PROJECT REPORT

Developing A Car Resale Value PredictionModel

UsingMachine Learning

Submitted By

PNT2022TMID32311

Karthik N- 810019104028

Januson J-810019104023

Jefni A.K – 810019104027

Deshma A.N-810019104016

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

a. PROJECT OVERVIEW

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models across cities.

b. PURPOSE

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

CHAPTER - 2

LITERATURE SURVEY

a. EXISTING PROBLEM

In the past year the world of automobiles has seen a drastic change with the semiconductor shortages after the pandemic, which led to spike in used car prices. Hence, there was fast change in car prices during this study which will affect the actual car pricing prediction future. As the current dataset will undervalue the cars in the market. Therefore, a model that is built on real time data can be best integrated into a mobile app for public use would be the idea solution.

b. REFERENCES

1.Predicting the Price of Used Cars using Machine Learning Techniques- Sameer Chand Pudaruth.

2. Car Price Prediction using Machine Learning Techniques- Enis gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric.

3. Price Evaluation Model In Second Hand Car System Based On BP Neural Network Theory- Ning sun, Hongxi Bai, Yuxia Geng, Huizhu Shi.

4. Prediction of Car Price using Linear Regression- A. Rengarajan , Ravi Shastri.

5. Vehicle Price Prediction System using Machine Learning Techniques-

Kanwal Noor, Sadaqat Jan.

6. New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit- Shen Gongqi, W. Yansong, Zhu Qiang.

7. Predicting the Price of Second-hand Cars using Artificial Neural Networks - Saamiyah Peerun, Sameerchand Pudaruth Nushrah Henna Chummun.

8. Used Car Price Prediction using K-Nearest Neighbor Based Model- K.Samruddhi ¸ Dr. R.Ashok Kumar.

9 Prediction of Prices for Used Car by Using Regression Models- Nitis Monburinon, Prajak Chertchom, Thongchai Kaewkiriya, Suwat Rungpheung, Sabir Buya, Pitchayakit Boonpou.

10. Used car price prediction using SVM - Gegic, Isakovic, Keco, Masetic, & Kevric.

C. PROBLEM STATEMENT DEFINITION

Survey 1:

Author: Sameer Chand Pudaruth.

Predicting the Price of Used Cars using Machine Learning Techniques

This paper is predicting the price of used car using machine learning techniques. In this paper, they investigate the application of supervised machine learning techniques to predict the price of used cars in Mauritius. Different techniques like naïve bayes and decision trees have been used to make the predictions. The predictions are then evaluated and compared in order to find those which provide the best performances. A seemingly easy problem turned out to be indeed very difficult to resolve with high accuracy. All the four methods provided comparable

performance.

Survey 2:

Author: Enis gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric

Car Price Prediction using Machine Learning Techniques

A car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars in Bosnia and Herzegovina, we applied three machine learning techniques (Artificial Neural Network, Support Vector Machine and Random Forest). However, the mentioned techniques were applied to work as an ensemble. The data used for the prediction was collected from the web portal autopijaca.ba using web scraper that was written in PHP programming language.

Respective performances of different algorithms were then compared to find one that best suits the available data set. The final prediction model was integrated into Java application. Furthermore, the model was evaluated using test data and the accuracy of 87.38% was obtained.

Survey 3:

Author: Ning sun, Hongxi Bai, Yuxia Geng, Huizhu Shi.

Price Evaluation Model In Second Hand Car System Based On BP Neural Network Theory

With the rapid growth of the number of private cars and the development of the second-hand car market, second-hand cars have become the main choice when people buy cars. The online second-hand car platform provides both buyers and sellers the chance of online P2P

trade. In such systems, the accuracy of second-hand car price evaluation largely determines whether the seller and the buyer can get more efficient trading experience.

Survey 4:

Author: A. Rengarajan , Ravi Shastri.

Prediction of Car Price using Linear Regression

In this paper, we look at how supervised machine learning techniques can be used to forecast car prices in India. Data from the online marketplace quikr was used to make the predictions. The predictions were made using a variety of methods, including multiple linear regression analysis, Random forest regressor and Randomized search CV. The predictions are then analyzed and compared to determine which ones provide the best results.

Survey 5:

Author: Kanwal Noor, Sadaqat Jan.

Vehicle Price Prediction System using Machine Learning Techniques

In this paper, they proposed a model to predict the price of the cars through multiple linear regression method. Here system were able to achieve high level of accuracy using Multiple linear regression models to predict the price of cars collected from used cars website in Pakistan called Pak Wheels that totalled to 1699 records after pre-processing, and where able to achieve accuracy of 98%, this was done after reducing the total amount of attributes using variable selection technique to include significant attributes only and to reduce the complexity of the model.

Survey 6:

Author: Shen Gongqi, W. Yansong, Zhu Qiang.

New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit

A comprehensive method combined by the BP neural network and nonlinear curve fit was introduced for optimizing the model due to its flexible nonlinearity. Firstly, 6 some distribution curves of residual value of the used cars were analyzed in time domain. Then, the BP neural network (NN) was established and used to extract the feature of the distribution curves in various conditions. A set of schemed data was used to train the NN and reached the training goal. Finally, the schemed data as inputs and the NN outputs were organized for nonlinear curve fit. Conclusion was drawn that the newly proposed model is feasible and accurate for residual value prediction of the used cars with various conditions.

Survey 7:

Author: Saamiyah Peerun, Sameerchand Pudaruth Nushrah Henna Chummun.

Predicting the Price of Second-hand Cars using Artificial Neural Networks

The aim of this study is to assess whether it is possible to predict the price of second-hand cars using artificial neural networks. Thus, data for 200 cars from different sources was gathered and fed to four different machine learning algorithms. We found that support vector machine regression produced slightly better results than using a neural network or linear regression. However, some of the predicted values are quite far away from the actual prices, especially for higher priced cars.

Survey 8:

Author: K.Samruddhi ¸ Dr. R.Ashok Kumar

Used Car Price Prediction using K-Nearest Neighbor Based Model

In this paper, a machine learning model is proposed to estimate the cost of the used

cars using the K-Nearest Neighbor algorithm. The model is trained with used cars 7 data for different trained and test ratios. Then the proposed model is cross-validated using K fold method to examine the performance to avoid the over fit.

Survey 9:

Author: Nitis Monburinon, Prajak Chertchom, Thongchai Kaewkiriya, Suwat Rungpheung, Sabir Buya, Pitchayakit Boonpou.

Prediction of Prices for Used Car by Using Regression Models

In this paper, the authors selected the data from the German ecommerce site. The main goal of this work is to find a suitable predictive model to predict the used cars price. They used different machine learning techniques for comparison and used the mean absolute error(MAE) as the metric. They proposed that their model with gradient boosted regression has a lower error with MAE value 0.28 and this gives the higher performance where linear regression has the MAE value 0.55, random forest with MAE value 0.35.

Survey 10:

Author: Gegic, Isakovic, Keco, Masetic, & Kevric.

Used car price prediction using SVM

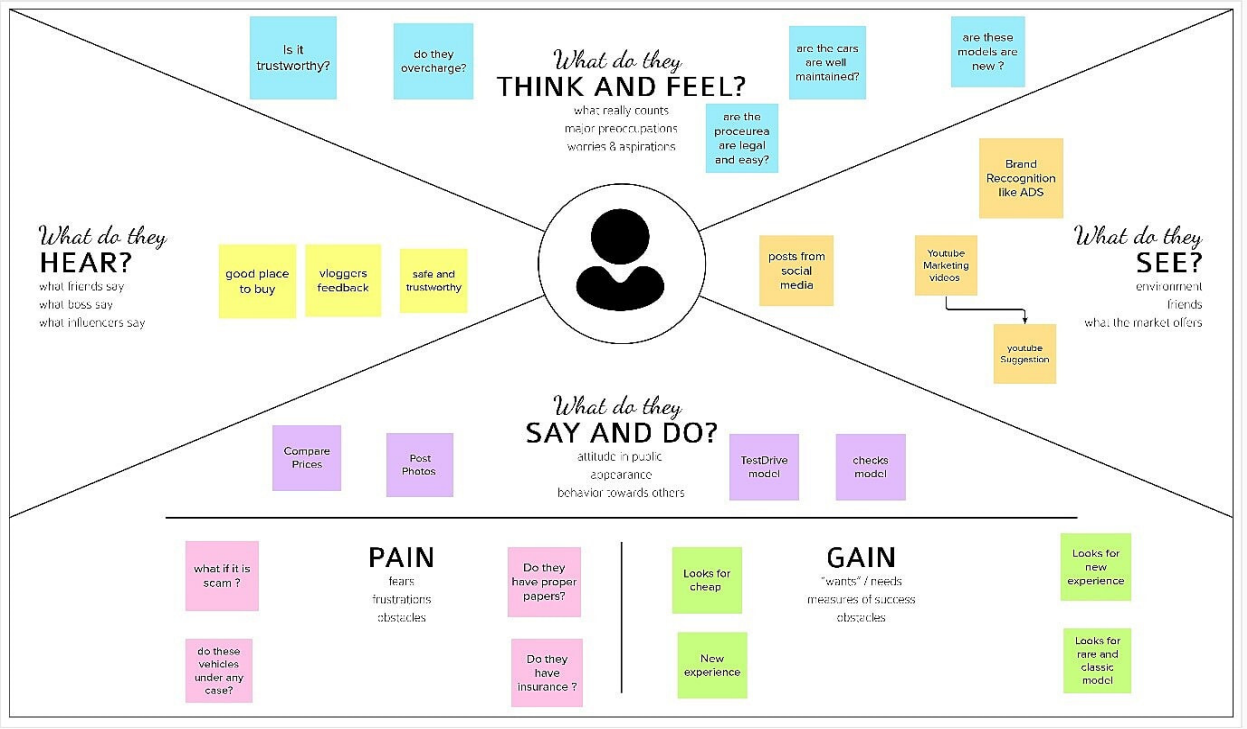
In this paper, using data scrapped from a local Bosnian website for used cars totalled at 797 car samples after pre-processing, and proposed using these methods: Support Vector Machine, Random Forest and Artificial Neural network. Results have shown using only one machine learning algorithm achieved results less than 50%, whereas after combing the algorithms with pre calcification of prices using Random Forest, results with accuracies up to 87.38% was recorded.

CHAPTER 3

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS

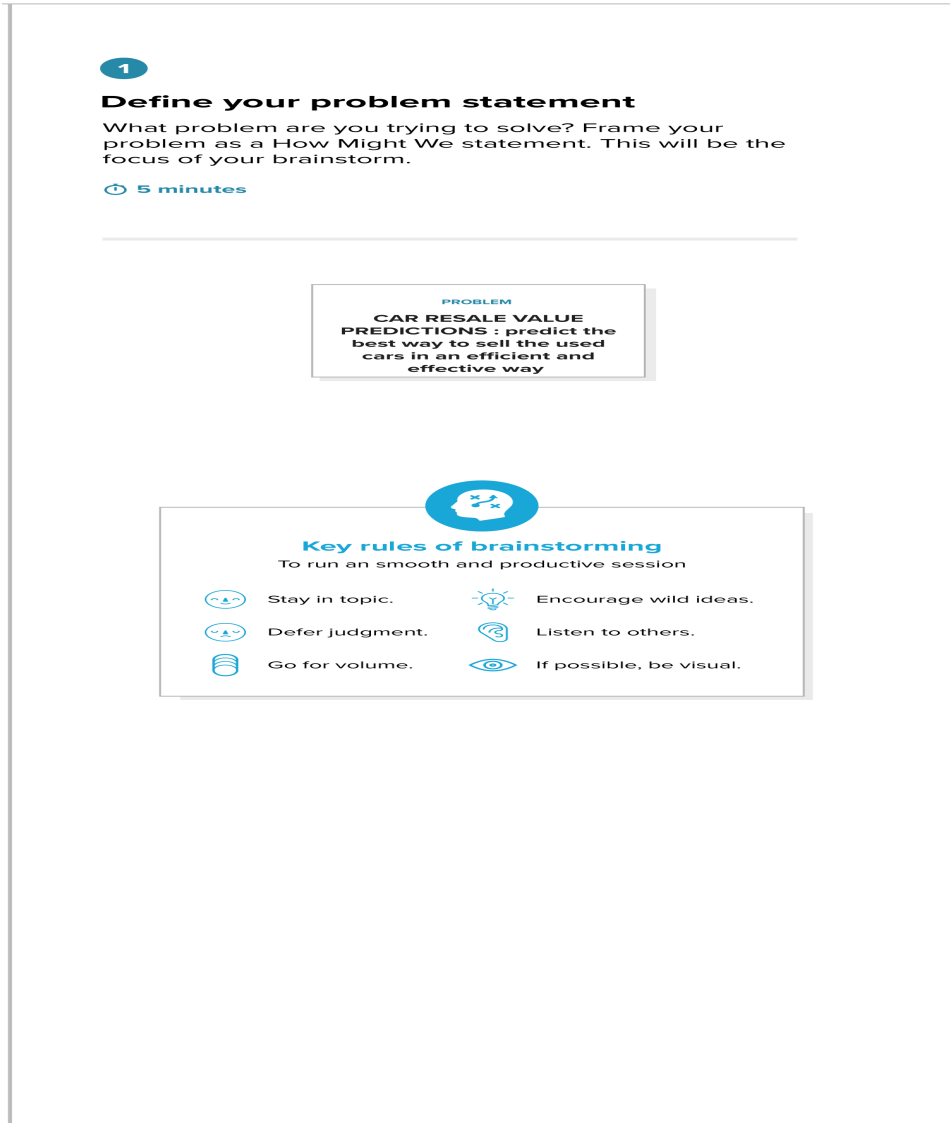
- An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.
- It is a useful tool to helps teams better understand their users.
- Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.



IDEATION & BRAINSTORMING

- Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving.

- Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.



2

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

KARTHIK

- Building model
- checking accuracy
- checking car condition
- Finding insights
- creating profile
- list the no of available cars

JANUSON

- provide feasible price for cars
- collect dataset
- privacy policy formulation
- creating user profiles
- Asking customer queries
- collect the car details

JEFNI

- Use large quantity of dataset
- pre process the data
- Building the UI to interact
- finding efficient algorithm
- finding the missed data

DESHMA

- filling missed out details
- Verify the dataset
- verify customer details
- fixed UI using flask
- list car company
- profile verification digitally

3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

collect dataset

- collect the car details
- collect user details
- finding the missed data
- list the no of available cars
- list car company

CHECKING CONSTRAINTS:

- verify customer details
- Verify the dataset
- checking car condition
- filling missed out details
- pre process the data

EXTRACTING FEATURES :

- categorize the cars based on company
- categorize the cars based on condition and mileage
- categorize the cars based on users need

Build data model

- finding efficient algorithm
- Building data model
- performing analysis using regression algorithm
- Finding insights
- checking accuracy

FINAL DELIVERABLE:

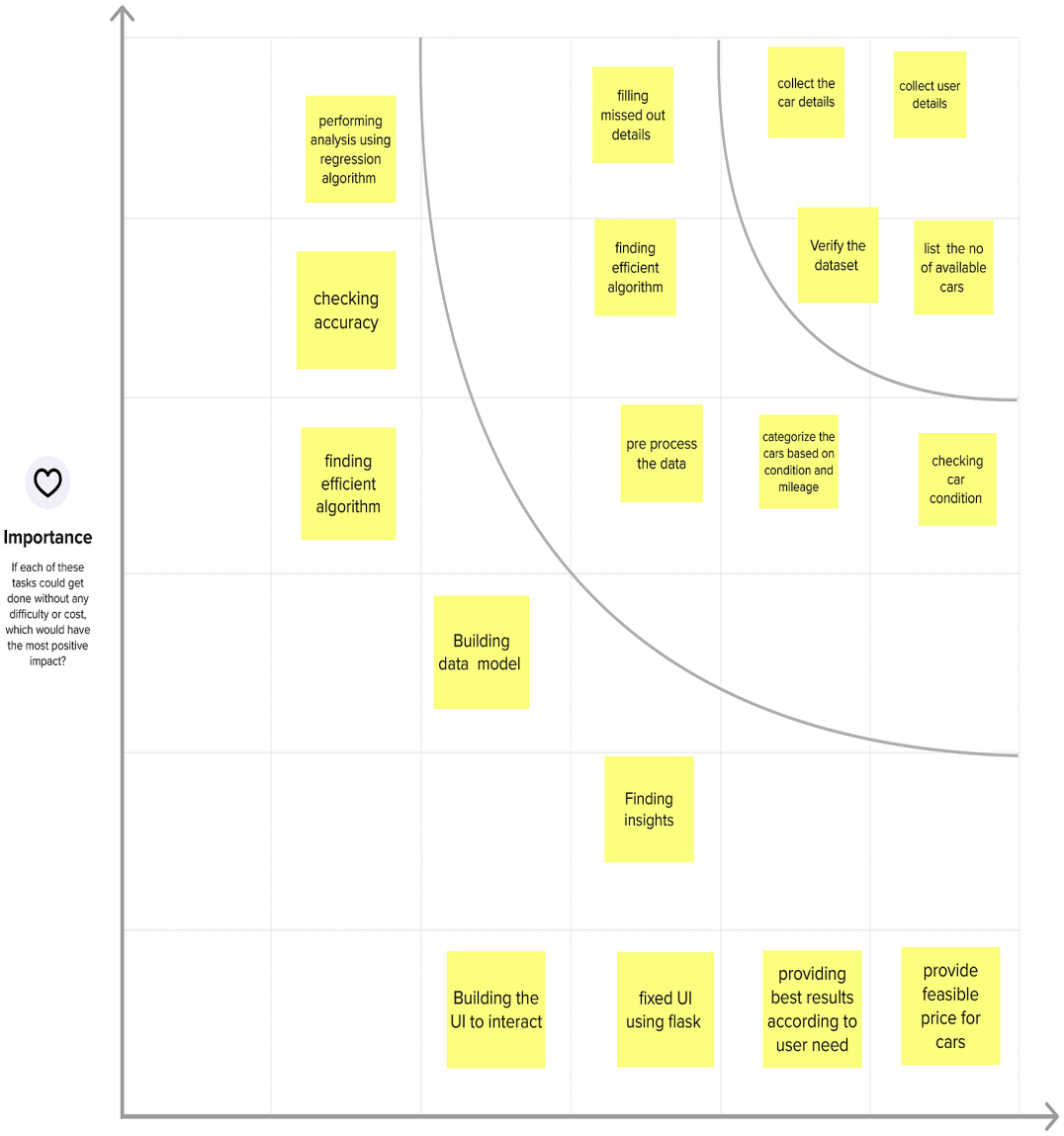
- provide feasible price for cars
- Building the UI to interact
- fixed UI using flask
- providing best results according to user need

4

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

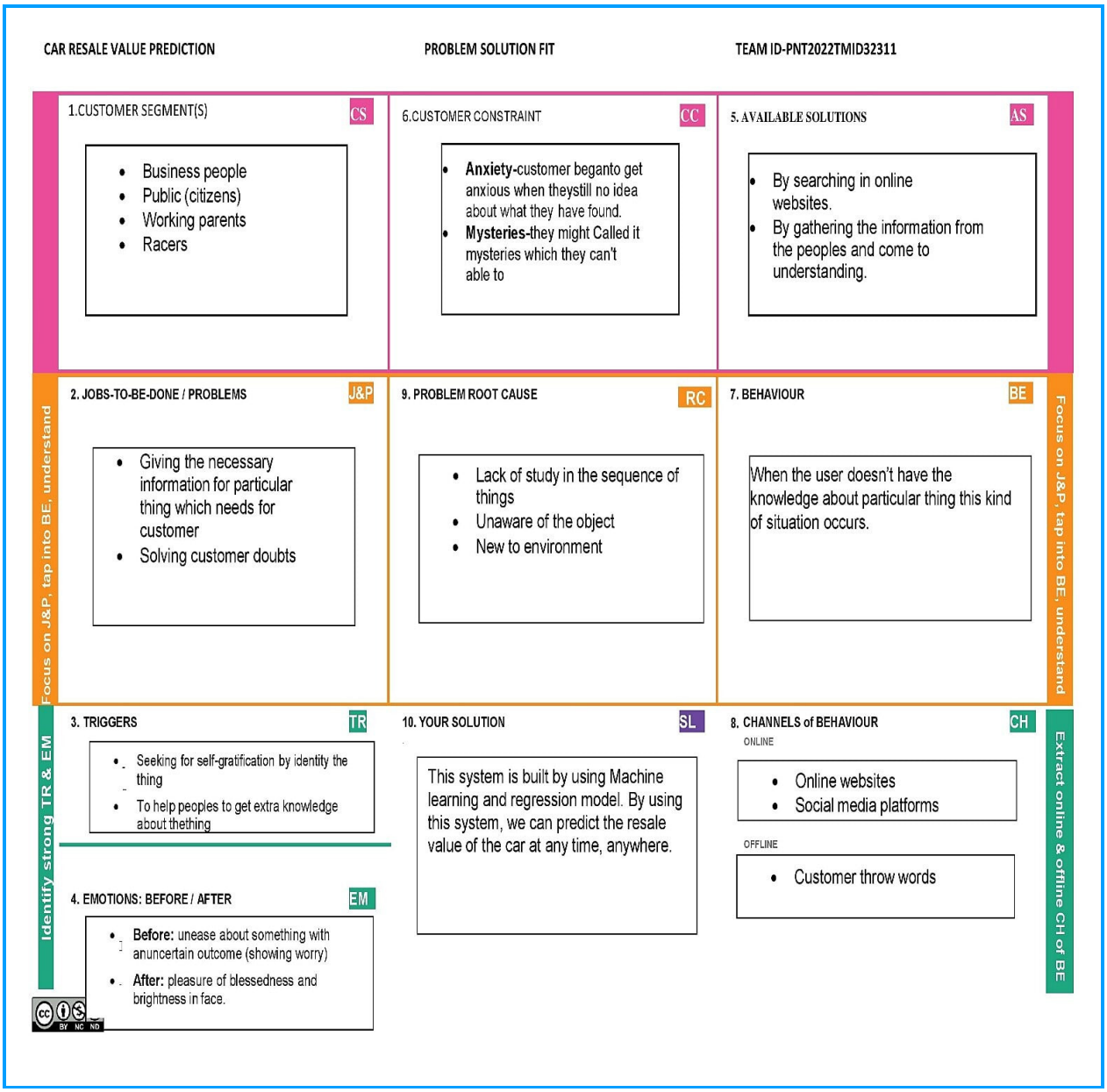
20 minutes



PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Sales prediction is the currentnumerous trend in which all the business companies thrive and it also aids the organization or concern in determining the future goals for it and its plan and procedure to achieve it. Resales of cars almost occupy a major part in every sales economy. In that regard various factors like registration year, engine condition, company servicerecord, spare partscondition, tyre condition, car body condition, kilometers covered, Interior look, color, mileage, number of owners, battery condition are taken into consideration beforebuying it alongwithengine condition and insurance. The prediction using the factors would suggest thefinal product to be brought.But these data may be inaccurate at times and there is a need for a proper algorithm that will provide a result witha good accuracy rate.
2.	Idea / Solution description	1. The overall proposed idea is to predict the car resale value and show it to the required people. 2. This idea can be implemented and could be presented to the customer. This involves two phases. 3. One phase is collecting the dataset for training the car resale value prediction model. 4. Testing the car resale value prediction model. 5. The second phase involves creating a website (front end) for presenting the entire solution as a customized GUI so that this would be very useful for the user to utilize this solution

PROPOSED SOLUTIONFIT



CHAPTER 4

REQUIREMENT A NALYSIS

a. FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through Linked IN.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP.
FR-3	Users Profile	Personal details, Bank account, If customer interested to buy a car
FR-4	Gather information about the vehicle	Then get the registered websites by customer collect information.
FR-5	Display the functionality of the vehicle	Through the registered websites they collect information about the vehicles.

b. NON-FUNCTIONAL REQUIREMENTS

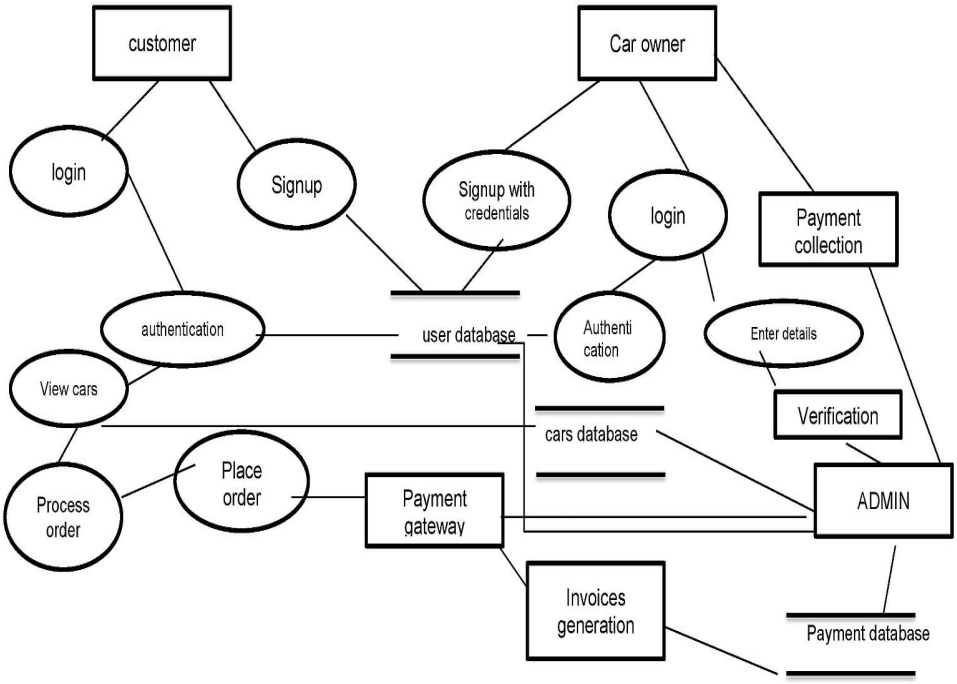
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	To create an UI makes as a user friendly, it makes a simple way to Understand
NFR-2	**Security**	Aware about fraudulent sites, it gives a fake information about the vehicle.
NFR-3	**Reliability**	Application must perform good and without failure
NFR-4	**Performance**	Website performance measures how quickly the pages of a website load and display in the web browser.
NFR-5	**Availability**	Website availability (also called website uptime) refers to the ability of the users to access and use a website or web service. A website's availability is typically communicated as a percentage for a given span of time.

NFR-6	**Scalability**	Application scalability is the ability of an application to handle a growing number of users and load, without compromising on performance and causing disruptions to user experience. To put it another way, scalability reflects the ability of the software to grow or change with the user's demands.

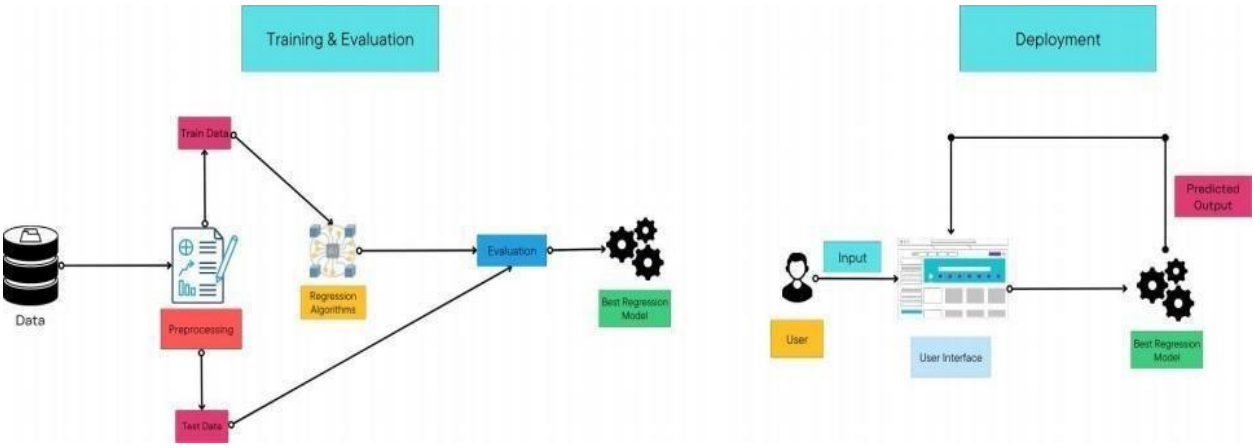
CHAPTER - 5

PROJECT DESIGN

DATA FLOW DIAGRAM



b. SOLUTION &TECHNICAL ARCHITECTURE



CHAPTER 6

PROJECT PLANNING AND SCHEDULING

a. SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement(Epic)	User Story Number	UserStory / Task	Priority
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, andconfirming my password.	High
Sprint-2	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	High
Sprint-1	Registration	USN-3	As a user, I can register for the application through Facebook	High
Sprint-1	Registration	USN-4	As a user, I can register for the application through	Medium

			Gmail	
Sprint-2	Login	USN-5	As a user, I can log into the application by entering email &password	High
Sprint-3	Dashboard	USN-6	After logging into the application, I can view different section and view cars model appropriate time.	Medium
Sprint-2	Login	USN-7	As an admin I can login using username and password.	High
Sprint-2	Training the model	USN-8	As an admin I can train the prediction model through various datasets	High
Sprint-1	Providing user friendly Interface	USN-9	As an admin I can provide specific labels for specific fields.	High

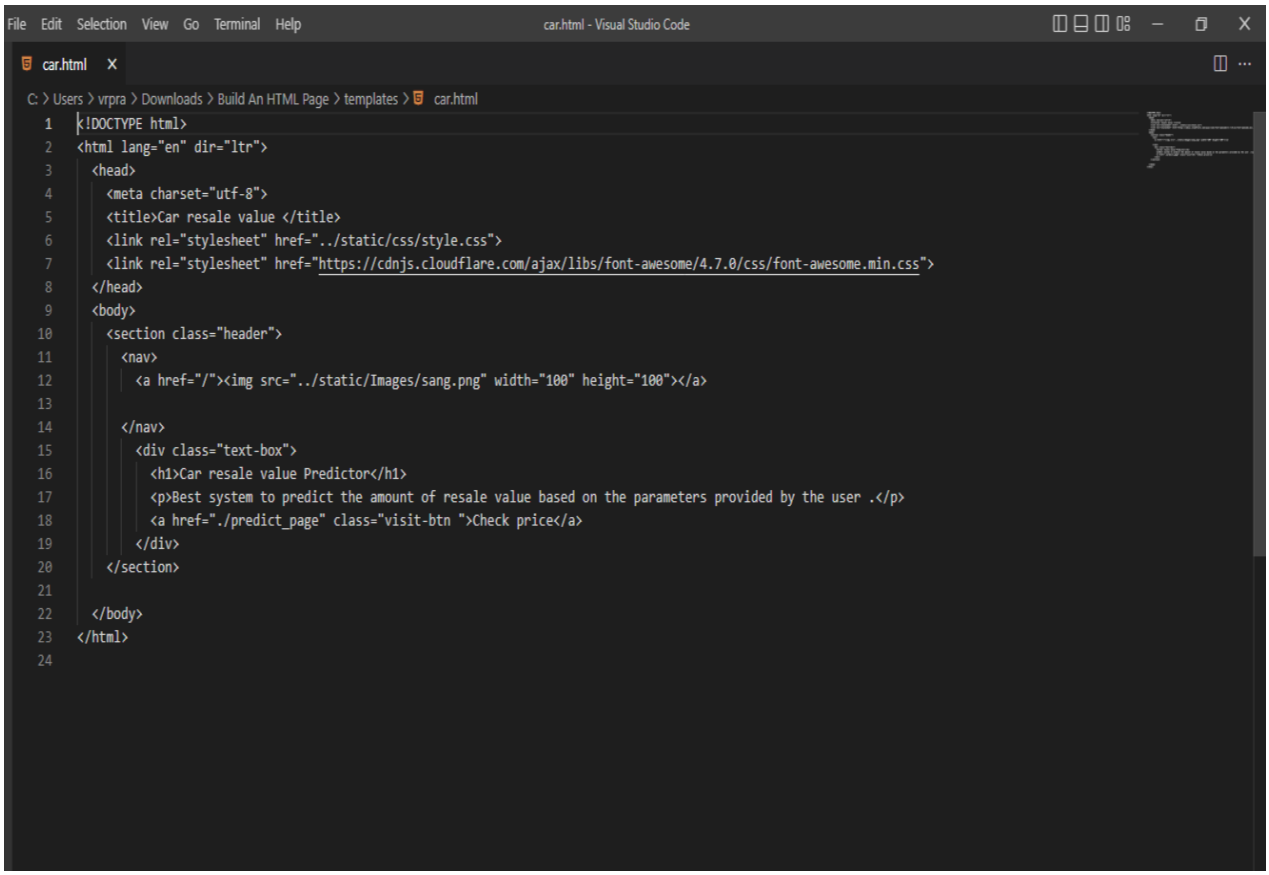
Sprint-2	View options	USN-10	As a car owner I can choose the sections and read policies.	Medium
Sprint-3	Data Entry	USN-11	As a car owner, I can provide data about the car in relevant fields.	High
Sprint 4	Place order	USN-12	As a customer, I select the car suitable to me from the recommended section and place Order	High
Sprint 4	Payment gateway	USN-13	As a customer, pay for the price of the car through payment gateway.	High
Sprint 5	Payment and invoice	USN -14	As an admin, collected amount is distributed to the car owner and invoice is provided to both car Owner and customer.	High

b. SPRINT DELIVERYSCHEDULE

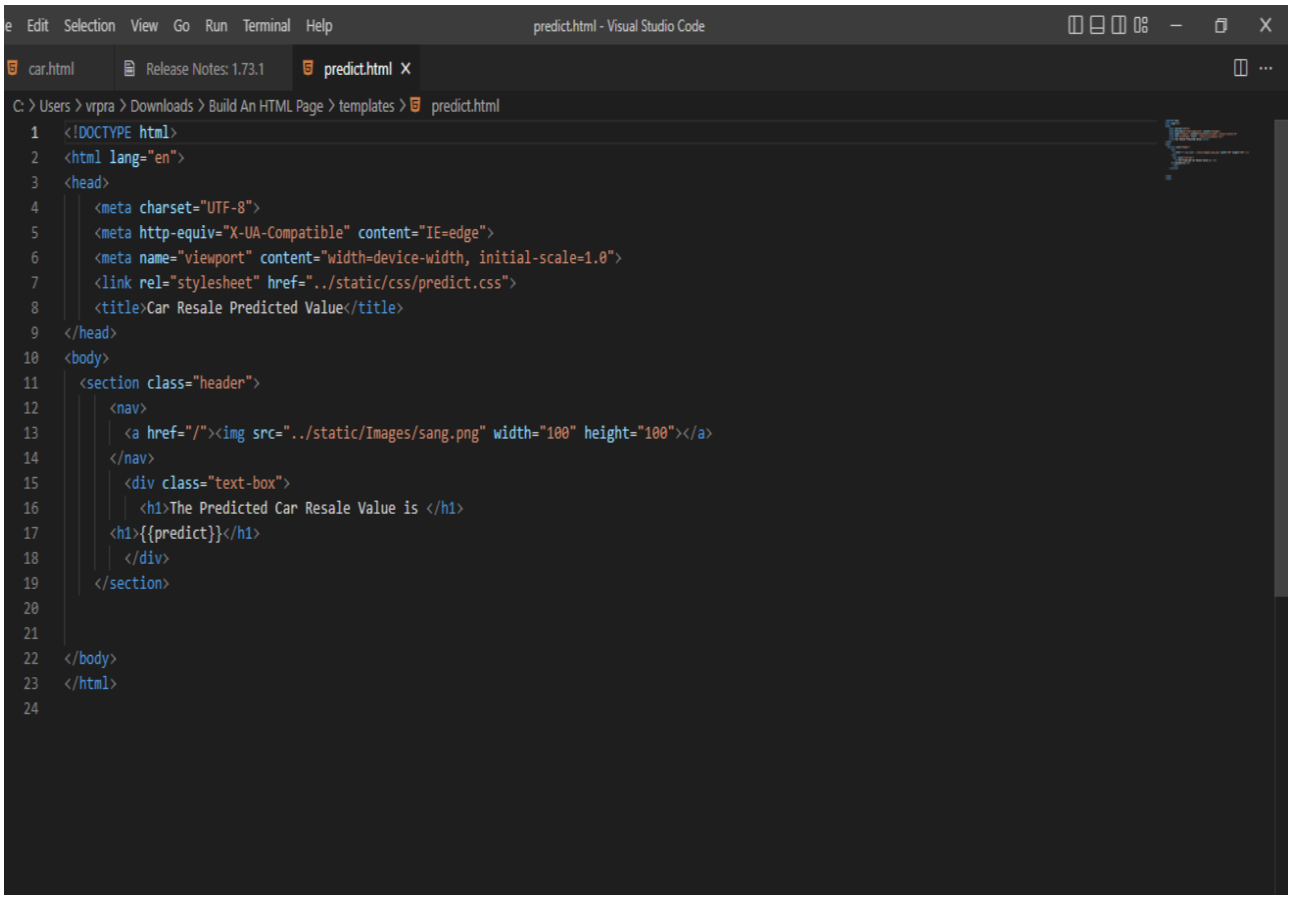
Sprint	Total Story Points	Due	StartDate	Sprint EndDate (Planned)	Story PointsCompleted (as onPlanned End Date)	Sprint Released Date (Actual)
Sprint1	20	3 Days	04 Nov 2022	07 Nov 2022	20	07 Nov 2022
Sprint2	20	3 Days	07 Nov 2022	10 Nov 2022	20	10 Nov2022
Sprint3	20	3 Days	10 Nov 2022	13 Nov 2022	20	13 Nov2022
Sprint4	20	3 Days	13 Nov 2022	16 Nov 2022	20	16 Nov 2022

CHAPTER -7
CODING AND SOLUTION

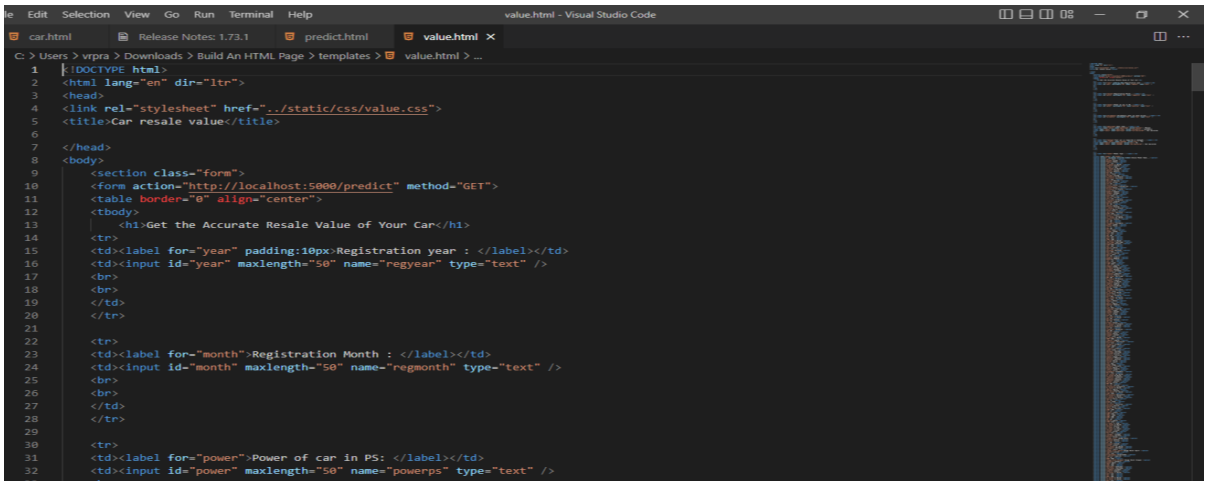
Car.html

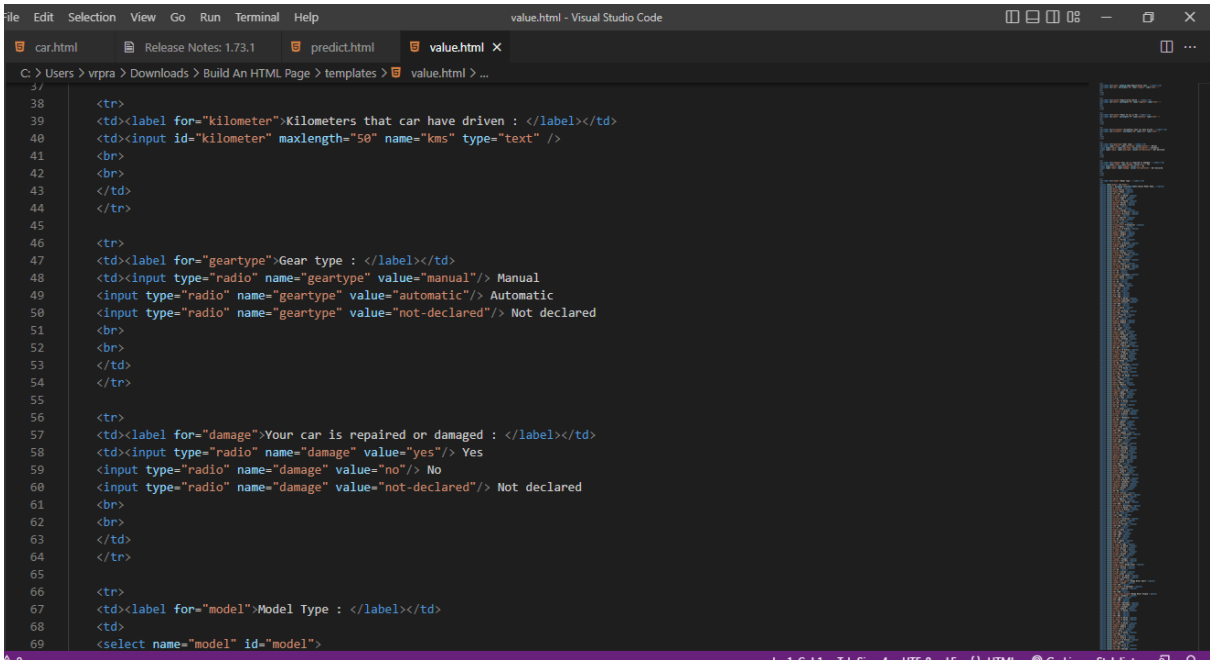


Predict.html



Value.html

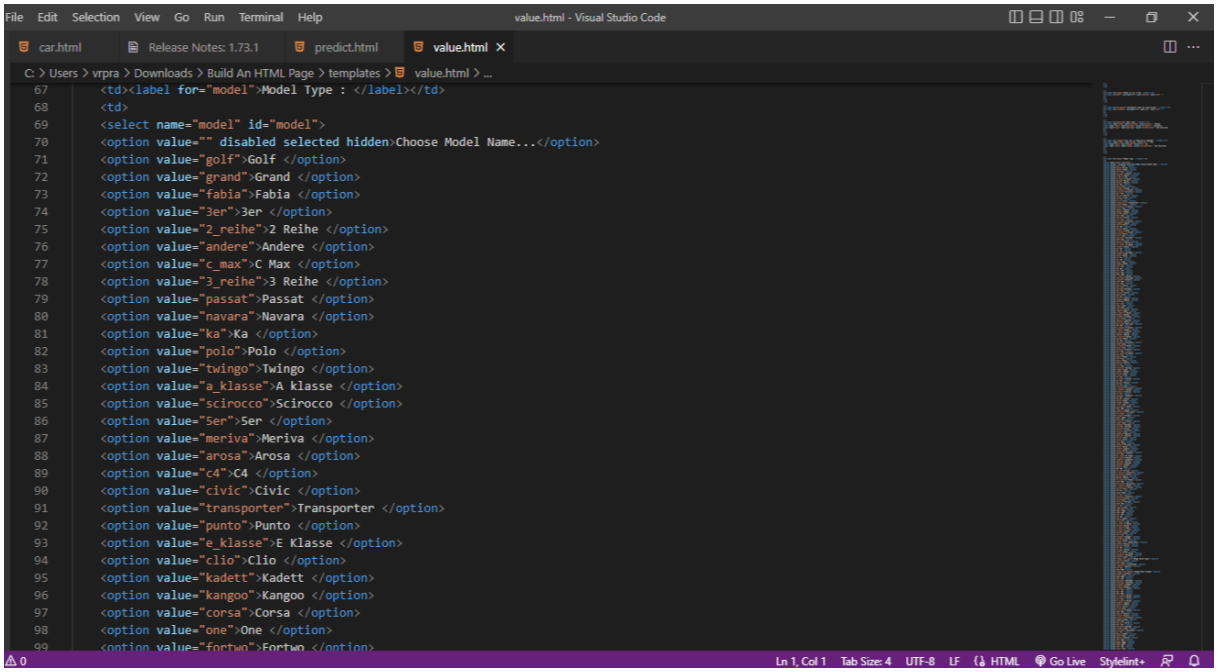


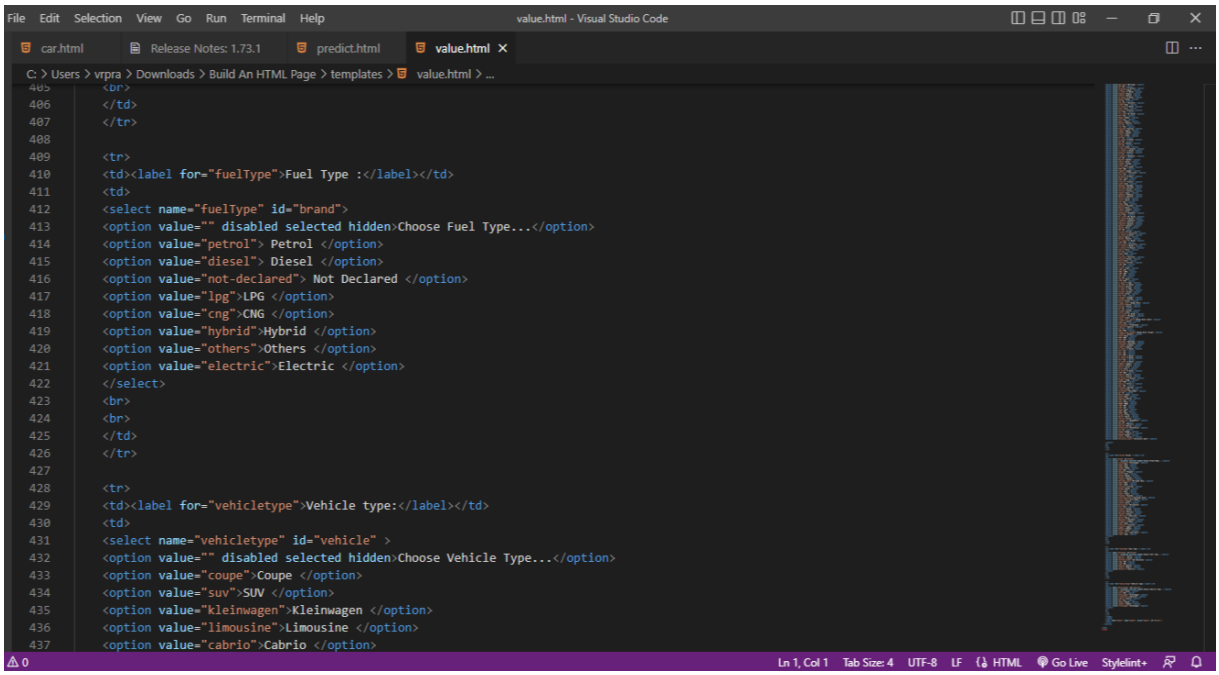

File Edit Selection View Go Run Terminal Help
value.html - Visual Studio Code
car.html
Release Notes: 1.73.1
predict.html
value.html
C: > Users > vrpra > Downloads > Build An HTML Page > templates > value.html > ...
<tr>
<td><label for="kilometer">Kilometers that car have driven : </label></td>
<td><input id="kilometer" maxlength="50" name="kms" type="text" />

</td>
</tr>
<tr>
<td><label for="geartype">Gear type : </label></td>
<td><input type="radio" name="geartype" value="manual"/> Manual
<input type="radio" name="geartype" value="automatic"/> Automatic
<input type="radio" name="geartype" value="not-declared"/> Not declared

</td>
</tr>
<tr>
<td><label for="damage">Your car is repaired or damaged : </label></td>
<td><input type="radio" name="damage" value="yes"/> Yes
<input type="radio" name="damage" value="no"/> No
<input type="radio" name="damage" value="not-declared"/> Not declared

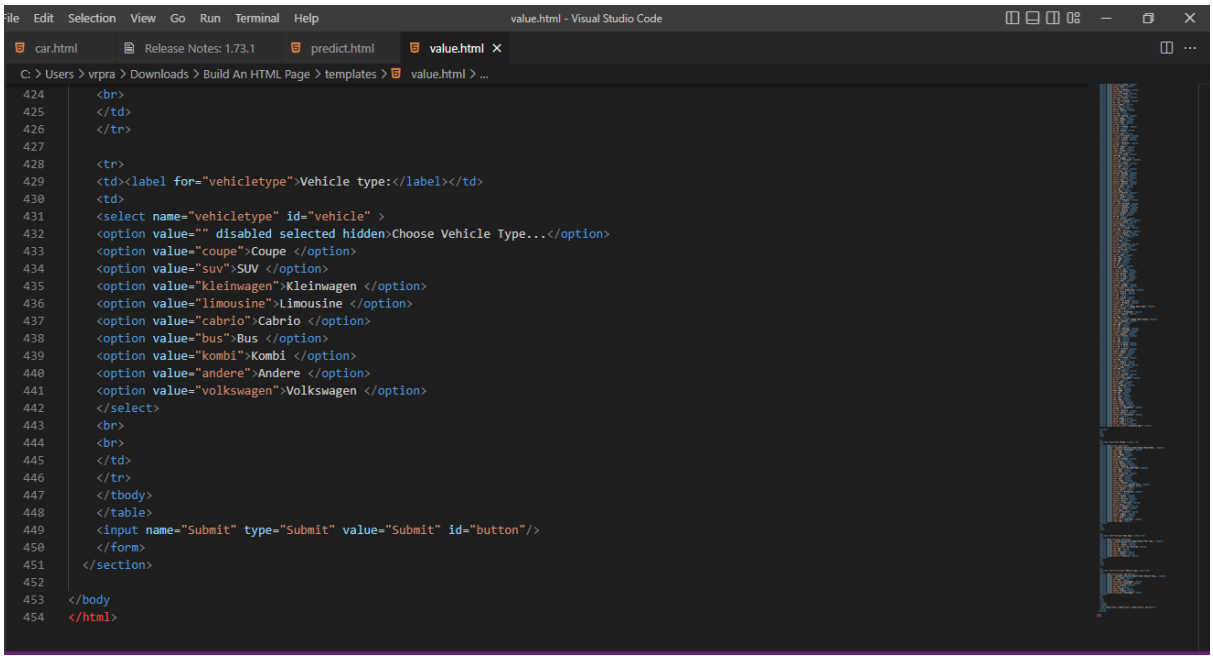
</td>
</tr>
<tr>
<td><label for="model">Model Type : </label></td>
<td>
<select name="model" id="model">


File Edit Selection View Go Run Terminal Help
value.html - Visual Studio Code
car.html
Release Notes: 1.73.1
predict.html
value.html
C: > Users > vrpra > Downloads > Build An HTML Page > templates > value.html > ...
<td><label for="model">Model Type : </label></td>
<td>
<select name="model" id="model">
<option value="" disabled selected hidden>Choose Model Name...</option>
<option value="golf">Golf </option>
<option value="grand">Grand </option>
<option value="fabia">Fabia </option>
<option value="3er">3er </option>
<option value="2_reihe">2 Reihe </option>
<option value="andere">Andere </option>
<option value="c_max">C Max </option>
<option value="3_reihe">3 Reihe </option>
<option value="passat">Passat </option>
<option value="navara">Navara </option>
<option value="ka">Ka </option>
<option value="polo">Polo </option>
<option value="twingo">Twingo </option>
<option value="a_klasse">A klasse </option>
<option value="scirocco">Scirocco </option>
<option value="5er">5er </option>
<option value="meriva">Meriva </option>
<option value="arosa">Arosa </option>
<option value="c4">C4 </option>
<option value="civic">Civic </option>
<option value="transporter">Transporter </option>
<option value="punto">Punto </option>
<option value="e_klasse">E Klasse </option>
<option value="clio">Clio </option>
<option value="kadett">Kadett </option>
<option value="kangoo">Kangoo </option>
<option value="corsa">Corsa </option>
<option value="one">One </option>
Ln 1, Col 1
Tab Size: 4
UTF-8
LF
HTML
Go Live
Stylelint+


File Edit Selection View Go Run Terminal Help
value.html - Visual Studio Code
car.html
Release Notes: 1.73.1
predict.html
value.html
C: > Users > vrpra > Downloads > Build An HTML Page > templates > value.html > ...

</td>
</tr>
<tr>
<td><label for="fuelType">Fuel Type :</label></td>
<td>
<select name="fuelType" id="brand">
<option value="" disabled selected hidden>Choose Fuel Type...</option>
<option value="petrol"> Petrol </option>
<option value="diesel"> Diesel </option>
<option value="not-declared"> Not Declared </option>
<option value="lpg">LPG </option>
<option value="cng">CNG </option>
<option value="hybrid">Hybrid </option>
<option value="others">Others </option>
<option value="electric">Electric </option>
</select>

</td>
</tr>
<tr>
<td><label for="vehicletype">Vehicle type:</label></td>
<td>
<select name="vehicletype" id="vehicle" >
<option value="" disabled selected hidden>Choose Vehicle Type...</option>
<option value="coupe">Coupe </option>
<option value="suv">SUV </option>
<option value="kleinwagen">Kleinwagen </option>
<option value="limousine">Limousine </option>
<option value="cabrio">Cabrio </option>
Ln 1, Col 1 Tab Size: 4 UTF-8 LF HTML Go Live Stylelint+


File Edit Selection View Go Run Terminal Help
value.html - Visual Studio Code
car.html
Release Notes: 1.73.1
predict.html
value.html
C: > Users > vrpra > Downloads > Build An HTML Page > templates > value.html > ...

</td>
</tr>
<tr>
<td><label for="vehicletype">Vehicle type:</label></td>
<td>
<select name="vehicletype" id="vehicle" >
<option value="" disabled selected hidden>Choose Vehicle Type...</option>
<option value="coupe">Coupe </option>
<option value="suv">SUV </option>
<option value="kleinwagen">Kleinwagen </option>
<option value="limousine">Limousine </option>
<option value="cabrio">Cabrio </option>
<option value="bus">Bus </option>
<option value="kombi">Kombi </option>
<option value="andere">Andere </option>
<option value="volkswagen">Volkswagen </option>
</select>

</td>
</tr>
</tbody>
</table>
<input name="Submit" type="Submit" value="Submit" id="button"/>
</form>
</section>
</body>
</html>

CHAPTER 8
TESTING

8.1 TEST CASES

Test case ID	**Feature**	**Component**	**Scenario**	**Expected Result**	**Actual Result**	**Status**
HP_TC_001	UI	Home Page	Elements in the home page	The Homepage must be displayed properly	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the details of cars should display properly	Working as expected	PASS
RP_TC_001	UI	Result	Verify UI elements in the Result Page	The Result page must be display properly	Working as expected	PASS

a. USER ACCEPTANCE TESTING

i. DEFECT ANALYSIS

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	1	4	2	3	10
Duplicate	0	0	3	0	3
External	0	1	0	0	1
Fixed	1	3	2	6	12
Not Reproduced	0	0	0	0	0
Skipped	1	1	0	0	2
Won't Fix	1	0	0	0	1
Totals	4	9	7	9	29

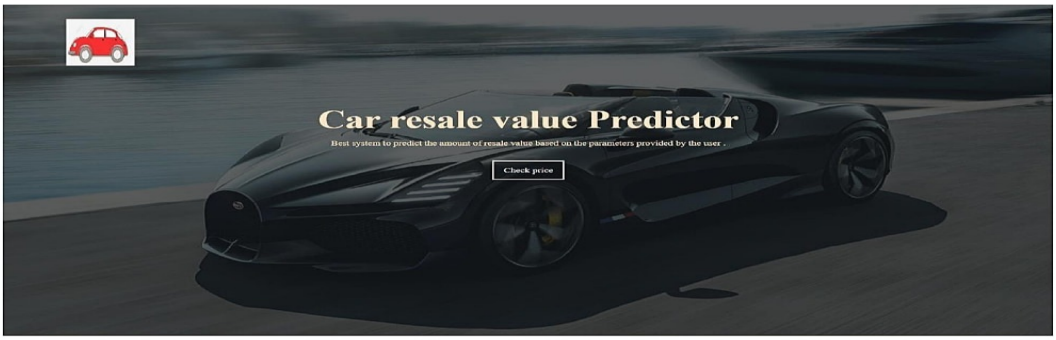
ii. TEST CASE ANALYSIS

Section	Total Cases	Not Tested	Fail	Pass
Model Evaluation	10	0	0	10
Client Application	20	0	0	20
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4

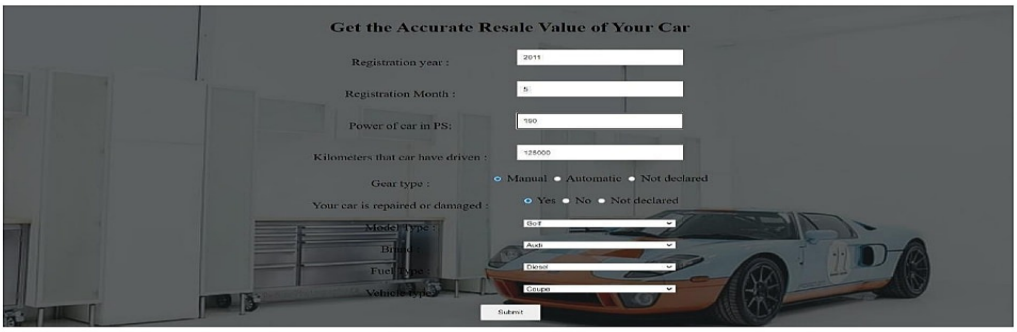
CHAPTER - 9
RESULTS

9.1 PERFORMANCE METRICS

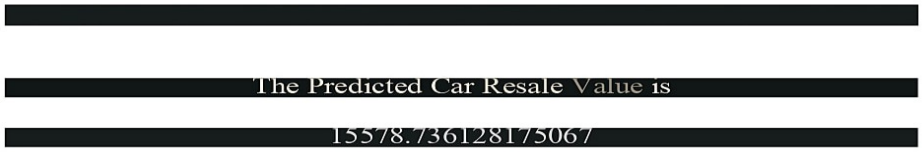
1) Home Page



2) Data Entry Page



3) Car Resale Value Display Page



CHAPTER - 10
ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. Reduces manual work

1. More accurate than average human

2. Capable of handlinga lot of data

3. Can be used anywhere from any device

DISADVANTAGES

4. Cannot handle complexdata

5. All the data must be in digital format

6. Requires a high performance server for faster predictions

7. Prone to occasional errors

CHAPTER 11
CONCLUSION

This project demonstrated a web application that uses machine learning to predict the car resale value for customers. Flask, HTML, CSS, JavaScript, and a few other technologies were used to create this project. The model predicts the handwritten digit using a CNN network. During testing, the model achieved a 99.61% recognition rate. The proposed project is scalable and can easily handle a huge number of users. Since it is a web application, it is compatible with any device that can run a browser. This project is extremely useful in real-world scenarios such as buying a second handed car in optimal best , sell the old cars in single place in best price and so on. There is so much room for improvement, which can be implemented in subsequent versions.

CHAPTER - 12

FUTURE SCOPE

For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset. To correct for overfitting in Random Forest, different selections of features and number of trees will be tested to check for change in performance.

APPENDIX
SOURCE CODE

MODEL CREATION

Integrateflask.py

integrate_flask.py - C:\Users\vrpra\Downloads\integrate_flask.py (3.8.2)

File Edit Format Run Options Window Help

```
# Import Libraries
import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder
import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "Qo9j8ni7qMJ8jlC8VFDRFHbuGRAhYWcTlkVqnYglAGkE"
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__)#initiate flask app

def load_model(file='../Result/resale_model.sav'):#load the saved model
        return pickle.load(open(file, 'rb'))

@app.route('/')
def index():#main page
        return render_template('car.html')

@app.route('/predict_page')
def predict_page():#predicting page
        return render_template('value.html')

@app.route('/predict', methods=['GET','POST'])
def predict():
        reg_year = int(request.args.get('regyear'))
        powerps = float(request.args.get('powerps'))
        kms= float(request.args.get('kms'))
        reg_month = int(request.args.get('regmonth'))

        gearbox = request.args.get('geartype')
        damage = request.args.get('damage')
        model = request.args.get('model')
        brand = request.args.get('brand')
        fuel_type = request.args.get('fuelType')
        veh_type = request.args.get('vehicletype')
```

integrate_flask.py - C:\Users\vrpra\Downloads\integrate_flask.py (3.8.2)

File Edit Format Run Options Window Help

```
        damage = request.args.get('damage')
        model = request.args.get('model')
        brand = request.args.get('brand')
        fuel_type = request.args.get('fuelType')
        veh_type = request.args.get('vehicletype')

        new_row = {'yearOfReg':reg_year, 'powerPS':powerps, 'kilometer':kms,
                                'monthOfRegistration':reg_month, 'gearbox':gearbox,
                                'notRepairedDamage':damage,
                                'model':model, 'brand':brand, 'fuelType':fuel_type,
                                'vehicletype':veh_type}

        print(new_row)

        new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','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('../Result/'+str('classes'+i+'.npy'), allow_pickle=True)
                transform = mapper[i].fit_transform(new_df[i])
                new_df.loc[:,i+'_labels'] = pd.Series(transform, index=new_df.index)

        labeled = new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] + [x+'_labels' for x in labels]]

        X = labeled.values.tolist()
        print('\n\n', X)
        #predict = reg_model.predict(X)

        # NOTE: manually define and pass the array(s) of values to be scored in the next line
        payload_scoring = {"input_data": [{"fields": [['yearOfReg', 'powerPS', 'kilometer', 'monthOfRegistration','gearbox_labels', 'notRepairedDamage_labels', 'model_

        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f67cbed-6222-413b-9901-b2a72807ac82/predictions?version=2022-10-30', jso
        predictions = response_scoring.json()
        print(response_scoring.json())
        predict = predictions['predictions'][0]['values'][0][0]
        print("Final prediction :",predict)
```

integrate_flask.py - C:\Users\vrpra\Downloads\integrate_flask.py (3.8.2)

File Edit Format Run Options Window Help

```
        new_row = {'yearOfReg':reg_year, 'powerPS':powerps, 'kilometer':kms,
                                'monthOfRegistration':reg_month, 'gearbox':gearbox,
                                'notRepairedDamage':damage,
                                'model':model, 'brand':brand, 'fuelType':fuel_type,
                                'vehicletype':veh_type}

        print(new_row)

        new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','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('../Result/'+str('classes'+i+'.npy'), allow_pickle=True)
                transform = mapper[i].fit_transform(new_df[i])
                new_df.loc[:,i+'_labels'] = pd.Series(transform, index=new_df.index)

        labeled = new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] + [x+'_labels' for x in labels]]

        X = labeled.values.tolist()
        print('\n\n', X)
        #predict = reg_model.predict(X)

        # NOTE: manually define and pass the array(s) of values to be scored in the next line
        payload_scoring = {"input_data": [{"fields": [['yearOfReg', 'powerPS', 'kilometer', 'monthOfRegistration','gearbox_labels', 'notRepairedDamage_labels', 'model_

        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/7f67cbed-6222-413b-9901-b2a72807ac82/predictions?version=2022-10-30', jso
        predictions = response_scoring.json()
        print(response_scoring.json())
        predict = predictions['predictions'][0]['values'][0][0]
        print("Final prediction :",predict)

        return render_template('predict.html',predict=predict)

if __name__=='__main__':
        reg_model = load_model()#load the saved model
        app.run(host='localhost', debug=True, threaded=False)
```

GITHUB LINK :

https://github.com/IBM-EPBL/IBM-Project-7140-1658848269