**account.css**

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

}

.topnav {

overflow: hidden;

background-color: #0d3e4f;

}

.topnav a {

float: right;

color: #f2f2f2;

text-align: center;

padding: 14px 16px;

text-decoration: none;

font-size: 17px;

}

.topnav a:hover {

background-color: #ddd;

color: black;

}

\* {

box-sizing: border-box;

}

body {

font-family: Arial, Helvetica, sans-serif;

}

/\* Float four columns side by side \*/

.column {

float: left;

width: 25%;

padding-top: 10px;

padding-right: 10px;

padding-bottom: 10px;

padding-left: 10px;

}

/\* Remove extra left and right margins, due to padding \*/

.row {margin: 0 -5px;}

/\* Clear floats after the columns \*/

.row:after {

content: "";

display: table;

clear: both;

}

/\* Responsive columns \*/

@media screen and (max-width: 600px) {

.column {

width: 100%;

display: block;

margin-bottom: 20px;

}

}

/\* Style the counter cards \*/

.card {

box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2);

padding: 16px;

text-align: center;

background-color: #f1f1f1;

}

.foo {

float: medal;

width: 20px;

height: 20px;

margin: 5px;

border: 1px solid rgba(0, 0, 0, .2);

}

.green {

background: #4dff4d;

}

.grey {

background: #5e5e5e;

}

body {font-family: Arial, Helvetica, sans-serif;}

.callout {

position: fixed;

bottom: 35px;

right: 20px;

margin-left: 20px;

max-width: 300px;

}

.callout-header {

padding: 25px 15px;

background: #0d3e4f;

font-size: 30px;

color: white;

}

.callout-container {

padding: 15px;

background-color: #ccc;

color: black

}

.closebtn {

position: absolute;

top: 5px;

right: 15px;

color: white;

font-size: 30px;

cursor: pointer;

}

.closebtn:hover {

color: lightgrey;

}

**login.css**

body {

background-color: #D6DBDF;

font-family: Arial, Helvetica, sans-serif;

}

.container {

margin: auto;

width: 30%;

padding: 100px;

}

/\* Full-width input fields \*/

input[type=text], input[type=password] {

width: 100%;

padding: 12px 20px;

margin: 8px 0;

display: inline-block;

border: 1px solid #ccc;

box-sizing: border-box;

}

/\* Set a style for all buttons \*/

button {

background-color: #0d3e4f;

color: white;

font-size: 17px;

border: none;

padding: 14px 16px;

cursor: pointer;

width: 100%;

}

button:hover {

opacity: 0.8;

}

/\* Extra styles for the cancel button \*/

.cancelbtn {

width: auto;

padding: 10px 18px;

background-color: #f44336;

}

/\* Center the image and position the close button \*/

.imgcontainer {

text-align: center;

margin: 24px 0 12px 0;

position: relative;

}

.container {

padding: 16px;

}

span.psw {

float: right;

padding-top: 16px;

}

/\* The Modal (background) \*/

.modal {

display: none; /\* Hidden by default \*/

position: fixed; /\* Stay in place \*/

z-index: 1; /\* Sit on top \*/

left: 0;

top: 0;

width: 100%; /\* Full width \*/

height: 100%; /\* Full height \*/

overflow: auto; /\* Enable scroll if needed \*/

background-color: rgb(0,0,0); /\* Fallback color \*/

background-color: rgba(0,0,0,0.4); /\* Black w/ opacity \*/

padding-top: 60px;

}

/\* Modal Content/Box \*/

.modal-content {

background-color: #fefefe;

margin: 5% auto 15% auto; /\* 5% from the top, 15% from the bottom and centered \*/

border: 1px solid #888;

width: 40%; /\* Could be more or less, depending on screen size \*/

}

/\* The Close Button (x) \*/

.close {

position: absolute;

right: 25px;

top: 0;

color: #000;

font-size: 35px;

font-weight: bold;

}

.close:hover,

.close:focus {

color: red;

cursor: pointer;

}

/\* Add Zoom Animation \*/

.animate {

-webkit-animation: animatezoom 0.6s;

animation: animatezoom 0.6s

}

@-webkit-keyframes animatezoom {

from {-webkit-transform: scale(0)}

to {-webkit-transform: scale(1)}

}

@keyframes animatezoom {

from {transform: scale(0)}

to {transform: scale(1)}

}

/\* Change styles for span and cancel button on extra small screens \*/

@media screen and (max-width: 300px) {

span.psw {

display: block;

float: none;

}

.cancelbtn {

width: 100%;

}

}

.login {

font-size: 25px;

}

**main.css**

body {

margin: 0;

font-family: Arial, Helvetica, sans-serif;

}

.topnav {

overflow: hidden;

background-color: #0d3e4f;

}

.topnav a {

float: right;

color: #f2f2f2;

text-align: center;

text-decoration: none;

font-size: 17px;

}

.topnav a:hover {

background-color: #ddd;

color: black;

}

.topnav a.active {

background-color: #2a7a57;

color: white;

}

\* {box-sizing: border-box;}

body {font-family: Verdana, sans-serif;}

.mySlides {display: none;}

img {vertical-align: middle;}

/\* Slideshow container \*/

.slideshow-container {

max-width: 1000px;

padding: 40px 12px;

position: relative;

margin: auto;

}

/\* Caption text \*/

.text {

color: #f2f2f2;

font-size: 15px;

padding: 8px 12px;

position: absolute;

bottom: 8px;

width: 100%;

text-align: center;

}

/\* Number text (1/3 etc) \*/

.numbertext {

color: #f2f2f2;

font-size: 12px;

padding: 8px 12px;

position: absolute;

top: 0;

}

/\* The dots/bullets/indicators \*/

.dot {

height: 15px;

width: 15px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active {

background-color: #717171;

}

/\* Fading animation \*/

.fade {

-webkit-animation-name: fade;

-webkit-animation-duration: 1.5s;

animation-name: fade;

animation-duration: 1.5s;

}

@-webkit-keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

/\* On smaller screens, decrease text size \*/

@media only screen and (max-width: 300px) {

.text {font-size: 11px}

}

**main.js**

var slideIndex = 0;

showSlides();

function showSlides() {

var i;

var slides = document.getElementsByClassName("mySlides");

var dots = document.getElementsByClassName("dot");

for (i = 0; i < slides.length; i++) {

slides[i].style.display = "none";

}

slideIndex++;

if (slideIndex > slides.length) {slideIndex = 1}

for (i = 0; i < dots.length; i++) {

dots[i].className = dots[i].className.replace(" active", "");

}

slides[slideIndex-1].style.display = "block";

dots[slideIndex-1].className += " active";

setTimeout(showSlides, 3100); // Change image every 2 seconds

}

//var age = ["Female"];

//for (i = 0; i < gender.length; i++) {

// var opt = document.createElement("option");

// document.getElementById("gender").innerHTML += '<option id="' + i + '">' + gender[i] + '</option>';

//}

**main10.css**

body {

background: #fafafa;

color: #333333;

margin-top: 5rem;

}

h1, h2, h3, h4, h5, h6 {

color: #444444;

}

.bg-steel {

background-color: #5f788a;

}

.site-header .navbar-nav .nav-link {

color:black;

}

.site-header .navbar-nav .nav-link:hover {

color: #ffffff;

}

.site-header .navbar-nav .nav-link.active {

font-weight: 500;

}

.content-section {

background: #ffffff;

padding: 10px 20px;

border: 1px solid #dddddd;

border-radius: 3px;

margin-bottom: 20px;

}

.article-title {

color: #444444;

}

a.article-title:hover {

color: #428bca;

text-decoration: none;

}

.article-content {

white-space: pre-line;

}

.article-img {

height: 65px;

width: 65px;

margin-right: 16px;

}

.article-metadata {

padding-bottom: 1px;

margin-bottom: 4px;

border-bottom: 1px solid #e3e3e3

}

.article-metadata a:hover {

color: #333;

text-decoration: none;

}

.article-svg {

width: 25px;

height: 25px;

vertical-align: middle;

}

.account-img {

height: 125px;

width: 125px;

margin-right: 20px;

margin-bottom: 16px;

}

.account-heading {

font-size: 2.5rem;

}

table {

border-collapse: collapse;

width: 50%;

}

th, td {

text-align: left;

padding: 15px;

}

tr:nth-child(even){background-color: #f2f2f2}

th {

background-color: #4CAF50;

color: white;

}

style.css

body {

font-family: Garamond;

}

h1 {

color: black;

margin-bottom: 0;

margin-top: 0;

text-align: center;

font-size: 40px;

}

h3 {

color: black;

font-size: 20px;

margin-top: 3px;

text-align: center;

}

#chatbox {

margin-left: auto;

margin-right: auto;

width: 40%;

margin-top: 60px;

}

#userInput {

margin-left: auto;

margin-right: auto;

width: 40%;

margin-top: 60px;

}

#textInput {

width: 87%;

border: none;

border-bottom: 3px solid #009688;

font-family: monospace;

font-size: 17px;

}

#buttonInput {

padding: 3px;

font-family: monospace;

font-size: 17px;

}

.userText {

color: white;

font-family: monospace;

font-size: 17px;

text-align: right;

line-height: 30px;

}

.userText span {

background-color: #009688;

padding: 10px;

border-radius: 2px;

}

.botText {

color: white;

font-family: monospace;

font-size: 17px;

text-align: left;

line-height: 30px;

}

.botText span {

background-color: #EF5350;

padding: 10px;

border-radius: 2px;

}

#tidbit {

position:absolute;

bottom:0;

right:0;

width: 300px;

}

**about.html**

{% extends "layout.html" %}

{% block content %}

<center><h3>

<p>Fetal Health is the indicator of fetal wellbeing and regular contact in the uterus of pregnant women during pregnancy.

Most pregnancy period complication leads fetus to a severe problem which restricts proper growth that causes

impairment or death. Safe pregnancy period by predicting risk levels before the occasion of complications encourage

proper fetal growth. Prediction of fetal health state from a set of pre-classified patterns knowledge is a common

important in developing a predictive classifier model using data mining algorithms. In this paper the University of

California Irvine (UCI) Cardiotocogram (CTG) dataset is the source of information required for building a model which

predicts whether the fetal is healthy or unhealthy. It consists of 2126 instances with 22 attributes obtained from FHR and

UC which are multivariate datatypes. The major goal of this study is to choose efficient data mining algorithm which

develops the accurate predictive model to manage knowledge of fetal health during pregnancy.</p>

{% endblock content %}

**home.html**

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='main.css') }}">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='login.css') }}">

<title>{{ title }}</title>

</head>

<body>

<div class="topnav">

<a href="about"><button>About</button></a>

<a href="user\_login"><button onclick="document.getElementById('id01').style.display='block'">User</button></a>

<!--a href="doctor\_login"><button onclick="document.getElementById('id02').style.display='block'">Doctor</button></a-->

<a href="/home"><button class="active">Home</button></a>

</div>

<h1><center>Fetal Health Prediction Using Machine Learning</center></h1>

<center>

<div class="slideshow-container">

<div class="mySlides fade">

<img src="{{url\_for('static', filename='1.jpg')}}" align="middle" style="width:1000px;height:400px;"/>

</div>

<div class="mySlides fade">

<img src="{{url\_for('static', filename='2.jpg')}}" align="middle"

style="width:1000px;height:400px;"/>

</div>

<div class="mySlides fade">

<img src="{{url\_for('static', filename='3.jpg')}}" align="middle" style="width:1000px;height:400px;"/>

</div>

<div class="mySlides fade">

<img src="{{url\_for('static', filename='4.jpg')}}" align="middle" style="width:1000px;height:400px;"/>

</div>

<div class="mySlides fade">

<img src="{{url\_for('static', filename='5.jpg')}}" align="middle" />

</div>

</div>

</center>

<br>

<div style="text-align:center">

<span class="dot"></span>

<span class="dot"></span>

<span class="dot"></span>

<span class="dot"></span>

</div>

<script src="{{url\_for('static', filename='login.js')}}"></script>

<script src="{{url\_for('static', filename='main.js')}}"></script>

</body>

</html>

**layout.html**

<!DOCTYPE html>

<html lang="en">

<head>

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='main.css') }}">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='login.css') }}">

<title>Title</title>

</head>

<body>

<div class="topnav">

<a href="about"><button>About</button></a>

<a href="user\_login"><button onclick="document.getElementById('id01').style.display='block'">User</button></a>

<!--a href="doctor\_login"><button onclick="document.getElementById('id02').style.display='block'">Admin</button></a-->

<a href="/home"><button class="active">Home</button></a>

</div>

<center><h1>Fetal Health Prediction Using Machine Learning</h1></center>

{% block content %}{% endblock %}

</body>

</html>

**Search.html**

{% extends "layout.html" %}

{% block content %}

<form action = "/analyse" method = "post" enctype="multipart/form-data">

<input class="btn btn-outline-info" type="file" name="file1" />

<input class="btn btn-outline-info" type="file" name="file2" />

<input class="btn btn-outline-info" type = "submit" value="predict">

</form>

<center>{% with messages = get\_flashed\_messages() %}

{% if messages %}

<ul class="flashes">

{% for message in messages %}

<div class="message\_flash">{{ message }}</div>

{% endfor %}

</ul>

{% endif %}

{% endwith %}</center>

{% endblock content %}

**User\_account.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='account.css') }}">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='login.css') }}">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='main.css') }}">

<title>Title</title>

</head>

<body>

<div class="topnav">

<a class="active" href>

<a class="active" href="/logout">logout</a></a>

</div>

<center>{% with messages = get\_flashed\_messages() %}

{% if messages %}

<ul class="flashes">

{% for message in messages %}

<div class="message\_flash">{{ message }}</div>

{% endfor %}

</ul>

{% endif %}

{% endwith %}</center>

<div class="container">

<br>

<br>

<center>

<form action = "/predict" method = "post" enctype="multipart/form-data">

<label for="psw"><b></b></label> baseline\_value <input type="text" placeholder="" name="baseline\_value" ><br>

<label for="psw"><b></b></label> accelerations <input type="text" placeholder="" name="accelerations" ><br>

<label for="psw"><b></b></label> fetal\_movement <input type="text" placeholder="" name="fetal\_movement" ><br>

<label for="psw"><b></b></label> uterine\_contractions <input type="text" placeholder="" name="uterine\_contractions" ><br>

<label for="psw"><b></b></label> light\_decelerations <input type="text" placeholder="" name="light\_decelerations" ><br>

<label for="psw"><b></b></label> severe\_decelerations <input type="text" placeholder="" name="severe\_decelerations" ><br>

<br>

<br>

<button type="submit">Predict</button>

</form></center>

</div>

</body>

</html>

**User login.html**

{% extends "layout.html" %}

{% block content %}

<center>{% with messages = get\_flashed\_messages() %}

{% if messages %}

<ul class="flashes">

{% for message in messages %}

<div class="message\_flash">{{ message }}</div>

{% endfor %}

</ul>

{% endif %}

{% endwith %}</center>

<form action="/user\_login" method="post">

<div class="container">

<label for="email"><b>User name</b></label>

<input type="text" placeholder="Enter Email" name="email" >

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="psw" >

<button type="submit">Login</button>

<label><b>New user</b></label>

<a href="/user\_register" class ='btn2' type="submit">register</a>

</div>

</form>

{% endblock content %}

</body>

</html>

**User\_register.html**

{% extends "layout.html" %}

{% block content %}

<form action="/user\_register" method="post">

<div class="container">

<label for="uname"><b>Username</b></label>

<input type="text" placeholder="Enter Name" name="uname" >

<label for="email"><b>Email id</b></label>

<input type="text" placeholder="Enter Email" name="email" >

<label for="psw"><b>Password</b></label>

<input type="password" placeholder="Enter Password" name="psw" >

<!--<label for="gender"><b>gender</b></label>

<input type="text" placeholder="" name="gender" />-->

<label for="gender"><b>Choose a Gender:</b></label>

<select id="gender" name="gender">

<!--<option value="male">male</option>-->

<option value="female">female</option>

</select><br>

<br>

<!--<label for="age"><b>Age</b></label>

<input type="text" placeholder="Enter your age" name="age" />-->

<label for="age"><b>Choose a age:</b></label>

<select id="age" name="age">

<option value="15">15</option>

<option value="16">16</option>

<option value="17">17</option>

<option value="18">18</option>

<option value="19">19</option>

<option value="20">20</option>

<option value="21">21</option>

<option value="22">22</option>

<option value="23">23</option>

<option value="24">24</option>

<option value="25">25</option>

<option value="26">26</option>

<option value="27">27</option>

<option value="28">28</option>

<option value="29">29</option>

<option value="30">30</option>

<option value="31">31</option>

<option value="32">32</option>

<option value="33">33</option>

<option value="34">34</option>

<option value="35">35</option>

<option value="36">36</option>

<option value="37">37</option>

<option value="38">38</option>

<option value="39">39</option>

<option value="40">40</option>

</select><br>

<br>

<button type="submit">Submit</button>

</div>

</form>

{% endblock content %}

**Fetal\_database.txt**

name varchar(20) DEFAULT NULL,

email varchar(50) DEFAULT NULL,

password varchar(20) DEFAULT NULL,

gender varchar(10) DEFAULT NULL,

age int(11) DEFAULT NULL

**Main.py**

import os

#-------------------------------------------------model\_code------------------------------------------------------------

import sqlite3

conn = sqlite3.connect('Fetal\_database')

cur = conn.cursor()

try:

cur.execute('''CREATE TABLE user (

name varchar(20) DEFAULT NULL,

email varchar(50) DEFAULT NULL,

password varchar(20) DEFAULT NULL,

gender varchar(10) DEFAULT NULL,

age int(11) DEFAULT NULL

)''')

except:

pass

#!/usr/bin/env python

# coding: utf-8

# importing libraries

import pandas as pd #Easily handles missing data,It provides an efficient way to slice the data,It includes a powerful time series tool to work

import numpy as np

#import matplotlib.pyplot as plt

#import seaborn as sns

#uploading dataset

data = pd.read\_csv('fetal\_health.csv',encoding= 'unicode\_escape')

#independent and dependent variable

X = data.iloc[:,:-7:-1]

y = data.iloc[:,-1]

# spliting up to testing and training dataset

from sklearn.model\_selection import train\_test\_split

X\_train , X\_test , y\_train , y\_test = train\_test\_split(X,y,test\_size=0.2,random\_state=100)

# navie bayes

import sklearn

from sklearn.naive\_bayes import BernoulliNB

from sklearn import metrics

from sklearn.metrics import accuracy\_score

BernNB = BernoulliNB(binarize=.1)

BernNB.fit(X\_train,y\_train)

print(BernNB)

y\_expect = y\_test

y\_pred = BernNB.predict(X\_test)

print(accuracy\_score(y\_expect,y\_pred)\*100)

score\_nb = accuracy\_score(y\_expect,y\_pred)\*100

# logistic regression

from sklearn.linear\_model import LogisticRegression

reg = LogisticRegression().fit(X\_train, y\_train)

score\_lr = reg.score(X\_test, y\_test)

score\_lr= score\_lr \* 100

score\_lr

# Random forest

from sklearn.ensemble import RandomForestClassifier

regr = RandomForestClassifier(max\_depth=3, random\_state=0)

regr.fit(X\_train, y\_train)

score\_rf = regr.score(X\_test,y\_test)

score\_rf =score\_rf\*100

score\_rf

# Decision tree

from sklearn.tree import DecisionTreeClassifier

clf = DecisionTreeClassifier(max\_depth=7)

clf.fit(X\_train, y\_train)

score\_dt = clf.score(X\_test,y\_test)

score\_dt = score\_dt \* 100

score\_dt

scores = [score\_nb,score\_lr,score\_rf,score\_dt]

algorithems = ['navie bayes','logistic regressor','random forest','Decision Tree']

for i in range(len(algorithems)):

print('the accuracy score achieved using' +algorithems[i]+ 'is :'+str(scores[i])+" %")

from flask import Flask,render\_template, url\_for,request, flash, redirect, session

app = Flask(\_\_name\_\_)

app.config['SECRET\_KEY'] = '881e69e15e7a528830975467b9d87a98'

#-------------------------------------home\_page-------------------------------------------------------------------------

@app.route('/')

@app.route('/home')

def home():

if not session.get('logged\_in'):

return render\_template('home.html')

else:

return redirect(url\_for('user\_account'))

#-------------------------------------about\_page-------------------------------------------------------------------------

@app.route("/about")

def about():

return render\_template('about.html')

#-------------------------------------about\_page-------------------------------------------------------------------------

#-------------------------------------user\_login\_page-------------------------------------------------------------------------

@app.route('/user\_login',methods = ['POST', 'GET'])

def user\_login():

conn = sqlite3.connect('Fetal\_database')

cur = conn.cursor()

if request.method == 'POST':

email = request.form['email']

password = request.form['psw']

print('asd')

count = cur.execute('SELECT \* FROM user WHERE email = "%s" AND password = "%s"' % (email, password))

print(count)

#conn.commit()

#cur.close()

l = len(cur.fetchall())

if l > 0:

flash( f'Successfully Logged in' )

return render\_template('user\_account.html')

else:

print('hello')

flash( f'Invalid Email and Password!' )

return render\_template('user\_login.html')

# -------------------------------------user\_login\_page-----------------------------------------------------------------

# -------------------------------------user\_register\_page-------------------------------------------------------------------------

@ app.route('/user\_register', methods=['POST', 'GET'])

def user\_register():

conn = sqlite3.connect('Fetal\_database')

cur = conn.cursor()

if request.method == 'POST':

name = request.form['uname']

email = request.form['email']

password = request.form['psw']

gender = request.form['gender']

age = request.form['age']

cur.execute("insert into user(name,email,password,gender,age) values ('%s','%s','%s','%s','%s')" % (name, email, password, gender, age))

conn.commit()

# cur.close()

print('data inserted')

return redirect(url\_for('user\_login'))

return render\_template('user\_register.html')

# -------------------------------------user\_register\_page-------------------------------------------------------------------------

@app.route('/predict', methods=['POST', 'GET'])

def predict():

baseline\_value = request.form['baseline\_value']

accelerations = request.form['accelerations']

fetal\_movement = request.form['fetal\_movement']

uterine\_contractions = request.form['uterine\_contractions']

light\_decelerations = request.form['light\_decelerations']

severe\_decelerations = request.form['severe\_decelerations']

global clf

if request.method == 'POST':

out = clf.predict([[float(baseline\_value),

float(accelerations),

float(fetal\_movement),

float(uterine\_contractions),

float(light\_decelerations),

float(severe\_decelerations)]])

if out == 1:

print("Abnormal")

flash('Abnormal')

elif out == 2:

print("neutral")

flash('neutral')

else:

print("Normal")

flash('Normal')

return render\_template('user\_account.html')

# ------------------------------------predict\_page-----------------------------------------------------------------

# -------------------------------------user\_logout\_page-------------------------------------------------------------------------

@app.route("/logout")

def logout():

session['logged\_in'] = False

return home()

@app.route("/logoutd",methods = ['POST','GET'])

def logoutd():

return home()

# -------------------------------------user\_logout\_page-------------------------------------------------------------------------

if \_\_name\_\_ == '\_\_main\_\_':

app.secret\_key = os.urandom(12)

app.run(debug=True)