**CHAPTER 4-Methodology**

**Introduction:**

In these chapter we are going to understand the need of machine learning algorithm. How we can use machine learning in healthcare to process huge datasets beyond the scope of human capability is discussed in these chapter. For prediction of heart disease using machine learning algorithm there are some steps such as identification of attributes which include age, sex, chest pain type, serum cholesterol, fasting blood sugar and many more which are discussed briefly below. For processing these attributes we need machine learning algorithms

for example SVM,KNN, Decision Tree, Logistic Regression, Random Forest .The working of these algorithms and there implementation is discussed below. Algorithms have some advantages as well as some disadvantages. How the disadvantage of one algorithm is solved by another algorithm is also discussed below. Visualization of test results using histograms, graphs, images is also explained in these chapter.

**Fig No 4.1 Block Diagram showing Proposed Methodology**

**Understanding the need of Machine Learning algorithm**

Machine Learning is the core subarea of artificial intelligence. It makes computers get into a self-learning mode without explicit programming. When fed new data, these computers learn, grow, change, and develop by themselves.  The value of machine learning in healthcare is its ability to process huge datasets beyond the scope of human capability, and then reliably convert analysis of that data into clinical insights that aid physicians in planning and providing care, ultimately leading to better outcomes, lower costs of care, and increased patient satisfaction.

**Collecting the information for dataset**

This step in the checklist is akin to what is often referred to as Exploratory Data Analysis (EDA). The goal is to try and gain insights from the data prior to modeling. Recall that in the first step assumptions about the data were to be identified and explored; this is a good time to more deeply investigate these assumptions. Human experts can be of particular use in this step, answering questions about correlations which may not be obvious to the machine learning practitioner. Studying features and their characteristics is done here, as is general visualization of features and their values (think of how much easier it is, for example, to quickly identify outliers by box plot than by numerical interrogation). Documenting the findings of your exploration for later use is good practice.

**Identification of attributes causing Cardiac Arrest and pre-processing on it**

The dataset consists of 303 individual data. There are 14 columns in the dataset, which are described below.

1. **Age:** displays the age of the individual.
2. **Sex:** displays the gender of the individual using the following format :  
   1 = male  
   0 = female
3. **Chest-pain type:** displays the type of chest-pain experienced by the individual using the following format:  
   1 = typical angina  
   2 = atypical angina  
   3 = non — anginal pain  
   4 = asymptotic
4. **Resting Blood Pressure:** displays the resting blood pressure value of an individual in mmHg (unit)
5. **Serum Cholesterol:** displays the serum cholesterol in mg/dl (unit)
6. **Fasting Blood Sugar:** compares the fasting blood sugar value of an individual with 120mg/dl.  
   If fasting blood sugar > 120mg/dl then: 1 (true)  
   else: 0 (false)
7. **Resting ECG:** displays resting electrocardiographic results  
   0 = normal  
   1 = having ST-T wave abnormality  
   2 = left ventricular hyperthrophy
8. **Max heart rate achieved:** displays the max heart rate achieved by an individual.
9. Exercise induced angina:  
   1 = yes  
   0 = no
10. **ST depression induced by exercise relative to rest:** displays the value which is an integer or float.
11. **Peak exercise ST segment:**  
    1 = upsloping  
    2 = flat  
    3 = down sloping
12. **Number of major vessels (0–3) colored by fluoroscopy:** displays the value as integer or float.
13. **Thal:** displays the thalassemia:  
    3 = normal  
    6 = fixed defect  
    7 = reversible defect
14. **Diagnosis of heart disease:** Displays whether the individual is suffering from heart disease or not:  
    0 = absence  
    1, 2, 3, 4 = present.

**Fitting the classifier algorithm to the Training set**

* Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes. A model that is overfitted matches the data too closely. A model that is underfitted doesn’t match closely enough. Each machine learning algorithm has a basic set of parameters that can be changed to improve its accuracy.
* During the fitting process, you run an algorithm on data for which you know the target variable, known as “labeled” data, and produce a machine learning model. Then, you compare the outcomes to real, observed values of the target variable to determine their accuracy.
* Next, we use that information to adjust the algorithm’s standard parameters to reduce the level of error, making it more accurate in uncovering patterns and relationships between the rest of its features and the target.
* We repeat this process until the algorithm finds the optimal parameters that produce valid, practical, applicable insights for our practical medical application problem.

**Predicting the test result**

* Our primary focus for reducing waste in healthcare expenditure is identifying and discouraging unnecessary repeat lab tests. A machine learning model which could reliably predict low information lab tests could provide personalized, real-time predictions to discourage over- testing.to predict when the next measurement of a lab test is likely to be the “same” as the previous one.
* This points to potential areas where machine learning approaches may identify and prevent unneeded testing before it occurs, and a methodological framework for how these tasks can be accomplished.

**Test accuracy of the result (Creation of Confusion matrix)**

* A confusion matrix is a technique for summarizing the performance of a classification algorithm.
* Classification accuracy is the ratio of correct predictions to total predictions made.
* The confusion matrix shows the ways in which your classification model is confused when it makes predictions.
* It gives you insight not only into the errors being made by your classifier but more importantly the types of errors that are being made.
* It is this breakdown that overcomes the limitation of using classification accuracy alone.

**Visualizing the test set result**

* During the training of models, it is necessary to visualize the training process and helps to understand debug models and track errors and progress of the models. Visualize the test and train result values at each step.
* Scalars (Loss and Accuracy) – Scalar can be used to show the trends of error during training. Besides logging the loss and accuracy to the stdout regularly, we record and plot them to analyze its long-term trend.
* Histograms – Visualize how the distribution of tensors in the model graph has changed over time. Showing many histograms visualizations of tensor at different points in time.
* Weight and bias –Monitor the weights and the biases during training time by visualizing them on histograms.
* Activation – For gradient descent to perform the best, the node outputs before the activation functions usually distributed.
* Gradients – Gradients can be visualized for each layer to identify deep learning problems like gradient diminishing or exploding problems.
* Graphs –Graphs visualize the internal structure or architecture of the model.
* Image –Images at each step of training means intermediate images generated can be visualized and visualize tensors.
* Projector – Visualize the principal component analysis and t-sne algorithm model results. This technique mainly uses for dimensionality reduction.

**Algorithms Used**

**Support Vector Machines (SVM)**

* A support vector machine (SVM) is a supervised [machine learning](http://www.monkeylearn.com/machine-learning/) model that uses classification algorithms for two-group classification problems. After giving an SVM model sets of labelled training data for each category, they’re able to categorize new text. So SVM is a fast and dependable classification algorithm that performs very well with a limited amount of [data to analyse](http://www.monkeylearn.com/data-analysis/).
* SVM algorithm can be used for **Face detection, image classification, text categorization,** etc.
* [SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC), [NuSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVC.html#sklearn.svm.NuSVC) and [LinearSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC) are classes capable of performing binary and multi-class classification on a dataset. [SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC) and [NuSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVC.html#sklearn.svm.NuSVC) are similar methods, but accept slightly different sets of parameters and have different mathematical formulations (see section [Mathematical formulation](https://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation)). On the other hand, [LinearSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC) is another (faster) implementation of Support Vector Classification for the case of a linear kernel. Note that [LinearSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC) does not accept parameter kernel, as this is assumed to be linear. It also lacks some of the attributes of [SVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html#sklearn.svm.SVC) and [NuSVC](https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVC.html#sklearn.svm.NuSVC), like support\_.
* SVM can be of two types:

1. Linear SVM: Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
2. Non-linear SVM: Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

* SVM can be understood with the example that we have used in the KNN classifier. Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog, so such a model can be created by using the SVM algorithm. We will first train our model with lots of images of cats and dogs so that it can learn about different features of cats and dogs, and then we test it with this strange creature. So as support vector creates a decision boundary between these two data (cat and dog) and choose extreme cases (support vectors), it will see the extreme case of cat and dog. On the basis of the support vectors, it will classify it as a cat. Consider the below diagram:



Fig No 4.2 SVM Model

**Advantages of support vector machines**

* Effective in high dimensional spaces.
* Still effective in cases where number of dimensions is greater than the number of samples.
* Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
* Versatile: different [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

**Disadvantages of support vector machines**

* If the number of features is much greater than the number of samples, avoid over-fitting in choosing [Kernel functions](https://scikit-learn.org/stable/modules/svm.html#svm-kernels) and regularization term is crucial.
* SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.

## **Working of SVM**

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

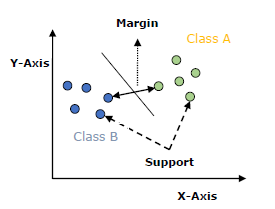


Fig No 4.3 SVM Graph

The followings are important concepts in SVM −

* Support Vectors − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.
* Hyperplane − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.
* Margin − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps −

* First, SVM will generate hyperplanes iteratively that segregates the classes in best way.
* Then, it will choose the hyperplane that separates the classes correctly.

**Algorithm of SVC:**

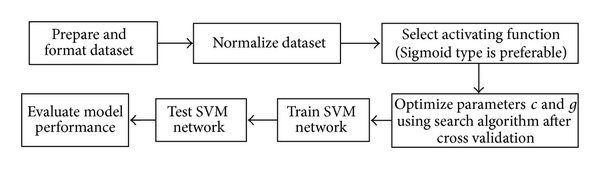


Fig No 4.4 SVM Block Diagram.

* Step 1: In this step we prepare the dataset and format it according to our needs.
* Step 2: Here we reduce data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
* Step 3: Selection of an activating function for the algorithm.
* Step 4: Optimization of parameters using search algorithm.
* Step 5: Training of SVM by the help of dataset collected in step 1.
* Step 6: Testing of the model performance in terms of accuracy/precision of the model.
* Step 7: Evaluation of model’s performance on various aspects.

**Decision Tree Algorithm**

A decision tree is a flowchart tree-like structure that is made from training set tuples. The dataset is broken down into smaller subsets and is present in the form of nodes of a tree. The tree structure has a root node, internal nodes or decision nodes, leaf node, and branches.

The root node is the topmost node. It represents the best attribute selected for classification. Internal nodes of the decision nodes represent a test of an attribute of the dataset leaf node or terminal node which represents the classification or decision label. The branches show the outcome of the test performed.

* A decision tree is a supervised learning algorithm that works for both discrete and continuous variables. It splits the dataset into subsets on the basis of the most significant attribute in the dataset. How the decision tree identifies this attribute and how this splitting is done is decided by the algorithms.
* The most significant predictor is designated as the root node, splitting is done to form sub-nodes called decision nodes, and the nodes which do not split further are terminal or leaf nodes.
* In the decision tree, the dataset is divided into homogeneous and non-overlapping regions. It follows a top-down approach as the top region presents all the observations at a single place which splits into two or more branches that further split. This approach is also called a greedy approach as it only considers the current node between the worked on without focusing on the future nodes.
* The decision tree algorithms will continue running until a stop criteria such as the minimum number of observations etc. is reached.
* Once a decision tree is built, many nodes may represent outliers or noisy data. Tree pruning method is applied to remove unwanted data. This, in turn, improves the accuracy of the classification model.
* To find the accuracy of the model, a test set consisting of test tuples and class labels is used. The percentages of the test set tuples are correctly classified by the model to identify the accuracy of the model. If the model is found to be accurate then it is used to classify the data tuples for which the class labels are not known.
* Some of the decision tree algorithms include Hunt’s Algorithm, ID3, CD4.5, and CART.
* Decision tree algorithm is the process in which the data given is first split according to the given decision. The data splitting is based on which category thus the data belong (e.g.: age, height, material, etc). Then the split data is compared to the condition given. Then the compared data is checked weather it satisfies the condition. If satisfy then the value is 1 or high and weather not satisfied the value is 0 or low then the high data is considered as a result and further process is taken. There are many steps that are involved in the working of a decision tree. It is the process of the partitioning of data into subsets. Splitting can be done on various factors as shown below i.e. on gender basis, height basis or based on class.

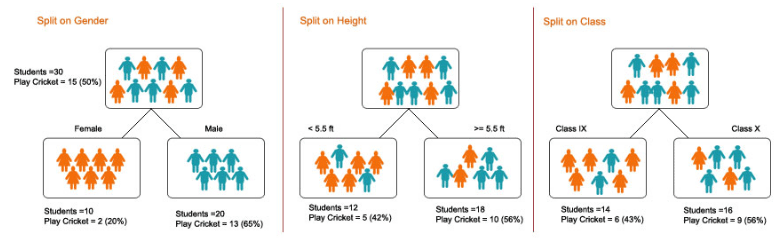


Fig No 4.5 Decision Tree Model

**Advantages of Decision tree**

* It is very easy to understand and interpret.
* The data for decision trees require minimal preparation.
* They force you to find many possible outcomes of a decision.
* Can be easily used with many other decision tools.
* Helps you to make the best decisions and best guesses on the basis of the information you have.
* Helps you to see the difference between controlled and uncontrolled events.
* Helps you estimate the likely results of one decision against another.

**Disadvantages of Decision tree**

* Sometimes decision trees can become too complex.
* The outcomes of decisions may be based mainly on your expectations. This can lead to unrealistic decision trees.
* The diagrams can narrow your focus to critical decisions and objectives.

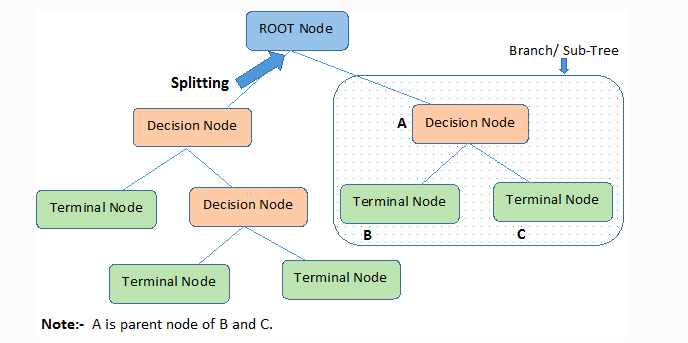
**Flow Chart**

Fig No 4.6 Decision Tree Flowchart

* Step 1: Root Node It represents the entire population or sample and this further gets divided into two or more homogeneous sets.
* Step 2: Splitting is a process of dividing a node into two or more sub-nodes.
* Step 3: When a sub-node splits into further sub-nodes, then it is called the decision node.
* Step 4: A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.
* Step 5: When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
* Step 6: A subsection of the entire tree is called branch or sub-tree.
* Step 7: Nodes do not split is called Leaf or Terminal node.
* Step 8: Here A is parent node B and C.

**K-Nearest Neighbour (KNN) Algorithm**

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.

* K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
* It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
* KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
* Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So, for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cat’s and dog’s images and based on the most similar features it will put it in either cat or dog category.



Fig No 4.7 KNN Model

## **Need of K-NN Algorithm:**

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

## 

Fig No 4.8 KNN Graphs

**Working Of K-NN:**

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbours
* **Step-2:** Calculate the Euclidean distance of **K number of neighbours**
* **Step-3:** Take the K nearest neighbours as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbours, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbour is maximum.
* **Step-6:** Our model is ready.

Suppose we have a new data point and we need to put it in the required category. Consider the below image:



Fig No 4.9 KNN Graph

* Firstly, we will choose the number of neighbours, so we will choose the k=5.
* Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:



Fig No 4.10 Euclidean Distance (KNN)

* As we can see the 3 nearest neighbours are from category A, hence this new data point must belong to category A.

## **Selection of K value in the K-NN Algorithm**

Below are some points to remember while selecting the value of K in the K-NN algorithm:

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

## **Advantages of KNN**

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

## **Disadvantages of KNN**

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

**Steps to implement the K-NN algorithm:**

* Data Pre-processing step
* Fitting the K-NN algorithm to the Training set
* Predicting the test result
* Test accuracy of the result(Creation of Confusion matrix)
* Visualizing the test set result.

**Random Forest Algorithm**

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, *"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.

The below diagram explains the working of the Random Forest algorithm:



Fig No 4.11 Random Forest Model

**Assumptions for Random Forest**

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

* There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
* The predictions from each tree must have very low correlations.

**Use of Random Forest:**

Below are some points that explain why we should use the Random Forest algorithm:

* It takes less training time as compared to other algorithms.
* It predicts output with high accuracy, even for the large dataset it runs efficiently.
* It can also maintain accuracy when a large proportion of data is missing.

**Working of Random Forest algorithm:**

* Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.
* The Working process can be explained in the below steps and diagram:
  + Step-1: Select random K data points from the training set.
  + Step-2: Build the decision trees associated with the selected data points (Subsets).
  + Step-3: Choose the number N for decision trees that you want to build.
  + Step-4: Repeat Step 1 & 2.
  + Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.
* The working of the algorithm can be better understood by the below example:

Example: Suppose there is a dataset that contains multiple fruit images. So, this dataset is given to the Random forest classifier. The dataset is divided into subsets and given to each decision tree. During the training phase, each decision tree produces a prediction result, and when a new data point occurs, then based on the majority of results, the Random Forest classifier predicts the final decision. Consider the below image:



Fig No 4.12 Random Forest (Fruit) Example

**Applications of Random Forest**

There are mainly four sectors where Random forest mostly used:

* Banking: Banking sector mostly uses this algorithm for the identification of loan risk.
* Medicine: With the help of this algorithm, disease trends and risks of the disease can be identified.
* Land Use: We can identify the areas of similar land use by this algorithm.
* Marketing: Marketing trends can be identified using this algorithm.

**Advantages of Random Forest**

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

**Disadvantages of Random Forest**

Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

**Python Implementation of Random Forest Algorithm**

* Now we will implement the Random Forest Algorithm tree using Python. For this, we will use the same dataset "user\_data.csv", which we have used in previous classification models. By using the same dataset, we can compare the Random Forest classifier with other classification models such as [Decision tree Classifier,](https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm) [KNN,](https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning)[SVM,](https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm) [Logistic Regression,](https://www.javatpoint.com/logistic-regression-in-machine-learning) etc.
* Implementation Steps are given below:
  + Data Pre-processing step
  + Fitting the Random forest algorithm to the Training set
  + Predicting the test result
  + Test accuracy of the result (Creation of Confusion matrix)
  + Visualizing the test set result.

**Logistic Regression**

* Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
* Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
* Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
* In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
* The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
* Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
* Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:



Fig No 4.13 Logistic Regression Graph

## **Type of Logistic Regression:**

On the basis of the categories, Logistic Regression can be classified into three types:

* **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
* **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
* **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Steps in Logistic Regression**

To implement the Logistic Regression using Python, we will use the same steps as we have done in previous topics of Regression. Below are the steps:

* Data Pre-processing step
* Fitting Logistic Regression to the Training set
* Predicting the test result
* Test accuracy of the result (Creation of Confusion matrix)
* Visualizing the test set result.

**Working of Logistic Regression:**

* Logistic Regression measures the relationship between the dependent variable (our label, what we want to predict) and the one or more independent variables (our features), by estimating probabilities using it’s underlying logistic function.
* These probabilities must then be transformed into binary values in order to actually make a prediction. This is the task of the logistic function, also called the sigmoid function. The Sigmoid-Function is an S-shaped curve that can take any real-valued number and map it into a value between the range of 0 and 1, but never exactly at those limits. This values between 0 and 1 will then be transformed into either 0 or 1 using a threshold classifier.
* The picture below illustrates the steps that logistic regression goes through to give you your desired output.

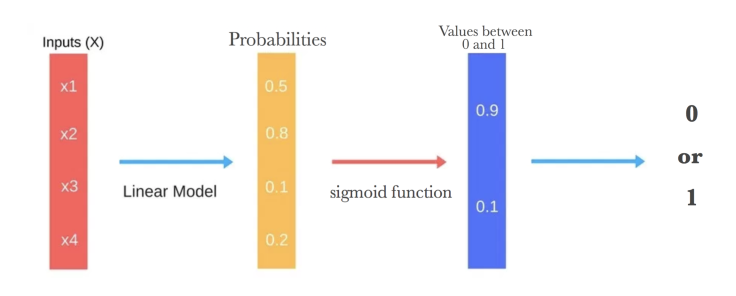


Fig No 4.14 Steps of Logistic Regression

Below you can see how the logistic function (sigmoid function) looks like:

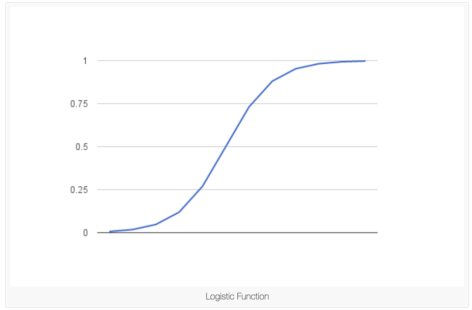


Fig No 4.15 Logistic function (sigmoid function)

* We want to maximize the likelihood that a random data point gets classified correctly, which is called Maximum Likelihood Estimation. Maximum Likelihood Estimation is a general approach to estimating parameters in statistical models.
* We can maximize the likelihood using different methods like an optimization algorithm. Newton’s Method is such an algorithm and can be used to find maximum (or minimum) of many different functions, including the likelihood function. Instead of Newton’s Method, you could also use Gradient Descent.

## **Advantages**

* Logistic Regression is **one of the simplest machine learning algorithms** and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
* The predicted parameters (trained weights) give **inference about the importance of each feature**. The direction of association i.e. positive or negative is also given. So we can use logistic regression to find out the relationship between the features.
* This algorithm allows models to be **updated easily to reflect new data**, unlike decision trees or support vector machines. The update can be done using stochastic gradient descent.
* Logistic Regression **outputs well-calibrated probabilities** along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.

## **Disadvantages**

* Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On **high dimensional datasets**, this may lead to the model being **over-fit on the training set**, which means overstating the accuracy of predictions on the training set and thus the model **may not be able to predict accurate results on the test set**. This usually happens in the case when the model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.
* **Non linear problems can't be solved** with logistic regression **since it has a linear decision surface**. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.
* It is **difficult to capture complex relationships** using logistic regression. More powerful and complex algorithms such as Neural Networks can easily outperform this algorithm.
* The training features are known as independent variables. Logistic Regression **requires moderate or no multicollinearity between independent variables**. This means if two independent variables have a high correlation, only one of them should be used. **Repetition of information could lead to wrong training of parameters** (weights) during minimizing the cost function. Multicollinearity can be removed using dimensionality reduction techniques.

References:

* https://www.javatpoint.com/k-nearest-neighbor-algorithm-for- machine-learning
* https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm
* https://scholar.google.co.in/scholar?q=prediction+of+heart+disease+using+machine+learning+algorithms&hl=en&as\_sdt=0&as\_vis=1&oi=scholart
* https://www.kaggle.com/faressayah/predicting-heart-disease-using-machine-learning
* https://www.researchgate.net/publication/326733163\_Prediction\_of\_Heart\_Disease\_Using\_Machine\_Learning\_Algorithms
* https://www.javatpoint.com/machine-learning