Car Resale value Prediction using Applied Data Science

by

Mothish Lakshman S - 727619BIT003

Ranjthkumar S - 727619BIT025

Madhavan R - 727619BIT005

Arjun G - 727619BIT033

Department of Information Technology

Dr. Mahalingam College of Engineering and Technology

November 2022

INDEX

CHAPTER – 1

INTRODUCTION

1.1 Project Overview

Car resale value prediction is the system to predict the amount of resale value based on the parameters provided by the user. User enters the details of the car into the form given and accordingly the car resale value is predicted.In order to predict the resale value of the car, we proposed an intelligent, flexible, and effective system that is based on using regression algorithms. Considering the main factors which would affect the resale value of a vehicle a regression model is to be built that would give the nearest resale value of the vehicle. We will be using various regression algorithms and algorithm with the best accuracy will be taken as a solution, then it will be integrated to the web-based application where the user is notified with the status of his product.

1.2 Purpose

The purpose of this project is to decide whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately[2-3]. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

CHAPTER – 2

LITERATURE SURVEY

2.1 Existing System

Existing System includes a process where a seller decides a price randomly and buyer has no idea about the car and it's value in the present day scenario. In fact, seller also has no idea about the car's existing value or the price he should be selling the car at. In existing systems, the prediction of second-hand car price is not accurate.

2.2 Literature Survey

Paper 1- Car Price Prediction UsingMachine Learning Techniques

In the first existingsurvey **Ref- TEM Journal.Volume 8, Issue 1, Pages 113-118,ISSN 2217-8309, DOI:10.18421/TEM81-16, February 2019.**

'Car Price Prediction Using Machine Learning Techniques' according to authors Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric, in this paper they mainly concentrate on collecting various data from web portal by using web scrap techniques. And those have been compared with the help of different machine learning algorithms to predict the vehicle price in easy manner. They classified the price according to different ranges of price that is already given. Artificial neural network, support vector machine, random forest algorithms were used on different datasets to build classifiers model. In the existing system, to predict the price of vehiclesboth two wheelers and four wheelers, a lot of data mining algorithms and machine learning algorithms were widely used. The major drawback of this existing system is they needmore attributes in order to predict the vehicle price. More comparison techniques must be used to get the result more efficiently. It is highly complicated to get sufficient data sets that were spread widely all over the world. The datasets can be collected only through online. But not on the offline mode. It is not possible for everyone to collect thedata sets through online mode particularly in rural areas. The data sets will not have about the vehicles which were not used for long time and also the traditional model vehicles may or may not be included in the data sets.

Paper 2 - Predicting the Price of Used Cars Using Machine Learning Techniques

In the second existingsurvey **Ref: International Journalof Information and Computation Technology. ISSN 0974-2239 Volume 4, Number 7 (2014).**

'Predicting the Price of Used Cars Using Machine Learning Techniques' according to author Sameerchand Pudaruth they have done the predictions of car price from the historical data that has been collected from daily newspapers. For predicting the price of vehicles, they 4 have used supervised machine learning techniques. Other algorithmswere also used to predictsuch as multiple linear regression, some decision tree algorithms. All these algorithms were compared and found the best algorithm for prediction. They have faced some difficulties in comparing the algorithms, somehow they have managed. The major drawbacks of existing system are the system is very slow due to most of the works about the keyword query just analyze individual points, and they are inappropriate to many applications that call for analysis of groups of different vehicle points.In the existing system shown above, authorsproposed prediction model based on the single machine learningalgorithm. However, it is noticeable that single machine learning algorithm approach did not give remarkable prediction resultsand could be enhanced by assembling various machine learning methods in an ensemble.In future, gainingadvancement in this system model we couldrely on this to predict the value. Generally, e-commerce platforms attract customers in different ways to use their system for buying or selling and the algorithm which is usedin their system is such that the value is not accurate It is overpriced when a customer wants to buy a car and vice versa while selling. Also referring to the tremendous loss reported of worth more than a billion dollarsin Germany due to mis - calculation of the car value which could be overcome using this prediction application. Further, we may add large historical data of car price which can help to improveaccuracy of the machinelearning model. We can build an android app for better user interaction and for betterperformance, we plan to judiciously design deep learningnetwork structures, use adaptive learning rates and train on clusters of data ratherthan the wholedataset.

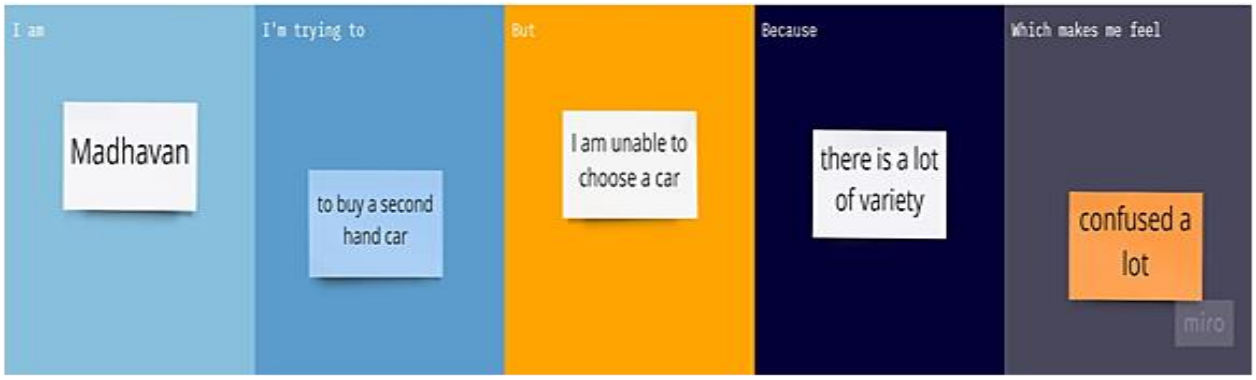
2.3 Problem Statement Definition

Due to the huge requirement of used cars and lack of experts who can determine the correct valuation . A Effective solution to predict used cars prices by scraping data from websites that sell used cars, and analysing the different aspects and factors that lead to the actual used car price valuation.

- To help guide the individuals looking to buy or sell cars and to give them a better insight into the automotive sector.

- Therefore, to help consumers avoid falling victims to some dealer, this car resale value prediction hopes to equip consumers with right tools to guide them in their shopping experience.
- Another goal of the project is to explore new methods to evaluate used cars prices and to compare their accuracie.

Example:



Problem Statement(PS)	**I am (Customer)**	**I'm trying to**	**But**	**Because**	**Which makes me feel**
PS-1	Entrepreneur	Buy a used car	I am unaware of the varieties available	I don't have correct guidance	Inferior to others
PS-2	Business women	Find a good second hand car	I am unaware about the price and other factors	I am unable to know the current updates	Stressed

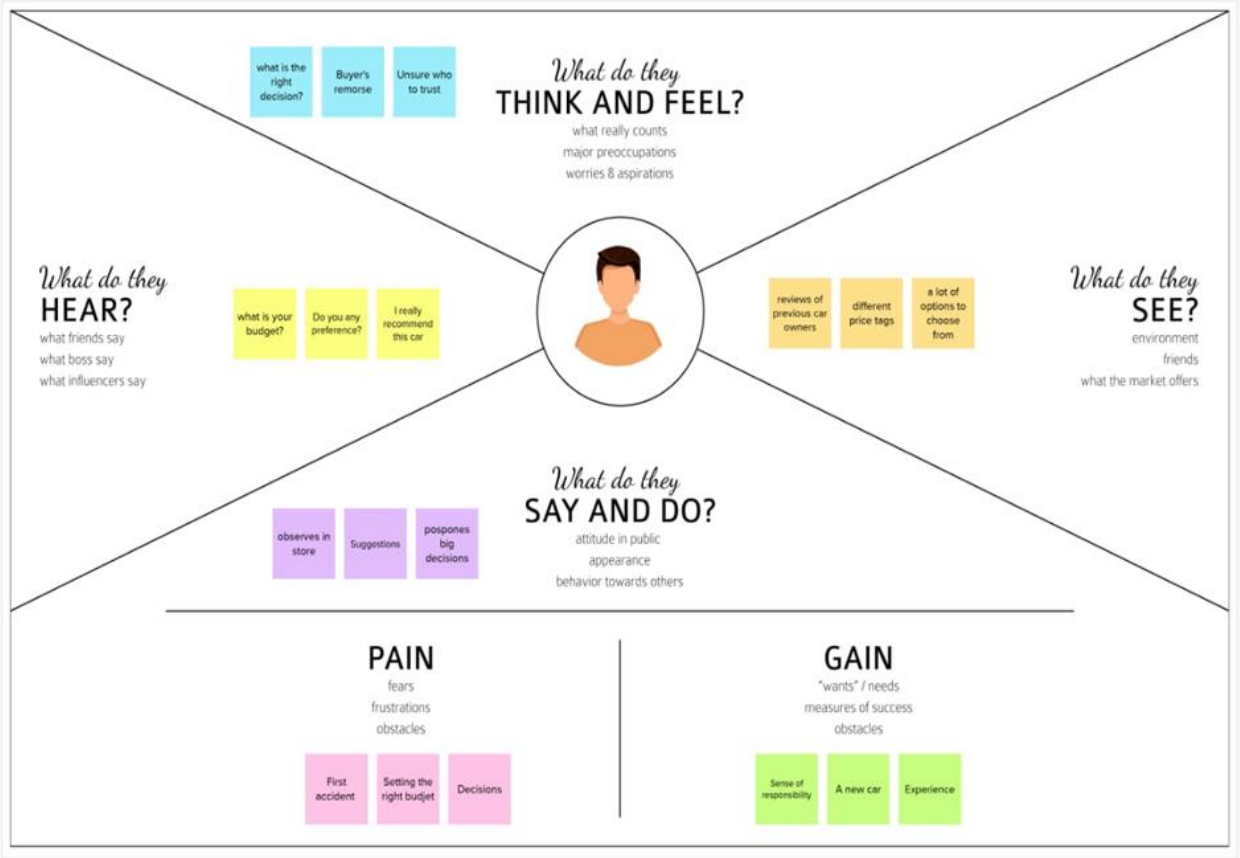
CHAPTER – 3

IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes.

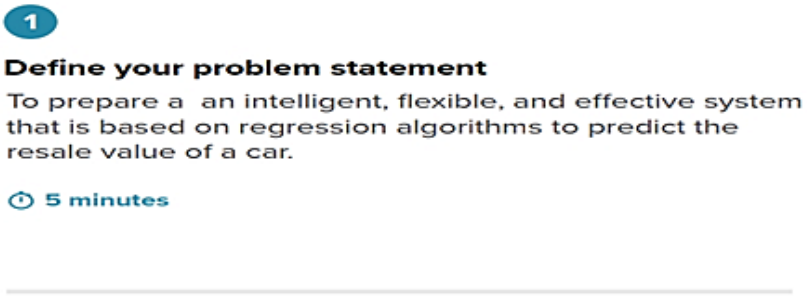
Example:

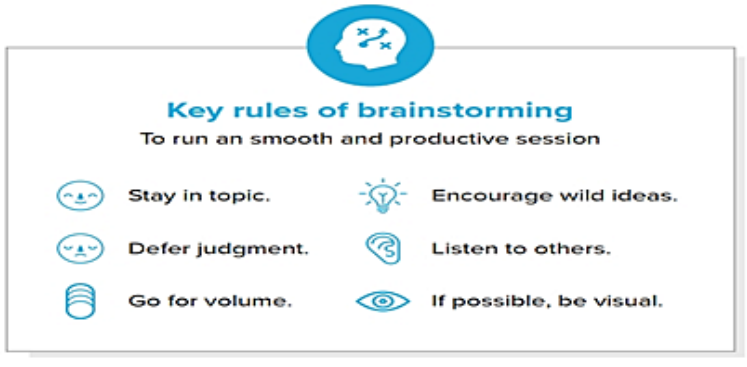


3.2 Ideation & Brainstorming

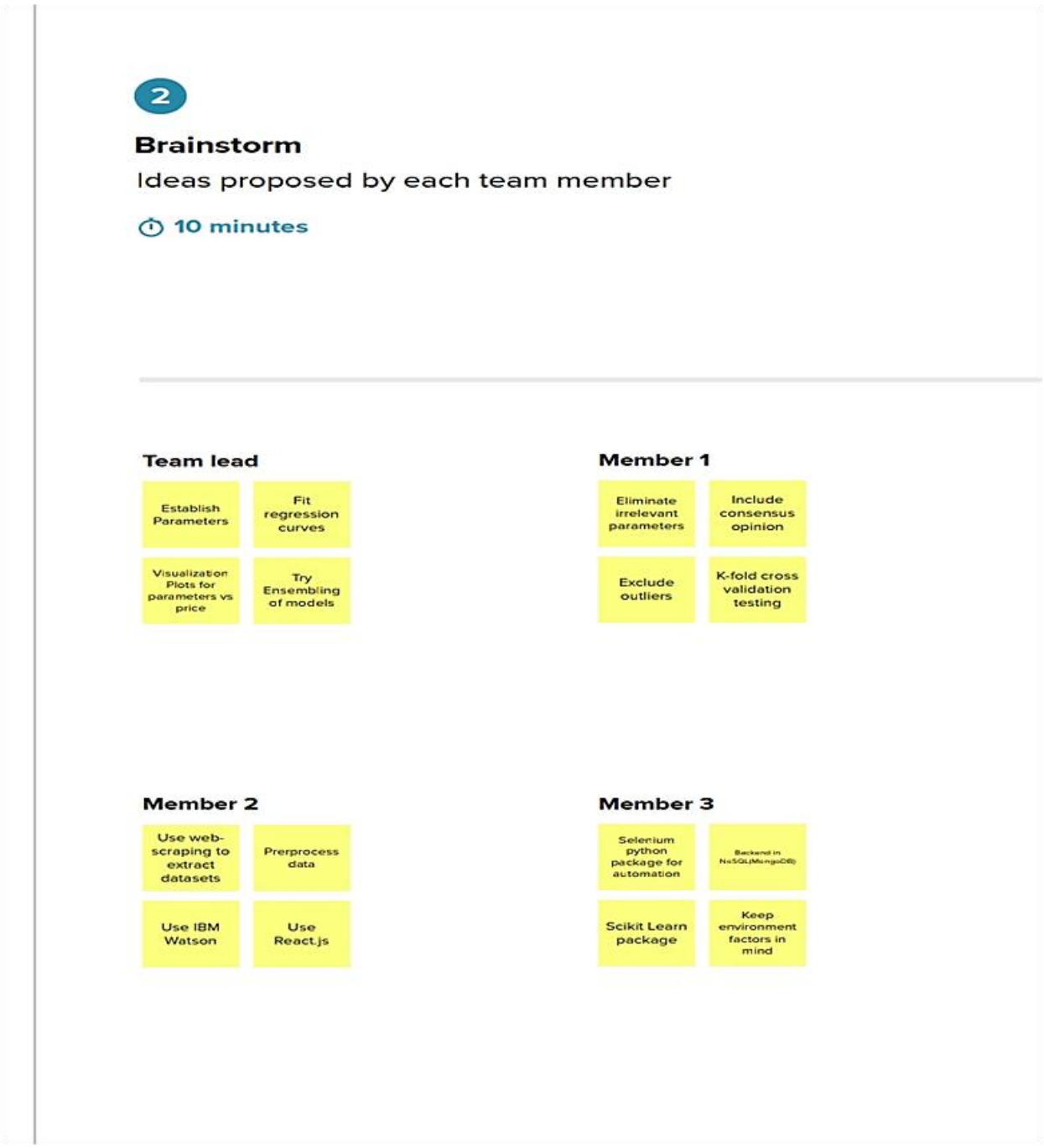
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

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

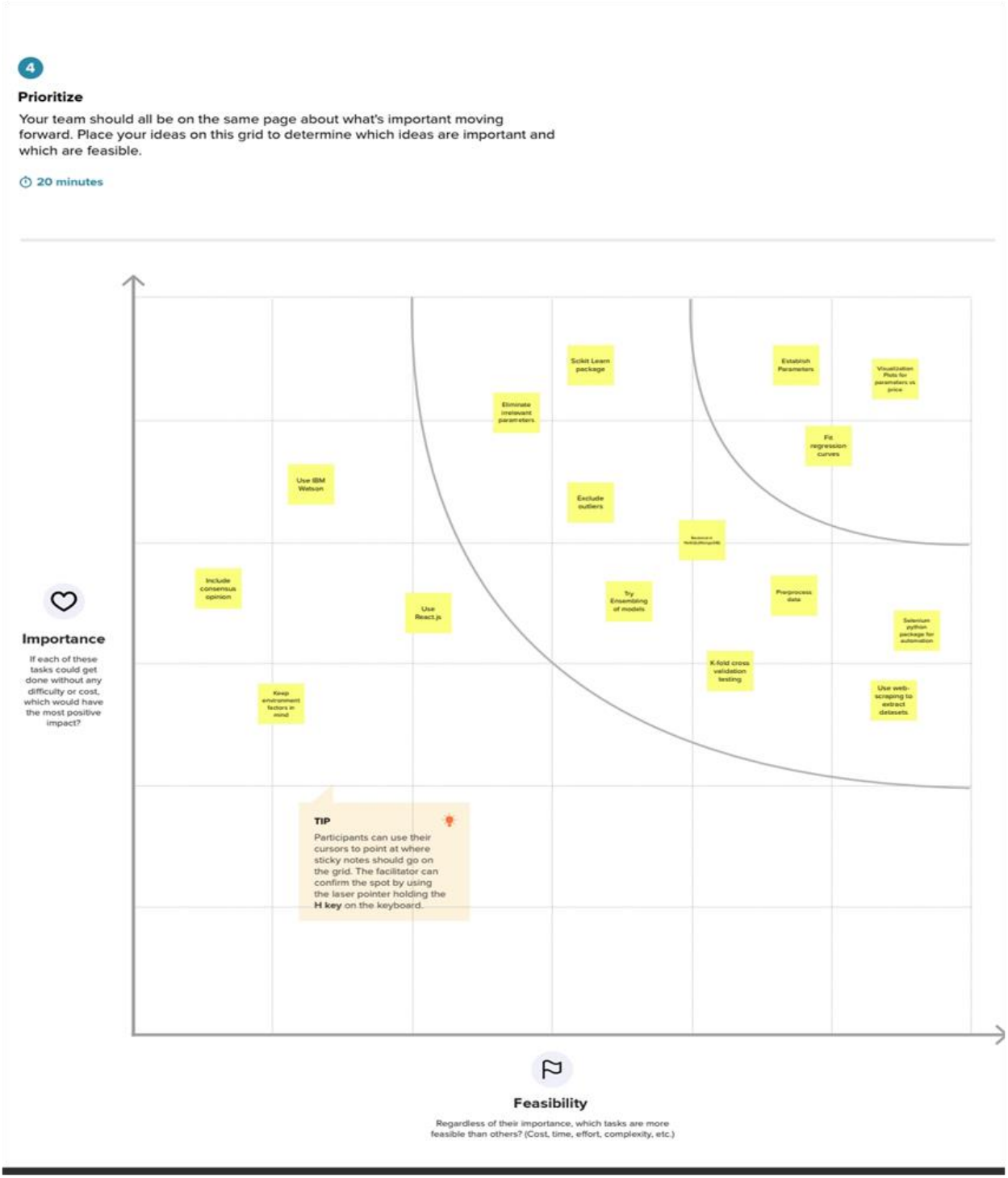




Step-2: Brainstorm, Idea Listing and Grouping



Step-3: Idea Prioritization



3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	For the purposes of car valuation, popular guides tend not to use machine learning. Instead, they source data from local sales and average the prices of many similar cars. This method works well if you have a common car with a common set of features. The condition of the car is judged very roughly, typically on a scale of one to three. Cars that are "unusual" are therefore hard to evaluate. Effectively, no inferences are drawn from similar cars but from a different make and model, whereas with machine learning, the entirety of the dataset and its features are used to train the model predictions. Using machine learning is a solution to the problem of utilization of all the data and will assist in utilizing all the features of a car to make valuations.
2.	Idea / Solution description	New cars of a particular make, model, and year all have the same retail price, excluding optional features. This price is set by the manufacturer. Used car, however, are subject to supply-and-demand pricing. Further, used cars have additional attributes that factor into the price. These include the condition, milage, and repair history, which sets cars that may have shared a retail price apart.
3.	Novelty /Uniqueness	The purpose of this thesis is to evaluate several different machine learning models for used car price prediction and draw conclusions about how they behave. This will deepen the knowledge of machine learning applied to car valuations and other similar price prediction problems.

4.	Social Impact / Customer Satisfaction	This work will focus on answering the research questions. They all entail a comparison of different ML algorithms for price prediction. This will be accomplished by sourcing and preparing a dataset on which all the algorithms can be trained on and compared fairly. The algorithms selected must therefore be similar enough for the same dataset to be used for all of them. This also means that no large optimization efforts on the dataset will be made to boost the performance, if these changes do not benefit the other models. Maximizing price prediction performance of any one algorithm in ways that do not offer better comparisons is outside the scope of this work.
5.	Business Model (Revenue Model)	A revenue model is a blueprint that shows how a startup business will earn revenue or gross income from its standard business operations, and how it will pay for operating costs and expenses
6.	Scalability of the Solution	which of the models and parameters gives the best overall accuracy in making price predictions for used cars. The optimal parameters were determined in the process of implementing the models, and thus each model was implemented with the parameters that yielded the best performance by trial and error All of the models approximated geometric appreciation, meaning that a constant percentage of value is lost every year independent of the age of the vehicle. Random Forest Regression had a significantly higher assessed average depreciation at approximately 13.8%, compared to the others with 9.7%. This is closer to the range of 15%- 31% assessed by Karl Storchmann in his analysis of international depreciation rates

3.4 Problem Solution Fit

CHAPTER – 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	Car Registration	Registration through website
FR-2	Car Value Prediction	Predicting the car resale value
FR-3	Suggesting Buyers	Suggesting the buyer through website

4.2 Non-Functional Requirements

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

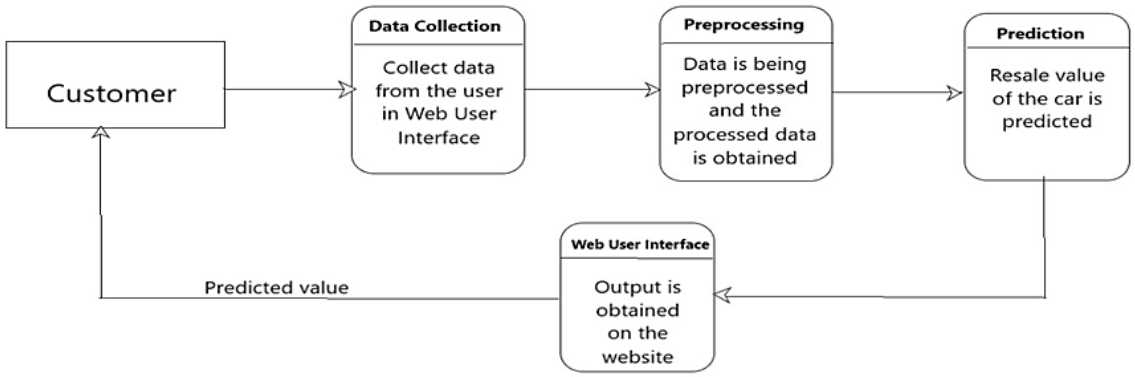
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Predicting the car's resale value
NFR-2	**Security**	Providing base level security for website
NFR-3	**Reliability**	Providing accurate resale value which is reliable
NFR-4	**Performance**	Enhancing performance by using efficient machine learning algorithm
NFR-5	**Availability**	Available anytime
NFR-6	**Scalability**	Can handle many users at a time

CHAPTER – 5

PROJECT DESIGN

5.1 Data Flow Diagram

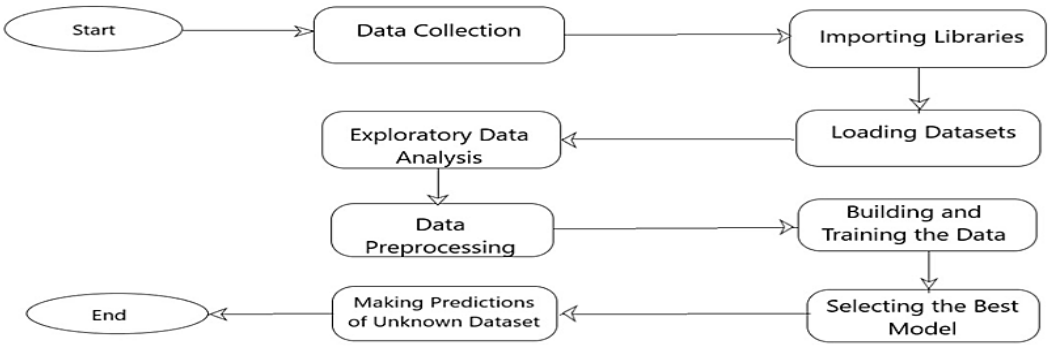
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.



5.2 Solution Architecture

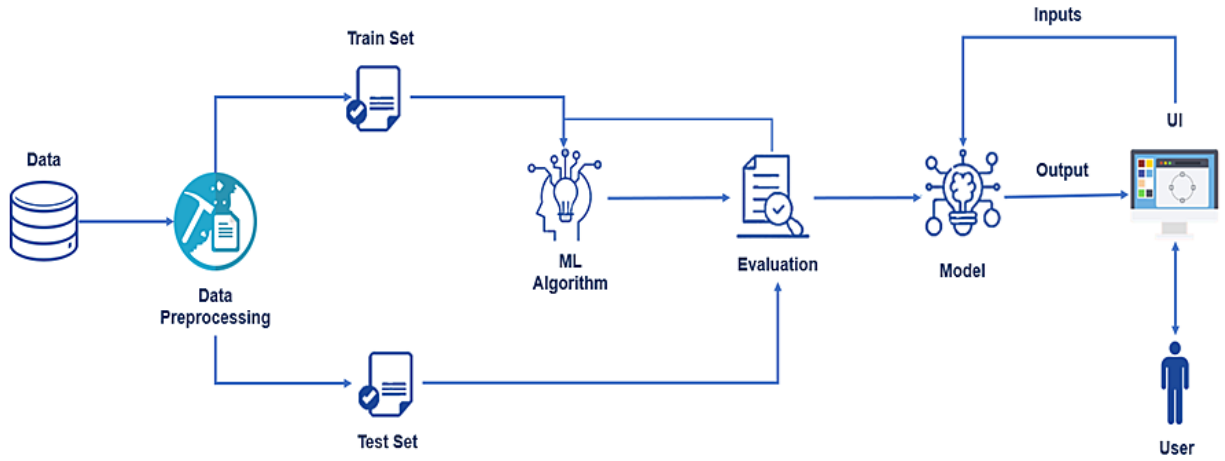
Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

Example - Solution Architecture Diagram:



Technical Architecture

The deliverable shall include the architectural diagram as below and the information as per the table.



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Desktop user)	Data Entry	USN-1	User can either the car details in the website	Enter the car details	Medium	Sprint-1
Customer (Desktop User)	Obtain Output	USN-2	User will receive car resale value in the website	Receives the car resale value	High	Sprint-1

CHAPTER – 6

PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Dataset reading and Pre processing	USN-1	Cleaning the dataset and splitting to dependent and independent variables	2	High	Madhavan R Mothish Lakshman S
Sprint-2	Building the model	USN-2	Choosing the appropriate model for building and saving the model as pickle file	1	High	Arjun G Ranjithkumar S
Sprint-3	Application building	USN-3	Using flask deploying the ML model	2	Medium	Ranjithkumar S Arjun G
Sprint-4	Train the model in IBM	USN-4	Finally train the model on IBM cloud and deploy the application	2	Medium	Mothish Lakshman S Ranjithkumar S

Project Tracker, Velocity & Burndown Chart:

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	15	5 Days	01 Oct 2022	06 Oct 2022	15	06 Oct 2022
Sprint-2	15	5 Days	07 Oct 2022	12 Nov 2022	15	12 Oct 2022
Sprint-3	15	5 Days	13 Nov 2022	17 Nov 2022	15	17 Oct 2022
Sprint-4	15	5 Days	18 Nov 2022	23 Nov 2022	15	23 Oct 2022

Velocity:

Imagine we have a 5-day sprint duration, and the velocity of the team is 15(points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

Actual Velocity = Sprint Duration/Velocity=15/5=3

Burndown Chart:

A burn down chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

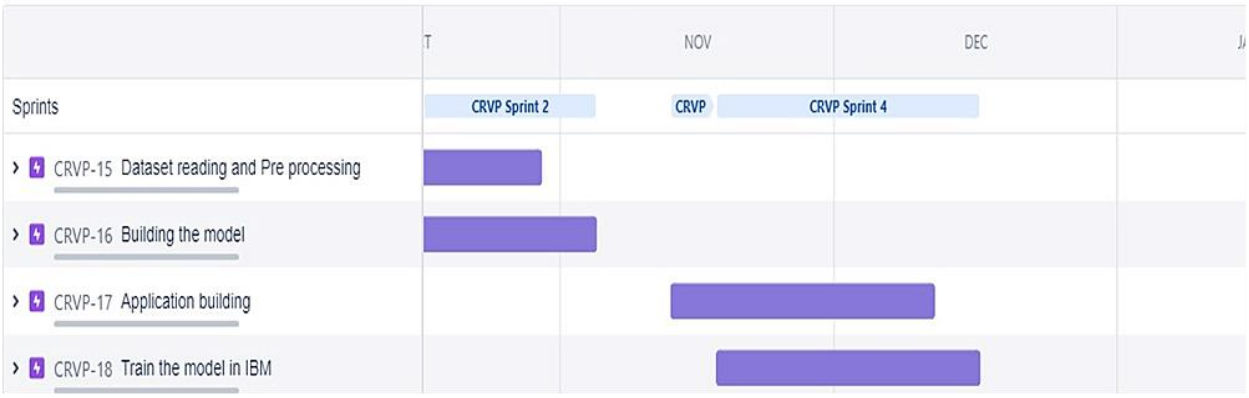
6.2 Sprint Delivery Schedule

TITLE	DESCRIPTION	DATE
Literature Survey & Information Gathering	Literature survey on the selected project & gathering information by referring the, technical papers, research publications etc	19 September 2022
Prepare Empathy Map	Prepare Empathy Map Canvas to capture the user Pains & Gains, Prepare list of problem statements	19 September 2022
Ideation	List the by organizing the brainstorming session and prioritize the top 3 ideas based on the feasibility & importance.	19 September 2022
Proposed Solution	Prepare the proposed solution document, which includes the novelty, feasibility of idea, business model, social impact, scalability of solution, etc.	24 September 2022
Problem Solution Fit	Prepare problem - solution fit document.	30 September 2022
Solution Architecture	Prepare solution architecture document.	20 September 2022

Customer Journey	Prepare the customer journey maps to understand the user interactions & experiences with the application (entry to exit).	3 October 2022
Functional Requirement	Prepare the functional requirement document	4 October 2022
Data Flow Diagrams	Draw the data flow diagrams and submit for review.	4 October 2022
Technology Architecture	Prepare the technology architecture diagram	4 October 2022
Prepare Milestone & Activity List	Prepare the milestones & activity list of the project.	10 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4 Develop & submit the delivery of Sprint.	Develop & submit the developed code by testing it.	IN PROGRESS...

6.3 Reports from JIRA

Project Milestones:



Project Planning Tool:

- Atlassian Jira

Reference:

https://www.atlassian.com/roadmap/cloud

CHAPTER – 7

CODING & SOLUTIONING

7.1 Analysis of Algorithm:

There are several algorithms a person can use to predict a value as every algorithm itself has different significance and its own equation to predict. Algorithms like linear regression, Lasso regression, Ridge regression, Random forest regression, XGBoost, etc. have different equation through which we get a predicted value using past data.

Random Forest Regression:

We will see steps that has helped us to build this model for car value prediction. Firstly, we have import dataset of different cars and their year, kms driven, fuel type, transmission, etc. These are the attributes which we are going to need to predict. Then we will be using feature engineering where we will correlate between attributes in our dataset.

ALGORITHMS	ACCURACY
LinearRegression	67.11 %
ElasticNet	58.11 %
RidgeRegression	67.11 %
RandomForest Regressor	86.72 %
LassoRegression	58.11 %

Here we can see Random forest regression is getting more accuracy than rest of the algorithms after comparing with our data which we are using in this project to predict the car value.

Dataset Description:

Dataset was collected from the website called Kaggle. Kaggle allows users to find and publish datasets, explore and build in a web based data science environment, work with other data science environment, work with other data scientists and machine learning engineers. The data set includes car name, year, selling price, present price, kms driven, fuel type, seller type, Transmission type and owner.

Explanation of code:

Importing Dataset:

The data consists of some rows and columns.Since our target is to find the selling price,the target attribute y is also selling price,remaining features are taken for analysis and predictions.

Reading Data

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
df=pd.read_csv("train.csv")
df.head(5)
```

Out[2]:

	Name	Location	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	New_Price	Price
0	Maruti Wagon R LXI CNG	Mumbai	2010	72000	CNG	Manual	First	26.6 km/kg	998 CC	58.16 bhp	5.0	NaN	1.75
1	Hyundai Creta 1.6 CRDi SX Option	Pune	2015	41000	Diesel	Manual	First	19.67 kmpl	1582 CC	126.2 bhp	5.0	NaN	12.50
2	Honda Jazz V	Chennai	2011	46000	Petrol	Manual	First	18.2 kmpl	1199 CC	88.7 bhp	5.0	8.61 Lakh	4.50
3	Maruti Ertiga VDI	Chennai	2012	87000	Diesel	Manual	First	20.77 kmpl	1248 CC	88.76 bhp	7.0	NaN	6.00
4	Audi A4 New 2.0 TDI Multitronic	Coimbatore	2013	40670	Diesel	Automatic	Second	15.2 kmpl	1968 CC	140.8 bhp	5.0	NaN	17.74

In [3]:
```
df.shape
```

Out[3]: (6019, 13)

Data Pre-Processing:

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. While creating a machine learning project, it is not always a case that we come across the clean and formated data and while doing a any operation with data, it is mandatory to clean it and put in a formatted way.So for this, we use data preprocessing task.

Cleaning and Pre-Processing

In [8]:
```
#removing kmpl and km/kg from mileage column
df['Mileage'] = df['Mileage'].apply(lambda x: str(x).replace('kmpl', '') if 'kmpl' in str(x) else str(x))
df['Mileage'] = df['Mileage'].apply(lambda x:str(x).replace('km/kg', '') if 'km/kg' in str(x) else str(x))
#removing CC from engine column
df['Engine'] = df['Engine'].apply(lambda x: str(x).replace('CC', '') if 'CC' in str(x) else str(x))
#removing bhp from power column
df['Power'] = df['Power'].apply(lambda x: str(x).replace('bhp', '') if 'bhp' in str(x) else str(x))
```

In [9]:
```
df['Mileage'] = pd.to_numeric(df['Mileage'], errors='coerce')
df['Engine'] = pd.to_numeric(df['Engine'], errors='coerce')
df['Power'] = pd.to_numeric(df['Power'], errors='coerce')
```

In [10]:
```
df['Mileage'].mode()
```

Out[10]:
```
0    17.0
Name: Mileage, dtype: float64
```

In [11]:
```
df['Mileage'].fillna(value=17.0,inplace=True)
```

In [12]:
```
df['Engine'].mode()
```

Out[12]:
```
0    1197.0
Name: Engine, dtype: float64
```

In [13]:
```
df['Engine'].fillna(value=17.0,inplace=True)
```

Elimination of unwanted features:

The Extra Trees Regressor library allows you to view feature importance and thereby remove the less important features from the data. It is always advised to remove the unnecessary feature because they can definitely yield better accuracy scores.

```
In [20]: #dropping the Name ,Location and new_price column
         df1_map=df.drop(["Name","Location","New_Price"],axis='columns')

In [21]: df1_map.head(5)
```

Out[21]:

	Year	Kilometers_Driven	Fuel_Type	Transmission	Owner_Type	Mileage	Engine	Power	Seats	Price	Brand_Name
0	2010	72000	CNG	Manual	First	26.60	998.0	58.16	5.0	1.75	Maruti
1	2015	41000	Diesel	Manual	First	19.67	1582.0	126.20	5.0	12.50	Hyundai
2	2011	46000	Petrol	Manual	First	18.20	1199.0	88.70	5.0	4.50	Honda
3	2012	87000	Diesel	Manual	First	20.77	1248.0	88.76	7.0	6.00	Maruti
4	2013	40670	Diesel	Automatic	Second	15.20	1968.0	140.80	5.0	17.74	Audi

```
In [22]: sns.heatmap(df1_map.isnull())
         plt.show()
```

Training and Test Split:

The train-test split used to estimate the performance of machine learning algorithms that are applicable for prediction based algorithms. This method is a fast and easy procedure to perform such that we can compare our own machine learning model results to machine results.

Train Test Split

```
In [53]: from sklearn.model_selection import train_test_split
         X_train,X_test,y_train,y_test=train_test_split(df3_inputs,df3_target,test_size=0.3,random_state=123)
```

Model Selection

```
In [54]: X_train
```

Out[54]:

	Brand_Name_n	Year	Kilometers_Driven	Fuel_Type_n	Transmission_n	Owner_Type_n	Mileage	Engine	Power	Seats
425	10	2017	23163	4	1	0	15.29	1591.0	121.3	5.0
2557	10	2013	84600	4	1	2	21.10	814.0	55.2	5.0
187	17	2017	24773	4	1	0	21.40	1197.0	83.1	5.0
5115	17	2010	90246	4	1	0	16.10	1298.0	88.2	5.0
3933	29	2015	100000	1	0	0	13.50	2400.0	215.0	5.0
...	...	...	...	...	...	...	...	...	...	...
5220	10	2015	14925	4	1	0	18.50	1197.0	82.9	5.0
4062	17	2015	30000	4	1	0	20.73	1373.0	91.1	5.0
1346	9	2012	66245	4	1	0	17.00	1497.0	118.0	5.0
3455	17	2014	41000	0	1	0	26.20	998.0	58.2	5.0
3583	10	2011	58000	4	1	2	17.01	1591.0	121.3	5.0

4211 rows × 10 columns

Training the model:

We have used the random forest Regressor to predict the selling prices since this is a regression problem and that random forest uses multiple decision trees and has shown good results for my model.

Best Model - Random Forest

In [14]:
```
Model_RandomForest = RandomForestRegressor(max_features='sqrt', bootstrap=True)
Model_RandomForest.fit(X_train, y_train)
Model_RandomForest.score(X_test, y_test)
```

Out[14]: 0.8560922241801848

In [15]:
```
import pickle
filename = 'predict_model.pkl'
pickle.dump(Model_RandomForest, open(filename, 'wb'))
```

In [16]:
```
pwd
```

Out[16]: 'C:\\Users\\ELCOT\\Desktop\\sprint2'

In [ ]:

7.2 Hyper Parameter Optimization:

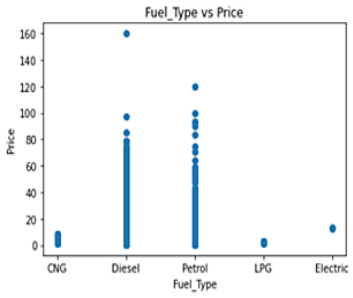
Hyper parameter optimization, also called hyper parameter tuning, is the process of searching for a set of hyper parameters that gives the best model results on a given dataset.

In [30]:
```
#year vs price
plt.title("Year vs Price")
plt.xlabel("Year")
plt.ylabel("Price")
plt.scatter(df1_map.Year,df1_map.Price)
```

Out[30]:

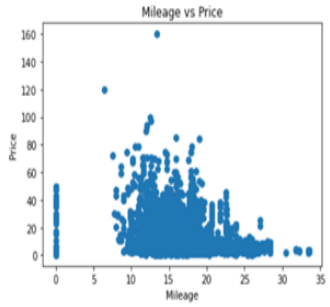
In [31]:
```
#fuel type vs price
plt.title("Fuel_Type vs Price")
plt.xlabel("Fuel_Type")
plt.ylabel("Price")
plt.scatter(df1_map.Fuel_Type,df1_map.Price)
```

Out[31]:



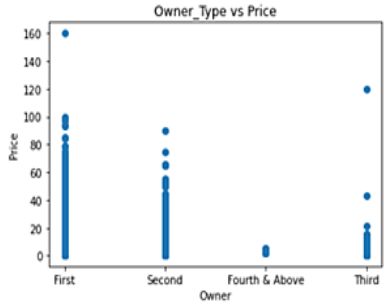
In [35]:
```
#mileage vs price
plt.title("Mileage vs Price")
plt.xlabel("Mileage")
plt.ylabel("Price")
plt.scatter(df1_map.Mileage,df1_map.Price)
```

Out[35]:



In [33]:
```
#owner type vs price
plt.title("Owner_Type vs Price")
plt.xlabel("Owner")
plt.ylabel("Price")
plt.scatter(df1_map.Owner_Type,df1_map.Price)
```

Out[33]:



Selection of Best Model :

Best Model - Random Forest

In [14]:
```
Model_RandomForest = RandomForestRegressor(max_features='sqrt', bootstrap=True)
Model_RandomForest.fit(X_train, y_train)
Model_RandomForest.score(X_test, y_test)
```

Out[14]: 0.8560922241801848

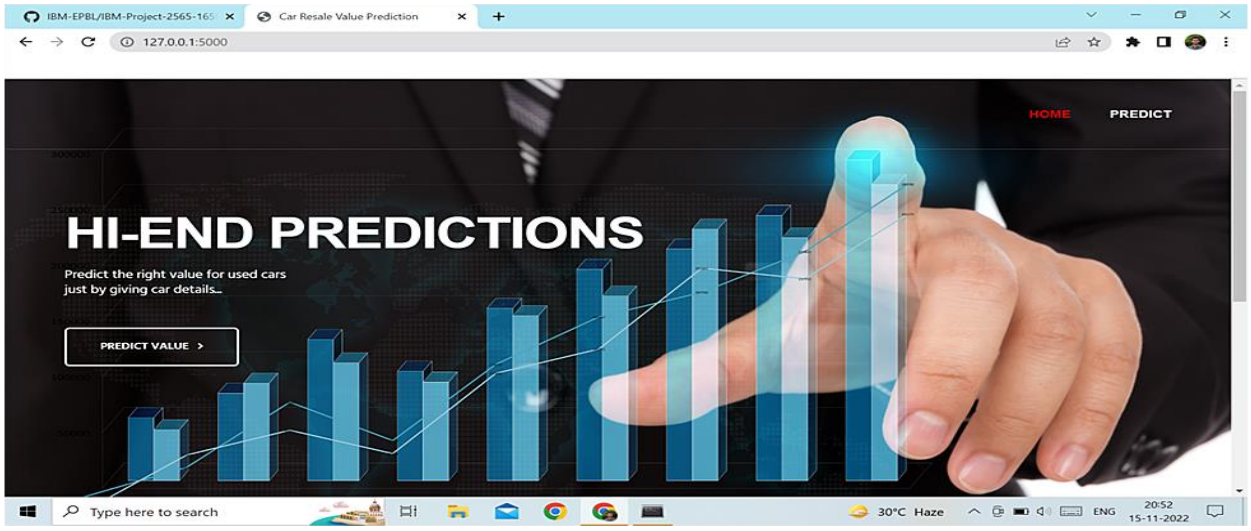
In [15]:
```
import pickle
filename = 'predict_model.pkl'
pickle.dump(Model_RandomForest, open(filename, 'wb'))
```

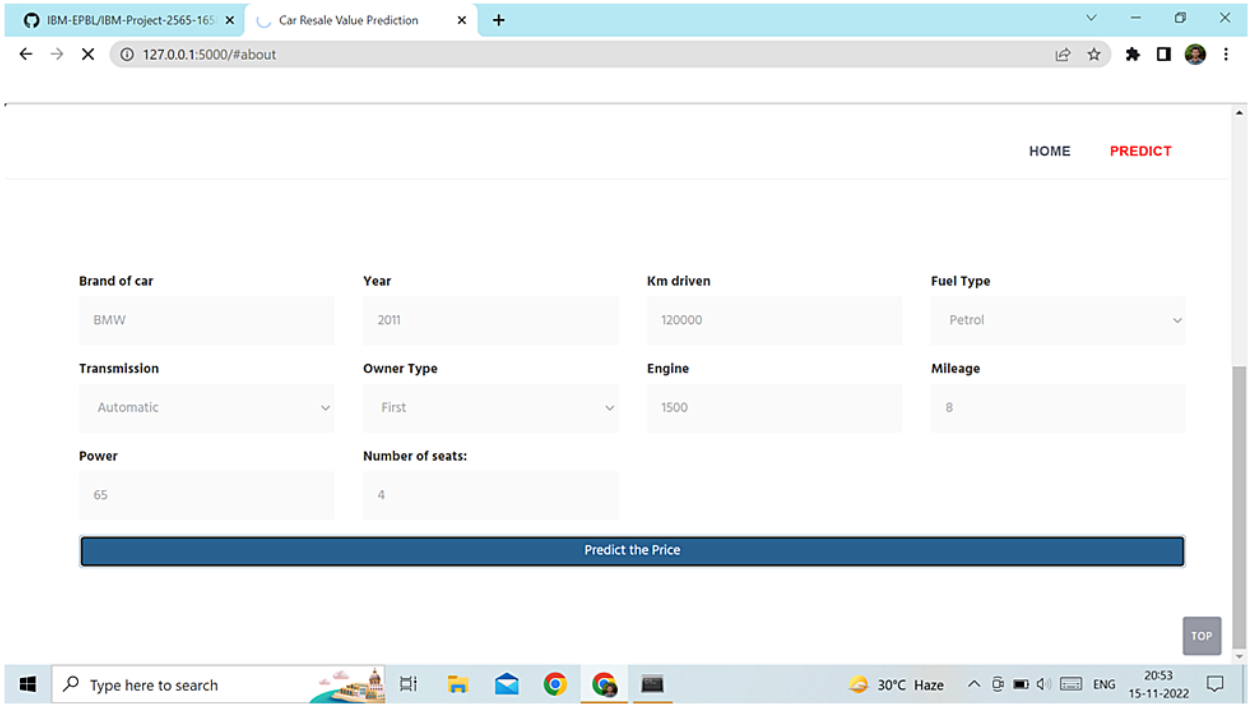
In [16]:
```
pwd
```

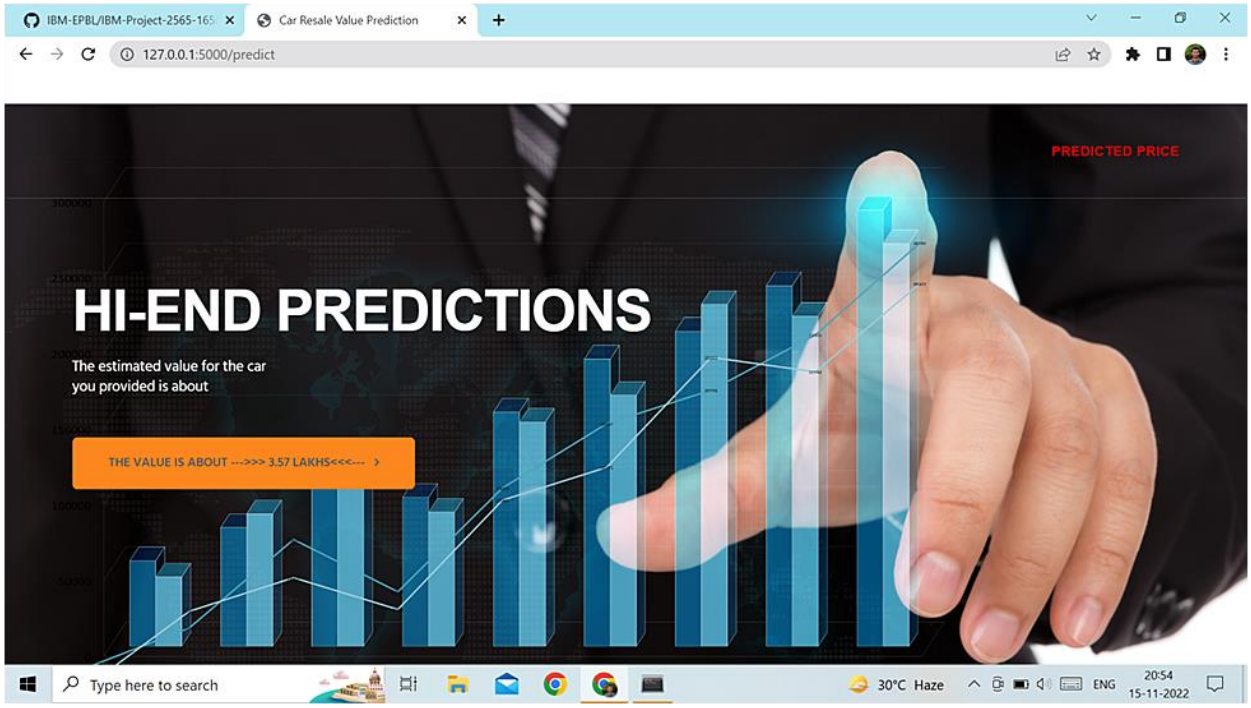
Out[16]: 'C:\\Users\\ELCOT\\Desktop\\sprint2'

In [ ]:

OUTPUTS :




IBM-EPBL/IBM-Project-2565-165
Car Resale Value Prediction
127.0.0.1:5000/#about
HOME
PREDICT
Brand of car
BMW
Year
2011
Km driven
120000
Fuel Type
Petrol
Transmission
Automatic
Owner Type
First
Engine
1500
Mileage
8
Power
65
Number of seats:
4
Predict the Price
TOP
Type here to search
30°C Haze
ENG
20:53
15-11-2022


IBM-EPBL/IBM-Project-2565-165
Car Resale Value Prediction
127.0.0.1:5000/predict
PREDICTED PRICE
HI-END PREDICTIONS
The estimated value for the car
you provided is about
THE VALUE IS ABOUT --->>> 3.57 LAKHS<<<---
Type here to search
30°C Haze
ENG
20:54
15-11-2022

CHAPTER – 8

TESTING

In the testing phase, we test the model using cross validation, we check the model is well or not and going is right or not, there are some technique of cross validation and we use confusion matrix for checking the model performance. We will test in all algorithms and we will get best suitable Seaborn is a python based data visualization library based on matplotlib.

8.1 Test Cases

				Date	03-Nov-22								
				Team ID	PNT2022TMID08828								
				Project Name	Project - Car Resales Value Predi								
				Maximum Marks	4 marks								
Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Commnets	TC for Automation(Y/N)	BUG ID	Executed By
HomePage_TC_OO 1	UI	Home Page	Verify whether Predict the Price button works		1.Enter URL and click go 2.Click the Predict Value button	127.0.0.1:5000/	It loads the Car resales Value prediction home page	Working as expected	Pass	Executed Perfectly	N		MADHAVAN MOTHISH LAKSHMAN
GetDetails_TC_OO 2	Functional	Prediction Page	Verify the UI elements for getting car details used for prediction	User Should know the details of the car to predict the value	1.Fill the details of the car	127.0.0.1:5000/home	Get the details of car such as: 1.Brand of car 2.year 3.KM driven 4.Fuel type 5.transmission 6.Owner Type 7.Engine 8.Power 9.Number of seats 10.Milleage	Working as expected	Pass		Y		RANJITHKUMAR ARJUN
ResultPage_TC_OO 3	Functional	Result page	Verify user is able to see the Predict the price button		1.Click on Predict the Value button and wait for the result	127.0.0.1:5000/home	User should able to see the predicted value in this page	Working as expected	Pass		Y		MADHAVAN RANJITHKUMAR

8.2 User Acceptance Testing

User Acceptance Testing (UAT), or application testing, is the final stage of any software development or change request lifecycle before go-live. It is the final stage of any development process to determine that the software does what it was designed to do in real-world situations. Actual users test the software to determine if it does what it was designed to do in real-world situations, validating changes made and assessing adherence to their organization's business requirements. The main purpose of acceptance testing is to validate end-to-end business flow.

1. **Purpose of Document**

The purpose of this document is to briefly explain the test coverage and open issues of the Car Resale value Prediction project at the time of the release to User Acceptance Testing (UAT).

2. **Defect Analysis**

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

Resolution	Severity1	Severity2	Severity3	Severity4	Subtotal
By Design	5	3	1	0	9
Duplicate	1	0	2	0	3
External	2	2	0	1	5
Fixed	6	4	2	1	13
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	3	2	1	6
Totals	14	12	9	4	39

3. **Test Case Analysis**

This reports how the number of test cases that have passed, failed and untested.

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	3	0	0	3
Client Application	3	0	0	3
Security	2	1	0	1
Outsource Shipping	3	1	0	2
Exception Reporting	1	0	0	1
Final Report Output	4	0	0	4
Version Control	2	0	0	2

CHAPTER – 9

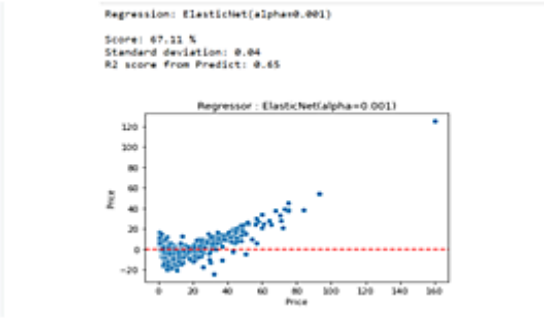
RESULTS

9.1 Performance Metrics

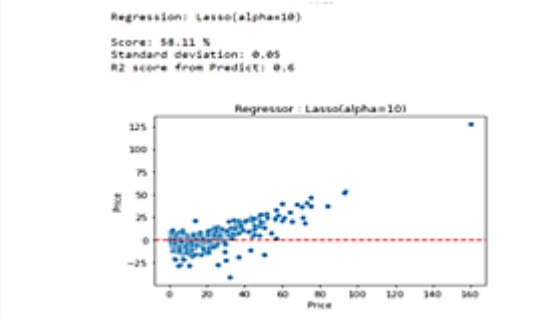
Model Performance Testing:

1. Metrics : Regression Model

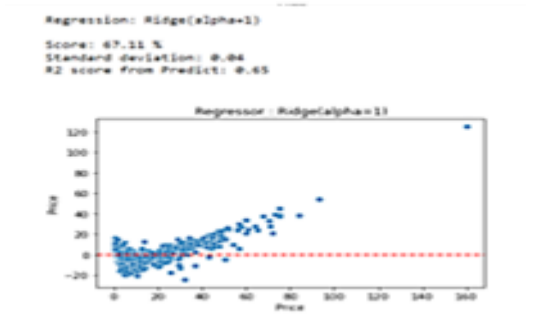
i. Elastic Net – 67.11%



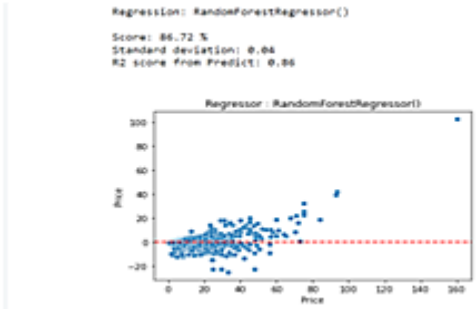
ii. Lasso – 58.11%



iii. Ridge – 67.11%

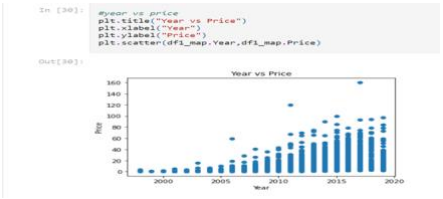


iv. RandomForest -86.72%

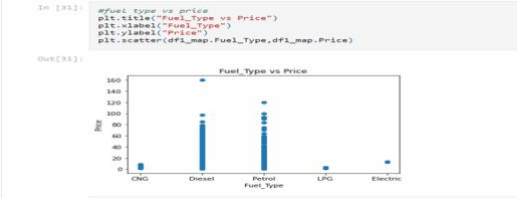


2. Tune the model - Hyperparameter Tuning Validation Method

i. Year vs Price



ii. Fuel type vs Price



iii. Transmission vs Price

iv. Owner type vs Price

CHAPTER – 10

ADVANTAGES & DISADVANTAGES

Advantages

- It saves user time that is wasted in search of required second hand vehicles.
- User can find the various vehicles information along with images at a single place.
- This system is effective and saves time, efforts and cost of users.
- Sellers can also easily find a buyer for their vehicle easily by predicting the price.
- Easy registration.

Disadvantages

- The user cannot view the vehicle in person.
- No human interaction.
- If there are poor quality photographs then it's of no use

CHAPTER – 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 Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction.

CHAPTER – 12

FUTURE SCOPE

In future this machine learning model may bind with various website which can provide real time data for price prediction. Also we may add large historical data of car price which can help to improve accuracy of the machine learning model. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.

CHAPTER – 13

Source Code

Index.html

```
<!DOCTYPE html>

<html lang="en" class="no-js">

<head>

<meta charset="utf-8"/>

<title>Car Resale Value Prediction</title>

<link href="http://fonts.googleapis.com/css?family=Hind:300,400,500,600,700" rel="stylesheet" type="text/css">

<link href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css" rel="stylesheet">

<link href="vendor/simple-line-icons/simple-line-icons.min.css" rel="stylesheet" type="text/css"/>

<link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>

<link href="css/animate.css" rel="stylesheet">

<link href="vendor/swiper/css/swiper.min.css" rel="stylesheet" type="text/css"/>

<link href="css/layout.min.css" rel="stylesheet" type="text/css"/>

<link rel="shortcut icon" href="favicon.ico"/>

</head>

<body id="body" data-spy="scroll" data-target=".header">

<header class="header navbar-fixed-top">

<nav class="navbar" role="navigation">

<div class="container">

<div class="menu-container js_nav-item">

</div>

<div class="collapse navbar-collapse nav-collapse">

<div class="menu-container">

<ul class="nav navbar-nav navbar-nav-right">
```

```
<li class="js_nav-item nav-item"><a class="nav-item-child nav-item-hover"
href="#body">Home</a></li>

<li class="js_nav-item nav-item"><a class="nav-item-child nav-item-hover"
href="#about">Predict</a></li>

</ul>

</div>

</div>

</div>

</nav>

</header>

<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

<div class="container">

<ol class="carousel-indicators">

<li data-target="#carousel-example-generic" data-slide-to="0" class="active"></li>

<li data-target="#carousel-example-generic" data-slide-to="1"></li>

</ol>

</div>

<div class="carousel-inner" role="listbox">

<div class="item active">

<img class="img-responsive" src="img/1920x1080/01.jpg" alt="Slider Image">

<div class="container">

<div class="carousel-centered">

<div class="margin-b-40">

<h1 class="carousel-title">Hi-End Predictions</h1>

<p class="color-white">Predict the right value for used cars <br/>just by giving car details... </p>

</div>

<a href="#about" class="btn-theme btn-theme-sm btn-white-brd text-uppercase">Predict Value</a>

</div>
```

```
</div>
</div>
<div id="about">
<div class="content-lg container">
<section class="featured-places">
<div class="container">
<form  method="POST" action ="/predict">
<div class="row">
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Brand of car</label>
<input type="text" class="form-control" id="" name=""
placeholder="Enter the brand " required>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Year</label>
<input type="text" class="form-control" id="input2" name="input2"
placeholder="Enter the year"required>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Km driven</label>
<input type="text" class="form-control" id="input3" name="input3"
placeholder="Enter the km" required>
```

```
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Fuel Type</label>
<select class="form-control" id="input4" name="input4" required>
<option value="3">Petrol</option>
<option value="2">Disel</option>
<option value="1">CNG</option>
</select>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Transmission</label>
<select class="form-control" id="input5" name="input5" required>
<option value="0">Manual</option>
<option value="1">Automatic</option>
</select>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Owner Type</label>
<select class="form-control" id="input6" name="input6" required>
<option value="0"> First</option>
<option value="1"> Second</option>
```

```
<option value="2"> Third or more</option>
</select>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Engine </label>
<input type="text" class="form-control" id="input7" name="input7"
placeholder="Enter the cc" required>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Mileage</label>
<input type="text" class="form-control" id="input8" name="input8"
placeholder="Enter the mileage" required>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
<label>Power</label>
<input type="text" class="form-control" id="input9" name="input9"
placeholder="Enter the bhp" required>
</div>
</div>
<div class="col-lg-3 col-md-4 col-sm-6 col-xs-12">
<div class="form-group">
```

<label>Number of seats:</label>

<input type="text" class="form-control" id="input10" name="input10"

placeholder="Enter no. of seates" required>

<button type="submit" class="section-btn btn btn-primary btn-block">Predict the Price</button>

</form>

</section>

Top

<script src="vendor/jquery.min.js" type="text/javascript"></script>

<script src="vendor/jquery-migrate.min.js" type="text/javascript"></script>

<script src="vendor/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>

<script src="vendor/jquery.easing.js" type="text/javascript"></script>

<script src="vendor/jquery.back-to-top.js" type="text/javascript"></script>

<script src="vendor/jquery.smooth-scroll.js" type="text/javascript"></script>

<script src="vendor/jquery.wow.min.js" type="text/javascript"></script>

<script src="vendor/swiper/js/swiper.jquery.min.js" type="text/javascript"></script>

<script src="vendor/masonry/jquery.masonry.pkgd.min.js" type="text/javascript"></script>

<script src="vendor/masonry/imagesloaded.pkgd.min.js" type="text/javascript"></script>

<script src="js/layout.min.js" type="text/javascript"></script>

<script src="js/components/wow.min.js" type="text/javascript"></script>

<script src="js/components/swiper.min.js" type="text/javascript"></script>

<script src="js/components/masonry.min.js" type="text/javascript"></script>

</html>

Predict

<!DOCTYPE html>

<html lang="en" class="no-js">

<head>

<meta charset="utf-8"/>

<title>Car Resale Value Prediction</title>

<link href="http://fonts.googleapis.com/css?family=Hind:300,400,500,600,700" rel="stylesheet" type="text/css">

<link href="//netdna.bootstrapcdn.com/font-awesome/4.0.3/css/font-awesome.css" rel="stylesheet">

<link href="vendor/simple-line-icons/simple-line-icons.min.css" rel="stylesheet" type="text/css"/>

<link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet" type="text/css"/>

<link href="css/animate.css" rel="stylesheet">

<link href="vendor/swiper/css/swiper.min.css" rel="stylesheet" type="text/css"/>

<link href="css/layout.min.css" rel="stylesheet" type="text/css"/>

<link rel="shortcut icon" href="favicon.ico"/>

</head>

<body id="body" data-spy="scroll" data-target=".header">

<header class="header navbar-fixed-top">

<nav class="navbar" role="navigation">

```
<ul class="nav navbar-nav navbar-nav-right">

<li class="js_nav-item nav-item"><a class="nav-item-child nav-item-hover" href="#body">Predicted price</a></li>

</ul>

</div>

</div>

</div>

</nav>

</header>

<div id="carousel-example-generic" class="carousel slide" data-ride="carousel">

<div class="carousel-inner" role="listbox">

<div class="item active">

<img class="img-responsive" src="img/1920x1080/01.jpg" alt="Slider Image">

<div class="container">

<div class="carousel-centered">

<div class="margin-b-40">

<h1 class="carousel-title">Hi-End Predictions</h1>

<p class="color-white">The estimated value for the car <br/>you provided is about</p>

</div>

<a  class="btn-theme btn-theme-sm btn-white-brd text-uppercase">The Value is about --->>> {{predict}}<<<---</a>

</div>

</div>

</div>

<a href="javascript:void(0);" class="js-back-to-top back-to-top">Top</a>

<script src="vendor/jquery.min.js" type="text/javascript"></script>

<script src="vendor/jquery-migrate.min.js" type="text/javascript"></script>

<script src="vendor/bootstrap/js/bootstrap.min.js" type="text/javascript"></script>
```

```
<script src="vendor/jquery.easing.js" type="text/javascript"></script>
<script src="vendor/jquery.back-to-top.js" type="text/javascript"></script>
<script src="vendor/jquery.smooth-scroll.js" type="text/javascript"></script>
<script src="vendor/jquery.wow.min.js" type="text/javascript"></script>
<script src="vendor/swiper/js/swiper.jquery.min.js" type="text/javascript"></script>
<script src="vendor/masonry/jquery.masonry.pkgd.min.js" type="text/javascript"></script>
<script src="vendor/masonry/imagesloaded.pkgd.min.js" type="text/javascript"></script>
<script src="js/layout.min.js" type="text/javascript"></script>
<script src="js/components/wow.min.js" type="text/javascript"></script>
<script src="js/components/swiper.min.js" type="text/javascript"></script>
<script src="js/components/masonry.min.js" type="text/javascript"></script>
</html>
```

Local deployment : Python(app.py)

```
import flask
from flask import request, render_template
from flask_cors import CORS
import joblib
 app = flask.Flask(__name__, static_url_path='')
CORS(app)
 @app.route('/', methods=['GET'])
def sendHomePage():
    return render_template('index.html')
 @app.route('/predict', methods=['POST'])
def predictprice():
            input1 = int(12)
            input2 = int(request.form['input2'])
```

```
        input3 = int(request.form['input3'])
        input4 = int(request.form['input4'])
        input5 = int(request.form['input5'])
        input6 = int(request.form['input6'])
        input7 = int(request.form['input7'])
        input8 = int(request.form['input8'])
        input9 = int(request.form['input9'])
        input10 = int(request.form['input10'])
        X = [[input1,input2,input3,input4,input5,input6,input7,input8,input9,input10]]
        model = joblib.load('predict_model.pkl')
        price = model.predict(X)[0]
        return render_template('predict.html',predict=("{:.2f} Lakhs".format(price)))
if __name__ == '__main__':
   app.run()
```

Cloud deployment : Python(app.py)

```
import requests
import flask
from flask import request, render_template
from flask_cors import CORS
import requests
API_KEY = "5BR0_y5fBo2IbTXwV9OPjpyXJddc-hVoizqrBZOo3Uxh"
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.Flask(__name__, static_url_path='')
```

```
CORS(app)

@app.route('/', methods=['GET'])

def sendHomePage():

    return render_template('index.html')

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

def predictprice():

    input1 = int(12)

    input2 = int(request.form['input2'])

    input3 = int(request.form['input3'])

    input4 = int(request.form['input4'])

    input5 = int(request.form['input5'])

    input6 = int(request.form['input6'])

    input7 = int(request.form['input7'])

    input8 = int(request.form['input8'])

    input9 = int(request.form['input9'])

    input10 = int(request.form['input10'])

    X = [[input1,input2,input3,input4,input5,input6,input7,input8,input9,input10]]

   payload_scoring = {"input_data": [{"field":
[['input1','input2','input3','input4','input5','input6','input7','input8','input9','input10']], "values": X}]}

    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/98b8a604-
b27a-4f5d-a7cc-ee8c7a680d49/predictions?version=2022-11-15', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})

    print(response_scoring)

    predictions = response_scoring.json()

    predict = predictions['predictions'][0]['values'][0][0]

    print("Final prediction :",predict)

        # showing the prediction results in a UI# showing the prediction results in a UI

    return render_template('predict.html',predict=("{:.2f} Lakhs".format(predict)))
```

```
if __name__ == '__main__' :
    app.run(debug= False)
```

Github Link : https://github.com/IBM-EPBL/IBM-Project-2565-1658474970

Project Demo Link : https://drive.google.com/file/d/1dw8SgUyX4lHGpr3pI2_IuVEzWY1Br3jD/view?usp=drivesdk