Professional Readiness for Innovation, Employability, and Entrepreneurship

PROJECT REPORT

Title	**:** Car Resale Value Prediction	
Team ID	**:** PNT2022TMID34489	
Team Lead	**:** AJAY S	(961819104301)
Members	**:** NIHAAL M S	(961819104059)
	ABILASH KUMAR M	(961819104003)
	PRABIN KUMAR K K	(961819104063)

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Project Overview

This system “Car Resale Value Prediction” aims to build a regression model to predict used cars' resale value based on multiple aspects, including vehicle mileage, year of manufacturing, fuel consumption, transmission, road tax, fuel type, and engine size. This model can benefit sellers, buyers, and car manufacturers in the used cars market. Upon completion, it can output a relatively accurate price prediction based on the information that user’s input. Various regression methods, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression, were applied in the research to achieve the highest accuracy.
This system was implemented as a web application where the user enters the details of the car to get an estimation of the car’s resale value.

1.2 Purpose

Car resale value prediction helps the user to predict the resale value of the car depending upon various features like kilometers driven, fuel type, etc. The purpose of this system is of commercial interest to sellers/financer to be able to predict the resale value of cars with better accuracy. The most essential elements for forecast are brand and model, period use of vehicle, mileage of vehicle, gear type and fuel type utilized in the vehicle just as fuel utilization per mile profoundly influences cost of a vehicle because of continuous changes in the cost of a fuel. In view of the differing highlights and factors, and furthermore with the assistance of master information the vehicle resale value forecast has been done precisely.

2. LITERATURE SURVEY

2.1 Existing problem

With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. In many developed countries, it is common to lease a car rather than buying it outright. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e., its expected resale value. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy.

2.2 References

Project Title	**Author**	**Abstract**
Price Prediction of Used Cars Using Machine Learning	Chuyang Jin	This work aims to build a model to predict used cars' reasonable prices based on multiple aspects. Various regression methods, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression, were applied in the work to obtain highest accuracy. Compared to previous research, the resulting model includes more aspects of used cars while also having a higher prediction accuracy.
Prediction of Prices for Used Car by using Regression Models (2018)	Nitis Monburinon, Prajak Chertchom, Thongchai Kaewkiriya, Suwat Rungpheung, Sabir Buya, Pitchayakit Boonpou.	In this work, a model to evaluate price based on big data analysis is proposed. It takes advantage of vehicle data and vehicle transaction data to analyze the price data for each type of vehicles. The work uses optimized Back Propagation neural network algorithm.

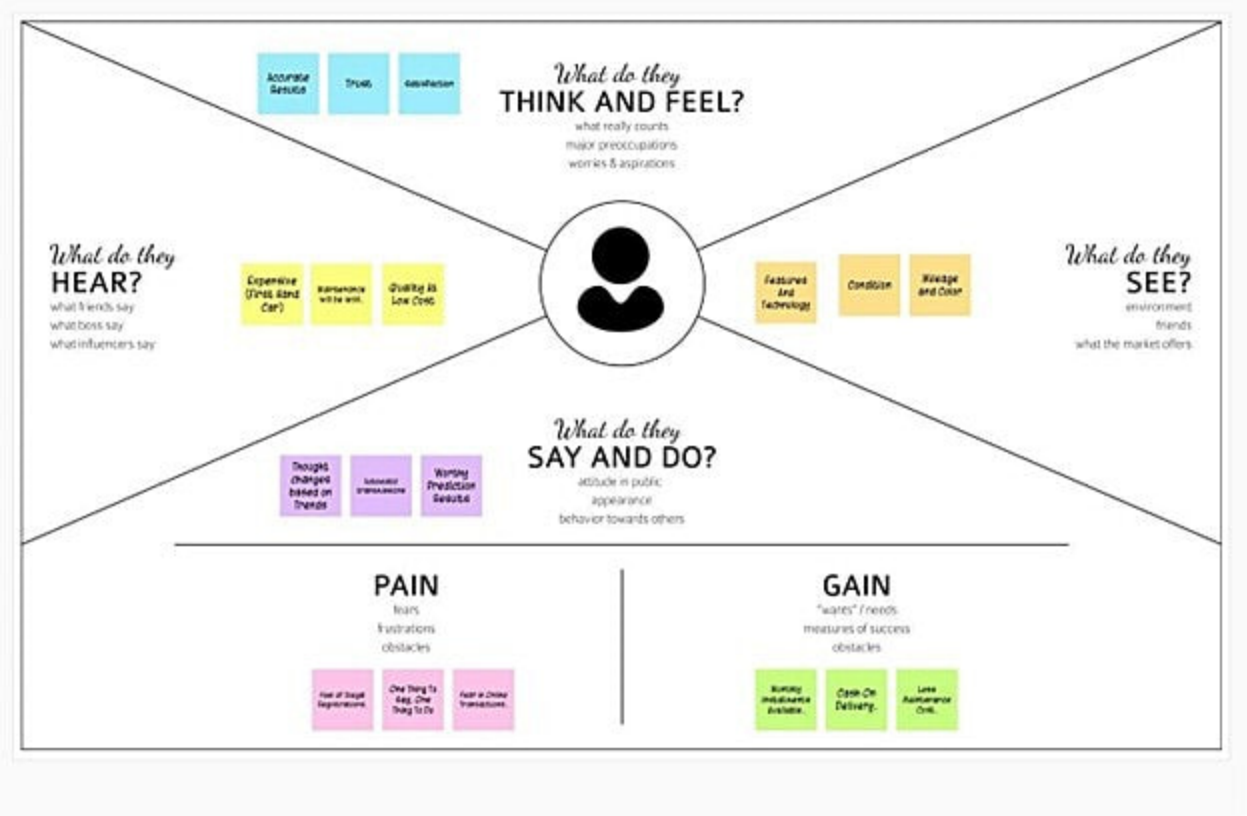
Car Price Prediction Using Machine Learning (2019)	Enis gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric.	In this work, several distinct attributes are analyzed for the reliable and accurate prediction. The work is to build a model to predict the resale price of cars in Bosnia and Herzegovina
Used Car price prediction (2021)	Praful Rane, Deep Pandya, Dhawal Kotak.	In this work, machine learning models that can accurately predict the price of a used car based on its features was built. They have implemented and evaluated various learning methods on dataset consisting of the sale prices of different models.
Prediction of Used Car Price Based on Supervised Learning Algorithm (2021)	Feng Wang, Xusong Zhang; Qiang Wang	In this work, Extra Trees Regressor, Random Forest Regressor was used. Finally, the algorithm was optimized by using the hyperparameter function. The results show that R2 = 0.9807 obtained from extreme random numbers is the best performance. The algorithm was obtained and validated with new data to derive the final algorithm model.

2.3 Problem Statement Definition

It is easy for any company to price their new cars based on the manufacturing and marketing cost it involves. But when it comes to a used car it is quite difficult to define a price because it involves it is influenced by various parameters like car brand, manufactured year etc. The goal of our system is to predict the best price for a used car in the based on the previous data related to sold cars using machine learning.

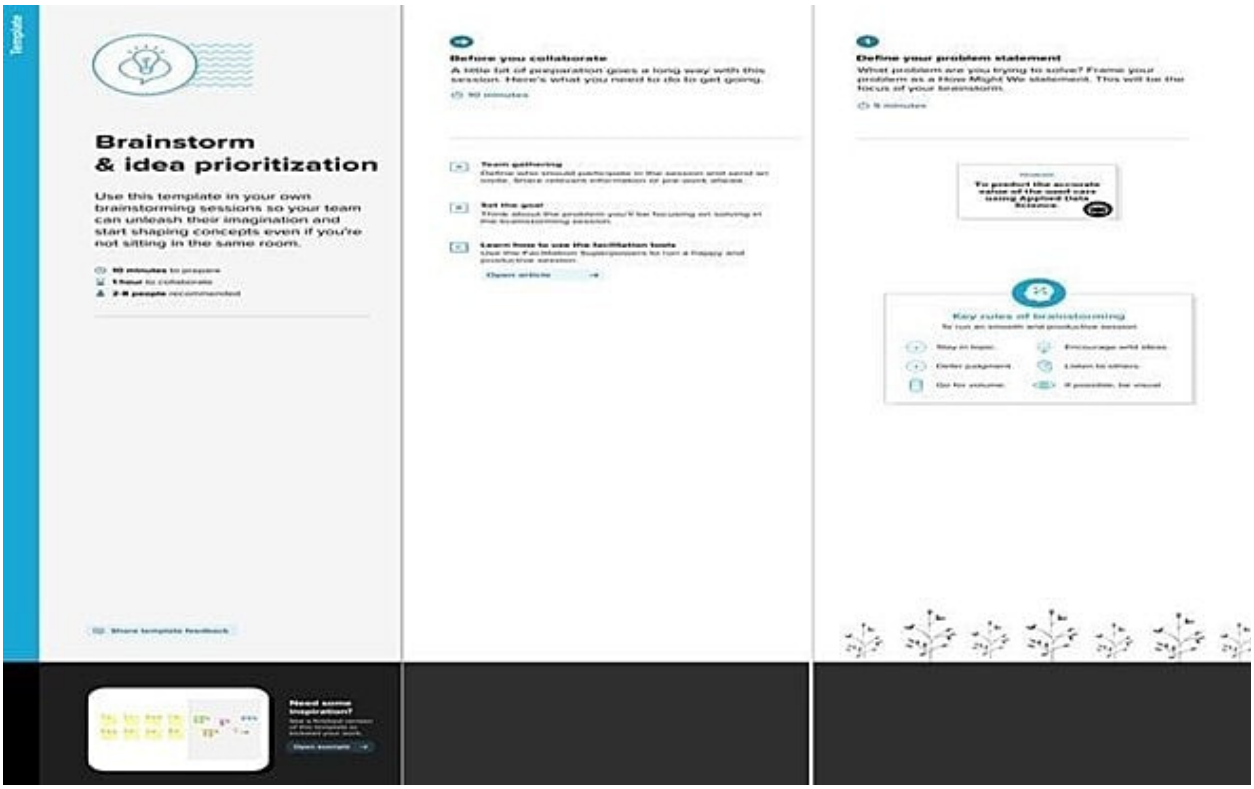
3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

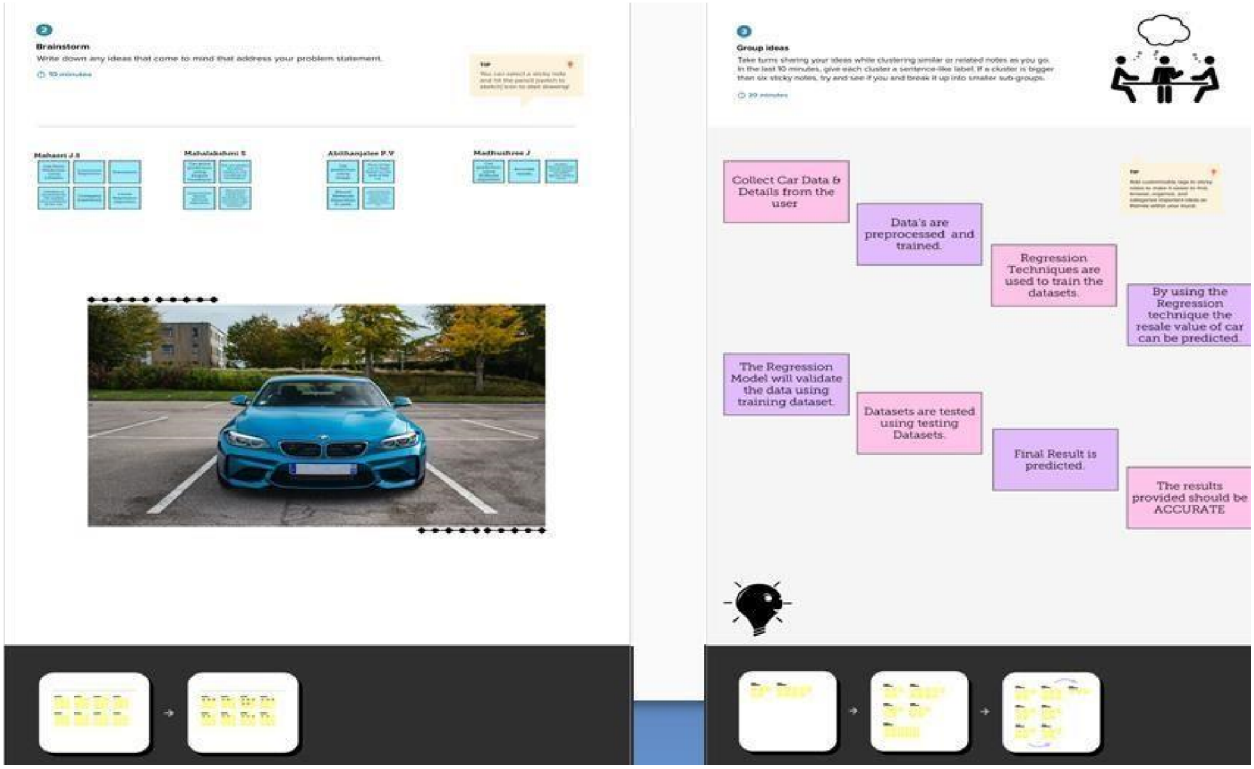


3.2 Ideation & Brainstorming

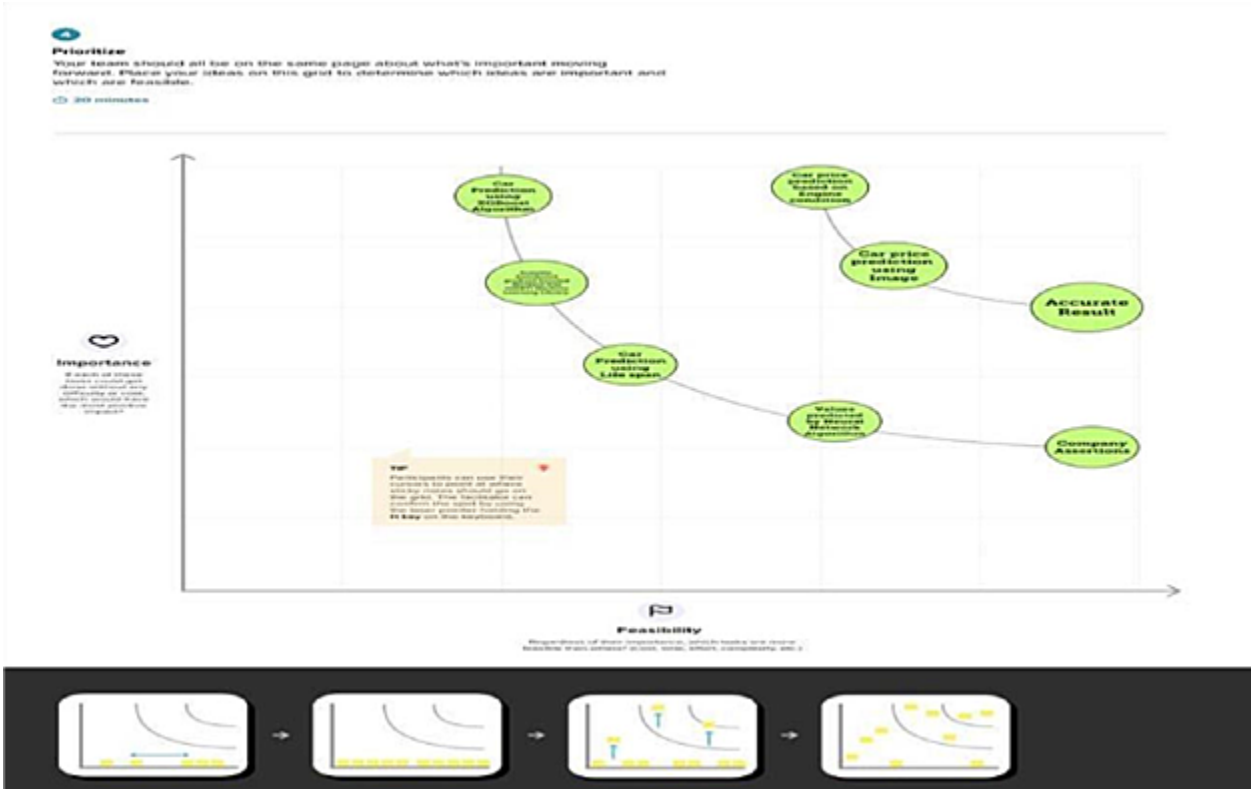
Step-1: Team Gathering, Collaboration and Select the Problem Statement



Step-2: Brainstorm, Idea Listing and Grouping



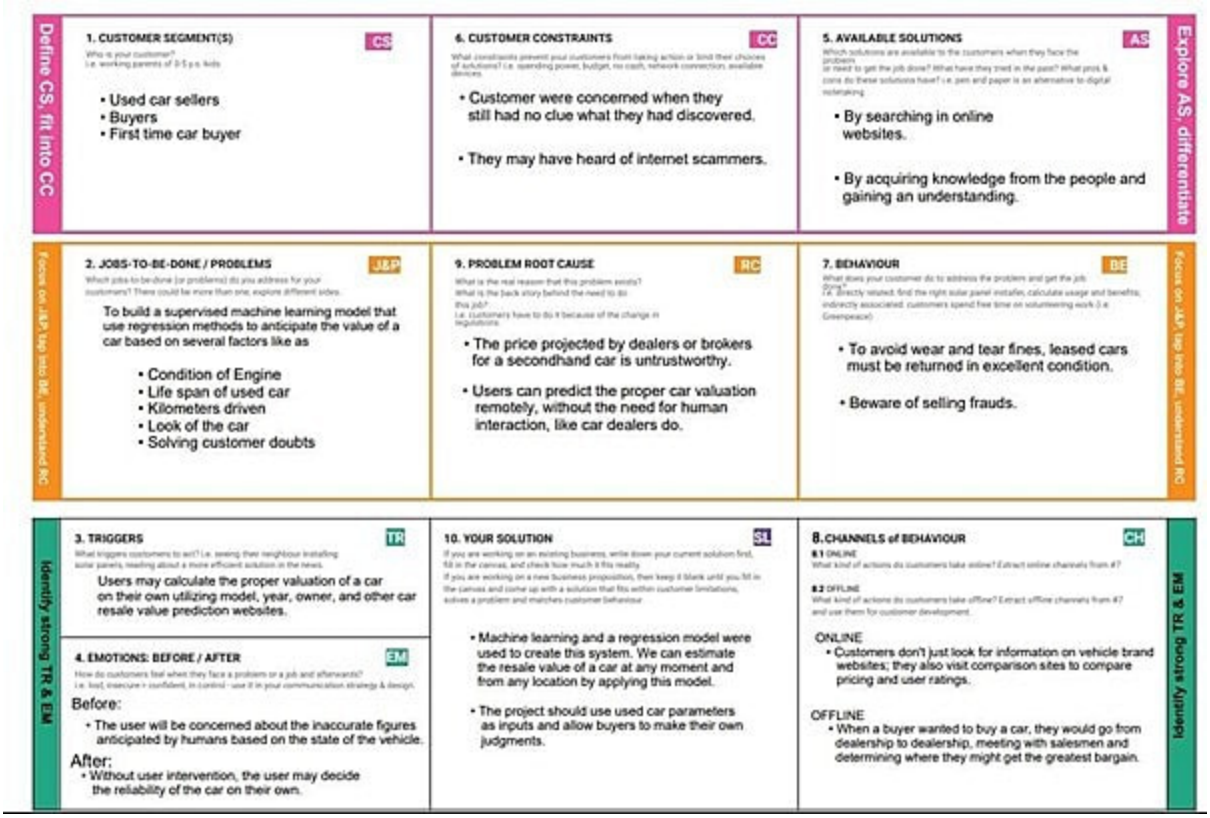
Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict the resale value of second hand car or used car considering its features.
2.	Idea / Solution description	To develop a Machine learning algorithm which predicts the resale value of any used car which is shown in web design.
3.	Novelty / Uniqueness	The model predicts the resale value of car with high accuracy.
4.	Social Impact / Customer Satisfaction	A good platform with more reliability and portability.
5.	Business Model (Revenue Model)	The model deployed in cloud so anyone can access it anywhere and anytime.
6.	Scalability of the Solution	It is a web page model so it can be viewed and accessed in both computer as well as mobile phones.

3.4 Problem solution fit



4. REQUIREMENT ANALYSIS

4.1 Functional Requirements

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via website
FR-3	Car Registration	Registering the car details
FR-4	Value Prediction	Predicting the car resale value

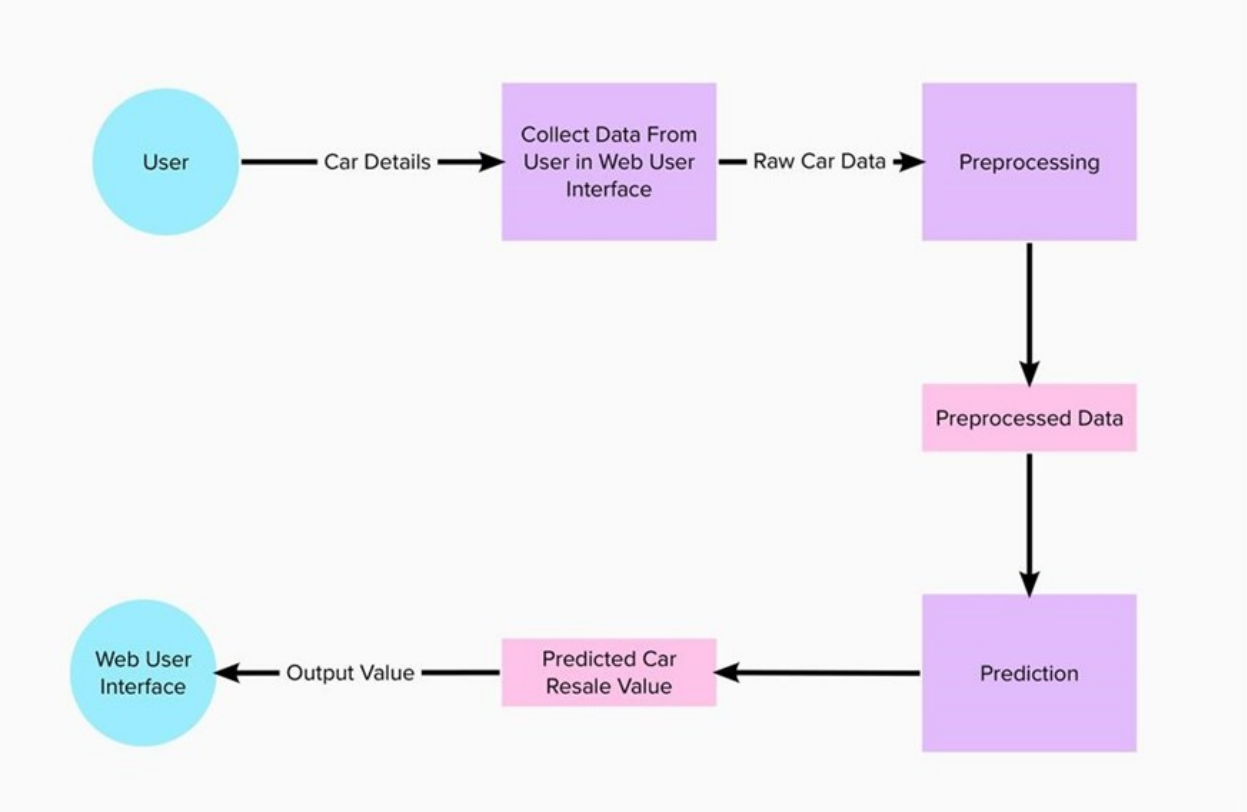
4.2 Non-Functional requirements

Following are the non-functional requirements of the proposed solution.

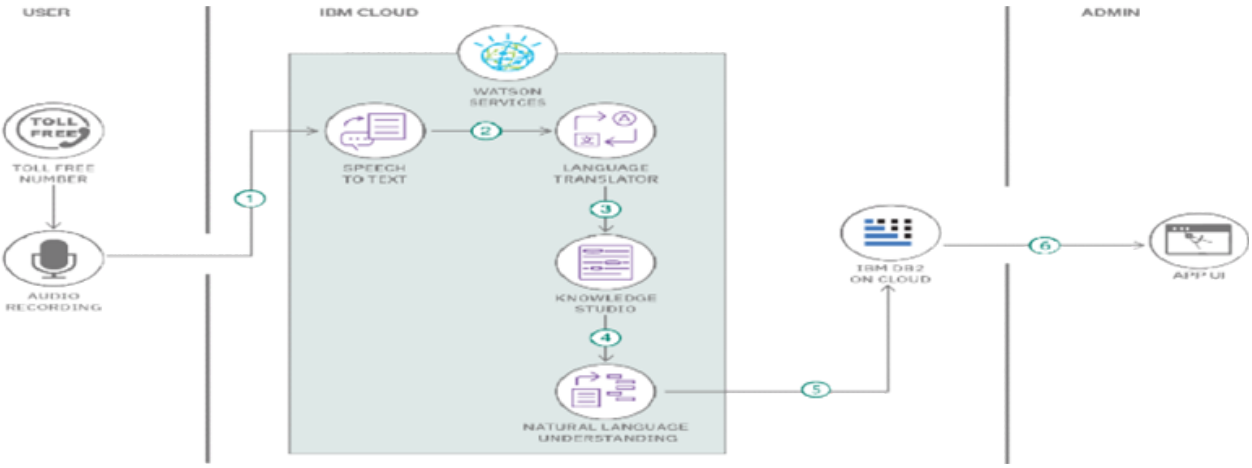
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Predicting the resale value
NFR-2	**Security**	Providing security to the website
NFR-3	**Reliability**	Providing high reliability by predicting values for different types of cars
NFR-4	**Performance**	Providing high performance by using some machine learning techniques
NFR-5	**Availability**	It is used for all types of cars
NFR-6	**Scalability**	Predicting values for different types of cars

5. PROJECT DESIGN

5.1 Data Flow Diagrams



5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement	User Story	User Story / Task	Acceptance criteria	Priority	Release

	(Epic)	**Number**				
Customer (Desktop user)	Home Page	USN-1	As a user, I can view the home page of the web application.	I can view the homepage	Low	Sprint-1
Customer (Desktop user)	Data Entry	USN-2	As a user, I can enter my car details in the application.	I can enter the car details	Medium	Sprint-2
Customer (Desktop user)	View car Resale value	USN-3	As a user, I can view the resale value of my car.	I can view my car's resale value	Medium	Sprint-3
Customer (Desktop user)	Resale Value Prediction	USN-4	As a user, I expect the application to predict the resale value of my car.	I expect the application to predict my car resale price	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Title	Description	Date
Literature Survey a Information Gathering nd	Gathering Information by referring the technical papers, research publications etc.	3 September 2022
Prepare Empathy Map	To capture user pain and gains Prepare List of Problem Statement	10 September 2022
Ideation	Prioritize a top 3 ideas based on feasibility and Importance	17 September 2022
Proposed Solution	Solution include novelty, feasibility, business model, social impact and scalability of solution	24 September 2022
Problem Solution Fit	Solution fit document	1 October 2022
Solution Architecture	Solution Architecture	1 October 2022
Customer Journey	To Understand User Interactions and experiences with application	8 October 2022
Functional Requirement	Prepare functional Requirement	12 October 2022
Data flow Diagrams	Data flow diagram	12 October 2022
Technology Architecture	Technology Architecture diagram	12 October 2022
Milestone & sprint delivery plan	Activity what we done &further plans	22 October 2022
Project Development-Delivery of sprint 1,2,3 &4	Develop and submit the developed code by testing it	24 October 2022 – 19 November 2022

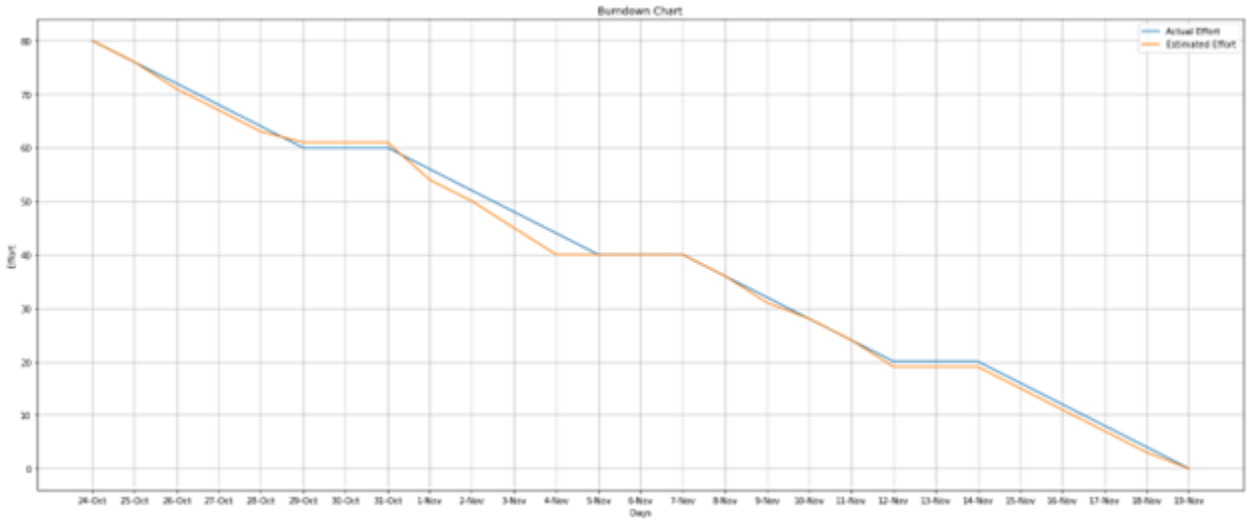
6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home Page	USN-1	As a user, I can view the home page of the web application.	20	Low	PRABIN KUMAR K.K
Sprint-2	Data Entry	USN-2	As a user, I can enter my car details in the application.	20	Medium	AJAY S
Sprint-3	Car resale value display	USN-3	As a user, I can view the resale value of my car.	20	Medium	NIHAAL M.S
Sprint-4	Resale Value Prediction	USN-4	As a user, I expect the application to predict the resale value of my car.	20	Medium	ABILASH KUMAR M

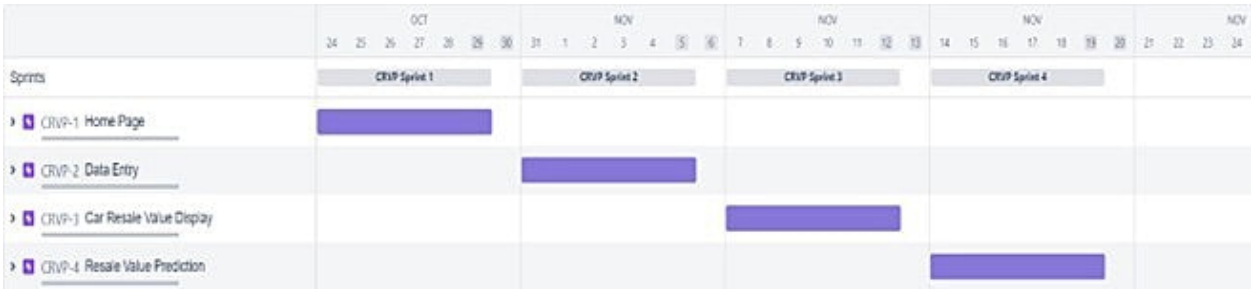
6.3 Project Tracker

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	12 Nov 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	19 Nov 2022

6.4 Burn down Chart



6.5 Reports from JIRA



7. CODING & SOLUTIONING

7.1Home Page

Displays the home page of the application.

Code

1) car.html

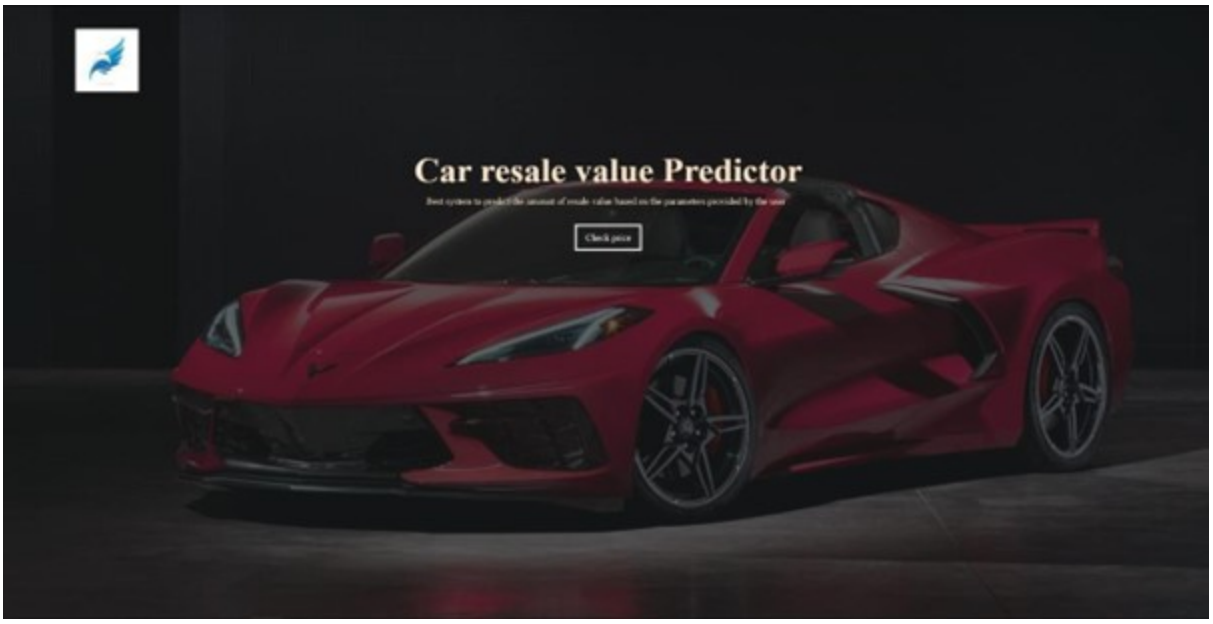
```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Car Resale Value Predicting Application</title>
<link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
<link rel="stylesheet" href="../static/css/style.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<section class="header">
<nav>
<a href="/"><img src="../static/Images/logo.png" width="100" height="100"></a>

</nav>
<div class="text-box">
<h1>Car resale value Predictor</h1>
<p>Best system to predict the amount of resale value based on the parameters provided
by the user .</p>
<a href="./predict_page" class="visit-btn ">Check price</a> </div>
</section>

</body>
</html>
```

2) style.css

```
*{ margin: 0; padding: 0;
}

.header
{ min-height: 100vh; width: 100%; background-image: linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car1.png); background-position:
center; background-size: cover; position: relative;
```

```
}
nav
{
  display:flex;   padding: 2% 6%;   justify-content: space-between;   align-items: center;
}
.nav-links
{
flex: 1;   text-align: right;
}
.nav-links ul li
{
 list-style: none;   display: inline-block;   padding: 8px 12px;   position: relative;
}
.nav-links ul li a
{
color:white;   text-decoration: none;   font-size: 13px;
}
text-box
{
text-align: center;   position: relative;   color:  #FFE4C4;   top:50%;
}
.text-box h1
{
margin-top: 50px;   font-size: 55px;
}
.text-box p
{
  margin: 10px 0 40px;   font-size: 15px;
}
.visit-btn
{
  display: inline;   border: 3px solid #fff;   padding:10px 14px;   font-size: 15px;   background:
transparent;   color: white;   text-decoration:none;

}
```

Output:



7.2 Data Entry Page

Allows user to enter the details about the car for which the resale value is to be predicted.

Code:

1) value.html

```
  <!DOCTYPE html>
  <html lang="en" dir="ltr">
  <head>
<link rel="stylesheet" href="../static/css/value.css">
<title>Car Resale Value Predicting Application</title>
<link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
<script  src="https://kit.fontawesome.com/b9b6bac803.js" crossorigin="anonymous"></script>
<link     rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">
<style>
..... table, th, td {
.....   padding: 10px;
..... }
..... </style>
</head>
<body>
<div class="container">
..... <div class="header">
..... <h1>Get the Accurate Resale Value of Your Car</h1>
..... </div>
.....<form action="http://localhost:5000/predict" class="form">
..... <div class="form-control">
..... <label for="year" padding:10px>Registration year : </label>
<input id="year" maxlength="50" name="regyear" type="text" autocomplete="off"/>
..... <i class="fas fa-check-circle"></i>
..... <i class="fas fa-exclamation-circle"></i>
..... <span></span>
..... </div>
..... <div class="form-control">
..... <label for="month">Registration Month : </label>
<input   id="month"   maxlength="50"   name="regmonth"   type="text"
autocomplete="off"/>
..... <i class="fas fa-check-circle"></i>
..... <i class="fas fa-exclamation-circle"></i>
..... <span></span>
..... </div>
..... <div class="form-control">
.....<label for="power">Power of car in PS: </label>
```

<input id="power" maxlength="50" name="powerps" type="text" autocomplete="off"/>
.....
.....
.....
.....
.....
..... <label for="kilometer">Kilometers that car have driven : </label>
<input id="kilometer" maxlength="50" name="kms" type="text" autocomplete="off"/>
.....
.....
.....
.....
.....
.....
..... <h3>Gear Type</h3>
..... <table style="width:50%">
..... <tr>
.....
..... <th></th>
.....
..... <th></th>
.....
..... </tr>
..... <tr>
.....
..... <td>Manual</td>
.....
<td><input type="radio" name="geartype" value="manual" id="manual" /></td>
..... </tr>
..... <tr>
.....
..... <td>Automatic</td>
.....
<td><input type="radio" name="geartype" value="automatic" id="automatic" /></td>
..... </tr>
..... <tr>
.....
..... <td>Not mentioned</td>
.....

```
<td><input type="radio" name="geartype" value="not-declared" id="not" /></td>
..... </tr>
.....  </table>
..... <i class="fas fa-check-circle"></i>
..... <i class="fas fa-exclamation-circle"></i>
..... <span></span>
..... </div>
..... <div class="form-control">
.....
..... <h3>Your car is repaired or damaged :</h3>
.....<table
style="width:50%">
.....
..... <tr>
        .....
..... <th></th>
        .....
..... <th></th>
.....
.....
..... </tr>
.....
..... <tr>
        .....
..... <td>Yes</td>
        .....
..... <td><input type="radio" name="damage" value="yes" id="yes"/></td>
.....
..... </tr>
.....
..... <tr>
        .....
..... <td>No</td>
        .....
.....  <td><input type="radio" name="damage" value="no" id="no"/></td>
.....
..... </tr>
.....
..... <tr>
        .....
..... <td>Not Declared</td>
```

..... <td><input type="radio" name="damage" value="not-declared" id="notdec"/></td>
.....
..... </tr>
.....
</table>
.....
.....
.....
.....
.....
..... <label for="model">Model Type : </label>
..... <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>
.....<option value="fortwo">Fortwo </option>
..... <option value="1er">1er </option>
..... <option value="b_klasse">B Klasse </option>
.....<option value="signum">Signum </option>
.....<option value="astra">Astra </option>
..... <option value="a8">A8 </option>
..... <option value="jetta">Jetta </option>
..... <option value="fiesta">Fiesta </option>
..... <option value="c_klasse">C Klasse </option>
..... <option value="micra">Micra </option>
..... <option value="vito">Vito </option>
..... <option value="sprinter">Sprinter </option>
..... <option value="156">156 </option>
..... <option value="escort">Escort </option>
..... <option value="forester">Forester </option>
..... <option value="xc_reihe">Xc Reihe </option>
..... <option value="scenic">Scenic </option>
..... <option value="a4">A4 </option>
..... <option value="a1">A1 </option>
..... <option value="insignia">Insignia </option>
..... <option value="combo">Combo </option>
..... <option value="focus">Focus </option>
..... <option value="tt">Tt </option>
..... <option value="a6">A6 </option>
..... <option value="jazz">Jazz </option>
.....<option value="omega">Omega </option>
..... <option value="slk">Slk </option>
..... <option value="7er">7er </option>
..... <option value="80">80 </option>
..... <option value="147">147 </option>
..... <option value="glk">Glk </option>
..... <option value="100">100 </option>
..... <option value="z_reihe">Z Reihe </option>
..... <option value="sportage">Sportage </option>
.....<option value="sorento">Sorento </option>
..... <option value="v40">V40 </option>
..... <option value="5er">5er </option>
..... <option value="ibiza">Ibiza </option>
..... <option value="3er">3er </option>
..... <option value="mustang">Mustang </option>

..... <option value="eos">Eos </option>
..... <option value="touran">Touran </option>
.....<option value="getz">Getz </option>
..... <option value="a3">A3 </option>
..... <option value="almera">Almera </option>
..... <option value="megane">Megane </option>
..... <option value="7er">7er </option>
..... <option value="1er">1er </option>
..... <option value="lupo">Lupo </option>
..... <option value="r19">R19 </option>
..... <option value="zafıra">Zafıra </option>
..... <option value="caddy">Caddy </option>
..... <option value="2_reihe">2 Reihe </option>
.....<option value="mondeo">Mondeo </option>
..... <option value="cordoba">Cordoba </option>
..... <option value="colt">Colt </option>
..... <option value="impreza">Impreza </option>
..... <option value="vectra">Vectra </option>
..... <option value="berlingo">Berlingo </option>
..... <option value="80">80 </option>
..... <option value="m_klasse">M Klasse </option>
..... <option value="tiguan">Tiguan </option>
..... <option value="i_reihe">I Reihe </option>
..... <option value="espace">Espace </option>
..... <option value="sharan">Sharan </option>
..... <option value="6_reihe">6 Reihe </option>
..... <option value="panda">Panda </option>
..... <option value="up">Up </option>
..... <option value="seicento">Seicento </option>
..... <option value="ceed">Ceed </option>
..... <option value="5_reihe">5 Reihe </option>
..... <option value="yeti">Yeti </option>
..... <option value="octavia">Octavia </option>
..... <option value="mii">Mii </option>
..... <option value="rx_reihe">Rx Reihe </option>
..... <option value="6er">6er </option>
..... <option value="modus">Modus </option>
..... <option value="fox">Fox </option>
..... <option value="matiz">Matiz </option>
..... <option value="beetle">Beetle </option>
..... <option value="c1">C1 </option>

.....<option value="rio">Rio </option>
..... <option value="touareg">Touareg </option>
..... <option value="logan">Logan </option>
..... <option value="spider">Spider </option>
..... <option value="cuore">Cuore </option>
..... <option value="s_max">S Max </option>
..... <option value="a2">A2 </option>
..... <option value="x_reihe">X Reihe </option>
..... <option value="a5">A5 </option>
.....<option value="galaxy">Galaxy </option>
..... <option value="c3">C3 </option>
..... <option value="viano">Viano </option>
.....<option value="s_klasse">S Klasse </option>
..... <option value="1_reihe">1 Reihe </option>
..... <option value="avensis">Avensis </option>
..... <option value="sl">Sl </option>
..... <option value="roomster">Roomster </option>
..... <option value="q5">Q5 </option>
..... <option value="kaefer">Kaefer </option>
..... <option value="santa">Santa </option>
..... <option value="cooper">Cooper </option>
..... <option value="leon">Leon </option>
..... <option value="4_reihe">4 Reihe </option>
..... <option value="500">500 </option>
..... <option value="laguna">Laguna </option>
.....<option value="ptcruiser">Ptcruiser </option>
..... <option value="clk">Clk </option>
..... <option value="primera">Primera </option>
..... <option value="exeo">Exeo </option>
..... <option value="159">159 </option>
..... <option value="transit">Transit </option>
.....<option value="juke">Juke </option>
..... <option value="qashqai">Qashqai </option>
..... <option value="carisma">Carisma </option>
..... <option value="accord">Accord </option>
..... <option value="corolla">Corolla </option>
..... <option value="lanos">Lanos </option>
..... <option value="phaeton">Phaeton </option>
.....<option value="boxster">Boxster </option>
.....<option value="verso">Verso </option>
..... <option value="swift">Swift </option>

..... <option value="rav">Rav </option>
..... <option value="kuga">Kuga </option>
..... <option value="picanto">Picanto </option>
..... <option value="kalos">Kalos </option>
..... <option value="superb">Superb </option>
..... <option value="stilo">Stilo </option>
..... <option value="alhambra">Alhambra </option>
..... <option value="911">911 </option>
..... <option value="mx_reihe">Mx Reihe </option>
..... <option value="m_reihe">M Reihe </option>
..... <option value="roadster">Roadster </option>
..... <option value="ypsilon">Ypsilon </option>
..... <option value="cayenne">Cayenne </option>
..... <option value="galant">Galant </option>
..... <option value="justy">Justy </option>
..... <option value="90">90 </option>
..... <option value="sirion">Sirion </option>
..... <option value="crossfire">Crossfire </option>
..... <option value="6_reihe">6 Reihe </option>
..... <option value="agila">Agila </option>
..... <option value="duster">Duster </option>
..... <option value="cr_reihe">Cr Reihe </option>
..... <option value="v50">V50 </option>
..... <option value="discovery">Discovery </option>
..... <option value="c_reihe">C Reihe </option>
..... <option value="v_klasse">V Klasse </option>
..... <option value="yaris">Yaris </option>
..... <option value="c5">C5 </option>
..... <option value="aygo">Aygo </option>
..... <option value="cc">Cc </option>
..... <option value="carnival">Carnival </option>
..... <option value="fusion">Fusion </option>
..... <option value="bora">Bora </option>
..... <option value="forfour">Forfour </option>
..... <option value="100">100 </option>
..... <option value="cl">Cl </option>
..... <option value="tigra">Tigra </option>
..... <option value="156">156 </option>
..... <option value="300c">300c </option>
..... <option value="100">100 </option>
..... <option value="147">147 </option>

..... <option value="q3">Q3 </option>
..... <option value="spark">Spark </option>
..... <option value="v70">V70 </option>
..... <option value="x_type">X Type </option>
..... <option value="5_reihe">5 Reihe </option>
.....<option value="ducato">Ducato </option>
..... <option value="s_type">S Type </option>
..... <option value="x_trail">X Trail </option>
..... <option value="toledo">Toledo </option>
.....<option value="altea">Altea </option>
..... <option value="7er">7er </option>
.....<option value="voyager">Voyager </option>
..... <option value="calibra">Calibra </option>
..... <option value="bravo">Bravo </option>
..... <option value="range_rover">Range Rover </option>
..... <option value="antara">Antara </option>
.....<option value="tucson">Tucson </option>
..... <option value="q7">Q7 </option>
..... <option value="citigo">Citigo </option>
..... <option value="jimny">Jimny </option>
..... <option value="cx_reihe">Cx Reihe </option>
..... <option value="wrangler">Wrangler </option>
..... <option value="lybra">Lybra </option>
..... <option value="range_rover_sport">Range Rover Sport </option>
..... <option value="lancer">Lancer </option>
..... <option value="159">159 </option>
..... <option value="freelander">Freelander </option>
..... <option value="captiva">Captiva </option>
..... <option value="c2">C2 </option>
..... <option value="500">500 </option>
..... <option value="range_rover_evoque">Range Rover Evoque </option>
..... <option value="sandero">Sandero </option>
..... <option value="note">Note </option>
..... <option value="900">900 </option>
..... <option value="147">147 </option>
..... <option value="defender">Defender </option>
..... <option value="cherokee">Cherokee </option>
..... <option value="clubman">Clubman </option>
..... <option value="samara">Samara </option>
..... <option value="2_reihe">2 Reihe </option>
..... <option value="1er">1er </option>

..... <option value="3er">3er </option>
..... <option value="601">601 </option>
..... <option value="3_reihe">3 Reihe </option>
..... <option value="4_reihe">4 Reihe </option>
..... <option value="5er">5er </option>
..... <option value="6_reihe">6 Reihe </option>
..... <option value="legacy">Legacy </option>
.....<option value="pajero">Pajero </option>
..... <option value="auris">Auris </option>
..... <option value="niva">Niva </option>
..... <option value="5_reihe">5 Reihe </option>
..... <option value="s60">S60 </option>
..... <option value="nubira">Nubira </option>
..... <option value="vivaro">Vivaro </option>
..... <option value="g_klasse">G Klasse </option>
..... <option value="lodgy">Lodgy </option>
..... <option value="850">850 </option>
..... <option value="serie_2">Serie 2 </option>
..... <option value="6er">6er </option>
..... <option value="charade">Charade </option>
..... <option value="croma">Croma </option>
..... <option value="outlander">Outlander </option>
..... <option value="gl">Gl </option>
..... <option value="doblo">Doblo </option>
..... <option value="musa">Musa </option>
..... <option value="amarok">Amarok </option>
..... <option value="156">156 </option>
..... <option value="move">Move </option>
..... <option value="9000">9000 </option>
..... <option value="v60">V60 </option>
..... <option value="145">145 </option>
.....<option value="aveo">Aveo </option>
..... <option value="200">200 </option>
..... <option value="300c">300c </option>
..... <option value="b_max">B Max </option>
..... <option value="delta">Delta </option>
..... <option value="terios">Terios </option>
..... <option value="rangerover">RangeRover </option>
..... <option value="90">90 </option>
..... <option value="materia">Materia </option>
..... <option value="kalina">Kalina </option>

..... <option value="elefantino">Elefantino </option>
..... <option value="i3">I3 </option>
..... <option value="kappa">Kappa </option>
..... <option value="serie_3">Serie 3 </option>
..... <option value="48429">48429 </option>
..... <option value="serie_1">Serie 1 </option>
..... <option value="discovery_sport">Discovery Sport </option>
..... </select>
.....
.....
.....
.....
.....
..... <label for="brand">Brand :</label>
..... <select name="brand" id="brand">
... <option value="" disabled selected hidden>Choose Brand Name...</option>
..... <option value="volkswagen">Volkswagen </option>
..... <option value="audi">Audi </option>
..... <option value="jeep">Jeep </option>
..... <option value="skoda">Skoda </option>
..... <option value="bmw">Bmw </option>
.....<option value="peugeot">Peugeot </option>
..... <option value="ford">Ford </option>
..... <option value="mazda">Mazda </option>
..... <option value="nissan">Nissan </option>
..... <option value="renault">Renault </option> .....
<option value="mercedes_benz">Mercedes Benz </option>
..... <option value="opel">Opel </option>
..... <option value="seat">Seat </option>
..... <option value="citroen">Citroen </option>
..... <option value="honda">Honda </option>
..... <option value="fiat">Fiat </option>
..... <option value="mini">Mini </option>
..... <option value="smart">Smart </option>
..... <option value="hyundai">Hyundai </option>
..... <option value="sonstige_autos">Sonstige Autos </option>
..... <option value="alfa_romeo">Alfa Romeo </option>
..... <option value="subaru">Subaru </option>
..... <option value="volvo">Volvo </option>
..... <option value="mitsubishi">Mitsubishi </option>
..... <option value="kia">Kia </option>

..... <option value="suzuki">Suzuki </option>
..... <option value="lancia">Lancia </option>
..... <option value="porsche">Porsche </option>
..... <option value="toyota">Toyota </option>
..... <option value="chevrolet">Chevrolet </option>
..... <option value="dacia">Dacia </option>
..... <option value="daihatsu">Daihatsu </option>
..... <option value="trabant">Trabant </option>
..... <option value="saab">Saab </option>
..... <option value="chrysler">Chrysler </option>
..... <option value="jaguar">Jaguar </option>
..... <option value="daewoo">Daewoo </option>
.....<option value="rover">Rover </option>
..... <option value="land_rover">Land Rover </option>
..... <option value="lada">Lada </option>
..... </select>
.....
.....
.....
.....
.....
..... <label for="fuelType">Fuel Type :</label>
..... <select name="fuelType" id="fuel"> ..... <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>
.....
.....
.....
.....
.....
..... <label for="vehicletype">Vehicle type: </label>
..... <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>
.....
.....
.....
.....
..... <input type="submit" id="submit"></input>
..... </form>

</body>
</html>

2) value.css

```
*{ padding:0px; margin:0; box-sizing:border-box; font-family: cursive; font-weight: bold;
color: #E74C3C;
}

body{
                background-image:
linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car2.png);
..... min-height:100vh;
.....display:flex;
..... justify-content:center;
.....align-items:center;
}

.header{
..... color:Black;
..... text-align:center;
..... padding:10px 0px 10px 100px; }

#model{
```

..... width:500px;
..... color: black;
}

#brand{
..... width:500px;
..... color: black;
}

#fuel{
..... width:500px;
..... color: black;
}

#vehicle{
..... width:500px;
..... color: black;
}

.form{
..... padding:30px 40px; }

.form-control{
..... margin-bottom:10px;
..... padding-bottom: 20px;
..... position:relative;
..... margin-left: 100px; }

.form-control label{
..... display:block;
..... margin-bottom:5px; }

.form-control input{
..... border: 2px solid #f0f0f0;
..... width:80%;
..... font-size :.8rem;
..... padding:5px;
.....display:inline-table;
}

.form-control i{

..... position:absolute;
..... right:20px;
..... top:35px;
..... visibility:hidden;
}

.form-control span{
..... position:absolute;
..... left:0;
..... bottom:0;
..... visibility:hidden;
..... font-weight:bolder;
..... font-style:italic;
..... font-size:1rem;
}

.form-control.success input{
..... border-color:#2ecc71; }

.form-control.error input{
..... border-color:#e743c3; }

.form-control.error span{
..... color:red;
..... visibility:visible; }

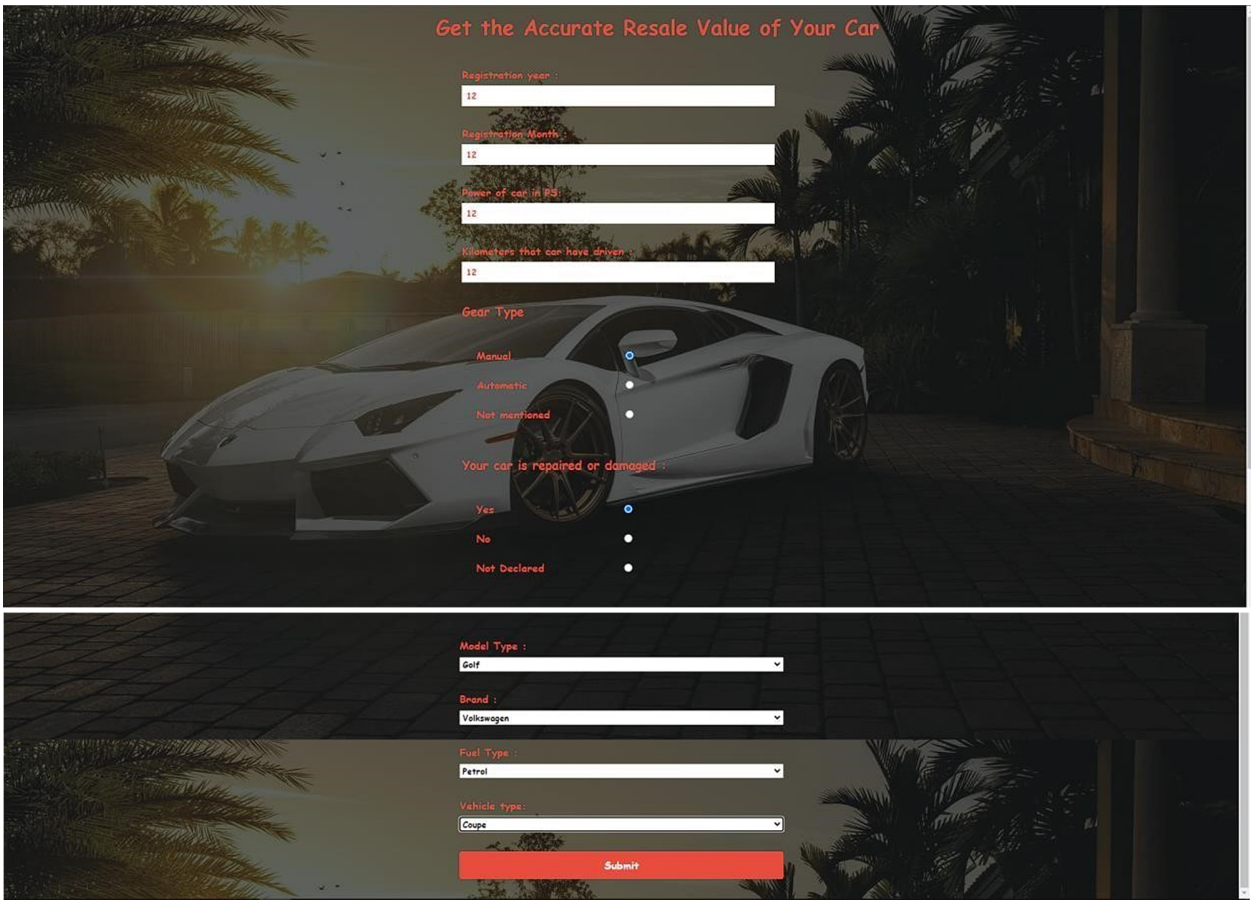
.form-control.success i.fa-check-circle {
..... border-color:#2ecc71;
..... visibility:visible;
}

.form-control.error i.fa-exclamation-circle {
..... border-color:#e73c3c;
..... visibility:visible;
}

.form #submit{
..... background-color:#E74C3C;
..... border:none;

```
..... outline:none;
..... color:white;
..... width:500px;
..... border-radius:4px;
..... padding:10px;
..... cursor:pointer;
..... transition:all .5s;
..... font-size:1rem;
..... margin-left: 100px;
}

.form #submit:hover{
..... background-color:#6441a5; }

.form-control #manual{
..... padding-bottom: 20%; }
```

Output



7.3 Output Display Page

The predicted resale car value is displayed in this page.

Code

1) predict.html

```
   <!DOCTYPE html>
   <html lang="en">
   <head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="../static/css/predict.css">
   <title>Car Resale Value Predicting Application</title>
   <link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
</head>
<body>
..... <section class="header">       <nav>
       <a href="/"><img src="../static/Images/logo.png" width="100" height="100"></a>
     </nav>
       <div class="text-box">
         <h1>The Predicted Car Resale Value is </h1>
                        .....
<h1>{{predict}}</h1>
       </div>
   </section>
.....
.....
</body>
</html>
```

2) predict.css

```
   .header{   min-height: 100vh;   width: 100%;
                        background-image:
linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car3.jpg);   background-position:
center;   background-size: cover;   position: relative;
}

.text-box{   text-align: center;   position: relative;   color:  #FFE4C4;   top:50%;
}
.text-box h1{   margin-top: 50px;   font-size: 55px;
}
```

```
.text-box p{   margin: 10px 0 40px;   font-size: 15px;
}

body{
..... margin: 0;
}

nav{   display:flex;   padding: 2% 6%;
  justify-content: space-between;   align-items: center;
}
```

OUTPUT:

7.4 Model Selection & Hyperparameter Tuning

Code

```
import pandas as pd import numpy as np
```

```
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV from sklearn.metrics
import mean_absolute_error, mean_squared_error, r2_score
import pickle
import wandb

#regression models
                from      sklearn.ensemble      import      BaggingRegressor,
RandomForestRegressor,
HistGradientBoostingRegressor, ExtraTreesRegressor from xgboost.sklearn import
XGBRegressor from lightgbm import LGBMRegressor
wandb.login(key='b75e0564aba32dce859c60044418df71ce7389a8')

data          =          pd.read_csv('../input/naalaiya-thiran/Preprocessed/autos_preprocessed.csv',
header=0, sep=',', encoding='Latin1')

labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

mapper = {} for i in labels:
   mapper[i] = LabelEncoder()     mapper[i].fit(data[i])     tr = mapper[i].transform(data[i])
np.save(str('classes'+i+'.npy'), mapper[i].classes_)     data.loc[:, i+'_labels'] = pd.Series(tr,
index=data.index)

labeled = data[['price', 'yearOfRegistration','powerPS','kilometer','monthOfRegistration']
+[x+"_labels" for x in labels]]

print(labeled.columns)

def find_scores(Y_actual, Y_pred, X_train):
   mae = mean_absolute_error(Y_actual, Y_pred)     mse = mean_squared_error(Y_actual,
Y_pred)
   rmse = np.sqrt(mse)     rmsle = np.log(rmse)     r2 = r2_score(Y_actual, Y_pred)     n, k =
X_train.shape
   adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))

   wandb.log({"mae": mae, "mse": mse, 'rmse':rmse, 'rmsle':rmsle, 'r2':r2, 'adj_r2':adj_r2_score})

def bagging_regressor():     config_defaults = {
            'n_estimators':100,
            'max_samples':0.4,
            'bootstrap':True,
```

```
            'random_state':42
        }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = BaggingRegressor(       n_estimators=config.n_estimators,
bootstrap=config.bootstrap,       max_samples=config.max_samples,
      random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

bagging_regressor_configs = {
    "name":'BaggingRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "n_estimators": {
            "values": [100, 200, 300]
        },
        "max_samples": {
            "values": [0.4,0.5, 0.6]
        }
    }
}

sweep_id     =          wandb.sweep(sweep=bagging_regressor_configs,
project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=bagging_regressor)

def random_forest_regressor():     config_defaults = {
```

```
            'n_estimators':100,
            'max_samples':0.4,
            'criterion':'squared_error',
            'bootstrap': True,
            'random_state':42
        }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = RandomForestRegressor(       n_estimators=config.n_estimators,       criterion =
config.criterion,       bootstrap=config.bootstrap,       max_samples=config.max_samples,
      random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

random_forest_configs = {     "name":'RandomForestRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "n_estimators": {
            "values": [100, 200, 300]
        },
        "max_samples": {
            "values": [0.4,0.5, 0.6]
        }
    }
}

sweep_id = wandb.sweep(sweep=random_forest_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=random_forest_regressor)
```

```
def hist_gradient_boost_regressor():
    config_defaults = {
                'loss':'squared_error',
                'learning_rate': 0.1,
                'max_iter':100,
                'random_state':42
            }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = HistGradientBoostingRegressor(       loss=config.loss,
      learning_rate = config.learning_rate,       max_iter=config.max_iter,       random_state =
config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

hist_gradient_boost_configs = {
    "name":'HistGradientBoostingRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "loss": {
            "values": ['squared_error', 'absolute_error']
        },
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "max_iter": {
            "values": [100,200,300]
```

```
      },
      "random_state": {
         "values": [42]
      }
   }
}

sweep_id    =        wandb.sweep(sweep=hist_gradient_boost_configs,
project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=hist_gradient_boost_regressor)

def extra_tree_regressor():
   config_defaults = {
            'criterion':'squared_error',
            'max_samples':0.4,
            'bootstrap': True,
            'random_state':42
         }
   wandb.init(config=config_defaults)    config = wandb.config

     A.  = labeled.iloc[:,1:].values
     B.  = labeled.iloc[:,0].values.reshape(-1,1)

   X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

   model = ExtraTreesRegressor(      criterion=config.criterion,      bootstrap = config.bootstrap,
max_samples=config.max_samples,
    random_state = config.random_state)

   model.fit(X_train, Y_train)

   Y_pred = model.predict(X_test)

   find_scores(Y_test, Y_pred, X_train)

extra_tree_configs = {
   "name":'ExtraTreesRegressor',
   "method": "grid",
   "metric": {
      "name": "adj_r2",
      "goal": "maximize"
```

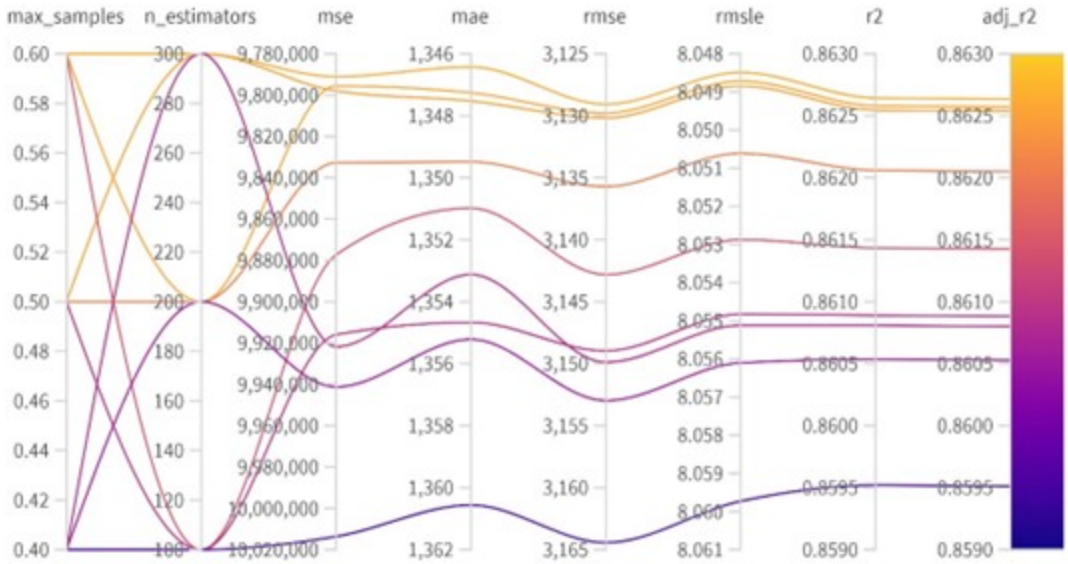
```
  },
  "parameters": {
    "criterion": {
      "values": ['squared_error', 'absolute_error']
    },
    "max_samples": {
      "values": [0.4,0.5, 0.6] }
  }
}

sweep_id = wandb.sweep(sweep=extra_tree_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=extra_tree_regressor)

def XGB_regressor():    config_defaults = {
            'learning_rate':0.1,
            'n_estimators': 500,
            'booster':'gbtree',
            'eta':0.01,
            'random_state':42
        }
  wandb.init(config=config_defaults)    config = wandb.config

    A.  = labeled.iloc[:,1:].values
    B.  = labeled.iloc[:,0].values.reshape(-1,1)

  X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

  model = XGBRegressor(      learning_rate=config.learning_rate,      n_estimators =
config.n_estimators,
    random_state = config.random_state)

  model.fit(X_train, Y_train)

  Y_pred = model.predict(X_test)

  find_scores(Y_test, Y_pred, X_train)

extra_tree_configs = {    "name":'XGBRegressor',
  "method": "grid",
  "metric": {
    "name": "adj_r2",
```

```
      "goal": "maximize"
    },
    "parameters": {
      "learning_rate": {
        "values": [0.01, 0.03, 0.05, 0.07]
      },
"n_estimators": {
        "values": [100,200,300]
      },
      "booster": {
        "values": ['gbtree','gblinear']
      },
      "eta": {
        "values": [0.01, 0.03, 0.05, 0.07]
      }
    }
}

sweep_id = wandb.sweep(sweep=extra_tree_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=XGB_regressor)

def LGBM_regressor():    config_defaults = {
            'objective':'root_mean_squared_error',
            'reg_sqrt': True,
            'metric':'rmse',
            'random_state':42
        }
    wandb.init(config=config_defaults)    config = wandb.config

     A.  = labeled.iloc[:,1:].values
     B.  = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = LGBMRegressor(      learning_rate=config.learning_rate,      n_estimators =
config.n_estimators,
     random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)
```

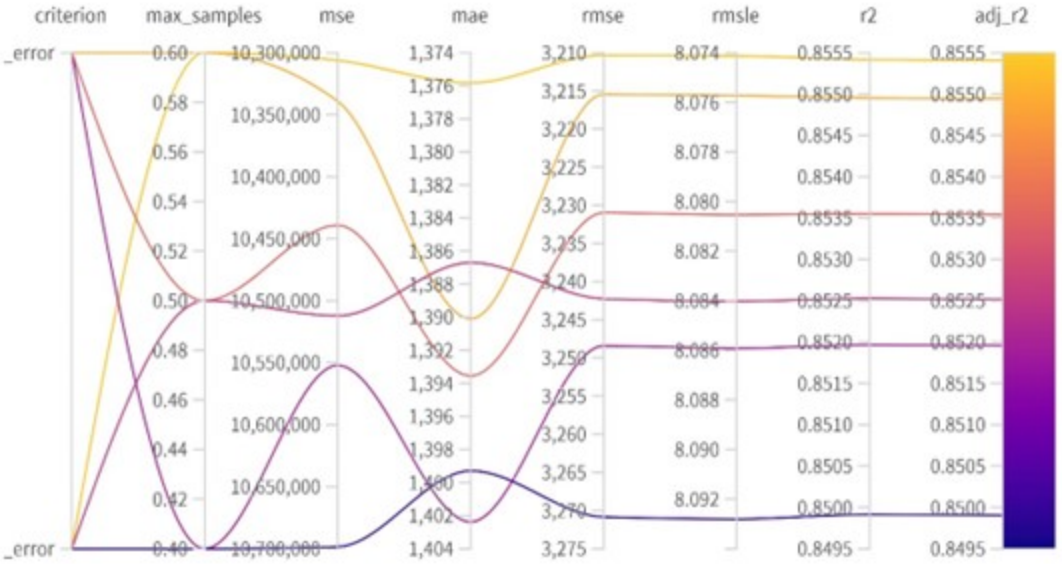
```
    find_scores(Y_test, Y_pred, X_train)

lgbm_configs = {
    "name":'LGBMRegressor',
    "method": "grid",
    "metric": {
"name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "objective": {
            "values": ['root_mean_squared_error']
        },
        "boosting_type": {
            "values": ['gbdt','dart','goss','rf']
        },
        "reg_sqrt": {
            "values": [True]
        },
        "metric": {
            "values": ['rmse']
        },
        "n_estimators": {
            "values": [100,200,300]
        },
        "random_state": {
            "values": [42]
        }
    }
}

sweep_id = wandb.sweep(sweep=lgbm_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=LGBM_regressor
```

Output:

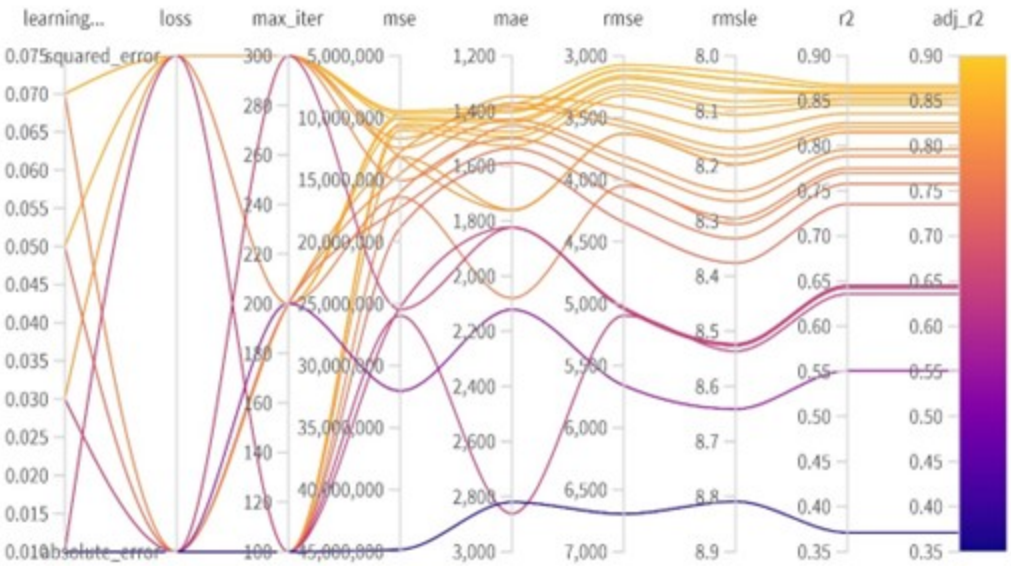
1 Bagging Regressor



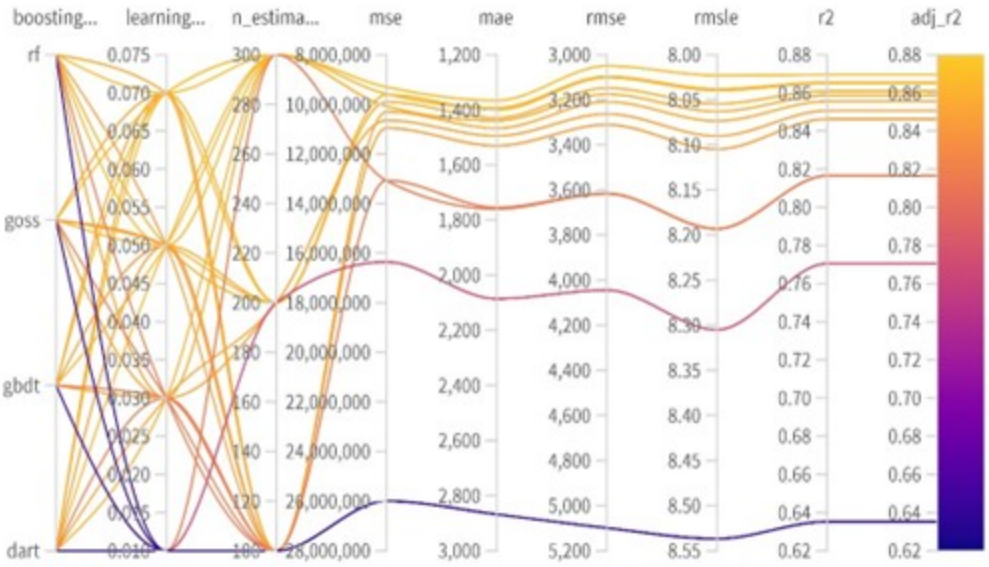
2 Extra Tree Regressor



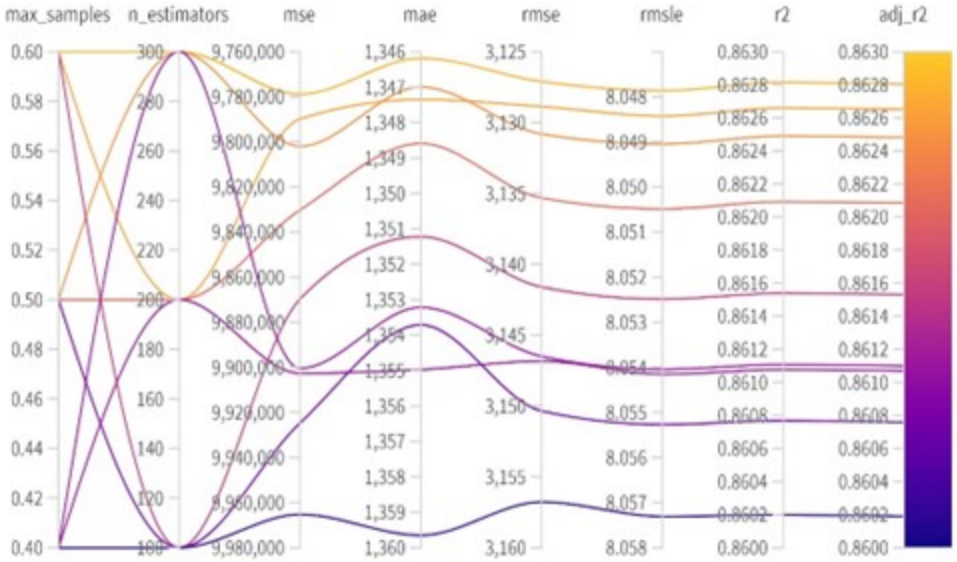
3 HOG Boosting Regressor



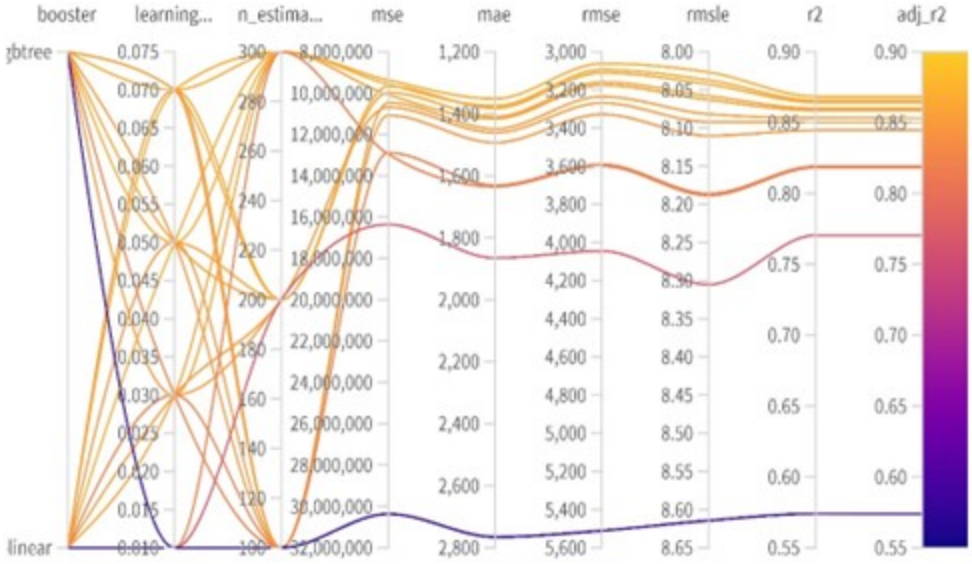
4 LGBM Regressor



5 Random Forest Regressor



6 XGB Regressor



7.5 Flask Integration

```
# Import Libraries
import pandas as pd
import numpy as np
m 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 = "04ZW6LlrLwAfofEU2VHPt69RKCWVc9U1o5LXkAU_66qA" 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')

..... 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_labels','brand_labels', 'fuelType_labels', 'vehicletype_labels']], "values": X}]}

                response_scoring                                =
requests.post('https://us-
south.ml.cloud.ibm.com/ml/v4/deployments/c0f74260-1f5f-43ad-
8d71eb12ef099507/predictions?version=2022-11-13', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken}) .....
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)
```

8. TESTING

8.1 Test Cases Scenarios

1	Verify user is able to see home page?
2	Verify user is able to navigate to data entry page?
3	Verify user is able to see data entry page?
4	Verify user is able to enter values in the fields?
5	Verify user is able to navigate to output display page?
6	Verify user is able to view the output display page?
7	Verify user is able to view the car resale value output in the output display page?

8.2 User Acceptance Testing

	C	D	E	F	G	H	I	J	K	L	M	N
1			Date	19-Nov-22								
2			Team ID	PNT2022TMID34489								
3			Project Name	Project - Car Resale Value Prediction								
4			Maximum Marks	4 marks								
5	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comments**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
6	Home Page	Verify all the UI elements in Home page rendered properly		1.Enter URL and click go 2.Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Nihaal
7	Home Page	Verifiy the Data Entry page can be reachable.		1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button.	-	User should navigate to Data Entry Page	Working as expected	Pass		N		Prabin kumar
8	Data Entry Page	Verify all the UI elements in Data Entry page rendered properly		1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button in the home page 4. Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Abilash kuma
9	Data Entry Page	Verify user is able to enter all values		1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered	2012 12 12 12 Manual Yes Golf Volkswagen Petrol Coupe	User should be able to enter all values in data entry page	Working as expected	Pass		N		Ajay
10	Data Entry Page	Verifiy the Output Display page can be reachable.		1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered 6. Press the submit Button	-	User should navigate to Output Display Page	Working as expected	Pass		N		Ajay
11	Output Display Page	Verify all the UI elements in Output Display page rendered properly		1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered 6. Press the submit Button 7. Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Abilash kuma
				1.Enter URL and click go 2.Verify all the UI elements displayed or not. 3.Press the Check Price button in the home page		Predited Car Resale Value is displayed on the page						

8.3 UAT Report

8.3.1 Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	2	3	1	1	7
Duplicate	1	0	3	0	4
External	2	0	0	1	3
Fixed	2	2	1	2	7
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	2	2	1	5
Totals	7	7	9	6	29

8.3.2 Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Home Page	5	0	0	5
Data Entry Page	15	0	0	15
Output Page	4	0	0	4
Hyper Parameter Tuning	3	0	0	3
Final Model Building	2	0	0	2
Flask Application	10	0	0	10
Train Model on IBM	3	0	0	3
Final Report Output	4	0	0	4

9. RESULTS

9.1 Performance Metrics

S No.	Name	Description
1.	Metrics	**Regression Model**: LGBM Regressor **MAE**: 1327.55 **MSE**: 9492244.28 **RMSE**: 3080.95 **RMSLE**: 8.03 **R2 Score**: 0.8668 **Adjusted R2 Score**: 0.8668
2.	Tune the Model	Hyperparameter Tuning: 1. **Learning Rate**: [0.01, 0.03, 0.05, 0.07] 2. **Boosting Type**: ['gbdt','dart','goss','rf'] 3. **Number of Estimators**: [100,200,300] **Validation Method**: Grid Search Cross Validation **Best Parameters:** a. Learning Rate – 0.07 b. Boosting Type – 'gbdt' c. Number of Estimators - 300

10 ADVANTAGES & DISADVANTAGES

Advantages

1. Application is easy to use
2. User Friendly
3. No Cost
4. No need to commission any agent to get car resale value estimate

Disadvantages

5. User needs to fill every asked detail of the car
6. Doesn’t work for cars from different distributions
7. Not always accurate

11. CONCLUSION

The increased prices of new cars and the financial incapability of the customers to buy them, used Car sales are on a global increase. Therefore, there is an urgent need for a Car Resale Value Prediction system which effectively determines the worthiness of the car in terms of cost. The proposed system is a web application that will help users to determine the accurate price of used cars.

12. FUTURE SCOPE

In future, large historical data of car price can be used to train the model, and which can help improve the estimation of the machine learning model. Moreover, we can build an application for mobile phone platforms like android, iOS for interacting with users. For better performance, we plan to judiciously design deep learning neural networks.

13. APPENDIX

Source Code

User Interfacecar.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<meta charset="utf-8">
<title>Car Resale Value Predicting Application</title>
<link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
<link rel="stylesheet" href="../static/css/style.css">
<link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/fontawesome/4.7.0/css/font-awesome.min.css">
</head>
<body>
<section class="header">
<nav>
<a href="/"><img src="../static/Images/logo.png" width="100" height="100"></a>

</nav>
<div class="text-box">
<h1>Car resale value Predictor</h1>
<p>Best system to predict the amount of resale value based on the parameters provided
by the user .</p>
<a href="./predict_page" class="visit-btn ">Check price</a> </div>
</section>

</body>
</html>

style.css *{ margin: 0; padding: 0;
}

.header{ min-height: 100vh; width: 100%; background-image: linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car1.png); background-position:
center;
background-size: cover; position: relative; } nav{
display:flex; padding: 2% 6%; justify-content: space-between; align-items: center; } .nav-
links{ flex: 1; text-align: right; } .nav-links ul li{ list-style: none; display: inline-block; padding:
```

```
8px 12px;   position: relative; } .nav-links ul li a{   color:white;   text-decoration: none;   font-size:
13px; } .text-box{   text-align: center;   position: relative;   color:  #FFE4C4;   top:50%; } .text-box
h1{   margin-top: 50px;   font-size: 55px; } .text-box p{   margin: 10px 0 40px;   font-size: 15px; }
.visit-btn{   display: inline;   border: 3px solid #fff;   padding:10px 14px;   font-size: 15px;
background: transparent;   color: white;   text-decoration:none;
}
```

value.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<link rel="stylesheet" href="../static/css/value.css">
<title>Car Resale Value Predicting Application</title>
<link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
<script src="https://kit.fontawesome.com/b9b6bac803.js" crossorigin="anonymous"></script>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/4.7.0/css/fontawesome.min.css">
<style>       table, th, td {
              padding: 10px;
             }
             </style>
</head>
<body>
<div class="container">
             <div class="header">
             <h1>Get the Accurate Resale Value of Your Car</h1>
             </div>
             <form action="http://localhost:5000/predict" class="form">
              <div class="form-control">
             <label for="year" padding:10px>Registration year : </label>
             <input id="year" maxlength="50" name="regyear" type="text" autocomplete="off"/>
              <i class="fas fa-check-circle"></i>
              <i class="fas fa-exclamation-circle"></i>
             <span></span>
             </div>
              <div class="form-control">
             <label for="month">Registration Month : </label>
             <input id="month" maxlength="50" name="regmonth" type="text" autocomplete="off"/>
              <i class="fas fa-check-circle"></i>
              <i class="fas fa-exclamation-circle"></i>
             <span></span>
```

```
            </div>
             <div class="form-control">
            <label for="power">Power of car in PS: </label>
            <input id="power" maxlength="50" name="powerps" type="text" autocomplete="off"/>
<i class="fas fa-check-circle"></i>
<i class="fas fa-exclamation-circle"></i>
<span></span>
</div>
<div class="form-control">
<label for="kilometer">Kilometers that car have driven : </label>
<input id="kilometer" maxlength="50" name="kms" type="text" autocomplete="off"/>
<i class="fas fa-check-circle"></i>
<i class="fas fa-exclamation-circle"></i>
<span></span>
</div>
             <div class="form-control">
                    <h3>Gear Type</h3>
            <table style="width:50%">
                    <tr>
                            <th></th>
                            <th></th>

                    </tr>
                    <tr>
                            <td>Manual</td>
                            <td><input type="radio" name="geartype" value="manual" id="manual"
/></td>
                    </tr>
                    <tr>
                            <td>Automatic</td>
   <td><input type="radio" name="geartype" value="automatic" id="automatic" /></td>
                    </tr>
                    <tr>
                            <td>Not mentioned</td>
                            <td><input type="radio" name="geartype" value="not-declared" id="not"
/></td>
                    </tr>
             </table>
             <i class="fas fa-check-circle"></i>
             <i class="fas fa-exclamation-circle"></i>
            <span></span>
```

```
            </div>
             <div class="form-control">
                    <h3>Your car is repaired or damaged :</h3>
                    <table style="width:50%">
                            <tr>
                                    <th></th>
                                    <th></th>

                            </tr>
                            <tr>
                                    <td>Yes</td>
   <td><input type="radio" name="damage" value="yes" id="yes"/></td>
                                </tr>
                                <tr>
                                        <td>No</td>
                                         <td><input type="radio" name="damage" value="no"
id="no"/></td>
                            </tr>
                            <tr>
                                    <td>Not Declared</td>
                                 <td><input type="radio" name="damage" value="not-declared"
id="notdec"/></td>
                            </tr>
                        </table>
             <i class="fas fa-check-circle"></i>
             <i class="fas fa-exclamation-circle"></i>
            <span></span>
            </div>
             <div class="form-control">
            <label for="model">Model Type : </label>
            <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>
<option value="fortwo">Fortwo </option>
<option value="1er">1er </option>
<option value="b_klasse">B Klasse </option>
<option value="signum">Signum </option>
<option value="astra">Astra </option>
<option value="a8">A8 </option>
<option value="jetta">Jetta </option>
<option value="fiesta">Fiesta </option>
<option value="c_klasse">C Klasse </option>
<option value="micra">Micra </option>
<option value="vito">Vito </option>
<option value="sprinter">Sprinter </option>
<option value="156">156 </option>
<option value="escort">Escort </option>
<option value="forester">Forester </option>
<option value="xc_reihe">Xc Reihe </option>
<option value="scenic">Scenic </option>
<option value="a4">A4 </option>
<option value="a1">A1 </option>
<option value="insignia">Insignia </option>
<option value="combo">Combo </option>
<option value="focus">Focus </option>
<option value="tt">Tt </option>

```
            <option value="a6">A6 </option>
            <option value="jazz">Jazz </option>
            <option value="omega">Omega </option>
        <option value="slk">Slk </option>   <option value="7er">7er </option>
            <option value="80">80 </option>
            <option value="147">147 </option>
            <option value="glk">Glk </option>
            <option value="100">100 </option>
            <option value="z_reihe">Z Reihe </option>
            <option value="sportage">Sportage </option>
            <option value="sorento">Sorento </option>
            <option value="v40">V40 </option>
            <option value="5er">5er </option>
            <option value="ibiza">Ibiza </option>
            <option value="3er">3er </option>
<option value="mustang">Mustang </option>
<option value="eos">Eos </option>
<option value="touran">Touran </option>
<option value="getz">Getz </option>
<option value="a3">A3 </option>
<option value="almera">Almera </option>
<option value="megane">Megane </option>
<option value="7er">7er </option>
<option value="1er">1er </option>
<option value="lupo">Lupo </option>
<option value="r19">R19 </option>
            <option value="zafıra">Zafıra </option>
            <option value="caddy">Caddy </option>
            <option value="2_reihe">2 Reihe </option>
            <option value="mondeo">Mondeo </option>
            <option value="cordoba">Cordoba </option>
            <option value="colt">Colt </option>
            <option value="impreza">Impreza </option>
            <option value="vectra">Vectra </option>
            <option value="berlingo">Berlingo </option>
            <option value="80">80 </option>
            <option value="m_klasse">M Klasse </option>
            <option value="tiguan">Tiguan </option>
            <option value="i_reihe">I Reihe </option>
            <option value="espace">Espace </option>
            <option value="sharan">Sharan </option>
```

<option value="6_reihe">6 Reihe </option>
<option value="panda">Panda </option>
<option value="up">Up </option>
<option value="seicento">Seicento </option>
<option value="ceed">Ceed </option>
<option value="5_reihe">5 Reihe </option>
<option value="yeti">Yeti </option>
<option value="octavia">Octavia </option>
<option value="mii">Mii </option>
<option value="rx_reihe">Rx Reihe </option>
<option value="6er">6er </option>
<option value="modus">Modus </option>
<option value="fox">Fox </option>
<option value="matiz">Matiz </option>
<option value="beetle">Beetle </option>
<option value="c1">C1 </option>
<option value="rio">Rio </option>
<option value="touareg">Touareg </option>
<option value="logan">Logan </option>
<option value="spider">Spider </option>
<option value="cuore">Cuore </option>
<option value="s_max">S Max </option>
<option value="a2">A2 </option>
<option value="x_reihe">X Reihe </option>
<option value="a5">A5 </option>
<option value="galaxy">Galaxy </option>
<option value="c3">C3 </option>
<option value="viano">Viano </option>
<option value="s_klasse">S Klasse </option>
<option value="1_reihe">1 Reihe </option>
<option value="avensis">Avensis </option>
<option value="sl">Sl </option>
<option value="roomster">Roomster </option>
<option value="q5">Q5 </option>
<option value="kaefer">Kaefer </option>
<option value="santa">Santa </option>
<option value="cooper">Cooper </option>
<option value="leon">Leon </option>
<option value="4_reihe">4 Reihe </option>
<option value="500">500 </option>
<option value="laguna">Laguna </option>

<option value="ptcruiser">Ptcruiser </option>
<option value="clk">Clk </option>
<option value="primera">Primera </option>
<option value="exeo">Exeo </option>
<option value="159">159 </option>
<option value="transit">Transit </option>
<option value="juke">Juke </option>
<option value="qashqai">Qashqai </option>
<option value="carisma">Carisma </option>
<option value="accord">Accord </option>
<option value="corolla">Corolla </option>
<option value="lanos">Lanos </option>
<option value="phaeton">Phaeton </option>
<option value="boxster">Boxster </option>
<option value="verso">Verso </option>
<option value="swift">Swift </option> <option value="rav">Rav </option>
<option value="kuga">Kuga </option>
<option value="picanto">Picanto </option>
<option value="kalos">Kalos </option>
<option value="superb">Superb </option>
<option value="stilo">Stilo </option>
<option value="alhambra">Alhambra </option> <option value="911">911 </option>
<option value="mx_reihe">Mx Reihe </option>
<option value="m_reihe">M Reihe </option> <option value="roadster">Roadster </option>
<option value="ypsilon">Ypsilon </option>
<option value="cayenne">Cayenne </option>
<option value="galant">Galant </option>
<option value="justy">Justy </option>
<option value="90">90 </option>
<option value="sirion">Sirion </option>
<option value="crossfire">Crossfire </option>
<option value="6_reihe">6 Reihe </option>
<option value="agila">Agila </option>
<option value="duster">Duster </option>
<option value="cr_reihe">Cr Reihe </option>
<option value="v50">V50 </option>
<option value="discovery">Discovery </option>
<option value="c_reihe">C Reihe </option>
<option value="v_klasse">V Klasse </option>
<option value="yaris">Yaris </option> <option value="c5">C5 </option>
<option value="aygo">Aygo </option>

<option value="cc">Cc </option>
<option value="carnival">Carnival </option>
<option value="fusion">Fusion </option>
<option value="bora">Bora </option>
<option value="forfour">Forfour </option>
<option value="100">100 </option>
<option value="cl">Cl </option>
<option value="tigra">Tigra </option>
<option value="156">156 </option>
<option value="300c">300c </option>
<option value="100">100 </option>
<option value="147">147 </option>
<option value="q3">Q3 </option>
<option value="spark">Spark </option>
<option value="v70">V70 </option>
<option value="x_type">X Type </option>
<option value="5_reihe">5 Reihe </option>
<option value="ducato">Ducato </option>
<option value="s_type">S Type </option>
<option value="x_trail">X Trail </option>
<option value="toledo">Toledo </option>
<option value="altea">Altea </option>
<option value="7er">7er </option>
<option value="voyager">Voyager </option>
<option value="calibra">Calibra </option>
<option value="bravo">Bravo </option>
<option value="range_rover">Range Rover </option>
<option value="antara">Antara </option>
<option value="tucson">Tucson </option>
<option value="q7">Q7 </option>
<option value="citigo">Citigo </option>
<option value="jimny">Jimny </option>
<option value="cx_reihe">Cx Reihe </option>
<option value="wrangler">Wrangler </option>
<option value="lybra">Lybra </option>
<option value="range_rover_sport">Range Rover Sport </option>
<option value="lancer">Lancer </option>
<option value="159">159 </option>
<option value="freelander">Freelander </option>
<option value="captiva">Captiva </option>
<option value="c2">C2 </option>

<option value="500">500 </option>
<option value="range_rover_evoque">Range Rover Evoque </option>
<option value="sandero">Sandero </option>
<option value="note">Note </option>
<option value="900">900 </option>
<option value="147">147 </option>
<option value="defender">Defender </option>
<option value="cherokee">Cherokee </option>
<option value="clubman">Clubman </option>
<option value="samara">Samara </option>
<option value="2_reihe">2 Reihe </option>
<option value="1er">1er </option>
<option value="3er">3er </option>
<option value="601">601 </option>
<option value="3_reihe">3 Reihe </option>
<option value="4_reihe">4 Reihe </option>
<option value="5er">5er </option>
<option value="6_reihe">6 Reihe </option>
<option value="legacy">Legacy </option>
<option value="pajero">Pajero </option>
<option value="auris">Auris </option>
<option value="niva">Niva </option>
<option value="5_reihe">5 Reihe </option>
<option value="s60">S60 </option>
<option value="nubira">Nubira </option>
<option value="vivaro">Vivaro </option>
<option value="g_klasse">G Klasse </option>
<option value="lodgy">Lodgy </option>
<option value="850">850 </option>
<option value="serie_2">Serie 2 </option>
<option value="6er">6er </option>
<option value="charade">Charade </option>
<option value="croma">Croma </option>
<option value="outlander">Outlander </option>
<option value="gl">Gl </option>
<option value="doblo">Doblo </option>
<option value="musa">Musa </option>
<option value="amarok">Amarok </option>
<option value="156">156 </option>
<option value="move">Move </option>
<option value="9000">9000 </option>

```
<option value="v60">V60 </option>
            <option value="145">145 </option>
            <option value="aveo">Aveo </option>
            <option value="200">200 </option>
            <option value="300c">300c </option>
            <option value="b_max">B Max </option>
            <option value="delta">Delta </option>
            <option value="terios">Terios </option>
            <option value="rangerover">RangeRover </option>
            <option value="90">90 </option>
            <option value="materia">Materia </option>
            <option value="kalina">Kalina </option>
            <option value="elefantino">Elefantino </option>
            <option value="i3">I3 </option>
            <option value="kappa">Kappa </option>
            <option value="serie_3">Serie 3 </option>
            <option value="48429">48429 </option>
            <option value="serie_1">Serie 1 </option>
            <option value="discovery_sport">Discovery Sport </option>
            </select>
             <i class="fas fa-check-circle"></i>
             <i class="fas fa-exclamation-circle"></i>
            <span></span>
            </div>
             <div class="form-control">
            <label for="brand">Brand :</label>
            <select name="brand" id="brand">
            <option value="" disabled selected hidden>Choose Brand Name...</option>
            <option value="volkswagen">Volkswagen </option>
            <option value="audi">Audi </option>
            <option value="jeep">Jeep </option>
            <option value="skoda">Skoda </option>
            <option value="bmw">Bmw </option>
            <option value="peugeot">Peugeot </option>
            <option value="ford">Ford </option>
        <option value="mazda">Mazda </option> <option value="nissan">Nissan </option>
<option value="renault">Renault </option>
<option value="mercedes_benz">Mercedes Benz </option>
<option value="opel">Opel </option>
<option value="seat">Seat </option>
<option value="citroen">Citroen </option>
```

```
<option value="honda">Honda </option>
<option value="fiat">Fiat </option>
<option value="mini">Mini </option>
<option value="smart">Smart </option>
<option value="hyundai">Hyundai </option>
                <option value="sonstige_autos">Sonstige Autos </option>
                <option value="alfa_romeo">Alfa Romeo </option>
                <option value="subaru">Subaru </option>
                <option value="volvo">Volvo </option>
                <option value="mitsubishi">Mitsubishi </option>
                <option value="kia">Kia </option>
                <option value="suzuki">Suzuki </option>
                <option value="lancia">Lancia </option>
                <option value="porsche">Porsche </option>
                <option value="toyota">Toyota </option>
                <option value="chevrolet">Chevrolet </option>
                <option value="dacia">Dacia </option>
                <option value="daihatsu">Daihatsu </option>
                <option value="trabant">Trabant </option>
                <option value="saab">Saab </option>
                <option value="chrysler">Chrysler </option>
                <option value="jaguar">Jaguar </option>
                <option value="daewoo">Daewoo </option>
                <option value="rover">Rover </option>
                <option value="land_rover">Land Rover </option>
                <option value="lada">Lada </option>
                </select>
                 <i class="fas fa-check-circle"></i>
                 <i class="fas fa-exclamation-circle"></i>
                <span></span>
                </div>
                 <div class="form-control">
                <label for="fuelType">Fuel Type :</label>
                <select name="fuelType" id="fuel">
                <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>
<i class="fas fa-check-circle"></i>
<i class="fas fa-exclamation-circle"></i>
<span></span>
</div>
<div class="form-control">
<label for="vehicletype">Vehicle type: </label>
<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>
             <i class="fas fa-check-circle"></i>
             <i class="fas fa-exclamation-circle"></i>
            <span></span>
            </div>
            <input type="submit" id="submit"></input>
            </form>
 </div>
</body>
</html>

value.css *{ padding:0px; margin:0; box-sizing:border-box; font-family: cursive; font-weight: bold;
color: #E74C3C;
}

body{
            background-image: linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car2.png);
      min-height:100vh;     display:flex;         justify-content:center;
            align-items:center;
}
```

```
.header{
      color:Black;    text-align:center;
            padding:10px 0px 10px 100px;
}

#model{
      width:500px;   color: black;
}

#brand{
      width:500px;   color: black;
}

#fuel{
      width:500px;   color: black;
}

#vehicle{
      width:500px;   color: black;
}

.form{
            padding:30px 40px;
}

.form-control{
      margin-bottom:10px;            padding-bottom: 20px;        position:relative;
            margin-left: 100px;
}

.form-control label{
      display:block;           margin-bottom:5px;
}

.form-control input{          border: 2px solid #f0f0f0;   width:80%;
font-size :.8rem;
            padding:5px;
             display:inline-table;
}

.form-control i{
```

```
    position:absolute;      right:20px;     top:35px;
        visibility:hidden;
}

.form-control span{
    position:absolute;      left:0;   bottom:0;
    visibility:hidden;      font-weight:bolder;     font-style:italic;
        font-size:1rem;
}

.form-control.success input{
        border-color:#2ecc71;
}

.form-control.error input{
        border-color:#e743c3;
}

.form-control.error span{
        color:red;
        visibility:visible;
}



.form-control.success i.fa-check-circle {   border-color:#2ecc71;
        visibility:visible;
}

.form-control.error i.fa-exclamation-circle {       border-color:#e73c3c;
        visibility:visible;
}
.form #submit{
    background-color:#E74C3C;       border:none;    outline:none;       color:white;
width:500px;        border-radius:4px;      padding:10px;       cursor:pointer;     transition:all
.5s;        font-size:1rem;
        margin-left: 100px;
}

.form #submit:hover{        background-color:#6441a5;
}
```

```
.form-control #manual{
            padding-bottom: 20%;
}

predict.html
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="../static/css/predict.css">
   <title>Car Resale Value Predicting Application</title>
   <link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
</head>
<body>
            <section class="header">
    <nav>
      <a href="/"><img src="../static/Images/logo.png" width="100" height="100"></a>      </nav>
      <div class="text-box">
        <h1>The Predicted Car Resale Value is </h1>
                    <h1>{{predict}}</h1>
      </div>
   </section>

       </body> </html>

predict.css .header{   min-height: 100vh;   width: 100%;   background-image: linear-
gradient(rgba(25,30,30,0.7),rgba(25,30,30,0.7)),url(../Images/car3.jpg);   background-position:
center;   background-size: cover;   position: relative;
}

.text-box{   text-align: center;   position: relative;   color:  #FFE4C4;   top:50%; } .text-box h1{
margin-top: 50px;   font-size: 55px;
}

.text-box p{   margin: 10px 0 40px;   font-size: 15px;
}

body{
```

```
        margin: 0;
}

nav{
  display:flex;   padding: 2% 6%;   justify-content: space-between;   align-items: center;
}
```

app.py

```
# 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 = "04ZW6LlrLwAfofEU2VHPt69RKCWVc9U1o5LXkAU_66qA"
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')

         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_labels','brand_labels', 'fuelType_labels', 'vehicletype_labels']], "values": X}]}

    response_scoring =
requests.post('https://ussouth.ml.cloud.ibm.com/ml/v4/deployments/c0f74260-1f5f-43ad-
8d71eb12ef099507/predictions?version=2022-11-13', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})    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)

car_resale_value_prediction_modelling.py
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pickle
import wandb

#regression models
from sklearn.ensemble
 import BaggingRegressor, RandomForestRegressor,
HistGradientBoostingRegressor, ExtraTreesRegressor
from xgboost.sklearn import XGBRegressor
from lightgbm import LGBMRegressor

wandb.login(key='b75e0564aba32dce859c60044418df71ce7389a8')

data = pd.read_csv('../input/naalaiya-thiran/Preprocessed/autos_preprocessed.csv', header=0,
sep=',', encoding='Latin1')

labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

mapper = {} for i in labels:
   mapper[i] = LabelEncoder()     mapper[i].fit(data[i])     tr = mapper[i].transform(data[i])
np.save(str('classes'+i+'.npy'), mapper[i].classes_)     data.loc[:, i+'_labels'] = pd.Series(tr,
index=data.index)

labeled = data[['price', 'yearOfRegistration','powerPS','kilometer','monthOfRegistration']
+[x+"_labels" for x in labels]]

print(labeled.columns)

def find_scores(Y_actual, Y_pred, X_train):
   mae = mean_absolute_error(Y_actual, Y_pred)     mse = mean_squared_error(Y_actual,
```

```
Y_pred)
   rmse = np.sqrt(mse)     rmsle = np.log(rmse)     r2 = r2_score(Y_actual, Y_pred)     n, k =
X_train.shape
   adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))

   wandb.log({"mae": mae, "mse": mse, 'rmse':rmse, 'rmsle':rmsle, 'r2':r2, 'adj_r2':adj_r2_score})

def bagging_regressor():     config_defaults = {
            'n_estimators':100,
            'max_samples':0.4,
            'bootstrap':True,
            'random_state':42
         }
   wandb.init(config=config_defaults)     config = wandb.config

     A.   = labeled.iloc[:,1:].values
     B.   = labeled.iloc[:,0].values.reshape(-1,1)

   X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

   model = BaggingRegressor(       n_estimators=config.n_estimators,
bootstrap=config.bootstrap,       max_samples=config.max_samples,
     random_state = config.random_state)

   model.fit(X_train, Y_train)

   Y_pred = model.predict(X_test)

   find_scores(Y_test, Y_pred, X_train)

bagging_regressor_configs = {
   "name":'BaggingRegressor',
   "method": "grid",
   "metric": {
      "name": "adj_r2",
      "goal": "maximize"
   },
   "parameters": {
      "n_estimators": {
         "values": [100, 200, 300]
      },
```

```
      "max_samples": {
        "values": [0.4,0.5, 0.6]
      }
    }
}

sweep_id = wandb.sweep(sweep=bagging_regressor_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=bagging_regressor)

def random_forest_regressor():    config_defaults = {
            'n_estimators':100,
            'max_samples':0.4,
            'criterion':'squared_error',
            'bootstrap': True,
            'random_state':42
        }
    wandb.init(config=config_defaults)    config = wandb.config

     A.  = labeled.iloc[:,1:].values
     B.  = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = RandomForestRegressor(      n_estimators=config.n_estimators,      criterion =
config.criterion,      bootstrap=config.bootstrap,      max_samples=config.max_samples,
     random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

random_forest_configs = {    "name":'RandomForestRegressor',
    "method": "grid",
    "metric": {
      "name": "adj_r2",
      "goal": "maximize"
    },
    "parameters": {
      "n_estimators": {
```

```
            "values": [100, 200, 300]
        },
        "max_samples": {
            "values": [0.4,0.5, 0.6]
        }
    }
}

sweep_id = wandb.sweep(sweep=random_forest_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=random_forest_regressor)

def hist_gradient_boost_regressor():
    config_defaults = {
                'loss':'squared_error',
                'learning_rate': 0.1,
                'max_iter':100,
                'random_state':42
            }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = HistGradientBoostingRegressor(       loss=config.loss,
      learning_rate = config.learning_rate,       max_iter=config.max_iter,
      random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

hist_gradient_boost_configs = {
    "name":'HistGradientBoostingRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
```

```
    },
    "parameters": {
        "loss": {
            "values": ['squared_error', 'absolute_error']
        },
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "max_iter": {
            "values": [100,200,300]
        },
        "random_state": {
            "values": [42]
        }
    }
}

sweep_id = wandb.sweep(sweep=hist_gradient_boost_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=hist_gradient_boost_regressor)

def extra_tree_regressor():
    config_defaults = {
                'criterion':'squared_error',
                'max_samples':0.4,
                'bootstrap': True,
                'random_state':42
            }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = ExtraTreesRegressor(       criterion=config.criterion,       bootstrap = config.bootstrap,
max_samples=config.max_samples,
      random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)
```

```
    find_scores(Y_test, Y_pred, X_train)

extra_tree_configs = {
    "name":'ExtraTreesRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "criterion": {
            "values": ['squared_error', 'absolute_error']
        },
        "max_samples": {
            "values": [0.4,0.5, 0.6]
        }
    }
}

sweep_id = wandb.sweep(sweep=extra_tree_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=extra_tree_regressor)

def XGB_regressor():     config_defaults = {
              'learning_rate':0.1,
              'n_estimators': 500,
              'booster':'gbtree',
              'eta':0.01,
              'random_state':42
          }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)

    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = XGBRegressor(       learning_rate=config.learning_rate,       n_estimators =
config.n_estimators,
      random_state = config.random_state)
```

```
    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

extra_tree_configs = {     "name":'XGBRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "n_estimators": {
            "values": [100,200,300]
        },
        "booster": {
            "values": ['gbtree','gblinear']
        },
        "eta": {
            "values": [0.01, 0.03, 0.05, 0.07]
        }
    }
}

sweep_id = wandb.sweep(sweep=extra_tree_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=XGB_regressor)

def LGBM_regressor():     config_defaults = {
              'objective':'root_mean_squared_error',
              'reg_sqrt': True,
              'metric':'rmse',
              'random_state':42
            }
    wandb.init(config=config_defaults)     config = wandb.config

      A.   = labeled.iloc[:,1:].values
      B.   = labeled.iloc[:,0].values.reshape(-1,1)
```

```
    X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

    model = LGBMRegressor(       learning_rate=config.learning_rate,       n_estimators =
config.n_estimators,
      random_state = config.random_state)

    model.fit(X_train, Y_train)

    Y_pred = model.predict(X_test)

    find_scores(Y_test, Y_pred, X_train)

lgbm_configs = {
    "name":'LGBMRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "objective": {
            "values": ['root_mean_squared_error']
        },
        "boosting_type": {
            "values": ['gbdt','dart','goss','rf']
        },
        "reg_sqrt": {
            "values": [True]
        },
        "metric": {
            "values": ['rmse']
        },
        "n_estimators": {
            "values": [100,200,300]
        },
        "random_state": {
            "values": [42]
```

```
      }
    }
}

sweep_id = wandb.sweep(sweep=lgbm_configs, project="car_resale_value")
wandb.agent(sweep_id=sweep_id, function=LGBM_regressor)

car_resale_value_prediction_LGBM.py
import pandas as pd
import numpy as np from sklearn.preprocessing
 import LabelEncoder from sklearn.model_selection
import train_test_split
from sklearn.metrics
 import mean_absolute_error, mean_squared_error, r2_score
 import pickle

#regression model
from lightgbm import LGBMRegressor


import os, types import pandas as pd from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your
credentials.
# You might want to remove those credentials before you share the notebook. cos_client =
ibm_boto3.client(service_name='s3',
    ibm_api_key_id='8DImq73hywb09uzAo_T_TsAZI_ocZgFLuhQdwmfUJZTX',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'carresalevalueprediction-donotdelete-pr-whcxr42j79mqcv' object_key =
'autos_preprocessed.csv'

body = cos_client.get_object(Bucket=bucket,Key=object_key)['Body'] # add missing __iter__
method, so pandas accepts body as file-like object
if not hasattr(body, "__iter__"): body.__iter__ = types.MethodType( __iter__, body )
```

```
data = pd.read_csv(body)
data.head()

labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

mapper = {} for i in labels:
   mapper[i] = LabelEncoder()     mapper[i].fit(data[i])     tr = mapper[i].transform(data[i])
np.save(str('classes'+i+'.npy'), mapper[i].classes_)     data.loc[:, i+'_labels'] = pd.Series(tr,
index=data.index)

labeled = data[['price', 'yearOfRegistration','powerPS','kilometer','monthOfRegistration']
+[x+"_labels" for x in labels]]

print(labeled.columns)

def find_scores(Y_actual, Y_pred, X_train):
   scores = dict()
   mae = mean_absolute_error(Y_actual, Y_pred)     mse = mean_squared_error(Y_actual,
Y_pred)
   rmse = np.sqrt(mse)     rmsle = np.log(rmse)     r2 = r2_score(Y_actual, Y_pred)     n, k =
X_train.shape
   adj_r2_score = 1 - ((1-r2)*(n-1)/(n-k-1))

   scores['mae']=mae     scores['mse']=mse     scores['rmse']=rmse     scores['rmsle']=rmsle
scores['r2']=r2
   scores['adj_r2_score']=adj_r2_score

   return scores

     A.   = labeled.iloc[:,1:].values
     B.   = labeled.iloc[:,0].values.reshape(-1,1)

X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.4, random_state=42)

model =
LGBMRegressor(boosting_type="gbdt",learning_rate=0.07,metric="rmse",n_estimators=300,obj
ective="root_mean_squared_error",random_state=42,reg_sqrt=True)

model.fit(X_train, Y_train)
```

```
Y_pred = model.predict(X_test)

find_scores(Y_test, Y_pred, X_train)

pickle.dump(model, open('resale_model.sav', 'wb'))

get_ipython().system('pip install -U ibm-watson-machine-learning')

from ibm_watson_machine_learning import APIClient import json

wml_credentials = {
   "apikey":"Qo9j8ni7qMJ8j1C8VFDRFHbuGRAhYWcTlkVqnYg1AGkE",
   "url":"https://us-south.ml.cloud.ibm.com"
}

wml_client = APIClient(wml_credentials)
wml_client.spaces.list()

SPACE_ID= "bf7bc386-40bf-4d85-91e6-eedd2c53f245"

wml_client.set.default_space(SPACE_ID)

wml_client.software_specifications.list(100)

import sklearn
sklearn.__version__

MODEL_NAME = 'CRVP'
DEPLOYMENT_NAME = 'CRVP'
DEMO_MODEL = model

software_spec_uid = wml_client.software_specifications.get_id_by_name('runtime-22.1-py3.9')
model_props = {     wml_client.repository.ModelMetaNames.NAME: MODEL_NAME,
wml_client.repository.ModelMetaNames.TYPE: 'scikit-learn_1.0',
   wml_client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid
}

model_details = wml_client.repository.store_model(
   model=DEMO_MODEL,
   meta_props=model_props,      training_data=X_train,
   training_target=Y_train
```

```
)

model_details
model_id = wml_client.repository.get_model_id(model_details) model_id deployment_props = {
   wml_client.deployments.ConfigurationMetaNames.NAME:DEPLOYMENT_NAME,
wml_client.deployments.ConfigurationMetaNames.ONLINE: {}
}
deployment = wml_client.deployments.create(
   artifact_uid=model_id,
   meta_props=deployment_props
)
```

GitHub Link:

1. Click here to redirect Git hub repository