Machine Learning Models for Heart Disease Prediction

Jingwen Feng, Puneet Khanna, Lana Popovic

February 21, 2026

Overview

Factors Influencing Heart Disease

- **Age:** The risk increases as people get older, especially for people over 65 and those of color.
  - Heart Disease is responsible for 1 in 3 deaths.
  - Nearly 20% of those deaths are people under 65.
- **Gender:** Men are generally at greater risk of heart disease.
- **Genetics:** Family history can increase risk through factors such as high blood pressure or even common environments.
- **Lifestyle Factors:**
  - Smoking: Makes blood more likely to clot, damages cells that line blood vessels, increases buildup of plaque.
  - Diet: Too much sodium intake from outside sources.
  - Physical inactivity: Lack of aerobic and muscle-strengthening activity.
- **Medical Conditions:**
  - High blood pressure: Plaque builds up within artery linings, narrowing blood flow to heart and brain.
  - Type 2 Diabetes: High blood sugar damages blood vessels in heart.

Data Breakdown

- Trained models upon 820 individuals.
- Tested models using 205 individuals.

1	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
2	49	1	2	118	149	0	0	126	0	0.8	2	3	2

Figure: Sample X_{train}/X_{test} data

Age: Age of individual (ages 29 to 77). **Sex:** Sex of individual (0 = female, 1 = male). **Cp:** Chest pain ranked from 0 to 3. **Trest:** Resting blood pressure. **Chol:** Serum cholesterol in mg/dl. **Fbs:** Fasting blood sugar $>$ 120 mg/dl (0 = false, 1 = true). **Restecg:** Resting ECG results from 0 to 2. **Thalach:** Maximum heart rate achieved. **Exang:** Exercise-induced angina (chest pain caused by reduced blood flow to heart) (0 = no, 1 = yes). **Oldpeak:** Presence of ST depression on ECG reading post-exercise. **Slope:** Slope of peak exercise ST depression segment. **Ca:** Number of blood vessels colored during fluoroscopy from 0 to 3. **Thal:** Presence of Thalassemia (inherited blood disorder causing less hemoglobin and RBC) from 0 to 3.

- Sample Y_{train}/Y_{test} data: 0 for No Heart Disease and 1 for Heart Disease.

Why Choose Logistic Regression, KNN, and SVM?

- **K-Nearest Neighbors (KNN):**
  - Non-parametric and lazy learning algorithm.
  - Effective if the decision boundary is irregular.
  - Simple to understand and implement.
  - Curse of dimensionality.
- **Support Vector Machine (SVM):**
  - Effective in high-dimensional spaces.
  - Works well with clear margin of separation.
  - Robust against overfitting in high-dimensional spaces.
  - Sensitive to outliers.
  - Data needs to be linearly separable.
- **Logistic Regression:**
  - Good for binary classification.
  - Provides probabilities for outcomes.
  - Easy to implement and interpret.
  - Assumes linearity.
  - Requires no multicollinearity between independent variables.

Heart Disease Dataset

The dataset used to train the model is from 1988 and consists of four databases: Cleveland, Hungary, Switzerland, and Long Beach V.
`https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset`

KNN for Heart Disease Prediction

The KNN algorithm assigns a class to a sample based on the majority class among its k nearest neighbors. It calculates the Euclidean distance in an n-dimensional space as follows:

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

where $D(x_i, x_j)$ is the Euclidean distance between points x_i and x_j, and n is the number of features.

Pipeline Construction

The KNN pipeline consists of two main components:

1. **StandardScaler:** Normalizes features by removing the mean and scaling to unit variance.

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

2. **KNeighborsClassifier:** Implements the KNN algorithm where the class of a sample is determined based on the majority vote of its k nearest neighbors.

$$D(x, x_i) = \sqrt{\sum_{d=1}^{n} (z_{xd} - z_{id})^2}$$

$$C(x) = \text{argmax}_c \sum_{i=1}^{k} \mathbf{1}(y_i = c \wedge d(x, x_i) \leq D_k(x))$$

Hyperparameter Tuning in KNN

Optimize the following parameters to minimize validation error:

- k: Number of neighbors
- w: Weight function (uniform or distance)
- p: Power parameter for the Minkowski metric

Objective function:

$$(\hat{k}, \hat{w}, \hat{p}) = \arg\min_{k,w,p} \left(\frac{1}{v} \sum_{i=1}^{v} \frac{1}{|\text{test}_i|} \sum_{j \in \text{test}_i} \mathbf{1}(y_j \neq \text{Vote}(x_j)) \right)$$

where

$$\text{Vote}(x_j) = \text{argmax}_c \left(\sum_{\ell \in N_k(x_j)} w(x_j, x_\ell) \cdot \mathbf{1}(y_\ell = c) \right)$$

$$w(x_j, x_\ell) = \begin{cases} 1 & \text{if uniform weights} \\ \frac{1}{d(x_j, x_\ell)^p} & \text{if distance-based weights} \end{cases}$$

KNN Cross-Validation

The execution of cross-validation for KNN involves careful training and evaluation:

1. **Train KNN:** Train the KNN model on the combined training set folds. Consider varying k to find the optimal number of neighbors.
2. **Evaluate Model:** Assess the KNN model on the test fold using a metric like accuracy.
3. **Compute the Score:** Calculate the accuracy for the test set as:

$$\text{score} = \frac{|\{j : y_j = \hat{y}_j\}|}{|\text{test}_i|}$$

where y_j are the true labels, $\hat{y}_j$ are the predicted labels by KNN, and $|\text{test}_i|$ is the number of samples in the test fold.

Finally, average the scores from all folds to calculate the overall cross-validation score:

$$\text{CV Score} = \frac{1}{v} \sum_{i=1}^{v} \text{score}$$

KNN Classification Example

Feature Scaling and Distance Calculation

For two patients A and B:

$$z_{\text{age,A}} = -0.75, \qquad z_{\text{trestbps,A}} = -0.67$$
$$z_{\text{age,B}} = -0.5, \qquad z_{\text{trestbps,B}} = 0.33$$

Euclidean distance between Patient A and B:

$$D(A, B) = \sqrt{(-0.75 + 0.5)^2 + (-0.67 - 0.33)^2} \approx 1.031$$

Classification with KNN

Assuming $k = 1$, if Patient B is the closest neighbor with $\textit{target} = 0$, then Patient A would also be predicted to not have heart disease.

SVM for Heart Disease Prediction (1/2)

SVM Optimization:

$$\min_{w,b} \frac{1}{2}|w|^2 + b \text{ s.t.} \qquad y_i(w^T x_i + b) \geq 0, \forall i$$

where:

- b: allowing for misclassifications
- y_i: class label (-1 or $+1$) of the i-th training example

SVM for Heart Disease Prediction (2/2)

Kernel Functions for Non-linear Decision Boundaries:

- Linear: $K(x_i, x_j) = x_i^T x_j$
- Polynomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$
- Radial Basis Function (RBF): $K(x_i, x_j) = \exp(-\gamma |x_i - x_j|^2)$

Model Prediction:

$$f(x) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b\right)$$

- $f(x)$: predicted class label for a new input feature vector x
- α_i: Lagrange multipliers obtained by solving the dual optimization problem
- x_i, y_i: feature vectors and class labels of the training examples
- $K(x_i, x)$: kernel function computing the similarity between the new input x and the i-th training example x_i
- b: bias term of the hyperplane

Logistic Regression for Heart Disease Prediction (1/2)

Logistic Regression Model:

- Binary classification: Predict presence (1) or absence (0) of heart disease
- Probability of heart disease presence: $P(y = 1|x) = \frac{1}{1+e^{-(w^T x+b)}}$

Limitations for Heart Disease Prediction:

- Assumes linear relationship
- Does not inherently capture interactions between features

Logistic Regression for Heart Disease Prediction (2/2)

Principal Component Analysis (PCA) for Logistic Regression:

- PCA transformation: $X_{PCA} = XV$, where V is the matrix of eigenvectors from the covariance matrix of X
- Logistic regression with PCA features: $P(y = 1|X_{PCA}) = \frac{1}{1+e^{-(w^T X_{PCA}+b)}}$

How PCA Optimizes Logistic Regression:

- Reduces dimensionality by selecting top principal components
- Captures most of the variance in the data
- Improves model stability and generalization performance

Evaluation Metrics (1/2)

- **Accuracy:** Measures the overall performance of the model. A high accuracy indicates that the model makes correct predictions most of the time. However, accuracy alone may not be sufficient for imbalanced datasets or when the cost of false positives and false negatives differs significantly.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$$

- **Precision:** Focuses on the model's ability to avoid false positives. A high precision means that when the model predicts a positive instance, it is likely to be correct. Precision is important when the cost of false positives is high, such as in spam email detection or medical diagnosis.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Evaluation Metrics (2/2)

- **Recall (Sensitivity):** Focuses on the model's ability to find all positive instances. A high recall means that the model identifies most of the positive instances, with few false negatives. Recall is important when the cost of false negatives is high, such as in cancer detection or fraud detection.

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1 Score:** Provides a balanced measure of the model's performance by combining Precision and Recall. A high F1 score indicates that the model performs well in both identifying positive instances and avoiding false positives. The F1 score is particularly useful when dealing with imbalanced datasets or when both false positives and false negatives have significant consequences.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Model Evaluation and Results

Best parameters found for KNN:
'classifier__n_neighbors': 3, 'classifier__p': 2, 'classifier__weights': 'distance'
Best parameters found for Logistic Regression:
'classifier__C': 1, 'classifier__penalty': 'l2', 'pca__n_components': None

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Logistic Regression	0.839	0.797	0.913	0.851
K-Nearest Neighbors	0.985	1.000	0.971	0.985

Table: Performance Metrics

Visual Representation of KNN Results

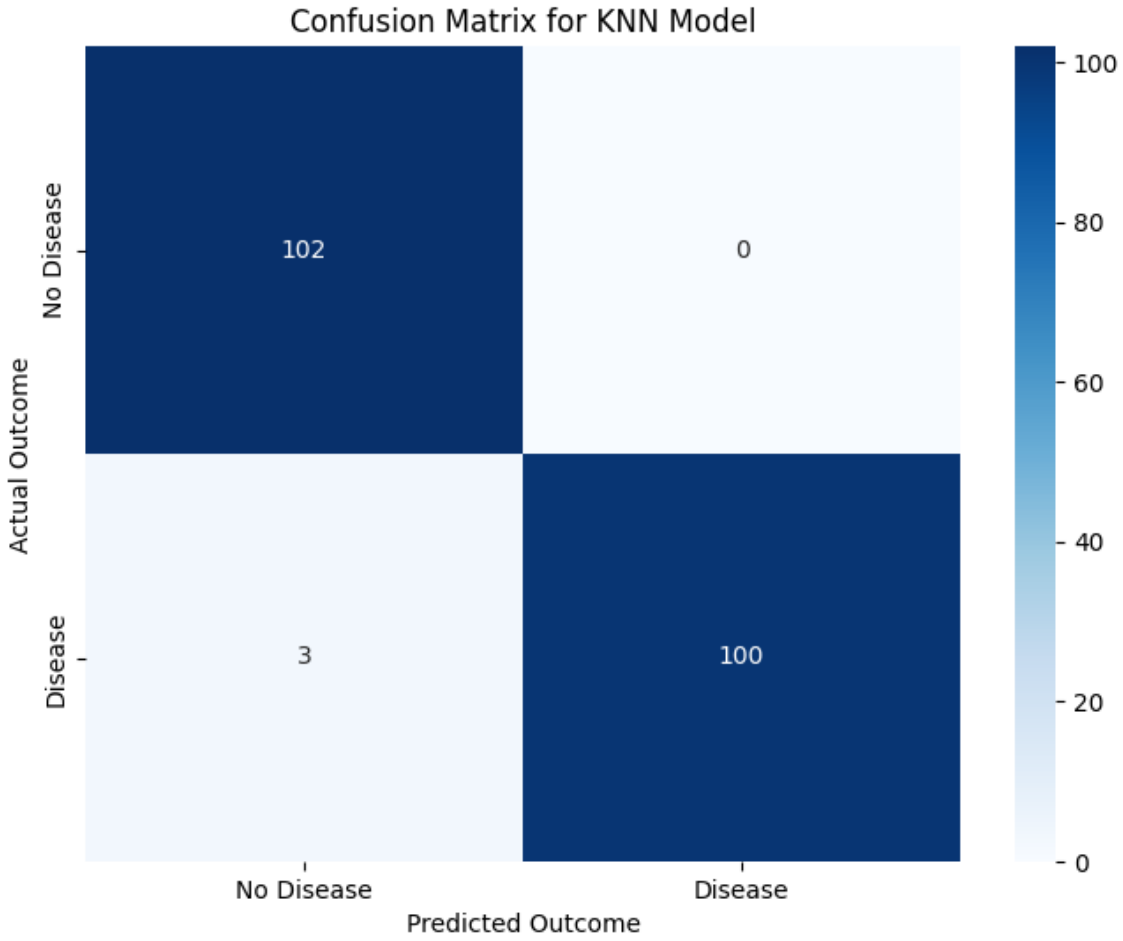

Figure: KNN Results

Visual Representation of Logistic Regression Results

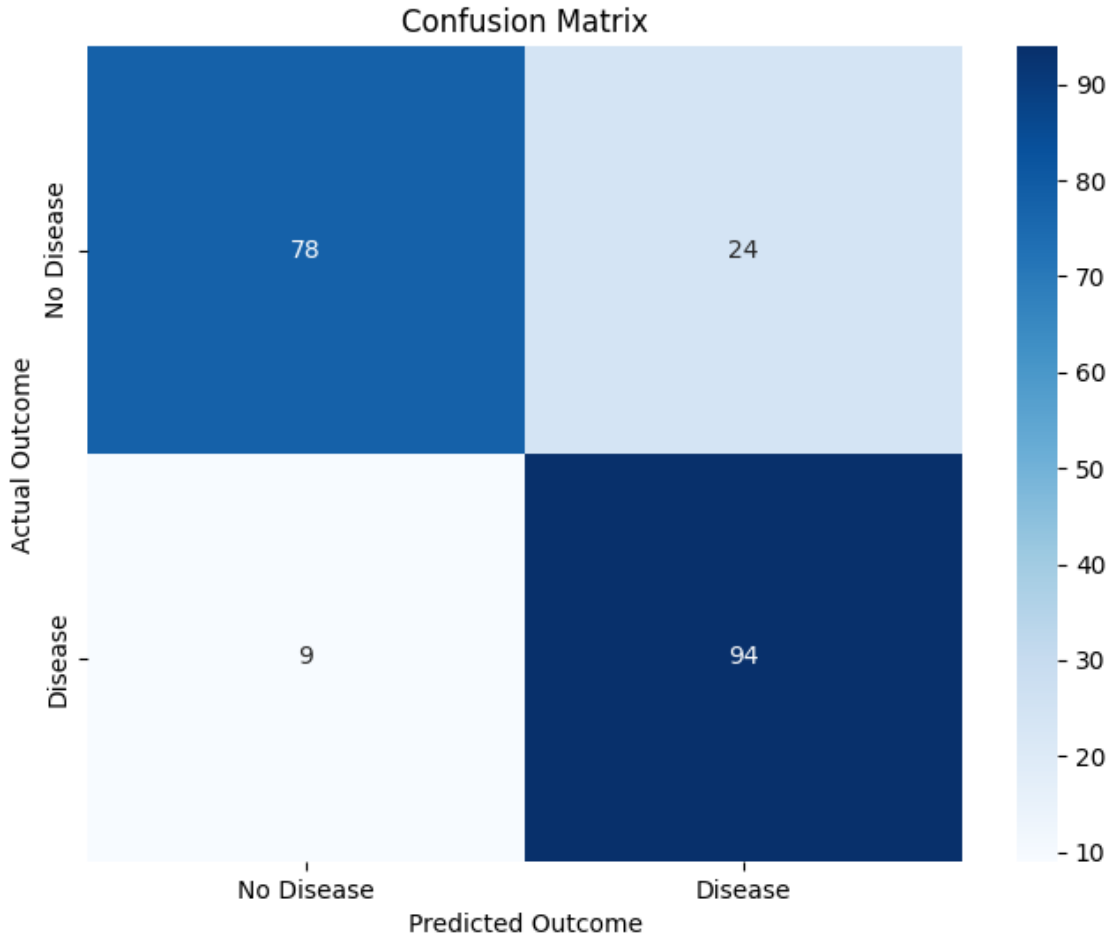

Figure: Logistic Regression Results

Conclusion: KNN vs Logistic Regression

Why KNN Worked Better:

- Handles non-linear data well
- Automatically considers interactions between features
- Makes predictions based on trained data patterns
- Less affected by irrelevant features

Things to Keep in Mind:

- The best model depends on the specific data and problem
- Logistic Regression is simpler and easier to interpret
- Both models can be improved by fine-tuning parameters and selecting better features

Link to code

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