Index(['age', 'bmi', 'children', 'female', 'male', 'no', 'yes', 'Diabetes',

'Heart disease', 'High blood pressure', 'None', 'Diabetes',

'Heart disease', 'High blood pressure', 'None', 'Frequently', 'Never',

'Occasionally', 'Rarely', 'Blue collar', 'Student', 'Unemployed',

'White collar', 'Basic', 'Premium', 'Standard'],

dtype='object')

// Send an HTTP request to the backend server

$.ajax({

    type: "POST",

    url: "http://127.0.0.1:5000/estimator",

    data: JSON.stringify(client\_data),

    contentType: "application/json",

    dataType: "json",

    success: function(response) {

      // Process the predicted results

    },

    error: function(xhr, status, error) {

      console.log(xhr.responseText);

    }

});

// Get estimate result

function processResults(response) {

    var prediction = response.estimator;

    // Display the prediction on the frontend

  }

method="POST" action="/estimator" role="form" , name="estimator"

def estimator():

    # Get the client data from "insurance\_estimator.html"

    data = json.loads(request.data)

    # transform data to feature to fit the model

    features = [[data["age"], data["bmi"], data["children"], data["female"], data["male"], data['No'], data["Yes"],  \

                 data['Diabetes'], data['Heart\_disease'], data['High\_blood\_pressure'], data['None'], \

                 data['Family\_Diabetes'], data['Family\_Heart\_disease'], data['Family\_High\_blood\_pressure'], \

                 data['Family\_None'], data['Frequently'], data['exercise\_Never'], data['Occasionally'], data['Rarely'], \

                 data['Blue\_collar'], data['Student'], data['Unemployed'], data['White\_collar'], \

                 data['Basic'], data['Premium'], data['Standard']]]

    # make predition by using model

    estimator = model.predict (features)

    return jsonify({'prediction': estimator[0]})

import joblib

# Load the machine learning model from the .joblib file

model = joblib.load('model\_LinearRegression.joblib')

# create route for model prediction

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

// Collect information

function showValue() {

    // Get the value of the input & select element from front end "insurance\_estimator.html"

    var client\_age = document.getElementById("age").value;

    var children\_no = document.getElementById("children").value

    var bmi = document.getElementById("bmi").value

    var gender = document.getElementById("gender").value

    var smoker = document.getElementById("smoker").value

    var medical\_history = document.getElementById("medical\_history").value

    var family\_medical\_history = document.getElementById("family\_medical\_history").value

    var exercise\_frequency = document.getElementById("exercise\_frequency").value

    var occupation = document.getElementById("occupation").value

    var coverage\_level = document.getElementById("coverage\_level").value

    // get dummy data for smoker

    var Yes;

    var No;

    if (smoker === "Yes") {

        Yes = 1;

        No = 0;

    } else {

        Yes = 0;

        No = 1;

    }

    // get dummy data for medical\_history & family\_medical\_history

    var Diabetes;

    var Heart\_disease;

    var High\_blood\_pressure;

    var None;

    var Family\_Diabetes;

    var Family\_Heart\_disease;

    var Family\_High\_blood\_pressure;

    var Family\_None;

    if (medical\_history === "Diabetes") {

        Diabetes = 1;

        Heart\_disease = 0;

        High\_blood\_pressure = 0;

        None = 0;

    } else if (medical\_history === "Heart disease"){

        Diabetes = 0;

        Heart\_disease = 1;

        High\_blood\_pressure = 0;

        None = 0;

    } else if (medical\_history === "High blood pressure") {

        Diabetes = 0;

        Heart\_disease = 0;

        High\_blood\_pressure = 1;

        None = 0;

    } else {

        Diabetes = 0;

        Heart\_disease = 0;

        High\_blood\_pressure = 0;

        None = 1;

    }

    if (family\_medical\_history === "Diabetes") {

        Family\_Diabetes = 1;

        Family\_Heart\_disease = 0;

        Family\_High\_blood\_pressure = 0;

        Family\_None = 0;

    } else if (family\_medical\_history === "Heart disease"){

        Family\_Diabetes = 0;

        Family\_Heart\_disease = 1;

        Family\_High\_blood\_pressure = 0;

        Family\_None = 0;

    } else if (family\_medical\_history === "High blood pressure") {

        Family\_Diabetes = 0;

        Family\_Heart\_disease = 0;

        Family\_High\_blood\_pressure = 1;

        Family\_None = 0;

    } else {

        Family\_Diabetes = 0;

        Family\_Heart\_disease = 0;

        Family\_High\_blood\_pressure = 0;

        Family\_None = 1;

    }

    // get dummy data for occupation

    var Blue\_collar;

    var White\_collar;

    var Student;

    var Unemployed;

    if (occupation === "Blue collar") {

        Blue\_collar = 1;

        White\_collar = 0;

        Student = 0;

        Unemployed = 0;

    } else if (occupation === "White collar"){

        Blue\_collar = 0;

        White\_collar = 1;

        Student = 0;

        Unemployed = 0;

    } else if (occupation === "Student") {

        Blue\_collar = 0;

        White\_collar = 0;

        Student = 1;

        Unemployed = 0;

    } else {

        Blue\_collar = 0;

        White\_collar = 0;

        Student = 0;

        Unemployed = 1;

    }

    //get dummy data for coverage level

    var Premium;

    var Standard;

    var Basic;

    if (coverage\_level === "Premium") {

        Premium = 1;

        Standard = 0;

        Basic = 0;

    } else if (coverage\_level === "Standard"){

        Premium = 0;

        Standard = 1;

        Basic = 0;

    } else {

        Premium = 0;

        Standard = 0;

        Basic = 1;

    }

    // list client data

    var client\_data = {

        "age": client\_age,

        "female": female,

        "male": male,

        "bmi": bmi,

        "children": children\_no,

        'Yes': Yes,

        'No': No,

        'Diabetes': Diabetes,

        'Heart\_disease': Heart\_disease,

        'High\_blood\_pressure': High\_blood\_pressure,

        'None': None,

        'Family\_Diabetes': Family\_Diabetes,

        'Family\_Heart\_disease': Family\_Heart\_disease,

        'Family\_High\_blood\_pressure': Family\_High\_blood\_pressure,

        'Family\_None': Family\_None,

        'Frequently': Frequently,

        'exercise\_Never': exercise\_Never,

        'Occasionally': Occasionally,

        'Rarely': Rarely,

        'exercise\_Never': exercise\_Never,

        'Blue\_collar': Blue\_collar,

        'Student': Student,

        'Unemployed': Unemployed,

        'White\_collar': White\_collar,

        'Basic': Basic,

        'Premium': Premium,

        'Standard': Standard

    }

    // Display the value in an alert box

    alert(JSON.stringify(client\_data));

    fetch('http://127.0.0.1:5000/estimator', {

        method: 'post',

        body: JSON.stringify(client\_data),

      }).then(function(response) {

        return response.json();

      }).then(function(data) {

        console.log("Data returned from python server", data)

      });

}

 <script src=".\static\js\app\_estimator.js"></script>

<https://simpledev.io/lesson/create-branch-gh-desktop-1/#:~:text=To%20create%20a%20new%20branch,appears%20and%20click%20Create%20Branch>.

<https://pythonbasics.org/flask-tutorial-templates/>

<https://stackoverflow.com/questions/50863789/in-flask-how-can-i-modify-html-with-code-in-python>

Index([

       , ,

       , 'family\_medical\_history\_Diabetes',

       'family\_medical\_history\_Heart disease',

       'family\_medical\_history\_High blood pressure',

       'family\_medical\_history\_None',

       'coverage\_level\_Premium', 'coverage\_level\_Standard'],

      dtype='object')

      if exercise\_frequency == "Frequently":

          exercise\_frequency\_Frequently = 1

          exercise\_frequency\_Occasionally = 0

          exercise\_frequency\_Rarely = 0

          exercise\_frequency\_Never = 0

      elif exercise\_frequency == "Occasionally":

          exercise\_frequency\_Frequently = 0

          exercise\_frequency\_Occasionally = 1

          exercise\_frequency\_Rarely = 0

          exercise\_frequency\_Never = 0

      elif exercise\_frequency == "Rarely":

          exercise\_frequency\_Frequently = 0

          exercise\_frequency\_Occasionally = 0

          exercise\_frequency\_Rarely = 1

          exercise\_frequency\_Never = 0

      else:

          exercise\_frequency\_Frequently = 0

          exercise\_frequency\_Occasionally = 0

          exercise\_frequency\_Rarely = 0

          exercise\_frequency\_Never = 1