**ABSTRACT:**

Heart plays significant role in living organisms. Diagnosis and prediction of heart related diseases requires more precision, perfection and correctness because a little mistake can cause fatigue problem or death of the person, there are numerous death cases related to heart and their counting is increasing exponentially day by day. To deal with the problem there is essential need of prediction system for awareness about diseases. Machine learning is the branch of Artificial Intelligence(AI), it provides prestigious support in predicting any kind of event which take training from natural events. In this paper, we calculate accuracy of machine learning algorithms for predicting heart disease, for this algorithms are k-nearest neighbor, decision tree, linear regression and support vector machine(SVM) by using UCI repository dataset for training and testing. For implementation of Python programming Anaconda(jupytor) notebook is best tool, which have many type of library, header file, that make the work more accurate and precise.

**INTRODUCTION**

Heart is one of the most extensive and vital organ of human body so the care of heart is essential. Most of diseases are related to heart so the prediction about heart diseases is necessary and for this purpose comparative study needed in this field, today most of patient are died because their diseases are recognized at last stage due to lack of accuracy of instrument so there is need to know about the more efficient algorithms for diseases prediction. Machine Learning is one of the efficient technology for the testing, which is based on training and testing. It is the branch of Artificial Intelligence(AI) which is one of broad area of learning where machines emulating human abilities, machine learning is a specific branch of AI. On the other hand machines learning systems are trained to learn how to process and make use of data hence the combination of both technology is also called as Machine Intelligence. As the definition of machine learning, it learns from the natural phenomenon, natural things so in this project we uses the biological parameter as testing data such as cholesterol, Blood pressure, sex, age, etc. and on the basis of these, comparison is done in the terms of accuracy of algorithms such as in this project we have used four algorithms which are decision tree, linear regression, k-neighbour, SVM. In this paper, we calculate the accuracy of four different machine learning approaches and on the basis of calculation we conclude that which one is best among them.

**LITERATURE SURVEY**

Heart is one of the core organ of human body, it play crucial role on blood pumping in human body which is as essential as the oxygen for human body so there is always need of protection of it, this is one of the big reasons for the researchers to work on this. So there are number of researchers working on it .There is always need of analysis of heart related things either diagnosis or prediction or you can say that protection of heart disease .There are various fields like artificial intelligence, machine learning, data mining that contributed on this work . Performance of any algorithms depends on variance and biasness of dataset[4]. As per research on the machine learning for prediction of heart diseases

* Himanshu et al.[4] naive bayes perform well with low variance and high biasness as compare to high variance and low biasness which is knn. With low biasness and high variance knn suffers from the problem of over fitting this is the reason why performance of knn get decreased. There are various advantage of using low variance and high biasness because as the dataset small it take less time for training as well as testing od algorithm but there also some disadvantages of using small size of dataset. When the dataset size get increasing the asymptotic errors are get introduced and low biasness, low variance based algorithms play well in this type of cases. Decision tree is one of the nonparametric machine learning algorithm but as we know it suffers from the problem over fitting but it cloud be solve by some over fitting removable techniques. Support vector machine is algebraic and statics background algorithm, it construct a linear separable n-dimensional hyper plan for the classification of datasets. The nature of heart is complex, there is need of carefully handling of it otherwise it cause death of the person. The severity of heart diseases is classified based on various methods like knn, decision tree, generic algorithm and naïve bayes [3].
* Mohan et al.[3] define how you can combine two different approaches to make a single approach called hybrid approach which have the accuracy 88.4% which is more than of all other. Some of the researchers have worked on data mining for the prediction of heart diseases.
* Kaur et al.[6] have worked on this and define how the interesting pattern and knowledge arederived from the large dataset. They perform accuracy comparison on various machine learning and data mining approaches for finding which one is best among then and get the result on the favor of svm.
* Kumar et al.[5] have worked on various machine learning and data mining algorithms and analysis of these algorithms are trained by UCI machine learning dataset which have 303 samples with 14 input feature and found svm is best among them, here other different algorithms are naivy bayes, knn and decision tree.
* Gavhane et al.[1] have worked on the multi layer perceptron model for the prediction of heart diseases in human being and the accuracy of the algorithm using CAD technology. If the number of person using the prediction system for their diseases prediction then the awareness about the diseases is also going to increases and it make reduction in the death rate of heart patient. Some researchers have work on one or two algorithm for predication diseases.
* Krishnan et al.[2] proved that decision tree is more accurate as compare to the naïve bayes classification algorithm in their project. Machine learning algorithms are used for various type of diseases predication and many of the researchers have work on this like
* Kohali et al.[7] work on heart diseases prediction using logistic regression, diabetes prediction using support vector machine, breast cancer prediction using Adaboot classifier and concluded that the logistic regression give the accuracy of 87.1%, support vector machine give the accuracy of 85.71%, Adaboot classifier give the accuracy up to 98.57% which good for predication point of view. A survey paper on heart diseases predication have proven that the old machine learning algorithms does not perform good accuracy for the predication while hybridization perform good and give better accuracy for the predication[8].

1)**Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques**

**AUTHORS:** Senthil kumar mohan, chandrasegar thirumalai and Gautam Srivastva

Heart disease is one of the most significant causes of mortality in the world today. Prediction of cardiovascular disease is a critical challenge in the area of clinical data analysis. Machine learning (ML) has been shown to be effective in assisting in making decisions and predictions from the large quantity of data produced by the healthcare industry. We have also seen ML techniques being used in recent developments in different areas of the Internet of Things (IoT). Various studies give only a glimpse into predicting heart disease with ML techniques. In this paper, we propose a novel method that aims at finding significant features by applying machine learning techniques resulting in improving the accuracy in the prediction of cardiovascular disease. The prediction model is introduced with different combinations of features and several known classification techniques. We produce an enhanced performance level with an accuracy level of 88.7% through the prediction model for heart disease with the hybrid random forest with a linear model (HRFLM).

2)**Prediction of Heart Disease using Machine Learning Algorithms: A Survey**

**AUTHORS:**  Himanshu Sharma and M A Rizvi

Health care field has a vast amount of data, for processing those data certain techniques are used. Data mining is one of the techniques often used. Heart disease is the Leading cause of death worldwide. This System predicts the arising possibilities of Heart Disease. The outcomes of this system provide the chances of occurring heart disease in terms of percentage. The datasets used are classified in terms of medical parameters. This system evaluates those parameters using data mining classification technique. The datasets are processed in python programming using two main Machine Learning Algorithm namely Decision Tree Algorithm and Naive Bayes Algorithm which shows the best algorithm among these two in terms of accuracy level of heart disease.

3)**Prediction of Heart Diseases Using Data Mining and Machine Learning Algorithms and Tools**

**AUTHORS:** M. Nikhil Kumar, K. V. S. Koushik, K. Deepak

Objectives: The objective of our work is to analyse various data mining tools and techniques in health care domain that can be employed in prediction of heart disease system and their efficient diagnosis. Methods/Statistical Analysis: A heart disease prediction model, which implements data mining technique, can help the medical practitioners in detecting the heart disease status based on the patient's clinical data. Data mining classification techniques for good decision making in the field of health care addressed are namely Decision trees, Naive Bayes, Neural Networks and Support Vector Machines. Hybridizing or combining any of these algorithms helps to make decisions quicker and more precise. Findings: Data mining is a powerful new technology for the extraction of hidden predictive and actionable information from large databases that can be used to gain deep and novel insights. Using advanced data mining techniques to excavate valuable information, has been considered as an activist approach to improve the quality and accuracy of healthcare service while lowering the healthcare cost and diagnosis time. Using this technique presence of heart disease can be predicted accurately. Using more input attributes such as controllable and uncontrollable risk factors, more accurate results could be achieved. Applications/Improvements: This method can be further expanded. It can use many of input attributes. Other data mining techniques are also be used for predication such as Clustering, Time series, Association rules. The unstructured data available in healthcare industry database can also be mined using text mining.

4)**Application of Machine Learning in Diseases Prediction**

**AUTHORS:**Pahulpreet Singh Kohli and Shriya Arora

The application of machine learning in the field of medical diagnosis is increasing gradually. This can be contributed primarily to the improvement in the classification and recognition systems used in disease diagnosis which is able to provide data that aids medical experts in early detection of fatal diseases and therefore, increase the survival rate of patients significantly. In this paper, we apply different classification algorithms, each with its own advantage on three separate databases of disease (Heart, Breast cancer, Diabetes) available in UCI repository for disease prediction. The feature selection for each dataset was accomplished by backward modeling using the p-value test. The results of the study strengthen the idea of the application of machine learning in early detection of diseases.

5)**An Extensive Review on Swarm Robotics**

**AUTHORS:**S. Kumra, R. Saxena, and S. Mehta.,

Swarm robotics is a new approach to the coordination of multi-robot systems which consist of large numbers of relatively simple robots which takes its inspiration from social insects. The most remarkable characteristic of swarm robots are the ability to work cooperatively to achieve a common goal. In this paper, classification of existing researches, problems and algorithms aroused in the study of swarm robotics are presented. The existing studies are classified into major areas and relevant sub-categories in the major areas.

**IMPLEMENTATION**

In this project student want to detect heart disease from dataset using Bio Inspired 4 features optimizing algorithms such as Genetic Algorithm, Bat, Bee and ACO. Here ACO algorithm is design in python to solve Travelling Salesman Problem to find shortest path and it cannot be implemented with heart disease dataset, so I am implementing 3 algorithms called Genetic, Bat and Bee.

Bio inspired algorithms design to optimized features used in dataset for training classification algorithms to increase prediction accuracy, sometime some datasets may have irrelevant values inside dataset and those irrelevant attributes or values may degrade classification accuracy so using optimize algorithms we can reduce features (attribute values) from dataset. This optimize algorithms will be applied on dataset to check whether all values are related to dataset or not, if any attribute found unrelated then it will removed from dataset.

To implement this algorithms I am using Heart disease dataset which contains 14 attributes and 4 class labels where 0 refers to No heart Disease and 1 refers to stage1 disease and 2 and 3 refers stage 3 and 4 disease.

Below are some values from dataset to train algorithms

age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal,class

63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0,2.3,3.0,0.0,6.0,0

67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0,1.5,2.0,3.0,3.0,2

67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0,2.6,2.0,2.0,7.0,1

37.0,1.0,3.0,130.0,250.0,0.0,0.0,187.0,0.0,3.5,3.0,0.0,3.0,0

First records contains dataset column names and remaining records are the values of dataset. In last column we have class values as 0, 2, 1 and 3 as disease stage.

Test dataset also contains record values but it will not have class labels and application will apply that test values on train dataset to predict it class labels. Some values from test dataset.

age,sex,cp,trestbps,chol,fbs,restecg,thalach,exang,oldpeak,slope,ca,thal

63.0,1.0,1.0,145.0,233.0,1.0,2.0,150.0,0.0,2.3,3.0,0.0,6.0

67.0,1.0,4.0,160.0,286.0,0.0,2.0,108.0,1.0,1.5,2.0,3.0,3.0

67.0,1.0,4.0,120.0,229.0,0.0,2.0,129.0,1.0,2.6,2.0,2.0,7.0

In above test dataset we can see there is no class name and application will predict it. All this files are available inside ‘heart\_dataset’ folder.

In that ‘heart\_dataset’ folder I kept dataset URL and information of dataset for references.

**ALGORITHMS**

What is Machine Learning?

Machine Learning (ML) is that field of computer science with the help of which computer systems can provide sense to data in much the same way as human beings do.

In simple words, ML is a type of artificial intelligence that extract patterns out of raw data by using an algorithm or method. The main focus of ML is to allow computer systems learn from experience without being explicitly programmed or human intervention.

Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

Why & When to Make Machines Learn?

We have already discussed the need for machine learning, but another question arises that in what scenarios we must make the machine learn? There can be several circumstances where we need machines to take data-driven decisions with efficiency and at a huge scale. The followings are some of such circumstances where making machines learn would be more effective −

Lack of human expertise

The very first scenario in which we want a machine to learn and take data-driven decisions, can be the domain where there is a lack of human expertise. The examples can be navigations in unknown territories or spatial planets.

Dynamic scenarios

There are some scenarios which are dynamic in nature i.e. they keep changing over time. In case of these scenarios and behaviors, we want a machine to learn and take data-driven decisions. Some of the examples can be network connectivity and availability of infrastructure in an organization.

Difficulty in translating expertise into computational tasks

There can be various domains in which humans have their expertise,; however, they are unable to translate this expertise into computational tasks. In such circumstances we want machine learning. The examples can be the domains of speech recognition, cognitive tasks etc.

Machine Learning Model

Before discussing the machine learning model, we must need to understand the following formal definition of ML given by professor Mitchell −

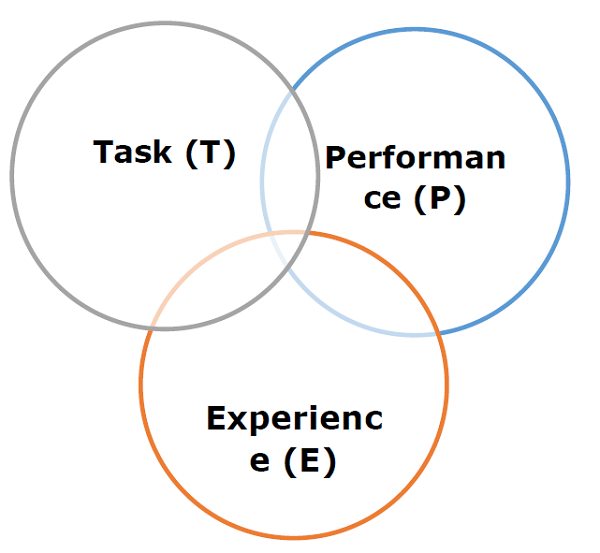
“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

The above definition is basically focusing on three parameters, also the main components of any learning algorithm, namely Task(T), Performance(P) and experience (E). In this context, we can simplify this definition as −

ML is a field of AI consisting of learning algorithms that −

* Improve their performance (P)
* At executing some task (T)
* Over time with experience (E)

Based on the above, the following diagram represents a Machine Learning Model −



Let us discuss them more in detail now −

Task(T)

From the perspective of problem, we may define the task T as the real-world problem to be solved. The problem can be anything like finding best house price in a specific location or to find best marketing strategy etc. On the other hand, if we talk about machine learning, the definition of task is different because it is difficult to solve ML based tasks by conventional programming approach.

A task T is said to be a ML based task when it is based on the process and the system must follow for operating on data points. The examples of ML based tasks are Classification, Regression, Structured annotation, Clustering, Transcription etc.

Experience (E)

As name suggests, it is the knowledge gained from data points provided to the algorithm or model. Once provided with the dataset, the model will run iteratively and will learn some inherent pattern. The learning thus acquired is called experience(E). Making an analogy with human learning, we can think of this situation as in which a human being is learning or gaining some experience from various attributes like situation, relationships etc. Supervised, unsupervised and reinforcement learning are some ways to learn or gain experience. The experience gained by out ML model or algorithm will be used to solve the task T.

Performance (P)

An ML algorithm is supposed to perform task and gain experience with the passage of time. The measure which tells whether ML algorithm is performing as per expectation or not is its performance (P). P is basically a quantitative metric that tells how a model is performing the task, T, using its experience, E. There are many metrics that help to understand the ML performance, such as accuracy score, F1 score, confusion matrix, precision, recall, sensitivity etc.

Challenges in Machines Learning

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

Applications of Machines Learning

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping.

An Introduction to Python

Python is a popular object-oriented programing language having the capabilities of high-level programming language. Its easy to learn syntax and portability capability makes it popular these days. The followings facts gives us the introduction to Python −

* Python was developed by Guido van Rossum at Stichting Mathematisch Centrum in the Netherlands.
* It was written as the successor of programming language named ‘ABC’.
* It’s first version was released in 1991.
* The name Python was picked by Guido van Rossum from a TV show named Monty Python’s Flying Circus.
* It is an open source programming language which means that we can freely download it and use it to develop programs. It can be downloaded from [www.python.org](https://www.python.org/).
* Python programming language is having the features of Java and C both. It is having the elegant ‘C’ code and on the other hand, it is having classes and objects like Java for object-oriented programming.
* It is an interpreted language, which means the source code of Python program would be first converted into bytecode and then executed by Python virtual machine.

Strengths and Weaknesses of Python

Every programming language has some strengths as well as weaknesses, so does Python too.

Strengths

According to studies and surveys, Python is the fifth most important language as well as the most popular language for machine learning and data science. It is because of the following strengths that Python has −

**Easy to learn and understand** − The syntax of Python is simpler; hence it is relatively easy, even for beginners also, to learn and understand the language.

**Multi-purpose language** − Python is a multi-purpose programming language because it supports structured programming, object-oriented programming as well as functional programming.

**Huge number of modules** − Python has huge number of modules for covering every aspect of programming. These modules are easily available for use hence making Python an extensible language.

**Support of open source community** − As being open source programming language, Python is supported by a very large developer community. Due to this, the bugs are easily fixed by the Python community. This characteristic makes Python very robust and adaptive.

**Scalability** − Python is a scalable programming language because it provides an improved structure for supporting large programs than shell-scripts.

Weakness

Although Python is a popular and powerful programming language, it has its own weakness of slow execution speed.

The execution speed of Python is slow as compared to compiled languages because Python is an interpreted language. This can be the major area of improvement for Python community.

Installing Python

For working in Python, we must first have to install it. You can perform the installation of Python in any of the following two ways −

* Installing Python individually
* Using Pre-packaged Python distribution − Anaconda

Let us discuss these each in detail.

Installing Python Individually

If you want to install Python on your computer, then then you need to download only the binary code applicable for your platform. Python distribution is available for Windows, Linux and Mac platforms.

The following is a quick overview of installing Python on the above-mentioned platforms −

**On Unix and Linux platform**

With the help of following steps, we can install Python on Unix and Linux platform −

* First, go to [www.python.org/downloads/](https://www.python.org/downloads/).
* Next, click on the link to download zipped source code available for Unix/Linux.
* Now, Download and extract files.
* Next, we can edit the Modules/Setup file if we want to customize some options.
  + Next, write the command **run ./configure script**
  + make
  + make install

**On Windows platform**

With the help of following steps, we can install Python on Windows platform −

* First, go to [www.python.org/downloads/](https://www.python.org/downloads/).
* Next, click on the link for Windows installer python-XYZ.msi file. Here XYZ is the version we wish to install.
* Now, we must run the file that is downloaded. It will take us to the Python install wizard, which is easy to use. Now, accept the default settings and wait until the install is finished.

**On Macintosh platform**

For Mac OS X, Homebrew, a great and easy to use package installer is recommended to install Python 3. In case if you don't have Homebrew, you can install it with the help of following command −

$ ruby -e "$(curl -fsSL

https://raw.githubusercontent.com/Homebrew/install/master/install)"

It can be updated with the command below −

$ brew update

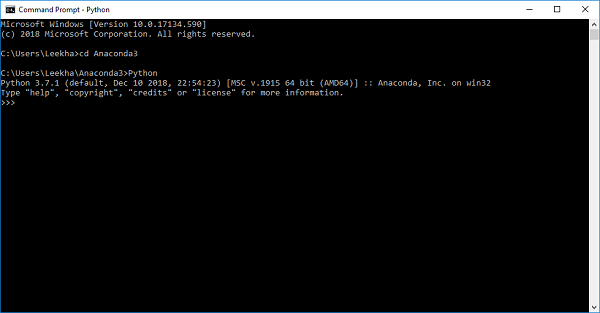
Now, to install Python3 on your system, we need to run the following command −

$ brew install python3

Using Pre-packaged Python Distribution: Anaconda

Anaconda is a packaged compilation of Python which have all the libraries widely used in Data science. We can follow the following steps to setup Python environment using Anaconda −

* **Step 1** − First, we need to download the required installation package from Anaconda distribution. The link for the same is [www.anaconda.com/distribution/](https://www.anaconda.com/products/individual). You can choose from Windows, Mac and Linux OS as per your requirement.
* **Step 2** − Next, select the Python version you want to install on your machine. The latest Python version is 3.7. There you will get the options for 64-bit and 32-bit Graphical installer both.
* **Step 3** − After selecting the OS and Python version, it will download the Anaconda installer on your computer. Now, double click the file and the installer will install Anaconda package.
* **Step 4** − For checking whether it is installed or not, open a command prompt and type Python as follows −



You can also check this in detailed video lecture at [www.tutorialspoint.com/python\_essentials\_online\_training/getting\_started\_with\_anaconda.asp](https://www.tutorialspoint.com/python_essentials_online_training/getting_started_with_anaconda.asp).

Why Python for Data Science?

Python is the fifth most important language as well as most popular language for Machine learning and data science. The following are the features of Python that makes it the preferred choice of language for data science −

Extensive set of packages

Python has an extensive and powerful set of packages which are ready to be used in various domains. It also has packages like **numpy, scipy, pandas, scikit-learn** etc. which are required for machine learning and data science.

Easy prototyping

Another important feature of Python that makes it the choice of language for data science is the easy and fast prototyping. This feature is useful for developing new algorithm.

Collaboration feature

The field of data science basically needs good collaboration and Python provides many useful tools that make this extremely.

One language for many domains

A typical data science project includes various domains like data extraction, data manipulation, data analysis, feature extraction, modelling, evaluation, deployment and updating the solution. As Python is a multi-purpose language, it allows the data scientist to address all these domains from a common platform.

Components of Python ML Ecosystem

In this section, let us discuss some core Data Science libraries that form the components of Python Machine learning ecosystem. These useful components make Python an important language for Data Science. Though there are many such components, let us discuss some of the importance components of Python ecosystem here −

Jupyter Notebook

Jupyter notebooks basically provides an interactive computational environment for developing Python based Data Science applications. They are formerly known as ipython notebooks. The following are some of the features of Jupyter notebooks that makes it one of the best components of Python ML ecosystem −

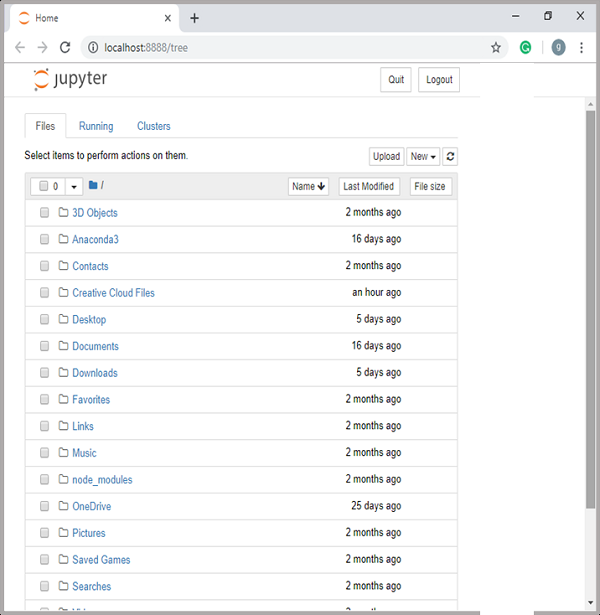
* Jupyter notebooks can illustrate the analysis process step by step by arranging the stuff like code, images, text, output etc. in a step by step manner.
* It helps a data scientist to document the thought process while developing the analysis process.
* One can also capture the result as the part of the notebook.
* With the help of jupyter notebooks, we can share our work with a peer also.

Installation and Execution

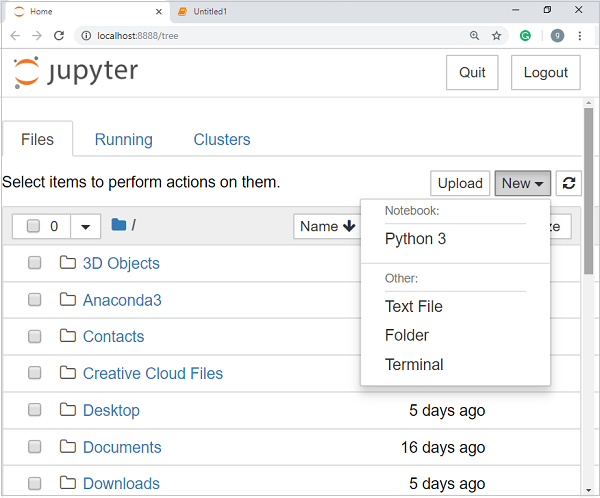
If you are using Anaconda distribution, then you need not install jupyter notebook separately as it is already installed with it. You just need to go to Anaconda Prompt and type the following command −

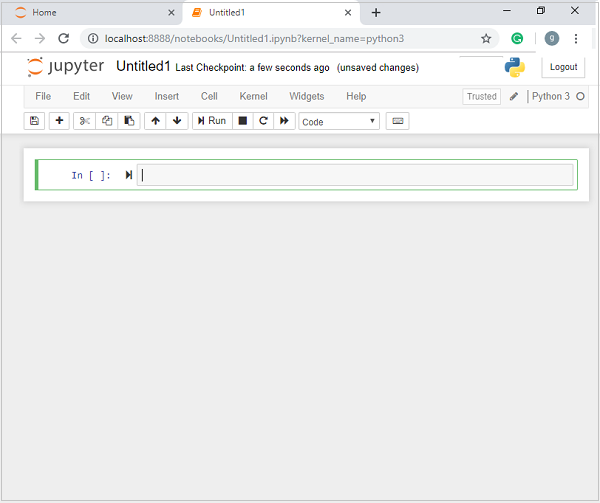
C:\>jupyter notebook

After pressing enter, it will start a notebook server at localhost:8888 of your computer. It is shown in the following screen shot −



Now, after clicking the New tab, you will get a list of options. Select Python 3 and it will take you to the new notebook for start working in it. You will get a glimpse of it in the following screenshots −





On the other hand, if you are using standard Python distribution then jupyter notebook can be installed using popular python package installer, pip.

pip install jupyter

Types of Cells in Jupyter Notebook

The following are the three types of cells in a jupyter notebook −

**Code cells** − As the name suggests, we can use these cells to write code. After writing the code/content, it will send it to the kernel that is associated with the notebook.

**Markdown cells** − We can use these cells for notating the computation process. They can contain the stuff like text, images, Latex equations, HTML tags etc.

**Raw cells** − The text written in them is displayed as it is. These cells are basically used to add the text that we do not wish to be converted by the automatic conversion mechanism of jupyter notebook.

For more detailed study of jupyter notebook, you can go to the link [www.tutorialspoint.com/jupyter/index.htm](https://www.tutorialspoint.com/jupyter/index.htm).

NumPy

It is another useful component that makes Python as one of the favorite languages for Data Science. It basically stands for Numerical Python and consists of multidimensional array objects. By using NumPy, we can perform the following important operations −

* Mathematical and logical operations on arrays.
* Fourier transformation
* Operations associated with linear algebra.

We can also see NumPy as the replacement of MatLab because NumPy is mostly used along with Scipy (Scientific Python) and Mat-plotlib (plotting library).

**Installation and Execution**

If you are using Anaconda distribution, then no need to install NumPy separately as it is already installed with it. You just need to import the package into your Python script with the help of following −

import numpy as np

On the other hand, if you are using standard Python distribution then NumPy can be installed using popular python package installer, pip.

pip install NumPy

For more detailed study of NumPy, you can go to the link[www.tutorialspoint.com/numpy/index.htm](https://www.tutorialspoint.com/numpy/index.htm).

Pandas

It is another useful Python library that makes Python one of the favorite languages for Data Science. Pandas is basically used for data manipulation, wrangling and analysis. It was developed by Wes McKinney in 2008. With the help of Pandas, in data processing we can accomplish the following five steps −

* Load
* Prepare
* Manipulate
* Model
* Analyze

Data representation in Pandas

The entire representation of data in Pandas is done with the help of following three data structures −

**Series** − It is basically a one-dimensional ndarray with an axis label which means it is like a simple array with homogeneous data. For example, the following series is a collection of integers 1,5,10,15,24,25...

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 5 | 10 | 15 | 24 | 25 | 28 | 36 | 40 | 89 |

**Data frame** − It is the most useful data structure and used for almost all kind of data representation and manipulation in pandas. It is basically a two-dimensional data structure which can contain heterogeneous data. Generally, tabular data is represented by using data frames. For example, the following table shows the data of students having their names and roll numbers, age and gender −

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Roll number** | **Age** | **Gender** |
| Aarav | 1 | 15 | Male |
| Harshit | 2 | 14 | Male |
| Kanika | 3 | 16 | Female |
| Mayank | 4 | 15 | Male |

**Panel** − It is a 3-dimensional data structure containing heterogeneous data. It is very difficult to represent the panel in graphical representation, but it can be illustrated as a container of DataFrame.

The following table gives us the dimension and description about above mentioned data structures used in Pandas −

|  |  |  |
| --- | --- | --- |
| **Data Structure** | **Dimension** | **Description** |
| Series | 1-D | Size immutable, 1-D homogeneous data |
| DataFrames | 2-D | Size Mutable, Heterogeneous data in tabular form |
| Panel | 3-D | Size-mutable array, container of DataFrame. |

We can understand these data structures as the higher dimensional data structure is the container of lower dimensional data structure.

Installation and Execution

If you are using Anaconda distribution, then no need to install Pandas separately as it is already installed with it. You just need to import the package into your Python script with the help of following −

import pandas as pd

On the other hand, if you are using standard Python distribution then Pandas can be installed using popular python package installer, pip.

pip install Pandas

After installing Pandas, you can import it into your Python script as did above.

Example

The following is an example of creating a series from ndarray by using Pandas −

In [1]: import pandas as pd

In [2]: import numpy as np

In [3]: data = np.array(['g','a','u','r','a','v'])

In [4]: s = pd.Series(data)

In [5]: print (s)

0 g

1 a

2 u

3 r

4 a

5 v

dtype: object

For more detailed study of Pandas you can go to the link [www.tutorialspoint.com/python\_pandas/index.htm](https://www.tutorialspoint.com/python_pandas/index.htm).

Scikit-learn

Another useful and most important python library for Data Science and machine learning in Python is Scikit-learn. The following are some features of Scikit-learn that makes it so useful −

* It is built on NumPy, SciPy, and Matplotlib.
* It is an open source and can be reused under BSD license.
* It is accessible to everybody and can be reused in various contexts.
* Wide range of machine learning algorithms covering major areas of ML like classification, clustering, regression, dimensionality reduction, model selection etc. can be implemented with the help of it.

Installation and Execution

If you are using Anaconda distribution, then no need to install Scikit-learn separately as it is already installed with it. You just need to use the package into your Python script. For example, with following line of script we are importing dataset of breast cancer patients from **Scikit-learn** −

from sklearn.datasets import load\_breast\_cancer

On the other hand, if you are using standard Python distribution and having NumPy and SciPy then Scikit-learn can be installed using popular python package installer, pip.

pip install -U scikit-learn

After installing Scikit-learn, you can use it into your Python script as you have done above.

Different Types of Methods

The following are various ML methods based on some broad categories −

Based on human supervision

In the learning process, some of the methods that are based on human supervision are as follows −

**Supervised Learning**

Supervised learning algorithms or methods are the most commonly used ML algorithms. This method or learning algorithm take the data sample i.e. the training data and its associated output i.e. labels or responses with each data samples during the training process.

The main objective of supervised learning algorithms is to learn an association between input data samples and corresponding outputs after performing multiple training data instances.

For example, we have

**x**: Input variables and

**Y**: Output variable

Now, apply an algorithm to learn the mapping function from the input to output as follows −

Y=f(x)

Now, the main objective would be to approximate the mapping function so well that even when we have new input data (x), we can easily predict the output variable (Y) for that new input data.

It is called supervised because the whole process of learning can be thought as it is being supervised by a teacher or supervisor. Examples of supervised machine learning algorithms includes **Decision tree, Random Forest, KNN, Logistic Regression** etc.

Based on the ML tasks, supervised learning algorithms can be divided into following two broad classes −

* Classification
* Regression

**Classification**

The key objective of classification-based tasks is to predict categorial output labels or responses for the given input data. The output will be based on what the model has learned in training phase. As we know that the categorial output responses means unordered and discrete values, hence each output response will belong to a specific class or category. We will discuss Classification and associated algorithms in detail in the upcoming chapters also.

**Regression**

The key objective of regression-based tasks is to predict output labels or responses which are continues numeric values, for the given input data. The output will be based on what the model has learned in its training phase. Basically, regression models use the input data features (independent variables) and their corresponding continuous numeric output values (dependent or outcome variables) to learn specific association between inputs and corresponding outputs. We will discuss regression and associated algorithms in detail in further chapters also.

Unsupervised Learning

As the name suggests, it is opposite to supervised ML methods or algorithms which means in unsupervised machine learning algorithms we do not have any supervisor to provide any sort of guidance. Unsupervised learning algorithms are handy in the scenario in which we do not have the liberty, like in supervised learning algorithms, of having pre-labeled training data and we want to extract useful pattern from input data.

For example, it can be understood as follows −

Suppose we have −

**x: Input variables**, then there would be no corresponding output variable and the algorithms need to discover the interesting pattern in data for learning.

Examples of unsupervised machine learning algorithms includes K-means clustering, **K-nearest neighbors** etc.

Based on the ML tasks, unsupervised learning algorithms can be divided into following broad classes −

* Clustering
* Association
* Dimensionality Reduction

**Clustering**

Clustering methods are one of the most useful unsupervised ML methods. These algorithms used to find similarity as well as relationship patterns among data samples and then cluster those samples into groups having similarity based on features. The real-world example of clustering is to group the customers by their purchasing behavior.

**Association**

Another useful unsupervised ML method is **Association** which is used to analyze large dataset to find patterns which further represents the interesting relationships between various items. It is also termed as **Association Rule Mining** or **Market basket analysis** which is mainly used to analyze customer shopping patterns.

**Dimensionality Reduction**

This unsupervised ML method is used to reduce the number of feature variables for each data sample by selecting set of principal or representative features. A question arises here is that why we need to reduce the dimensionality? The reason behind is the problem of feature space complexity which arises when we start analyzing and extracting millions of features from data samples. This problem generally refers to “curse of dimensionality”. PCA (Principal Component Analysis), K-nearest neighbors and discriminant analysis are some of the popular algorithms for this purpose.

**Anomaly Detection**

This unsupervised ML method is used to find out the occurrences of rare events or observations that generally do not occur. By using the learned knowledge, anomaly detection methods would be able to differentiate between anomalous or a normal data point. Some of the unsupervised algorithms like clustering, KNN can detect anomalies based on the data and its features.

Semi-supervised Learning

Such kind of algorithms or methods are neither fully supervised nor fully unsupervised. They basically fall between the two i.e. supervised and unsupervised learning methods. These kinds of algorithms generally use small supervised learning component i.e. small amount of pre-labeled annotated data and large unsupervised learning component i.e. lots of unlabeled data for training. We can follow any of the following approaches for implementing semi-supervised learning methods −

* The first and simple approach is to build the supervised model based on small amount of labeled and annotated data and then build the unsupervised model by applying the same to the large amounts of unlabeled data to get more labeled samples. Now, train the model on them and repeat the process.
* The second approach needs some extra efforts. In this approach, we can first use the unsupervised methods to cluster similar data samples, annotate these groups and then use a combination of this information to train the model.

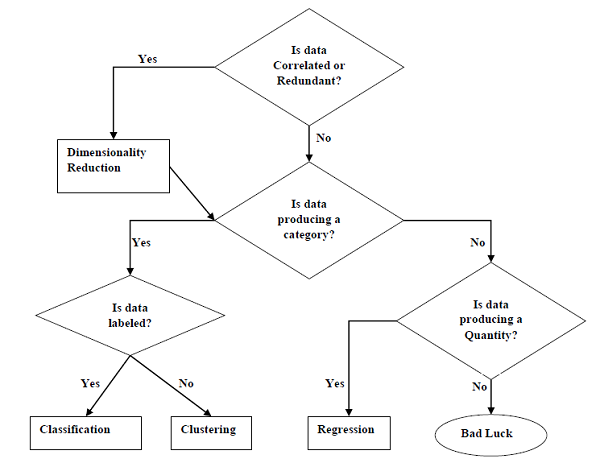
Reinforcement Learning

These methods are different from previously studied methods and very rarely used also. In this kind of learning algorithms, there would be an agent that we want to train over a period of time so that it can interact with a specific environment. The agent will follow a set of strategies for interacting with the environment and then after observing the environment it will take actions regards the current state of the environment. The following are the main steps of reinforcement learning methods −

* **Step 1** − First, we need to prepare an agent with some initial set of strategies.
* **Step 2** − Then observe the environment and its current state.
* **Step 3** − Next, select the optimal policy regards the current state of the environment and perform important action.
* **Step 4** − Now, the agent can get corresponding reward or penalty as per accordance with the action taken by it in previous step.
* **Step 5** − Now, we can update the strategies if it is required so.
* **Step 6** − At last, repeat steps 2-5 until the agent got to learn and adopt the optimal policies.

Tasks Suited for Machine Learning

The following diagram shows what type of task is appropriate for various ML problems −



Based on learning ability

In the learning process, the following are some methods that are based on learning ability −

**Batch Learning**

In many cases, we have end-to-end Machine Learning systems in which we need to train the model in one go by using whole available training data. Such kind of learning method or algorithm is called **Batch or Offline learning**. It is called Batch or Offline learning because it is a one-time procedure and the model will be trained with data in one single batch. The following are the main steps of Batch learning methods −

* **Step 1** − First, we need to collect all the training data for start training the model.
* **Step 2** − Now, start the training of model by providing whole training data in one go.
* **Step 3** − Next, stop learning/training process once you got satisfactory results/performance.
* **Step 4** − Finally, deploy this trained model into production. Here, it will predict the output for new data sample.

Online Learning

It is completely opposite to the batch or offline learning methods. In these learning methods, the training data is supplied in multiple incremental batches, called mini-batches, to the algorithm. Followings are the main steps of Online learning methods −

* **Step 1** − First, we need to collect all the training data for starting training of the model.
* **Step 2** − Now, start the training of model by providing a mini-batch of training data to the algorithm.
* **Step 3** − Next, we need to provide the mini-batches of training data in multiple increments to the algorithm.
* **Step 4** − As it will not stop like batch learning hence after providing whole training data in mini-batches, provide new data samples also to it.
* **Step 5** − Finally, it will keep learning over a period of time based on the new data samples.

Based on Generalization Approach

In the learning process, followings are some methods that are based on generalization approaches −

Instance based Learning

Instance based learning method is one of the useful methods that build the ML models by doing generalization based on the input data. It is opposite to the previously studied learning methods in the way that this kind of learning involves ML systems as well as methods that uses the raw data points themselves to draw the outcomes for newer data samples without building an explicit model on training data.

In simple words, instance-based learning basically starts working by looking at the input data points and then using a similarity metric, it will generalize and predict the new data points.

Model based Learning

In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called hyperparameters and in which input data is used to extract the features. In this learning, hyperparameters are optimized based on various model validation techniques. That is why we can say that Model based learning methods uses more traditional ML approach towards generalization.

**Genetic Algorithm**

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. **They are commonly used to generate high-quality solutions for optimization problems and search problems.**

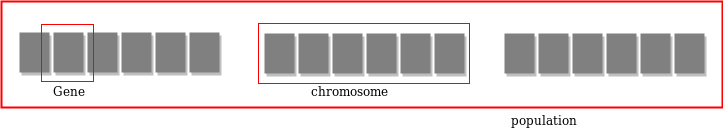
**Genetic algorithms simulate the process of natural selection** which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate “survival of the fittest” among individual of consecutive generation for solving a problem. **Each generation consist of a population of individuals** and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

**Foundation of Genetic Algorithms**

Genetic algorithms are based on an analogy with genetic structure and behavior of chromosome of the population. Following is the foundation of GAs based on this analogy –

1. Individual in population compete for resources and mate
2. Those individuals who are successful (fittest) then mate to create more offspring than others
3. Genes from “fittest” parent propagate throughout the generation, that is sometimes parents create offspring which is better than either parent.
4. Thus each successive generation is more suited for their environment.

**Search space**

The population of individuals are maintained within search space. Each individual represent a solution in search space for given problem. Each individual is coded as a finite length vector (analogous to chromosome) of components. These variable components are analogous to Genes. Thus a chromosome (individual) is composed of several genes (variable components).  


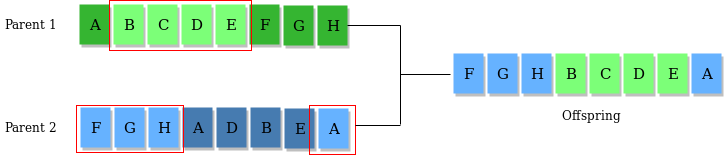
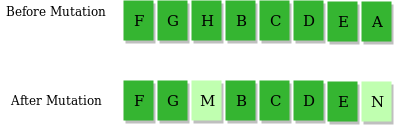
**Fitness Score**

A Fitness Score is given to each individual which **shows the ability of an individual to “compete”**. The individual having optimal fitness score (or near optimal) are sought.

The GAs maintains the population of n individuals (chromosome/solutions) along with their fitness scores.The individuals having better fitness scores are given more chance to reproduce than others. The individuals with better fitness scores are selected who mate and produce **better offspring** by combining chromosomes of parents. The population size is static so the room has to be created for new arrivals. So, some individuals die and get replaced by new arrivals eventually creating new generation when all the mating opportunity of the old population is exhausted. It is hoped that over successive generations better solutions will arrive while least fit die.

Each new generation has on average more “better genes” than the individual (solution) of previous generations. Thus each new generations have better **“partial solutions”** than previous generations. Once the offsprings produced having no significant difference than offspring produced by previous populations, the population is converged. The algorithm is said to be converged to a set of solutions for the problem.

**Operators of Genetic Algorithms**

Once the initial generation is created, the algorithm evolve the generation using following operators –  
**1) Selection Operator:** The idea is to give preference to the individuals with good fitness scores and allow them to pass there genes to the successive generations.  
**2) Crossover Operator:** This represents mating between individuals. Two individuals are selected using selection operator and crossover sites are chosen randomly. Then the genes at these crossover sites are exchanged thus creating a completely new individual (offspring). For example –  
  
**3) Mutation Operator:** The key idea is to insert random genes in offspring to maintain the diversity in population to avoid the premature convergence. For example –  
  
The whole algorithm can be summarized as –

1) Randomly initialize populations p

2) Determine fitness of population

3) Untill convergence repeat:

a) Select parents from population

b) Crossover and generate new population

c) Perform mutation on new population

d) Calculate fitness for new population

**Example problem and solution using Genetic Algorithms**

Given a target string, the goal is to produce target string starting from a random string of the same length. In the following implementation, following analogies are made –

* Characters A-Z, a-z, 0-9 and other special symbols are considered as genes
* A string generated by these character is considered as chromosome/solution/Individual

**Application of Genetic Algorithms**

Genetic algorithms have many applications, some of them are –

* Recurrent Neural Network
* Mutation testing
* Code breaking
* Filtering and signal processing
* Learning fuzzy rule base etc

**BEE ALGORITHM**

In computer science and operations research, the artificial bee colony algorithm (ABC) is an optimization algorithm based on the intelligent foraging behaviour of honey bee swarm, proposed by Derviş Karaboğa (Erciyes University) in 2005

**ALGORITHM**

In the ABC model, the colony consists of three groups of bees: employed bees, onlookers and scouts. It is assumed that there is only one artificial employed bee for each food source. In other words, the number of employed bees in the colony is equal to the number of food sources around the hive. Employed bees go to their food source and come back to hive and dance on this area. The employed bee whose food source has been abandoned becomes a scout and starts to search for finding a new food source. Onlookers watch the dances of employed bees and choose food sources depending on dances. The main steps of the algorithm are given below:[1]

Initial food sources are produced for all employed bees

REPEAT

Each employed bee goes to a food source in her memory and determines a closest source, then evaluates its nectar amount and dances in the hive

Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbour around that, she evaluates its nectar amount.

Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.

The best food source found so far is registered.

UNTIL (requirements are met)

In ABC, a population based algorithm, the position of a food source represents a possible solution to the optimization problem and the nectar amount of a food source corresponds to the quality (fitness) of the associated solution. The number of the employed bees is equal to the number of solutions in the population. At the first step, a randomly distributed initial population (food source positions) is generated. After initialization, the population is subjected to repeat the cycles of the search processes of the employed, onlooker, and scout bees, respectively. An employed bee produces a modification on the source position in her memory and discovers a new food source position. Provided that the nectar amount of the new one is higher than that of the previous source, the bee memorizes the new source position and forgets the old one. Otherwise she keeps the position of the one in her memory. After all employed bees complete the search process, they share the position information of the sources with the onlookers on the dance area. Each onlooker evaluates the nectar information taken from all employed bees and then chooses a food source depending on the nectar amounts of sources. As in the case of the employed bee, she produces a modification on the source position in her memory and checks its nectar amount. Providing that its nectar is higher than that of the previous one, the bee memorizes the new position and forgets the old one. The sources abandoned are determined and new sources are randomly produced to be replaced with the abandoned ones by artificial scouts.

Swarm intelligence (SI) is the collective behavior of decentralized, self-organized systems, whether natural or artificial. It’s an emerging field of biologically-inspired artificial intelligence based on the behavioral models of social insects such as ants, bees, wasps, termites, etc.

*Ever thought that ants patter of movement and strategy while finding food can be adopted to train our model. It’s just learning from animals, how they find their food. So that we can also apply those techniques to find our food (or solution of a problem) :D*

All swarm intelligence agents have two properties in common:

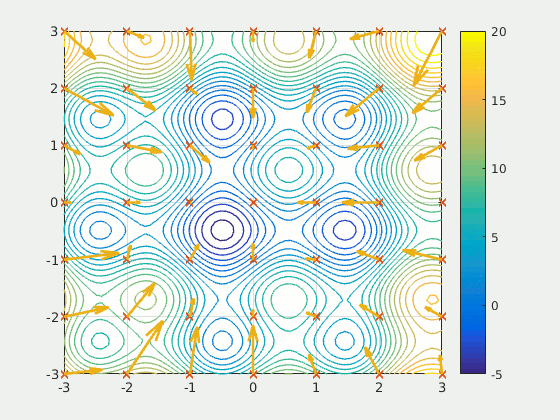
1. Self-organization: Any interaction is executed based on local information rather than the global context.
2. Division of Labor: Task/Workload is distributed among the entire population as needed.

To learn more about the foundations of swarm intelligence, refer to this post:

[Swarm Intelligence](http://www.techferry.com/articles/swarm-intelligence.html" \t "_blank)

[Swarm intelligence is an emerging field of biologically-inspired artificial intelligence based on the behavioral models…](http://www.techferry.com/articles/swarm-intelligence.html" \t "_blank)

[www.techferry.com](http://www.techferry.com/articles/swarm-intelligence.html" \t "_blank)



An Artificial Bee Colony (ABC) is one of the most recently defined algorithms by Dervish Karaboga under the larger umbrella of swarm intelligence, motivated by the intelligent behavior of honey bees, who aim to discover food sources with progressively higher amounts of nectar. The base paper can be found [here](https://www.researchgate.net/publication/221498082_Artificial_Bee_Colony_ABC_Optimization_Algorithm_for_Solving_Constrained_Optimization_Problems).

Main Components of Honey Bee Swarms

Compared to the natural swarms of bees, these main components can be somewhat linked to our model replicating honey bee movements.

1. **Food Source:** Represented by profitability, whose value depends on proximity, richness, and how easily it can be extracted.
2. **Employed foragers**: The foragers exploit a particular food source (solution) and pass information about the direction and profit value to other agents, with a certain probability.
3. **Unemployed foragers:** The agents that are not currently looking for a solution. These “onlookers” are a type of unemployed bee that watches the activity of the employed ones and starts searching for a source while scouts start in its neighborhood or nest.

A comparative study of real honey bee swarms and the algorithm variety:

* The swarm size can be considered to be our search space
* The food source replicates our solutions (as the bee looks for the food source with the most nectar, we look for the most optimized solution)
* Employee bees and onlooker bees are components in our model that work in the same way as the bees do (division of labor).
* The better the food source, the more fit it is for the bee, which can be linked to our objective function.

If the swarm size is S, then we can consider the number of food sources, employee bees, and onlooker bees to be S/2.

Before looking at how this kind of model works to find the most optimized solution, I should first introduce you to the various applications of the ABC algorithm in AI-related area.

When to use artificial bee colony algorithm?

The ABC algorithm is an optimization approach that requires a search space, a set of decision variables, and an objective function. Some of its popular use cases are in fields of:

1. Time series forecasting
2. Clustering problems
3. Feature selection for classification problems
4. Cost estimation problems

Machine learning is rapidly moving closer to where data is collected — edge devices

Phases in searching of a solution

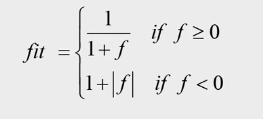
The meta-heuristic artificial bee colony algorithm finds its applications in the optimization of numerical problems. The intelligent searching behavior of honey bees forms the base of this algorithm. The artificial bee colony algorithm is responsible for performing both a global and local search, just like the bees do.

*Note: To better introduce you all to the phases of finding an optimized solution, I will refer to model components as their corresponding honey bee swarm unit.*

**Employed bee phase**

In the employed bee phase, each bee is associated with a solution, and the bees start to identify a better food source than the one they’re currently associated with. They generate a new solution based on the partner solution and the algorithm undergoes a greedy selection between them.

In an ABC algorithm, the fitness value directly corresponds to the objective function value(f), following the below formula:

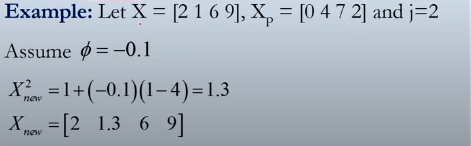


Since each employee bee is exploiting a particular food source, there is a one-to-one connection. Each bee evolves the solution by exploiting by the formula:

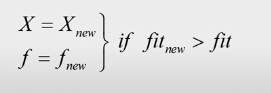




Φ is a random number in the range [-1,1], and i,j is the dimension of the decision variables of the solution. Below is an example showing how a new solution is generated by a randomly-selected variable:



If the fitness of current solution (X) is less than the new solution (Xnew), the new solution is greedily considered more optimum.

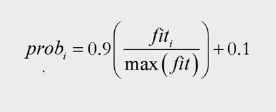


A*trial*counter is used to keep track of failures encountered by each solution. A *trial*value mapped with the current solution is increased by one every time a new solution is inferior. The *trial*value is set to zero if a new optimum solution is generated.

Onlooker bee phase

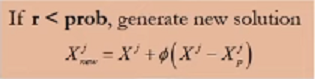
Onlookers bees evaluate food sources based on an employee bee’s waggle dance (movement done by bees to inform other bees of nectar). Talking about its counterpart in the algorithm, it implies the solutions exploited by the employed bees are the ones mapped to the onlooker bee in this phase. Each bee is not necessarily linked with a food source. Onlooker bees may/may not exploit a particular food source (i) based on the probability determined by the below equation:





So the probability of all solutions is determined beforehand, and solutions associated with a higher fitness value (objective function value) have a higher probability of being exploited in the onlooker phase.

For every onlooker bee, a random number (r) is generated. An onlooker bee will exploit a food source (solution) only if it meets the below criteria:



If the value of *trial*that’s associated with a particular solution becomes greater than the *limit*(a user-defined variable), then that particular solution enters scout phase.

**BAT ALGORITHM**

The Bat algorithm is a metaheuristic algorithm for global optimization. It was inspired by the echolocation behaviour of microbats, with varying pulse rates of emission and loudness.[1][2] The Bat algorithm was developed by Xin-She Yang in 2010

**METAPHOR**

The idealization of the echolocation of microbats can be summarized as follows: Each virtual bat flies randomly with a velocity VI at position (solution) xi with a varying frequency or wavelength and loudness Ai. As it searches and finds its prey, it changes frequency, loudness and pulse emission rate {\displaystyle r}r. Search is intensified by a local random walk. Selection of the best continues until certain stop criteria are met. This essentially uses a frequency-tuning technique to control the dynamic behaviour of a swarm of bats, and the balance between exploration and exploitation can be controlled by tuning algorithm-dependent parameters in bat algorithm.

A detailed introduction of metaheuristic algorithms including the bat algorithm is given by Yang[4] where a demo program in MATLAB/GNU Octave is available, while a comprehensive review is carried out by Parpinelli and Lopes.[5] A further improvement is the development of an evolving bat algorithm (EBA) with better efficiency

**ANT COLONY OPTIMIZATION(ACO)**

A Probabilistic Technique for finding Optimal Paths



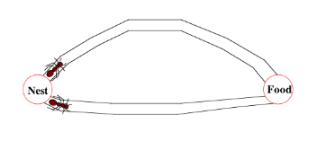
Photo by [Maksim Shutov](https://unsplash.com/@maksimshutov?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText) on [Unsplash](https://unsplash.com/s/photos/ant?utm_source=unsplash&utm_medium=referral&utm_content=creditCopyText)

Optimization problems are very important in the field of both scientific and industrial. Some real-life examples of these optimization problems are time table scheduling, nursing time distribution scheduling, train scheduling, capacity planning, traveling salesman problems, vehicle routing problems, Group-shop scheduling problem, portfolio optimization, etc. Many optimizations algorithms are developed for this reason. Ant colony optimization is one of them. Ant colony optimization is a probabilistic technique for finding optimal paths. In computer science and researches, the ant colony optimization algorithm is used for solving different computational problems.

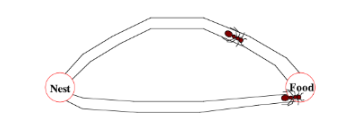
Ant colony optimization(ACO) was first introduced by Marco Dorigo in the 90s in his Ph.D. thesis. This algorithm is introduced based on the foraging behavior of an ant for seeking a path between their colony and source food. Initially, it was used to solve the well-known traveling salesman problem. Later, it is used for solving different hard optimization problems.

Ants are social insects. They live in colonies. The behavior of the ants is controlled by the goal of searching for food. While searching, ants roaming around their colonies. An ant repeatedly hops from one place to another to find the food. While moving, it deposits an organic compound called pheromone on the ground. Ants communicate with each other via pheromone trails. When an ant finds some amount of food it carries as much as it can carry. When returning it deposits pheromone on the paths based on the quantity and quality of the food. Ant can smell pheromone. So, other ants can smell that and follow that path. The higher the pheromone level has a higher probability of choosing that path and the more ants follow the path, the amount of pheromone will also increase on that path.

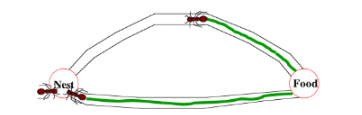
Let’s see an example of this. Let consider there are two paths to reach the food from the colony. At first, there is no pheromone on the ground. So, the probability of choosing these two paths is equal that means 50%. Let consider two ants choose two different paths to reach the food as the probability of choosing these paths is fifty-fifty.



The distances of these two paths are different. Ant following the shorter path will reach the food earlier than the other.



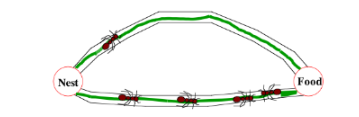
After finding food, it carries some food with itself and returns to the colony. When it tracking the returning path it deposits pheromone on the ground. The ant following the shorter path will reach the colony earlier.



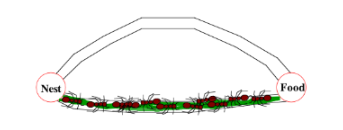
When the third ant wants to go out for searching food it will follow the path having shorter distance based on the pheromone level on the ground. As a shorter path has more pheromones than the longer, the third ant will follow the path having more pheromones.



By the time the ant following the longer path returned to the colony, more ants already have followed the path with more pheromones level. Then when another ant tries to reach the destination(food) from the colony it will find that each path has the same pheromone level. So, it randomly chooses one. Let consider it choose the above one(in the picture located below)



Repeating this process again and again, after some time, the shorter path has a more pheromone level than others and has a higher probability to follow the path, and all ants next time will follow the shorter path.



For solving different problems with ACO, there are three different proposed version of Ant-System:

**Ant Density**& **Ant Quantity:**Pheromone is updated in each movement of an ant from one location to another.

**Ant Cycle:**Pheromone is updated after all ants completed their tour.

Let see the pseudocode for applying the ant colony optimization algorithm. An artificial ant is made for finding the optimal solution. In the first step of solving a problem, each ant generates a solution. In the second step, paths found by different ants are compared. And in the third step, paths value or pheromone is updated.

**procedure** ACO\_MetaHeuristic **is**  
 **while** not\_termination **do**  
 generateSolutions()  
 daemonActions()  
 pheromoneUpdate()  
 **repeat**  
end procedure

There are many optimization problems where you can use ACO for finding the optimal solution. Some of them are:

1. Capacitated vehicle routing problem
2. Stochastic vehicle routing problem (SVRP)
3. Vehicle routing problem with pick-up and delivery (VRPPD)
4. Group-shop scheduling problem (GSP)
5. Nursing time distribution scheduling problem
6. Permutation flow shop problem (PFSP)
7. Frequency assignment problem
8. Redundancy allocation problem
9. Traveling salesman problem(TSP)

Let see the mathematical terms of ACO(typically for a TSP problem).

**Pheromone update**



The left side on the equation indicates the amount of pheromone on the given edge x,y

ρ — the rate of pheromone evaporation

And the last term on the right side indicated the amount of pheromone deposited.



Where L is the cost of an ant tour length and Q is a constant.

**EXISTING SYSTEM:**

Very few systems use the available clinical data for prediction purposes and even if they do ,they are restricted by the large number of association rules that apply.Diagnosis of the condition soley depends upon the Doctor's intuition and patient's records.The decision support system and will prove to be an aid for the physicians with the diagnosis.The algorithm,Fuzzy c means uses clustering and makes use of clusters and data points to predict the relativity of an attribute .Each data point is associated with multiple clusters depending upon the membership degrees

**DISADAVANTAGES:**

❖ Detection is not possible at an earlier stage

❖ Parctical use of various collected data is time consuming

**PROPOSED SYSTEM:**

To deal with the problem there is essential need of prediction system for awareness about diseases. Machine learning is the branch of Artificial Intelligence(AI), it provides prestigious support in predicting any kind of event which take training from natural events.we calculate accuracy of machine learning algorithms for predicting heart disease, for this algorithms are k-nearest neighbor, decision tree, linear regression and support vector machine(SVM) by using UCI repository dataset for training and testing

**ADAVANTAGES:**

* High performance and accurancy rate
* Machine Learning Algorithms is very flexible and is widely in various domains with high rates of success

**IMPLEMENTATION**

**MODULES:**

● Users

● Data Collection

● Attribute Selection

● Preprocessing of data.

**Users:**

User add the data to the database and view the data to the view data and predict the heart disease using ml.

**Data Collection:**

First step for predication system is data collection and deciding about the training and testing dataset. In this project we have used 73% training dataset and 37% dataset used as testing dataset the system.

**Attribute Selection:**

Attribute of dataset are property of dataset which are used for system and for heart many attributes are like heart bit rate of person, gender of the person, age of the person and many more predication system.

**Preprocessing of data:**

Preprocessing needed for achieving prestigious result from the machine learning algorithms. For example Random forest algorithm does not support null values dataset and for this we have to manage null values from original raw data. For our project we have to convert some categorized value by dummy value means in the form of “0”and “1” by using following code

**Admin:**

Admin will give authority to Users. In order to activate the users. the admin can Prediction Heart Disease.

**SOFTWARE REQUIREMENT**

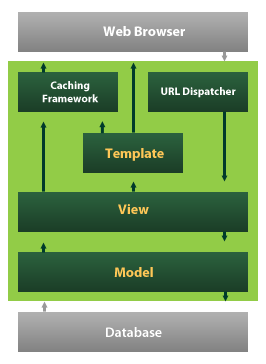
**PYTHON**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An [interpreted language](https://en.wikipedia.org/wiki/Interpreted_language), Python has a design philosophy that emphasizes code [readability](https://en.wikipedia.org/wiki/Readability) (notably using [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation to delimit [code blocks](https://en.wikipedia.org/wiki/Code_block) rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer [lines of code](https://en.wikipedia.org/wiki/Source_lines_of_code) than might be used in languages such as [C++](https://en.wikipedia.org/wiki/C%2B%2B)or [Java](https://en.wikipedia.org/wiki/Java_(programming_language)). It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many [operating systems](https://en.wikipedia.org/wiki/Operating_system). [CPython](https://en.wikipedia.org/wiki/CPython), the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python, is [open source](https://en.wikipedia.org/wiki/Open_source) software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation). Python features a [dynamic type](https://en.wikipedia.org/wiki/Dynamic_type) system and automatic [memory management](https://en.wikipedia.org/wiki/Memory_management). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigm), including [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming), [imperative](https://en.wikipedia.org/wiki/Imperative_programming), [functional](https://en.wikipedia.org/wiki/Functional_programming) and [procedural](https://en.wikipedia.org/wiki/Procedural_programming), and has a large and comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library)

**DJANGO**

Django is a high-level Python Web framework that encourages rapid development and clean, pragmatic design. Built by experienced developers, it takes care of much of the hassle of Web development, so you can focus on writing your app without needing to reinvent the wheel. It’s free and open source.

Django's primary goal is to ease the creation of complex, database-driven websites. Django emphasizes [reusability](https://en.wikipedia.org/wiki/Reusability)and "pluggability" of components, rapid development, and the principle of [don't repeat yourself](https://en.wikipedia.org/wiki/Don%27t_repeat_yourself). Python is used throughout, even for settings files and data models.



Django also provides an optional administrative [create, read, update and delete](https://en.wikipedia.org/wiki/Create,_read,_update_and_delete) interface that is generated dynamically through [introspection](https://en.wikipedia.org/wiki/Introspection_(computer_science)) and configured via admin models



**CONCLUSION AND FUTURE SCOPE**

Heart is one of the essential and vital organ of human body and prediction about heart diseases is also important concern for the human beings so that the accuracy for algorithm is one of parameter for analysis of performance of algorithms. Accuracy of the algorithms in machine learning depends upon the dataset that used for training and testing purpose. When we perform the analysis of algorithms on the basis of dataset whose attributes are shown in TABLE.1 and on the basis of confusion matrix, we find KNN is best one. For the Future Scope more machine learning approach will be used for best analysis of the heart diseases and for earlier prediction of diseases so that the rate of the death cases can be minimized by the awareness about the diseases.

**REFERENCES**

[1] Santhana Krishnan J and Geetha S, “Prediction of Heart Disease using Machine Learning Algorithms” ICIICT, 2019.

[2] Aditi Gavhane, Gouthami Kokkula, Isha Panday, Prof. Kailash Devadkar, “Prediction of Heart Disease using Machine Learning”, Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology(ICECA), 2018.

[3] Senthil kumar mohan, chandrasegar thirumalai and Gautam Srivastva, “Effective Heart Disease Prediction Using Hybrid Machine Learning Techniques” IEEE Access 2019.

[4] Himanshu Sharma and M A Rizvi, “Prediction of Heart Disease using Machine Learning Algorithms: A Survey” International Journal on Recent and Innovation Trends in Computing and Communication Volume: 5 Issue: 8 , IJRITCC August 2017.

[5] M. Nikhil Kumar, K. V. S. Koushik, K. Deepak, “Prediction of Heart Diseases Using Data Mining and Machine Learning Algorithms and Tools” International Journal of Scientific Research in Computer Science, Engineering and Information Technology ,IJSRCSEIT 2019.

[6] Amandeep Kaur and Jyoti Arora,“Heart Diseases Prediction using Data Mining Techniques: A survey” International Journal of Advanced Research in Computer Science , IJARCS 2015-2019.

[7] Pahulpreet Singh Kohli and Shriya Arora, “Application of Machine Learning in Diseases Prediction”, 4th International Conference on Computing Communication And Automation(ICCCA), 2018.

[8] M. Akhil, B. L. Deekshatulu, and P. Chandra, “Classification of Heart Disease Using K- Nearest Neighbor and Genetic Algorithm,” Procedia Technol., vol. 10, pp. 85–94, 2013.

[9] S. Kumra, R. Saxena, and S. Mehta, “An Extensive Review on Swarm Robotics,” pp. 140–145, 2009.

[10] Hazra, A., Mandal, S., Gupta, A. and Mukherjee, “ A Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review” Advances in Computational Sciences and Technology , 2017.

[11] Patel, J., Upadhyay, P. and Patel, “Heart Disease Prediction Using Machine learning and Data Mining Technique” Journals of Computer Science & Electronics , 2016.

[12] Chavan Patil, A.B. and Sonawane, P.“To Predict Heart Disease Risk and Medications Using Data Mining Techniques with an IoT Based Monitoring System for Post-Operative Heart Disease Patients” International Journal on Emerging Trends in Technology, 2017.

[13] V. Kirubha and S. M. Priya, “Survey on Data Mining Algorithms in Disease Prediction,” vol. 38, no. 3, pp. 124–128, 2016.

[14] M. A. Jabbar, P. Chandra, and B. L. Deekshatulu, “Prediction of risk score for heart disease using associative classification and hybrid feature subset selection,” Int. Conf. Intell. Syst. Des. Appl. ISDA, pp. 628–634, 2012.

[15] https://archive.ics.uci.edu/ml/datasets/Heart+Disease