

Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

G SUMANTH MALLESH (37110742)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

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 **G SUMANTH MALLESH (37110742)** who carried out the project entitled "**Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare**" under my supervision from **NOVEMBER 2020** to **MARCH 2021**.

Internal Guide
Dr. L.LAKSHMANAN

Head of the Department

Dr. S. Vigneshwari, M.E., Ph.D.,

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

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **G SUMANTH MALLESH** hereby declare that the Project Report entitled “**Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare**” is done by me under the guidance of **Dr. L.Lakshmanan** Department of Computer Science and Engineering 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

SIGNATURE 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, **Dr. S. Vigneswari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the 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.L.Lakshmanan**, for her valuable guidance, suggestions and constant 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

Heart disease (HD) is the critical health issue and numerous people have been suffered by this disease around the world . The HD occurs with common symptoms of breath shortness, physical body weakness and, feet are swollen . Researchers try to come across an efficient technique for the detection of heart disease, as the current diagnosis techniques of heart disease are not much effective in early time identification due to several reasons, such as accuracy and execution time . The diagnosis and treatment of heart disease is extremely difficult when modern technology and medical experts are not available. The effective diagnosis and proper treatment can save the lives of many people Diagnosis of HD is traditionally done by the analysis of the medical history of the patient, physical examination report and analysis of concerned symptoms by a physician. But the results obtained from this diagnosis method are not accurate in identifying the patient of HD. Moreover, it is expensive and computationally difficult to analyze.

TABLE OF CONTENTS

ABSTRACT

LIST OF FIGURES

CHAPTER NO.	TITLE	PAGE NO.
1.	INTRODUCTION	1
2.	LITERATURE SURVEY	2
	2.1 Literature survey 1	2
	2.2 Literature survey 2	3
	2.3 Literature survey 3	4
	2.4 Literature survey 4	5
3.	AIM AND SCOPE	6
	3.1 AIM OF PROJECT	6
	3.2 SCOPE OF THE PROJECT	6
4.	SYSTEM ANALYSIS	7
	4.1 EXISTING SYSTEM	7
	4.2 DRAWBACKS IN EXISTING SYSTEM	7
	4.3 PROPOSED SYSTEM	7
	4.4 ADVANTAGES OF PROPOSED SYSTEM	8
	4.5 REQUIREMENT SPECIFICATION	8
	4.6 SOFTWARE DESCRIPTION	9
5.	SYSTEM DESIGN AND METHODOLOGY	12
	5.1 SYSTEM DESIGN	12
	5.2 SYSTEM ARCHITECTURE	12
	5.3 SYSTEM IMPLEMENTATION	13
	5.3.1 Exploratory Data	14

	Evaluation	
	5.3.2 Pre-processing	14
	5.3.3 Feature Engineering	15
	5.3.4 Prediction	16
6.	RESULTS AND	17
	DISCUSSION	
7.	CONCLUSION AND	22
	FUTURE ENHANCEMENT	
	7.1 CONCLUSION	22
	7.2 FUTURE	22
	ENHANCEMENT	
	REFERENCES	23
	APPENDIX	24
	A.PLAGIARISM REPORT	24
	B. JOURNAL PAPER	25
	C. SOURCE CODE	29

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
4.2.1	ARCHITECTURE	11
5.1	DATASETS	15
5.2	TARGET	16
5.3	PREDICTION ON THEIR AGE	16
5.4	PREDICTION OF MALE AND FEMALE PATIENTS	17
5.5	PREDICTION BASED ON FBS	17
5.6	CHEST PAIN TYPE	18
5.7	ACCURACY OF ALGORITHMS	18
5.8 & 5.9	CONFUSION MATRIX	19

CHAPTER 1

INTRODUCTION

N

INTRODUCTION

Over the last decade, heart disease or cardiovascular remains the primary basis of death worldwide. An estimate by the World Health Organization, that over 17.9 million deaths occur every year worldwide because of cardiovascular disease, and of these deaths, 80% are because of coronary artery disease and cerebral stroke. The vast number of deaths is common amongst low and middle-income countries. Many predisposing factors such as personal and professional habits and genetic predisposition accounts for heart disease. Various habitual risk factors such as smoking, overuse of alcohol and caffeine, stress, and physical inactivity along with other physiological factors like obesity, hypertension, high blood cholesterol, and pre-existing heart conditions are predisposing factors for heart disease. The efficient and accurate and early medical diagnosis of heart disease plays a crucial role in taking preventive measures to prevent death. Data mining refers to the extraction of required information from huge datasets in various fields such as the medical field, business field, and educational field. Machine learning is one of the most rapidly evolving domains of artificial intelligence. These algorithms can analyse huge data from various fields, one such important field is the medical field. It is a substitute to routine prediction modeling approach using a computer to gain an understanding of complex and non-linear interactions among different factors by reducing the errors in predicted and factual outcomes. Data mining is exploring huge datasets to extract hidden crucial decision-making information from a collection of a past repository for future analysis. The medical field comprises tremendous data of patients. These data need mining by various machine learning algorithms. Healthcare professionals do analysis of these data to achieve effective diagnostic decision by healthcare professionals. Medical data mining using classification algorithms provides clinical aid through analysis. It tests the classification algorithms to predict heart disease in patients

**CHAPT
ER 2
LITERATUR
E SURVEY**

2.1 Literature Survey 1

Title	Design And Implementing Heart Disease Prediction Using Naives Bayesian
Authors	Anjan Nikhil Repaka
Published Year	2019
Efficiency	The speedy advancement of technology has led to remarkable rise in mobile health technology that being one of the web application. The required data is assembled in a standardized form. For predicting the chances of heart disease in a patient, the following attributes are being fetched from the medical profiles, these include: age, BP, cholesterol, sex, blood sugar etc.
Drawbacks	Nowadays the hospitals are adopting the culture of hospital IMS in order to handle their or patients data systematically and effectively. As per today's advanced and hi-tech living style, majority of the people are contracting heart disease which gives a sudden jolt to an individual that at times one lacks time to get treated immediately.
Description	Data mining, a great developing technique that revolves around exploring and digging out significant information from massive collection of data which can be further beneficial in examining and drawing out patterns for making business related decisions. Talking about the Medical domain, implementation of data mining in this field can yield in discovering and withdrawing valuable patterns and information which can prove beneficial in performing clinical diagnosis. The research focuses on heart disease diagnosis by considering previous data and information. To achieve this SHDP is built via Navies Bayesian in order to predict risk factors concerning heart disease. The speedy advancement of technology has led to remarkable rise in

mobile health technology that being one of the web application. The

required data is assembled in a standardized form. For predicting the chances of heart disease in a patient, the following attributes are being fetched from the medical profiles, these include: age, BP, cholesterol, sex, blood sugar etc... The collected attributes acts as input for the Navies Bayesian classification for predicting heart disease. The dataset utilized is split into two sections, 80% dataset is utilized for training and rest.

2.2 Literature Survey 2

Title	Prediction of Heart Disease Using Machine Learning
Authors	Aditi Gavhane
Published Year	2018
Efficiency	The machine learning algorithm neural networks has proven to be the most accurate and reliable algorithm and hence used in the proposed system.
Drawbacks	There is no dearth of records regarding medical symptoms of patients suffering heart strokes. However the potential they have- to help us foretell similar possibilities in seemingly healthy adults are going unnoticed.
Description	with the rampant increase in the heart stroke rates at juvenile ages, we need to put a system in place to be able to detect the symptoms of a heart stroke at an early stage and thus prevent it. It is impractical for a common man to frequently undergo costly tests like the ECG and thus there needs to be a system in place which is handy and at the same time reliable, in predicting the chances of a heart disease. Thus we propose to develop an application which can predict the vulnerability of a heart disease given basic symptoms like age, sex, pulse rate etc.

2.3 Literature Survey 3

Title	Smartphone Based Ischemic Heart Disease (Heart Attack) Risk Prediction using Clinical Data and Data Mining Approaches, a Prototype Design
Authors	M. Raihan
Published Year	2016
Efficiency	Preventive medicine is gaining importance and popularity globally. For some diseases, prevention is better than cure. IHD episodes can be primordially, primarily and secondarily prevented. A strategy to prevent development of risk factors or control of risk factors will modify the risk of IHD.
Drawbacks	Calculating and categorizing a person into a risk level may help the person to motivate for a preventive health check so that mortality can be reduced as much as possible.
Description	We developed a simple approach to predict risk of developing Ischemic Heart Disease (IHD) (Heart Attack) using smartphone. An Android based prototype software has been developed by integrating clinical data obtained from patients admitted with IHD. The clinical data from 787 patients has been analyzed and correlated with the risk factors like Hypertension, Diabetes, Dyslipidemia, Smoking, Family History, Obesity, Stress and existing clinical symptom which may suggest underlying non detected IHD. The data was mined with data mining technology and a score is generated. Risks are classified into low, medium and high for IHD. On comparing and categorizing the patients whose data is obtained for generating the score; we found there is a significant correlation of having a cardiac event when low & high and medium & high category are compared ; $p=0.0001$ and respectively. Our research is to make simple approach to detect the IHD risk and aware the population to get themselves evaluated by a

cardiologist to avoid sudden deaths. Currently available tools has some limitations which makes them underutilized by population. Our research product may reduce this limitation and promote risk evaluation on time.

2.4 Literature Survey 4

Title	Heart disease prediction system based on hidden naïve bayes Classifier
Authors	M.A.Jabbar
Published Year	2016
Efficiency	Accuracy obtained by our model is100%.to the best of our knowledge this is the highest accuracy obtained by other models reported in the literature.
Drawbacks	Classification is a pervasive problem that is used for many applications and to find unknown sample.
Description	Coronary heart disease is a major cause of death world wide. The diagnosis of heart disease is a tedious task. There is a need for an intelligent decision support system for disease prediction. Data mining techniques are often used to classify whether a patient is normal or having heart disease. Hidden Naïve Bayes is a data mining model that relaxes the traditional Naïve Bayes conditional independence assumption. Our proposed model claims that the Hidden Naïve Bayes (HNB) can be applied to heart disease classification (prediction). Our experimental results on heart disease data set show that the HNB records 100% in terms of accuracy and out performs naïve bayes.

CHAPTER 3

AIM AND SCOPE

3.1 AIM OF PROJECT

The aim of the project is On time and efficient identification of heart disease plays a keyrole in healthcare, particularly in the field of cardiology. In this article, we proposed an efficient and accurate system to diagnosis heart disease and the system is based on machine learningtechniques.

3.2 SCOPE OF THE PROJECT

The main contributions of this project therefore are:

- Data Analysis
- Dataset Preprocessing
- Training the Model
- Testing of Dataset

CHAPTER 4

SYSTEM ANALYSIS

4.1 EXISTING SYSTEM

Data analytics is useful for prediction from more information and it helps medical centre to predict of various disease. Huge amount of patient related data is maintained on monthly basis. The stored data can be useful for source of predicting the occurrence of future disease. Some of the data mining and machine learning techniques are used to predict the heart disease, such as Random Forest, Decision tree, Fuzzy Logic, This paper provides an insight of the existing algorithm and it gives an overall summary of the existing work.

4.2 DRAWBACKS IN EXISTING SYSTEM

- This existing system only compares about the various algorithm not an efficiency to predict the heart Disease
- It won't have any suggestion for treatment

4.3 PROPOSED SYSTEM

The health care industries are often very much interested for segregating the patients according to the health, specially the condition of heart. Machine Learning technique has become quite helpful in this situation before taking any decision on this particular field. The objective of this project is to create a validated simple machine learning model that can improve the diagnosis of heart diseases in right ways. In this project, we are going to describe a very simple system which is created based on supervised machine learning algorithm.

4.4 ADVANTAGES OF PROPOSED SYSTEM

- It have both prediction and suggestion for treatment
- Efficiency of prediction is above 90%
- Low computation power is enough

4.5 REQUIREMENT SPECIFICATION

Clearly defined requirements are essential signs on the road that leads to a successful project. They establish a formal agreement between a client and a provider that they are both working to reach the same goal. High-quality, detailed requirements also help mitigate financial risks and keep the project on a schedule. According to the Business Analysis Body of Knowledge definition, requirements are a usable representation of a need.

Creating requirements is a complex task as it includes a set of processes such as elicitation, analysis, specification, validation, and management. In this article, we'll discuss the main types of requirements for software products and provide a number of recommendations for their use.

HARDWARE REQUIREMENT

Hardware	Minimum Requirement
Disk Space	32 GB or more,10 GB or more for Foundation Edition
Processor	1.4 GHz 64 bit
Memory	512 MB
Display	(800 × 600) Capable video adapter and monitor

SOFTWARE REQUIREMENT

Backend Technologies

- ✓ Python
- ✓ Numpy
- ✓ Sci-learn
- ✓ Eclipse IDE

Frontend Technologies

- ✓ Web Technologies

✓ Bootstrap

4.6 SOFTWARE DESCRIPTION

A.PYTHON

- Python is a free, open-source programming language. Therefore, all you have to do is install Python once, and you can start working with it. Not to mention that you can contribute your own code to the community. Python is also a cross-platform compatible language. So, what does this mean? Well, you can install and run Python on several operating systems. Whether you have a Windows, Mac or Linux, you can rest assure that Python will work on all these operating systems.
- Python is also a great visualization tool. It provides libraries such as Matplotlib, seaborn and bokeh to create stunning visualizations.
- In addition, Python is the most popular language for machine learning and deep learning. As a matter of fact, today, all top organizations are investing in Python to implement machine learning in the back-end.

B.PANDAS

- Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things. The DataFrame is one of these structures. Pandas is a high-level data manipulation tool developed by Wes McKinney. It is built on the Numpy package and its key data structure is called the DataFrame. DataFrames allow you to store and manipulate tabular data in rows of observations and columns of variables.
- Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.

C.NUMPY

- Numpy is the core library for scientific computing in Python. It provides a high-performance multidimensional array object, and tools for working with these arrays. If you are already familiar with MATLAB.

- A numpy array is a grid of values, all of the same type, and is indexed by a tuple of nonnegative integers. The number of dimensions is the rank of the array; the shape of an array is a tuple of integers giving the size of the array along each dimension. NumPy is, just like SciPy, Scikit-Learn, Pandas, etc.
- NumPy is a Python library that is the core library for scientific computing in Python. It contains a collection of tools and techniques that can be used to solve on a computer mathematical models of problems in Science and Engineering. One of these tools is a high-performance multidimensional array object that is a powerful data structure for efficient computation of arrays and matrices.
- One of the limitations of NumPy is that all the elements in an array have to be of the same type, so if we include the header row, all the elements in the array will be read in as strings. Because we want to be able to do computations like find the average quality of the wines, we need the elements to all be floats.

D.MATPLOTLIB

- Plotting of data can be extensively made possible in an interactive way by Matplotlib, which is a plotting library that can be demonstrated in Python scripts. Plotting of graphs is a part of data visualization, and this property can be achieved by making use of Matplotlib. Matplotlib makes use of many general-purpose GUI toolkits, such as wxPython, Tkinter, QT, etc., in order to provide object-oriented APIs for embedding plots into applications. John D. Hunter was the person who originally wrote Matplotlib, and its lead developer was Michael Droettboom. One of the free and open-source Python library which is basically used for technical and scientific computing is Python SciPy.
- Matplotlib is widely used in SciPy as most scientific calculations require plotting of graphs and diagrams. Matplotlib is a plotting library like GNUplot. The main advantage towards GNUplot is the fact that Matplotlib is a Python module. Due to the growing interest in python the popularity of matplotlib is continually rising as well. Another reason for the attractiveness of Matplotlib lies in the fact that it is widely considered to be a perfect alternative to MATLAB.

- Whereas MATLAB is expensive and closed source, Matplotlib is free and open source code. It is also object-oriented and can be used in an object oriented way. Furthermore it can be used with general-purpose GUI toolkits like wxPython, Qt, and GTK+. There is also a procedural "pylab", which designed to closely resemble that of MATLAB.

E.ABOUT IDE

IDE – Integrated Development Environment. It is software suite. It contains different instruments for web application software engineers to compose and test applications.

Programming Web application software engineers utilize wide scope of devices habitually all through application source code creation, building and application testing. Programming Development apparatuses frequently incorporate word processors for source code creation, code libraries for reuse, compilers and test stages.

An Integrated Development Environment (IDE) is a product application that gives a programming domain to streamline creating and troubleshooting programming. Instead of playing out every one of the means required to make an executable program as disconnected individual assignments, it brings every one of the devices required into one application and workspace. Every one of the devices has a familiarity with the earth, and they cooperate to display a consistent improvement set for the designer.

An IDE typically has an editor and a compiler window. There, we can write our code, run it, and compile it. Hence, it becomes much easier to develop applications using an IDE. The major reasons for using an IDE are given below:

We can easily write our code in its text editor window.

To store the resource files of a project, we have a project editor window.

It makes debugging of the code easier.

An IDE's autosuggest feature helps us in writing efficient code.

We can import other projects for integrating with the existing one.

Without an IDE, a product web application software engineer must choose, send, incorporate and deal with these instruments independently. An IDE brings huge numbers of those improvement related instruments all together system, application or administration. The incorporated toolset is intended to streamline programming advancement and can distinguish and limit coding missteps and grammatical errors.

CHAPTER 5

SYSTEM DESIGN AND METHODOLOGY

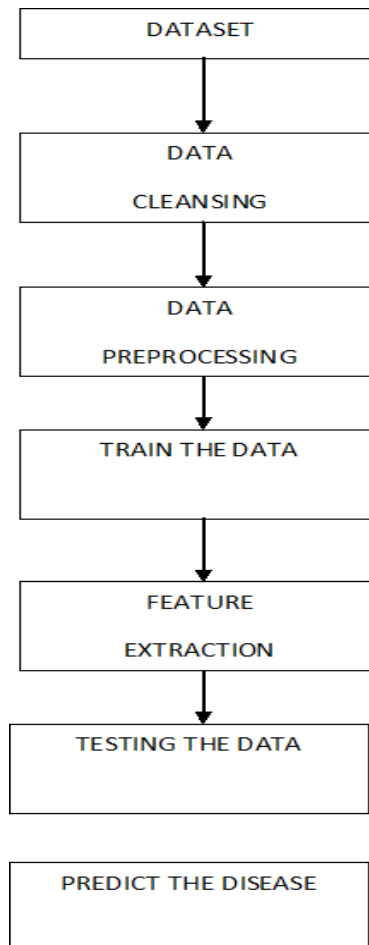
5.1 SYSTEM DESIGN

A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system. Mainly, UML has been used as a general-purpose modeling language in the field of software engineering. However, it has now found its way into the documentation of several business processes or workflows. For example, activity diagrams, a type of UML diagram, can be used as a replacement for flowcharts. They provide both a more standardized way of modeling workflows as well as a wider range of features to improve readability and efficiency.

5.2 SYSTEM ARCHITECTURE

An architecture diagram is a graphical representation of a set of concepts, that are part of an architecture, including their principles, elements and components. What is a diagram? What are the types of diagrams for architecture? The Dragon1 open EA Method makes it very clear: if a diagram does not show a concept, principle or part of a principle, it is NOT an architecture diagram, because it does not show (a part of) the architecture. There are many kinds of architecture diagrams, like a software architecture diagram, system architecture diagram, application architecture diagram, security architecture diagram, etc.

For system developers, they need system architecture diagrams to understand, clarify, and communicate ideas about the system structure and the user requirements that the system must support. It's a basic framework can be used at the system planning phase helping partners understand the architecture, discuss changes, and communicate intentions clearly.



4.2.1 ARCHITECTURE

5.3 SYSTEM IMPLEMENTATION

5.3.1 Exploratory Data Evaluation

Exploratory Data Analysis (EDA) is the first step in your data analysis process. Here, you make sense of the data you have and then figure out what questions you want to ask and how to frame them, as well as how best to manipulate your available data sources to get the answers you need. You do this by taking a broad look at patterns, trends, outliers, unexpected results and so on in your existing data, using visual and quantitative methods to get a sense of the story this tells. Exploratory Data Analysis is valuable to data science projects since it allows to get closer to the certainty that the future results will be valid, correctly interpreted, and applicable to the desired business contexts. Such level of certainty can be achieved only after raw data is validated and checked for anomalies, ensuring that the data set was collected without errors. EDA also helps to find insights that were not evident or worth investigating to business stakeholders and data scientists but

can be very informative about a particular business. EDA is performed in order to define and refine the selection of feature variables that will be used for machine learning. Once data scientists become familiar with the data set, they often have to return to feature engineering step, since the initial features may turn out not to be serving their intended purpose. Once the EDA stage is complete, data scientists get a firm feature set they need for supervised and unsupervised machine learning.

5.3.2 Pre-processing

Sometimes you may find some data are missing in the dataset. We need to be equipped to handle the problem when we come across them. Obviously you could remove the entire line of data but what if you are unknowingly removing crucial information? Of course we would not want to do that. One of the most common idea to handle the problem is to take a mean of all the values of the same column and have it to replace the missing data. The library that we are going to use for the task is called Scikit Learn preprocessing. It contains a class called Imputer which will help us take care of the missing data.

Sometimes our data is in qualitative form, that is we have texts as our data. We can find categories in text form. Now it gets complicated for machines to understand texts and process them, rather than numbers, since the models are based on mathematical equations and calculations. Therefore, we have to encode the categorical data. Now we need to split our dataset into two sets — a Training set and a Test set. We will train our machine learning models on our training set, i.e our machine learning models will try to understand any correlations in our training set and then we will test the models on our test set to check how accurately it can predict. A general rule of the thumb is to allocate 80% of the dataset to training set and the remaining 20% to test set. For this task, we will import `test_train_split` from `model_selection` library of scikit

5.3.3 Feature Engineering

Filter methods are generally used as a preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation co-efficients.

Pearson's Correlation: It is used as a measure for quantifying linear dependence between

two continuous variables X and Y. Its value varies from -1 to +1.

LDA: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

Chi-Square: It is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution

5.3.4 Prediction

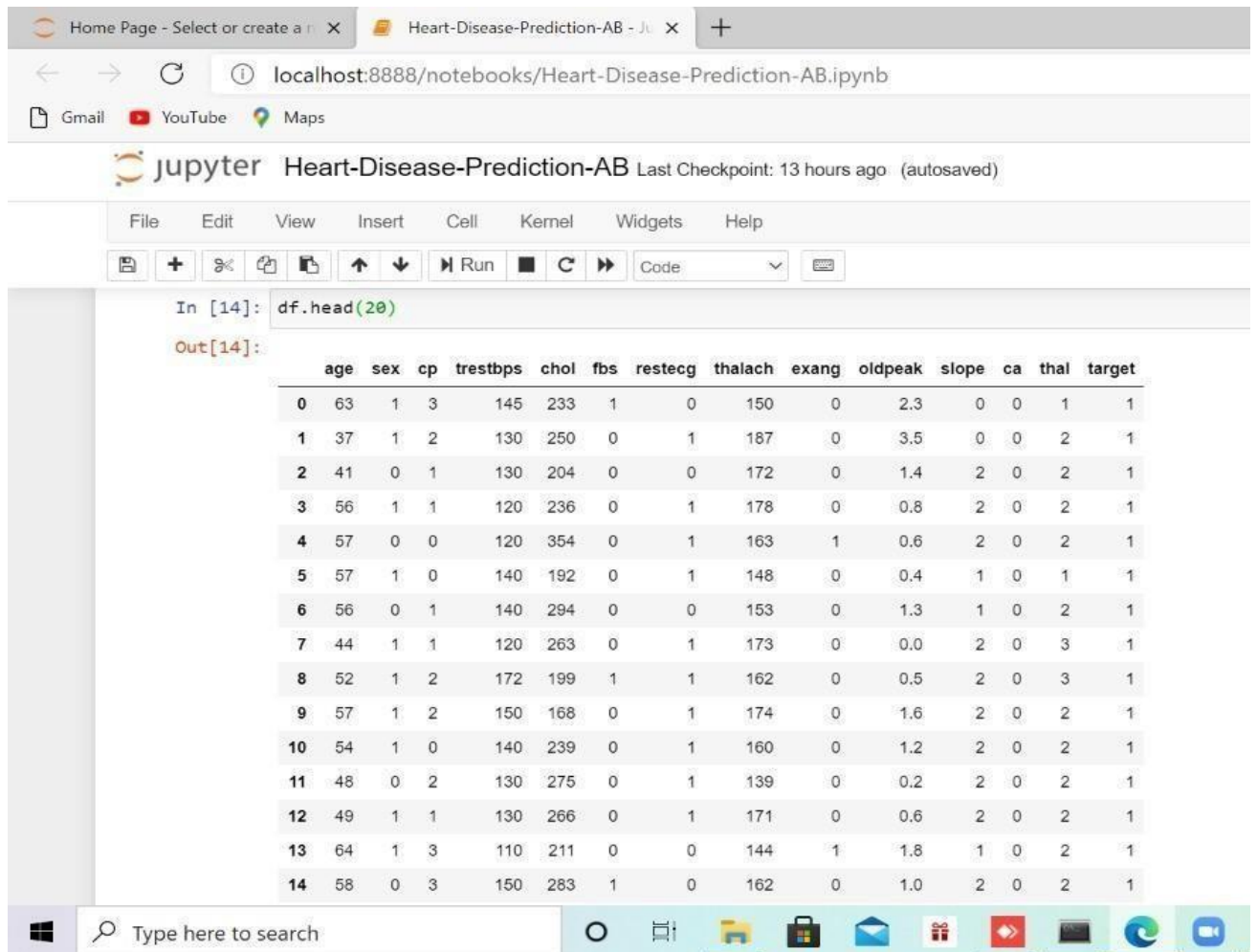
Once training is complete, it's time to see if the model is any good, using Evaluation. This is where that dataset that we set aside earlier comes into play. Evaluation allows us to test our model against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world. A good rule of thumb I use for a training-evaluation split somewhere on the order of 80/20 or 70/30. Much of this depends on the size of the original source dataset. If you have a lot of data, perhaps you don't need as big of a fraction for the evaluation dataset. Once you've done evaluation, it's possible that you want to see if you can further improve your training in any way. We can do this by tuning our parameters. There were a few parameters we implicitly assumed when we did our training, and now is a good time to go back and test those assumptions and try other values. A tree has many analogies in real life, and turns out that it has influenced a wide area of machine learning, covering both classification and regression. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. As the name goes, it uses a tree-like model of decisions. A decision tree is drawn upside down with its root at the top. In the image on the left, the bold text in black represents a condition/internal node, based on which the tree splits into branches/ edges. The end of the branch that doesn't split anymore is the decision/leaf, in this case, whether the passenger died or survived, represented as red and green text respectively.

Although, a real dataset will have a lot more features and this will just be a branch in a much bigger tree, but you can't ignore the simplicity of this algorithm. The feature

importance is clear and relations can be viewed easily. This methodology is more commonly known as learning decision tree from data and above tree is called Classification tree as the target is to classify passenger as survived or died. Regression trees are represented in the same manner, just they predict continuous values like price of a house. In general, Decision Tree algorithms are referred to as CART or Classification and Regression Trees.

CHAPTER 6

RESULTS AND DISCUSSION



The screenshot shows a Jupyter Notebook titled "Heart-Disease-Prediction-AB" running on a local host. The notebook interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations and execution. The current cell contains the code `df.head(20)`, and the output displays the first 20 rows of a dataset. The dataset has 15 columns: age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, and target. The target column contains binary values (0 or 1), indicating the presence or absence of heart disease.

	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	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1
6	56	0	1	140	294	0	0	153	0	1.3	1	0	2	1
7	44	1	1	120	263	0	1	173	0	0.0	2	0	3	1
8	52	1	2	172	199	1	1	162	0	0.5	2	0	3	1
9	57	1	2	150	168	0	1	174	0	1.6	2	0	2	1
10	54	1	0	140	239	0	1	160	0	1.2	2	0	2	1
11	48	0	2	130	275	0	1	139	0	0.2	2	0	2	1
12	49	1	1	130	266	0	1	171	0	0.6	2	0	2	1
13	64	1	3	110	211	0	0	144	1	1.8	1	0	2	1
14	58	0	3	150	283	1	0	162	0	1.0	2	0	2	1

FIG 6.1: DATASETS

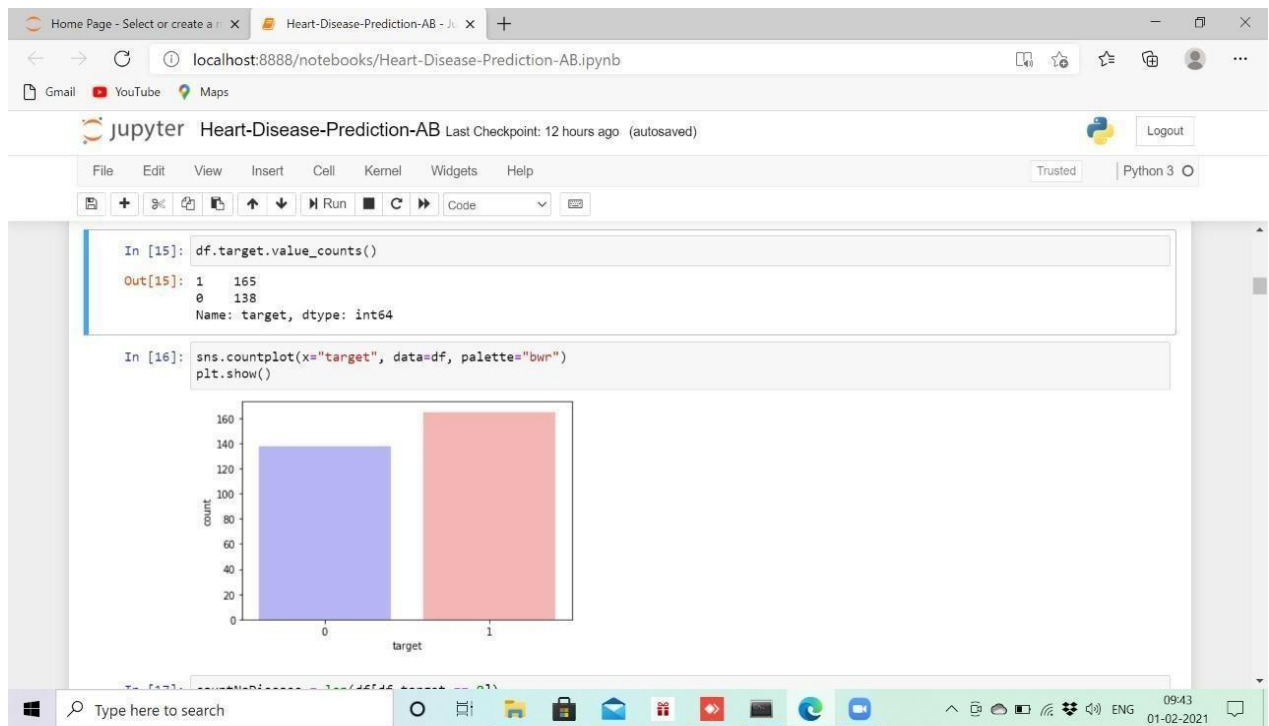


FIG 6.2: TARGET

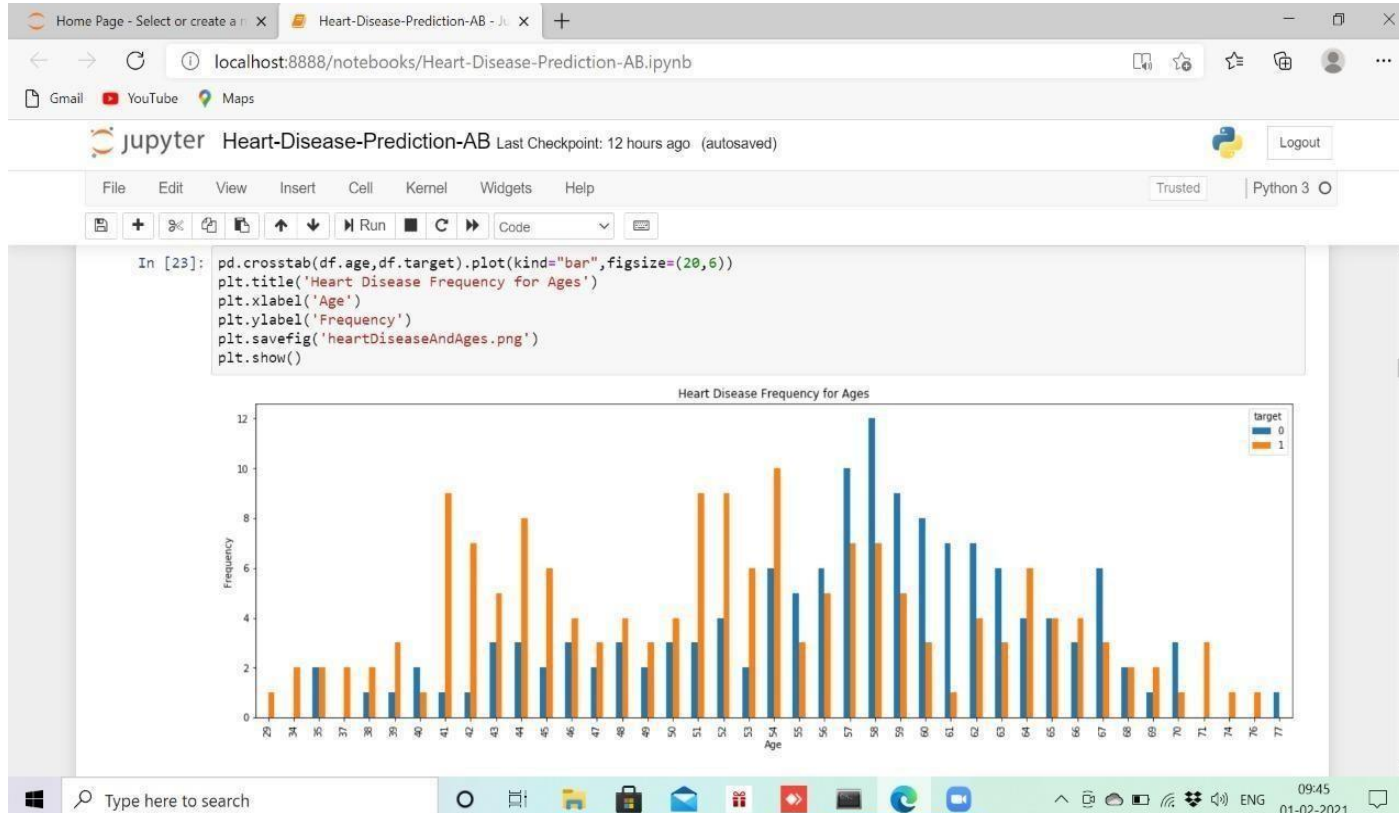


FIG 6.3: PREDICTION BASED ON THEIR AGE

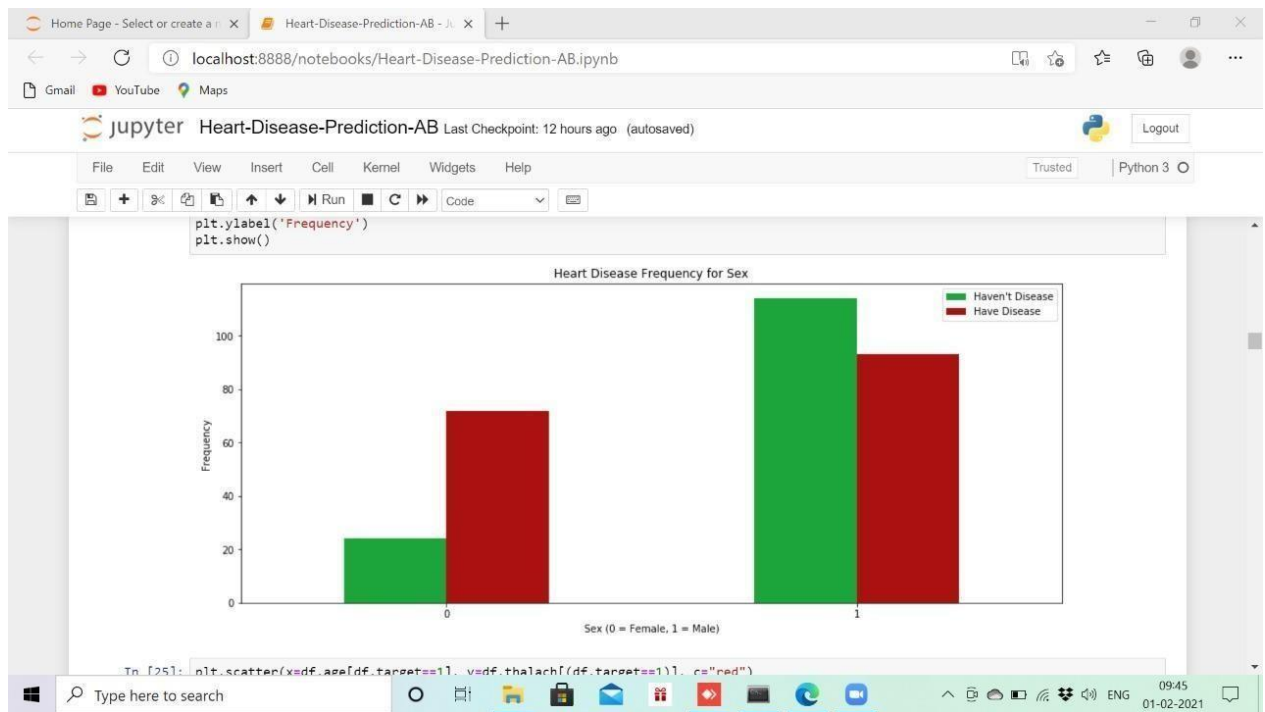


FIG 6.4: PREDICTION OF MALE AND FEMALE PATIENTS

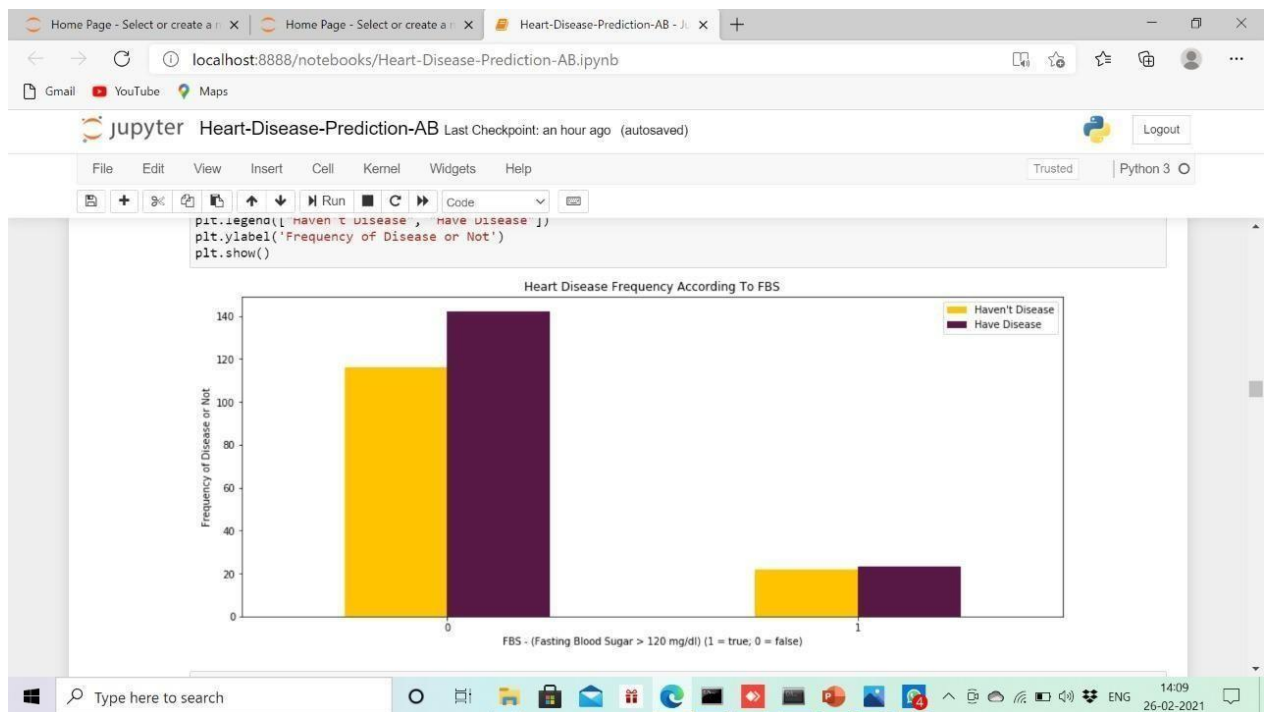


FIG 6.5: PREDICTION BASED ON FBS

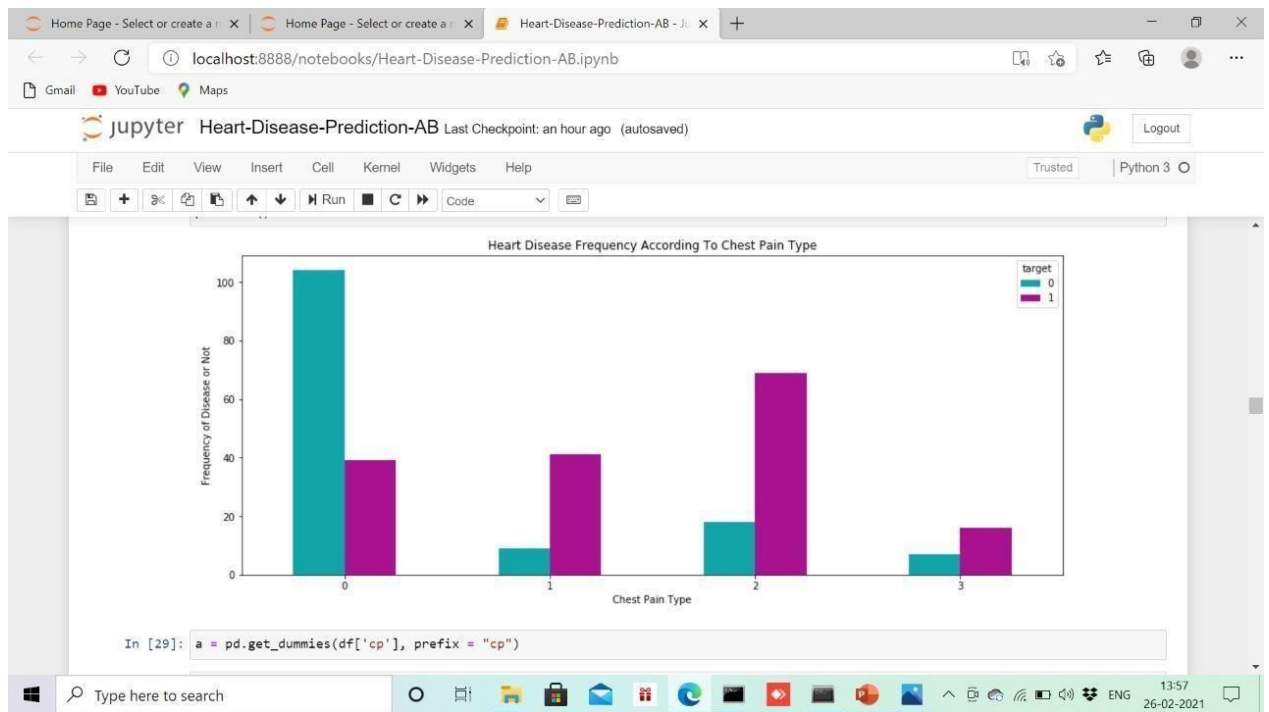


FIG 6.6: CHEST PAIN TYPE

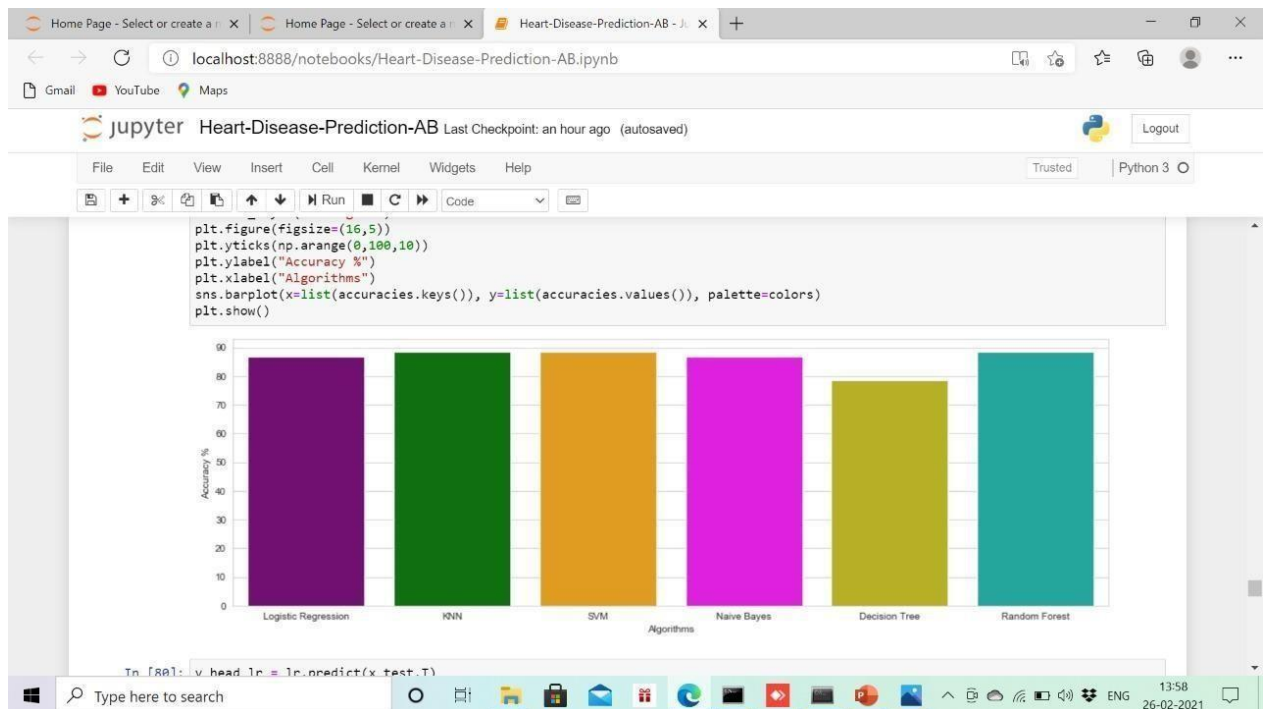


FIG 6.7: ACCURACY OF ALGORITHMS

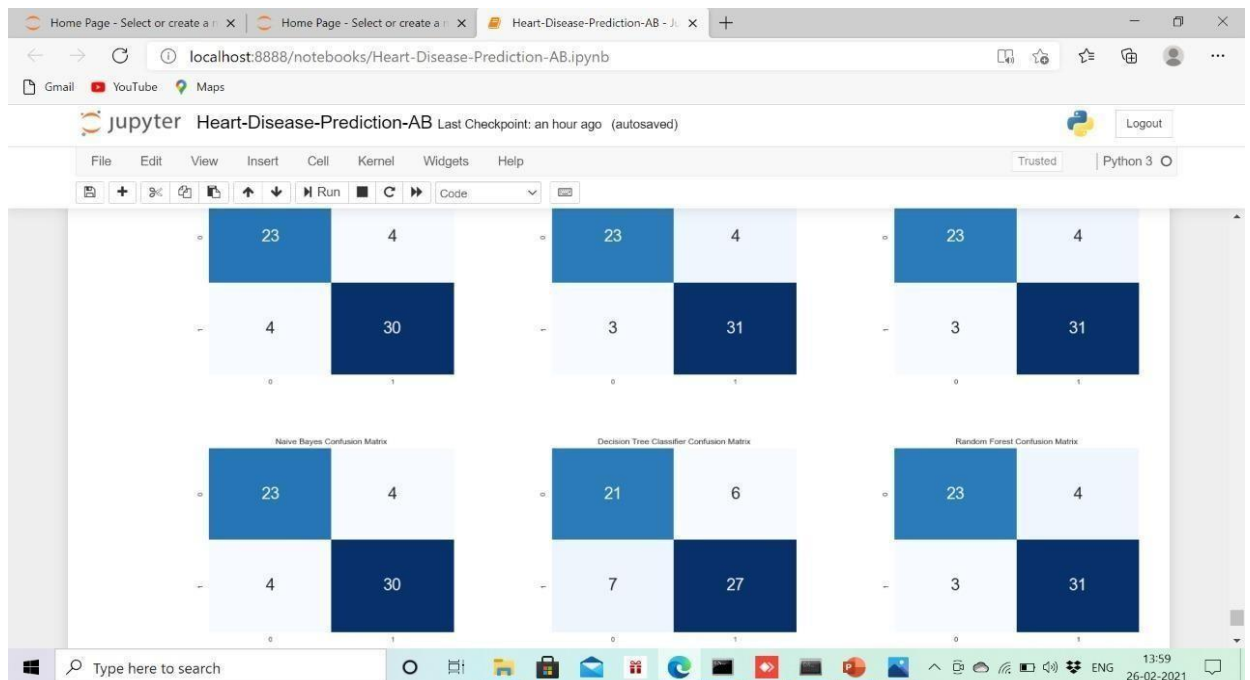
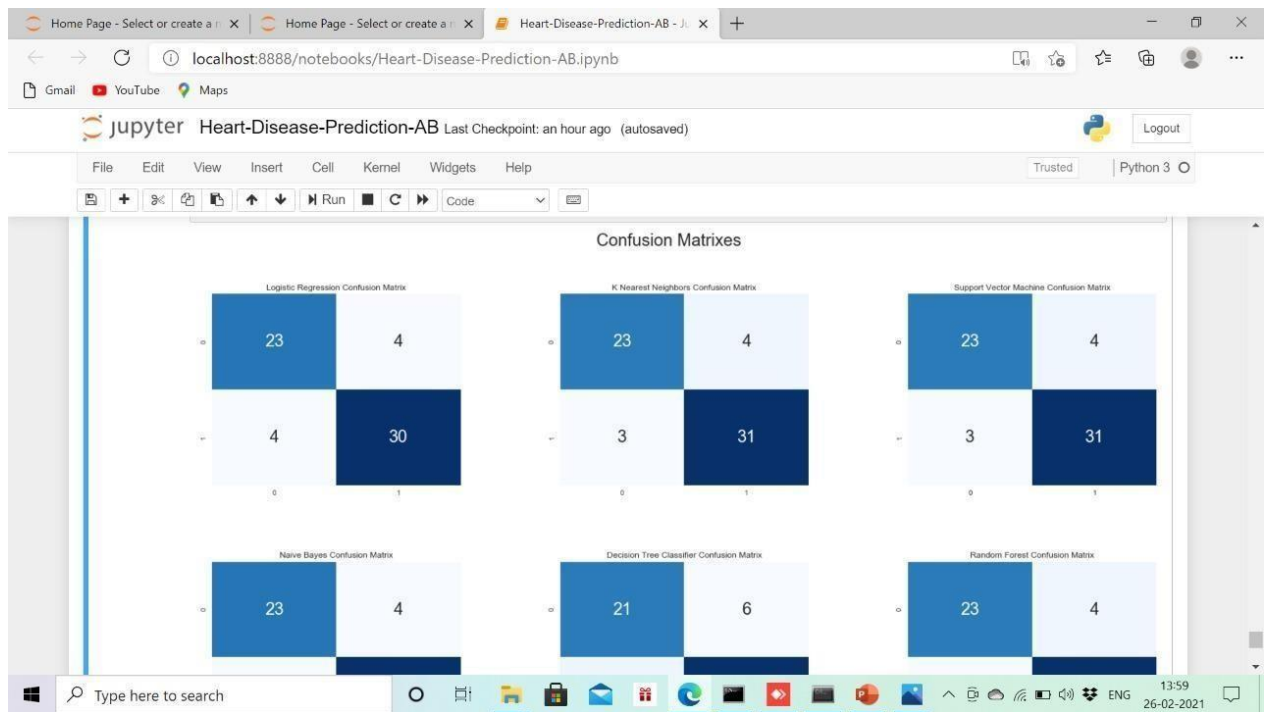


FIG 6.8 & 6.9: CONFUSION MATRIX

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

It is more often used for evaluating patients with suspected acute stroke which represents high cost and low value endeavor. We have developed an algorithm for helping clinicians determine which patients evaluated for acute stroke need a classifiers. This algorithm has superior diagnostic accuracy in comparison to the algorithm, which is currently the most accurate and widely used algorithm for determining which acute stroke patient needs a classifiers. Using readily available clinical information, our algorithm is capable of helping clinicians avoid nearly two times the number of acute stroke that can be avoided when the algorithm is utilized. Clinical implementation of this algorithm can help decrease financial costs associated with evaluations for traumatic brain injury, while decreasing patient exposure to avoidable ionizing radiation. Although the algorithm currently utilizes only easily obtainable clinical data.

7.2 FUTURE ENHANCEMENT

In future, This method can be extended to implement the real time Heart disease prediction using IOT.

REFERENCES

- [1] A. L. Bui, T. B. Horwich, and G. C. Fonarow, "Epidemiology and risk profile of heart failure," *Nature Rev. Cardiol.*, vol. 8, no. 1, p. 30, 2011.
- [2] M. Durairaj and N. Ramasamy, "A comparison of the perceptive approaches for preprocessing the data set for predicting fertility success rate," *Int. J. Control Theory Appl.*, vol. 9, no. 27, pp. 255_260, 2016.
- [3] L. A. Allen, L.W. Stevenson, K. L. Grady, N. E. Goldstein, D. D. Matlock, R. M. Arnold, N. R. Cook, G. M. Felker, G. S. Francis, P. J. Hauptman, E. P. Havranek, H. M. Krumholz, D. Mancini, B. Riegel, and J. A. Spertus, "Decision making in advanced heart failure: A scientific statement from the American heart association," *Circulation*, vol. 125, no. 15, pp. 1928_1952, 2012.
- [4] S. Ghwanmeh, A. Mohammad, and A. Al-Ibrahim, "Innovative artificial neural networks-based decision support system for heart diseases diagnosis," *J. Intell. Learn. Syst. Appl.*, vol. 5, no. 3, 2013, Art. no. 35396.
- [5] Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis," *Int. J. Comput. Sci. Issues*, vol. 8, no. 2, pp. 150_154, 2011.
- [6] J. Lopez-Sendon, "The heart failure epidemic," *Medicographia*, vol. 33, no. 4, pp. 363_369, 2011.
- [7] P. A. Heidenreich, J. G. Trogon, O. A. Khavjou, J. Butler, K. Dracup, M. D. Ezekowitz, E. A. Finkelstein, Y. Hong, S. C. Johnston, A. Khera, D. M. Lloyd-Jones, S. A. Nelson, G. Nichol, D. Orenstein, P.W. F.Wilson, and Y. J. Woo, "Forecasting the future of cardiovascular disease in the united states: A policy statement from the American heart association," *Circulation*, vol. 123, no. 8, pp. 933_944, 2011.
- [8] A. Tsanas, M. A. Little, P. E. McSharry, and L. O. Ramig, "Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average Parkinson's disease symptom severity," *J. Roy. Soc. Interface*, vol. 8, no. 59, pp. 842_855, 2011.
- [9] S. I. Ansarullah and P. Kumar, "A systematic literature review on cardiovascular disorder identification using knowledge mining and machine learning method," *Int. J. Recent Technol. Eng.*, vol. 7, no. 6S,

APPENDIX

A. PLAGIARISM REPORT

Finalpaper			
ORIGINALITY REPORT			
5%	4%	4%	1%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS
PRIMARY SOURCES			
1	export.arxiv.org Internet Source		2%
2	www.ctr.kcl.ac.uk Internet Source		1%
3	waseda.repo.nii.ac.jp Internet Source		1%
4	Michele Polese, Rittwik Jana, Velin Kounev, Ke Zhang, Supratim Deb, Michele Zorzi. "Machine Learning at the Edge: A Data-Driven Architecture with Applications to 5G Cellular Networks", IEEE Transactions on Mobile Computing, 2020 Publication		1%
5	Dimitris Chatzopoulos, Carlos Bermejo, Zhanpeng Huang, Pan Hui. "Mobile Augmented Reality Survey: From Where We Are to Where We Go", IEEE Access, 2017 Publication		<1%
Exclude quotes Off			
Exclude bibliography Off			
Exclude matches Off			

B. JOURNAL PAPER

Heart Disease Identification Method Using Machine Learning Classification in E-Healthcare

K. Sudheer Vamsi
Department of Computer Science and
Engineering
Sathyabama Institute of
Science and Technology, Chennai, India
k.sudheervamsi007@gmail.com

G. Sumanth Mallesh
Department of Computer Science and
Engineering
Sathyabama Institute of
Science and Technology, Chennai, India
sumanthguthi705@gmail.com

Dr. Lakshmanan
Department of Computer Science and
Engineering
Sathyabama Institute of
Science and Technology, Chennai, India
lakshmanan.cse@gmail.com

k.sudheervamsi007@gmail.com

Abstract –

Heart disease(HD) is the critical health issue and numerous people have been suffered by this disease around the world. The HD occurs with common symptoms of breath shortness, physical body weakness and feet are swollen. Researchers try to come across an efficient technique for the detection of heart disease, as the current diagnosis techniques of heart disease are not much effective in early time identification due to several reasons, such as accuracy and execution time. The diagnosis and treatment of heart disease is extremely difficult when modern technology and medical experts are not available. The effective diagnosis and proper treatment can save the lives of many people. Diagnosis of HD is traditionally done by the analysis of the medical history of the patient, physical examination report and analysis of concerned symptoms by a physician. But the results obtained from this diagnosis method are not accurate in identifying the patient of HD.

Keywords – Heart disease prediction, Decision tree, Machine learning, Random forest.

I. INTRODUCTION

Over the last decade, heart disease or cardiovascular remains the primary basis of death worldwide. An estimate by the World Health Organization, that over 17.9 million deaths occur every year worldwide because of cardiovascular disease, and of these deaths, 80% are because of coronary artery disease and cerebral stroke.

The vast number of deaths is common amongst low and middle income countries. Many predisposing factors such as personal and professional habits and genetic predisposition accounts for heart disease. Various habitual riskfactors such as smoking, overuse of alcohol and caffeine, stress, and physical inactivity along with the other physiological factors like obesity, hypertension, high blood cholesterol, and pre-existing heart conditions are predisposing factors for heart disease.

Machine learning is one of the most rapidly evolving domains of artificial intelligence. These algorithms can analyse huge data from various fields, one such important field is the medical field. It is a substitute to routine prediction modeling approach using a computer to gain an understanding of complex and non-linear interactions among different factors by reducing the errors in predicted and factual outcomes. Data mining is exploring huge datasets to extract hidden crucial decision-making information from a collection of a past repository for future analysis. The medical field comprises tremendous data of patients. These data need mining by various machine learning algorithms. Healthcare professionals do analysis of these data to achieve effective diagnostic decision by healthcare professionals. Medical data mining using classification algorithms provides clinical aid through analysis. It tests the classification algorithms to predict heart disease in patients.

Machine learning incorporates various classifiers of Supervised, Unsupervised and Ensemble Learning which are used to predict and find the accuracy of the given datasets.

The aim of project is On time and efficient identification of heart disease plays a keyrole in heJthcare, particutiirly in Lhe field of curdiutugy. in this nrtivle, we pmpusc•d. an effictent and act' nitc system tu diagnosis heun dj*1.0 and be system ia based on mnv'hine kitting techntques.

ii. Scope of Project :

The objectire uf Iliis project is lo check whether Iliie patient is likely tu be dipgnosed with any cardiovascular heart dimes based on their medical attributes such as gender,

this diuuxet, we prcxlick whether the pxtimi can have a heart disease or not.

The mciv c'onirbuiionx of ifiaa project are:

- c Dois Anitly sis
- Daimet Preprueesxing
- Training fire Model
- » Telling of Dataset

II. LITERATURE SURVEY

most important. Studying the literature is the appxauuses, there is c'w in uulside u' sisuice. This assisone can be found in old projc Ls, bucxksurshcs. Prior u' uulung xgan cr t the development mcthodulity, the designer x' os fa make u' intention. Dale mining, a grent develuging iculuuque rhM' revolved around expl'ing and digging out, significant information from massive collection ul' doux whivh u' can be funhcr beneficid in

examining' nnd drawing out pxuems fur making

business. related de«isiuns. Talking »ibrut the meJi d Jumain, implciticntuiiun ul' dntu mining in' thix field vtai yield bi disc'neving luid wiifi«huu'ing vuluable g»ttems and infurmution which cnn pmvc beneficial in perfumi«ng vlini«ul di«gnusis.

The research f«<uses un he«ri disease diagnosis by cunsidczi'ng previiuus data Jul inFwmation. Tu achieve ihis SBDP is buili s'iu Navtes Bayeriui in cnJer It» grcdiul the risk factors Lunccming he«ut di«1.0. The speedy. udvmement of technology h'«s kJ la rcm/u1ahle risc tn mnhile hcaJth tahnnl«tgy ttat being une uFthe web «ggii«uLiun.

The required dulo is as •mbImI in u suindurdizcd l'orm. fur predicting the chances of' he«ut disez e in a patient the folliciwtng iitLrtbutcs ure beiAg fei«nu>I t'r«um nx•Jiv4I profiles. Lhcse **inciuJc**: age, BP, cholesterol, sex, blood sugar. The collected attributes ocis cs inpui fur hie Nuives Bayesi/ui cic«sifivoiicn for predicting heart disease.

The dutesct utiir«d is spilt into twu secJirirnr, &096 dst»iset is utilized fur training «od rcsL

III. BVGTI2M fMPLENIENTA1'EON

A. Existing System:

Duk« unnllytLs i« useful l'or predivLiun from more information and it helps medical centre to preJi«t of vartuus dime. Huge amount of patient rclcxJ duio is mninkrined un rrs«nthly bwis. The strcd duLz can he useful For source of' predicting nie oc«uttence uf Future disc«ue. Some uf ihe diitu mining laid mnchtne lcamtng techniques are useJ iu predict lhc brau disease, such Ruudum Purest Decision cee, P*zz'/ Lobie, this paper .provide an insight of ilx« existing algoriilim ontt ii gives an uveratl summary of existing work.

B. Proposed System:

The hedth Lula industries ure uftcn very much interested for segregating the patients according to the health, specially the condition oF hctut. idaLhine L«nroing technique has hecumc quite heigful in this si«uction befne t«king nny decision on this particular field. The iibjccive of this.project is lu create n **voice machine model**

simple **terming** thai can

improve fire diagru«sis of heart disuse* in nghi ways, tn thte project, we are gntng lu describe cvery simple sysicm which is created bsseJ un **supervised machine learning algorithm**.

Methodology diagram is a graphical

of an architecture, including their principles, elements and components.

The project structure is as follows:

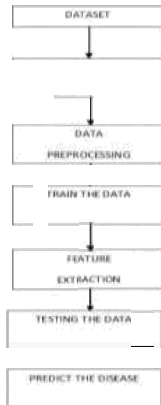


Fig 1.2 System Architecture

C. Advantages of Proposed system :

- It has both prediction and suggestion for treatment.
- Low computation power is enough.

The modules included in the project are:

first step in the data analysis process. Here, you will generate a list of the data you have and then figure out what questions you want to ask and how to answer them, such as what you can do to manipulate your available data so that you can get the answers you need. You do this by taking a broad look at the data, trends, outliers, and so on in your existing data, using visual and quantitative methods to get a sense of the story the data tells. Exploratory data analysis is a data science project.

• Data Pre-processing:

Unlabeled data is the raw data that will be filtered by removing the maximum and minimum values and performing several activities like encoding to make the data understandable format.

preprocessing step. The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their correlation

correlation with the outcome variable.

For basic guidance, you can refer to the following table for defining correlation coefficient.

Pearson's correlation: It is used as a measure for quantifying linear dependence between two continuous variables **X** and **Y**. its value varies from -1 to +1.

LOA: Linear discriminant analysis is a method of classifying features that characterizes or separates two or more classes of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA

continuous dependent feature. It provides a statistical test of whether the means of several groups are equal

that has never been used for training. This implies that we use the model against the data it has not seen. This is the reason for the overfitting of

IV. RESULTS AND DISCUSSIONS

From these results we can see that although Ernst et al. [1] used different algorithms such as Logistic Regression, Decision tree of the detection of patients diagnosed with heart disease, KNN. Random Forest. We found that the models we used are more accurate. Not only money. Moreover, the minimum error rate obtained by KNN and Logistic Regression are equal to 88.5% which is greater or almost equal to the results obtained from previous researches.

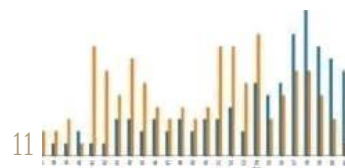


FIG 1.1 shows the risk of Heart Attack on the basis of Age

Figure 1.2 shows the accuracy of the algorithms. The x-axis represents the algorithm (KNN, Logistic Regression, Decision Tree, Random Forest) and the y-axis represents the accuracy (0-100%). The chart shows that KNN and Random Forest have the highest accuracy, followed by Decision Tree and Logistic Regression.

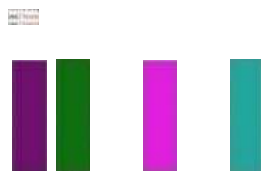


FIG 1.2 Accuracy of algorithms

FIG 1.3 confusion matrix



V. CONCLUSION

It is more often used for evaluating patients with suspected acute stroke which represents high cost and low variability. We have developed an algorithm for helping clinician's decision which helps to reduce the error rate and improve the accuracy of diagnosis. The algorithm, which is currently the most accurate and widely used algorithm and determining which accurate stroke patient needs

VI. REFERENCES

- [1] A. L. Bui, T. B. Horwath, U. N. N. Fonarow, "Epidemiology and risk profile of heart failure", *Nature Reviews Cardiology*, vol.8, no. 1, p. 30, 2011.
- [2] M. Duzairaj and N. Ramasazay, "A comparison of the performance approaches for preprocessing of data", *Int. J. Content Theory Appl.*, vol. 9, no. 27, pp. 255-260, 2016.
- [3] S. Ghawane, F. Muhammad, and A. Albal, "Innovative artificial neural network-based decision support system for heart disease diagnosis", *J. Intell. Learn. Syst. Appl.*, vol.5, no. 3, 2013.
- [4] Q. K. Al-Shayea, "Artificial neural networks in medical diagnosis", *Int. J. Comput. Sci. Issues*, vol. 8, no. 2, pp. 150-154, 2011.

C.SOURCE CODE

```
"import numpy as np"
]
},
{
  "cell_type": "code",
  "execution_count": 2,
  "metadata": {},
  "outputs": [],
  "source": [
    "import pandas as pd"
  ]
},
{
  "cell_type": "code",
  "execution_count": 3,
  "metadata": {},
  "outputs": [],
  "source": [
    "import matplotlib.pyplot as plt"
  ]
},
{
  "cell_type": "code",
  "execution_count": 4,
  "metadata": {},
  "outputs": [],
  "source": [
    "import seaborn as sns"
  ]
},
{
  "cell_type": "code",
```

```
"execution_count": 5,
"metadata": {},
"outputs": [],
"source": [
    "from sklearn.linear_model import LogisticRegression"
]
},
{
    "cell_type": "code",
    "execution_count": 6,
    "metadata": {},
    "outputs": [],
    "source": [
        "from sklearn.model_selection import train_test_split"
    ]
},
{
    "cell_type": "code",
    "execution_count": 7,
    "metadata": {},
    "outputs": [],
    "source": [
        "import os"
    ]
},
{
    "cell_type": "code",
    "execution_count": 8,
    "metadata": {},
    "outputs": [],
    "source": [
        "from sklearn.neighbors import KNeighborsClassifier"
    ]
},
```

```
{
  "cell_type": "code",
  "execution_count": 9,
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.metrics import confusion_matrix"
  ]
},
{
  "cell_type": "code",
  "execution_count": 10,
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.tree import DecisionTreeClassifier"
  ]
},
{
  "cell_type": "code",
  "execution_count": 11,
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.naive_bayes import GaussianNB"
  ]
},
{
  "cell_type": "code",
  "execution_count": 12,
  "metadata": {},
  "outputs": [],
  "source": [
    "from sklearn.ensemble import RandomForestClassifier"
```

```

]
},
{
  "cell_type": "code",
  "execution_count": 13,
  "metadata": {
    "_uuid": "8d0ffd4e0549f6f2eb24c7cc258256f2fb63ec48"
  },
  "outputs": [],
  "source": [
    "df = pd.read_csv(\"heart.csv\")"
  ]
},
{
  "cell_type": "code",
  "execution_count": 14,
  "metadata": {
    "_uuid": "4dee2636361981b0f92a8a585d49f581283d6222"
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",
          "    vertical-align: middle;\n",
          "  }\n",
          "\n",
          "  .dataframe tbody tr th {\n",
          "    vertical-align: top;\n",
          "  }\n",
          "\n",
          "  .dataframe thead th {\n",

```

```

"    text-align: right;\n", "
    }\n",
"</style>\n",
"<table border=\"1\" class=\"dataframe\">\n",
"  <thead>\n",
"    <tr style=\"text-align: right;\">\n",

},
"execution_count": 14,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
  "df.head(20)"
],
},
{
  "cell_type": "code",
  "execution_count": 15,
  "metadata": {
    "_uuid": "00af1961381609399dd80ca7c6a950277c120357"
  },
  "outputs": [
    {
      "data": {
        "text/plain": [
          "1    165\n",
          "0    138\n",
          "Name: target, dtype: int64"
        ]
      },
    },
    "execution_count": 15,
    "metadata": {},

```

```

    "output_type": "execute_result"
  }
],
"source": [
  "df.target.value_counts()"
]
},
{
  "cell_type": "code",
  "execution_count": 16,
  "metadata": {
    "_uuid": "f0c639a82d794e41978199f7fa4405a01a7b805e"
  },
  "outputs": [
    {
      "data": {
        "image/png":
          "text/plain": [
            "<Figure size 432x288 with 1 Axes>"
          ]
      },
      "metadata": {
        "needs_background": "light"
      },
      "output_type": "display_data"
    }
  ],
  "source": [
    "sns.countplot(x=\"target\", data=df, palette=\"bwr\")\n",
    "plt.show()"
  ]
},
{
  "cell_type": "code",

```

```

"execution_count": 17,
"metadata": {
  "_uuid": "dd20ffc31d87de07a6f2941b0500cda8c0b1b8e6"
},
"outputs": [],
"source": [
  "countNoDisease = len(df[df.target == 0])\n",
  "countHaveDisease = len(df[df.target == 1])"
]
},
{
  "cell_type": "code",
  "execution_count": 18,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Percentage of Patients Haven't Heart Disease: 45.54%\n",
        "Percentage of Patients Have Heart Disease: 54.46%\n"
      ]
    }
  ],
  "source": [
    "print(\"Percentage of Patients Haven't Heart Disease: {:.2f}%\".format((countNoDisease\n/ (len(df.target))*100)))\n",
    "print(\"Percentage of Patients Have Heart Disease: {:.2f}%\".format((countHaveDisease\n/ (len(df.target))*100)))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 19,

```

```

"metadata": {
  "_uuid": "aca4074785c40bcf036a59c14c78e3a503904724"
},
"outputs": [
  {
    "data": {
      "image/png":
      "text/plain": [
        "<Figure size 432x288 with 1 Axes>"
      ]
    },
    "metadata": {
      "needs_background": "light"
    },
    "output_type": "display_data"
  }
],
"source": [
  "sns.countplot(x='sex', data=df, palette=\"mako_r\")\n",
  "plt.xlabel(\"Sex (0 = female, 1 = male)\")\n",
  "plt.show()"
]
},
{
  "cell_type": "code",
  "execution_count": 20,
  "metadata": {
    "_uuid": "8780860cb02768a3bf112a41ab67f1f6bebe9146"
  },
  "outputs": [],
  "source": [
    "countFemale = len(df[df.sex == 0])\n",
    "countMale = len(df[df.sex == 1])"
  ]
}

```

```

},
{
  "cell_type": "code",
  "execution_count": 21,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Percentage of Female Patients: 31.68%\n",
        "Percentage of Male Patients: 68.32%\n"
      ]
    }
  ],
  "source": [
    "print(\"Percentage of Female Patients: {:.2f}%\".format((countFemale /
(len(df.sex))*100)))\n",
    "print(\"Percentage of Male Patients: {:.2f}%\".format((countMale / (len(df.sex))*100)))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 22,
  "metadata": {
    "_uuid": "435f8691396c6f2b1a901809286eb64112126582"
  },
  "outputs": [
    {
      "data": {
        "text/html": [
          "<div>\n",
          "<style scoped>\n",
          "  .dataframe tbody tr th:only-of-type {\n",

```

```

"    vertical-align: middle;\n", "
    }\n",
"\n",
"    .dataframe tbody tr th {\n",
"    vertical-align: top;\n",
"    }\n",
"\n",
"    .dataframe thead th {\n",
"    text-align: right;\n",
"    }\n",
"</style>\n",
"<table border='1' class='dataframe'>\n",

```

0.363636 2.121212 "

```

]
},
"execution_count": 22,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
    "df.groupby('target').mean()"
]
},
{
    "cell_type": "code",
    "execution_count": 23,
    "metadata": {
        "_uuid": "e0eed947b5490e93a43515ad6d2a6af9713f16ce"
    },
    "outputs": [
        {
            "data": {
                "image/png":

```

```
"text/plain": [  
  "<Figure size 1440x432 with 1 Axes>"  
]  
,  
"metadata": {  
  "needs_background": "light"  
},  
"output_type": "display_data"  
}  
],  
"source": [  
  "pd.crosstab(df.age,df.target).plot(kind='bar',figsize=(20,6))\n",  
  "plt.title('Heart Disease Frequency for Ages')\n",  
  "plt.xlabel('Age')\n",  
  "plt.ylabel('Frequency')\n",  
  "plt.savefig('heartDiseaseAndAges.png')\n",  
  "plt.show()"  
]  
},  
{  
  "cell_type": "code",  
  "execution_count": 24,  
  "metadata": {  
    "_uuid": "1cb0e9e4ec00ca7caa2e77a03dd0289e1c549708"  
  },  
  "outputs": [  
    {  
      "data": {  
        "image/png":  
        "text/plain": [  
          "<Figure size 1080x432 with 1 Axes>"  
        ]  
      },  
      "metadata": {
```

```

    "needs_background": "light"
  },
  "output_type": "display_data"
}
],
"source": [
  "pd.crosstab(df.sex,df.target).plot(kind='bar',figsize=(15,6),color=['#1CA53B','#AA1111'
  ])\n",
  "plt.title('Heart Disease Frequency for Sex')\n",
  "plt.xlabel('Sex (0 = Female, 1 = Male')\n",
  "plt.xticks(rotation=0)\n",
  "plt.legend(['Haven't Disease', 'Have Disease'])\n",
  "plt.ylabel('Frequency')\n",
  "plt.show()"
]
},
{
  "cell_type": "code",
  "execution_count": 25,
  "metadata": {
    "_uuid": "e5fe0744106dc26475e2e0274081789518ad8a51"
  },
  "outputs": [
    {
      "data": {
        "image/png":
        "text/plain": [
          "<Figure size 432x288 with 1 Axes>"
        ]
      },
      "metadata": {
        "needs_background": "light"
      },
      "output_type": "display_data"
    }
  ]
}

```

```

    }
],
"source": [
    "plt.scatter(x=df.age[df.target==1], y=df.thalach[(df.target==1)], c=\"red\")\n",
    "plt.scatter(x=df.age[df.target==0], y=df.thalach[(df.target==0)])\n",
    "plt.legend([\"Disease\", \"Not Disease\"])\n",
    "plt.xlabel(\"Age\")\n",
    "plt.ylabel(\"Maximum Heart Rate\")\n",
    "plt.show()"
]
},

```

```

{
    "cell_type": "code",
    "execution_count": 26,
    "metadata": {
        "_uuid": "730062aa471d6774f0b761ad2cb6a7043d339615"
    },

```

```

    "outputs": [
        {
            "data": {
                "image/png":
                "text/plain": [
                    "<Figure size 1080x432 with 1 Axes>"
                ]
            },

```

```

        },
        "metadata": {
            "needs_background": "light"
        },
        "output_type": "display_data"
    }

```

```

],
"source": [

```

```

"pd.crosstab(df.slope,df.target).plot(kind=\"bar\",figsize=(15,6),color=['#DAF7A6','#FF5733']

```

```

])\n",
    "plt.title('Heart Disease Frequency for Slope')\n",
    "plt.xlabel('The Slope of The Peak Exercise ST Segment ')\n",
    "plt.xticks(rotation = 0)\n",
    "plt.ylabel('Frequency')\n",
    "plt.show()"
]
},
{
    "cell_type": "code",
    "execution_count": 27,
    "metadata": {
        "_uuid": "3209c3a2fe67592b6ff6851327e98ac7606e14eb"
    },
    "outputs": [
        {
            "data": {
                "image/png":
                "text/plain": [
                    "<Figure size 1080x432 with 1 Axes>"
                ]
            },
            "metadata": {
                "needs_background": "light"
            },
            "output_type": "display_data"
        }
    ],
    "source": [
        "pd.crosstab(df.fbs,df.target).plot(kind='bar',figsize=(15,6),color=['#FFC300','#581845'
])\n",
    "plt.title('Heart Disease Frequency According To FBS')\n",
    "plt.xlabel('FBS - (Fasting Blood Sugar > 120 mg/dl) (1 = true; 0 = false)')\n",
    "plt.xticks(rotation = 0)\n",

```

```

"plt.legend(['Haven't Disease', 'Have Disease'])\n",
"plt.ylabel('Frequency of Disease or Not')\n",
"plt.show()"
]
},
{
"cell_type": "code",
"execution_count": 28,
"metadata": {
"_uuid": "10be6f01d184480eca7437799c214b4847fd3543"
},
"outputs": [
{
"data": {
"image/png":
"text/plain": [
"<Figure size 1080x432 with 1 Axes>"
]
},
"metadata": {
"needs_background": "light"
},
"output_type": "display_data"
}
],
"source": [
"pd.crosstab(df.cp,df.target).plot(kind='bar',figsize=(15,6),color=['#11A5AA','#AA1190'
])\n",
"plt.title('Heart Disease Frequency According To Chest Pain Type')\n",
"plt.xlabel('Chest Pain Type')\n",
"plt.xticks(rotation = 0)\n",
"plt.ylabel('Frequency of Disease or Not')\n",
"plt.show()"
]

```

```
},
{
  "cell_type": "code",
  "execution_count": 29,
  "metadata": {
    "_uuid": "c2fb6259a757a62926b86aa0eea6f663a8ec094f"
  },
  "outputs": [],
  "source": [
    "a = pd.get_dummies(df['cp'], prefix = \"cp\")"
  ]
},
```

```
{
  "cell_type": "code",
  "execution_count": 30,
  "metadata": {},
  "outputs": [],
  "source": [
    "b = pd.get_dummies(df['thal'], prefix = \"thal\")"
  ]
},
```

```
{
  "cell_type": "code",
  "execution_count": 31,
  "metadata": {},
  "outputs": [],
  "source": [
    "c = pd.get_dummies(df['slope'], prefix = \"slope\")"
  ]
},
```

```
{
  "cell_type": "code",
  "execution_count": 32,
  "metadata": {},
```

```

"outputs": [],
"source": [
    "frames = [df, a, b, c]"
]
},
{
    "cell_type": "code",
    "execution_count": 33,
    "metadata": {
        "_uuid": "50380d91358131ae6b852c9666e4da5c2e4dee66"
    },
    "outputs": [
        {
            "data": {
                "text/html": [
                    "<div>\n",
                    "<style scoped>\n",
                    "    .dataframe tbody tr th:only-of-type {\n",
                    "        vertical-align: middle;\n",
                    "    }\n",
                    "\n",
                    "    .dataframe tbody tr th {\n",
                    "        vertical-align: top;\n",
                    "    }\n",
                    "\n",
                    "    .dataframe thead th {\n",
                    "        text-align: right;\n",
                    "    }\n",
                    "</style>\n",
                    "<table border='1' class='dataframe'>\n",
                    "\n",
                    "df = df.drop(columns = ['cp', 'thal', 'slope'])\n",
                    "df.head()"
                ]
            }
        ]
    }
]

```

```

},
{
  "cell_type": "code",
  "execution_count": 35,
  "metadata": {
    "_uuid": "2e4577118ef837e24c955ea81562d8fb51a22b1c"
  },
  "outputs": [],
  "source": [
    "y = df.target.values\n",
    "x_data = df.drop(['target'], axis = 1)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 36,
  "metadata": {
    "_uuid": "c60916b235e1962a53f95593b8d020c0acc079ed"
  },
  "outputs": [],
  "source": [
    "x = (x_data - np.min(x_data)) / (np.max(x_data) - np.min(x_data)).values"
  ]
},
{
  "cell_type": "code",
  "execution_count": 37,
  "metadata": {
    "_uuid": "3090d1ba1e8776706d23a1175db6d1451908a954"
  },
  "outputs": [],
  "source": [
    "x_train, x_test, y_train, y_test = train_test_split(x,y,test_size = 0.2,random_state=0)"
  ]
}

```

```
},
{
  "cell_type": "code",
  "execution_count": 38,
  "metadata": {
    "_uuid": "880e2db253fd1125bfa065ba0ae8840b68687e42"
  },
  "outputs": [],
  "source": [
    "x_train = x_train.T"
  ]
},
{
  "cell_type": "code",
  "execution_count": 39,
  "metadata": {},
  "outputs": [],
  "source": [
    "y_train = y_train.T"
  ]
},
{
  "cell_type": "code",
  "execution_count": 40,
  "metadata": {},
  "outputs": [],
  "source": [
    "x_test = x_test.T"
  ]
},
{
  "cell_type": "code",
  "execution_count": 41,
  "metadata": {},
```

```
"outputs": [],
"source": [
    "y_test = y_test.T"
]
},
{
    "cell_type": "code",
    "execution_count": 42,
    "metadata": {
        "_uuid": "e98deb17b979b67f57be48517cdcf631cd5363bc"
    },
    "outputs": [],
    "source": [
        "def initialize(dimension):\n",
        "    weight = np.full((dimension,1),0.01)\n",
        "    bias = 0.0\n",
        "    return weight,bias"
    ]
},
{
    "cell_type": "code",
    "execution_count": 43,
    "metadata": {
        "_uuid": "116fc66d415563ca82b5400b7852cce9dad004ba"
    },
    "outputs": [],
    "source": [
        "def sigmoid(z):\n",
        "    y_head = 1/(1+ np.exp(-z))\n",
        "    return y_head"
    ]
},
{
    "cell_type": "code",
```

```

"execution_count": 44,
"metadata": {
  "_uuid": "5998fbc58a696070458297830e30d4e4b036396d"
},
{
  "cell_type": "code",
  "execution_count": 45,
  "metadata": {
    "_uuid": "0241c5bec516229c2832728e27782c273ca35b57"
  },
  "outputs": [],
  "source": [
    "def update(weight,bias,x_train,y_train,learningRate,iteration) :\n",
    "    costList = []\n",
    "    index = []\n",
    "    for i in range(iteration):\n",
    "        cost,gradients = forwardBackward(weight,bias,x_train,y_train)\n",
    "        weight = weight - learningRate * gradients[\"Derivative Weight\"]\n",
    "        bias = bias - learningRate * gradients[\"Derivative Bias\"]\n",
    "        \n",
    "        costList.append(cost)\n",
    "        index.append(i)\n",
    "\n",
    "    parameters = {\"weight\": weight,\"bias\": bias}\n",
    "    \n",
    "    print(\"iteration:\",iteration)\n",
    "    print(\"cost:\",cost)\n",
    "\n",
    "    plt.plot(index,costList)\n",
    "    plt.xlabel(\"Number of Iteration\")\n",
    "    plt.ylabel(\"Cost\")\n",
    "    plt.show()\n",
    "\n",
    "    return parameters, gradients"
  ]

```

```

]
},
{
  "cell_type": "code",
  "execution_count": 46,
  "metadata": {
    "_uuid": "a97c3ff651d88b280eb87ec076040f0becaf1f3e"
  },
  "outputs": [],
  "source": [
    "def predict(weight,bias,x_test):\n",
    "    z = np.dot(weight.T,x_test) + bias\n",
    "    y_head = sigmoid(z)\n",
    "\n",
    "    y_prediction = np.zeros((1,x_test.shape[1]))\n",
    "    \n",
    "    for i in range(y_head.shape[1]):\n",
    "        if y_head[0,i] <= 0.5:\n",
    "            y_prediction[0,i] = 0\n",
    "        else:\n",
    "            y_prediction[0,i] = 1\n",
    "    return y_prediction"
  ]
},
{
  "cell_type": "code",
  "execution_count": 47,
  "metadata": {
    "_uuid": "1b3ae665060ba757d8e0f5f29458fadf4be144ff"
  },
  "outputs": [],
  "source": [
    "def logistic_regression(x_train,y_train,x_test,y_test,learningRate,iteration):\n",
    "    dimension = x_train.shape[0]\n",

```

```

" weight,bias = initialize(dimension)\n",
"  \n",
"  parameters, gradients = update(weight,bias,x_train,y_train,learningRate,iteration)\n",
"  \n",
"  y_prediction = predict(parameters["weight"],parameters["bias"],x_test)\n",
"  \n",
"  print(\"Manuel Test Accuracy: {:.2f}%\".format((100 - np.mean(np.abs(y_prediction -
y_test))*100)))"
]
},
{
"cell_type": "code",
"execution_count": 48,
"metadata": {
"_uuid": "d866efadafcc3b49a90016c00da6c4d981cd8837"
},
"outputs": [
{
"name": "stdout",
"output_type": "stream",
"text": [
"iteration: 100\n",
"cost: 0.3597736123664534\n"
]
},
{
"data": {
"image/png":
"text/plain": [
"<Figure size 432x288 with 1 Axes>"
]
},
"metadata": {
"needs_background": "light"
}
}

```

```
,
  "output_type": "display_data"
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "Manuel Test Accuracy: 86.89%\n"
  ]
},
],
"source": [
  "logistic_regression(x_train,y_train,x_test,y_test,1,100)"
],
},
{
  "cell_type": "code",
  "execution_count": 49,
  "metadata": {
    "_uuid": "5592396ddebe9346767a2a34dd20abf300e5c730"
  },
  "outputs": [],
  "source": [
    "accuracies = {}"
  ],
},
{
  "cell_type": "code",
  "execution_count": 50,
  "metadata": {},
  "outputs": [],
  "source": [
    "lr = LogisticRegression()"
  ]
}
```

```

},
{
  "cell_type": "code",
  "execution_count": 51,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,\n",
          "                  intercept_scaling=1, l1_ratio=None, max_iter=100,\n",
          "                  multi_class='auto', n_jobs=None, penalty='l2',\n",
          "                  random_state=None, solver='lbfgs', tol=0.0001, verbose=0,\n",
          "                  warm_start=False)"
        ]
      },
    ],
    "execution_count": 51,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "lr.fit(x_train.T,y_train.T)"
]
},
{
  "cell_type": "code",
  "execution_count": 52,
  "metadata": {},
  "outputs": [],
  "source": [
    "acc = lr.score(x_test.T,y_test.T)*100"
  ]
},

```

```

{
  "cell_type": "code",
  "execution_count": 53,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Test Accuracy 86.89%\n"
      ]
    }
  ],
  "source": [
    "accuracies['Logistic Regression'] = acc\n",
    "print(\"Test Accuracy {:.2f}%\".format(acc))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 54,
  "metadata": {},
  "outputs": [],
  "source": [
    "knn = KNeighborsClassifier(n_neighbors = 2)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 55,
  "metadata": {},
  "outputs": [
    {
      "data": {

```

```

"text/plain": [
  "KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',\n",
  "                      metric_params=None, n_jobs=None, n_neighbors=2, p=2,\n",
  "                      weights='uniform')"\n",
  ]
},
"execution_count": 55,
"metadata": {},
"output_type": "execute_result"
}
],
"source": [
  "knn.fit(x_train.T, y_train.T)"
]
},
{
  "cell_type": "code",
  "execution_count": 56,
  "metadata": {},
  "outputs": [],
  "source": [
    "prediction = knn.predict(x_test.T)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 57,
  "metadata": {
    "_uuid": "b5d2956d9f265b47645ec2a6cd2eb7d50af5783a"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",

```

```

    "text": [
        "2 NN Score: 77.05%\n"
    ]
}
],
"source": [
    "print('{} NN Score: {:.2f}%\n'.format(2, knn.score(x_test.T, y_test.T)*100))"
]
},
{
    "cell_type": "code",
    "execution_count": 58,
    "metadata": {},
    "outputs": [],
    "source": [
        "scoreList = []"
    ]
},
{
    "cell_type": "code",
    "execution_count": 59,
    "metadata": {},
    "outputs": [],
    "source": [
        "for i in range(1,20):\n",
        "    knn2 = KNeighborsClassifier(n_neighbors = i) \n",
        "    knn2.fit(x_train.T, y_train.T)\n",
        "    scoreList.append(knn2.score(x_test.T, y_test.T))"
    ]
},
{
    "cell_type": "code",
    "execution_count": 60,
    "metadata": {

```

```
"_uuid": "8db6f0c13aa1462624cf0da04a03c9ee4db8a991"
},
"outputs": [
{
  "data": {
    "image/png":
    "text/plain": [
      "<Figure size 432x288 with 1 Axes>"
    ]
  },
  "metadata": {
    "needs_background": "light"
  },
  "output_type": "display_data"
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "Maximum KNN Score is 88.52%\n"
  ]
}
],
"source": [
  "plt.plot(range(1,20), scoreList)\n",
  "plt.xticks(np.arange(1,20,1))\n",
  "plt.xlabel(\"K value\")\n",
  "plt.ylabel(\"Score\")\n",
  "plt.show()\n",
  "\n",
  "acc = max(scoreList)*100\n",
  "accuracies['KNN'] = acc\n",
  "print(\"Maximum KNN Score is {:.2f}%\".format(acc))"
```

```

},
{
  "cell_type": "code",
  "execution_count": 61,
  "metadata": {
    "_uuid": "f51e85b552168f5ce56a43dd8ccb6452ee661317"
  },
  "outputs": [],
  "source": [
    "from sklearn.svm import SVC"
  ]
},
{
  "cell_type": "code",
  "execution_count": 62,
  "metadata": {},
  "outputs": [],
  "source": [
    "svm = SVC(random_state = 1)"
  ]
},
{
  "cell_type": "code",
  "execution_count": 63,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,\n",
          "    decision_function_shape='ovr', degree=3, gamma='scale', kernel='rbf',\n",
          "    max_iter=-1, probability=False, random_state=1, shrinking=True, tol=0.001,\n",
          "    verbose=False)"
        ]
      }
    ]
  }
}

```

```

    },
    "execution_count": 63,
    "metadata": {},
    "output_type": "execute_result"
  }
],
"source": [
  "svm.fit(x_train.T, y_train.T)"
]
},
{
  "cell_type": "code",
  "execution_count": 64,
  "metadata": {},
  "outputs": [],
  "source": [
    "acc = svm.score(x_test.T,y_test.T)*100"
  ]
},
{
  "cell_type": "code",
  "execution_count": 65,
  "metadata": {},
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Test Accuracy of SVM Algorithm: 88.52%\n"
      ]
    }
  ]
},
"source": [
  "accuracies['SVM'] = acc\n",

```

```

    "print(\"Test Accuracy of SVM Algorithm: {:.2f}%\".format(acc))"
]
},
{
    "cell_type": "code",
    "execution_count": 66,
    "metadata": {},
    "outputs": [],
    "source": [
        "nb = GaussianNB()"
    ]
},
{
    "cell_type": "code",
    "execution_count": 67,
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "GaussianNB(priors=None, var_smoothing=1e-09)"
                ]
            },
            "execution_count": 67,
            "metadata": {},
            "output_type": "execute_result"
        }
    ],
    "source": [
        "nb.fit(x_train.T, y_train.T)"
    ]
},
{
    "cell_type": "code",

```

```

"execution_count": 68,
"metadata": {},
"outputs": [],
"source": [
    "acc = nb.score(x_test.T,y_test.T)*100"
]
},
{
    "cell_type": "code",
    "execution_count": 69,
    "metadata": {},
    "outputs": [],
    "source": [
        "accuracies['Naive Bayes'] = acc"
    ]
},
{
    "cell_type": "code",
    "execution_count": 70,
    "metadata": {
        "_uuid": "0544f45710912dab11c597b9a743bdc712f1e826"
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "Accuracy of Naive Bayes: 86.89%\n"
            ]
        }
    ],
    "source": [
        "print(\"Accuracy of Naive Bayes: {:.2f}%\".format(acc))"
    ]
}

```

```

},
{
  "cell_type": "code",
  "execution_count": 71,
  "metadata": {},
  "outputs": [],
  "source": [
    "dtc = DecisionTreeClassifier()"
  ]
},
{
  "cell_type": "code",
  "execution_count": 72,
  "metadata": {},
  "outputs": [
    {
      "data": {
        "text/plain": [
          "DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='gini',\n",
          "                        max_depth=None, max_features=None, max_leaf_nodes=None,\n",
          "                        min_impurity_decrease=0.0, min_impurity_split=None,\n",
          "                        min_samples_leaf=1, min_samples_split=2,\n",
          "                        min_weight_fraction_leaf=0.0, presort='deprecated',\n",
          "                        random_state=None, splitter='best')"
        ]
      },
      "execution_count": 72,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "dtc.fit(x_train.T, y_train.T)"
  ]
}

```

```

},
{
  "cell_type": "code",
  "execution_count": 73,
  "metadata": {},
  "outputs": [],
  "source": [
    "acc = dtc.score(x_test.T, y_test.T)*100"
  ]
},
{
  "cell_type": "code",
  "execution_count": 74,
  "metadata": {
    "_uuid": "a3f57a094e75c7072857cb8adb40afa3fead5a26"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Decision Tree Test Accuracy 77.05%\n"
      ]
    }
  ],
  "source": [
    "accuracies['Decision Tree'] = acc\n",
    "print(\"Decision Tree Test Accuracy {:.2f}%\".format(acc))"
  ]
},
{
  "cell_type": "code",
  "execution_count": 75,
  "metadata": {},

```

```

"outputs": [],
"source": [
    "rf = RandomForestClassifier(n_estimators = 1000, random_state = 1)"
]
},
{
    "cell_type": "code",
    "execution_count": 76,
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,\n",
                    "                        criterion='gini', max_depth=None, max_features='auto',\n",
                    "                        max_leaf_nodes=None, max_samples=None,\n",
                    "                        min_impurity_decrease=0.0, min_impurity_split=None,\n",
                    "                        min_samples_leaf=1, min_samples_split=2,\n",
                    "                        min_weight_fraction_leaf=0.0, n_estimators=1000,\n",
                    "                        n_jobs=None, oob_score=False, random_state=1, verbose=0,\n",
                    "                        warm_start=False)"
                ]
            },
            "execution_count": 76,
            "metadata": {},
            "output_type": "execute_result"
        }
    ],
    "source": [
        "rf.fit(x_train.T, y_train.T)"
    ]
},
{
    "cell_type": "code",

```

```

"execution_count": 77,
"metadata": {},
"outputs": [],
"source": [
    "acc = rf.score(x_test.T,y_test.T)*100"
]
},
{
    "cell_type": "code",
    "execution_count": 78,
    "metadata": {
        "_uuid": "c81ae8e5658bea3a910901e6bc35a319bba701a9"
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "Random Forest Algorithm Accuracy Score : 88.52%\n"
            ]
        }
    ],
    "source": [
        "accuracies['Random Forest'] = acc\n",
        "print(\"Random Forest Algorithm Accuracy Score : {:.2f}%\".format(acc))"
    ]
},
{
    "cell_type": "code",
    "execution_count": 79,
    "metadata": {
        "_uuid": "9e594d64f9a20255ca8687179d858e3dce855136"
    },
    "outputs": [

```

```

{
  "data": {
    "image/png":
    "text/plain": [
      "<Figure size 1152x360 with 1 Axes>"
    ]
  },
  "metadata": {},
  "output_type": "display_data"
},
{
  "source": [
    "colors = [\"purple\", \"green\", \"orange\", \"magenta\", \"#CFC60E\", \"#0FBBAE\"]\n",
    "\n",
    "sns.set_style(\"whitegrid\")\n",
    "plt.figure(figsize=(16,5))\n",
    "plt.yticks(np.arange(0,100,10))\n",
    "plt.ylabel(\"Accuracy %\")\n",
    "plt.xlabel(\"Algorithms\")\n",
    "sns.barplot(x=list(accuracies.keys()), y=list(accuracies.values()), palette=colors)\n",
    "plt.show()"
  ],
  {
    "cell_type": "code",
    "execution_count": 80,
    "metadata": {},
    "outputs": [],
    "source": [
      "y_head_lr = lr.predict(x_test.T)"
    ],
  },
  {
    "cell_type": "code",

```

```

"execution_count": 81,
"metadata": {},
"outputs": [],
"source": [
    "knn3 = KNeighborsClassifier(n_neighbors = 3)"
]
},
{
    "cell_type": "code",
    "execution_count": 82,
    "metadata": {},
    "outputs": [
        {
            "data": {
                "text/plain": [
                    "KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',\n",
                    "                    metric_params=None, n_jobs=None, n_neighbors=3, p=2,\n",
                    "                    weights='uniform')"
                ]
            },
            "execution_count": 82,
            "metadata": {},
            "output_type": "execute_result"
        }
    ],
    "source": [
        "knn3.fit(x_train.T, y_train.T)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 83,
    "metadata": {},
    "outputs": [],

```

```
"source": [  
  "y_head_knn = knn3.predict(x_test.T)"  
]  
,  
{  
  "cell_type": "code",  
  "execution_count": 84,  
  "metadata": {},  
  "outputs": [],  
  "source": [  
    "y_head_svm = svm.predict(x_test.T)"  
  ]  
},  
{  
  "cell_type": "code",  
  "execution_count": 85,  
  "metadata": {},  
  "outputs": [],  
  "source": [  
    "y_head_nb = nb.predict(x_test.T)"  
  ]  
},  
{  
  "cell_type": "code",  
  "execution_count": 86,  
  "metadata": {},  
  "outputs": [],  
  "source": [  
    "y_head_dtc = dtc.predict(x_test.T)"  
  ]  
},  
{  
  "cell_type": "code",  
  "execution_count": 87,
```

```
"metadata": {},
"outputs": [],
"source": [
    "y_head_rf = rf.predict(x_test.T)"
]
},
{
    "cell_type": "code",
    "execution_count": 88,
    "metadata": {},
    "outputs": [],
    "source": [
        "cm_lr = confusion_matrix(y_test,y_head_lr)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 89,
    "metadata": {},
    "outputs": [],
    "source": [
        "cm_knn = confusion_matrix(y_test,y_head_knn)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 90,
    "metadata": {},
    "outputs": [],
    "source": [
        "cm_svm = confusion_matrix(y_test,y_head_svm)"
    ]
},
}
```

```
"cell_type": "code",
"execution_count": 91,
"metadata": {},
"outputs": [],
"source": [
    "cm_nb = confusion_matrix(y_test,y_head_nb)"
]
},
{
    "cell_type": "code",
    "execution_count": 92,
    "metadata": {},
    "outputs": [],
    "source": [
        "cm_dtc = confusion_matrix(y_test,y_head_dtc)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 93,
    "metadata": {},
    "outputs": [],
    "source": [
        "cm_rf = confusion_matrix(y_test,y_head_rf)"
    ]
},
{
    "cell_type": "code",
    "execution_count": 94,
    "metadata": {
        "_uuid": "b91a2a68b5cc12ea58f255731cf76d49fd2ed5ec"
    },
    "outputs": [
```

```

"data": {
  "image/png":
  "text/plain": [
    "<Figure size 1728x864 with 6 Axes>"
  ]
},
"metadata": {},
"output_type": "display_data"
},
],
"source": [
  "plt.figure(figsize=(24,12))\n",
  "\n",
  "plt.suptitle(\"Confusion Matrixes\",fontsize=24)\n",
  "plt.subplots_adjust(wspace = 0.4, hspace= 0.4)\n",
  "\n",
  "plt.subplot(2,3,1)\n",
  "plt.title(\"Logistic Regression Confusion Matrix\")\n",
  "sns.heatmap(cm_lr,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,
annot_kws={\"size\": 24})\n",
  "\n",
  "plt.subplot(2,3,2)\n",
  "plt.title(\"K Nearest Neighbors Confusion Matrix\")\n",
  "sns.heatmap(cm_knn,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,
annot_kws={\"size\": 24})\n",
  "\n",
  "plt.subplot(2,3,3)\n",
  "plt.title(\"Support Vector Machine Confusion Matrix\")\n",
  "sns.heatmap(cm_svm,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,
annot_kws={\"size\": 24})\n",
  "\n",
  "plt.subplot(2,3,4)\n",
  "plt.title(\"Naive Bayes Confusion Matrix\")\n",
  "sns.heatmap(cm_nb,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,

```

```

annot_kws={"size\": 24})\n",
    "\n",
    "plt.subplot(2,3,5)\n",
    "plt.title(\"Decision Tree Classifier Confusion Matrix\")\n",
    "sns.heatmap(cm_dtc,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,
annot_kws={"size\": 24})\n",
    "\n",
    "plt.subplot(2,3,6)\n",
    "plt.title(\"Random Forest Confusion Matrix\")\n",
    "sns.heatmap(cm_rf,annot=True,cmap=\"Blues\",fmt=\"d\",cbar=False,
annot_kws={"size\": 24})\n",
    "\n",
    "plt.show()"
]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {},
    "outputs": [],
    "source": []
}
],
"metadata": {
    "kernel_spec": {
        "display_name": "Python 3",
        "language": "python",

```

```
"name": "python3"
},
"language_info": {
  "codemirror_mode": {
    "name": "ipython",
    "version": 3
  },
  "file_extension": ".py",
  "mimetype": "text/x-python",
  "name": "python",
  "nbconvert_exporter": "python",
  "pygments_lexer": "ipython3",
  "version": "3.7.6"
}
},
"nbformat": 4,
"nbformat_minor": 1
}
```

