

**Exploring Heart Disease Prediction Dataset with Machine Learning Report**



**A pair of hands holding a heart beat

Description automatically generated**

January 1, 2024

**Introduction:**

The document provides a comprehensive analysis of the Heart Disease Prediction Dataset using machine learning algorithms. It encompasses data preprocessing, exploratory data analysis, and the implementation of various classifiers such as Support Vector Machine, Decision Tree, Logistic Regression, and Random Forest. The analysis includes visualization of decision trees, estimator trees, confusion matrices, and accuracy assessments for both training and test sets. Additionally, the document utilizes Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn for data analysis and visualization. The dataset is transformed and scaled using techniques such as Principal Component Analysis (PCA) and Standard Scaler. The document also addresses missing values in the dataset and maps categorical variables for analysis. Overall, it offers valuable insights into the application of machine learning in predicting heart disease based on health parameters.

**Abstract:**

The document presents an analysis of a heart prediction dataset using machine learning models. It involves preprocessing, visualization, and application of various classifiers such as Support Vector Machine (SVM), Naive Bayes, Logistic Regression, Decision Tree, and Random Forest. The dataset contains features such as age, sex, chest pain type, resting blood pressure, cholesterol, fasting blood sugar, resting ECG, maximum heart rate, exercise-induced angina, ST depression induced, peak ST, number of major vessels, thal, and target disease. The report aims to provide insights into the effectiveness of machine learning models in predicting heart disease.

**Problem Definition:**  
The problem involves predicting heart disease based on various features present in the heart prediction dataset. The challenge is to analyze the dataset, preprocess the data, and apply machine learning models to accurately predict the presence of heart disease based on the given features.

**Description:**  
The document includes the preprocessing of the heart prediction dataset, visualization using seaborn and matplotlib, and the application of machine learning models. It also involves the evaluation of the models' performance using accuracy metrics, visualization of confusion matrices, and decision tree plots. The report aims to provide a comprehensive analysis of the dataset and the effectiveness of different machine learning models in predicting heart disease.

**Aims & Objectives:**  
The primary aim is to analyze the heart prediction dataset and evaluate the performance of various machine learning models in predicting heart disease. The objectives include preprocessing the dataset, visualizing the data, training, and evaluating machine learning models, and providing insights into the predictive capabilities of the models.

**Requirements Specification:**  
The project requires a heart prediction dataset, Python programming language, and various libraries such as pandas, numpy, matplotlib, seaborn, and sklearn for data analysis and machine learning model implementation. Additionally, the project requires access to a computing environment capable of running machine learning algorithms.

**Research Methodology:**The research methodology involves data preprocessing, exploratory data analysis, visualization of the dataset, training and evaluation of machine learning models, and interpretation of the results. The analysis includes the use of various machine learning algorithms to predict heart disease based on the provided dataset.

**Ethical Issues:**  
The project adheres to ethical considerations related to data privacy, confidentiality, and responsible use of machine learning algorithms for healthcare applications. The handling of sensitive medical data is conducted with utmost care and in compliance with data protection regulations.

**Deliverables:**  
The deliverables include a comprehensive report on the analysis of the heart prediction dataset, the performance of machine learning models, and insights into the predictive capabilities of the models. Additionally, the project aims to provide visualizations and interpretations that can contribute to the understanding of heart disease prediction.

**Project Resources:**  
The project requires access to the heart prediction dataset, computational resources for data analysis and model training, and expertise in machine learning and data analysis techniques. Additionally, access to relevant libraries and tools for visualization and model implementation is essential.

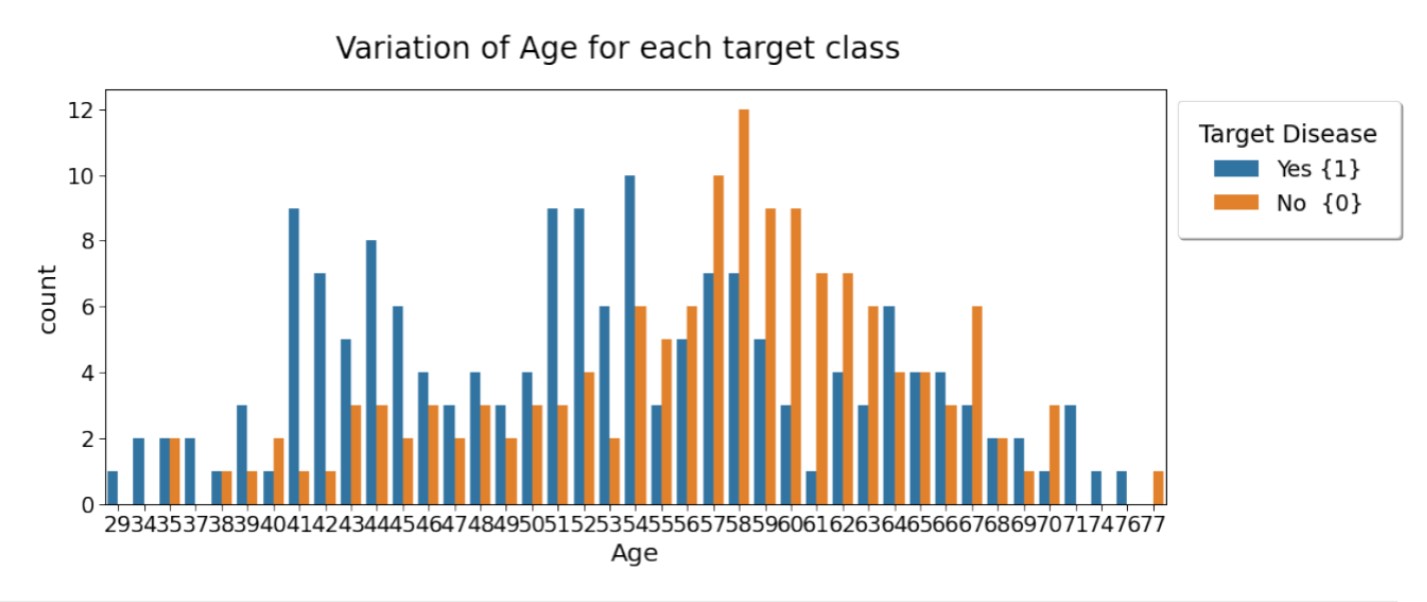
**Milestones:**  
The project milestones include data preprocessing, visualization of the dataset, training and evaluation of machine learning models, interpretation of results, and the generation of a comprehensive report. The project timeline is structured to achieve these milestones in a systematic and efficient manner.

Overall, the project aims to contribute to the understanding of heart disease prediction using machine learning models and provide valuable insights for healthcare professionals and researchers in the field of cardiovascular health.

**Conclusion:**

As a result, the report and findings focus on the analysis of a Heart Disease Prediction Dataset using machine learning algorithms. The analysis involves the use of various classifiers such as Support Vector Machine (SVM), Logistic Regression, Decision Tree, and Random Forest. The document includes the visualization of decision boundaries, confusion matrices, and accuracy assessments for both training and test sets. Additionally, it involves data preprocessing, exploratory data analysis, and the implementation of machine learning models.

**Graph Analysis:**

****

**Diagram 1.1**

A comparison of a graph

Description automatically generated with medium confidenceFrom **diagram 1.1**, it can be observed that certain age groups have higher counts of individuals with heart disease compared to those without, and vice versa. The chart enables a visual comparison of the prevalence of heart disease across various age groups.

**Plot 1.1**

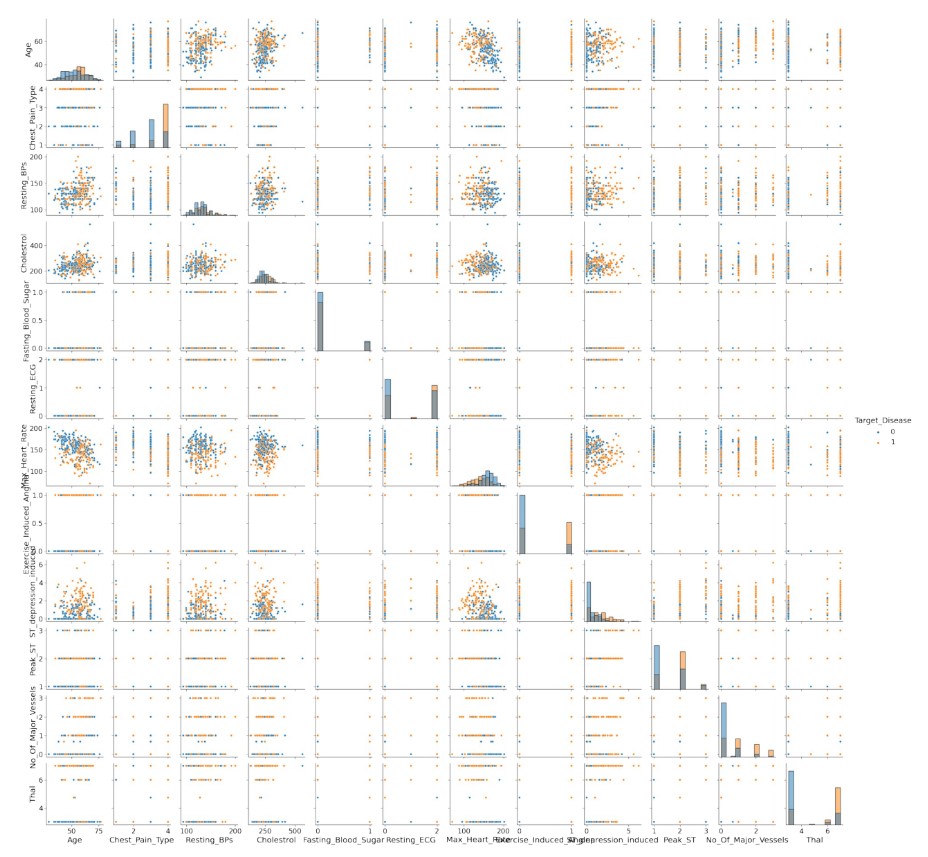
**Plot 1.1** displays two types of plots that represent the distribution of age versus sex within the context of a heart disease study. On the left, there is a swarm plot, and on the right, there is a violin plot. Both plots are segmented by sex: male and female.

Swarm Plot (Left):

* The x-axis categorizes individuals by sex: male or female.
* The y-axis represents the age of individuals.
* Each dot represents an individual, and the dots are spread out to show the distribution of ages within each sex category without overlapping.

Violin Plot (Right):

* Like the swarm plot, the x-axis categorizes by sex and the y-axis by age.
* The violin plot shows the density of individuals at different ages, with wider sections indicating a higher density of individuals.
* The plot combines aspects of a box plot with a kernel density plot to show the distribution of ages for each sex.



**Plot 1.2**

From **plot 1.2**, one can analyze the age distribution and density differences between males and females within the target class of a heart disease dataset.

The image shows a pair plot (also known as a scatterplot matrix) of various features from a dataset related to heart disease prediction. Each small plot represents the relationship between two variables, with histograms on the diagonal showing the distribution of a single variable. The data points are colored based on two categories of a ‘Target\_Disease’ variable, with ‘0’ indicating no disease and ‘1’ indicating the presence of disease.

Here are some obs’rvations that ca‘ ’e made from the plots:

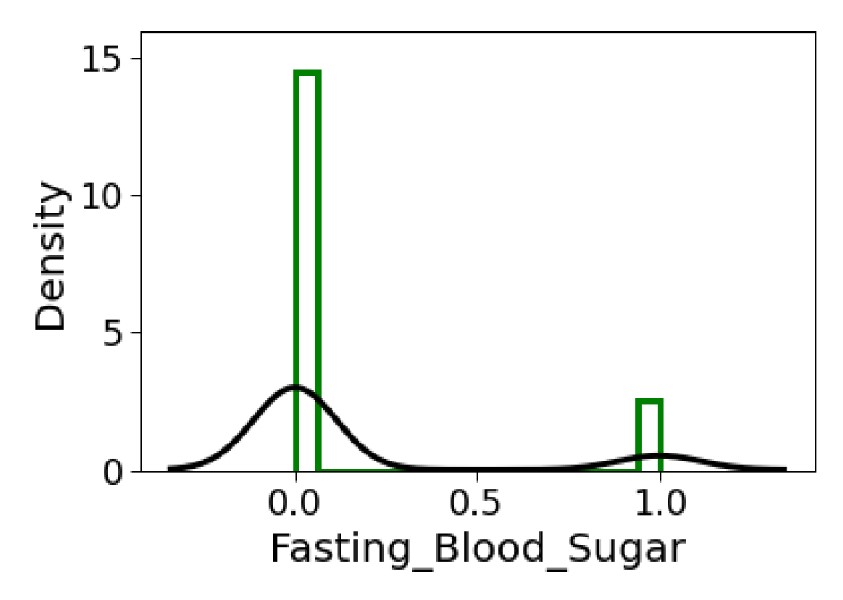
1. The distributions of some variables like ‘Age’, ‘Resting\_BP’ (Resting Blood Pressure), and ‘Cholesterol’ are relatively smooth and continuous, while others like ‘Chest\_Pain\_Type’ and ‘Fasting\_Blood\_Sugar’ show discrete groupings, indicating they are categorical or binary features.
2. In several plots, the two categories of the ‘Target\_Disease’ show distinct clusters, suggesting that these features could be good predictors of heart disease. For example, ‘Max\_Heart\_Rate’ shows a noticeable difference in distribution between the two categories.
3. The histograms along the diagonal allow us to see the frequency distribution of each feature for patients with and without heart disease. Some features, such as ‘Oldpeak’ (ST depression induced by exercise relative to rest), show a visibly different distribution between the two categories of the target variable.

A graph of a normal distribution

Description automatically generated

**Plot 1.3**

**Plot 1.3** shows a density plot, which is used to visualize the distribution of a continuous variable, in this case, 'Cholesterol'. The x-axis represents the cholesterol levels, while the y-axis represents the density of the observations. The plot appears to show a unimodal distribution that is roughly symmetric, indicating that the cholesterol levels in the dataset are clustered around a central value with fewer observations toward the tails of the distribution.

The green histogram in the background represents the actual data points, while the black line is likely a kernel density estimate (KDE), which provides a smooth curve that approximates the underlying probability density function of the variable.

**Plot 1.4**

**Plot 1.4** shows a density plot for a variable called "Fasting\_Blood\_Sugar". On the horizontal axis (x), two different values, 0.0 and 1.0, are represented, which probably correspond to binary categories, while on the vertical axis (y), the density is shown, which is an estimation of the probability distribution of the variable. There are two prominent peaks, one at 0.0 and a much smaller one at 1.0, indicating that most of the data falls into the category 0.0 with a much lower density of observations in the category 1.0.

A graph of a normal distribution

Description automatically generated

**Plot 1.5**

**Plot 1.5** shows a density plot with two overlapping distributions and vertical lines indicating specific values on the x-axis, labeled "Resting\_ECG."

Here's an analysis of the image showing a density plot with overlaid histograms. Here's the plot:

* **X-axis**: Represents 'Resting\_ECG' values, rangingDensity Plot\*\*: Represents the distribution of from approximately -1 to 3.
* **Y-axis**: Represents 'Density', ranging from 0 to 1.
* **Histogram**: The bars indicate the frequency distribution of 'Resting a continuous variable.
* **Two Distributions**: Suggests\_E comparison between two different groups or conditions.
* **Vertical Lines**: TypicallyCG mark mean or' median values values of the distributions.
* **X-axis**: Labeled "Resting\_ECG," likely indicating the variable measured is related to electro-cardiogram results.
* **Y-axis**: Labeled "Density," showing the probability density of the values.

The plot is used to visualize the distribution and central tendency of Resting\_ECG values for two different groups or conditions. The overlap of the distributions indicates areas of common values between the groups, while the distinct peaks suggest differences in the central values. **Density Curve**: The smooth curve suggests the probability density function of the data, estimating where values are more likely to occur.

* **Two Peaks**: The plot has two prominent peaks, suggesting a bimodal distribution of 'Resting\_ECG' values.

A graph of a normal heart rate

Description automatically generated

**Plot 1.6**

**Plot 1.6** demonstrates a statistical graph, specifically a histogram overlaid with a density plot. The histogram is displayed with green bars and the density plot with a black curve. The x-axis is labeled "Max\_Heart\_Rate" which indicates that the data represents the maximum heart rate values of a sample population or dataset. The y-axis is labeled "Density," which suggests that the graph shows the probability density of the different maximum heart rate values.

**A graph of exercise and angina

Description automatically generated**

**Plot 1.7**

**Plot 1.7** demonstrates another statistical graph, like the previous one, where a histogram is overlaid with a density plot. The histogram is shown with green bars and the density plot with a black curve. The x-axis is labeled "Exercise\_Induced\_Angina," which indicates that the data represents whether individuals experienced angina (a type of chest pain) induced by exercise. This is likely coded as a binary variable, where 0 represents "no" and 1 represents "yes."

The y-axis is labeled "Density," which represents the probability density for the occurrences of exercise-induced angina in the sample or dataset. The histogram bars show the frequency of the dataset's binary outcomes, while the density plot gives a smoothed representation of this distribution.

A graph of a normal distribution

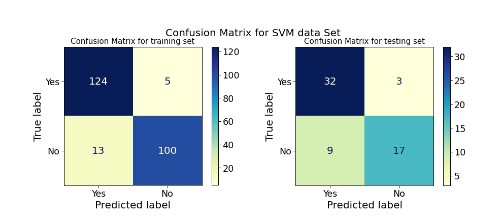
Description automatically generated

**Plot 1.8**

**Plot 1.8** depicts a density plot, which is a visualization tool used to show the distribution of a continuous variable. The plot is specifically for a variable labeled "ST\_depression\_induced." The x-axis represents the values of ST depression, and the y-axis represents the density of these values.

The plot includes a black line that shows the kernel density estimation (KDE), which is a smooth, continuous line that provides an estimate of the probability density function of the variable. There are also vertical green lines at the base of the plot, which appear to be a histogram representing the frequency of the observed values in discrete intervals.

From the shape of the KDE, we can infer that most of the data points are clustered around the lower values of ST depression, with the peak of the density occurring near zero. The distribution then tails off as the ST depression values increase, indicating fewer occurrences of higher ST depression values in the dataset.

**Matrix 1.1**

**Matrix 1.1** shows two confusion matrices for the performance of a Support Vector Machine (SVM) classifier on a dataset. The left matrix is the confusion matrix for the training set, and the right matrix is for the testing set.

In the confusion matrix for the training set:

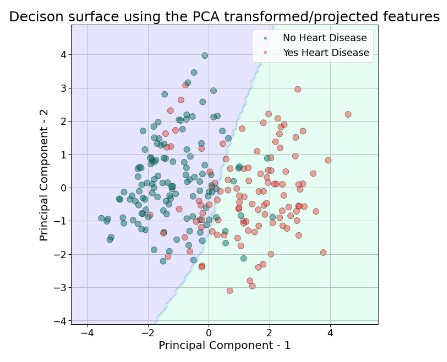
* The classifier correctly predicted 'Yes' 124 times and incorrectly predicted 'Yes' 5 times (these are false positives).
* It incorrectly predicted 'No' 13 times (these are false negatives) and correctly predicted 'No' 100 times.

In the confusion matrix for the testing set:

* The classifier correctly predicted 'Yes' 32 times and incorrectly predicted 'Yes' 3 times (false positives).
* It incorrectly predicted 'No' 9 times (false negatives) and correctly predicted 'No' 17 times.

Below the matrices, there is a command in Python:

This command seems to be a custom function call (not a standard library function) that probably generates a plot to visualize the decision boundaries or performance of theclassifier, where **classifier** is the trained SVM model, **X\_train** is the training data, and **X\_test** is the testing data.

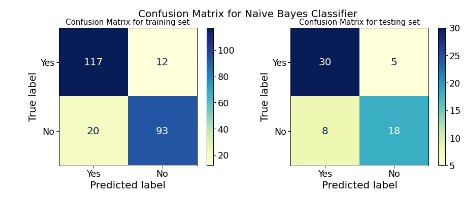


**Plot 2.1**

**Plot 2.1** shows a scatter plot which represents the decision surface of a classifier using two principal components obtained from a PCA (Principal Component Analysis) transformation of the features. The data points are colored based on the presence of heart disease: teal for 'No Heart Disease' and coral for 'Yes Heart Disease'.

The axes are labeled "Principal Component - 1" and "Principal Component - 2", indicating the first and second principal components, respectively. These components are linear combinations of the original features that capture the most variance in the data. The decision boundary appears to be represented by the line where the color of the background changes, distinguishing the two classes.

The overlap of the two colors in the scatter plot suggests that there is some degree of misclassification by the classifier, as points from different classes are found on both sides of the decision boundary.



**Matrix 1.2**

**Matrix 1.2** shows two confusion matrices for the Naive Bayes Classifier applied to a heart disease prediction dataset. Confusion matrices are used to evaluate the performance of a classification algorithm. The left matrix represents the performance on the training set, while the right matrix represents the performance on the testing set.

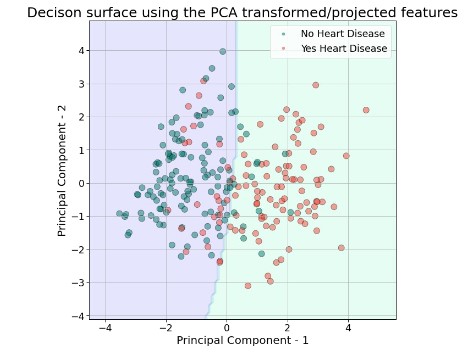
For the training set:

* 117 instances were correctly predicted as 'Yes' (true positives).
* 93 instances were correctly predicted as 'No' (true negatives).
* 12 instances were incorrectly predicted as 'No' when they were 'Yes' (false negatives).
* 20 instances were incorrectly predicted as 'Yes' when they were 'No' (false positives).

For the testing set:

* 30 instances were correctly predicted as 'Yes' (true positives).
* 18 instances were correctly predicted as 'No' (true negatives).
* 5 instances were incorrectly predicted as 'No' when they were 'Yes' (false negatives).
* 8 instances were incorrectly predicted as 'Yes' when they were 'No' (false positives).

These matrices provide insight into the classifier's ability to correctly predict the presence or absence of heart disease, as well as the types of errors it is making.



**Plot 2.2**

**Plot 2.2** depicts a decision surface created using Principal Component Analysis (PCA) transformed or projected features from a dataset concerning heart disease. PCA is a dimensionality reduction technique that transforms the data into a new coordinate system with the axes called principal components, which are ordered by the amount of variance they explain in the data.

In the scatter plot:

* The horizontal axis is labeled "Principal Component - 1" and the vertical axis is labeled "Principal Component - 2."
* Data points are color-coded: teal dots represent individuals without heart disease ("No Heart Disease"), and coral dots represent individuals with heart disease ("Yes Heart Disease").
* The background shows a decision surface, which is the area where the model's decision boundary is visualized, separating the two classes based on the PCA features.

This visualization helps to understand how well the machine learning model can distinguish between the two classes after reducing the feature space to two dimensions. Overlapping points suggest that there may be some difficulty in perfectly separating the A comparison of a logistic regression class

Description automatically generatedclasses using the given features.

**Matrix 1.3**

**Plot 1.3** demonstrates two confusion matrices for a logistic regression classifier, one for the training set and one for the testing set. A confusion matrix is a table often used to describe the performance of a classification model on a set of data for which the true values are known.

The left matrix, labeled "Confusion Matrix for training set," shows the results of the logistic regression classifier when applied to the training data. The matrix has two rows and two columns that indicate the number of predictions that fall into each possible outcome category. The rows represent the true classes (Yes or No), whereas the columns represent the predicted classes (Yes or No). From the matrix, we can see that for the "Yes" class, the classifier correctly predicted 115 instances (true positives) and incorrectly predicted 14 instances (false negatives). For the "No" class, there were 20 false positives and 93 true negatives.

The right matrix, labeled "Confusion Matrix for testing set," shows similar information but for the testing data. In this case, the classifier correctly predicted 4 true positives (Yes) and 17 true negatives (No). However, there are errors in both classes: 1 false negative (Yes that was incorrectly predicted as No) and 9 false positives (No that was incorrectly predicted as Yes). These confusion matrices are pivotal for understanding not just the overall accuracy of the classifier, but also to give insights into the types of mistakes it is making (e.g., favoring false positives over false negatives or vice versa). It’s also important to evaluate how the classifier performs on both the training and testing sets to identify if the model is overfitting to the training data.

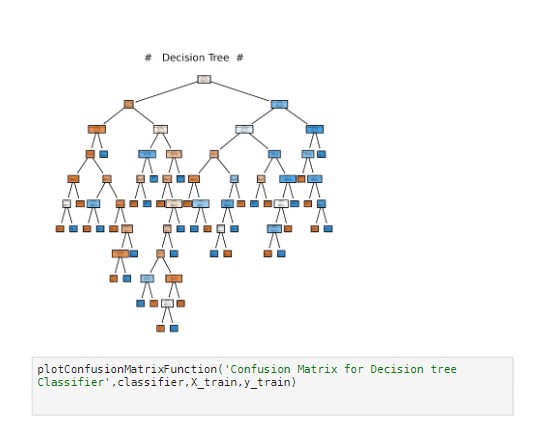
A graph showing a diagram of a patient's surface

Description automatically generated with medium confidence

**Plot 2.2**

**Plot 2.2** is a scatter plot showing the decision surface resulting from a classification algorithm applied to a dataset with two classes: individuals with heart disease (marked as red dots) and those without heart disease (marked as green dots). The data has been transformed using Principal Component Analysis (PCA), which is a dimensionality reduction technique. The axes represent the first two principal components, which are linear combinations of the original variables that capture the maximum variance in the data.

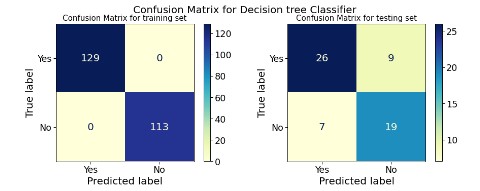
The regions marked in blue and light green show the decision boundary determined by the classification algorithm, where the boundary line is the division between the two regions. Each region predicts the class label for new instances based on where they would fall in this two-dimensional PCA space. If a new data point falls into the blue area, the model will predict it as a case without heart disease, and if it falls into the light green area, the model will predict it as a case with heart disease. This visualization helps in understanding how well the model might perform in separating the two classes, although it reveals some overlap between the classes, indicating potential misclassifications or complexity in the data that might not be fully captured by the first two principal components.



**Model 3.1**

**Model 3.1** is used in machine learning for classification and regression tasks. The tree-like structure represents decisions and their possible consequences, including chance event outcomes, resource costs, and utility.

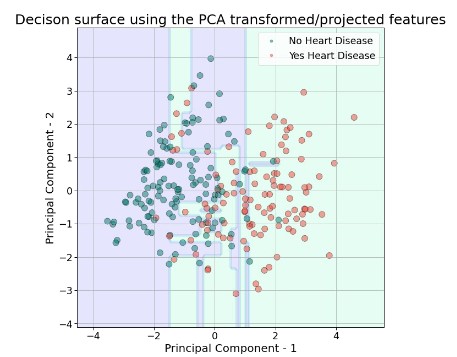
In the graphic, each node (represented by the small boxes) indicates a point where a decision must be split based on some information. Blue boxes indicate a decision to split further, while orange boxes suggest a terminal node, which is the outcome or the class prediction. The arrows suggest the flow of decisions (or data splits) from one condition to another. Below the Decision Tree, there is a line of code that is part of a script likely written in Python (given the syntax), which is calling a function `plotConfusionMatrixFunction`. This function seems to be designed to plot the confusion matrix for the Decision Tree Classifier using the training data `X\_train` and `y\_train`. The confusion matrix is a performance measurement tool for machine learning classification, where the output of a classifier is compared to the known labels and summarized in a matrix to show the model's accuracy – that is, its ability to correctly predict the class labels of new, unseen data.



**Matrix 1.4**

On the left, the confusion matrix (**Matrix 1.4**) for the training set shows that the model made correct predictions for all instances, as indicated by the numbers 113 for "No" predictions and 0 for "Yes" predictions along the diagonal. This indicates a perfect classification on the training set, with no false positives or false negatives.

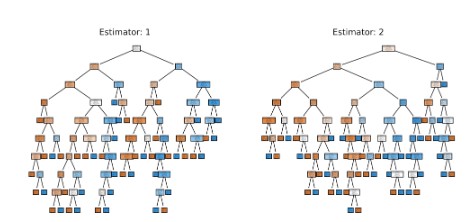
On the right, the confusion matrix (**Matrix 1.4**) for the testing set shows the model's performance on unseen data. The diagonal shows the number of correct predictions: 19 for "No" and 26 for "Yes". The off-diagonal numbers represent the misclassifications: 9 false positives (instances incorrectly labeled as "Yes") and 7 false negatives (instances incorrectly labeled as "No"). These values provide insights into the model's generalization capabilities and areas where it may be making errors.



**Plot 2.3**

**Plot 2.3** depicts a scatter plot illustrating the decision surface using PCA (Principal Component Analysis) transformed/projected features for a dataset concerning heart disease. The horizontal axis represents the first principal component (Principal Component 1), while the vertical axis represents the second principal component (Principal Component 2). These two principal components are linear combinations of the original dataset features that are constructed to capture the most variance within the dataset.

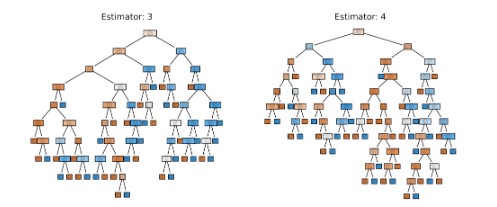
In the plot, you can see two distinct groups of points: one group is marked by blue dots, representing cases without heart disease, and the other group is marked by red dots, indicating cases with heart disease. The shaded areas seem to be decision boundaries that separate the two classes. Machine learning algorithms utilize such decision surfaces to classify new data points—based on where a new point falls on the plot, it can be predicted to belong to one of the categories (Yes Heart Disease or No Heart Disease). The goal of PCA in this context is to reduce the dimensionality of the data, simplifying the dataset while preserving as much of the relevant information as possible for the classification task.



**Model 3.2**

**Model 3.2** shows two diagrams labeled "Estimator 1" and "Estimator 2," which appear to be representations of decision trees. Decision trees are a type of predictive modeling algorithm used in statistics, data mining, and machine learning. They are typically used for classification and regression tasks, helping to predict the outcome of a variable by learning decision rules inferred from data features.

Each rectangular or square node in these trees represents a decision point based on certain criteria, and the paths from these nodes denote the outcome of each decision, leading to either another decision node or a terminal node (often represented by leaf-like symbols), which provides the predicted outcome. The different colors within the nodes may indicate various classes or value ranges of the predicted outcome depending on how the tree is structured. Estimator 1 and Estimator 2 might be two different models or two iterations of a model with varying parameters or rules, potentially as a part of an ensemble method like a random forest, where multiple trees contribute to a final prediction.



**Model 3.3**

**Model 3.3** depicts diagrams of two different decision trees, labeled as "Estimator: 3" and "Estimator: 4." These decision trees are likely part of an ensemble learning model, which is a machine learning approach that uses multiple learning algorithms to obtain better predictive performance than could be obtained from any of the constituent learning algorithms alone. Each "estimator" is a term commonly used to refer to the individual models within an ensemble. The trees in the given image are examples of such individual models.

In each tree, the topmost node, also known as the root node, represents the point from which the data splitting starts based on a chosen attribute or feature. As we progress down each tree, the nodes represent decision points where the dataset is further split based on other attributes. The branches connecting the nodes correspond to the outcome of the decision at that point, and the leaf nodes at the bottom (shown as rectangles) indicate the final decision or prediction made by the tree. The different colors of nodes typically represent classes or outcomes in classification problems, and the decision path followed from the root to a leaf node constitutes the decision rule associated with that particular outcome. Estimator 3 and Estimator 4 would make slightly different predictions based on their unique splits and decisions, contributing to the diversity of the overall ensemble prediction.

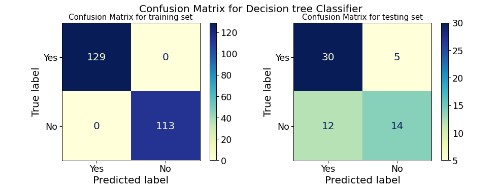
A diagram of a structure

Description automatically generated with medium confidence

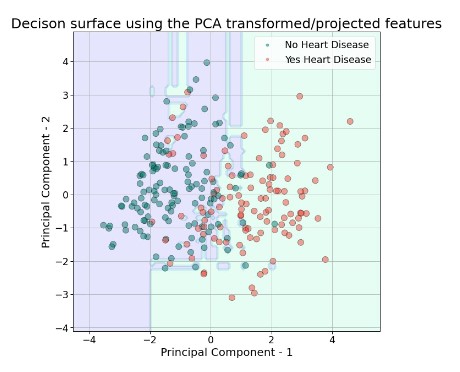
**Model 3.4**

**Model 3.4** depicts two decision trees, specifically Estimator 5 and Estimator 6. Decision trees are a type of machine learning model used for both regression and classification tasks. They work by splitting the data into subsets based on the value of input features, which in the context of these trees, are represented by the nodes (the rectangles). Each split is a decision made based on some criteria, typically the one that maximizes the information gain or minimizes a cost function like the Gini impurity or mean squared error, depending on whether the task at hand is classification or regression.

The squares at the bottom of the trees are the leaves of the decision trees, where the final decisions or predictions are made. For classification tasks, the leaf represents the majority class among the samples within its section of the feature space. For regression tasks, it typically represents the average of the target values. These two trees appear to be part of an ensemble method, perhaps a Random Forest or a form of Gradient Boosting, as indicated by the naming of 'estimator.' Ensemble methods like these combine the predictions of several base estimators to improve generalizability and robustness over a single estimator.

**Matrix 1.5**

* The matrix on the left (**Matrix 1.5**) indicates the performance of the classifier on the training set. In this matrix, there are 129 instances that were `Yes` and the model predicted all of them correctly as `Yes` since there are no instances in the off-diagonal cell. Similarly, there are 113 instances that were `No` and the model predicted all of them correctly as `No`. This matrix shows perfect classification on the training data; however, such a perfect score often raises concerns about overfitting, where a model learns the training data too well, including noise and outliers, which may not generalize well to new, unseen data.
* The matrix on the right (**Matrix 1.5)** shows how the model performed on the testing set, which is a set of data not seen by the model during training. This is a more realistic indication of how the model might perform on new data. Here, there were 35 instances labeled `Yes`, and the model correctly predicted 30 of them, while it incorrectly predicted 5 instances as `No` that were truly `Yes`. For the instances that were `No`, there were 26 in total, with the model correctly predicting 14 and incorrectly predicting 12 as `Yes`. These numbers indicate that while the model has learned some generalizable patterns, it's not as accurate on the testing set compared to the training set, which is common in machine learning applications.

**Plot 2.4**

**Plot 2.4** demonstrates a scatter plot visualizing a decision surface using features transformed or projected by Principal Component Analysis (PCA). In this context, the scatter plot is color-coded to represent two categories of a medical dataset: individuals with heart disease (marked in red) and individuals without heart disease (marked in greenish blue).

PCA is a statistical method commonly used to reduce the dimensionality of data by transforming the original variables into a new set of uncorrelated variables, called principal components. These principal components are ordered such that the first few retain most of the variation present in the original data. In the scatter plot, each point represents an individual patient, plotted according to their scores on the first two principal components. By projecting the data onto these components, it may become easier to visualize patterns and clusters, which, in this case, could potentially differentiate between patients with and without heart disease. This kind of visualization clarifies the distribution of data and the separation of classes for machine learning classification tasks.