1.Introduction

1.1 Objective

The main objective of this project is to classify various types of Diseases using the Machine Learning Algorithms. We plan to predict different types of diseases like Breast Cancer, Diabetes, Kidney Diseases, Heart Disease and Malaria, with the help of the data sets which we will be getting from an online website called “Kaggle.” We will try to use different types of Algorithms under the Supervised Learning Category and in the end, we will compare the Algorithms with their achieved Accuracies.

1.2 Motivation

Our main Motivation is to predict the disease at most early stage, so that the disease diagnosed person can able to get the treatment at early stage. So, we tried to predict as much as diseases as we can those are Breast Cancer, Kidney disease, Diabetes, Heart Disease

Breast cancer accounts for 14% of cancers in Indian women. It is reported that with every four minutes, an Indian woman is diagnosed with breast cancer. Breast cancer is on the rise, both in rural and urban India. A 2018 report of Breast Cancer statistics recorded 1,62,468 new registered cases and 87,090 reported deaths.

Cancer survival becomes more difficult in higher stages of its growth, and more than 50% of Indian women suffer from stage 3 and 4 of breast cancer. Post cancer survival for women with breast cancer was reported 60% for Indian women, as compared to 80% in the U.S.

Kidney Disease, per year over 100,000 patients are diagnosed with End Stage Kidney Disease (ESKD) in India. The common kidney disease is chronic kidney disease and it affects about 10% of the world's population. Due to the lack of accurate national data collection, the incidence of CKD in India is not clear but studies estimate that the number of new patients diagnosed with End Stage Kidney Disease (ESKD) who are started on dialysis or transplantation is over 100,000 per year.

India has an estimated 77 million people (1 in 11 Indians) formally diagnosed with diabetes, which makes it the second most affected in the world, after China. Furthermore, 700,000 Indians died of diabetes, hyper glycemia, kidney disease or other complications of diabetes in 2020.

An estimated 17.9 million people died from CVDs in 2016, representing 31% of all global deaths. Of these deaths, 85% were due to heart attack and stroke. n 2016 India reported 63% of total deaths due to NCDs, of which 27% were attributed to CVDs. CVDs also account for 45% of deaths in the 40–69-year age group.

Individuals at risk of CVD may demonstrate raised blood pressure, glucose, and lipids as well as overweight and obesity.

And finally coming to Malaria, with approximately 4.2 million estimated malaria cases and 7341 estimated malaria- deaths, India accounted for a total of 83% estimated malaria cases and 82% estimated malaria-deaths in the WHO South-East Asia Region.

Detection of these diseases at early stages of infection can, in many cases, drastically increase the chances of survival and can prevent a large number of deaths. Some of the common examinations done to determine the presence of these diseases. The presence of trained professionals who can read these scans and determine the infection are necessary in every locality. But in most cases, some of the doctors cover vast areas and their presence at all times is not possible.

1.3 Background

Nowadays the computer vision technology is booming, and various identification techniques are developed. Also, advances in the Machine learning led to delivering accurate results and giving rise to new technologies. we are going to take down disease dataset from the Kaggle website and evaluate them by applying algorithms such as Decision Tree, Random Forest, Naïve bayes, KNN and CNN which will help in getting accurate prediction. One of the best techniques currently used in medical image analysis are CNNs which have a remarkable efficiency in classifying the images. Some of the Contemporary CNN models are Pre- Trained, Functional, Sequential. John Ross Quinlan proposed a new concept: trees with multiple answers. Important note: CART and all other decision tree classification algorithms only have two answers for each question called binary trees. k-nearest neighbour algorithm (k-NN) is a non-parametric supervised learning method first developed by Evelyn Fix and Joseph Hodges in 1951, and later expanded by Thomas Cover. Naive Bayes is one of the simplest Machine Learning algorithms that has always been a favourite for classifying data. Naive Bayes is based on Bayes Theorem, which was proposed by Reverend Thomas Bayes back in the 1760's. Random Forest approach, a machine learning technique, was first proposed by Breiman (2001) by combining classification and regression tree.

2. Project Description and Goals

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.

In this proposed system we are going to take down six disease datasets 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.

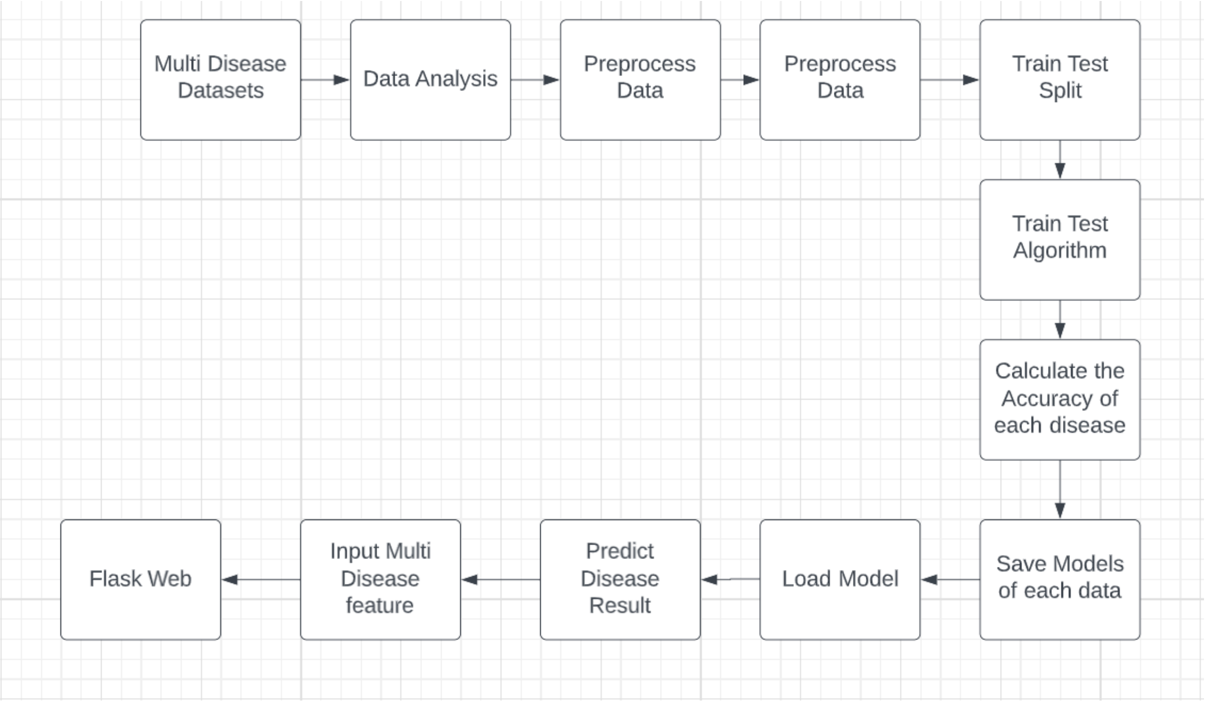


FIG.1

3. Technical Specifications

3.1 Software Used

3.1.1 Anaconda

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

3.1.2 Keras

Keras is an API designed for human beings, now no longer machines. Keras follows best practices for lowering cognitive load: it gives consistent & easy APIs, it minimizes the quantity of consumer movements required for not un-usual place use cases, and it offers clean and actionable comments upon consumer error. This makes Keras smooth to research and smooth to use. As a Keras consumer, you are greater productive, permitting you to strive greater thoughts than your competition, faster which in flip facilitates you win system getting to know competitions. This ease of use does now no longer come on the price of decreased flexibility: because Keras integrates deeply with low-level TensorFlow capability, it enables you to increase extraordinarily hackable workflows in which any piece of capability may be customized.

3.1.3 Jupyter

Jupyter is, in a nutshell: it is a device for collaborating. It is constructed for writing and sharing code and text, in the context of an internet web page. The code runs on a server, and the outcomes are become HTML and included in the web page you are writing. That server may be anywhere: to your laptop, in the back of your firewall, or on the general public internet. Your web page carries your thoughts, your code, and the outcomes of running the code.

Code is in no way simply code. It is a part of a concept process, an argument, even a test. This is especially authentic for facts evaluation, however it is authentic for nearly any application. Jupyter helps you to construct a “lab notebook” that indicates your work: the code, the facts, the outcomes, at the side of your rationalization and reasoning. As IBM places it, Jupyter helps you to construct a “computational narrative that distills facts into insights.” Data means nothing in case you cannot flip it into insight, in case you cannot discover it, proportion it, and speak it. Data evaluation manner little in case you cannot discover and test with a person else’s outcomes. Jupyter is a device for exploring, sharing, and discussing.

3.1.4 Flask

Flask is a micro internet framework written in Python. It is classed as a microframework as it does now no longer requires precise equipment or libraries. It has no database abstraction layer, shape validation, or some other additives wherein pre-present third-birthday birthday celebration libraries offer not un usual place functions. However, Flask helps extensions which could upload utility capabilities as though they had been carried out in Flask itself. Extensions exist for object-relational mappers, shape validation, add handling, diverse open authentication technology and numerous un usual place framework associated equipment.

3.2 Algorithms and Libraries used

3.2.1 Decision Tree Algorithm

The decision tree Algorithm belongs to the family of supervised machine learning algorithms. It can be used for both a classification problem as well as for regression problem. The goal of this algorithm is to create a model that predicts the value of a target variable, for which the decision tree uses the tree representation to solve the problem in which the leaf node corresponds to a class label and attributes are represented on the internal node of the tree.

- Decision Tree is a Supervised learning technique
- That can be used for both classification and Regression problems
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome
- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.

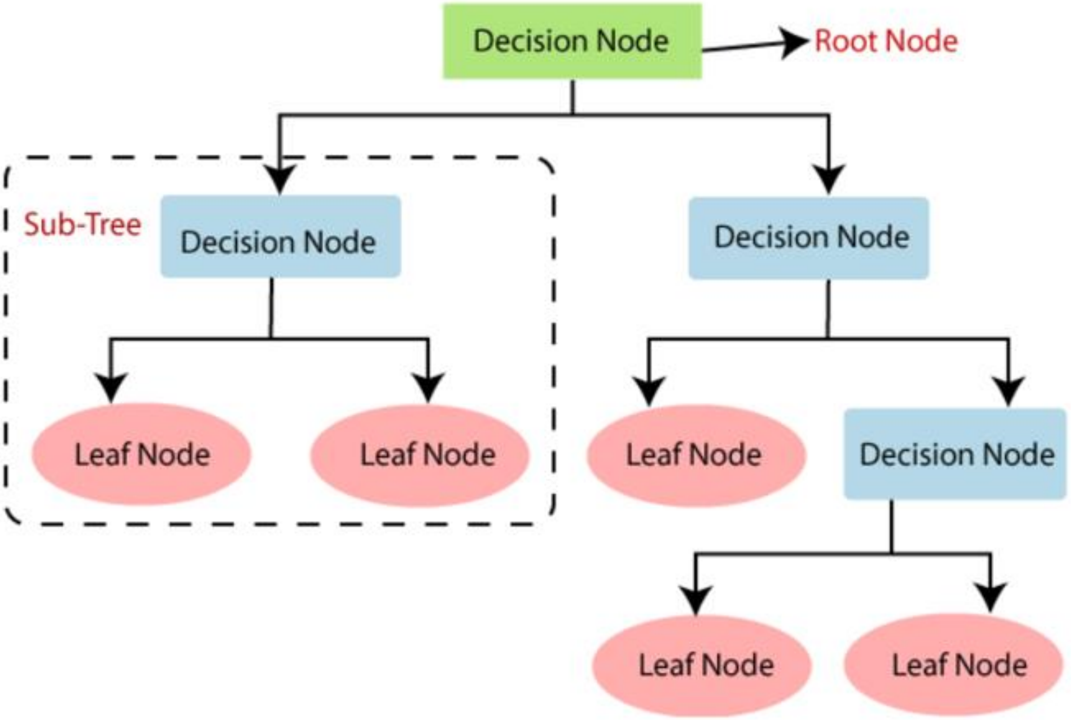


FIG.2

3.2.2 Random Forest Algorithm

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

- 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.
- 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.
- It is capable of handling large datasets with high dimensionality and it enhances the accuracy of the model and prevents the overfitting issue.
- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

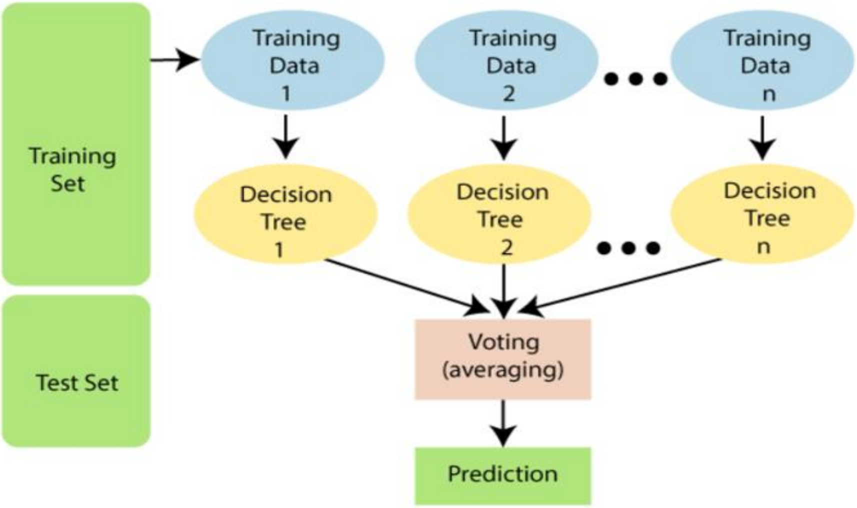


FIG.3

3.2.3 Naïve Bayes Algorithm

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

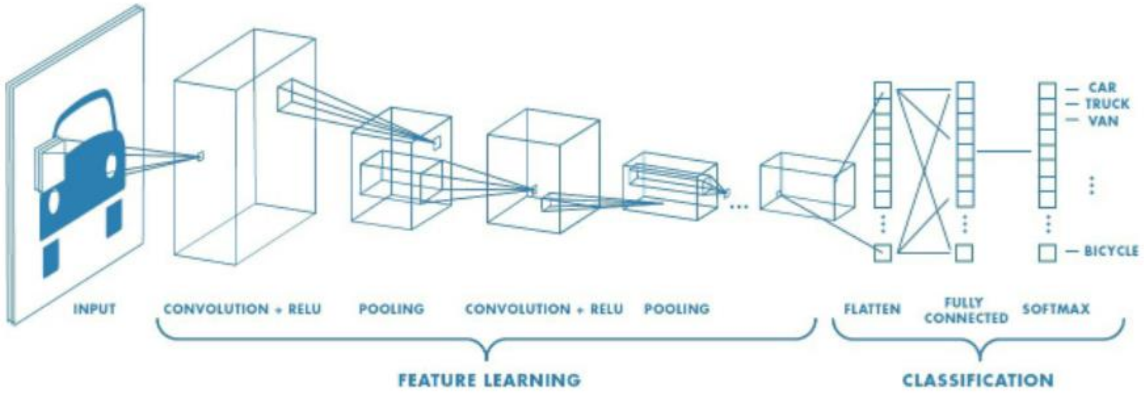
3.2.4 K- Nearest neighbour Algorithm

- 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.

3.2.5 CNN Algorithm

Convolutional Neural Network (CNN) is a Deep Learning set of rules that can absorb an enter photograph, assign significance to numerous aspects/gadgets withinside the photograph and have the ability to distinguish one from the different. The pre-processing required in a CNN is a lot decrease in comparison to different type algorithms. While in primitive strategies filters are hand-engineered, with sufficient training, CNN have the capacity to examine those filters/characteristics.



The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

4. Design Approach and Details

4.1 Collection of datasets.

Heart, kidney, diabetic, liver and malaria 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.

4.2. Understanding features of dataset.

To understand the features which are given in the dataset and able to study that.

4.3 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.

4.4 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.

4.5 Apply ML algorithms to dataset to predict which type of disease

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.

4.6 Accuracy results

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

4.7 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.

5. Scheduled, Tasks and Milestones

- February 1st review - Literature Survey, basic setup.
- March 2nd review - Implementation of the classification model.
- March (After 2nd review) – Improving accuracy of the classification model.
- April – Draft of paper and Poster.
- May – Final touches and completion of project. Submission of report and poster.

6. Project Demonstration

- The dataset is collected from the website called “Kaggle” and the training model is designed.
- The dataset will be randomly divided in to Training set and Testing set.
- The model is trained with the dataset and classifies the different sample test cases which have not been seen by the model yet.
- The accuracy of the model is obtained by the respective algorithms used.
- The models are taken into the web application in order to predict the outcome for given input.

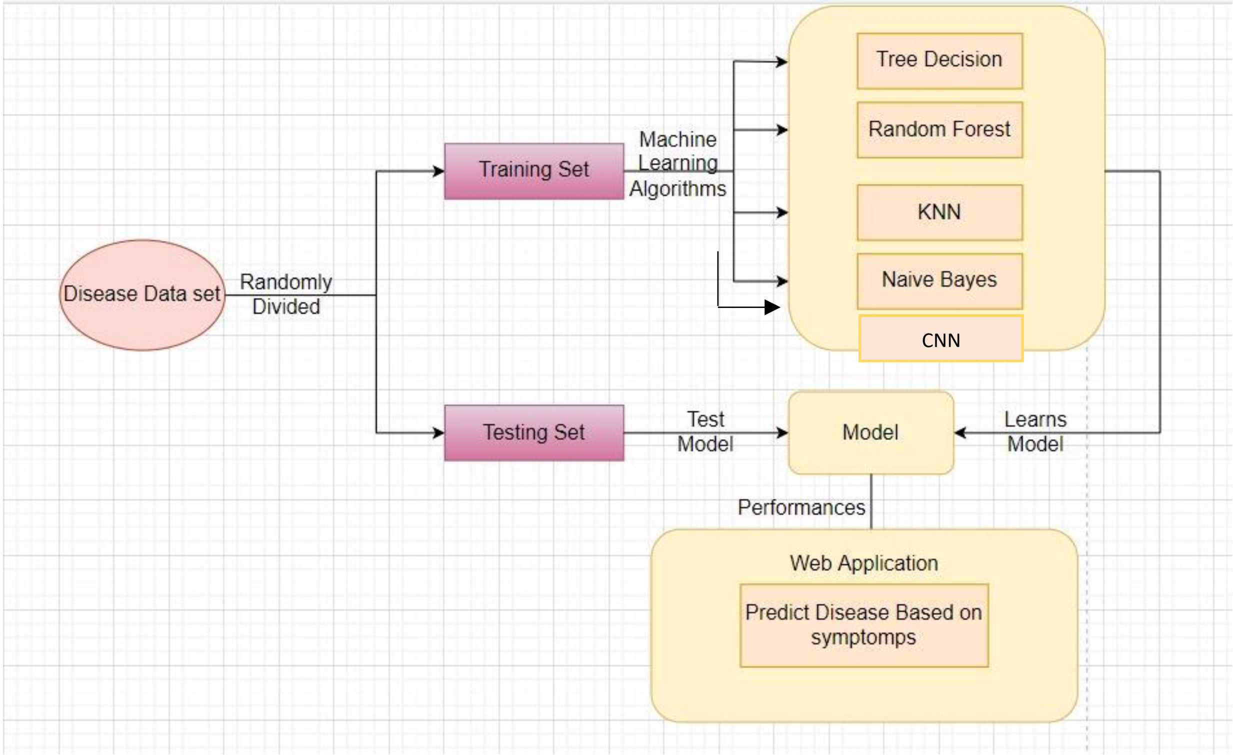


FIG.4

We will be taking the dataset from the Kaggle. Coming to the Breast cancer Dataset we have taken the data where the total patients are 569 and, in those breast cancer Diagnosed patients are 212 in number and the remaining i.e., 357 patients are not Diagnosed with Breast Cancer. In this taken dataset, Breast Cancer Diagnosed patients are indicated with “m” which Is known for “Malign” and the patients that are not Diagnosed with cancer are indicated with “b” which is known as “Benign”.

For Breast Cancer. (Diagnosed=Malign=+, Normal=Benign=-)
Total =569; Diagnosed=212; Normal =357

id	dia	R_m	T_m	P_m	A_m	S_m	C_m	Co_m	Co_p_m	Sy_m	radius_se	F_m
842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	1.095	0.07871
842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.5435	0.05667
84300903	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.7456	0.05999
84348301	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.4956	0.09744
84358402	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.7572	0.05883
843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.3345	0.07613
844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.4467	0.05742
84458202	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.5835	0.07451
844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.3063	0.07389
84501001	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.2976	0.08243
845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.3795	0.05697
84610002	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.5058	0.06082
846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.9555	0.078
846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.4033	0.05338
84667401	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.2121	0.07682
84799002	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.37	0.07077
848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.4727	0.05922
84862001	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.5692	0.07356
849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.7582	0.05395
8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.2699	0.05766
8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.1852	0.06811
8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.2773	0.06905
8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.4388	0.07032
851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.6917	0.05278
852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.8068	0.0633
852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	1.046	0.07413

The Diabetes Dataset we have taken from the Kaggle contains the data where the total patients are 768 and, in those Diabetes, diagnosed patients are 268 in number and the remaining i.e., 500 patients are not Diagnosed with Diabetes.

In this taken dataset, Diabetes Diagnosed patients are indicated with “1” and the patients that are not Diagnosed with cancer are indicated with “0”.

For Diabetes. (Diagnosed=1=+, Normal=0=-)
Total=768; Diagnosed = 268; Normal =500.

Pregnancies	Glucose	BP	ST	Insulin	BMI	Diabe_ped-fun	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1
11	143	94	33	146	36.6	0.254	51	1
10	125	70	26	115	31.1	0.205	41	1

Coming to the Kidney Disease Dataset we have taken the data where the total patients are 400 and, in those Kidney, Disease Diagnosed patients are 250 in number and the remaining i.e., 150 patients are not Diagnosed with Kidney Disease.

In this taken dataset, Kidney Disease Diagnosed patients are indicated with “Ckd” which is a representation for “chronic kidney disease” and the patients that are not Diagnosed with Kidney Disease are indicated with “Notckd” which is a representation for “No Chronic Disease.”

For Kidney Disease. (Diagnosed= Ckd=+, Normal= Notckd=-)
Total =400; Diagnosed =250; Normal=150

id	age	bp	sg	al	bu	sc	hemo	classification
0	48	80	1.02	1	36	1.2	15.4	ckd
1	7	50	1.02	4	18	0.8	11.3	ckd
2	62	80	1.01	2	53	1.8	9.6	ckd
3	48	70	1.005	4	56	3.8	11.2	ckd
4	51	80	1.01	2	26	1.4	11.6	ckd
5	60	90	1.015	3	25	1.1	12.2	ckd
6	68	70	1.01	0	54	24	12.4	ckd
7	24		1.015	2	31	1.1	12.4	ckd
8	52	100	1.015	3	60	1.9	10.8	ckd
9	53	90	1.02	2	107	7.2	9.5	ckd
10	50	60	1.01	2	55	4	9.4	ckd
11	63	70	1.01	3	60	2.7	10.8	ckd
12	68	70	1.015	3	72	2.1	9.7	ckd
13	68	70	1.005	2	86	4.6	9.8	ckd
14	68	80	1.01	3	90	4.1	5.6	ckd
15	40	80	1.015	3	162	9.6	7.6	ckd
16	47	70	1.015	2	46	2.2	12.6	ckd
17	47	80	1.05	1	87	5.2	12.1	ckd
18	60	100	1.025	0	27	1.3	12.7	ckd
19	62	60	1.015	1	31	1.6	10.3	ckd
20	61	80	1.015	2	148	3.9	7.7	ckd
21	60	90	1.02	3	180	76	10.9	ckd
22	48	80	1.025	4	163	7.7	9.8	ckd
23	21	70	1.01	0	57	3.5	10.5	ckd
24	42	100	1.015	4	50	1.4	11.1	ckd
25	61	60	1.025	0	75	1.9	9.9	ckd

And after collecting all the datasets the data will be divided in to the Training and Test sets and the Training set will be get trained to the particular Algorithm that we choose. We will be using different types of machine learning algorithms under the supervised learning category. then we need the will be getting their accuracies and after that we will be comparing the Training Data and the Test data in-order to get the results.

In the Liver Disease Dataset, we have taken the data where the total patients are 583 and, in those Liver, Disease Diagnosed patients are 167 in number and the remaining i.e., 416 patients are not Diagnosed with Liver Disease.

In this taken dataset, Liver Disease Diagnosed patients are indicated with “2” and the patients that are not Diagnosed with Liver Disease are indicated with “1”.

For Liver Disease. (Diagnosed= 2=+, Normal= 1=-)

Total =583; Diagnosed =167; Normal=416

Age	Gender	T_B	D_B	A_P	A_A	A_Am	Total_Protiens	Albumin	A_G	Dataset
65	Female	0.7	0.1	187	16	18	6.8	3.3	0.9	1
62	Male	10.9	5.5	699	64	100	7.5	3.2	0.74	1
62	Male	7.3	4.1	490	60	68	7	3.3	0.89	1
58	Male	1	0.4	182	14	20	6.8	3.4	1	1
72	Male	3.9	2	195	27	59	7.3	2.4	0.4	1
46	Male	1.8	0.7	208	19	14	7.6	4.4	1.3	1
26	Female	0.9	0.2	154	16	12	7	3.5	1	1
29	Female	0.9	0.3	202	14	11	6.7	3.6	1.1	1
17	Male	0.9	0.3	202	22	19	7.4	4.1	1.2	1
55	Male	0.7	0.2	290	53	58	6.8	3.4	1	1
57	Male	0.6	0.1	210	51	59	5.9	2.7	0.8	1
72	Male	2.7	1.3	260	31	56	7.4	3	0.6	1
64	Male	0.9	0.3	310	61	58	7	3.4	0.9	1
74	Female	1.1	0.4	214	22	30	8.1	4.1	1	1
61	Male	0.7	0.2	145	53	41	5.8	2.7	0.87	1
25	Male	0.6	0.1	183	91	53	5.5	2.3	0.7	1
38	Male	1.8	0.8	342	168	441	7.6	4.4	1.3	1
33	Male	1.6	0.5	165	15	23	7.3	3.5	0.92	1
40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
40	Female	0.9	0.3	293	232	245	6.8	3.1	0.8	1
51	Male	2.2	1	610	17	28	7.3	2.6	0.55	1
51	Male	2.9	1.3	482	22	34	7	2.4	0.5	1

In the taken Heart Disease Dataset it contains total patients of 397 and in those Heart, Disease Diagnosed patients are 160 in number and the remaining i.e., 137 patients are not Diagnosed with Heart Disease.

In this taken dataset, Heart Disease Diagnosed patients are indicated with “0” and the patients that are not Diagnosed with Heart Disease are indicated with “1”.

For Heart Disease. (Diagnosed= 0 =+, Normal= 1=-)

Total =397; Diagnosed =160; Normal=137

age	sex	c	t	c	f	r	t	ex	oldpeak	slope	ca	thal	condition
69	1	0	160	234	1	2	131	0	0.1	1	1	0	0
69	0	0	140	239	0	0	151	0	1.8	0	2	0	0
66	0	0	150	226	0	0	114	0	2.6	2	0	0	0
65	1	0	138	282	1	2	174	0	1.4	1	1	0	1
64	1	0	110	211	0	2	144	1	1.8	1	0	0	0
64	1	0	170	227	0	2	155	0	0.6	1	0	2	0
63	1	0	145	233	1	2	150	0	2.3	2	0	1	0
61	1	0	134	234	0	0	145	0	2.6	1	2	0	1
60	0	0	150	240	0	0	171	0	0.9	0	0	0	0
59	1	0	178	270	0	2	145	0	4.2	2	0	2	0
59	1	0	170	288	0	2	159	0	0.2	1	0	2	1
59	1	0	160	273	0	2	125	0	0	0	0	0	1
59	1	0	134	204	0	0	162	0	0.8	0	2	0	1
58	0	0	150	283	1	2	162	0	1	0	0	0	0
56	1	0	120	193	0	2	162	0	1.9	1	0	2	0

At last, coming to the Malaria Disease, in this take dataset it totally contains of 27,559 images combined of both Parasitized and Uninfected and in those Parasitized are “13,779” and the Uninfected are “13,780”.

Now we will be showing a glympse of the dataset of Parasitized and Uninfected.

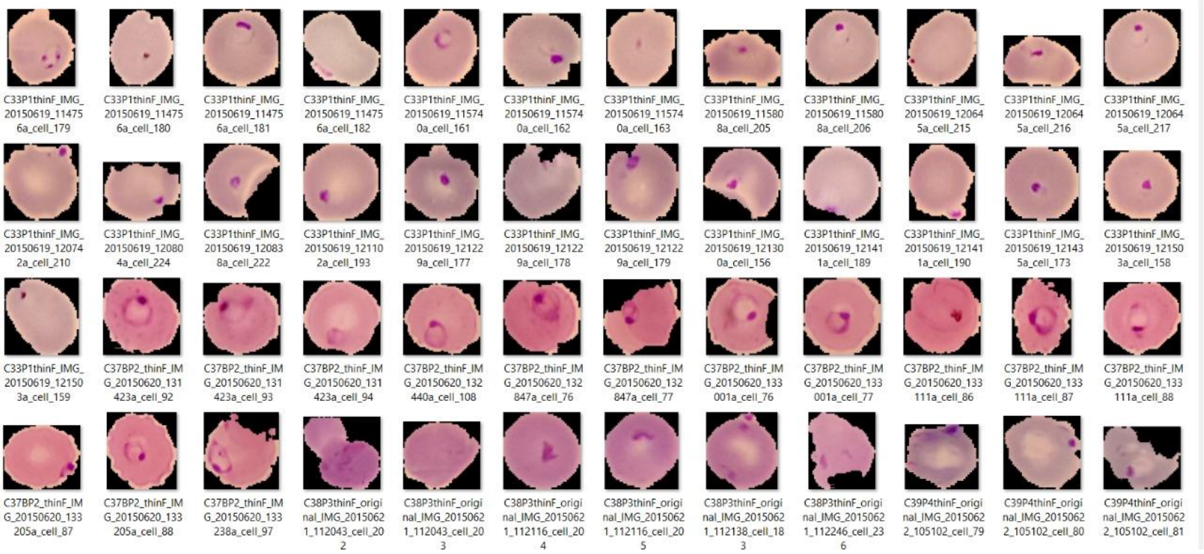

C33P1thinF_IMG_20150619_114756a_cell_179
C33P1thinF_IMG_20150619_114756a_cell_180
C33P1thinF_IMG_20150619_114756a_cell_181
C33P1thinF_IMG_20150619_114756a_cell_182
C33P1thinF_IMG_20150619_115740a_cell_161
C33P1thinF_IMG_20150619_115740a_cell_162
C33P1thinF_IMG_20150619_115740a_cell_163
C33P1thinF_IMG_20150619_115808a_cell_205
C33P1thinF_IMG_20150619_115808a_cell_206
C33P1thinF_IMG_20150619_120645a_cell_215
C33P1thinF_IMG_20150619_120645a_cell_216
C33P1thinF_IMG_20150619_120645a_cell_217
C33P1thinF_IMG_20150619_120742a_cell_210
C33P1thinF_IMG_20150619_120804a_cell_224
C33P1thinF_IMG_20150619_120838a_cell_222
C33P1thinF_IMG_20150619_121102a_cell_193
C33P1thinF_IMG_20150619_121229a_cell_177
C33P1thinF_IMG_20150619_121229a_cell_178
C33P1thinF_IMG_20150619_121229a_cell_179
C33P1thinF_IMG_20150619_121300a_cell_156
C33P1thinF_IMG_20150619_121411a_cell_189
C33P1thinF_IMG_20150619_121411a_cell_190
C33P1thinF_IMG_20150619_121435a_cell_173
C33P1thinF_IMG_20150619_121503a_cell_158
C33P1thinF_IMG_20150619_121503a_cell_159
C37BP2_thinF_IMG_20150620_131423a_cell_92
C37BP2_thinF_IMG_20150620_131423a_cell_93
C37BP2_thinF_IMG_20150620_131423a_cell_94
C37BP2_thinF_IMG_20150620_132440a_cell_108
C37BP2_thinF_IMG_20150620_132847a_cell_76
C37BP2_thinF_IMG_20150620_132847a_cell_77
C37BP2_thinF_IMG_20150620_133001a_cell_76
C37BP2_thinF_IMG_20150620_133001a_cell_77
C37BP2_thinF_IMG_20150620_133111a_cell_86
C37BP2_thinF_IMG_20150620_133111a_cell_87
C37BP2_thinF_IMG_20150620_133111a_cell_88
C37BP2_thinF_IMG_20150620_133205a_cell_87
C37BP2_thinF_IMG_20150620_133205a_cell_88
C37BP2_thinF_IMG_20150620_133238a_cell_97
C38P3thinF_original_IMG_20150621_112043_cell_202
C38P3thinF_original_IMG_20150621_112043_cell_203
C38P3thinF_original_IMG_20150621_112116_cell_204
C38P3thinF_original_IMG_20150621_112116_cell_205
C38P3thinF_original_IMG_20150621_112138_cell_183
C38P3thinF_original_IMG_20150621_112246_cell_236
C39P4thinF_original_IMG_20150622_105102_cell_79
C39P4thinF_original_IMG_20150622_105102_cell_80
C39P4thinF_original_IMG_20150622_105102_cell_81

FIG. 5

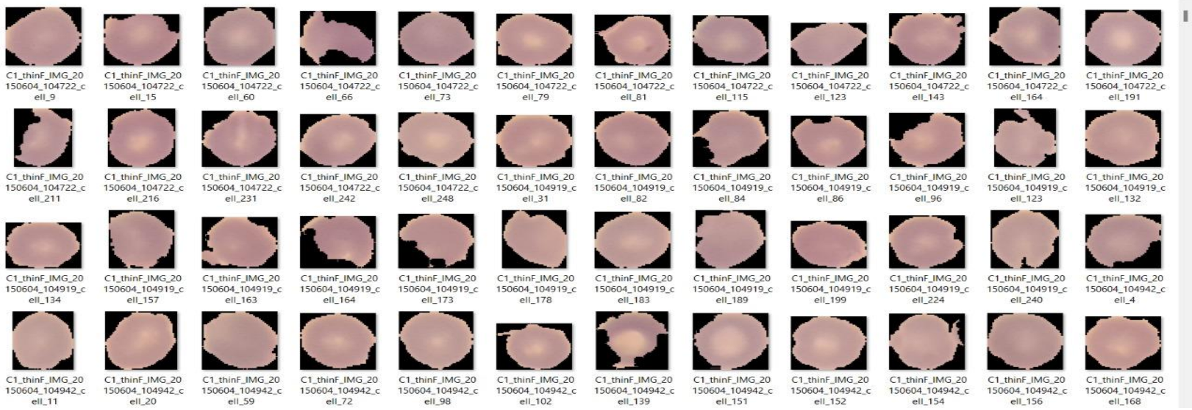

C1_thinF_IMG_20150604_104722_cell_9
C1_thinF_IMG_20150604_104722_cell_15
C1_thinF_IMG_20150604_104722_cell_60
C1_thinF_IMG_20150604_104722_cell_66
C1_thinF_IMG_20150604_104722_cell_73
C1_thinF_IMG_20150604_104722_cell_79
C1_thinF_IMG_20150604_104722_cell_81
C1_thinF_IMG_20150604_104722_cell_115
C1_thinF_IMG_20150604_104722_cell_123
C1_thinF_IMG_20150604_104722_cell_143
C1_thinF_IMG_20150604_104722_cell_164
C1_thinF_IMG_20150604_104722_cell_191
C1_thinF_IMG_20150604_104722_cell_211
C1_thinF_IMG_20150604_104722_cell_216
C1_thinF_IMG_20150604_104722_cell_231
C1_thinF_IMG_20150604_104722_cell_242
C1_thinF_IMG_20150604_104722_cell_248
C1_thinF_IMG_20150604_104919_cell_31
C1_thinF_IMG_20150604_104919_cell_82
C1_thinF_IMG_20150604_104919_cell_84
C1_thinF_IMG_20150604_104919_cell_86
C1_thinF_IMG_20150604_104919_cell_96
C1_thinF_IMG_20150604_104919_cell_123
C1_thinF_IMG_20150604_104919_cell_132
C1_thinF_IMG_20150604_104919_cell_134
C1_thinF_IMG_20150604_104919_cell_157
C1_thinF_IMG_20150604_104919_cell_163
C1_thinF_IMG_20150604_104919_cell_164
C1_thinF_IMG_20150604_104919_cell_173
C1_thinF_IMG_20150604_104919_cell_178
C1_thinF_IMG_20150604_104919_cell_183
C1_thinF_IMG_20150604_104919_cell_189
C1_thinF_IMG_20150604_104919_cell_199
C1_thinF_IMG_20150604_104919_cell_224
C1_thinF_IMG_20150604_104919_cell_240
C1_thinF_IMG_20150604_104942_cell_4
C1_thinF_IMG_20150604_104942_cell_11
C1_thinF_IMG_20150604_104942_cell_20
C1_thinF_IMG_20150604_104942_cell_59
C1_thinF_IMG_20150604_104942_cell_72
C1_thinF_IMG_20150604_104942_cell_98
C1_thinF_IMG_20150604_104942_cell_102
C1_thinF_IMG_20150604_104942_cell_139
C1_thinF_IMG_20150604_104942_cell_151
C1_thinF_IMG_20150604_104942_cell_152
C1_thinF_IMG_20150604_104942_cell_154
C1_thinF_IMG_20150604_104942_cell_156
C1_thinF_IMG_20150604_104942_cell_168

FIG. 6

For Malaria Disease.

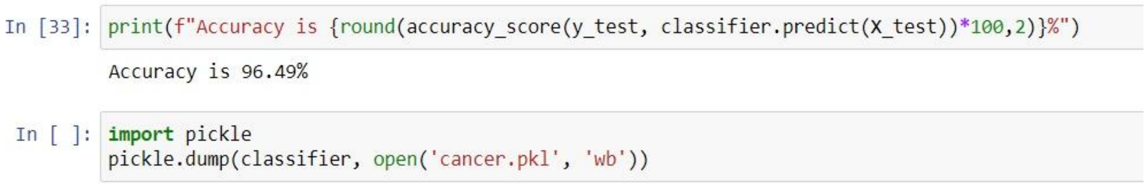
Total =27,558; Parasitized =13,779; Normal= 13,780

7.Result and Discussion

For Breast Cancer.

Here we will be writing the particular code for Breast cancer of their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

Random Forest: -



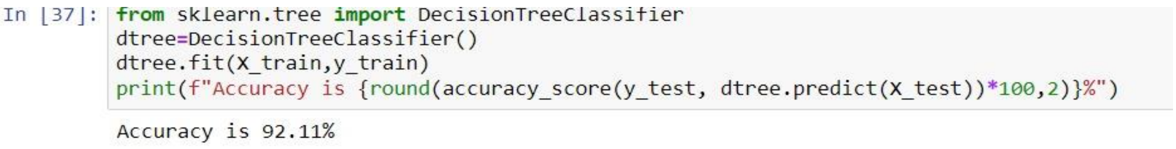
```
In [33]: print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%")

         Accuracy is 96.49%

In [ ]:  import pickle
         pickle.dump(classifier, open('cancer.pkl', 'wb'))
```

FIG. 7

Decision Tree: -

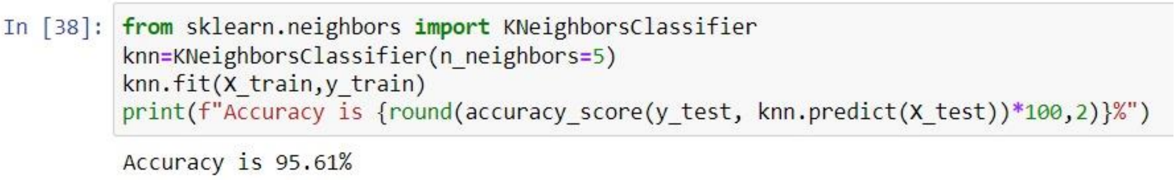


```
In [37]: from sklearn.tree import DecisionTreeClassifier
         dtree=DecisionTreeClassifier()
         dtree.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, dtree.predict(X_test))*100,2)}%")

         Accuracy is 92.11%
```

FIG. 8

KNN: -

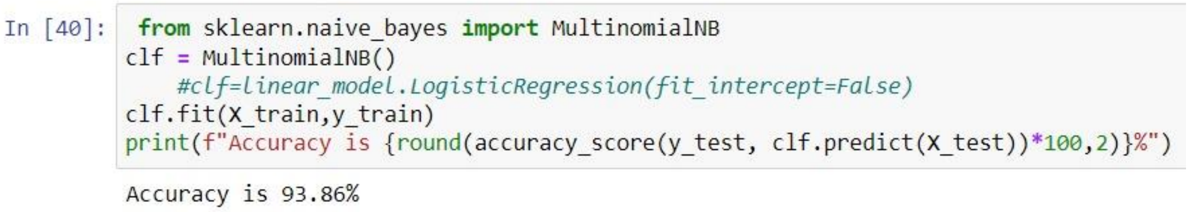


```
In [38]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier(n_neighbors=5)
         knn.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, knn.predict(X_test))*100,2)}%")

         Accuracy is 95.61%
```

FIG. 9

Naïve Bayes: -



```
In [40]:  from sklearn.naive_bayes import MultinomialNB
         clf = MultinomialNB()
             #clf=linear_model.LogisticRegression(fit_intercept=False)
         clf.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, clf.predict(X_test))*100,2)}%")

         Accuracy is 93.86%
```

FIG. 10

And after that we will be aiming ourselves to the Flask to gain the results. We will also develop the code in Jupyter Notebook for the particular appearances that we want in Flask.

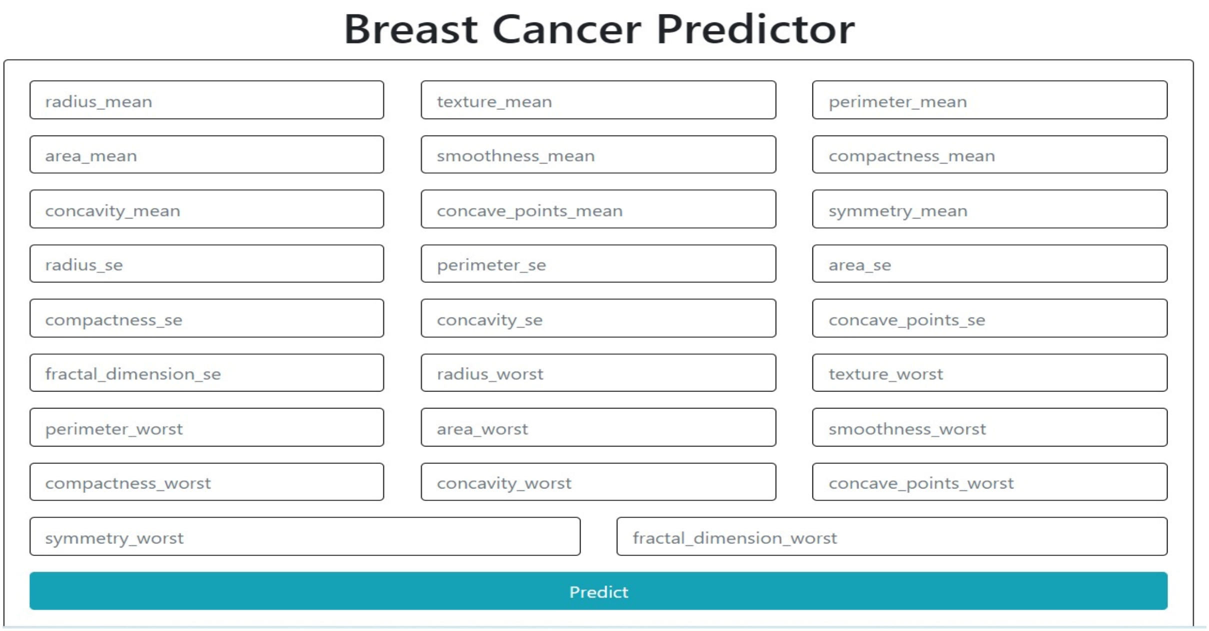


FIG. 11

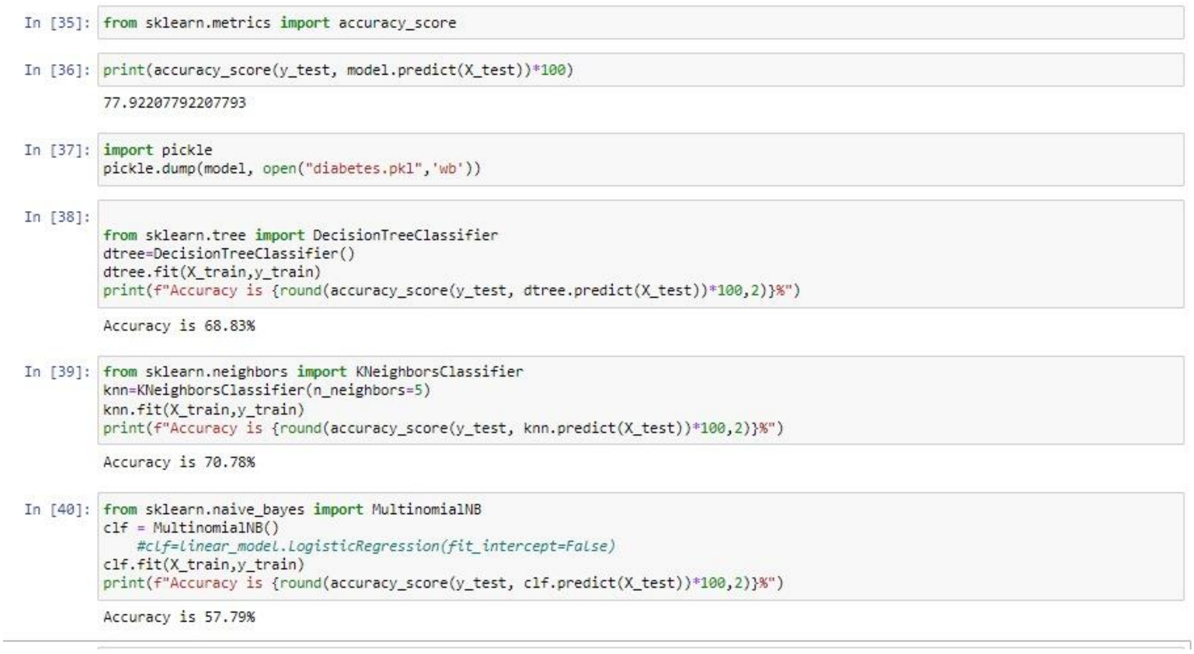
The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with that Breast Cancer or not. These parameters can be taken from the dataset or our Own.

We will be doing the same thing for the all-other disease that we want to predict.

For Diabetes

Same as what we did for the Breast Cancer here also, we will be writing the particular code for Diabetes of their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 36, 38, 39 and 40 respectively in the below given Figure.



```
In [35]: from sklearn.metrics import accuracy_score

In [36]: print(accuracy_score(y_test, model.predict(X_test))*100)

         77.92207792207793

In [37]: import pickle
         pickle.dump(model, open("diabetes.pkl",'wb'))

In [38]:
         from sklearn.tree import DecisionTreeClassifier
         dtree=DecisionTreeClassifier()
         dtree.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, dtree.predict(X_test))*100,2)}%")

         Accuracy is 68.83%

In [39]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier(n_neighbors=5)
         knn.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, knn.predict(X_test))*100,2)}%")

         Accuracy is 70.78%

In [40]: from sklearn.naive_bayes import MultinomialNB
         clf = MultinomialNB()
             #clf=linear_model.LogisticRegression(fit_intercept=False)
         clf.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, clf.predict(X_test))*100,2)}%")

         Accuracy is 57.79%
```

FIG.12

As we can see in the Above Figure that it shows for the Random Forest Algorithm shows the highest accuracy comparing to the all-other Algorithms.

And after that we will be aiming ourselves to the Flask to gain the results

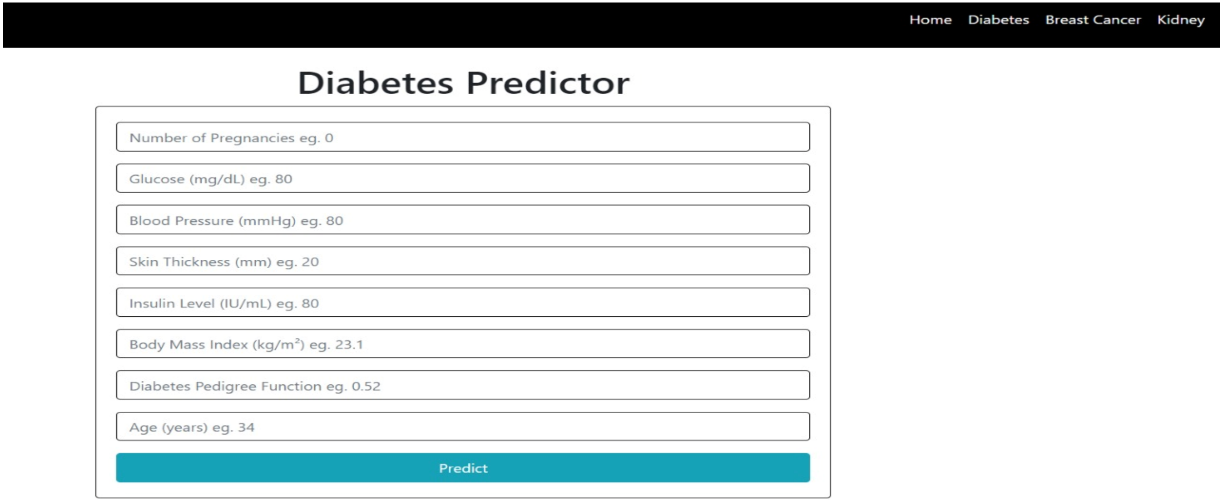

Home
Diabetes
Breast Cancer
Kidney
Diabetes Predictor
Number of Pregnancies eg. 0
Glucose (mg/dL) eg. 80
Blood Pressure (mmHg) eg. 80
Skin Thickness (mm) eg. 20
Insulin Level (IU/mL) eg. 80
Body Mass Index (kg/m²) eg. 23.1
Diabetes Pedigree Function eg. 0.52
Age (years) eg. 34
Predict

FIG.13

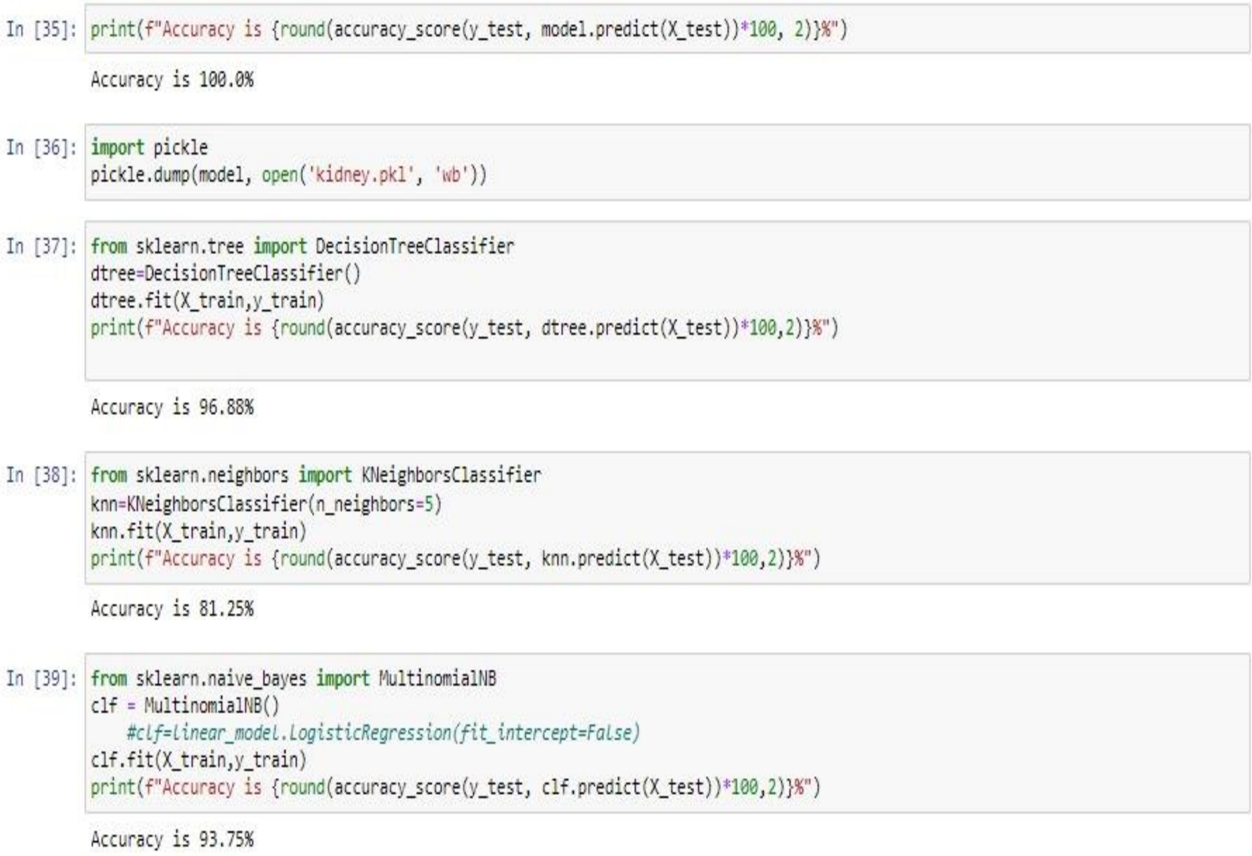
The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with Diabetes or not. These parameters can be taken from the dataset or our Own.

For Kidney Disease

Same as what we did for the Above Diseases here also, we will be writing the particular code for kidney disease for their particular Algorithms in order to calculate their accuracies. These Code will be written in the Jupyter Notebook.

By Comparing the all accuracies, we conclude that the Random Forest Algorithm attains the best accuracy. So, we will be writing only for random forest algorithm.

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 35, 37, 38 and 39 respectively in the below given Figure.



```
In [35]: print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100, 2)}%")

         Accuracy is 100.0%

In [36]: import pickle
         pickle.dump(model, open('kidney.pkl', 'wb'))

In [37]: from sklearn.tree import DecisionTreeClassifier
         dtree=DecisionTreeClassifier()
         dtree.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, dtree.predict(X_test))*100,2)}%")

         Accuracy is 96.88%

In [38]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier(n_neighbors=5)
         knn.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, knn.predict(X_test))*100,2)}%")

         Accuracy is 81.25%

In [39]: from sklearn.naive_bayes import MultinomialNB
         clf = MultinomialNB()
             #clf=linear_model.LogisticRegression(fit_intercept=False)
         clf.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, clf.predict(X_test))*100,2)}%")

         Accuracy is 93.75%
```

FIG. 14

And after that we will be aiming ourselves to the Flask to gain the results.

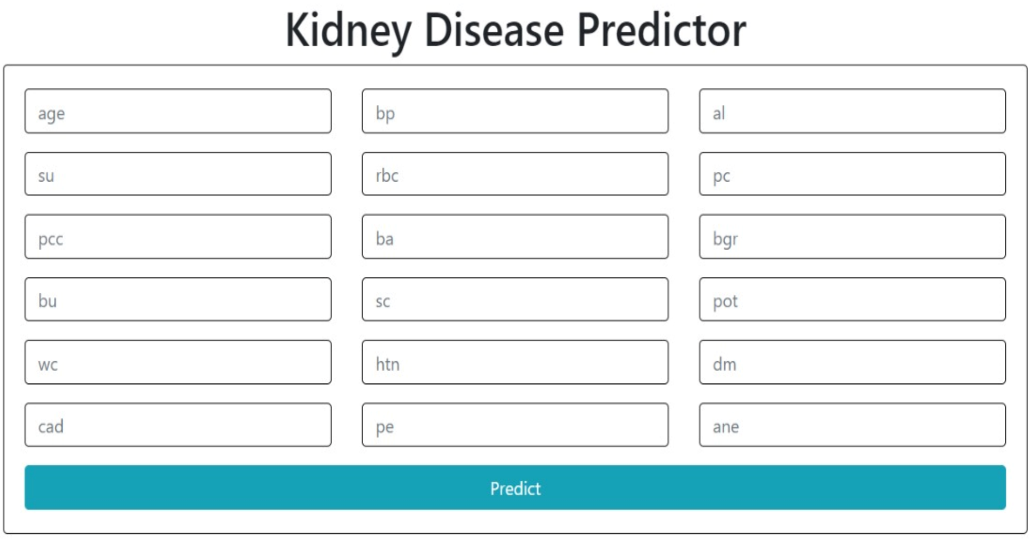
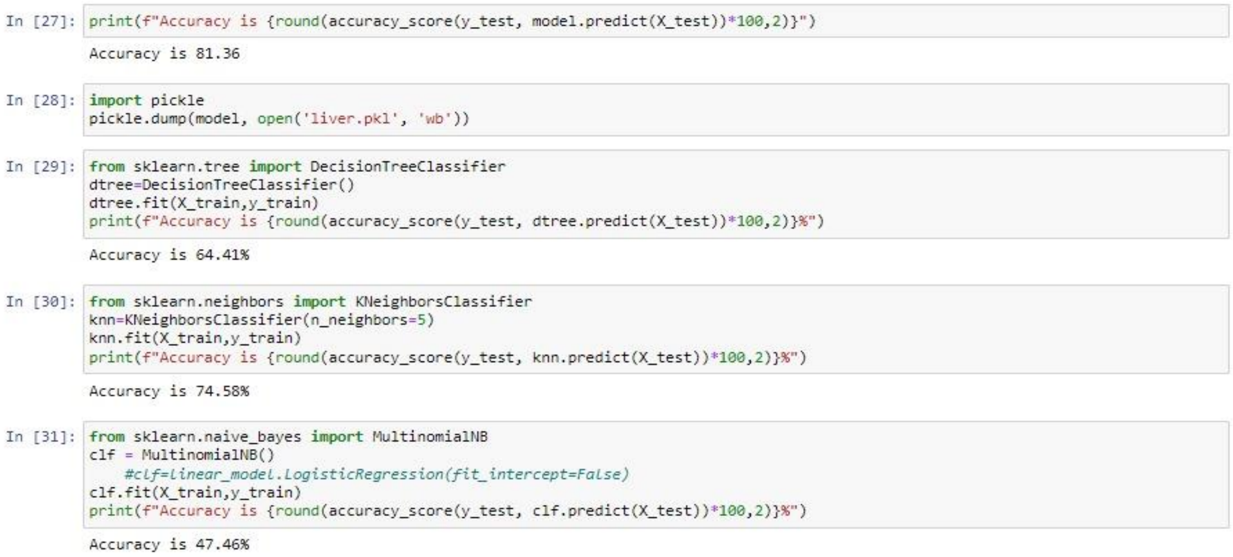


FIG.15

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with kidney disease or not. These parameters can be taken from the dataset or our Own.

For Liver Disease

Coming to the Liver Disease we will be writing the code we will be writing the particular code for their particular Algorithms in Jupyter Notebook in order to calculate their accuracies.



```
In [27]: print(f"Accuracy is {round(accuracy_score(y_test, model.predict(X_test))*100,2)}")

         Accuracy is 81.36

In [28]: import pickle
         pickle.dump(model, open('liver.pkl', 'wb'))

In [29]: from sklearn.tree import DecisionTreeClassifier
         dtree=DecisionTreeClassifier()
         dtree.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, dtree.predict(X_test))*100,2)}%")

         Accuracy is 64.41%

In [30]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier(n_neighbors=5)
         knn.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, knn.predict(X_test))*100,2)}%")

         Accuracy is 74.58%

In [31]: from sklearn.naive_bayes import MultinomialNB
         clf = MultinomialNB()
             #clf=linear_model.LogisticRegression(fit_intercept=False)
         clf.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, clf.predict(X_test))*100,2)}%")

         Accuracy is 47.46%
```

FIG. 16

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 27, 29, 30 and 31 respectively in the below given Figure.

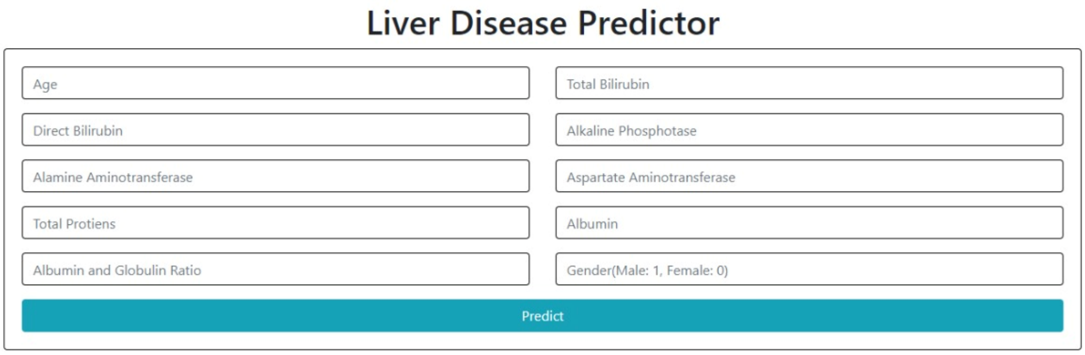
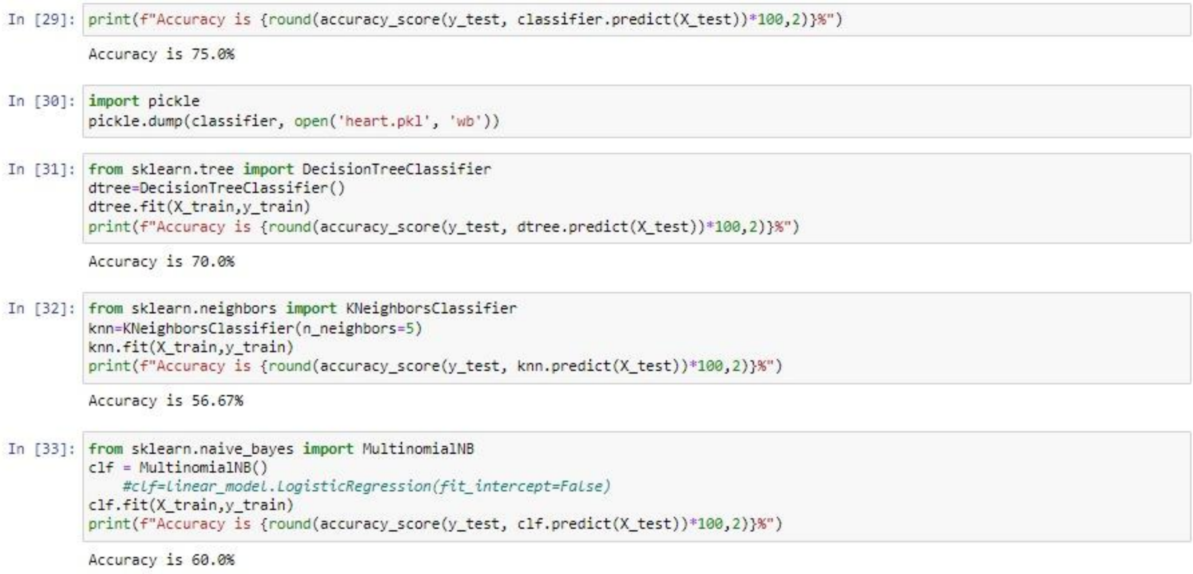


FIG.17

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with Liver Disease or not. These parameters can be taken from the dataset or our Own.

For Heart Disease

For Heart Disease also like same as the others we will be writing the code we will be writing the particular code for their particular Algorithms in Jupyter Notebook in order to calculate their accuracies.



```
In [29]: print(f"Accuracy is {round(accuracy_score(y_test, classifier.predict(X_test))*100,2)}%")

         Accuracy is 75.0%

In [30]: import pickle
         pickle.dump(classifier, open('heart.pkl', 'wb'))

In [31]: from sklearn.tree import DecisionTreeClassifier
         dtree=DecisionTreeClassifier()
         dtree.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, dtree.predict(X_test))*100,2)}%")

         Accuracy is 70.0%

In [32]: from sklearn.neighbors import KNeighborsClassifier
         knn=KNeighborsClassifier(n_neighbors=5)
         knn.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, knn.predict(X_test))*100,2)}%")

         Accuracy is 56.67%

In [33]: from sklearn.naive_bayes import MultinomialNB
         clf = MultinomialNB()
             #clf=linear_model.LogisticRegression(fit_intercept=False)
         clf.fit(X_train,y_train)
         print(f"Accuracy is {round(accuracy_score(y_test, clf.predict(X_test))*100,2)}%")

         Accuracy is 60.0%
```

FIG. 18

The Code for Random Forest, Decision tree, KNN and Naïve bayes are written in the line 29, 31, 32 and 33 respectively in the below given Figure.

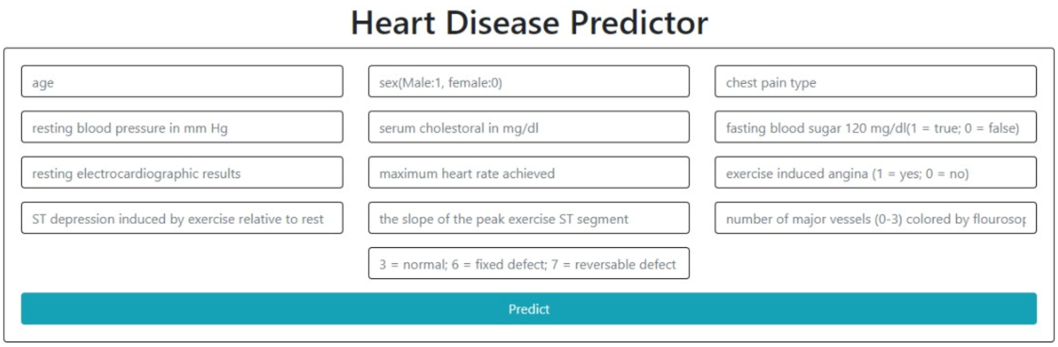

Heart Disease Predictor
age
sex(Male:1, female:0)
chest pain type
resting blood pressure in mm Hg
serum cholestoral in mg/dl
fasting blood sugar 120 mg/dl(1 = true; 0 = false)
resting electrocardiographic results
maximum heart rate achieved
exercise induced angina (1 = yes; 0 = no)
ST depression induced by exercise relative to rest
the slope of the peak exercise ST segment
number of major vessels (0-3) colored by flourosoj
3 = normal; 6 = fixed defect; 7 = reversable defect
Predict

FIG.19

The Above Figure shows how it will be when we open the flask through Jupyter notebook. And then we need to enter all the parameters of a particular patient that are shown in the above Figure to get whether that the patient is diagnosed with heart disease or not. These parameters can be taken from the dataset or our Own.

For Malaria Disease

Coming to the Malaria Disease it is a bit different cause unlike all the other disease we will be using Deep Learning for this. So, the all collected data i.e., the image data will be divided into the training set and testing set and after we train the training data, we will compare that data to the Testing data inorder to know whether the patient is diagnosed with the Malaria Disease or not.

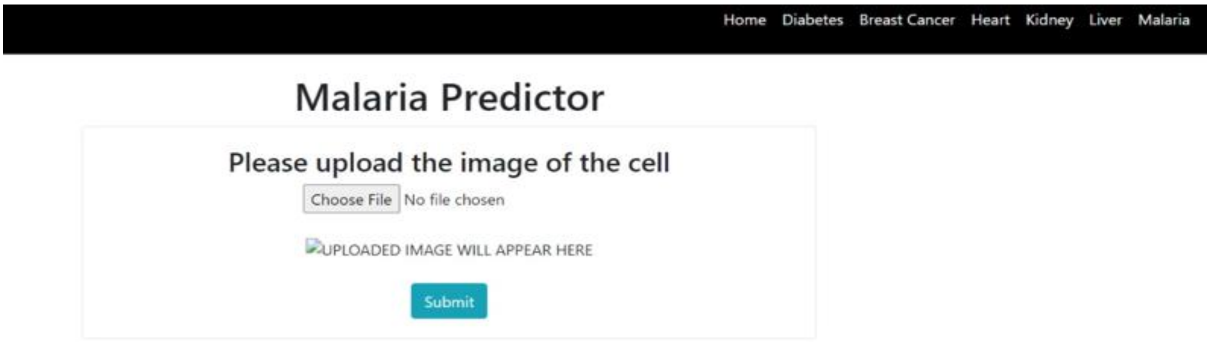
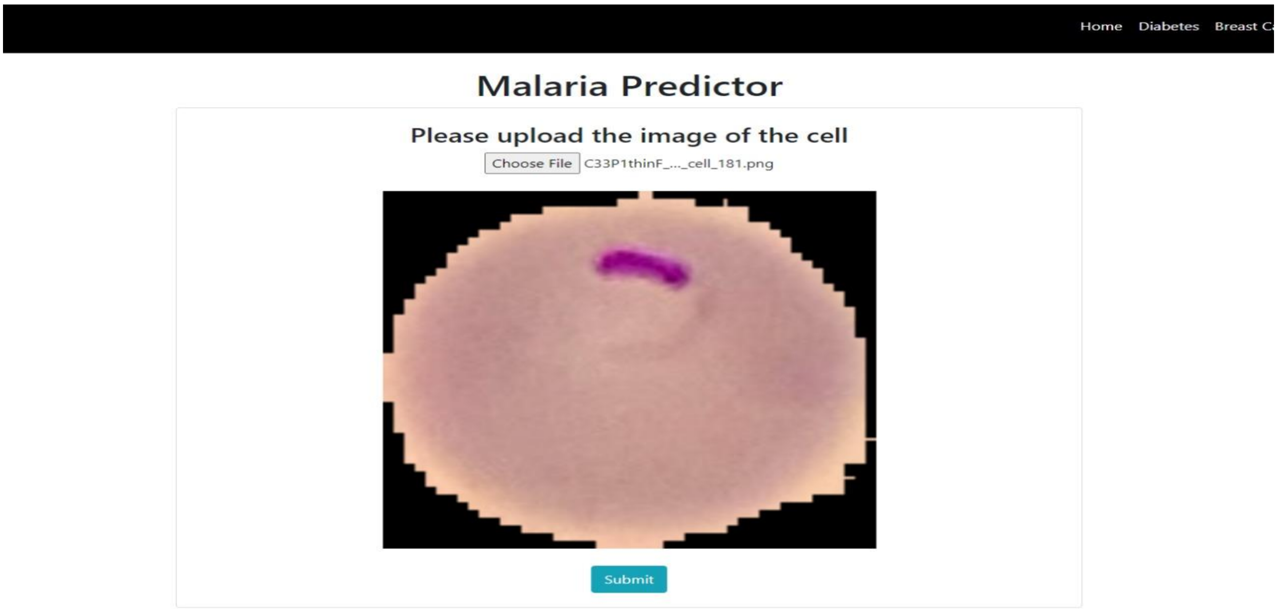


FIG. 20

In the given Above Figure we need to upload the image of the cell of a particular person whom we want to check whether he is diagnosed with malaria disease or not at the Given choose File option.

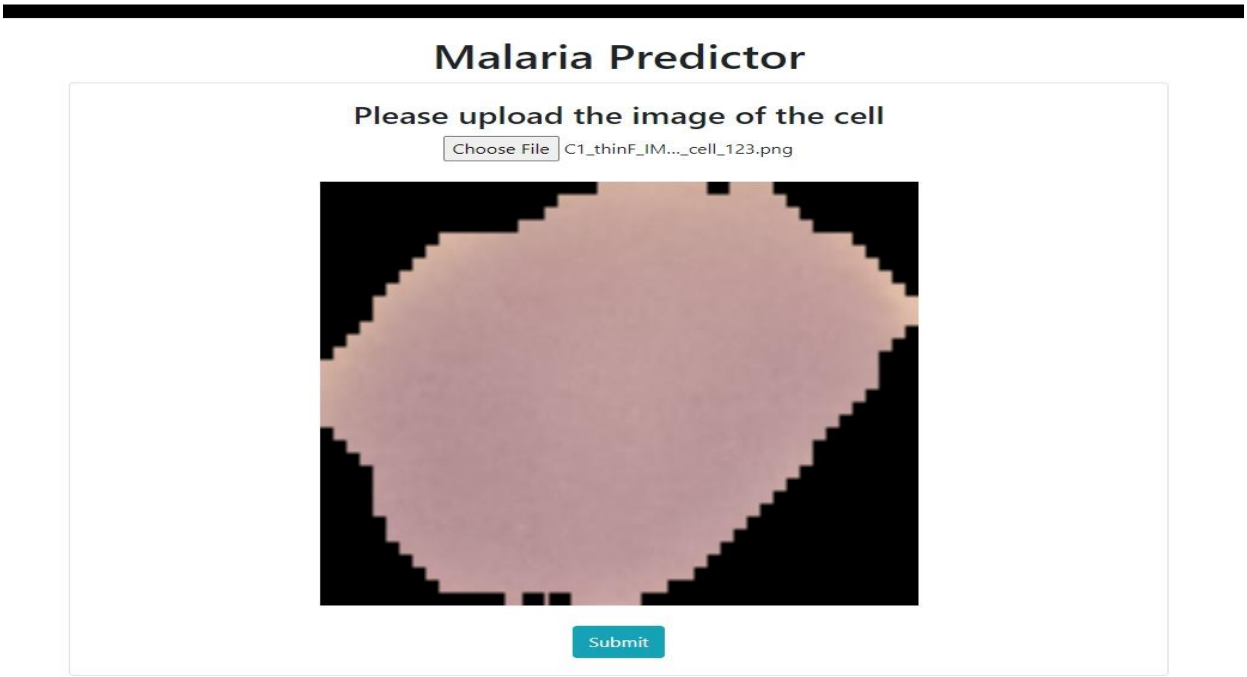


Malaria Predictor

Please upload the image of the cell

C33P1thinF_..._cell_181.png

FIG. 21


Malaria Predictor

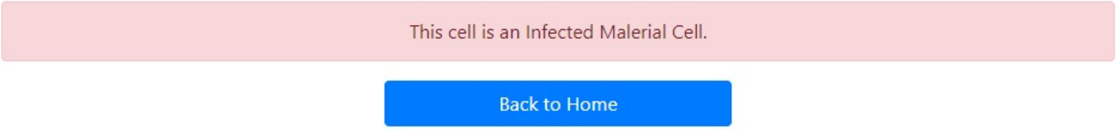
Please upload the image of the cell

C1_thinF_IM..._cell_123.png

FIG. 22

After uploading the image, we will get to see the image that we uploaded in the website as shown in the above Figure.

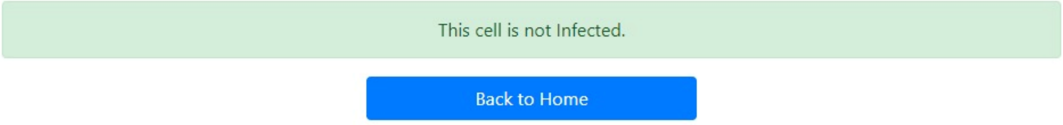
In The Above Figures, Fig .no is Infected with the malaria disease so the result will be represented like as figure given below.



This cell is an Infected Malerial Cell.

FIG. 23

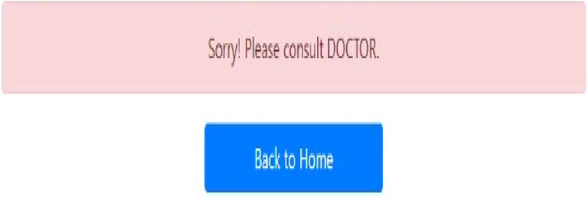
In The Above Figures, Fig .no is not Infected with the malaria disease so the result will be represented like as figure given below.



This cell is not Infected.

FIG. 24

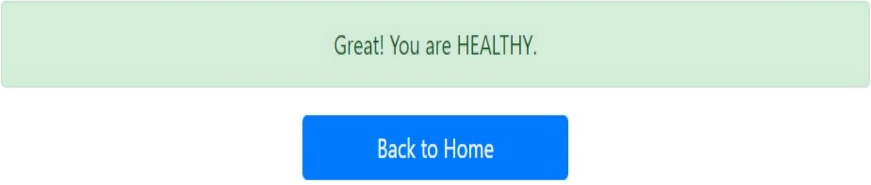
When the Particular Patient is Diagnosed with the Disease except the Malaria Disease then It will represent like the below given Figure in the Flask



Sorry! Please consult DOCTOR.

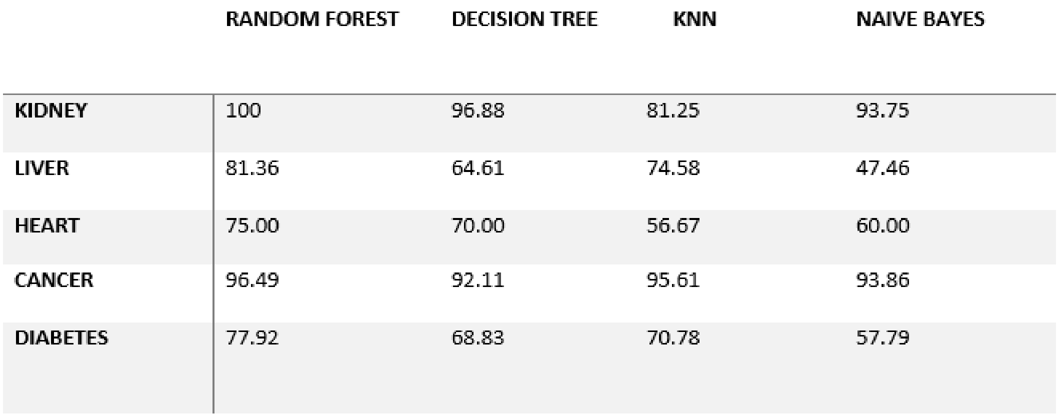
FIG. 25

When the Particular Patient is not Diagnosed with the Disease Excluding the Malaria Disease i.e., the Person is in Healthy Condition then It will represent like the below given Figure in the Flask.



Great! You are HEALTHY.

FIG. 26



	RANDOM FOREST	DECISION TREE	KNN	NAIVE BAYES
KIDNEY	100	96.88	81.25	93.75
LIVER	81.36	64.61	74.58	47.46
HEART	75.00	70.00	56.67	60.00
CANCER	96.49	92.11	95.61	93.86
DIABETES	77.92	68.83	70.78	57.79

FIG. 27