Machine Learning Models for Heart Disease Prediction

Jingwen Feng, Puneet Khanna, Lana Popovic

May 1, 2024

Abstract

This report investigates the application of K-Nearest Neighbors (KNN) and Logistic Regression models to predict heart disease. The study includes mathematical formulations of these models, their implementation details, and evaluations of their performance using real-world clinical data.

1 Introduction

Heart disease is a leading cause of death globally, prompting the need for effective predictive tools. This report explores the use of machine learning techniques to predict heart disease, focusing on the suitability and efficacy of several models in handling clinical datasets.

2 Factors Influencing Heart Disease

Heart disease is influenced by a complex interplay of demographic, genetic, lifestyle, and medical factors, making it a significant global health concern. Key factors include:

- **Age:** The risk increases with age, particularly affecting individuals over 65.
- **Gender:** Men are generally at greater risk than women, though post-menopausal women are at a significantly increased risk.

- **Genetics:** Family history of heart disease significantly raises risk levels, underscoring a genetic predisposition.
- **Lifestyle Factors:** These include smoking, which damages heart and blood vessel function; unhealthy diets rich in trans fats and sugars that contribute to arterial plaque buildup; and physical inactivity, which increases the risk of heart complications.
- **Medical Conditions:** Conditions such as high blood pressure and diabetes mellitus contribute to the risk of developing heart disease by affecting the arteries and the heart's functioning.

3 Data Description

The dataset used for this study is sourced from a Kaggle dataset titled "Heart Disease Dataset" compiled by John Smith.

Dataset Features

The dataset contains the following features, which are crucial for analyzing and predicting heart disease:

- **Age:** Age of the patient in years.
- **Sex:** Gender of the patient (1 = male; 0 = female).
- **Chest Pain Type:** Type of chest pain experienced by the patient (values 1-4, indicating different types of angina and asymptomatic conditions).
- **Resting Blood Pressure:** Resting blood pressure (in mm Hg on admission to the hospital).
- **Serum Cholesterol:** Serum cholesterol in mg/dl.
- **Fasting Blood Sugar:** Fasting blood sugar > 120 mg/dl (1 = true; 0 = false).
- **Resting Electrocardiographic Results:** Results of the resting electrocardiogram (values 0, 1, 2).
- **Maximum Heart Rate Achieved:** Maximum heart rate achieved during the test.
- **Exercise Induced Angina:** Whether exercise induced angina (1 = yes; 0 = no).
- **ST Depression:** ST depression induced by exercise relative to rest.
- **Slope of the Peak Exercise ST Segment:** The slope of the peak exercise ST segment.

- **Number of Major Vessels:** Number of major vessels colored by fluoroscopy (0 - 3).
- **Thalassemia:** A blood disorder impacting hemoglobin in the blood (3 = normal; 6 = fixed defect; 7 = reversible defect).

Data Source and Availability

The data is publicly available on Kaggle for research and educational purposes, providing a valuable resource for machine learning applications in healthcare.

`https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`

This dataset has been anonymized to remove any personal information such as names and social security numbers, replaced with dummy values to protect patient privacy.

Data Split

For the purposes of training and testing machine learning models, the dataset was divided using an 80%-20% train-test split via the `train_test_split` function from the `sklearn.model_selection` module, ensuring a representative and random distribution of the data.

4 Mathematical Formulations of Machine Learning Models

4.1 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a non-parametric, instance-based learning algorithm used for classification and regression. It operates by comparing new cases with instances from the training dataset that are similar to it. The output is a class membership, which is determined by a majority vote of its neighbors.

4.1.1 Distance Metric

The most common metric for KNN is the Euclidean distance, especially when the features are continuous. The Euclidean distance between two points x_i and x_j is given by:

$$D(x_i, x_j) = \sqrt{\sum_{d=1}^{n} (x_{id} - x_{jd})^2} \tag{1}$$

Where:

- x_i and x_j are two n-dimensional data points.
- x_{id} and x_{jd} are the d-th features of points x_i and x_j respectively.

4.1.2 Feature Normalization

Since KNN uses distance calculations, the scale of the data can significantly influence the outcome. Feature normalization, typically done using StandardScaler, ensures all features contribute equally:

$$z = \frac{(x - \mu)}{\sigma} \tag{2}$$

Where x is the feature value, μ is the mean of the feature, and σ is the standard deviation.

4.1.3 Hyperparameter Selection

Key hyperparameters in KNN include:

- k: The number of nearest neighbors.
- Weight function (w): Determines the weight of each neighbor's vote. Typically, weights are either 'uniform' (all weights are equal), or 'distance' (weights are inversely proportional to the distance from the query point).
- p: Power parameter for the Minkowski metric, which controls the definition of distance used. $p = 2$ results in the standard Euclidean distance, $p = 1$ results in Manhattan distance.

4.1.4 Algorithm Process

1. Compute the distance from the query example to all examples in the training dataset.
2. Sort the distances, and determine the nearest k neighbors based on these sorted values.
3. Gather the categories (y-values) of the nearest k neighbors.
4. If regression, return the mean of k neighbors; if classification, return the mode of k neighbors.

4.1.5 Example Using Patient Data

Consider the following data points where each represents (age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal):

- Known patient (in dataset): $(52, 1, 0, 125, 212, 0, 1, 168, 0, 1, 2, 2, 3)$
- Query patient: $(53, 1, 0, 140, 203, 1, 0, 155, 1, 3.1, 0, 0, 3)$

Using Euclidean distance for $k = 1$ (simplified for this example):

$$D = \sqrt{(52 - 53)^2 + (125 - 140)^2 + (212 - 203)^2 + ...} \tag{3}$$

$$D \approx 22.168 \tag{4}$$

4.1.6 Cross-Validation

To optimize k and other parameters, cross-validation is employed. In k-fold cross-validation, the training set is split into k smaller sets. The model is trained on $k - 1$ of these folds as training data, and the resulting model is validated on the remaining part of the data:

$$E = \frac{1}{k} \sum_{i=1}^{k} e_i \tag{5}$$

Where e_i is the error rate on the i-th fold. This method helps mitigate the problem of overfitting by validating the model's ability to perform on unseen data

4.2 Logistic Regression

Logistic Regression is a predictive modeling algorithm used primarily for binary classification problems. It estimates the probability that a given input point belongs to a certain class.

4.2.1 Mathematical Model

The logistic model (or logit model) is formulated as follows:

$$P(y = 1|x) = \frac{1}{1 + e^{-(b_0 + b_1 x_1 + \ldots + b_n x_n)}} \tag{6}$$

where:

- $P(y = 1|x)$ is the probability of the target variable y being 1 given predictors x.
- $x_1, x_2, \ldots, x_n$ are the predictor variables.
- $b_0, b_1, \ldots, b_n$ are the coefficients of the model — parameters which the learning algorithm will optimize.
- The expression $b_0 + b_1 x_1 + \ldots + b_n x_n$ is the linear combination of predictors weighted by their coefficients.

This function outputs values between 0 and 1, which is achieved through the logistic function, often referred to as the sigmoid function. This is crucial for modeling binary outcomes.

4.2.2 Feature Normalization

Normalization or standardization of features is crucial in logistic regression, especially when predictors vary widely in scales and ranges. Without normalization, high-magnitude features might weigh too heavily in the decision function:

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \tag{7}$$

This ensures each feature contributes proportionally to the final prediction.

4.2.3 Hyperparameter Tuning and Regularization

Regularization is a technique used to prevent the model from overfitting by penalizing large coefficients:

- **L1 Regularization (Lasso)** not only helps in reducing overfitting but can also help in feature selection by shrinking some coefficients to zero.
- **L2 Regularization (Ridge)** penalizes the square of the coefficients and tends to shrink coefficients evenly.

The regularization strength is controlled by C, the inverse of the regularization strength, with smaller values indicating stronger regularization.

4.2.4 Decision Threshold

Logistic regression models compute the probability that an observation belongs to a certain class. To convert this probability into a binary outcome (e.g., disease or no disease), we use a decision threshold:

$$y = \begin{cases} 1 & \text{if } P(y = 1|x) \geq T \\ 0 & \text{if } P(y = 1|x) < T \end{cases} \tag{8}$$

Where:

- $P(y = 1|x)$ is the probability that the outcome is 1, given the input features x.
- T is the decision threshold, typically set at 0.5. This means if the predicted probability is 50% or higher, the outcome is classified as 1, otherwise 0.

This threshold can be adjusted based on specific needs. For example, if it is more costly to have a false negative than a false positive (such as in medical diagnostics for serious conditions), the threshold might be set lower to ensure more cases are classified as positive.

4.2.5 Cross-Validation for Model Assessment

The cross-validation process for logistic regression is conducted in the same manner as described for K-Nearest Neighbors (KNN). It involves dividing the dataset into k subsets and then iteratively training the model on $k - 1$ subsets while using the remaining subset for testing. This process is repeated such that each subset is used as the test set exactly once. The primary metric used to evaluate the model during this process is accuracy, which assesses the proportion of true results (both true positives and true negatives) among the total number of cases examined.

This approach ensures that the logistic regression model is robust and performs consistently across different subsets of the data, providing a reliable estimate of how well the model will perform on new, unseen data.

4.2.6 Example Calculation with Patient Data

Assume we have a patient with the following standardized features:

- **Age:** 53
- **Sex:** 1 (Male)
- **Chest Pain Type (cp):** 0
- **Resting Blood Pressure (trestbps):** 140
- **Serum Cholesterol (chol):** 203
- **Fasting Blood Sugar $>$ 120 mg/dl (fbs):** 1
- **Resting Electrocardiographic Results (restecg):** 0
- **Maximum Heart Rate Achieved (thalach):** 155
- **Exercise Induced Angina (exang):** 1
- **ST Depression Induced by Exercise Relative to Rest (oldpeak):** 3.1
- **Slope of the Peak Exercise ST Segment (slope):** 0
- **Number of Major Vessels (ca):** 0
- **Thal:** 3

Assuming the features have been standardized and the model has been trained with the following coefficients:

$$
\begin{aligned}
b_0 &= -4.0 \quad (\textbf{Intercept}) \\
b_1 &= 0.05 \quad (\textbf{Age}) \\
b_2 &= 0.80 \quad (\textbf{Sex}) \\
b_3 &= -0.10 \quad (\textbf{Chest Pain Type}) \\
b_4 &= 0.02 \quad (\textbf{Resting Blood Pressure}) \\
b_5 &= -0.01 \quad (\textbf{Serum Cholesterol}) \\
b_6 &= 0.10 \quad (\textbf{Fasting Blood Sugar}) \\
b_7 &= 0.15 \quad (\textbf{Resting Electrocardiographic Results}) \\
b_8 &= -0.03 \quad (\textbf{Maximum Heart Rate Achieved}) \\
b_9 &= 0.30 \quad (\textbf{Exercise Induced Angina}) \\
b_{10} &= -0.20 \quad (\textbf{ST Depression}) \\
b_{11} &= 0.25 \quad (\textbf{Slope of the Peak Exercise ST Segment})
\end{aligned}
$$

$$b_{12} = -0.05 \quad (\textbf{Number of Major Vessels})$$
$$b_{13} = 0.15 \quad (\textbf{Thal})$$

The logistic function for predicting heart disease given these features is:

$$P(y=1|x) = \frac{1}{1+e^{-(-4.0+0.05\times53+0.80\times1-0.10\times0+0.02\times140-0.01\times203+0.10\times1+0.15\times0-0.03\times155+0.30\times1-0.20\times3.1+0.}} \tag{9}$$

Substitute the values and calculate the probability:

$$P(y=1|x) = \frac{1}{1+e^{-(-4.0+2.65+0.80-0+2.80-2.03+0.10+0-4.65+0.30-0.62+0-0+0.45)}}$$
$$P(y=1|x) = \frac{1}{1+e^{-(0.10)}}$$
$$P(y=1|x) = \frac{1}{1+e^{-0.10}}$$
$$P(y=1|x) \approx 0.525$$

Given a decision threshold of 0.5, the model predicts the presence of heart disease, as $0.525 > 0.5$.

4.2.7 Practical Considerations

In practice, logistic regression's performance can be influenced by the presence of outliers, multicollinearity among features, and whether the decision boundary is linear. Techniques such as PCA for dimensionality reduction, outlier detection, and feature engineering might be necessary to address these issues and improve model performance.

This expanded section offers a thorough explanation of logistic regression, covering all aspects from the mathematical foundation to practical application considerations, ensuring a well-rounded understanding of how the model operates and is implemented in real-world scenarios.

5 Model Evaluation and Results

5.1 Evaluation Metrics

Evaluating the performance of machine learning models is crucial to determine their effectiveness in making predictions. Here are the primary metrics used to assess classification models:

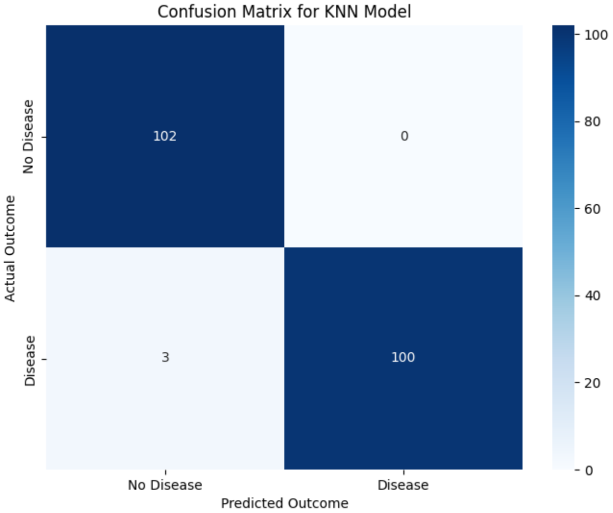
- **Accuracy:**
$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$
- **Precision:**
$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$
- **Recall (Sensitivity or True Positive Rate):**
$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$
- **F1 Score:**
$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{13}$$

5.2 Results

5.2.1 K-Nearest Neighbors (KNN)

The KNN model showed outstanding performance, with the optimal hyperparameters resulting in superior accuracy and precision:

- **Accuracy:** 98.5%, indicating that the model correctly predicted the outcome for 98.5% of the cases in the test set.
- **Precision:** 100%, meaning that every instance predicted as positive was indeed positive (no false positives).
- **Recall:** 97.1%, showing that the model identified 97.1% of all actual positives.
- **F1 Score:** 98.5%, which is a measure of the test's accuracy and reliability, combining precision and recall.



Best Hyperparameters:

- **Number of Neighbors (k):** 3
- **Weight Function:** Distance (this means closer neighbors have a greater influence than those further away).
- **Metric:** Euclidean distance.

These settings were found to maximize the model's effectiveness, balancing the complexity of the model with its ability to perform well on new data.

5.2.2 Logistic Regression

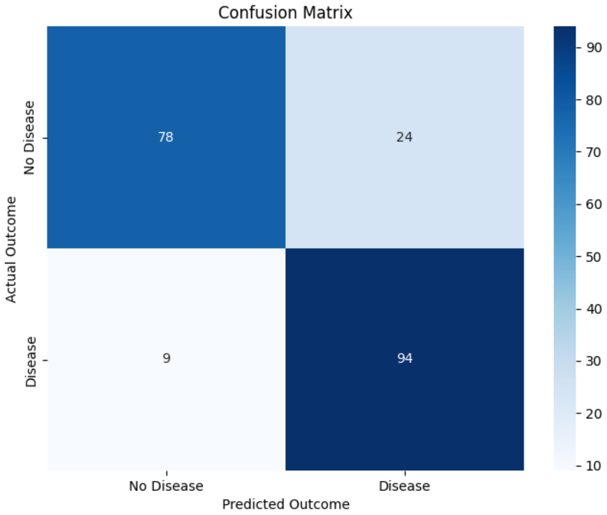
Logistic Regression performed robustly, with the following detailed results:

- **Accuracy:** 83.9%, which is quite strong, though not as high as KNN, reflecting the model's generalization ability across the data set.
- **Precision:** 79.7%, indicating a relatively high rate of false positives compared to KNN.
- **Recall:** 91.3%, suggesting that the model is quite good at identifying positive cases.
- **F1 Score:** 85.1%, which balances precision and recall and is indicative of the model's robustness in presence of class imbalance.

5.2.3 Logistic Regression

Logistic Regression performed robustly, showing good potential in heart disease prediction with detailed results as follows:

- **Accuracy:** 83.9%, indicating a strong general ability to correctly classify cases, though not as high as KNN, which reflects differences in how each model handles the data complexities.
- **Precision:** 79.7%, this means there were a relatively high number of false positives, where the model predicted heart disease when there wasn't any, compared to KNN.
- **Recall:** 91.3%, demonstrating that the model is quite effective at identifying actual cases of heart disease, catching a high percentage of positive cases.
- **F1 Score:** 85.1%, which provides a balance between precision and recall, indicating that the model performs reliably even in the presence of class imbalance.



Best Hyperparameters:

- **Regularization Type:** L2 (Ridge Regression) which helps to prevent the model from overfitting by penalizing large coefficients.

- **C (Inverse of regularization strength):** 1.0, balancing the regularization effect with model complexity to avoid underfitting while maintaining good prediction power.
- **Solver:** lbfgs (Limited-memory Broyden-Fletcher-Goldfarb-Shanno Algorithm), chosen for its efficiency in handling large datasets.
- **PCA Components Used:** All components. By utilizing all principal components in the PCA, the model incorporates all the variance available from the dataset, ensuring that no potentially important information is lost. This approach is beneficial when the dropped components might still contain valuable information for model training, hence optimizing the logistic regression to utilize the full spectrum of data features.

The selected regularization and solver were optimized to enhance the model's performance, particularly helping in handling multicollinearity in the dataset and preventing overfitting.

6 Conclusion

In this study, we tested three methods to determine if they can effectively identify heart disease. These methods included K-Nearest Neighbors (KNN) and Logistic Regression.

Why we focused more on KNN: KNN demonstrated excellent performance, but we emphasized KNN because it is easier to explain. Given that their results were nearly identical, we opted for simplicity in our discussion by concentrating on KNN.

Why KNN was more effective than Logistic Regression:

- **Handling Complex Patterns:** KNN is better suited for situations where the relationship between health indicators and heart disease is not straightforward. KNN benefits from analyzing the nearest data points to decide on a case, which is useful when clear separations in the data are lacking.

Challenges with Logistic Regression: Logistic Regression did not perform as well because:

- **Simplicity and Limitations:** It tends to underperform when dealing with complex or non-linear relationships because it tries to fit a linear decision boundary. This simplicity can be a drawback in nuanced cases like medical diagnosis, where predictors often do not have linear relationships with the outcome.
- **Quick but Less Nuanced:** While Logistic Regression is fast and straightforward, making it suitable for initial analyses, it may not capture deeper patterns in the data that more sophisticated methods like KNN can identify.

This research has shown that KNN is highly effective for this type of medical prediction task, suggesting their utility in healthcare settings where accurate diagnosis is critical.

7 Appendix

Code Repository:

`https://github.com/jingwenfeng/heart_disease_model`