DOAA CA1

CARVALUIFY

YOUR ONE-STOP CAR PRICE PREDICTOR WEBSITE

Name : Dario Prawara Teh Wei Rong

Class : DAAA / FT / 2B / 04

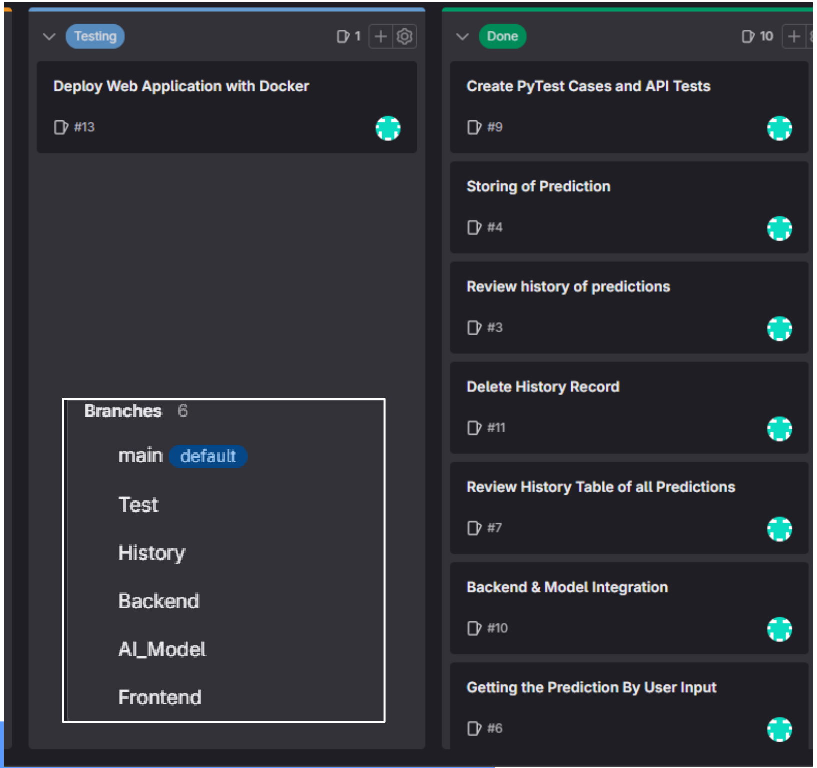
Admin Number : 2201858

PROJECT OBJECTIVE

Using **regression modelling** for car price prediction, **predict car prices,** given a set of input car feature data.

For our **‘100,000 UK Cars’** dataset, it contained extensive data on car model, car registration year, mileage, tax and more.

DEVOPS PROCESS



Testing 1

Deploy Web Application with Docker
#13

Branches 6

main default

Test

History

Backend

AI_Model

Frontend

Done 10

Create PyTest Cases and API Tests
#9

Storing of Prediction
#4

Review history of predictions
#3

Delete History Record
#11

Review History Table of all Predictions
#7

Backend & Model Integration
#10

Getting the Prediction By User Input
#6

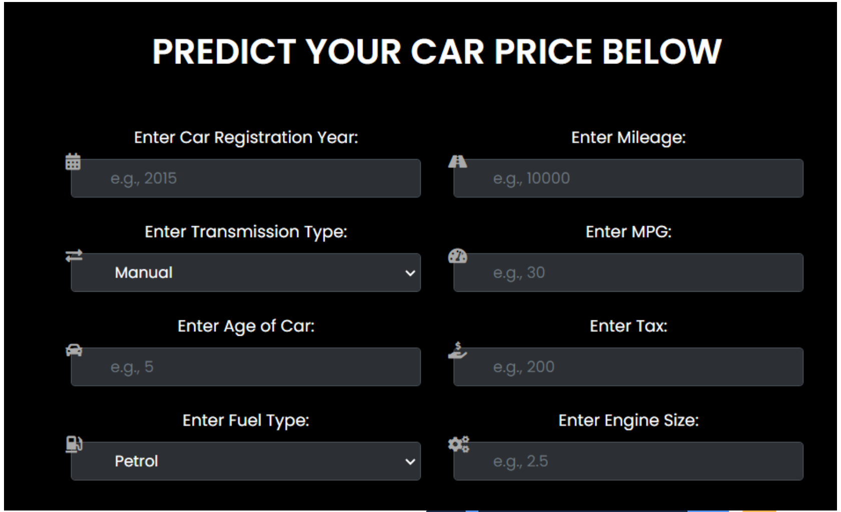
SETTING UP GITLAB AND SCRUM BOARD

- Managed a **scrum board** to enable us to keep track of our web application development.
- The scrum board contained 4 key labels, ‘To-Do’, ‘In Progress’, ‘Testing’, and ‘Done’.
- Created **6** branches for frontend, backend, model building and testing for us to make changes to the folder in a controlled environment.

PRESENTING : CARVALUIFY

CARVALUIFY

CarValuify is a **simple-to-use web application** that enables you to predict your car prices & valuations through inputting your car details. Simply enter the necessary fields, and our AI algorithm will work out the rest!



PREDICT YOUR CAR PRICE BELOW

Enter Car Registration Year: e.g., 2015

Enter Mileage: e.g., 10000

Enter Transmission Type: Manual

Enter MPG: e.g., 30

Enter Age of Car: e.g., 5

Enter Tax: e.g., 200

Enter Fuel Type: Petrol

Enter Engine Size: e.g., 2.5

GENERAL INFORMATION

Authentication required for non-public pages.

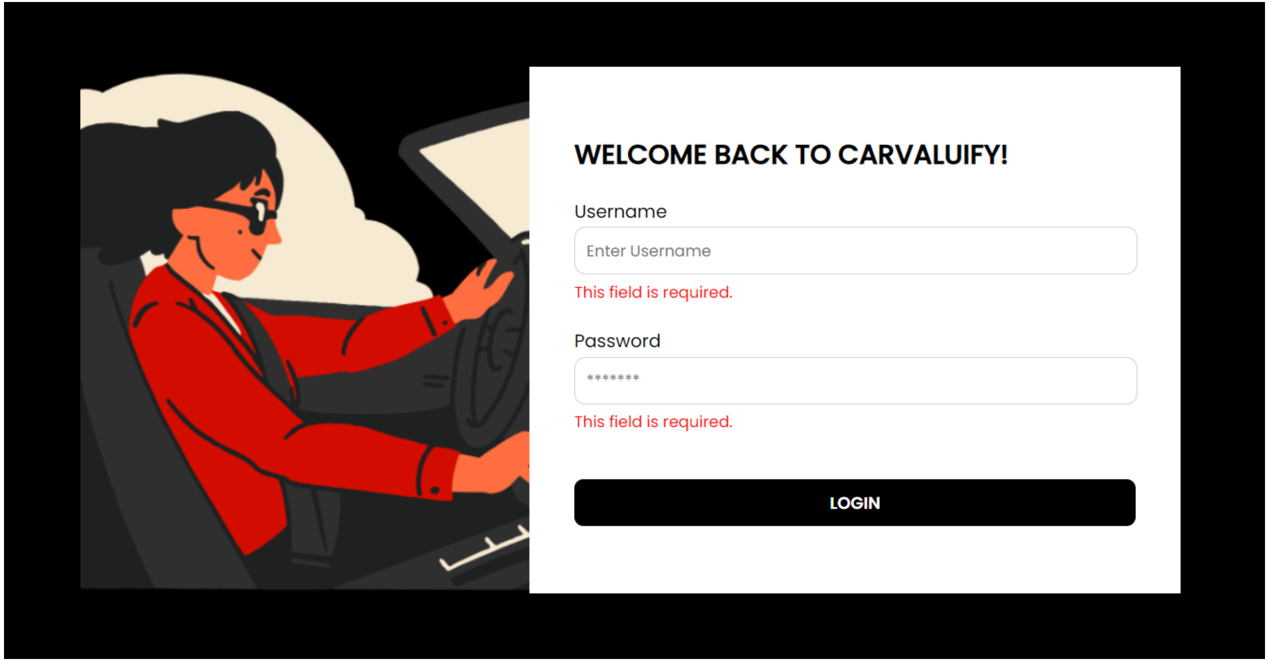
Bypassing the authentication by typing in direct link **is addressed and is prevented**.

Username : student
Password : student

Responsive + Flexible Website Design using Bootstrap 4 and TailwindCSS.

Webpage is viewable on multiple devices and is customizable to the screen size.

CARVALUIFY | CAR PRICE PREDICTOR



WELCOME BACK TO CARVALUIFY!

Username

Enter Username

This field is required.

Password

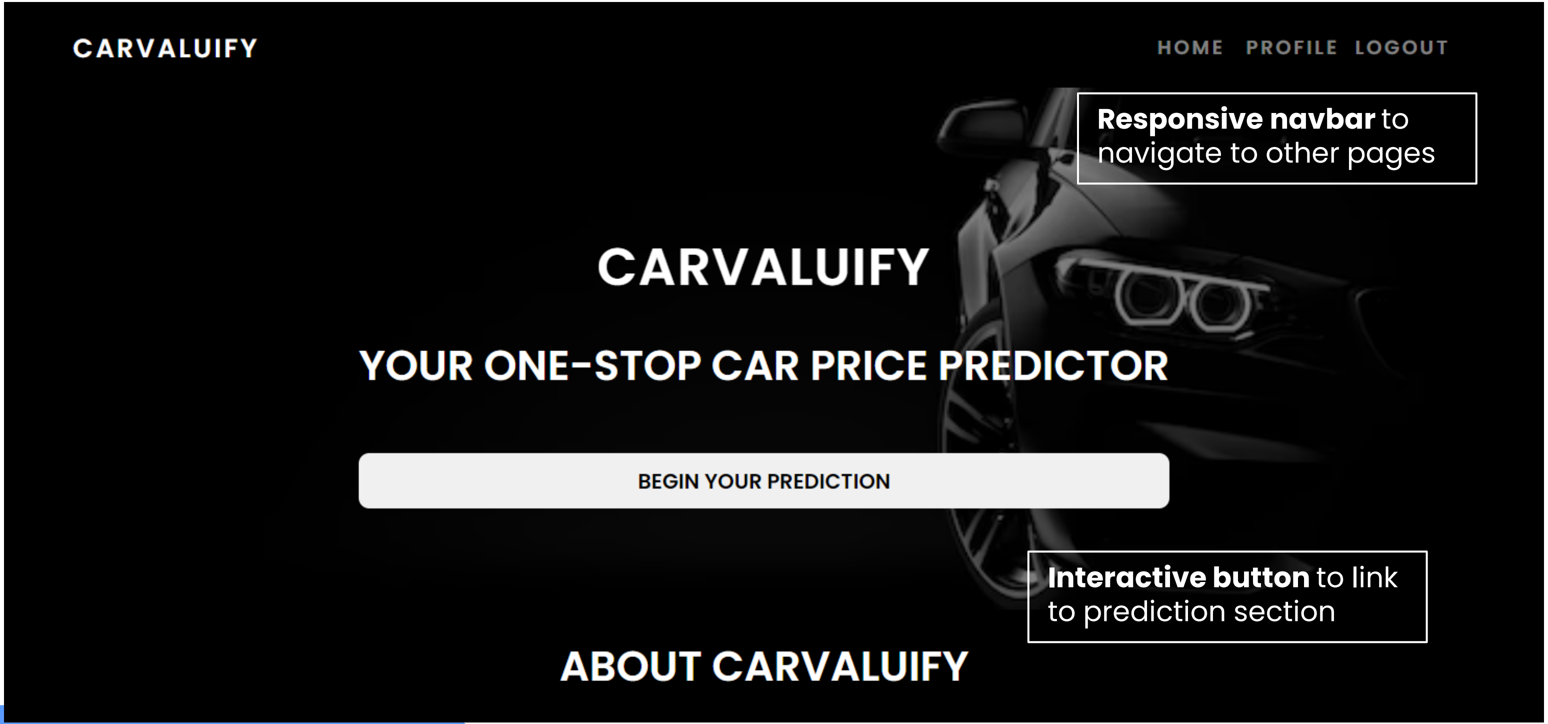
****

This field is required.

Login Page : Prompt users to enter the username and password before proceeding

Errors & validation checks are displayed using Flask's **flash()** function.

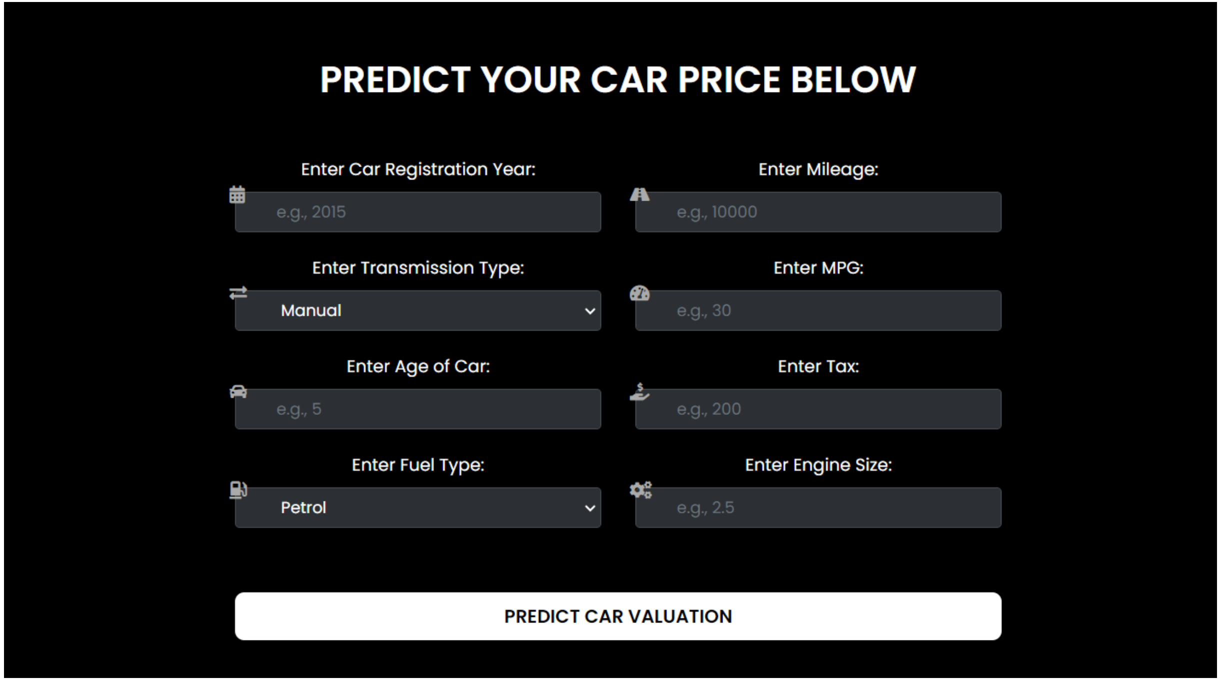
CARVALUIFY | CAR PRICE PREDICTOR



Responsive navbar to navigate to other pages

Interactive button to link to prediction section

CARVALUIFY | CAR PRICE PREDICTOR



PREDICT YOUR CAR PRICE BELOW

Enter Car Registration Year: e.g., 2015

Enter Mileage: e.g., 10000

Enter Transmission Type: Manual

Enter MPG: e.g., 30

Enter Age of Car: e.g., 5

Enter Tax: e.g., 200

Enter Fuel Type: Petrol

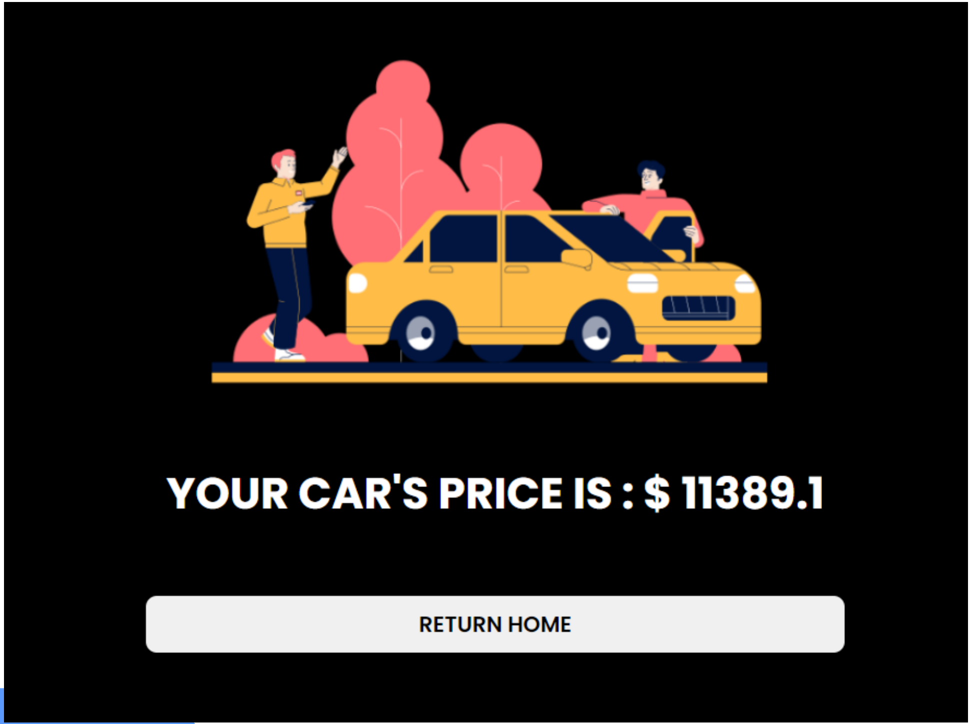
Enter Engine Size: e.g., 2.5

Added **attractive icons** for user engagement

Dropdown box options for easy selection for our user

CARVALUIFY | CAR PRICE PREDICTOR

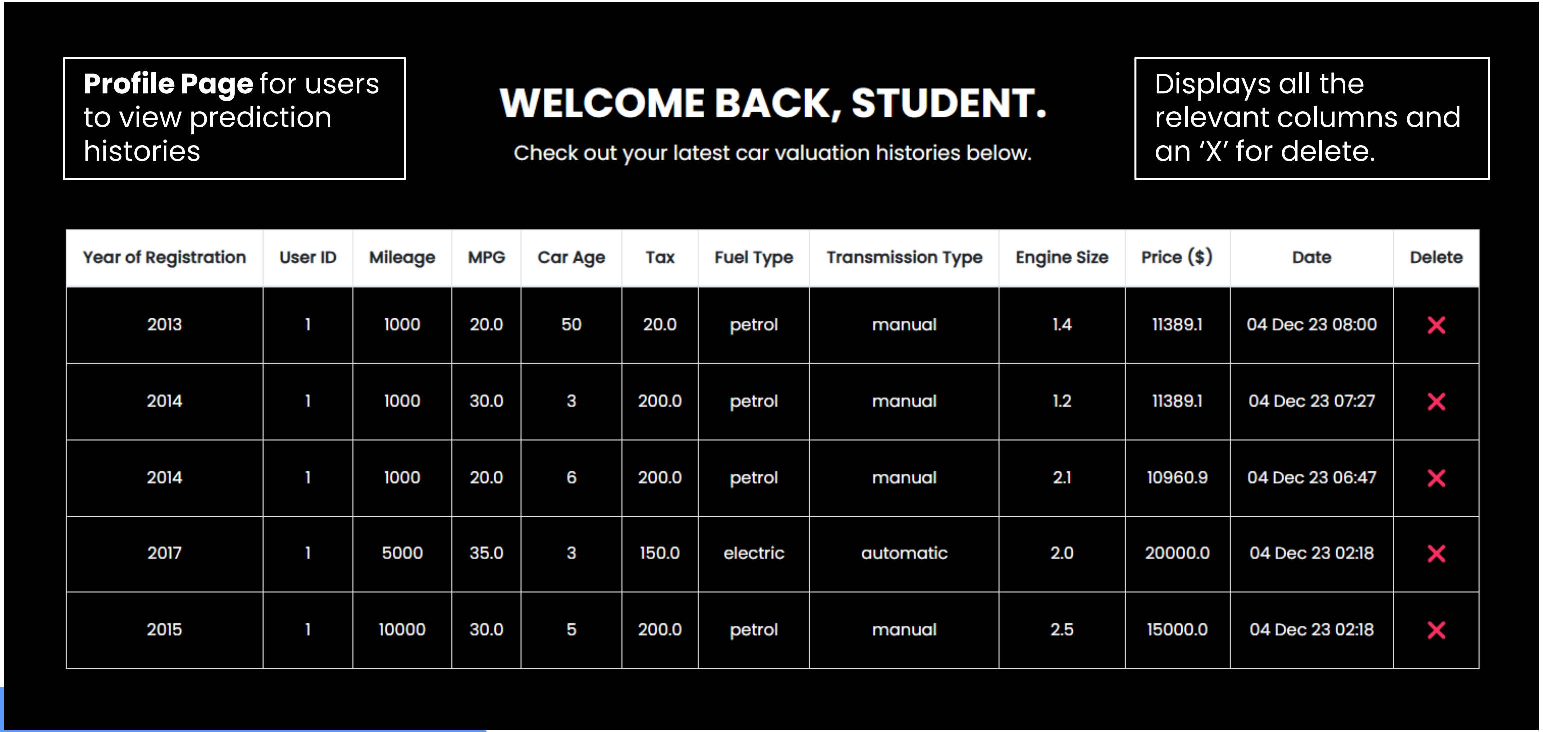
Attractive graphic for viewing of the predicted price



YOUR CAR'S PRICE IS : $ 11389.1

Simple 'Return Home' button to easily return to the home page

CARVALUIFY | CAR PRICE PREDICTOR



Profile Page for users to view prediction histories

WELCOME BACK, STUDENT.

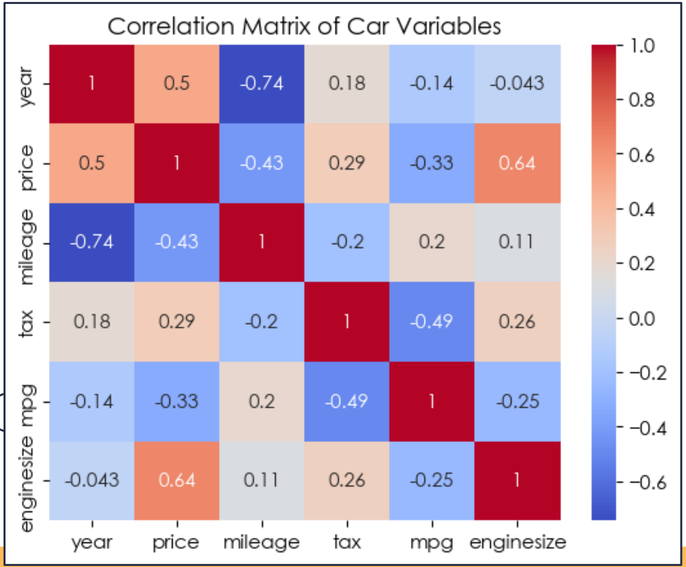
Check out your latest car valuation histories below.

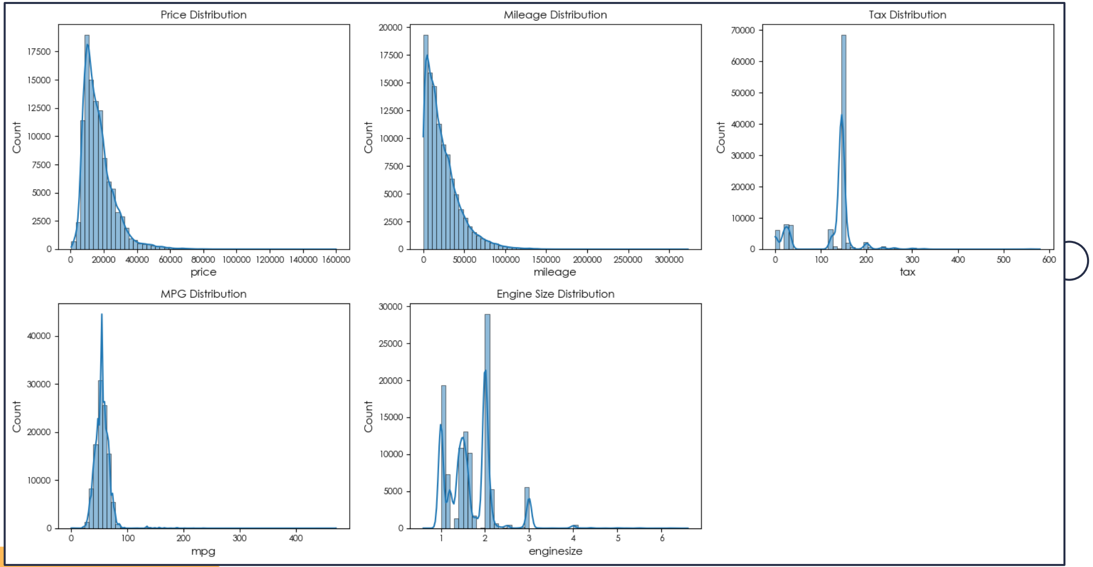
Displays all the relevant columns and an 'X' for delete.

Year of Registration	User ID	Mileage	MPG	Car Age	Tax	Fuel Type	Transmission Type	Engine Size	Price ($)	Date	Delete
2013	1	1000	20.0	50	20.0	petrol	manual	1.4	11389.1	04 Dec 23 08:00	✕
2014	1	1000	30.0	3	200.0	petrol	manual	1.2	11389.1	04 Dec 23 07:27	✕
2014	1	1000	20.0	6	200.0	petrol	manual	2.1	10960.9	04 Dec 23 06:47	✕
2017	1	5000	35.0	3	150.0	electric	automatic	2.0	20000.0	04 Dec 23 02:18	✕
2015	1	10000	30.0	5	200.0	petrol	manual	2.5	15000.0	04 Dec 23 02:18	✕

MACHINE LEARNING REGRESSOR

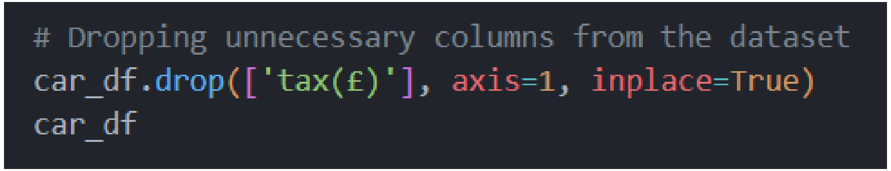
- First, conducted extensive EDA on the dataset to discover insights and patterns in data.
- Found **moderate to high correlation** between most of the car variables in dataset.
- Most variables like Price and Mileage are **slightly skewed** (imbalanced distribution).





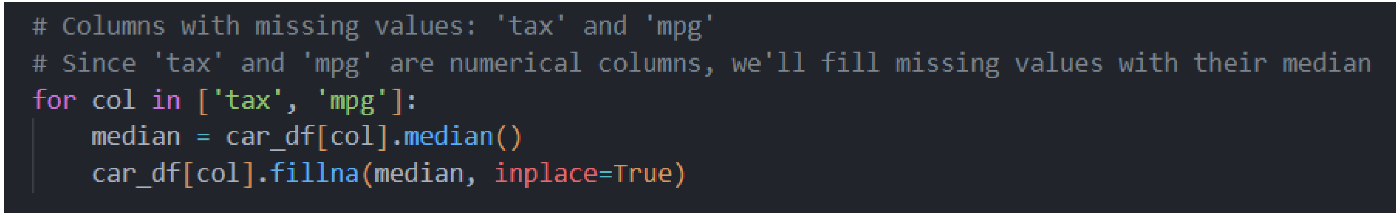
MACHINE LEARNING REGRESSOR

- Dropped unnecessary columns not useful for the analysis.



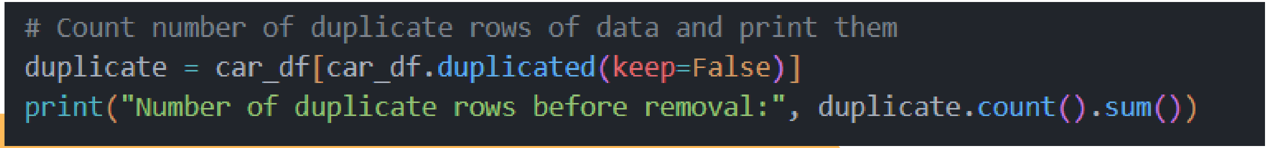
```
# Dropping unnecessary columns from the dataset
car_df.drop(['tax(£)'], axis=1, inplace=True)
car_df
```

- Imputed missing values with **MEDIAN** to remove NULLS and prevent loss of data rows.



```
# Columns with missing values: 'tax' and 'mpg'
# Since 'tax' and 'mpg' are numerical columns, we'll fill missing values with their median
for col in ['tax', 'mpg']:
    median = car_df[col].median()
    car_df[col].fillna(median, inplace=True)
```

- Dropped **DUPLICATE** rows of data to prevent biasness to certain groups of values.



```
# Count number of duplicate rows of data and print them
duplicate = car_df[car_df.duplicated(keep=False)]
print("Number of duplicate rows before removal:", duplicate.count().sum())
```

MACHINE LEARNING REGRESSOR

Feature Engineering Performed

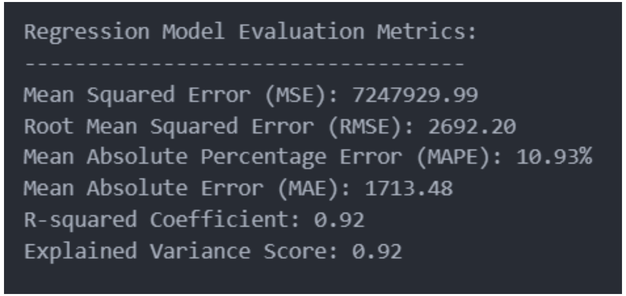
- Robust Scaling of Data
- Added new column ‘Car Age’ (Now – Registration Year)
- One-hot Encoded Labels
- Train-Test Split X and Y

Cross-Validation Performance

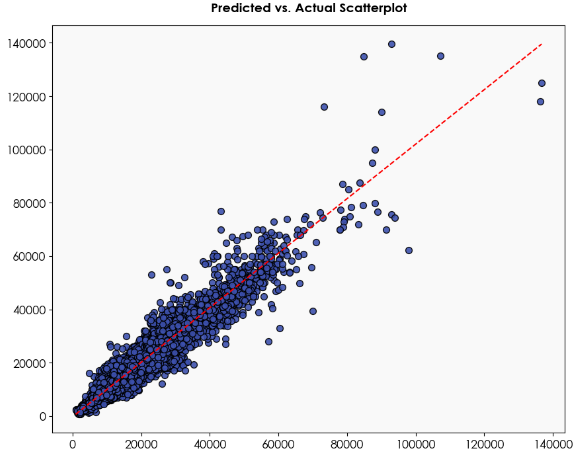
- Best-performing Model : RANDOM FOREST REGRESSOR
- MAPE : 0.1156
- RMSE : 2962.17

Hyperparameter Tuning – Randomized Search

- **R2 : 0.92**
- **MAPE : 10.93%**



```
Regression Model Evaluation Metrics:
-----------------------------------
Mean Squared Error (MSE): 7247929.99
Root Mean Squared Error (RMSE): 2692.20
Mean Absolute Percentage Error (MAPE): 10.93%
Mean Absolute Error (MAE): 1713.48
R-squared Coefficient: 0.92
Explained Variance Score: 0.92
```



WEB APIs DEVELOPED

POST
API/ADD

API used to add a new prediction record to the database table.

Login is needed.

GET
API/GET

API used to get an existing prediction record from the database table.

Login is needed.

DELETE
API/DELETE

API used to delete an existing prediction record from the database table.

Login is needed.

POST
API/PREDICT

API used to perform predictions and gain price insights on the car.

Login is needed.

TESTING WITH PYTEST

54 TESTS

VALIDITY TESTING

Tested on valid data to determine if ordinary data can be used. Ensured data entered was acceptable.

RANGE TESTING

Extreme values for columns such as Year, MPG, Car Age and Mileage etc. Tested both lower and upper bounds.

CONSISTENCY TESTING

Given identical inputs, checked if prediction and output prices are the same.

EXPECTED FAILURE TESTING

Tested XFAIL on:

- Invalid Year of Registration
- Invalid Mileage (< 0)
- Invalid MPG
- Invalid Car Age
- Invalid tax value

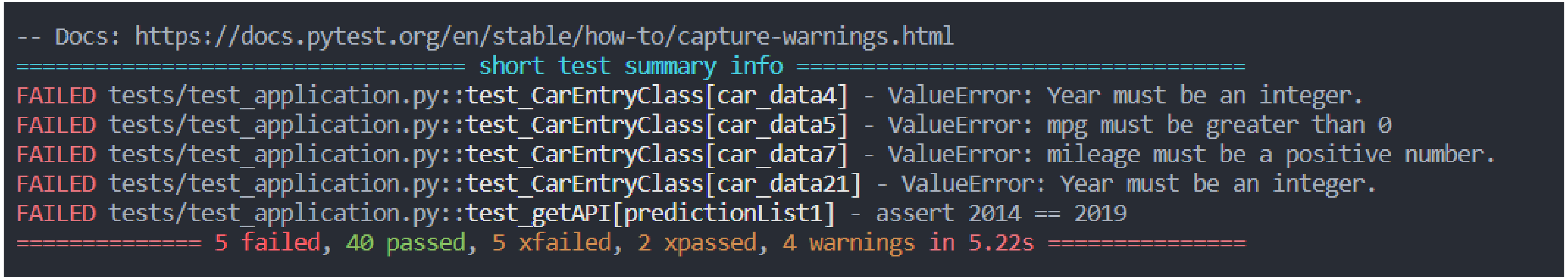
UNEXPECTED FAILURE TESTING

Tested invalid inputs on:

- Year of Registration (Non-integer)
- MPG (Less than 0)
- Mileage (Negative value)

TESTING WITH PYTEST

52 TESTS DONE



```
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
================================== short test summary info ==================================
FAILED tests/test_application.py::test_CarEntryClass[car_data4] - ValueError: Year must be an integer.
FAILED tests/test_application.py::test_CarEntryClass[car_data5] - ValueError: mpg must be greater than 0
FAILED tests/test_application.py::test_CarEntryClass[car_data7] - ValueError: mileage must be a positive number.
FAILED tests/test_application.py::test_CarEntryClass[car_data21] - ValueError: Year must be an integer.
FAILED tests/test_application.py::test_getAPI[predictionList1] - assert 2014 == 2019
============== 5 failed, 40 passed, 5 xfailed, 2 xpassed, 4 warnings in 5.22s ===============
```

THANK YOU!