**

Train/test split (80/20) for unbiased evaluation:

A computer code with text

AI-generated content may be incorrect.

**4.2 Standardization**

Features are standardized using StandardScaler to ensure consistent scales, especially for models like Logistic Regression and Neural Networks.

**5. Model Training & Evaluation**

This section details each model, its implementation, rationale, and associated visualizations (available in the ml\_data\_diab\_.ipynb notebook).

**5.1 Logistic Regression**

* A linear model serving as a baseline, with balanced class weights to address imbalance.
* **Implementation:**

A screenshot of a computer program

AI-generated content may be incorrect.

Fast and interpretable, with coefficients revealing feature impacts (e.g., how num\_lab\_procedures affects readmission probability).

**5.1.1 Performance Visualization:**

* **Confusion Matrix:** A heatmap showing the number of correct and incorrect predictions across the three classes (<30, >30, NO). The diagonal values represent correct predictions, with higher values indicating better performance. Available in the notebook under the Logistic Regression evaluation section.
* **ROC Curve:** Plots the True Positive Rate vs. False Positive Rate for each class, with the Area Under the Curve (AUC) indicating discrimination ability. A higher AUC (closer to 1) suggests better performance. Found in the notebook’s evaluation plots.
* **Precision-Recall Curve:** Focuses on the trade-off between precision and recall, particularly for the minority class (<30). Useful for imbalanced data, located in the notebook.

A screenshot of a graph

AI-generated content may be incorrect.

**5.2 Random Forest**

* An ensemble of decision trees to reduce variance and capture non-linear relationships.
* **Implementation:**

A screenshot of a computer code

AI-generated content may be incorrect.

Captures non-linear interactions (e.g., number\_emergency and diag\_1) and provides feature importance scores.

5.2.1 **Performance Visualization:**

* **Confusion Matrix:** Displays the classification results across the three classes, with a focus on improving predictions for <30. High diagonal values indicate strong performance. Available in the notebook.
* **Feature Importance Plot:** A bar chart showing the relative importance of features (e.g., number\_emergency may rank highest). Found in the notebook under Random Forest results.
* **ROC Curve:** Illustrates the model’s ability to distinguish between classes, with AUC values typically higher than Logistic Regression due to its ensemble nature. Located in the notebook.
* **Precision-Recall Curve:** Emphasizes performance on the minority class, showing improved recall compared to the baseline. Available in the notebook.

A screenshot of a graph

AI-generated content may be incorrect.

**5.3Decision Tree**

* A single tree-based model for classification.
* Implementation:



Interpretable but prone to overfitting without tuning.

* **Performance Visualization:**
  + **Confusion Matrix:** Moderate performance, with potential overfitting. Available in the notebook.
  + **Tree Visualization:** A diagram showing splits (e.g., based on num\_lab\_procedures). Found in the notebook.
  + **ROC Curve:** Lower AUC due to overfitting. Located in the notebook.
  + **Precision-Recall Curve:** Limited recall for <30. Available in the notebook.  
    A screenshot of a computer

    AI-generated content may be incorrect.

**5.4 XGBoost**

* An optimized gradient boosting algorithm for high performance.
* **Implementation:**A screenshot of a computer program

  AI-generated content may be incorrect.

Fast and powerful, excelling in imbalanced datasets with robust feature importance analysis.

* **Performance Visualization:**
  + **Confusion Matrix:** Strong performance, similar to Gradient Boosting, with good <30 detection. Available in the notebook.
  + **Feature Importance Plot:** Ranks features like number\_emergency, num\_lab\_procedures. Found in the notebook.
  + **ROC Curve:** High AUC, often outperforming Gradient Boosting. Located in the notebook.
  + **Precision-Recall Curve:** Strong recall for <30, reflecting optimization for imbalanced data. Available in the notebook.

A screenshot of a computer screen

AI-generated content may be incorrect.

**5.5 K-Nearest Neighbors (KNN)**

* Classifies based on the majority class of the nearest neighbors.
* **Implementation:**A screenshot of a computer program

  AI-generated content may be incorrect.

Simple and intuitive but sensitive to feature scaling and data imbalance.

 **Performance Visualization:**

* **Confusion Matrix:** May show lower recall for <30 unless balanced. Available in the notebook.
* **ROC Curve:** Moderate AUC, reflecting sensitivity to data distribution. Found in the notebook.
* **Precision-Recall Curve:** Limited performance on <30 due to imbalance. Located in the notebook.

A screenshot of a graph

AI-generated content may be incorrect.

**5.6Gradient Boosting**

* Sequential trees optimizing residual errors.
* **Implementation:**

A screenshot of a computer

AI-generated content may be incorrect.

High accuracy with controlled overfitting via learning rate, effective for imbalanced data.

* **Performance Visualization:**
  + **Confusion Matrix:** Strong performance across classes, especially <30. Available in the notebook.
  + **Feature Importance Plot:** Highlights num\_lab\_procedures, number\_emergency. Found in the notebook.
  + **ROC Curve:** High AUC, competitive with Random Forest. Located in the notebook.
  + **Precision-Recall Curve:** Improved recall for minority class. Available in the notebook.

A chart with numbers and a few labels

AI-generated content may be incorrect.

**5.7Gaussian Naive Bayes**

* A probabilistic model assuming feature independence.
* **Implementation:** **A screenshot of a computer program

  AI-generated content may be incorrect.**

Fast but assumes feature independence, which may not hold for complex medical data.

* **Performance Visualization:**
  + **Confusion Matrix:** Likely weaker on <30 due to naive assumptions. Available in the notebook.
  + **ROC Curve:** Lower AUC compared to ensemble methods. Found in the notebook.
  + **Precision-Recall Curve:** Poor recall for <30, reflecting limitations. Located in the notebook.

A screenshot of a computer

AI-generated content may be incorrect.

**6. Models Accuracy:  
A screenshot of a computer program

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**6.1 Visualization of Models Accuracy:  
A red and white bars

AI-generated content may be incorrect.**

**7. . Hyperparameter Tuning**

**Streamlined grid search for key parameters:**

**8. K-Means Accuracy Calculation**

Evaluates K-Means clustering against true labels by testing label mappings:

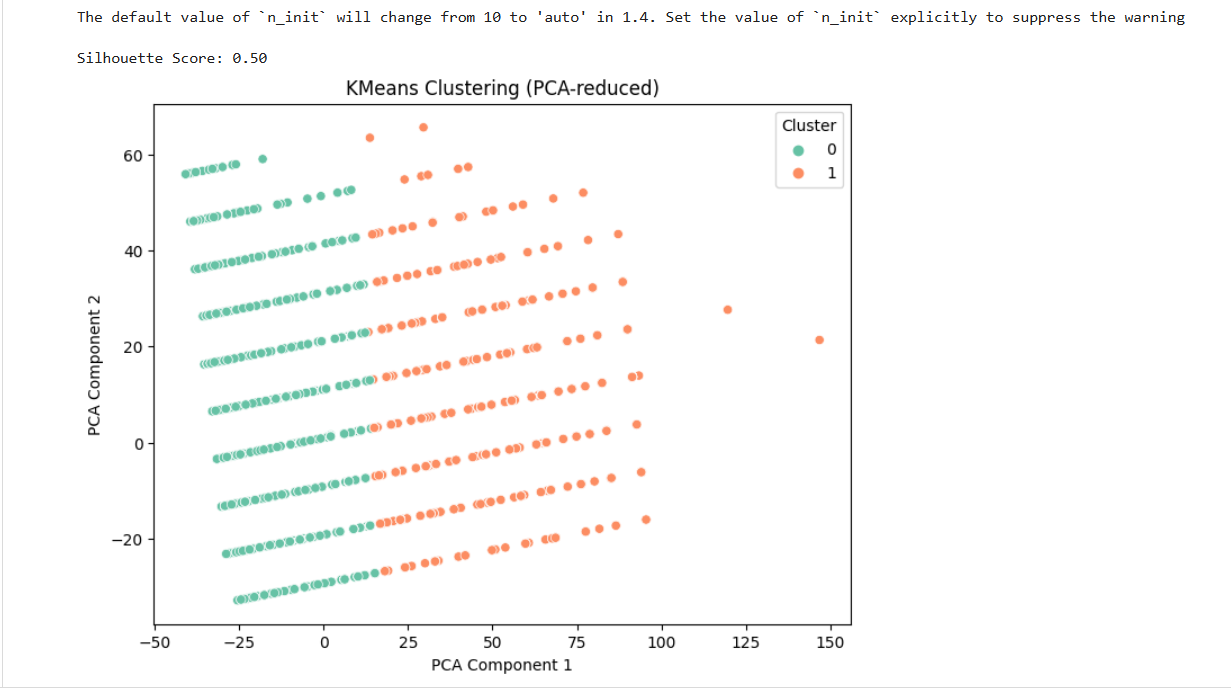
**A screenshot of a computer program

AI-generated content may be incorrect.**

**A computer screen shot of a program

AI-generated content may be incorrect.**

Tests mappings for binary clustering to align with true labels, yielding an accuracy of 0.72.

****