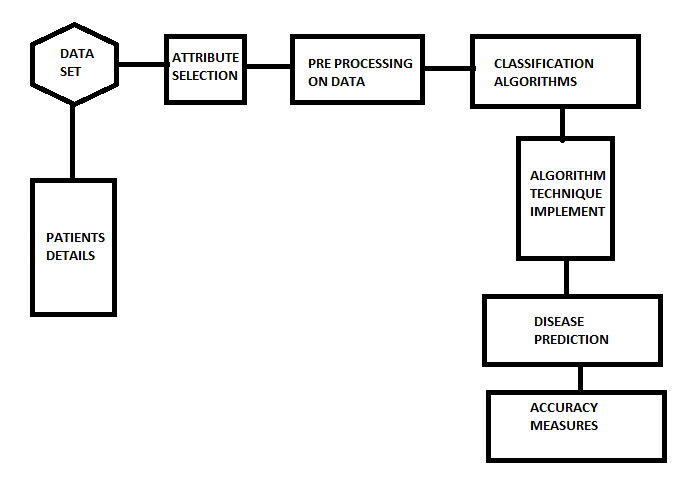
METHODOLOGY

Architecture Diagram and Explaination:



The working of this system is described in a step by step:

1. Dataset collection which contains patient details.

2. Attributes selection process selects the useful attributes for the prediction of heart disease.

3. After identifying the available data resources, they are further selected, cleaned, made into the desired form.

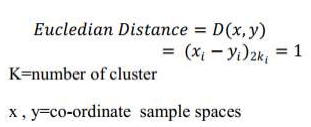
4. Different classification techniques as stated will be applied on preprocessed data to predict the accuracy of heart disease.

5. Accuracy measure compares the accuracy of different classifiers

Algorithm wise methodology and Explanation:

K- NEAREST NEIGHBOUR ALGORITHM (KNN):

KNN is slow supervised learning algorithm, it take more time to get trained classification like other algorithm is divided into two step training from data and testing it on new instance . The K Nearest Neighbour working principle is based on assignment of weight to the each data point which is called as neighbour. In K Nearest Neighbour distance is calculate for training dataset for each of the K Nearest data points now classification is done on basis of majority of votes there are three types of distances need to be measured in KNN Euclidian, Manhattan, Minkowski distance in which Euclidian will be consider most one the following formula is used to calculate their distance.



The algorithm for KNN is defined in the steps given below: 1. D represents the samples used in the training and k denotes the number of nearest neighbour. 2. Create super class for each sample class. 3. Compute Euclidian distance for every training sample 4. Based on majority of class in neighbour, classify the sample

Random Forest Algorithm:

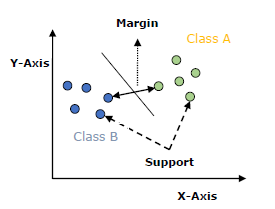
Random forest or decision tree is a method for classifiaction and other tasks that operates by constructing a multitude of decision tree at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

A decision tree is built by determining the questions (called splits of nodes) that, when answered, lead to the greatest reduction in Gini impurity. What this means is the decision tree tries to form nodes containing a high proportion of samples (data points) from a single class by finding values in the features that cleanly divide the data into classes.

Support Vector Machine (SVM) :

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.

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).



* **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.