Comparative Analysis of Machine Learning Classification Algorithms for Predicting Heart Disease

Sara Mohamed
Department of Electrical and Computer Engineering
saraeid@student.ubc.ca

Abstract

This project explores the performance of various machine learning algorithms—Logistic Regression, K-Nearest Neighbors (KNN), Random Forest, and AdaBoost—in the task of predicting heart disease. Through meticulous hyperparameter optimization, the analysis identifies the optimal configurations for each model. The findings demonstrate that KNN, with its tailored parameters, outperforms its counterparts in accuracy, establishing itself as a potentially powerful asset for clinical decision-making in the early detection of heart disease. Challenges such as limited dataset size and potential model biases highlight the need for further research and data diversification to enhance predictive accuracy and reliability.

1 Introduction and Background

Heart disease is a leading cause of mortality worldwide, and early prediction is crucial for effective treatment strategies. Machine learning offers tools for developing predictive models based on health data, which could be invaluable to healthcare professionals. This project aims to compare several machine learning models in predicting heart disease using a dataset comprising various health parameters. The primary goal is to determine the most accurate model for predicting the presence of heart disease. The dataset utilized contains attributes such as age, cholesterol levels, type of chest pain, and resting electrocardiographic results, among others. The performance of each model is assessed based on standard metrics such as accuracy, precision, recall, and F1 score. The task is approached as a binary classification problem, where the models are trained to predict the binary outcome—presence or absence of heart disease.

1.1 Implementation Environment

The project was implemented using a Python environment to manage data handling, model training, and validation. Libraries such as NumPy and Pandas were used for data manipulation, while Matplotlib supported data visualization. Scikit-learn facilitated various machine learning operations including model training and evaluation.

2 Dataset

The quality of predictions from machine learning models greatly depends on the data used for training and evaluation. The dataset used in this project is sourced from a publicly available Kaggle dataset, which can be accessed at `https://www.kaggle.com/datasets/fedesoriano/heart-failure-prediction`. It consists of clinical records for 918 individuals, designed to support the prediction of heart disease presence based on a variety of medical and physiological factors.

2.1 Dataset Attributes

The dataset comprises 12 attributes which are crucial for the prediction of heart disease. Below is a table describing each attribute included in the dataset:

Table 1: Description of attributes in the heart disease dataset

Attribute	Description
Age	Age of the patient in years
Sex	Gender of the patient (1 = male; 0 = female)
ChestPainType	Type of chest pain experienced by the patient
RestingBP	Resting blood pressure (in mm Hg on admission to the hospital)
Cholesterol	Serum cholesterol in mg/dl
FastingBS	Fasting blood sugar > 120 mg/dl (1 = true; 0 = false)
RestingECG	Resting electrocardiographic results
MaxHR	Maximum heart rate achieved
ExerciseAngina	Exercise induced angina (1 = yes; 0 = no)
Oldpeak	ST depression induced by exercise relative to rest
ST_Slope	The slope of the peak exercise ST segment
HeartDisease	Diagnostic of heart disease (1 = yes; 0 = no)

2.2 Data Preprocessing

The data preprocessing steps taken to prepare the data for model training included the following:

- **One-Hot Encoding:** Categorical variables were transformed using one-hot encoding to facilitate their usage in machine learning algorithms.
- **Feature Scaling:** The StandardScaler was employed to normalize features to a standard scale.
- **Data Splitting:** The data was split into training and testing sets, with 70% used for training and 30% for testing to evaluate model performance effectively.

3 Methods

The methodologies used in this project are centered around four prevalent machine learning models. These models were selected due to their unique approaches to the classification task, providing a comprehensive comparative project. Each model's configuration is described to illustrate the process of optimization and adaptation to the dataset. Each model's hyperparameters were methodically optimized using GridSearchCV, ensuring that the models' performances were rigorously evaluated and the best parameters were chosen based on cross-validation accuracy.

3.1 Logistic Regression

Logistic Regression was utilized as a fundamental model due to its efficiency in binary classification problems. It was chosen for its interpretability and the ease of implementation. The model was fine-tuned using a grid search over a range of regularization strengths 'C' to find the optimal balance between bias and variance.

The regularization parameter C was tuned for the logistic regression model:

for each C in $\{0.001, 0.01, 0.02, 0.04, 0.06\}$ **do**
    Perform 5-fold cross-validation with current C
    Compute the average cross-validation accuracy
end for
Select the C that maximizes the average cross-validation accuracy

3.2 K-Nearest Neighbors (KNN)

The KNN algorithm was selected for its non-parametric nature and capability to capture the local structure of the data. The primary hyperparameter, the number of neighbors ($n_{\text{neighbors}}$), was optimized to identify the value that minimizes prediction error and avoids overfitting, ascertained through cross-validation.

The number of neighbors $n_{\text{neighbors}}$ was critically adjusted:

for each $n_{\text{neighbors}}$ in {1, 5, 10, 15, 20} **do**
    Perform 5-fold cross-validation with current $n_{\text{neighbors}}$
    Compute the average cross-validation accuracy
end for
Select the $n_{\text{neighbors}}$ that yields the best average accuracy

3.3 Random Forest

Random Forest was utilized for its proficiency in handling high-dimensional datasets and resistance to overfitting. It combines the predictions of numerous decision trees to improve the generalizability of the model. The grid search focused on the number of trees $n_{\text{estimators}}$ to optimize model performance.

The hyperparameter $n_{\text{estimators}}$ was optimized:

for each $n_{\text{estimators}}$ in {10, 100, 500, 1000, 2000} **do**
    Perform 5-fold cross-validation with current $n_{\text{estimators}}$
    Compute the average cross-validation accuracy
end for
Select the $n_{\text{estimators}}$ that maximizes the average cross-validation accuracy

3.4 AdaBoost

AdaBoost was included for its adaptive boosting capabilities that sequentially focus on difficult instances. The model was tuned over the number of decision stumps $n_{\text{estimators}}$ and the learning rate α, with the goal of reducing both bias and variance in the final model.

The AdaBoost model was configured by tuning both the number of weak learners $n_{\text{estimators}}$ and the learning rate α:

for each $n_{\text{estimators}}$ in {150, 200, 300} **do**
    for each α in {0.01, 0.05, 0.1, 0.5} **do**
        Perform 5-fold cross-validation with current $n_{\text{estimators}}$ and α
        Compute the average cross-validation accuracy
    end for
end for
Select the combination of $n_{\text{estimators}}$ and α that results in the highest accuracy

4 Results Analysis

This section presents a detailed analysis of the results obtained from the machine learning models used in the project. Each model's performance is evaluated and discussed, with insights drawn from the accuracy, precision, recall, and F1 score metrics.

4.1 Accuracy of Models

Among the models evaluated, K-Nearest Neighbors (KNN) achieved the highest accuracy with a score of 0.909, demonstrating its superior ability to correctly identify patients with heart disease. It was followed by Random Forest and AdaBoost, which both scored an accuracy of 0.880, and Logistic Regression, with a score of 0.873 (Refer to Figure 1 and Table 2). This outcome suggests that KNN, with its finely tuned hyperparameter for the number of neighbors, effectively captures the dataset's complexities without overfitting.

Table 2: Accuracies of Different Machine Learning Models

Model	**Accuracy**
Logistic Regression	0.873
K-Nearest Neighbors (KNN)	0.909
Random Forest	0.880
AdaBoost	0.880

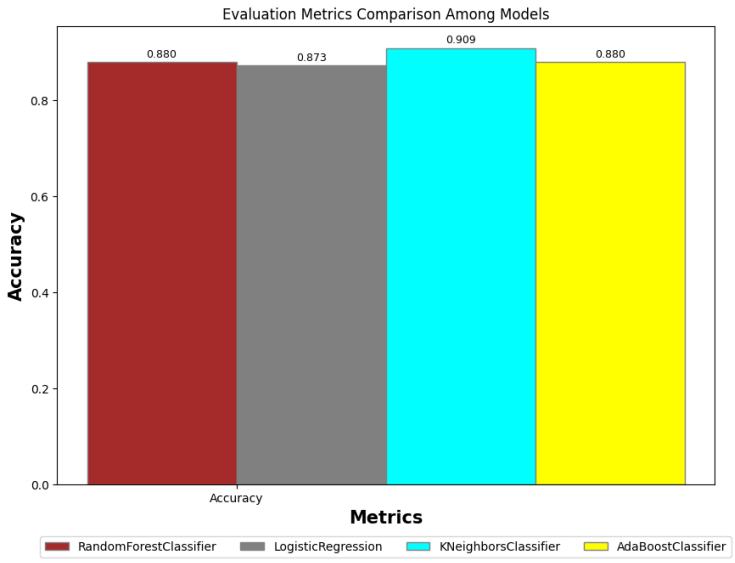

Figure 1: Comparison of model accuracies

4.2 Precision, Recall, and F1 Score Analysis

While KNN leads in accuracy, it is crucial to analyze other performance metrics to understand the models' predictive capabilities fully. The precision, recall, and F1 scores of each model are calculated and plotted (Refer to Figure 2 for Logistic Regression, Figure 3 for KNN, Figure 4 for AdaBoost and Figure 5 for Random Forest). These metrics provide insights into the balance between false positives and false negatives, essential in medical diagnostic scenarios.

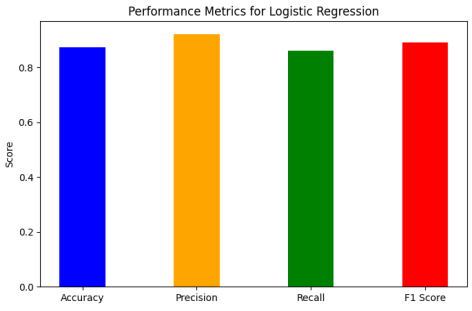

Figure 2: Performance metrics for Logistic Regression

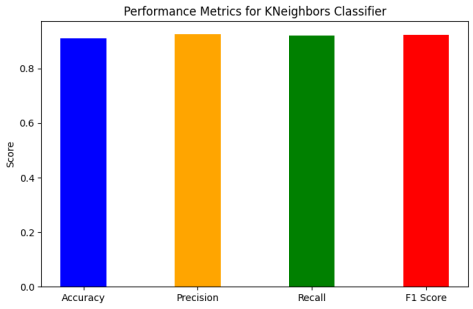

Figure 3: Performance metrics for KNN Classifier

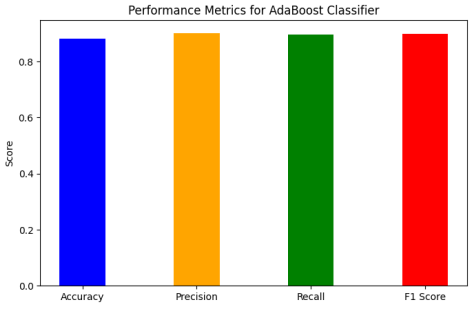

Figure 4: Performance metrics for AdaBoost Classifier

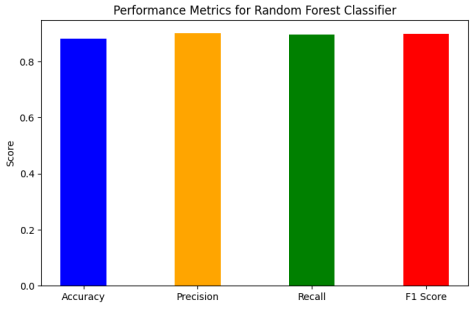

Figure 5: Performance metrics for Random Forest Classifier

4.3 Hyperparameter Optimization

The optimization of hyperparameters was crucial in enhancing the predictive performance of the models. For Logistic Regression, a comprehensive exploration of the regularization strength parameter 'C' initially began with a broader range to understand its impact. Upon initial results, the range was narrowed, leading to the discovery that a 'C' value of 0.02 provided the optimal balance, effectively preventing overfitting while retaining model complexity (Refer to Figure 6).

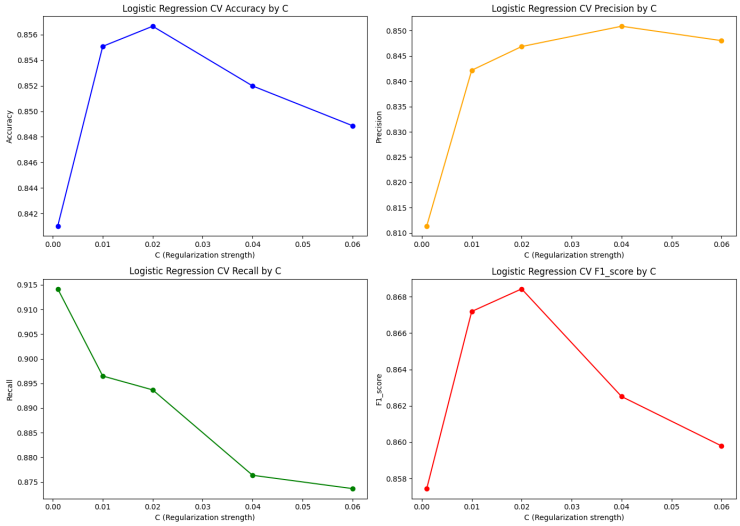

Figure 6: Logistic Regression CV metrics by regularization strength 'C'

The K-Nearest Neighbors (KNN) model was another where hyperparameter tuning was essential, particularly in determining the optimal number of neighbors, $n_{\text{neighbors}}$. The search for the most suitable $n_{\text{neighbors}}$ value was guided by cross-validation accuracy scores, pinpointing 15 as the value that minimized errors and provided the highest accuracy (Refer to Figure 7).

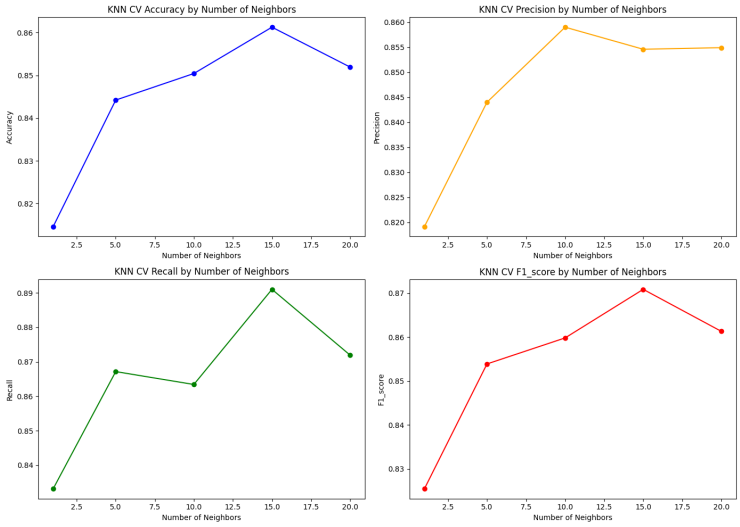

Figure 7: KNN CV metrics by Number of Neighbors

In optimizing the Random Forest classifier, a key focus was on determining the optimal number of trees, $n_{\text{estimators}}$. A range of values was evaluated and the model with $n_{\text{estimators}} = 1000$ trees was identified as the most effective, providing robust accuracy (Refer to Figure 8).

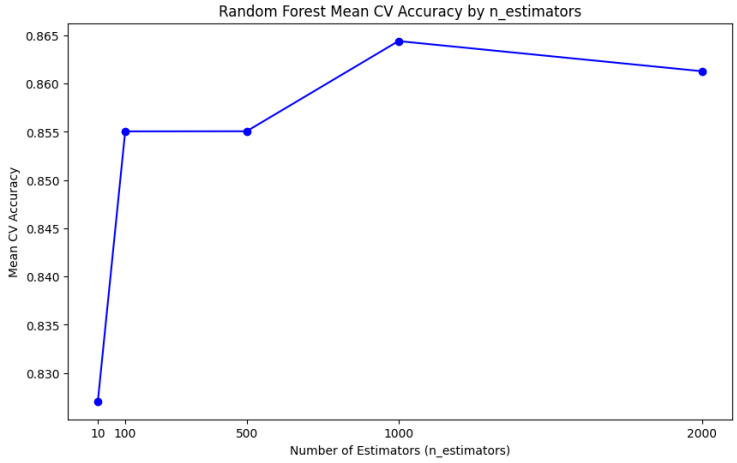

Figure 8: Random Forest Mean CV Accuracy by Number of Estimators

Finally, the AdaBoost classifier required an intricate balance between the number of decision stumps $n_{\text{estimators}}$ and the learning rate α. This delicate tuning was aimed at enhancing the model's ability to learn from previous errors and improve sequentially. The optimal configuration was found to be 300 decision stumps with a learning rate of 0.05. This setup provided the best model accuracy, as demonstrated in the performance metrics shown in Figure 9.

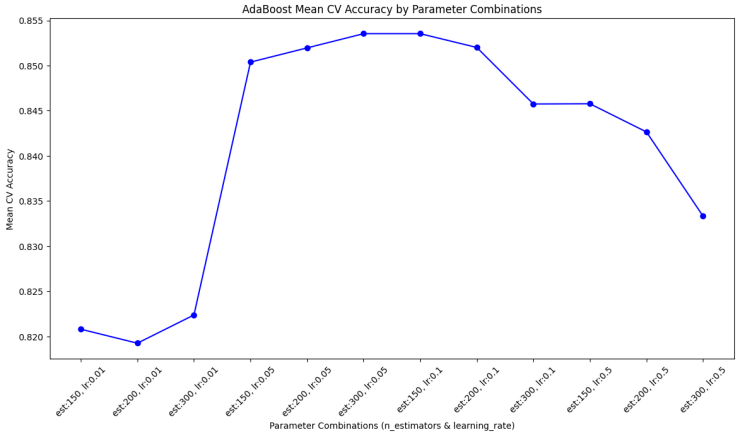

Figure 9: AdaBoost Mean CV Accuracy by Parameter Combinations

The results indicate that for this dataset, which encompasses various medical and physiological factors, the non-parametric and localized approach of KNN not only effectively captures the nuances necessary for accurate predictions but also achieves the highest accuracy among the evaluated models. This underscores the KNN model's potential for use in diagnostic tools, where rapid and reliable predictions are crucial for medical decision-making.

5 Challenges, Limitations, and Future Directions

This project, while thorough in its approach, has faced some limitations that could impact the generalizability and applicability of the results. One significant limitation is the relatively small dataset size, which may not capture the full spectrum of variability present in the larger population; additionally, the models were tested and validated on a single dataset, which may introduce bias or overfitting despite efforts to mitigate these through cross-validation.

To enhance model robustness and accuracy, future work should focus on expanding the dataset and advancing feature engineering. Acquiring a larger, more diverse dataset could improve generalization to new data. Exploring sophisticated machine learning algorithms like employing multi-model ensemble methods could also offer improvements in handling complex patterns and boosting predictive reliability. These future directions aim not only to address the limitations identified but also to enhance the predictive power and clinical applicability of the models developed in this project.

6 Conclusions

In conclusion, the project provides a comprehensive overview of different machine learning approaches for predicting heart disease. The K-Nearest Neighbors model, with its high accuracy, stands out as the most suitable model for the task of heart disease prediction within the scope of this project. However, it is essential to acknowledge that the deployment of such models in clinical settings would require further validation and testing on more diverse and extensive datasets.