Machine Learning with Python

(Health Insurance cost prediction)

A Summer Internship Report Submitted in partial fulfillment of the requirement for undergraduate degree of

## BACHELOR OF TECHNOLOGY IN

**COMPUTER SCIENCE AND ENGINEERING**

## Submitted by

## Harshitha

## 222010322057

Under the guidance of

**Dr.Mula Malyadri**

Assistant Professor



**Department of Computer Science and Engineering GITAM School of Technology**

**GITAM (Deemed to be University) Hyderabad-502329**

## GANDHI INSTITUTE OF TECHNOLOGY AND MANAGEMENT

**(GITAM)**

## (Declared as Deemed-to-be-University u/s 3 of UGC Act 1956) HYDERABAD CAMPUS



**DECLARATION**

I submit this industrial training work entitled “**Heart Disease Classification** ” to GITAM (Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “**Bachelor of Technology**” in “**Computer Science and Engineering**”. I declare that it was carried out independently by me under the guidance of **Dr. Mula Malyadri**, Asst. Professor, GITAM (Deemed To Be University), Hyderabad, India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD Name: Pothula Harshitha Date: 07-09-2023 Student Roll No: 222010322057

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GITAM SCHOOL OF TECHNOLOGY

**GITAM**

## (DEEMED TO BE UNIVERSITY) HYDERABAD CAMPUS



**CERTIFICATE**

This is to certify that the Internship report entitled “**MACHINE LEARNING WITH PYTHON** ” is a bonafide record of work carried out by **POTHULA HARSHITHA(222010322057)** submitted in partial fulfillment of the requirement for the award of the degree of Bachelors of Technology in Computer Science and Engineering.

**Dr. Mula Malyadri Dr. K .S. Sudeep**

Assistant Professor Professor & HOD

Dept. of CSE

**CERTIFICATE OF COMPLETION**



**ACKNOWLEDGEMENT**

My project would not have been successful without the help of several people. We would like to thank the personalities who were part of our project in numerous ways, those who gave us outstanding support from the birth of the project.

We are extremely thankful to our honorable Pro-Vice Chancellor, **Dr. D. Shambasiva rao** for providing necessary infrastructure and resources for the accomplishment of our project.

We are highly indebted to **Prof. N. Seetharamaiah**, Principal, School of Technology, for his support during the tenure of the project.

We are very much obliged to our beloved **Dr. K.S.Sudeep** , Head of the Department of Computer Science & Engineering for providing the opportunity to undertake this project and encouragement in completion of this project.

We hereby wish to express our deep sense of gratitude to **Dr. Mula Malyadri , Assistant Professor**, Department of Computer Science and Engineering, School of Technology for the esteemed guidance, moral support and invaluable advice provided by her for the success of the project.

We are also thankful to all the staff members of the Computer Science and Engineering department who have cooperated in making our project a success. We would like to thank all our parents and friends who extended their help, encouragement and moral support either directly or indirectly in our project work.

Sincerely,

Harshitha Pothula, 222010322057.

**ABSTRACT**

This report is the reflection and also the journey of my two month internship period along with the highlights of what I learned through errors, work responsibilities, and the importance of the internship program at LearnWik Solutions Pvt Lmt. The knowledge I have gained on Machine Learning with Python, new frameworks, and also how to work in an office environment is splendid. As an intern, my work was to learn and focus on Machine learning projects. In this report, I have focused on my work territory and explain my achievements which I have got through my internship at LearnWik Solutions.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| S.NO | CONTENT | PGNO |
| 1 | **MACHINE LEARNING**   * 1. Introduction   2. Importance of Machine Learning   3. Uses of Machine Learning   4. Types of Learning Algorithms      1. Supervised Learning      2. Unsupervised Learning      3. Semi Supervised Learning | 01 |
| 2 | PYTHON  * 1. Introduction To Python      1. History Of Python   2. How to Setup Python      1. Installation (using python IDLE)      2. Installation (using Anaconda)   3. Features Of Python   4. Python Variable Types      1. Python Numbers      2. Python Strings      3. Python Lists      4. Python Tuples      5. Python Dictionary   5. Python Functions      1. Defining a Function      2. Calling a Function   6. Python Using OOPs Concepts      1. Class      2. \_\_\_init\_\_\_Method in Class | 06 |
| 3 | HEART DISEASE CLASSIFICATION  * 1. Project Requirements      1. Packages used      2. Algorithms used   2. Problem Statement   3. Dataset Description   4. Objective of the Case Study | 14 |
| 4 | DATA PREPROCESSING/FEATURE ENGINEERING AND EDA  * 1. Loading The Data   2. Statistical Analysis   3. Handling Missing Values | 17 |
| 5 | MODEL BUILDING AND EVALUATION 5.1. Brief About the Algorithms Used and Building Model  5.2. Train the Models   * 1. Validate the Models   2. Make Predictions | 21 |
| 6 | **CONCLUSION** | 32 |
| 7 | **REFERENCES** | 33 |

**CHAPTER 1: MACHINE LEARNING**

### INTRODUCTION:

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

### IMPORTANCE OF MACHINE LEARNING:

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide

variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps

analyse those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques. The process flow depicted here represents how machine learning works.

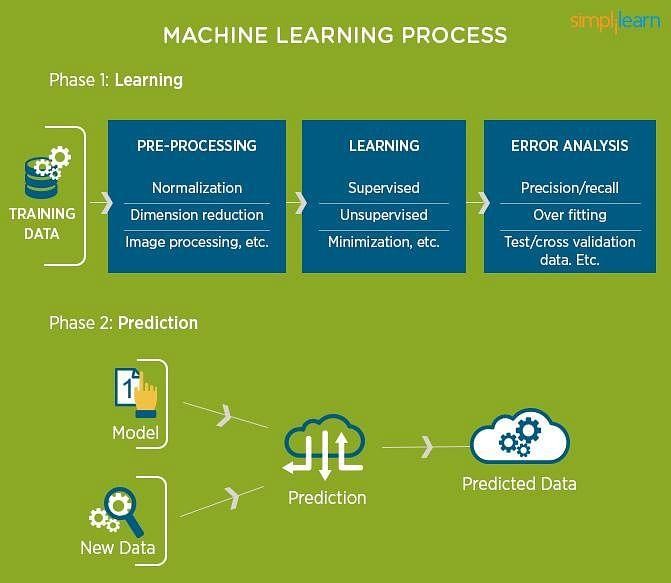


Fig 1.2.1: The Process Flow

### USES OF MACHINE LEARNING:

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let’s consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data.

Traditionally, data analysis was always characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

### TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

### Supervised Learning:

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labeled training dataset – that is, a dataset that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

### Unsupervised Learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

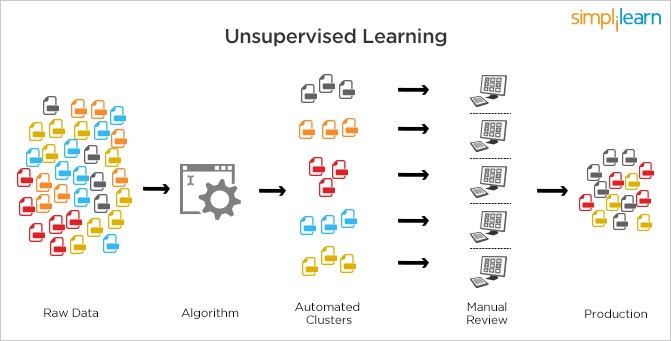


Fig 1.4.2.1: Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

### Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabelled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabelled data.

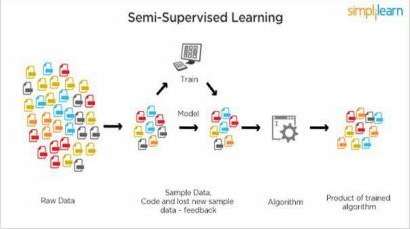


Fig 1.4.3.1: Semi Supervised Learning

# CHAPTER 2: PYTHON

Basic programming language used for machine learning is: PYTHON

### INTRODUCTION TO PYTHON:

* Python is a high-level, interpreted, interactive and object-oriented scripting language.
* Python is a general-purpose programming language that is often applied in scripting roles
* Python is Interpreted: Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP. ∙ Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
* Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

### History of python:

Python was developed by GUIDO VAN ROSSUM in early 1990’s ∙ Its latest version is 3.7.3, it is generally called as python3

### HOW TO SETUP PYTHON:

* Python is available on a wide variety of platforms including Linux and Mac OS X. Let’s understand how to set up our Python environment.
* The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

### Installation (using python IDLE):

* Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.
* Download python from [www.python.org](http://www.python.org/)
* When the download is completed, double click the file and follow the instructions to install it.
* When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.

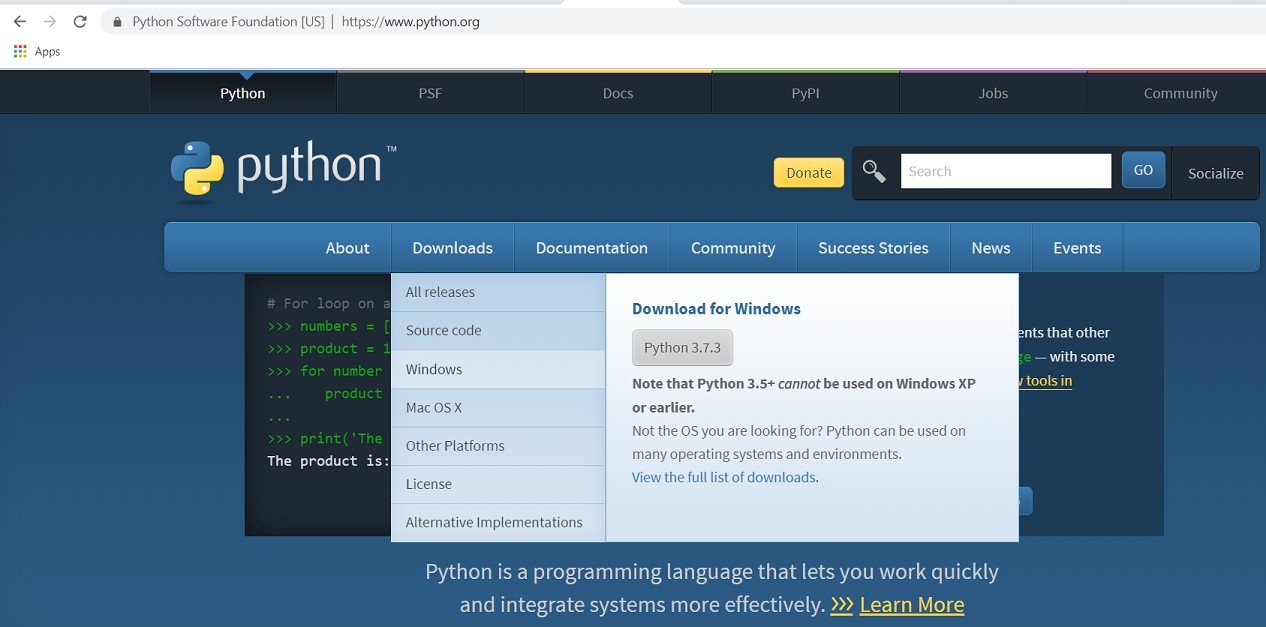


Fig 2.2.1.1: Python download

### Installation (using Anaconda):

* Python programs are also executed using Anaconda.
* Anaconda is a free open-source distribution of python for large scale data processing, predictive analytics and scientific computing.
* Conda is a package manager that quickly installs and manages packages.
* In WINDOWS:
* Step 1: Open Anaconda.com/downloads in the web browser.
* Step 2: Download python 3.4 version for (32-bits graphic installer/64 -bit graphic installer)
* Step 3: select installation type (all users)
* Step 4: Select path (i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish
* Step 5: Open Jupiter notebook (it opens in default browser)

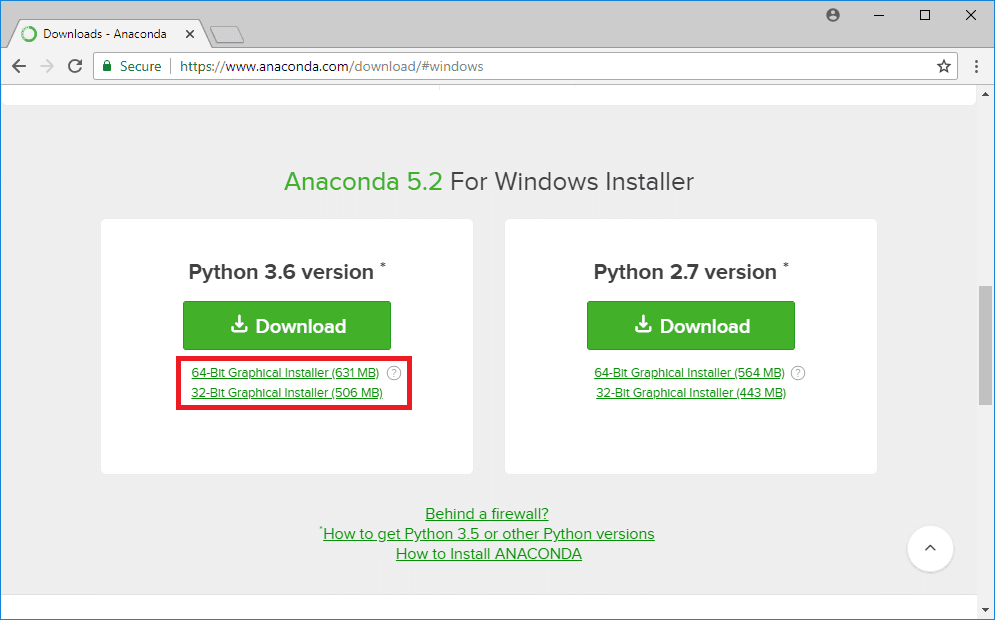


Fig 2.2.2.1: Anaconda download

### FEATURES OF PYTHON:

* Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* Easy-to-read: Python code is more clearly defined and visible to the eyes. ∙ Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
* A broad standard library: Python's bulk of the library is very portable and cross platform compatible on UNIX, Windows, and Macintosh.
* Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* Extendable: You can add low-level modules to the Python interpreter.
* These modules enable programmers to add to or customize their tools to be more efficient.
* Databases: Python provides interfaces to all major commercial databases.
* GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

### PYTHON VARIABLE TYPES:

Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory. Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory. Therefore, by assigning different data types to variables, you can store integers, decimals or characters in these variables.

Python has five standard data types are:

* Numbers
* Strings
* Lists
* Tuples
* Dictionary

### Python Numbers:

* Number data types store numeric values. Number objects are created when you assign a value to them.
* Python supports four different numerical types − int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

### Python Strings:

* Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
* Python allows for either pairs of single or double quotes.
* Subsets of strings can be taken using the slice operator ([ ] and [:] ) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
* The plus (+) sign is the string concatenation operator and the asterisk (\*) is the repetition operator.

### Python Lists:

* Lists are the most versatile of Python's compound data types.
* A list contains items separated by commas and enclosed within square brackets ([]).
* To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data types.
* The values stored in a list can be accessed using the slice operator ([ ] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1. ∙ The plus (+) sign is the list concatenation operator, and the asterisk (\*) is the repetition operator.

### Python Tuples:

* A tuple is another sequence data type that is similar to the list.
* A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
* The main differences between lists and tuples are: Lists are enclosed in brackets ( [ ] ) and their elements and size can be changed, while tuples are enclosed in parentheses ( ( ) ) and cannot be updated.
* Tuples can be thought of as read-only lists.
* For example − Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple.
* Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

### Python Dictionary:

* Python's dictionaries are a kind of hash table type.
* They work like associative arrays or hashes found in Perl and consist of key-value pairs. ∙ A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.
* Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
* You can use numbers to "index" into a list, meaning you can use numbers to find out what’s in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.
* What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

### PYTHON FUNCTION:

* + 1. **Defining a Function:**

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword def followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses.

The code block within every function starts with a colon (:) and is indented. The statement returns [expression] exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as return None.

### Calling a Function:

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

### PYTHON USING OOPs CONCEPTS:

* + 1. **Class:**

A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

* Class variable: A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.
* Data member: A class variable or instance variable that holds data associated with a class and its objects.
* Instance variable: A variable that is defined inside a method and belongs only to the current instance of a class.
* Defining a Class:
* We define a class in a very similar way to how we define a function.
* Just like a function ,we use parentheses and a colon after the class name(i.e. ():). Fig 3.6.1.1: Defining a Class

### init method in Class:

* The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
* The init method has a special name that starts and ends with two underscores: init ().

# CHAPTER 3: HEART DISEASE CLASSIFICATION

### PROJECT REQUIREMENTS:

* + 1. **Packages used:**

**Numpy:** In Python we have lists that serve the purpose of arrays, but they are slow to process. Numpy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in Numpy is called ndata, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

**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. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

**Matplotlib:** Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of Numpy, the numerical mathematics extension of Python. It provides an

object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers. Matplotlib has a procedural interface named Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with Numpy can be considered as the open-source equivalent of MATLAB.

**Scikit-learn:** also called Sklearn, is a robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling, including classification, regression, clustering, and dimensionality reduction via a consistent interface.

**Seaborn:** is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas’ data structures. Seaborn helps you explore and understand your data.

**Collection Module:** in Python provides different types of containers. A Container is an object that is used to store different objects and provide a way to access the contained objects and iterate over them. Some of the built-in containers are Tuple, List, Dictionary, etc.

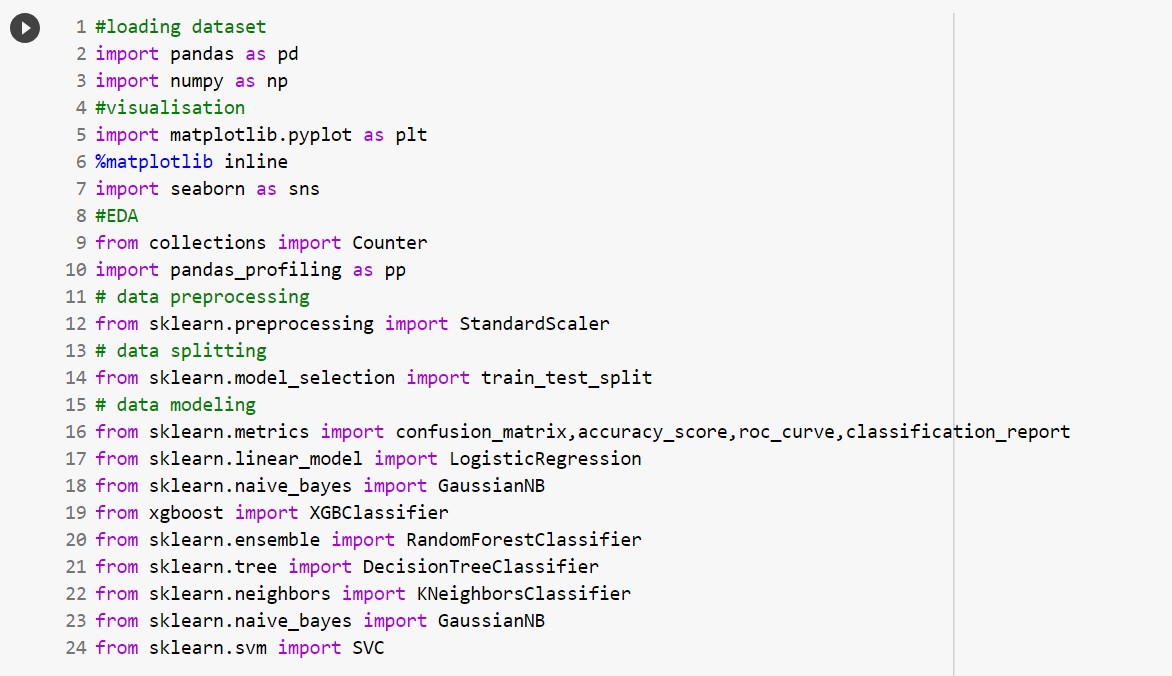


Fig 3.1.1.1: Packages used

### Algorithms used:

* + - * Logistic Regression
      * Naives Bayes
      * Random Forest
      * K-nearest neighbor
      * Decision Tree
      * Support Vector Machine

### PROBLEM STATEMENT:

The aim of this dataset is to help with the difficult task of detecting heart disease.

### DATASET DESCRIPTION:

The dataset is also distributed as a CSV formatted file hearts.csv. The loaded object contains 'age', 'sex', 'cp', 'trestbps', 'chol', 'fbs', 'restecg', 'thalach','exang', 'oldpeak', 'slope', 'ca', 'thal'.

### 3.4. OBJECTIVE OF THE CASE STUDY:

Objective of the problem is to help address the difficult task of predicting heart disease in a person. Automating this process can be applied to many issues including monitoring port activity levels and supply chain analysis. Goal is to predict heart disease in humans.

# CHAPTER 4: DATA PREPROCESSING/ FEATURE ENGINEERING AND EDA

### LOADING THE DATA:

Python has a built-in head () function to read a file. This function returns a file object, also called a handle, as it is used to read or modify the file accordingly.

We can specify the mode while opening a file. In mode, we specify whether we want to read, write or append to the file. We can also specify if we want to open the file in text mode or binary mode.

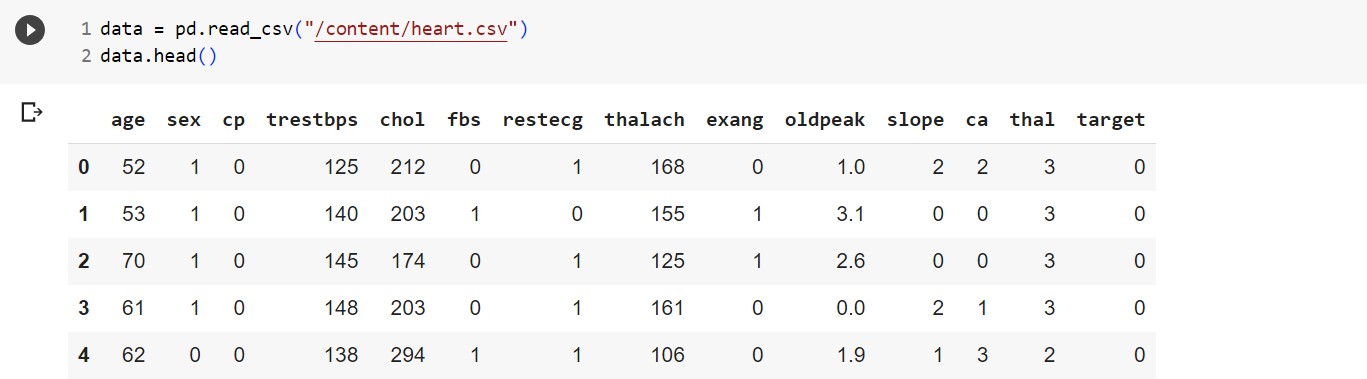


Fig 4.1.1: Loding data

Pandas DataFrame to which all the operations can be performed which helps us to access each and every row as well as columns and each and every value can be accessed using the dataframe. Any missing value or NaN value have to be cleaned.

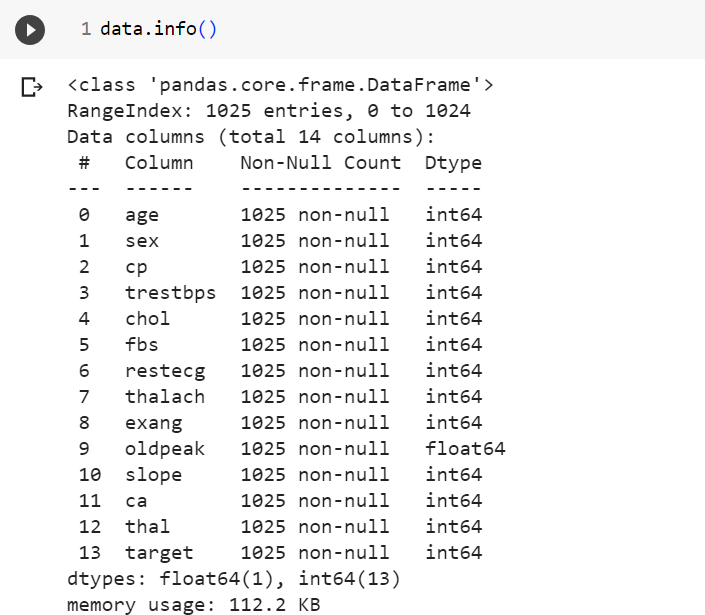


Fig 4.1.2: Reading Data

### STATISTICAL ANALYSIS:

Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset’s distribution, excluding nan values. Analyses both numeric and object series, as well as Data Frame column sets of mixed data types. The output will vary depending on what is provided.

For numeric data, the result’s index will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default, the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

For object data (e.g. strings or timestamps), the result’s index will include count, unique, top and freq. The top is the most common value. The freq is the most common value’s frequency.

Timestamps also include the first and last items.

If multiple object values have the highest count, then the count and top results will be arbitrarily chosen from among those with the highest count.

For mixed data types provided via a DataFrame , the default is to return only an analysis of numeric columns. If the data frame consists only of object and categorical data without any numeric columns, the default is to return an analysis of both the object and categorical columns. If include='all' is provided as an option, the result will include a union of attributes of each type.

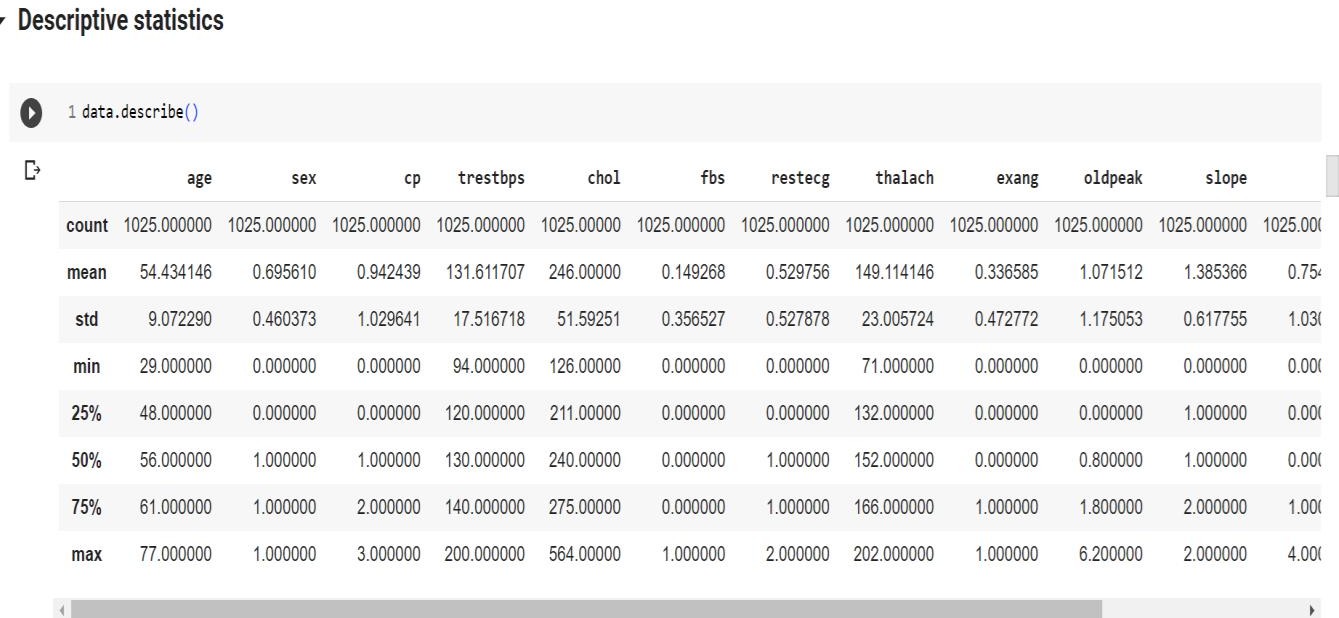


Fig 4.2.1: Statistical data of labels

### HANDLING MISSING VALUES:

There are a number of schemes that have been developed to indicate the presence of missing data in a table or DataFrame. Generally, they revolve around one of two strategies: using a mask that globally indicates missing values, or choosing a sentinel value that indicates a missing entry.

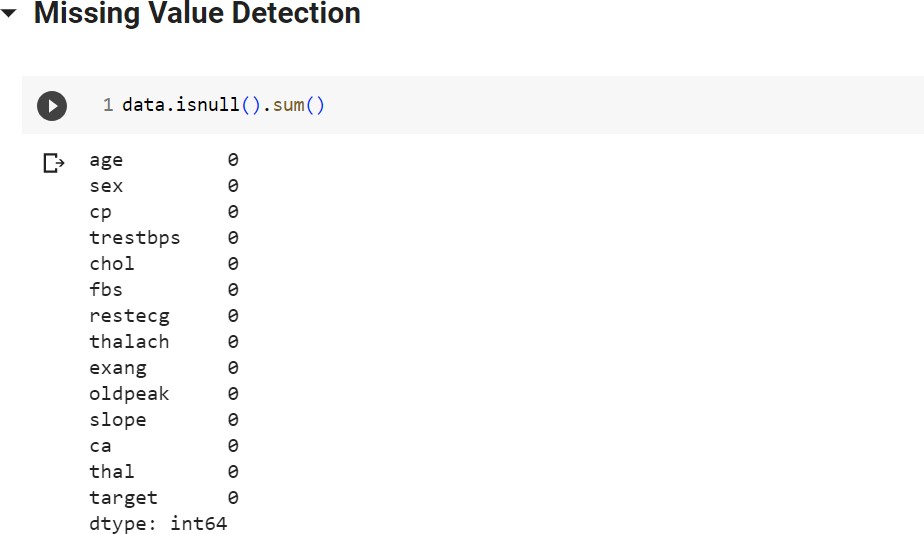


Fig 4.3.1: No Missing Values in the data set

# CHAPTER 5: MODEL BUILDING AND EVALUATION

### BRIEF ABOUT THE ALGORITHMS USED AND BUILDING MODEL

Machine learning models are majorly classified as supervised and unsupervised. If the model is supervised, it is divided into two categories: regression and classification. We will focus on the following machine learning models:

1. **Logistic Regression:** It is a basic classification algorithm which predicts the probability of a target variable.

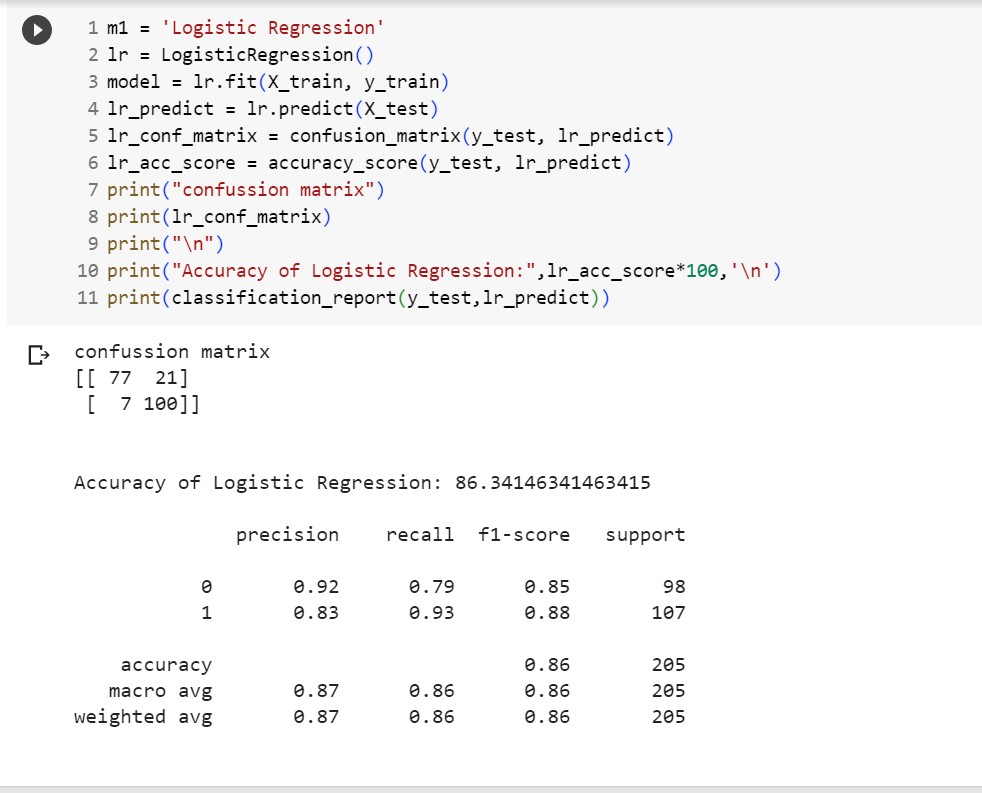


Fig 5.1.1: Logistic Regression

1. **K-nearest Neighbors:** It's a machine learning algorithm that's supervised. The idea behind nearest neighbor methods is to find a predetermined number of training samples that are closest in distance to the new point and use them to predict the mark. It makes no assumptions about the data and is typically used for classification tasks where little to no prior knowledge of the data distribution is available. Finding the k closest data points in the training set to the data point for which a target value is unavailable and assigning the average value of the identified data points to it is the aim of this algorithm.

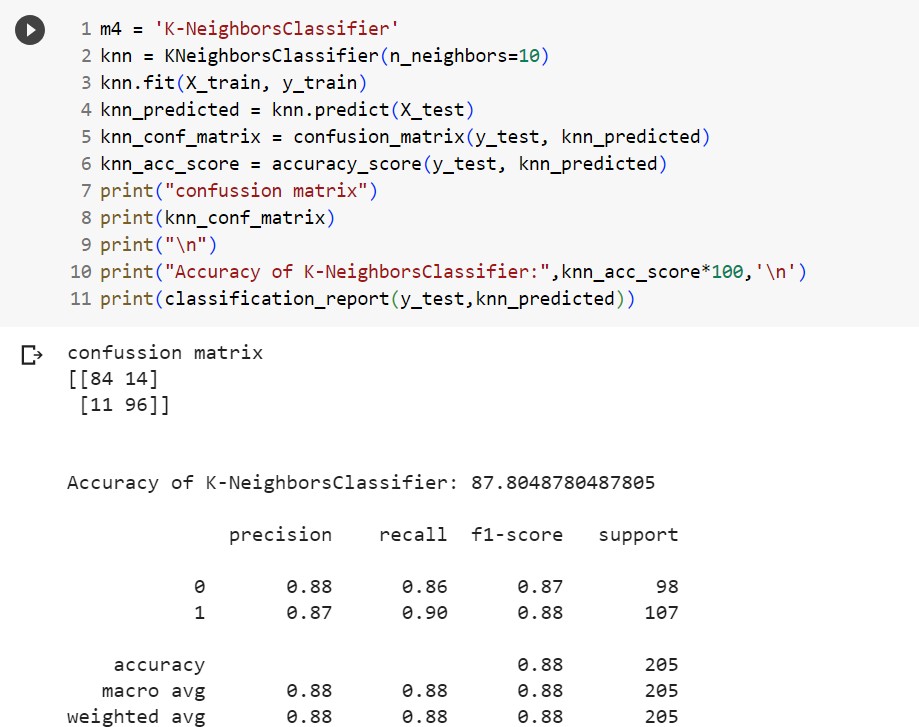


Fig 5.1.2: K-Nearest Neighbors

1. **Random Forest:** Random Forest is a supervised machine learning algorithm that can be used to solve problems in both classification and regression. It builds decision trees out of data samples, then gets predictions from each of them before voting on the best solution.

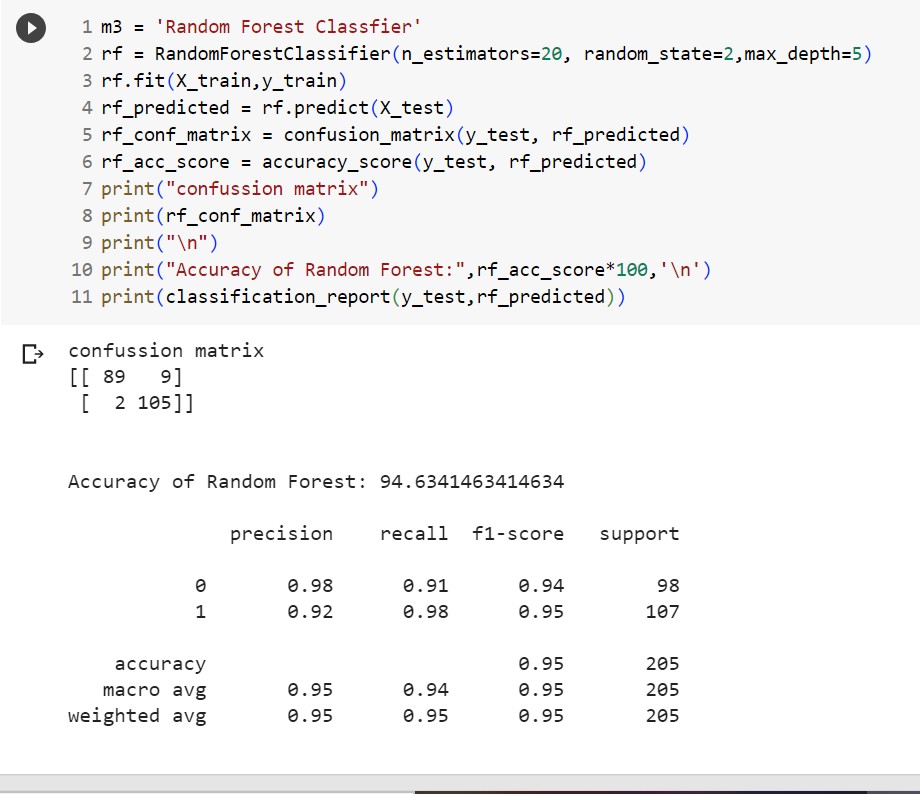


Fig 5.1.3: Random Forest

1. **Naive Bayes:** The Naïve Bayes classifier is a supervised machine learning algorithm, which is used for classification tasks, like text classification. It is also part of a family of generative learning algorithms, meaning that it seeks to model the distribution of inputs of a given class or category. Unlike discriminative classifiers, like logistic regression, it does not learn which features are most important to differentiate between classes.

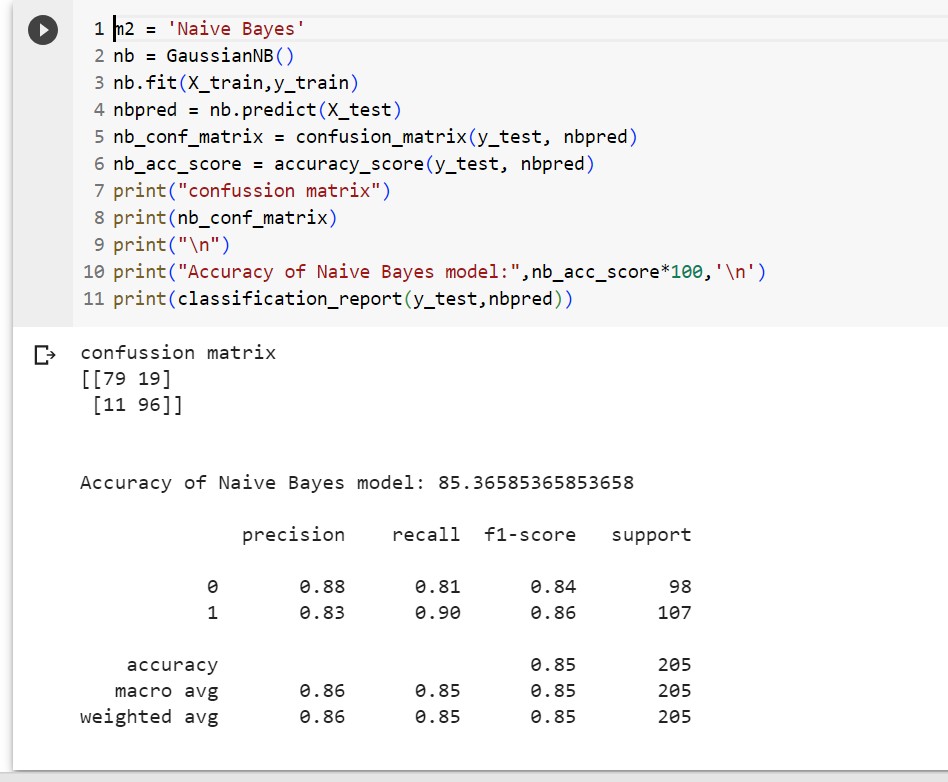


Fig 5.1.4: Naïve Bayes

1. **Decision Tree:** Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

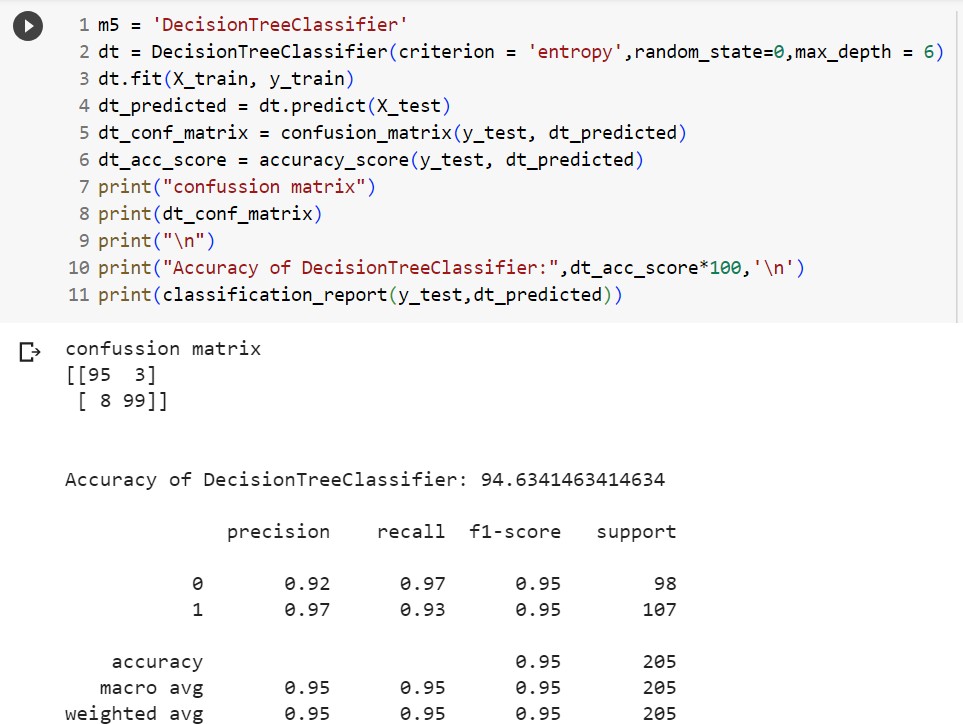


Fig 5.1.5: Decision Tree

1. **Support Vector:** Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning.The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future.

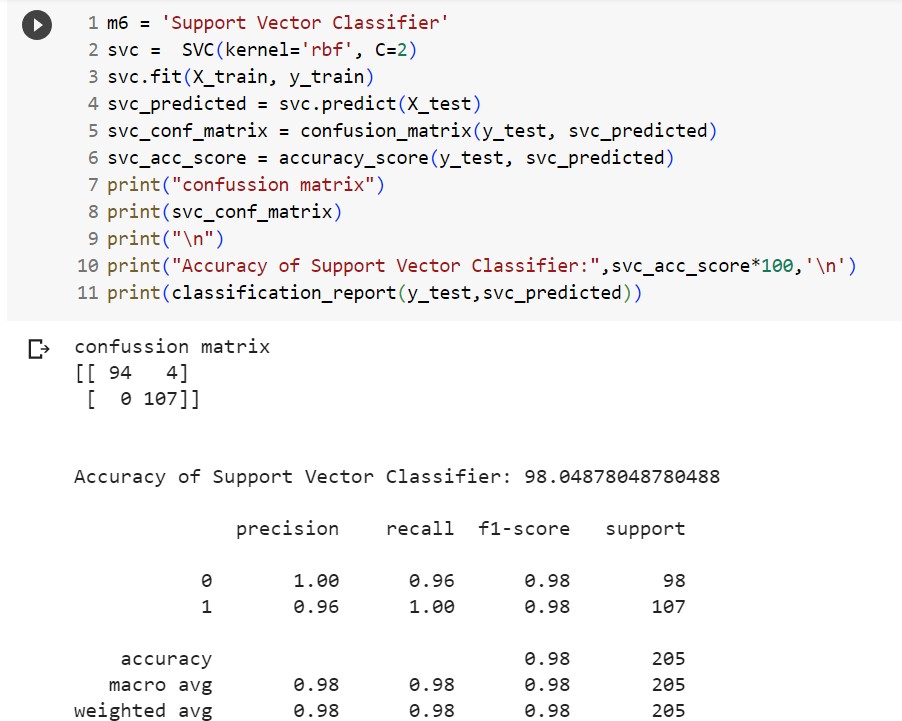


Fig 5.1.6: Support Vector

### TRAIN THE MODELS

Splitting the data: after the preprocessing is done then the data is split into train and Validation sets

* In deep learning in order to assess the performance of the classifier. You train the classifier using a 'training set' and then test the performance of your classifier on an unseen ‘validation set'. An important point to note is that during training the classifier only uses the training set. The validation set must not be used during training the classifier. The validation set will only be available during testing the classifier.
* Training set - a subset to train a model. (Model learns patterns between Input and Output)
* Validation set - a subset to test the trained model.(To test whether the model has correctly learnt )
* The amount or percentage of Splitting can be taken as specified.
* First we need to identify the input and output variables and we need to separate the input set and output set.
* Datasets are usually grouped into batches (especially when the amount of data is very large). Some people use the term iteration loosely and refer to putting one batch through the model as an iteration.. This helps in attaining good accuracy to the model.
* Here, the model has been split into 80% for trained data and the rest 20% of the data is of validation data by using valididation\_split parameter.

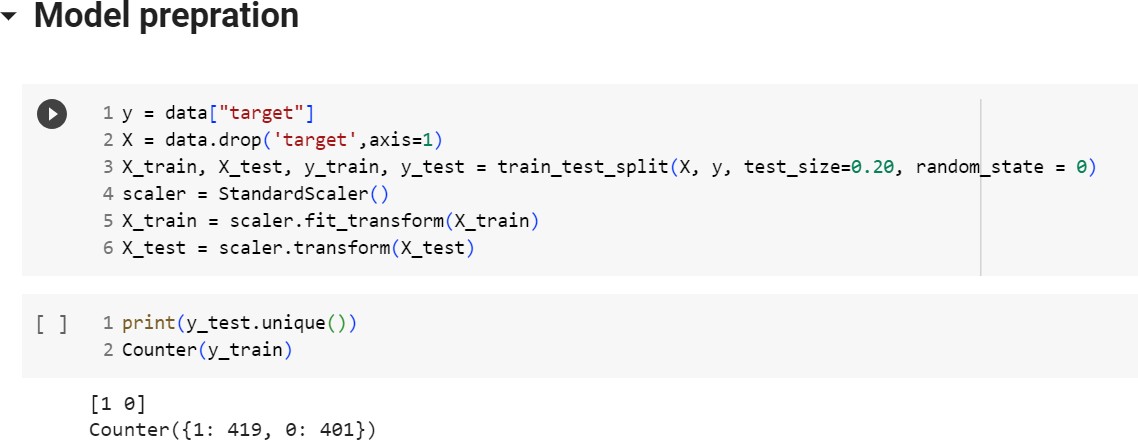


Fig 5.2.1 Building the Model

### VALIDATE THE MODELS:

Model validation is the process of evaluating a trained model on a test data set. This provides the generalization ability of a trained model. Here I provide a step by step approach to complete the first iteration of model validation in minutes.

* The models are validated after completion of training and testing the model. ∙ Checking the accuracy scores as metrics to validate the models.

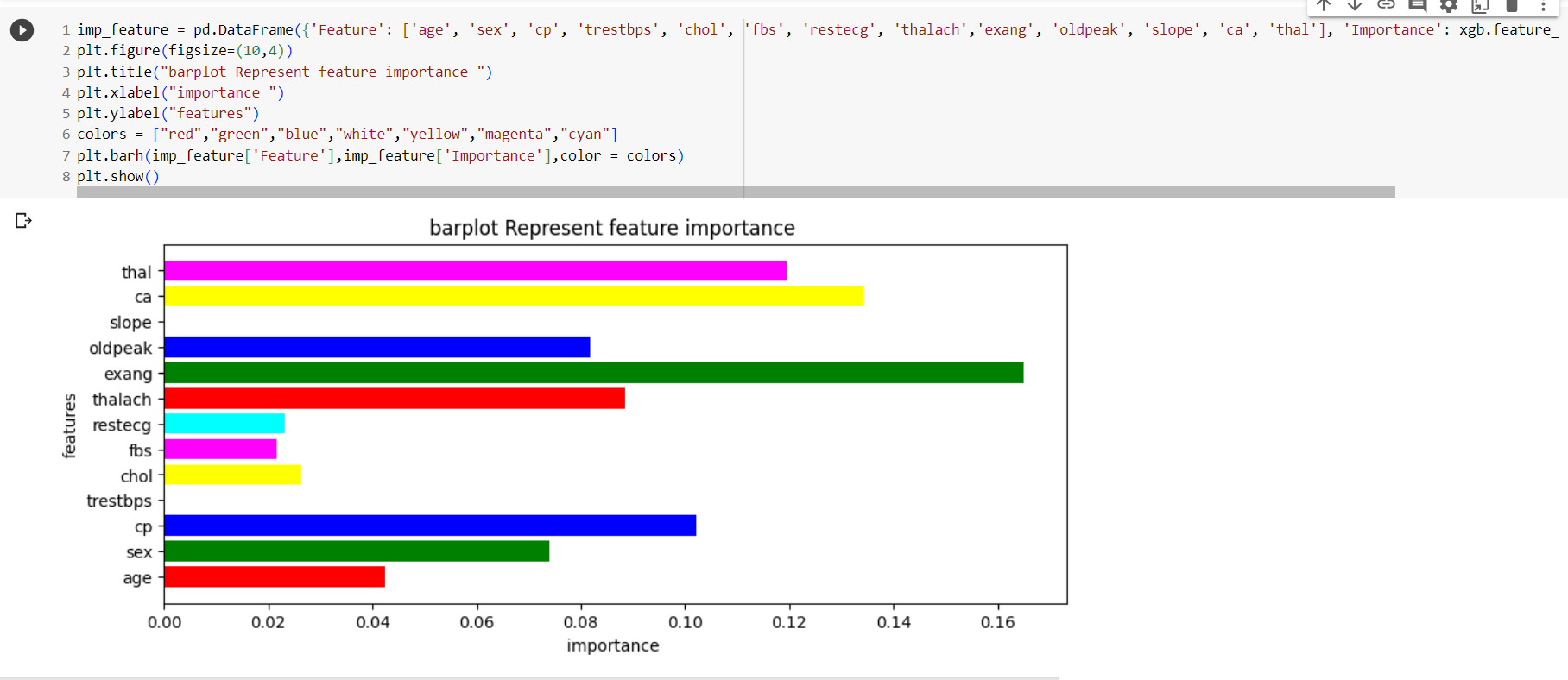


Fig 5.3.1: Decision Tree

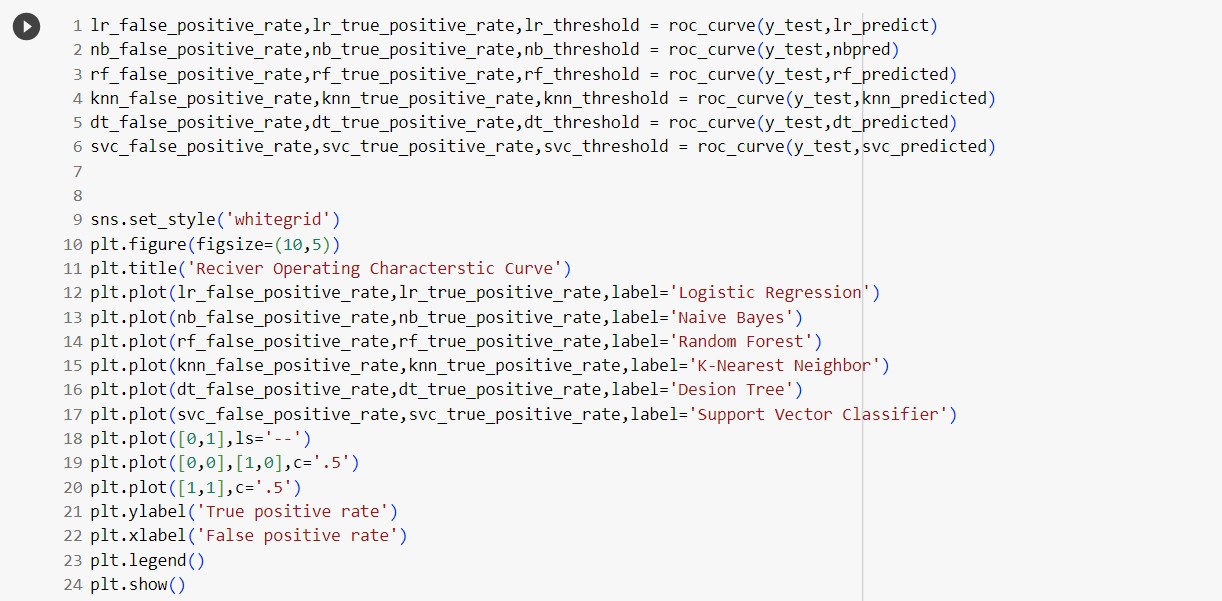


Fig 5.3.2: Valida

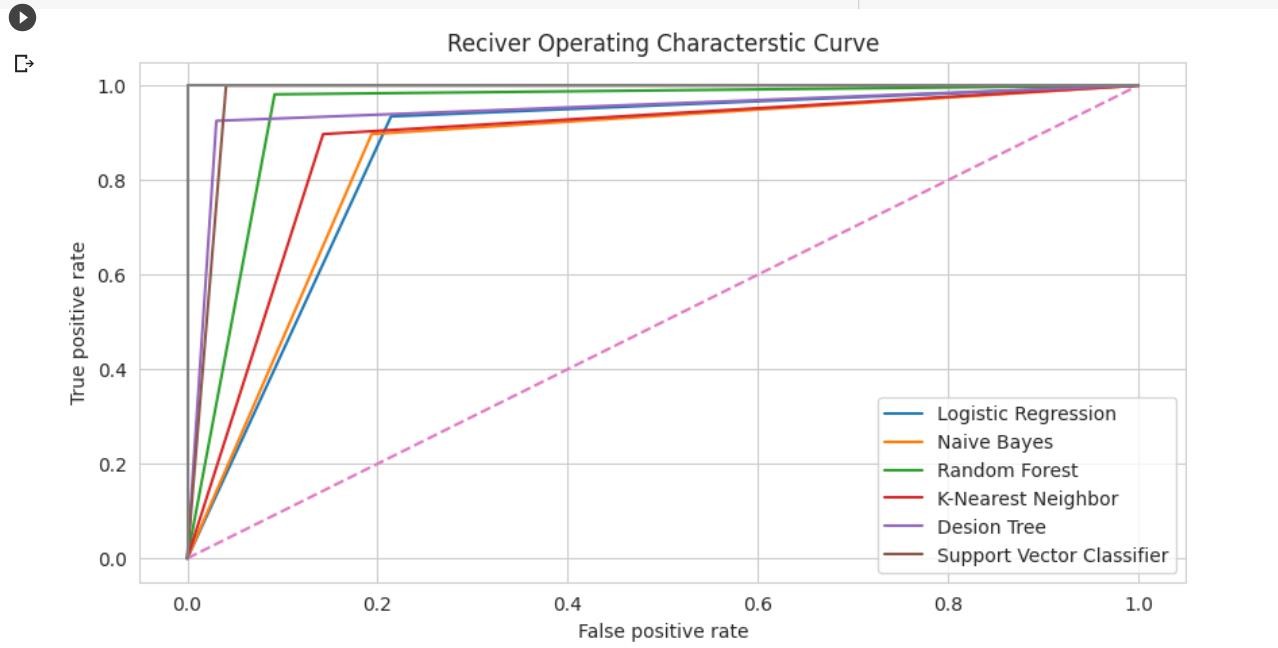


Fig 5.3.3 Model Validation

### MODEL EVALUATION

Model Evaluation can give us a brief information on the accuracies of the machine learning models we have used to predict the correct information. In all the models which has high accuracy can give us a correct prediction of heart disease.



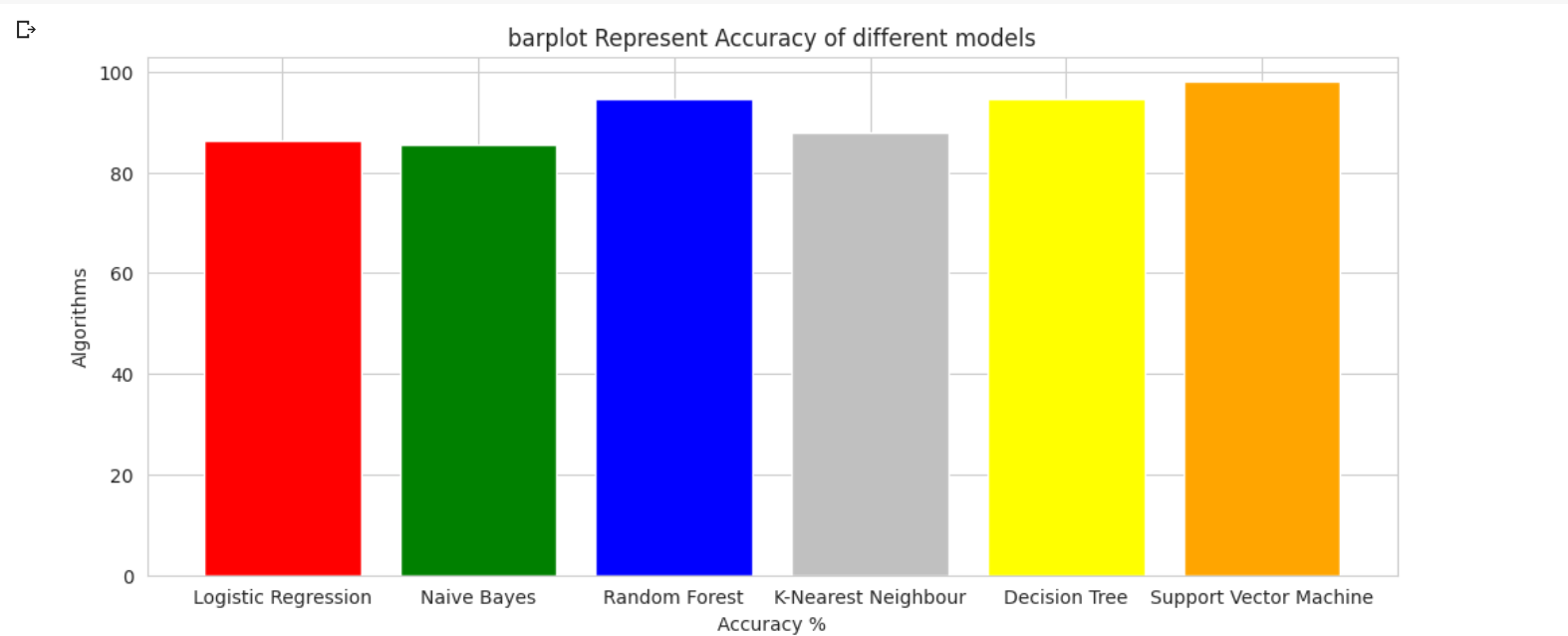


Fig 5.4.1 Model Validation

* + - Hence, here the accuracy of the support vector machine is high.
    - So, it gives us an accurate and reliable prediction on heart disease.

# CONCLUSION

With the rising number of deaths due to heart disease, it is becoming increasingly important to build a system that can effectively and accurately forecast heart disease. The motivation for the study was to find the most efficient ML algorithm for detection of heart diseases. This study compares the accuracy score of KNN, Logistic Regression and Random Forest for predicting heart disease using hearts dataset. The result of this study indicates that the Support Vector Machine algorithm is the most efficient algorithm with an accuracy score of 98% for prediction of heart disease. Accuracy of the algorithms in machine learning depends upon the dataset that is used for training and testing purposes.

**REFERENCES**