

APPLICATION BUILDING

EXECUTE AND TEST YOUR MODEL

1) Execution

```
Dropping unnecessary columns

In 18: 1 df.drop(['name', 'abtest', 'dateCrawled', 'nrOfPictures', 'lastSeen', 'postalCode', 'dateCreated'], axis='columns', inplace=True)

In 19: 1 df

Out 19: 1
   price  vehicleType  yearOfRegistration  gearbox  powerPS  model  kilometer  monthOfRegistration
1  18300         coupe             2011    manuell      190    NaN      125000              5
2   9800         suv             2004    automatik     163    grand     125000              8
3   1500    kleinwagen             2001    manuell      75    golf     150000              6
4   3600    kleinwagen             2008    manuell      69    fabia      90000              7
5    650    limousine             1995    manuell     102     3er     150000             10
6   2200         cabrio             2004    manuell     109    2_reihe     150000              8
8  14500         bus             2014    manuell     125    c_max      30000              8
9    999    kleinwagen             1998    manuell     101    golf     150000              0

309166 rows x 11 columns  Open in new tab

In 20: 1 new_df = df.copy()

In 21: 1 new_df.columns

Out 21: 1
Index(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
      'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand',
      'notRepairedDamage'],
      dtype='object')
```

Model Building

```
Project: IBM-Project C:\Users\ASUS\Desktop\IBM-Project
> DATASET
> Model Building
> .ipynb_checkpoints
  classesbrand.npy
  classesfuelType.npy
  classesgearbox.npy
  classesmodel.npy
  classesnotRepairedDamage.npy

1 from flask import Flask, render_template, Response, request
2 import pickle
3 from sklearn.preprocessing import LabelEncoder
4 import numpy as np
5 import pandas as pd
6 app=Flask(__name__)
7 filename= 'C:\\Users\\ASUS\\Desktop\\IBM-Project\\Model_Building\\crvpy.sav'

Terminal: Local x Local (2) + v
(IBM-Project) C:\Users\ASUS\Desktop\IBM-Project>flask --app apps.py
Usage: flask [OPTIONS] COMMAND [ARGS]...
Try 'flask --help' for help.

Error: Missing command.

(IBM-Project) C:\Users\ASUS\Desktop\IBM-Project>flask --app apps.py --debug run
* Serving Flask app 'apps.py'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 116-429-565
```

Flask execution

2) MODEL TESTING



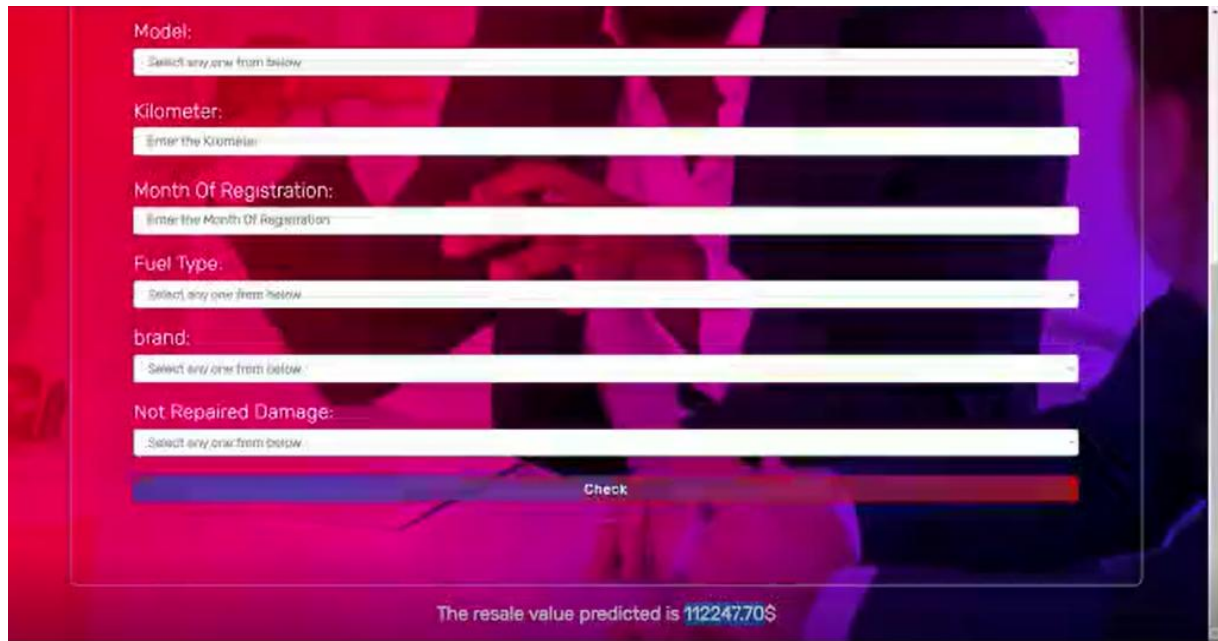
The screenshot shows the 'Home Page' of a 'Car Resale Value Prediction' application. The page has a dark blue header with the title 'Car Resale Value Prediction' in white. Below the header, there is a light blue box with the text 'Please fill in your details here:'. The form contains several input fields: 'Vehicle Type:' with a dropdown menu showing 'Suv'; 'Year Of Registration:' with a dropdown menu showing 'Select the Year Of Registration'; 'Gearbox:' with a dropdown menu showing 'Select any one from below'; 'Power PS:' with a text input field showing 'Enter the Power PS'; 'Model:' with a dropdown menu showing 'Select any one from below'; 'Kilometer:' with a text input field showing 'Enter the Kilometer'; and 'Month Of Registration:' with a dropdown menu. The background of the page is a blurred image of a man in a suit.

Home Page



The screenshot shows the 'Data Entry Page' of the 'Car Resale Value Prediction' application. The page has a dark blue header with the title 'Car Resale Value Prediction' in white. Below the header, there is a light blue box with the text 'Please fill in your details here:'. The form contains several input fields: 'Vehicle Type:' with a dropdown menu showing 'Suv'; 'Year Of Registration:' with a dropdown menu showing '2020'; 'Gearbox:' with a dropdown menu showing 'Automatic'; 'Power PS:' with a text input field showing '2400'; 'Model:' with a dropdown menu showing 'Civic'; 'Kilometer:' with a text input field showing '1000'; 'Month Of Registration:' with a dropdown menu showing '4'; and 'Fuel Type:' with a dropdown menu showing 'Petrol'. The background of the page is a blurred image of a man in a suit.

Data Entry Page



The image shows a web application for calculating car resale value. It features a series of input fields for 'Model', 'Kilometer', 'Month Of Registration', 'Fuel Type', 'brand', and 'Not Repaired Damage'. Each field has a placeholder text indicating how to use it (e.g., 'Select any one from below' for dropdowns, 'Enter the Kilometer' for text input). A 'Check' button is positioned below the inputs. At the bottom, a message states 'The resale value predicted is 112247.70\$'.

Model:
Select any one from below

Kilometer:
Enter the Kilometer

Month Of Registration:
Enter the Month Of Registration

Fuel Type:
Select any one from below

brand:
Select any one from below

Not Repaired Damage:
Select any one from below

Check

The resale value predicted is 112247.70\$

Car Resale Value Display Page