Exploring the Potential of Analytical Models in Heart Disease Prediction

Hanan Saleh Al-Messabi
College of Technological Innovation
Zayed University,
Abu Dhabi, UAE.
M80008665@zu.ac.ae

Fatma Mohamed Al-Ali
College of Technological Innovation
Zayed University,
Abu Dhabi, UAE.
M80008676@zu.ac.ae

Feras Al-Obeidat
College of Technological Innovation
Zayed University,
Abu Dhabi, UAE.
Feras.Al-Obeidat@zu.ac.ae

***Abstract*— In 2021, the annual death toll due to various heart diseases reached a staggering 18 million individuals. This excessive mortality rate has become a pressing concern for scientists and medical professionals alike. Fortunately, the emergence of artificial intelligence has provided a valuable tool for decision-makers to tackle the challenges posed by heart disease. Consequently, numerous algorithms have been proposed to develop diverse models tailored to specific applications. By utilizing different analytical models, including logistic regression, decision trees, random forests, neural networks, and deep learning models, it has been determined that the logistic regression model achieves the highest and most favorable metric scores. With an impressive accuracy rate of 83%, a precision rate of 88%, and a recall rate of 86%, this model proves to be the most effective in predicting heart disease. Therefore, this study will significantly contribute to the advancement of healthcare practices by harnessing the power of big data and advanced analytical models. These insights will provide valuable guidance in addressing critical health issues in society in the future.**

***Keywords*— large data set, regression, analytical tools, heart disease, decision tree, machine learning, algorithms, neural network, deep learning.**

I. INTRODUCTION

Heart disease encompasses a range of conditions that impact the heart. Recently, cardiovascular diseases (CVDs) have been the primary cause of death worldwide, resulting in 17.9 million deaths each year, as reported by the World Health Organization [1]. Unhealthy behaviors such as obesity, high triglyceride levels, high cholesterol, and hypertension contribute to the increased risk of heart disease. The American Heart Association has compiled a list of specific symptoms to be aware of, including irregular heartbeat, sleep problems, swollen legs, and, in some cases, rapid weight gain of up to 1 to 2 kg per day [2]. These symptoms can be similar to those of other diseases, particularly those common among the elderly, making accurate diagnosis challenging and leading to a rise in mortality rates [3].

Over time, a plethora of research, data, and medical records from hospitals have become increasingly accessible. As a result, a substantial volume of data is being generated, serving as the driving force behind today's analytics applications. The advancement of big data technologies has unlocked a wealth of information for organizations, enabling them to process, manage, and analyze various types of data [4]. It is now widely recognized that machine learning and artificial intelligence play a pivotal role in the healthcare sector. Various machine learning and deep learning models can be utilized to diagnose, classify, or predict diseases. Nowadays, machine learning models can effortlessly perform comprehensive genomic data analysis while also being trained to support knowledge-based pandemic predictions. Additionally, medical records can undergo in-depth analysis to construct models that yield more accurate predictions [3].

Computer science heavily relies on data structures and algorithms, with a wide array of algorithms designed to achieve different objectives. These algorithms work with diverse data structures at the same computational complexity level. Understanding data structures is crucial for developing efficient algorithms [6], as many algorithms are dependent on specific data structures for optimal functionality [5]. The research discussed in this paper focuses on the application of machine learning and deep learning algorithms on a *Heart Disease* dataset, showcasing the superior predictive capabilities of logistic regression algorithms over other analytical models in both Spark and Weka environments. The research paper primarily focuses on investigating the efficiency of several machine learning algorithms and analytical tools in predicting heart diseases at an early stage. The main goal is to determine whether logistic regression is the most effective model for accurately detecting the possibility of developing heart disease.

The objective of the paper is to employ various algorithms on a dataset and compare the outcomes of each method. In this study, five machine learning algorithms for classification and clustering, namely Logistic Regression, Decision Tree, Random Forest, Neural Network, and Deep Learning models, were utilized. These algorithms were executed on both the Kaggle and Weka platforms to conduct multiple accuracy tests on the dataset. The rationale behind selecting these algorithms is their suitability for the dataset, as the dependent variable (outcome) is categorical rather than a constant value. In simpler terms, the dataset's outcome value falls into a yes-or-no category rather than being a continuous value.

Furthermore, these algorithms are widely recognized as sophisticated and robust instruments for identifying and forecasting diverse conditions. Their primary objective is to draw logical conclusions by uncovering and analyzing concealed patterns within a given dataset. Consequently, employing carefully chosen machine learning algorithms can prove invaluable in the prediction and categorization of individuals afflicted with heart diseases. Early detection of these indicators can potentially mitigate the exacerbation of symptoms and the emergence of further complications.

The outcomes of this study will make a substantial contribution to the progress of healthcare practices, showcasing the impact of transitioning towards a digital and societal transformation, especially in the healthcare sector. Additionally, this research provides valuable perspectives on the obstacles and possibilities for enhancing heart disease prediction, diagnosis, and prevention. By addressing a crucial health concern in society, the influence of digital transformation becomes evident, enabling the utilization of artificial intelligence to streamline processes more effectively and shape future concepts.

II. RELATED WORK

In earlier times, the analysis of collected data was relatively simpler due to its limited size and lack of diversity, owing to the slow pace of technological advancements. However, in our present dynamic world, data is being generated on a massive scale and in various formats, such as videos, pdfs, and spreadsheets. Compared to the past, the rapid emergence of technological advancements has made it easier to handle the large volumes of data generated on a daily basis. The significance of effectively managing this vast amount of data stems from its immense value across numerous fields. For instance, in the realm of manufacturing process management, large data sets play a crucial role in enhancing cost and operational performance by optimizing supply chains and resource allocation within organizations [7]. Similarly, in the healthcare sector, the utilization of large data sets aids decision-makers and patients alike in making more informed choices, leading to improved accuracy in diagnoses and disease prediction [8].

Nevertheless, extensive datasets require powerful analytical instruments to examine information gathered from diverse sources and interpret it. One of the most commonly utilized tools for this purpose is machine learning, a component of artificial intelligence that concentrates on utilizing data to gain insights into human thought processes in order to enhance their precision and efficiency [9]. There are numerous data analysis tools employed in machine learning in conjunction with large datasets. While Microsoft Excel serves as a user-friendly tool for this task, more sophisticated and up-to-date tools such as the R and Python languages, Matlab, and Spark are also available [10]. For instance, the Python language can be utilized to handle large datasets, enabling users to program and document data within an interactive environment like Jupyter Notebook, which comes equipped with pre-existing libraries for various fields, including machine learning [10]. Additionally, Apache Spark, another framework that utilizes Python, can be employed for machine learning to facilitate more precise and in-depth analysis of extensive datasets.

These tools include a wide range of models for data analysis, such as regression and decision tree approaches. As a result, learning and understanding how to apply existing analytical models is critical for their effectiveness in a variety of applications. For example, the regression analysis approach aids in forecasting event occurrences. Furthermore, it identifies the elements that are more potent in comparison to others and have a greater impact on results, allowing for the identification of the relationship between key components and outcomes. As a result, it would be simpler to completely comprehend the situation and draw an informed conclusion [11]. Some of the most prominent regression approaches are linear regression, logistic regression, stepwise regression, and LASSO regression. Random forest is another analytical model that, like decision trees, may do classification and regression analysis depending on the issue [11].

Several methods are utilized in conjunction with the heart disease dataset. Designing classification algorithms has long been a critical topic of research in machine learning and pattern recognition. Supervised machine learning methods include the well-known linear and logistic regression classifiers [11]. There are several distinctions between the two classifiers. In machine learning, linear regression is a predictive modeling approach. The model predicts values based on independent factors and helps identify the link between dependent and independent variables. Logistic regression, on the other hand, is used to divide elements of a collection into two groups (binary classification) by computing the likelihood of each member in the set. In other words, logistic regression is the appropriate regression technique to use when the dependent variable has a binary outcome [12].

Among the supervised machine learning algorithms, the decision tree is another notable technique. It utilizes frequency tables to make predictions and is capable of handling both categorical and numerical variables [11]. The decision tree algorithm constructs a tree structure where the internal nodes represent the dataset's features, the branches depict the decision rules, and the leaf nodes signify the final outcome and prediction [13]. As the algorithm proceeds, it systematically breaks down the dataset into smaller subsets while simultaneously developing the associated decision tree. Generally, frequency tables are employed to make predictions within this model [13].

In addition, there is a crucial algorithm referred to as Artificial Neural Networks (ANN) or neural networks. It is a fundamental element of machine learning and serves as the foundation of the deep learning methodology. ANN is a computational structure comprised of numerous neurons, which are mathematically illustrated to connect entities in the physical world, functioning similarly to the biological nervous system in detecting and understanding patterns within the data [14]. Neural networks enable computers to make intelligent decisions independently with minimal human intervention. This is possible as they can learn and replicate the complex and nonlinear relationships between input and output data [15].

Deep learning refers to the creation of learning algorithms that have the ability to train and make predictions based on complex data. The term "deep" in deep learning refers to the number of layers in a neural network. Specifically, it involves neural networks that consist of more than three layers, including the input and output layers [16]. In the field of healthcare, significant efforts have been made to develop a system for the early detection of heart disease using various clinical principles. Among the algorithmic methods employed, neural networks and deep learning play a crucial role in identifying and predicting patients with heart disease. For example, a study [17] proposed a strategy for predicting cardiac disease using an ANN. The authors utilized a self-administered questionnaire and trained the neural network with a backpropagation algorithm, which consisted of three hidden layers. The architecture of the network was validated using the Dundee rank factor score and achieved a high relative operating characteristic value (98%) on their dataset. Overall, deep learning and neural networks offer valuable tools for diagnosing and predicting heart disease, contributing to early identification and improved patient outcomes.

According to [17], a further advancement was made through the proposal of a deep neural network and a statistical model for feature selection. The authors of this study employed multiple strategies to mitigate overfitting and underfitting, resulting in a 94% accuracy rate and 93% sensitivity. Additionally, they investigated the effectiveness of ANNs with different quantities of hidden layers, achieving close to 95.5% accuracy with five hidden layers. Furthermore, they proposed an ANN with a significant number of neurons in the hidden layer that utilizes *the radial basis function.* In general, they achieved approximately 97% accuracy with this setup.

The integration of these models, along with technological advancements, has the potential to aid in the prediction of various health issues, thereby assisting in the prevention, treatment, and even potential cure of numerous diseases. Heart disease has been a prominent cause of death globally in recent years, accounting for approximately 16% of total deaths, surpassing other factors such as stroke, chronic illnesses, cancer, and diabetes [18]. Therefore, it is imperative to address and research this critical issue in order to reduce mortality rates and subsequently enhance life expectancy. A study conducted in the United Arab Emirates (UAE) revealed that cardiovascular diseases are the primary cause of death in the region, largely attributed to factors such as the unhealthy lifestyle practices of UAE residents [19]. The study also indicates that unhealthy habits like smoking and obesity contribute to elevated levels of blood pressure and cholesterol, consequently heightening the risk of developing heart disease in the future.

By employing various analytical models, this paper demonstrates the application of a heart disease dataset to gain a better understanding of the issue at hand.

III.PROPOSED METHODOLOGY

This section provides a deeper insight into the dataset employed, any modifications made to the attributes, the algorithm utilized, and the equations provided to offer a clearer understanding.

A. Data Extraction and Transformation

Within this study, the dataset employed for application and analysis is referred to as *Heart Disease Classification*, comprising 14 columns and 303 rows. Of these columns, 13 serve as independent attributes that impact the output attribute, known as the target. The variables influencing the output attribute and their respective terms in this investigation are detailed below:

1. Individual's age (age).
2. Individual's sex (sex).
3. Type of chest pain experienced (chest_pain_type).
4. Blood pressure reading at rest (resting_bp).
5. Level of cholesterol in an individual (cholestoral).
6. Fasting blood sugar level measurement (fasting_blood_sugar).
7. Resting ECG results (restecg).
8. Maximum heart rate achieved by an individual (max_hr).
9. The presence of exercise-induced angina (exang).
10. ST depression induced by exercise relative to rest (oldpeak).
11. Slope of the ST segment on the ECG (slope).
12. Number of major vessels colored by fluoroscopy (num_major_vessels).
13. Degree of thalassemia present (thal).

Various research studies have highlighted the significance of understanding certain features to predict and prevent heart disease in patients promptly [20]. As individuals age, the risk of experiencing a heart attack, for instance, tends to rise [21]. Moreover, research indicates that men are at a higher risk of developing heart disease [22]. Hence, it is crucial to analyze the impact and interplay of these factors with other variables on the probability of developing heart disease.

Further elaboration on additional characteristics reveals that an electrocardiogram (ECG) is utilized to assess the electrical activity of the heart during a state of rest without any movement. The old peak refers to the measurement of ST depression that occurs as a result of exercise in comparison to the resting state, taking into account its slope value.

The attributes mentioned above are labeled as 'age','sex','chest_pain_type','resting_bp', 'cholestoral','fasting_blood_sugar','restecg','max_hr', 'exang','oldpeak','slope', 'num_major_vessels', 'thal', and 'target' in the subsequent sections.

When it comes to data transformation, no conversion is required as all values in the dataset are numerical, with 12 being integers and one attribute being decimals.

B. The Algorithms, diagrams, and flowcharts used.

In this paper, advanced algorithms are harnessed to predict future outcomes and conduct comparisons to evaluate their efficiency. As outlined below, the algorithms employed encompass Logistic Regression, Decision Tree, Random Forest, Neural Networks, and Deep Learning.

Logistic Regression

A logistic regression is a statistical modeling technique utilized for classification and predictive analytics. It provides a binary classification outcome for the output variable. By analyzing a dataset of independent variables, logistic regression calculates the probability of an event occurring, such as determining whether an email is spam or not. As the result is a probability, the dependent variable is constrained within the range of 0 and 1 [23].

Logistic regression plays a crucial role in artificial intelligence (AI) and machine learning (ML). ML models, created through logistic regression, enable users to automate intricate data processing tasks without manual interference. These models empower companies to extract valuable insights from their data, which can be utilized for predictive analysis to cut down on operational expenses, boost productivity, and accelerate growth. For example, businesses can identify patterns that improve employee retention or drive more profitable product development [23].

Logistic regression offers several advantages and benefits compared to other machine learning techniques. Firstly, it is known for its simplicity, making it easier to understand and implement. Additionally, logistic regression excels at processing large volumes of data quickly, enabling efficient analysis. Moreover, it provides developers with enhanced visibility into internal software processes, surpassing other data analysis techniques [23].

Furthermore, logistic regression is widely utilized in a multitude of industries for various practical purposes. In the healthcare sector, medical professionals employ logistic regression models to anticipate the likelihood of diseases in patients, enabling them to plan preventive care and treatment accordingly. By comparing the impact of family history or genetic factors on diseases, researchers can gain valuable insights. Similarly, the financial industry can leverage logistic regression to its advantage. Financial institutions, for instance, rely on this technique to detect fraudulent activities by analyzing financial transactions. Additionally, logistic regression aids in evaluating loan and insurance applications for risk assessment. These scenarios are well-suited for logistic regression models due to their distinct outcomes, such as high or low risk and fraudulent or non-fraudulent activities [23].

A comprehensive understanding of basic regression analysis is essential to comprehending logistic regression. The first step in any data analysis is to establish a business question. In the case of logistic regression, it is crucial to frame the question in a manner that produces specific outcomes. After identifying the question, it is necessary to identify the relevant data factors. Subsequently, past data for all factors must be collected and processed using regression software. The software will mathematically connect the diverse data points through equations. By utilizing the logistic regression equation, the software can predict unknown values [23].

Equations in mathematics establish the connection between two variables, namely x and y. By illustrating the correlation between various x and y values, these functions or equations generate a graph on the x-axis and y-axis [23].

Within statistics, variables represent the diverse factors or characteristics of data with varying values. Specific variables act as independent or explanatory factors in analyses, potentially affecting the outcome. Conversely, there are dependent variables whose values are reliant on the independent variables. Logistic regression is a method used to explore and evaluate the influence of independent variables on a dependent variable by examining the historical data values of both [23].

Furthermore, logistic regression is a statistical model that employs the logistic function, also known as the logit function, represented mathematically as the equation linking x and y [24]. Hereafter, the primary logistic equation is presented for your reference:

$$Y = \frac{e^{(b_0+b_1X)}}{1+e^{(b_0+b_1X)}}$$

where,

- Y = predicted output
- X = input value
- b_0 = bias or intercept term
- b_1 = coefficient for input (X)

The equation provided here shares similarities with the linear regression formula, as it involves combining input values in a linear manner to predict an output value using weights or coefficient values. However, unlike linear regression, the output value demonstrated in this instance is binary, taking on either the value of 0 or 1, rather than a numerical value [24].

In other words, the primary objective of logistic regression is to identify an optimal model that accurately characterizes the output variable as either 0 (negative class) or 1 (positive class). Logistic regression, which is a specialized form of linear regression, calculates the coefficients of a formula to forecast the likelihood of the dependent variable occurring. Consequently, it selects the parameters that increase or decrease the probability of the dependent event transpiring while acknowledging that the probability of an event falls within the range of 0 to 1. Conversely, the linear regression model does not ensure this probability range [11].

Two crucial factors to take into account when assessing the likelihood of the suggested binary result of the independent variable are as follows: First, it must be positive, which can be achieved by utilizing an exponential function, and secondly, the probability must not exceed one, which can be accomplished by dividing the outcome by the sum of the outcome and one [11].

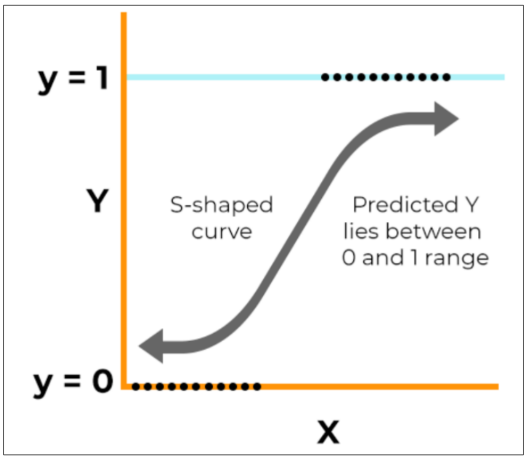


Fig.1. Sigmoid Function of logistic regression [24].

The S-curve displayed in Figure 1 represents the graphical representation of the logistic regression equation. It is worth noting that the logit function exclusively generates values ranging from 0 to 1 for the dependent variable, regardless of the independent variable's values. This characteristic is pivotal to logistic regression's predictive capability for the dependent variable's value [24].

Logistic Regression Flowchart

Within Figure 2, there is a depiction of the flow diagram for the logistic regression model, which controls the independent and dependent variables. This model employs the sigmoid function to predict probabilities and define decision boundaries [25].

Logistic regression consists of two distinct phases, namely forward propagation and backward propagation. In the initial stage of forward propagation, the weights are multiplied by the features. As the weights are initially unknown, random values can be assigned to them. Subsequently, a sigmoid function is employed to allocate a probability ranging from 0 to 1. The prediction is then made based on this probability, considering a specified threshold value. Following this, the predicted value is compared to the observed and detected values, leading to the creation of a loss function.

Furthermore, the loss function determines the distance between the predicted value and the actual value. In cases where the loss function yields a significantly high value, backward propagation is implemented. The primary objective of backward propagation is to optimize the weights by utilizing the cost function, which involves calculating derivatives [24].

Sigmoid functions play a crucial role in logistic regression models by transforming real values into a range between 0 and 1. These mathematical functions exhibit a characteristic S-shaped curve and include popular variations like the logistic function, the hyperbolic tangent, and the arctangent. Figure 3 illustrates these common sigmoid functions [26], with the logistic function being commonly referred to as the sigmoid function in machine learning.

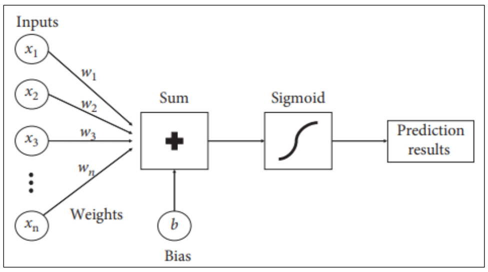


Fig.2. Logistic regression flowchart [25].

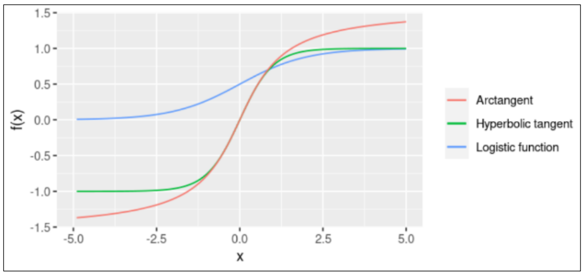


Fig.3. Common Sigmoid functions [26].

All sigmoid functions share the characteristic of mapping the entire number line into a limited range, such as 0 to 1 or -1 to 1. This property enables sigmoid functions to convert real values into probabilities that can be easily interpreted [26]. Among the various sigmoid functions, the logistic sigmoid function is particularly popular, as it transforms any real-valued input into a value between 0 and 1 [26].

The logistic sigmoid function is defined in the following way:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Decision Tree

Decision tree methodology serves as a widely utilized and powerful tool for data mining endeavors, facilitating the establishment of classification systems based on multiple covariates or the development of prediction algorithms for a specific target variable. The decision tree itself takes the form of a flowchart resembling a tree structure, featuring root nodes, internal nodes, and leaf nodes. This non-parametric algorithm is well-equipped to handle extensive, intricate datasets without the need for a complex parametric structure. In cases where the sample size is sufficiently large, the data can be partitioned into training and validation/testing datasets. The training dataset is employed for constructing a decision tree model, while the validation dataset assists in determining the optimal tree size for achieving the best final model.

In the field of medical research, decision tree methodology has become increasingly prevalent. A notable example is its use in diagnosing medical conditions by analyzing symptom patterns. The decision tree enables the classification of conditions into distinct clinical subtypes or identifies patients who necessitate diverse treatments.

Decision tree models come in different types depending on their application, including categorical variable decision trees and continuous variable decision trees. Moreover, machine learning employs two types of decision tree algorithms: classification trees and regression trees. These algorithms are part of machine learning methodologies and are utilized for developing prediction models from specific datasets. In particular, the regression tree algorithm is effective when dealing with continuous or numeric response variables, as opposed to categorical ones.

Moreover, the tree is employed for predicting target values. Regression trees are applicable for datasets with quantitative data such as temperature and price [27]. In contrast, classification trees are utilized when the target variable is categorical, aiding in identifying the class where the target variable is most likely to be categorized. Classification trees are useful for dividing the response variable into mainly two classes: *Yes* or *No* [27]. For instance, they can be used to predict which students will or will not graduate from high school [28].

There are several typical applications of decision tree models, which are as follows [29]:

1. Variable selection: The number of variables that are routinely monitored in clinical settings has increased intensely with the introduction of electronic data storage. Many of these variables are not crucial and relevant and, thus, should probably not be included in data mining exercises. Like stepwise variable selection in regression analysis, decision tree methods can be used to select the most appropriate input variables that should be used to form decision tree models, which can subsequently be used to formulate clinical hypotheses and update following research.

2. Evaluating the comparative importance of variables: Once a set of relevant variables is identified, researchers may want to know which variables play key roles. One way to compute variable importance is through model accuracy reduction if a variable is removed. In most situations, the more records a variable influences, the greater the importance of the variable will be.

3. Handling missing values: A common method of handling missing data is to exclude cases with missing values. This is ineffective and runs the risk of introducing bias in the analysis. Decision tree analysis can deal with missing data in two ways. Firstly, it can classify missing values as a separate category that can be analyzed with the other categories. Secondly, it can use a built-in decision tree model that sets the variable with many missing values as a target variable to make a prediction and replace these missing ones with the predicted one.

4. Prediction: This is one of the most significant usages of decision tree models. Using the tree model derived from historical data, it is easy to predict the results in future analyses.

5. Data manipulation: Too many categories of one categorical variable or heavily continuous data are common in medical research. In these conditions, decision tree models can help in deciding how to best collapse and breakdown categorical variables into a more manageable number of categories or how to split the heavy variables into series.

A decision tree serves as a classification technique that repetitively segments a dataset into smaller subdivisions by applying a series of tests at each branch or node of the tree (see Figure 4). The tree comprises a root node at the top, internal nodes for splits, and terminal nodes for leaves. Each node in a decision tree has a single parent node and two or more descendant nodes. In this process, the dataset is classified by progressively dividing it based on the decision framework established by the tree, assigning a class label to each observation according to the leaf node it is associated with.

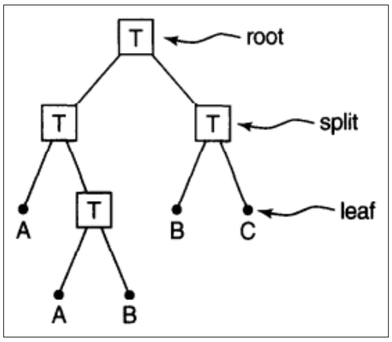


Fig.4. A decision tree classifier [30].

At every node, denoted as a box, tests (represented as T) are conducted to divide the data into progressively smaller groups. The class labels (A, B, and C) assigned to each observation are associated with the leaf nodes [30].

How decision tree works

There are multiple approaches to choosing the optimal attribute at each node in decision tree models. Among these approaches, information gain and Gini impurity serve as widely used splitting criteria. Both methods aid in evaluating the effectiveness of test conditions and their ability to classify samples into specific classes. Entropy, a metric derived from information theory, quantifies the impurity of sample values. It is mathematically defined by the formula [31]:

$$Entropy(S) = -\sum_{c\epsilon C} p(c) log_2\, p(c)$$

where,

- S is the data set that entropy is calculated
- c is the classes in set S
- p(c) is the proportion of data points that belong to class c to the number of total data points in set S.

Entropy values can vary from 0 to 1. When all samples in a data set, S, are assigned to a single class, the entropy will be zero. Conversely, if half of the samples are classified as one class and the other half as another class, the entropy will reach its maximum value of 1. In order to select the optimal feature and discover the most suitable decision tree, the attribute with the lowest entropy should be utilized. Information gain measures the difference in entropy before and after a split based on a specific attribute. The attribute with the highest information gain will result in the most effective split, accurately classifying the training data based on its target classification. Information gain is commonly represented by the formula [31]:

$$Information\ Gain\ (S,a) = Entropy\ (S) - \sum_{V \in Vcalues(a)} \frac{|S_v|}{|S|} Entropy(S_v)$$

where,

- a is a specific attribute or class label
- *Entropy(S)* is the entropy of dataset S
- |Sv|/|S| is the proportion of the values in S_v to the number of values in dataset S
- *Entropy* (S_v) is the entropy of dataset S_v

Gini impurity quantifies the chance of misclassifying random data points in a dataset when labels are assigned based on the dataset's class distribution. If a set S is pure (i.e., it consists of only one class), then its impurity is zero. The formula for Gini impurity is given by [31]:

$$Gini\ Impurity = 1 - \sum_i (p_i)^2$$

In order to ensure a successful machine learning model, it is crucial to achieve a 'good fit'. This entails finding the right balance between underfitting and overfitting. Therefore, it is important to consider the metrics of classification evaluation, such as precision and recall [32]. Precision can be defined as the proportion of relevant instances among all retrieved instances. In the context of our problem statement, precision would measure the number of patients correctly classified as having a heart disease out of all the patients classified as positive. Mathematically, this can be represented as follows [32]:

$$Precision = \frac{True\ Positives\ (TP)}{True\ Positives\ (TP) + False\ Positives\ (FP)}$$

In contrast, the recall signifies the ratio of retrieved instances compared to all relevant instances. Another mathematical representation of this concept is provided as follows [32]:

$$Recall = \frac{TP}{TP + False\ Negatives\ (FN)}$$

Both recall and precision metrics are valuable for conducting an analysis. In this paper, you will discover the practical application of both metric tools as well as a comprehensive examination of the accuracy score for each model. The evaluation metric, which will be thoroughly analyzed and discussed, is the focal point of this research.

Decision trees offer numerous benefits over conventional supervised classification methods, like maximum likelihood classification in remote sensing. They are nonparametric, eliminating the need for assumptions or extensive computation on input data distributions [30]. Moreover, decision trees can effectively manage nonlinear relationships between features and classes, accommodate missing values, address multi-output issues, and process both numeric and categorical inputs in a visually intuitive manner.

Decision trees provide a clear indication of the most important fields or variables for prediction or classification. Additionally, they are capable of managing large datasets and can be parallelized for improved processing time. It is worth noting that decision trees are appealing due to their explicit classification structure, which is both computationally efficient to construct and easy to interpret [33].

Despite its advantages, the decision tree method has its weaknesses. Complex decision trees are prone to overfitting and struggle to generalize the data effectively, a phenomenon known as overfitting. To address this issue, various techniques can be employed, such as simplifying the tree, setting the minimum number of samples required at a leaf node, or limiting the maximum depth of the tree. Furthermore, decision trees can be sensitive to minor data variations, resulting in significantly different trees. Additionally, Scikit-learn, a popular machine learning library in Python, does not fully support decision trees. Although it includes a decision tree module, it lacks support for categorical variables. Lastly, training a decision tree model can be computationally intensive compared to other algorithms [34].

Random Forest

The research also utilizes the Random Forest algorithm, which is widely favored among data scientists. This supervised machine learning algorithm is commonly applied in classification and regression tasks. By constructing decision trees on various samples and aggregating their majority votes for classification, as well as calculating averages in regression models, Random Forest proves to be a versatile tool. Notably, it excels at handling datasets with both continuous and categorical variables, making it suitable for various classification and regression tasks [35].

Random forest is an ensemble learning technique that consists of multiple individual decision trees. The concept of ensemble involves combining several models together. Instead of relying on a single model, a collection of models is utilized to make predictions [35]. Ensemble learning employs two methods: bagging and boosting. Bagging is a meta-algorithm in machine learning that aims to enhance the stability and accuracy of statistical classification and regression algorithms [36]. This is achieved by randomly sampling a replacement from the original dataset. Each element in the bagging process has an equal probability of appearing in a new dataset. In the case of random forest, bagging is used when decision tree models with higher variance are present. Additionally, random feature selection is employed to grow the trees. A random forest is formed by combining several random trees [37].

Nevertheless, the boosting technique aims to construct a robust classifier by combining multiple weak classifiers. This is achieved by sequentially building a model using weak models. The final model, such as ADA BOOST and

XG BOOST, exhibits the highest accuracy [35]. Increasing the number of trees in the forest results in improved accuracy and helps avoid overfitting issues [38]. The diagram provided below illustrates the functionality of the random forest algorithm:

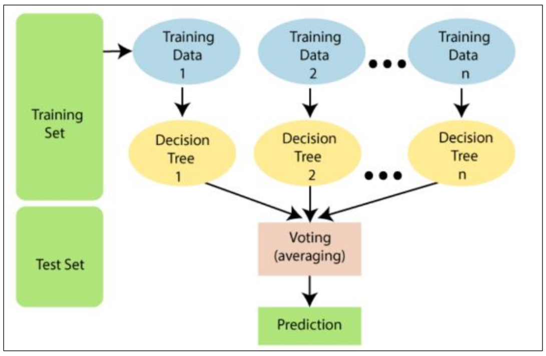


Fig.5. The random forest algorithm process [38].

There are certain steps involved in random forest algorithms. Initially, a subgroup of data points and a subgroup of features are selected for building each decision tree. Simply put, *n* random records and *m* features are taken from the dataset with *k* records. Secondly, individual decision trees are constructed for each sample. Then, each decision tree will generate an output. Lastly, final output is considered based on majority voting or averaging for classification and regression, respectively [38].

When utilized for classification or regression problems, the random forest algorithm presents several significant advantages and challenges. A key advantage of this approach is its capacity to address the issue of overfitting. To elaborate, by incorporating a large number of decision trees in a random forest, the classifier avoids overfitting the model, as the averaging of uncorrelated trees effectively reduces both the overall variance and prediction error.

In addition, random forests offer flexibility by predicting missing values while maintaining accuracy even with incomplete data. Stability is ensured through majority voting or averaging, leading to consistent outcomes. The algorithm also simplifies the assessment of variable importance and contribution to the model. Diversity is a crucial characteristic of random forests, where not all attributes are considered when building individual trees, resulting in diverse trees. Each tree in the random forest randomly selects a subset of features at the node's splitting point [39].

On the other hand, the random forest method presents certain obstacles. Despite its ability to effectively handle large datasets and generate precise predictions, this model tends to slow down when processing extensive data due to the computation required for each individual decision tree. Additionally, utilizing a random forest necessitates a greater allocation of resources for computation and data storage [39].

The random forest algorithm has been implemented across diverse industries, enabling organizations to enhance their decision-making processes and improve operational efficiency. In the finance sector, random forest is a preferred choice due to its effectiveness in minimizing the time required for data management and preprocessing tasks. It is commonly utilized for evaluating high-risk credit card customers, detecting fraud, and addressing pricing issues.

Moreover, health professionals within the healthcare sector have the opportunity to employ this algorithm. It can be integrated into computational biology, empowering doctors to tackle challenges such as gene expression classification, sequence annotation, and biomarker discovery. As a result, doctors can forecast drug responses to specific medications [39]. Despite being comprised of decision trees, random forests exhibit distinct behaviors. The following table delineates the primary differences between the two algorithms.

TABLE I: COMPARISON BETWEEN DECISION TREE AND RANDOM FOREST MODELS [35].

Decision trees	Random Forest
Decision trees normally encounter an overfitting problem if they are allowed to grow without any control.	Random forests are created from subsections of data, and the final output is based on average or majority ranking; hence, the problem of overfitting is taken care of.
A single decision tree is quicker to compute.	It is relatively slower.
When a data set with features is taken as input by a decision tree, it will frame some rules to make predictions.	It does not use any set of formulas since it randomly selects observations, builds a decision tree, and takes the average result.

Random Forest Flowchart

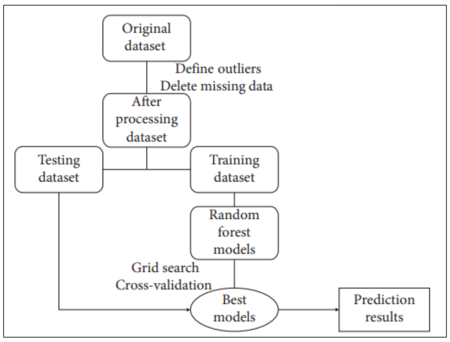


Fig.6. Random Forest Flowchart [40].

In Figure 6, the flow diagram illustrates the random forest model, which comprises a group of decision trees. The model preprocesses the data and randomly selects samples from the dataset for training. For each selected sample, a decision tree is created within the random forest model, which is trained without fine-tuning. Subsequently, grid search is conducted with 5-fold cross-validation and various parameter combinations, including the number of trees in the random forest. Additionally, the model determines the optimal function for feature numbers at each split, the number of levels in the tree, and the method of selecting samples for training each tree. Both the Gini criterion and the entropy criterion are utilized to evaluate the quality of the tree and the accuracy of the model [40].

Neural Network & Deep learning

Furthermore, apart from the previously mentioned models, there are two groundbreaking algorithms employed to enhance the precision of daily predicaments: neural networks and deep learning. Essentially, a neural network comprises three components: an input layer, hidden layers, and an output layer. On the other hand, deep learning encompasses a collection of neural networks that collaborate in a manner that efficiently handles vast quantities of data [41] [42]. Neural networks aid in the identification of concealed patterns and connections within data. This approach is widely utilized in various applications, such as financial forecasting, analysis of user behavior, and disease mapping [41].

Deep learning is often seen as more challenging and intricate than other methods, such as decision trees and random forests, due to the involvement of multiple neural networks. Despite this complexity, deep learning is essential for handling large datasets, which is a necessity for data scientists and specialists [42]. Multilayer perceptrons, a popular deep learning technique, are commonly used on social media platforms like Instagram for tasks such as image data compression and classification problem-solving [42].

While there are several models that can be utilized for the heart disease dataset, certain methods, such as linear regression and natural language processing (NLP), are not suitable. Linear regression is an analytical approach that employs algorithms to demonstrate the correlation between a dependent variable and independent variables. Its objective is to forecast future events or outcomes [43]. This regression model is typically employed with categorical or continuous variables. Consequently, it becomes challenging to apply this model to the heart disease dataset due to the binary nature of the output.

NLP, or natural language processing, is a significant aspect of computer science that involves training computers to comprehend and analyze text and spoken words. This method serves various purposes, including summarizing text rapidly, responding to spoken commands, and translating text into multiple languages [44]. NLP finds applications in digital assistants and consumer services, enhancing business operations, productivity, and the management of crucial processes. However, similar to linear regression, NLP cannot be applied to the heart disease dataset due to the absence of a text variable [44].

IV. APPLICATION

The study was conducted using an accessible database focusing on heart disease. The dataset consists of 14 attributes and 303 records, which were split into a training set (80%) and a testing set (20%). Two data mining tools—the Spark environment in Kaggle and Weka 3.8.6 software—were utilized for this purpose. Various analytical models, such as regression with multilayer perceptron using deep learning, multilayer perceptron using a neural network with backpropagation, and multinomial logistic regression, were employed as training algorithms. Detailed information on the codes used, coding environment, data visualization, statistical analysis, and findings of the analysis can be found.

A. *Code used and environment*

The software environment utilized in this document is referred to as *Apache Spark*. Essentially, it is a platform that offers APIs designed for handling extensive data sets across distributed datasets, along with a wide array of libraries and tools to enhance code productivity [11]. As illustrated in Figure 7, the Spark modeling model comprises three main processes: initiating Spark applications through the driver, executing designated tasks via executors, and managing resource allocation through the cluster manager [11].

In order to take advantage of this environment, users make use of an online platform called Kaggle. This platform allows users to create models and work with datasets. As shown in Figure 8, the initial code is used to install the Spark session and download all the necessary packages and libraries for the subsequent analysis. Additional libraries are imported to perform statistical operations such as plotting, vector assembly, confusion matrix, and more when applying the three models: logistic regression, decision tree, random forest, ANN, and deep learning models.

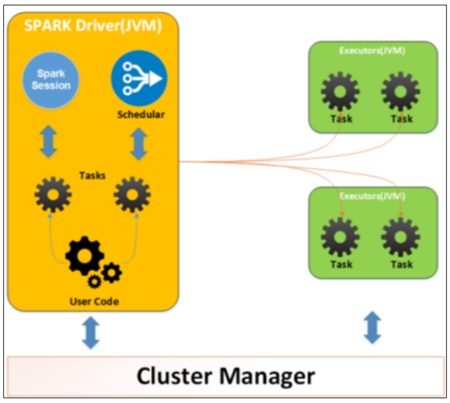
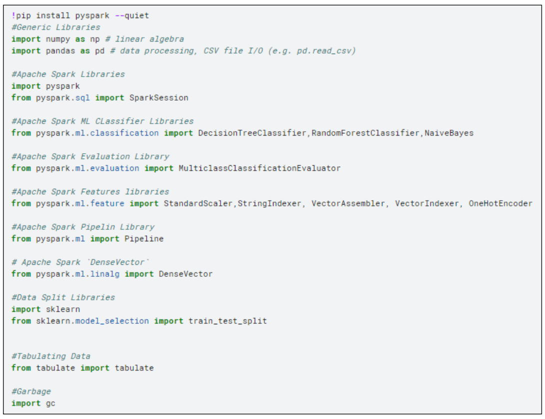


Fig.7. Spark programming model [11].



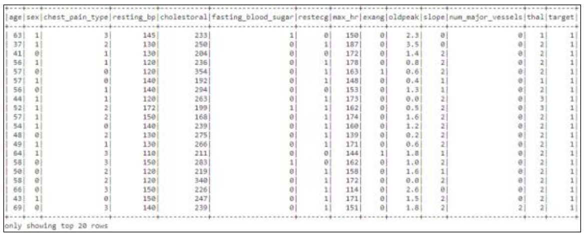
```
!pip install pyspark --quiet
#Generic Libraries
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

#Apache Spark Libraries
import pyspark
from pyspark.sql import SparkSession

#Apache Spark ML CLassifier Libraries
from pyspark.ml.classification import DecisionTreeClassifier,RandomForestClassifier,NaiveBayes

#Apache Spark Evaluation Library
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

#Apache Spark Features libraries
from pyspark.ml.feature import StandardScaler,StringIndexer, VectorAssembler, VectorIndexer, OneHotEncoder

#Apache Spark Pipelin Library
from pyspark.ml import Pipeline

# Apache Spark `DenseVector`
from pyspark.ml.linalg import DenseVector

#Data Split Libraries
import sklearn
from sklearn.model_selection import train_test_split


#Tabulating Data
from tabulate import tabulate

#Garbage
import gc
```

Fig.8. Installing Spark and libraries code.

Additionally, Weka is a user-friendly and no-cost data mining tool equipped with a collection of Java libraries that facilitate the creation of machine learning algorithms and classifiers. The acronym Weka stands for Waikato Environment for Knowledge Analysis [45]. During the execution of models in Weka, the *10-fold cross-validation* technique is employed. This technique involves randomly dividing the dataset into 10 portions and repeating this process 10 times, with one portion designated for testing and the remaining portions utilized for training [46].

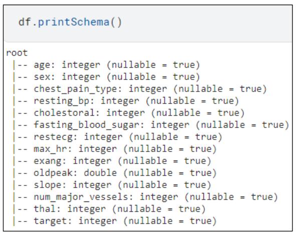
B. Data Visualization and distribution

After uploading the dataset and starting a new Spark session, you can see a table of the first 20 rows in Figure 9. To ensure that all values are numerical, the printed schema shows that all types are either integers or decimals without any string or text values. From there, there is no need to do any conversion (see Figure 10).



```
+---+---+---------------+----------+-----------+-------------------+-------+------+-----+-------+-----+-----------------+----+------+
|age|sex|chest_pain_type|resting_bp|cholestoral|fasting_blood_sugar|restecg|max_hr|exang|oldpeak|slope|num_major_vessels|thal|target|
+---+---+---------------+----------+-----------+-------------------+-------+------+-----+-------+-----+-----------------+----+------+
| 63|  1|              3|       145|        233|                  1|      0|   150|    0|    2.3|    0|                0|   1|     1|
| 37|  1|              2|       130|        250|                  0|      1|   187|    0|    3.5|    0|                0|   2|     1|
| 41|  0|              1|       130|        204|                  0|      0|   172|    0|    1.4|    2|                0|   2|     1|
| 56|  1|              1|       120|        236|                  0|      1|   178|    0|    0.8|    2|                0|   2|     1|
| 57|  0|              0|       120|        354|                  0|      1|   163|    1|    0.6|    2|                0|   2|     1|
| 57|  1|              0|       140|        192|                  0|      1|   148|    0|    0.4|    1|                0|   1|     1|
| 56|  0|              1|       140|        294|                  0|      0|   153|    0|    1.3|    1|                0|   2|     1|
| 44|  1|              1|       120|        263|                  0|      1|   173|    0|    0.0|    2|                0|   3|     1|
| 52|  1|              2|       172|        199|                  1|      1|   162|    0|    0.5|    2|                0|   3|     1|
| 57|  1|              2|       150|        168|                  0|      1|   174|    0|    1.6|    2|                0|   2|     1|
| 54|  1|              0|       140|        239|                  0|      1|   160|    0|    1.2|    2|                0|   2|     1|
| 48|  0|              2|       130|        275|                  0|      1|   139|    0|    0.2|    2|                0|   2|     1|
| 49|  1|              1|       130|        266|                  0|      1|   171|    0|    0.6|    2|                0|   2|     1|
| 64|  1|              3|       110|        211|                  0|      0|   144|    1|    1.8|    1|                0|   2|     1|
| 58|  0|              3|       150|        283|                  1|      0|   162|    0|    1.0|    2|                0|   2|     1|
| 50|  0|              2|       120|        219|                  0|      1|   158|    0|    1.6|    1|                0|   2|     1|
| 58|  0|              2|       120|        340|                  0|      1|   172|    0|    0.0|    2|                0|   2|     1|
| 66|  0|              3|       150|        226|                  0|      1|   114|    0|    2.6|    0|                0|   2|     1|
| 43|  1|              0|       150|        247|                  0|      1|   171|    0|    1.5|    2|                0|   2|     1|
| 69|  0|              3|       140|        239|                  0|      1|   151|    0|    1.8|    2|                2|   2|     1|
+---+---+---------------+----------+-----------+-------------------+-------+------+-----+-------+-----+-----------------+----+------+
only showing top 20 rows
```

Fig.9. Heart disease dataset values – showing top 20 rows.

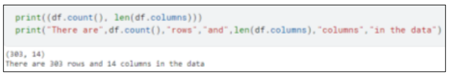


```
df.printSchema()
```

```
root
 |-- age: integer (nullable = true)
 |-- sex: integer (nullable = true)
 |-- chest_pain_type: integer (nullable = true)
 |-- resting_bp: integer (nullable = true)
 |-- cholestoral: integer (nullable = true)
 |-- fasting_blood_sugar: integer (nullable = true)
 |-- restecg: integer (nullable = true)
 |-- max_hr: integer (nullable = true)
 |-- exang: integer (nullable = true)
 |-- oldpeak: double (nullable = true)
 |-- slope: integer (nullable = true)
 |-- num_major_vessels: integer (nullable = true)
 |-- thal: integer (nullable = true)
 |-- target: integer (nullable = true)
```

Fig.10. Heart disease dataset variables and their types.

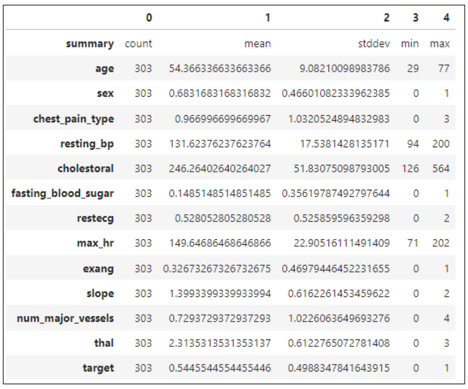
Another code is used to count the columns and rows of this dataset. You can see that there are 303 rows and 14 columns in the used data, where the class variable of this dataset is 'target' (see Figure 11). Also, you can find a summary of the various statistical measures of this dataset: mean, standard deviation, minimum, and maximum values (see Figure 12).



```
print((df.count(), len(df.columns)))
print("There are",df.count(),"rows","and",len(df.columns),"columns","in the data")
```

```
(303, 14)
There are 303 rows and 14 columns in the data
```

Fig.11. Number of columns and rows.

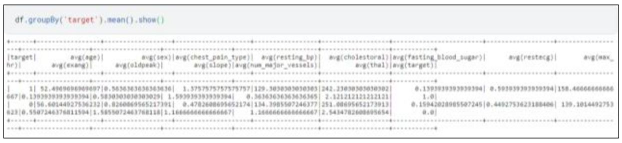


	0	1	2	3	4
summary	count	mean	stddev	min	max
age	303	54.366336633663366	9.08210098983786	29	77
sex	303	0.6831683168316832	0.46601082333962385	0	1
chest_pain_type	303	0.966996699669967	1.0320524894832983	0	3
resting_bp	303	131.62376237623764	17.53814281351 71	94	200
cholestoral	303	246.26402640264027	51.83075098793005	126	564
fasting_blood_sugar	303	0.1485148514851485	0.356197874927976 44	0	1
restecg	303	0.528052805280528	0.525859596359298	0	2
max_hr	303	149.64686468646866	22.90516111491409	71	202
exang	303	0.32673267326732675	0.46979446452231655	0	1
slope	303	1.3993399339933994	0.6162261453459622	0	2
num_major_vessels	303	0.7293729372937293	1.0226063649693276	0	4
thal	303	2.3135313531353137	0.6122765072781408	0	3
target	303	0.5445544554455446	0.4988347841643915	0	1

Fig.12. Summary of the numerical variables.

The summary shows that the average age of participants is around 54 years, and the majority are males. Also, the oldest one is 77 years old, while the youngest one is 29 years old. Most of the participants have the first type of chest pain, considering that there are three types of chest pain. The highest blood pressure measured is 200 mmHg, while the lowest measured is 94 mmHg. This explains the high standard deviation of this variable (~17.5). However, cholesterol values have the highest standard deviation (~52) compared to other variables. This can be explained by the vast gap between the calculated values; its minimum and maximum values are 126 and 564 mg/dL, respectively.

When *the groupBy* function is used to see the summary of the output variable, it shows that the average age of people with heart diseases is ~52 years, and the average age of healthy people is ~57 years. The differences that can be noticed are having chest pain, an increase in *resting heart rate (restecg)*, an increase in maximum heart rate, and a decrease in both *exang* and *oldpeak* values (see Figure 13).

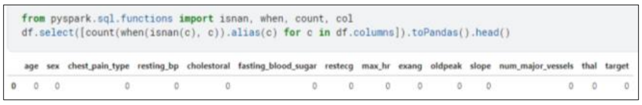


```
df.groupBy('target').mean().show()
```

```
+------+-----------------+------------------+--------------------+-----------------+------------------+------------------------+------------------+--------
---+------------------+------------------+------------------+----------------------+------------------+-----------+
|target|         avg(age)|          avg(sex)|avg(chest_pain_type)|  avg(resting_bp)|  avg(cholestoral)|avg(fasting_blood_sugar)|      avg(restecg)|    avg(max_
hr)|        avg(exang)|      avg(oldpeak)|        avg(slope)|avg(num_major_vessels)|         avg(thal)|avg(target)|
+------+-----------------+------------------+--------------------+-----------------+------------------+------------------------+------------------+--------
---+------------------+------------------+------------------+----------------------+------------------+-----------+
|     1| 52.4969696969697|0.5636363636363636|  1.3757575757575757|129.3030303030303|242.23030303030302|      0.1393939393939394| 0.593939393939394|158.4666666666
667|0.1393939393939394|0.5830303030303029| 1.593939393939394|    0.3636363636363636| 2.121212121212121|        1.0|
|     0|56.60144927536232|0.8260869565217391|  0.4782608695652174|134.3985507246377|251.08695652173913|     0.15942028985507245|0.4492753623188406| 139.1014492753
623|0.5507246376811594|1.5855072463768118|1.1666666666666667|    1.1666666666666667|2.5434782608695654|        0.0|
+------+-----------------+------------------+--------------------+-----------------+------------------+------------------------+------------------+--------
---+------------------+------------------+------------------+----------------------+------------------+-----------+
```

Fig.13. Summary of the target variable.

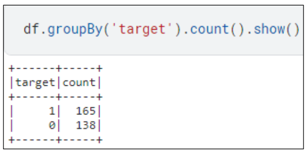
One of the advantages of using this dataset is that there are no missing values (see Figure 14). The importance of having a dataset without missing values comes from the fear of algorithm failure, having incorrect results, and a lack of precision in the analysis afterwards [47]. Also, the *groupBy* function shows that 165 out of 303 patients are diagnosed with heart disease (see Figure 15).



```
from pyspark.sql.functions import isnan, when, count, col
df.select([count(when(isnan(c), c)).alias(c) for c in df.columns]).toPandas().head()
```

	age	sex	chest_pain_type	resting_bp	cholestoral	fasting_blood_sugar	restecg	max_hr	exang	oldpeak	slope	num_major_vessels	thal	target
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Fig.14. Checking for any missing values.



```
df.groupBy('target').count().show()
```

```
+------+-----+
|target|count|
+------+-----+
|     1|  165|
|     0|  138|
+------+-----+
```

Fig.15. Number of patients with heart disease (1 = Have heart disease; 0 = healthy).

To see the distribution of features within the dataset, you can find graphs of that in Figure 16. You can notice that several values follow roughly a normal distribution shape, like *age*, *resting_bp*, *cholesterol*, and *max_hr*. Figures 17 to 23 show various visualizations of different variables. For instance, the second chest pain type is the dominant compared to the other types, as shown in the pie chart. Also, scatter plots show a null-or-no relationship between the drawn variables (see Figures 19 and 20). If you look at the boxplots, you will find that most participants in this dataset are between 50 and 60 years old, *thal* type is more likely to fall between 2 and 3, and most cholesterol levels are between 200 and 300 mg/dL (see Figures 21–23).

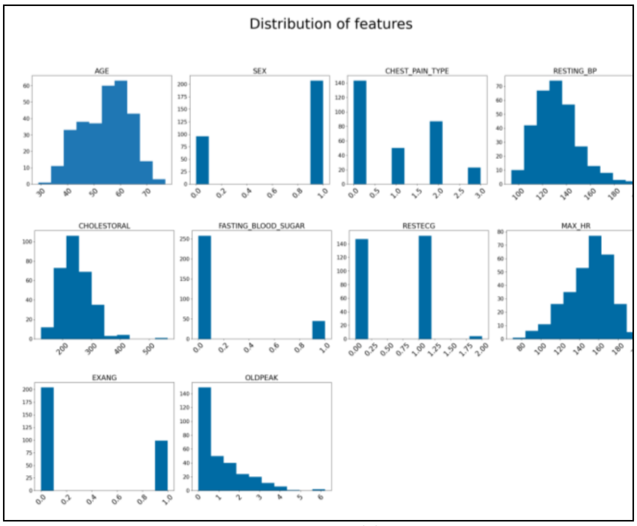


Fig.16. Distribution of features.

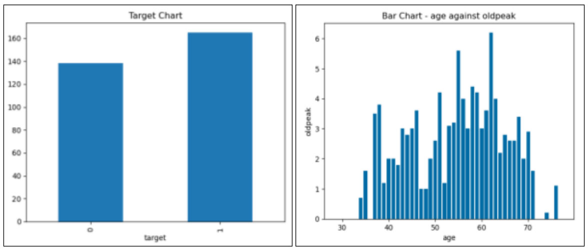


Fig.17. Bar chart of *target* variable (left), and *age* and *oldpeak* (right).

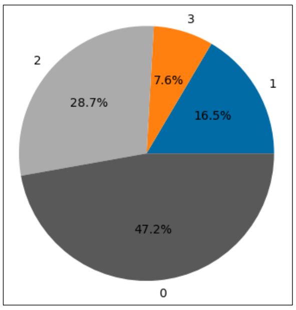


Fig.18. Pie chart of *chest_pain_type* categories

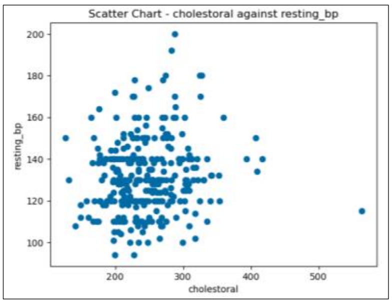


Fig.19. Scatter plot of *cholestoral* against *resting_bp* categories

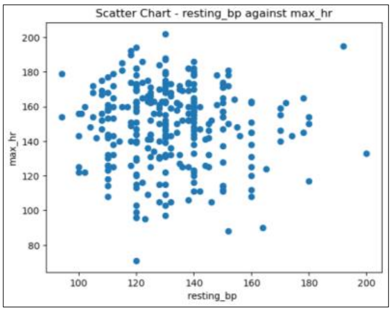


Fig.20. Scatter plot of *resting_bp* against *max_hr* categories

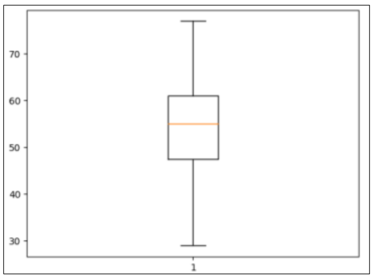


Fig.21. Boxplot of *age* variable

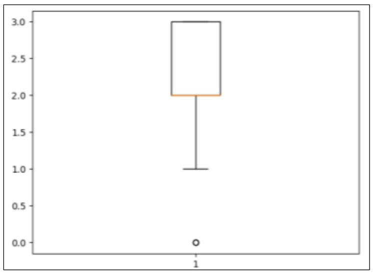


Fig.22. Boxplot of *thal* variable

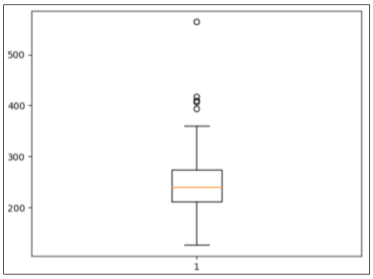
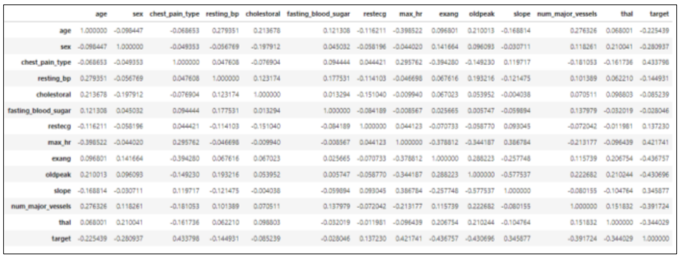


Fig.23. Boxplot of *cholestoral* variable

C. *Statistical analysis*

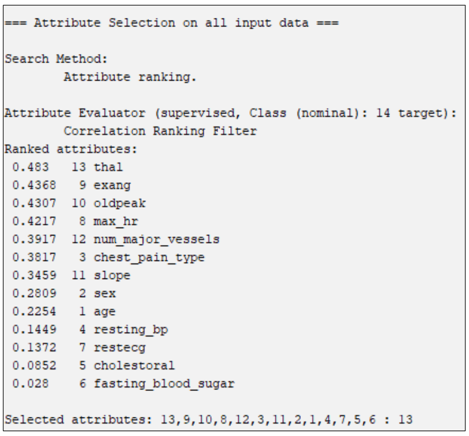
When calculating the correlation between the input variables and the output variable using the “pearson” method, four variables showed a good correlation between them and the output: *chest_pain_type*, *restecg*, *max_hr*, and *slope* (see Figure 24). *Chest_pain_type* and *max_hr* have the highest correlation with the class label with ~0.4 correlation, followed by *slope* with ~0.3 correlation, and *restecg* with ~0.1 correlation. This gives us an indicator of the effect of all these factors on the possibility of having heart disease.



	age	sex	chest_pain_type	resting_bp	cholestoral	fasting_blood_sugar	restecg	max_hr	exang	oldpeak	slope	num_major_vessels	thal	target
age	1.000000	-0.098447	-0.068653	0.279351	0.213678	0.121308	-0.116211	-0.398522	0.096801	0.210013	-0.168814	0.276326	0.068001	-0.225439
sex	-0.098447	1.000000	-0.049353	-0.056769	-0.197912	0.045032	-0.058196	-0.044020	0.141664	0.096093	-0.030711	0.118261	0.210041	-0.280937
chest_pain_type	-0.068653	-0.049353	1.000000	0.047608	-0.076904	0.094444	0.044421	0.295762	-0.394280	-0.149230	0.119717	-0.181053	-0.161736	0.433798
resting_bp	0.279351	-0.056769	0.047608	1.000000	0.123174	0.177531	-0.114103	-0.046698	0.067616	0.193216	-0.121475	0.101389	0.062210	-0.144931
cholestoral	0.213678	-0.197912	-0.076904	0.123174	1.000000	0.013294	-0.151040	-0.009940	0.067023	0.053952	-0.004038	0.070511	0.098803	-0.085239
fasting_blood_sugar	0.121308	0.045032	0.094444	0.177531	0.013294	1.000000	-0.084189	-0.008567	0.025665	0.005747	-0.059894	0.137979	-0.032019	-0.028046
restecg	-0.116211	-0.058196	0.044421	-0.114103	-0.151040	-0.084189	1.000000	0.044123	-0.070733	-0.058770	0.093045	-0.072042	-0.011981	0.137230
max_hr	-0.398522	-0.044020	0.295762	-0.046698	-0.009940	-0.008567	0.044123	1.000000	-0.378812	-0.344187	0.386784	-0.213177	-0.096439	0.421741
exang	0.096801	0.141664	-0.394280	0.067616	0.067023	0.025665	-0.070733	-0.378812	1.000000	0.288223	-0.257748	0.115739	0.206754	-0.436757
oldpeak	0.210013	0.096093	-0.149230	0.193216	0.053952	0.005747	-0.058770	-0.344187	0.288223	1.000000	-0.577537	0.222682	0.210244	-0.430696
slope	-0.168814	-0.030711	0.119717	-0.121475	-0.004038	-0.059894	0.093045	0.386784	-0.257748	-0.577537	1.000000	-0.080155	-0.104764	0.345877
num_major_vessels	0.276326	0.118261	-0.181053	0.101389	0.070511	0.137979	-0.072042	-0.213177	0.115739	0.222682	-0.080155	1.000000	0.151832	-0.391724
thal	0.068001	0.210041	-0.161736	0.062210	0.098803	-0.032019	-0.011981	-0.096439	0.206754	0.210244	-0.104764	0.151832	1.000000	-0.344029
target	-0.225439	-0.280937	0.433798	-0.144931	-0.085239	-0.028046	0.137230	0.421741	-0.436757	-0.430696	0.345877	-0.391724	-0.344029	1.000000

Fig.24. Correlation between variables.

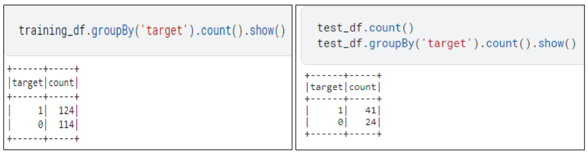
Moreover, when repeating the same process using Weka software, seven variables showed a good correlation between them and the output: *thal*, *exang*, *oldpeak*, *max_hr*, *num_major_vessels*, *chest_pain_type*, and *slope* (see Figure 25). *thal* has the highest correlation with the class label with ~0.48 correlation, followed by *exang* with ~0.44 correlation, followed by *oldpeak* with ~0.43 correlation. This emphasizes the effect of *chest_pain_type*, *max_hr* and *slope* on the possibility of having heart disease.



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 14 target):
        Correlation Ranking Filter
Ranked attributes:
 0.483    13 thal
 0.4368    9 exang
 0.4307   10 oldpeak
 0.4217    8 max_hr
 0.3917   12 num_major_vessels
 0.3817    3 chest_pain_type
 0.3459   11 slope
 0.2809    2 sex
 0.2254    1 age
 0.1449    4 resting_bp
 0.1372    7 restecg
 0.0852    5 cholestoral
 0.028     6 fasting_blood_sugar

Selected attributes: 13,9,10,8,12,3,11,2,1,4,7,5,6 : 13
```

Fig.25. Correlation between variables and the output using Weka.

For the logistic regression, all the input columns are assembled into one single vector, including all the input features of the model. Then, the dataset is split to train and evaluate the performance of the logistic regression model using an 80/20 ratio to train our model on 80% of the dataset, where 80% of this data is 238 out of 303. After that, the splitting for the training and testing sets is verified, as shown below.



```
training_df.groupBy('target').count().show()

+------+-----+
|target|count|
+------+-----+
|     1|  124|
|     0|  114|
+------+-----+
```

```
test_df.count()
test_df.groupBy('target').count().show()

+------+-----+
|target|count|
+------+-----+
|     1|   41|
|     0|   24|
+------+-----+
```

Fig.26. Splitting verification of the training set (left) and testing set (right).

After training the predictions, the correct predictions based on the training set are 124, with around 83% prediction accuracy. As shown in Figure 27, the model predicts 36 true positives and 18 true negatives, while it gives predictions of 6 false positives and 5 false negatives.

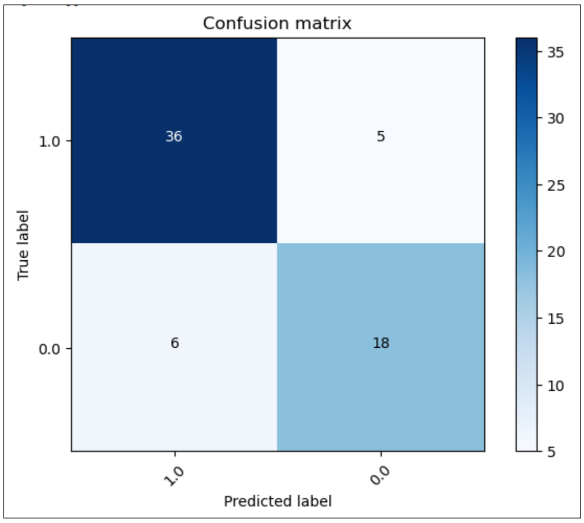
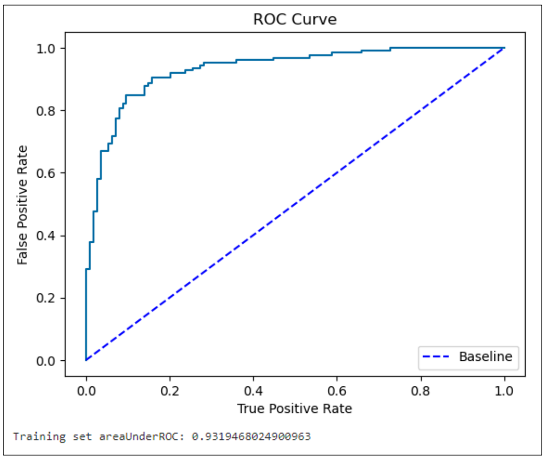


Fig.27. Confusion Matrix of the logistic regression model.

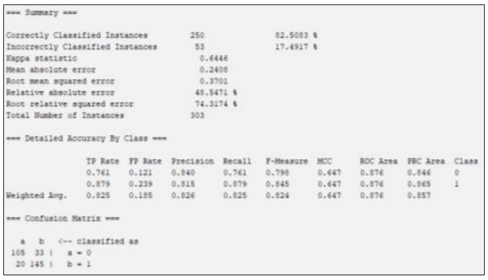
As shown below, the Receiver Operating Characteristic curve (ROC) helps in deciding the threshold value for the model. Two parameters are used in plotting this curve: the true positive rate and the false positive rate. A random classifier is drawn as a dashed diagonal line to see how far this curve is from this dashed line. A good classifier stays toward the top-left corner, as far as possible from that line, and this graph shows that this classifier performs well. Since this is close to 0.93, this confirms that the model does a good job of classifying data.



```
Training set areaUnderROC: 0.9319468024900963
```

Fig.28. ROC of the logistic regression.

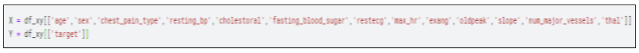
When using Weka to run the logistic regression, it gives an ~82.5% prediction accuracy, which is equal to the value calculated by Spark (see Figure 29). The model predicted 53 out of 303 values incorrectly, where the correct ones reflect a good indicator of the model's performance.



```
=== Summary ===

Correctly Classified Instances         250               82.5083 %
Incorrectly Classified Instances        53               17.4917 %
Kappa statistic                          0.6446
Mean absolute error                      0.2408
Root mean squared error                  0.3701
Relative absolute error                 48.5471 %
Root relative squared error             74.3174 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.761    0.121    0.840      0.761    0.798      0.647    0.876     0.846     0
                 0.879    0.239    0.815      0.879    0.845      0.647    0.876     0.865     1
Weighted Avg.    0.825    0.185    0.826      0.825    0.824      0.647    0.876     0.857

=== Confusion Matrix ===

   a   b   <-- classified as
 105  33 |   a = 0
  20 145 |   b = 1
```

Fig.29. Summary of the logistic regression analysis using Weka.

For the decision tree model, all the input features are assigned to X and the output variable to Y (see Figure 30). Then, the dataset is split again to train and evaluate the performance of the decision tree model using the 80/20 ratio.



```
X = df_xy[['age','sex','chest_pain_type','resting_bp','cholestoral','fasting_blood_sugar','restecg','max_hr','exang','oldpeak','slope','num_major_vessels','thal']]
Y = df_xy[['target']]
```

Fig.30. Separating variables to X and Y.

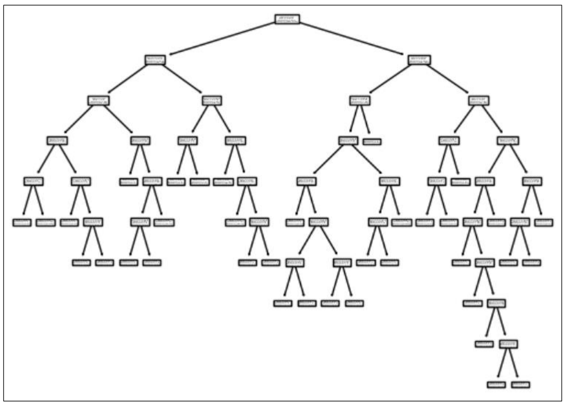


Fig.31. Initial Decision Tree plot.

As you can see above, it shows the initial plot of the model. To make it clearer for interpretation and analysis, another code is used, and the outcome is shown below (see Figure 32).

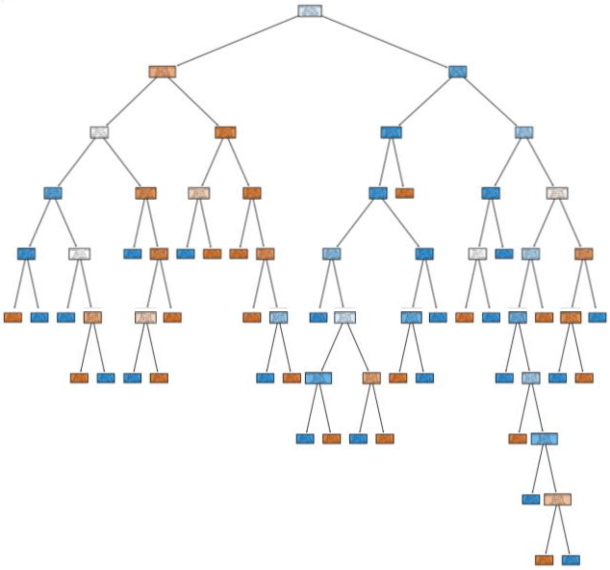


Fig.32. Another Decision Tree plot.

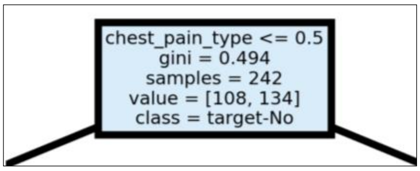


Fig.33. Zoomed image of the root node of the model.

Based on Gini ratio, *chest_pain_type* is the root node with the most strength and influence (see Figure 33), followed by *num_major_vessels* and *age* variables—an internal and decision node—with more decision-making based on the other variables to decide if the patient has a less or more chance of having a heart disease. Therefore, chest pain type is the root factor and most informative attribute that increases or decreases the possibility of having a heart attack.

After training the model and creating a decision tree classifier object, the calculated accuracy prediction is ~74%. As shown in Figure 34, the model predicts 22 true positives and 23 true negatives, while giving predictions of 8 false positives and 8 false negatives.

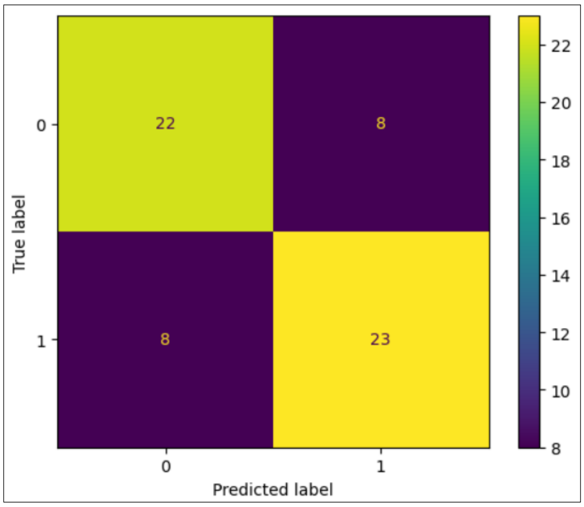
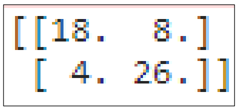


Fig.34. Confusion Matrix of the Decision Tree model.

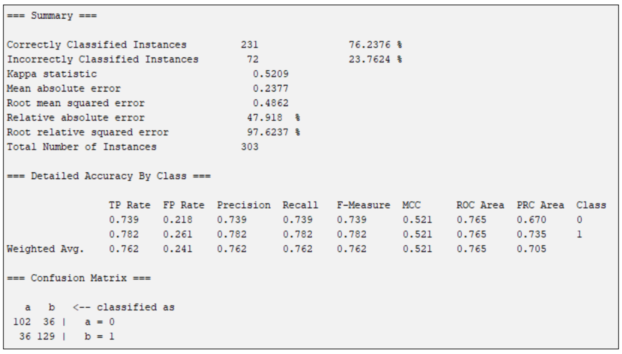
In regard to the last model, a *Random Forest Classifier* is imported after splitting data again to train and evaluate the performance of the random forest model using an 80/20 ratio. From that, the accuracy score is around 78% with ~22% error rate. As shown in Figure 35, the model predicts 18 true positives and 26 true negatives, while it gives predictions of 4 false positives and 8 false negatives.



```
[[18.  8.]
 [ 4. 26.]]
```

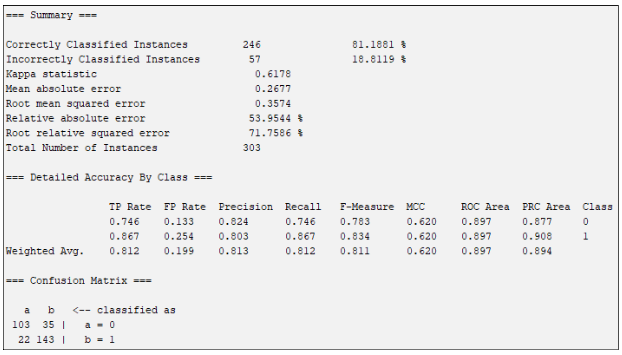
Fig.35. Confusion Matrix of the Random Forest model.

When using Weka to run the decision tree and random forest models, it gives an ~76.2% prediction accuracy for the decision tree model and an ~81.2% prediction accuracy for the other model. The accuracy scores are nearly similar to the results calculated by Spark (see Figures 36–38). From that, it is emphasized that Weka is a time-saving tool to help a data scientist, for instance, work efficiently and predict outcomes effectively.



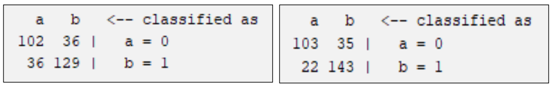
```
=== Summary ===

Correctly Classified Instances         231               76.2376 %
Incorrectly Classified Instances        72               23.7624 %
Kappa statistic                          0.5209
Mean absolute error                      0.2377
Root mean squared error                  0.4862
Relative absolute error                 47.918  %
Root relative squared error             97.6237 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.739    0.218    0.739      0.739    0.739      0.521    0.765     0.670     0
                 0.782    0.261    0.782      0.782    0.782      0.521    0.765     0.735     1
Weighted Avg.    0.762    0.241    0.762      0.762    0.762      0.521    0.765     0.705

=== Confusion Matrix ===

   a   b   <-- classified as
 102  36 |   a = 0
  36 129 |   b = 1
```

Fig.36. Summary of the decision tree analysis using Weka.



```
=== Summary ===

Correctly Classified Instances         246               81.1881 %
Incorrectly Classified Instances        57               18.8119 %
Kappa statistic                          0.6178
Mean absolute error                      0.2677
Root mean squared error                  0.3574
Relative absolute error                 53.9544 %
Root relative squared error             71.7586 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.746    0.133    0.824      0.746    0.783      0.620    0.897     0.877     0
                 0.867    0.254    0.803      0.867    0.834      0.620    0.897     0.908     1
Weighted Avg.    0.812    0.199    0.813      0.812    0.811      0.620    0.897     0.894

=== Confusion Matrix ===

   a   b   <-- classified as
 103  35 |   a = 0
  22 143 |   b = 1
```

Fig.37. Summary of the random forest analysis using Weka.

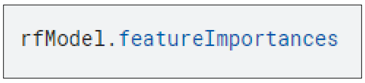


```
   a   b   <-- classified as
 102  36 |   a = 0
  36 129 |   b = 1
```

```
   a   b   <-- classified as
 103  35 |   a = 0
  22 143 |   b = 1
```

Fig.38. Confusion Matrices of decision tree model (left) and random forest (right) using Weka.

To check the importance of each feature, the following code is used:



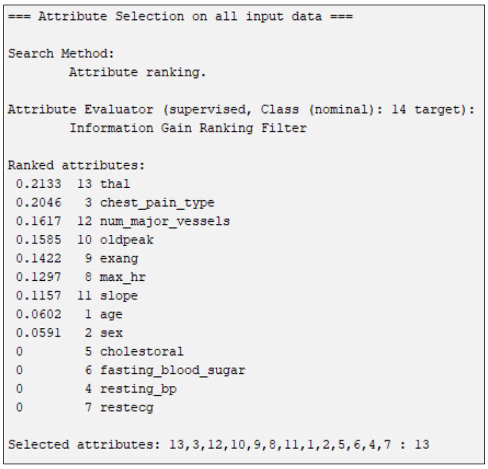
```
rfModel.featureImportances
```

TABLE II. FEATURE IMPORTANCE

Feature Name	Importance Score
thal	0.2286
chest_pain_type	0.1806
num_major_vessels	0.1341
max_hr	0.0981
oldpeak	0.09
age	0.0813
resting_bp	0.0388
exang	0.0367
cholestoral	0.0319
sex	0.0255
restecg	0.0242
slope	0.0218
fasting_blood_sugar	0.0083

It shows that *thal* variable is the most important and useful variable that contributes the most to model predictions (0.2286 score), followed by *chest_pain_type* and *num_major_vessels* (0.1806 and 0.1341, respectively). The variable with the least effect according to this model is *fasting_blood_sugar* with a 0.0083 score (see Table II).

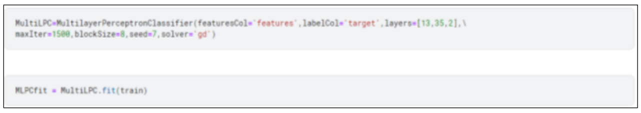
Using Weka, the following screenshot shows the effect of each feature on the model (see Figure 39). You can notice that the results are approximately the same, where *thal*, *chest_pain_type*, and *num_major_vessels* are the most influencing factors compared to the other ones.



```
=== Attribute Selection on all input data ===

Search Method:
        Attribute ranking.

Attribute Evaluator (supervised, Class (nominal): 14 target):
        Information Gain Ranking Filter

Ranked attributes:
 0.2133   13 thal
 0.2046    3 chest_pain_type
 0.1617   12 num_major_vessels
 0.1585   10 oldpeak
 0.1422    9 exang
 0.1297    8 max_hr
 0.1157   11 slope
 0.0602    1 age
 0.0591    2 sex
 0         5 cholestoral
 0         6 fasting_blood_sugar
 0         4 resting_bp
 0         7 restecg

Selected attributes: 13,3,12,10,9,8,11,1,2,5,6,4,7 : 13
```

Fig.39. Feature importance based on Weka calculations.

In addition, a neural network analysis is done for the dataset using both Spark and Weka. To do that, *a Perceptron Classifier* is used along with other needed functions and libraries. In Spark, the best accuracy result is ~68.7% using 35 hidden layers. In contrast, the accuracy score calculated by Weka is ~77.9% using 10 hidden layers (see Figures 40–43). The difference between them is around 10%, and that comes from the difference between the two tools, their libraries, and the hidden layers effect.



```
MultiLPC=MultilayerPerceptronClassifier(featuresCol='features',labelCol='target',layers=[13,35,2],\
maxIter=1500,blockSize=8,seed=7,solver='gd')
```

```
MLPCfit = MultiLPC.fit(train)
```

Fig.40. Codes used in Spark to run the neural network model.

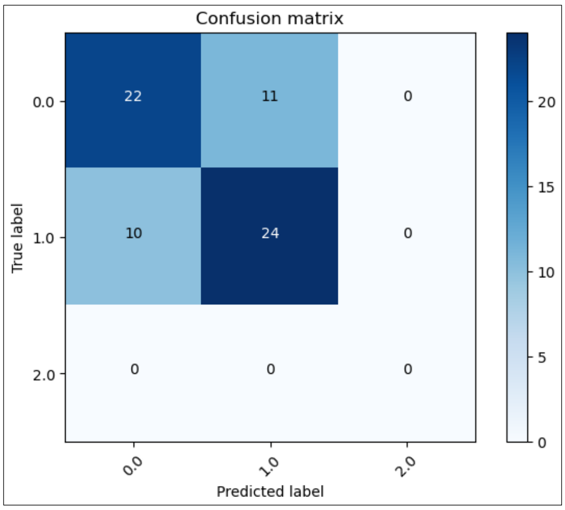
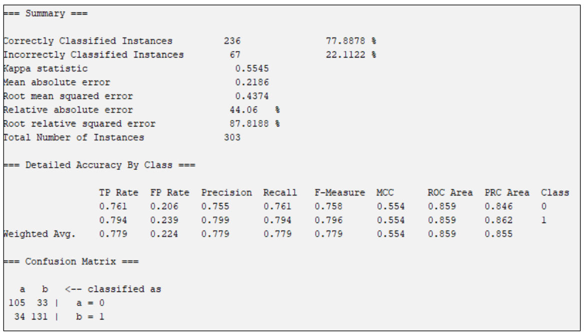


Fig.41. Confusion Matrix of the neural network model.



```
=== Summary ===

Correctly Classified Instances         236               77.8878 %
Incorrectly Classified Instances        67               22.1122 %
Kappa statistic                          0.5545
Mean absolute error                      0.2186
Root mean squared error                  0.4374
Relative absolute error                 44.06   %
Root relative squared error             87.8188 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.761    0.206    0.755      0.761    0.758      0.554    0.859     0.846     0
                 0.794    0.239    0.799      0.794    0.796      0.554    0.859     0.862     1
Weighted Avg.    0.779    0.224    0.779      0.779    0.779      0.554    0.859     0.855

=== Confusion Matrix ===

   a   b   <-- classified as
 105  33 |   a = 0
  34 131 |   b = 1
```

Fig.42. Summary of the neural network analysis using Weka.

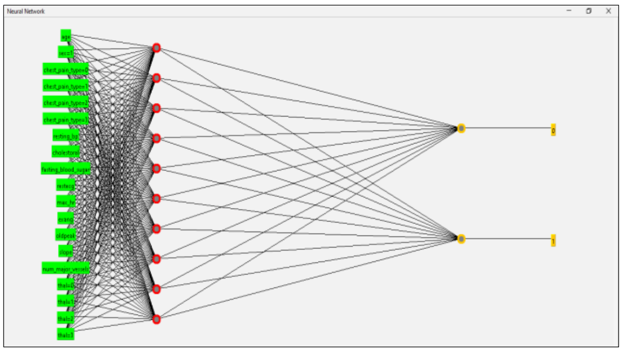
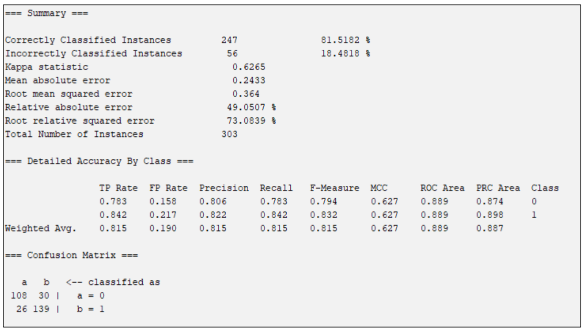


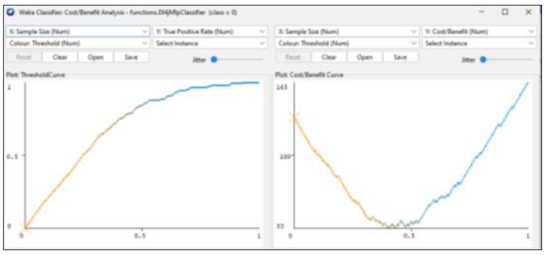
Fig.43. The neural network of the heart dataset.

Finally, a deep learning model is applied to this dataset again, along with a multilayer perceptron using Weka only. The calculated accuracy of this model is ~81.5%, which is a good one, showing the benefit of using such a model in doing heart disease analysis to help diagnose urgent cases (see Figures 44–46).



```
=== Summary ===

Correctly Classified Instances         247               81.5182 %
Incorrectly Classified Instances        56               18.4818 %
Kappa statistic                          0.6265
Mean absolute error                      0.2433
Root mean squared error                  0.364
Relative absolute error                 49.0507 %
Root relative squared error             73.0839 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.783    0.158    0.806      0.783    0.794      0.627    0.889     0.874     0
                 0.842    0.217    0.822      0.842    0.832      0.627    0.889     0.898     1
Weighted Avg.    0.815    0.190    0.815      0.815    0.815      0.627    0.889     0.887

=== Confusion Matrix ===

   a   b   <-- classified as
 108  30 |   a = 0
  26 139 |   b = 1
```

Fig.44. Summary of the deep learning analysis using Weka.

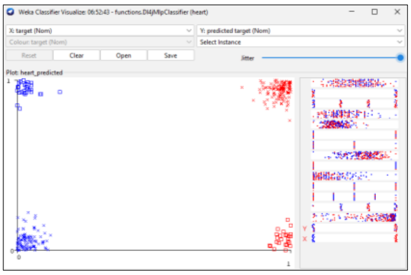


Weka Classifier: Cost/Benefit Analysis - functions.Dl4jMlpClassifier (class = 0)

X: Sample Size (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance
Plot: ThresholdCurve

X: Sample Size (Num)
Y: Cost/Benefit (Num)
Colour: Threshold (Num)
Select Instance
Plot: Cost/Benefit Curve

Fig.45. Threshold plot (left) and cost/benefit curve (right) of the deep learning analysis using Weka.



Weka Classifier Visualize: 06:52:43 - functions.Dl4jMlpClassifier (heart)

X: target (Nom)
Y: predicted target (Nom)
Colour: target (Nom)
Select Instance
Plot: heart_predicted

Fig.46. Classifier error of the deep learning analysis using Weka.

D. *Results*

From the above discussion and analysis, you can notice that both logistic regression and deep learning models have the highest accuracy scores compared to the other models, which still have acceptable accuracy scores over 70%. With this type of data, where there are two results, it is better to use more than one analytical model, like deep learning supported by logistic regression algorithms. The analysis shows that the error rate of the decision tree model is the highest, taking into consideration that the best error rate needs to be close to zero.

TABLE III. COMPARISON BETWEEN ACCURACY SCORES OF THE FIVE MODELS USING ALL ATTRIBUTES

	Error Rate	Accuracy Score	Accuracy Score using 75/25 ratio
Logistic Regression	~16.9%	~83.1%	~82.0%
Decision Tree	~26.0%	~74.0%	~72.0%
Random Forest	~22.0%	~78.6%	~76.0%
Neural Network (using Weka)	~22.1%	~77.9%	~84.2%
Deep Learning (using Weka)	~18.5%	~81.5%	~81.6%
Average Score	-	78.9%	~79.2%

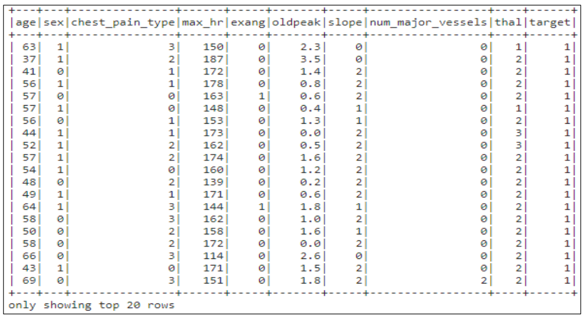
When comparing it to the Random Forest model, you can see that the last one gives more accurate predictions because of its more complex and networked structure. Therefore, it is considered better for making future predictions, and that is understandable since random forest models represent a collection of decision trees. Similarly, accuracy scores are not that different from the ones calculated using the 75/25 ratio; the accuracy decreased for all models except for neural networks and deep learning models (see Table III).

E. *Using selected attributes*

After calculating the significance of attributes, it is a good point to run analytical models using selected attributes to test the effect of that on model accuracy, as shown below. When comparing the results in *Table IV* with the previous ones in *Table III*, you can notice that the accuracy value decreased for all analytical models except for the logistic regression model. Two possible reasons for that can be the nature of the data—the binary outcome—or the fact that deep learning analytical methods give better accuracy throughout random voting and taking the average of the outcome. The average accuracy score is between 76.0% and 80.4%, which is an indicator of the good efficiency of the model's performance. Still, further analysis is needed using additional datasets and attributes to improve the models' accuracy.

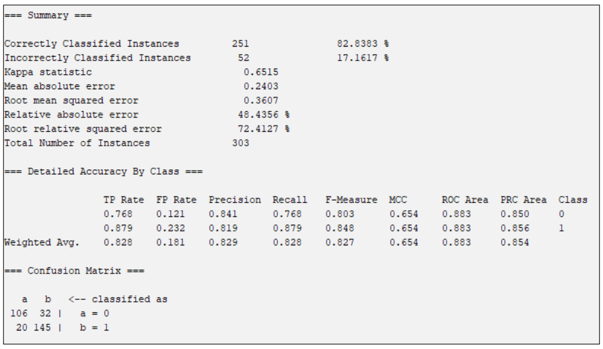
TABLE IV. COMPARISON BETWEEN ACCURACY SCORES OF THE FIVE MODELS USING SELECTED ATTRIBUTES

Modified Dataset	Accuracy Score (Spark) -80/20 ratio	Accuracy Score (Weka)- 10-fold cross validation
Logistic Regression	~83.8%	~82.8%
Decision Tree	~73.8%	~74.6%
Random Forest	~73.1%	~80.5%
Neural Network	~73.1%	~80.5%
Deep Learning	-	~83.8%
Average Score	~76.0%	~80.4%



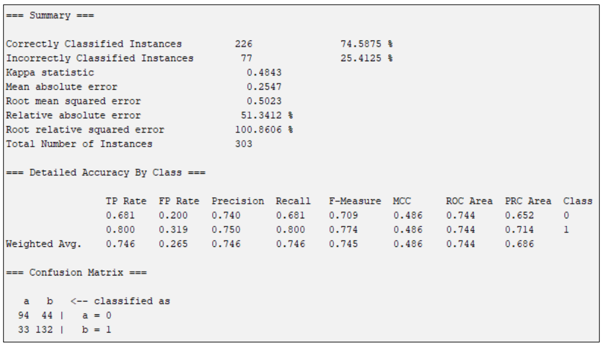
```
+---+---+---------------+------+-----+-------+-----+------------------+----+------+
|age|sex|chest_pain_type|max_hr|exang|oldpeak|slope|num_major_vessels|thal|target|
+---+---+---------------+------+-----+-------+-----+------------------+----+------+
| 63|  1|              3|   150|    0|    2.3|    0|                 0|   1|     1|
| 37|  1|              2|   187|    0|    3.5|    0|                 0|   2|     1|
| 41|  0|              1|   172|    0|    1.4|    2|                 0|   2|     1|
| 56|  1|              1|   178|    0|    0.8|    2|                 0|   2|     1|
| 57|  0|              0|   163|    1|    0.6|    2|                 0|   2|     1|
| 57|  1|              0|   148|    0|    0.4|    1|                 0|   1|     1|
| 56|  0|              1|   153|    0|    1.3|    1|                 0|   2|     1|
| 44|  1|              1|   173|    0|    0.0|    2|                 0|   3|     1|
| 52|  1|              2|   162|    0|    0.5|    2|                 0|   3|     1|
| 57|  1|              2|   174|    0|    1.6|    2|                 0|   2|     1|
| 54|  1|              0|   160|    0|    1.2|    2|                 0|   2|     1|
| 48|  0|              2|   139|    0|    0.2|    2|                 0|   2|     1|
| 49|  1|              1|   171|    0|    0.6|    2|                 0|   2|     1|
| 64|  1|              3|   144|    1|    1.8|    1|                 0|   2|     1|
| 58|  0|              3|   162|    0|    1.0|    2|                 0|   2|     1|
| 50|  0|              2|   158|    0|    1.6|    1|                 0|   2|     1|
| 58|  0|              2|   172|    0|    0.0|    2|                 0|   2|     1|
| 66|  0|              3|   114|    0|    2.6|    0|                 0|   2|     1|
| 43|  1|              0|   171|    0|    1.5|    2|                 0|   2|     1|
| 69|  0|              3|   151|    0|    1.8|    2|                 2|   2|     1|
+---+---+---------------+------+-----+-------+-----+------------------+----+------+
only showing top 20 rows
```

Fig.47. Selected attributes with high info gain.



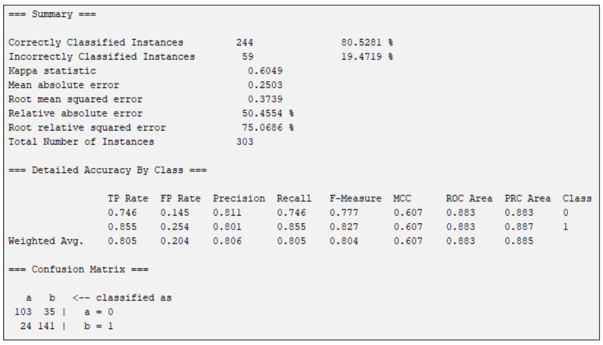
```
=== Summary ===

Correctly Classified Instances         251               82.8383 %
Incorrectly Classified Instances        52               17.1617 %
Kappa statistic                          0.6515
Mean absolute error                      0.2403
Root mean squared error                  0.3607
Relative absolute error                 48.4356 %
Root relative squared error             72.4127 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.768    0.121    0.841      0.768    0.803      0.654    0.883     0.850     0
                 0.879    0.232    0.819      0.879    0.848      0.654    0.883     0.856     1
Weighted Avg.    0.828    0.181    0.829      0.828    0.827      0.654    0.883     0.854

=== Confusion Matrix ===

   a   b   <-- classified as
 106  32 |   a = 0
  20 145 |   b = 1
```

Fig.48. Logistic regression model for selected attributes using Weka.



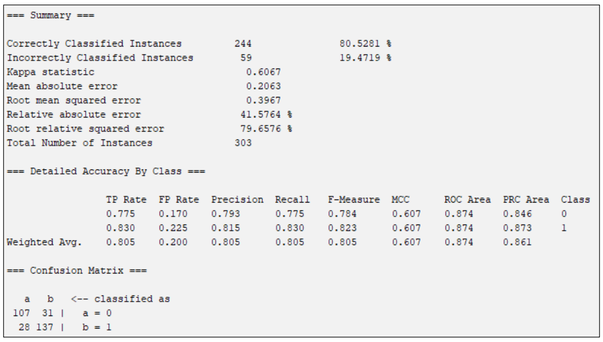
```
=== Summary ===

Correctly Classified Instances         226               74.5875 %
Incorrectly Classified Instances        77               25.4125 %
Kappa statistic                          0.4843
Mean absolute error                      0.2547
Root mean squared error                  0.5023
Relative absolute error                 51.3412 %
Root relative squared error            100.8606 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.681    0.200    0.740      0.681    0.709      0.486    0.744     0.652     0
                 0.800    0.319    0.750      0.800    0.774      0.486    0.744     0.714     1
Weighted Avg.    0.746    0.265    0.746      0.746    0.745      0.486    0.744     0.686

=== Confusion Matrix ===

   a   b   <-- classified as
  94  44 |   a = 0
  33 132 |   b = 1
```

Fig.49. Decision tree model for selected attributes using Weka.



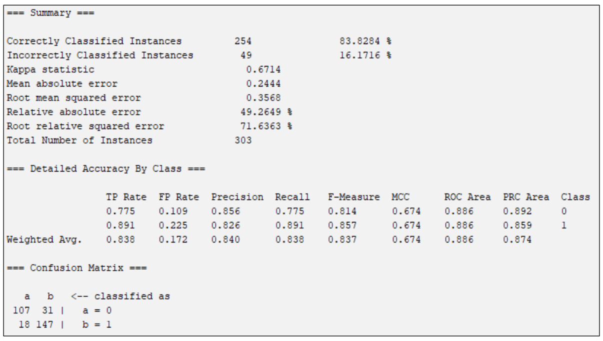
```
=== Summary ===

Correctly Classified Instances         244               80.5281 %
Incorrectly Classified Instances        59               19.4719 %
Kappa statistic                          0.6049
Mean absolute error                      0.2503
Root mean squared error                  0.3739
Relative absolute error                 50.4554 %
Root relative squared error             75.0686 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.746    0.145    0.811      0.746    0.777      0.607    0.883     0.883     0
                 0.855    0.254    0.801      0.855    0.827      0.607    0.883     0.887     1
Weighted Avg.    0.805    0.204    0.806      0.805    0.804      0.607    0.883     0.885

=== Confusion Matrix ===

   a   b   <-- classified as
 103  35 |   a = 0
  24 141 |   b = 1
```

Fig.50. Random forest model for selected attributes using Weka.



```
=== Summary ===

Correctly Classified Instances         244               80.5281 %
Incorrectly Classified Instances        59               19.4719 %
Kappa statistic                          0.6067
Mean absolute error                      0.2063
Root mean squared error                  0.3967
Relative absolute error                 41.5764 %
Root relative squared error             79.6576 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.775    0.170    0.793      0.775    0.784      0.607    0.874     0.846     0
                 0.830    0.225    0.815      0.830    0.823      0.607    0.874     0.873     1
Weighted Avg.    0.805    0.200    0.805      0.805    0.805      0.607    0.874     0.861

=== Confusion Matrix ===

   a   b   <-- classified as
 107  31 |   a = 0
  28 137 |   b = 1
```

Fig.51. Neural network model for selected attributes using Weka.



```
=== Summary ===

Correctly Classified Instances         254               83.8284 %
Incorrectly Classified Instances        49               16.1716 %
Kappa statistic                          0.6714
Mean absolute error                      0.2444
Root mean squared error                  0.3568
Relative absolute error                 49.2649 %
Root relative squared error             71.6363 %
Total Number of Instances              303

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.775    0.109    0.856      0.775    0.814      0.674    0.886     0.892     0
                 0.891    0.225    0.826      0.891    0.857      0.674    0.886     0.859     1
Weighted Avg.    0.838    0.172    0.840      0.838    0.837      0.674    0.886     0.874

=== Confusion Matrix ===

   a   b   <-- classified as
 107  31 |   a = 0
  18 147 |   b = 1
```

Fig.52. Deep learning model for selected attributes using Weka.

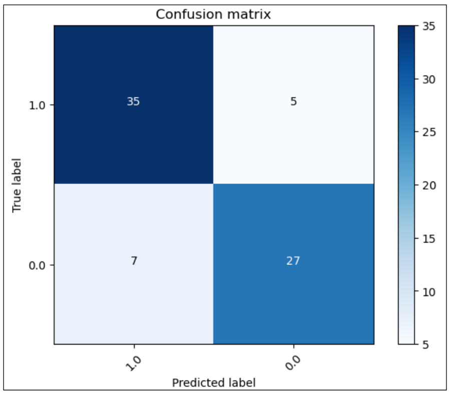


Fig.53. Confusion matrix of logistic regression model for selected attributes using Spark.

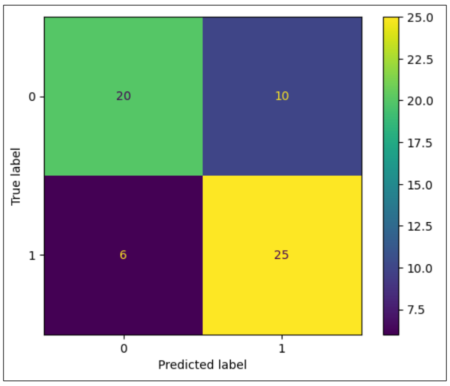
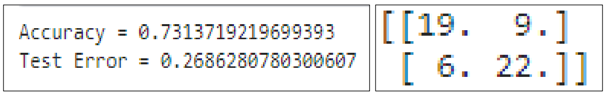


Fig.54. Confusion matrix of decision tree model for selected attributes using Spark.



```
Accuracy = 0.7313719219699393
Test Error = 0.2686280780300607
```

```
[[19.  9.]
 [ 6. 22.]]
```

Fig.55. Accuracy calculations and confusion matrix of random forest model for selected attributes using Spark.

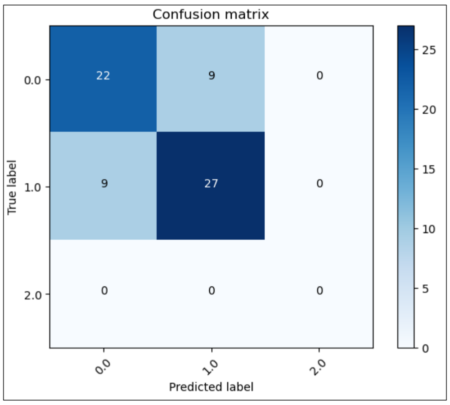


Fig.56. Confusion matrix of neural network model for selected attributes using Spark.

Other metrics to be discussed and analyzed in this section are precision and recall. Using the previous confusion matrices of the analytical models above, you can find in Table V an overview of the results for the three values—precision, recall, and accuracy—to visualize them easily so as to compare between them and take advantage of these values.

Precision measures how often our models have been true in predicting and computing the probability that cases or patients are correctly classified with positive values. It is a useful tool in terms of having a target that your built system is as correct as possible without considering the negative values. In regard to heart disease analysis, precision is important since it can be used to build an effective system that can identify a user's heart condition. This can be done only if it successfully predicts users' heart conditions, which means that high precision is needed to save both time and money for an organization intending to own such a system.

From Table V, you can notice that the higher precision value is achieved by the logistic regression model with around 88%, followed by deep learning and neural network models with around 78% and 76%, respectively. The model with the lowest precision score is the random forest one with a value less than 70%, which is most probably not acceptable in the healthcare sector—besides other sectors where precision value is critical. Interestingly, precision values are higher for models that used all variables, which is an indication of the significance of the most variables included in the analysis instead of removing some of them.

In contrast, the recall metric measures how often our models have been true in correctly classifying a case or patient with the highest positive value among all the real positives. It is a useful tool in terms of classifying an event that has already happened, where the focus is on the real positives as much as possible while the real negatives are neglected. Since our discussion is about a health issue, a high recall value is needed for heart disease detection in patients, considering that this event has already occurred. In other words, it will be risky for a patient to be falsely classified as having a heart disease, so it is important to avoid false negatives by working with recall values.

Remarkably, recall values of the five models are high (over 70%), which is a good indicator of the strength of using these models along with the used dataset. You can notice from Table V that the logistic regression, random forest, and deep learning models achieved the highest recall scores with around 86%, 82%, and 81%, respectively. The lowest recall values go to the decision tree and neural network models with around 73% and 76%, respectively, and this can be explained by the lower complexity and efficiency of these models compared to the other ones. Surprisingly, the recall values of deep learning and decision tree models when using selected attributes increased noticeably, but that is not evidence of the accuracy of these values since the precision values do not support the recall ones. From there, more work needs to be done in analyzing the recall value, since it has a critical role in determining patients' conditions.

TABLE V. COMPARISON BETWEEN PRECISION, RECALL AND ACCURACY VALUES OF THE FIVE MODELS

Analytical Model*	**Precision**	**Recall**	**Accuracy**
Logistic Regression	87.8%	85.7%	83.1%
Decision Tree	73.3%	73.3%	73.8%
Random Forest	69.2%	81.8%	78.6%
Neural Network (Weka)	76.1%	75.5%	77.9%
Deep Learning (Weka)	78.3%	80.6%	81.5%
Logistic Regression**	87.5%	83.3%	83.8%
Decision Tree**	66.7%	76.9%	73.8%
Random Forest**	67.9%	76.0%	73.2%
Neural Network (Weka)**	71.0%	71.0%	73.1%
Deep Learning (Weka)**	77.5%	85.6%	83.8%

* Models that were run using Weka software are noted, otherwise calculations were taken from Spark analysis.

** Results for the five models applied and analyzed at selected attributes as discussed previously.

V. CONCLUSION

This study has examined the impact and advantages of utilizing big data to predict the risk of heart disease in patients. Various analytical models were employed to predict heart disease based on several factors, with the most significant ones being the type of chest pain, thalassemia degree, and number of major vessels. Additionally, secondary factors such as age, maximum heart rate value, old peak, and ST segment's slope were considered. Each model has its own specific applications, and it appears that logistic regression and deep learning offer better analysis of the heart disease dataset. The findings of this paper are supported by numerous papers and discussions. According to [48], the neural network technique can effectively design a diagnostic system to predict heart disease risk levels with an accuracy of approximately 100%, using attributes similar to those used in this study. Another analysis, using the same heart disease dataset as this paper, yielded similar results: an accuracy score of approximately 87% using the Random Forest Classifier, approximately 79% using both the Logistic Regression Model and the Support Vector Classifier, and approximately 81% using the Multilayer Perceptron Classifier [49].

As per a different publication [20], the Naive Bayes classifier and Sequential Minimal Optimization models can be effectively applied with similar datasets and attributes to yield superior outcomes. These two models exhibited enhanced performance with an accuracy rate of approximately 84.5%, surpassing the Multilayer Perceptron's accuracy of around 82%. To delve deeper into this subject, exploring alternative machine learning models, such as diverse deep learning techniques, on comparable datasets is recommended. Moreover, implementing additional strategies, such as augmenting data volume to address weak correlations, creating novel features, focusing solely on essential attributes, and integrating hidden layers, can further enhance results. In the UAE, heart disease emerges as the primary cause of fatalities, emphasizing the critical nature of leveraging artificial intelligence to address and prevent this issue in the future. The outcomes of this study underscore the potential of artificial intelligence and data utilization in driving digital and societal transformations. By harnessing the power of artificial intelligence alongside data-driven methodologies, healthcare professionals can strengthen their capacity to detect, manage, and prevent heart disease, ultimately leading to improved patient outcomes and societal well-being.

REFERENCES

[1] World Health Organization: WHO, “Cardiovascular diseases,” Jun. 11, 2019. https://www.who.int/health-topics/cardiovascular-diseases/#tab=tab_1

[2] “Classes of Heart Failure,” *www.heart.org*, May 31, 2017. https://www.heart.org/en/health-topics/heart-failure/what-is-heart-failure/classes-of-heart-failure

[3] R. Bharti, A. Khamparia, M. Shabaz, G. Dhiman, S. Pande, and P. Singh, “Prediction of Heart Disease Using a Combination of Machine Learning and Deep Learning,” *Computational Intelligence and Neuroscience*, vol. 2021, pp. 1–11, Jul. 2021, doi: 10.1155/2021/8387680.

[4] C. Stedman, “The ultimate guide to big data for businesses,” *Data Management*, Feb. 23, 2022. https://www.techtarget.com/searchdatamanagement/The-ultimate-guide-to-big-data-for-businesses

[5] S. Gupta, “What Are Data Structures and Algorithms?,” *Springboard Blog*, Jul. 08, 2020. https://www.springboard.com/blog/software-engineering/data-structures-and-algorithms/

[6] GeeksforGeeks, “Introduction to Data Structures,” *GeeksforGeeks*, Mar. 15, 2023. https://www.geeksforgeeks.org/introduction-to-data-structures/

[7] R. Dubey, A. Gunasekaran, S. J. Childe, C. Blome, and T. Papadopoulos, “Big Data and Predictive Analytics and Manufacturing Performance: Integrating Institutional Theory, Resource-Based View and Big Data Culture,” *British Journal of Management*, vol. 30, no. 2, pp. 341–361, Apr. 2019, doi: 10.1111/1467-8551.12355.

[8] L. B. Furstenau *et al.*, “Big data in healthcare: Conceptual

network structure, key challenges and opportunities," *Digital Communications and Networks*, Mar. 2023, doi: 10.1016/j.dcan.2023.03.005.

[9] "What is Machine Learning? | IBM." https://www.ibm.com/topics/machine-learning

[10] J. A. R. Neto, "Tools for Data Analysis used in Data Science, ML and Big Data," *Medium*, Jan. 15, 2023. https://medium.com/xnewdata/tools-for-data-analysis-used-in-data-science-ml-and-big-data-87e07e1ddb0

[11] A. Deshpande and M. Kumar, *Artificial Intelligence for Big Data: Complete guide to automating Big Data solutions using Artificial Intelligence techniques*. Packt Publishing Ltd, 2018.

[12] World Health Organization: WHO, "WHO reveals leading causes of death and disability worldwide: 2000-2019," Dec. 09, 2020. https://www.who.int/news/item/09-12-2020-who-reveals-leading-causes-of-death-and-disability-worldwide-2000-2019

[13] A. Shehab *et al.*, "Prevalence of Cardiovascular Risk Factors and 10-Years Risk for Coronary Heart Disease in the United Arab Emirates," *Current Diabetes Reviews*, vol. 19, no. 3, Apr. 2022, doi: 10.2174/1573399818666220421113607.

[14] N. Seth, "Fundamentals Of Neural Networks & Deep Learning | AnalytixLabs," *Blogs & Updates on Data Science, Business Analytics, AI Machine Learning*, Sep. 27, 2022. https://www.analytixlabs.co.in/blog/fundamentals-of-neural-networks/

[15] "What is a Neural Network? - Artificial Neural Network Explained - AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/what-is/neural-network/

[16] "What are Neural Networks? | IBM." https://www.ibm.com/topics/neural-networks#:~:text=A%20neural%20network%20that%20consists, considered%20a%20deep%20learning%20algorithm

[17] S. F. Hussain, S. K. Nanda, S. Barigidad, S. Akhtar, Md. Suaib, and N. K. Ray, *Novel Deep Learning Architecture for Predicting Heart Disease using CNN*. Cornell University, 2021. doi: 10.1109/ocit53463.2021.00076.

[18] S. Mondal, "Regression Analysis | Beginners Comprehensive Guide (Updated 2023)," *Analytics Vidhya*, Feb. 14, 2023. https://www.analyticsvidhya.com/blog/2020/12/beginners-take-how-logistic-regression-is-related-to-linear-regression/#:~:text=The%20Differences%20between%20Linear%20Regression,Logistic%20regression%20provides%20discreet%20output.

[19] "Decision Tree Algorithm in Machine Learning - Javatpoint," www.javatpoint.com. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[20] "Recommendation of Attributes for Heart Disease Prediction using Correlation Measure," *International Journal of Recent Technology and Engineering*, vol. 8, no. 2S3, pp. 870–875, Aug. 2019, doi: 10.35940/ijrte.b1163.0782s319.

[21] "Heart Health and Aging," *National Institute on Aging*. https://www.nia.nih.gov/health/heart-health-and-aging

[22] J. Corliss, "The heart disease gender gap," *Harvard Health*, Sep. 01, 2022. https://www.health.harvard.edu/heart-health/the-heart-disease-gender-

[23] "What is Logistic Regression? - Logistic Regression Model Explained - AWS," *Amazon Web Services, Inc.* https://aws.amazon.com/what-is/logistic-regression/

[24] V. Kanade, "What Is Logistic Regression? Equation, Assumptions, Types, and Best Practices - Spiceworks," *Spiceworks*, Apr. 18, 2022. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-logistic-regression/

[25] M. B. Antor *et al.*, "A Comparative Analysis of Machine Learning Algorithms to Predict Alzheimer's Disease," *Journal of Healthcare Engineering*, vol. 2021, pp. 1–12, Jan. 2021, doi: 10.1155/2021/9917919.

[26] T. Wood, "Sigmoid Function," DeepAI, Sep. 27, 2020. https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

[27] G. Pulipaka, "An essential guide to classification and regression trees in R Language," *Medium*, Jun. 06, 2016. [Online]. Available: https://medium.com/@gp_pulipaka/an-essential-guide-to-classification-and-regression-trees-in-r-language-4ced657d176b

[28] GeeksforGeeks, "CART Classification And Regression Tree in Machine Learning," GeeksforGeeks, Sep. 2022, [Online]. Available: https://www.geeksforgeeks.org/cart-classification-and-regression-tree-in-machine-learning/

[29] Y. Y. Song and Y. Lu, "Decision tree methods: applications for classification and prediction," Shanghai Archives of Psychiatry, vol. 27, no. 2, pp. 130–135, Apr. 2015, doi: 10.11919/j.issn.1002-0829.215044.

[30] M. Friedl and C. E. Brodley, "Decision tree classification of land cover from remotely sensed data," Remote Sensing of Environment, vol. 61, no. 3, pp. 399–409, Sep. 1997, doi: 10.1016/s0034-4257(97)00049-7.

[31] "What is a Decision Tree | IBM." https://www.ibm.com/ae-en/topics/decision-trees

[32] P. Huilgol, "Precision and Recall | Essential Metrics for Data Analysis (Updated 2023)," Analytics Vidhya, Feb. 2023, [Online]. Available: https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/#What_is_Precision?

[33] GeeksforGeeks, "Decision Tree," GeeksforGeeks, Mar. 2023, [Online]. Available: https://www.geeksforgeeks.org/decision-tree/

[34] "1.10. Decision Trees," Scikit-learn. https://scikit-learn.org/stable/modules/tree.html

[35] S. E. R, "Understand Random Forest Algorithms With Examples (Updated 2023)," Analytics Vidhya, Mar. 24, 2023. https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

[36] GeeksforGeeks, "Bagging vs Boosting in Machine Learning," GeeksforGeeks, Jun. 01, 2022. https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/

[37] P. Vadapalli, "Bagging vs Boosting in Machine Learning: Difference Between Bagging and Boosting," upGrad Blog, Oct. 27, 2022. https://www.upgrad.com/blog/bagging-vs-boosting/

[38] "Machine Learning Random Forest Algorithm - Javatpoint," www.javatpoint.com. https://www.javatpoint.com/machine-learning-random-forest-algorithm

[39] "What is Random Forest? | IBM." https://www.ibm.com/ae-en/topics/random-forest#:~:text=Random%20forest%20is%20a%20commonly,both%20classification%20and%20regression%20problems

[40] M. B. Antor et al., "A Comparative Analysis of Machine Learning Algorithms to Predict Alzheimer's Disease," Journal of Healthcare Engineering, vol. 2021, pp. 1–12, Jan. 2021, doi: 10.1155/2021/9917919.

[41] P. Storage, "Deep Learning vs. Neural Networks," Pure Storage Blog, Mar. 2023, [Online]. Available: https://blog.purestorage.com/purely-informational/deep-learning-vs-neural-networks/

[42] "Top 10 Deep Learning Algorithms in Machine Learning [2023]," ProjectPro, Apr. 24, 2023. https://www.projectpro.io/article/deep-learning-algorithms/443

[43] V. Kanade, "What Is Linear Regression? Types, Equation, Examples, and Best Practices for 2022 - Spiceworks," Spiceworks, Apr. 03, 2023. https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-linear-regression/

[44] "What is Natural Language Processing? | IBM." https://www.ibm.com/topics/natural-language-processing

[45] I. H. Witten, E. Frank, L. E. Trigg, M. A. Hall, G. Holmes, and S. J. Cunningham, "Weka: Practical machine learning tools and techniques with Java implementations," Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems, Aug. 1999.

[46] R. Zacharski, "Training Sets, Test Sets, and 10-fold Cross-validation - KDnuggets," KDnuggets. https://www.kdnuggets.com/2018/01/training-test-sets-cross-validation.html

[47] "Decision Tree Algorithm in Machine Learning - Javatpoint," *www.javatpoint.com*. https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm

[48] P. Singh, S. Singh, and G. S. Pandi-Jain, "Effective heart disease prediction system using data mining techniques," *International Journal of Nanomedicine*, vol. Volume 13, pp. 121–124, Mar. 2018, doi: 10.2147/ijn.s124998.

[49] Rhnbyrm, "Heart Disease Classification," Kaggle, Mar. 2023, [Online]. Available: https://www.kaggle.com/code/rhnbyrm/heart-disease-classification