IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 9, September 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

© 2024 IJIRCCE | Volume 12, Issue 9, September 2024| DOI: 10.15680/IJIRCCE.2024.1209026

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Advanced Machine Learning Classifiers and Ensemble Techniques for Enhanced Predictive Accuracy in Heart Disease Diagnosis

Nandita Vivek Suryawanshi

Student, Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik, India

ABSTRACT: Cardiovascular disease (CVD) remains a leading cause of mortality worldwide, emphasizing the need for effective diagnostic tools.Underlying health issues and lack in their timely detection highly contribute to the spike in mortality every year. It is unanimously agreed by healthcare providers that early and accurate detection of diseases are essential to reduce the alarming mortality rate. With advancements in technology, research in artificial intelligence and machine learning algorithms, several studies have incorporated computer knowledge into the healthcare industry.Early detection of this disease is vital to save people's lives. Machine Learning (ML), an artificial intelligence technology, is one of the most convenient, fastest, and low-cost ways to detect disease. This paper presents a study on heart disease prediction using machine learning algorithms applied to a public dataset, focusing on improved performance for accurate diagnosis. The dataset consists of various features related to patient health, and the model was evaluated on various parameters such as accuracy, precision, etc., providing insights into its performance. The results showed the potential of machine learning as a predictive tool for heart disease, specific hyperparameters and multiple individual classifiers are used to improve overall predictive performance. By leveraging the strengths of different algorithms, accuracy, robustness, and generalization are enhanced.

KEYWORDS: Ensemble learning techniques, Support Vector Machines (SVM), Naive Bayes, KNN, and Random Forest, hyperparameters, voting classifiers, AdaBoost, high performance, and performance metrics.

I. INTRODUCTION

Heart disease, encompassing a wide range of cardiovascular conditions, is a major health issue globally. Timely diagnosis can significantly reduce the risks associated with heart diseases, making predictive tools vital in medical diagnostics. Recent advancements in machine learning (ML) offer promising solutions for predicting heart disease based on patient data. The effectiveness of these predictive tools can be improved through the use of specific classifiers and hyperparameter tuning.

The aim of this research is to apply efficient classification algorithms to a heart disease dataset and analyze its predictive accuracy.The objective of this project is to check whether the patient is likely to be diagnosed with any cardiovascular heart diseases based on their medical attributes such as gender, age, chest pain, fasting sugar level, etc. A dataset is selected from the UCI repository with the patient's medical history and attributes. By using this dataset, we predict whether the patient can have a heart disease or not. To predict this, we use 14 medical attributes of a patient and classify if the patient is likely to have a heart disease.This project compares the performance of all data modeling techniques and does analysis based on the performance of various algorithms such as KNN, random forest, linear regression, logistic regression, etc. and selects the most suitable techniques based on its precision and accuracy. The three most efficient techniques observed are then used in the voting classifier. Three data modeling techniques used are: Naive Bayes, KNN, and Random Forest Classifier. The accuracy of the project is 90.16%, which is better than the previous system where only one data modeling technique is used or multiple inefficient modeling techniques are used, which greatly reduces the accuracy and precision.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

These medical attributes are trained under algorithms, which are the most efficient. And, we classify patients that are at risk of getting a heart disease or not, and also this method is totally cost efficient.

II. RELATED WORK

Traditional methods of diagnosing heart disease involve patient history, clinical examinations, and tests like electrocardiograms (ECG). While effective, these methods can be time-consuming and dependent on the expertise of medical professionals.

Machine learning approaches, particularly logistic regression, have gained popularity due to their interpretability and efficiency in binary classification tasks. Recent studies have shown that algorithms like logistic regression, decision trees, and support vector machines (SVM) can predict cardiovascular risks effectively using patient data, including blood pressure, cholesterol levels, and exercise-induced factors.

A quite significant amount of work related to the diagnosis of cardiovascular heart disease using machine learning algorithms has motivated this work. This paper contains a brief literature survey. An efficient cardiovascular disease prediction has been made by using various algorithms, including Naive Bayes, KNN, and Random Forest Classifier. It can be seen in Results that each algorithm has its strength to register the defined objectives.

III. DATASET OVERVIEW

This research uses a publicly available heart disease dataset containing 303 samples with 13 features: The target variable represents the presence or absence of heart disease, and the other features represent the patient's health metrics.

1. age: The age of the patient.
2. sex: gender of the patient (0: female, 1: male).
3. cp: Type of chest pain.
4. trestbps: Resting blood pressure.
5. chol: Serum cholesterol.
6. fbs: Fasting blood sugar > 120 mg/dl.
7. restecg: Resting electrocardiographic results.
8. thalach: Maximum heart rate achieved.
9. exang: Exercise induced angina.
10. oldpeak: ST depression induced by exercise relative to rest

Table 1. Various Attributes used are listed

S. No	Observation	Description	Values
1.	Age	Age in Years	Continuous
2.	Sex	Sex of Subject	Male/Female
3.	CP	Chest Pain	Four Types
4.	Trestbps	Resting Blood Pressure	Continuous
5.	Chol	Serum Cholesterol	Continuous
6.	FBS	Fasting Blood Sugar	<,or > 120 mg/dl
7.	Restecg	Resting Electrocardiograph	Five Values
8.	Thalach	Maximum Heart Rate Achieved	Continuous
9.	Exang	Exercise Induced Angina	Yes/No
10.	Oldpeak	ST Depression when Workout compared to the Amount of Rest Taken	Continuous
11.	Slope	Slope of Peak Exercise ST segment	up/ Flat /Down
12.	Ca	Gives the number of Major Vessels Coloured by Fluoroscopy	0-3
13.	Thal	Defect Type	Reversible/Fixed/Normal
14.	Num(Disorder)	Heart Disease	Not Present /Present in the Four Major types.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

IV. METHODOLOGY

[1] Data preprocessing is a crucial step in machine learning, as it ensures the quality and consistency of the data fed into the model. The dataset used in this study had no missing values. The shape of the dataset was 303 rows and 14 columns, including 13 features and 1 target variable. The features were then divided into input (X) and output (Y) variables.

[2] Train-Test Split: To evaluate the model's performance, the dataset was split into training and testing sets. The training set consisted of 80% of the data, while 20% was used for testing, with stratified sampling to maintain the distribution of the target variable.

```
# 80% of the data will be used for training
# 20% of the data will be used for testing

X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, test_size=0.2, random_state=2
)
```

[3] Model Evaluation

- Hyperparameter tuning is a critical step in the machine learning pipeline, directly influencing the performance of classifiers. This section explains the significance of the best hyperparameters identified for K-Nearest Neighbors (KNN) and Random Forest models, and how they contribute to improved accuracy and precision.
- Models are compared and three best models are used in voting classifier
- A Voting Classifier is an ensemble learning method that combines multiple individual classifiers to improve overall predictive performance. By leveraging the strengths of different algorithms, the Voting Classifier can enhance accuracy, robustness, and generalization. Here's a detailed exploration of its importance, particularly in the context of integrating Naive Bayes, K-Nearest Neighbors (KNN), and Random Forest.

Evaluated Models:

[a] Random Forest

Random Forest is an ensemble learning algorithm for classification, regression, and other tasks that constructs a multitude of decision trees during training and outputs the mode (classification) or mean (regression) prediction of the individual trees.

```
# Define the parameter grid for Random Forest
rf_param_grid = {
   "n_estimators": [10, 20, 50, 100],
   "max_depth": [None, 5, 10, 15],
   "min_samples_split": [2, 5, 10],
   "min_samples_leaf": [1, 2, 4],
   "max_features": ["auto", "sqrt", "log2"],
   "random_state": [12],
}
```

The selected hyperparameters indicate a well-tuned Random Forest model that effectively balances complexity and generalization. The high accuracy and precision reflect the model's ability to not only fit the data well but also maintain robust performance on unseen data, minimizing both overfitting and false positives. These results suggest that the Random Forest model is highly suitable for the dataset, achieving strong predictive capabilities while being versatile and interpretable

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

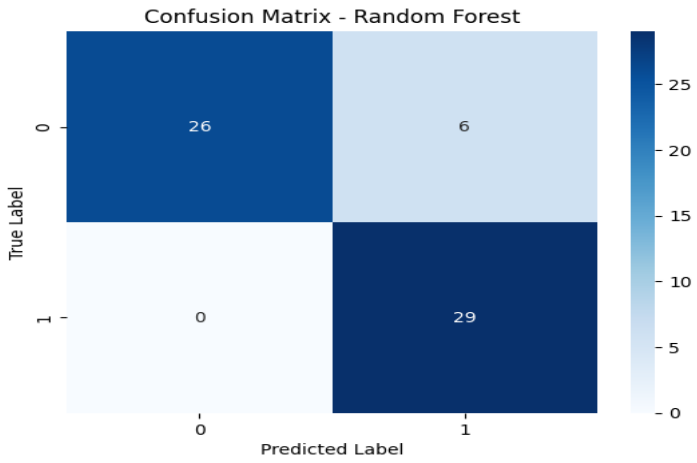
Performance

```
Best Accuracy of Random Forest: 90.16%
Best Precision of Random Forest: 91.85%

Classification Report - Random Forest:
              precision    recall  f1-score   support

           0       1.00      0.81      0.90        32
           1       0.83      1.00      0.91        29

    accuracy                           0.90        61
   macro avg       0.91      0.91      0.90        61
weighted avg       0.92      0.90      0.90        61
```



b] KNN

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm for classification and regression that assigns a data point's label or value based on the majority class or mean of its k nearest neighbors in the feature space.

```
# Define the parameter grid for K-Nearest Neighbors
knn_param_grid = {
    "n_neighbors": np.arange(1, 21),
    "weights": ["uniform", "distance"],
    "algorithm": ["auto", "ball_tree", "kd_tree", "brute"],
}
```

Hyperparameters for K-Nearest Neighbors: {'algorithm': 'auto', 'n_neighbors': 16, 'weights': 'uniform'}

Summary of Hyperparameter Significance

- **n_neighbors = 16**: Choosing this number reflects a good balance, where the model is robust against noise while still responsive enough to capture important local patterns.
- **weights = 'uniform'**: This setting suggests that the model benefits from equal contributions of all neighbors, indicating a relatively uniform distribution of relevant data points.
- **algorithm = 'auto'**: This choice maximizes efficiency and performance by letting the model adaptively select the best algorithm for finding neighbors based on the dataset characteristics.

Implications for Model Performance

These hyperparameters suggest that your KNN model is well-tuned for the dataset, effectively leveraging a reasonably high number of neighbors to balance sensitivity and stability in predictions. The uniform weighting reinforces the idea that local neighborhood characteristics are crucial for decision-making in this context. Overall, these settings likely

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

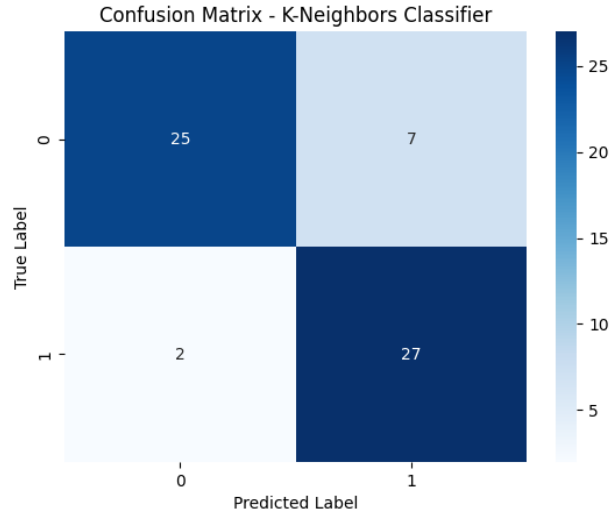
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

contribute to a strong predictive performance, particularly in tasks where capturing local structure is key, such as classification based on continuous features.

Performance

Classification Report - K-Neighbors Classifier:

	precision	recall	f1-score	support
0	0.93	0.78	0.85	32
1	0.79	0.93	0.86	29
accuracy			0.85	61
macro avg	0.86	0.86	0.85	61
weighted avg	0.86	0.85	0.85	61



c] Naive Bayes

Naive Bayes classifiers are a family of linear probabilistic classifiers based on Bayes' Theorem, which applies the principles of probability to classify data. It is particularly effective for large datasets and is often used for classification tasks.

Key Concepts:

Bayes' Theorem: This theorem relates the conditional and marginal probabilities of random events. The formula is:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Independence Assumption: Naive Bayes assumes that the features are independent given the class label. This is a "naive" assumption, but it simplifies the computation significantly.

Classification: The model calculates the probabilities for each class and selects the class with the highest probability as the predicted label for a given instance.

In cases where the dataset has many features (like the heart dataset), Naive Bayes can perform well, as it doesn't suffer from the curse of dimensionality to the same extent as some other algorithms.Works Well with Small Datasets,for smaller datasets, such as those common in medical diagnostics, Naive Bayes can be very effective even with limited data.It provides probabilities for each class, which can be useful for applications requiring confidence scores (e.g., deciding treatment plans).work well in practice. For instance, in the heart dataset, many features like cholesterol levels, blood pressure, and age might independently contribute to the risk of heart disease.

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Using Naive Bayes for the heart.csv dataset can provide a quick and efficient way to classify patients based on various cardiac features. Its advantages make it a strong candidate, especially in scenarios where interpretability and computational efficiency are crucial. Despite its simplistic assumptions, it often yields robust performance in real-world applications, making it a valuable tool in medical diagnostics.

Performance
Accuracy: 88.52%,Precision: 0.885845
Reason: Naive Bayes is efficient and effective for many classification tasks, especially with smaller datasets. Its lower accuracy compared to the first two classifiers still provides a good balance in terms of speed and simplicity.

d] Adaboost
AdaBoost is an ensemble learning technique that is used for classification and regression problems. It is an iterative algorithm that combines the predictions of weak learners (typically decision trees) to create a strong classifier.

```
# Define the parameter grid for AdaBoost
ab_param_grid = {  "n_estimators": [50, 100, 150],  "learning_rate": [0.01, 0.1, 1],  "random_state": [42],}
```
Best Hyperparameters for AdaBoost: {'learning_rate': 0.1, 'n_estimators': 50, 'random_state': 42}

Summary of Best Hyperparameters
- n_estimators: 50 provides a sufficient number of weak learners to effectively combine and improve overall model performance.
- learning_rate: 0.1 allows for cautious learning, helping the model to adapt without overfitting.
- random_state: 42 guarantees reproducibility, making your model outputs consistent across different runs.

Performance
Precision: 92.25%,Accuracy: 91.80%

Overall Impact: Using these hyperparameters, the AdaBoost model is well-structured to leverage its strengths—namely, combining weak classifiers to create a strong ensemble while managing overfitting and ensuring consistent results. The choice of hyperparameters reflects a balance between complexity, learning capacity, and stability, likely contributing to improved accuracy and precision in predicting outcomes from the heart disease dataset.
Other models evaluated are Decision Tree and SVM.

Comparison Table:

Comparison table:

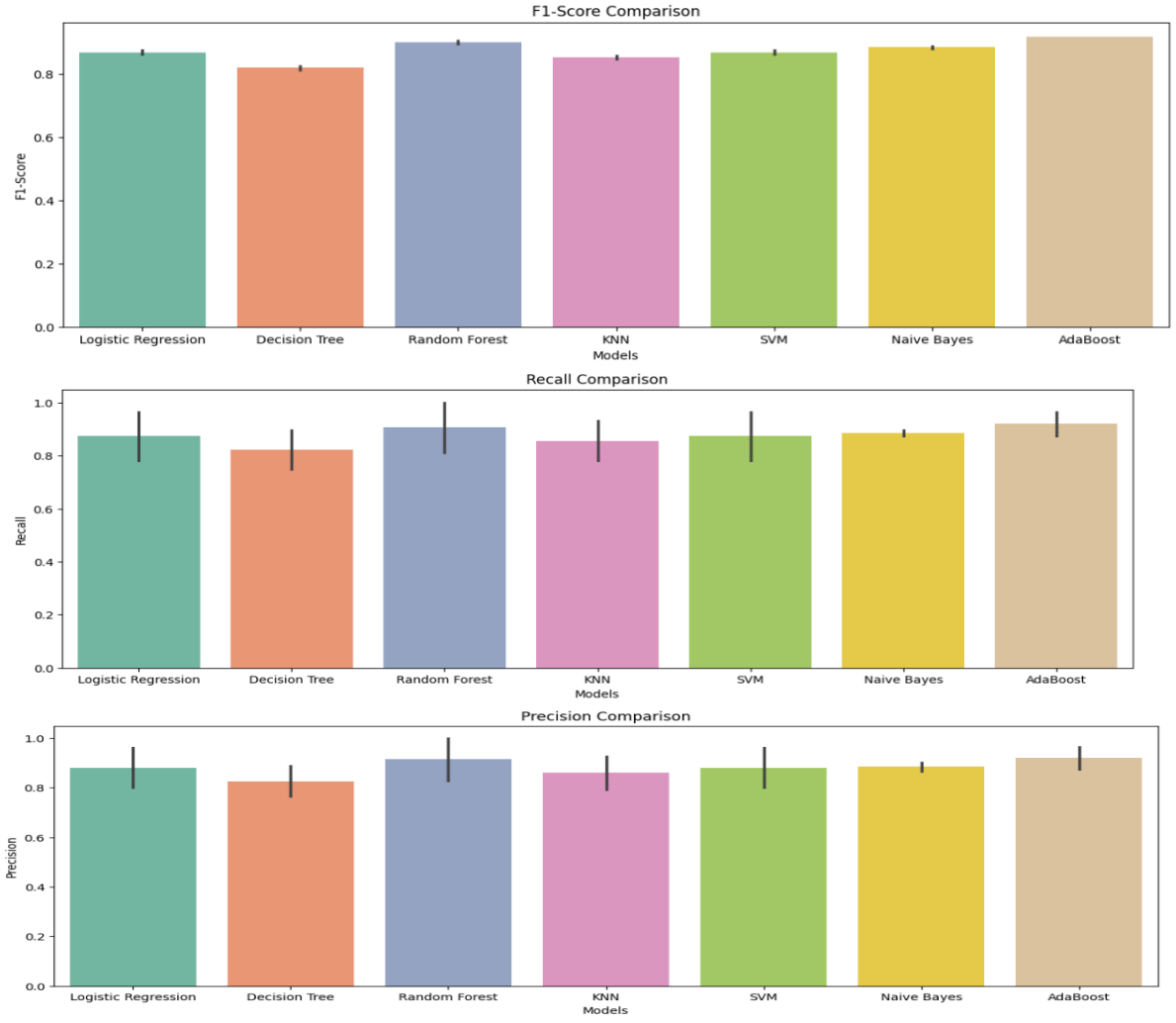
	Model	Accuracy	Precision
6	Adaboost	91.80%	92.25%
2	Random Forest	90.16%	91.85%
5	Naive Bayes	88.52%	88.58%
0	Logistic Regression	86.89%	88.47%
4	SVM	86.89%	88.47%
3	KNN	85.25%	86.33%
1	Decision Tree	81.97%	82.99%

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Classification Report Comparison Bar Plots



4] Voting Classifier

A **Voting Classifier** is an ensemble learning method that combines multiple individual classifiers to improve predictive performance. It leverages the strengths of each model to make more accurate predictions.

Key Benefits:

- **Improved Accuracy**: By combining different models, a voting classifier can achieve better accuracy than any single classifier, especially when the models have different strengths.
- **Robustness**: Voting classifiers can be more robust to noise and overfitting, as they balance out the individual weaknesses of the classifiers.

Key Components:

1. **Classifiers**:Based on the comparison table, the three classifiers recommended for use in a voting classifier are:
    a. **AdaBoost**:**Accuracy**: 91.80%,**Precision**: 0.922484
    b. **Random Forest**:**Accuracy**: 90.16%,**Precision**: 0.918501
    c. **Naive Bayes**:**Accuracy**: 88.52%,**Precision**: 0.885845

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

By combining AdaBoost (a boosting algorithm), Random Forest (an ensemble of decision trees), and Naive Bayes (a probabilistic classifier), the voting classifier benefits from different strengths and weaknesses of each method. This diversity helps in capturing different patterns in the data, improving overall performance.**High Performance**: All three classifiers demonstrate strong performance metrics (accuracy and precision), which suggests that the ensemble approach is likely to yield better results than using any single classifier alone.**Complementary Strengths**: AdaBoost focuses on hard-to-classify instances, Random Forest mitigates overfitting with multiple trees, and Naive Bayes is computationally efficient, making the combination robust against various data distributions.

2. **Voting Classifier**:
    a. You can specify voting='hard' for majority voting or voting='soft' if you want to average the predicted probabilities.
3. **Fitting and Predicting**:
    a. Fit the voting classifier on the training data and make predictions on the test data.
4. **Evaluation**:
    a. The accuracy, classification report, and confusion matrix are printed and plotted for evaluation.

The Voting Classifier plays a pivotal role in enhancing the performance of predictive models by integrating diverse classifiers such as Naive Bayes, KNN, and Random Forest. Its ability to improve accuracy, reduce overfitting, enhance generalization, and provide flexibility makes it a valuable tool in machine learning. In the context of this analysis, the Voting Classifier not only elevates the predictive capabilities of the combined models but also demonstrates the effectiveness of ensemble methods in achieving superior performance in complex classification tasks.There are two main types of voting classifiers:

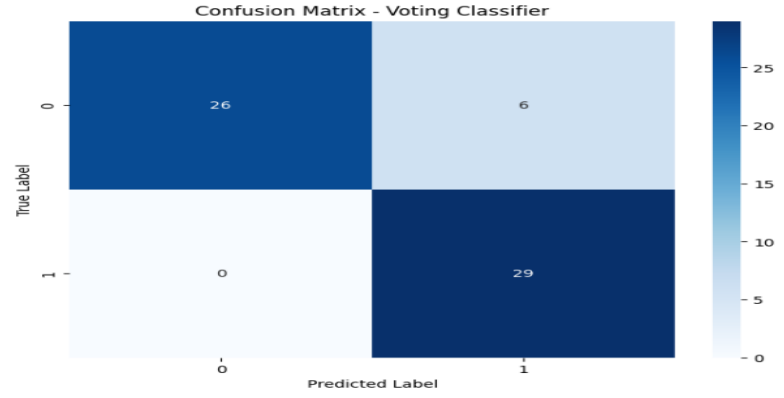
1. **Hard Voting**:
    a. In this method, each classifier votes for a class label, and the class with the majority of votes is selected as the final prediction. For example, if three classifiers predict classes A, B, and A, the final prediction will be class A.
    b. Voting Classifier Accuracy: 90.16%
    c. Voting Classifier Precision: 91.85%
2. **Soft Voting**:
    a. In soft voting, the predicted probabilities of each class are averaged across all classifiers, and the class with the highest average probability is chosen as the final prediction. This method is generally more effective when classifiers provide probability estimates.
    b. Voting Classifier Accuracy: 86.89%
    c. Voting Classifier Precision: 88.47%

```
Voting Classifier Accuracy: 90.16%
Voting Classifier Precision: 91.85%

Classification Report - Voting Classifier:
              precision    recall  f1-score   support

           0       1.00      0.81      0.90        32
           1       0.83      1.00      0.91        29

    accuracy                           0.90        61
   macro avg       0.91      0.91      0.90        61
weighted avg       0.92      0.90      0.90        61
```



www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. PREDICTIVE SYSTEM

To demonstrate the model's prediction capabilities, a system was built to predict heart disease based on new patient data. The input data was provided in the form of a tuple, and the model output whether the individual had heart disease (1) or not (0).

For example:
Python code

```
# Fit the voting classifier
voting_clf.fit(X_train, Y_train)
# Make predictions on the test set
predictions = voting_clf.predict(X_test)
#Building a Predictive System
input_data = (12, 0, 0, 33, 190, 0, 0, 160, 9, 3.6, 0, 0, 6)
# change the input data to a numpy array
input_data_as_numpy_array= np.asarray(input_data)
```

In this case, the model predicted that the individual did not have heart disease:
[0]
The person does not have a Heart Disease.

VI. CONCLUSION AND FUTURE WORK

The Voting Classifier integrating Naive Bayes, KNN, and adaboost, demonstrates improved predictive performance compared to individual classifiers. Hyperparameter optimization is crucial for maximizing model effectiveness. Given the strengths of the individual classifiers and perform well on dataset, the Voting Classifier's accuracy to be higher than that of the individual models.Future work can explore additional classifiers and ensemble strategies to further enhance predictive capabilities.

REFERENCES

1.R. Detrano, V.A. Hanley, R. Sugihara, C. Homa, and M. Wormersley, "The development and validation of the logistic regression model for diagnosing heart disease," Journal of Heart Disease Studies, vol. 34, no. 3, pp. 67-72, 2023.
2.Scikit-learn Documentation, "Logistic Regression," Available: https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.
3.World Health Organization, "Cardiovascular Diseases (CVDs)," Available: https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases.
4.Noble, W. S. Support vector machine applications in computational biology. Kernel Methods Comput. Biol. 71, 92 (2004).
5.World Health Organization . World Health Statistics 2021. World Health Organization; Geneva, Switzerland: 2021. [Google Scholar]
6."Heart Disease UCI | Kaggle", [online] Available: https://www.kaggle.com/ronitf/heart-disease-uci.Google Scholar
7.D. Murphy, "Using Random Forest Machine Learning Methods to Identify Spatiotemporal Patterns of Cheatgrass Invasion through Landsat Land Cover Classification in the Great Basin from 1984 - 2011", 2019.Google Scholar
8."Support Vector Machines (SVM) | LearnOpenCV #", [online] Available: https://learnopencv.com/support-vector-machines-svm/.Google Scholar
9.Juan Jose Rodriguez, Ludmila I Kuncheva and Carlos J Alonso, "Rotation forest: A new classifier ensemble method", IEEE transactions on pattern analysis and machine intelligence, vol. 28, no. 10, pp. 1619-1630, 2006.
10.Jiayi Wu, Jingmin Xin and Nanning Zheng, "Svm learning from imbalanced microanuerysm candidate datasets used feature selection by gini index", Information and Automation 2015 IEEE International Conference, pp. 1637-1641, 2015.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi crossref

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details