International Journal of Decision Inelligence
Vol 1, Issue 2, Spring 2023 ,65 - 81

Optimal Prediction in the Diagnosis of Existing Heart Diseases using Machine Learning: Outlier Data Strategies

Seyyed Amir Mahdi Ghoreishi Zadeh[a],Omid Rahmani[b,*], Mostafa Setak[c]

[a]*M.Sc. Student in Industrial Engineering Majoring In Macro Engineering systems, K. N. Toosi University*
[b]*M.Sc. Student in Industrial Engineering Majoring In Healthcare Systems, K. N. Toosi University*
[c]*Associate Professor, Department of Industrial Engineering, Economic and Social Systems, K. N. Toosi University*

Received 03 April 2023; Accepted 10 June 2023

Abstract

Heart disease is a prevalent and life-threatening condition that poses significant challenges to healthcare systems worldwide. Accurate and timely diagnosis of heart disease is crucial for effective treatment and patient management. In recent years, machine learning algorithms have emerged as powerful tools for predicting and identifying individuals at risk of heart disease. This article highlights the importance of heart disease diagnosis and explores the potential of machine learning algorithms in enhancing diagnostic accuracy. This article presents a study aimed at developing a model for predicting heart disease in Cleveland patients. The innovation of this research is that The research involved identifying and handling outlier data using Winsorized and Logarithmic transformation methods. We also used Wrapper and Embedded methods to determine the most important features for diagnosing heart disease. In addition to the usual features, Exercise-induced angina and No. of major vessels were found to be important. We then compared the performance of four machine learning algorithms, including KNN, Naïve Bayes' Classifier, Decision Tree, and Support Vector Classifier, to determine the best algorithm for predicting heart disease. The findings showed that the Decision Tree algorithm had the best performance with an accuracy of 97.95%.

Keywords: Heart Disease, Diagnosis, Machine Learning Algorithms, Winsorized and Logarithmic Transform Methods, Decision Tree, Support Vector Classification

1-Introduction

Heart disease holds a prominent position among the leading causes of mortality worldwide, making it a highly concerning ailment. It is responsible for a significant number of fatalities, as indicated by the World Health Organization data, which reported approximately 17.9 million deaths attributed to heart diseases in 2016 (tandfonline, n.d.). Heart diseases, including Coronary Heart Disease, Hypertension, and Stroke, claim the highest number of lives among all fatal causes of death in the United States. Specifically, Coronary Heart Disease alone accounts for approximately 1 out of every 7 deaths in the country, resulting in around 366,800 fatalities each year. The estimated incidence of heart attacks in the United States stands at roughly 7.9 million cases, representing about 3% of heart attack occurrences among American adults. Moreover, in 2015, the United States witnessed 114,023 individuals succumbing to heart attacks (Heart and Stroke Statistics, n.d.). Heart disease is influenced by numerous factors, including personal and professional habits as well as genetic predisposition. Engaging in habits like smoking, excessive alcohol, and caffeine consumption, experiencing high levels of stress, and leading a sedentary lifestyle contribute to the risk of heart disease. Physiological factors such as obesity, hypertension, elevated blood cholesterol levels, and pre-existing heart conditions further increase vulnerability to heart disease. Consequently, the timely and accurate diagnosis of heart disease is of utmost importance as it enables proactive measures to prevent fatalities and improve preventive interventions. Early diagnosis empowers healthcare professionals to promptly intervene, provide appropriate treatments, and implement lifestyle modifications. Furthermore, it facilitates the implementation of preventive strategies that can impede disease progression, enhance quality of life, and potentially extend lifespan. Efficient and accurate diagnostic methods, encompassing medical history assessments, physical examinations, diagnostic tests, and imaging techniques, play a pivotal role in identifying heart disease. Integrating machine learning algorithms with these diagnostic tools augments diagnostic accuracy, efficiency, and precision, thereby promoting better disease management and patient outcomes. Machine learning algorithms can analyze extensive patient data, uncover hidden patterns, and identify subtle risk factors that may not be readily apparent through conventional diagnostic approaches. By leveraging the capabilities of machine learning algorithms, healthcare professionals can enhance diagnostic accuracy by considering multiple factors simultaneously and developing predictive models that facilitate early detection and intervention. In conclusion, efficient and accurate medical diagnosis is paramount in the prevention of heart disease-related deaths. By incorporating machine learning algorithms into the diagnostic process, healthcare professionals can improve their ability to identify individuals at risk, tailor preventive measures accordingly, and ultimately achieve improved patient outcomes while reducing the burden of heart disease on society. Consequently, there exists an

* Corresponding Author. Email: rahmanniomid@gmail.com

ongoing and pressing requirement for a precise and reliable system capable of analyzing medical data to uncover concealed patterns related to heart diseases and predict occurrences of heart attacks proactively. The ultimate goal is to enhance the management and prevention of heart attacks for improved outcomes. Data mining involves extracting relevant information from extensive datasets across diverse domains, including medicine, business, and education. Machine learning, a rapidly advancing area of artificial intelligence, plays a pivotal role in analyzing vast amounts of data from various fields. In the medical field, machine learning algorithms provide an alternative to conventional prediction modeling approaches, utilizing computers to comprehend intricate and nonlinear relationships among different factors. By reducing errors in predicted outcomes and aligning them with actual results, machine learning enables a deeper understanding of complex interactions (Weng et.al.,2017).

2-Literature Review

Heart diseases are a leading cause of mortality, and researchers have explored the use of predictive models to identify indicators of heart disease based on health data. Recent advancements in machine learning have spurred research into the development of methods and techniques for diagnosing heart disease. Various techniques, including classification and clustering, have been examined to address this problem.

In 2020 Ahmed, H et al. proposed a system that m aims to identify the most effective machine learning algorithm for accurately predicting heart disease. To achieve this, two feature selection algorithms, namely univariate feature selection, and Relief, are utilized to select important features from the dataset. Four machine learning algorithms, namely Decision Tree, Support Vector Machine, Random Forest Classifier, and Logistic Regression Classified are compared using the selected features and the full set of features. Hyperparameter tuning and cross-validation techniques are applied to improve accuracy. An advantage of the proposed system is its efficient handling of Twitter data streams that contain patient data, accomplished by integrating Apache Kafka with Apache Spark as the underlying infrastructure. The experimental results show that the random forest classifier outperforms the other models, achieving the highest accuracy of 94.9% (Ahmed et.al.,2020).

In 2020 Shah, D et al. examined the attributes related to heart disease and a supervised learning model was developed using Naïve Bayes, Decision Tree, K-Nearest Neighbor, and Random Forest algorithms. The dataset used in the study includes 303 instances and 76 attributes from the Cleveland database of the UCI repository of heart disease patients. Out of these 76 attributes, only 14 are considered for testing to evaluate the performance of different algorithms. The main objective of the study is to predict the likelihood of heart disease in patients. The results indicate that the K-nearest neighbor algorithm achieves the highest accuracy score (Shah et.al.,2020).

Developing accurate and efficient methods for the early prediction of heart diseases through machine learning and data mining methods is challenging, but it can provide valuable support for clinical decision-making with digital patient records. In many countries, there is a lack of cardiovascular expertise, leading to high rates of misdiagnosis. In 2021 Ali, M.M. et al. aimed at identifying the most accurate machine learning classifiers for predicting heart disease. Several supervised machine learning algorithms were compared in terms of performance and accuracy with feature importance scores for each feature, except for MLP and KNN. This study showed that the Random Forests (RF) algorithm achieved 100% accuracy, sensitivity, and specificity when applied to a heart disease dataset collected from Kaggle. This shows that a simple supervised machine learning algorithm can effectively predict heart disease with high accuracy and potential clinical application (Ali et.al.,2021).

In 2022 Riyaz, L et al. provided a comprehensive survey of various machine learning techniques and evaluated their performance for efficient prediction, diagnosis, and treatment of heart diseases. The study examines several machine learning techniques, including support vector machine (SVM), decision tree (DT), Naïve Bayes (NB), K-nearest neighbor (KNN), and artificial neural network (ANN), used for predicting heart disease occurrence. The average prediction accuracy of each technique is calculated to determine the overall best and worst-performing techniques. The results show that ANN achieved the highest average prediction accuracy of 86.91%, while the C4.5 decision tree technique had the lowest average prediction accuracy of 74.0% (Ayob, 2022).

Telediagnostic equipment has become increasingly important for monitoring heart disease in today's world. Early diagnosis of heart disease is crucial for effective treatment and quick recommendations from clinical experts. However, feature extraction poses a significant challenge for heart disease prediction, as high-dimensional data leads to longer learning times for existing machine learning classifiers. In 2022 Shekar, J et al. proposed a novel and efficient Internet of Things-based classifier called the tuned adaptive neuro-fuzzy inference system (TANFIS) for accurate heart disease prediction. The TANFIS tuning parameters are optimized using the Laplace Gaussian mutation-based moth flame optimization and grasshopper optimization algorithm. The proposed method is evaluated using 11 datasets from the UCI repository and achieves an impressive 99.76% accuracy for heart disease prediction, which is an improvement of 5.4% compared to existing algorithms (Sekar et.al.,2022).

In 2021 Katarya et al. discussed heart disease and its risk factors and explored various machine learning techniques for predicting heart disease. A comparative analysis of the

machine learning algorithms used in the experiment is provided. The research objective is focused on predicting heart disease using machine learning techniques and analyzing their performance (Katarya & Meena,2021).As medical datasets increase in size, it has become difficult for clinicians to understand the complex relationships of features and accurately predict disease. In 2021 Pathan, M et al. In an article with the aim of research, they presented the identification of the most important risk factors in a multi-dimensional data set for the possibility of accurate classification of heart disease with fewer complications. To achieve this goal, two heart disease datasets with different medical characteristics were analyzed for correlation and interdependence of different characteristics in the context of heart disease. A filter-based feature selection technique was then applied to select the most relevant features, resulting in optimal feature subset reduction for heart disease detection. Different machine learning classification models were tested using full and reduced feature subsets as input, and the trained classifiers were evaluated based on accuracy, receiver operating characteristic (ROC) curve, and F1 score. The results showed that the relevant features have a high impact on the classification accuracy, and even with the reduction of the number of features, the performance of the classification models was significantly improved by reducing the training time compared to the models trained on the full feature set (Pathan et.al.,2022).

In 2022 El-Shafiey, M. G. et al. proposed a novel hybrid genetic algorithm (GA) and particle swarm optimization (PSO) optimized approach called GAPSO-RF, which uses random forest (RF) to select the most optimal features for increasing heart disease prediction accuracy. The proposed approach utilizes multivariate statistical analysis in the first step to select the most significant features used in the initial population. Then, a discriminate mutation strategy is employed in GA, and GAPSO-RF combines a modified GA for global search and a PSO for local search. Additionally, PSO incorporates the concept of rehabilitating individuals who were rejected in the selection process. The performance of the GAPSO-RF approach is evaluated using two datasets from the University of California, Cleveland, and Statlog, using evaluation metrics such as accuracy, specificity, sensitivity, and area under the receiver operating characteristic (ROC) curve. The experimental results demonstrate that the GAPSO-RF approach achieved high heart disease prediction accuracies of 95.6% and 91.4% on the Cleveland and Statlog datasets, outperforming other state-of-the-art prediction methods (El-Shafiey et.al.,2022).

Early prediction of heart disease (HD) risks is crucial for prevention and treatment, but current clinical procedures for diagnosing HD are expensive and often require expert intervention. To address this issue, researchers have developed various intelligent systems for automated HD diagnosis, with artificial neural networks (ANNs) being a popular choice due to their promising prediction results. However, no research has explored the use of ANNs for feature extraction, which represents a gap in the research that needs to be addressed to improve predictions. This study proposes a new approach for HD prediction that uses a pre-trained Deep Neural Network (DNN) for feature extraction, Principal Component Analysis (PCA) for dimensionality reduction, and Logistic Regression (LR) for prediction. The efficacy of the proposed approach (DNN + PCA + LR) was investigated using the Cleveland HD dataset, with experimental results showing that it performs well on both training and testing data, achieving accuracy rates of 91.79% and 93.33%, respectively. Furthermore, the proposed approach outperformed state-of-the-art approaches in most evaluation metrics (Hassan et.al.,2023).They discussed the interdisciplinary field of data mining for healthcare, which includes the use of machine learning and data visualization to investigate the effectiveness of medical treatments, in the research of Arumugam, K. et al. In 2023, there is a special focus on diabetes-related heart disease, a type of heart disease that affects people with diabetes. While there are various data mining classification algorithms to predict heart disease, there is not enough data to predict heart disease in diabetics. The decision tree model is known to perform better than simple Bayes and support vector machine models, and they recommend tuning the decision tree model for optimal performance in predicting the probability of heart disease in diabetic subjects. Therefore, it is recommended to use a decision tree model to predict the probability of heart disease in people with diabetes and fine-tune it for optimal performance (Arumugam et.al.,2023).

In 2023 Ozcan, M., & Peker, S. utilized the Classification and Regression Tree (CART) algorithm, a supervised machine learning method, to predict heart disease and extract decision rules that clarify the relationships between input and output variables. also ranks the features that influence heart disease based on their importance. The model's accuracy of 87% validates its reliability when considering all performance parameters. Additionally, the extracted decision rules simplify the use of clinical purposes, making them accessible to healthcare professionals and patients who may face time and cost constraints in the diagnosis and treatment of heart disease. Overall, the proposed algorithm can potentially support healthcare professionals and patients in diagnosing and treating heart disease (Ozcan & Peker,2023).In 2023 Dileep, P. et al. investigated the effectiveness of deep learning techniques compared to traditional methods in predicting heart disease using the UCI heart disease dataset and a real-time dataset. To enhance the accuracy of traditional methods, a cluster-based bi-directional long-short-term memory (C-BiLSTM) approach is proposed. The experimental results are obtained by inputting both datasets through the K-Means clustering algorithm to remove duplicate data, and then the heart disease is predicted using the C-BiLSTM approach. The performance of the C-BiLSTM approach is compared with conventional classifier methods such as Regression Tree, SVM, Logistic

Regression, KNN, Gated Recurrent Unit, and Ensemble. The efficiency of the system is evaluated based on accuracy, sensitivity, and F1 score. The results show that the C-BiLSTM approach outperforms the six conventional methods, achieving 94.78% accuracy for the UCI dataset and 92.84% accuracy for the real-time dataset, providing better prediction of heart disease (Dileep et.al.,2023).

3-Approach Methodology

This study aims to analyze various machine learning algorithms to identify the most suitable model for the given dataset for heart disease probability, which can be beneficial for doctors and patients in the medical field. Various machine learning algorithms have been used on the dataset, and the research paper presents the data analysis. Additionally, the paper highlights the features that have a greater impact on accuracy, which can potentially reduce the need for multiple tests and save costs for patients.

3-1. Data Source

The primary objective of this research is to create a model that can attain a high level of accuracy by utilizing the Cleveland heart disease dataset from 2016 (Cleveland et.al.,2019). The Cleveland heart disease dataset from 2016 contains 13 independent variables or features, as well as one dependent variable or class label that is utilized to predict the presence of heart disease. Table.1 provides complete information about these features. The target label has five values, with a value of 0 indicating the absence of heart disease and values of 1, 2, 3, and 4 indicating the presence of different heart problems. However, in this study, the focus is on binary classification, distinguishing between individuals with heart disease (represented by a value of 1) and those without (represented by a value of 0). Therefore, values of 2, 3, 4, and 5 have been replaced with 1 because they indicate the presence of heart disease.

Table 1
Features information and description of Cleveland heart disease dataset

Sr. no.	Attribute	Representative icon	Details
1	Age	AGE	Patient's age, in years
2	Sex	SEX	0=female; 1 =male
3	Chest pain	CPT	4 types of chest pain (1_typical angina; 2_atypical angina; 3_non-anginal pain; 4_asymptomatic)
4	Rest blood pressure	RBP	Resting systolic blood pressure (in mm Hg on admission to the hospital)
5	Serum cholester	SCH	Serum cholesterol in mg/dl
6	Fasting blood su	FBS	Fasting blood sugar>120 mg/dl (0_false; 1_true)
7	Rest electrocardiogra	RES	0_normal; 1_having ST-T wave abnormality; 2_left ventricular hypertrophy
8	MaxHeart rate	MHR	Maximum heart rate achieved
9	Exercise-induce angina	EIA	Exercise-induced angina (0_no; 1_yes)
10	ST depression	OPK	ST depression induced by exercise relative to rest
11	Slope	PES	The slope of the peak exercise ST segment (1_upsloping; 2_flat; downsloping)
12	No. of vessels	VCA	No. of major vessels (0–3) colored by fluoroscopy
13	Thalassemia	THE	Defect types; 3_normal; 6_fixed defect; 7_reversible defect
14	Target (Class)	target	0 = no disease and 1 = disease

According to the original data, plotting a correlation heatmap (Figure 1) would show the degree and direction of the correlation between each pair of variables in the dataset. The heatmap would display a color-coded matrix with each cell representing the correlation coefficient between two variables. This visualization would help to identify which variables are strongly related to each other and which variables have weak or no relationship. Additionally, it could help to identify any patterns or trends in the data, such as multicollinearity or the presence of confounding variables. Ultimately, the correlation heatmap would provide valuable insights into the relationships between variables in the dataset and help guide subsequent data analysis and modeling efforts.

3-2- Data Preprocessing

Real-world information often contains a significant amount of missing or noisy data, which can hinder accurate predictions. To address these issues and improve prediction accuracy, the data must undergo preprocessing. Figure 2 illustrates the sequential chart of the proposed model used in this study.

It is common for collected data to contain outliers and missing values, which can lead to inaccurate and ineffective results. To achieve accurate and effective results, it is necessary to clean the data by removing outliers and filling in any missing values.

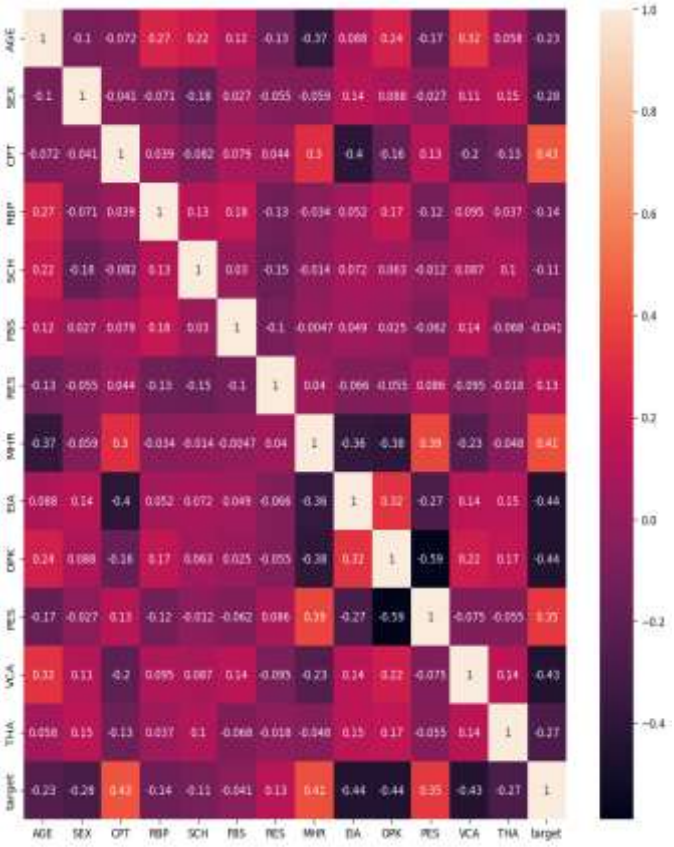

Fig.1. A Correlation Matrix

Winsorization: is a method of handling outlier data that involves replacing extreme values with less extreme values to reduce the impact of outliers on statistical analysis. This method is named after Charles P. Winsor, an economist who developed this technique in 1940. The Winsorization process involves identifying extreme values in a dataset and then replacing them with trimmed or modified values that are closer to the median or mean of the data. For example, if a dataset has extreme values in the upper tail, the Winsorization process will replace those values with the maximum value that is within a certain range or percentile, rather than simply removing them from the analysis altogether.

Similarly, if there are extreme values in the lower tail, they will be replaced with the minimum value within a certain range or percentile. Winsorization is considered a robust method of handling outlier data because it preserves the overall distribution of the data while minimizing the impact of outliers. However, it should be used with caution and only after careful consideration of the data and the research question. Additionally, the specific method of Winsorization employed may vary depending on the nature of the data and the research question.

Transformation involves converting data from one format to another to enhance its comprehensibility. The data is then categorized and divided into training and test sets, which are utilized to run several algorithms and achieve accuracy score results.

Logarithmic transformations: can also be used in machine learning algorithms for handling or transforming outlier data. In some cases, outliers can have a significant impact on the performance of a machine learning algorithm, leading to poor accuracy or overfitting. Logarithmic transformations can help to reduce the impact of outliers by compressing the data and making extreme values less influential. When using logarithmic transformations to handle or transform outlier data, it is important to choose an appropriate base for the logarithm. Some bases, such as the natural logarithm (base e), can be more effective at reducing the impact of outliers than others. Additionally, it is important to evaluate the performance of the algorithm both before and after the transformation to ensure that it is improving the accuracy of the model. However, it is important to note that logarithmic transformations may not always be appropriate for handling or transforming outlier data, and the choice of transformation should be based on the nature of the data and the research question.

Based on the primary data, we used boxplot diagrams to visualize the distribution of all the features in the dataset. This allowed us to identify any outliers that may be present and develop appropriate strategies for handling them. Two common methods for handling outliers that we employed were Winsorization and logarithmic transformations. After applying these methods, we re-evaluated the distribution of the data to ensure that the outliers had been effectively handled. This preprocessing step is important in ensuring the accuracy and reliability of any subsequent analysis or modeling.

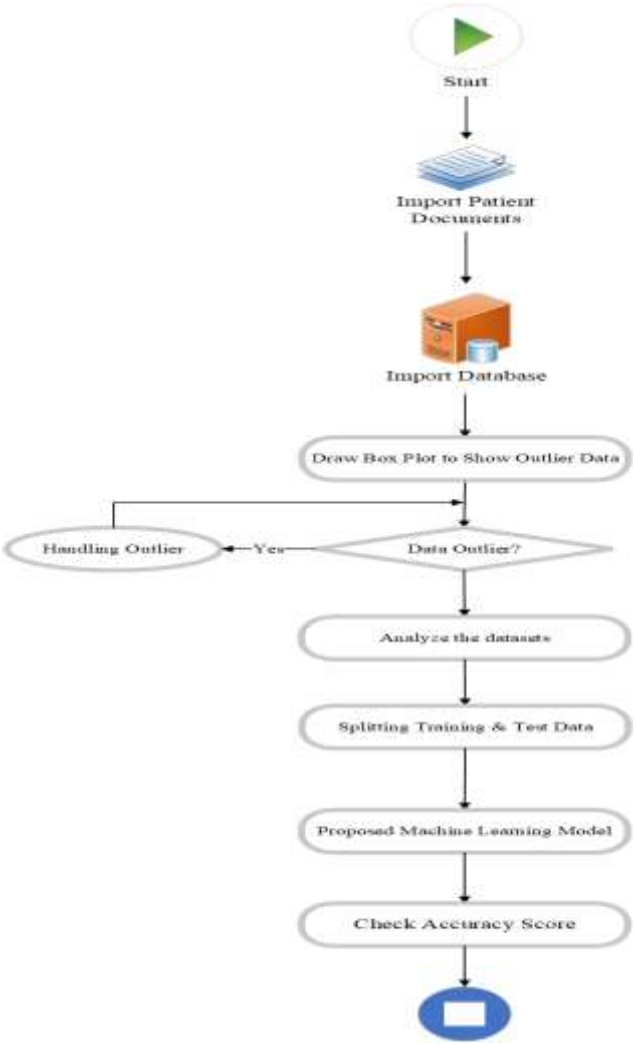

Fig. 2. Sequential Chart of the Proposed Model

Based on the box plot diagrams shown in Figures 3 and 4, it is evident that several features in the dataset contain outlier data. Specifically, the features RBF, SCH, MHR, FBS, OPK, VCA, and THA all appear to have at least one outlier value that is significantly different from the rest of the data. Outliers can have a significant impact on statistical analysis and modeling, as they can skew the results and lead to inaccurate conclusions. Therefore, it is important to carefully handle and preprocess outliers before performing any further analysis or modeling. Techniques such as Winsorization or logarithmic transformations can be used to handle outliers and improve the accuracy and reliability of subsequent analyses.

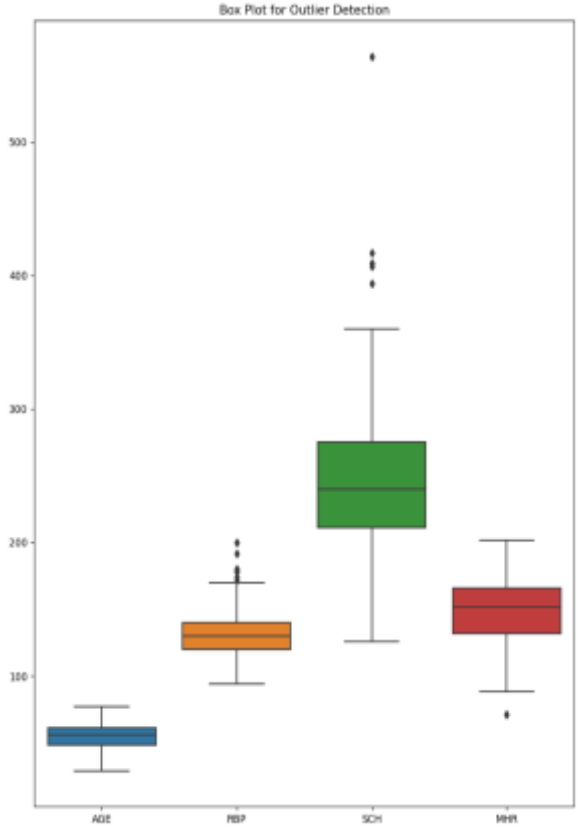

Fig.3. Box Plot for Outlier Detection

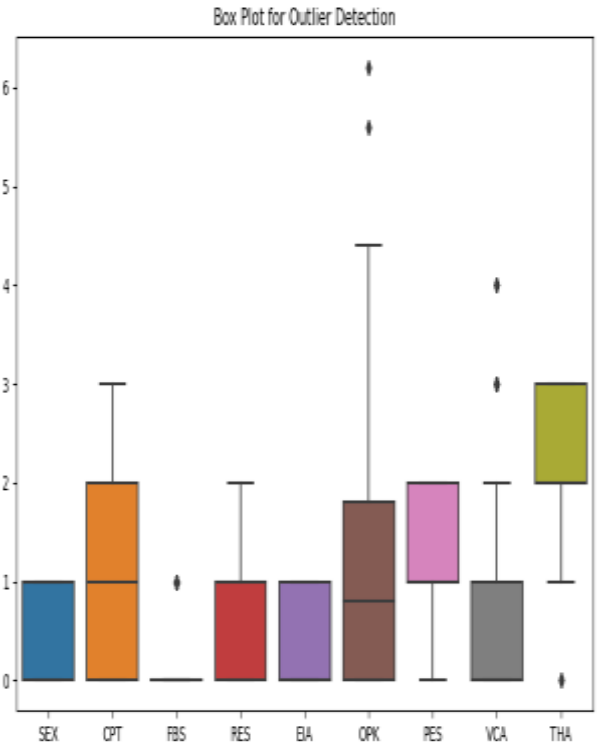

Fig.4. Box Plot for Outlier Detection

To handle the outlier data present in the dataset, we employed two different algorithms Winsorization and a logarithmic transformation. After applying these algorithms, we compared the statistical information obtained from three different datasets: the original dataset(Table.2), the dataset obtained from applying the Winsorization algorithm(Table.3), and the dataset obtained from applying the logarithmic transformation algorithm (Table.4). Based on this analysis, we selected the best dataset after applying the logarithmic transformation algorithm to use in building our prediction model because logarithmic transformations are used to address issues related to skewed distributions or heteroscedasticity. Taking the logarithm of a variable can help normalize its distribution and reduce the influence of extreme values. This transformation is often applied when the data exhibits exponential growth or decay patterns.

This dataset had been preprocessed using the most effective algorithm for handling outliers and provided the most accurate and reliable results. Using this dataset as a basis, we were able to build a robust and effective prediction model that could be used to make accurate predictions about the target variable.

Table 2
Original Dataset Summary Statistics

	RBP	SCH	MHR	FBS	OPK	VCA	THA
Count	1025	1025	1025	1025	1025	1025	1025
Mean	131.611	246	149.114	0.1492	1.0715	0.754	2.323
Std	17.5167	51.59	23.005	0.3565	1.1750	1.0307	0.620
Min	94	126	71	0	0	0	0
25%	120	211	132	0	0	0	2
50%	130	240	152	0	0	0	2
75%	140	275	166	0	1.8	1	3
Max	200	564	202	1	6.2	4	3

Table 3
Winsorized Dataset Summary Statistics

	RBP	SCH	MHR	FBS	OPK	VCA	THA
Count	1025	1025	1025	1025	1025	1025	1025
Mean	131.297	244.77	149.363	0.149	1.034	0.7365	2.3307
Std	15.4053	43.871	21.293	0.356	1.072	0.982	0.6001
Min	108	175	108	0	0	0	1
25%	120	211	132	0	0	0	2
50%	130	240	152	0	0.8	0	2
75%	140	275	166	0	1.8	1	3
Max	164	330	182	1	3.4	3	3

Table 4
Log-Transformed Dataset Summary Statistics

	RBP	SCH	MHR	FBS	OPK	VCA	THA
Count	1025	1025	1025	1025	1025	1025	1025
Mean	4.8790	5.4886	4.9985	0.1034	0.5841	0.4180	1.1808
Std	0.1288	0.2027	0.1647	0.2471	0.5278	0.5138	0.2118
Min	4.5538	4.8441	4.2766	0	0	0	0
25%	4.7957	5.3565	4.8934	0	0	0	1.0986
50%	4.8751	5.4847	5.0304	0	0.5877	0	1.0986
75%	4.948	5.6204	5.1179	0	1.0296	0.6931	1.3862
Max	5.3033	6.3368	5.3132	0.9631	1.9740	1.6094	1.3862

We have researched the dataset and concluded that all the features in it are important and require analysis. Therefore, we have decided not to use any dimensionality reduction techniques such as PCA and will use the transformed dataset to develop a predictive learning model. In the following, we only use feature selection to show which of these features is important in our dataset, otherwise, we did not apply any of them in the use of machine learning algorithms.

3-3- Feature Selection

To determine the importance of features in our dataset, we will use two feature selection methods: wrapper methods and embedded methods. We will apply these methods to both the Winsorized Dataset and Log-Transformed Dataset, and compare the results.

*Wrapper methods:*are a type of feature selection technique in machine learning that involves selecting a subset of features based on the performance of a machine learning algorithm. This method evaluates different subsets of features by training and testing the model on each subset. The basic idea behind wrapper methods is to use a machine learning algorithm as a black box to evaluate the utility of different subsets of features. The algorithm is trained on a subset of features and tested on a validation set to measure its performance. This process is repeated for different subsets of features until the optimal subset is found. The advantage of wrapper methods is that they can take into account the interactions between features. For example, two features that are individually weak predictors may be strong predictors when used together. By evaluating different subsets of features, wrapper methods can identify these interactions and select the most informative subset of features. wrapper methods can be computationally expensive since they involve training and testing a machine learning algorithm for each subset of features. Also, wrapper methods can be prone to overfitting, especially when the number of features is large. Some popular wrapper methods include recursive feature elimination (RFE), forward selection, and backward elimination. Each method has its strengths and weaknesses and the choice of method depends on the specific problem and the characteristics of the data.

*Embedded methods:*are a type of feature selection technique in machine learning that incorporates feature selection into the training process of a machine learning algorithm. These methods select the most important features during model training. The basic idea behind embedded methods is to use a machine learning algorithm that has built-in feature selection mechanisms. These algorithms select the most informative features during the training process, which can improve the accuracy of the model and reduce overfitting. One popular example of an embedded method is regularization, which involves adding a penalty term to the loss function of a machine learning algorithm that encourages the model to select fewer features. Regularization is commonly used in linear regression and logistic regression algorithms. Another example of an embedded method is decision tree-based algorithms, such as random forests and gradient boosting machines, which use feature importance scores to determine the most informative features. These algorithms evaluate the importance of each feature based on its contribution to the reduction of impurity in the splits of the decision trees. The advantage of embedded methods is that they can be less computationally expensive than wrapper methods since they select the most informative features during the model training process. Additionally, since the feature selection is embedded in the training process, embedded methods can be less prone to overfitting compared to wrapper methods. embedded methods may not consider all possible subsets of features and may not be able to capture interactions between features as well as wrapper methods. The choice of embedded method depends on the specific problem and the characteristics of the data.

Based on the results presented in Table 5, it is evident that the Wrapper method was applied to two datasets, the Winsorized Dataset and the Log-Transformed Dataset, and the important features were identified. This information can be used to refine the prediction model and improve its accuracy by reducing the number of features used in the model. By selecting only the most 5 important features, we can reduce the computational complexity of the model and potentially improve its generalizability to new data. the results of the Wrapper method provide valuable insights into the importance of each feature in the dataset and can be used to optimize the prediction model for better performance.

Table 5
Selected Feature by Wrapper Method

	Selected Feature Indices				
	Selected Feature Names				
Winsorized Dataset	2	9	12	13	14
	SEX	EIA	VCA	THE	target
log_transformed Dataset	2	9	10	12	14
	SEX	EIA	OPK	VCA	target

Table 6 presents the results of the embedded method applied to two datasets, the Winsorized Dataset and the Log-Transformed Dataset. The embedded method was used to identify the feature importance scores for each feature in the dataset. The results of the embedded method provide insights into which features are the most important for the prediction model. This information can be used to refine the model and improve its accuracy by reducing the number of features used in the model. By selecting only the most important features, we can reduce the model's computational complexity and improve its generalizability to new data. The feature importance scores indicated in Table 6 provide a quantitative measure of each feature's importance, allowing us to prioritize which features to include in the model.

Table 6
Feature Importance Scores

Feature	Winsorized Dataset	log_trans Dataset
AGE	- 0.0	-0.0040571482842
SEX	- 0.0	-0.0
CPT	0.0	0.0
RBP	-0.00101997949450818	-0.0
SCH	-0.00053853598694008	-0.0
FBS	- 0.0	-0.0
RES	0.0	0.0
MHR	0.00464603581067881	0.0
EIA	- 0.0	-0.0
OPK	- 0.0	-0.0
PES	0.0	0.0
VCA	- 0.0	-0.0
THE	- 0.0	-0.0

Feature importance scores are a measure of the relative importance of each feature in a machine learning algorithm. In embedded methods for feature selection, feature importance scores are used to identify the most informative features in the dataset. The feature importance scores are calculated during the training process of the machine learning algorithm and are based on how much each feature contributes to the model's performance. The scores can be calculated using different methods, such as the Gini importance or the permutation importance. The importance scores allow us to identify which features have the most significant impact on the prediction model's accuracy. By analyzing the importance scores, we can determine which features are the most informative and prioritize them for inclusion in the model. Feature importance scores can be useful for several reasons.

Firstly, they can help to identify which features are most relevant to the problem we are trying to solve. This can help to reduce the dimensionality of the dataset and improve the model's performance by focusing on the most informative features. Secondly, feature importance scores can provide insights into the underlying relationships between features in the dataset. For example, if two features have high importance scores, it may indicate that they are highly correlated or that they interact in a complex way. Feature importance scores provide a way to identify the most informative features in a dataset and can be used to optimize the performance of a machine learning algorithm. They can also provide insights into the underlying relationships between features, helping us to better understand the problem we are trying to solve.

4-Machine Learning Algorithm Used

4-1-K-Means

K-means: is a popular unsupervised machine learning algorithm used for clustering data points into groups based on their similarity. The algorithm seeks to find a predetermined number of clusters in a dataset by minimizing the sum of squared distances between each data point and its assigned cluster centroid. K-means is a simple and efficient algorithm, but it has some limitations. One of the main limitations is the sensitivity to the initial random centroid positions, which can lead to different results on different runs. Therefore, it's important to run the algorithm multiple times with different initializations to ensure that optimal clustering is achieved. Additionally, K-means assumes that the clusters are spherical and equally sized, which may not be the case in some datasets. The paper discusses the application of the K-means algorithm to a Winsorized dataset. As previously mentioned, the main objective of the K-means algorithm is to group data points into clusters based on their similarity. This is achieved by minimizing the sum of the squared distances between each data point and its assigned cluster centroid.

Figure 5 in the paper illustrates the relationship between the number of clusters (K) and the sum of the squared distances. The graph shows that as the number of clusters increases, the sum of the squared distances decreases. This implies that the algorithm can identify more distinct patterns in the data as the number of clusters increases. However, there is a point of diminishing returns where the reduction in the sum of squared distances becomes less significant as K increases. Therefore, the optimal number of clusters should be chosen based on a trade-off between the reduction in the sum of squared distances and the complexity of the resulting clusters.

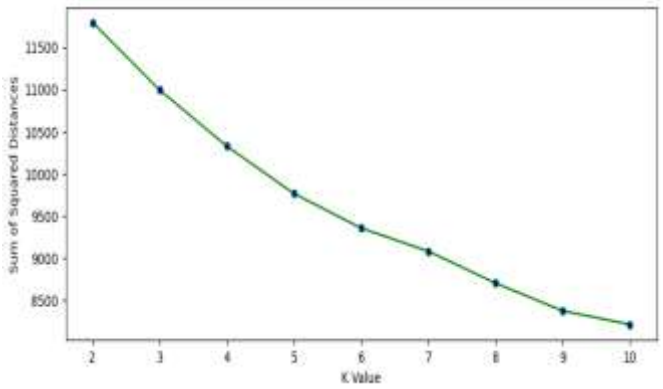

Fig. 5. Squared Distances / K Value

4-2-Hierarchical Clustering

Hierarchical clustering: is a popular unsupervised machine learning algorithm used for clustering data points into groups based on their similarity. The algorithm builds a hierarchy of clusters by iteratively merging or splitting clusters until a desired number of clusters is achieved. There are two types of hierarchical clustering:

1. Agglomerative clustering: This is a bottom-up approach where each data point is initially considered as a separate cluster. The algorithm then iteratively merges the closest

pairs of clusters until all data points belong to a single cluster.

2. Divisive clustering: This is a top-down approach where all data points are initially considered as a single cluster. The algorithm then iteratively splits the clusters into smaller clusters until the desired number of clusters is achieved.

The distance between clusters is typically calculated using a linkage criterion, which determines the distance between two clusters based on the distances between their data points. The most commonly used linkage criteria are:

1. Single linkage: The distance between two clusters is defined as the shortest distance between any two data points in the two clusters.

2. Complete linkage: The distance between two clusters is defined as the longest distance between any two data points in the two clusters.

3. Average linkage: The distance between two clusters is defined as the average distance between all pairs of data points in the two clusters.

Hierarchical clustering results in a dendrogram, a tree-like diagram that illustrates the hierarchy of clusters. The dendrogram can be cut at a certain height to obtain the desired number of clusters. The advantage of hierarchical clustering is that it can identify clusters at different levels of granularity, while the disadvantage is that it can be computationally expensive for large datasets.

The article discusses the use of the Hierarchical Clustering algorithm on a Winsorized dataset. The algorithm creates a hierarchy of clusters by merging or splitting clusters until a desired number of clusters is achieved.

Figure 6 in the article shows the dendrogram produced by the Hierarchical Clustering algorithm on the Winsorized dataset. The dendrogram illustrates the hierarchy of clusters and can be cut at a certain height to obtain the desired number of clusters. The advantage of Hierarchical Clustering is that it can identify clusters at different levels of granularity. However, it can be computationally expensive for large datasets due to its iterative nature.

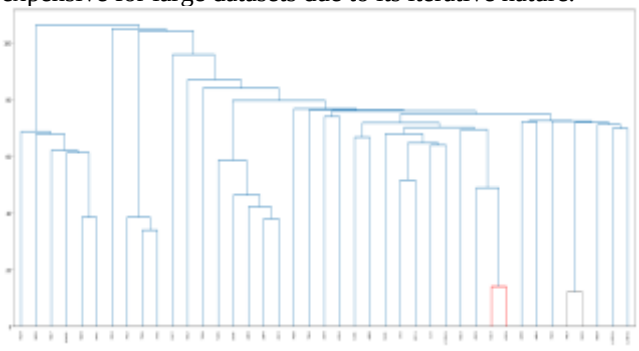

Fig. 6. Dendrogram for Hierarchical Clustering

4-3- Performance Evaluation Metrics

The confusion matrix is a useful tool for evaluating the performance of classification algorithms, as it provides a clear understanding of the accuracy of the results. The confusion matrix is composed of four elements: true positive (TP), true negative (TN), false negative (FN), and false positive (FP). In the context of medical diagnosis, false negatives are considered the most dangerous predictions as they may lead to a failure to diagnose a disease. The different performance metrics such as accuracy (Acc) are calculated using the confusion matrix. Accuracy measures the proportion of correctly classified instances out of all the instances. The true positive rate (TPR), also known as sensitivity, measures the proportion of positive cases that are correctly identified. The precision measures the proportion of true positive cases out of all positive cases identified by the classifier. The F1 score is the harmonic mean of precision and recall, which provides a single score that balances both metrics.

The formula for calculating accuracy was given by

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}} \quad (1)$$

The formula for calculating accuracy was given by

$$\text{Perecision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

Recall(TPR, Sensitivity) identified the proportion of patients with heart disease given by

$$\text{Perecision} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

The F1 score considered a harmonic average between precision in Eq. (2) and recall in Eq. (3) defined by

$$\text{F1 score} = 2\left(\frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}\right) \quad (4)$$

4-4-K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm used for classification based on the concept of finding the closest neighboring data points to a given point. This algorithm belongs to the instance-based learning family of algorithms. The distance between the attributes of a given data point and its neighbors is computed using Euclidean distance. KNN is a type of instance-based learning algorithm where the classification of a new data point is based on the classification of its nearest neighbors [3]. KNN is a simple and effective algorithm, but it has some limitations. One limitation is that it can be computationally expensive for large datasets, especially when the number of features is high. Another

limitation is that it requires careful selection of the value of k to avoid overfitting or underfitting. The summary of the K-Nearest Neighbors (KNN) algorithm is illustrated in Figure 7.

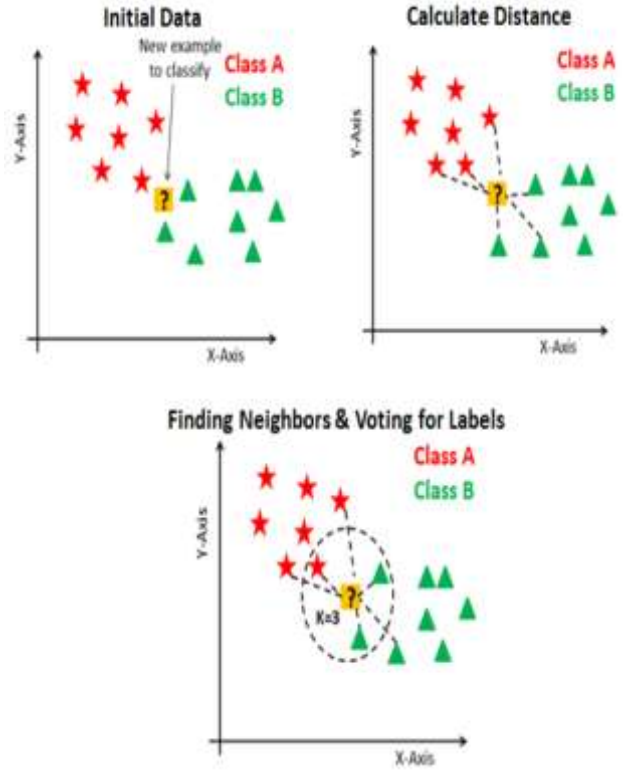

Fig .7. Summary of the K-Nearest Neighbors (KNN) algorithm

The article describes the application of the K-Nearest Neighbors (KNN) algorithm to the Log-Transformed Dataset. In this application, the number of nearest neighbors was set to 9, and the Minkowski distance metric with power parameter 2 was used to compute the distance between data points. The KNN algorithm was applied to classify the data points into different groups based on their attributes. The confusion matrix was calculated to evaluate the performance of the algorithm, and the results were presented in Figure 8. The confusion matrix is a table that shows the number of correct and incorrect predictions made by the classifier. It is a useful tool for evaluating the performance of classification algorithms. KNN is a popular algorithm for classification tasks due to its simplicity and effectiveness. However, the performance of the algorithm is highly dependent on the choice of hyperparameters such as the number of nearest neighbors and the distance metric used.

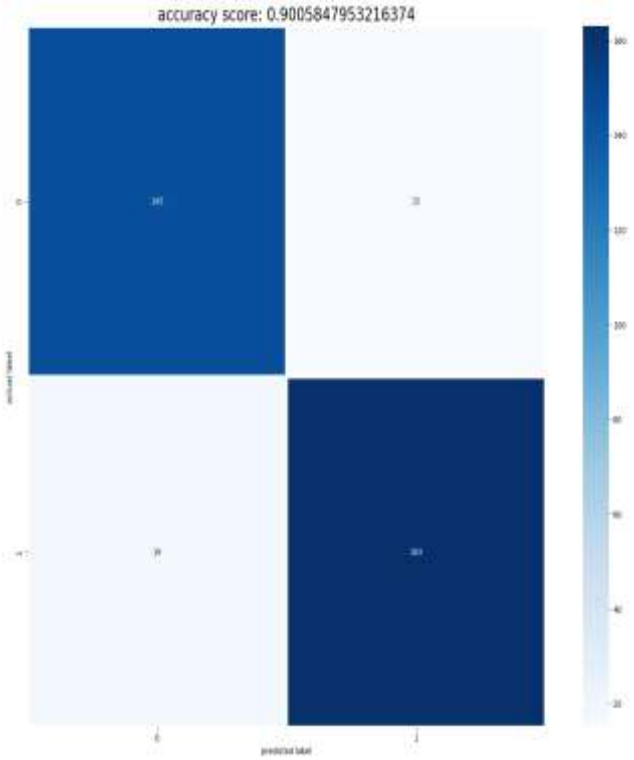

Fig. 8. Confusion matrix and accuracy score for K-Nearest Neighbors

In K-Nearest Neighbors (KNN) algorithm, the error rate is the proportion of misclassified instances in the test dataset. It is calculated by dividing the number of incorrectly classified instances by the total number of instances in the test dataset. The error rate is an important metric for evaluating the performance of a KNN model and determining its accuracy. A lower error rate indicates better performance of the model in classifying data points into their respective classes. However, it is important to note that the choice of k value, distance metric, and data preprocessing techniques can significantly affect the error rate of the KNN model. Therefore, it is recommended to experiment with different hyperparameters and techniques to optimize the performance of the KNN model.

Figure 9 shows the error graph based on the value of k in the KNN algorithm. As the value of k increases, the error rate also increases. This is because a larger value of k means that the algorithm is less sensitive to local variations in the data and more biased toward the overall distribution. On the other hand, a smaller value of k may result in overfitting to the training data and poor generalization to new data. Therefore, it is important to choose the optimal value of k based on the characteristics of the dataset and the desired performance of the algorithm. The error graph provides a useful tool for selecting the appropriate value of k to achieve the best performance of the KNN algorithm.

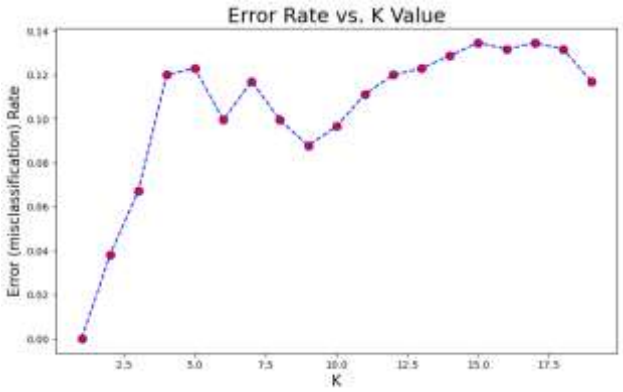

Fig. 9. Error Rate vs. K Value in KNN Algorithm

Figure 10 shows the accuracy graph based on the value of k in the KNN algorithm. As the value of k increases, the accuracy of the algorithm decreases. This is because a larger value of k means that the algorithm is less sensitive to local variations in the data and more biased toward the overall distribution. As a result, the model may fail to accurately capture the underlying patterns in the data. On the other hand, a smaller value of k may result in overfitting to the training data and poor generalization to new data. Therefore, it is important to select the optimal value of k based on the characteristics of the dataset and the desired performance of the algorithm. The accuracy graph provides a useful tool for selecting the appropriate value of k to achieve the best performance of the KNN algorithm. Therefore, the accuracy graph is used to identify the optimal value of k that results in the highest accuracy of the KNN algorithm on a given dataset. The optimal value of k is typically the one that results in the highest accuracy while avoiding overfitting the training data. It is important to note that the optimal value of k may vary for different datasets and depends on the characteristics of the data and the desired performance of the algorithm. In summary, the accuracy graph provides a useful tool for selecting the appropriate value of k to achieve the best performance of the KNN algorithm. It helps to balance the trade-off between bias and variance in the algorithm and avoid overfitting or underfitting the training data.

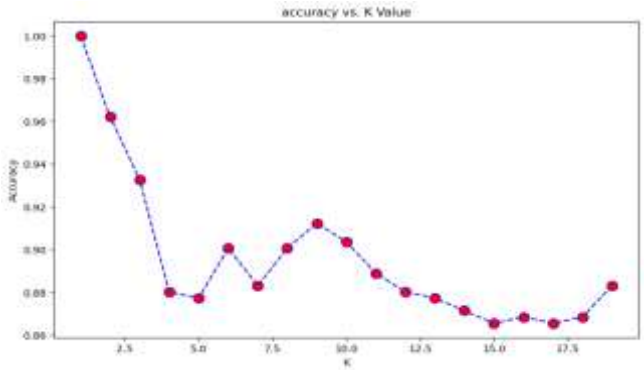

Fig. 10. Accuracy vs. K Value in KNN Algorithm

4-5- Naïve Bayes' Classifier

Naïve Bayes' classifier is a probabilistic algorithm used for classification tasks in machine learning. It is based on Bayes' theorem, which states that the probability of a hypothesis (class) given the evidence (data) is proportional to the probability of the evidence given the hypothesis multiplied by the prior probability of the hypothesis.

The "naïve" assumption in Naïve Bayes' classifier is that the attributes of the data are independent of each other, given the class variable. This simplifies the computation of the conditional probabilities and makes the algorithm computationally efficient and scalable, even for high-dimensional datasets. To classify new data, the algorithm computes the conditional probabilities of each class given the attributes of the data. The class with the highest probability is assigned as the predicted class for the new data. The conditional probabilities are estimated from the training data using maximum likelihood estimation or Bayesian estimation techniques. Naïve Bayes' classifier is widely used in text classification, spam filtering, sentiment analysis, and other classification tasks. It is known for its simplicity, scalability, and effectiveness, especially in cases where the number of features is large compared to the size of the dataset. However, the naïve assumption of independence of attributes may not hold for all datasets, and the algorithm may suffer from the problem of "zero-frequency" when a feature is not present in the training data for a particular class. Many complex real-world situations use Naive Bayes classifiers

$$P(X\ /\ Y) = \frac{P(Y\ /\ X)\ \times\ P(X)}{P(Y)} \qquad (5)$$

P(X/Y) is the posterior probability, P(X) is the class prior probability, P(Y) is the predictor prior probability, and P(Y/X) is the likelihood, and probability of predictor. Naïve Bayes is a simple, easy-to-implement, and efficient classification algorithm that handles non-linear, complicated data. However, there is a loss of accuracy as it is based on assumption and class conditional independence. Figure 11 presents the confusion matrix and accuracy score of the Naïve Bayes' Classifier algorithm on a given dataset.

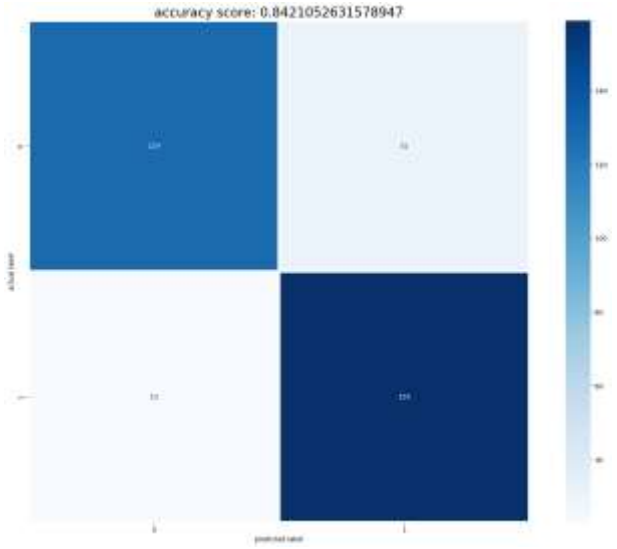

Fig. 11. Confusion matrix and accuracy score for Naïve Bayes' Classifier

4-6- Decision Tree Classification

Decision Tree (DT) is a widely used and one of the oldest machine learning algorithms. The algorithm works by constructing a tree-like structure where each node represents a decision based on a feature in the dataset. The decision leads to a branch that corresponds to the possible outcomes of the decision, and the process continues recursively until a leaf node is reached that represents the final classification decision. The DT algorithm is designed to evaluate and match the results of the classification of data items, by constructing a decision logic in the form of a tree structure. The tree structure represents a set of rules that can be used to classify new data instances based on their features. The algorithm aims to find the best split at each node that maximizes the information gain or minimizes the impurity of the data. DT is a popular algorithm for classification and regression tasks in machine learning due to its simplicity, interpretability, and scalability. The resulting tree structure can be easily visualized and understood, making it a useful tool for decision-making and analysis. However, the algorithm may suffer from overfitting and instability issues, and its performance may depend on the choice of hyperparameters and the characteristics of the dataset. The decision tree algorithm is a popular machine learning algorithm that can be used for classification tasks. In this research, the decision tree algorithm has been applied to the dataset using the Gini coefficient, which is a measure of impurity used to determine the best split at each node of the decision tree.

The Gini coefficient measures the probability of incorrectly classifying a randomly chosen instance from the dataset, which is used to evaluate the quality of each split. After applying the decision tree algorithm using the Gini coefficient, the resulting model can be evaluated using a confusion matrix. Figure 12 shows the confusion matrix for the decision tree model on the given dataset. The confusion matrix is a table that summarizes the number of correct and incorrect predictions made by the model. It consists of four elements: true positive, false positive, true negative, and false negative. The true positive and true negative elements indicate the number of instances that were correctly classified as positive and negative, respectively. The false positive and false negative elements indicate the number of instances that were incorrectly classified as positive and negative, respectively.

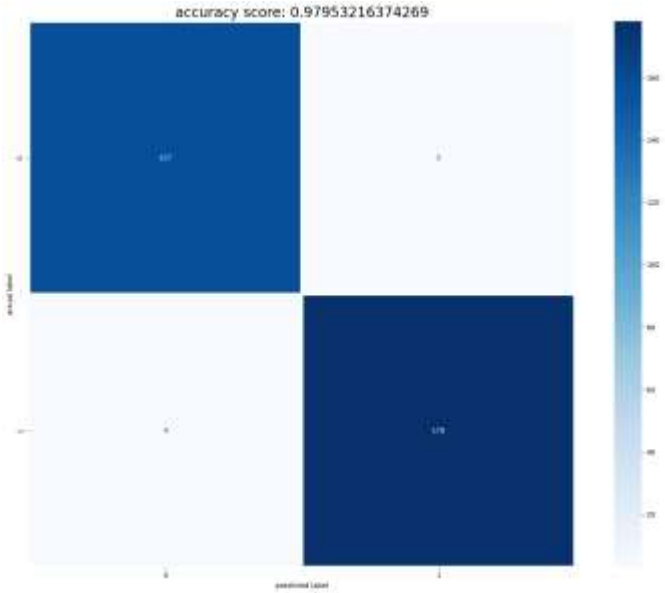

Fig. 12. Confusion matrix and accuracy score for Decision Tree Classification

4-7- Support Vector Classifier (SVC)

SVC (Support Vector Classifier) is a popular machine learning algorithm used for classification tasks. It is a type of supervised learning algorithm that uses a boundary called a hyperplane to separate the data into different classes. The hyperplane is constructed in a way that maximizes the distance between the closest points of the different classes, known as support vectors. The SVC algorithm can be used for both linear and nonlinear classification tasks. In linear classification, the hyperplane is a straight line that separates the data into different classes. In nonlinear classification, the data is transformed into a higher-dimensional feature space where a hyperplane can be used to separate the data into different classes. The SVC algorithm is based on the principle of structural risk minimization, which aims to find the hyperplane that minimizes the classification error while maximizing the margin between the support vectors. The optimization problem is typically solved using quadratic programming techniques. SVC is a powerful and flexible algorithm that has been shown to perform well in many real-world applications, including image classification,

text classification, and bioinformatics. However, the algorithm may suffer from overfitting or underfitting issues, and the choice of hyperparameters can significantly affect its performance. Therefore, it is important to carefully select the hyperparameters and perform cross-validation to ensure optimal performance of the SVC algorithm on a given dataset.

In this research, the Support Vector Classifier (SVC) algorithm has been utilized for the classification task on the given dataset. The SVC algorithm constructs a hyperplane to separate the data into different classes, based on the principle of structural risk minimization. After applying the SVC algorithm can be evaluated using a confusion matrix. Figure 13 shows the confusion matrix for the SVC model on the given dataset.

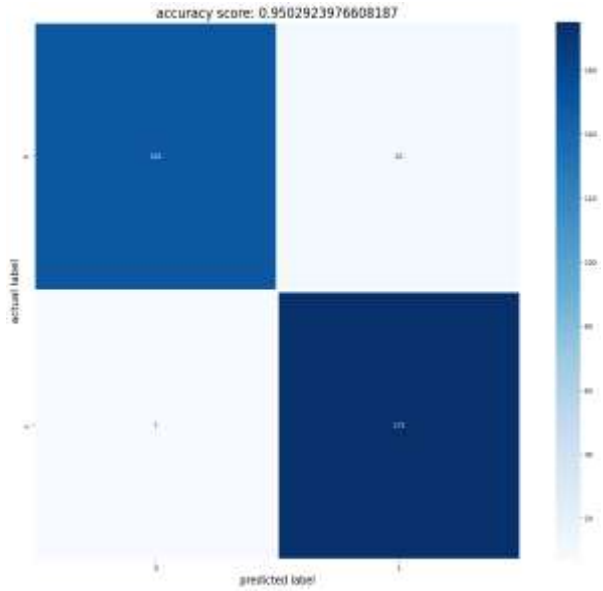

Fig. 13. Confusion matrix and accuracy score for Support Vector Classifier

Drawing the Support Vector Classifier (SVC) or Support Vector Machine (SVM) plot involves plotting the data points in a two-dimensional or three-dimensional space and drawing the hyperplane that separates the data into different classes. The hyperplane is constructed in a way that maximizes the margin between the closest points of the different classes, known as support vectors. In a two-dimensional space, the hyperplane is a straight line that separates the data into different classes. The position and orientation of the line are determined by the weights assigned to the features of the data. The distance between the line and the closest points of the different classes is known as the margin. The margin is maximized by finding the hyperplane that minimizes the classification error while maximizing the margin between the support vectors. In three-dimensional space, a hyperplane is a plane that separates the data into different classes. The position and orientation of the plane are determined by the weights assigned to the features of the data. The margin between the plane and the closest points of the different classes is maximized by finding the hyperplane that minimizes the classification error while maximizing the margin between the support vectors. Drawing the SVC or SVM can help visualize the decision boundary and understand how the algorithm separates the data into different classes. It can also help identify potential issues such as overfitting or underfitting, which can affect the performance of the algorithm on new data.

The percentage of explained variance diagram is a tool used in exploratory data analysis and dimensionality reduction techniques such as Principal Component Analysis (PCA). It shows the proportion of variance in the data that is explained by each principal component. Figure 14 shows the percentage of explained variance diagrams for a dataset that has been analyzed using PCA. The diagram presents a scree plot, which is a line graph that shows the proportion of variance explained by each principal component. Each principal component represents a linear combination of the original variables that capture the maximum amount of variance in the data.

The percentage of explained variance diagram can be used to determine the optimal number of principal components to retain in the analysis. The diagram shows that the first principal component explains the majority of the variance in the data, followed by the second and third principal components. Subsequent principal components explain progressively less variance in the data. By examining the percentage of explained variance diagrams, analysts can identify which principal components are the most important in explaining the variability in the data. This information can be used to inform decisions about how many principal components to retain in the analysis, and how to interpret the resulting reduced-dimensional representation of the data.

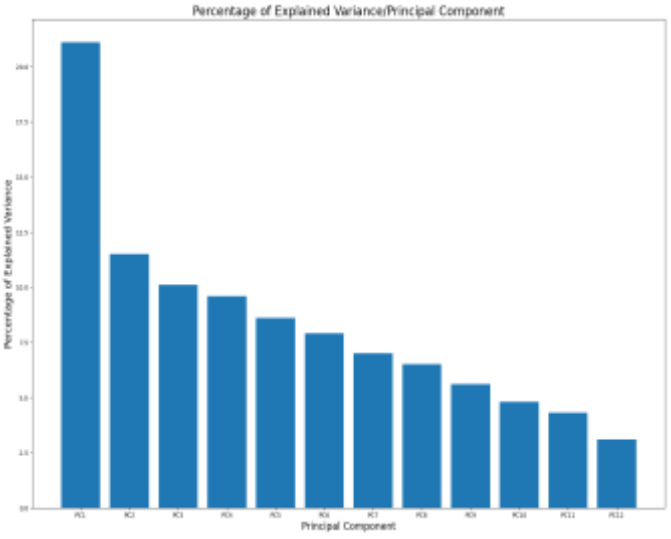

Fig. 14. Percentage of Explained Variance/Principal Component

In this research, the Support Vector Classifier (SVC) algorithm has been applied to the given dataset for classification tasks. To visualize the decision boundary and the separation of the data into different classes, the SVC algorithm has been used to draw the classification. The resulting classification can be seen in Figure 15. Drawing the classification of the applied support vector can help visualize the decision boundary and understand how the algorithm separates the data into different classes. The classification is typically plotted in a two-dimensional or three-dimensional space, where the x and y (or x, y, and z) axes represent the features of the data, and the color or shape of the data points represents their class. The decision boundary is typically shown as a line or surface that separates the data points of different classes. In Figure 15, the classification of the applied support vector is shown, where the blue and green colors represent the two classes of data. The decision boundary is shown as a curved line that separates the two classes. The classification plot provides valuable insights into the performance of the SVC algorithm on the given dataset. It can be used to identify potential issues such as overfitting or underfitting, and to fine-tune the hyperparameters of the algorithm to improve its performance.

In summary, drawing the classification of the applied support vector involves plotting the data points in a two-dimensional or three-dimensional space and visualizing the decision boundary that separates the data into different classes. It can help understand how the algorithm works and identify potential issues that can affect its performance.

5-Results & Discussion

For this study, Machine Learning was conducted on a dataset related to heart disease. we have processed the dataset and identified any outliers present in the data. To handle the outliers, they have employed

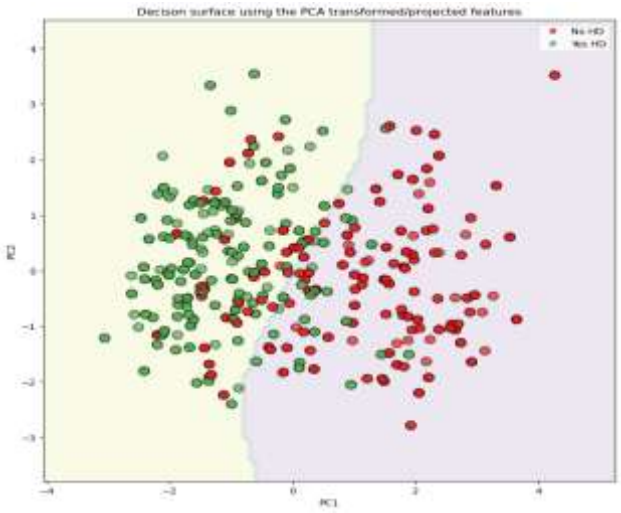

Fig. 15. Decision surface using the PCA transformed/projected features

Winsorization methods involve replacing the extreme values with less extreme ones and logarithmic transformation methods which involve transforming the data to a logarithmic scale. The researchers have decided to use the data obtained through logarithmic transformation for their analysis and classification algorithms. we have employed various classification algorithms such as K-Nearest Neighbor (KNN), Naive Bayes (NB), Decision Tree (DT), and Support Vector Classification (SVC) to classify the data and draw meaningful conclusions.

The results of implementing the KNN algorithm on the developed model can be observed in Table 7. Along with the accuracy score of 90.05%, we have also evaluated the precision, recall, f1-score, and support numbers of the model. These metrics are used to assess the model's performance in terms of precision and recall of the positive and negative labels, the harmonic mean of precision and recall (f1-score), and the support number of each class. Additionally, the accuracy of the training score was found to be 89.89%, indicating that the model is also performing well on the training data.

Macro-averaging and Weighted averaging: are two commonly used techniques to evaluate the performance of classification models on imbalanced datasets. Macro-averaging involves calculating the metrics for each class individually and then taking the average of those metrics across all classes. This approach treats each class equally, regardless of its size or prevalence in the dataset. Macro-averaging is useful when the dataset is well-balanced, i.e., there are roughly equal numbers of instances in each class. Weighted averaging, on the other hand, takes into account the class imbalance in the dataset by weighting the metrics according to the number of instances in each class. This approach gives more weight to the metrics for the larger classes and less weight to the smaller classes. Weighted averaging is useful when the dataset is imbalanced, i.e., there are significantly more instances in one class than in the others.

Table 7
Classification Report of K-Nearest Neighbors

	Precision	Recall	F1-score	Support
Accuracy			0.90	342
Macro avg	0.90	0.90	0.90	342
Weighted avg	0.90	0.90	0.90	342
Training Score	89.89751098096632 %			
Model accuracy score	90.05847953216374 %			

The results of implementing the NB algorithm on the developed model can be observed in Table 8. Along with the accuracy score of 84.21%, we have also evaluated the precision, recall, f1-score, and support numbers of the model. These metrics are used to assess the model's performance in terms of precision and recall of the positive and negative labels, the harmonic mean of

precision and recall (f1-score), and the support number of each class. Additionally, the accuracy of the training score was found to be 80.52%, indicating that the model is also performing well on the training data.

Table 8
Classification Report of NB

	Precision	Recall	F1-score	Support
Accuracy			0.84	342
Macro avg	0.84	0.84	0.84	342
Weighted avg	0.84	0.84	0.84	342
Training Score	80.52708638360176 %			
Model accuracy score	84.21052631578947 %			

The results of implementing the DT algorithm on the developed model can be observed in Table 9. Along with the accuracy score of 97.95%, we have also evaluated the precision, recall, f1-score, and support numbers of the model. These metrics are used to assess the model's performance in terms of precision and recall of the positive and negative labels, the harmonic mean of precision and recall (f1-score), and the support number of each class. Additionally, the accuracy of the training score was found to be 100%, indicating that the model is also performing well on the training data.

Table 9
Classification Report of DT

	Precision	Recall	F1-score	Support
Accuracy			0.98	342
Macro avg	0.98	0.98	0.98	342
Weighted avg	0.98	0.98	0.98	342
Training Score	100.0 %			
Model accuracy score	97.953216374269 %			

The results of implementing the SVC algorithm on the developed model can be observed in Table 10. Along with the accuracy score of 95.02%, we have also evaluated the precision, recall, f1-score, and support numbers of the model. These metrics are used to assess the model's performance in terms of precision and recall of the positive and negative labels, the harmonic mean of precision and recall (f1-score), and the support number of each class. Additionally, the accuracy of the training score was found to be 93.41%, indicating that the model is also performing well on the training data.

Table 10
Classification Report of SVC

	Precision	Recall	F1-score	Support
Accuracy			0.65	342
Macro avg	0.95	0.95	0.95	342
Weighted avg	0.95	0.95	0.95	342
Training Score	93.41142020497803 %			
Model accuracy score	95.02923976608187 %			

Table 11& Figure 16 presented a comparison of the accuracy scores obtained by applying four different classification algorithms on the dataset. The algorithms used for this comparison are K-Nearest Neighbor (KNN), Naive Bayes (NB), Decision Tree (DT), and Support Vector Classification (SVC). we have evaluated the performance of each algorithm and presented their corresponding accuracy scores in the table. The comparison helps to identify which algorithm performs better in terms of accuracy and can be used to make informed decisions about which algorithm to use for future classification tasks.

Table 11

Comparison of the Accuracy Scores

Algorithm	Accuracy SCORE
KNN - K Nearest Neighbors	90.0058 %
Naive Bayes	84.2105 %
Decision Tree Classifier	97.9532 %
Support vector classifier	95.0292 %

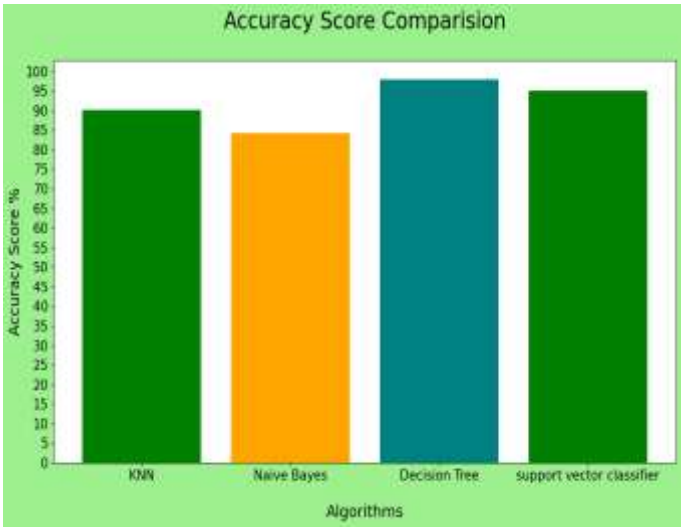

Fig. 16. Comparison of the Accuracy Scores

Based on the results presented in the table and figure, it is evident that the Decision Tree algorithm using the Gini coefficient has the highest accuracy among the four classification algorithms considered. The accuracy score obtained for this algorithm is 0.979532, indicating that it is highly effective in classifying instances in the dataset.

A comparison of the accuracy score of heart disease prediction in the proposed model with different authors is given in Table 12.

Table 12
Accuracy of Heart Disease Prediction with Different Techniques

Authors	Technique	Accuracy
Otoom et al. [17]	Naïve Bayes	84.5%
	SVM	84.5%
	Functional trees	84.5%
Vembandasamy et al. [18]	Naïve Bayes	86.419%
Chaurasia et al. [19]	J48	84.35%
	Bagging	85.03%
	SVM	94.60%
Parthiban et al. [20]	Naïve Bayes	74%
Seema et al. [21]	Naïve Bayes	95.556%
Kumar Dwivedi [22]	Naïve Bayes	83%
	Classification tree	77%
	K-NN	80%
	Logistic regression	85%
	SVM	82%
	ANN	84%
The proposed model	KNN	90.05 %
	Naive Bayes	84.21 %
	DT	97.95 %
	SVC	95.02 %

6-Conclusion

To predict heart disease in Cleveland patients, we attempted to offer a model in the current research. In this study, we first attempted to identify the outlier data by creating a boxplot diagram. Once the features of the outlier data were identified, we handled these outlier data using Winsorized and Logarithmic transformations methods, which according to Better Handling of the Logarithmic Transformations method for use in machine learning algorithms has been used from the dataset obtained after applying this method. Using Wrapper approaches, we then identified which features of our dataset were more crucial for doctors to detect heart disease, which showed that in addition to the classic features for diagnosing heart disease, such as blood pressure and chest pain. This demonstrated that in addition to the typical features for diagnosing heart disease, such as blood pressure and chest pain, other features, like Exercise-induced angina and the number of major vessels, can also have special importance. Finally, using Embedded methods, we determined the importance score of each feature. Following, we report the outcomes of two unsupervised machine learning algorithms we applied, K-Means and Hierarchical Clustering. Lastly, we used four machine learning algorithms such as KNN, Naive Bayes' Classifier, Decision Tree, and Support Vector Classifier to decide on the best algorithm for predicting heart disease. Based on the accuracy of these algorithms, it was determined that the Decision Tree algorithm performed the best among them, with a 97.95% accuracy rate. Future investigation might include the following:

1. Genetic factors contribute to the development of heart disease. Future studies may attempt to establish genetic indicators that might be used to predict a person's propensity to establish heart disease.

2. Lifestyle habits such as nutrition, exercise, and stress can contribute to the development of heart disease. Further research could look into how these characteristics affect the risk of heart disease.

3. Examining different methods to solving outliers or missing data.

Reference

[1] Cardiovascular diseases n.d. https://www.who.int/health-topics/cardiovascular-diseases#tab=tab_1

[2] Heart and Stroke Statistics. WwwHeartOrg n.d. https://www.heart.org/en/about-us/heart-and-stroke-association-statistics

[3] Weng, S. F., Reps, J., Kai, J., Garibaldi, J. M., & Qureshi, N. (2017). Can machine-learning improve cardiovascular risk prediction using routine clinical data?. PloS one, 12(4), e0174944.

[4] Ahmed, H., Younis, E. M., Hendawi, A., & Ali, A. A. (2020). Heart disease identification from patients' social posts, machine learning solution on Spark. Future Generation Computer Systems, 111, 714-722.

[5] Shah, D., Patel, S., & Bharti, S. K. (2020). Heart disease prediction using machine learning techniques. SN Computer Science, 1, 1-6.

[6] Ali, M. M., Paul, B. K., Ahmed, K., Bui, F. M., Quinn, J. M., & Moni, M. A. (2021). Heart disease prediction using supervised machine learning algorithms: Performance analysis and comparison. Computers in Biology and Medicine, 136, 104672.

[7] Riyaz, L., Butt, M. A., Zaman, M., & Ayob, O. (2022). Heart disease prediction using machine learning techniques: a quantitative review. In International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 3 (pp. 81-94). Springer Singapore.

[8] Sekar, J., Aruchamy, P., Sulaima Lebbe Abdul, H., Mohammed, A. S., & Khamuruddeen, S. (2022). An efficient clinical support system for heart disease prediction using TANFIS classifier. Computational Intelligence, 38(2), 610-640.

[9] Katarya, R., & Meena, S. K. (2021). Machine learning techniques for heart disease prediction: a comparative study and analysis. Health and Technology, 11, 87-97.

[10] Pathan, M. S., Nag, A., Pathan, M. M., & Dev, S. (2022). Analyzing the impact of feature selection on the accuracy of heart disease prediction. Healthcare Analytics, 2, 100060.
[11] El-Shafiey, M. G., Hagag, A., El-Dahshan, E. S. A., & Ismail, M. A. (2022). A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. Multimedia Tools and Applications, 81(13), 18155-18179.
[12] Hassan, D., Hussein, H. I., & Hassan, M. M. (2023). Heart disease prediction based on pre-trained deep neural networks combined with principal component analysis. Biomedical Signal Processing and Control, 79, 104019.
[13] Arumugam, K., Naved, M., Shinde, P. P., Leiva-Chauca, O., Huaman-Osorio, A., & Gonzales-Yanac, T. (2023). Multiple disease prediction using Machine learning algorithms. Materials Today: Proceedings, 80, 3682-3685.
[14] Ozcan, M., & Peker, S. (2023). A classification and regression tree algorithm for heart disease modeling and prediction. Healthcare Analytics, 3, 100130.
[15] Dileep, P., Rao, K. N., Bodapati, P., Gokuruboyina, S., Peddi, R., Grover, A., & Sheetal, A. (2023). An automatic heart disease prediction using cluster-based bi-directional LSTM (C-BiLSTM) algorithm. Neural Computing and Applications, 35(10), 7253-7266.
[16] Cleveland, Hungary, S., the VA Long Beach, 2019. Heart disease data set. https://archive.ics.uci.edu/ml/datasets/heart+Disease
[17] Otoom, A. F., Abdallah, E. E., Kilani, Y., Kefaye, A., & Ashour, M. (2015). Effective diagnosis and monitoring of heart disease. International Journal of Software Engineering and Its Applications, 9(1), 143-156.
[18] Vembandasamy, K., Sasipriya, R., & Deepa, E. (2015). Heart diseases detection using Naive Bayes algorithm. International Journal of Innovative Science, Engineering & Technology, 2(9), 441-444.
[19] Chaurasia, V., & Pal, S. (2014). Data mining approach to detect heart diseases. International Journal of Advanced Computer Science and Information Technology (IJACSIT) Vol, 2, 56-66.
[20] Parthiban, G., & Srivatsa, S. K. (2012). Applying machine learning methods in diagnosing heart disease for diabetic patients. International Journal of Applied Information Systems, 3(7), 25-30.
[21] Deepika, K., & Seema, S. (2016, July). Predictive analytics to prevent and control chronic diseases. In 2016 2nd international conference on applied and theoretical computing and communication technology (iCATccT) (pp. 381-386). IEEE.
[22] Dwivedi, A. K. (2018). Performance evaluation of different machine learning techniques for prediction of heart disease. Neural Computing and Applications, 29, 685-693.