TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

TAPTAP
A BLACKBUCKS PRODUCT

ABSTRACT

In the medical field, the diagnosis of heart disease is the most difficult task. The diagnosis of heart disease is difficult as a decision relied on grouping of large clinical and pathological data. Due to this complication, the interest increased in a significant amount between the researchers and clinical professionals about the efficient and accurate heart disease prediction. In case of heart disease, the correct diagnosis in early stage is important as time is the very important factor. Heart disease is the principal source of deaths widespread, and the prediction of heart disease is significant at an untimely phase. Machine learning in recent years has been the evolving, reliable and supporting tools in medical domain and has provided the greatest support for predicting disease with correct case of training and testing. The main idea behind this work is to study diverse prediction models for the heart disease and selecting important heart disease feature using Random Forests algorithm. Random Forests is the Supervised Machine Learning algorithm which has the high accuracy compared to other Supervised Machine Learning algorithms such as logistic regression etc. By using Random Forests algorithm, we are going to predict if a person has heart disease or not.

CHAPTER - 1

INTRODUCTION

The heart is a kind of muscular organ which pumps blood into the body and is the central part of the body‘s cardiovascular system which also contains lungs. Cardiovascular system also comprises a network of blood vessels, for example, veins, arteries, and capillaries. These blood vessels deliver blood all over the body. Abnormalities in normal blood flow from the heart cause several types of heart diseases which are commonly known as cardiovascular diseases (CVD). Heart diseases are the main reasons for death worldwide. According to the survey of the World Health Organization (WHO), 17.5 million total global deaths occur because of heart attacks and strokes. More than 75% of deaths from cardio-vascular diseases occur mostly in middle-income and low-income countries. Also, 80% of the deaths that occur due to CVDs are because of stroke and heart attack. Therefore, prediction of cardiac abnormalities at the early stage and tools for the prediction of heart diseases can save a lot of life and help doctors to design an effective treatment plan which ultimately reduces the mortality rate due to cardiovascular diseases.

Due to the development of advance healthcare systems, lots of patient data are nowadays available (i.e., Big Data in Electronic Health Record System) which can be used for designing predictive models for cardiovascular diseases. Data mining or machine learning is a discovery method for analyzing big data from an assorted perspective and encapsulating it into useful information. ―Data Mining is a non-trivial extraction of implicit previously unknown and potentially useful information about data‖. Nowadays, a huge amount of data pertaining to disease diagnosis, patients etc. are generated by healthcare industries. Data mining provides a number of techniques which discover hidden patterns or similarities from data.

Therefore, in this paper, a machine learning algorithm is proposed for the implementation of a heart disease prediction system which was validated on two open access heart disease prediction datasets.

Data mining is the computer-based process of extracting useful information from enormous sets of databases. Data mining is most helpful in an explorative analysis because of nontrivial information from large volumes of evidence.

Medical data mining has great potential for exploring the cryptic patterns in the data sets of the clinical domain.

These patterns can be utilized for healthcare diagnosis. However, the available raw medical data are widely distributed, voluminous and heterogeneous in nature. This data needs to be collected in an organized form. This collected data can be then integrated to form a medical information system. Data mining provides a user-oriented approach to novel and hidden patterns in the Data The data mining tools are useful for answering business questions and techniques for predicting the various diseases in the healthcare field. Disease prediction plays a significant role in data mining. This paper analyzes the heart disease predictions using classification algorithms. These invisible patterns can be utilized for health diagnosis in healthcare data.

Data mining technology affords an efficient approach to the latest and indefinite patterns in the data. The information which is identified can be used by the healthcare administrators to get better services. Heart disease was the most crucial reason for victims in the countries like India, United States. In this project we are predicting the heart disease using classification algorithms. Machine learning techniques like Classification algorithms such as Random Forest, Logistic Regression are used to explore different kinds of heart-based problems.

CHAPTER 2

LITERATURE SURVEY

Machine Learning techniques are used to analyze and predict the medical data information resources. Diagnosis of heart disease is a significant and tedious task in medicine. The term heart disease encompasses the various diseases that affect the heart. The exposure of heart disease from various factors or symptom is an issue which is not complimentary from false presumptions often accompanied by unpredictable effects. The data classification is based on Supervised Machine Learning algorithm which results in better accuracy. Here we are using the Random Forest as the training algorithm to train the heart disease dataset and to predict the heart disease. The results showed that the medicinal prescription and designed prediction system is capable of prophesying the heart attack successfully. Machine Learning techniques are used to indicate the early mortality by analyzing the heart disease patients and their clinical records (Richards, G. et al., 2001). (Sung, S.F. et al., 2015) have brought about the two Machine Learning techniques, k-nearest neighbor model and existing multi linear regression to predict the stroke severity index (SSI) of the patients. Their study show that k-nearest neighbor performed better than Multi Linear Regression model. (Arslan, A. K. et al.,2016) have suggested various Machine Learning techniques such as support vector machine (SVM), penalized logistic regression (PLR) to predict the heart stroke. Their results show that SVM produced the best performance in prediction when compared to other models. Boshra Brahmi et al, [20] developed different Machine Learning techniques to evaluate the prediction and diagnosis of heart disease. The main objective is to evaluate the different classification techniques such as J48, Decision Tree, KNN and Naïve Bayes. After this, evaluating some performance in measures of accuracy, precision, sensitivity, specificity is evaluated.

Data source:

Clinical databases have collected a significant amount of information about patients and their medical conditions. Records set with medical attributes were obtained from the Cleveland Heart Disease database. With the help of the dataset, the patterns significant to the heart attack diagnosis are extracted.

The records were split equally into two datasets: training dataset and testing dataset.
A total of 303 records with 76 medical attributes were obtained. All the attributes are numeric-valued. We are working on a reduced set of attributes, i.e., only 14 attributes.
All these restrictions were announced to shrink the digit of designs, these are as follows:

1. The features should seem on a single side of the rule.
2. The rule should distinct various features into the different groups.
3. The count of features available from the rule is organized by medica history people having heart disease only.

The following table shows the list of attributes on which we are working.

Table 2.1: List of Attributes

S no	Attribute Name	Description
1	Age	age in years
2	Sex	(1 = male; 0 = female)
3	Cp	Chest Pain
4	Trest bps	resting blood pressure (in mm Hg on admission to the hospital)
5	Chol	serum cholesterol in mg/d

6	Fbs	(Fasting blood sugar >120 mg/dl) (1 = true; 0 = false)
7	Restecg	Resting electrocardiographic results
8	Thalach	Maximum heart rate achieved
9	Exang	Exercise induced angina (1=yes;0=no)
10	Old peak	ST depression induced by exercise relative to rest
11	Slope	The slope of the peak exercise ST segment
12	Ca	Number of major vessels (0-3) colored by fluoroscopy
13	Thal	3 = normal; 6 = Fixed defect; 7 = reversible fluoroscopy
14	Target	1 or 0

TAPTAP
A BLACKBUCKS PRODUCT

CHAPTER 3
AIM AND SCOPE OF PRESENT INVESTIGATION

3.1 EXISTING SYSTEM:

Clinical decisions are often made based on doctors‘ intuition and experience rather than on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. There are many ways that a medical misdiagnosis can present itself. Whether a doctor is at fault, or hospital staff, a misdiagnosis of a serious illness can have very extreme and harmful effects. The National Patient Safety Foundation cites that 42% of medical patients feel they have had experienced a medical error or missed diagnosis. Patient safety is sometimes negligently given the back seat for other concerns, such as the cost of medical tests, drugs, and operations. Medical Misdiagnoses are a serious risk to our healthcare profession. If they continue, then people will fear going to the hospital for treatment. We can put an end to medical misdiagnosis by informing the public and filing claims and suits against the medical practitioners at fault.

Disadvantages:

Prediction is not possible at early stages.

In the Existing system, practical use of collected data is time consuming.

Any faults occurred by the doctor or hospital staff in predicting would lead to fatal incidents.

Highly expensive and laborious process needs to be performed before treating the patient to find out if he/she has any chances to get heart disease in future.

3.2 PROPOSED SYSTEM:

This section depicts the overview of the proposed system and illustrates all of the components, techniques and tools are used for developing the entire system. To develop an intelligent and user-friendly heart disease prediction system, an efficient software tool is needed in order to train huge datasets and compare multiple machine learning algorithms. After choosing the robust algorithm with best accuracy and performance measures, it will be implemented on the development of the smartphone-based application for detecting and predicting heart disease risk level. Hardware components like Arduino/Raspberry Pi, different biomedical sensors, display monitor, buzzer etc. are needed to build the continuous patient monitoring system.

3.3 FEASIBILITY STUDY:

A Feasibility Study is a preliminary study undertaken before the real work of a project starts to ascertain the likely hood of the project's success. It is an analysis of possible alternative solutions to a problem and a recommendation on the best alternative.

3.3.1 Economic Feasibility:

It is defined as the process of assessing the benefits and costs associated with the development of project. A proposed system, which is both operationally and technically feasible, must be a good investment for the organization. With the proposed system the users are greatly benefited as the users can be able to detect the fake news from the real news and are aware of most real and most fake news published in the recent years. This proposed system does not need any additional software and high system configuration. Hence the proposed system is economically feasible.

3.3.2 Technical Feasibility:

The technical feasibility infers whether the proposed system can be developed considering the technical issues like availability of the necessary technology, technical capacity, adequate response and extensibility. The project is decided to build using Python. Jupyter Notebook is designed for use in distributed environment

of the internet and for the professional programmer it is easy to learn and use effectively. As the developing organization has all the resources available to build the system therefore the proposed system is technically feasible.

3.3.3 Operational Feasibility:

Operational feasibility is defined as the process of assessing the degree to which a proposed system solves business problems or takes advantage of business opportunities. The system is self-explanatory and doesn‘t need any extra sophisticated training. The system has built-in methods and classes which are required to produce the result. The application can be handled very easily with a novice user. The overall time that a user needs to get trained is 14 less than one hour. As the software that is used for developing this application is very economical and is readily available in the market. Therefore, the proposed system is operationally feasible.

3.4 EFFORT, DURATION AND COST ESTIMATION USING COCOMO MODEL:

The COCOMO (Constructive Cost Model) model is the most complete and thoroughly documented model used in effort estimation. The model provides detailed formulas for determining the development time schedule, overall development effort, and effort breakdown by phase and activity as well as maintenance effort. COCOMO estimates the effort in person months of direct labor. The primary effort factor is the number of source lines of code (SLOC) expressed in thousands of delivered source instructions (KDSI). The model is developed in three versions of different level of detail basic, intermediate, and detailed. The overall modeling process takes into account three classes of systems.

3.4.1. Embedded:

This class of system is characterized by tight constraints, changing environment, and unfamiliar surroundings. Projects of the embedded type are model to the company and usually exhibit temporal constraints.

3.4.2. Organic:

This category encompasses all systems that are small relative to project size

and team size and have a stable environment, familiar surroundings and relaxed interfaces. These are simple business systems, data processing systems, and small software libraries.

3.4.3. Semidetached:

The software systems falling under this category are a mix of those of organic and embedded in nature. Some examples of software of this class are operating systems, database management system, and inventory management systems class are operating systems, database management system, and inventory management systems.

Table 3.1: Organic, Semidetached and Embedded system values

TYPE OF PRODUCT	A	B	C	D
Organic	2.4	1.02	2.5	0.38
Semi Detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

For basic COCOMO Effort = a*(KLOC) b

Type =c*(effort)d

For Intermediate and Detailed COCOMO Effort = a * (KLOC) b* EAF (EAF = product of cost drivers).

Intermediate COCOMO model is a refinement of the basic model, which comes in the function of 15 attributes of the product. For each of the attributes the user of the model has to provide a rating using the following six-point scale.

LO (Low) HI (High)

VL (Very Low) VH (Very High)

NM (Normal) XH (Extra High)

The list of attributes is composed of several features of the software and includes

product, computer, personal and project attributes as follows.

3.4.4Product Attributes:

Required reliability (RELY): It is used to express an effect of software faults ranging from slight inconvenience (VL) to loss of life (VH). The nominal value (NM) denotes moderate recoverable losses.

Data bytes per DSI (DATA): The lower rating comes with lower size of a database. Complexity (CPLX): The attribute expresses code complexity again ranging from straight batch code (VL) to real time code with multiple resources scheduling (XH).

3.4.5Computer Attributes:

Execution time (TIME) and memory (STOR) constraints: This attribute identifies the percentage of computer resources used by the system. NM states that less than 50% is used; 95% is indicated by XH.]

Virtual machine volatility (VIRT): It is used to indicate the frequency of changes made to the hardware, operating system, and overall software environment. More frequent and significant changes are indicated by higher ratings.

Development turnaround time (TURN): This is a time from when a job is submitted until output becomes received. LO indicated a highly interactive environment, VH quantifies a situation when this time is longer than 12 hours.

3.4.6 Personal Attributes:

Analyst capability (ACAP) and Analyst programmer capability (PCAP).

This describes skills of the developing team. The higher the skills, the higher the rating.

Application experience (AEXP), language experience (LEXP), and virtual machine experience (VEXP)

These are used to quantify the number of experiences in each area by the development team, more experience, higher rating.

3.4.7 Project Attributes:

Modern development practices (MODP): deals with the amount of use of modern software practices such as structural programming and object-oriented approach.

Use of software tools (TOOL): is used to measure a level of sophistication of automated tools used in software development and a degree of integration among the tools being used. Higher rating describes levels in both aspects.

Schedule effects (SCED): concerns the amount of schedule compression (HI or VH), or schedule expansion (LO or VL) of the development schedule in comparison to a nominal (NM) schedule.

Table 3.2: Project Attributes

	VL	LO	NM	HI	VH	XH
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.15	1.30	
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.29	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
LEXP	1.14	1.07	1.00	0.95		

TAPTAP
A BLACKBUCKS PRODUCT

VEXP	1.21	1.10	1.00	0.90		
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	

Our project is an organic system and for intermediate

COCOMO Effort = a* (KLOC) b * EAF

KLOC = 115

For organic system a

= 2.4

b = 1.02

EAF = product of cost

Driver‘s effort = 2.4 * (0.115) ^ 1.02 * 1.30 =

1.034

Programmer month's Time for Development = C * (Effort) d =

2.5 * (1.034) ^ 0.38

= 2.71 months

Cost of programmer = Effort * cost of Programmer per month =

1.034 * 20000

= 20680

Project cost = 20000 + 20680 =

40680

TAPTAP
A BLACKBUCKS PRODUCT

CHAPTER 4
EXPERIMENTAL OR MATERIALS AND METHODS

4.1 INTRODUCTION TO REQUIREMENT SPECIFICATION:

Software Engineering by James F Peters & WitoldPedrycz Headfirst Java by Ka. A Software Requirements Specification (SRS) is a description of software product, program or set of programs that performs a set of functions in a target environment (IEEE Std. 830-1993).

a. Purpose:

The purpose of software requirements specification specifies the intentions and intended audience of the SRS.

b. Scope:

The scope of the SRS identifies the software product to be produced, the capabilities, application, relevant objects etc. We are proposed to implement Passive Aggressive Algorithm which takes the test and trained data set.

c. Definitions, Acronyms and Abbreviations Software Requirements Specification:

It‘s a description of a particular software product, program or set of programs that performs a set of function in target environment.

d. References:

IEEE Std. 830-1993, IEEE Recommended Practice for Software Requirements specifications thy Sierra and Bert Bates.

e. Overview:

The SRS contains the details of process, DFD‘s, functions of the product, user characteristics. The non-functional requirements if any are also specified.

f. Overall description:

The main functions associated with the product are described in this section of SRS. The characteristics of a user of this product are indicated. The assumptions in this section result from interaction with the project stakeholders.

4.2 REQUIREMENT ANALYSIS:

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.) Under requirement specification, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity. The Requirement phases terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase. The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium though which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

4.2.1 Product Perspective:

The application is developed in such a way that any future enhancement can be easily implementable. The project is developed in such a way that it requires minimal maintenance. The software used are open source and easy to install. The application developed should be easy to install and use. This is an independent application which can be easily run on to any system which has Python installed and Jupiter Notebook.

4.2.2 Product Features:

The application is developed in a way that _heart disease' accuracy is predicted using Random Forest. The dataset is taken from https://www.datacamp.com/community/tutorials/scikit-learn-credit-card. We can compare the accuracy for the implemented algorithms. User characteristics Application is developed in such a way that its users are v Easy to use v Error free 20 v Minimal training or no training v Patient regular monitor Assumption & Dependencies It is considered that the dataset taken fulfils all the requirements.

4.2.3 Domain Requirements:

This document is the only one that describes the requirements of the system. It is meant for the use by the developers and will also be the bases for validating the final heart

disease system. Any changes made to the requirements in the future will have to go through a formal change approval process. User Requirements User can decide on the prediction accuracy to decide on which algorithm can be used in real-time predictions. Non-Functional Requirements ÿ Dataset collected should be in the CSV format ÿ. The column values should be numerical values ÿ Training set and test set are stored as CSV files ÿ Error rates can be calculated for prediction algorithms product.

4.2.4 Requirements Efficiency:

Less time for predicting the Heart Disease Reliability: Maturity, fault tolerance and recoverability. Portability: can the software easily be transferred to another environment, including install ability.

4.2.5 Usability:

How easy it is to understand, learn and operate the software system Organizational. Requirements: Do not block some available ports through the windows firewall. Internet connection should be available Implementation Requirements The dataset collection, internet connection to install related libraries. Engineering Standard Requirements User interface is developed in python, which gets input such stock symbol.

4.2.6 Hardware Interfaces:

Ethernet on the AS/400 supports TCP/IP, Advanced Peer-to-Peer Networking (APPN) and advanced program-to-program communications (APPC). ISDN To connect AS/400 to an Integrated Services Digital Network (ISDN) for faster, more accurate data transmission. An ISDN is a public or private digital communications network that can support data, fax, image, and other services over the same physical interface. We can use other protocols on ISDN, such as IDLC and X.25. Software Interfaces Anaconda Navigator and Jupiter Notebook are used.

4.2.7 Operational Requirements:

a. **Economic:** The developed product is economic as it is not required any hardware interface etc. Environmental Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS). The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer.

b. **Health and Safety:** The software may be safety critical. If so, there are issues associated with its integrity level. The software may not be safety-critical although it forms part of a safety-critical system.

There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable. If a computer system is to run software of a high integrity level, then that system should not at the same time accommodate software of a lower integrity level.

Systems with different requirements for safety levels must be separated. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

4.3 SYSTEM REQUIREMENTS

4.3.1 Hardware Requirements:

Processor	:	above 500MHz
Ram	:	4GB
Hard Disk	:	4GB
Input device	:	Standard keyboard and Mouse
Output device	:	VGA and High-Resolution Monitor

4.3.2 SOFTWARE REQUIREMENTS:

Operating System	:	Windows 7 or higher
Programming	:	python 3.6 and related libraries :
Software		Anaconda Navigator, Jupyter Notebook and Google colab

4.4 SOFTWARE DESCRIPTION

4.4.1 Python:

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type of system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library. Python interpreters are available for many operating systems. C Python, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all its variant implementations. C Python is managed by the non-profit Python Software Foundation.

4.4.2 Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data mining and preparation. It had very little contribution towards data analysis. Pandas solved this problem.

Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

Key Features of Pandas:

Fast and efficient Data Frame object with default and customized indexing.

Tools for loading data into in-memory data objects from different file formats.

Data alignment and integrated handling of missing data.

Reshaping and pivoting of date sets.

Label-based slicing, indexing and subsetting of large data sets.

Columns from a data structure can be deleted or inserted.

Group by data for aggregation and transformations.

High performance merging and joining of data.

Time Series functionality.

4.4.3 NumPy:

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

A powerful N-dimensional array object

Sophisticated (broadcasting) functions

Tools for integrating C/C++ and Fortran code

Useful linear algebra, Fourier transform, and random number capabilities 24

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary datatypes can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

4.4.4 Sckit-Learn:

Simple and efficient tools for data mining and data analysis

Accessible to everybody, and reusable in various contexts

Built on NumPy, SciPy, and matplotlib

Open source, commercially usable - BSD license **4.4.5**

Matploit lib:

Matplotlib is a python library used to create 2D graphs and plots by using python scripts.

It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc.

It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc.

4.4.6 Jupyter Notebook:

The Jupyter Notebook is an incredibly powerful tool for interactively developing and presenting data science projects.

A notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media.

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

Uses include data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

The Notebook has support for over 40 programming languages, including Python, R, Julia, and Scala.

Notebooks can be shared with others using email, Drop box, Git Hub and the Jupyter Notebook.

Your code can produce rich, interactive output: HTML, images, videos, LATEX, and custom MIME types.

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, Tensor Flow.

4.5 ALGORITHMS

4.5.1 Logistic Regression

A popular statistical technique to predict binomial outcomes (y = 0 or 1) is Logistic Regression. Logistic regression predicts categorical outcomes (binomial / multinomial values of y). The predictions of Logistic Regression (henceforth, LogR in this article) are in the form of probabilities of an event occurring, i.e., the probability of y=1, given certain values of input variables x. Thus, the results of LogR range between 0-1.

LogR models the data points using the standard logistic function, which is an S-shaped

curve also called as sigmoid curve and is given by the equation.

$$f(x) = \frac{1}{1+e^{-(x)}}$$

Logistic Regression Assumptions:

Logistic regression requires the dependent variable to be binary.

For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.

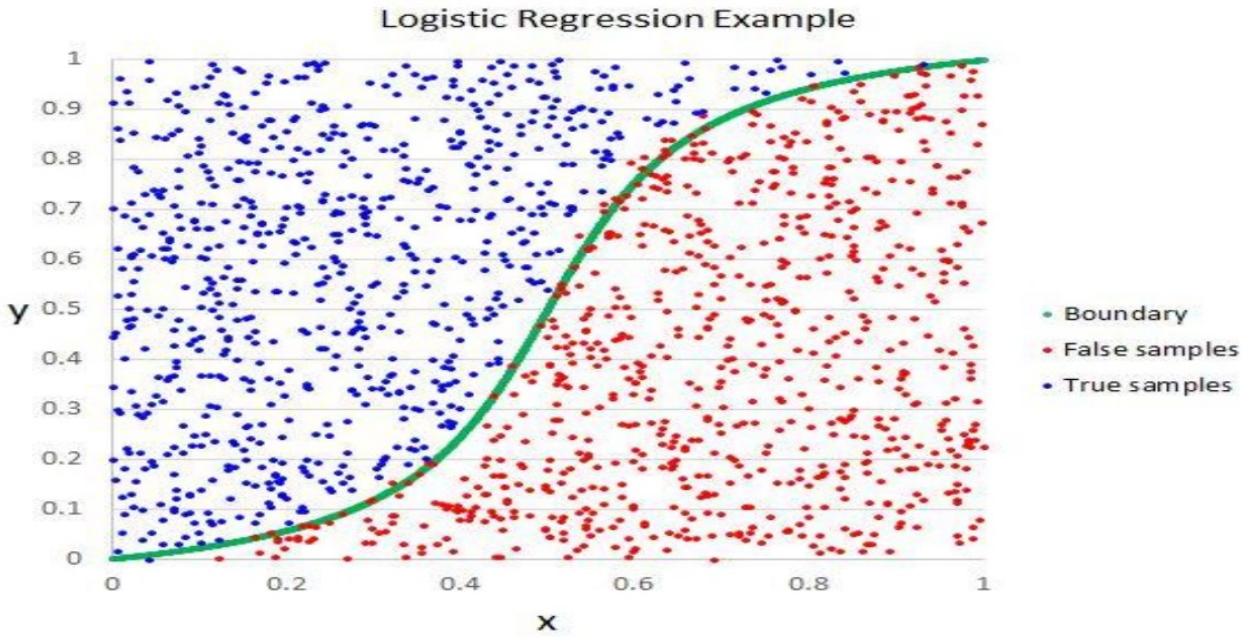
Only the meaningful variables should be included.

The independent variables should be independent of each other.

Logistic regression requires quite large sample sizes.

Even though, logistic (logit) regression is frequently used for binary variables (2 classes), it can be used for categorical dependent variables with more than 2 classes. In this case it‘s called Multinomial Logistic Regression. **Fig**

4.5.1: Logistic Regression



4.5.2 Random Forest:

Random Forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest.

Similarly, random forest creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Working of Random Forest with the help of following steps:

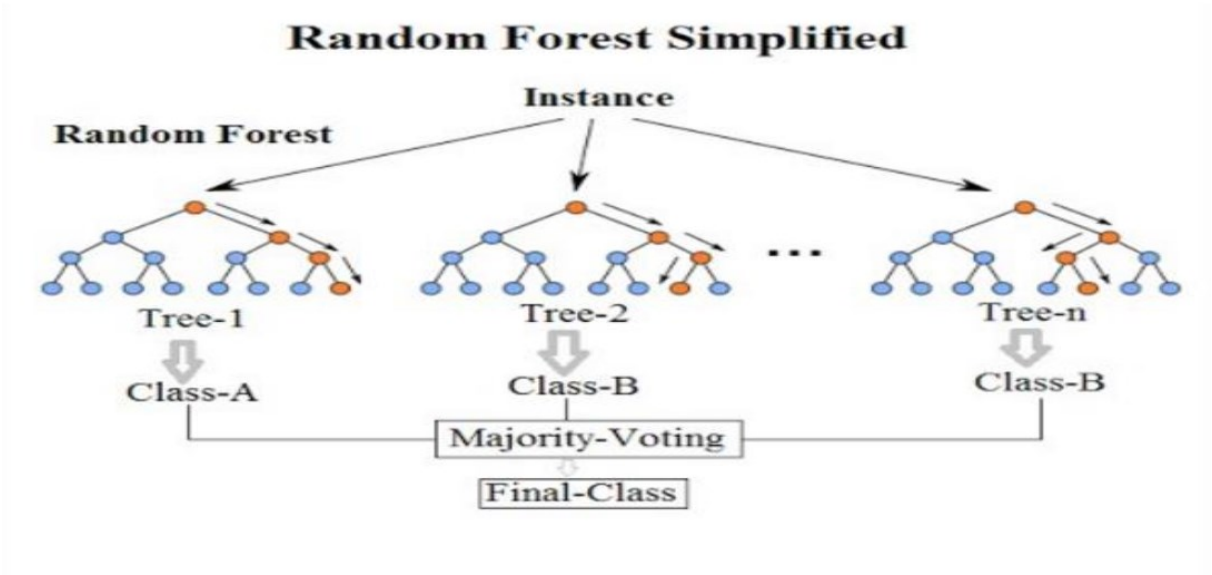
First, start with the selection of random samples from a given dataset.

Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

In this step, voting will be performed for every predicted result.

At last, select the most voted prediction results as the final prediction result. The following diagram will illustrate its working-

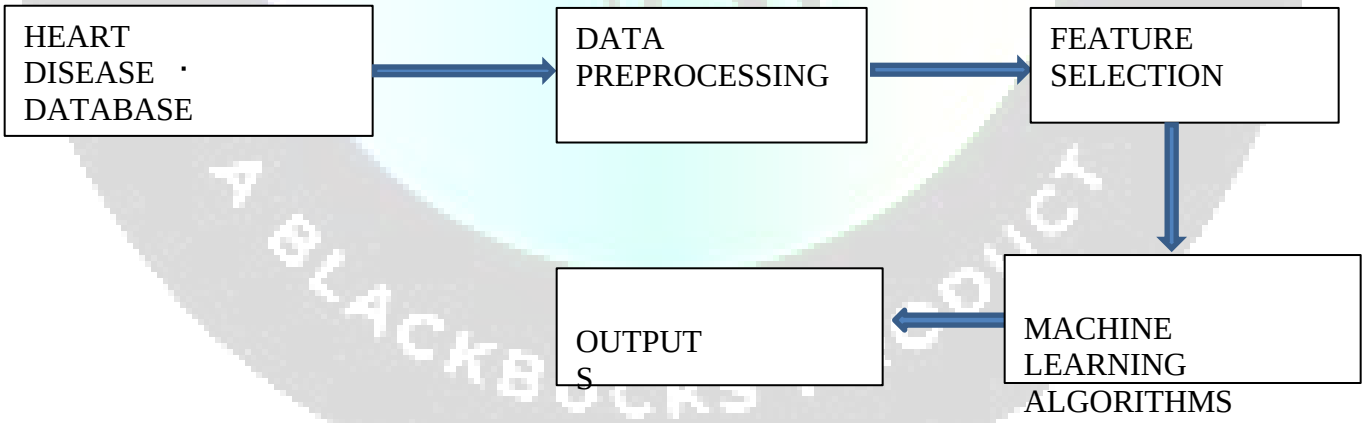
Fig 4.5.2: Random Forest Classifier



4.6 SYSTEM ARCHITECTURE

The below figure shows the process flow diagram or proposed work. First, we collected the Cleveland Heart Disease Database from UCI website then pre-processed the dataset and select 16 important features

Fig 4.6: System Architecture



For feature selection we used Recursive feature Elimination Algorithm using Chi2 method and get 16 top features. After that applied ANN and Logistic algorithm individually and compute the accuracy. Finally, we used proposed Ensemble Voting method and compute

best method for diagnosis of heart disease.

4.7 MODULES:

The entire work of this project is divided into 4 modules.

They are:

a. **Data Pre-Processing b.**

Feature

c. **Classification d.**

Prediction

a. Data Pre-processing:

This file contains all the pre-processing functions needed to process all input documents and texts. First, we read the train, test and validation data files then performed some preprocessing like tokenizing, stemming etc. There are some exploratory data analyses is performed like response variable distribution and data quality checks like null or missing values etc.

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Preprocessing of data is mainly to check the data quality. The quality can be checked by the following-

Accuracy: To check whether the data entered is correct or not.

Completeness: To check whether the data is available or not recorded.

Consistency: To check whether the same data is kept in all the places that do or do not match.

Timeliness: The data should be updated correctly

Believability: The data should be trustable.

Interpretability: The understandability of the data.

b. Feature:

Extraction In this file we have performed feature extraction and selection

methods from sci-kit learn python libraries. For feature selection, we have used methods like simple bag-of-words and n-grams and then term frequency like tf-idf weighting. We have also used word2vec and POS tagging to extract the features, though POS tagging and word2vec has not been used at this point in the project

Bag of Words:

It‘s an algorithm that transforms the text into fixed-length vectors. This is possible by counting the number of times the word is present in a document. The word occurrences allow to compare different documents and evaluate their similarities for applications, such as search, document classification, and topic modeling.

The reason for its name, ―Bag-Of-Words‖, is due to the fact that it represents the sentence as a bag of terms. It doesn‘t consider the order and the structure of the words, but it only checks if the words appear in the document.

N-grams:

N-grams are continuous sequences of words or symbols or tokens in a document. In technical terms, they can be defined as the neighbouring sequences of items in a document. They come into play when we deal with text data in NLP(Natural Language Processing) tasks.

TF-IDF Weighting:

TF-IDF stands for term frequency-inverse document frequency and it is a measure, used in the fields of information retrieval (IR) and machine learning, that can quantify the importance or relevance of string representations (words, phrases, lemmas, etc) in a document amongst a collection of documents (also known as a corpus).

c. Classification:

Here we have built all the classifiers for the breast cancer diseases detection. 26

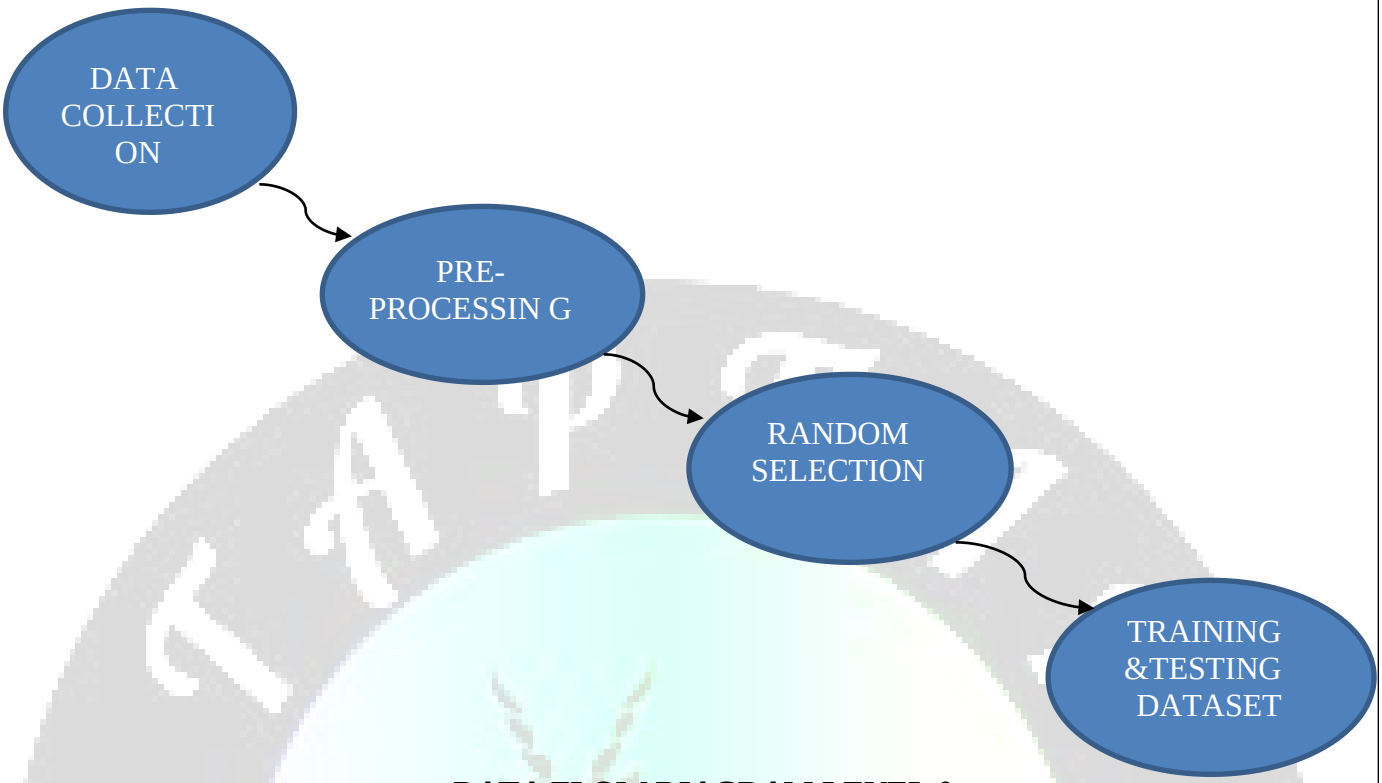
The extracted features are fed into different classifiers. We have used Naive-bayes, Logistic Regression, Linear SVM, Stochastic gradient decent and Random Forest classifiers from sklearn. Each of the extracted features was used in all the classifiers. Once fitting the model, we compared the f1 score and checked the confusion matrix. After fitting all the classifiers, 2 best performing models were selected as candidate models for heart diseases classification. We have performed parameter tuning by implementing GridSearchCV methods on these candidate models and chosen best performing parameters for these classifiers. Finally selected model was used for heart disease detection with the probability of truth. In Addition to this, we have also extracted the top 50 features from our term-frequency tf-idf Vectorizer to see what words are most and important in each of the classes. We have also used Precision-Recall and learning curves to see how training and test set performs when we increase the amount of data in our classifiers.

d. Prediction:

Our finally selected and best performing classifier was algorithm which was then saved on disk with name final_model.sav. Once you close this repository, this model will be copied to user's machine and will be used by prediction.py file to classify the heart diseases. It takes a news article as input from user then model is used for final classification output that is shown to user along with probability of truth. **4.8 DATA FLOW DIAGRAM:**

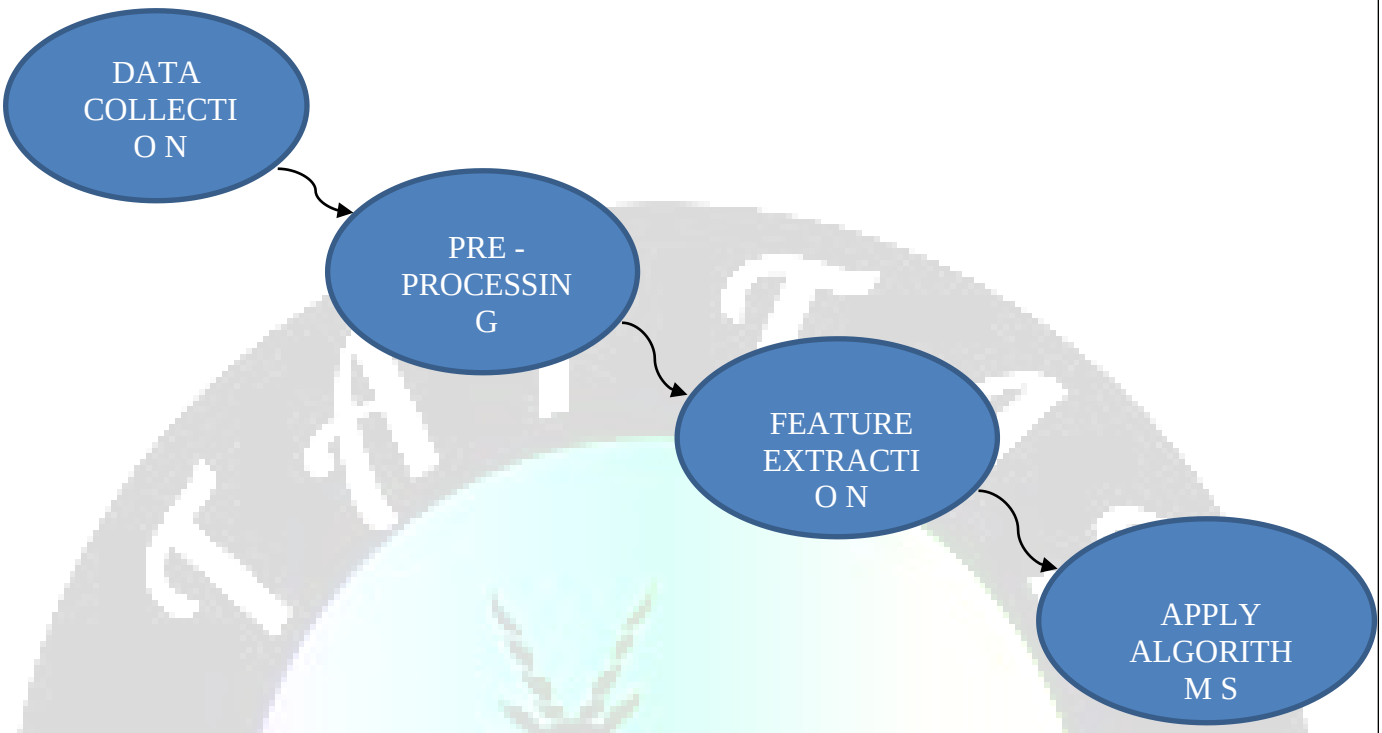
The data flow diagram (DFD) is one of the most important tools used by system analysis. Data flow diagrams are made up of number of symbols, which represents system components. Most data flow modeling methods use four kinds of symbols: Processes, Data stores, Data flows and external entities. These symbols are used to represent four kinds of system components. Circles in DFD represent processes. Data Flow represented by a thin line in the DFD, and each data store has a unique name and square or rectangle represents external entities.

LEVEL:0



DATA FLOW DIAGRAM LEVEL 0

LEVEL 1:

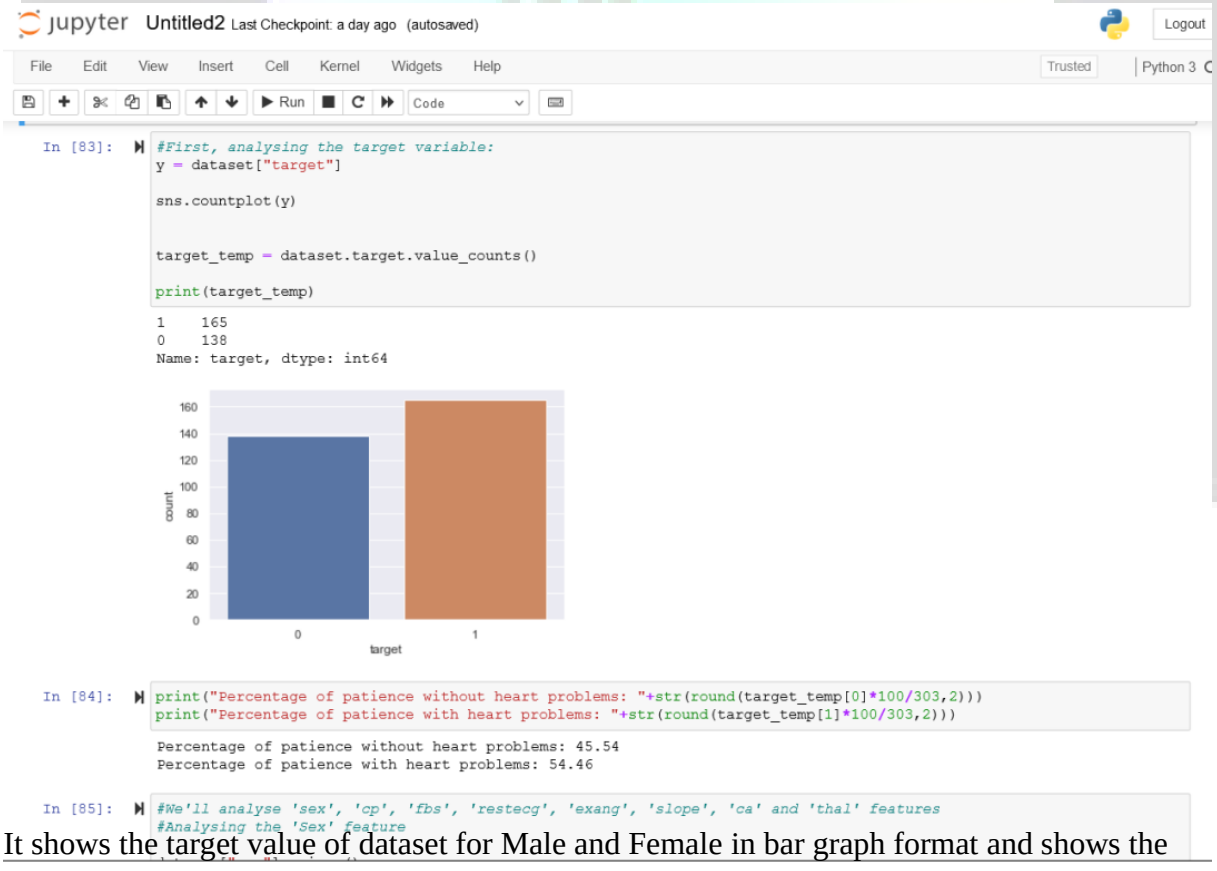


Data Flow Diagram Level 1

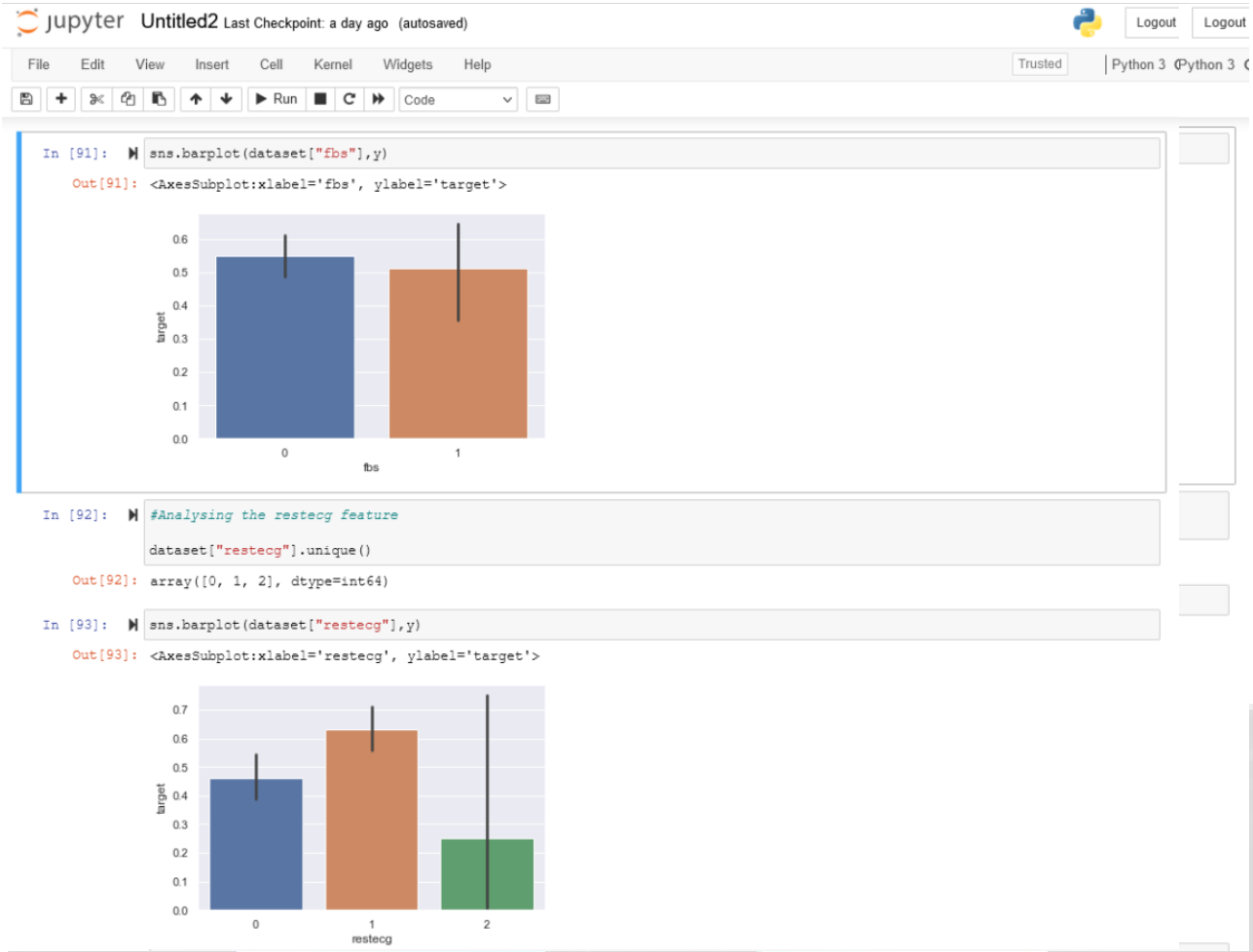
CHAPTER 5

RESULT AND DISCUSSION, PERFORMANCE ANALYSIS

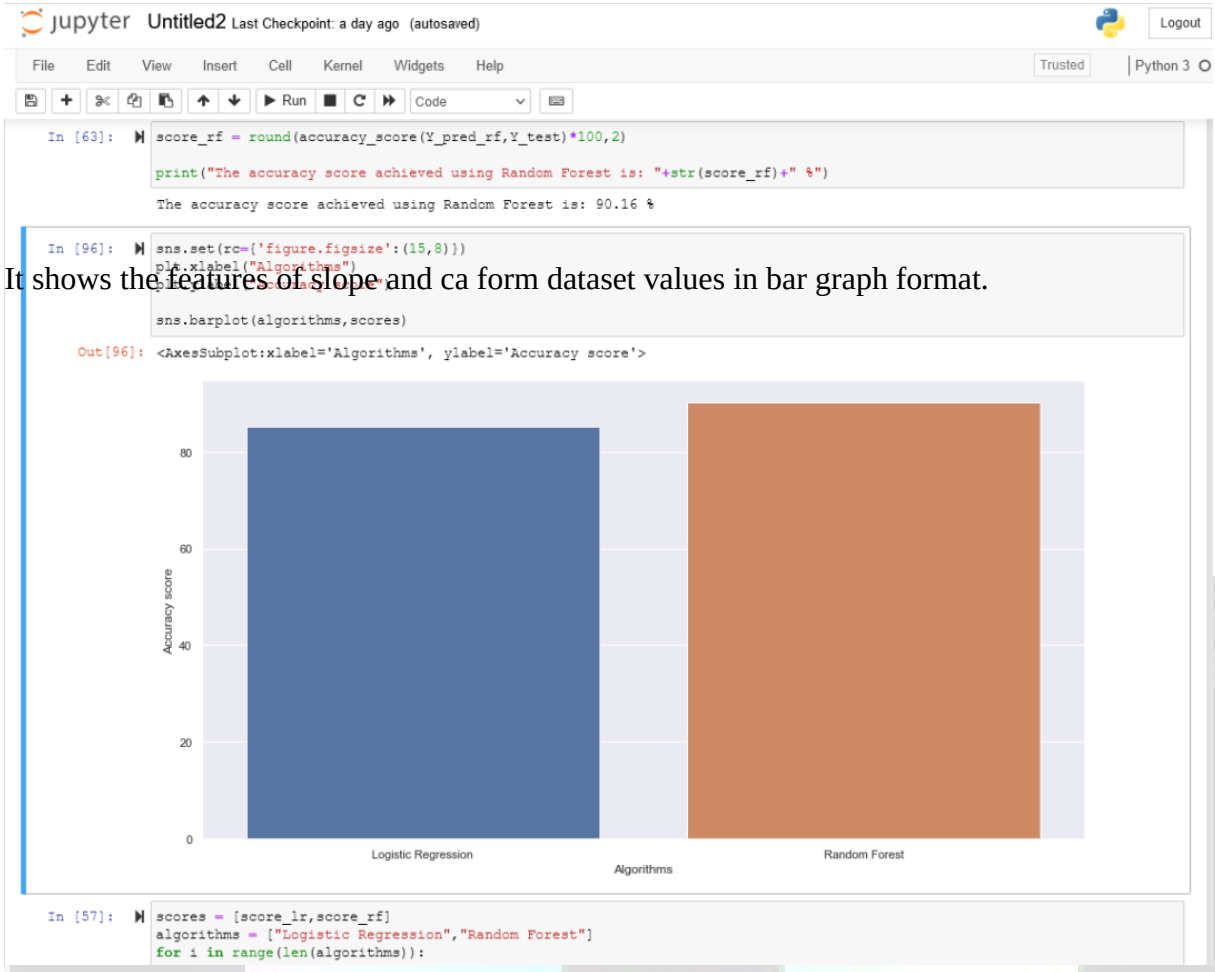
In this project, we introduce about the heart disease prediction system with different classifier techniques for the prediction of heart disease. The techniques are Random Forest and Logistic Regression: we have analyzed that the Random Forest has better accuracy as compared to Logistic Regression. Our purpose is to improve the performance of the Random Forest by removing unnecessary and irrelevant attributes from the dataset and only picking those that are most informative for the classification task.



It shows the target value of dataset for Male and Female in bar graph format and shows the percentage of patience having with or without heart problem.



It shows the values of fbs and restecg with a bar graph and analyzing the restecg and fbs features.



It shows the features of slope and ca form dataset values in bar graph format.

Finally, it shows that Random Forest algorithm has more accuracy than Logistic Regression from the dataset values and also shows the accuracy percentage of Random Forest Algorithm.

Chapter 6

SUMMARY AND CONCLUSION

6.1 Summary

This project objective is to predict the Heart Disease Using Machine Learning. So, this paper a machine learning algorithm is proposed for the implementation of a heart disease prediction system which was validated on two open access heart disease prediction datasets.

6.2 Conclusion

In this project, we introduce about the heart disease prediction system with different classifier techniques for the prediction of heart disease. The techniques are Random Forest and Logistic Regression: we have analyzed that the Random Forest has better accuracy as compared to Logistic Regression. Our purpose is to improve the performance of the Random Forest by removing unnecessary and irrelevant attributes from the dataset and only picking those that are most informative for the classification task

References

[1] P.K. Anooj, ―Clinical decision support system: Risk level prediction of heart disease using weighted fuzzy rules‖; Journal of King Saud University – Computer and Information Sciences (2012) 24, 27–40. Computer Science & Information Technology (CS & IT) 59.

[2] Nidhi Bhatla, Kiran Jyoti "An Analysis of Heart Disease Prediction using Dif ferent Data Mining Techniques". International Journal of Engineering Research & Technology.