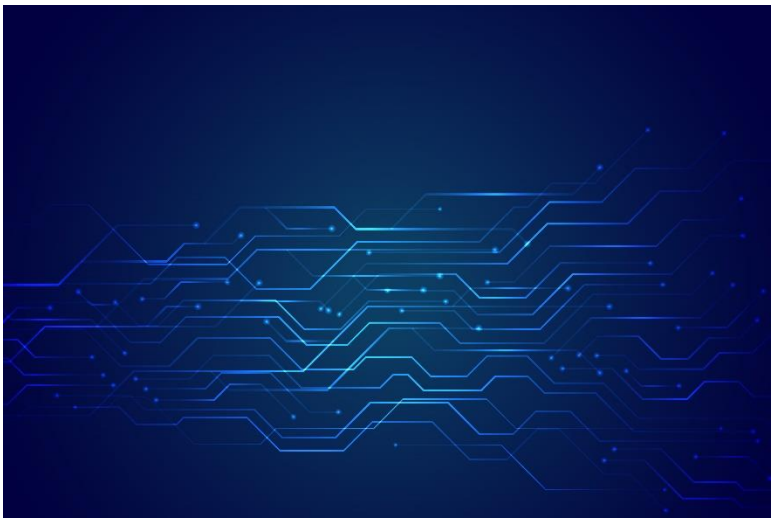


Report

Assignment for Software Engineer

Ayush Kumar
B20MT013
Indian Institute of Technology, Jodhpur

Repository Link

Machine Learning Model

In this project, I've developed two ML models for my website:

1) Breast Cancer Predictor: I've used the dataset from UCI machine learning repository. For knowing which algorithm will be the best for my model, I performed a quick test of the following appropriate non-linear algorithms keeping ten fold cross validation for each testing:

    i) Classification and Regression trees (CART)

    ii) Support Vector Machines (SVM)

    iii) Gaussian Naïve Bayes (NB)

    iv) k-Nearest Neighbors (KNN)

    After evaluating each model, I got following results:

    For CART Model: Mean accuracy is 0.944770

    For SVM Model: Mean accuracy is 0.971386

    For NB Model: Mean accuracy is 0.963223

    For KNN Model: Mean accuracy is 0.969345

    As I got highest mean accuracy for SVM, I used this for my final prediction.

2) Heart Disease Predictor: For this predictor, I took the dataset from a doctor's research project, the data is included in the Dataset folder of the repository. In this model, I used 6 models keeping random state value to be 2022 and max depth to be 15 in required models to check which model will be the best, these models are:

    i) Decision Tree classifier

    ii) Logistic Regression

    iii) SGD Classifier

    iv) Random Forest Classifier

    v) Adaboost Classifier

    vi) XGB Classifier

The results are as follows:

	Model	Train Score	Test Score	Recall	Precision	f1-score
0	DT	0.935	0.929	88.32	97.23	92.56
1	Logistic	0.943	0.944	89.82	98.91	94.15
2	SGD	0.944	0.945	89.03	100.00	94.19
3	RF	0.940	0.935	91.20	95.65	93.37
4	Ada	0.924	0.925	89.72	95.07	92.32
5	XGB	0.947	0.946	90.15	99.03	94.38

Based on the result, I chose XGB classifier to be the final model for my prediction as it was the best model to predict people with heart disease with accuracy of 91% and people without heart disease with accuracy of 99%.

```
...                precision    recall  f1-score   support

           0        0.90962   0.99120   0.94866     73105
           1        0.99034   0.90151   0.94384     73106

    accuracy                             0.94636    146211
   macro avg        0.94998   0.94636   0.94625    146211
weighted avg        0.94998   0.94636   0.94625    146211
```

BACKEND

Backend of a website is the server-side of the site which is used for saving and organizing data and at the same time it ensures that the client-side of the site works properly. I've used Python for backend development using Flask framework. The advantage of using flask is that it has a built-in development server and request dispatching is done through REST.

For API, I imported 'Api' from 'flask_restplus' library and then I developed it for both the models having name 'HeartDisease' and 'BreastDisease' respectively. POST and GET are the HTTP methods being used. POST is used to send data to a server to update or create resource and GET is used to fetch the information from a website.

I've kept the API on the backend so that I can easily fetch it on frontend using the API routes.

FRONTEND

Frontend is the client-side of the website. It is what a user sees when one loads a website. I've used ReactJS for developing the frontend keeping the atomic design in my mind at the same time.

Atomic design refers to designing in a building block manner. The distinct levels of atomic design are:

i) Atoms

ii) Molecules

iii) Organisms

iv) Pages

I've kept atomic design in my mind while making the project:

HOW TO RUN THE PROJECT

1. The project is made on python version= 3.9.0 and react version = 16.8.6.

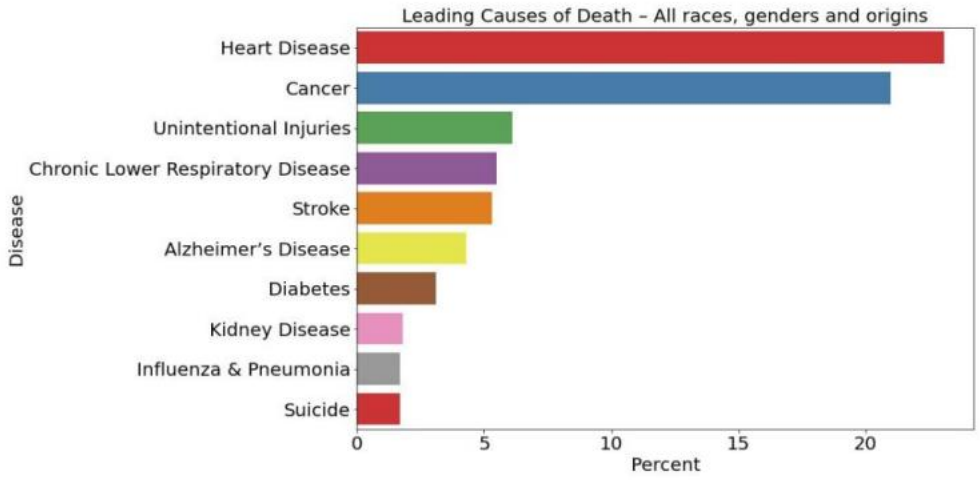
2. Open two terminals:

a) For Frontend: Type cd frontend in the terminal so that you're in the frontend folder. For avoiding dependencies issues, it's preferrable to run a virtual environment first which can be done by typing 'virtualenv <name>', (if you haven't installed virtualenv , it can be done by running 'pip install virtualenv'). Now type '<name>\scripts\activate' to start the virtual environment. Now type 'npm install' so that all dependencies are installed and then run 'npm start' to start the website.

b) For Backend: In the second terminal (Command prompt is preferrable if your windows doesn't allow scripts in code editor terminal.), type cd backend so that you're in the backend folder. Now create a virtual environment but by using 'virtualenv -p python3.9 <name>' (because I used python 3.9 while developing it). Now in the same way as above, type '<name>\scripts\activate' to start the virtual environment. Now run 'pip install -r requirements.txt' to install all the dependencies being used. After all the dependencies are installed, run 'flask run' to start the server.

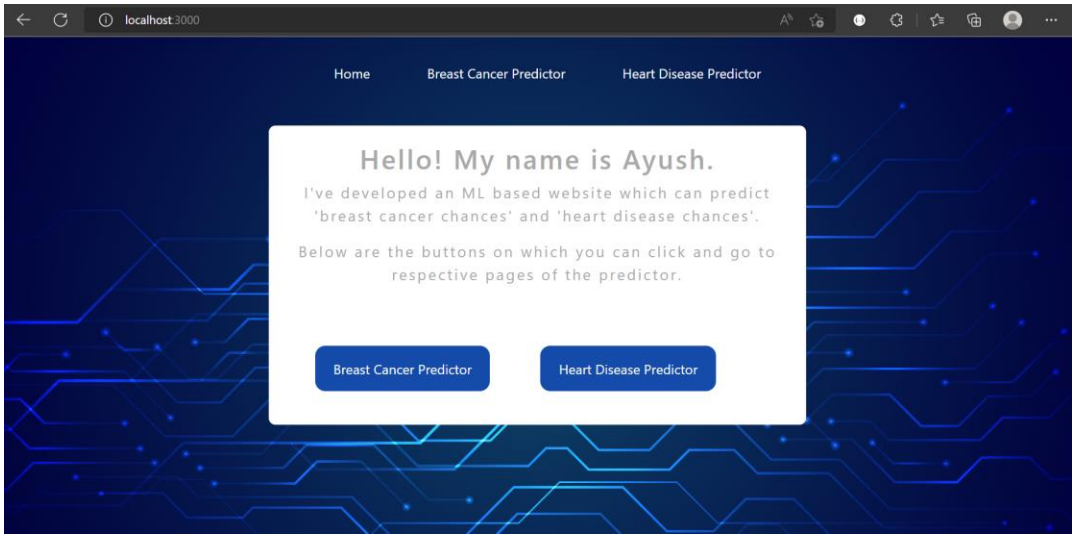
SOCIAL IMPACT

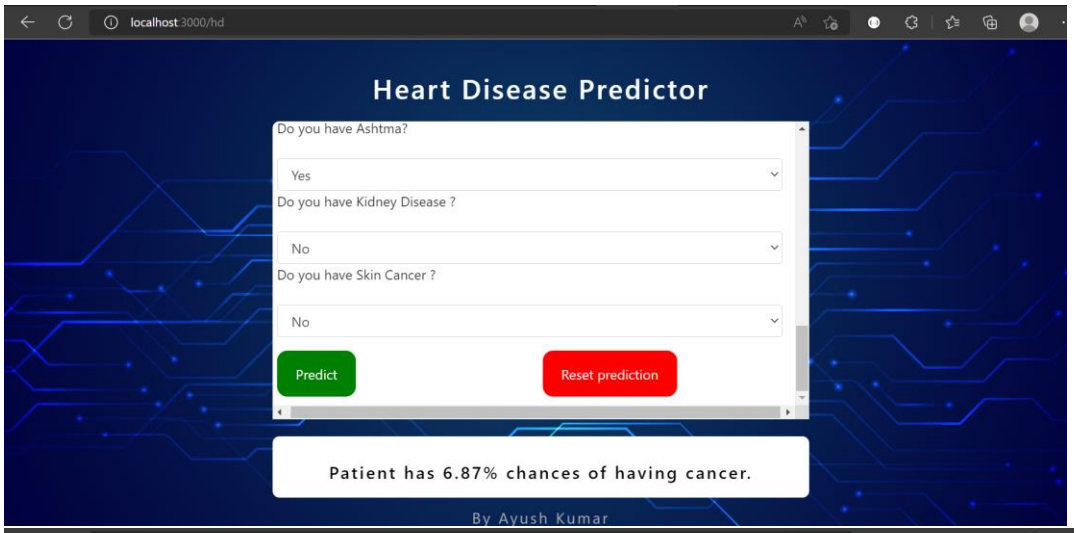
Heart disease and cancer can last a long time in a latent form and these are the leading cause of death:

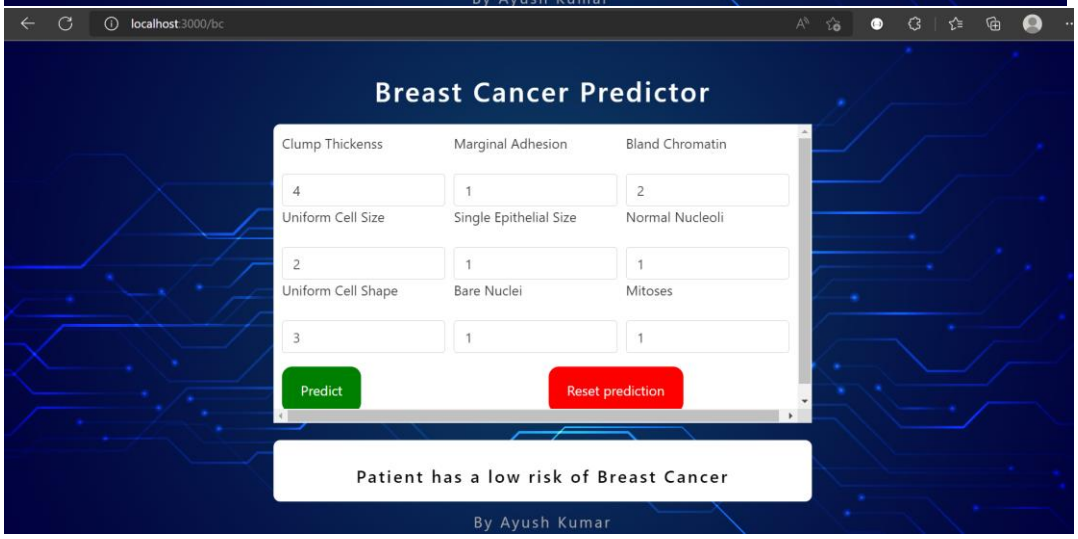


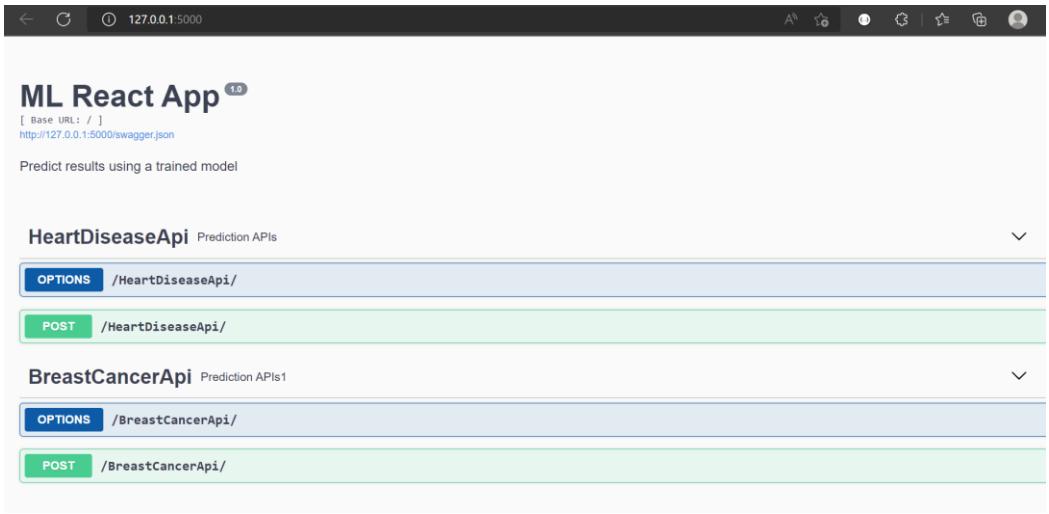
The main objective is to find what personal factors may increase the risk of getting breast cancer or heart disease. At the same time, it can be used to aware people towards heart disease.

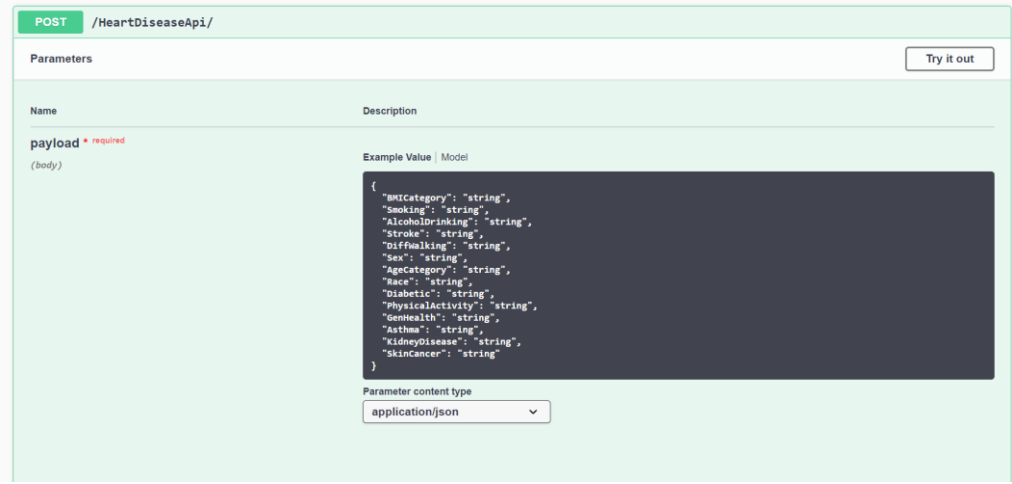
SCREENSHOTS OF THE WEBSITE

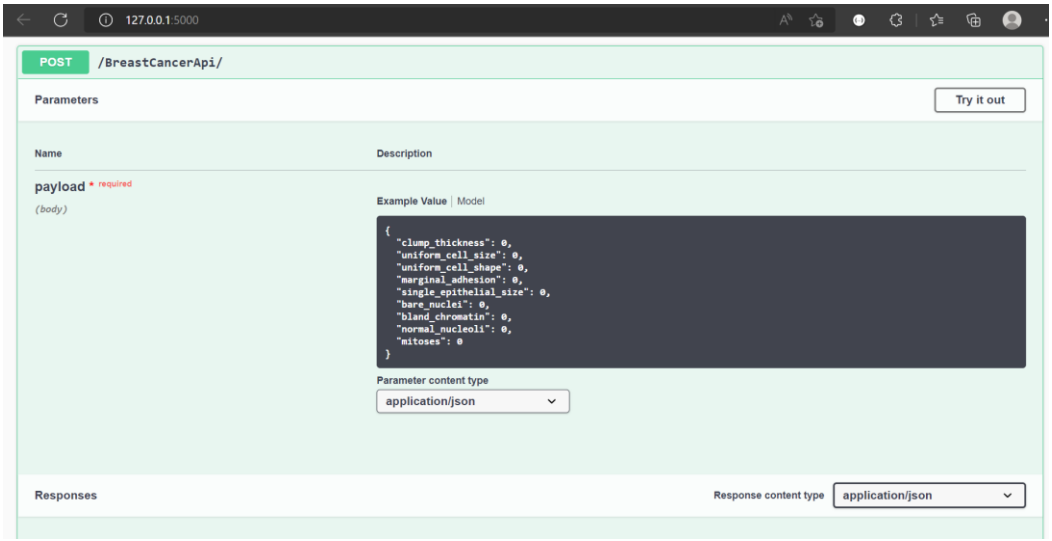







127.0.0.1:5000
ML React App
1.0
[Base URL: /]
http://127.0.0.1:5000/swagger.json
Predict results using a trained model
HeartDiseaseApi Prediction APIs
OPTIONS /HeartDiseaseApi/
POST /HeartDiseaseApi/
BreastCancerApi Prediction APIs1
OPTIONS /BreastCancerApi/
POST /BreastCancerApi/


POST /HeartDiseaseApi/
Parameters
Try it out
Name
Description
payload * required
(body)
Example Value | Model
{
"BMICategory": "string",
"Smoking": "string",
"AlcoholDrinking": "string",
"Stroke": "string",
"DiffWalking": "string",
"Sex": "string",
"AgeCategory": "string",
"Race": "string",
"Diabetic": "string",
"PhysicalActivity": "string",
"GenHealth": "string",
"Asthma": "string",
"KidneyDisease": "string",
"SkinCancer": "string"
}
Parameter content type
application/json


127.0.0.1:5000
POST /BreastCancerApi/
Parameters
Try it out
Name
Description
payload * required
(body)
Example Value | Model
{
"clump_thickness": 0,
"uniform_cell_size": 0,
"uniform_cell_shape": 0,
"marginal_adhesion": 0,
"single_epithelial_size": 0,
"bare_nuclei": 0,
"bland_chromatin": 0,
"normal_nucleoli": 0,
"mitoses": 0
}
Parameter content type
application/json
Responses
Response content type
application/json