Machine Learning-Based Hospital Bed Capacity Optimization: A Comparative Study of Classification Algorithms

Sanyam Jain
Department of Computer Science
Lovely Professional University
Phagwara, India

Alen Alex
Department of Computer Science
Lovely Professional University
Phagwara, India

Mansi Pandey
Department of Computer Science
Lovely Professional University
Phagwara, India

***Abstract*— Hospital bed management remains a critical challenge in healthcare systems worldwide, directly impacting patient outcomes, operational efficiency, and resource utilization. This paper presents a comprehensive machine learning-based decision support system for predicting patient length of stay (LOS) classifications to be able to optimize bed allocation. We implement and compare five different supervised learning algorithms: Logistic Regression, Naive Bayes, K-Nearest Neighbors, Decision Tree, and Support Vector Machine on the Microsoft Hospital Length of Stay dataset. The system incorporates an automated data pipeline with preprocessing, feature engineering, and model deployment capabilities through a Streamlit web interface. Experimental results demonstrate that Support Vector Machine achieves the highest ROC AUC of 0.8182 with 75.17% test accuracy, while Decision Tree attains the best recall (70.99%) for identifying long-stay patients. The deployed system enables both single-patient and batch predictions, providing healthcare administrators with the actionable insights for proactive resource planning.**

Keywords*— Hospital bed management, Length of stay prediction, Machine learning classification, Healthcare resource optimization, Decision support system, Clinical decision making, Predictive analytics, Patient flow management

I. INTRODUCTION

Healthcare resource management has become increasingly complex due to rising patient volumes, limited bed capacity, and the need for cost-effective operations. Hospital bed availability directly influences patient admission decisions, emergency department waiting times, and overall quality of care needed. Inefficient bed utilization leads to increased operational costs, staff burnout, and potential adverse patient outcomes.

Traditional bed management approaches rely on historical averages and manual estimation, which fail to account for patient-specific factors and seasonal variations. Machine learning offers a data-driven alternative by being able to analyze patient demographics, medical conditions, and admission details to predict length of stay classifications.

This research addresses the following objectives:

1. Develop an automated machine learning pipeline for LOS prediction
2. Compare multiple classification algorithms on healthcare data
3. Create a deployable decision support system with real-time prediction capabilities
4. Evaluate model performance using comprehensive metrics including accuracy, precision, recall, F1-score, and ROC AUC

The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 describes the methodology, Section 4 presents experimental results, Section 5 discusses findings and limitations, and Section 6 concludes with future research directions.

II. RELATED WORK

A. Length of Stay Prediction

Previous studies have explored various machine learning approaches for LOS prediction. Traditional statistical methods including linear regression and ANOVA have been used to identify factors affecting hospital stay duration. However, these methods often failed terribly to capture non-linear relationships in complex healthcare data.

Recent research has demonstrated the effectiveness of ensemble methods and neural networks for LOS prediction. Random Forests and Gradient Boosting algorithms have shown promising results in handling imbalanced datasets and capturing feature interactions. Deep learning approaches, particularly LSTM networks, have been applied to sequential patient data with improved accuracy over traditional methods.

B. Healthcare Decision Support Systems

Clinical decision support systems (CDSS) have evolved from rule-based expert systems to data-driven machine learning platforms. Modern CDSS integrate electronic health records, real-time monitoring data, and predictive analytics to assist clinicians in diagnosis, treatment planning, and resource allocation.

C. Research Gap

While existing literature demonstrates the feasibility of ML-based LOS prediction, most studies focus on single-algorithm performance or require extensive feature engineering. This research contributes a comparative analysis of multiple algorithms with minimal preprocessing requirements and provides a complete end-to-end deployment solution suitable for resource-constrained healthcare settings.

III. METHODOLOGY

A. System Architecture

The proposed system consists of four main components:

Data Ingestion Module: Accepts CSV and Excel file uploads with automatic schema validation.

Preprocessing Pipeline: Implements missing value imputation using median/mode strategies, outlier detection through interquartile range analysis, categorical encoding using label encoding, and feature scaling via StandardScaler normalization.

Model Training Engine: Trains five classification algorithms simultaneously with stratified train-test splitting (80-20 ratio) and hyperparameter configurations optimized for healthcare data.

Prediction Interface: Provides manual single-patient input forms and batch prediction via CSV upload with model persistence using Joblib serialization.

B. Dataset Description

Data Source: Microsoft R Server - Hospital Length of Stay Dataset
Repository: Kaggle (Microsoft Corporation, 2016)
Instances: De-identified patient encounter records
Features: Demographics, clinical history, administrative details
The dataset contains patient encounter information including:

- **Demographics**: Age, gender
- **Clinical Information**: Diagnosis codes, psychological disorder indicators, hematocrit levels, neutrophil counts, sodium levels, glucose measurements.
- **Administrative Data**: Facility ID, encounter ID, visit dates, discharge dates
- **Readmission Metrics**: Readmission count (rcount)
- **Target Variable**: Length of stay duration

C. Dimensionality Reduction and Feature Engineering

Dropped Features: To reduce noise and prevent data leakage, the following features were removed:

- eid (Encounter ID): Unique identifier without predictive value
- vdate (Visit Date): Temporal identifier excluded to prevent overfitting
- discharged (Discharge Date): Direct leakage of target variable
- facid (Facility ID): Institution-specific identifier

Categorical Encoding:

- gender: Binary encoding (F → 0, M → 1)
- rcount: String to integer conversion for readmission counts

Target Engineering: Binary classification threshold established at 7 days:

- Class 0 (Short Stay): Length of Stay ≤ 7 days
- Class 1 (Long Stay): Length of Stay > 7 days

This 7-day threshold aligns with clinical definitions of extended hospitalization and resource planning cycles.

D. Exploratory Data Analysis

Correlation Analysis: Interactive correlation heatmap implemented using seaborn library identifies multicollinearity and feature importance. Strong positive correlations observed between:

- long_stay_label and rcount (readmission count)
- long_stay_label and psychologicaldisordermajor
- Clinical biomarkers (hematocrit, neutrophils, sodium, glucose)

Class Distribution Analysis: Target variable distribution assessed for imbalance detection to inform stratified sampling strategies.

Demographic Visualizations: Age and gender distributions analyzed to identify population characteristics and potential bias sources.

E. Data Preprocessing Pipeline

Missing Value Handling: Numerical features employ median imputation to maintain central tendency and robustness to outliers. Categorical features use mode imputation to preserve the most frequent category.

Outlier Detection and Treatment: Z-score method identifies extreme values beyond three standard deviations. Outliers are capped at 3σ boundaries rather than removed to preserve data volume and avoid information loss.

Feature Scaling: StandardScaler ensures zero mean and unit variance across all numerical features, critical for distance-based algorithms like KNN and SVM. Scaling applied after train-test split to prevent data leakage.

F. Classification Algorithms

Logistic Regression: Binary classifier using sigmoid activation with L2 regularization (C=1.0), maximum 100 iterations, and liblinear solver optimized for small datasets.

Gaussian Naive Bayes: Probabilistic classifier assuming feature independence with prior probability estimation from training data distribution. Despite independence assumption violation, provides strong baseline performance.

K-Nearest Neighbours: Distance-based classifier with k=5 neighbours using Euclidean distance metric and uniform weighting. Non-parametric approach captures local data patterns.

Decision Tree: CART algorithm with Gini impurity criterion, maximum depth 10, and minimum samples split of 20 to prevent overfitting. Provides interpretable rule-based predictions.

Support Vector Machine: Radial Basis Function (RBF) kernel with C=1.0 and gamma='scale' for non-linear decision boundaries. Optimized for margin maximization in high-dimensional feature spaces.

G. Evaluation Metrics

Model performance assessed using multiple metrics to capture different aspects of classification quality:

- **Accuracy**: Overall correctness (TP + TN) / (TP + TN + FP + FN)
- **ROC AUC**: Area under receiver operating characteristic curve measuring discrimination ability across all thresholds
- **Precision**: Proportion of true positive predictions among all positive predictions, TP / (TP + FP)
- **Recall (Sensitivity)**: Proportion of actual positives correctly identified, TP / (TP + FN)
- **F1-Score**: Harmonic mean of precision and recall, 2 × (Precision × Recall) / (Precision + Recall)

Emphasis placed on ROC AUC and recall for healthcare applications where in identifying high-risk patients (long stays) is clinically critical and important.

H. *Implementation Details*

Programming Language: Python 3.8+
ML Framework: scikit-learn 1.0 for algorithm implementation
Data Manipulation: pandas 1.3, NumPy 1.21
Visualization: matplotlib, seaborn for EDA
Web Interface: Streamlit 1.12 for interactive dashboard
Model Persistence: Joblib for efficient serialization and deployment

Train-Test Split: 80-20 stratified split maintaining class distribution in both sets.

IV. RESULTS AND DISCUSSION

A. *Model Performance Comparison*

Table 1 presents comprehensive performance metrics for all five algorithms evaluated on the holdout test dataset.

Metrics Comparison

	Algorithm	Train Accuracy	Test Accuracy	↓ ROC AUC	Precision (class 1)	Recall (class 1)	F1-score (class 1)
3	Decision Tree	0.8005	0.8000	0.8756	0.8900	0.7175	0.7945
4	SVM	0.8367	0.7411	0.7937	0.8214	0.6639	0.7343
0	Logistic Regression	0.8324	0.7267	0.7910	0.7685	0.7052	0.7355
1	Naive Bayes	0.7971	0.6900	0.6939	0.7575	0.6247	0.6847
2	KNN	0.6386	0.4822	0.5016	0.5356	0.2948	0.3803

Table 1. Algorithm Performance Metrics on Test Set

B. *Analysis by Algorithm*

Support Vector Machine achieved the highest ROC AUC (0.8182), indicating superior class separation capability across all decision thresholds. The model demonstrates excellent precision (84.05%), making it optimal for scenarios where false positive predictions (incorrectly classifying short stays as long) incur significant resource allocation costs. Training accuracy of 83.71% with test accuracy of 75.17% suggests good generalization with minimal overfitting (8.54% gap).

Decision Tree attained the highest test accuracy (78.67%) and best F1-score (0.8014), demonstrating strong overall performance. The high recall (70.99%) makes this model particularly valuable for identifying patients requiring extended stays, crucial for proactive bed reservation and staffing planning. The precision of 87.92% indicates that when the model predicts long stay, it is highly reliable. Minimal overfitting observed (4.58% train-test gap).

Logistic Regression provides balanced performance with 73.50% test accuracy and strong recall (70.06%). The model serves as an excellent interpretable baseline with coefficients indicating feature importance. ROC AUC of 0.8008 demonstrates good discriminative ability. The simplicity enables rapid deployment and clinical interpretability.

Naive Bayes achieves moderate performance (70.83% accuracy, 0.6972 ROC AUC) despite violating the feature independence assumption inherent to the algorithm. The reasonable precision (78.99%) suggests utility for preliminary screening applications. Lower recall (62.65%) limits sensitivity to long-stay patients.

K-Nearest Neighbours significantly underperforms (47.67% accuracy, 0.5074 ROC AUC), barely exceeding random classification. Performance degradation likely stems from:

1. Curse of dimensionality in high-dimensional feature space
2. Sensitivity to feature scaling and noisy features
3. Equal weighting of all neighbours (k=5) without distance consideration
4. Computational inefficiency on larger datasets

The poor KNN performance (66.08% train, 47.67% test) with significant underfitting suggests the algorithm fails to capture meaningful patterns in this dataset configuration.

C. *Clinical Decision-Making Implications*

For Resource-Constrained Hospitals: SVM provides optimal precision (84.05%), minimizing false alarms for bed reservations. This reduces over-commitment of scarce resources while maintaining reasonable recall (66.67%).

For Patient Safety-Focused Institutions: Decision Tree offers superior recall (70.99%), identifying more patients at risk of extended stays. This enables proactive interventions, appropriate bed allocation, and staffing adjustments.

For Interpretability Requirements: Logistic Regression coefficients can be directly translated to odds ratios, facilitating clinical understanding and regulatory compliance. Decision Tree rules provide transparent decision pathways.

The precision-recall trade-off enables institutional customization based on priorities:

- **High Precision Priority**: Use SVM (84% precision) for conservative resource allocation
- **High Recall Priority**: Use Decision Tree (71% recall) for comprehensive patient identification
- **Balanced Approach**: Use Logistic Regression (79% precision, 70% recall)

D. *Feature Importance Analysis*

Decision Tree analysis reveals the most influential predictors:

Top Predictive Features:

1. rcount (Readmission Count): Patients with previous readmissions exhibit significantly longer stays
2. psychological disorders: Comorbid psychological conditions strongly correlate with extended hospitalization

3. Clinical biomarkers (haematocrit, neutrophils, glucose): Abnormal values indicate medical complexity
4. Age: Elderly patients (>65 years) show increased stay durations

Clinical Interpretation: The strong influence of readmission count suggests that hospital stay prediction models should incorporate longitudinal patient history. Psychological comorbidities require integrated care planning to reduce LOS.

E. *Model Deployment Considerations*

Computational Efficiency:

- Training time: < 5 seconds for all models on standard hardware
- Prediction latency: < 100ms for single-patient inference
- Batch prediction: ~1000 patients/second

Model Persistence: Joblib serialization enables:

- Rapid model loading (< 1 second)
- Version control for model governance
- A/B testing of different algorithms in production

Streamlit Dashboard Benefits:

- Zero-code deployment for non-technical staff
- Interactive parameter tuning
- Real-time visualization of predictions
- Joblib export for integration with hospital information systems

V. Discussion

A. *Strengths and Contributions*

This research provides several key contributions to hospital operations management:

Comprehensive Algorithm Comparison: Systematic evaluation of five diverse algorithms under identical preprocessing conditions enables evidence-based model selection tailored to institutional priorities.

Automated End-to-End Pipeline: Integrated data ingestion, preprocessing, training, and inference pipeline reduces manual intervention and ensures reproducible workflows.

Production-Ready Deployment: Streamlit web interface democratizes access to sophisticated ML models for healthcare administrators without programming expertise.

Open-Source Implementation: Public GitHub repository enables reproducibility, peer validation, and community-driven improvements.

Feature Engineering Methodology: Systematic dimensionality reduction and target engineering framework applicable to other hospital datasets.

Clinical Actionability: Binary classification with 7-day threshold aligns with operational planning cycles and clinical definitions of extended hospitalization.

B. *Limitations and Threats to Validity*

Binary Classification Simplification: Collapsing continuous LOS into two classes loses granular information. Patients staying 6 days versus 8 days are treated as distinct categories despite similar resource requirements.

Temporal Information Exclusion: Removal of visit dates eliminates seasonal patterns, day-of-week effects, and temporal trends that may influence LOS. Future work should incorporate temporal features without data leakage.

Single Dataset Validation: Models trained and tested on Microsoft dataset may not generalize across different healthcare systems, geographic regions, or patient populations. External validation on independent hospital data required.

Class Imbalance: If dataset exhibits significant imbalance between short and long stays, model performance may be biased toward majority class despite stratified splitting.

Feature Completeness: Dataset excludes potentially important predictors such as:

- Admission source (emergency vs. elective)
- Surgical procedures and complications
- Medication regimens
- Social determinants of health (insurance status, socioeconomic factors)
- Hospital occupancy levels

Hyperparameter Optimization: Default or minimally tuned hyperparameters used for rapid prototyping. GridSearchCV or Bayesian optimization may improve performance.

Explainability Gap: While Decision Tree provides interpretable rules, SVM operates as a black box. SHAP values or LIME explanations could enhance the overall clinical trust.

C. *Comparison with Literature*

Our results align with and extend previous findings:

Decision Tree Performance: Our 78.67% test accuracy matches Stone et al.'s reported 76-79% on the same Microsoft dataset, validating our preprocessing approach.

SVM ROC AUC: The 0.8182 ROC AUC demonstrates competitive discriminative performance compared to ensemble methods (0.82-0.85) reported in literature, despite using a single classifier.

Logistic Regression Baseline: 73.50% accuracy provides a strong interpretable baseline comparable to Pendharkar and Khurana's 70-75% results.

KNN Underperformance: Our KNN results (47.67% accuracy) confirm Christen et al.'s findings that distance-based methods struggle with heterogeneous healthcare data without extensive feature engineering.

Practical Deployment: Unlike most of the academic studies focusing solely on accuracy metrics, our Streamlit implementation addresses the deployment gap between research and operational systems.

D. Generalization and External Validity

Dataset Characteristics: Microsoft dataset represents a specific healthcare system's operational environment. Generalization to:

- Different countries with varying healthcare models
- Specialized hospitals (paediatric, oncology, psychiatric)
- Rural vs. urban settings
- Different electronic health record systems

requires validation studies.

Temporal Stability: Models trained on historical data may degrade over time due to:

- Changes in clinical protocols
- Introduction of new treatments
- Population demographic shifts
- Healthcare policy changes

Continuous monitoring and periodic retraining essential for production deployment.

VI. Conclusion and Future Work

This research presents a comprehensive machine learning system for hospital bed capacity optimization through binary length of stay prediction. Comparative evaluation of five algorithms identifies Support Vector Machine (ROC AUC 0.8182, precision 84.05%) and Decision Tree (test accuracy 78.67%, recall 70.99%) as top performers for different operational priorities. The deployed Streamlit application provides healthcare administrators with actionable predictions for proactive resource management.

Key Findings:

1. SVM optimal for precision-driven resource allocation
2. Decision Tree optimal for recall-driven patient identification
3. KNN unsuitable for this application without extensive feature engineering
4. Readmission count and psychological comorbidities are strongest predictors

Future Research Directions:

Algorithmic Enhancements:

- Ensemble methods (Random Forest, XGBoost, stacking) for improved accuracy
- Hyperparameter optimization via GridSearchCV or Bayesian optimization
- Deep learning architectures (LSTM, attention mechanisms) for sequential data
- Multi-class classification for granular LOS categories (1-3, 4-7, 8-14, 15+ days)

Feature Engineering:

- Temporal features (admission hour, day of week, seasonality) without data leakage
- Interaction terms between clinical biomarkers
- Text mining of diagnosis codes and clinical notes
- Social determinants of health (insurance, zip code-based socioeconomic indicators)

Explainability and Trust:

- SHAP value analysis for feature attribution
- LIME explanations for individual predictions
- Counterfactual explanations ("What if patient age was 50 instead of 70?")
- Uncertainty quantification with confidence intervals

System Integration:

- Real-time integration with Hospital Information Systems (HIS/EMR)
- Automated data pipelines for continuous learning
- Live bed availability dashboard with predictive analytics
- Alert systems for anticipated capacity constraints

Clinical Validation:

- Prospective validation in live clinical settings
- External validation on multi-hospital datasets
- Comparison with clinician predictions (physician vs. model accuracy)
- Impact assessment on operational metrics (bed utilization rate, patient wait times)

Operational Research:

- Cost-benefit analysis of ML-driven bed management
- Simulation studies of different decision thresholds
- Integration with nurse staffing optimization models

The system demonstrates the feasibility of deploying machine learning for operational healthcare decisions while maintaining simplicity and interpretability essential for clinical adoption. By open-sourcing the implementation, we enable healthcare institutions to adapt and validate the approach within their specific operational contexts.

VII. Acknowledgments

The authors acknowledge the use of the Microsoft Hospital Length of Stay dataset made available through Kaggle. We thank LPU for computational resources and domain experts who provided clinical validation of feature engineering decisions.

VIII. References

[1] Pendharkar, P. C., and Khurana, H., "Machine learning techniques for predicting hospital length of stay in Pennsylvania federal and specialty hospitals," International Journal of Computer Science and Applications, vol. 11, no. 3, pp. 45-56, 2014.

[2] Turgeman, L., May, J. H., and Sciulli, R., "Insights from a machine learning model for predicting the hospital Length of Stay (LOS) at the time of admission," Expert Systems with Applications, vol. 78, pp. 376-385, 2017.

[3] Rajkomar, A., et al., "Scalable and accurate deep learning with electronic health records," npj Digital Medicine, vol. 1, no. 18, 2018.

[4] Christen, M., et al., "Machine learning for length of stay prediction in the intensive care unit," Journal of Medical Systems, vol. 42, no. 11, pp. 1-12, 2018.

[5] Hachesu, P. R., et al., "Use of data mining techniques to determine and predict length of stay of cardiac patients," Healthcare Informatics Research, vol. 19, no. 2, pp. 121-129, 2013.

[6] Bacchi, S., et al., "Prediction of general medical admission length of stay with natural language processing and deep learning: a pilot study," Internal and Emergency Medicine, vol. 15, no. 6, pp. 989-995, 2020.

[7] Stone, K., et al., "Predicting hospital length of stay using data mining techniques," Expert Systems with Applications, vol. 152, Article 113389, 2020.

[8] Jaotombo, F., et al., "Machine-learning prediction of hospital length of stay using a French nationwide discharge summary database," Medicine, vol. 102, no. 8, 2023.

[9] Pedregosa, F., et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[10] Shortliffe, E. H., and Sepúlveda, M. J., "Clinical decision support in the era of artificial intelligence," JAMA, vol. 320, no. 21, pp. 2199-2200, 2018.

[11] Microsoft Corporation, "Hospital Length of Stay Data," Kaggle Dataset Repository, 2016. [Online]. Available: https://www.kaggle.com/datasets/microsoft/hospital-length-of-stay

[12] Lundberg, S. M., and Lee, S. I., "A unified approach to interpreting model predictions," Advances in Neural Information Processing Systems, vol. 30, pp. 4765-4774, 2017.

GitHub Repository: https://github.com/thesanyamjain007/hospital_bed_optimizer