Heart Disease Prediction Using Machine Learning Algorithms

Sahithi Katakam ID: 11652934
Sai Charan Reddy Chinthareddy ID: 11580538
Ganesh Sai Dontineni ID: 11651338
Harini Konda ID: 11648692
Neelima Vanukuri ID: 11723884

***Abstract*—The project underscores the importance of precise and reliable prediction of heart diseases due to their significant implications on human health. Given the limitations of traditional diagnostic methods, which are prone to delays and human error, there is a critical demand for more advanced predictive systems. This project aims to enhance the accuracy of heart disease predictions by employing sophisticated machine learning models and leveraging comprehensive datasets alongside cutting-edge processing technologies. The goal is to improve the timeliness and precision of heart disease detection, thereby potentially reducing the incidence of preventable complications and improving overall patient outcomes.**

***Index Terms*—Pyspark, Random Forest Classifier, GBT, Big-Data,**

I. Introduction

Heart disease is a major health issue worldwide, leading to many deaths each year. Detecting heart disease early is crucial because it can help save lives. Traditional methods of analyzing health data are often slow and can't handle large amounts of information effectively. This is where big data technology, like Apache Spark, comes in handy. It allows us to process large datasets quickly and efficiently.

In this project, we use PySpark, which combines Python with Spark's capabilities, to predict heart disease using data from the Behavioral Risk Factor Surveillance System (BRFSS). This data includes various health indicators like body mass index (BMI), cholesterol levels, smoking status, and physical activity, which are important for predicting heart disease.

Our goals for this project are:

Data Handling: We'll use PySpark to manage and prepare large datasets for analysis.

Model Building and Testing: We'll create and test different machine learning models, such as Random Forest, Decision Trees, and Gradient Boosting Trees, to see how well they can predict heart disease.

Model Comparison: We'll compare these models to find out which one works best.

Understanding Scalability: We'll explore how well our data processing can scale up when using Spark, discussing the benefits and challenges.

II. Existing Solutions

The current approach utilizes decision tree classifiers , linear regression models, SVM and KNN as foundational tools for predicting heart disease:

A. Decision Tree Classifier:

Characteristics: Utilizes a non-linear, rule-based approach to segment data into branches, efficiently handling both categorical and numerical data.

Strengths: Highly interpretable, mimicking human decision-making processes and requiring no feature scaling, which simplifies data preparation.

Limitations: Prone to over-fitting with complex tree structures; sensitive to minor data variations, which can destabilize model consistency.

B. Linear Regression:

Characteristics: Establishes a linear relationship between dependent and independent variables, commonly used for predictions and forecasting.

Strengths: Simple to implement and interpret, offering insights into the significance of each predictor.

Limitations: Assumes linearity between variables, which may not always hold, and is sensitive to outliers that can skew results.

In this project, we have employed both decision tree classifiers and linear regression models as our foundational analytical tools.

C. K-Nearest Neighbors (KNN):

K-Nearest Neighbors (KNN) is a simple, instance-based learning algorithm where the function is only approximated locally and all computation is deferred until classification. The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other.

Distance Metrics: KNN works by finding the distances between a query and all the examples in the data, selecting the specified number. K of examples (K-nearest neighbors) closest to the query, then votes for the most frequent label (in the case of classification).

Lazy Learning: KNN is a lazy learning algorithm because it does not have a specialized training phase or it is very minimal, which means the training phase is pretty fast.

Simplicity: Very easy to understand and implement.

No assumptions about data: No assumptions about the underlying data distribution, which is an advantage since many real-world datasets do not follow mathematical theoretical assumptions.

Adaptability: Highly adaptable. New training examples can be added easily.

Computationally expensive: The algorithm stores all training data, leading to potentially huge storage requirements.

Sensitive to the scale of the data and irrelevant features.

High memory requirement as it needs to store all of the training data. Performance degrades with high dimensionality (curse of dimensionality).

D. Support Vector Machines (SVM)

Support Vector Machine (SVM) is a powerful, supervised machine learning algorithm used for classification or regression problems. It works by finding the best hyperplane that separates data points into two classes in the feature space.

Maximizing the Margin: SVM attempts to maximize the margin between the different classes. The data points that are closest to the hyperplane (which influences where the hyperplane is placed) are called support vectors.

Kernel Trick: Uses a technique called the kernel trick to transform data into a higher dimension where a hyperplane can be used to separate data points. Kernels commonly used include linear, polynomial, RBF (radial basis function), and sigmoid.

Effective in high-dimensional spaces, even in cases where the number of dimensions exceeds the number of samples.

Memory efficient: Uses a subset of training points (support vectors), so it is also memory efficient.

Versatility: Different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

Requires full labeling of input data: Unsuitable for unsupervised tasks, such as clustering.

Sensitive to the tuning of parameters and the choice of the kernel.

Poor performance with overlapping classes: Struggles with datasets where classes are heavily overlapping.

Scalability: Computationally intensive, thus not suitable for large datasets.

III. Comparative Analysis

While examining big data components like spark, Hadoop and Cassandra for our problem statement, we focused on their strengths and suitability depending upon the requirement of our project.

Table: Accuracy comparison

Algorithm	Accuracy
Support Vector machine	83%
Decision tree	79%
Linear regression	78%
k-nearest neighbor	87%

Fig. 1. Accuracy Percentages of Different Models

IV. Comparison among the three components:

A. Hadoop:

- Hadoop's model is comparatively slower in performance compared to spark, especially for iterative algorithms which are quite common in machine learning tasks.
- Integration of machine learning pipelines using Hadoop is quite complicated, it demands extra frameworks which would increase the complexity of the project, so choosing spark is beneficial.
- Maintenance of Hadoop cluster is not that easy; it requires more effort to set up and manage when compared with spark solution.

B. Cassandra:

- Cassandra supports decentralized data models which is not suitable for performing complex operations such as those in predictive modeling for heart diseases.
- Cassandra does not have built-in capability to support complicated analytical tasks such as data preprocessing, model training, these are well supported by spark as it has built-in libraries and tools

C. Spark:

- Spark has the ability to process real time data which makes it quite suitable to carry out predictive modeling and monitoring.
- Spark has better in-memory computation capacity which helps in faster data processing when compared with traditional frameworks like hadoop.
- Spark has distributed processing engine which yields better performance and also has better scope to perform huge range of analytical tasks

Due to the flexibility and easy way of implementation, spark is well-suited for our project compared to Hadoop or Cassandra

V. Method/Algorithm from the paper

The suggested model for predicting heart disease leverages Random Forest, a sophisticated machine learning algorithm, alongside Gradient Boosting, an ensemble technique. This combination aims to enhance the accuracy and robustness of the predictive analysis.

A. Gradient Boosted Trees (GBT)

Gradient Boosted Trees is an ensemble learning technique that builds models incrementally in the form of a series of decision trees added sequentially to correct the errors of the previous trees. Here are key points about GBT:

Sequential Model Building: Unlike bagging methods (e.g., Random Forest), GBT builds one tree at a time, where each new tree is created to correct the mistakes made by previously built trees.

Each tree tries to improve on the residual errors of the entire ensemble. Loss Minimization: GBT minimizes a specified loss function (like squared error for regression or logistic loss

for classification), which makes it highly flexible in handling various types of predictive modeling tasks.

Regularization Techniques: Includes several regularization techniques, such as learning rate (shrinkage) and depth control, to prevent overfitting. This makes GBT very robust, especially for complex data sets with many features.

High Accuracy: Often provides substantial predictive accuracy that can outperform many other models, particularly on heterogeneous datasets with complex, non-linear relationships.

B. Random Forest

Random Forest is another ensemble learning method known for its simplicity and effectiveness, particularly in classification and regression tasks.

Here are some critical points about Random Forest: Parallel Model Building: Builds multiple decision trees in parallel and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Robustness and Accuracy: Typically exhibits high performance with default parameter settings, making it very user-friendly for practitioners.

Handling Overfitting: Due to the method of averaging multiple trees, Random Forest is less prone to overfitting compared to individual decision trees.

VI. Model Development

The planned project seeks to address the limitations of current solutions by utilizing Random Forest and Gradient Boosting models within the PySpark environment. This approach is designed to develop a dynamic and effective system for predicting heart disease.

A. Key Steps

- Importing Libraries: The script initiates by importing essential libraries and modules such as Spark session from PySpark, pandas for data handling, and other utilities necessary for data manipulation.
- Establishing Spark Session: A Spark session is initiated using PySpark, providing a critical environment for processing and analyzing the dataset.
- Loading Data: The URL dataset is read from a CSV file into a panda DataFrame for detailed examination. This dataset includes features of URLs and their labels, where ”Label” indicates whether a URL is ”Malicious” (1) or ”Safe” (0).
- Visualizing Data: Visualization techniques like box plots and histograms are employed to explore the distribution of features in the dataset.
- Correlation Analysis: A correlation heatmap is generated to display the relationships between features, which assists in evaluating the importance of each feature.
- Splitting Data: The dataset is split into training and testing portions, with 70% allocated for training purposes and 30
- Implementing Gradient Boosting: Besides the Random Forest model, a Gradient Boosting classifier is also trained using the same training data to evaluate and compare the effectiveness of different models.

VII. Results

A. Correlation Heatmap

The heatmap Fig:2 illustrates the correlations between key numeric variables, with darker colors representing stronger relationships. A notable positive correlation exists between age and heart disease, supporting medical literature. In contrast, physical activity shows a negative correlation, aligning with the understanding that active lifestyles can reduce heart disease risk.

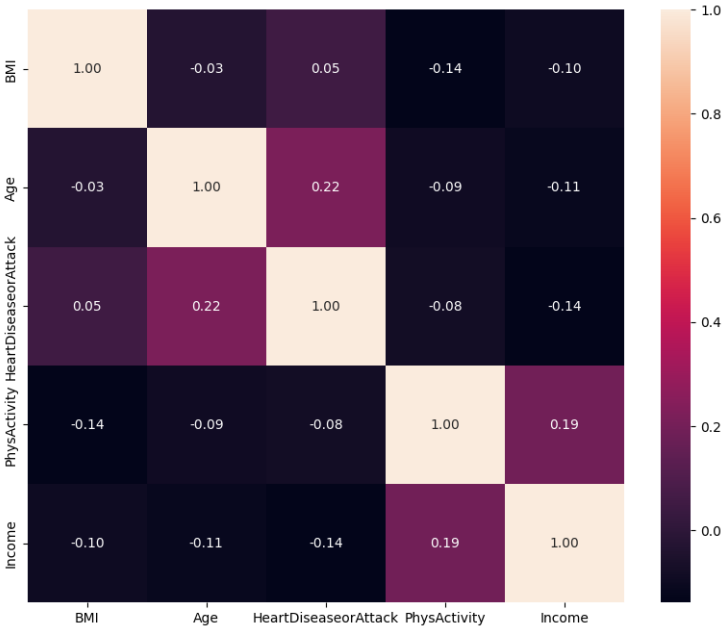

Fig. 2. Heat Map Distribution

B. Boxplot of Age Distribution by Heart Disease or Attack Status

The boxplot Fig:3 offers insight into the age distribution among individuals with and without heart disease or attacks. Older age groups tend to have a higher occurrence of heart disease, which is consistent with the known risk increase as age advances.

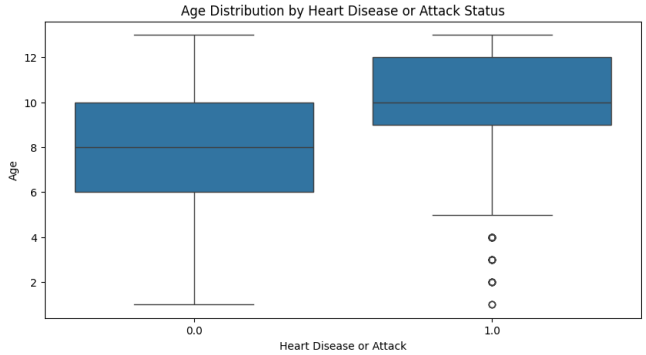

Fig. 3. Box Plot of Age Distribution

C. Histogram of BMI Distribution

Lastly, the histogram of BMI distribution Fig:4 across the dataset shows a right-skewed pattern, indicating a prevalence of higher BMI scores. This skewness could be an indicator of the general population's obesity trend, which is a risk factor for heart disease.

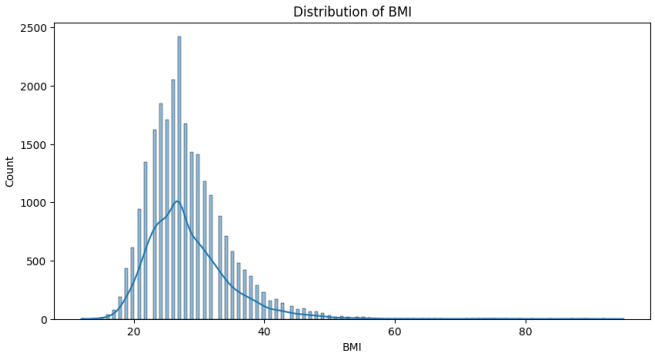

Fig. 4. BMI Histrogram

Our project evaluated three machine learning models: Random Forest, Decision Tree, and Gradient Boosted Trees (GBT). The line graph presents a comparison of their performance across four key metrics: accuracy, weighted precision, weighted recall, and F1 score. The Random Forest model exhibited consistent high performance across all metrics, marking its robustness. The Decision Tree showed slightly lower values, indicating potential overfitting or simplicity. The GBT had similar precision and recall but improved slightly in F1 score compared to the Decision Tree, highlighting its strength in balancing false positives and negatives.

D. Gradient Boosted Trees (GBT) Radar Chart

The radar chart for GBT Fig:**??** captures its performance, plotting the accuracy, precision, recall, and F1 score as axes emanating from a central point. The model's scores stretch towards the edges of the web, indicating high performance. The close proximity of precision and recall suggests a balanced prediction for both classes.

E. Random Forest Radar Chart

Similar to the GBT, the radar chart for the Random Forest classifier Fig:6 displays a balance across all metrics. The model excels particularly in accuracy, with all points pushing towards the outer limits of the chart, signaling a well-rounded model performance.

F. Decision Tree Radar Chart

The radar chart for the Decision Tree Fig:7 shows a slight reduction in metrics compared to the Random Forest, with the F1 score exhibiting the most significant decrease. This indicates a lower performance in handling false positives and negatives, which is critical in medical predictions.

VIII. Comparison

Model Performance Comparison:

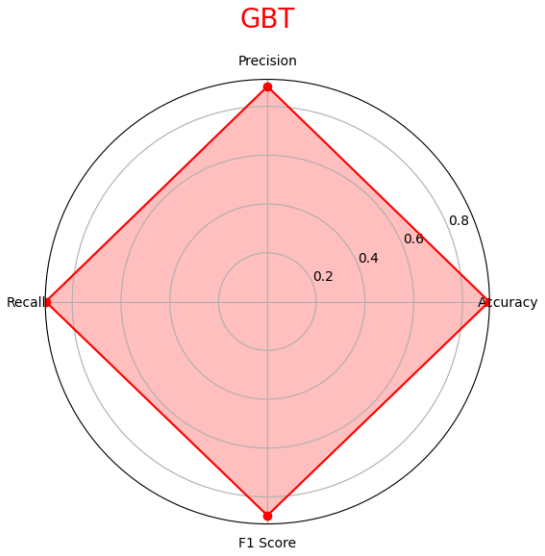

Fig. 5. Radar Charts of GBT model metrics

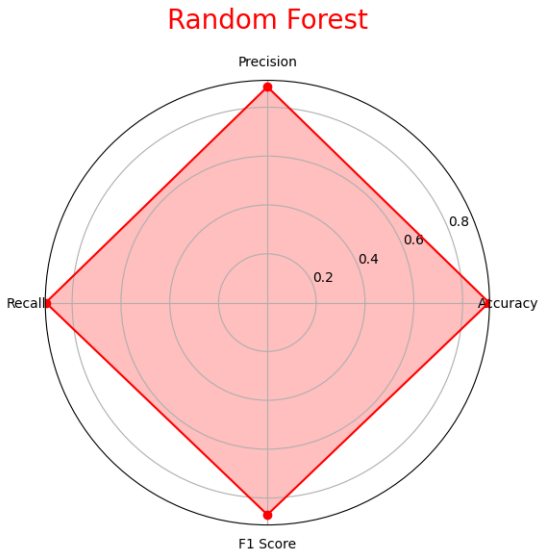

Fig. 6. Radar Charts of Random Forest model metrics

A. Random Forest

- Accuracy: Random Forest typically exhibits high accuracy due to its ensemble method, which aggregates the predictions of multiple decision trees to make a final decision. This method reduces the variance and avoids over-fitting, often resulting in superior accuracy.
- Weighted Precision: Due to its ensemble nature, Random Forest can handle imbalanced classes better than single decision trees, often resulting in higher precision.
- Weighted Recall: Similarly, the recall is generally high as Random Forest can correctly classify more positive instances by averaging out biases from individual trees.
- F1 Score: Combining high precision and recall, Random Forest usually achieves a high F1 score, indicating a

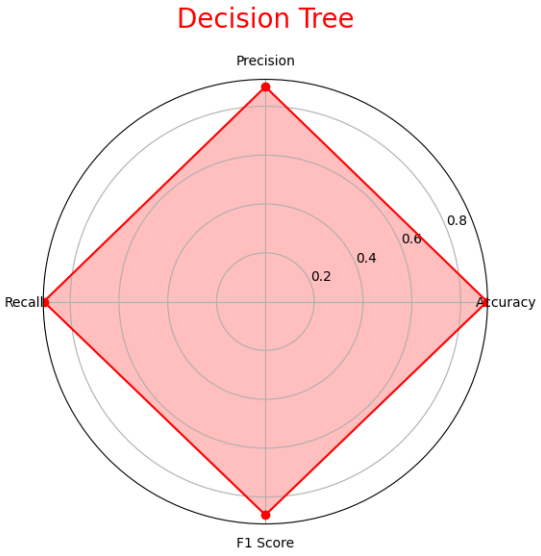

Fig. 7. Radar Charts of Decision Tree model metrics

strong balance between recall and precision.

B. Decision Tree

- Accuracy: While potentially very accurate, decision trees often suffer from over-fitting, especially with more complex datasets. This might result in lower accuracy on unseen data compared to Random Forest.
- Weighted Precision: Precision can vary significantly with Decision Trees, particularly if the tree depth isn't adequately controlled.
- Weighted Recall: Recall might be high, but it is usually less reliable than Random Forest due to over-fitting risks that can skew the tree's decision boundaries.
- F1 Score: Decision Trees might achieve a reasonable F1 score, but it can be less consistent across different runs or data splits due to the model's sensitivity to the specific structure of the training data.

C. Gradient Boosted Trees (GBT)

- Accuracy: GBT often rivals or exceeds Random Forest in accuracy as it builds trees sequentially to correct errors from previous trees, thus refining predictions.
- Weighted Precision: GBT can provide very high precision, as it focuses on correcting misclassified data points in successive iterations.
- Weighted Recall: Recall rates are generally high, as GBT effectively learns from mis-classifications and improves its ability to identify difficult cases.
- F1 Score: Due to high precision and recall, GBT typically scores well on the F1 metric, reflecting robust overall performance.

IX. Potential Applications

- 1.Enhanced Cybersecurity Measures: The model can be integrated into cybersecurity systems to automatically detect and flag malicious URLs. This helps in preventing phishing attacks and other forms of cyber threats, enhancing the overall security of digital platforms.
- Real-Time Threat Detection: By implementing the model within real-time monitoring systems, organizations can actively scan and analyze URLs accessed within their networks. This allows for immediate detection and mitigation of threats, safeguarding sensitive data and infrastructure from potential breaches.
- Content Filtering: The project can be used to develop sophisticated content filtering solutions for educational institutions and workplaces. It can help in blocking access to harmful or inappropriate websites based on their URL classification, ensuring compliance with regulatory policies, and maintaining a safe online environment.
- Consumer Protection Applications: The model can be deployed in consumer protection software, such as parental control applications or web browsing safety tools. It helps in automatically screening and blocking malicious websites, thereby protecting users from scams, malware, and other online risks.
- Data Analytics for Cyber Intelligence: The insights gained from the classification and analysis of URLs can be utilized by governmental and non-governmental organizations for cyber intelligence purposes. This would involve tracking evolving cyber threat patterns and developing strategies to counter new malware and phishing tactics effectively.

A. Analysis

As we can see in below Fig:8 all three models performed similarly with high accuracy, precision, and recall rates, which is excellent for a predictive healthcare application where the cost of false negatives can be high (missing a diagnosis). However, the models' precision at 88% indicates that there is still room for improvement, as 12% of the positive predictions were false positives. The GBT model had a slightly better balance of precision and recall as indicated by the F1 score, but the difference is minimal.

X. Pros and Cons

A. Pros

- Improved Accuracy:
  Machine learning models can achieve high levels of accuracy in predicting heart diseases by learning from large datasets, which often contain complex and non-linear relationships that traditional statistical methods might miss.
- Early Detection:
  These algorithms can identify risk factors and symptoms much earlier than conventional methods, allowing for timely intervention that can prevent the progression of the disease.
- Personalization:
  Machine learning enables the development of personalized medicine strategies. Algorithms can analyze data

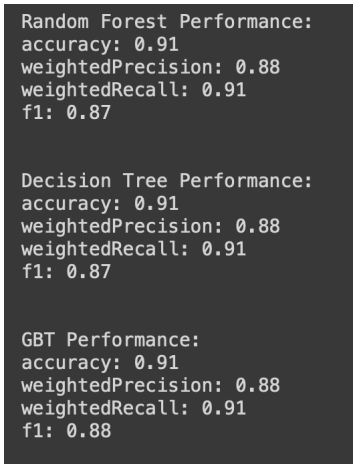

Fig. 8. Evaluation Metrics

specific to individual patients, thus tailoring treatment plans that are optimized for better outcomes based on predictive analytics.

- Cost-Effectiveness:
  By automating the diagnostic process and reducing the need for repetitive tests, machine learning can help lower healthcare costs and improve resource utilization.
- Dynamic Learning:
  Machine learning systems can continuously learn and improve from new data, which helps in refining the algorithms and enhancing their accuracy over time without human intervention.

B. Cons

- 1. Data Privacy Concerns: The use of sensitive health data in machine learning models raises significant privacy concerns. Ensuring the security and confidentiality of patient data is crucial and challenging.
- Dependency on Quality Data: The performance of machine learning models is heavily dependent on the quality, completeness, and bias of the data used for training. Poor data quality can lead to inaccurate predictions and potentially harmful outcomes.
- Complexity in Interpretation: Some advanced machine learning models, especially deep learning models, act as ”black boxes” where it’s difficult to interpret how decisions are made. This lack of transparency can be a major issue in medical applications where understanding the decision-making process is essential.
- Over-fitting Risks: Machine learning models, particularly those that are complex, can over-fit the training data, making them perform well on training data but poorly on unseen data. This can be misleading in predicting heart diseases accurately.
- High Initial Costs: Developing robust machine learning systems for heart disease prediction requires significant upfront investments in terms of technology, professional expertise, and time, which might not be feasible for all healthcare providers.

These pros and cons highlight the potential and challenges of applying machine learning algorithms in the field of heart disease prediction. Careful consideration and balanced approach are necessary to leverage the benefits while mitigating the drawbacks effectively.

REFERENCES

[1] Singh, A., Kumar, R. (2020). Heart Disease Prediction Using Machine Learning Algorithms. 2020 International Conference on Electrical and Electronics Engineering (ICE3). https://doi.org/10.1109/ice348803.2020.912295.

[2] S. Ouyang, ”Research of Heart Disease Prediction Based on Machine Learning,” 2022 5th International Conference on Advanced Electronic Materials, Computers and Software Engineering (AEMCSE), Wuhan, China, 2022.

[3] A. Lakshmi and R. Devi, ”Heart Disease Prediction Using Enhanced Whale Optimization Algorithm Based Feature Selection With Machine Learning Techniques,” 2023 12th International Conference on System Modeling Advancement in Research Trends (SMART), Moradabad, India, 2023.

[4] N. Mohan, V. Jain and G. Agrawal, ”Heart Disease Prediction Using Supervised Machine Learning Algorithms,” 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India.

[5] P. Sujatha and K. Mahalakshmi, ”Performance Evaluation of Supervised Machine Learning Algorithms in Prediction of Heart Disease,” 2020 IEEE International Conference for Innovation in Technology (INOCON), Bangluru, India.

[6] H. E. Hamdaoui, S. Boujraf, N. E. H. Chaoui and M. Maaroufi, ”A Clinical support system for Prediction of Heart Disease using Machine Learning Techniques,” 2020 5th International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Sousse, Tunisia, 2020.