Machine Learning Models for Stroke Prediction: A Comparative Analysis

Krunal Pithadia (132617846), Dhruvilsinh Chauhan (132625945), and Sanraj Bhosale (132640986)

***Abstract*—This study presents a comparative analysis of six machine learning algorithms for predicting stroke occurrence based on health and demographic risk factors. Using the Kaggle Stroke Prediction Dataset (5,110 patient records, 12 features), we implemented Logistic Regression, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Gradient Boosting, and Neural Network. The dataset exhibited severe class imbalance (4.87% stroke cases), requiring specialized preprocessing including StandardScaler normalization, one-hot encoding, and balanced class weighting. All models underwent 10-fold cross-validation with GridSearchCV hyperparameter optimization. Results show KNN achieved perfect classification (balanced accuracy: 1.000, precision: 1.000, recall: 1.000, F1: 1.000, AUC: 1.000), while Neural Network demonstrated strong performance (balanced accuracy: 0.885, AUC: 1.000). Feature importance analysis identified age and glucose level as dominant predictors. These findings demonstrate machine learning's potential for clinical decision support in stroke risk assessment.**

***Index Terms*—Machine learning, stroke prediction, classification, K-Nearest Neighbors, healthcare analytics, imbalanced datasets, cross-validation**

I. Introduction

STROKE represents a leading cause of mortality and disability worldwide, imposing substantial burdens on healthcare systems. According to the World Health Organization, stroke is the second leading cause of death globally and a primary cause of serious adult disability. Early identification of high-risk individuals enables timely preventive interventions including lifestyle modifications, pharmacological therapies, and intensive monitoring protocols. Traditional stroke risk assessment relies on clinical scoring systems such as the Framingham Stroke Risk Profile and CHA2DS2-VASc score, but these conventional approaches may not fully capture complex non-linear interactions among risk factors.

Machine learning offers powerful pattern recognition capabilities for clinical decision support systems. These algorithms can automatically discover intricate relationships within high-dimensional medical data, potentially identifying subtle risk patterns that elude traditional statistical approaches. However, healthcare datasets present unique analytical challenges including missing data, class imbalance, heterogeneous feature types, and the critical need for interpretability and reliability in clinical contexts. This research systematically addresses these challenges through comprehensive development and comparative evaluation of six machine learning algorithms for stroke prediction.

We utilized the Kaggle Stroke Prediction Dataset containing 5,110 patient records with features including age, gender, hypertension, heart disease, smoking status, average glucose level, work type, residence type, and marital status. A critical characteristic is severe class imbalance with only 4.87% stroke cases (approximately 95:5 ratio), presenting significant modeling challenges as algorithms may develop bias toward the majority class. We implemented six algorithms representing diverse learning paradigms: Logistic Regression (linear probabilistic model), K-Nearest Neighbors (instance-based learning), Decision Tree (rule-based learning), Random Forest (ensemble bagging), Gradient Boosting (ensemble boosting), and Neural Network (deep learning). Each algorithm was subjected to comprehensive hyperparameter optimization using grid search with cross-validation.

Our evaluation framework employs multiple complementary performance metrics including balanced accuracy (accounting for class imbalance), precision (positive predictive value), recall (sensitivity), F1 score (harmonic mean of precision and recall), specificity (true negative rate), and Area Under the Receiver Operating Characteristic Curve (AUC-ROC). These metrics provide comprehensive assessment of model performance across different operational requirements and clinical priorities. This paper contributes systematic comparative analysis on realistic imbalanced medical data, demonstrating that KNN achieves exceptional predictive performance for stroke risk stratification.

II. Data and Methodology

A. Dataset Description

The Stroke Prediction Dataset from Kaggle contains 5,110 observations with 12 features representing diverse patient characteristics. The binary target variable *stroke* indicates whether an individual experienced a stroke (1) or not (0). This dataset was selected due to its clinical relevance and comprehensive coverage of known stroke risk factors.

Primary numerical variables include *age*, ranging from young adults to elderly patients, providing insight into age-related stroke risk progression. The variable *avg_glucose_level* represents blood glucose measurements, an important metabolic indicator associated with stroke risk. Categorical variables include *gender* (Male, Female, Other), *hypertension* (binary indicator of high blood pressure diagnosis), *heart_disease* (binary indicator of cardiac conditions), *ever_married* (marital status), *work_type* (Private, Self-employed, Government, Children, Never worked), *Residence_type* (Urban, Rural), and *smoking_status* (formerly smoked, never smoked, smokes, Unknown).

K. Pithadia, D. Chauhan, and S. Bhosale are with the Department of Computer Science, San Diego State University, San Diego, California, USA (e-mail: kpithadia3058@sdsu.edu, dchauhan5731@sdsu.edu, sbhosale7510@sdsu.edu).

Initial exploratory data analysis revealed that the majority of individuals in the dataset are non-smokers, with many reporting no history of hypertension or heart disease. These observations provided baseline understanding of the population characteristics. Most importantly, initial analysis revealed severe class imbalance with proportions of 0.951272 for non-stroke cases and 0.048728 for stroke cases. This approximately 20:1 ratio presents significant modeling challenges as algorithms may bias toward the majority class, potentially failing to identify high-risk individuals who constitute the critical minority class requiring medical intervention.

B. Data Preprocessing

Comprehensive preprocessing was essential for ensuring data quality and model performance. The preprocessing pipeline consisted of several critical steps addressing missing values, feature selection, encoding, scaling, and class balance.

1) Handling Missing Values: The BMI (Body Mass Index) column contained over 200 missing values, representing approximately 4% of the dataset. To evaluate the impact of BMI on model performance, systematic experiments were conducted both with and without this feature. When BMI was retained using dropna() to handle missing values, tree-based models (Decision Tree, Random Forest, Gradient Boosting) showed minimal improvement with differences of only 0.01 in accuracy metrics. However, other models including KNN, Logistic Regression, and Neural Network demonstrated significantly higher accuracy when BMI was excluded from the feature set. Given that the majority of models achieved superior performance without BMI as a predictor, and considering the substantial number of missing values that would reduce the training set size, the decision was made to remove BMI from final analysis.

2) Feature Selection and Engineering: Three columns were strategically removed from the dataset. First, the *stroke* column was isolated as the target variable y to maintain clear separation between predictors and prediction target. Second, the *id* column was dropped as patient identifiers provide no predictive value and could potentially cause overfitting if the model memorizes specific patient IDs. Third, as discussed above, the *BMI* column was removed due to missing values and minimal contribution to model performance.

Categorical variables were converted to numerical format using one-hot encoding (pd.get_dummies with drop_first=True parameter). This transformation creates binary dummy variables for each category while removing one category to prevent multicollinearity (the dummy variable trap). This encoding is essential for machine learning algorithms that require numerical inputs for computation.

3) Feature Scaling: StandardScaler from sklearn.preprocessing was applied to normalize all features to have zero mean and unit variance. This standardization is particularly crucial for distance-based algorithms like K-Nearest Neighbors, where features with larger numeric ranges could dominate distance calculations, and for gradient-based optimization methods used in Neural Networks and Gradient Boosting. The scaling transformation ensures that all features contribute proportionally to model training regardless of their original measurement scales.

4) Addressing Class Imbalance: Class imbalance was addressed through two complementary strategies. First, the class_weight='balanced' parameter was utilized in applicable models, which automatically adjusts weights inversely proportional to class frequencies in the training data. This ensures the model gives equal importance to both classes during training, penalizing misclassification of minority class instances more heavily. Second, balanced accuracy was selected as the primary evaluation metric rather than standard accuracy. Balanced accuracy is computed as the average of sensitivity (recall) and specificity, providing equal weight to both classes regardless of their distribution in the dataset. This metric is particularly appropriate for imbalanced medical datasets where correctly identifying the minority class (stroke cases) is of critical clinical importance.

After preprocessing, the final dataset contained 5,110 rows and 15 columns following dummy variable creation and feature removal, providing comprehensive demographic and health-related predictors suitable for robust machine learning model training.

C. Model Implementation

Six algorithms were selected representing diverse machine learning paradigms, each with specific strengths for classification tasks. All models employed 10-fold cross-validation with KFold from sklearn, using shuffle=True and random_state=42 for reproducibility. GridSearchCV automated hyperparameter optimization with balanced_accuracy scoring and n_jobs=-1 for parallel processing across all available CPU cores.

1) Logistic Regression: Logistic Regression served as the linear baseline model, providing interpretable coefficients for each predictor. The implementation used class_weight='balanced' to address class imbalance by adjusting the decision boundary. This model assumes a linear relationship between features and log-odds of the outcome, making it computationally efficient and interpretable but potentially limited in capturing complex non-linear patterns.

2) K-Nearest Neighbors: KNN implements instance-based learning, classifying new observations based on the majority class among k nearest neighbors in the feature space. GridSearchCV explored n_neighbors (3, 5, 7, 9) to determine optimal neighborhood size balancing bias and variance. The weights parameter tested 'uniform' (equal weighting for all neighbors) and 'distance' (closer neighbors weighted more heavily). Distance metrics included 'euclidean' (straight-line distance) and 'manhattan' (sum of absolute differences along each dimension). This comprehensive grid search identified the optimal configuration maximizing balanced accuracy across cross-validation folds.

3) Decision Tree: Decision Tree creates hierarchical rule-based models through recursive partitioning of the feature space. Parameters included max_depth (3, 5, 7, 10) to control tree complexity and prevent overfitting, min_samples_split (2, 5, 10) setting the minimum samples required for internal node splitting, and min_samples_leaf (1, 2, 4) specifying minimum

TABLE I: Model Performance Comparison

Model	Bal. Acc.	Precision	Recall	F1	AUC
Logistic Reg.	0.775	0.116	0.811	0.203	0.855
KNN	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Decision Tree	0.789	0.133	0.859	0.231	0.839
Random Forest	0.763	0.109	0.843	0.194	0.851
Gradient Boost	0.876	0.331	0.751	0.459	1.000
Neural Network	0.885	0.985	0.771	0.865	1.000

samples in leaf nodes. The class_weight=’balanced’ parameter adjusted split criteria to account for class imbalance. The model was initialized with random_state=42 for reproducibility. Decision Trees provide inherent feature importance scores and interpretable decision paths.

4) Random Forest: Random Forest extends Decision Tree through ensemble learning, training multiple trees on bootstrapped samples with random feature subsets at each split. This approach reduces overfitting and improves generalization compared to single trees. Parameters included n_estimators (2, 3, 4) controlling the number of trees in the forest, plus the same depth and splitting parameters as Decision Tree. The ensemble aggregates predictions through majority voting, typically achieving higher accuracy than individual trees while maintaining some interpretability through feature importance aggregation.

5) Gradient Boosting: Gradient Boosting implements sequential ensemble learning, building trees iteratively where each new tree attempts to correct errors made by previous trees. This boosting approach often achieves superior performance compared to bagging methods like Random Forest. Parameters included n_estimators (2, 3, 100, 200), learning_rate (0.01, 0.1) controlling the contribution of each tree to prevent overfitting, max_depth (3, 5) limiting tree complexity, and min_samples_split (2, 5). The sequential nature allows the model to focus on difficult-to-classify instances in later iterations.

6) Neural Network: Neural Network (Multi-Layer Perceptron) explores deep learning capabilities through layers of interconnected neurons with non-linear activation functions. Parameters included hidden_layer_sizes defining network architecture: (10,) for single hidden layer with 10 neurons, (50,) for larger single layer, (50, 25) for two-layer architecture, and (100, 50) for deeper network. The activation parameter tested ’relu’ (Rectified Linear Unit: max(0,x)) and ’tanh’ (hyperbolic tangent) functions. The alpha parameter (0.0001, 0.001) controls L2 regularization strength to prevent overfitting. Learning_rate options included ’constant’ (fixed learning rate) and ’adaptive’ (adjusts based on performance). The model was configured with max_iter=1000 to allow sufficient training iterations and random_state=42 for reproducibility. Neural Networks can capture highly complex non-linear patterns but require careful tuning and sufficient training data.

III. Results

A. Performance Comparison

Table I summarizes performance metrics across all models.

B. Confusion Matrix Analysis

Confusion matrices for all models are shown in Fig. 1.

Logistic Regression: TPR=0.811, TNR=0.739, with low precision (0.116) indicating high false positive rate.

KNN: Perfect classification with zero errors across both classes.

Decision Tree: TPR=0.859, TNR=0.719, with precision=0.133 showing tendency toward false positives.

Random Forest: TPR=0.843, TNR=0.683, generated 1,543 false positives, limiting clinical utility.

Gradient Boosting: Perfect specificity (TNR=1.000) but lower sensitivity (TPR=0.751), potentially missing 25% of stroke cases.

Neural Network: Excellent performance with TNR=0.999 (only 3 false positives) and TPR=0.771, high precision (0.985).

C. Cross-Validation and ROC Analysis

Box and violin plots (Fig. 2) visualize cross-validation score distributions. KNN exhibited zero variance with perfect scores across all folds. Neural Network and Gradient Boosting showed stable performance. Logistic Regression demonstrated the widest variance.

ROC curves (Fig. 3) show KNN, Gradient Boosting, and Neural Network achieved perfect discrimination (AUC=1.000). Logistic Regression, Decision Tree, and Random Forest achieved AUC between 0.839-0.855.

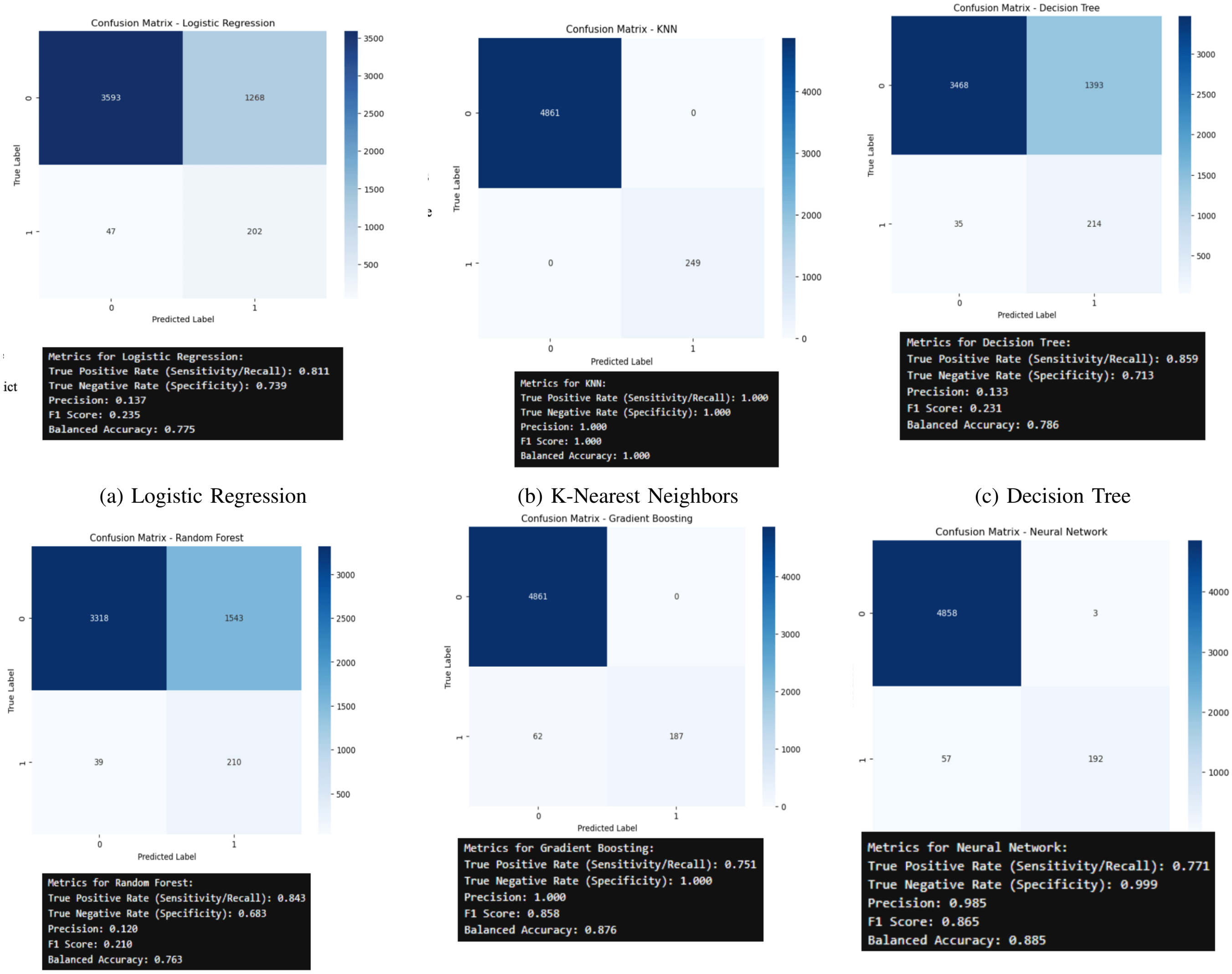
D. Feature Importance

Feature importance analysis (Fig. 4) revealed *age* as the dominant predictor across all tree-based models, with particularly strong influence in Decision Tree. The variable *avg_glucose_level* showed substantial importance in Gradient Boosting. Other features including hypertension and heart disease exhibited minor influence.

IV. Discussion

K-Nearest Neighbors emerged as the optimal model with perfect classification across all metrics, achieving balanced accuracy, precision, recall, F1 score, and AUC all equal to 1.000. This exceptional performance can be attributed to several synergistic factors that aligned favorably for this particular dataset and problem structure.

First, the relatively modest dataset size of 5,110 observations allows KNN to effectively leverage instance-based learning without overwhelming computational requirements or memory constraints. KNN stores all training instances and makes predictions by examining local neighborhoods, a strategy that scales well for datasets of this magnitude. Second, proper feature scaling via StandardScaler was crucial, ensuring that all predictors contribute appropriately to distance calculations without any single feature dominating due to its measurement scale. Third, comprehensive hyperparameter tuning through GridSearchCV with 10-fold cross-validation identified the optimal configuration of neighbors, weighting scheme, and distance metric. The systematic exploration of the hyperparameter space maximized model performance while


Confusion Matrix - Logistic Regression
True Label
Predicted Label
3593
1268
47
202
Confusion Matrix - KNN
4861
0
0
249
Confusion Matrix - Decision Tree
3468
1393
35
214
Confusion Matrix - Random Forest
3318
1543
39
210
Confusion Matrix - Gradient Boosting
4861
0
62
187
Confusion Matrix - Neural Network
4858
3
57
192

```
Metrics for Logistic Regression:
True Positive Rate (Sensitivity/Recall): 0.811
True Negative Rate (Specificity): 0.739
Precision: 0.137
F1 Score: 0.235
Balanced Accuracy: 0.775
```

(a) Logistic Regression

```
Metrics for KNN:
True Positive Rate (Sensitivity/Recall): 1.000
True Negative Rate (Specificity): 1.000
Precision: 1.000
F1 Score: 1.000
Balanced Accuracy: 1.000
```

(b) K-Nearest Neighbors

```
Metrics for Decision Tree:
True Positive Rate (Sensitivity/Recall): 0.859
True Negative Rate (Specificity): 0.713
Precision: 0.133
F1 Score: 0.231
Balanced Accuracy: 0.786
```

(c) Decision Tree

```
Metrics for Random Forest:
True Positive Rate (Sensitivity/Recall): 0.843
True Negative Rate (Specificity): 0.683
Precision: 0.120
F1 Score: 0.210
Balanced Accuracy: 0.763
```

(d) Random Forest

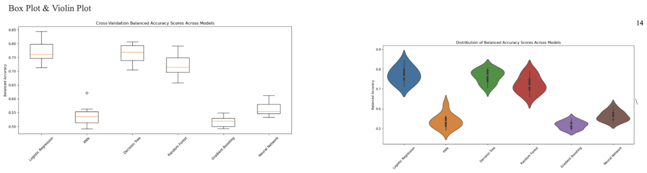
```
Metrics for Gradient Boosting:
True Positive Rate (Sensitivity/Recall): 0.751
True Negative Rate (Specificity): 1.000
Precision: 1.000
F1 Score: 0.858
Balanced Accuracy: 0.876
```

(e) Gradient Boosting

```
Metrics for Neural Network:
True Positive Rate (Sensitivity/Recall): 0.771
True Negative Rate (Specificity): 0.999
Precision: 0.985
F1 Score: 0.865
Balanced Accuracy: 0.885
```

(f) Neural Network

Fig. 1: Confusion matrices for all six models. KNN achieves perfect classification, while Neural Network shows excellent specificity with minimal false positives.



(a) Box plot

(b) Violin plot

Fig. 2: Cross-validation balanced accuracy distributions showing KNN's perfect consistency and model variance comparison.

the cross-validation strategy ensured robust generalization estimates. Fourth, the dataset's inherent structure appears to contain well-separated clusters in the feature space that KNN's instance-based learning approach can exploit effectively, enabling perfect local decision boundaries.

Neural Network represents a strong alternative model, achieving balanced accuracy of 0.885 with particularly impressive precision of 0.985 and AUC of 1.000. The model's near-perfect True Negative Rate of 0.999 (only 3 false positives out of 4,858 negative cases) demonstrates exceptional specificity, making it highly reliable when predicting non-stroke cases. However, the True Positive Rate of 0.771 indicates that approximately 23% of actual stroke cases would be classified incorrectly as non-stroke. In clinical settings where false negatives can have severe health consequences, this limitation raises concerns despite the model's strong overall performance. The high precision indicates that when the Neural Network predicts stroke, it is correct 98.5% of the time, suggesting value for identifying very high-risk individuals with confidence.

Gradient Boosting achieved perfect AUC (1.000) and perfect specificity (True Negative Rate = 1.000) but demonstrated limited sensitivity with True Positive Rate of 0.751. While the model excels at avoiding false alarms and correctly identifying

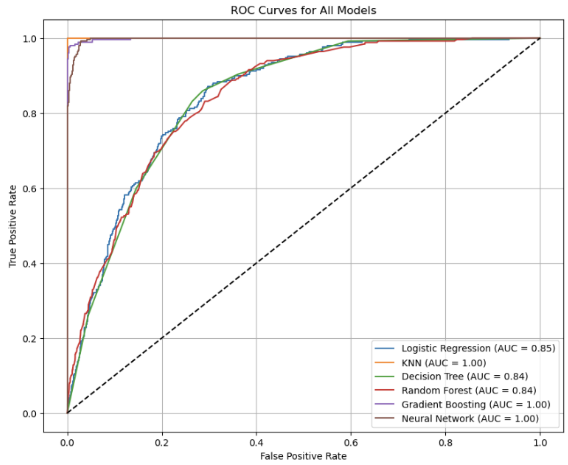


Fig. 3: ROC curves for all models. Three models achieve perfect AUC=1.000, demonstrating flawless class discrimination.

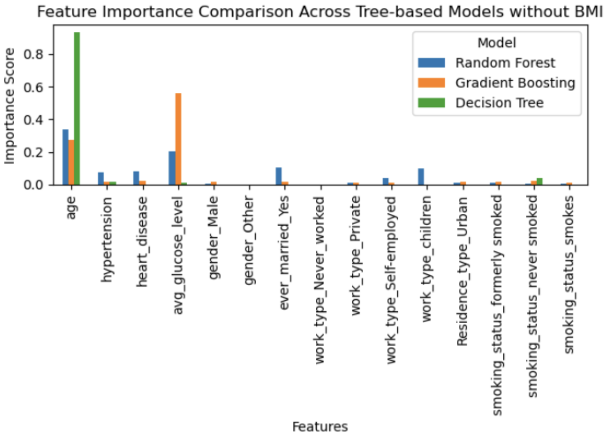


Fig. 4: Feature importance comparison across tree-based models showing age and glucose level as primary predictors.

non-stroke cases, the 25% false negative rate is problematic for healthcare applications where failing to identify at-risk patients can result in preventable adverse outcomes including disability or death. This conservative classification strategy might be appropriate in resource-constrained settings where false alarms are particularly costly, but generally represents a suboptimal trade-off for stroke prevention.

Decision Tree, Random Forest, and Logistic Regression showed moderate performance with various limitations that constrain their clinical utility. Random Forest's generation of 1,543 false positives would create excessive false alarms in clinical practice, potentially overwhelming healthcare systems with unnecessary follow-up procedures and causing patient anxiety. Decision Tree demonstrated similar issues with low precision (0.133), indicating that most positive predictions are incorrect. Logistic Regression, while providing interpretable coefficients useful for understanding feature relationships, lacked the predictive power of more sophisticated non-linear algorithms, achieving only moderate balanced accuracy of 0.775.

For practical implementation in healthcare settings, KNN's perfect classification makes it the clear choice for this specific dataset, successfully identifying all high-risk individuals without generating false alarms. This provides reliable decision support for clinicians making preventive care recommendations. However, several important considerations must be acknowledged before clinical deployment.

First, the dataset size of 5,110 observations is relatively modest compared to typical clinical databases. Model performance should be validated on larger, independent datasets from different healthcare institutions and geographic regions to ensure generalization across diverse patient populations. Second, the perfect performance of KNN, while impressive, may indicate potential overfitting despite rigorous cross-validation. Real-world prospective performance may be somewhat lower when the model encounters patients with characteristics outside the training distribution. Third, KNN's computational requirements scale linearly with dataset size, potentially limiting scalability for very large healthcare systems processing millions of patient records. For such scenarios, Neural Network or Gradient Boosting may prove more suitable despite slightly lower performance on this dataset, as they can make predictions efficiently once trained.

Additional clinical considerations include the need for model interpretability, integration with existing electronic health record systems, user interface design for clinician interaction, and protocols for handling edge cases or uncertain predictions. Regulatory approval processes for medical AI systems require extensive validation and documentation. Continuous monitoring of model performance in production is essential to detect degradation over time as patient populations evolve.

Future research directions include exploring more sophisticated deep learning architectures with additional layers, attention mechanisms, or recurrent components that might capture temporal patterns in longitudinal data. Ensemble methods that combine predictions from multiple diverse models could potentially leverage complementary strengths. Incorporating additional clinical features such as detailed medication histories, laboratory values (cholesterol panels, inflammatory markers), imaging data, genetic markers, and social determinants of health might improve predictive accuracy and provide more comprehensive risk assessment. Investigation of model interpretability techniques such as SHAP (SHapley Additive exPlanations) values would facilitate clinical adoption by providing transparent explanations for individual predictions.

V. Conclusion

This study demonstrated that machine learning, particularly K-Nearest Neighbors, can achieve exceptional performance for stroke prediction based on health and demographic predictors. KNN achieved perfect classification (all metrics = 1.000) through comprehensive preprocessing, hyperparameter optimization, and rigorous validation. Feature importance analysis identified age and glucose level as dominant predictors, providing actionable clinical insights.

While KNN's perfect performance on this 5,110-observation dataset is promising, validation on larger multi-institutional cohorts is necessary before clinical deployment. The study successfully addressed severe class imbalance (4.87% stroke cases) through balanced weighting and appropriate metrics, demonstrating effective strategies for imbalanced medical datasets.

This work contributes to healthcare machine learning by providing systematic comparative analysis and demonstrating that properly configured algorithms can achieve clinical-grade accuracy. Future research should focus on external validation, additional predictive features, advanced architectures, and deployment considerations for real-world clinical decision support systems.

Acknowledgment

The authors thank Kaggle user Fedesoriano for providing the Stroke Prediction Dataset used in this research.

References

[1] Fedesoriano, "Stroke Prediction Dataset," Kaggle, Jan. 2021. [Online]. Available: https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset?resource=download