

# The Classification and Prediction of Cardiovascular Disease Based on Machine Learning Algorithms

## Summary

The diagnosis and treatment of cardiovascular diseases (CVD) have imposed considerable pressure on the healthcare industry. Therefore, researchers in the field are exploring the integration of machine learning into the domain of CVD classification prediction to assist doctors in accurate diagnoses and provide higher-quality medical services for patients. Building upon this, our team introduced a substantial number of individual classifiers based on different ensemble methods. Through continuous training, optimization, and comparison, we sought high-quality solutions to address the CVD classification prediction problem.

Firstly, we conducted meticulous preprocessing of the dataset provided. The entire preprocessing phase was divided into three parts: data cleansing to remove redundancy and noise, eliminating interference and enhancing data purity and consistency; Min-Max normalization of data features to maintain comparability due to the extensive scale of the data; and the use of a brute-force algorithm to select outstanding feature combinations for different models. In this part, through systematic feature selection, we optimized the input for subsequent classifiers, achieving superior performance.

Next, we performed exploratory data analysis on the effectively preprocessed dataset. Based on data types, we categorized the data into two groups, presented the basic distribution of data in tabular form, and further created a correlation heatmap, forming a clear overview of attribute feature correlations. Subsequently, we binary-classified the dataset based on disease status, presenting the correlation between attributes and disease status more directly.

Based on the preprocessed dataset, we selected fundamental classifiers such as KNN, Naive Bayes, and Logistic Regression for model construction and training. We then introduced ensemble learning methods such as Voting, Bagging, and Boosting to enhance model effectiveness. Additionally, feature selection was introduced, further strengthening the model's performance by choosing appropriate feature combinations. By comparing the accuracy of the model under different methods and feature combinations, we determined that the optimal model is based on feature-selected LightGBM, achieving an accuracy of 72.97%.

Finally, we discussed the strengths of this study and areas for improvement, providing directions for future research.

**Key Words:** cardiovascular; machine learning; ensemble learning; feature selection; classification; KNN; logistic regression; decision tree;

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Background	1
1.2	Related Work	1
1.3	Our approach	2
<b>2</b>	<b>Assumptions and Data Preprocessing</b>	<b>3</b>
2.1	Assumptions	3
2.2	Data Preprocessing	3
2.2.1	Data Cleaning	3
2.2.2	Data Transformation	4
2.2.3	Feature Selection	4
<b>3</b>	<b>Exploratory Data Analysis</b>	<b>5</b>
<b>4</b>	<b>Machine Learning Methods for Classification</b>	<b>8</b>
4.1	Individual Learning Algorithms	8
A	K-Nearest Neighbors	8
B	Naive Bayes	8
C	Decision Tree	9
D	Logistic Regression	9
E	Support Vector Machine	10
F	Multilayer Perceptron	10
4.2	Ensemble Methods	10
A	Voting	11
B	Bagging	11
C	Boosting	11
<b>5</b>	<b>Simulation Result and Model evaluation</b>	<b>13</b>
5.1	Result of Individual Classifier	13
5.2	Result of Ensemble Methods	14
A	Voting	14
B	Bagging	14
C	Boosting	15
5.3	Result of Feature Selection	15
5.4	Result of Comprehensive Evaluation	16
<b>6</b>	<b>Conclusion</b>	<b>17</b>
6.1	Strengths:	17
6.2	Possible Improvement:	18
	<b>Appendices</b>	<b>19</b>

# 1 Introduction

## 1.1 Problem Background

The modern world has cardiovascular disease as its deadliest enemy. [2]

As a severe health issue involving the heart and vascular system, cardiovascular disease, has resulted in a substantial number of deaths and disabilities globally, characterized by its high incidence and severity.

The incidence of CVD is closely associated with changes in modern lifestyles, increased work-related stress, environmental pollution, and unhealthy habits. In contemporary society, the fast-paced and unhealthy way of life exposes a greater number of individuals to the risk of cardiovascular diseases. Therefore, accurate and prompt diagnosis is crucial. Much care should be taken to introduce machine learning methods, aiming to uncover hidden patterns and regularities within data. These methods assist healthcare professionals in making accurate diagnoses, reducing misunderstandings and misdiagnoses, and ultimately providing higher-quality medical services for individuals suffering from CVD.

## 1.2 Related Work

In the field of clinical decision support development for cardiovascular diseases, research primarily focuses on the integration of machine learning models with information systems to form hybrid models. Ongoing efforts in the development, selection, and optimization of models aim to ensure the feasibility and robustness of these models in real medical environments, ultimately achieving high precision and efficiency.

Various machine learning algorithms applied in the prediction of cardiovascular diseases show varying degrees of accuracy, with the emphasis of research placed on continuous optimization:

- Some groundbreaking developments in research are concentrated on the **application of model ensemble methods and the improvement of ensemble techniques**. For instance, [3] adopts an ensemble model incorporating various classifiers such as naive Bayes, random forest (RF), and utilizes bagging and boosting ensemble techniques. This successful integration enhances the accuracy of weak classifiers, and through majority voting, significantly improves the accuracy of the cardiovascular disease prediction model. The performance in identifying heart disease risk is considered satisfactory.
- On another note, some studies focus on the **precise identification of key risk factors for cardiovascular diseases**. Amin et al. [1] apply machine learning algorithms including K-NN, SVM, and a hybrid model with NB and LB voting. Through comprehensive comparative analysis, the hybrid model, along with selected attributes, achieves a significant improvement.
- Furthermore, through the **improvement of feature selection algorithms** and the combination of classification model techniques with the selection of feature sub-

sets, some studies have successfully enhanced model performance. In the research by Saqlain et al. [6], the application of the MFSFSA and SVM combination yields ideal results, improving the accuracy, sensitivity, and specificity of the predictive model.

### 1.3 Our approach

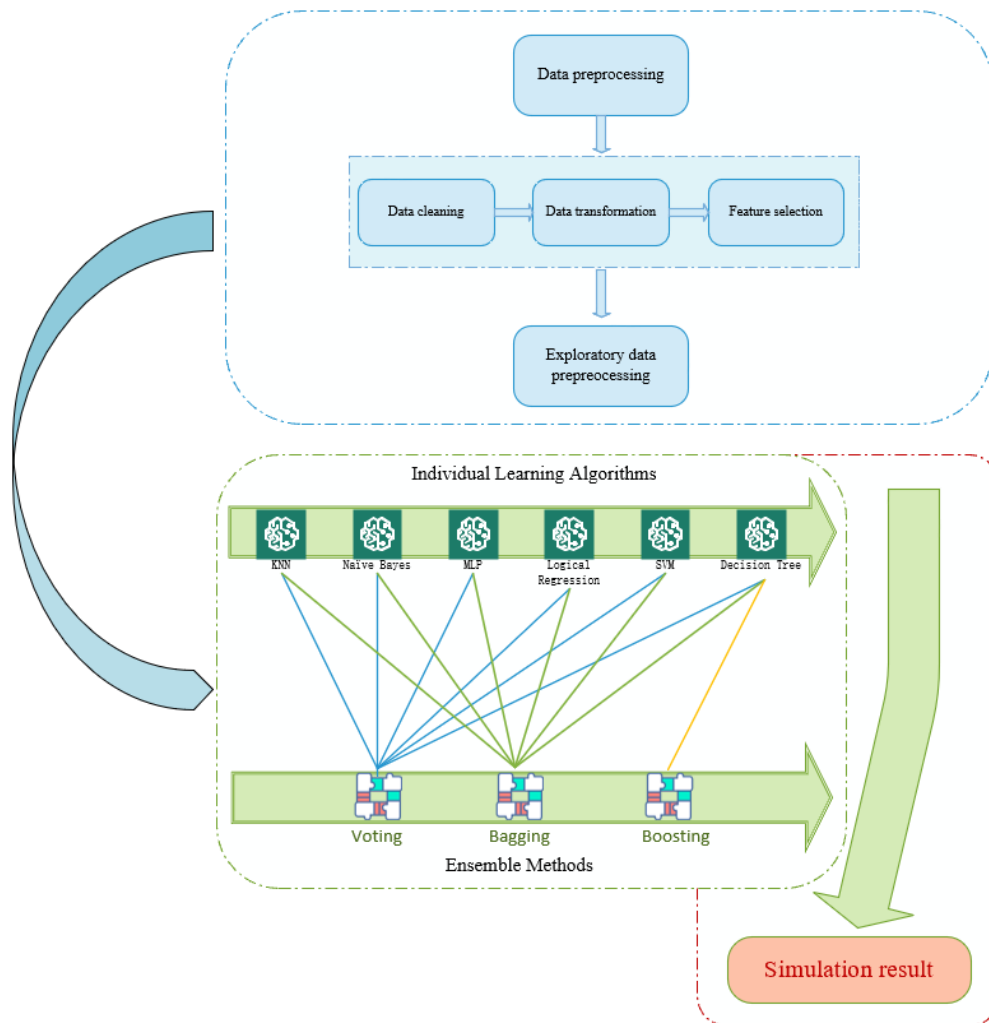


Figure 1: Overview of our work

According to the attached database, we conducted a detailed preprocessing of the provided dataset to ensure its high quality and adaptability. We performed data cleansing, eliminating redundancy and noise, thereby enhancing the purity and consistency of the data. To maintain consistency and comparability, we normalized the data features to ensure they share similar scales, mitigating potential impacts arising from scale differences. Subsequently, we utilized a brute-force algorithm to select outstanding feature combinations, optimizing the model's accuracy. This step aims to systematically enhance the classifier's input through feature selection, achieving a

higher level of performance.

Next, we trained multiple individual learners on the preprocessed dataset, optimizing and adjusting them to obtain the best hyperparameter configurations and maximize the performance of each classifier. Various ensemble methods were introduced to strengthen the overall performance of these weak classifiers. This includes optimizing hyperparameters of the ensemble model to ensure optimal collaborative effects. Finally, we conducted a comprehensive performance evaluation and comparison of individual learners and ensemble learners, identifying the model with the best performance and providing a reliable foundation for further research and applications.

## 2 Assumptions and Data Preprocessing

### 2.1 Assumptions

In order to simplify the problem, we have made the following fundamental assumptions, each of which is assumed to be accurate and reasonable:

- All the data is authentic.
- The features provided by the dataset play a significant role in predicting heart disease.
- The dataset is representative of the studied problem, effectively reflecting the characteristics of the target population.
- The machine learning model demonstrates good generalization capabilities in medical environments and across diverse populations.
- Ethical considerations and patient privacy have been taken into account in the research.

### 2.2 Data Preprocessing

#### 2.2.1 Data Cleaning

Based on the files `cardio_train.csv` provided in the appendix, a screening and cleaning process was conducted to address the noise present in the data. This aimed at resolving inconsistencies in the data, mitigating the impact of sensor malfunctions, human input errors, or abnormal events on the model detection:

- **Noise Data Handling:** Initially, illogical data was manually removed to minimize its impact on the model quality, despite its relatively small proportion in the dataset.
- **Outlier Treatment:**

Visual statistical analysis using box plots was employed for rapid identification of outliers. Given that the quantity of outlier points represents a marginal

share of the total dataset, exerting a marginal influence and lacking statistical significance, these outlier records were directly removed.

Applying the 3 principle, points falling within the  $P(|x - \mu| > 3\sigma) \leq 0.003$  range are defined as outliers. An examination and refinement process for outlier cleaning is undertaken to ensure thoroughness.

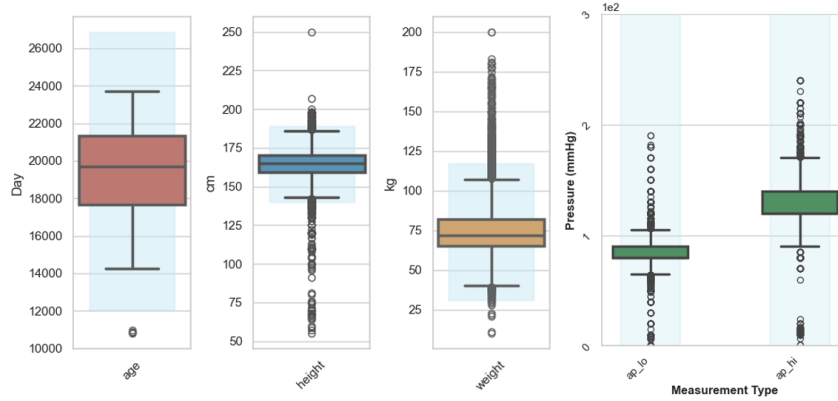


Figure 2: Outlier cleaning for the original dataset

## 2.2.2 Data Transformation

The data in the cardio\_train dataset exhibits disparate scales; therefore, a Min-Max normalization strategy is employed. This approach scales the data to the [0,1] interval, as illustrated by the linear transformation in equation  $Y = \frac{X - \min}{\max - \min}$ .

The mapped dataset eliminates magnitude discrepancies, facilitating machine learning models by balancing feature scales, enhancing robustness, and improving visualization effects.

## 2.2.3 Feature Selection

We performed further engineering on the raw data, selecting key features from the original feature set that significantly impact model performance, and completing the feature selection process. Impurities and redundancies were eliminated from the original data, reducing the risk of model overfitting.

Utilizing a brute-force algorithm, we individually identified the most effective feature combinations for each classifier. These selected feature combinations were then tested on other classifiers, resulting in a total of 7 distinct feature combinations. Different classifiers were employed to conduct tests on various feature combinations.

From the results of feature set selection, it is evident that the features "age," "ap\_hi," and "cholesterol" are consistently included in all chosen feature sets, indicating their significant impact on model performance. Additionally, the absence of the "smoke" feature in the selected sets suggests a potential detrimental effect of "smoke" on the model's performance.

### 3 Exploratory Data Analysis

Based on the Kaggle data science platform, an authoritative diagnostic database for cardiovascular diseases was retrieved. The original database comprised nearly one hundred thousand records. Through data preprocessing, approximately 63,000 valid entries were retained.

The given dataset includes 13 features, such as patient ID (Id), four physiological indicators (Age, Height, Weight, Gender), four medical indicators (Systolic pressure, Diastolic pressure, Cholesterol, Glucose), three subjective information (Condition of Smoke, Alcohol consumption, Physical Activity), and the patient's health condition.

For data analysis, the dataset is divided into two groups based on features, and the basic distribution of the data is presented in tabular form. The first group consists of indicators with statistical significance in the parameters themselves, and Table [1] provides the corresponding maximum, average, and standard deviation for each attribute. The second group comprises options representing data parameters, and Table [2] presents the frequency of corresponding indicator options.

Table 1: Descriptive Statistics of the First Group of Attributes

	Minimum	Maximum	Mean	Std. Deviation
age /(Day)	14277	23713	19492.109	2456.0378
height /(cm)	140	188	164.48	7.6616
weight /(kg)	32	117	73.7086	13.03788
ap_hi /(mmHg)	95	169	126.29	13.732
ap_lo /(mmHg)	66	104	81.7	7.5852

From the table statistics, it is observed that the age range of the dataset is concentrated in the middle-aged group, with the majority falling between 40 and 65 years old. More than 90% of the subjects are heavy consumers of both tobacco and alcohol, while those with a habit of regular physical activity constitute less than 20%. Additionally, over three-quarters of the subjects exhibit normal levels of cholesterol and blood glucose.

Furthermore, to provide a clear overview of the feature correlations among attributes, a correlation heatmap for all valid data is presented in Figure [3] .

Table 2: Frequency Statistics of the Second Group of Attribute Categories

Statistical Variable	Classification	Frequency	Percent	/%
Gender	Female	40835	64.9	
	Male	22061	35.1	
Cholesterol	Normal	47435	75.4	
	Above normal	8268	13.1	
	Substantially above normal	7193	11.4	
Gluc	Normal	53651	85.3	
	Above normal	4466	7.1	
Smoke	Substantially above normal	4779	7.6	
	Regular	57421	91.3	
Alco	Infrequent	5475	8.7	
	Regular	59597	94.8	
Active	Infrequent	3299	5.2	
	Regular	12330	19.6	
Cardio	Infrequent	50566	80.4	
	Yes	31796	50.6	
	No	31100	49.4	

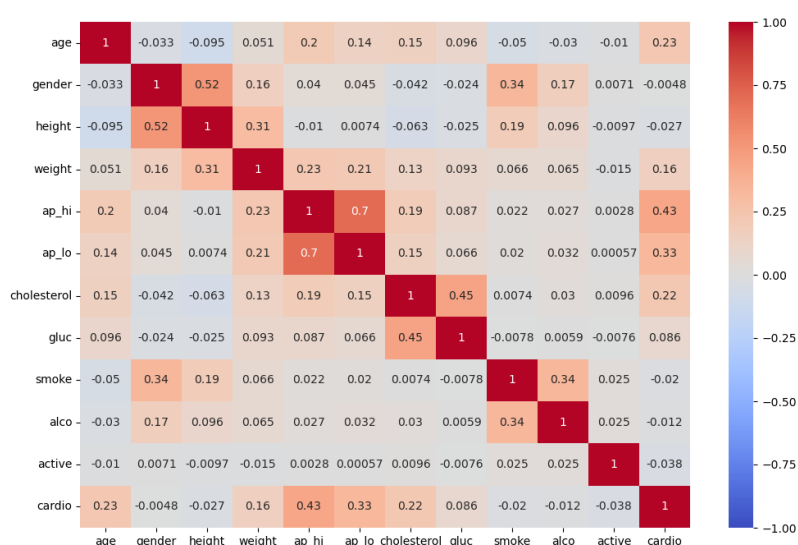


Figure 3: Mutual correlation heatmap

Additionally, we have categorized the obtained valid dataset into two groups based on health conditions: positive (cardio=1) and negative. The pie chart in Figure [3] illustrates that the survey participants are predominantly female, constituting



over 60%, while males account for less than 40%.

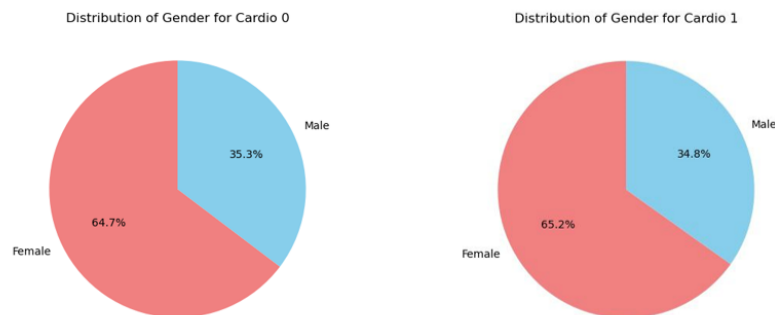


Figure 4: Distribution of gender after classification

Moreover, distribution histograms with overlapping bars were plotted for continuous attributes, enabling a visual representation of the sample data. An overview of the data distribution reveals a positive correlation between extreme values of age, weight, and arterial blood pressure with the incidence of cardiovascular disease.

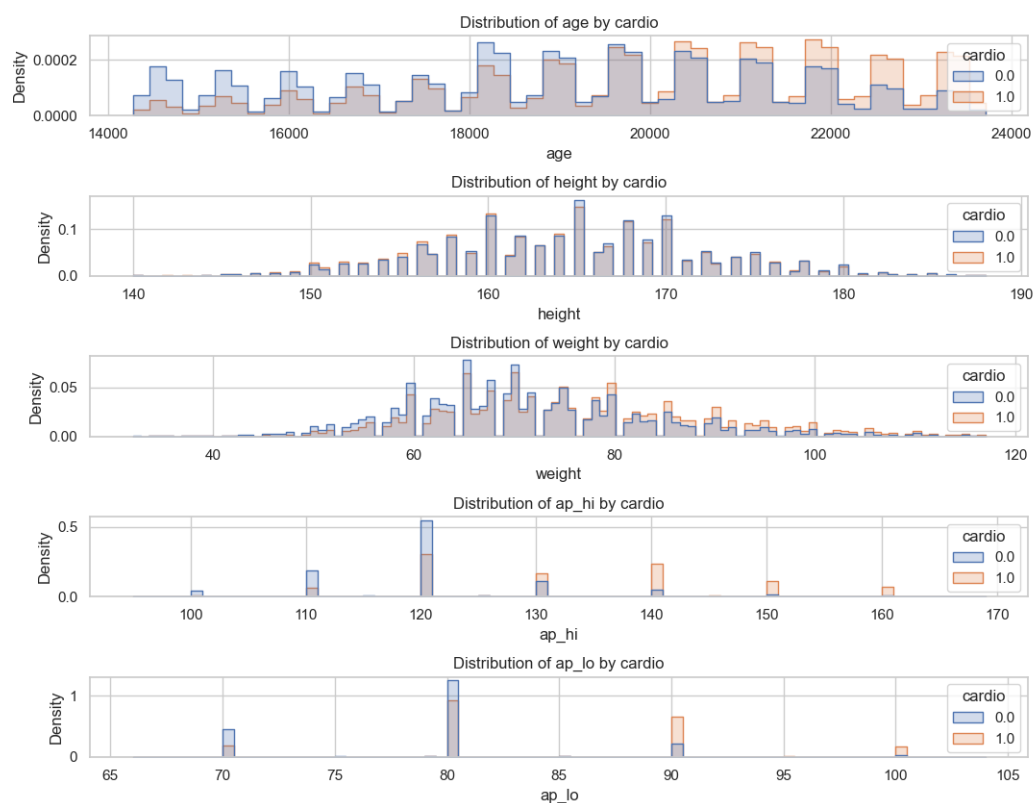


Figure 5: Continuous Attribute Distribution in the Sample Population

## 4 Machine Learning Methods for Classification

To validate the model's generalization ability and facilitate the assessment of its application performance, we conducted an analysis of the dataset and split it into a training set (70%) and a test set (30%). Simultaneously, the 'random\_state' parameter was employed to fix the pseudo-random number seed, ensuring result reproducibility and transparency in outcomes.

Based on this, the following entails the training, optimization, and testing of several classification models:

### 4.1 Individual Learning Algorithms

#### A K-Nearest Neighbors

KNN, as a supervised instance-based learning algorithm, stores the given training samples and their corresponding class labels during the training phase. In the testing phase, the Euclidean distance  $\left(d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}\right)$  is employed as the distance metric to calculate the distance between the test data and all training data. The K-nearest training data points are selected based on this distance, and classification is achieved through a majority vote based on the selected data points.

A combined approach of grid search and cross-validation is utilized for hyperparameter tuning and performance evaluation of the model. **( In this study, we consistently employ the same methodology for the optimization of all models. )**Based on the optimal results from the optimization process, we choose the value of K as 39, representing the number of nearest neighbors.

#### B Naive Bayes

The second model is constructed based on a probability model using the Bayesian theorem, known as Naive Bayes. It assumes that all features are conditionally independent given the category.

During the training phase, a labeled training dataset is input, and the model learns feature attributes and training samples to generate the classifier. The training process initially computes the prior probability  $P(Y = y) = \frac{\text{Number of samples with class } y}{\text{Total number of samples}}$  for each category using Bayesian theorem, and then calculates the conditional probability of the categories based on the feature attributes, as shown in formula (1).

$$\begin{cases} P(X_i = x_j | Y = y) = \frac{\text{Number of samples with class } y \text{ and feature } X_i \text{ value } x_j}{\text{Number of samples with class } y}, & X \in \text{Discrete feature} \\ P(X = x | y) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), & X \in \text{Continuous feature} \end{cases} \quad (1)$$

The computed prior probabilities and conditional probabilities are stored and applied during the prediction process. For predictions, the category with the maximum posterior probability is selected as the predicted result.[5]

### C Decision Tree

The third model adopted a tree-based decision tree model, which divides the dataset into different subsets through feature partitions at each branching node and ultimately represents categories through leaf nodes.

During the training process, the model selects the optimal partitioning feature for the loaded dataset by computing information gain, as shown in Formula (2). Based on the partitioning feature, the data is divided, and the partitioning process is recursively applied to each subset. A complete decision tree is constructed using the Gini coefficient  $\text{Gini}(D) = 1 - \sum_{i=1}^c p_i^2$ .

$$\text{Information Gain}(D, A) = \text{Entropy}(D) - \sum_{v=1}^V \frac{|D_v|}{|D|} \cdot \text{Entropy}(D_v) \quad (2)$$

In this context,  $D$  represents the original dataset,  $A$  is the feature to be partitioned,  $V$  denotes the number of values for feature  $A$ , and  $D_v$  represents the subset of data where feature  $A$  takes the value  $v$ .

After building the tree, pruning is performed to remove unnecessary branches, preventing overfitting and enhancing the model's generalization performance. Through continuous optimization, the finalized ideal tree depth is determined to be 6.

### D Logistic Regression

The fourth model introduces the logistic regression algorithm for binary classification problems. It models the relationship between the target label and multiple independent variables based on the foundational Sigmoid function, estimating probabilities. The training data matrix is introduced, and predictions are obtained through hypothesis function  $h_{\theta}(x)$  operations to assess sample classification. The hypothesis function employs the Sigmoid function, as shown in Formula (3)(4).

$$f(x) = \frac{1}{1 + e^{-z}} \quad (3)$$

Here,  $f(x)$  represents the output probability,  $e$  is the base of the natural logarithm, and  $z$  is the linear combination of the model:

$$z = b_0 + b_1 \cdot x_1 + b_2 \cdot x_2 + \dots + b_n \cdot x_n \quad (4)$$

In this equation,  $b_0$  is the intercept, and  $b_1, b_2, \dots, b_n$  are the weights associated with features  $x_1, x_2, \dots, x_n$ .

To minimize the loss, train the model using the training data and the loss function.

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[ y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] \quad (5)$$

Here,  $m$  is the number of training samples, and  $y^{(i)}$  is the actual label for the  $i$ -th sample.

Apply model optimization algorithms to adjust model parameters, introducing a regularization strength  $\lambda$  of 0.01 to prevent overfitting. For predictions on new data, output a value between 0 and 1, where samples with a predicted value  $\geq 0.5$  are classified as positive class, and those with a value  $< 0.5$  are predicted as negative class.

## E Support Vector Machine

The SVM algorithm operates in a high-dimensional space by **finding an optimal hyperplane** that projects data points, effectively classifying binary data regions.

During training, the model selects a suitable kernel function, simultaneously constructing an objective function to maximize the margin. This involves calculating the weights and bias for support vectors, resulting in the equation for the hyperplane.

$$w \cdot x + b = 0 \quad (6)$$

Where  $w$  is the normal vector,  $x$  is the input feature vector, and  $b$  is the bias

By experimenting with various parameter combinations, grid search identifies the model configuration that performs the best within the specified parameter space. Regularization parameters are explored with three different C values [0.1, 1, 10], and the value 1 is selected. The kernel function is determined to be a radial basis function (**rbf**) with a coefficient of **Scale**, resulting in the ideal decision boundary. This optimized model can be applied for accurate classification or prediction of new samples.

## F Multilayer Perceptron

For the final multilayer perceptron (MLP) model, we initialized the weights and bias parameters in the neural network. The parameter configuration was set as 'hidden\_layer\_sizes=(100, 100, 100)', indicating the presence of 3 hidden layers, each comprising 100 neurons. Simultaneously, the maximum number of iterations for the learning rate was specified.

The training dataset was fed into the network, and, in conjunction with forward and backward propagation, the parameter updates were performed iteratively. Subsequently, the model underwent actual deployment.

## 4.2 Ensemble Methods

After completing the training optimization for individual classifiers, the concept of ensemble learning is introduced. By judiciously selecting individual classifiers

and adopting effective ensemble strategies, the performance of weak classifiers is enhanced (Polikar, 2012)[4].

For this purpose, we explore three different combination strategies:

### A Voting

Voting is adopted as the first ensemble learning strategy, integrating predictions from individual classifiers by voting on their results. We have experimented with two widely used voting strategies:

- **Hard Voting** : Relying on the discrete prediction outcomes of the individual classifiers selected earlier, hard voting determines the final predicted category through a majority voting principle.
- **Soft Voting** : Taking into account the probability outputs of individual base learners, obtain the ultimate ensemble prediction through a weighted average of the probabilities assigned by each learner to every category.

### B Bagging

The second ensemble learning strategy, Bagging, achieves bootstrapped sampling through Bootstrap to generate multiple training sets. Each dataset is utilized to train an individual base learner, and the ultimate fusion of the predictive outcomes of these base learners occurs through a voting mechanism. It generates multiple sets of data from a random sample of the original dataset with replacement to reduce variance.

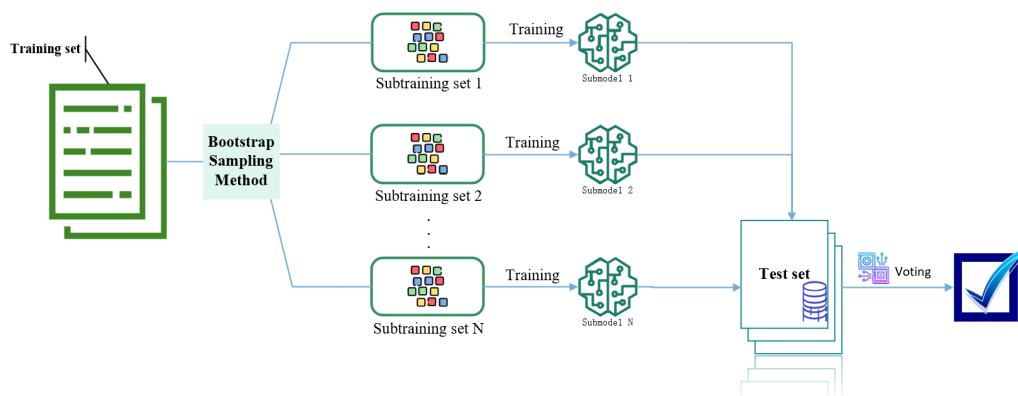


Figure 6: Overview of Bagging

### C Boosting

Introducing the iterative machine learning technique of boosting involves sequentially training a series of individual learners through iterations. In this process, each

subsequent learner continually corrects the errors of its predecessor, gradually improving the overall model performance. We employed four ensemble learning algorithms: AdaBoost, GBDT (Gradient Boosting Decision Trees), XGBoost, and LightGBM. Each algorithm was individually tested for its performance on the dataset.

To begin with, AdaBoost iteratively trains a series of weak learners, demonstrating its strength in effectively adjusting to samples with different features and enhancing the overall model performance.

Gradient Boosting Decision Trees, representing gradient boosting decision trees, sequentially train decision trees, with each tree correcting the residuals of the previous one. The final prediction results are obtained by summing the predictions of all trees, endowing the model with strong fitting capabilities, particularly suitable for complex non-linear relationships.

XGBoost, building upon GBDT, introduces optimizations such as regularization terms, second-order gradient boosting algorithms, and parallel computation of feature column blocks. These techniques enhance both the training speed and generalization ability of the model, leading to superior performance.

LightGBM, a lightweight machine learning algorithm, adopts histogram-based decision tree learning. It reduces memory and computational costs through histogram-based splitting, while also supporting parallel learning and processing of large-scale datasets.

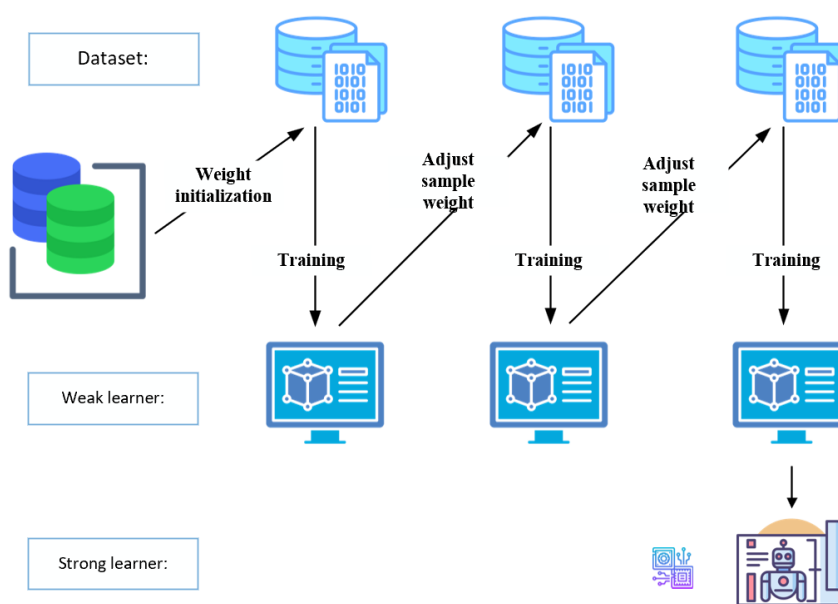


Figure 7: Overview of Boosting

## 5 Simulation Result and Model evaluation

### 5.1 Result of Individual Classifier

Regarding the performance comparison of individual classifiers, we observed that multi-layer perceptrons (MLP) showed the best accuracy, followed by support vector machines (SVM) and logistic regression. In contrast, the performance of naive Bayes and decision trees is relatively low. Digging into the confusion matrix revealed that the number of false positive cases (FP) was significantly higher than false negative cases (FN), suggesting that the model was more likely to misdiagnose normal samples as cases. This reflects that the model has some bias in classifying normal and sick samples.

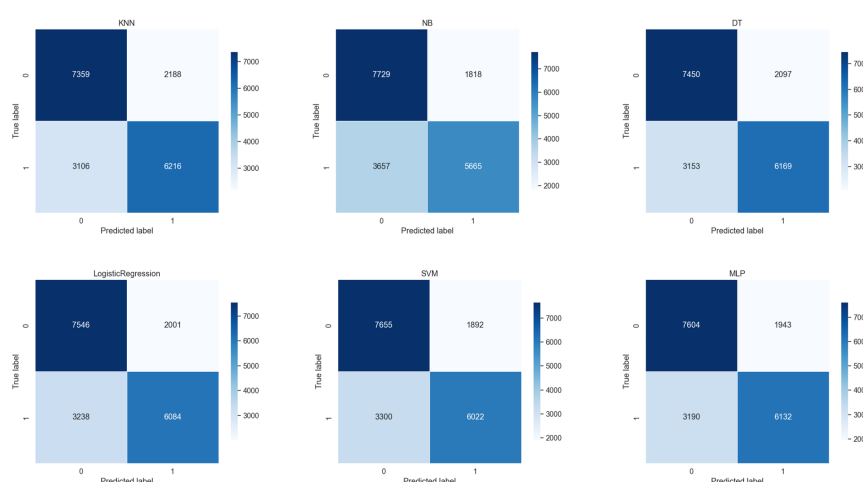


Figure 8: Confusion matrix of all individual classifier

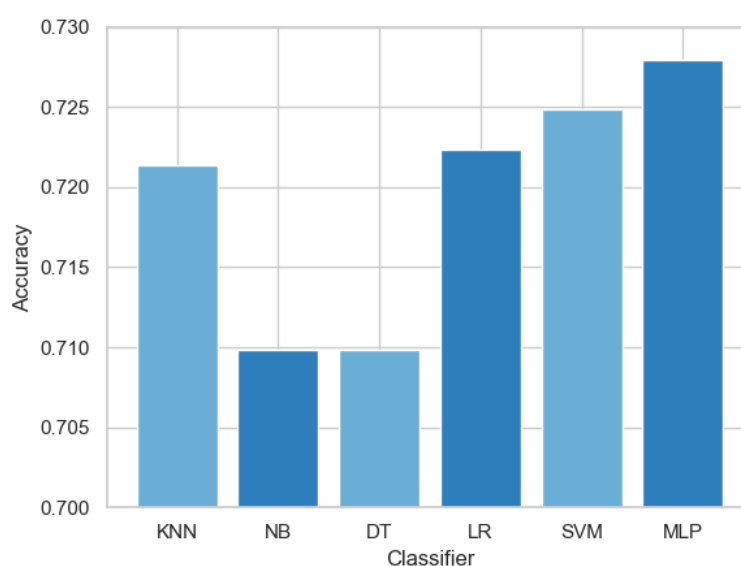


Figure 9: Result of individual classifier

## 5.2 Result of Ensemble Methods

### A Voting

The experimental results show that the integrated method of soft voting can get higher accuracy than hard voting when processing CVD detection. However, it is also possible that the accuracy of the integrated method using voting is lower than that of a single classifier algorithm.

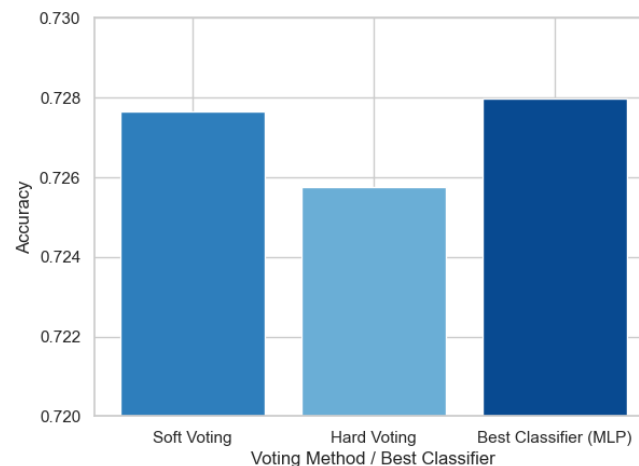


Figure 10: Result of Voting

### B Bagging

As shown in Figure [B], after applying the Bagging ensemble method, the accuracy of most models either decreased or remained unchanged. However, there was a notable improvement in the accuracy of the decision tree model, increasing from 70.9% to 72.0%.

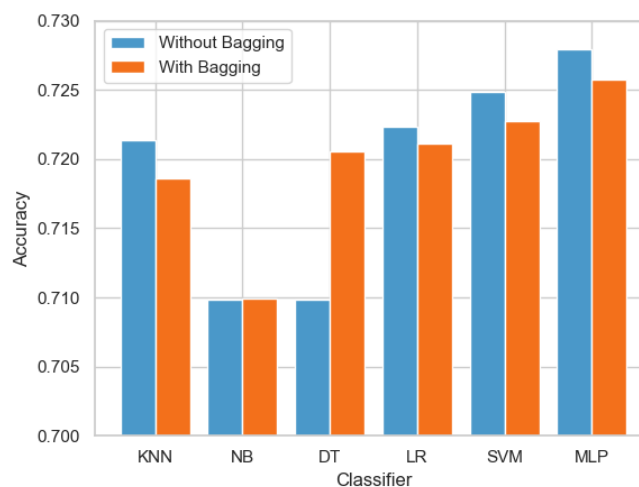


Figure 11: Result of Bagging



## C Boosting

Following the application of GridSearch for parameter selection, substantial disparities in accuracy performance among the four Boosting methods were observed. XGBoost exhibited the optimal performance, achieving a notable accuracy of 72.86%. Following closely was LightGBM, attaining an accuracy of 72.81%. In contrast, AdaBoost and Gradient Boosting Decision Trees demonstrated comparatively inferior performance in terms of accuracy.

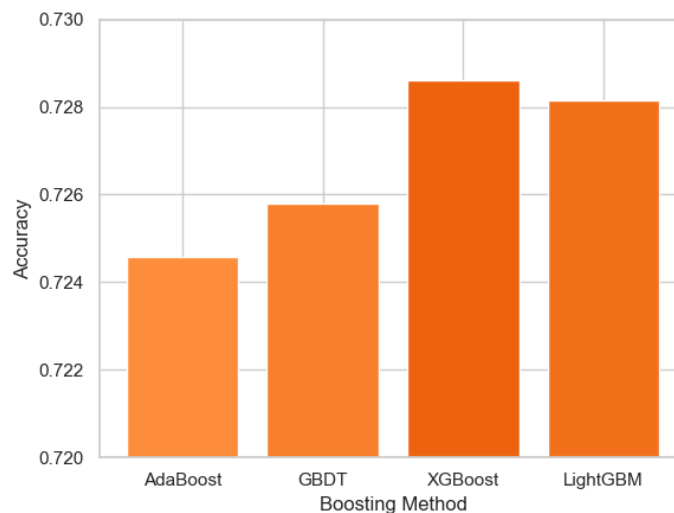


Figure 12: Result of Boosting

## 5.3 Result of Feature Selection

Through a brute-force algorithm, the most effective feature combinations were individually selected for each classifier. These selected feature combinations were then tested on other classifiers, resulting in the identification of five distinct feature combinations. Different classifiers were evaluated using these feature combinations.

Analyzing the results of feature set selection, it is notable that the features "age," "ap\_hi," and "cholesterol" are consistently present in all the selected feature sets. This implies that these three features have the most significant impact on the model's performance. Additionally, it is observed that the feature "smoke" is not included in any of the selected feature sets, suggesting that "smoke" may have a negative impact on the model's performance.

Table 3: Selected feature combinations

F1	('age', 'weight', 'gender', 'ap_hi', 'ap_lo', 'cholesterol', 'gluc', 'alco', 'active')
F2	('age', 'ap_hi', 'ap_lo', 'cholesterol', 'active')
F3	('age', 'gender', 'ap_hi', 'ap_lo', 'cholesterol', 'active')
F4	('age', 'gender', 'ap_hi', 'cholesterol', 'alco')
F5	('age', 'gender', 'ap_hi', 'ap_lo', 'cholesterol', 'gluc', 'active')

After feature engineering was used to select features, the accuracy of all models except MLP and SVM improved, and the two models NB and DT improved the most. In addition, the LightGBM model has the best overall effect, and the accuracy rate can reach 72.97% after feature selection.

Table 4: **Model representation based on feature selection**

	F1	F2	F3	F4	F5
LR	0.72351	0.71890	0.71853	0.71837	0.72097
NB	0.714467	0.72576	0.72576	0.72051	0.71701
DT	0.72258	0.72353	0.72464	0.71606	0.72289
RF	0.72129	0.72283	0.72150	0.72606	0.72171
KNN	0.72051	0.71987	0.72369	0.72130	0.72442
SVM	0.72394	0.72304	0.72357	0.72007	0.72319
MLP	0.72589	0.72303	0.72531	0.72150	0.72568

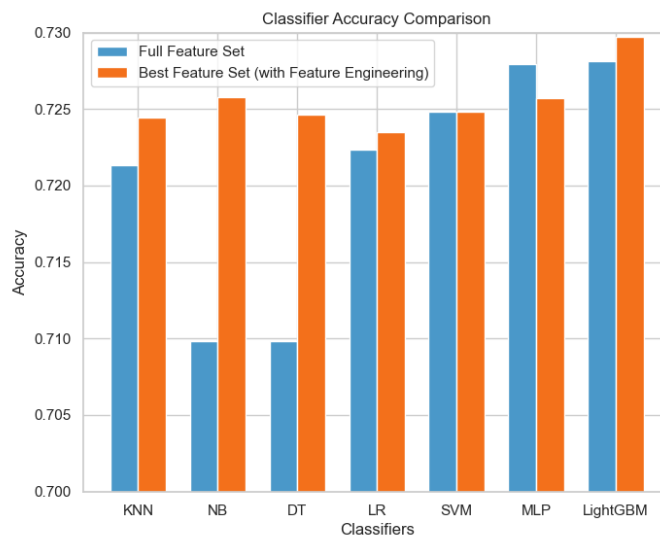


Figure 13: Feature Selection Result

## 5.4 Result of Comprehensive Evaluation

Upon synthesizing all the experiments, the model that exhibited the highest performance was the LightGBM model after feature selection, achieving an accuracy of 72.97%. Following closely was the XGBoost model. It is noteworthy that the MLP (Multi-Layer Perceptron) model demonstrated outstanding accuracy across all individual models and after applying the Bagging ensemble method.

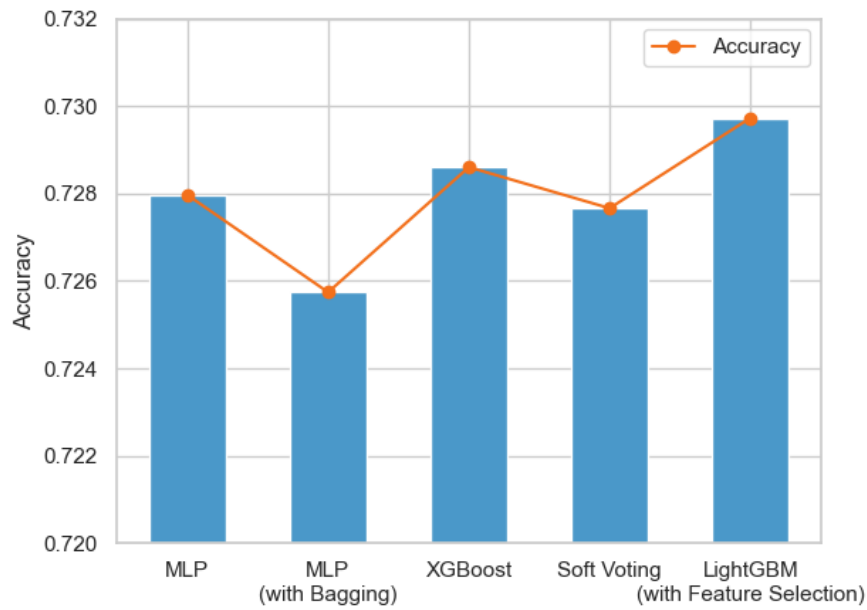


Figure 14: Comprehensive Evaluation Result

## 6 Conclusion

In this paper, we employed both individual learning algorithms and ensemble methods. A comprehensive performance evaluation and comparison, incorporating both horizontal and vertical integration, were conducted. The results identified the model with the optimal performance, which is the Feature-selected LightGBM Model.

### 6.1 Strengths:

- **Data Quality Improvement:** In the data preparation phase, spanning from data cleaning to feature engineering, we conducted comprehensive data preprocessing. We identified outstanding feature combinations for the models, effectively enhancing the accuracy and generalization ability of the models.
- **Multi-Model Comparison:** This study introduced a sufficient number of models. Such multi-model testing provides a more comprehensive and reliable model assessment, aiding in making more informed decisions and selections.
- **Comprehensive Evaluation:** In the final analysis, we conducted a thorough performance evaluation and comparison of both individual and ensemble learners. The optimal model obtained has high credibility.

## 6.2 Possible Improvement:

- **Insufficient Data Samples:** The results of this study heavily rely on the quality of the original data and the correlation of features. Due to the limited diversity, scarcity of features, and lower feature correlation in the original data, the effectiveness of the final model might be affected.
- **Lack of domain expertise:** Due to a lack of specialized medical knowledge, the understanding of medical features is not sufficiently profound, leading to an insufficiently refined model construction.

## References

- [1] Mohammad Shafenoor Amin, Yin Kia Chiam, and Kasturi Dewi Varathan. Identification of significant features and data mining techniques in predicting heart disease. *Telematics and Informatics*, 36:82–93, March 2019.
- [2] Kumar G Dinesh, K Arumugaraj, Kumar D Santhosh, and V Mareeswari. Prediction of Cardiovascular Disease Using Machine Learning Algorithms. In *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, pages 1–7, Coimbatore, March 2018. IEEE.
- [3] C. Beulah Christalin Latha and S. Carolin Jeeva. Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques. *Informatics in Medicine Unlocked*, 16:100203, January 2019.
- [4] Robi Polikar. Ensemble Learning. In *Ensemble Machine Learning*, pages 1–34. Springer, New York, NY, 2012.
- [5] I Rish. An empirical study of the naive Bayes classifier. 3(22):41–46, August 2001.
- [6] Syed Muhammad Saqlain, Muhammad Sher, Faiz Ali Shah, Imran Khan, Muhammad Usman Ashraf, Muhammad Awais, and Anwar Ghani. Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines. *Knowledge and Information Systems*, 58(1):139–167, January 2019.

# Appendices

## Appendix A: Programmes Codes

Here are simulation programmes we used in our model as follow.

**Input Python source:**

---

```
"""
Soft Voting Classifier for Cardiovascular Disease Prediction

This script uses a Soft Voting Classifier, combining RandomForest, KNN,
    Logistic Regression, Naive Bayes, SVM, and MLP classifiers,
to predict cardiovascular disease.

"""

from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, confusion_matrix,
    classification_report
import pandas as pd
from sklearn.neural_network import MLPClassifier
from sklearn.preprocessing import StandardScaler
from sklearn.svm import SVC

# Create StandardScaler object
scaler = StandardScaler()

file_path = 'cardio_train_cleaned_2.csv'

# Read the dataset
df = pd.read_csv(file_path)

# Select features and target variable
features = ['age', 'height', 'weight', 'ap_hi', 'ap_lo', 'cholesterol',
    'gluc', 'smoke', 'alco', 'active']
target = 'cardio'

# Extract features (X) and target variable (y)
X = df[features]
y = df[target]

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
    random_state=66)

# Standardize the features
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

```
# Create individual classifiers
rf_classifier = RandomForestClassifier(n_estimators=50, random_state=42)
knn_classifier = KNeighborsClassifier(n_neighbors=39)
logreg_classifier = LogisticRegression(max_iter=1000, random_state=42)
nb_classifier = GaussianNB()
svm_classifier = SVC(kernel='rbf', probability=True)
mlp_classifier = MLPClassifier(hidden_layer_sizes=(100, 100, 100),
                               max_iter=30, random_state=42, learning_rate='constant')

# Create a Soft Voting Classifier
voting_classifier = VotingClassifier(
    estimators=[
        ('random_forest', rf_classifier),
        ('knn', knn_classifier),
        ('logistic_regression', logreg_classifier),
        ('naive_bayes', nb_classifier),
        ('svm', svm_classifier),
        ('mlp', mlp_classifier)
    ],
    voting='soft' # Use soft voting
)

# Train the Soft Voting Classifier
voting_classifier.fit(X_train_scaled, y_train)

# Make predictions
y_pred_voting = voting_classifier.predict(X_test_scaled)

# Evaluate model performance
accuracy_voting = accuracy_score(y_test, y_pred_voting)
conf_matrix_voting = confusion_matrix(y_test, y_pred_voting)
class_report_voting = classification_report(y_test, y_pred_voting)

# Print results
print(f'Accuracy using Soft Voting Classifier: {accuracy_voting}')
print(f'Confusion Matrix:\n{conf_matrix_voting}')
print(f'Classification Report:\n{class_report_voting}')



---


"""
This script uses the LightGBM classifier to predict cardiovascular disease
based on a cleaned dataset.
It includes data preprocessing, model training with hyperparameter tuning,
and evaluation using accuracy metrics.
"""

import pandas as pd
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from lightgbm import LGBMClassifier
from sklearn.metrics import accuracy_score, classification_report,
confusion_matrix
import seaborn as sns
```

```
import matplotlib.pyplot as plt

# Read CSV file
file_path =
    r'C:\Users\wangsiyu\Desktop\Problem\Problem\cardio_train_cleaned.csv'
df = pd.read_csv(file_path)

# Extract features and labels
X = df[['age', 'height', 'weight', 'gender', 'ap_hi', 'ap_lo',
        'cholesterol', 'gluc', 'smoke', 'alco', 'active']]
y = df['cardio']

# Split the dataset
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1,
    random_state=42)

# Standardize numerical features
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize the LightGBM model
model = LGBMClassifier()

# Define the range of LightGBM parameters
param_grid = {
    'learning_rate': [0.1, 0.05, 0.01],
    'n_estimators': [50, 100, 200],
    'max_depth': [3, 5, 7]
}

# Create a GridSearchCV object
grid_search = GridSearchCV(model, param_grid, cv=3, scoring='accuracy')

# Perform grid search
grid_search.fit(X_train_scaled, y_train)

# Output the best parameters
print("Best Parameters:", grid_search.best_params_)

# Use the model with the best parameters for prediction
best_model = grid_search.best_estimator_
y_pred_best = best_model.predict(X_test_scaled)

# Evaluate model performance
print("Accuracy with Best Parameters:", accuracy_score(y_test,
    y_pred_best))
print("Classification Report with Best Parameters:\n",
    classification_report(y_test, y_pred_best))

# Generate confusion matrix
cm = confusion_matrix(y_test, y_pred_best)
```

```
# Use Seaborn to plot the heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['No
    Cardio', 'Cardio'], yticklabels=['No Cardio', 'Cardio'])
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

---



## Report on Use of AI

1. OpenAI ChatGPT (Jan 19, 2024 version, ChatGPT-3.5,)

Query1: Please help me explain lightgbm

Output1:LightGBM is a machine learning library based on a gradient lifting framework, specifically designed to handle large-scale data sets and high-dimensional features. Its full name is "Light Gradient Boosting Machine". Here are some of the key features and explanations of LightGBM:

1.Gradient Lifting Framework: LightGBM uses the gradient lifting approach, an ensemble learning technique that builds a robust model by combining multiple weak learners.

2.Distributed training: LightGBM supports distributed training, enabling it to efficiently process large data sets. This is important for tasks dealing with large amounts of data and high-dimensional features.

3.Histogram optimization: LightGBM uses histogram algorithms to discretize features, thereby reducing memory usage and speeding up the training process. This helps with efficiency issues when dealing with large data sets.

4.Leaf-wise growth: LightGBM uses the leaf-wise growth strategy to reduce the loss function more efficiently than traditional level-wise growth.

5.Support for Class features: LightGBM can handle class features directly without additional coding. This makes it more convenient when working with data that contains categorical variables.

6.High performance: Thanks to the optimization techniques described above, LightGBM generally offers higher training speeds and a lower memory footprint, making it ideal for handling large amounts of data and complex models.

Overall, LightGBM is a powerful and efficient machine learning library, especially suited for tasks that require processing large and high-dimensional data sets, such as click-through rate prediction, ranking, and other similar problems.