**1. INTRODUCTION**

**1.1 Heart Disease:-**

Cardiovascular disease (CVD) is a class of diseases that involve the [heart](https://en.wikipedia.org/wiki/Heart) or [blood vessels](https://en.wikipedia.org/wiki/Blood_vessel). CVD includes [coronary artery diseases](https://en.wikipedia.org/wiki/Coronary_artery_disease) (CAD) such as [angina](https://en.wikipedia.org/wiki/Angina_pectoris) and [myocardial infarction](https://en.wikipedia.org/wiki/Myocardial_infarction) (commonly known as a heart attack). Other CVDs include [stroke](https://en.wikipedia.org/wiki/Stroke), [heart failure](https://en.wikipedia.org/wiki/Heart_failure), [hypertensive heart disease](https://en.wikipedia.org/wiki/Hypertensive_heart_disease), [rheumatic heart disease](https://en.wikipedia.org/wiki/Rheumatic_heart_disease), [cardiomyopathy](https://en.wikipedia.org/wiki/Cardiomyopathy), [heart arrhythmia](https://en.wikipedia.org/wiki/Heart_arrhythmia), [congenital heart disease](https://en.wikipedia.org/wiki/Congenital_heart_disease), [valvular heart disease](https://en.wikipedia.org/wiki/Valvular_heart_disease), [carditis](https://en.wikipedia.org/wiki/Carditis), [aortic aneurysms](https://en.wikipedia.org/wiki/Aortic_aneurysm), [peripheral artery disease](https://en.wikipedia.org/wiki/Peripheral_artery_disease), [thromboembolic disease](https://en.wikipedia.org/wiki/Thrombosis), and [venous thrombosis](https://en.wikipedia.org/wiki/Venous_thrombosis).

The underlying mechanisms vary depending on the disease. Coronary artery disease, stroke, and peripheral artery disease involve [atherosclerosis](https://en.wikipedia.org/wiki/Atherosclerosis). This may be caused by [high blood pressure](https://en.wikipedia.org/wiki/Hypertension), [smoking](https://en.wikipedia.org/wiki/Tobacco_smoking), [diabetes mellitus](https://en.wikipedia.org/wiki/Diabetes_mellitus), lack of [exercise](https://en.wikipedia.org/wiki/Physical_exercise), [obesity](https://en.wikipedia.org/wiki/Obesity), [high blood cholesterol](https://en.wikipedia.org/wiki/Hypercholesterolaemia), poor diet, and excessive [alcohol](https://en.wikipedia.org/wiki/Alcoholic_beverage) consumption, among others. High blood pressure is estimated to account for approximately 13% of CVD deaths, while tobacco accounts for 9%, diabetes 6%, lack of exercise 6% and obesity 5%. Rheumatic heart disease may follow untreated [strep throat](https://en.wikipedia.org/wiki/Streptococcal_pharyngitis).

It is estimated that up to 90% of CVD may be preventable. Prevention of CVD involves improving risk factors through: [healthy eating](https://en.wikipedia.org/wiki/Healthy_eating), exercise, avoidance of tobacco smoke and limiting alcohol intake. Treating risk factors, such as high blood pressure, blood lipids and diabetes is also beneficial. Treating people who have strep throat with [antibiotics](https://en.wikipedia.org/wiki/Antibiotic) can decrease the risk of rheumatic heart disease. The use of [aspirin](https://en.wikipedia.org/wiki/Aspirin) in people, who are otherwise healthy, is of unclear benefit.

Cardiovascular diseases are the [leading cause of death](https://en.wikipedia.org/wiki/List_of_causes_of_death_by_rate) globally. This is true in all areas of the world except Africa. Together CVD resulted in 17.9 million deaths (32.1%) in 2015, up from 12.3 million (25.8%) in 1990. [Deaths, at a given age](https://en.wikipedia.org/wiki/Age_standardized_deaths), from CVD are more common and have been increasing in much of the [developing world](https://en.wikipedia.org/wiki/Developing_world), while rates have declined in most of the [developed world](https://en.wikipedia.org/wiki/Developed_world) since the 1970s. Coronary artery disease and stroke account for 80% of CVD deaths in males and 75% of CVD deaths in females. Most cardiovascular disease affects older adults. In the United States 11% of people between 20 and 40 have CVD, while 37% between 40 and 60, 71% of people between 60 and 80, and 85% of people over 80 have CVD. The average age of death from coronary artery disease in the developed world is around 80 while it is around 68 in the developing world. Diagnosis of disease typically occurs seven to ten years earlier in men as compared to women

**1.2 Types:-**

There are many cardiovascular diseases involving the blood vessels. They are known as [vascular diseases](https://en.wikipedia.org/wiki/Vascular_disease).

* [Coronary artery disease](https://en.wikipedia.org/wiki/Coronary_artery_disease) (also known as coronary heart disease and ischemic heart disease)
* [Peripheral arterial disease](https://en.wikipedia.org/wiki/Peripheral_arterial_disease) – disease of blood vessels that supply blood to the arms and legs
* [Cerebrovascular disease](https://en.wikipedia.org/wiki/Cerebrovascular_disease) – disease of blood vessels that supply blood to the brain (includes [stroke](https://en.wikipedia.org/wiki/Stroke))
* [Renal artery stenosis](https://en.wikipedia.org/wiki/Renal_artery_stenosis)
* [Aortic aneurysm](https://en.wikipedia.org/wiki/Aortic_aneurysm)

There are also many cardiovascular diseases that involve the heart.

* [Cardiomyopathy](https://en.wikipedia.org/wiki/Cardiomyopathy) – diseases of cardiac muscle
* [Hypertensive heart disease](https://en.wikipedia.org/wiki/Hypertensive_heart_disease) – diseases of the heart secondary to high [blood pressure](https://en.wikipedia.org/wiki/Blood_pressure) or [hypertension](https://en.wikipedia.org/wiki/Hypertension)
* [Heart failure](https://en.wikipedia.org/wiki/Heart_failure) - a clinical syndrome caused by the inability of the heart to supply sufficient blood to the tissues to meet their metabolic requirements
* [Pulmonary heart disease](https://en.wikipedia.org/wiki/Pulmonary_heart_disease) – a failure at the right side of the heart with respiratory system involvement
* [Cardiac dysrhythmias](https://en.wikipedia.org/wiki/Cardiac_dysrhythmias) – abnormalities of heart rhythm
* Inflammatory heart disease
* [Endocarditis](https://en.wikipedia.org/wiki/Endocarditis) – [inflammation](https://en.wikipedia.org/wiki/Inflammation) of the inner layer of the heart, the [endocardium](https://en.wikipedia.org/wiki/Endocardium). The structures most commonly involved are the [heart valves](https://en.wikipedia.org/wiki/Heart_valve).
* Inflammatory [cardiomegaly](https://en.wikipedia.org/wiki/Cardiomegaly)
* [Myocarditis](https://en.wikipedia.org/wiki/Myocarditis) – inflammation of the [myocardium](https://en.wikipedia.org/wiki/Myocardium), the muscular part of the heart, caused most often by viral infection and less often by bacterial infections, certain medications, toxins, and autoimmune disorders. It is characterized in part by infiltration of the heart by [lymphocyte](https://en.wikipedia.org/wiki/Lymphocyte) and [monocyte](https://en.wikipedia.org/wiki/Monocyte) types of [white blood cells](https://en.wikipedia.org/wiki/White_blood_cells).
* [Eosinophilic myocarditis](https://en.wikipedia.org/wiki/Eosinophilic_myocarditis) - inflammation of the myocardium caused by pathologically activated [eosinophilic](https://en.wikipedia.org/wiki/Eosinophils) white blood cells. This disorder differs from myocarditis in its causes and treatments.
* [Valvular heart disease](https://en.wikipedia.org/wiki/Valvular_heart_disease)
* [Congenital heart disease](https://en.wikipedia.org/wiki/Congenital_heart_disease) – heart structure malformations existing at birth
* [Rheumatic heart disease](https://en.wikipedia.org/wiki/Rheumatic_heart_disease) – heart muscles and valves damage due to [rheumatic fever](https://en.wikipedia.org/wiki/Rheumatic_fever) caused by [Streptococcus pyogenes](https://en.wikipedia.org/wiki/Streptococcus_pyogenes) a [group A streptococcal infection](https://en.wikipedia.org/wiki/Group_A_streptococcal_infection).

**1.3 Statement of the problem:-**

The tests you'll need to diagnose your heart disease depend on what condition your doctor thinks you might have. No matter what type of heart disease you have, your doctor will likely perform a physical exam and ask about your personal and family medical history before doing any tests. Besides blood tests and a chest X-ray, tests to diagnose heart disease can include:

* **Electrocardiogram (ECG):-**An ECG records these electrical signals and can help your doctor detect irregularities in your heart's rhythm and structure. You may have an ECG while you're at rest or while exercising (stress electrocardiogram).
* **Holter monitoring:-** A Holter monitor is a portable device you wear to record a continuous ECG, usually for 24 to 72 hours. Holter monitoring is used to detect heart rhythm irregularities that aren't found during a regular ECG exam.
* **Echocardiogram:-**This noninvasive exam, which includes an ultrasound of your chest, shows detailed images of your heart's structure and function.
* **Stress test:-** This type of test involves raising your heart rate with exercise or medicine while performing heart tests and imaging to check how your heart responds.
* **Cardiac catheterization:-**In this test, a short tube (sheath) is inserted into a vein or artery in your leg (groin) or arm. A hollow, flexible and longer tube (guide catheter) is then inserted into the sheath. Aided by X-ray images on a monitor, your doctor threads the guide catheter through that artery until it reaches your heart.

The pressures in your heart chambers can be measured, and dye can be injected. The dye can be seen on an X-ray, which helps your doctor see the blood flow through your heart, blood vessels and valves to check for abnormalities.

* **Cardiac computerized tomography (CT) scan:-**This test is often used to check for heart problems. In a cardiac CT scan, you lie on a table inside a doughnut-shaped machine. An X-ray tube inside the machine rotates around your body and collects images of your heart and chest.
* **Cardiac magnetic resonance imaging (MRI):-**For this test, you lie on a table inside a long tube-like machine that produces a magnetic field. The magnetic field produces pictures to help your doctor evaluate your heart.

**1.4 Objective:-**

This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients.Thus we proposed the computer based patient records could reduce medical errors, enhance patient safety decrease unwanted practice variation, and improve patient outcome.

**1.5 Technical objective:-**

The main objective of this project is to develop a prototype Intelligent Heart Disease Prediction System using three data mining modeling techniques, namely, PCA, SVM.

**1.6 System Overview:**

This system named “Heart disease failure using machine learning” that aims to predict heart disease accuracy rate using machine learning .This project is intended to solve the mental and physical dilemma created in patients.

**1.7 System Features:-**

**Heart disease prediction**

Heart disease prediction is an action of predicting how accurate the heart disease is in a patient.This prediction feature of this system tries to predict the accurate rate of disease in return which involves an output i.e; linear SVM, non linear SVM and kernel RBF .

**1.8 Existing System:-**

Clinical decisions are often made based on doctors' intuition and experience rather than

on the knowledge rich data hidden in the database. This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. There are many ways that a medical misdiagnosis can present itself. Whether a doctor is at fault, or hospital staff, a misdiagnosis of a serious illness can have very extreme and harmful effects. Medical misdiagnoses are a serious risk to our healthcare profession. If they continue, then people will fear going to the hospital f treatment.

When we look into the ancient stages into history the disease identification to a person is made based on the prior or past medical records of the concerned person and that’s all was a medical alliances in order to get paid form the consumers. There is a lot of misdiagnosis going on in the current era too like producing fraudulent outcomes or results and keeping this in mind we look forward and wish to do project in this field in order to serve the people in utmost.

There is no proper clinical diagnosis of heart disease in these days everything seems to be commercial even in the medical field. The national patient safety foundation cites that fall to 42% of medical patience feel they have had experienced a medical error or a missed diagnosis. Patient safety is sometimes negligently given the backseat for other concerns, such as the cost of medical tests, drugs and operations.

Medical misdiagnosis are serious risk to our health care profession. If they continue, then people will fear going to the hospital. We can put an end to medical diagnosis is by informing the public and filling claims and suits against the medical practitioners at fault.

**1.9 Proposed System:-**

The main objective of this project is to develop a prototype Intelligent Heart Disease Prediction System using three data mining modeling techniques, namely, PCA, SVM .This practice leads to unwanted biases, errors and excessive medical costs which affects the quality of service provided to patients. Thus we proposed the computer based patient records could reduce medical errors, enhance patient safety decrease unwanted practice variation, and improve patient outcome.

This suggestion is promising as data modeling and analysis tools, e.g.,data mining, have the potential to generate a knowledge-rich environment which can help to significantly improve the quality of clinical decisions.

So, its providing effective treatments, it also helps to reduce treatment cost. To enhance visualization and ease of interpretation.

The most effective model to predict patients with heart disease appears to be SKleam library followed by Neural Network. Five mining goals are defined based on business intelligence and data exploration. The goals are evaluated against the trained models. All three models could answer complex queries, each with its own strength with respect ease of model interpretation, access to detailed information and accuracy. SKleam library could answer four out of the five goals; Decision Trees, three; and Neural Network, two. Although not the most effective model, Decision Trees results are easier to read and interpret. The drill through feature to access detailed patients’ profiles is only available in Decision Trees. SKleam library fared better than Decision Trees as it could identify all the significant medical predictors. The relationship between attributes produced by Neural Network is more difficult to understand. IHDPS can be further enhanced and expanded. For example, it can incorporate other medical attributes besides the 15 listed in Figure 1. It can also incorporate other data mining techniques, e.g., Time Series, Clustering and Association Rules. Continuous data can also be used instead of just categorical data. Another area is to use Text Mining to mine the vast amount of unstructured data available in healthcare databases. Another challenge would be to integrate data mining and text mining .

**2. SYSTEM REQUIREMENTS**

**2.1 Hardware requirements:-**

* 1 GB RAM or more
* 32 or 64 bit processor
* 1Ghz speed
* 6 GB memory or more.

**2.2 Software requirements:-**

Proposed model requires a python development platform which could support the required python library known as SKleam

* Windows operating system.
* Python 2.7 or above.
* SVM machine learning algorithm.

**3. TECHNOLOGY**

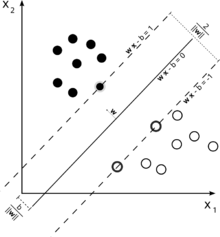
**3.1 What is Machine learning:-**

Machine learning (ML) is the [scientific study](https://en.wikipedia.org/wiki/Branches_of_science) of [algorithms](https://en.wikipedia.org/wiki/Algorithm) and [statistical models](https://en.wikipedia.org/wiki/Statistical_model) that [computer systems](https://en.wikipedia.org/wiki/Computer_systems) use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a mathematical model of sample data, known as "[training data](https://en.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as [email filtering](https://en.wikipedia.org/wiki/Email_filtering), and [computer vision](https://en.wikipedia.org/wiki/Computer_vision), where it is infeasible to develop an algorithm of specific instructions for performing the task. Machine learning is closely related to [computational statistics](https://en.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers. The study of [mathematical optimization](https://en.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.wikipedia.org/wiki/Data_mining) is a field of study within machine learning, and focuses on [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) through [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning). In its application across business problems, machine learning is also referred to as [predictive analytics](https://en.wikipedia.org/wiki/Predictive_analytics).

**3.1.1 Overview of Machine Learning**

The name machine learning was coined in 1959 by [Arthur Samuel](https://en.wikipedia.org/wiki/Arthur_Samuel). [Tom M. Mitchell](https://en.wikipedia.org/wiki/Tom_M._Mitchell) provided a widely quoted, more formal definition of the algorithms studied in the machine learning field: "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*." This definition of the tasks in which machine learning is concerned offers a fundamentally [operational definition](https://en.wikipedia.org/wiki/Operational_definition) rather than defining the field in cognitive terms. This follows [Alan Turing](https://en.wikipedia.org/wiki/Alan_Turing)'s proposal in his paper "[Computing Machinery and Intelligence](https://en.wikipedia.org/wiki/Computing_Machinery_and_Intelligence)", in which the question "Can machines think?" is replaced with the question "Can machines do what we (as thinking entities) can do?". In Turing's proposal the various characteristics that could be possessed by a thinking machine and the various implications in constructing one are exposed

**3.1.2Machine learning tasks**

[](https://en.wikipedia.org/wiki/File:Svm_max_sep_hyperplane_with_margin.png)

A [support vector machine](https://en.wikipedia.org/wiki/Support_vector_machine) is a supervised learning model that divides the data into regions separated by a [linear boundary](https://en.wikipedia.org/wiki/Linear_classifier). Here, the linear boundary divides the black circles from the white.

Machine learning tasks are classified into several broad categories. In [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning), the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the [training data](https://en.wikipedia.org/wiki/Training_data) for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. [Semi-supervised learning](https://en.wikipedia.org/wiki/Semi-supervised_learning) algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels.

[Classification](https://en.wikipedia.org/wiki/Statistical_classification) algorithms and [regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a [limited set](https://en.wikipedia.org/wiki/Discrete_number) of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the [Boolean](https://en.wikipedia.org/wiki/Boolean_data_type) values true and false. [Regression](https://en.wikipedia.org/wiki/Regression_analysis) algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object.

In [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning), the algorithm builds a mathematical model from a set of data which contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or [clustering](https://en.wikipedia.org/wiki/Cluster_analysis) of data points. Unsupervised learning can discover patterns in the data, and can group the inputs into categories, as in [feature learning](https://en.wikipedia.org/wiki/Feature_learning). [Dimensionality reduction](https://en.wikipedia.org/wiki/Dimensionality_reduction) is the process of reducing the number of "features", or inputs, in a set of data.

[Active learning](https://en.wikipedia.org/wiki/Active_learning_(machine_learning)) algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget, and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. [Reinforcement learning](https://en.wikipedia.org/wiki/Reinforcement_learning) algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment, and are used in [autonomous vehicles](https://en.wikipedia.org/wiki/Autonomous_vehicle) or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include [topic modeling](https://en.wikipedia.org/wiki/Topic_modeling), where the computer program is given a set of [natural language](https://en.wikipedia.org/wiki/Natural_language) documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable [probability density function](https://en.wikipedia.org/wiki/Probability_density_function) in [density estimation](https://en.wikipedia.org/wiki/Density_estimation) problems. [Meta learning](https://en.wikipedia.org/wiki/Meta_learning_(computer_science)) algorithms learn their own [inductive bias](https://en.wikipedia.org/wiki/Inductive_bias) based on previous experience. In [developmental robotics](https://en.wikipedia.org/wiki/Developmental_robotics), [robot learning](https://en.wikipedia.org/wiki/Robot_learning) algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

###### **3.2 Machine Learning Classifiers:-**

In order to classify the heart patients and healthy people, machine learning classification algorithms are used. Some popular classification algorithms and their theoretical background are discussed briefly in this paper.

**3.2.1 Logistic Regression:-**

A logistic regression is a classification algorithm [[27](https://www.hindawi.com/journals/misy/2018/3860146/#B27)–[29](https://www.hindawi.com/journals/misy/2018/3860146/#B29)]. For binary classification problem, in order to predict the value of predictive variable y when y ∈ [0, 1], 0 is negative class and 1 is positive class. It also uses multiclassification to predict the value of y when y ∈ [0, 1, 2, 3].

In order to classify two classes 0 and 1, a hypothesis  will be designed and threshold classifier output is  at 0.5. If the value of hypothesis , it will predict y = 1 which mean that the person has heart disease and if value of , then predict y = 0 which shows that the person is healthy.

Hence, the prediction of logistic regression under the condition  is done.

**3.3 Data mining algorithms:-**

Research on data mining has led to the formulation of several data mining

algorithms.These algorithms can be directly used on a dataset for creating some models

or to draw vital conclusions and inferences from that dataset. Some popular data mining

algorithms are Decision tree, Naïve Bayes, k-means, artificial neural network etc.

They are discussed in the follows section.

**3.4 Materials and methods:-**

**3.4.1 Dataset:-**

The “Cleveland heart disease dataset 2016” is used by various researchers and can be accessed from online data mining repository of the University of California, Irvine. This dataset was used in this research study for designing machine-learning-based system for heart disease diagnosis. The Cleveland heart disease dataset has a sample size of 303 patients, 76 features, and some missing values. During the analysis, 6 samples were removed due to missing values in feature columns and leftover samples size is 297 with 13 more appropriate independent input features, and target output label was extracted and used for diagnosing the heart disease. The target output label has two classes in order to represent a heart patient or a normal subject. Thus, the extracted dataset is of 29713 features matrix. The complete information and description of 297 instances of 13 features of the dataset is given in following Table

|  |
| --- |
| **3.4.2 Attributes used:-**  The dataset have 76 raw attributes, only 14 of them are actually used. The attributes used are : Age : age in yearsSex : sex (1 = male; 0 = female)Cp : Chest Pain Type Value 1: typical angina Value 2: atypical angina Value 3: non‐anginal pain Value 4: asymptomaticTrestbps : Resting blood pressure (in mm Hg on admission to the hospital)Chol : serum cholestoral in mg/dlFbs : (fasting blood sugar > 120 mg/dl) (1 = true; 0 = false)Thalach : maximum heart rate achievedRestecg : resting electrocardiographic results Value 0: normal Value 1: having ST‐T wave abnormality (T wave inversions and/or ST elevation or depression of > 0.05 mV) Value 2: showing probable or definite left ventricular hypertrophy by Estes' criteriaExang : exercise induced angina (1 = yes; 0 = no)Oldpeak : ST depression induced by exercise relative to restSlope : the slope of the peak exercise ST segment Value 1: upsloping Value 2: flat Value 3: downslopingCa : number of major vessels (0‐3) colored by flourosopyThal : 3 = normal; 6 = fixed defect; 7 = reversable defectNum : diagnosis of heart disease (angiographic disease status) Value 0: < 50% diameter narrowing  Table 1: Features information and description of Cleveland heart disease dataset 2016 |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | |  | | | | | | S. no. | Feature name | Feature code | Description | Domain of values (min-max) | |  | | | | | | 1 | Age | AGE | Age in years | 30 < age < 77 | | 2 | Sex | SEX | Male = 1 | 1 | | Female = 0 | 0 | | 3 | Type of chest pain | CPT | 1 = atypical angina | 1 | | 2 = typical angina | 2 | | 3 = asymptomatic | 3 | | 4 = nonanginal pain | 4 | | 4 | Resting blood pressure | RBP | mm Hg admitted at the hospital | 94–200 | | 5 | Serum cholesterol | SCH | In mg/dl | 120–564 | | 6 | Fasting blood sugar >120 mg/dl | FBS | Fasting blood sugar >120 mg/dl (1 = true; 0 = false) | 1 | | 0 | | 7 | Resting electrocardiographic results | RES | 0 = normal | 0 | | 1 = having ST-T | 1 | | 2 = hypertrophy | 2 | | 8 | Maximum heart rate achieved | MHR | — | 71–202 | | 9 | Exercise-induced angina | EIA | 1 = yes | 0 | | 0 = no | 1 | | 10 | Old peak = ST depression induced by exercise relative to rest | OPK | — | 0–6.2 | | 11 | Slope of the peak exercise ST segment | PES | 1 = up sloping | 1 | | 2 = flat | 2 | | 3 = down sloping | 3 | | 12 | Number of major vessels (0–3) colored by fluoroscopy | VCA | — | 0 | | 1 | | 2 | | 3 | | 13 | Thallium scan | THA | 3 = normal | 3 | | 6 = fixed defect | 6 | | 7 = reversible defect | 7 | |

**3.5 Support vector machine:-**

The SVM is a machine learning classification algorithm which has been mostly used for classification problems [[30](https://www.hindawi.com/journals/misy/2018/3860146/#B30)–[32](https://www.hindawi.com/journals/misy/2018/3860146/#B32)]. SVM used a maximum margin strategy that transformed into solving a complex quadratic programming problem. Due to the high performance of SVM in classification, various applications widely applied it.

In a binary classification problem, the instances are separated with a hyperplane , where  and dare dimensional coefficient vectors, which are normal to the hyperplane of the surface, b is offset value from the origin, and x is data set values. The SVM gets results of  and b.  can be solved by introducing Lagrangian multipliers in the linear case. The data points on borders are called support vectors. The solution of  can be written as , where n is the number of support vectors and yi are target labels to x. The value of and b are calculated, and the linear discriminant function can be written as follows:

The nonlinear scenario, for kernel trick and decision function, can be written as follows:

The positive semidefinite functions obey Mercer’s condition as kernel functions

Support Vector Machine is an extremely popular supervised machine learning technique(having a pre-defined target variable) which can be used as a classifier as well as a predictor. For classification, it finds a hyper-plane in the feature space that differentiates between the classes. An SVM model represents the training data points as points in the feature space, mapped in such a way that points belonging to separate classes are segregated by a margin as wide as possible.



The test data points are then mapped into that same space and are classified based on which side of the margin they fall. Fig. 1: Support Vector Machine Shan Xu et al. have used SVM to achieve an accuracy of 98.9% in People's Hospital dataset [5].In [9], SVM performs the best with 85.7655% of correctly classified instance and in [10] SVM is used with boosting technique to give an accuracy of 84.81%. HoudaMezrigui et al. have used SVM to attain a f-measure value of 93.5617 [11]. In [12] SVM classifies the pixel variation with an accuracy of 92.1% helping to identify the affected region accurately

**3.5.1 Need of support vector machine:-**

It is a supervised learning method which classifies data into two classes over a hyper

plane. Support vector machine performs a similar task like C4.5 except that it doesn’t

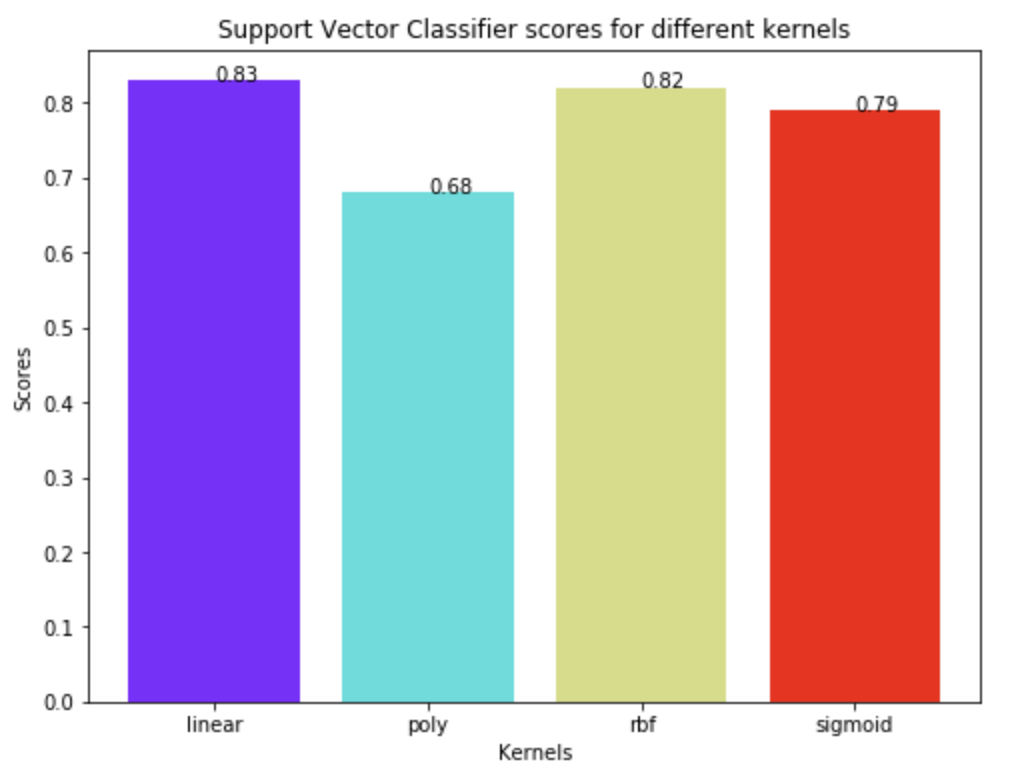
use Decision trees at all. Support vector machine attempts to maximize the margin

(distance between the hyper plane and the two closest data points from each

respective class) to decrease any chance of misclassification. Some

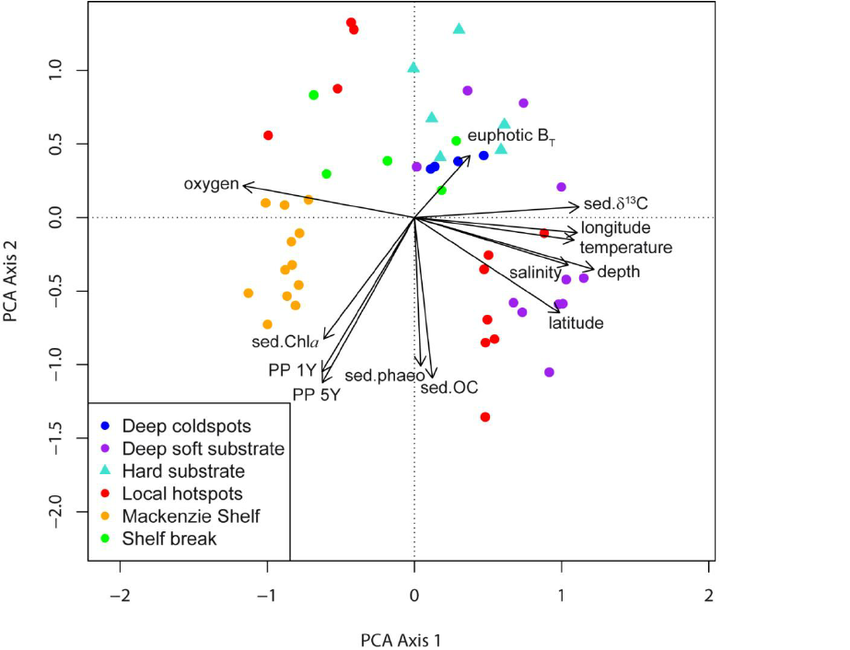
popular implementations of support vector machine are scikit-learn, MATLAB and of

LIBSVM.



**3.6 Principal component analysis:-**

Principal component analysis (PCA) is a statistical procedure that uses an [orthogonal transformation](https://en.wikipedia.org/wiki/Orthogonal_transformation) to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of [linearly uncorrelated](https://en.wikipedia.org/wiki/Correlation_and_dependence) variables called principal components. If there are {\displaystyle n}observations with {\displaystyle p}variables, then the number of distinct principal components is {\displaystyle \min(n-1,p)}. This transformation is defined in such a way that the first principal component has the largest possible [variance](https://en.wikipedia.org/wiki/Variance) (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it is [orthogonal](https://en.wikipedia.org/wiki/Orthogonal) to the preceding components. The resulting vectors (each being a [linear combination](https://en.wikipedia.org/wiki/Linear_combination) of the variables and containing *n* observations) are an uncorrelated [orthogonal basis set](https://en.wikipedia.org/wiki/Orthogonal_basis_set). PCA is sensitive to the relative scaling of the original variables.



PCA is mostly used as a tool in [exploratory data analysis](https://en.wikipedia.org/wiki/Exploratory_data_analysis) and for making [predictive models](https://en.wikipedia.org/wiki/Predictive_modeling). It is often used to visualize genetic distance and relatedness between populations. PCA can be done by [eigenvalue decomposition](https://en.wikipedia.org/wiki/Eigendecomposition_of_a_matrix) of a data [covariance](https://en.wikipedia.org/wiki/Covariance) (or [correlation](https://en.wikipedia.org/wiki/Correlation)) matrix or [singular value decomposition](https://en.wikipedia.org/wiki/Singular_value_decomposition) of a [data matrix](https://en.wikipedia.org/wiki/Data_matrix_(multivariate_statistics)), usually after a normalization step of the initial data. The normalization of each attribute consists of mean centering – subtracting each data value from its variable's measured mean so that its empirical mean (average) is zero – and, possibly, normalizing each variable's variance to make it equal to 1; see [Z-scores](https://en.wikipedia.org/wiki/Z-score). The results of a PCA are usually discussed in terms of component scores, sometimes called factor scores (the transformed variable values corresponding to a particular data point), and loadings (the weight by which each standardized original variable should be multiplied to get the component score). If component scores are standardized to unit variance, loadings must contain the data variance in them (and that is the magnitude of eigenvalues). If component scores are not standardized (therefore they contain the data variance) then loadings must be unit-scaled, ("normalized") and these weights are called eigenvectors; they are the cosines of orthogonal rotation of variables into principal components or back.

**3.6.1 Implementing PCA on a 2-D dataset in detail:-**

**Step 1:**

Normalize the data

First step is to normalize the data that we have so that PCA works properly. This is done by subtracting the respective means from the numbers in the respective column. So if we have two dimensions X and Y, all X become 𝔁- and all Y become 𝒚-. This produces a dataset whose mean is zero.

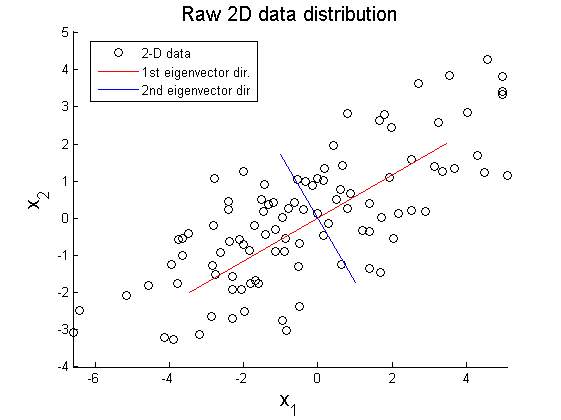
**Step 2:**

Calculate the covariance matrix

Since the dataset we took is 2-dimensional, this will result in a 2x2 Covariance matrix.

https://s3.amazonaws.com/files.dezyre.com/images/Tutorials/Covariance+Matrix.JPG

Please note that Var[X1] = Cov[X1,X1] and Var[X2] = Cov[X2,X2].



**Step 3:**

Calculate the eigenvalues and eigenvectors

Next step is to calculate the eigenvalues and eigenvectors for the covariance matrix. The same is possible because it is a square matrix. ƛ is an eigenvalue for a matrix A if it is a solution of the characteristic equation:

det( ƛI - A ) = 0

Where, I is the identity matrix of the same dimension as A which is a required condition for the matrix subtraction as well in this case and ‘det’ is the determinant of the matrix. For each eigenvalue ƛ, a corresponding eigen-vector v, can be found by solving:

( ƛI - A )v = 0

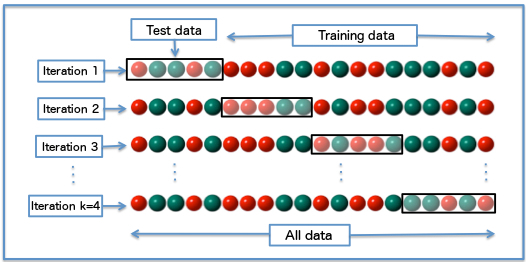
**Step 4:**

Choosing components and forming a feature vector:

We order the eigenvalues from largest to smallest so that it gives us the components in order or significance. Here comes the dimensionality reduction part. If we have a dataset with n variables, then we have the corresponding neigenvalues and eigenvectors. It turns out that the eigenvector corresponding to the highest eigenvalue is the principal component of the dataset and it is our call as to how many eigenvalues we choose to proceed our analysis with. To reduce the dimensions, we choose the first p eigenvalues and ignore the rest. We do lose out some information in the process, but if the eigenvalues are small, we do not lose much.

**3.7 Cross validation techniques:-**

Cross-validation, sometimes called rotation estimation, or out-of-sample testing is any of various similar [model validation](https://en.wikipedia.org/wiki/Model_validation) techniques for assessing how the results of a [statistical](https://en.wikipedia.org/wiki/Statistics) analysis will generalize to an independent data set. It is mainly used in settings where the goal is prediction, and one wants to estimate how [accurately](https://en.wikipedia.org/wiki/Accuracy) a [predictive model](https://en.wikipedia.org/wiki/Predictive_modelling) will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the [validation dataset](https://en.wikipedia.org/wiki/Validation_set) or testing set). The goal of cross-validation is to test the model's ability to predict new data that was not used in estimating it, in order to flag problems like [overfitting](https://en.wikipedia.org/wiki/Overfitting) or [selection bias](https://en.wikipedia.org/wiki/Selection_bias) and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem).



One round of cross-validation involves [partitioning](https://en.wikipedia.org/wiki/Partition_of_a_set) a [sample](https://en.wikipedia.org/wiki/Statistical_sample) of [data](https://en.wikipedia.org/wiki/Data) into [complementary](https://en.wikipedia.org/wiki/Complement_(set_theory)) subsets, performing the analysis on one subset (called the training set), and validating the analysis on the other subset (called the validation set or testing set). To reduce [variability](https://en.wikipedia.org/wiki/Variance), in most methods multiple rounds of cross-validation are performed using different partitions, and the validation results are combined (e.g. averaged) over the rounds to give an estimate of the model's predictive performance.

In summary, cross-validation combines (averages) measures of fitness in prediction to derive a more accurate estimate of model prediction performance.

**3.7.1 Purpose of cross validation:-**

Suppose we have a [model](https://en.wikipedia.org/wiki/Statistical_model) with one or more unknown [parameters](https://en.wikipedia.org/wiki/Parameters), and a data set to which the model can be fit (the training data set). The fitting process [optimizes](https://en.wikipedia.org/wiki/Optimization_(mathematics)) the model parameters to make the model fit the training data as well as possible. If we then take an [independent](https://en.wikipedia.org/wiki/Independence_(probability_theory)) sample of validation data from the same [population](https://en.wikipedia.org/wiki/Statistical_population) as the training data, it will generally turn out that the model does not fit the validation data as well as it fits the training data. The size of this difference is likely to be large especially when the size of the training data set is small, or when the number of parameters in the model is large. Cross-validation is a way to estimate the size of this effect.

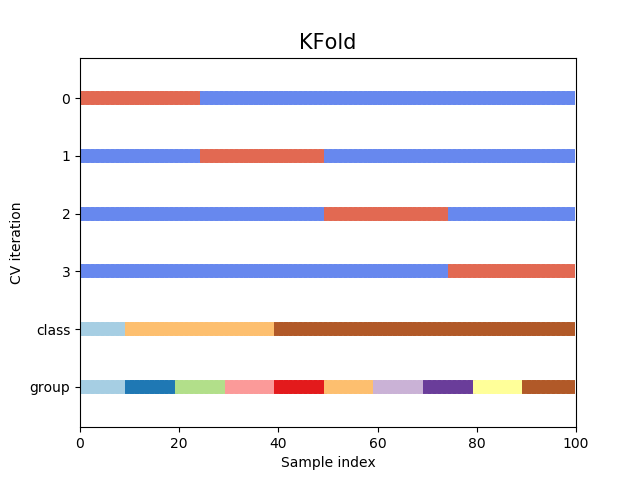
In linear regression we have [real](https://en.wikipedia.org/wiki/Real_number) response values y1, ..., yn, and n p-dimensional [vector](https://en.wikipedia.org/wiki/Euclidean_vector) covariates x1, ..., xn. The components of the vector xi are denoted xi1, ..., xip. If we use [least squares](https://en.wikipedia.org/wiki/Least_squares) to fit a function in the form of a [hyperplane](https://en.wikipedia.org/wiki/Hyperplane) y = a + βTx to the data (xi, yi)1 ≤ i ≤ n, we could then assess the fit using the [mean squared error](https://en.wikipedia.org/wiki/Mean_squared_error) (MSE). The MSE for given estimated parameter values a and β on the training set (xi, yi)1 ≤ i ≤ n

{\displaystyle {\frac {1}{n}}\sum \_{i=1}^{n}(y\_{i}-a-{\boldsymbol {\beta }}^{T}\mathbf {x} \_{i})^{2}={\frac {1}{n}}\sum \_{i=1}^{n}(y\_{i}-a-\beta \_{1}x\_{i1}-\dots -\beta \_{p}x\_{ip})^{2}}If the model is correctly specified, it can be shown under mild assumptions that the [expected value](https://en.wikipedia.org/wiki/Expected_value) of the MSE for the training set is (n − p − 1)/(n + p + 1) < 1 times the expected value of the MSE for the validation set[[8]](https://en.wikipedia.org/wiki/Cross-validation_(statistics)#cite_note-8) (the expected value is taken over the distribution of training sets). Thus if we fit the model and compute the MSE on the training set, we will get an optimistically [biased](https://en.wikipedia.org/wiki/Bias_(statistics)) assessment of how well the model will fit an independent data set. This biased estimate is called the in-sample estimate of the fit, whereas the cross-validation estimate is an out-of-sample estimate.

Since in linear regression it is possible to directly compute the factor (n − p − 1)/(n + p + 1) by which the training MSE underestimates the validation MSE under the assumption that the model specification is valid, cross-validation can be used for checking whether the model has been [overfitted](https://en.wikipedia.org/wiki/Overfitting), in which case the MSE in the validation set will substantially exceed its anticipated value. (Cross-validation in the context of linear regression is also useful in that it can be used to select an optimally regularized cost function). In most other regression procedures (e.g. [logistic regression](https://en.wikipedia.org/wiki/Logistic_regression)), there is no simple formula to compute the expected out-of-sample fit. Cross-validation is, thus, a generally applicable way to predict the performance of a model on unavailable data using numerical computation in place of theoretical analysis.

**K-fold cross validation:-**

In k-fold cross-validation, the original sample is randomly partitioned into k equal sized subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the model, and the remaining k − 1 subsamples are used as training data.



The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the validation data. The k results can then be averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling (see below) is that all observations are used for both training and validation, and each observation is used for validation exactly once. 10-fold cross-validation is commonly used, but in general kremains an unfixed parameter.

For example, setting k = 2 results in 2-fold cross-validation. In 2-fold cross-validation, we randomly shuffle the dataset into two sets d0 and d1, so that both sets are equal size (this is usually implemented by shuffling the data array and then splitting it in two). We then train on d0 and validate on d1, followed by training on d1 and validating on d0.

When k = n (the number of observations), the k-fold cross-validation is exactly the leave-one-out cross-validation.

In stratified k-fold cross-validation, the folds are selected so that the mean response value is approximately equal in all the folds. In the case of binary classification, this means that each fold contains roughly the same proportions of the two types of class labels.

**3.8 Methodology:-**

Get dataset

Preprocess dataset

(replace missing value, etc.)

Analyze dataset for different type of Machine Learning Algorithm

Apply Supervised Learning Algorithm

Create training set and testing set

Do cross-validation

Show the accuracy

If supervised

?

The initial phase of implementation level is start where it is followed by the get dataset and after getting the dataset it is going to preprocess the dataset attributes such as age,sex,cp, etc.., if it is missing x value it is going to be replaced then after it will be analyze the dataset for different type of machine learning algorithm if it is supervised then it is going to create training set and testing set then performs the cross validation and shows the accuracy rate and if it is not supervised then it is going to be terminated instantly.

**3.9 Python:-**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms.

The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, <https://www.python.org/>, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation.

The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

This introduces the reader informally to the basic concepts and features of the Python language and system. It helps to have a Python interpreter handy for hands-on experience, but all examples are self-contained, so the tutorial can be read off-line as well.For a description of standard objects and modules, see [The Python Standard Library](https://docs.python.org/2/library/index.html#library-index). [The Python Language Reference](https://docs.python.org/2/reference/index.html#reference-index) gives a more formal definition of the language. To write extensions in C or C++, read [Extending and Embedding the Python Interpreter](https://docs.python.org/2/extending/index.html#extending-index) and [Python/C API Reference Manual](https://docs.python.org/2/c-api/index.html#c-api-index). There are also several books covering Python in depth.

This does not attempt to be comprehensive and cover every single feature, or even every commonly used feature. Instead, it introduces many of Python’s most noteworthy features, and will give you a good idea of the language’s flavor and style. After reading it, you will be able to read and write Python modules and programs, and you will be ready to learn more about the various Python library modules described in [The Python Standard Library](https://docs.python.org/2/library/index.html#library-index).

**Libraries used in this:-**

* **SKLearn**

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

The library is built upon the SciPy (Scientific Python) that must be installed before you can use scikit-learn. This stack that includes:

* **Numpy**

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

* **Pandas**

Pandas is a popular Python package for data science, and with good reason: it offers powerful, expressive and flexible data structures that make data manipulation and analysis easy, among many other things.

* **Matplotlib**

[Matplotlib.pyplot](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html#module-matplotlib.pyplot) is a collection of command style functions that make matplotlib work like MATLAB. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

In [matplotlib.pyplot](https://matplotlib.org/api/_as_gen/matplotlib.pyplot.html" \l "module-matplotlib.pyplot" \o "matplotlib.pyplot) various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes.

**3.9.1 What is Keras?**

Keras is a minimalist Python library for machine learning that can run on top of Theano or TensorFlow.

It was developed to make implementing deep learning models as fast and easy as possible for research and development.

It runs on Python 2.7 or 3.5 and can seamlessly execute on GPUs and CPUs given the underlying frameworks. It is released under the permissive MIT license.

* **Sequential**

The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.

* **Dropout**

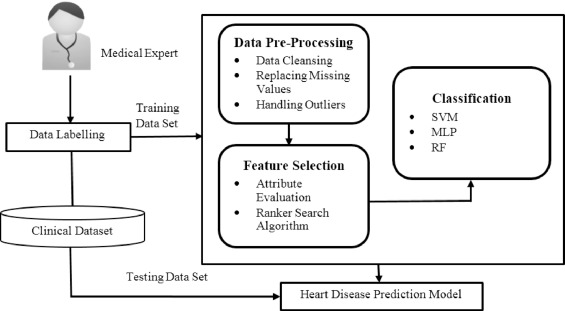
Dropout is a regularization technique for neural network models proposed by Srivastava, et al. in their 2014 paper Dropout: A Simple Way to Prevent Neural Networks from Over fitting (download the PDF). Dropout is a technique where randomly selected neurons are ignored during training. They are “dropped-out” randomly.

* **Dense layer**

A linear operation in which every input is connected to every output by a weight (so there are n\_inputs \* n\_outputs weights - which can be a lot!). Generally followed by a non-linear activation function. Convolutional layer a linear operation using a subset of the weights of a dense layer.

**4. SYSTEM DESIGN**

**4.1 System Architecture:-**



**Fig 4.1: System Architecture**

The initial phase of implementation level is start where it is followed by the get dataset and after getting the dataset it is going to preprocess the dataset attributes such as age,sex,cp, etc.., if it is missing x value it is going to be replaced then after it will be analyze the dataset for different type of machine learning algorithm if it is supervised then it is going to create training set and testing set then performs the cross validation and shows the accuracy rate and if it is not supervised then it is going to be terminated instantly.

**4.2 Data Flow Diagram / Flow Chart:-**

**5. IMPLEMENTATION**

**5.1 MODULES:-**

1. Data Preprocessing

2. Attribute evaluation

3. Making the Predictions

4. Visualization of Results

**5.1.1 Data Preprocessing:-**

In this data preprocessing process we import the training data set pandas to read the input data set. For scaling the giving data set we use Min-Max scalar to scale the values range (0, 1).

We have create a data structure with 60 time steps and 1 output, the taken data set has to be divided into x-train and y-train according to our need.

**5.1.2 Attribute evaluation:-**

K-Fold cross validation is used to minimize the bias associated with random sampling of training and test data samples in comparing predictive accuracy of two or more methods. Here the whole data set is randomly split into ‘k’ (in our case k=10) mutually exclusive subsets of approximately equal size. Classification model is trained and tested k times. The classification performance is evaluated by accuracy (ACC); [sensitivity](https://www.omicsonline.org/scholarly/sensitivity-analysis-journals-articles-ppts-list.php) (Sen); specificity (Spec) rates, and the positive predicted value (PPV) and negative predicted value (NPV), based on values residing in a confusion matrix (see **Table 2**).

|  | | **Predicted classes** | |
| --- | --- | --- | --- |
| **High risk** | **Low risk** |
| Expected/Actual | High risk | TP | FN |
| Classes | Low risk | FP | TN |

**Table 2:** Confusion matrix.

Assume that the cardiovascular classifier output set includes two typically risk prediction classes as: “High risk”, and “Low risk”. Each pattern xi (i=1, 2..n) is allocated into one element from the set (P, N) (positive or negative) of the risk prediction classes.

Hence, each input pattern might be mapped into one of four possible outcomes: true positive true high risk (TP) when the outcome is correctly predicted as High risk; true negativetrue low risk (TN) when the outcome is correctly predicted as Low risk; false negative-false Low risk (FN) when the outcome is incorrectly predicted as Low risk, when it is High risk (positive); or false positivefalse high risk (FP) when the outcome is incorrectly predicted as High risk, when it is Low risk (negative). The set of (P, N) and the predicted risk set can be built as a confusion matrix.

The accuracy of a classifier is calculated by:

Equation

The sensitivity is the rate of number correctly predicted “High risk” over the total number of correctly predicted “High risk” and incorrectly predicted “Low risk”. It is given by:

Equation

The specificity rate is the rate of correctly predicted “Low risk” over the total number of expected/actual “Low risk”. It is given by:

Equation

Higher accuracy does not always reflect a good classification outcome. For clinical data analysis it is important to evaluate the classifier based on how well the classifier predicts the “High Risk” patients. In many cases it has been found that the classification outcome is showing good accuracy as it can predict well the low risk patients (majority class) but failed to predict high risk patients (the minority class). For completeness, we also show positive predictive value (PPV) and negative predictive value (NPV), where

Equation

Equation

**5.1.3 Making the predictions:-**

To make predictions we have to read the test data set and reshape the data set according our requirements.

**5.1.4 Visualization of Results:-**

The predictions made are visualized on a graph using matplotlib that graph includes X label Y label which is accuracy rate and predicted heart disease rate of given data set. There are different attributes that are used to visualize the graph by color.

**5.2 Code Snippets:-**

from numpy import genfromtxt

import numpy as np

import matplotlib

matplotlib.use('Agg')

import matplotlib.pyplot as plt

from sklearn.svm import LinearSVC

from sklearn.decomposition import PCA

import pylab as pl

from itertools import cycle

from sklearn import cross\_validation

from sklearn.svm import SVC

#Loading and pruning the data

dataset = genfromtxt('cleveland\_data.csv',dtype = float, delimiter=',')

#print dataset

X = dataset[:,0:12] #Feature Set

y = dataset[:,13] #Label Set

#Method to plot the graph for reduced Dimesions

def plot\_2D(data, target, target\_names):

colors = cycle('rgbcmykw')

target\_ids = range(len(target\_names))

plt.figure()

for i, c, label in zip(target\_ids, colors, target\_names):

plt.scatter(data[target == i, 0], data[target == i, 1],

c=c, label=label)

plt.legend()

plt.savefig('Reduced\_PCA\_Graph')

# Classifying the data using a Linear SVM and predicting the probability of disease belonging to a particular class

modelSVM = LinearSVC(C=0.001)

pca = PCA(n\_components=5, whiten=True).fit(X)

X\_new = pca.transform(X)

# calling plot\_2D

target\_names = ['0','1','2','3','4']

plot\_2D(X\_new, y, target\_names)

#Applying cross validation on the training and test set for validating our Linear SVM Model

X\_train, X\_test, y\_train, y\_test = cross\_validation.train\_test\_split(X\_new, y, test\_size=0.4, train\_size=0.6, random\_state=0)

modelSVM = modelSVM.fit(X\_train, y\_train)

print "Testing Linear SVC values using Split"

print modelSVM.score(X\_test, y\_test)

# prediction score based on X\_new

modelSVMRaw = LinearSVC(C=0.001)

modelSVMRaw = modelSVMRaw.fit(X\_new, y)

cnt = 0

for i in modelSVMRaw.predict(X\_new):

if i == y[i]:

cnt = cnt+1

print "Score without any split"

print float(cnt)/303

# printing the Likelihood of disease belonging to a particular class

# predicting the outcome

count0 = 0

count1 = 0

count2 = 0

count3 = 0

count4 = 0

for i in modelSVM.predict(X\_new):

if i == 0:

count0 = count0+1;

elif i == 1:

count1 = count1+1;

elif i == 2:

count2 = count2+1;

elif i == 3:

count3 = count3+1;

elif modelSVM.predict(i) ==4:

count4 = count4+1

total = count0+count1+count2+count3+count4

#Predicting the Likelihood

print "The prediction is as follows:"

print " Likelihood of belonging to Class 0 is", float(count0)/total

print " Likelihood of belonging to Class 1 is", float(count1)/total

print " Likelihood of belonging to Class 2 is", float(count2)/total

print " Likelihood of belonging to Class 3 is", float(count3)/total

print " Likelihood of belonging to Class 4 is", float(count4)/total

#Applying the Principal Component Analysis on the data features

modelSVM2 = SVC(C=0.001,kernel='rbf')

#Applying cross validation on the training and test set for validating our Linear SVM Model

X\_train1, X\_test1, y\_train1, y\_test1 = cross\_validation.train\_test\_split(X\_new, y, test\_size=0.4, train\_size=0.6, random\_state=0)

modelSVM2 = modelSVM2.fit(X\_train1, y\_train1)

print "Testing with RBF using split"

print modelSVM2.score(X\_test1, y\_test1)

modelSVM2Raw = SVC(C=0.001,kernel='rbf')

modelSVM2Raw = modelSVM2Raw.fit(X\_new, y)

cnt1 = 0

for i in modelSVM2Raw.predict(X\_new):

if i == y[i]:

cnt1 = cnt1+1

print "RBF Score without split"

print float(cnt1)/303

#Using Stratified K Fold

skf = cross\_validation.StratifiedKFold(y, n\_folds=5)

for train\_index, test\_index in skf:

# print("TRAIN:", train\_index, "TEST:", test\_index)

X\_train3, X\_test3 = X[train\_index], X[test\_index]

y\_train3, y\_test3 = y[train\_index], y[test\_index]

modelSVM3 = SVC(C=0.001,kernel='rbf')

modelSVM3 = modelSVM3.fit(X\_train3, y\_train3)

print "Testing using stratified with K folds"

print modelSVM3.score(X\_test3, y\_test3)

modelSVM3Raw = SVC(C=0.001,kernel='rbf')

modelSVM3Raw = modelSVM3Raw.fit(X\_new, y)

cnt2 = 0

for i in modelSVM3Raw.predict(X\_new):

if i == y[i]:

cnt2 = cnt2+1

print "Stratified K Fold score on X\_New"

print float(cnt2)/303

**6. TESTING**

**6.1 System Testing:**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**6.2 Types of Tests:**

**Unit Testing:**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration Testing:**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional Test:**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Testing:**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing:**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach:**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives:**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested:**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

# Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

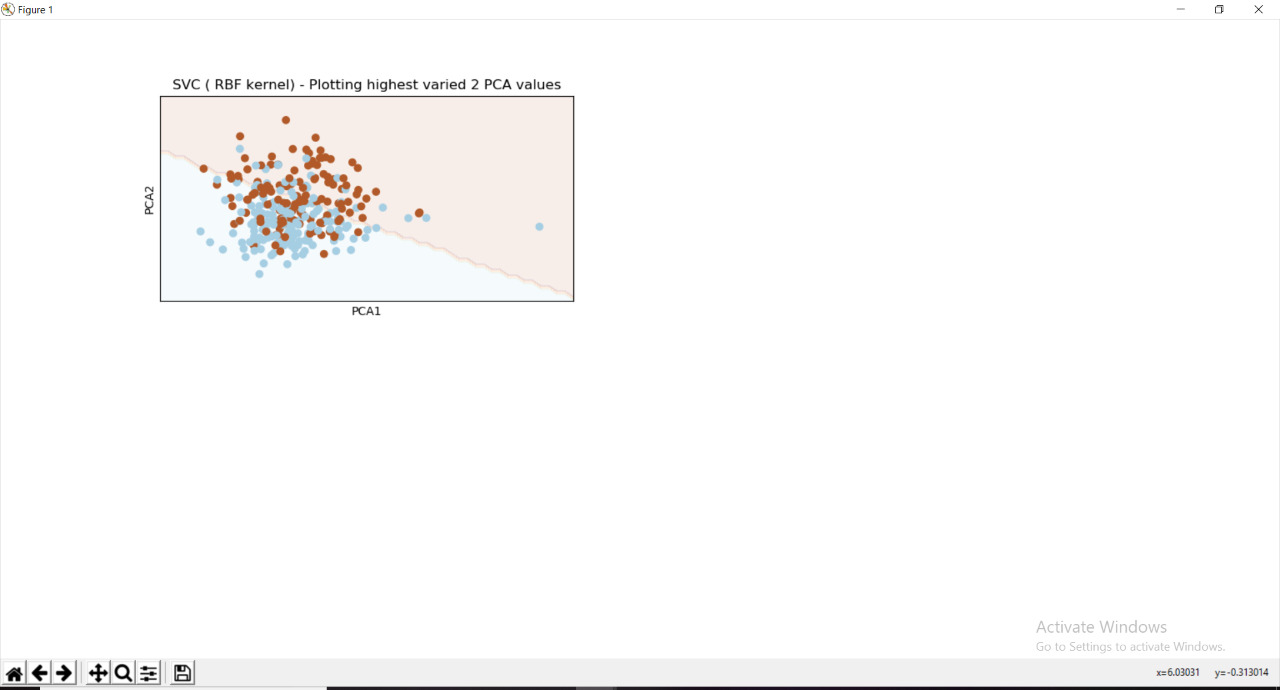
**Acceptance Testing:**

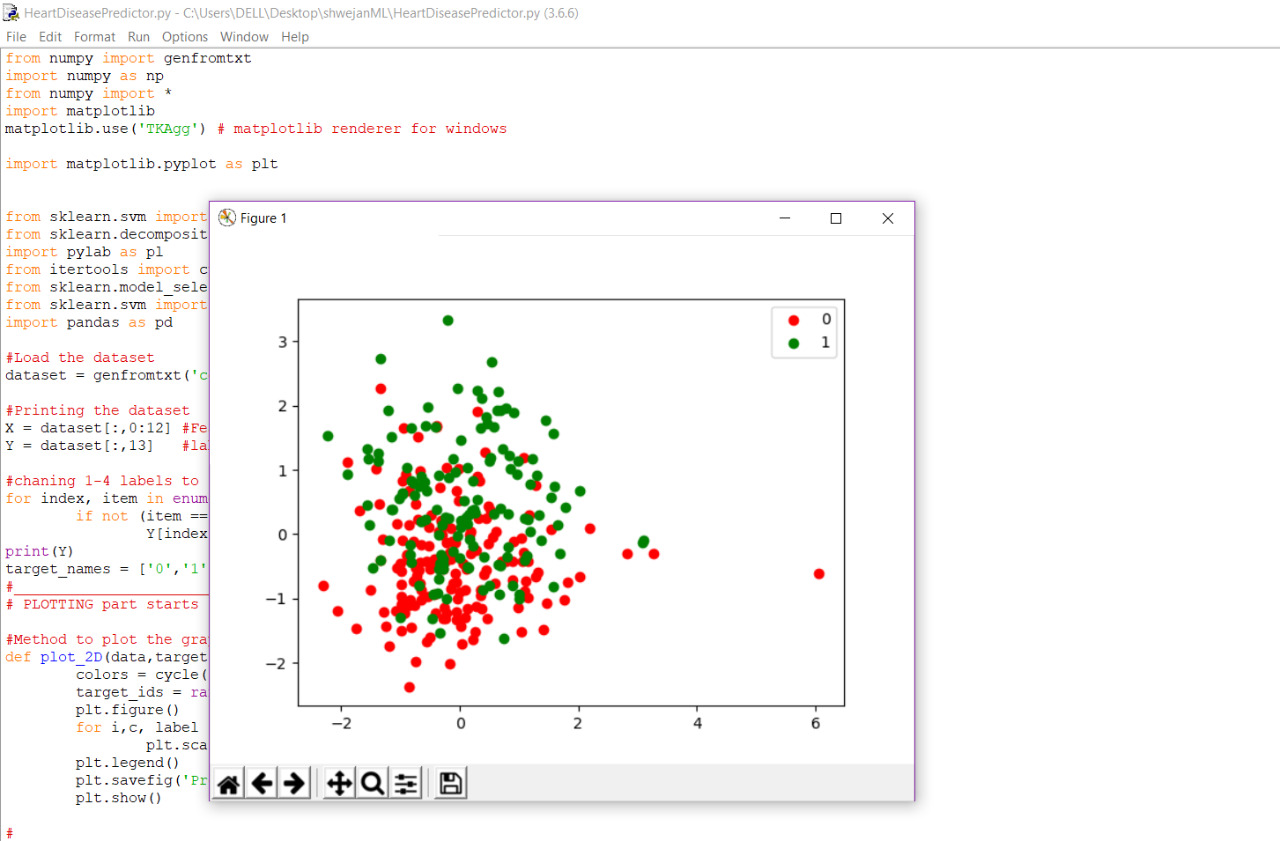
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7. SCREENSHOTS**

**7.1 RESULTS:-**

****

****

**8.CONCLUSION:-**

So,we can conclude that we can get the accurate result with accuracy of 89.7% from this project which uses machine learning techniques such as support vector machine(SVM) and random forests.

By this we can conclude that by using

LINEAR SVM we can acquire an classification accuracy of 83% without split

LINEAR SVM we can acquire an classification accuracy of 73.77% with split

NON-LINEAR SVM we can acquire an classification accuracy of 66.7% without split

NON-LINEAR SVM we can acquire an classification accuracy of 57.37% with split

**9.REFERENCES**

**1. SKLEARN OFFICIAL DOCUMENT**

**2. NPTEL MACHINE LEARNING COURSES**

**3. UDEMY MACHINE LEARNING COURSE USING PYTHON**