import numpy as np

import pandas as pd

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

from sklearn.svm import SVC

from flask import Flask,render\_template,request

# import CSV data

DataCsv = pd.read\_csv('data.csv')

# Convert CSV data into Excel format

DataCsv.to\_excel('CancerData.xlsx', index=None, header=True)

DataExcel = pd.read\_excel('CancerData.xlsx')

# Working on CSV files

# Drop last column having values of no use

DataCsv = DataCsv.dropna(axis=1)

# Count Malignant and Benign patients

# print(DataCsv['diagnosis'].value\_counts())

# Encode the categorical values

# Malignant->1

# Benign->0

for i in range(0, len(DataCsv)):

if DataCsv.iloc[i, 1] == 'M':

DataCsv.iloc[i, 1] = float(1)

elif DataCsv.iloc[i, 1] == 'B':

DataCsv.iloc[i, 1] = float(0)

# Split our data into independent (x->2-31) and Dependent (y->1) Data sets

x = DataCsv.iloc[:, 2:32].values

y = DataCsv.iloc[:, 1].values

# Train the model on 75% of our data

# Test the model with remaining 25% of the data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(x, y, test\_size=0.05)

print(X\_train)

# Validating Data

X\_train = np.array(X\_train)

y\_train = np.array(y\_train)

X\_train = X\_train.astype(np.float64)

y\_train = y\_train.astype(np.float64)

X\_test = np.array(X\_test)

y\_test = np.array(y\_test)

X\_test = X\_test.astype(np.float64)

y\_test = y\_test.astype(np.float64)

# Data Normalization

'''for row in range(len(X\_train)):

Xmax=np.max(X\_train[row,:])

Xmin=np.min(X\_train[row,:])

for col in range(29):

X=X\_train[row][col]

X\_train[row][col]=(X-Xmin)/(Xmax-Xmin)'''

# Use SVM modelling to train the data

svc\_lin=SVC(kernel='linear',random\_state=0)

svc\_lin.fit(X\_train,y\_train)

print('Training Accuracy:',svc\_lin.score(X\_train,y\_train))

# print('Testing Accuracy:',svc\_lin.score(X\_test,y\_test))

# Web App

Main=Flask(\_\_name\_\_)

@Main.route('/')

def design():

return render\_template('design.html')

@Main.route('/', methods=['POST'])

def getvalue():

Radiusmean=request.form['radiusmean']

Texturemean = request.form['texturemean']

Perimetermean = request.form['perimetermean']

Areamean=request.form['areamean']

Smoothnessmean=request.form['smothnessmean']

Compactnessmean = request.form['compactnessmean']

Concativiytymean = request.form['concativitymean']

Concavepointmean = request.form['concavepointmean']

Symmetrymean = request.form['symmetrymean']

FractalDimensionMean = request.form['fractaldimensionmean']

RadiusSe = request.form['radiusse']

Texturese=request.form['Texturese']

Perimeterse = request.form['Perimeterse']

AreaSE = request.form['arease']

SmoothnessSE=request.form['smoothnessse']

compactnessSE=request.form['compactnessse']

ConcativiytySE = request.form['concativityse']

concavepointse=request.form['concavepointse']

Symmetryse=request.form['symmetryse']

FractalDimensionSE = request.form['fractaldimensionse']

Radiusworst=request.form['radiusworst']

TextureWorst=request.form['textureworst']

PerimeterWorst=request.form['perimeterworst']

AreaWorst=request.form['areaworst']

SmoothnessWorst = request.form['smoothnessworst']

CompactnessWorst=request.form['compactnessworst']

ConcativtyWorst = request.form['concativtyworst']

Concavepointworst=request.form['concavepointworst']

SymmetryWorst = request.form['symmetryworst']

FractalDimansionWorst = request.form['fractaldimansionworst']

X\_test=[[Radiusmean,Texturemean,Perimetermean,Areamean,Smoothnessmean,Compactnessmean,

Concativiytymean,Concavepointmean,Symmetrymean,FractalDimensionMean,

RadiusSe,Texturese,Perimeterse,AreaSE,SmoothnessSE,compactnessSE,ConcativiytySE,

concavepointse,Symmetryse,FractalDimensionSE,

Radiusworst,TextureWorst,PerimeterWorst,AreaWorst,SmoothnessWorst,

CompactnessWorst,ConcativtyWorst,Concavepointworst,SymmetryWorst,FractalDimansionWorst],

[Radiusmean, Texturemean, Perimetermean, Areamean, Smoothnessmean, Compactnessmean,

Concativiytymean, Concavepointmean, Symmetrymean, FractalDimensionMean,

RadiusSe, Texturese, Perimeterse, AreaSE, SmoothnessSE, compactnessSE, ConcativiytySE,

concavepointse, Symmetryse, FractalDimensionSE,

Radiusworst, TextureWorst, PerimeterWorst, AreaWorst, SmoothnessWorst,

CompactnessWorst, ConcativtyWorst, Concavepointworst, SymmetryWorst, FractalDimansionWorst]]

y\_test=[1,0]

result=(svc\_lin.score(X\_test, y\_test))

print('Testing Accuracy:', svc\_lin.score(X\_test, y\_test))

return render\_template('pass.html',

RM=Radiusmean,

TM=Texturemean,

PM=Perimetermean,

AM=Areamean ,

SM=Smoothnessmean,

CM=Compactnessmean,

ConMN=Concativiytymean,

conpointmean=Concavepointmean,

symMn=Symmetrymean,

FM=FractalDimensionMean,

radiusSE=RadiusSe,

Texse=Texturese,

PerSE=Perimeterse,

ArSe=AreaSE ,

smoSE=SmoothnessSE,

compSE=compactnessSE,

ConSe=ConcativiytySE,

concvptse=concavepointse,

SymSE=Symmetryse,

FracDimSE=FractalDimensionSE,

RW=Radiusworst,

TW=TextureWorst,

PW=PerimeterWorst,

AW=AreaWorst,

SmoWorst=SmoothnessWorst,

CompW=CompactnessWorst,

ConWrst=ConcativtyWorst,

ConptW=Concavepointworst,

SymmWrst=SymmetryWorst,

FracDimWrst=FractalDimansionWorst)

@Main.route('/Result')

def Result():

Radiusmean = request.form['radiusmean']

Texturemean = request.form['texturemean']

Perimetermean = request.form['perimetermean']

Areamean = request.form['areamean']

Smoothnessmean = request.form['smothnessmean']

Compactnessmean = request.form['compactnessmean']

Concativiytymean = request.form['concativitymean']

Concavepointmean = request.form['concavepointmean']

Symmetrymean = request.form['symmetrymean']

FractalDimensionMean = request.form['fractaldimensionmean']

RadiusSe = request.form['radiusse']

Texturese = request.form['Texturese']

Perimeterse = request.form['Perimeterse']

AreaSE = request.form['arease']

SmoothnessSE = request.form['smoothnessse']

compactnessSE = request.form['compactnessse']

ConcativiytySE = request.form['concativityse']

concavepointse = request.form['concavepointse']

Symmetryse = request.form['symmetryse']

FractalDimensionSE = request.form['fractaldimensionse']

Radiusworst = request.form['radiusworst']

TextureWorst = request.form['textureworst']

PerimeterWorst = request.form['perimeterworst']

AreaWorst = request.form['areaworst']

SmoothnessWorst = request.form['smoothnessworst']

CompactnessWorst = request.form['compactnessworst']

ConcativtyWorst = request.form['concativtyworst']

Concavepointworst = request.form['concavepointworst']

SymmetryWorst = request.form['symmetryworst']

FractalDimansionWorst = request.form['fractaldimansionworst']

X\_test = [[Radiusmean, Texturemean, Perimetermean, Areamean, Smoothnessmean, Compactnessmean,

Concativiytymean, Concavepointmean, Symmetrymean, FractalDimensionMean,

RadiusSe, Texturese, Perimeterse, AreaSE, SmoothnessSE, compactnessSE, ConcativiytySE,

concavepointse, Symmetryse, FractalDimensionSE,

Radiusworst, TextureWorst, PerimeterWorst, AreaWorst, SmoothnessWorst,

CompactnessWorst, ConcativtyWorst, Concavepointworst, SymmetryWorst, FractalDimansionWorst],

[Radiusmean, Texturemean, Perimetermean, Areamean, Smoothnessmean, Compactnessmean,

Concativiytymean, Concavepointmean, Symmetrymean, FractalDimensionMean,

RadiusSe, Texturese, Perimeterse, AreaSE, SmoothnessSE, compactnessSE, ConcativiytySE,

concavepointse, Symmetryse, FractalDimensionSE,

Radiusworst, TextureWorst, PerimeterWorst, AreaWorst, SmoothnessWorst,

CompactnessWorst, ConcativtyWorst, Concavepointworst, SymmetryWorst, FractalDimansionWorst]]

y\_test = [1, 0]

result = (svc\_lin.score(X\_test, y\_test))

print('Testing Accuracy:', svc\_lin.score(X\_test, y\_test))

return render\_template("Result.html",RESULT=result)

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

Main.run(debug=True)

<!DOCTYPE html>

<html>

<head>

<style type="text/css">

p.c{

white-space:pre;

}

span{

background-color: #80ced6

}

</style>

</head>

<body>

<h1 style="color:#E75480;font-size:50px;"><span>Breast Cancer Prediction</span></h1>

<p class="c">

<img style="border:10px outset Black;" src="../static/images/Home.gif" alt="Home" style="width:50%;" height="470">

</body>

<head>

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

<style type="text/css">

table {

border-collapse: collapse;

width: 70%;

}

th {

height: 70px;

}

table, th, td {

border: 1px solid black;

border-collapse: collapse;

}

table.center {

margin-left: auto;

margin-right: auto;

}

img {

display: block;

margin-left: auto;

margin-right: auto;

}

h1{text-align: center;}

p{text-align: center;}

div{text-align: center;}

</style>

</head>

<body>

<table class="center" style="background-color:#FFD1DC;">

<head>

<style>

body{

background:url(../static/images/cancer\_back.jpg);

background-repeat: no-repeat;

background-attachment: fixed;

background-size: 100% 100%;

}

table.center{

margin-left: auto;

margin-right: auto;

}

</style>

</head>

<form name="passdata" action="." method="post">

<tbody>

<tr>

<th

scope="col" style="width:35%;font-size:20px;">Subject(Range Value)

</th>

<th

scope="col" style="width:70%;font-size:20px;">Information

</th>

</tr>

<tr>

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

Radius Mean

</td>

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

<input name="radiusmean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Texture Mean

</td>

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

<input name="texturemean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Perimeter Mean

</td>

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

<input name="perimetermean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Area Mean

</td>

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

<input name="areamean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Smoothness Mean

</td>

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

<input name="smothnessmean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Compactness Mean

</td>

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

<input name="compactnessmean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concavity Mean

</td>

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

<input name="concativitymean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concave Point Mean

</td>

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

<input name="concavepointmean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Symmetry Mean

</td>

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

<input name="symmetrymean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Fractal Dimension Mean

</td>

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

<input name="fractaldimensionmean" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Radius SE

</td>

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

<input name="radiusse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Texture SE

</td>

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

<input name="Texturese" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Perimeter SE

</td>

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

<input name="Perimeterse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Area SE

</td>

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

<input name="arease" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Smoothness SE

</td>

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

<input name="smoothnessse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Compactness SE

</td>

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

<input name="compactnessse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concavity SE

</td>

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

<input name="concativityse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concave Point SE

</td>

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

<input name="concavepointse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Symmetry SE

</td>

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

<input name="symmetryse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Fractal Dimension SE

</td>

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

<input name="fractaldimensionse" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Radius Worst

</td>

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

<input name="radiusworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Texture Worst

</td>

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

<input name="textureworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Perimeter Worst

</td>

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

<input name="perimeterworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Area Worst

</td>

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

<input name="areaworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Smothness Worst

(0.07-0.22)

</td>

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

<input name="smoothnessworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Compactness Worst

</td>

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

<input name="compactnessworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concavity Worst

</td>

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

<input name="concativtyworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Concave Points Worst

</td>

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

<input name="concavepointworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Symmetry Worst

(0.16-0.66)

</td>

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

<input name="symmetryworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

Fractal Dimension Worst

(0.06-0.21)

</td>

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

<input name="fractaldimansionworst" class="textbook" type ="text"><br><br>

</td>

</tr>

<tr>

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

<form action="Result.html">

<input type="submit">

</form>

</td>

</tr>

</tbody>

</form>

</table>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title Title></title>

</head>

<body>

<h1>{{RM}}</h1>

<h1>{{TM}}</h1>

<h1>{{PM}}</h1>

<h1>{{AM}}</h1>

<h1>{{SM}}</h1>

<h1>{{CM}}</h1>

<h1>{{ConMN}}</h1>

<h1>{{conpointmean}}</h1>

<h1>{{symMn}}</h1>

<h1>{{FM}}</h1>

<h1>{{radiusSE}}</h1>

<h1>{{Texse}}</h1>

<h1>{{PerSE}}</h1>

<h1>{{ArSe}}</h1>

<h1>{{smoSE}}</h1>

<h1>{{compSE}}</h1>

<h1>{{ConSe}}</h1>

<h1>{{concvptse}}</h1>

<h1>{{SymSE}}</h1>

<h1>{{FracDimSE}}</h1>

<h1>{{RW}}</h1>

<h1>{{TW}}</h1>

<h1>{{PW}}</h1>

<h1>{{AW}}</h1>

<h1>{{SmoWorst}}</h1>

<h1>{{CompW}}</h1>

<h1>{{ConWrst}}</h1>

<h1>{{ConptW}}</h1>

<h1>{{SymmWrst}}</h1>

<h1>{{FracDimWrst}}</h1>

</body>

</html>

<h1>Probabability that you have Malignant or benign cancer is {{RESULT}}</h1>