Heart Disease Prediction Using Effective Machine Learning Techniques

**ABSTRACT:**

In today’s era deaths due to heart disease has become a major issue approximately one person dies per minute due to heart disease. This is considering both male and female category and this ratio may vary according to the region also this ratio is considered for the people of age group 25-69. This does not indicate that the people with other age group will not be affected by heart diseases. This problem may start in early age group also and predict the cause and disease is a major challenge nowadays. Here in this paper, we have discussed various algorithms used for prediction of heart diseases. And create a GUI based system for predicting heart disease.

**Keywords:** dataset, Machine learning-Classification method

**Introduction:**

The major challenge that the Healthcare industry faces now-a-days is superiority of facility. Diagnosing the disease correctly & providing effective treatment to patients will define the quality of service. Poor diagnosis causes disastrous consequences that are not accepted.

Records or data of medical historyis very large, but these are from many dissimilarfoundations. The interpretations that are done by physicians are essential components of these data. The data in real world might be noisy, incomplete and inconsistent, so data preprocessing will be required in directive to fill the omitted values in the database.

Even if cardiovascular diseases is found as the important source of death in world in ancient years, these have been announced as the most avoidable and manageable diseases. The whole and accurate management of a disease rest on on the well-timed judgment of that disease. An correct and methodical tool for recognizing high-risk patients and mining data for timely analysis of heart infection looks a serious want.

Even though heart disease is acknowledged as the supreme chronic sort of diseasein the world, it can be most avoidable one also at the same time. A healthy way of life (main prevention) and timely analysis (inferior prevention) are the two major origins of heart disease director. Conducting steady check-ups (inferior prevention) showsoutstanding role in the judgment and early prevention of heart disease difficulties. Several tests comprising of angiography, chest X-rays, echocardiography and exercise tolerance test support to this significant issue. Nevertheless, these tests are expensive and involve availability of accurate medical equipment.

Heart expert’s create a good and huge record of patient’s database and store them. It also delivers a great prospect for mining a valued knowledge from such sort of datasets.

Researchers make use of several data mining techniques that are accessible to help the specialists or physicians identify the heart disease. Commonly used procedures used are decision tree, k-nearest and Naïve Bayes. Other different classification based techniques used are bagging algorithm, kernel density, sequential minimal optimization and neural networks, straight Kernel self-organizing map and SVM (Support Vector Machine).

**Domain overview:**

Machine learning is to predict the future from past data. Machine learning (ML) is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data and the basics of Machine Learning, implementation of a simple machine learning algorithm using python. Process of training and prediction involves use of specialized algorithms. It feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data. Machine learning can be roughly separated in to three categories. There are supervised learning, unsupervised learning and reinforcement learning. Supervised learning program is both given the input data and the corresponding labeling to learn data has to be labeled by a human being beforehand. Unsupervised learning is no labels. It provided to the learning algorithm. This algorithm has to figure out the clustering of the input data. Finally, Reinforcement learning dynamically interacts with its environment and it receives positive or negative feedback to improve its performance.

Data scientists use many different kinds of machine learning algorithms to discover patterns in python that lead to actionable insights. At a high level, these different algorithms can be classified into two groups based on the way they “learn” about data to make predictions: supervised and unsupervised learning. Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function from input variables(X) to discrete output variables(y). In machine learning and statistics, classification is a supervised learning approach in which the computer program learns from the data input given to it and then uses this learning to classify new observation. This data set may simply be bi-class (like identifying whether the person is male or female or that the mail is spam or non-spam) or it may be multi-class too. Some examples of classification problems are: speech recognition, handwriting recognition, bio metric identification, document classification etc.

Analyses Predicts

Machine Learning

Past Dataset

Trains

Fig: Process of Machine learning

[Supervised Machine Learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) **is the** majority of practical machine learning uses supervised learning. Supervised learning is where have input variables (X) and an output variable (y) and use an algorithm to learn the mapping function from the input to the output**is y = f(X).** The goal is to approximate the mapping function so well that when you have new input data (X) that you can predict the output variables (y) for that data. Techniques of Supervised Machine Learning algorithms include **logistic regression**, **multi-class classification**, **Decision Trees** and **support vector machines etc**. Supervised learning requires that the data used to train the algorithm is already labeled with correct answers. Supervised learning problems can be further grouped into **Classification** problems. This problem has as goal the construction of a succinct model that can predict the value of the dependent attribute from the attribute variables. The difference between the two tasks is the fact that the dependent attribute is numerical for categorical for classification. A classification model attempts to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. A classification problem is when the output variable is a category, such as “red” or “blue”.

Agriculture is one of the most important occupations practiced in our country. It is the broadest economic sector and plays an important role in overall development of the country. About 60 % of the land in the country is used for agriculture in order to suffice the needs of 1.2 billion people. Thus, modernization of agriculture is very important and thus will lead the farmers of our country towards profit. Data analytic (DA) is the process of examining data sets in order to draw conclusions about the information they contain, increasingly with the aid of specialized systems and software. Earlier yield prediction was performed by considering the farmer's experience on a particular field and crop. However, as the conditions change day by day very rapidly, farmers are forced to cultivate more and more crops. Being this as the current situation, many of them don’t have enough knowledge about the new crops and are not completely aware of the benefits they get while farming them. Also, the farm productivity can be increased by understanding and forecasting crop performance in a variety of environmental conditions. Thus, the proposed system takes the location of the user as an input. From the location, the nutrients of the soil such as Nitrogen, Phosphorous, Potassium is obtained. This static data is the crop production and data related to demands of various crops obtained from various websites. It applies machine learning and prediction algorithm to identify the pattern among data and then process it as per input conditions.

**Preparing the Dataset:**

The demo dataset is now supplied to machine learning model on the basis of this data set the model is trained. Every new detail filled at the time of application form acts as a test data set. After the operation of testing, model prediction based upon the inference it concludes on the basis of the training data sets. Satellite Imagery (Remote Sensing Data), has been widely used for predicting crop yield. This dataset is collected using the sensors mounted on satellites or planes, which detect the energy (electromagnetic waves), reflected or diffracted from surface of the earth. Remote sensing data has a lot of energy bands to offer, but mainly only few of them have been used for crop yield prediction. Yet, there are some people who have tried generating relevant features using the bands which are typically ignored, and they have been successful with improving results with that. In case of this dataset, most people rarely explore the high-order moments of the features. Based on these datasets people have used algorithms like Regression models, Random Forest and Nearest Neighbor etc.

Table shows details of the datasets:

|  |  |
| --- | --- |
| **Variable** | **Description** |
| Age | Age of person |
| Sex | Male or Female |
| cp | Chest pain type |
| trestbps | Resting blood pressure |
| chol | Cholesterol level |
| fbs | Fasting blood sugar |
| Rest ecg | Resting electrocardiogram |
| thalach | Maximum heart rate achieved |
| exang | Exercise induced angina |
| oldpeak | ST depression induced by exercise relative to rest |
| slope | the slope of the peak exercise ST segment |
| ca | number of major vessels (0-3) colored by flourosopy |
| thal | 3 = normal; 6 = fixed defect; 7 = reversable defect |

**Problem Description:**

India has witnessed an alarming rise in the occurrence of heart disease, stroke, diabetes and cancers in the past 25 years, a series of new studies published on Wednesday in [The Lancet](https://www.thelancet.com/journals/langlo/article/PIIS2214-109X(18)30407-8/fulltext) and its associated journals have revealed.

Detailed estimates of cardiovascular diseases, diabetes, chronic respiratory diseases, cancer, and suicide show that their prevalence has gone up in every Indian state between 1990 and 2016, but there is vast variation among states.

The prevalence of heart disease and stroke has increased by over 50% from 1990 to 2016 in India, with an increase observed in every state. The contribution of these diseases to total deaths and disease burden in the country has almost doubled in the past 25 years. Heart disease now is the leading individual cause of disease burden in India, and stroke is the fifth leading cause.

**Scope:**

The scope of this project is to investigate a dataset of heart disease records for patients using machine learning technique. To identifying heart disease is more difficult. We try to reduce this problem by applying machine learning techniques.

**Objectives:**

* Data validation
* Data Cleaning/ Preparing
* Data Visualization
* Using more algorithm with comparing to predict more accuracy (Like random forest, Decision tree Logistic classification algorithm)

**LITERATURE SURVEY**

**General:**

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

**Review of Literature Survey:**

1. **PROBLEMS WITH MINING MEDICAL DATA**

In year 2000, research conducted by ShusakuTsumoto says that as we human beings are unable to arrange data if it is huge in size we should use the data mining techniques that are available for finding different patterns from the available huge database and can be used again for clinical research and perform various operations on it.

2. **EVIDENCE COMBINATION IN MEDICAL DATA MINING**

Y. Alp Aslandogan, et. al. (2004), worked on three different classifiers called K-nearest Neighbour (KNN), Decision Tree, Naïve Bayesian and used Dempsters’ rule for this three viewpoint to appear as one concluding decision. This classification based on the combined idea show increased accuracy.

**3. IMPROVING HEART DISEASE PREDICTION USING CONSTRAINED ASSOCIATION RULES**

Carlos Ordonez (2004), Assessed the problematic to recognize and forecast the rule of relationship for the heart disease. Adataset involving medical history of the patients having heart disease with the aspects of risk factors was accessed by him, measurements of narrowed artery and heart perfusion. All these restrictions were announced to shrink the digit of designs, these are as follows:

1) The features should seem on a single side of the rule.

2) The rule should distinct variousfeatures into the different groups.

3) The count of featuresavailable from the rule is organized by medical history of people having heart disease only. The occurrence or the nonappearance of heart disease was predicted by the author in four heart veins with the two clusters of rules.

**4. PREDICTING SURVIVAL CAUSES AFTER OUT OF HOSPITAL CARDIAC ARREST USING DATA MINING METHOD**

Franck Le Duff (2004), worked on creating Decision tree quickly with clinical data of the physician or service. He suggested few data mining techniques which can help cardiologists in the predication survival of patients. The main drawback of the system was that the user needs to have knowledge of the techniques and we should collect sufficient data for creating an suitable model.

**5. USING EFFICIENT SUPANOVA KERNEL FOR HEART DISEASE DIAGNOSIS**

Boleslaw Szymanski, et. al. (2006), operated on a novel experiential to check the aptitude of calculation of scarce kernel in SUPANOVA. The author used this technique on a standard boston housing market dataset for discovering heart diseases, measurement of heart activities and prediction of heart diseases were found 83.7% correct which were measured with the help of support vector machine and kernel equivalent to it. A quality result is gained by spline kernel with the help of standard boston housing market database.

**6. ASSOCIATIVE CLASSIFICATION APPROACH FOR DIAGNOSING CARDIOVASCULAR DISEASE**

Kiyong Noh, et. al. (2006) made use of a classification technique for removal of multi-parametric structures by accessing HRV and ECG signals. Kiyong used the FP-growth algorithm as the foundation of this technique that is associative. A rule consistency degree was gained which allows a robust press on trimming designs in the method of producing designs.

**7. MININGBIOSIGNAL DATA: CORONARY ARTERY DISEASE DIAGNOSIS USING LINEAR AND NONLINEAR FEATURES OF HRV**

HeonGyu Lee, et. al. (2007), operated for the operation systems of Arithmetical and cataloguing for the addition chief of the multi-parametric feature through direct and nonlinear features of Heart Rate Variability (HRV). The dissimilar classifiers existing are cataloguing grounded on Decision Tree (C4.5), Multiple Association Rules (CMAR) and Bayesian classifiers, and Support Vector Machine (SVM) that are investigated for the valuation of the linear and nonlinear features of the HRV tables.

**8. DECISION SUPPORT SYSTEM FOR HEART DISEASE DIAGNOSIS USING NEURAL NETWORK**

Niti Guru, et. al. (2007), functioned for forecasting of heart disease, Blood Stress and Sugar by the aid of neural systems. Hearings were accepted out on example best ever of patients. The neural system is verified with 13 types, as blood pressure,period, angiography etc. [12].

**Existing System:**

In hospitals doctors use many procedures which are costly to find heart diseases.

These techniques include stress test, chest x-ray, computed tomography (CT), nuclear scanning, echocardiogram (**heart** ultrasound), or magnetic resonance (MR). ... A wide range of modalities may also be used to **monitor** patients with **heart disease**/damage during and after treatment, including: Electrocardiogram (ECG or EKG)

**Drawbacks:**

* Many patients cant afford huge amount for tests.
* Some times even after verifying testing reports its difficult to say that a patient has heart disease or not..

**Proposed System:**

## Exploratory Data Analysis:

## In this section of the report, you will load in the data, check for cleanliness, and then trim and clean your dataset for analysis. Make sure that you document your steps carefully and justify your cleaning decisions.

Training the Dataset:

* The first line imports iris data set which is already predefined in sklearn module. Iris data set is basically a table which contains information about various varieties of iris flowers.
* For example, to import any algorithm and train\_test\_split class from sklearn and numpy module for use in this program.
* Then we encapsulate load\_data() method in data\_dataset variable. Further we divide the dataset into training data and test data using train\_test\_split method. The X prefix in variable denotes the feature values and y prefix denotes target values.
* This method divides dataset into training and test data randomly in ratio of 67:33. Then we encapsulate any algorithm.
* In the next line, we fit our training data into this algorithm so that computer can get trained using this data. Now the training part is complete.

Testing the Dataset:

* Now we have dimensions of a new flower in a numpy array called ‘n’ and we want to predict the species of this flower. We do this using the predict method which takes this array as input and spits out predicted target value as output.
* So the predicted target value comes out to be 0. Finally we find the test score which is the ratio of no. of predictions found correct and total predictions made. We do this using the score method which basically compares the actual values of the test set with the predicted values.

Advantages:

* Our goal is push for assisting doctors and patients using our predictions. All these publications state they have done better than their competitors but there is no article or public mention of their work being used practically to assist the doctors. If there are some genuine problems in rolling out that work to next stage, then identify those problems and try solving them.

Application:

* medical sector to automate to identify the heart disease (real time world) and predicting by desktop application / web application.

Overview of the system:

This helps all others department to carried out other formalities. It have to find Accuracy of the training dataset, Accuracy of the testing dataset, Specification, False Positive rate, precision and recall by comparing algorithm using python code. The following Involvement steps are,

* Define a problem
* Preparing data
* Evaluating algorithms
* Improving results
* Predicting results

The steps involved in Building the data model is depicted below.

**Prediction (**Heart Disease prediction**)**

**Building classification Model**

**Pre Processing** (Outlier Detection, Outlier Removal, Imputation Removal)

**Data collection** (Splitting Training set & Test set)

Fig: data flow diagram for Machine learning model

Project Goals:

# Exploration data analysis of variable identification

* Loading the given dataset
* Import required libraries packages
* Analyze the general properties
* Find duplicate and missing values
* Checking unique and count values

# Uni-variate data analysis

* Rename, add data and drop the data
* To specify data type

# Exploration data analysis of bi-variate and multi-variate

* Plot diagram of pairplot, heatmap, bar chart and Histogram

# Method of Outlier detection with feature engineering

* Pre-processing the given dataset
* Splitting the test and training dataset
* Comparing the Decision tree and Logistic regression model and random forest

# Comparing algorithm to predict the result

* Based on the accuracy

Data collection:

The data set collected for predicting past farmer list of yield is split into Training set and Test set. Generally, 7:3 ratios are applied to split the Training set and Test set. The Data Model which was created using Random Forest , logistic , Decision tree algorithms are applied on the Training set and based on the test result accuracy, Test set prediction is done.

Data Gathering

Data Pre-Processing

Choose model

Train model

Test model

Tune model

Prediction

Fig: Process of dataflow diagram

General Properties:

Create cells freely to explore your data and you should not perform too many operations in each cell. One option that you can take with this project is to do a lot of explorations in an initial notebook. These don't have to be organized, but make sure you use enough comments to understand the purpose of each code cell. Then, after you're done with your analysis, create a duplicate notebook where you will trim the excess and organize your steps so that you have a flowing, cohesive report and make sure that you keep your reader informed on the steps that you are taking in your investigation. Follow every code cell, or every set of related code cells, with a markdown cell to describe to the reader what was found in the preceding cell. Try to make it so that the reader can then understand what they will be seeing in the following cell.

Patient details

Test dataset

Data Processing

Model

Classification ML Algorithm

Training dataset

Fig: Architecture of Proposed model

**Project Requirements**

**General:**

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements

2. Non-Functional requirements

3. Environment requirements

A. Hardware requirements

B. software requirements

**Functional requirements:**

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like sk-learn, pandas, numpy, matplotlib and seaborn.

**Non-Functional Requirements:**

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithms
4. Improving results
5. Prediction the result

**Environmental Requirements:**

1. Software Requirements:

Operating System : Windows

Tool : Anaconda with Jupyter Notebook

2. Hardware requirements:

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

**Software Description:**

Anaconda is a [free and open-source](https://en.wikipedia.org/wiki/Free_and_open-source) distribution of the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [R](https://en.wikipedia.org/wiki/R_(programming_language)) programming languages for [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing) ([data science](https://en.wikipedia.org/wiki/Data_science), [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications, large-scale data processing, [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics), etc.), that aims to simplify [package management](https://en.wikipedia.org/wiki/Package_management) and deployment. Package versions are managed by the [package management system](https://en.wikipedia.org/wiki/Package_manager) “Conda”. The Anaconda distribution is used by over 12 million users and includes more than 1400 popular data-science packages suitable for Windows, Linux, and MacOS. So, Anaconda distribution comes with more than 1,400 packages as well as the [Conda](https://en.wikipedia.org/wiki/Conda_(package_manager)" \o "Conda (package manager)) package and virtual environment manager called Anaconda Navigator and it eliminates the need to learn to install each library independently. The open source packages can be individually installed from the Anaconda repository with the conda install command or using the pip install command that is installed with Anaconda. [Pip packages](https://en.wikipedia.org/wiki/Pip_(package_manager)) provide many of the features of conda packages and in most cases they can work together. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, [PyPI](https://en.wikipedia.org/wiki/Python_Package_Index" \o "Python Package Index) or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, you can create new environments that include any version of Python packaged with conda.

Anaconda Navigator:

Anaconda Navigator is a desktop [graphical user interface (GUI)](https://en.wikipedia.org/wiki/Graphical_user_interface) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using [command-line commands](https://en.wikipedia.org/wiki/Command-line_interface). Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS" \o "MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux).

The following applications are available by default in Navigator:

* [JupyterLab](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Lab)
* [Jupyter Notebook](https://en.wikipedia.org/wiki/Project_Jupyter#Jupyter_Notebook)
* [QtConsole](https://qtconsole.readthedocs.io/en/latest/)
* [Spyder](https://en.wikipedia.org/wiki/Spyder_(software))
* [Glueviz](http://glueviz.org/)
* [Orange](https://en.wikipedia.org/wiki/Orange_(software))
* [Rstudio](https://en.wikipedia.org/wiki/Rstudio)
* [Visual Studio Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)

### Conda:

Conda is an [open source](https://en.wikipedia.org/wiki/Open-source_software), [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), language-agnostic [package manager](https://en.wikipedia.org/wiki/Package_manager) and environment management system that installs, runs and updates packages and their dependencies. It was created for Python programs, but it can package and distribute software for any language (e.g., [R](https://en.wikipedia.org/wiki/R_(programming_language))), including multi-languages. The Conda package and environment manager is included in all versions of Anaconda, Miniconda, and Anaconda Repository.

### The Jupyter Notebook:

#### The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

## [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id5):

Notebook documents (or “notebooks”, all lower case) are documents produced by the [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html" \l "notebook-app), which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc.) as well as executable documents which can be run to perform data analysis.

## [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id6):

The Jupyter Notebook App is a server-client application that allows editing and running [notebook documents](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document) via a web browser. The Jupyter Notebook App can be executed on a local desktop requiring no internet access (as described in this document) or can be installed on a remote server and accessed through the internet. In addition to displaying/editing/running notebook documents, the Jupyter Notebook App has a “Dashboard” ([Notebook Dashboard](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#dashboard)), a “control panel” showing local files and allowing to open notebook documents or shutting down their [kernels](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#kernel).

## [kernel](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id7):

A notebook kernel is a “computational engine” that executes the code contained in a [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document). The ipython kernel, referenced in this guide, executes python code. Kernels for many other languages exist ([official kernels](http://jupyter.readthedocs.org/en/latest/#kernels)). When you open a [Notebook document](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document), the associated kernel is automatically launched. When the notebook is executed (either cell-by-cell or with menu Cell -> Run All), the kernel performs the computation and produces the results. Depending on the type of computations, the kernel may consume significant CPU and RAM. Note that the RAM is not released until the kernel is shut-down.

## [Notebook Dashboard](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#id8):

The Notebook Dashboard is the component which is shown first when you launch [Jupyter Notebook App](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html" \l "notebook-app). The Notebook Dashboard is mainly used to open [notebook documents](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#notebook-document), and to manage the running [kernels](https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html#kernel) (visualize and shutdown). The Notebook Dashboard has other features similar to a file manager, namely navigating folders and renaming/deleting files.

Working Process:

* Download and install anaconda and get the most useful package for machine learning in Python.
* Load a dataset and understand its structure using statistical summaries and data visualization.
* machine learning models, pick the best and build confidence that the accuracy is reliable.

Python is a popular and powerful interpreted language. Unlike R, Python is a complete language and platform that you can use for both research and development and developing production systems. There are also a lot of modules and libraries to choose from, providing multiple ways to do each task. It can feel overwhelming.

The best way to get started using Python for machine learning is to complete a project.

* It will force you to install and start the Python interpreter (at the very least).
* It will give you a bird’s eye view of how to step through a small project.
* It will give you confidence, maybe to go on to your own small projects.

When you are applying machine learning to your own datasets, you are working on a project. A machine learning project may not be linear, but it has a number of well-known steps:

* Define Problem.
* Prepare Data.
* Evaluate Algorithms.
* Improve Results.
* Present Results.

The best way to really come to terms with a new platform or tool is to work through a machine learning project end-to-end and cover the key steps. Namely, from loading data, summarizing data, evaluating algorithms and making some predictions.

Here is an overview of what we are going to cover:

1. Installing the Python anaconda platform.
2. Loading the dataset.
3. Summarizing the dataset.
4. Visualizing the dataset.
5. Evaluating some algorithms.
6. Making some predictions.

### Model Selection:

This is the most exciting phase in Applying Machine Learning to any Dataset. It is also known as Algorithm selection for Predicting the best results. Usually Data Scientists use different kinds of Machine Learning algorithms to the large data sets. But, at high level all those different algorithms can be classified in two groups : supervised learning and unsupervised learning. Supervised learning : Supervised learning is a type of system in which both input and desired output data are provided. Input and output data are labeled for classification to provide a learning basis for future data processing. Supervised learning problems can be further grouped into **Regression** and **Classification** problems.

A **regression** problem is when the output variable is a real or continuous value, such as “salary” or “weight”. A **classification** problem is when the output variable is a category like filtering emails “spam” or “not spam”

Unsupervised Learning : Unsupervised learning is the [algorithm](https://whatis.techtarget.com/definition/algorithm) using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. In our dataset we have the outcome variable or Dependent variable i.e Y having only two set of values, either M (Malign) or B(Benign). So we will use Classification algorithm of supervised learning.

**Modules:**

* Data validation and pre-processing technique (Module-01)
* Exploration data analysis of visualization and training a model by given attributes (Module-02)
* Performance measurements of logistic regression and decision tree algorithms (Module-03)
* Performance measurements of Support vector classifier and Random forest (Module-04)
* Performance measurements of KNN and Naive Bayes (Module-05)
* GUI based prediction of heart disease (Module-06)

**Module-01:**

Variable Identification Process / data validation process:

Validation techniques in machine learning are used to get the error rate of the Machine Learning (ML) model, which can be considered as close to the true error rate of the dataset. If the data volume is large enough to be representative of the population, you may not need the validation techniques. However, in real-world scenarios, to work with samples of data that may not be a true representative of the population of given dataset. To finding the missing value, duplicate value and description of data type whether it is float variable or integer. The sample of data used to provide an unbiased evaluation of a model fit on the training dataset while tuning model hyper parameters. The evaluation becomes more biased as skill on the validation dataset is incorporated into the model configuration. The validation set is used to evaluate a given model, but this is for frequent evaluation. It as machine learning engineers uses this data to fine-tune the model hyper parameters. Data collection, data analysis, and the process of addressing data content, quality, and structure can add up to a time-consuming to-do list. During the process of data identification, it helps to understand your data and its properties; this knowledge will help you choose which algorithm to use to build your model. For example, time series data can be analyzed by regression algorithms; classification algorithms can be used to analyze discrete data. (For example to show the data type format of given dataset)

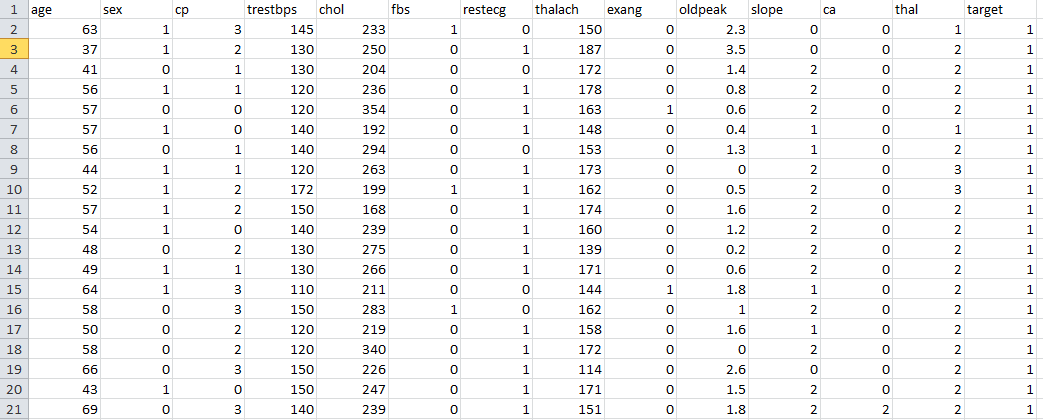


Fig: Given data frame

Data Validation/ Cleaning/Preparing Process:

Importing the library packages with loading given dataset. To analyzing the variable identification by data shape, data type and evaluating the missing values, duplicate values. A validation dataset is a sample of data held back from training your model that is used to give an estimate of model skill while tuning model's and procedures that you can use to make the best use of validation and test datasets when evaluating your models. Data cleaning / preparing by rename the given dataset and drop the column etc. to analyze the uni-variate, bi-variate and multi-variate process. The steps and techniques for data cleaning will vary from dataset to dataset. The primary goal of data cleaning is to detect and remove errors and anomalies to increase the value of data in analytics and decision making.

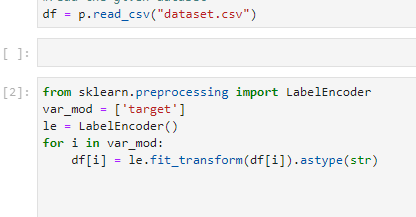


Fig: Spliting the given dataset

Data Pre-processing:

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. To achieving better results from the applied model in Machine Learning method of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format; for example, Random Forest algorithm does not support null values. Therefore, to execute random forest algorithm null values have to be managed from the original raw data set.

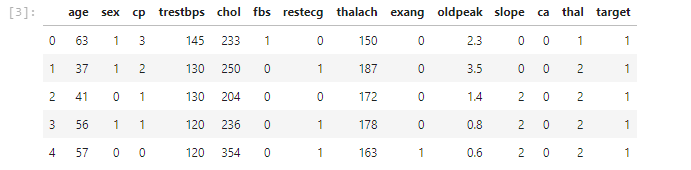


Fig: After preprocessing given data frame

**Module-02:**

Exploration data analysis of visualization:

Data visualization is an important skill in applied statistics and machine learning. Statistics does indeed focus on quantitative descriptions and estimations of data. Data visualization provides an important suite of tools for gaining a qualitative understanding. This can be helpful when exploring and getting to know a dataset and can help with identifying patterns, corrupt data, outliers, and much more. With a little domain knowledge, data visualizations can be used to express and demonstrate key relationships in plots and charts that are more visceral and stakeholders than measures of association or significance. Data visualization and exploratory data analysis are whole fields themselves and it will recommend a deeper dive into some the books mentioned at the end.

Sometimes data does not make sense until it can look at in a visual form, such as with charts and plots. Being able to quickly visualize of data samples and others is an important skill both in applied statistics and in applied machine learning. It will discover the many types of plots that you will need to know when visualizing data in Python and how to use them to better understand your own data.

* How to chart time series data with line plots and categorical quantities with bar charts.
* How to summarize data distributions with histograms and box plots.
* How to summarize the relationship between variables with scatter plots.

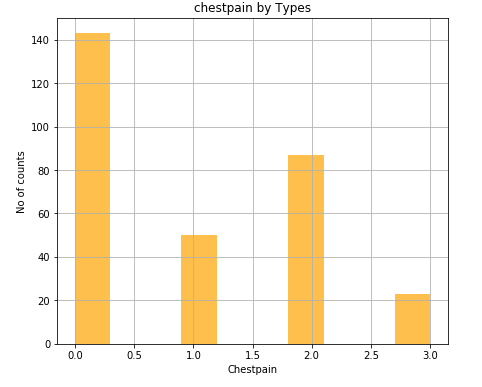


Fig: Chest pain types

Many machine learning algorithms are sensitive to the range and distribution of attribute values in the input data. Outliers in input data can skew and mislead the training process of machine learning algorithms resulting in longer training times, less accurate models and ultimately poorer results.

Even before predictive models are prepared on training data, outliers can result in misleading representations and in turn misleading interpretations of collected data. Outliers can skew the summary distribution of attribute values in descriptive statistics like mean and standard deviation and in plots such as histograms and scatterplots, compressing the body of the data. Finally, outliers can represent examples of data instances that are relevant to the problem such as anomalies in the case of fraud detection and computer security.

It couldn’t fit the model on the training data and can’t say that the model will work accurately for the real data. For this, we must assure that our model got the correct patterns from the data, and it is not getting up too much noise. Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

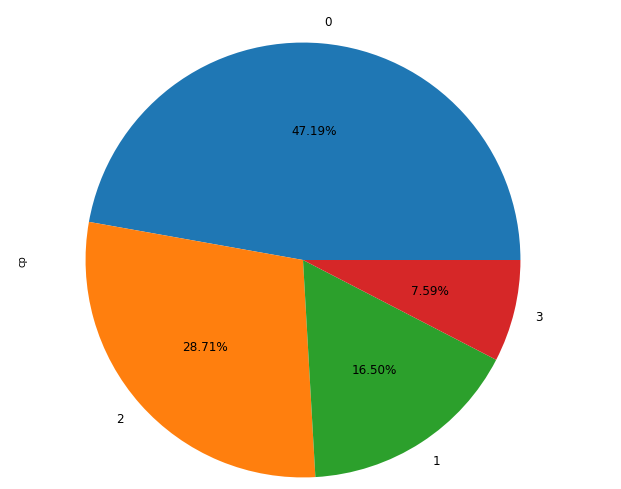
****

Fig: types of chest pain in percentage

**Module-03:**

[**Logistic Regression**](https://en.wikipedia.org/wiki/Logistic_regression)**:**

It is a statistical method for analysing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables. Logistic regression is a Machine Learning classification algorithm that is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.).

In other words, the logistic regression model predicts P(Y=1) as a function of X. Logistic regression Assumptions:

* Binary logistic regression requires the dependent variable to be binary.
* For a binary regression, the factor level 1 of the dependent variable should represent the desired outcome.
* Only the meaningful variables should be included.
* The independent variables should be independent of each other. That is, the model should have little.
* The independent variables are linearly related to the log odds.
* Logistic regression requires quite large sample sizes.

[Decision Tree](https://www.geeksforgeeks.org/decision-tree/):

It is one of the most powerful and popular algorithm. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. Assumptions of Decision tree:

* At the beginning, we consider the whole training set as the root.
* Attributes are assumed to be categorical for information gain, attributes are assumed to be continuous.
* On the basis of attribute values records are distributed recursively.
* We use statistical methods for ordering attributes as root or internal node.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed.

This process is continued on the training set until meeting a termination condition. It is constructed in a top-down recursive divide-and-conquer manner. All the attributes should be categorical. Otherwise, they should be discretized in advance. Attributes in the top of the tree have more impact towards in the classification and they are identified using the information gain concept. A decision tree can be easily over-fitted generating too many branches and may reflect anomalies due to noise or outliers.

**Module-04:**

**Support Vector Machines (SVM):**

A classifier that categorizes the data set by setting an optimal hyper plane between data. I chose this classifier as it is incredibly versatile in the number of different kernelling functions that can be applied and this model can yield a high predictability rate. Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms. They were extremely popular around the time they were developed in the 1990s and continue to be the go-to method for a high-performing algorithm with little tuning.

* How to disentangle the many names used to refer to support vector machines.
* The representation used by SVM when the model is actually stored on disk.
* How a learned SVM model representation can be used to make predictions for new data.
* How to learn an SVM model from training data.
* How to best prepare your data for the SVM algorithm.
* Where you might look to get more information on SVM.

**Random Forest:**

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees’ habit of over fitting to their training set. Random forest is a type of supervised machine learning algorithm based on [ensemble learning](https://en.wikipedia.org/wiki/Ensemble_learning). Ensemble learning is a type of learning where you join different types of algorithms or same algorithm multiple times to form a more powerful prediction model. The [random forest](https://en.wikipedia.org/wiki/Random_forest) algorithm combines multiple algorithm of the same type i.e. multiple decision trees, resulting in a forest of trees, hence the name "Random Forest". The random forest algorithm can be used for both regression and classification tasks.

The following are the basic steps involved in performing the random forest algorithm:

* Pick N random records from the dataset.
* Build a decision tree based on these N records.
* Choose the number of trees you want in your algorithm and repeat steps 1 and 2.
* In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output). The final value can be calculated by taking the average of all the values predicted by all the trees in forest. Or, in case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

**Module-05:**

K-Nearest Neighbor (KNN):

K-Nearest Neighbor is a supervised machine learning algorithm which stores all instances correspond to training data points in n-dimensional space. When an unknown discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors) and returns the most common class as the prediction and for real-valued data it returns the mean of k nearest neighbors. In the distance-weighted nearest neighbor algorithm, it weights the contribution of each of the k neighbors according to their distance using the following query giving greater weight to the closest neighbors.

Usually KNN is robust to noisy data since it is averaging the k-nearest neighbors. The k-nearest-neighbors algorithm is a classification algorithm, and it is supervised: it takes a bunch of labeled points and uses them to learn how to label other points. To label a new point, it looks at the labeled points closest to that new point (those are its nearest neighbors), and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point (the “k” is the number of neighbors it checks). Makes predictions about the validation set using the entire training set. KNN makes a prediction about a new instance by searching through the entire set to find the k “closest” instances. “Closeness” is determined using a proximity measurement (Euclidean) across all features.

Naive Bayes algorithm:

The Naive Bayes algorithm is an intuitive method that uses the probabilities of each attribute belonging to each class to make a prediction. It is the supervised learning approach you would come up with if you wanted to model a predictive modeling problem probabilistically. Naive bayes simplifies the calculation of probabilities by assuming that the probability of each attribute belonging to a given class value is independent of all other attributes. This is a strong assumption but results in a fast and effective method. The probability of a class value given a value of an attribute is called the conditional probability. By multiplying the conditional probabilities together for each attribute for a given class value, we have a probability of a data instance belonging to that class. To make a prediction we can calculate probabilities of the instance belonging to each class and select the class value with the highest probability.

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets. Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

**Module-06:**

Tkinter is a python library for developing GUI (Graphical User Interfaces). We use the tkinter library for creating an application of UI (User Interface), to create windows and all other graphical user interface and Tkinter will come with Python as a standard package, it can be used for security purpose of each users or accountants. There will be two kinds of pages like registration user purpose and login entry purpose of users.

Parameter calculations:

Accuracy calculation:

False Positives (FP): A person who will pay predicted as defaulter. When actual class is no and predicted class is yes. E.g. if actual class says this passenger did not survive but predicted class tells you that this passenger will survive.

False Negatives (FN): A person who default predicted as payer. When actual class is yes but predicted class in no. E.g. if actual class value indicates that this passenger survived and predicted class tells you that passenger will die.

True Positives (TP): A person who will not pay predicted as defaulter. These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes. E.g. if actual class value indicates that this passenger survived and predicted class tells you the same thing.

True Negatives (TN): A person who default predicted as payer. These are the correctly predicted negative values which means that the value of actual class is no and value of predicted class is also no. E.g. if actual class says this passenger did not survive and predicted class tells you the same thing.

It achieved precision, recall, true positive rate (TPR), and false positive rate (FPR) for each classification techniques as it is shown in the above tables and also achieved different interesting confusion matrix for each classification techniques and we can see the classification performance of each classifiers by the help of confusion matrix. We use a confusion matrix to compute the accuracy rate of each severity class. For each class, it demonstrates how instances from that class receive the various classifications. Here in the next table we have shown instances that are correctly classified and incorrectly classified in accordance with overall accuracy of each classification techniques. All classifiers perform similarly well with respect to the number of correctly classified instances.

Comparing Algorithm with prediction in the form of best accuracy result:

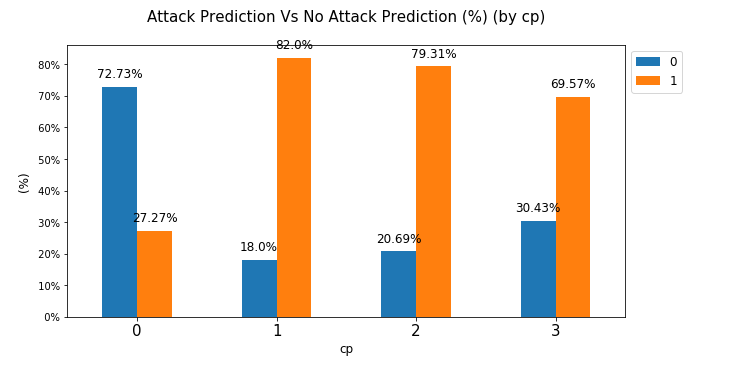
It is important to compare the performance of multiple different machine learning algorithms consistently and it will discover to create a test harness to compare multiple different machine learning algorithms in Python with scikit-learn. It can use this test harness as a template on your own machine learning problems and add more and different algorithms to compare. Each model will have different performance characteristics. Using resampling methods like cross validation, you can get an estimate for how accurate each model may be on unseen data. It needs to be able to use these estimates to choose one or two best models from the suite of models that you have created. When have a new dataset, it is a good idea to visualize the data using different techniques in order to look at the data from different perspectives. The same idea applies to model selection. You should use a number of different ways of looking at the estimated accuracy of your machine learning algorithms in order to choose the one or two to finalize. A way to do this is to use different visualization methods to show the average accuracy, variance and other properties of the distribution of model accuracies. 

Fig. Attack prediction by types

In the next section you will discover exactly how you can do that in Python with scikit-learn. The key to a fair comparison of machine learning algorithms is ensuring that each algorithm is evaluated in the same way on the same data and it can achieve this by forcing each algorithm to be evaluated on a consistent test harness.

In the example below 4 different algorithms are compared:

* Logistic Regression
* Random Forest
* Decision tree
* Support Vector Machines
* Now, the dimensions of new features in a numpy array called ‘n’ and it want to predict the species of this features and to do using the predict method which takes this array as input and spits out predicted target value as output.
* So, the predicted target value comes out to be 0. Finally to find the test score which is the ratio of no. of predictions found correct and total predictions made and finding accuracy score method which basically compares the actual values of the test set with the predicted values.

Sensitivity:

Sensitivity is a measure of the proportion of actual positive cases that got predicted as positive (or true positive). Sensitivity is also termed as Recall. This implies that there will be another proportion of actual positive cases, which would get predicted incorrectly as negative (and, thus, could also be termed as the false negative). This can also be represented in the form of a false negative rate. The sum of sensitivity and false negative rate would be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Sensitivity is a measure of the proportion of people suffering from the disease who got predicted correctly as the ones suffering from the disease. In other words, the person who is unhealthy actually got predicted as unhealthy.

Mathematically, sensitivity can be calculated as the following:

Sensitivity = (True Positive) / (True Positive + False Negative)

The following is the details in relation to True Positive and False Negative used in the above equation.

* True Positive = Persons predicted as suffering from the disease (or unhealthy) are actually suffering from the disease (unhealthy); In other words, the true positive represents the number of persons who are unhealthy and are predicted as unhealthy.
* False Negative = Persons who are actually suffering from the disease (or unhealthy) are actually predicted to be not suffering from the disease (healthy). In other words, the false negative represents the number of persons who are unhealthy and got predicted as healthy. Ideally, we would seek the model to have low false negatives as it might prove to be life-threatening or business threatening.

The higher value of sensitivity would mean higher value of true positive and lower value of false negative. The lower value of sensitivity would mean lower value of true positive and higher value of false negative. For healthcare and financial domain, models with high sensitivity will be desired.

Specificity:

Specificity is defined as the proportion of actual negatives, which got predicted as the negative (or true negative). This implies that there will be another proportion of actual negative, which got predicted as positive and could be termed as false positives. This proportion could also be called a false positive rate. The sum of specificity and false positive rate would always be 1. Let's try and understand this with the model used for predicting whether a person is suffering from the disease. Specificity is a measure of the proportion of people not suffering from the disease who got predicted correctly as the ones who are not suffering from the disease. In other words, the person who is healthy actually got predicted as healthy is specificity.

Mathematically, specificity can be calculated as the following:

Specificity = (True Negative) / (True Negative + False Positive)

The following is the details in relation to True Negative and False Positive used in the above equation.

* True Negative = Persons predicted as not suffering from the disease (or healthy) are actually found to be not suffering from the disease (healthy); In other words, the true negative represents the number of persons who are healthy and are predicted as healthy.
* False Positive = Persons predicted as suffering from the disease (or unhealthy) are actually found to be not suffering from the disease (healthy). In other words, the false positive represents the number of persons who are healthy and got predicted as unhealthy.

The higher value of specificity would mean higher value of true negative and lower false positive rate. The lower value of specificity would mean lower value of true negative and higher value of false positive.

Prediction result by accuracy:

Logistic regression algorithm also uses a linear equation with independent predictors to predict a value. The predicted value can be anywhere between negative infinity to positive infinity. We need the output of the algorithm to be classified variable data. Higher accuracy predicting result is logistic regression model by comparing the best accuracy.

True Positive Rate(TPR) = TP / (TP + FN)

False Positive rate(FPR) = FP / (FP + TN)

Accuracy: The Proportion of the total number of predictions that is correct otherwise overall how often the model predicts correctly defaulters and non-defaulters.

Accuracy calculation:

Accuracy = (TP + TN) / (TP + TN + FP + FN)

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost same.

Precision: The proportion of positive predictions that are actually correct. (When the model predicts default: how often is correct?)

Precision = TP / (TP + FP)

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. The question that this metric answer is of all passengers that labeled as survived, how many actually survived? High precision relates to the low false positive rate. We have got 0.788 precision which is pretty good.

Recall: The proportion of positive observed values correctly predicted. (The proportion of actual defaulters that the model will correctly predict)

Recall = TP / (TP + FN)

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to the all observations in actual class - yes.

F1 Scoreis the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it’s better to look at both Precision and Recall.

General Formula:

F- Measure = 2TP / (2TP + FP + FN)

F1-Score Formula:

F1 Score = 2\*(Recall \* Precision) / (Recall + Precision)

**Used Python Packages:**

**sklearn:**

* + In python, sklearn is a machine learning package which include a lot of ML algorithms.
  + Here, we are using some of its modules like train\_test\_split, DecisionTreeClassifier or Logistic Regression and accuracy\_score.

**NumPy:**

* + It is a numeric python module which provides fast maths functions for calculations.
  + It is used to read data in numpy arrays and for manipulation purpose.

**Pandas:**

* + Used to read and write different files.
  + Data manipulation can be done easily with data frames.

**Matplotlib:**

* + Data visualization is a useful way to help with identify the patterns from given dataset.
  + Data manipulation can be done easily with data frames.

**tkinter:**

* + Standard python interface to the GUI toolkit.
  + Accessible to everybody and reusable in various contexts.

**Work flow diagram**

Source Data

Data Processing and Cleaning

Testing Dataset

Training Dataset

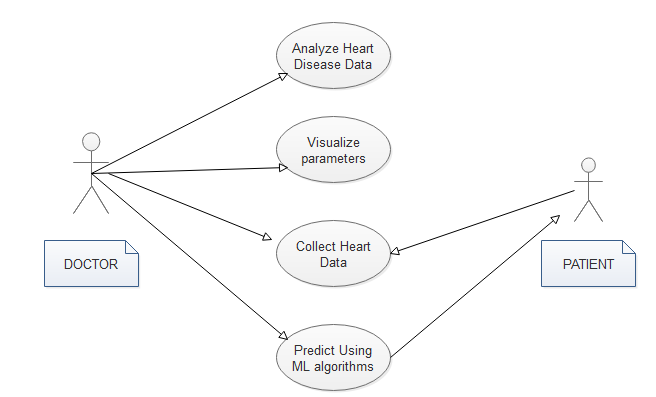
Best Model by Accuracy

Classification ML Algorithms

Heart disease prediction by accuracy result

Fig: Workflow Diagram

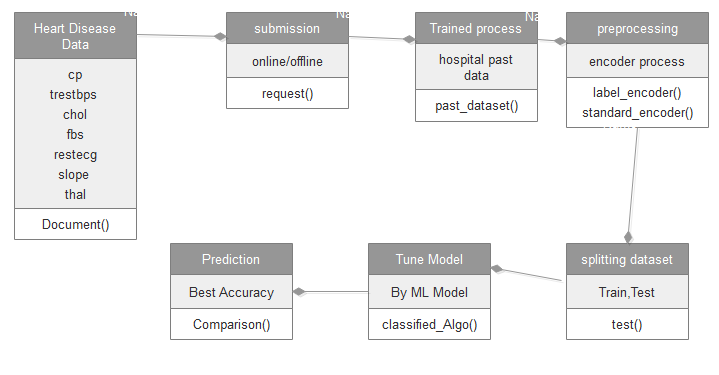
**Use Case Diagram**:



**Explanation**:

Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So we can say that uses cases are nothing but the system functionalities written in an organized manner. Now the second things which are relevant to the use cases are the actors. Actors can be defined as something that interacts with the system. The actors can be human user, some internal applications or may be some external applications. Functionalities to be represented as a use case, Actors and Relationships among the use cases and actors and the name of a use case is very important.

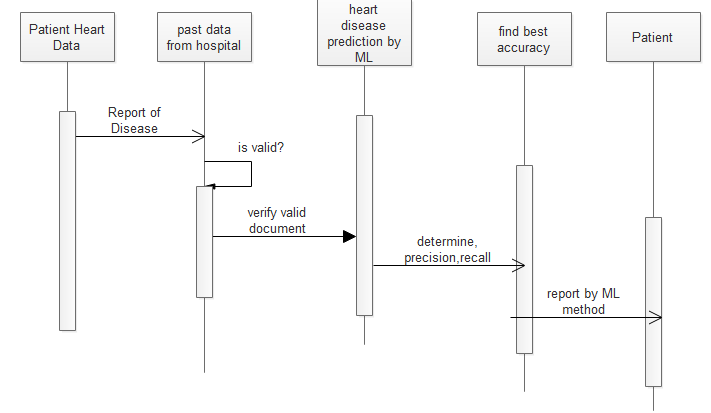
**Class Diagram**:



**Explanation**:

Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified. Because unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram. Because at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

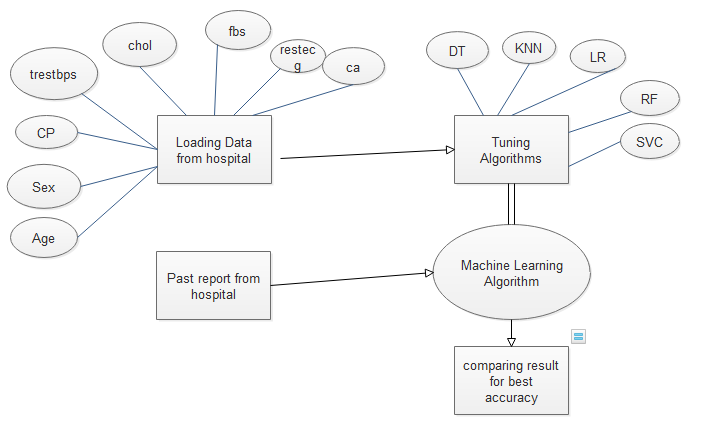
**Sequence Diagram**:



**Explanation**:

UML sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modeling, which focuses on identifying the behavior within your system. Other dynamic modeling techniques include [activity diagramming](http://agilemodeling.com/artifacts/activityDiagram.htm), Sequence diagrams, along with [class diagrams](http://agilemodeling.com/artifacts/classDiagram.htm) and [physical data models](http://agiledata.org/essays/dataModeling101.html) are in my opinion the most important design-level models for modern business application development.

**Entity Relationship Diagram (ERD):**



An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a [data modeling](https://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and be used as the foundation for a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

**Sample Code:**

#import libraries for access and functional purpose

import pandas as p

import numpy as n

import matplotlib.pyplot as plt

import seaborn as s

#read the given dataset

df = p.read\_csv("dataset.csv")

df.head()

chestpains = p.Categorical(df['cp'])

chestpains

df['cp'].value\_counts()

df['cp'].nunique()

df['cp'].unique()

df.shape

#To describe the dataframe

df.describe()

#Checking datatype and information about dataset

df.info()

#find sum of duplicate data

sum(df.duplicated())

#Checking minimum or maximum trestbps

print("Minimum Trestbps:", df["trestbps"].min())

print("Maximun Trestbps:", df["trestbps"].max())

#Checking minimum or maximum cholestrol

print("Minimum Cholestrol:", df["chol"].min())

print("Maximun Cholestrol:", df["chol"].max())

#Rename the data

df.rename(columns={'cp':'CPTYPE'}, inplace=True)

df.rename(columns={'chol':'CHOLESTROL'}, inplace=True)

df.rename(columns={'ca':'CARDIACARREST'}, inplace=True)

#show the dataframe

df.head()

df['CARDIACARREST'].unique()

**RESULT AND DISCUSSION**

**Software involvement steps:**

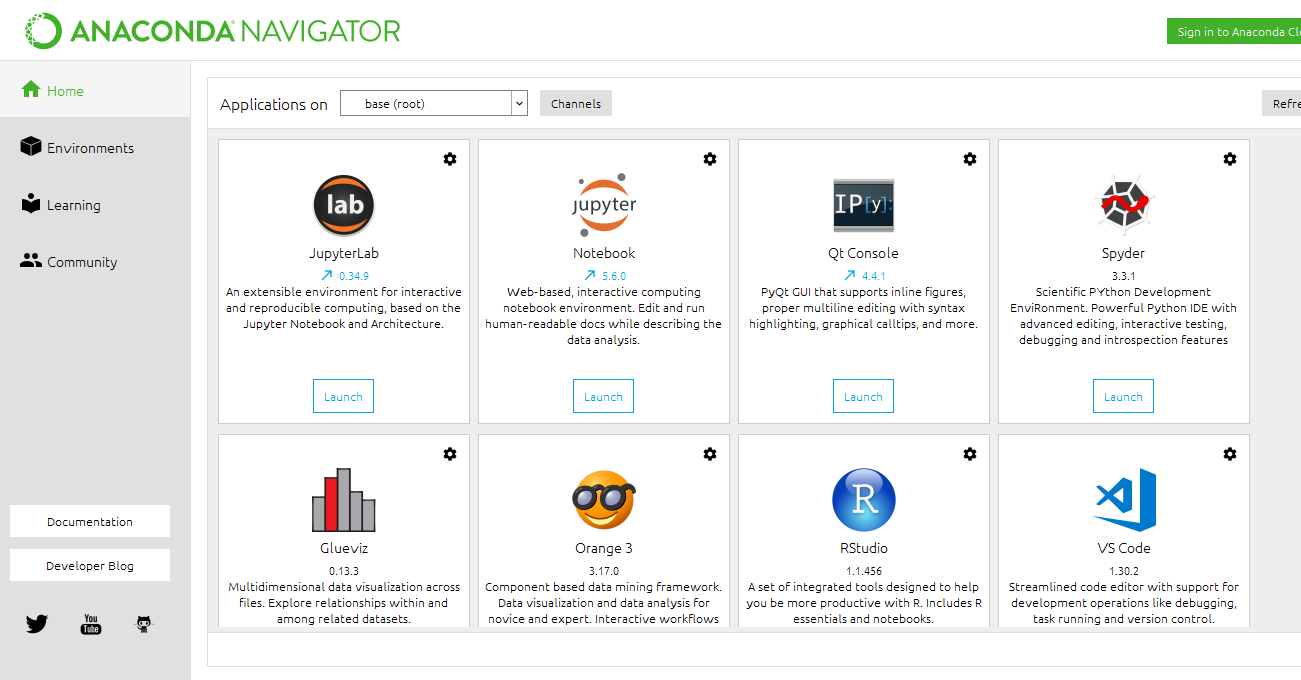
****

Fig: Open the anaconda navigator

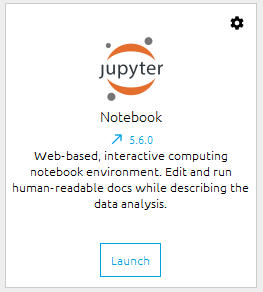
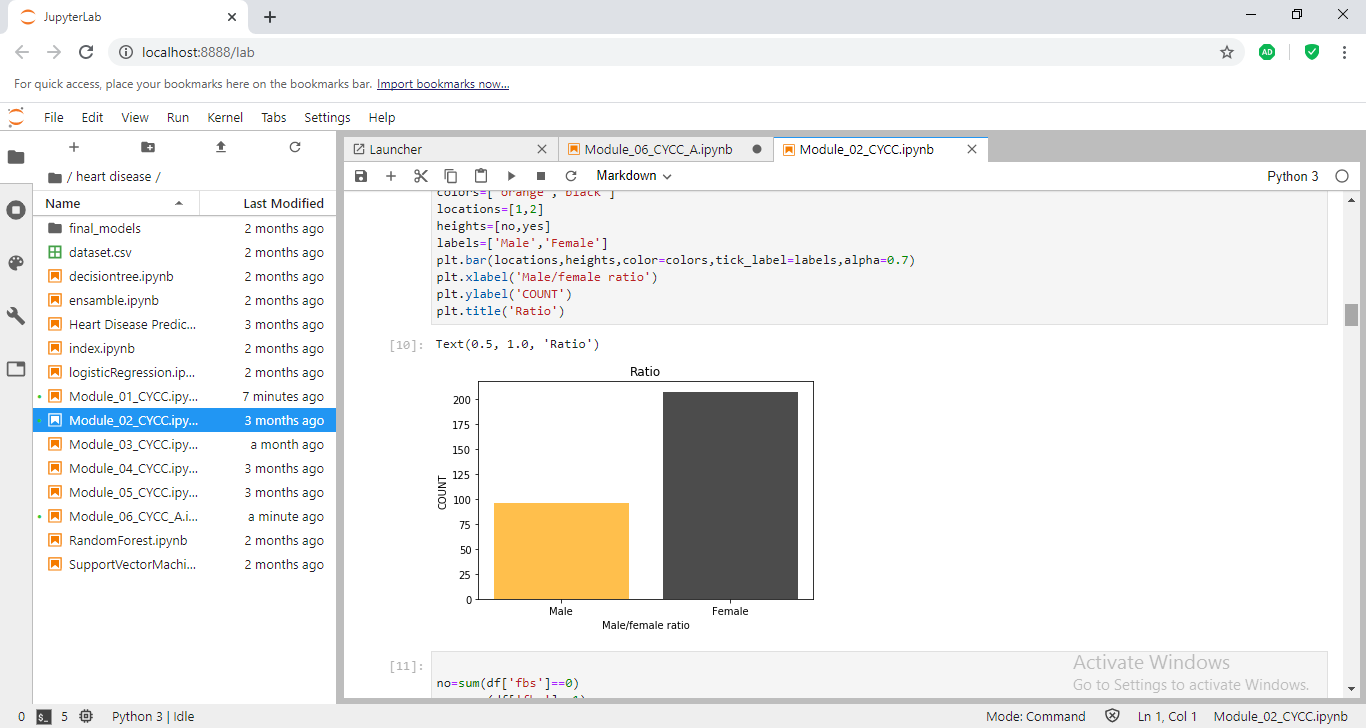
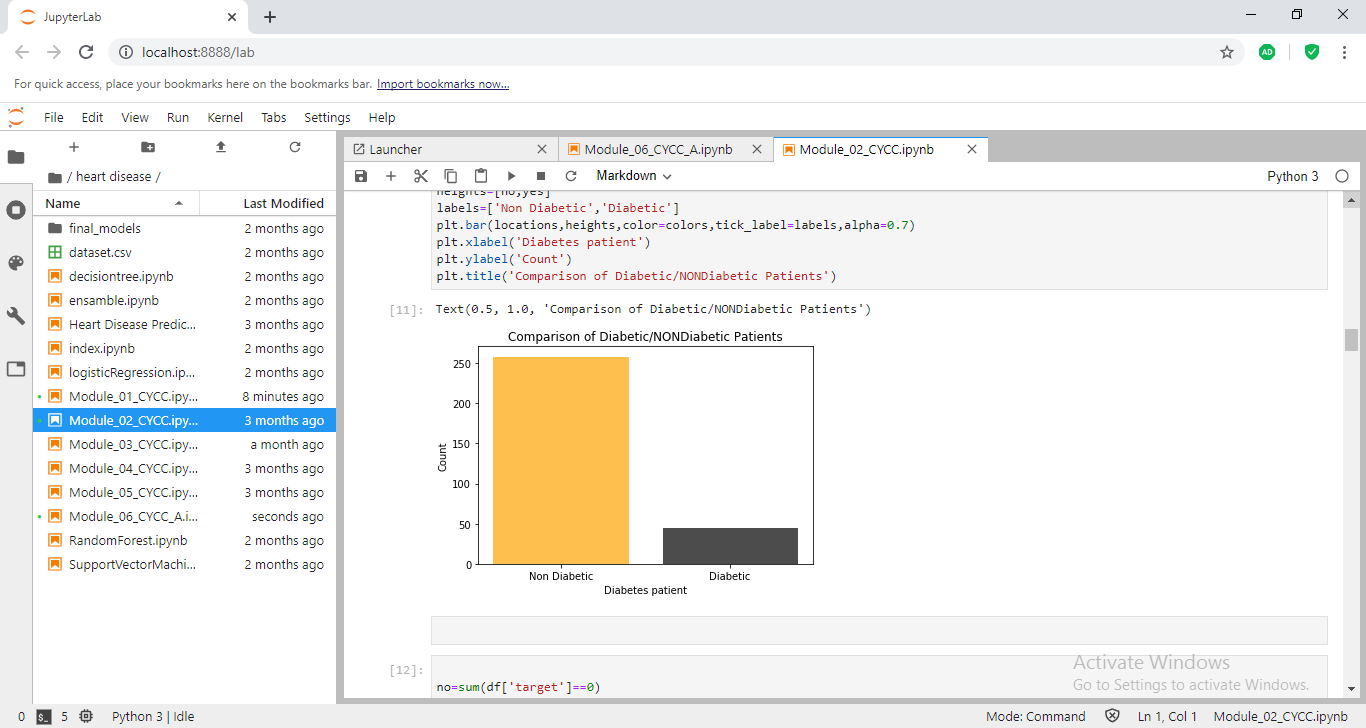
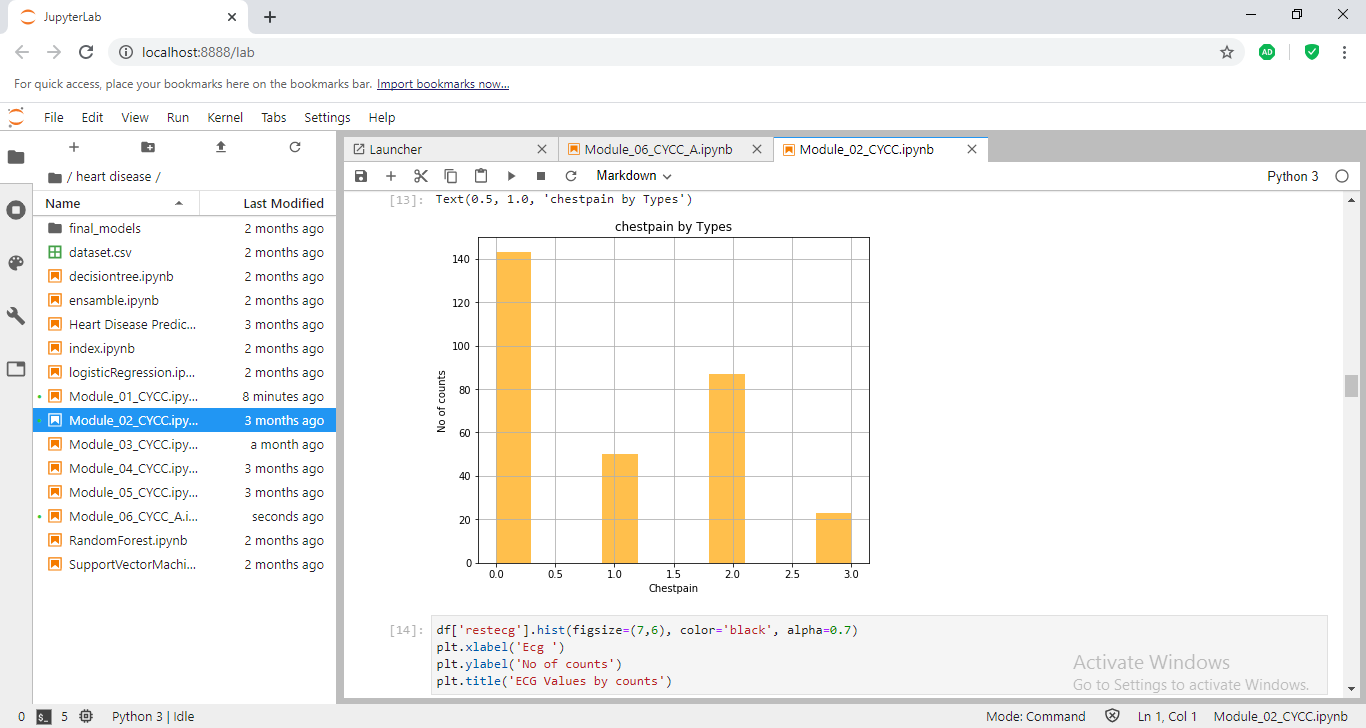
****

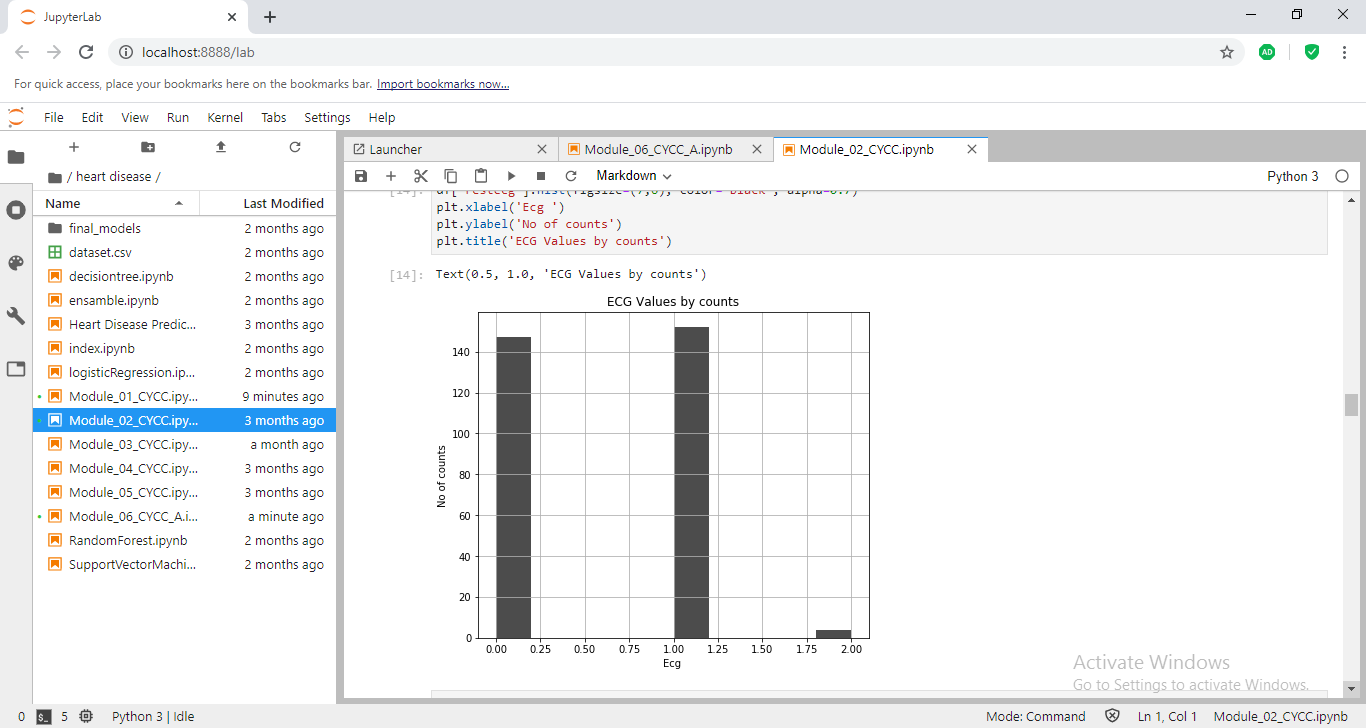
Fig: Launch the jupyter notebook platform

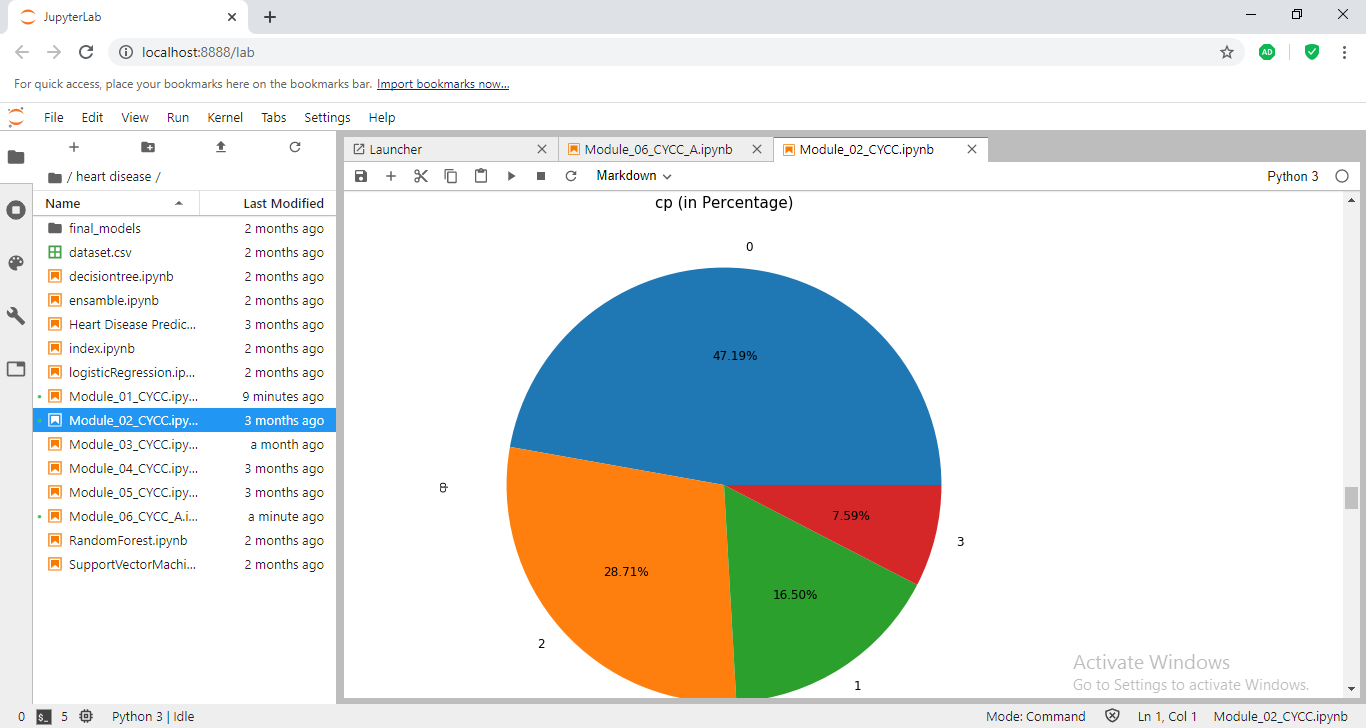
**Visualization**

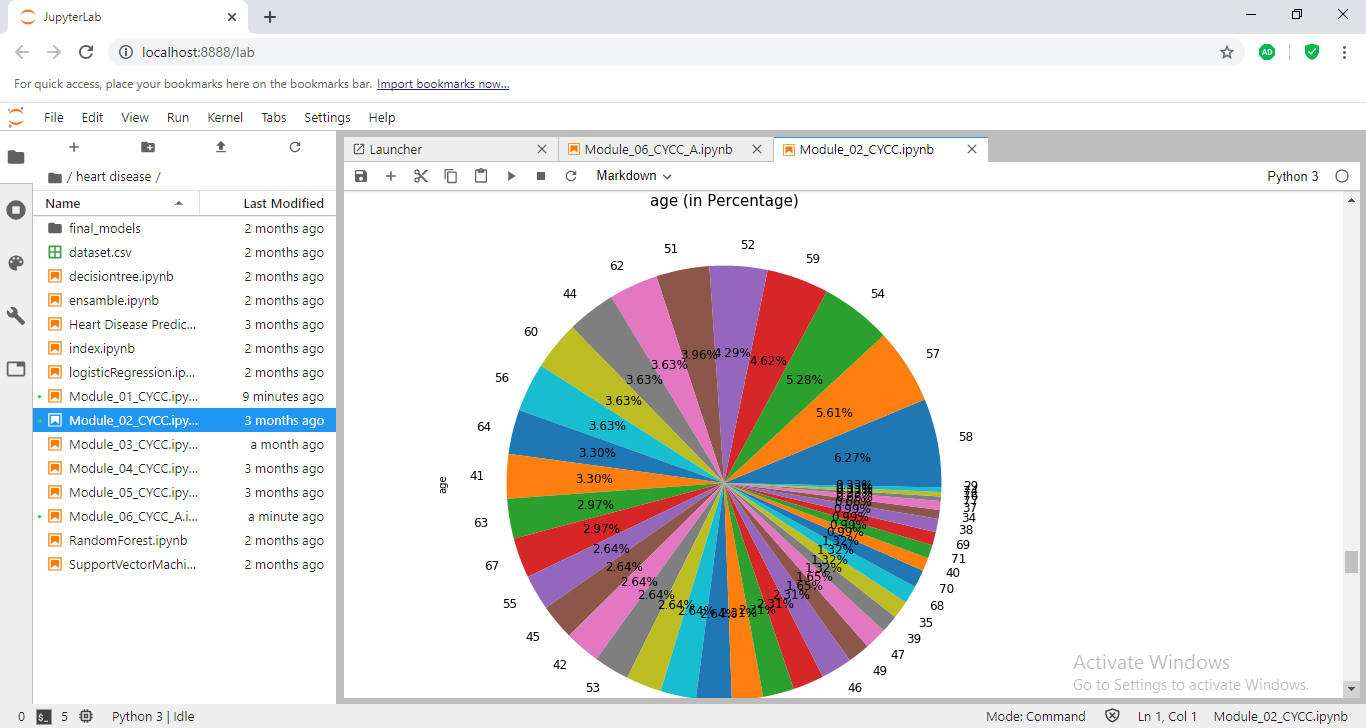


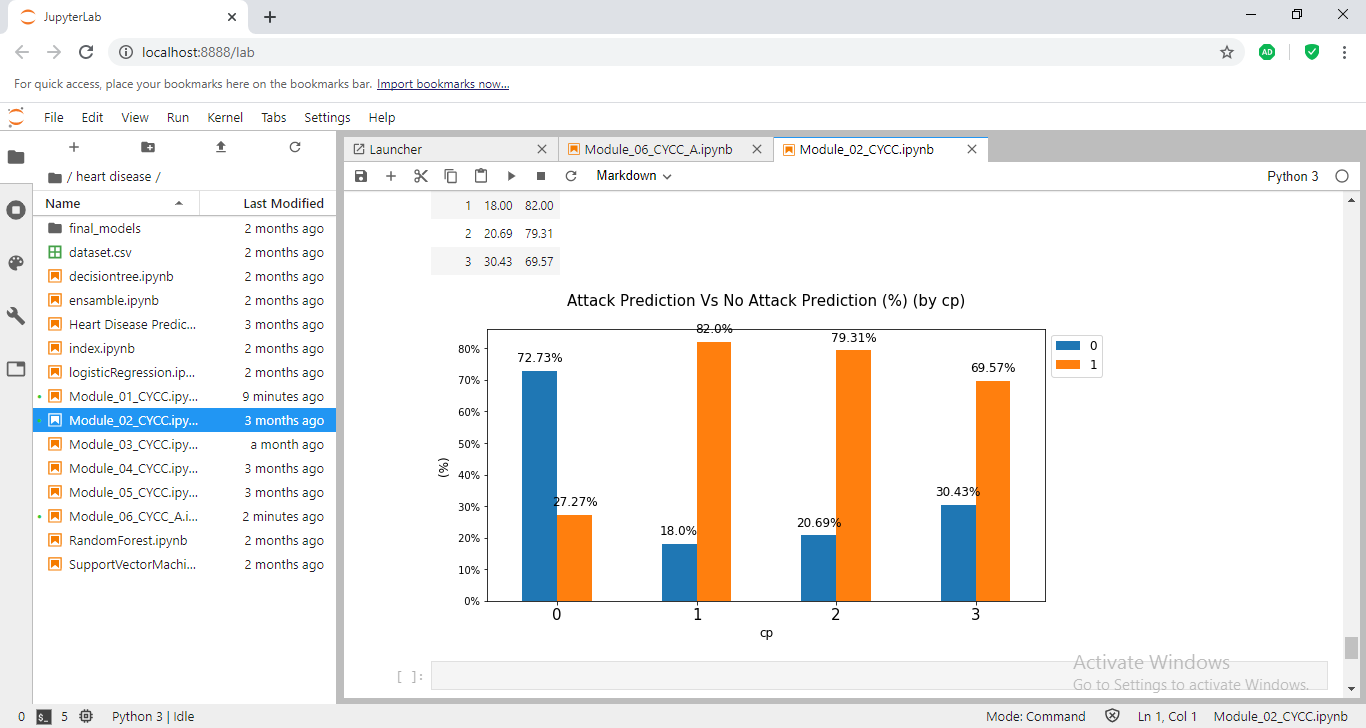












**Performance of Machine Learning parameters:**

Comparison of accuracy results of crop yield prediction:

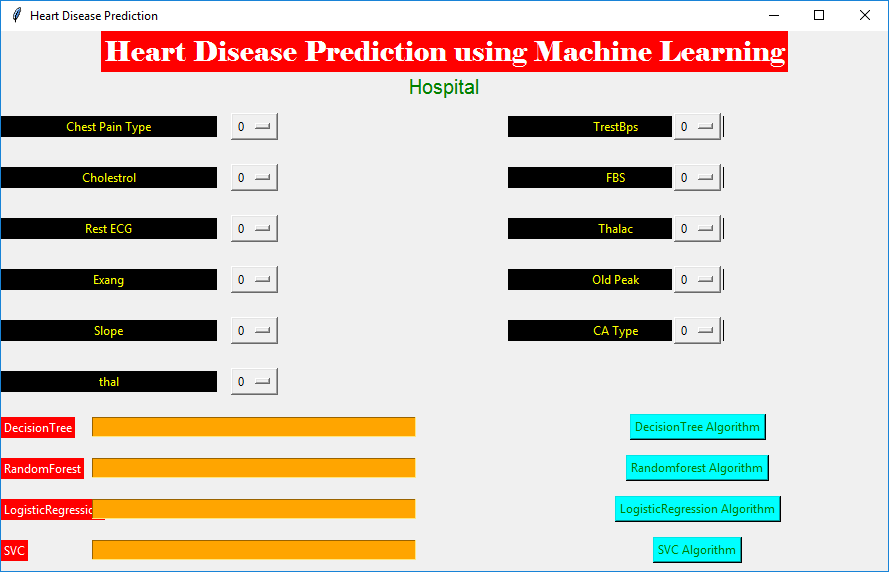
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters** | **LR** | **DT** | **RF** | **SVC** |
| **Precision** | 0.96 | 1 | 1 | 0.62 |
| **Recall** | 0.96 | 1 | 1 | 1 |
| **F1-Score** | 0.96 | 1 | 1 | 0.77 |
| **Sensitivity** | 0.95 | 1 | 1 | 1 |
| **Specificity** | 0.92 | 1 | 1 | 0 |
| **Accuracy (%)** | 94.73 | 100 | 100 | 62.06 |

Comparison of accuracy results of crop yield cost prediction:

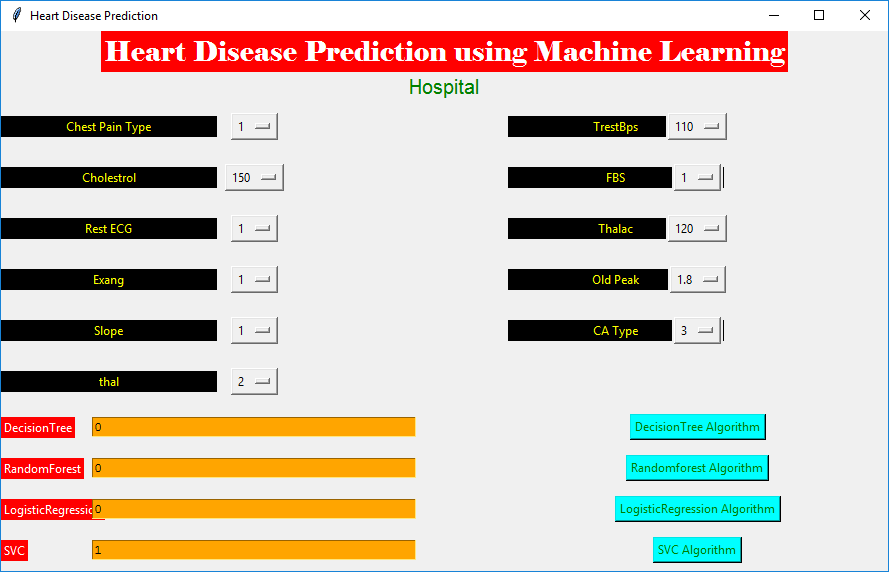
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Parameters** | **LR** | **DT** | **RF** | **SVC** |
| **Precision** | 0.70 | 1 | 1 | 0.69 |
| **Recall** | 0.88 | 1 | 1 | 1 |
| **F1-Score** | 0.78 | 1 | 1 | 0.82 |
| **Sensitivity** | 0.87 | 1 | 1 | 1 |
| **Specificity** | 0.13 | 1 | 1 | 0 |
| **Accuracy (%)** | 65.17 | 100 | 100 | 69.36 |

**Testing Results:**

INPUT:



OUTPUT:



**Conclusion:**

The analytical process started from data cleaning and processing, missing value, exploratory analysis and finally model building and evaluation. Finally we predict the heart disease using machine learning algorithm with different results. This brings some of the following insights about heart disease prediction. As maximum types of dataset will be covered under this system, doctor may get to know about the disease exactly using ML algorithms, it helps the doctor in decision making weather patient has heart disease or not.

**Future Work:**

* Remaining SMLT algorithms will be involve to finding the best accuracy with applying to predict the crop yield and cost.
* Hospitals wants to automate the detecting the heart disease from eligibility process (real time).
* To automate this process by show the prediction result in web application or desktop application.
* To optimize the work to implement in Artificial Intelligence environment.