Real Time Machine Learning Detection of Heart Disease

Submitted in partial fulfillment of the requirements

for the award of

Bachelor of Engineering Degree in Computer Science and Engineering

By

HANEESH CHELLUBOINA

(Reg. No. 38110105)

B.V.S.SATYANARAYANA

(Reg No. 38110071)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING
SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119, TAMILNADU

MARCH 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

(Established under Section 3 of UGC Act, 1956)

JEPPIAAR NAGAR, RAJIV GANDHI SALAI

CHENNAI– 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **CHELLUBOINA HANEESH (38110105) and B.V.S.SATYANARAYANA (3811071)** who carried out the project entitled "**Agriculture Crop Recommendation System using Machine Learning** " under my supervision from December 2021 to March 2022.

Internal Guide:
Dr. B. U. Anu Barathi, M. E., PhD

Head of the Department:

Dr. L. Lakshmanan M.E., Ph.D.

Dr. VIGNESHWARI, M.E., Ph.D.

Submitted for Viva voce Examination held on_____

Internal Examiner **External Examiner**

DECLARATION

I **CHELLUBOINA HANEESH (38110105) and B.V.S.SATYANARAYANA (3811071)** hereby declare that the Project Report entitled **Agriculture Crop Recommendation System using Machine Learning** is done by me under the guidance of **Dr. B. U. Anu Barathi., M.E., PhD** professor, Dept of CSE at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: Chennai **SIGNA TUR E OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D., Dean, School of Computing and Dr. S. VIGNESHWARI, M.E., Ph.D. and Dr.L. Lakshmanam M.E.,Ph.D., Head of the Department**, **Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide Dr.Anu Bharathi,M.E., P.hD., for her valuable guidance, suggestions andconstant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Computer Science and Engineering** who were helpful in many ways For the Completion of the project.

.

ABSTRACT:

According to recent survey by UN agency (World health organization) seventeen.9 million individuals die annually owing to heart connected diseases and it's increasing chop-chop. With the increasing population and illness, it's become a challenge to diagnosis illness and providing the suitable treatment at the proper time. however, there'sa light-weight of hope that recent advances in technologyhave accelerated the general public health sector by developing advanced useful medical specialty solutions. This paper aims at analyzing the assorted datamining techniques particularly Naive Thomas Bayes, Random Forest Classification, call tree and Support Vector Machine by employing a qualified dataset for cardiopathy prediction that is include varied attributes like gender, age, hurting sort, pressure level, blood glucose etc. The analysis includes finding the correlations between the assorted attributes of the dataset by utilizing the quality data processing techniques and thus mistreatment the attributes befittingly topredict the possibilities of a cardiopathy. These machine learning techniques take less time for the prediction of the illness with a lot of accuracy which can cut back the get rid of valuable lives everywhere the planet.

LIST OF TABLES

LIST OF FIGURES

TABLE OF CONTENTS

Chapter 1
INTRODUCTION

1.1 Overview:

Health is one in every of the planet challenges for humanity. World health organization (WHO) has mentioned that for a personal correct health is that the elementary right. thus to stay individuals match and healthy correct health care services ought to be provided. thirty-one proportion of all deaths worldwide square measure due to heart connected problems. identification and treatment of cardiovascular disease is incredibly complicated, significantly in developing countries, because of the shortage of diagnostic devices and a shortage of physicians and alternative resources poignant correct prediction and treatment of internal organ patients. With this concern within the recent times engineering and machine learning techniques square measure being employed to develop code to help doctors in creating call of cardiovascular disease within the preliminary stage. Early stage detection of the malady and predicting the likelihood of an individual to be in danger of cardiovascular disease will scale back the death rate. Medical data processing techniques square measure employed in medical knowledge to extract substantive patterns and data. Medical data has redundancy, multi- attribution, unity and an in depth relationship with time. downside the matter} of mistreatment the large volumes of information effectively becomes a serious problem for the health sector. data processing provides the methodology and technology to convert these knowledge mounds into helpful decision-making data. This postulation system for cardiovascular disease would facilitate Cardiologists intaking faster choices in order that a lot of patients will receive treatments inside a shorter amount of your time.

1.2 Scope Of the Project:

The main motivation of doing this research is to present a heart disease prediction model for the prediction of occurrence of heart disease. Further, this research work is aimed towards identifying the best classification algorithm for identifying the possibility of heart disease in a patient. This work is justified by performing a comparative study and analysis using several classification algorithms used at different levels of evaluations. Although these are commonly used machine learning algorithms, the heart disease prediction is a vital task involving highest possible accuracy.

1.3 Domain Overview:

1.3.1 MACHINE LEARNING

Machine Learning is a system that can learn from example through self-improvement and without being explicitly coded by programmer. The breakthrough comes with the idea that a machine can singularly learn from the data(i.e., example) to produce accurate results.

Machine learning combines data with statistical tools to predict an output. This output is then used by corporate to makes actionable insights. Machine learning is closely related to data mining and Bayesian predictive modeling. The machine receives data as input, use an algorithm to formulate answers.

A typical machine learning tasks are to provide a recommendation. For those who have a Netflix account, all recommendations of movies or series are based on the user's historical data. Tech companies are using unsupervised learning to improve the user experience with personalizing recommendation.

Machine learning is also used for a variety of task like fraud detection, predictive maintenance, portfolio optimization, automatize task and so on.

1.4 Machine Learning vs. Traditional Programming

Traditional programming differs significantly from machine learning. In traditional programming, a programmer code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become unsustainable to maintain.

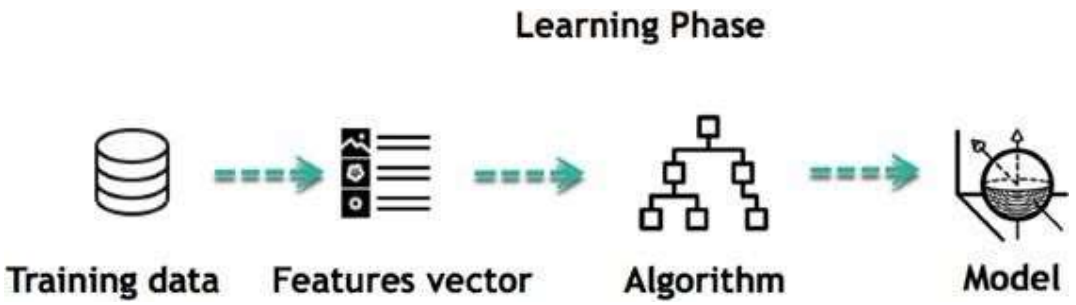
fig 1.1 Machine Learning

1.4.1 How does Machine learning work?

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feed a previously unseen example, the machine has difficulties to predict.

The core objective of machine learning is the **learning** and **inference**. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the **data**. One crucial part of the data scientist is to choose carefully which data to provide to the machine. The list of attributes used to solve a problem is called a **feature vector.** You can think of a feature vector as a subset of data thatis used to tackle a problem.

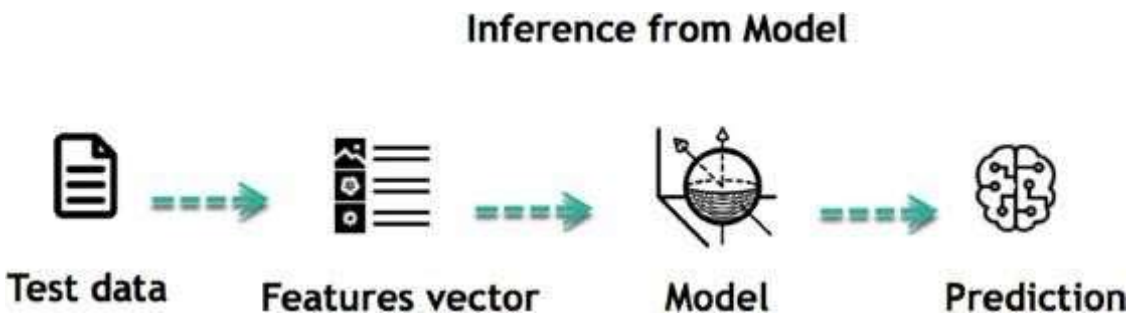
The machine uses some fancy algorithms to simplify the reality and transform this discovery into a **model**. Therefore, the learning stage is used to describe the data and summarize it into a model.



For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high-end restaurant: This is the model

1.5 Inferring

When the model is built, it is possible to test how powerful it is on never-seen- before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the modelpreviously trained to make inference on new data.

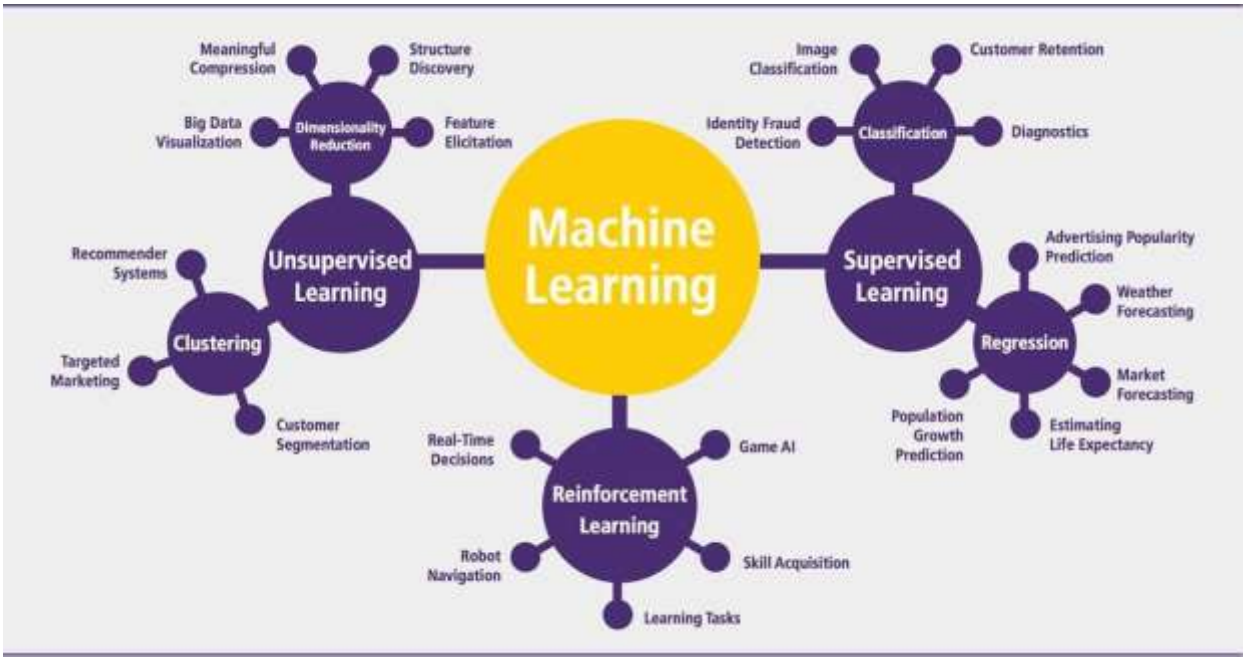


The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Fig 1.2 Machine learning Algorithms and where they are used?



Machine learning can be grouped into two broad learning tasks: Supervised and unsupervised. There are many other algorithms shown in above fig.1.2

Supervised learning

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.

You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can

only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Algorithm Name	Description	Type
Linear regression	Finds a way to correlate each feature to the output to help predict future values.	Regression
Logistic regression	Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors)	Classification
Decision	Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes	Regression

tree	(e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made	Classification
Naive Bayes	The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event.	Regression Classification
Support vector machine	Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non- linear solver.	Regression (nc very common Classification
Random forest	The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.	Regression Classification
AdaBoost	Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome	Regression Classification
Gradient-boosting trees	Gradient-boosting trees is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.	Regression Classification

1.7 Unsupervised learning

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns)

You can use it when you do not know how to classify the data, and you want the algorithm to find patterns and classify the data for you

Algorithm	Description	Type
K-means clustering	Puts data into some groups (k) that each contains data withsimilar characteristics (as determined by the model, not in advance by humans)	Clustering
Gaussian mixturemodel	A generalization of k-means clustering that provides moreflexibility in the size and shape of groups (clusters	Clustering
Hierarchical clustering	Splits clusters along a hierarchical tree to form aclassification system. Can be used for Cluster loyalty-card customer	Clustering
Recommendersystem	Help to define the relevant data for making arecommendation.	Clustering
PCA/T-SNE	Mostly used to decrease the dimensionality of the data.The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.	Dimension Reduction

1.8 Application of Machine learning

Augmentation:

- Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions. The primary user is to reduce errors due to human bias.

Automation:

- Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

Finance Industry

- Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

Government organization

- The government makes use of ML to manage public safety and utilities. Take the example of China with the massive face recognition. The government uses Artificial intelligence to prevent jaywalker.

Healthcare industry

- Healthcare was one of the first industry to use machine learning with image detection.

Marketing

- Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

1.9 Example of application of Machine Learning in Supply Chain

Machine learning gives terrific results for visual pattern recognition, opening up many potential applications in physical inspection and maintenance across the entire supply chain network.

Unsupervised learning can quickly search for comparable patterns in the diverse dataset. In turn, the machine can perform quality inspection throughout the logistics hub, shipment with damage and wear.

For instance, IBM's Watson platform can determine shipping container damage. Watson combines visual and systems-based data to track, report and make recommendations in real-time.

In past year stock manager relies extensively on the primary method to evaluate and forecast the inventory. When combining big data and machine learning, better forecasting techniques have been implemented (an improvement of 20 to 30 % over traditional forecasting tools). In term of sales, it means an increase of 2 to 3 %due to the potential reduction in inventory costs.

Example of Machine Learning Google Car

For example, everybody knows the Google car. The car is full of lasers on the roof which are telling it where it is regarding the surrounding area. It has radar in the front, which is informing the car of the speed and motion of all the cars around it. It uses all of that data to figure out not only how to drive the car but also to figure out and predict what potential drivers around the car are going to do. What's impressive is that the car is processing almost a gigabyte a second of data.

Deep Learning

Deep learning is a computer software that mimics the network of neurons in a brain. It is a subset of machine learning and is called deep learning because it makes use of deep neural networks. The machine uses different layers to learnfrom the data. The depth of the model is represented by the number of layers in the model. Deep learning is the new state of the art in term of AI. In deep learning, the learning phase is done through a neural network.

Reinforcement Learning

Reinforcement learning is a subfield of machine learning in which systems are trained by receiving virtual "rewards" or "punishments," essentially learning bytrial and error. Google's DeepMind has used reinforcement learning to beat a

human champion in the Go games. Reinforcement learning is also used in video games to improve the gaming experience by providing smarter bot.

One of the most famous algorithms are:

- Q-learning
- Deep Q network
- State-Action-Reward-State-Action(SARSA)
- Deep Deterministic Policy Gradient(DDPG)

1.10 Applications/ Examples of deep learning applications

AI in Finance: The financial technology sector has already started using AI tosave time, reduce costs, and add value. Deep learning is changing the lending industry by using more robust credit scoring. Credit decision-makers can use AIfor robust credit lending applications to achieve faster, more accurate risk assessment, using machine intelligence to factor in the character and capacity of applicants.

Underwrite is a Fintech company providing an AI solution for credit makers company. underwrite.ai uses AI to detect which applicant is more likely to payback a loan. Their approach radically outperforms traditional methods.

AI in HR: Under Armour, a sportswear company revolutionizes hiring and modernizes the candidate experience with the help of AI. In fact, Under Armour Reduces hiring time for its retail stores by 35%. Under Armour faced a growing popularity interest back in 2012. They had, on average, 30000 resumes a month. Reading all of those applications and begin to start the screening and interview process was taking too long. The lengthy process to get people hired and on- boarded impacted Under Armour's ability to have their retail stores fully staffed, ramped and ready to operate.

At that time, Under Armour had all of the 'must have' HR technology in place such as transactional solutions for sourcing, applying, tracking and onboarding but those tools weren't useful enough. Under armour choose **HireVue**, an AI provider for HR solution, for both on-demand and live interviews. The results were bluffing; they managed to decrease by 35% the time to fill. In return, the hired higher qualitystaffs.

AI in Marketing: AI is a valuable tool for customer service management and personalization challenges. Improved speech recognition in call-center management and call routing as a result of the application of AI techniques allowsa more seamless experience for customers.

For example, deep-learning analysis of audio allows systems to assess a customer's emotional tone. If the customer is responding poorly to the AI chatbot, the system can be rerouted the conversation to real, human operators that take over the issue.

Apart from the three examples above, AI is widely used in other sectors/industries.

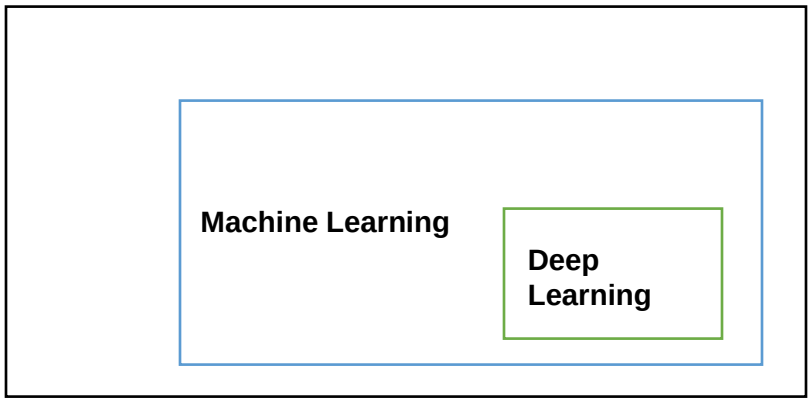


Fig 1.3 Artificial Intelligence

Difference between Machine Learning and Deep Learning

Machine Learning	Deep Learning

Data Dependencies	Excellent performances on a small/medium dataset	Excellent performance on a big dataset
Hardware dependencies	Work on a low-end machine.	Requires powerful machine, preferably with GPU: DL performs a significant amount of matrix multiplication
Feature engineering	Need to understand the features that represent the data	No need to understand the best feature that represents the data
Execution time	From few minutes to hours	Up to weeks. Neural Network needs to compute a significant number of weights
Interpretability	Some algorithms are easy to interpret (logistic, decision tree), some are almost impossible (SVM, XG Boost)	Difficult to impossible

When to use ML or DL?

In the table below, we summarize the difference between machine learning and deep learning.

	Machine learning	**Deep learning**
Training dataset	Small	Large
Choose features	Yes	No
Number of algorithms	Many	Few
Training time	Short	Long

With machine learning, you need fewer data to train the algorithm than deep learning. Deep learning requires an extensive and diverse set of data to identify the underlying structure. Besides, machine learning provides a faster-trained model. Most advanced deep learning architecture can take days to a week to train. The advantage of deep learning over machine learning is it is highly accurate. You do not need to understand what features are the best representation of the data; the neural network learned how to select critical features. In machine learning, you need to choose for yourself what features to include in the model.

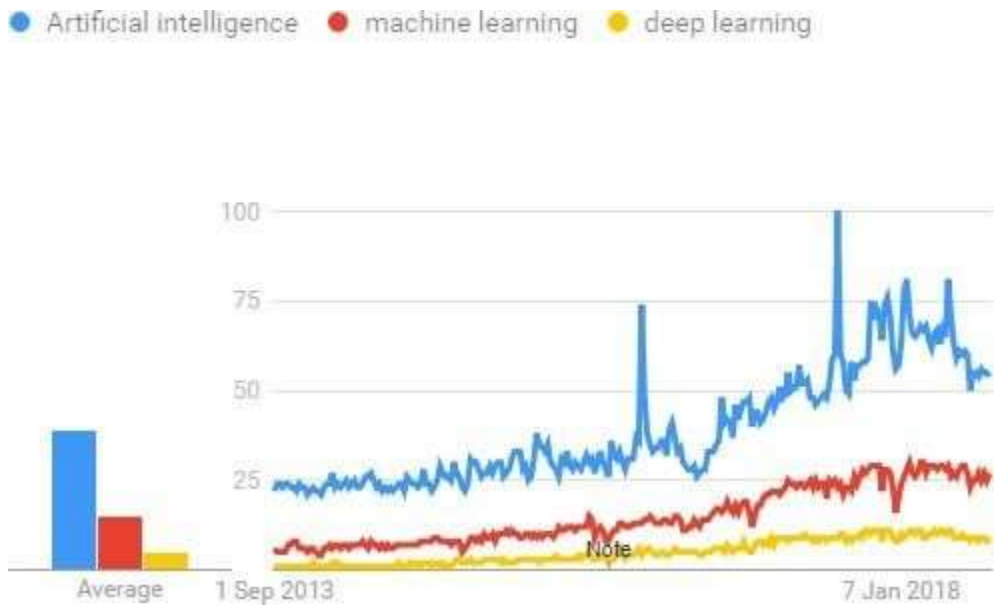


Fig 1.4 Tensor Flow

The most famous deep learning library in the world is Google's TensorFlow.Google product uses machine learning in all of its products to improve the search engine, translation, image captioning or recommendations.

To give a concrete example, Google users can experience a faster and more refined the search with AI. If the user types a keyword a the search bar, Google provides a recommendation about what could be the next word.

Google wants to use machine learning to take advantage of their massive datasets to give users the best experience. Three different groups use machine learning:

- Researchers
- Data scientists
- Programmers.

They can all use the same toolset to collaborate with each other and improve their efficiency.

Google does not just have any data; they have the world's most massive computer, so TensorFlow was built to scale. TensorFlow is a library developed by the Google Brain Team to accelerate machine learning and deep neural network research.

It was built to run on multiple CPUs or GPUs and even mobile operating systems, and it has several wrappers in several languages like Python, C++ or Java.

In this tutorial, you will learn

TensorFlow Architecture

Tensorflow architecture works in three parts:

- Preprocessing the data
- Build the model
- Train and estimate the model

It is called Tensorflow because it takes input as a multi-dimensional array, also known as **tensors**. You can construct a sort of **flowchart** of operations (called a Graph) that you want to perform on that input. The input goes in at one end, and then it flows through this system of multiple operations and comes out the otherend as output.

This is why it is called TensorFlow because the tensor goes in it flows through a list of operations, and then it comes out the other side.

Where can Tensorflow run?

TensorFlow can hardware, and software requirements can be classified into Development Phase: This is when you train the mode. Training is usually done on your Desktop or laptop.

Run Phase or Inference Phase: Once training is done Tensorflow can be run on many different platforms. You can run it on

- Desktop running Windows, macOS or Linux
- Cloud as a web service
- Mobile devices like iOS and Android

You can train it on multiple machines then you can run it on a different machine, once you have the trained model.

The model can be trained and used on GPUs as well as CPUs. GPUs were initially designed for video games. In late 2010, Stanford researchers found that GPU was also very good at matrix operations and algebra so that it makes them very fast for doing these kinds of calculations. Deep learning relies on a lot of matrix multiplication. TensorFlow is very fast at computing the matrix multiplication because it is written in C++. Although it is implemented in C++, TensorFlow can be accessed and controlled by other languages mainly, Python.

Finally, a significant feature of TensorFlow is the TensorBoard. The TensorBoard enables to monitor graphically and visually what TensorFlow is doing.

List of Prominent Algorithms supported by TensorFlow

- Linear regression: tf. estimator. LinearRegressor
- Classification:tf. estimator. Linear Classifier
- Deep learning classification: tf. estimator. DNNClassifier
- Deep learning wipe and deep: tf. estimator. DNNLinearCombinedClassifier
- Booster tree regression: tf. estimator. BoostedTreesRegressor
- Boosted tree classification: tf. estimator. BoostedTreesClassifier

Chapter 2

Literature Survey:

S.No	Title , Year	Author	props	cons
1.	Using the Extreme Learning Machine (ELM) technique for heart disease diagnosis , 2020	Salam Ismaeel, Ali Miri et al	Extreme learning machine (ELM), Neural Networks, Heart Disease, Prediction and Diagnosis Systems, Pattern Classification	•This method is not suitable for all types of data. •Efficiency is not good when compared to other methods.
2.	Prediction System for heart disease using Naive Bayes , 2020.	Shadab Adam Pattekari and Asma Parveen	Heart disease, Naive based classifier, Particle swarm optimization, Feature selection	•Accuracy is less when compared to other methods. •Heart disease cannot be predicted correctly.
3.	Health Gear: a real-time wearable system for monitoring and analyzing physiological signals , 2019	N. Oliver and F. F. Mangas	Bluetooth , health gear , sensors.	•This method is not suitable for all types of data. •Performance of the overall system is not very good.
4.	AMON: a wearable multi parameter medical monitoring and alert system , 2019	U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E.B. Schenker, F. Catarsi, L. Coluccini, A. Belardinelli, D. Shklarski, A. Menachem, E. Hirt, R. Schmid, and M. Vuskovic	Medical device, multiparameter, telemedicine, validation, wearable, wrist-worn.	• Less efficient and less robust when compared to other methods. •Consumes a large amount of time when compared to other methods.
5.	A biomedical decision support system using LS-SVM classifier with an efficient and new parameter regularization procedure for diagnosis of heart valve diseases , 2018.	Emre Çomak , Ahmet Arslan	Support Vector Machine (SVM) , Renyi's entropy , Logistic regression , Doppler Heart Sounds (DHS).	• Less flexible when compared to other methods. •The process involved in this method is very complex.
6.	Prediction of Heart Disease Using Machine Learning Algorithms.2020.	Sonam Nikhar	Used combination of algorithms like Naive Bayes and Rare forest to get the best accuracy.	•Naive Bayes accuracy should be improved.
7.	Multi Disease Prediction using Data Mining Techniques.	K. Gomathi Kamaraj	Evaluate the best classifier.	•Concentrate on only finding the best classifier.

Chapter 3

3.1 System Analysis:

3.1.1 Existing System:

Remote mobile health monitoring has already been recognized as not only a potential. Each stage such as data aggregation, data maintenance, data integration, and data analysis, and pattern interpretation, application faces many challenges while dealing with healthcare big data (HBD). There are many problems in complexity of analysis and scalable of data in parallelization computing model is processed. They have not accuracy in prediction of heart disease.

3.2 LIMITATION:

- Certain approaches being applicable only for small data.
- Certain combination of classifier over fit with data set while others are under fit.
- Some approaches are not adoptable for real time collection of database implementation.

3.3 PROPOSED SYSTEM

In our project, proposed system is accuracy prediction of heart disease problem in health care application. Easier to analyze the scalable of health care big data. Less time consumption with efficiency of data in heart disease. High performance in data maintained of heart disease prediction.

3.4 ADVANTAGES:

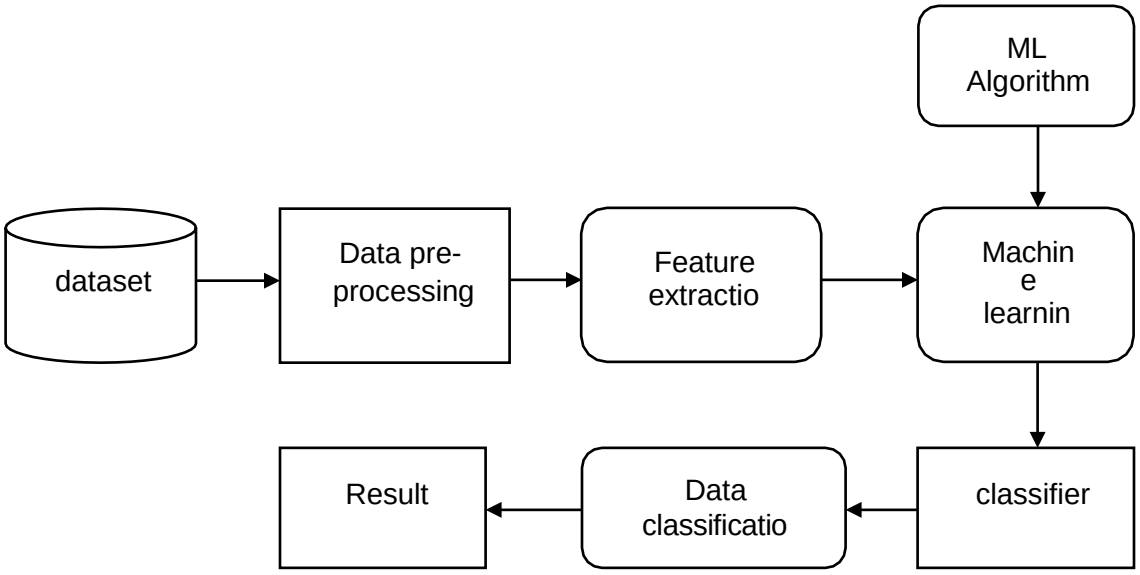
- The performance classification of heart based diseases is further improved.
- Time complexity and accuracy can have measured by various machine learning models, so that we can measures different.
- Different machine learning having high accuracy of result.
- Risky factors can be predicted early by machine learning models.

Programming Language : Python.

Chapter 4

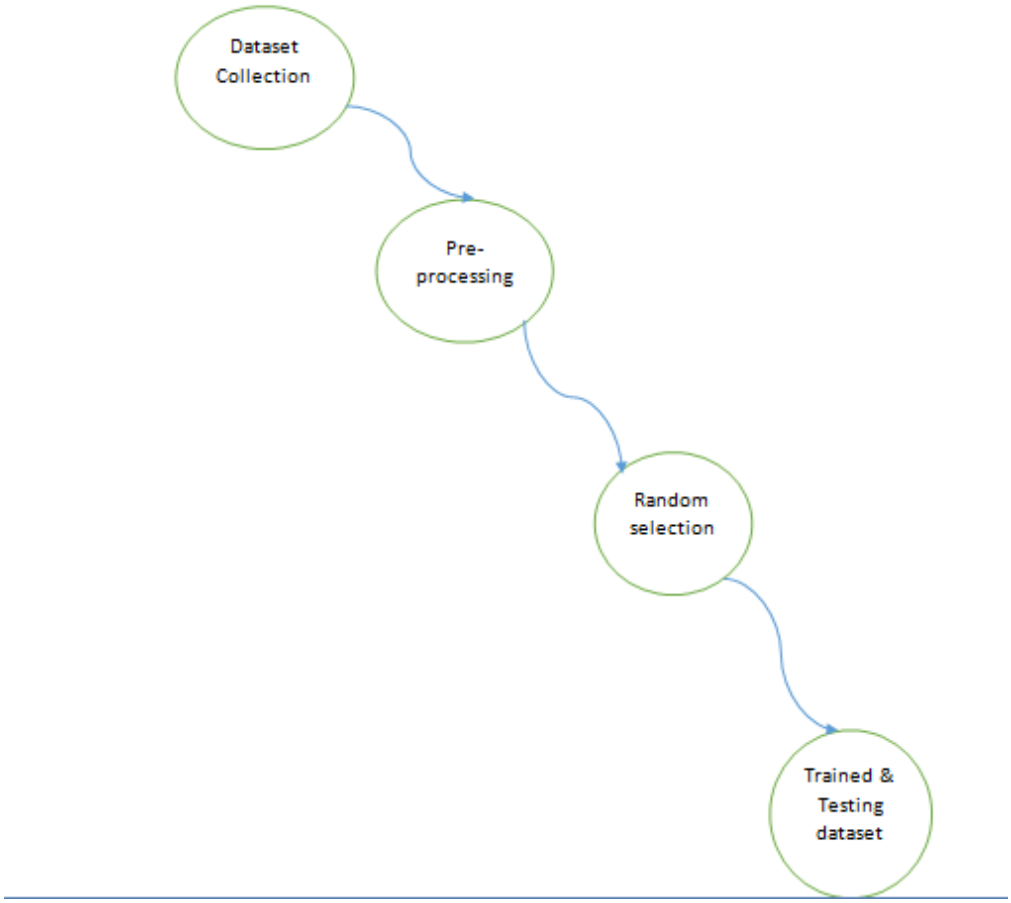
4.1 System Design:

Architecture Diagram:

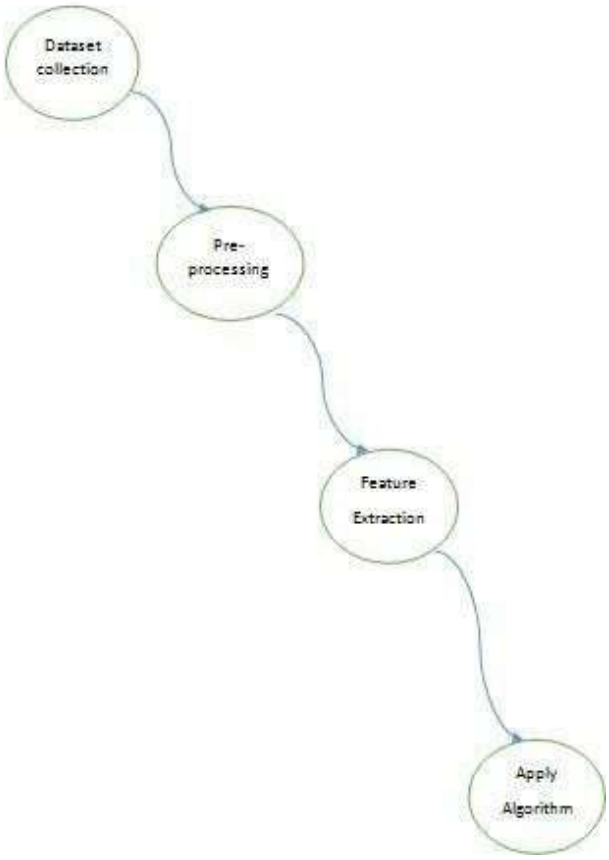


4.2 Data Flow Diagram:

LEVEL 0



LEVEL 1


Dataset collection
Pre-processing
Feature Extraction
Apply Algorithm

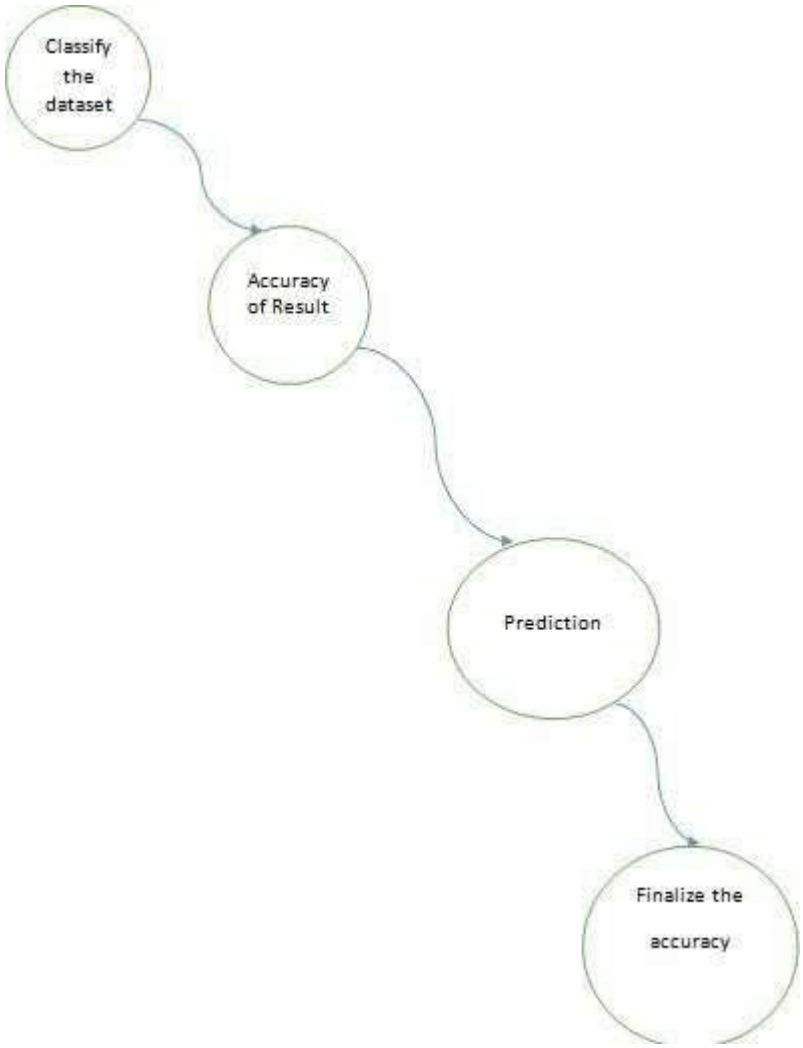


FIG 4.3 UML Diagram - Use Case Diagram:

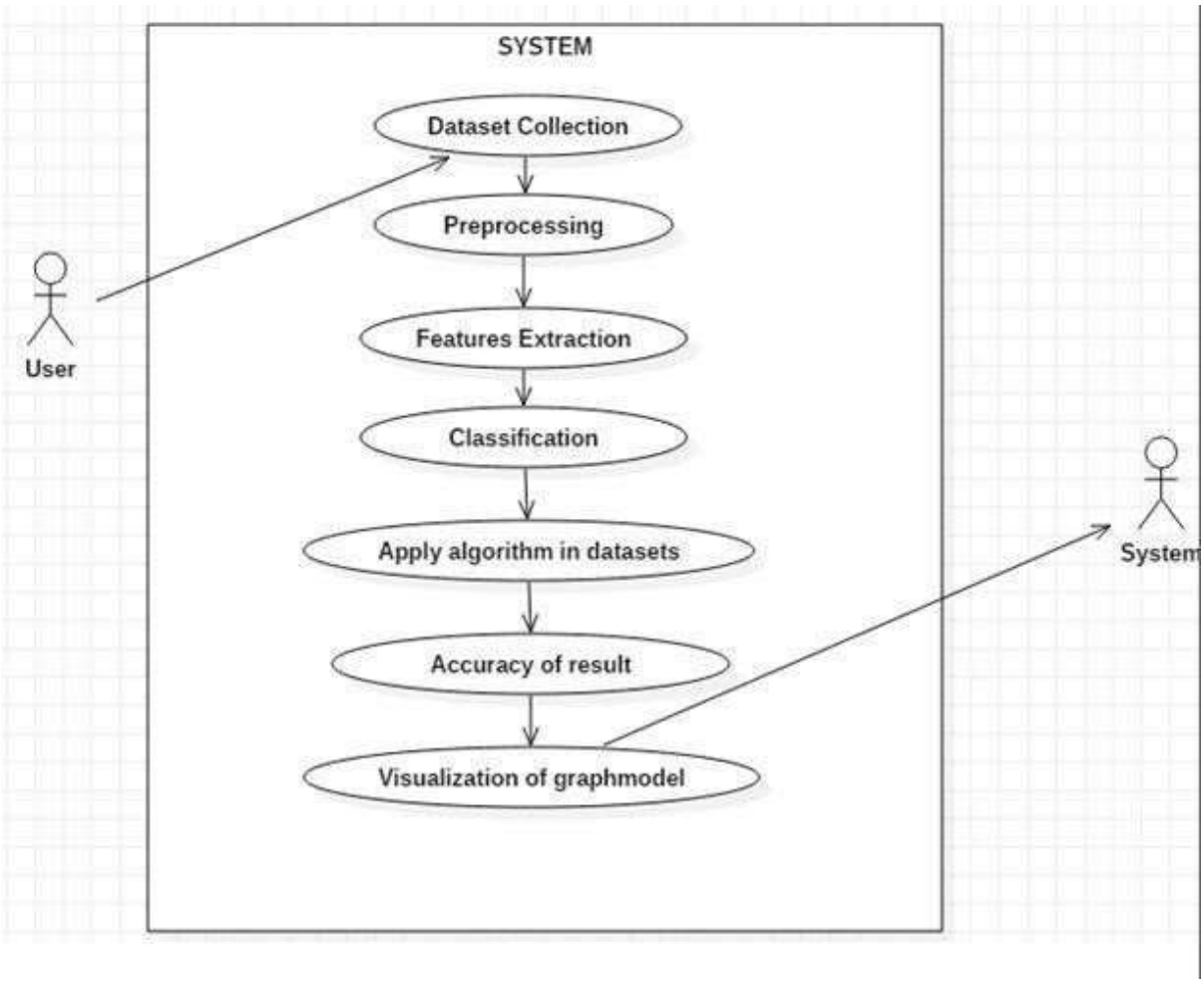

SYSTEM
Dataset Collection
Preprocessing
Features Extraction
Classification
Apply algorithm in datasets
Accuracy of result
Visualization of graphmodel
User
System

FIG 4.4 Class Diagram:

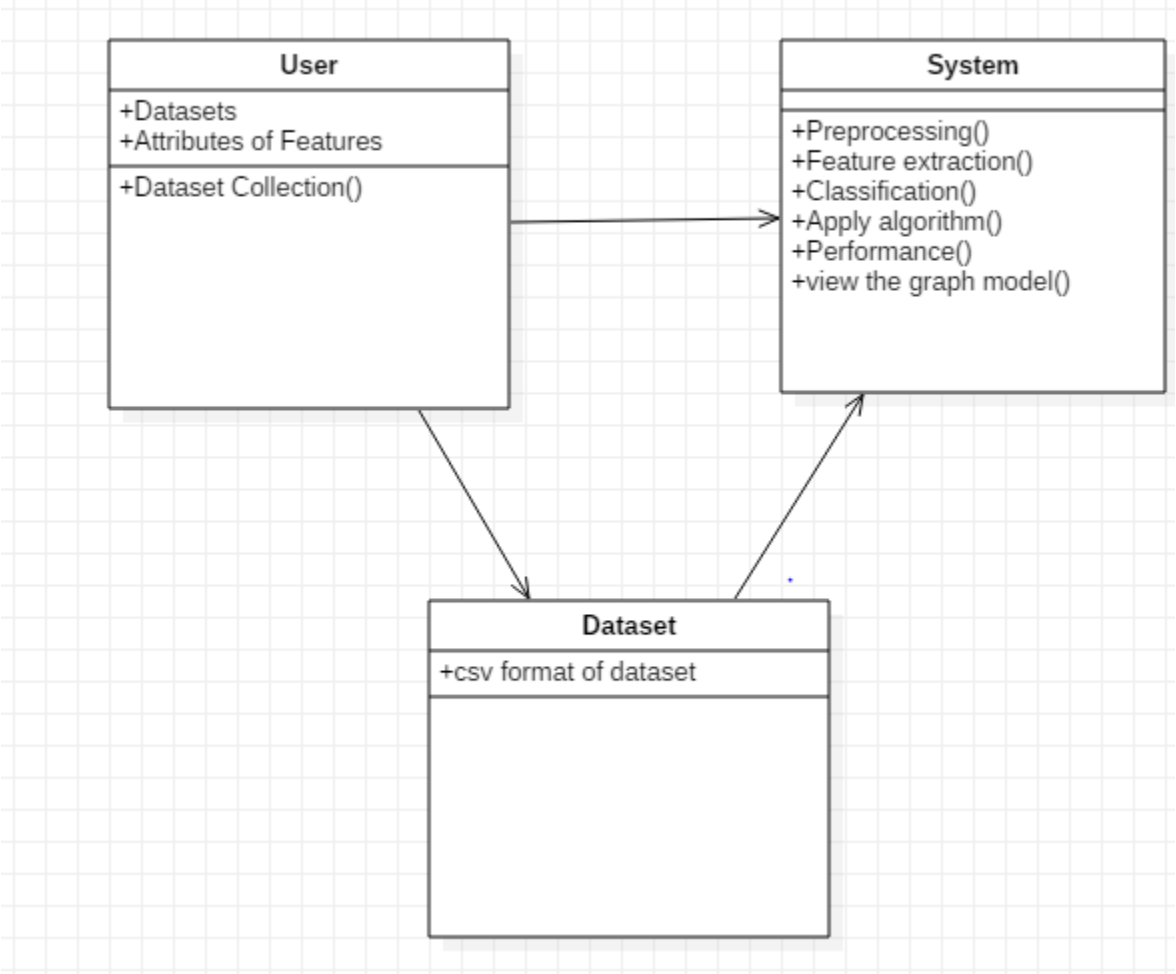


FIG 4.5 Activity Diagram:

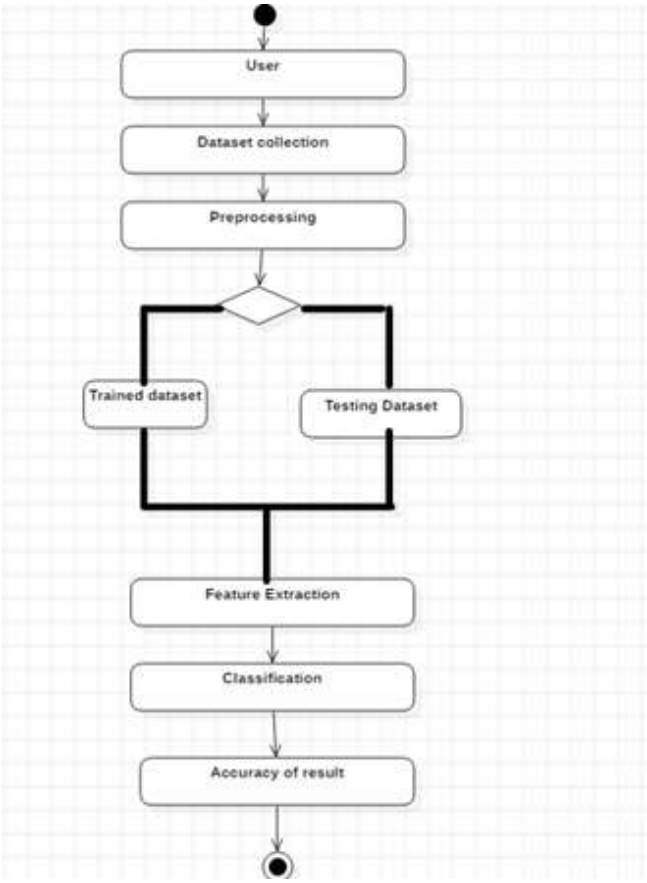
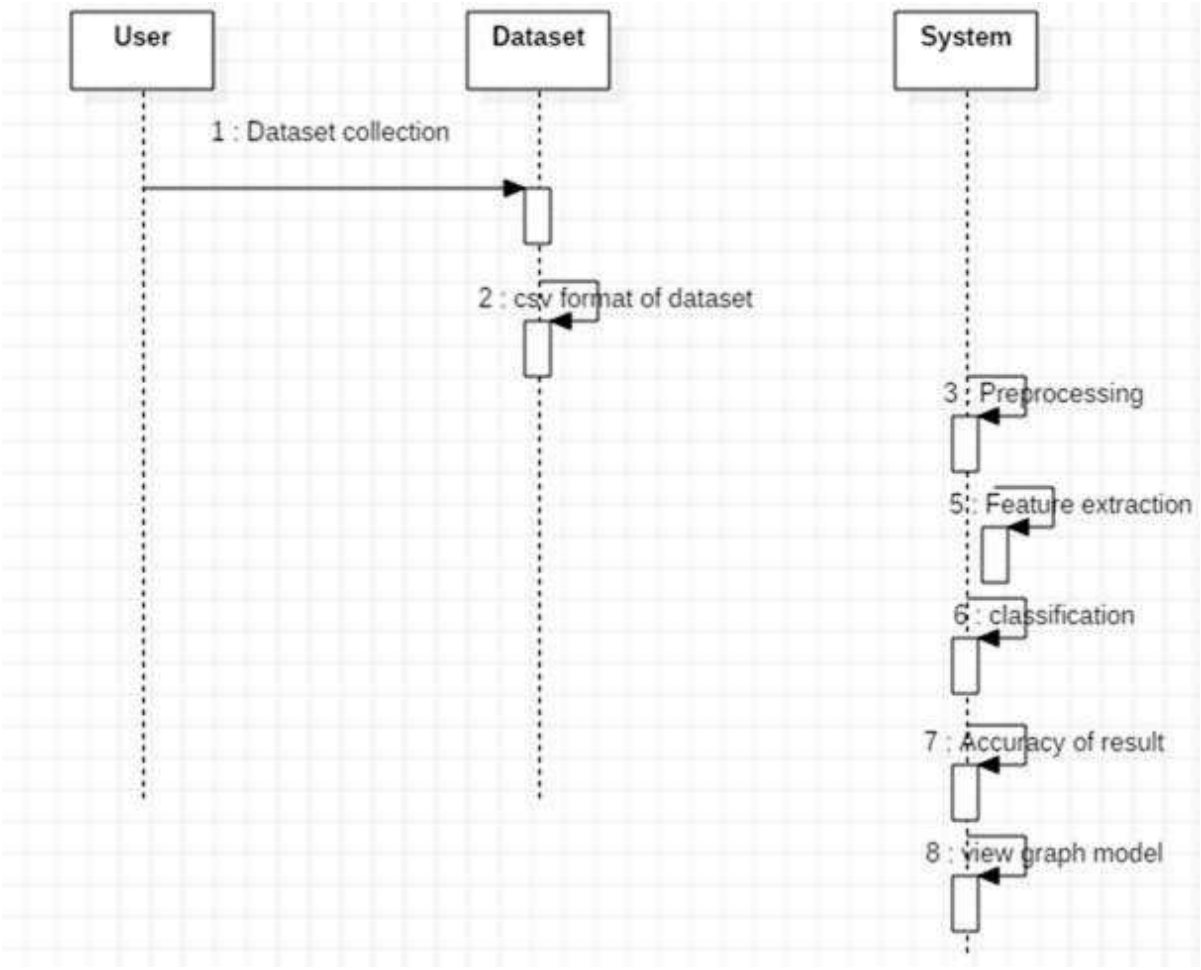

User
Dataset collection
Preprocessing
Trained dataset
Testing Dataset
Feature Extraction
Classification
Accuracy of result

Fig 4.6 Sequence Diagram:



4.7 Algorithms:

Algorithm Discussion

In machine learning we can use different algorithms otherwise known as classifiers to help us predict for our project. Here in our project we are looking forward to predict the number of patient that have heart disease and the number of patient that do not have heart disease running four algorithms our dataset. There as on we are going to use four is that it will allow us to get better and more reliable prediction. Because if we are using one algorithm or classifier and do not have anything else to compare it with then we cannot say that it a reliable prediction because it might

be giving us a very good accuracy but this algorithm might not be the best or more appropriate one to use for our scenario. Whereas if we use more than one algorithm or classifier in our case four of them, we can compare them with one another and if we find one classifier is giving us accuracy that is not even in the ball park of the other algorithm provided accuracy we can understand that something is going wrong. It can be that the algorithm itself is not suitable for the job or we made a mistake in our coding. So using more than one algorithm is essential for any prediction based system. Now the algorithms that we have chosen to use in our project are: 1. Decision tree, 2. Naïve Bayes, 3.SVM (support vector machine) and lastly 4. Random Forest. We will be discussing each ofthose algorithms below.

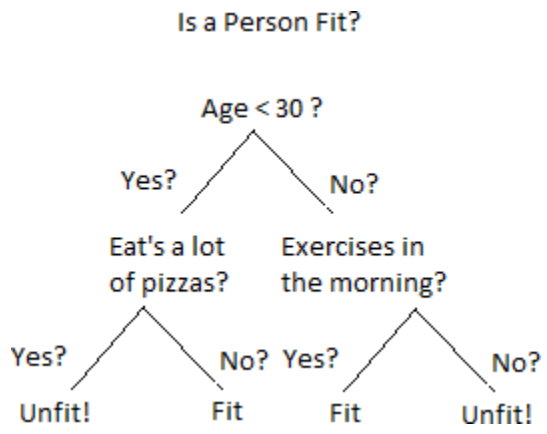
4.8 DECISION TREE(DT)

For our first algorithm we will be using Decision Tree classifier. It is one of the most popular machine learning algorithms to this date. They are used for both classification and regression problems. Now a question might arise why we are willing to use Decision tree classifier over other classifiers. To answer that question, we can bring about two reasons. One being, Decision trees often tries to mimic the same way human brain thinks so it is quite simple to understand the dataand come to some good conclusions or interpretations.

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. Decision trees learn from data to approximate a sine curve with a set of if-then-else decision rules. The deeper the tree, the more complex the decision rules and the fitter the model.

Decision tree builds classification or regression models in the form of a treestructure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result

is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.



Simple Decision Tree

4.9 How Do Decision Trees Work?

There are several steps involved in the building of a decision tree.

Splitting

The process of partitioning the data set into subsets. Splits are formed on a particular variable

Pruning

The shortening of branches of the tree. Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch

Tree Selection

The process of finding the smallest tree that fits the data. Usually this is the tree that yields the lowest cross-validated error.

4.10 NAÏVE BAYES(NB)

We have already talked out our first machine learning algorithm, the decision tree classifier. Now we are going to talk about our second machine algorithm which we are going to use for our prediction purposes which is named Naïve Bayes classifier. The working principle of naïve Bayes classifier is as follows:

•Training Step: By assuming predictors to be conditionally independent given for a class, the method estimates the parameters of a probability distribution known asthe prior probability from the training data.

•Prediction Step: For unknown test data, the method computes the posterior probability of the dataset which is belonging to each class. The method finally classifies the test data based upon the largest posterior probability from theset

4.11 Support Vector Machine(SVM)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper- plane that differentiates the two classesvery well. Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper- plane/ line).

Chapter 5

5.1 Implementation Process:

MODULES:

1. **Understanding the data**
2. **Data Pre-Processing**
3. **EXPLORATORY DATA ANALYSIS (EDA)**
4. **Splitting and Classification**

- **Understanding the data**

Data Source The dataset used here for predicting heart disease is taken from UCI Machine learning repository. UCI is a collection of databases that are used for implement machine learning algorithms. The dataset used here is real dataset. The dataset consists of 300 instance of data with the appropriate 14 clinical parameters. The clinical parameter of dataset is about tests which are taken related to the heart disease as like blood pressure level, chest pain type, electrocardiographic result and etc.

- **Data Pre-Processing:**

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

1. Formatting
2. Cleaning
3. Sampling

1. **Formatting:** The data you have selected may not be in a format that is suitable for you to work with. The data may be in a relational database and you would like it in a flat file, or the data may be in a proprietary file format and you would like it in a relational database or a text file.

2. **Cleaning:** Cleaning data is the removal or fixing of missing data. There may be data instances that are incomplete and do not carry the data you believe you need to address the problem. These instances may need to be removed. Additionally, there may be sensitive information in some of the attributes and these attributes may need to be anonym zed or removed from the data entirely.

3. **Sampling:** There may be far more selected data available than you need to work with. More data can result in much longer running times for algorithms and larger computational and memory requirements. You can take a smaller representative sample of the selected data that may be much faster for exploring and prototyping solutions before considering the whole dataset.

5.2 EXPLORATORY DATA ANALYSIS (EDA)

In this section we are going to distribute the target value is vital for choosing appropriate accuracy metrics and consequently properly assess different machine learning models. First of all, we are going to count values of explained variable otherwise known as the determining variable which is going to give us the prediction of a patient being affected by heart disease or not. Second of all we are going to separate numeric features from categorical features. Then we are going to show the relation between the categorical features in various plots and try to figure

out or rather observe the influence of those categorical features in the actual determining variable "diagnosis". It's really essential that the dataset we are working on should be approximately balanced. An extremely imbalanced dataset can render the whole model training useless and thus, will be of no use. If it is in imbalance dataset we have to do weather under sampling or over sampling to compensate the class data into balanced data.

- **Splitting and Classification:**

The whole database is split into training and testing database. The 80% data is taken for training while remaining 20% data is used for testing. Classification The training data is trained by using different machine learning algorithms Random Forest, Decision tree, Navy Bayes, SVM

5.3 CLASSIFICATION USINGDECISION TREE:

Decision Tree (DT) is a simple and easy to implement classifier. The bit through feature to access in depth patients' profiles is only obtainable in Decision Trees. Decision tree builds classification or regression models in the structure of a tree making it simple to debug and handle. Decision trees can handle both categorical and numerical data. The algorithm works by finding the information gain of the attributes and taking out the attributes for splitting the branches in threes. The information gain for the tree is identified using the below given Eq.(1).

$$E(S)= -P(P)\log2P(P)-P(N)\log2P(N)$$

The algorithm for the decision tree is given below:

Step 1:Identify the information gain for the attributes in the dataset.

Step 2:Sort the information gain for the heart disease datasets in descending order.

Step 3:After the identification of the information gain assign the best attribute of the dataset at the root of the tree.

Step 4:Then calculate the information gain using the same formula.

Step 5:Split the nodes based on the highest information gain value.

Step 6:Repeat the process until each attributesare set as leaf nodes in all the branches of the tree

5.4 CLASSIFICATION USING RANDOM FOREST

Random forests (RF) [13] are combination of tree predictors using decision tree such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. They are more robust with respect to noise. It is a supervised classification algorithm used for the prediction and it is considered as the superior due to its large number of trees in theforest giving improved accuracy than decision trees. Typically,the trees are trained independently and the predictions of the trees are combined through averaging. Random forest algorithm can use both for classification and the regression basedon the problem domain. The algorithm for random forest is given below:

Step 1:Randomly selectkfeatures from entiremfeatures, wherek<< m.

Step 2:Surrounded by thekfeatures, calculate the node“d”using the best split point.

Step 3:Split the node intodaughter nodesusing thebest split.

Step 4:Repeat1 to 3steps until l number of nodes has been reached.

Step 5:Construct forest by repeating steps1 to 4for n number times to createn number of trees.

Firstly, the k features are taken out of total m features. In the next stage, in each tree randomly select k features in order to find the root node by using the best split approach. The next stage involves calculating the daughter nodes using the same best split approach for the heart disease dataset. Similarly, the tree is formed from the root node and until all the leaf nodes are generated from the attributes. This randomly created tree forms the random forest that is used for making heart disease prediction in patients.

5.5 REAL LIFE NEED

Now a days People forgetting about their health. Heart diseases are becoming one of the most fatal diseases in several countries. Patients with Heart disease have been continuously increasing because of excessive consumption of alcohol, inhale of harmful gases, intake of contaminated food, pickles and drugs. Heart disease is a serious public health problem that affects patients. Heart disease is complex, varied and fluctuates, meaning that no one person's experience of Heart disease is the same as another. experience of Heart problems may vary from day to day. This is partly because the Heart has a huge number of functions and so Heart failure can affect almost every part of your body and the way you feel.

5.6 DATA ATTRIBUTES

The data we are using is from the records of the presence of heart disease in the patient. **Dataset** We found our data set that has been used in our book from kaggle (https://www.kaggle.com/ ronitf/heart-disease- uci/version/1). The dataset that we used in our thesis has in total 14 columns and 303 rows. First 13 of those columns are the features that we will be using later on in order to predict the final column

'diagnosis' which will tell us if the patient is going to be affected by heart disease or not. The 303 rows represent data of 303 patients that we found from the dataset.

ROW

It has a total of 303 rows which means we have with us data of over

```
In [8]: len(df)
Out[8]: 303
```

COLUMN

We have 14 columns each with different dimensions

```
In [5]: df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
age         303 non-null int64
sex         303 non-null int64
cp          303 non-null int64
trestbps    303 non-null int64
chol        303 non-null int64
fbs         303 non-null int64
restecg     303 non-null int64
thalach     303 non-null int64
exang       303 non-null int64
oldpeak     303 non-null float64
slope       303 non-null int64
ca          303 non-null int64
thal        303 non-null int64
target      303 non-null int64
dtypes: float64(1), int64(13)
memory usage: 33.2 KB
```

5.7 DIMENSION OF THE DATA

1. **age:** age in years

2. **sex:** sex (1 = male; 0 = female)
3. **cp:** chest pain type
   -- Value 1: typical angina
   -- Value 2: atypical angina
   -- Value 3: non-anginal pain
   -- Value 4: asymptomatic
4. **trestbps:** resting blood pressure (in mm Hg on admission to the hospital)
5. **chol:** serum cholestoral in mg/dl
6. **fbs:** (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)
7. **restecg:** resting electrocardiographic results
   -- Value 0: normal
   -- Value 1: having ST-T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV)
   -- Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteria
8. **thalach:** maximum heart rate achieved
9. **exang:** exercise induced angina (1 = yes; 0 = no)
10. **oldpeak** = ST depression induced by exercise relative to rest
11. **slope:** the slope of the peak exercise ST segment
   -- Value 1: upsloping
   -- Value 2: flat
   -- Value 3: downsloping
12. **ca:** number of major vessels (0-3) colored by flourosopy
13. **thal:** 3 = normal; 6 = fixed defect; 7 = reversable defect
14. **num:** diagnosis of heart disease (angiographic disease status)
   -- Value 0: < 50% diameter narrowing
   -- Value 1: > 50% diameter narrowing

5.8 DATASET READING USING PANDAS

We created an array called col names and put down all our columns on that array. Then we read the csv file also known as the dataset file.

```
In [3]: df.head()
```

Out[3]:

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1

5.9 PREPROCESSING

Beforewestartletusgiveabriefinformationaboutwhatdatapreprocessingactuallyis. Data preprocessing may be a data processing technique that involves remodeling data into a lucid format. Real-world data is often incomplete, inconsistent and lacking in certain behaviors or trends and is likely to contain many errors. Data preprocessing may be a tried technique of partitioning such problems. Data preprocessing prepares raw data for further processing. Data preprocessing is used in database-driven applications such as customer relationship management and rule-based applications. For our thesis we are using standard scaler from the sklearn library for preprocessing our data. We choose this one over the many other ones because it suits very well with our system.

5.10 MISSING VALUES

There are no missing values in the data set so we don’t need to replace the missing values

```
missing_values = df.isnull().mean()*100

missing_values.sum()
```

```
0.0
```

5.11 PYTHON OVERVIEW

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted:** Python is processed at runtime by the interpreter.You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive:** You can actually sit at a Python prompt and interactwith the interpreter directly to write your programs.

- **Python is Object-Oriented:** Python supports Object-Oriented style ortechnique of programming that encapsulates code within objects.

- **Python is a Beginner's Language:** Python is a great language for thebeginner-level programmers and supports the development of a widerange of applications from simple text processing to WWW browsers to games.

5.12 History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, Smalltalk, Unix shell, and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

5.13 Python Features

Python's features include:

- **Easy-to-learn:** Python has few keywords, simple structure, and a clearlydefined syntax. This allows the student to pick up the language quickly.

- **Easy-to-read:** Python code is more clearly defined and visible to the eyes.

- **Easy-to-maintain:** Python's source code is fairly easy-to-maintaining.

- **A broad standard library:** Python's bulk of the library is very portable andcross-platform compatible on UNIX, Windows, and Macintosh.

- **Interactive Mode:** Python has support for an interactive mode which allowsinteractive testing and debugging of snippets of code.

- **Portable:** Python can run on a wide variety of hardware platforms and has thesame interface on all platforms.

- **Extendable:** You can add low-level modules to the Python interpreter. Thesemodules enable programmers to add to or customize their tools to be more efficient.

- **Databases:** Python provides interfaces to all major commercial databases.

- **GUI Programming:** Python supports GUI applications that can be created and ported to many system calls, libraries, and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

- **Scalable:** Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below:

- IT supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

5.14 PYTHON ENVIRONMENT

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

ANACONDA NAVIGATOR

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows you to launch applications and easily manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository. It is available for Windows, mac OS and Linux.

Why use Navigator?

In order to run, many scientific packages depend on specific versions of other packages. Data scientists often use multiple versions of many packages, and use multiple environments to separate these different versions.

The command line program conda is both a package manager and an environment manager, to help data scientists ensure that each version of each package has all the dependencies it requires and works correctly.

Navigator is an easy, point-and-click way to work with packages and environments without needing to type conda commands in a terminal window. You can use it to find the packages you want, install them in an environment, run the packages and update them, all inside Navigator.

5.15 WHAT APPLICATIONS CAN I ACCESS USING NAVIGATOR?

The following applications are available by default in Navigator:

- JupyterLab
- Jupyter Notebook
- QTConsole
- Spyder
- VSCode
- Glueviz
- Orange 3 App
- Rodeo
- RStudio

Advanced conda users can also build your own Navigator applications

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code.

You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

What's new in 1.9?

- Add support for **Offline Mode** for all environment related actions.
- Add support for custom configuration of main windows links.
- Numerous bug fixes and performance enhancements.

5.16 PYTHON

Python is a general-purpose, versatile and popular programming language. It's great as a first language because it is concise and easy to read, and it is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and scientific applications.It has simple easy-to-use syntax, making it the perfect language for someone tryingto learn computer programming for the first time.

Features of Python

A simple language which is easier to learn, Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax. If you are a newbie, it's a great choice to start your journey with Python.

- **Free and open source**

  You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software's written in it, you can even make

changes to the Python's source code. Python has a large community constantly improving it in each iteration.

- **Portability**

  You can move Python programs from one platform to another, and run it without any changes.
  It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

- **Extensible and Embeddable**

  Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code. This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

- **A high-level, interpreted language**

  Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.
  Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower level operations.

- **Large standard libraries to solve common tasks** Python has a number of standard libraries which makes life of a programmermuch easier since you don't have to write all the code yourself. For example:Need to connect MySQL database on a Web Server You can use MySQL dBlibrary using import MySQL db. Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

- **Object-oriented**

  Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively. With OOP, you are able to divide these complex problems into smaller sets by creating object

Python

History and Versions:

Python is predominantly a dynamic typed programming language which was initiated by Guido van Rossum in the year 1989. The major design philosophy that was given more importance was the readability of the code and expressing an idea in fewer lines of code rather than the verbose way of expressing things as in C++ and Java. The other design philosophy that was worth mentioning was that, there should be always a single way and a single obvious way to express a given task which is contradictory to other languages such as C++, Perl etc. Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The initial versions of Python were heavily inspired from lisp (for functional programming constructs). Python had heavily borrowed the module system, exception model and also keyword arguments from Modula-3 language. Pythons' developers strive not to entertain premature optimization, even though it might increase the performance by a few basis points. During its design, the creators had conceptualized the language as being a very extensible language, and hence they had designed the language tohave a small core library which was extended by a huge standard library. Thus as aresult, python is used as a scripting language as it can be easily embedded into any

application, though it can be used to develop a full-fledged application. The reference implementation of python is Python. There are also other implementations like Python, Iron Python which can use python syntax as well as can use any class of Java (Python) or .Net class (Iron Python). Versions: Pythonhas two versions 2.x version and 3.x version. The 3.x version is a backward incompatible release was released to fix many design issues which plagued the 2.x series. The latest in the 2.x series is 2.7.6 and the latest in 3.x series is 3.4.0. 1.5.2 Paradigms: Python supports multi-paradigms such as: Object-Oriented, Imperative,Functional, Procedural, and Reflective. In Object-Oriented Paradigm, Python supports most of the OOPs concepts such as Inheritance (It also has support for Multiple Inheritance), Polymorphism but its lack of support for encapsulation is a blatant omission as Python doesn't have private, protected members: all class members are public. Earlier Python 2.6 versions didn't support some OOP's concepts such as Abstraction through Interfaces and Abstract Classes. It also supports Concurrent paradigm, but with Python we will not be able to make truly multitasking applications as the inbuilt threading API is limited by GIL (Global Interpreter Lock) and hence applications that use the threading API cannot run on multi-core parallelly. The only remedy is that, the user has to either use the multi- processing module which would fork processes or use Interpreters that haven't implemented GIL such as Python or Iron Python. Compilation, Execution and Memory Management: Compilation, Execution and Memory Management: 21 A Comparative Studies of Programming Languages (Comparative Studies of SixProgramming Language) Just like the other Managed Languages, Python compiles to an intermediary code and this in turn is interpreted by the Python Runtime Environment to the Native Machine Code. The reference implementation (i.e.CPython) doesn't come with a JIT compiler because of which the execution speed is slow compared to native programming languages. We can use PyPy interpreter

as it includes a JIT compiler rather than using the Python interpreter that comes by default with the python language, if speed of execution is one of the important factors. The Python Runtime Environment also takes care of all the allocation and deallocation of memory through the Garbage Collector. When a new object is created, the GC allocates the necessary memory, and once the object goes out of its scope, the GC doesn't release memory immediately but instead it becomes eligible for Garbage Collection, which would eventually release the memory. Typing Strategies: Python is a strongly dynamic typed language. Python 3 also supports optional static typing. There are a few advantages in using a dynamic typed language, the most prominent one would be that the code is more readable as there is less code (in other words has less boiler-plate code). But the main disadvantage in having python as a dynamic programming language is that there would be no way to guarantee that a particular piece of code would run successfully for all the different data-types scenarios simply because it had run successfully with one type. Basically, we don't have any means to find out an error in the code, till the code has started running. 1.5.4 Strengths and Weaknesses and Application Areas: Python is predominantly used as a scripting language used in developing standalone applications that are being developed with Static-Typed languages,because of the flexibility it provides due to its dynamic typed nature. Python favorsrapid application development, which qualifies it to be used for prototyping. To a certain extent, Python is also used in developing websites. Due to its dynamic typing and of the presence of a Virtual Machine, there is a considerable overhead which translates to way less performance when we compare with native programming languages. And hence it is not suited

5.17 NUMPY

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fastoperations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier transforms, basic linear algebra, basic statistical operations, random simulation and much more. At the core of the NumPy package,is the ndarray object. This encapsulates n-dimensional arrays of homogeneous data types, with many operations being performed in compiled code for performance. There are several important differences between NumPy arrays and the standard Python sequences:

• NumPy arrays have a fixed size at creation, unlike Python lists (which can grow dynamically). Changing the size of an ndarray will create a new array and delete the original.

• The elements in a NumPy array are all required to be of the same data type, and thus will be the same size in memory. The exception: one can have arrays of (Python, including NumPy) objects, thereby allowing for arrays of different sizedelements.

• NumPy arrays facilitate advanced mathematical and other types of operations on large numbers of data. Typically, such operations are executed more efficiently and with less code than is possible using Python's built-in sequences.

• A growing plethora of scientific and mathematical Python-based packages are using NumPy arrays; though these typically support Python-sequence input, they convert such input to NumPy arrays prior to processing, and they often output NumPy arrays. In other words, in order to efficiently use much (perhaps even most) of today's scientific/mathematical Python-based software, just knowing how to use Python's built-in sequence types is insufficient - one also needs to know

how to use NumPy arrays. The points about sequence size and speed are particularly important in scientific computing. As a simple example, consider the case of multiplying each element in a 1-D sequence with the corresponding element in another sequence of the same length. If the data are stored in two Python lists, a and b, we could iterate over each element:

The Numeric Python extensions (NumPy henceforth) is a set of extensions to the Python programming language which allows Python programmers to efficiently manipulate large sets of objects organized in grid-like fashion. These sets of objects are called arrays, and they can have any number of dimensions: one dimensional arrays are similar to standard Python sequences, two-dimensionalarrays are similar to matrices from linear algebra. Note that one-dimensional arraysare also different from any other Python sequence, and that two-dimensional matrices are also different from the matrices of linear algebra, in ways which we will mention later in this text. Why are these extensions needed? The core reason isa very prosaic one, and that is that manipulating a set of a million numbers in Python with the standard data structures such as lists, tuples or classes is much too slow and uses too much space. Anything which we can do in NumPy we can do in standard Python – we just may not be alive to see the program finish. A subtler reason for these extensions however is that the kinds of operations that programmers typically want to do on arrays, while sometimes very complex, can often be decomposed into a set of fairly standard operations. This decomposition has been developed similarly in many array languages. In some ways, NumPy is simply the application of this experience to the Python language – thus many of theoperations described in NumPy work the way they do because experience has shown that way to be a good one, in a variety of contexts. The languages which were used to guide the development of NumPy include the infamous APL family of languages, Basis, MATLAB, FORTRAN, S and S+, and others. This heritage

will be obvious to users of NumPy who already have experience with these other languages. This tutorial, however, does not assume any such background, and all that is expected of the reader is a reasonable working knowledge of the standard Python language. This document is the "official" documentation for NumPy. It is both a tutorial and the most authoritative source of information about NumPy with the exception of the source code. The tutorial material will walk you through a set of manipulations of simple, small, arrays of numbers, as well as image files. This choice was made because:

• Aconcrete data set makes explaining the behavior of some functions much easier to motivate than simply talking about abstract operations on abstract data sets;

• Every reader will at least an intuition as to the meaning of the data and organization of image files, and

• The result of various manipulations can be displayed simply since the data set has a natural graphical representation. All users of NumPy, whether interested in image processing or not, are encouraged to follow the tutorial with a working NumPy installation at their side,testing the examples, and, more importantly, transferring the understanding gained by working on images to their specific domain. The best way to learn is by doing – the aim of this tutorial is to guide you along this "doing."

5.18 Design of system

In this portion of our report we are going to discuss how we prepared or designed the whole system. In terms of how we executed the system it will be discussed later in the book.

5.19 Dataset

We found our data set that has been used in our book from kaggle (https://www.kaggle.com/ ronitf/heart-disease- uci/version/1). The dataset that we used in our thesis has in total 14 columns and 303 rows. First 13 of those columns are the features that we will be using later on in order to predict the final column 'diagnosis' which will tell us if the patient is going to be affected by heart disease or not. The 303 rows represent data of 303 patients that we found from thedataset.

5.20 Preprocessing

Before we start let us give a brief information about what data preprocessing actually is. Data preprocessing may be a data processing technique that involves remodeling data into a lucid format. Real-world data is often incomplete, inconsistent and lacking in certain behaviors or trends and is likely to contain manyerrors. Data preprocessing may be a tried technique of partitioning such problems. Data preprocessing prepares raw data for further processing. Data preprocessing is used in database-driven applications such as customer relationship managementand rule-based applications. For our thesis we are using standard scaler from the sklearn library for preprocessing our data. We choose this one over the many other ones because it suits very well with our system.

5.21 Load data

We created an array called col names and put down all our columns on that array. Then we read the csv file also known as thedataset file.

5.22 Analyze features

In this section we are going to distribute the target value is vital for choosing appropriate accuracy metrics and consequently properly assess different machine learning models. First of all, we are going to count values of explained variable otherwise known as the determining variable which is going to give us the prediction of a patient being affected by heart disease or not. Second of all we are going to separate numeric features from categorical features. Then we are going to show the relation between the categorical features in various plots and try to figure out or rather observe the influence of those categorical features in the actualdetermining variable "diagnosis".

5.23 Modeling and predicting with machine learning

The main goal of the entire project is to predict heart disease occurrence with the highest accuracy. In order to achieve this, we will test several classification algorithms. This section includes all results obtained from the study and introduces the best performer according to accuracy metric. I have chosen several algorithms typical for solving supervised learning problems throughout classification methods. First of all, let's equip ourselves with a handy tool that benefits from the cohesion of SciKit Learn library and formulate a general function for training our models. The reason for displaying accuracy on both, train and test sets, is to allow us to evaluate whether the model over fits or under fits the data (so-called bias/variance tradeoff). Then we are going to split the data then test and train them in the ratio of 70:30. Then we are going to create a model where we are going to run all our algorithms.

5.24 Finding the result

At the end we are going to create a summery table where we are going to show the different accuracy percentage of different algorithms. Where we are going to find out that it does not come as a surprise that the more complex algorithms like SVM and Random Forests generated better results compared to the basic ones. It isworth to emphasize that in most cases hyper parameter tuning is essential toachieve robust results out of these techniques. By producing decent results, simplermethods proved to be useful as well. Machine learning has absolutely bright future in medical field. Just imagine a place where heart disease experts are not available. With just basic information about a certain patient's medical history, we may quite accurately predict whether a disease will occur or not. We are going to discuss them more in details in the later section.

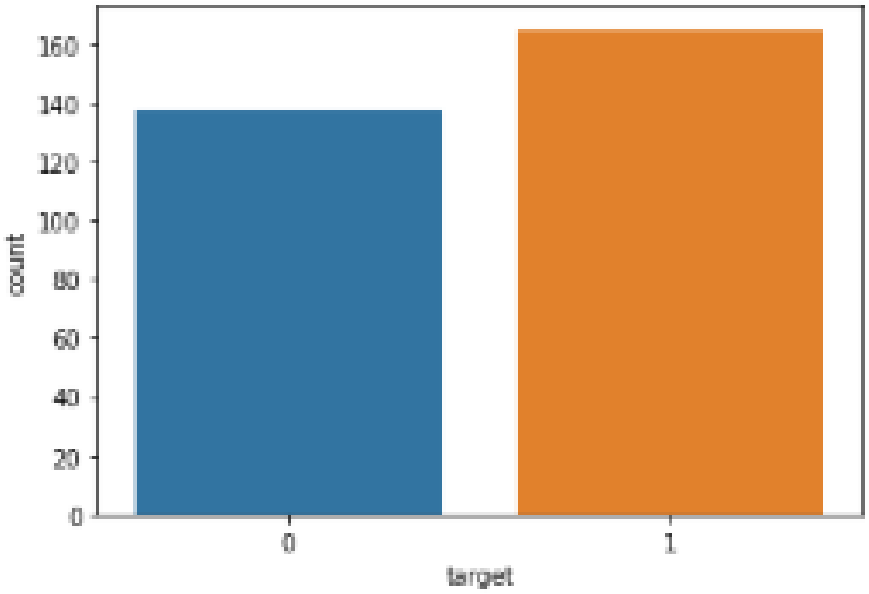
Chapter 6

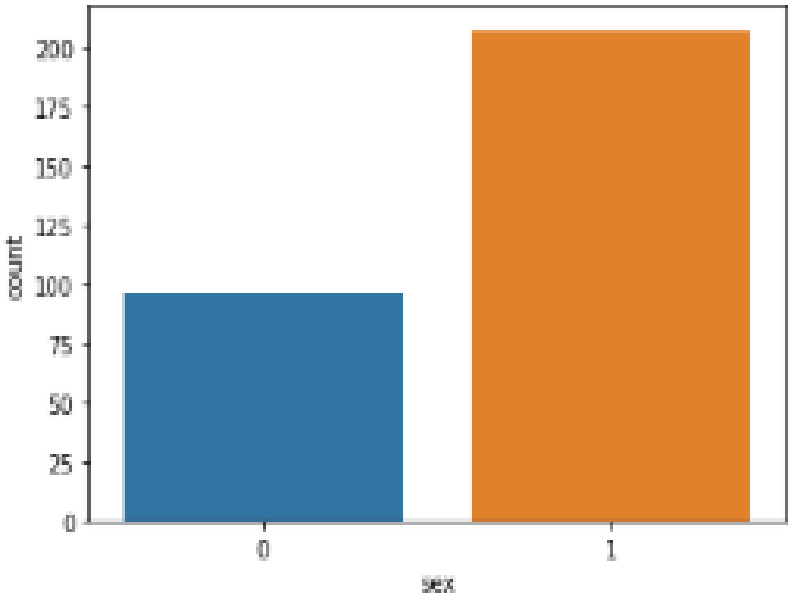
6.1 Results and Analysis:

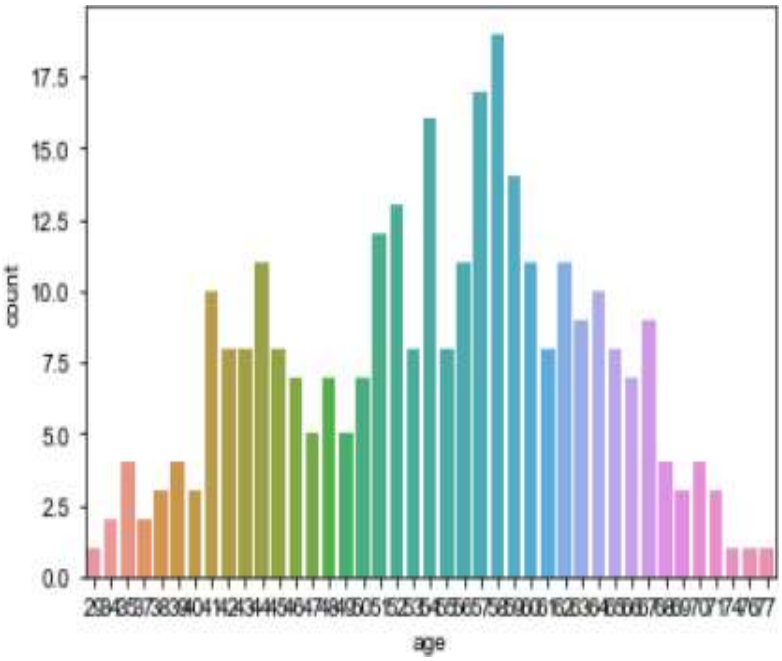
In our previous chapters we have discussed about different algorithms, previous works in this field and the dataset we used for our experiments. All those were the foundation for this chapter. In this chapter we discussed about results that we found after implementing the algorithms and analyzed them.

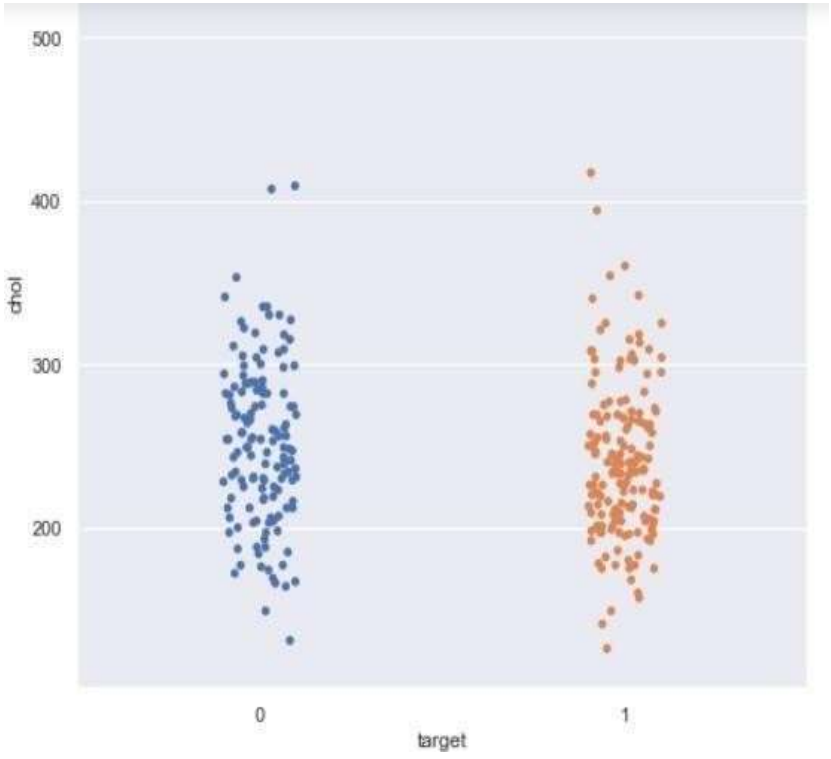
6.2 Exploratory Data Analysis

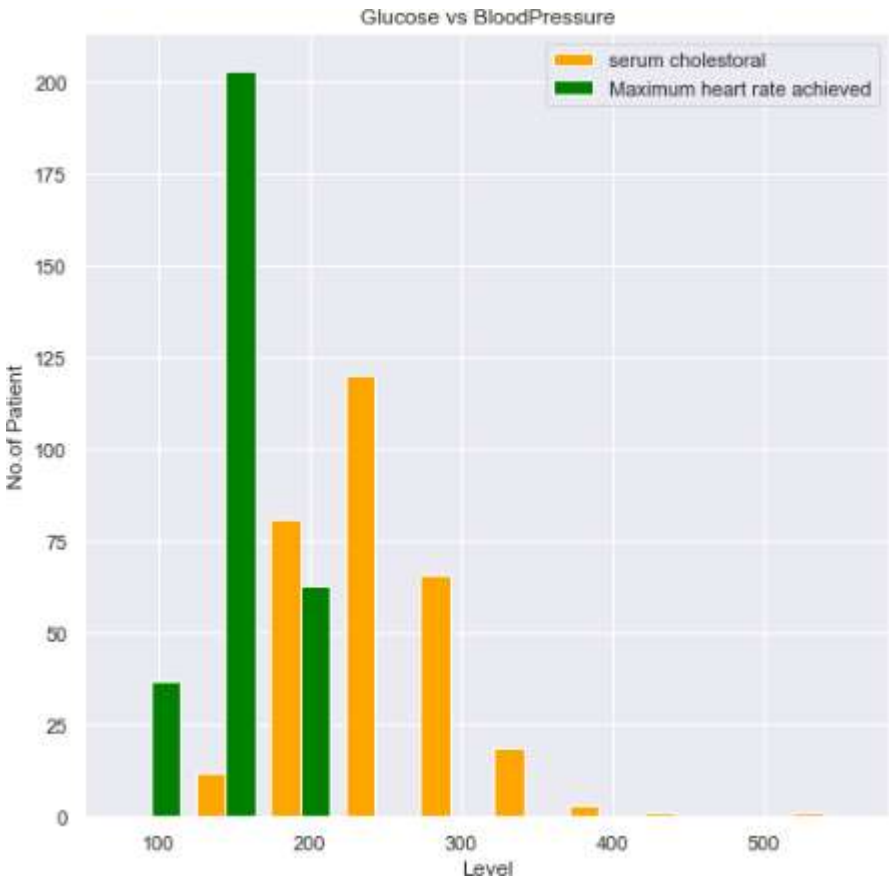
(EDA)Attribute wise graph plot

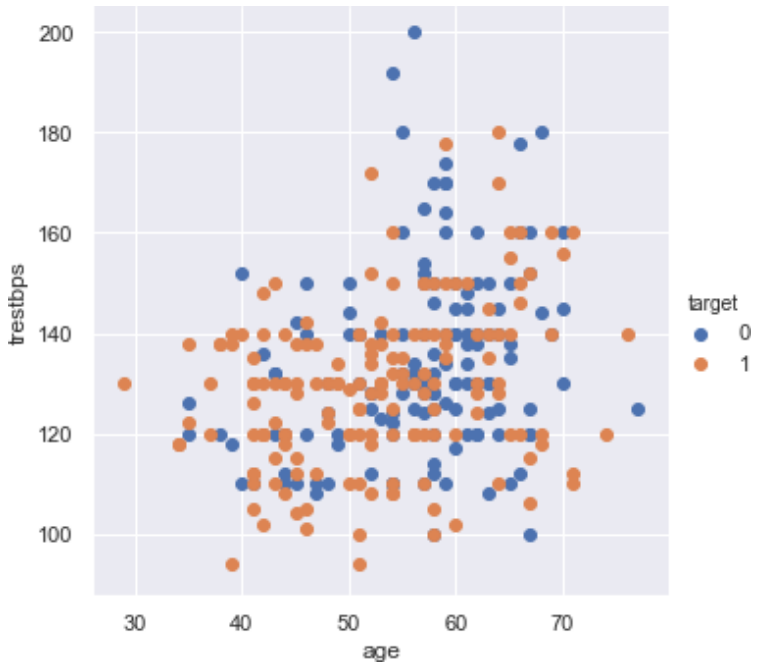

160
140
120
100
80
60
40
20
0
count
0
1
target


200
175
150
125
100
75
50
25
0
count
0
1
sex


count
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
age


chol
500
400
300
200
0
1
target


Glucose vs BloodPressure
serum cholestoral
Maximum heart rate achieved
No.of Patient
Level
0
25
50
75
100
125
150
175
200
100
200
300
400
500


trestbps
age
target
0
1
100
120
140
160
180
200
30
40
50
60
70

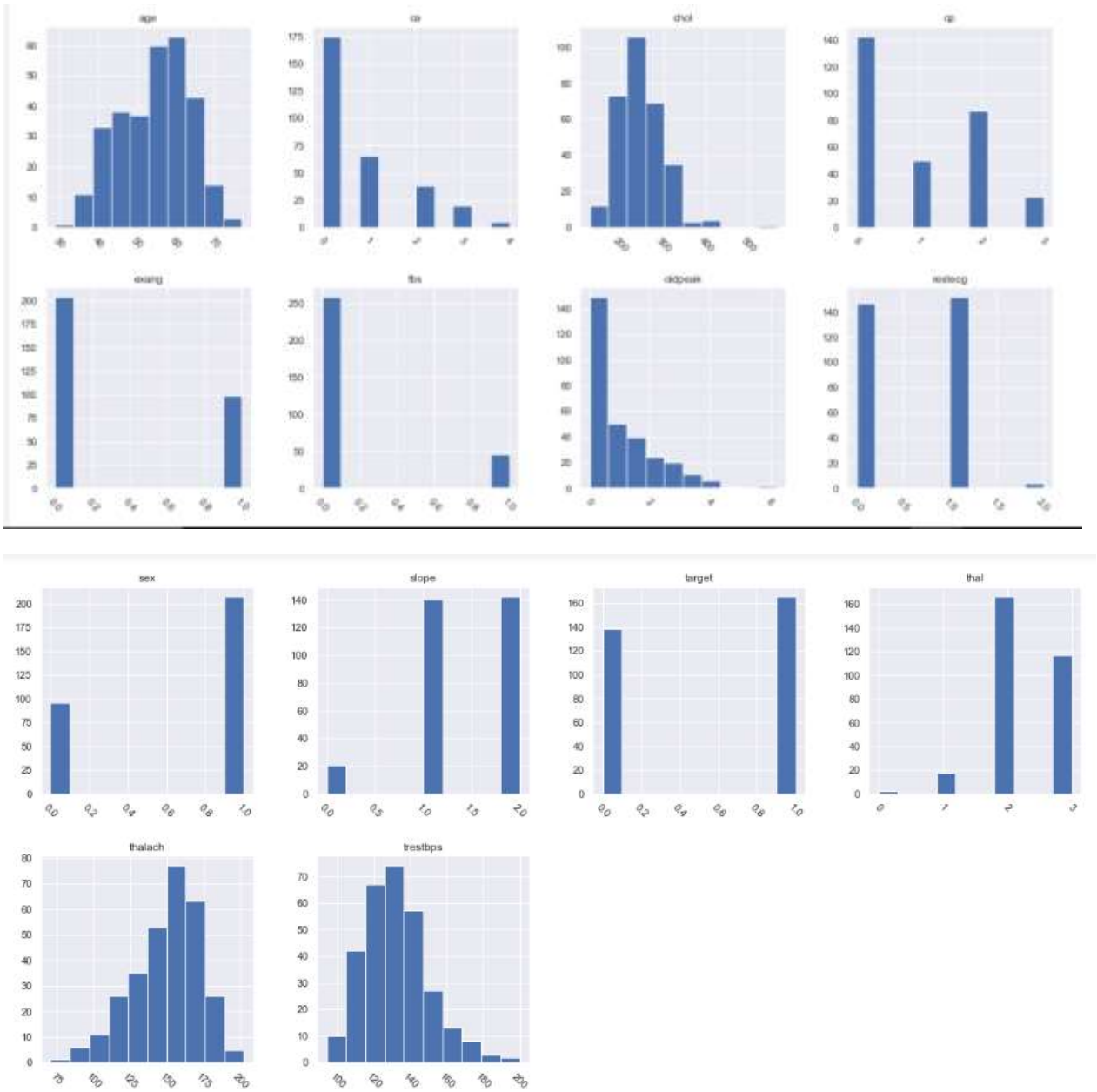


Fig 6.3 Density Plot with Oldpeak Attribute

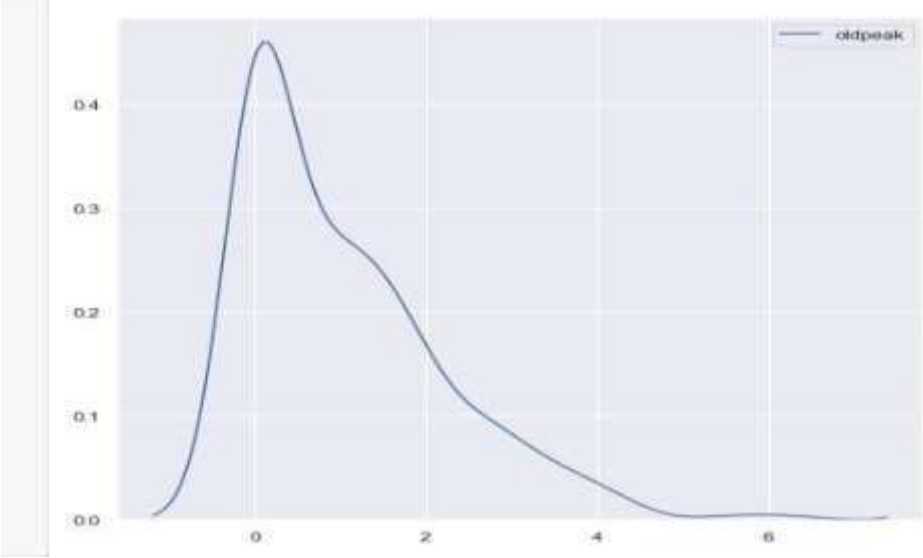


Fig 6.4 Correlation Matrix between attributes :

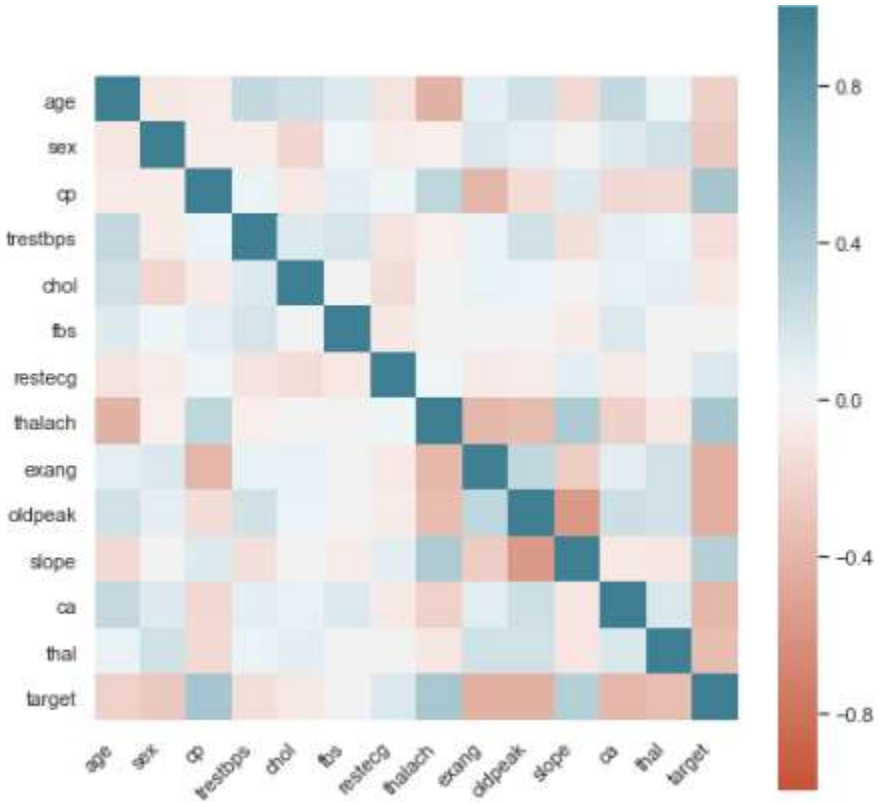


Fig 6.5 Confusion Matrix with Naïve Bayes

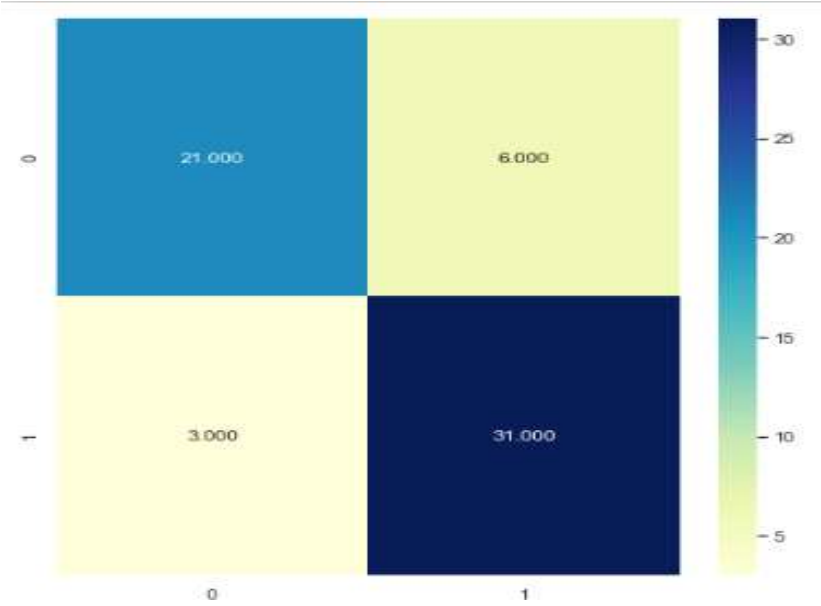


Fig 6.6 Confusion Matrix with Random Forest Classifier

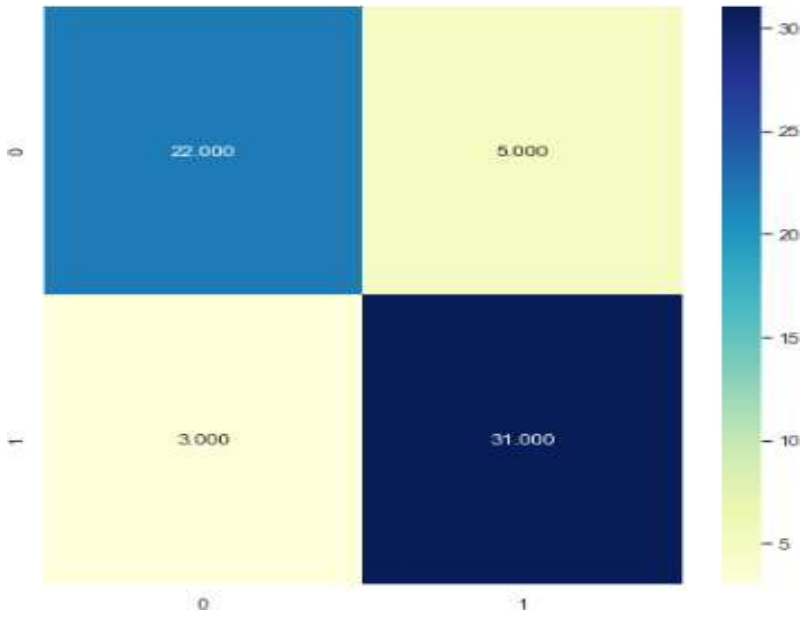


Fig 6.7 Confusion Matrix with Decision Tree Classifier

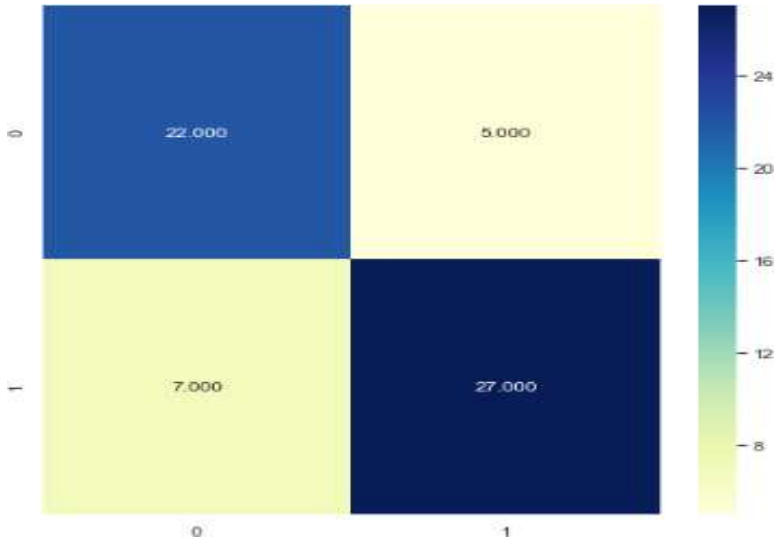


Fig 6.8 Confusion Matrix with SVM

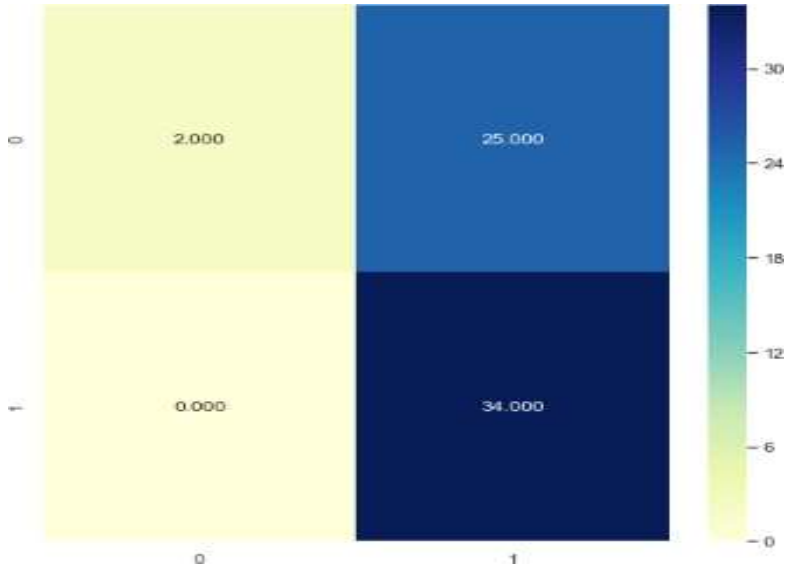
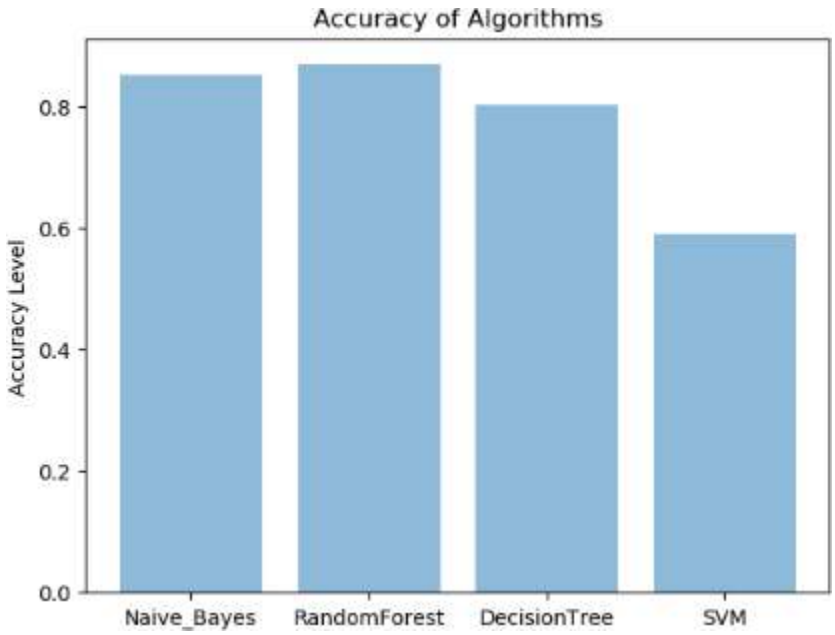


Fig 6.9 Comparative Result with Different Algorithm :



6.10 Accuracy of Models with All Features

Models / No. of runs	RUN1	RUN2	RUN3	RUN4	RUN5
Naive Bayes	0.85	0.85	0.85	0.85	0.85
Random Forest Classifier	0.83	0.81	0.83	0.81	0.88
Decision Tree	0.78	0.78	0.75	0.77	0.77
Support Vector	0.59	0.59	0.59	0.59	0.59

6.11 Conclusion:

In this research we have tried to compare different machine learning algorithmsand predict if a certain person, given various personal characteristics and symptoms, will get heart disease or not. The main motive of our report was to comparing the accuracy and analyzing the reasons behind the variation of different algorithms. We have used Cleveland dataset for heart diseases which contains 303 instances and used 10-fold Cross Validation to divide the data into two sections which are training and testing datasets. We have considered 13 attributes and implemented four different algorithms to analyze the accuracy. By the end of the implementation part, we have found Gaussian Naïve Bayes and Random Fore staregiving the maximum accuracy level in our dataset which is 91.21 percent and Decision Tree is performing the lowest level of accuracy which is 84.62 percent. Probably for other instances and other datasets other algorithm may work in better way but in our case we have found this result. Moreover, if we increase the attributes, maybe we can found more accurate result butitwilltakemoretimetoprocessandthesystemwillbeslowerthannowasitwillbelittle

more complex and will be handling more data's. So considering these possible things we took a decision which is better for us to work with.

6.12 Future Scope:

The dataset that is used in our thesis is very small and old. Moreover, no new dataset regarding heart disease has been introduced so far. There is a need of new dataset and we can collect that from various hospitals of India. We can alsoevaluate the efficiency of each individual classifier and also such classifiers in combination, by employing the bagging, boosting and stacking techniques.

6.13 References:

Machine learning based decision support systems (DSS) for heart disease diagnosis: a review. Online: 25 March 2017 DOI: 10.1007/s10462-01

[1]]Prerana T H M, Shivaprakash N C et al "Prediction of Heart Disease Using Machine Learning Algorithms- Naïve Bayes,Introduction to PAC Algorithm, Comparison of Algorithms and HDPS",Vol 3, PP: 90-99 ©IJSE, 2015

[2] Salam Ismaeel, Ali Miri et al "Using the Extreme Learning Machine (ELM) technique for heart disease diagnosis",IEEE Canada International Humanitarian TechnologyConference,DOI:10.1109/IHTC.2015.7238043, 03 September 2015

[4]F BrainBoudi,'Risk Factors for Coronary Artery Disease',2016.[Online]Available:https://emedicine.medscape.com/articl e/164163-overview.

[5] National Health Council,'Heart Health Screenings',2017.[Online]Available:http://www.heart.org/HEARTORG/ Conditions/Heart-HealthScreenings_UCM_428687_Article.jsp#.WnsOAeeYPIV

[6] ScikitLearn,'MLPCLassifier',Available:http://scikitlearn.org/stable/modules/generated/sklearn.neural_network.MLPClassifi er.html

[7] Prediction System for heart disease using Naïve Bayes *Shadab Adam Pattekari and Asma Parveen Department of Computer Science and Engineering Khaja Banda Nawaz College of Engineering

[8] Comak E, Arslan A (2012) A biomedical decision support system using LS-SVM classifier with an efficient and new parameter regularization procedure for diagnosis of heart valve diseases. J Med Syst 36:549–556

[9] Ahmed Fawzi Otoom , Emad E. Abdallah , Yousef Kilani , Ahmed Kefaye and Mohammad Ashour(2015)Effective Diagnosis and Monitoring of Heart Disease ISSN: 1738-9984 IJSEI

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
[DEEMED TO BE UNIVERSITY]
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

National Conference on Computational Intelligence and Communication Networks

Springer CCIS

N C C I C N

24TH – 25TH MARCH 2022

NCCICN

Certificate of Presentation

This is to certify that ***Dr./Mr./Ms. CHELLUBOINA HANEESH,*** *of* ***Sathyabama Institute of Science and Technology,*** *has presented a paper entitled "****Real-Time Machine Learning Detection of Heart Disease****", in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala
Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan
Convener
Professor & Head, CSE

Dr. S. Vigneshwari
Convener
Professor & Head, CSE

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
[DEEMED TO BE UNIVERSITY]
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

National Conference on Computational Intelligence and Communication Networks

Springer CCIS

N C C I C N

24TH – 25TH MARCH 2022

NCCICN

Certificate of Presentation

This is to certify that ***Dr./Mr./Ms. B.V.S SATHYANARAYANA,*** *of* ***Sathyabama Institute of Science and Technology,*** *has presented a paper entitled* ***"Real-Time Machine Learning Detection of Heart Disease",*** *in the National Conference on Computational Intelligence and Communication Networks (NCCICN 2022).*

Dr. T. Sasikala
Conference Chair, Professor & Dean
School of Computing

Dr. L. Lakshmanan
Convener
Professor & Head, CSE

Dr. S. Vigneshwari
Convener
Professor & Head, CSE