Date	01 November 2022
Team ID	PNT2022TMID37289
Project Name	Car resale value prediction

```
from flask import Flask, render_template, request, url_for, redirect
import os
import pandas as pd
import numpy as np
import flask
import pickle



app = Flask(__name__)


@app.route('/')
def home_page():
   return render_template('index.html')


@app.route('/input', methods = ["GET","POST"])
def input_page():

    return render_template('input.html')


@app.route('/dosubmit', methods = ["GET","POST"])
def dosubmit():
   if request.method == 'POST':
    model_index=0

    modelList=['golf', 'grand', 'fabia', '3er', '2_reihe', 'c_max', '3_reihe',
      'passat', 'navara', 'twingo', 'a_klasse', 'scirocco', '5er',
      'meriva', 'andere', 'c4', 'civic', 'e_klasse', 'one', 'fortwo',
      'clio', '1er', 'b_klasse', 'punto', 'a8', 'jetta', 'astra',
      'c_klasse', 'micra', 'vito', 'sprinter', 'escort', 'forester',
      'xc_reihe', 'fiesta', 'scenic', 'a1', 'transporter', 'focus', 'a4',
      'tt', 'a6', 'jazz', 'omega', 'polo', 'slk', '7er', 'combo', '80',
      '147', 'glk', 'z_reihe', 'sportage', 'sorento', 'ibiza', 'mustang',
      'eos', 'touran', 'getz', 'insignia', 'almera', 'megane', 'a3',
      'r19', 'mondeo', 'cordoba', 'colt', 'vectra', 'lupo', 'berlingo',
      'm_klasse', 'tiguan', '6_reihe', 'up', 'i_reihe', 'ceed', 'kangoo',
      '5_reihe', 'yeti', 'octavia', 'zafira', 'mii', 'rx_reihe', 'corsa',
      '6er', 'panda', 'beetle', 'rio', 'touareg', 'logan', 'caddy',
      'spider', 's_max', 'modus', 'a2', 'x_reihe', 'a5', 'galaxy', 'c3',
      'viano', 's_klasse', '1_reihe', 'sharan', 'avensis', 'sl',
      'roomster', 'q5', 'santa', 'leon', 'cooper', '4_reihe',
      'ptcruiser', 'clk', 'primera', 'espace', 'exeo', '159', 'transit',
      'juke', 'ka', 'v40', 'carisma', 'accord', 'corolla', 'phaeton',
      'boxster', 'verso', 'rav', 'kuga', 'qashqai', 'swift', 'picanto',
      'superb', 'stilo', 'alhambra', 'm_reihe', 'roadster', 'ypsilon',
      'galant', 'justy', 'impreza', '90', 'sirion', 'signum',
```

```
        'crossfire', 'duster', 'v50', 'mx_reihe', 'discovery', 'c_reihe',
        'v_klasse', 'yaris', 'c5', 'aygo', 'cc', 'carnival', 'fusion',
        'bora', 'agila', '911', 'cl', 'tigra', '156', '300c', '500', '100',
        'q3', 'cr_reihe', 'spark', 'x_type', 'ducato', 's_type', 'x_trail',
        'toledo', 'altea', 'voyager', 'matiz', 'v70', 'bravo',
        'range_rover', 'tucson', 'fox', 'q7', 'c1', 'kadett', 'jimny',
        'cx_reihe', 'cayenne', 'wrangler', 'lybra', 'range_rover_sport',
        'lancer', 'freelander', 'captiva', 'laguna', 'c2',
        'range_rover_evoque', 'sandero', 'note', 'antara', '900',
        'defender', 'clubman', 'forfour', 'legacy', 'pajero', 'auris',
        'niva', 's60', 'nubira', 'vivaro', 'g_klasse', 'cherokee', 'lodgy',
        'lanos', '850', 'calibra', 'serie_2', 'charade', 'croma', 'cuore',
        'citigo', 'outlander', 'gl', 'doblo', 'musa', 'amarok', 'arosa',
        '9000', 'kalos', 'v60', 'aveo', '200', '145', 'b_max', 'delta',
        'rangerover', 'materia', 'terios', 'move', 'kalina', 'i3',
        'kaefer', 'kappa', 'samara', 'discovery_sport', 'seicento']

    model_type = request.form['model_type']
    for i in modelList:
      if(i == model_type):
        model_index = modelList.index(i)



    pincode = int(request.form['pin_code'])
    abtest = int(request.form['abtest'])
    vehicletype = int(request.form.get('vehicle'))
    regyear = int(request.form['reg_year'])
    gearbox = int(request.form['gearBox'])
    powerps = float(request.form['power_ps'])
    kms = float(request.form['kilometer_driven'])
    regmonth = int(request.form.get('reg_month'))
    fuelType = int (request.form.get('fuel'))
    brand = int (request.form.get('brand'))
    damage = int (request.form[ 'carDamage'])
    to_predict_list =
[[abtest,vehicletype,regyear,gearbox,powerps,model_index,kms,regmonth,fuelType,brand,damage,pin
code]]
    loaded_model = pickle.load(open("finalmodel.pkl","rb"))
    result = loaded_model.predict(to_predict_list)
    ans = round(result[0],2)
    prediction = str(ans)


   return redirect(url_for('output_page',output_res = prediction))



@app.route('/output' ,methods = ["GET","POST"])
def output_page():
   output_res  = request.args.get('output_res')
   return render_template('output.html', prediction = output_res)
```

```
if __name__ == '__main__':
    app.run()
```