

# **PROJECT REPORT**

## **CAR RESALE VALUE PREDICTION**

TEAM ID : PNT2022TMID28400

### **TEAM LEADER**

Keerthna M.P - 312419205048

### **TEAM MEMBERS**

Akshaya Lakshmi.P - 312419205012

Akshaya P - 312419205009

Kabilesh Raja.R - 312419205043

## **1.INTRODUCTION**

### **1.1 Project overview**

The Car Valuation Tool is a free tool designed to help you get the estimated resale value of your car within seconds. Our automobile valuation algorithm is real-time updated, so it keeps up with the most recent modifications and market trends. However, the amounts displayed during the online assessment are only estimates and might alter when the retailer inspects your automobile. You don't even need to register to have your automobile valued; all you need to do is provide some basic information about it, such as its make, model, amount of miles driven, city of residence, and contact information.

### **1.2 Purpose**

In 2019, the Indian used automobile resale industry was valued at \$24.2 billion USD.

There is a critical need to close this gap between sellers and buyers due to the enormous demand for used automobiles and the shortage of professionals who can evaluate the proper valuation. The goal of this research is to create a system that can

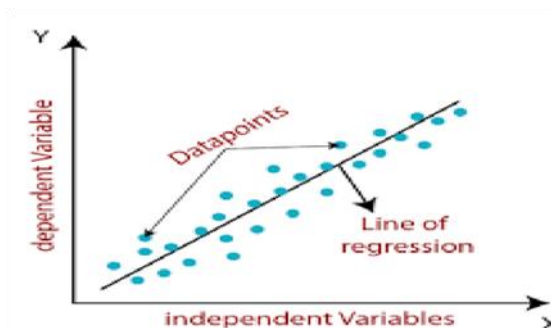
impartially forecast a car's resale value based on little information such as the number of miles travelled and the year of purchase. The process of determining the current used automobile pricing in a certain location is known as used car value. By selecting the brand, model, year, trim, and the number of kilometers travelled, a user of OBV may quickly determine the used car's price. The value of a used automobile is based on a number of variables, including its state right now, when it was bought, etc. Used automobile valuation will never have a precise price; instead, it will always fall within a reasonable price range.

## 2. LITERATURE SURVEY

### (1) Linear Regression:

It is an AI calculation dependent on administered learning. It plays out a relapse task. It is utilized to assess genuine qualities (cost of houses, number of calls, absolute deals and so forth) in view of nonstop variable(s). Here, we set up connection among free and ward factors by fitting a best line. This best fit line is known as relapse line and spoke to by a straight condition  $Y = a * X + b$ . Prior to understanding what direct relapse is, let us get ourselves acclimated with relapse. Relapse is a strategy for demonstrating an objective worth dependent on free indicators. This strategy is generally utilized for spreading and discovering circumstances and logical results connection between factors. Relapse methods generally vary dependent on the quantity of autonomous factors.

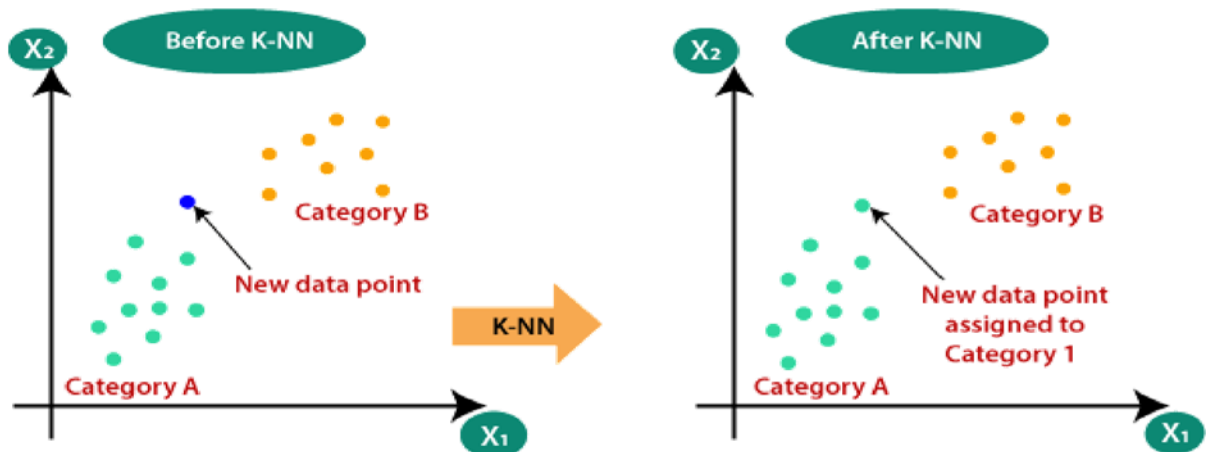
In fig(1) it is very clear how the linear regression algorithm will work.



### (2) K-Nearest Neighbours

K-nearest neighbour is a machine learning technique in which the new (unknown) data is

all the existing records in order to locate the best match(es). Despite its apparent simplicity, a lot of care has to be taken in pre-processing the data otherwise we can easily go off-track. Only three attributes were considered namely the make, year and cylinder volume. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data. It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data. **Example:** Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

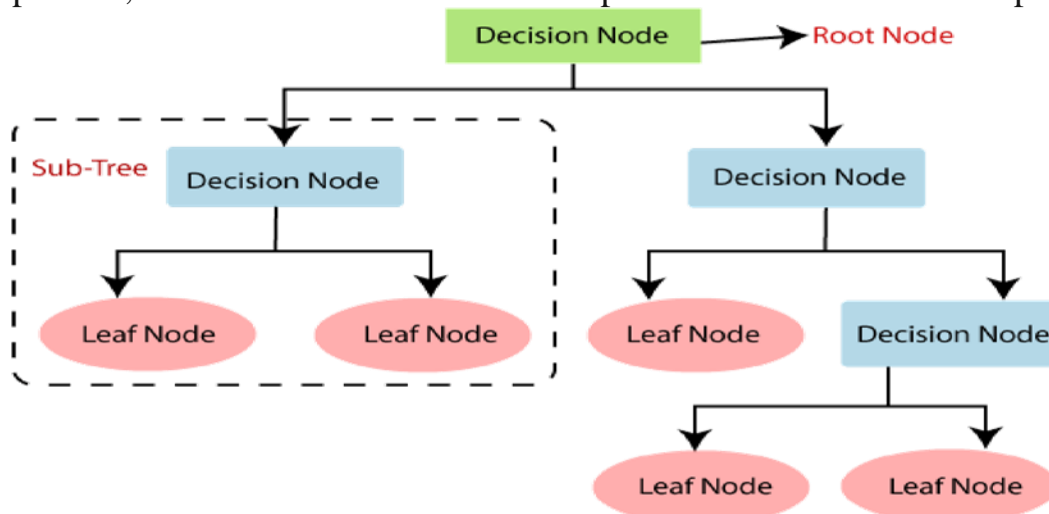


### (3)Decision Tree

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both

and regression tasks. It has a hierarchical, tree structure, which consists of a root node, branches, internal nodes and leaf nodes. Decision tree learning employs a divide and conquer strategy by conducting a greedy search to identify the optimal split points within a tree. This process of splitting is then repeated in a top-down, recursive manner until all, or the majority of records have been classified under specific class labels. Whether or not all data points are classified as homogenous sets is largely dependent on the complexity of the decision tree.

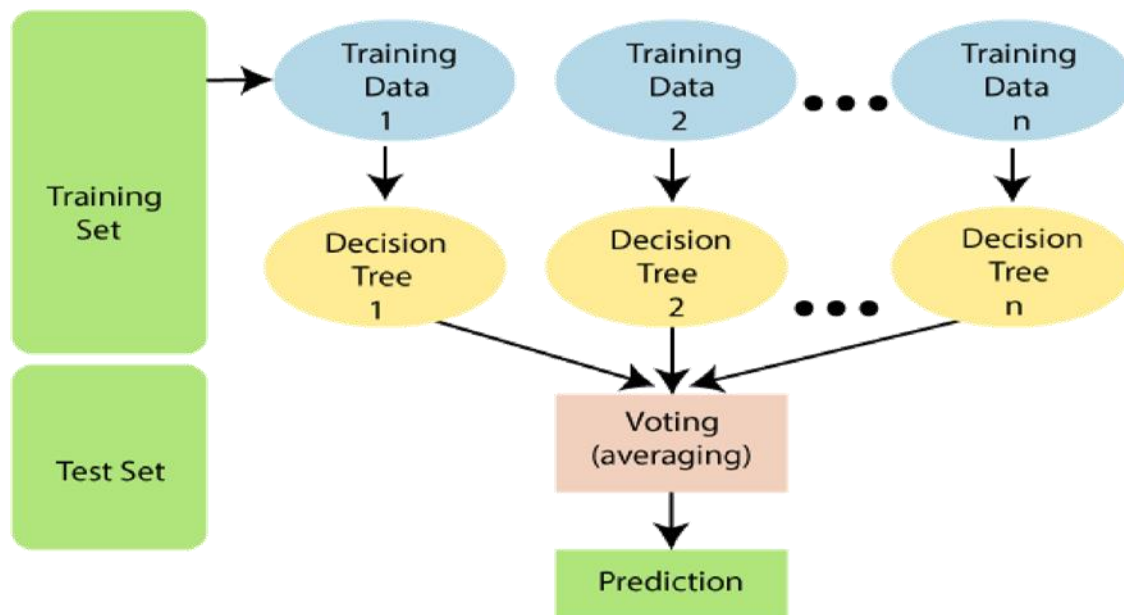
Smaller trees are more easily able to attain pure leaf nodes—i.e. data points in a single class. However, as a tree grows in size, it becomes increasingly difficult to maintain this purity, and it usually results in too little data falling within a given subtree. When this occurs, it is known as data fragmentation, and it can often lead to over fitting. As a result, decision trees have preference for small trees, which is consistent with the principle of parsimony in Occam's Razor; that is, "entities should not be multiplied beyond necessity." Said differently, decision trees should add complexity only if necessary, as the simplest explanation is often the best. To reduce complexity and prevent over fitting, pruning is usually employed; this is a process, which removes branches that split on features with low importance.



### (4) Random forest classifier

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression

problems in ML. It is based on the concept of **ensemble learning**, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of over fitting. Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of **ensemble learning**, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model*.



## (5) Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical

or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1.**

## **2.1 Existing problem**

Car Resale value prediction is one of the best to sell our in this market for an best and better price. Rather than giving our car to an less price , the customer those who uses the car will be benifitted and the seller will also be benefitted. The goal of this research is to create a system that can impartially forecast a car's resale value based on little information such as the number of miles travelled and the year of purchase. You don't even need to register to have your automobile valued; all you need to do is provide some basic information about it, such as its make, model, amount of miles driven, city of residence, and contact information.

## **2.2 References**

- [1] Pudaruth Sameerchand , Pudaruth Sameerchand , Predicting the price of Used Car Using Machine Learning Techniques
- [2] Enis gegic, Becir ,Isakovic, Dino Keco, ,Zerina Masetic,Jasmin Kevric Car Price Prediction Using Machine Learning
- [3] Ning sun,Hongxi Bai,Yuxia Geng,Huizhu Shi Price Evaluation model in second hand car system
- [4] Doan Van Thai, Luong Ngoc Son, Pham Vu Tien, Nguyen Nhat Anh, Nguyen Thi Ngoc Anh Prediction car prices using qualify qualitative data and knowledge-based system

## **2.3 Problem Statement Definition**

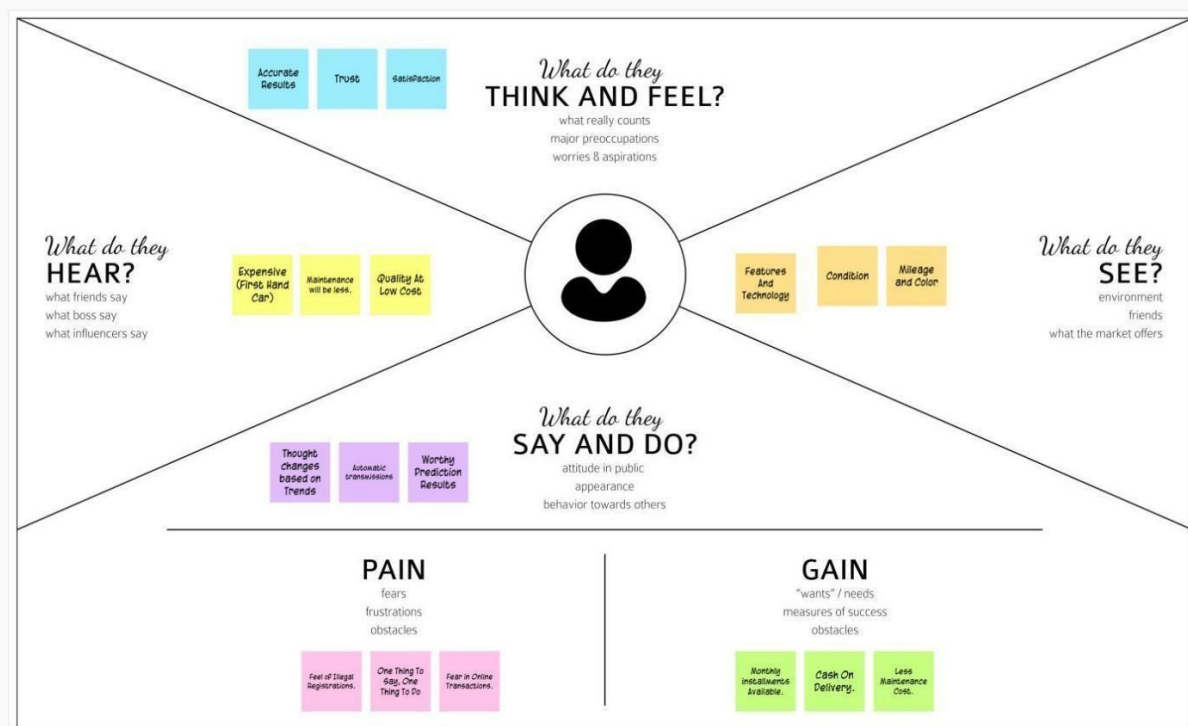
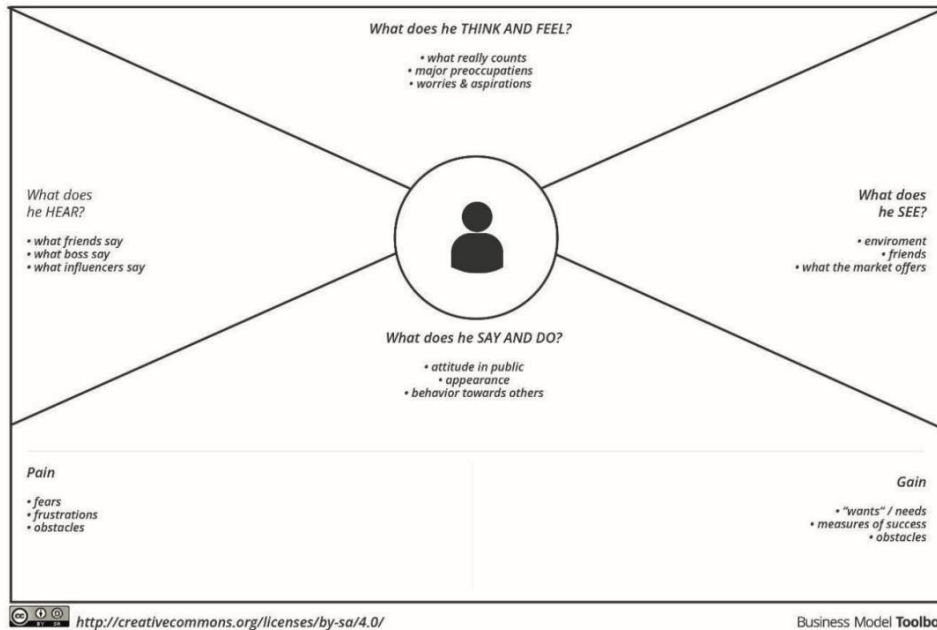
Car Resale value prediction is used to predict the value of the used cars to an reasonable price which satisfies the customer.

# 3 IDEATION AND PROPOSED SOLUTION

## 3.2 Ideation & Brainstroming

### 3.1 Empathy Map Canvas

#### Empathy Map



1

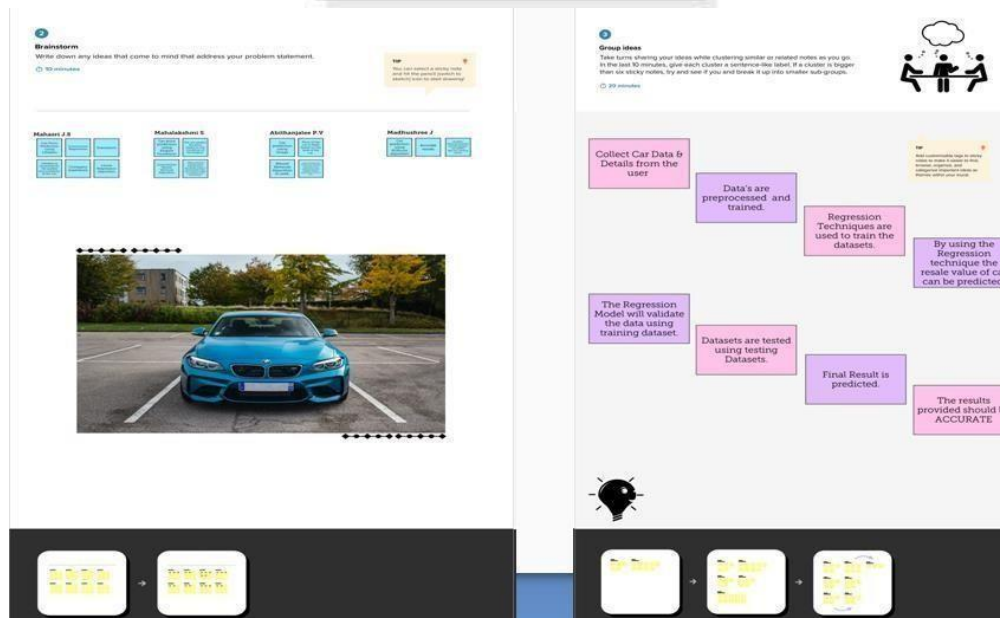
## Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

🕒 5 minutes

### PROBLEM

The main aim of this project is to predict the price of used cars using different machine learning models



Idea prioritization :

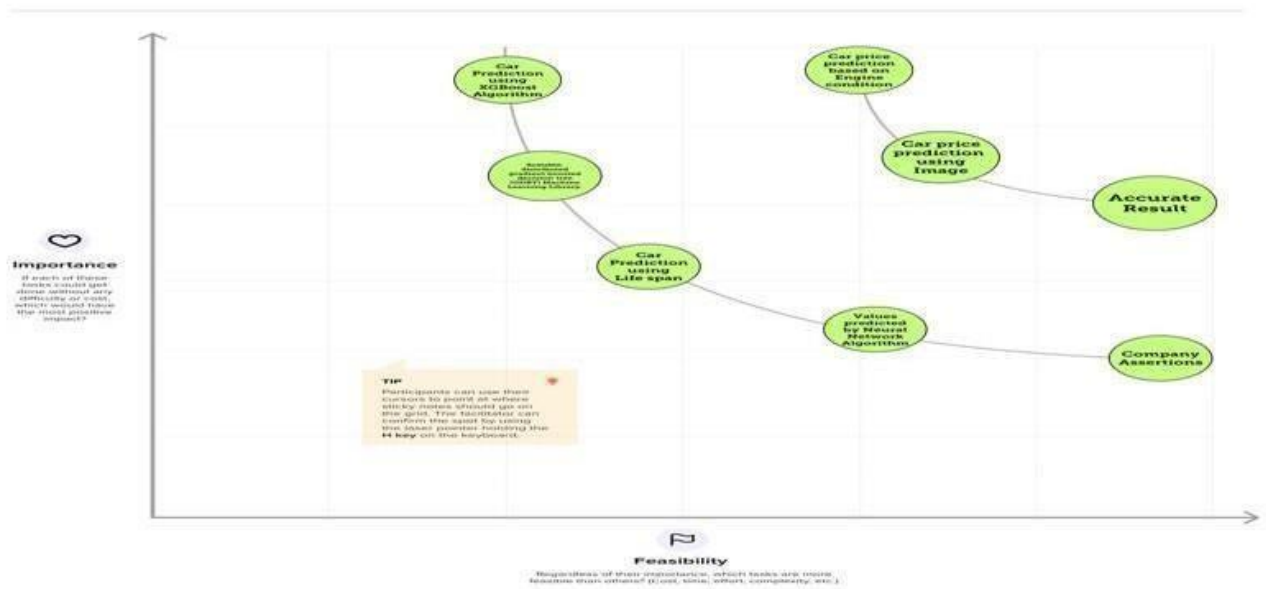


4

#### Prioritize

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

⌚ 30 minutes



### 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

Define CS, fit into CC

1. CUSTOMER SEGMENT(S)

CS

Who is your customer?  
i.e. working parents of 0-5 y.o. kids

- Used car sellers
- Buyers
- First time car buyer

6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

- Customer were concerned when they still had no clue what they had discovered.
- They may have heard of internet scammers.

5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

- By searching in online websites.
- By acquiring knowledge from the people and gaining an understanding.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

To build a supervised machine learning model that use regression methods to anticipate the value of a car based on several factors like as

- Condition of Engine
- Life span of used car
- Kilometers driven
- Look of the car
- Solving customer doubts

9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists?  
What is the back story behind the need to do this job?  
i.e. customers have to do it because of the change in regulations.

- The price projected by dealers or brokers for a secondhand car is untrustworthy.
- Users can predict the proper car valuation remotely, without the need for human interaction, like car dealers do.

7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?  
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- To avoid wear and tear fines, leased cars must be returned in excellent condition.
- Beware of selling frauds.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Users may calculate the proper valuation of a car on their own utilizing model, year, owner, and other car resale value prediction websites.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?  
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

Before:

- The user will be concerned about the inaccurate figures anticipated by humans based on the state of the vehicle.

After:

- Without user intervention, the user may decide the reliability of the car on their own.

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.

If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

- Machine learning and a regression model were used to create this system. We can estimate the resale value of a car at any moment and from any location by applying this model.
- The project should use used car parameters as inputs and allow buyers to make their own judgments.

8. CHANNELS of BEHAVIOUR

CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

ONLINE

- Customers don't just look for information on vehicle brand websites; they also visit comparison sites to compare pricing and user ratings.

OFFLINE

- When a buyer wanted to buy a car, they would go from dealership to dealership, meeting with salesmen and determining where they might get the greatest bargain.

Identify strong TR & EM

## 4 REQUIREMENT ANALYSIS

### 4.1 Functional requirement

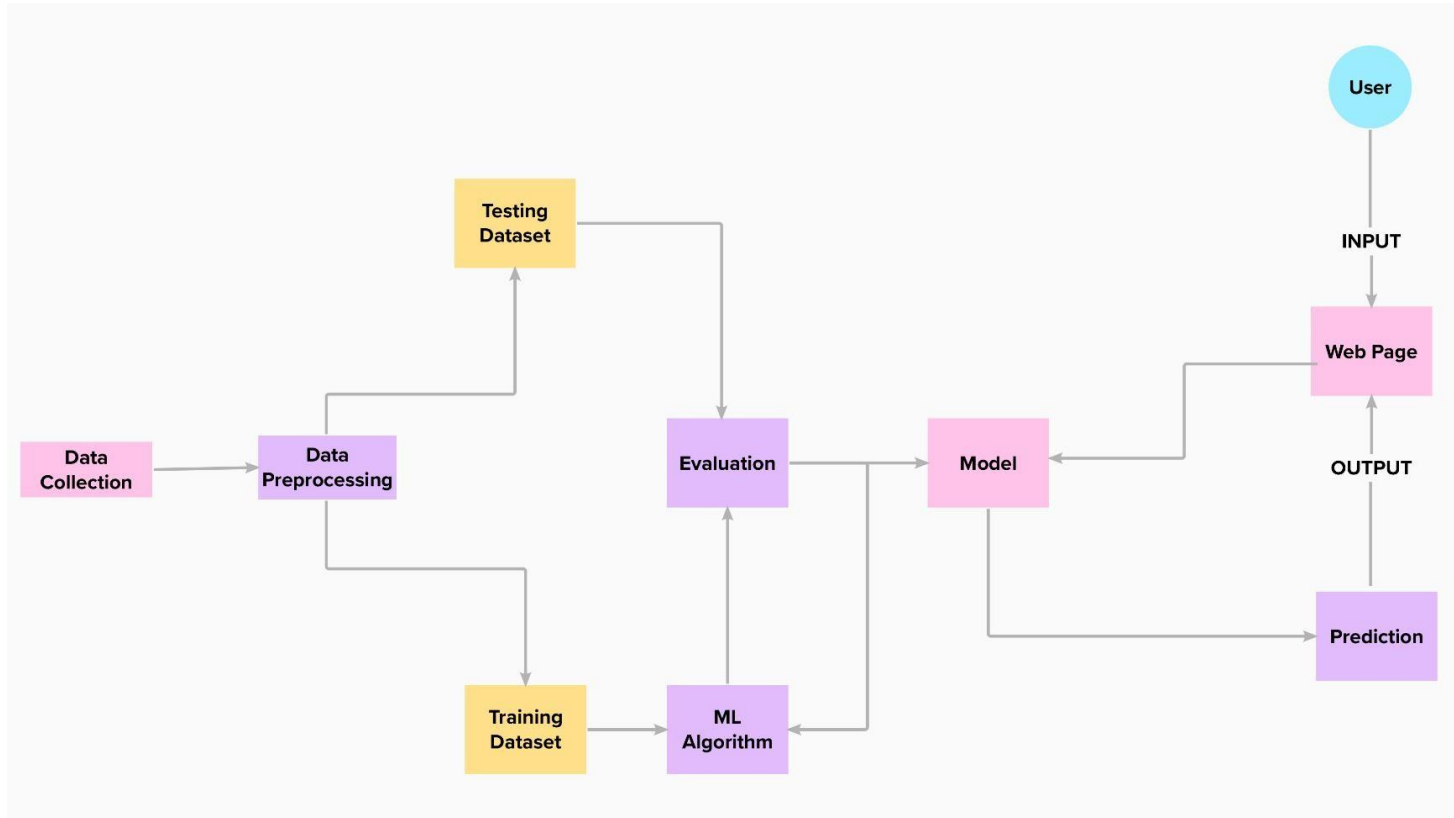
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Website
FR-2	User Confirmation	Confirmation via Website
FR-3	Car Registration	Registration through Website
FR-4	Car Information	Getting the car details through Website
FR-5	Value Prediction	Shows the resale value of the car through website

### 4.2 Non-Functional requirement

FR No.	Non-Functional Requirement	Description
NFR-1	<b>Usability</b>	The model predicts the resale value of the car with more accuracy.
NFR-2	<b>Security</b>	Protect the user information as well as their car details.
NFR-3	<b>Reliability</b>	The model performs consistently well and also it begins trust to the user.
NFR-4	<b>Performance</b>	The model performance has high accuracy and with portable from one machine to another machine.
NFR-5	<b>Availability</b>	The model can be available anywhere at anytime.
NFR-6	<b>Scalability</b>	The model can be viewed and accessed in both computer as well as mobile phone.

## 5 PROJECT

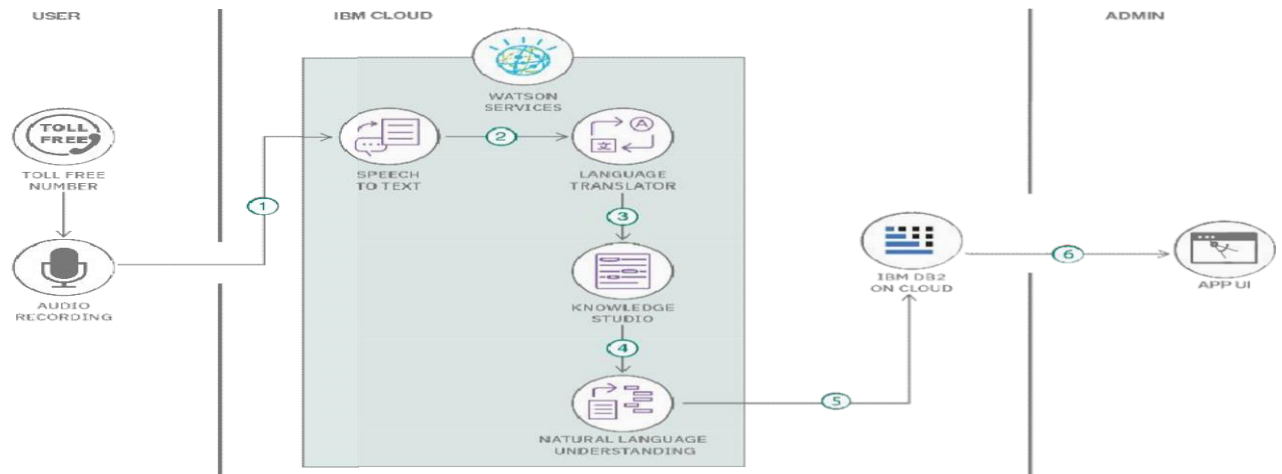
### DESIGN 5.1



Data Flow

Diagram

### 5.2 Solution & Technical Architecture



## 6 PROJECT PLANNING & SCHEDULING

### 6.1 Sprint Planning & Estimation

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Pre-process data	USN-1	Collect Dataset	1	Low	Akshaya.P
Sprint-1		USN-2	Import required libraries	1	Low	AkshayaLakshmi.P
Sprint-1		USN-3	Read and clean data sets	2	Low	Keerthna.MP
Sprint-2	Model building	USN-1	Split data into independent and dependent variables	3	Medium	KabileshRaja
Sprint-2		USN-2	Apply using regression model	3	Medium	AkshayaLakshmi.P
Sprint-3	Application building	USN-1	Build python flask application and HTML page	5	High	Akshaya.P & Keerthna.MP
Sprint-3		USN-2	Execute and test	5	High	AkshayaLakshmi.P
Sprint-4	Training the model	USN-1	Train machine learning model	5	High	KabileshRaja & Akshaya.P
Sprint-4		USN-2	Integrate flask	5	High	Keerthna.MP

## 6.2 Project Planning Phase

### Milestone and Activity List

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc.	2 SEPTEMBER 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	10 SEPTEMBER 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	15 SEPTEMBER 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	23 SEPTEMBER 2022
Problem Solution Fit	Prepare problem - solution fit document.	28 SEPTEMBER 2022
Solution Architecture	Prepare solution architecture document.	5 OCTOBER 2022
Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	7 OCTOBER 2022

<b>Functional Requirement</b>	Prepare the functional requirement document.	8 OCTOBER 2022
<b>Data Flow Diagrams</b>	Draw the data flow diagrams and submit for review.	10 OCTOBER 2022
<b>Technology Architecture</b>	Prepare the technology architecture diagram.	15 OCTOBER 2022
<b>Prepare Milestone &amp; Activity List</b>	Prepare the milestones & activity list of the project.	26 OCTOBER 2022
<b>Project Development - Delivery of Sprint-1, 2, 3 &amp; 4</b>	Develop & submit the developed code by testing it.	IN PROGRESS..



# **CODING & SOLUTIONING**

## **7.1 Feature 1**

- IoT device
- IBM Watson Platform
- Node red
- Cloudant DB
- Web UI
- MIT App Inventor
- Python code

## **7.2 Feature2**

- Login
- Wokwi

## 8 TESTING AND RESULTS

### 8.1 Test Cases

Test case 1:

Test case 2:

*Get the Accurate Resale Value of Your Car*

Registration Number

Registration Year

Registration Month

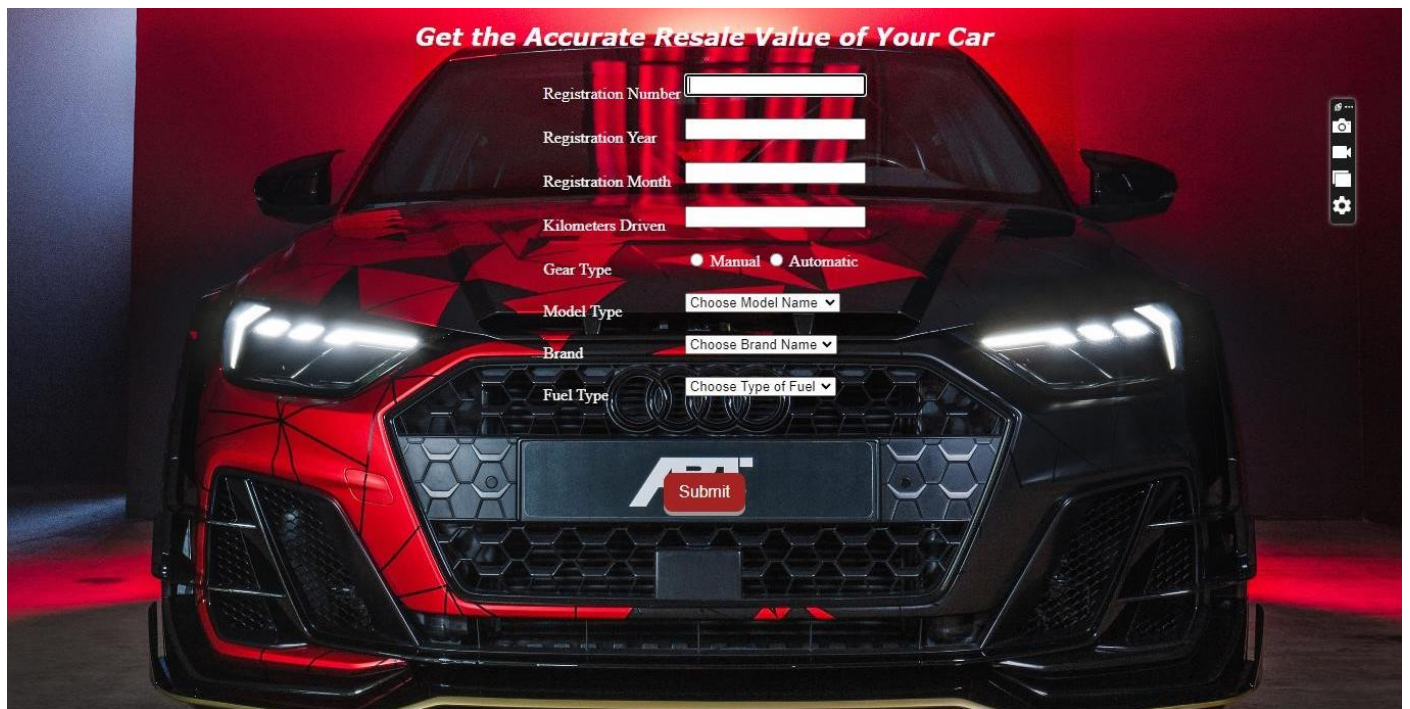
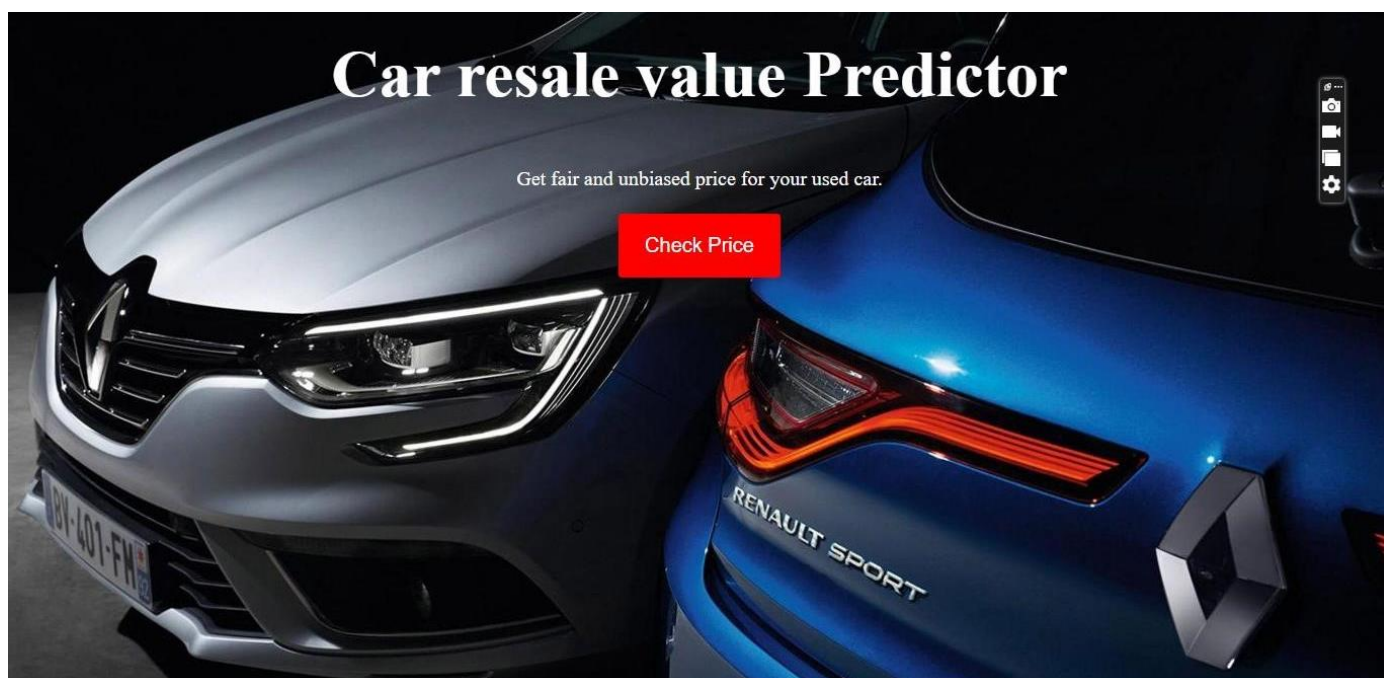
Kilometers Driven

Gear Type ☐ Manual ☐ Automatic

Model Type

Brand

Fuel Type

A screenshot of a web application for predicting car resale value. The background is a high-quality image of a red sports car, possibly an Audi RS, with its headlights on. Overlaid on the car is a white form with various input fields for user data. The form includes fields for Registration Number, Year, and Month; Kilometers Driven; Gear Type (with radio buttons for Manual and Automatic); Model Type, Brand, and Fuel Type (all with dropdown menus). A red 'Submit' button is at the bottom of the form. On the right side of the car image, there is a vertical toolbar with icons for camera, gallery, and settings.

## 9 Result

10.

### 10.1 ADVANTAGES

- This will reduced installation cost.
- It will monitor 24/7.
- Very useful to sale the car for reasonable price

### 10.2 DISADVANTAGES

- Car Resale value can not be used by the person who doesn't have access to the internet.
- Very hard to use for targeted range of people



## 11 CONCLUSION

Price prediction analyses a good or service based on its attributes, demand, and current market trends using an algorithm. The pricing is then adjusted by the programme at a level that it believes would both draw people and optimise sales. The method is known as price forecasting or predictive pricing in some quarters.

## 12 FUTURE SCOPE

When compared to February 2020, average prices were up 42.5% in September 2022. While it's possible that used vehicle prices have peaked, new car prices are expected to be high through the end of 2022. Prices are anticipated to drop for both new and used automobiles in 2023, by 2.5% to 5% for new cars and 10% to 20% for used cars.

## 13 APPENDIX

Source Code

HTML FILES:

home.html

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<style>  a:link,
```

```
  a:visited
```

```
{ background-color:
```

```
white;color:  black;
```

```
border:  2px  solid
```

```
black;border-radius:
```

```
25px; padding:
```

```
10px 20px;
```

```
text-align: center; text-
decoration: none; display:
inline-block;
}
```

```
a:hover, a:active {
background-color:
beige; color: black;
text-decoration: none;
}
</style>
```

```
<head>
```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge"> <link
rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
integrity="sha384-
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRx
T2MZw1T" crossorigin="anonymous">
<link rel="stylesheet" type="text/css" href="{ { url_for('static', filename='style.css') } }" />

<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Home</title>
</head>
```

```
<body>
```

```
<div class="col-md-8">
{ % with messages = get_flashed_messages(with_categories=true) % }
```

```
{% if messages %}
{% for category, message in messages %} <div
class="alert alert-{{ category }}">

    {{ message }}

</div>
{% endfor %}
```

```

    {% endif %}
{% endwith %}
    {% block content %} {% endblock %}
</div>
<center>
<h1>Car Resale Value Predictor</h1> <br>
<a href="{ { url_for('login') } }">Click to Login</a><br> <br>
<a href="{ { url_for('register') } }">Register here</a><br> <br>
<a href="{ { url_for('update') } }">Update Password</a><br> <br>
</center>

<script src="https://code.jquery.com/jquery-
3.3.1.slim.min.js" integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi
6jizo" crossorigin="anonymous"></script>

<script
src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/p
o
pper.min.js" integrity="sha384-
UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86
dIHNDz0W1" crossorigin="anonymous"></script>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootst
rap.min.js" integrity="sha384-
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0
7jRM" crossorigin="anonymous"></script>
</body>

</html>

```

Welcome.html(taking in inputs from the user)

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<link rel="stylesheet"
```

```
href="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/css/bootstrap.min.css"
```

```
integrity="sha384-
```

```
ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRx
```

```
T2MZw1T" crossorigin="anonymous">
```

```
<link rel="stylesheet" type="text/css" href="{ { url_for('static', filename='style.css') } }" />
```



```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Welcome Page</title>
</head>

<body>
  <marquee>Welcome!</marquee> <br>
  <center>
    <h3>Enter Car details</h3>
    </center>

    <center>
      <form method="POST" action="/predict">
        <label for="sell">Choose a Seller:</label>
        <select id="sell" name="sell">
          <option value=0>Commercial</option>
          <option value=1>Private</option>
        </select>
        <br>
        <label for="ot">Choose a OfferType:</label>
        <select id="ot" name="ot">
          <option value=0>Offer</option>
          <option value=1>Request</option>
        </select>
        <br>
        <label for="vt">Choose a Vehicle type:</label>
        <select id="vt" name="vt">
          <option value=3>Coupe</option>
```

<option value=1>Combination</option>

<option value=0>Bus</option>

<option value=2>Convertible</option>

<option value=4>Limousine</option>

<option value=7>Small</option>

```
<option value=8>SUV</option>
<option value=6>Others</option>
<option value=5>Not Declared</option>
</select>
<br>
<label for="gb">Choose a Gearbox:</label>
<select id="gb" name="gb">
<option value=0>Automatic</option>
<option value=1>Manual</option>
<option value=2>Not Declared</option>
</select>
<br>
<p>PowerPs</p>
<input name="pps" required>
<p>kilometers</p>
<input name="km" required>
<br>
<label for="ft">Choose a FuelType:</label>
<select id="ft" name="ft">
<option value=7>Petrol</option>
<option value=1>Diesel</option>
<option value=0>CNG</option>
<option value=4>LPG</option>
<option value=3>Hybrid</option>
<option value=2>Electric</option>
<option value=5>Not Declared</option>
<option values=6>Others</option>
</select>
<br>
<label for="brand">Choose a Brand:</label>
```

```
<select id="brand" name="brand">
<option value=0>Alpha Romeo</option>
<option value=1>Audi</option>
<option value=2>BMW</option>
```

<option value=3>Chevrolet</option>  
<option value=4>Chrysler</option>  
<option value=5>Citroen</option>  
<option value=6>Dacia</option>  
<option value=7>Daewoo</option>  
<option value=8>Daihatsu</option>  
<option value=9>Fiat</option>  
<option value=10>Ford</option>  
<option value=11>Honda</option>  
<option value=12>Hyundai</option>  
<option value=13>Jaguar</option>  
<option value=14>Jeep</option>  
<option value=15>Kia</option>  
<option value=16>Lada</option>  
<option value=17>Lancia</option>  
<option value=18>Land Rover</option>  
<option value=19>Mazda</option>  
<option value=20>Mercedes</option>  
<option value=21>Mini</option>  
<option value=22>Mitsubishi</option>  
<option value=23>Nissan</option>  
<option value=24>Opel</option>  
<option value=25>Peugeot</option>  
<option value=26>Porsche</option>  
<option value=27>Renault</option>  
  
<option value=28>Rover</option>  
<option value=29>Saab</option>  
<option value=30>Seat</option>  
<option value=31>Skoda</option>

<option value=34>Subaru</option>

<option value=35>Suzuki</option> <option value=36>Toyota</option>

<option value=38>Volkswagen</option>

<option value=39>Volvo</option>

```
</select>
<br>
<label for="nr">Repaired?</label>
<select id="nr" name="nr">
  <option value=1>Yes</option>
  <option value=0>No</option>
  <option value=2>Not Declared</option>
</select>
<br>
<p>Age(in months)</p>
<input name="age" required>
<br>
<br>
<button type="submit">Submit</button>
</form>
</center>

<br>
<br>
<a href="{ { url_for('logout') } }">Press here to logout</a>
```

```

      <script      src="https://code.jquery.com/jquery-
3.3.1.slim.min.js"      integrity="sha384-
q8i/X+965DzO0rT7abK41JStQIAqVgRVzpbzo5smXKp4YfRvH+8abtTE1Pi
6jizo" crossorigin="anonymous"></script>
```

```

      <script
      src="https://cdn.jsdelivr.net/npm/popper.js@1.14.7/dist/umd/p
o
pper.min.js" integrity="sha384-
```

UO2eT0CpHqdSJQ6hJty5KVphtPhzWj9WO1clHTMGa3JDZwrnQq4sF86  
dIHNDz0W1" crossorigin="anonymous"></script>

<script  
src="https://cdn.jsdelivr.net/npm/bootstrap@4.3.1/dist/js/bootst  
rap.min.js" integrity="sha384-  
JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B0  
7jRM" crossorigin="anonymous"></script>

</body>

</html>

Predict.html(For displaying the output)



```

<!DOCTYPE html>
<html lang="en">
  <style>
    body{
      background-color:
      pink;color:black;
    }
  </style>
<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">

  <title>Car Value</title>
</head>
<body>
  <h1>Car Value Predicted is</h1>

  <h1>{{predict1}}</h1>
  <br>
  <br>

</body>
</html>

```

app-ibm.py(FLASK FRAMEWORK)

```

from flask import Flask,request, render_template, url_for, redirect, flash
from flask_login import UserMixin, login_user, LoginManager, login_required,
logout_user,current_user

```

```
from wtforms.validators import InputRequired, Length,  
ValidationError  
from flask_bcrypt import Bcrypt
```

```

from wtforms import StringField, PasswordField,
SubmitField, IntegerField
from flask_wtf import FlaskForm
import sqlite3
import joblib

from sklearn.preprocessing import
OrdinalEncoder
import pandas as pd
import requests

```

# NOTE: you must manually set API\_KEY below using information retrieved from your IBM Cloud account.

```

API_KEY = "iX5xF0JyPhYIfWGg37VPe14p1D7OMDPgPG1cl1yQJfJg"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token',
data={"apikey":API_KEY, "grant_type":
'urn:ibm:params:oauth:granttype:apikey'})
mltoken = token_response.json()["access_token"]
header = {'Content-Type':
'application/json', 'Authorization': 'Bearer ' +

```

```

mltoken}
app = Flask( name )
bcrypt = Bcrypt(app)

```

---

<sup>1</sup> A3E'

```

login_manager = LoginManager()
login_manager.init_app(app)
login_manager.login_view =
'login'

```

```

app.config['SECRET_KEY'] =
'B7@login_manager.user_loa
derdef load_user(user_id): conn =
connect_db() user =
conn.execute('SELECT * FROM
user WHERE id = ?',
              (user_id,)).fetchone()
usr_obj = User(user[0], user[1],
user[2])return usr_obj

```

```
def connect_db():  
    conn =  
    sqlite3.connect('database.db')
```

```
return conn  
class User:
```

```
    def __init__(self, id, email,  
        username):  
        self.id = id  
        self.username =  
        username  
        self.email =  
        email
```

```
    def to_json(self):  
        return {"username":  
            self.username, "email":  
            self.email}
```

```
    def is_authenticated(self):  
        return True
```

```
    def is_active(self):  
        return  
        True
```

```
def is_anonymous(self):  
    return False  
def get_id(self):  
    return  
    str(self.id)
```

```
class  
    RegisterForm(FlaskForm):  
        email =  
        StringField(validators=[  
            InputRequired(),            Length(min=4,            max=50)],  
        render_kw={"placeholder": "Email"})username =  
        StringField(validators=[
```

```

            InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})
    rollnumber = StringField(validators=[
        InputRequired(), Length(min=5, max=10)], render_kw={"placeholder":
"RollNumber"})
    password = PasswordField(validators=[
        InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"})
    submit = SubmitField('Register')

```

```

def validate_username(self,
    username):
    conn = connect_db()
    existing_user_username = conn.execute('SELECT * FROM user WHERE username = ?',(username.data,)).fetchone()

    conn.commit()
    if existing_user_username:
        raise ValidationError('That username already exists. Try another one.')

```

```

class LoginForm(FlaskForm):
    username = StringField(validators=[
        InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":

```

```
"Username" }) password =  
  
    PasswordField(validators=[  
  
        InputRequired(), Length(min=8,  
  
        max=20)], render_kw={"placeholder":  
  
"Password" }) submit =  
  
    SubmitField('Login')
```

```
class UpdateForm(FlaskForm):
```



```
username = StringField(validators=[
    InputRequired(), Length(min=4, max=20)], render_kw={"placeholder":
"Username"})
```

```
oldpassword = PasswordField(validators=[
    InputRequired(), Length(min=8, max=20)],
render_kw={"placeholder": "PreviousPassword"})
```

```
password = PasswordField(validators=[
    InputRequired(), Length(min=8, max=20)], render_kw={"placeholder":
"Password"}) submit =
```

```
SubmitField('Update')
```

```
@app.route('
/')def
home(): return
    render_template('home.html')
```

```
@app.route('/login',
methods=['GET', 'POST'])def
login(): form = LoginForm()

    if
        form.validate_on_submit():conn = connect_db() user =
        conn.execute('SELECT * FROM user WHERE username = ?',
```

```
        (form.username.data,)).fetchone()
conn.comm

it()
conn.close()

if user:

    if      bcrypt.check_password_hash(user[4],
        form.password.data):usr_obj = User(user[0],
        user[1], user[2]) login_user(usr_obj) return
    redirect(url_for('welcome'))
```

```

    else:
        print('Hi') flash(f'Invalid credentials, check and try logging
        in again.',
        'danger')return redirect(url_for('login'))

return render_template('login.html', form=form)

```

```

@app.route('/welcome', methods=['GET',
'POST'])@login_required def welcome():
return render_template('welcome.html')

```

```

@app.route('/predict',
methods=['POST'])def
predictSpecies():

    sell =
    float(request.form['sell'])ot
    = float(request.form['ot'])
    vt =
    float(request.form['vt'])
    gb =
    float(request.form['gb'])
    pps=float(request.form['p ps'])
    km=float(request.form['k m'])
    ft=float(request.form['ft'])
    brand=float(request.form['brand']
) nr=float(request.form['nr'])

```

```
age=float(request.form['age']) arr
= [[sell, ot, vt,
gb,pps,km,ft,brand,nr,age]]
```

```
payload_scoring = {"input_data": [{"field": ['sell', 'ot', 'vt',
'gb','pps','km','ft','brand','nr','age']], "values":arr}}}
```

```
response_scoring
```

```
=requests.post('https://ussouth.ml.cloud.ibm.com/ml/v4/deployments/a4a92034-8fcd-4e79-
ab7c-
```

```

521a5d8cb7d5/predictions?version=2022-11-15', json=payload_scor
ing,headers={'Authorization': 'Bearer ' + mltoken})
print(response_scoring) predictions =
response_scoring.json() pr =
predictions['predictions'][0]['values'][0][0]
print("final prediction",pr) return
render_template('predict.html',predict1=pr)

```

```

@app.route('/logout', methods=['GET',
'POST'])@login_required
def logout(): logout_use

r() return
redirect(url_for('login'))

```

```

@ app.route('/register', methods=['GET',
'POST'])def register(): form =
RegisterForm()
conn =
connect_db() if
form.validate_on_submit():

hashed_password = bcrypt.generate_password_hash(form.password.data)
conn.execute('INSERT INTO user (email, username, roll_number,
pass_word) VALUES (?,
?, ?, ?)',

```

```
        (form.email.data, form.username.data,
form.rollnumber.data, hashed_password))conn.commit()
conn.close() return redirect(url_for('login'))
return render_template('register.html', form=form)
```

```
@ app.route('/update', methods=['GET',
'POST'])def update():
```

```

form =
UpdateForm()conn      =
connect_db()           if
form.validate_on_submit():

    conn = connect_db() user = conn.execute('SELECT * FROM user
    WHERE username = ?',

        (form.username.data,)).fetchone()
    if user:

        if bcrypt.check_password_hash(user[4],
            form.oldpassword.data):print(user)
            hashed_password1 =

                bcrypt.generate_password_hash(form.passw
                ord.data)

            conn.execute('UPDATE user set pass_word = ? where username = ?',
                (hashed_password1,
                form.username.data)) conn.commit()
            conn.close() flash(f'Password changed
            successfully.',

                'success')return      redirect(url_for('home'))
        else:

            flash(f'Invalid password, Enter valid password.',
                'danger')return      redirect(url_for('update'))
        else:

            flash(f'Invalid user, Enter valid User.',
                'danger')return      redirect(url_for('update'))
    return render_template('update.html', form=form)

```

```

if name == " main ": app.run(debug=True)

```

GITHUB LINK : <https://github.com/IBM-EPBL/IBM-Project-27634-1660061500>

DEMOLINK:

<https://drive.google.com/file/d/1MRwCx6NABK7wluvzyiT55QuKmDAM1L13/view?usp=sharing>

THANKYOU