**AIM: DEPLOYMENT OF MACHINE LEARNING MODELS**

**THEORY:**

# ML-Model-Flask-Deployment

This is a mini project to elaborate how Machine Learn Models are deployed on production using Flask API

# Prerequisites

You must have Numpy, Scikit Learn, Pandas (for Machine Learning Model) and Flask (for API) installed.

This project has four major parts :

model.py - This contains code for our Machine Learning model to predict employee salaries based on training data in 'hiring.csv' file.

app.py - This contains Flask APIs that receives employee details through GUI, computes the predicted value based on our model and returns it.

request.py - This uses a requests module to call APIs already defined in app.py and displays the returned value.

templates - This folder contains the HTML template to allow users to enter employee detail and displays the predicted employee salary.

# Running the project

Create the machine learning model by running below command - *python model.py*

This would create a serialized version of our model into a file model.pkl Run app.py using below command to start Flask API *python app.py*

By default, flask will run on port 5000.

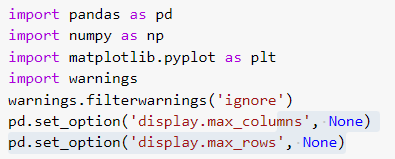
Navigate to URL http://localhost:5000

You should be able to view the homepage.

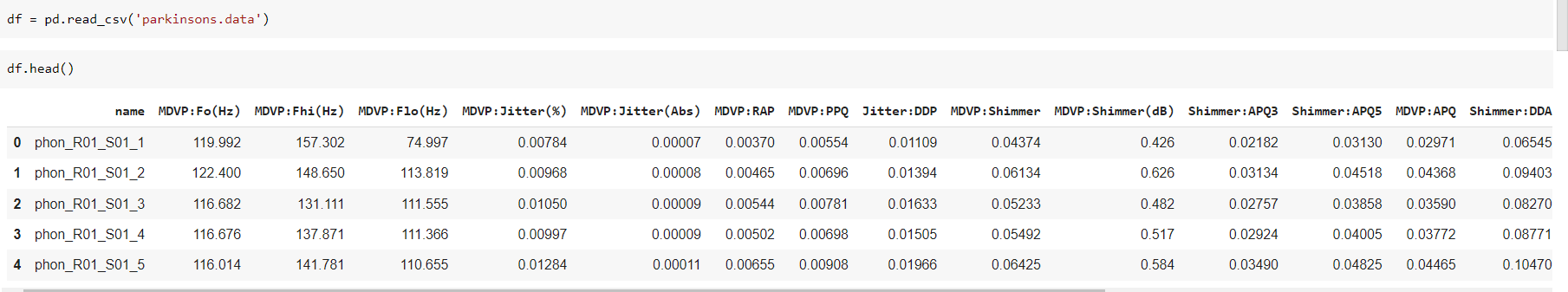
Enter valid numerical values in all 3 input boxes and hit Predict.

You will be able to see the predicted salary value on the HTML page.

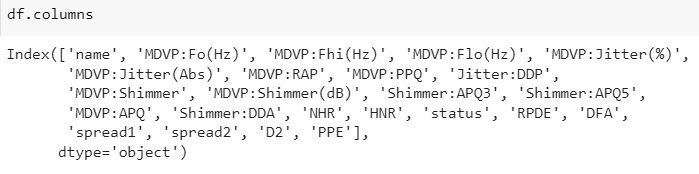
1. **CREATING MODEL:**
2. **IMPORTING LIBRARIES:**



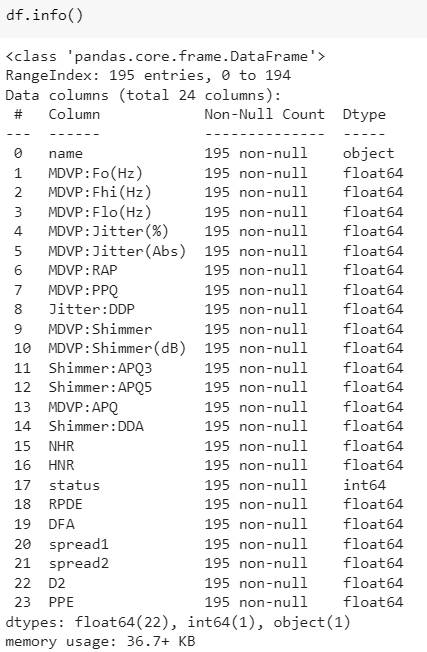
1. **READING DATASET:**



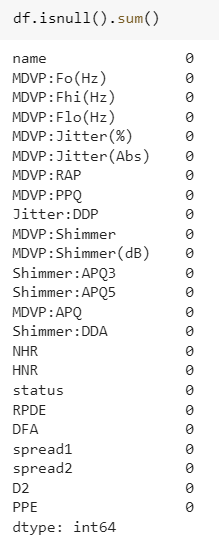
1. **LISITING ALL COLUMNS:**



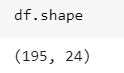
1. **DEPTH INFO OF DATASET:**



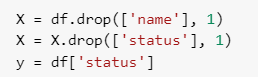
1. **COUNTING NULL VALUES:**



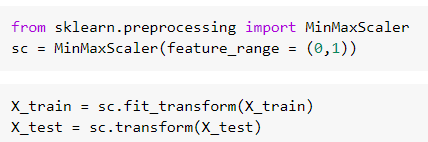
1. **DIMENSIONS OF DATASET:**



1. **SPLITTING DATASET:**



1. **NORMALIZATION:**

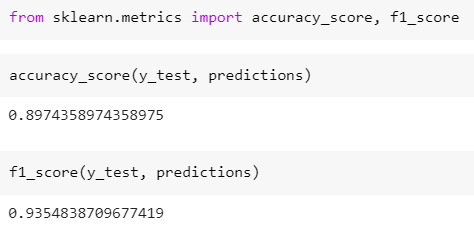


1. **BUILDING XGBOOST MODEL:**

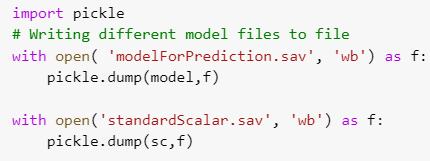




1. **CHECKING ACCURACY OF THE MODEL:**



1. **EXPORTING MODEL:**



1. **DEPLOYMENT OF MODEL:**

**FRONTEND:**

**HTML:**

**Index.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Parkinson's Diseases Prediction</title>

<link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@500&display=swap" rel="stylesheet" />

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"

    integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"

    integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"

    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"

    integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"

    crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"

    integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"

    crossorigin="anonymous"></script>

</head>

<body>

            <header>

                <div class="container">

                    <h1>

                        <p style="text-align: center">Parkinson's Diseases Prediction</p>

                    </h1>

                </div>

            </header>

    <div class="container">

            <form action="/predict" method="POST">

                <div class="mb-3">

                <input type="float" name="mdvp\_fo"  placeholder="MDVP:Fo(Hz) range(88,260)" class="form-control" required/><br/>

                <input type="float" name="mdvp\_fhi"  placeholder="MDVP:Fhi(Hz) range(102,592)" class="form-control" required><br/>

                <input type="float" name="mdvp\_flo" id="mdvp\_flo" placeholder="MDVP:Flo(Hz) range(65,240)" class="form-control" required><br />

                <input type="float" name="mdvp\_jitper" id="mdvp\_jitper" placeholder="MDVP:Jitter(%) range(0.001, 0.033)" class="form-control" required><br />

                <input type="float" name="mdvp\_jitabs" id="mdvp\_jitabs" placeholder="MDVP:Jitter(Abs) range(0.00002, 0.0002)" class="form-control" required><br />

                <input type="float" name="mdvp\_jitabs" id="mdvp\_jitabs" placeholder="MDVP:Jitter(Abs) range(0.00002, 0.0002)" class="form-control" required><br />

                <input type="float" name="mdvp\_rap" id="mdvp\_rap" placeholder="MDVP:RAP range(0.0006, 0.02)" class="form-control" required><br />

                <input type="float" name="mdvp\_ppq" id="mdvp\_ppq" placeholder="MDVP:PPQ range(0.0009, 0.02)" class="form-control" required><br />

                <input type="float" name="jitter\_ddp" id="jitter\_ddp" placeholder="Jitter:DDP range(0.002, 0.065)" class="form-control" required><br />

                <input type="float" name="mdvp\_shim" id="mdvp\_shim" placeholder="MDVP:Shimmer range(0.009, 0.12)" class="form-control" required><br />

                <input type="float" name="mdvp\_shim\_db" id="mdvp\_shim\_db" placeholder="MDVP:Shimmer(dB) range(0.085, 1.302)" class="form-control" required><br />

                <input type="float" name="shimm\_apq3" id="shimm\_apq3" placeholder="Shimmer:APQ3 range(0.004, 0.056)" class="form-control" required><br />

                <input type="float" name="shimm\_apq5" id="shimm\_apq5" placeholder="Shimmer:APQ5 range(0.005, 0.08)" class="form-control" required><br />

                <input type="float" name="mdvp\_apq" id="mdvp\_apq" placeholder="MDVP:APQ range(0.007, 0.14)" class="form-control" required><br />

                <input type="float" name="shimm\_dda" id="shimm\_dda" placeholder="Shimmer:DDA range(0.013, 0.17)" class="form-control" required><br />

                <input type="float" name="nhr" id="nhr" placeholder="NHR range(0.0006, 0.31)" class="form-control" required><br />

                <input type="float" name="hnr" id="hnr" placeholder="HNR range(8, 33)" class="form-control" required><br />

                <input type="float" name="rpde" id="rpde" placeholder="RPDE range(0.25, 0.68)" class="form-control" required><br />

                <input type="float" name="dfa" id="dfa" placeholder="DFA range(0.57, 0.82)" class="form-control" required><br />

                <input type="float" name="spread1" id="spread1" placeholder="Spread1 range(-7, -2)" class="form-control" required><br />

                <input type="float" name="spread2" id="spread2" placeholder="Spread2 range(0.006, 0.45)" class="form-control" required><br />

                <input type="float" name="d2" id="d2" placeholder="D2 range(1.42, 3.67)" class="form-control" required><br />

                <input type="float" name="ppe" id="ppe" placeholder="PPE range(0.04, 0.5)" class="form-control" required><br />

                <button type="submit" class="btn btn-primary" value="Predict">Predict</button>

                </div>

            </form>

        </div>

</body>

</html>

**Result.html:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Predicted Result</title>

    <link href="https://fonts.googleapis.com/css2?family=Quicksand:wght@500&display=swap" rel="stylesheet" />

    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet"

        integrity="sha384-1BmE4kWBq78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"

        integrity="sha384-ka7Sk0Gln4gmtz2MlQnikT1wXgYsOg+OMhuP+IlRH9sENBO0LRn5q+8nbTov4+1p"

        crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.10.2/dist/umd/popper.min.js"

        integrity="sha384-7+zCNj/IqJ95wo16oMtfsKbZ9ccEh31eOz1HGyDuCQ6wgnyJNSYdrPa03rtR1zdB"

        crossorigin="anonymous"></script>

    <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.min.js"

        integrity="sha384-QJHtvGhmr9XOIpI6YVutG+2QOK9T+ZnN4kzFN1RtK3zEFEIsxhlmWl5/YESvpZ13"

        crossorigin="anonymous"></script>

</head>

<body>

    <div class="container">

        <h2>Predicted Result</h2>

        <p>{{prediction}}</p>

    </div>

</body>

</html>

**BACKEND:**

**PYTHON:**

**app.py:**

# importing the necessary dependencies

from flask import Flask, render\_template, request

from flask\_cors import CORS,cross\_origin

import pickle

app = Flask(\_\_name\_\_) # initializing a flask app

@app.route('/',methods=['GET'])  # route to display the home page

@cross\_origin()

def homePage():

    return render\_template("index.html")

@app.route('/predict',methods=['POST','GET']) # route to show the predictions in a web UI

@cross\_origin()

def index():

    if request.method == 'POST':

            #  reading the inputs given by the user

        mdvp\_fo=float(request.form['mdvp\_fo'])

        mdvp\_fhi=float(request.form['mdvp\_fhi'])

        mdvp\_flo=float(request.form['mdvp\_flo'])

        mdvp\_jitper=float(request.form['mdvp\_jitper'])

        mdvp\_jitabs=float(request.form['mdvp\_jitabs'])

        mdvp\_rap=float(request.form['mdvp\_rap'])

        mdvp\_ppq=float(request.form['mdvp\_ppq'])

        jitter\_ddp=float(request.form['jitter\_ddp'])

        mdvp\_shim=float(request.form['mdvp\_shim'])

        mdvp\_shim\_db=float(request.form['mdvp\_shim\_db'])

        shimm\_apq3=float(request.form['shimm\_apq3'])

        shimm\_apq5=float(request.form['shimm\_apq5'])

        mdvp\_apq=float(request.form['mdvp\_apq'])

        shimm\_dda=float(request.form['shimm\_dda'])

        nhr=float(request.form['nhr'])

        hnr=float(request.form['hnr'])

        rpde=float(request.form['rpde'])

        dfa=float(request.form['dfa'])

        spread1=float(request.form['spread1'])

        spread2=float(request.form['spread2'])

        d2=float(request.form['d2'])

        ppe=float(request.form['ppe'])

        filename = 'modelForPrediction.sav'

        loaded\_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage

            # predictions using the loaded model file

        scaler = pickle.load(open('standardScalar.sav', 'rb'))

        prediction=loaded\_model.predict(scaler.transform([[mdvp\_fo,mdvp\_fhi,mdvp\_flo,mdvp\_jitper, mdvp\_jitabs,

                mdvp\_rap,mdvp\_ppq, jitter\_ddp, mdvp\_shim, mdvp\_shim\_db,shimm\_apq3,shimm\_apq5,mdvp\_apq,shimm\_dda,nhr,hnr,rpde,dfa,spread1,spread2,d2,ppe]]))

        print('prediction is', prediction)

        if prediction == 1:

            pred = "You have Parkinson's Disease. Please consult a specialist."

            return render\_template('results.html', prediction=pred)

        else:

            pred = "You are Healthy Person."

            # showing the prediction results in a UI

            return render\_template('results.html',prediction=pred)

    else:

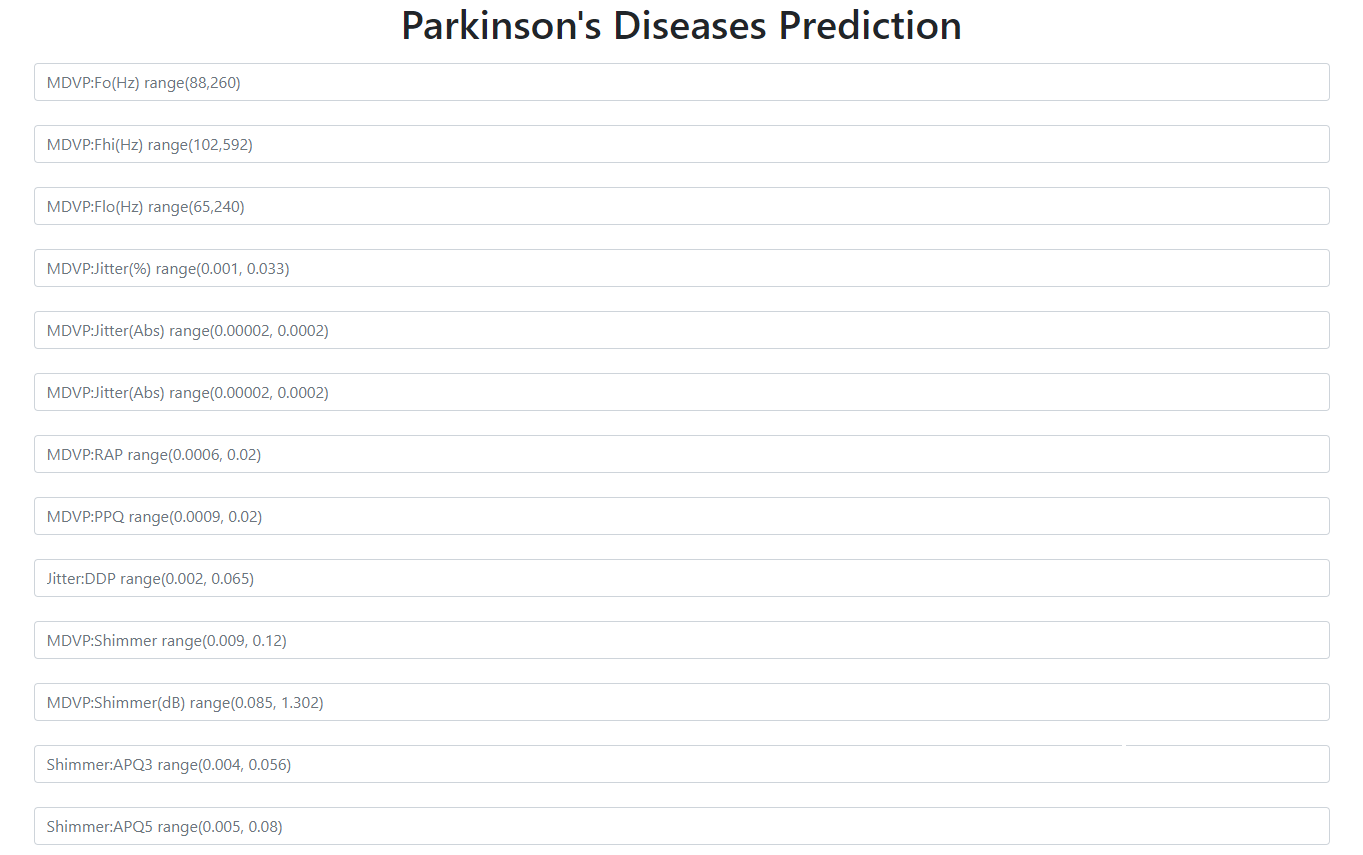
        return render\_template('index.html')

if \_\_name\_\_ == "\_\_main\_\_":

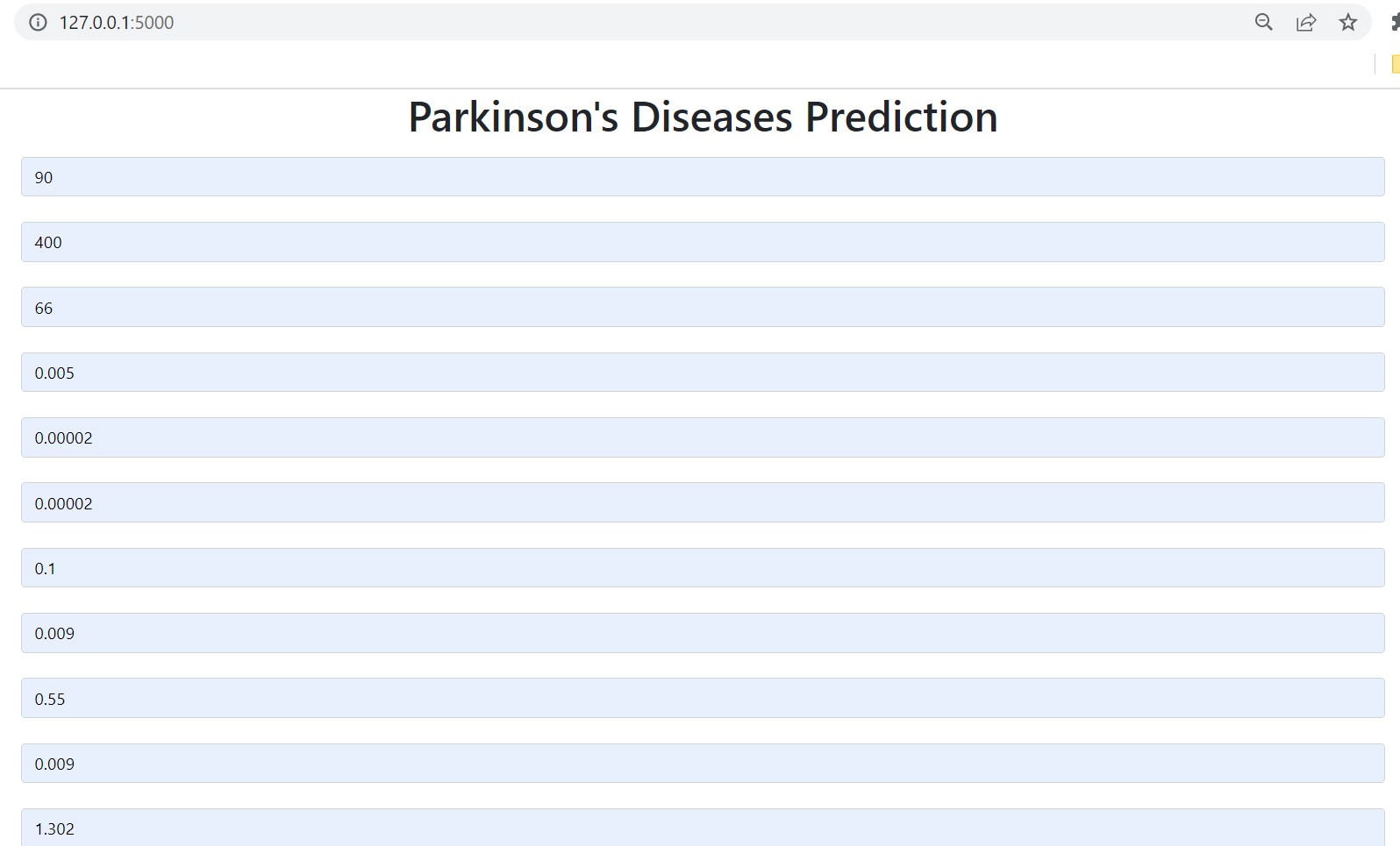
    #app.run(host='127.0.0.1', port=8001, debug=True)

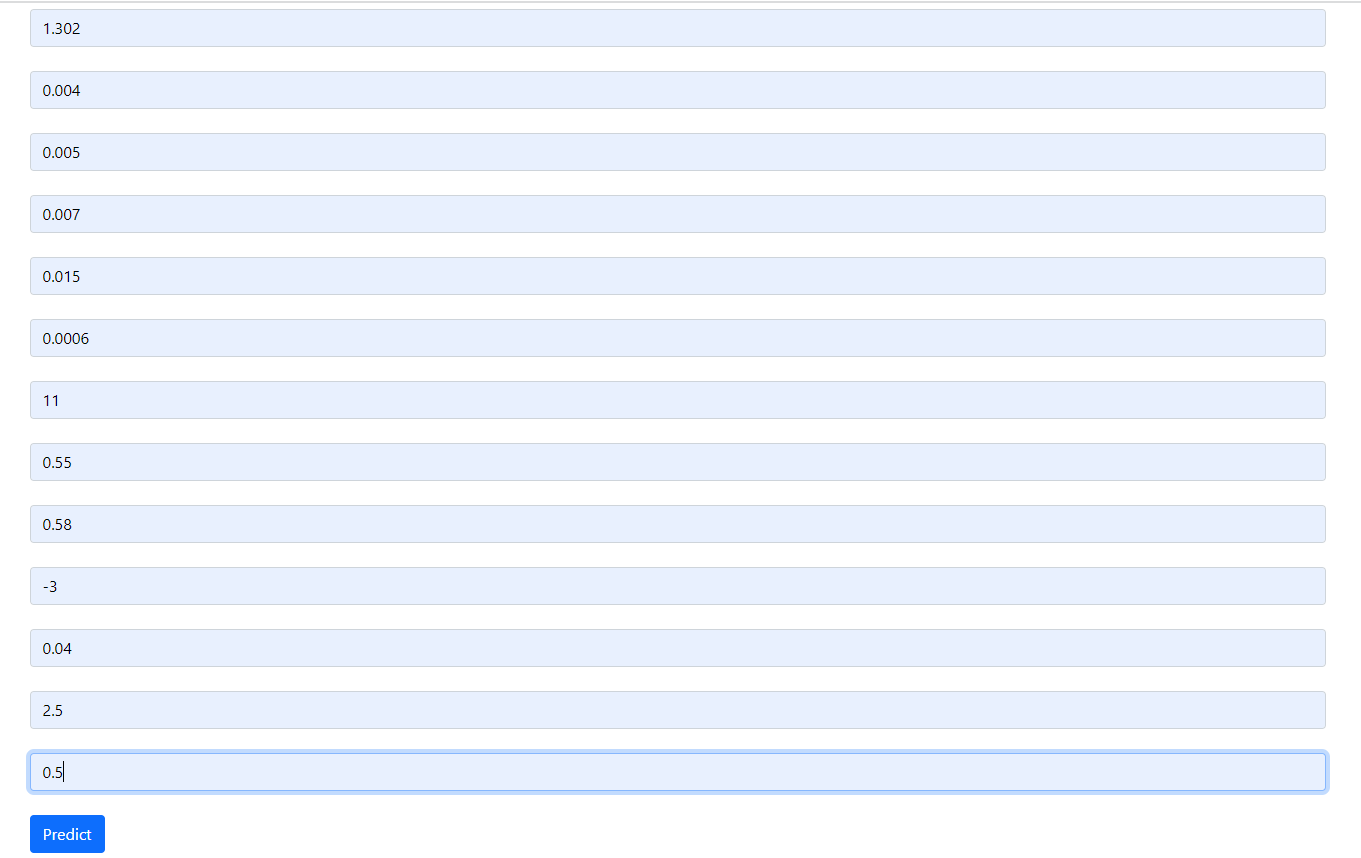
    app.run(debug=False) # running the app

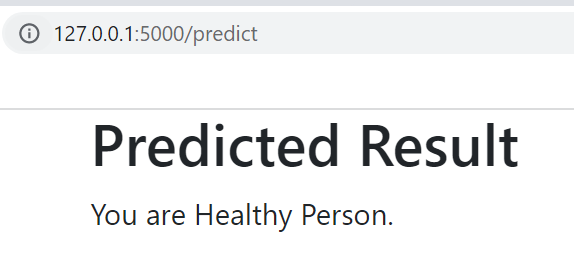
**OUTPUT:**











**CONCLUSION:**

From this practical, I have learned how to deploy machine learning applications using flask python framework.