Date	12 November 2022
Team ID	PNT2022TMID01544
Project Name	PROJECT-CAR RESALES VALUE PREDICTION
Maximum Marks	2 Marks

FLASK:

Flask is an open source web framework which offers us with the tools, library resources needed to create a web application. Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

PANDAS:

Data analysis and related manipulation of tabular data in Data frames are the major uses of Pandas. Data can be imported into Pandas from a variety of file types, including Microsoft Excel, JSON, Parquet, SQL database tables, and comma-separated values. Data wrangling, data cleaning, and other operations like merging, restructuring, and choosing are all possible with Pandas. Many of the R programming language's established functionality for working with data frames were brought into Python with the introduction of pandas. The NumPy library, which is focused on effectively working with arrays rather than the characteristics of working with Data frames, is the foundation upon which the Panda library is constructed.

NumPy:

A collection of the multidimensional matrix that facilitates complex mathematical operations. NumPy can be used to execute operations on arrays that are related to mathematics, such as algebraic, statistical, and trigonometric patterns. The image is transformed into a matrix. The Convolutional Neural Network is utilized to understand and analyze the image in its matrix form . The image's annotations then adopted a NumPy array style. Finally, the dataset contains the precise labels for each image. On this, SciPy was also developed. It provides more noteworthy execution that utilizes NumPy arrays and is required for various logical and engineering tasks.

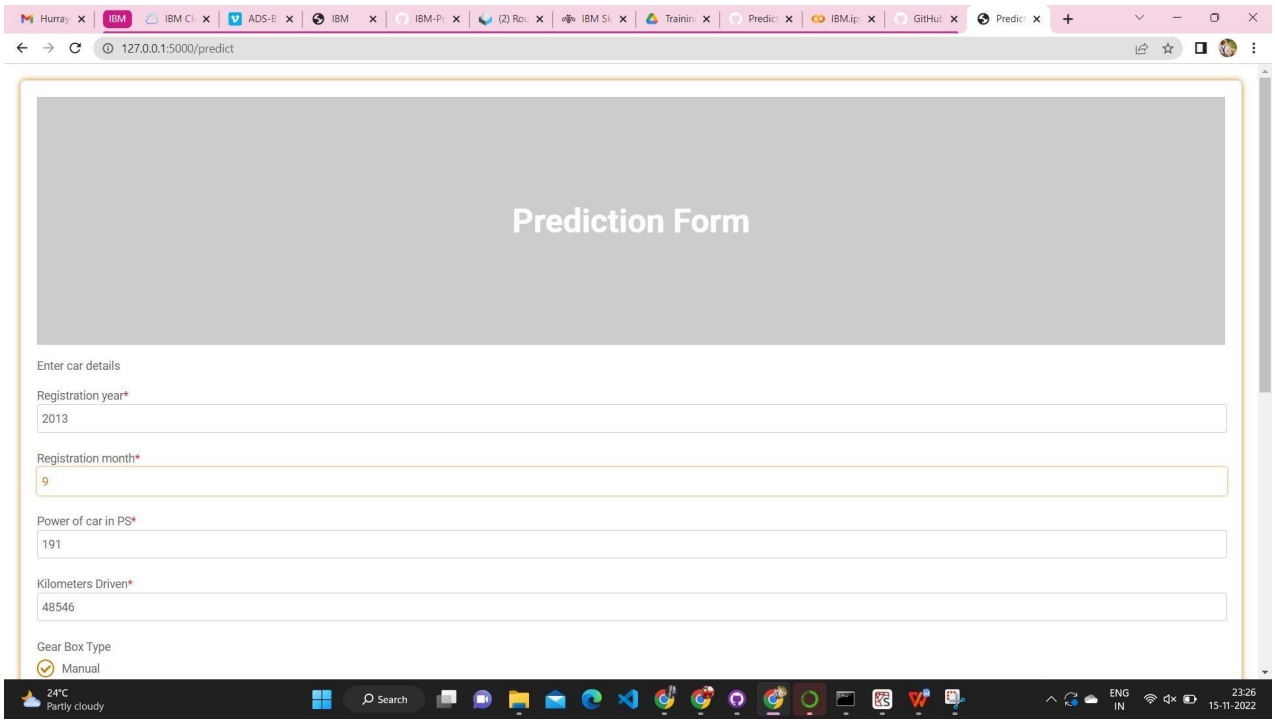
app.py

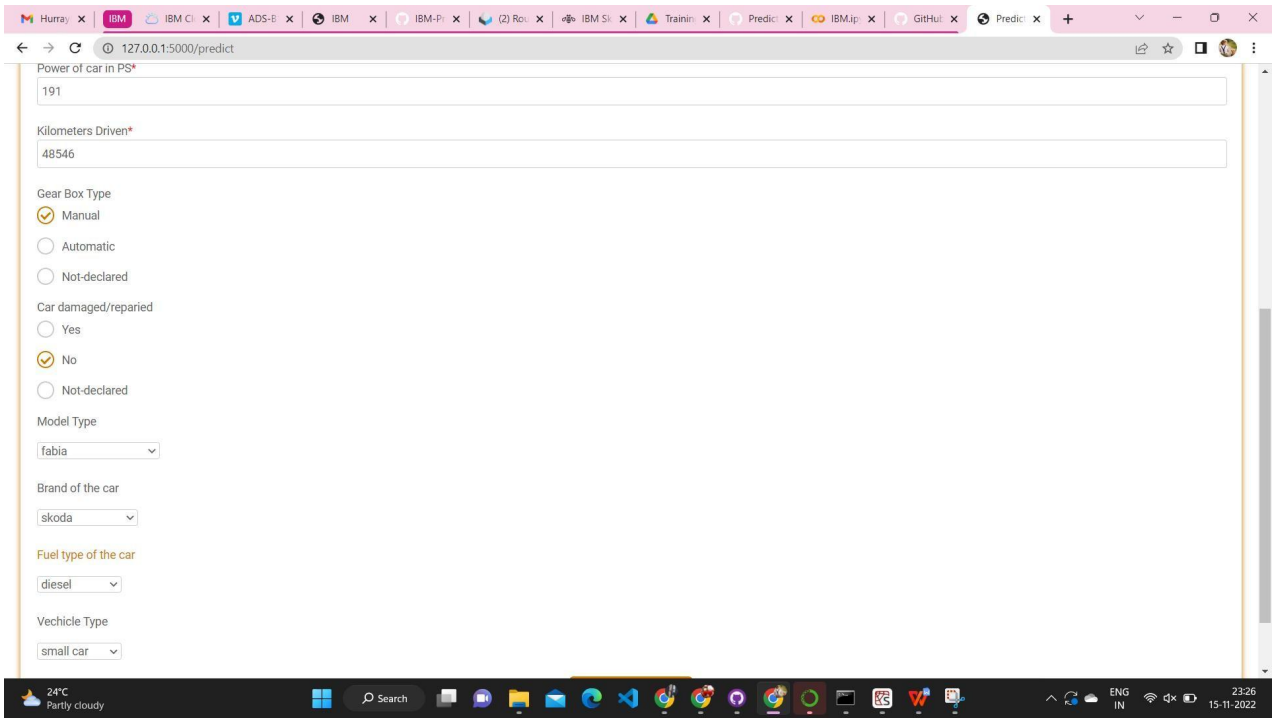
```
import pandas
from flask import Flask, render_template, request

import pandas as pd
import numpy as np


from sklearn.preprocessing import LabelEncoder

app =Flask(__name__)
import pickle
filename = 'resale_model.sav'
model_rand= pickle.load(open(filename, 'rb'))
@app.route('/')
def index():
    return render_template('home.html')

@app.route('/predict')
def predict():
    return render_template('predict.html')
@app.route('/y_predict', methods= ['POST'])
def y_predict():
    regyear =  int(request.form['Registrationyear'])
    powerps = int(request.form['PowerofcarinPS'])
    kms =int(request.form['KilometersDriven'])
    regmonth =int(request.form.get('Registrationmonth'))
```

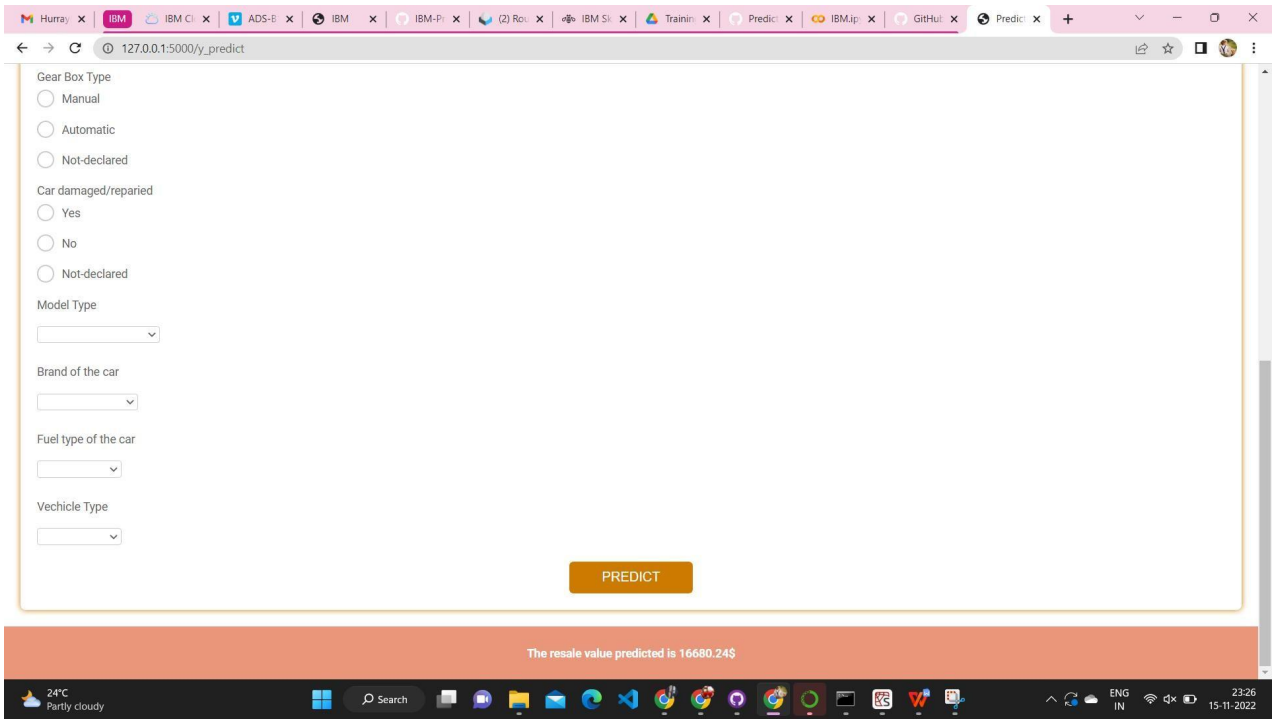
```
    gearbox =(request.form['Geartype'])
    damage =request.form['cd']
    model = request.form.get('model')
    brand =request.form.get('brand')
    fuelType =request.form.get('fueltype')
    vehicletype = request.form.get('vechicletype')
    row = {'vehicleType': vehicletype,'yearOfRegistration':regyear,
          'gearbox': gearbox,'powerPS': powerps, 'model':model,'kilometer': kms,
          'monthOfRegistration': regmonth,'fuelType': fuelType,
          'brand': brand,'notRepairedDamage': damage}
    print(row)
    new_row = pd.DataFrame([row])
    new_df = pd.DataFrame(columns = ['vehicleType', 'yearOfRegistration', 'gearbox',
                        'powerPS','model', 'kilometer', 'monthOfRegistration', 'fuelType',
                        'brand','notRepairedDamage' ])
    new_df=pd.concat([new_df,new_row], ignore_index = True)
    new_df['monthOfRegistration']= new_df['monthOfRegistration'].astype(int)
    labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
    mapper = {}
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].fit(new_df[i])
        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)
    df_ev = np.exp(y_prediction)
    print("df_ev: {} ".format(df_ev))
    return render_template('predict.html',y = 'The resale value predicted is {:.2f}$'.format(df_ev[0]))


if__name__=='__main__':
    app.run(debug= False)
```

OUTPUT:




127.0.0.1:5000/predict
Power of car in PS*
191
Kilometers Driven*
48546
Gear Box Type
Manual
Automatic
Not-declared
Car damaged/reparied
Yes
No
Not-declared
Model Type
fabia
Brand of the car
skoda
Fuel type of the car
diesel
Vechicle Type
small car


127.0.0.1:5000/y_predict
Gear Box Type
Manual
Automatic
Not-declared
Car damaged/reparied
Yes
No
Not-declared
Model Type
Brand of the car
Fuel type of the car
Vechicle Type
PREDICT
The resale value predicted is 16680.24$