```python
import pandas as pd
import numpy as np
from flask import Flask,render_template,Response,request
import pickle
from sklearn.preprocessing import LabelEncoder
import pickle

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "OBzxNKUthSA-ldV5xlJTAbOTwk6KHtbWq20PA339A1bI"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}


app = Flask(__name__)
filename = 'resale_model.sav'
model_rand = pickle.load(open(filename,'rb'))

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

@app.route('/predict')
def predict():
        return render_template('resalepredict.html')

@app.route('/y_predict',methods=['GET','POST'])
def y_predict():
```

```
    regyear = int(request.form['regyear'])
    powerps = float(request.form['powerps'])
    kms = float(request.form['kms'])
    regmonth = int(request.form.get('regmonth'))
    gearbox = request.form['gearbox']
    damage = request.form['damaged']
    model  = request.form.get('model_type')
    brand = request.form.get('brand')
    fuelType = request.form.get('fuel')
    vehicletype= request.form.get('vehicletype')
    new_row = {'yearOfRegistration':regyear,'powerPS':powerps,'kilometer':kms,'monthOfRegistration':regmonth,'gearbox':gearbox,'notRepairedDamage':damage,'model':model,'brand'

    print(new_row)
    new_df = pd.DataFrame(columns=['vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','brand','notRepairedDamage'])
    new_df = new_df.append(new_row,ignore_index=True)
    labels = ['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']
    mapper = {}
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].classes_ = np.load(str('classes'+i+'.npy'),allow_pickle=True)
        tr = mapper[i].fit_transform(new_df[i])
        new_df.loc[:,i+'_Labels'] = pd.Series(tr,index=new_df.index)
    labeled = new_df[ ['yearOfRegistration','powerPS','kilometer','monthOfRegistration'] + [x+"_Labels" for x in labels]]
    X = labeled.values
    print(X)
        #y_prediction = model_rand.predict(X)
        #print(y_prediction)
    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"fields": ['f0','f1','f2','f3','f4','f5','f6','f7','f8','f9'], "values":X.tolist()}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/d82d5a31-9678-4e51-b9fd-7a638e0e991e/predictions?version=2022-11-20', json=payload_sc
    print("Scoring response")
    predictions = response_scoring.json()
```

```
    output =  predictions['predictions'][0]['values'][0][0]
    print(output)
    return render_template('resalepredict.html',ypred="The resale value predicted is $  "+str(output))

if __name__ == '__main__':
        app.run(host='Localhost',debug=True,threaded=False)
```