**Software used:**

### HARDWARE REQUIREMENTS:

* System : Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz
* Hard Disk : 1 TB.
* Input Devices : Keyboard, Mouse
* Ram : 4 GB.

### SOFTWARE REQUIREMENTS:

* Operating system : Windows XP/7/10.
* Coding Language : Python
* Tool : Anaconda
* Interface : flask webapp
* Tool : For training Model Jupiter Notebook
* Libraries : keras, tensorflow, sklearn, numpy, pandas, flask

**Methodology:**

**Water fall Model:**

## **METHODOLOGY FOLLOWED**

**The steps followed to do this project are:**

* + 1. **Collection of datasets.**

Heart, kidney, diabetic, liver disease datasets are collected form Kaggle website which are in the form of csv format. These datasets have features and labels based on type of disease dataset we are using features and labels are changed.

* + 1. Understanding features of dataset.

**Cancer Dataset:**

**Features:**

id radius\_mean texture\_mean perimeter\_mean area\_mean smoothness\_mean compactness\_mean concavity\_mean concave points\_mean symmetry\_mean fractal\_dimension\_mean radius\_se texture\_se perimeter\_se area\_se smoothness\_se compactness\_se concavity\_se concave points\_se symmetry\_se fractal\_dimension\_se radius\_worst texture\_worst perimeter\_worst area\_worst smoothness\_worst compactness\_worst concavity\_worst concave points\_worst symmetry\_worst fractal\_dimension\_worst

Labels:

Diagnosis

**Diabetic Disease:**

**Features:**

Pregnancies Glucose BloodPressure SkinThickness Insulin BMI DiabetesPedigreeFunction Age Outcome

**Labels:**

Outcome

* + 1. **Pre-processing the data.**

In this stage data analysis of each dataset is performed to check relation between features and labels with graphical representation. Null values are removed from the dataset and balanced dataset is prepared for all diseases datasets.

* + 1. **Split data into training dataset and testing dataset.**

Data set is split in to two parts using test train split function ( 80 and 20 ) as test and train datasets. Train features are called as train x and labels as train y. These values are used to train algorithm and test data is used to check accuracy of each disease dataset**.**

* + 1. **Apply ML algorithms to dataset to predict the heart disease, diabetic, kidney disease, Liver diseases.**

In this stage pre processed dataset is taken as input of each disease dataset and trained features and labels are given as input to fit function to train model and model is saved in to system in the form of pkl file. The model is used in web application for prediction results based on user given input.

* + 1. **Accuracy results**

After training is done test set is given and input to algorithm to test accuracy of each dataset.

**Flask Web framework:**

For this project web application is developed using flask framework which takes trained model as input and html , css for web page design . Using this application own input is given to webpage and disease is predicted.

### Algorithms

#### Random Forest

Random forest is a tree-based algorithm which involves building several trees (decision trees), then combining their output to improve generalization ability of the model. The method of combining trees is known as an ensemble method. Ensembling is nothing but a combination of weak learners (individual trees) to produce a strong learner.

Definition: A random forest is a classifier consisting of a collection of tree structured classifiers *h*(*x,* Θ*k*)*, k* = 1*, ...* where the Θ*k* are independent identically distributed (*i.i.d*) random vectors and each tree casts a unit vote for the most popular class at input [[4](#_bookmark61)].

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

* + - * Pick N random records from the dataset.
      * Build a decision tree based on these N records.
      * Choose the number of trees you want in your algorithm and repeat steps (i) and (ii).
      * In case of a classification problem, each tree in the forest predicts the category to which the new record belongs. Finally, the new record is assigned to the category that wins the majority vote.

Figure [2.1](#_bookmark0) shows different trees labelling the class differently. What ensemble does is take the mode (maximum occurring class) of the output produced by n different trees to create a better model. To say it in simple words: Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction.

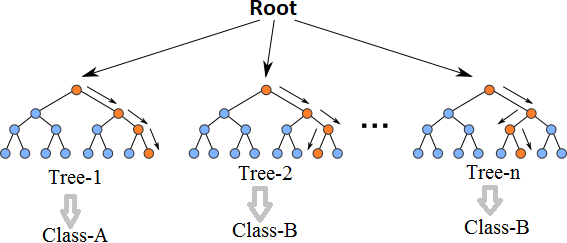


Fig. 2.1 Multiple decision trees [[12](#_bookmark69)]

Even though decision trees are pretty intuitive and easier to understand, they can be very noisy. Few changes in the data can lead to different splits and completely different models. The instability of the tree makes it unrealistic as a prediction model by itself. A single decision tree is insufficient and generally overfits the data, that is it can capture the structure of the in-sample data very well, but it tends to work poorly out-of-sample. In the context of statistics, decisions trees have low bias (as it can fit the data well) but high variances (the predictions are noisy).

Understanding the working principle of decision trees is imperative in the understanding of Random Forest Algorithm. The most popular algorithm for decision trees is ID3 algorithm. It finds the best attributes/features that best classifies the target attribute. One of the most commonly used way to figure out the best attribute is by calculating Information Gain which is, in turn, calculated using another property called Entropy.

The calculation of entropy of a system is done as follows:

*c*

*Entropy*(*S*) = ∑ *pilog*2 *pi* (2.1)

−

*i*=1

Here, c is the total number of classes or attributes and *pi* is number of examples belonging to the *ith* class. Information gain is simply the expected reduction in entropy caused by partitioning all our examples according to a given attribute. Mathematically, it is defined as:

*Gain*(*S, A*) ≡ *Entropy*(*S*) − ∑ |*Sv*| *Entropy*(*Sv*) (2.2)

*v*∈*Values*(*A*) |*S*|

S refers to the entire set of examples that we have. A is the attribute we want to partition or split. |S| is the number of examples and |*Sv*| is the number of examples for the current value of attribute A. The attribute with the highest information gain sits at the root node, and the tree is first split based on that attribute.

## **FUNCTIONAL AND NON FUNCTIONALREQUIREMENTS**

### FUNCTIONAL:

The functional requirement defines the system or the components of the system. A function is basically inputs, behaviours and outputs. Stuff that can be called functional requirements are: calculations, technical details, data manipulation and processing. It tells us what a system is supposed to do.

Here, the system has to perform the following tasks:

* Understand all the features as well as the data provided in the dataset.
* Map the data in the dataset with the given input data. Find patterns, if any, with both the dataset as well as input data.
* Check whether the input data of a patient will result in the diagnosis of breast cancer or not.
* If multiple disease is diagnosed, provide information on the type of disease with disease or without disease.
* Provide the percentage accuracy of the proposed prediction.

### NON-FUNCTIONAL:

A non-functional requirement is a requirement gives the criteria that can be used to judge how well a system can function. It comes under system/requirements engineering. It gives a judgement on the overall unlike functional requirements which define specific behaviour or functions. Functional requirements are implemented by using the system design whereas system architecture is what is used for implementing the non-functional requirements.

Non-functional requirements are also called constraints. Some of the quality attributes are as follows:

### ACCESSIBILITY:

Accessibility is a term that is used to describe if a product or software is accessible to the public and how easily can it be accessed.

It is easy to access as the dataset is open source and can be found on the University of California, Irvin’s ML dataset repository. Unlike breast cancer diagnosis tests in hospitals which cost a lot, anyone can access this dataset for free.

### MAINTAINABILITY:

Maintainability tells us how easily a software or tool or system can be modified in order to:

* + - * Correct defects
      * Meet new requirements

Different programming languages can be used to make the predictive model based on the programmer’s wishes. The datasets can also be modified and new data can be added as and when the data is updated by doctors. Different ML algorithms can also be used to check which algorithm will give the best result.

As python and R are both programming languages that can adapt to new changes easily, it is easy to maintain this type of system.

### SCALABILITY:

The system can work normally under situations such as low bandwidth and huge datasets. The

R studio as well as Excel can take care of these data and can perform the algorithms with ease.

### PORTABILITY:

Portability is a feature which tells us about the ease at which we can reuse an existing piece of code when we move from one location or environment to some other.

This system uses python and R programming languages and they can be executed under different operation conditions provided it meet its minimum configurations. Only system files and dependant assemblies would have to be configured in such case.

**What's next for review 3**

**Future scope:**

In future disease related to pneumonia, covid from x ray related diseases can be added to application by using image datasets and CNN model to train those dataset and models are integrated in to web application.

**Concept:**

We are proposing such a system that will flaunt a simple, cost effective , elegant User Interface and also be time efficient . Our proposed system bridges the gap between doctors and patients which will help both classes of users to achieve their goal. This system is used to predict below mentioned diseases

Diabetes

- Breast Cancer

- Heart Disease

- Kidney Disease

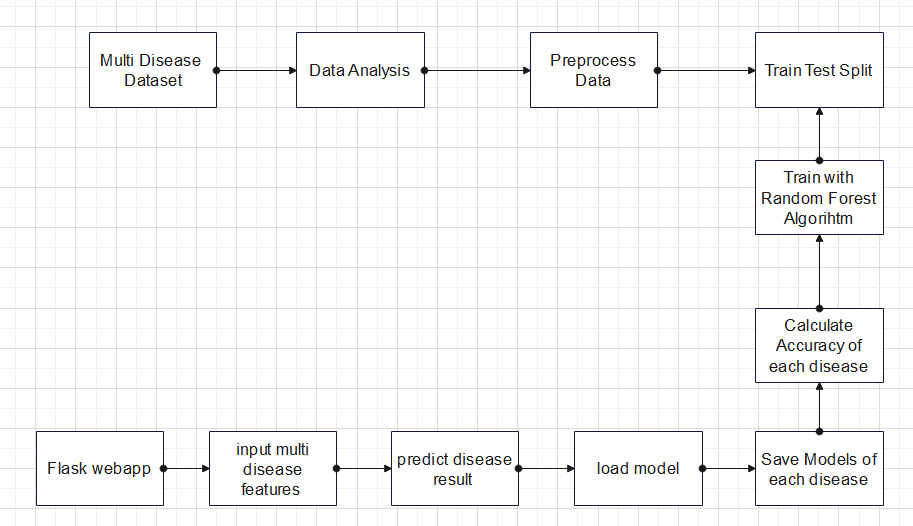
- Liver Disease

- Malaria

- Pneumonia

. In this proposed system we are going to take down six disease dataset from the Kaggle website and evaluate them by applying algorithms such as Decision Tree, Random Forest, Naïve bayes and KNN which will help in getting accurate prediction .Our system will explore and merge more datasets which includes large diversity of population to get more effective results and thus our system will improve and enhances the accuracy of the results. Along with the increased accuracy rate, we will proliferate the reliability of our system for this job and can gain the trust of patient in this system. Apart from all these, our system will comprise of a Database for storing the data entered by the users and the name of the disease the patient is suffering from which can be used as a reference in future for further treatment. Hence this system will contribute in easier health management with better satisfaction to the users.

**Block diagrams and flow chart:**

****

**Results**

**Current status:**