Car Resale Value Prediction

Date	**8 November 2022**
Team ID	**PNT2022TMID07152**
Project Name	**Project - Car Resale Value prediction**

Application Building:

Build the Python Flask App

#Importing required libraries

Import pandas as pd import

Numpy as pd

from flask import Flask, render_template, Response, request import

pickle

from sklearn.preprocessing import LabelEncoder import

pickle

#Load the model and initialize Flask app

app=Flask(__name__) filename='resale_model.sav'

model_rand=pickle.load(open(filename,'rb'))

#Configure app.py to fetch the parameter values from the ui,and return the prediction

@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['reg year'])

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

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

regmonth int(request.form.get('regmonth'))

gearbox = request.form['gearbox']

damage request.form['dam']

model request.form.get('modeltype') brand=
request.form.get('brand')

fuelType = request.form.get('fuel') vehicle type=
request.form.get('vehicle type')

new_row("yearOfRegistration':reg year, 'powerPS':power ps,
'kilo-meter':kms,

monthofRegistration': regmonth, gearbox gearbox,
'notRepairedDamage': damage,

'model':model, 'brand':brand, 'fuelType': fuelType,

'vehicleType': vehicle type)

```
print(new row)

new_df = pd.DataFrame(columns =['vehicleType',
'yearOfRegistration', 'gearbox'', 'powerPS', 'model',

'kilo-meter', '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'+inpy'))

tr= mapper[i].fit_transform(new_df[i])

new_df.loc[:, i +'_labels'] = pd.Series (tr, index-new_df.index)

label = new_df[ ['yearOfRegistration' ,"powerPS' 'kilo-meter'
"monthOfRegistration']+[x+'_labels' for x in

labels]]

X=labeled.values print(X)

y_prediction=model.rand.

predict(X)

print(y_prediction)

return render_template('resalespredict.html',ypred = 'The resale
value predicted is

{:.2f}$'.format(y_prediction[0]))
```

Run the app

```
If __name__ == '__main':

app.run(host='localhost',debug = True, threaded = False)
```