CAR RESALE VALUE PREDICTION USING RANDOM FOREST REGRESSION

1.INTRODUCTION

1.1Project Overview

Leading organizations are collecting tons of data every day to derive business decisions and solutions from it. With this huge amount of data,demand for data scientists and analysis is massively increasing.Machine learning and Artificial Intelligence are transforming the world for a better tomorrow.Data is the new “oil” of this 21st century,Machine Learning is the technology build over it.

Nowadays, Machine Learning and artificial intelligence are applicable in almost every sector.Companies are adopting smart AI adopting smart AI solutions in their product to eliminate manual inventions.Lets keep ourselves confined to cars,and we will see how it has changed the driving experiences.

1.2 PROJECT PURPOSE

The main idea of making a car resale value prediction system is to get hands-on practice for python using Data Science. 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.

2.LITERATURE SURVEY

CAR PRICE PREDICTION [Abhay Yadav, Chavi Ralhan ET AL, 2022]

India has a considerable size of car sales on top of the world day-to- day many buyers usually sell their cars after using for the time to another buyer, they name them as second possessor.Numerous platforms such as cars24.com, OLX.com that come up with these buyers with a platform where they can sell their old cars, but what should be the price of the car, this is the long-lasting query ever by using Machine Learning algorithms and they lead a response to this issue. Using a history of previous used car sales data and machine learning methodologies like Supervised Learning, they used to predict a fair price for the car.They also used machine learning techniques like Random Forest and Extra Tree Regression

USED CAR PRICE PREDICTION AND LIFE SPAN [Aditya Nikhade, Rohan Borde, 2021]

The predictions are based on dataset collected from various websites and Kaggle Websites mostly. This project will compare all this data to all regression algorithms and performance of various machine learning algorithms such as Linear Regression, Ridge Regression, Decision tree Regressor and choose the best out of it. Depending on various parameters the project will determine the price of a car and compare the prices of old cars with new cars. The lifespan of the car can be determined using Government regulations and Company claims.Apart from various factors, they also consider GPS navigator to predict the price of the car.

Car Price Prediction Using Machine Learning [Ketan Agrahari, Ayush Chaubey ET AL, 2021]

The rise of online websites and other tools like it have made it easier for both buyers and sellers to get a better understanding of the factors that determine the market value of a used car. Based on a set of factors, Machine Learning algorithms may be used to forecast the price of any automobile. The cost is calculated using the amount of characteristics. They used linear regression and lasso regression to develop a price model for used automobiles in a comparative research. The main goal of this study is to discover the best predictive model for estimating the price of a used car.

Used Car Price Prediction using K-Nearest Neighbor Based Model [Samruddhi, Ashok Kumar, 2020]

In this paper, they proposed a model to estimate the cost of the used cars using the K nearest neighbour algorithm which is simple and suitable for small data set. Here, they have collected a used cars dataset and analyzed the same. The data was trained by the model and examined the accuracy of the model among different ratios of trained and test set. The same model is cross-validated for assessing the performance of the model using the K- Fold method which is easy to understand and implement. They have used the K nearest Neighbor algorithm and got accuracy 85% where the accuracy of linear regression is 71%. The proposed model is also validated with 5 and 10 folds by using K FoldMethod. The experimental analysis shows that the proposed model is fitted as the optimized model.

2.1.EXISTING PROBLEM

The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase (Pal, Arora and Palakurthy, 2018). There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offers this service, their prediction method may not be the best. Besides, different models and systems may contribute on predicting power for a used car’s actual market value. It is important to know their actual market value while both buying and selling.

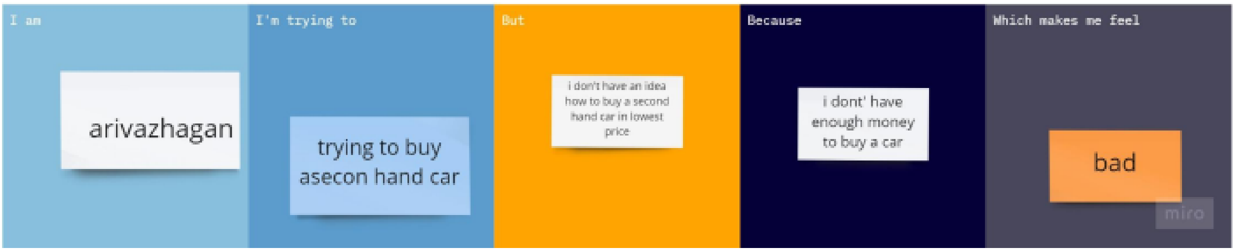
REFERENCES

1. https://www.kaggle.com/jpayne/852k-used-car-listings
2. N. Monburinon, P. Chertchom, T. Kaewkiriya, S. Rungpheung, S. Buya and P. Boonpou, "Prediction of prices for used car by using regression models," 2018 5th International Conference on Business and Industrial Research (ICBIR), Bangkok, 2018, pp. 115-119.
3. Listiani M. 2009. Support Vector Regression Analysis for Price Prediction in a Car Leasing Application. Master Thesis. Hamburg University of Technology
4. Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. ACM, 2016.
5. Ke, Guolin, et al. "Lightgbm: A highly efficient gradient boosting decision tree." Advances in Neural Information Processing Systems. 2017.
6. Fisher, Walter D. "On grouping for maximum homogeneity." Journal of the American statistical Association 53.284 (1958): 789-798.
7. https://scikit-learn.org/stable/modules/classes.html: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

2.3.PROBLEM STATEMENT DEFINITION:

Create a problem statement to understand your customer's point of view. The CustomerProblem Statement template helps you focus on what matters to create experiences peoplewill love. A well-articulated customer problem statement allows you and your teamto findtheideal solution for the challenges your customers face. Throughout the process, you’ll alsobeableto empathize with your customers, which helps you better understand howthey perceiveyour product or service.

REFERENCE LINK:https://miro.com/templates/customer-problem-statement/

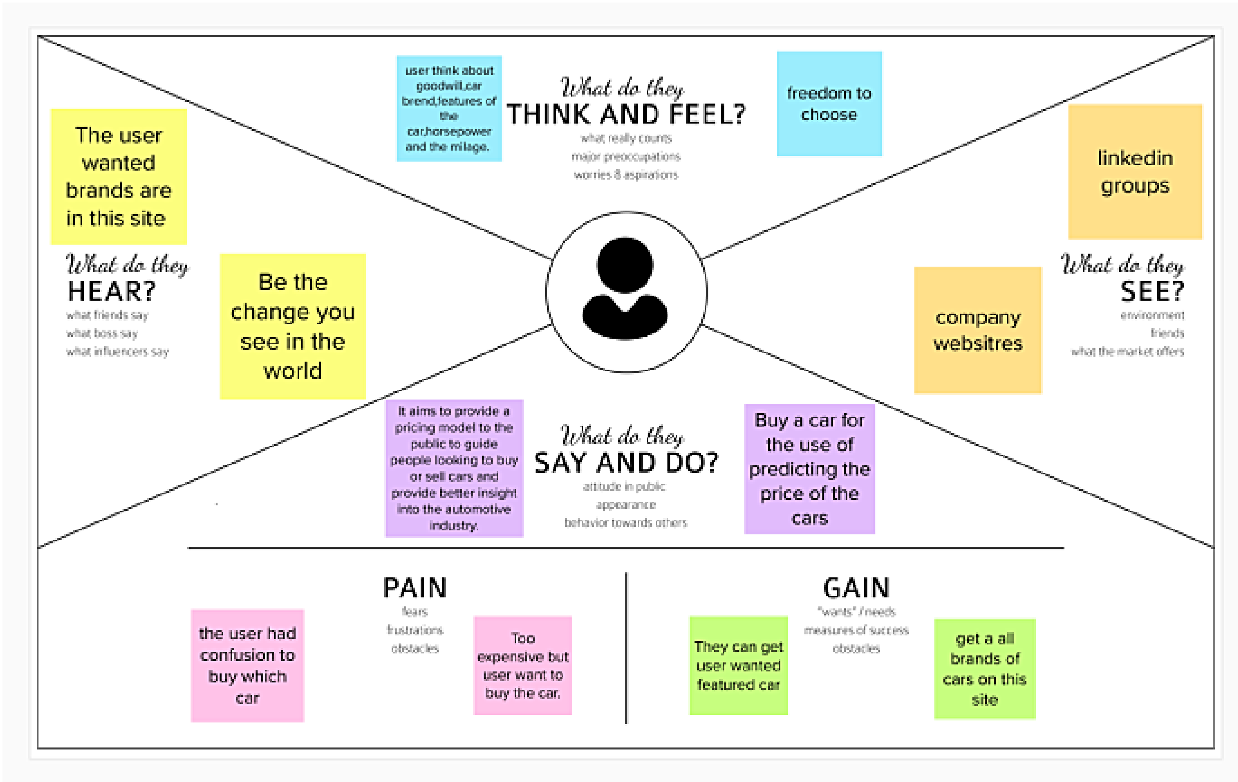


Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Buisness man	Buy a secound hand car for less amount of fuel consuming car	I am unaware of the used cars	I don't have any guidence	frustated
PS-2	Common man	Selling my old car for bought a new car	I amun aware of selling car in online	Idon't have any customer service in online	confused

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.
- It is a useful tool to helps teams better understand their users.
- Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.
- An empathy map is a **visual map** where the team member has to identify with different feelings, emotions, needs, goals, etc. of the target audience they are designing for. It helps the team to empathize with their users and understand their problems in order to design better products or services for them.



3.2 .IDEATION &BRAINSTROMING

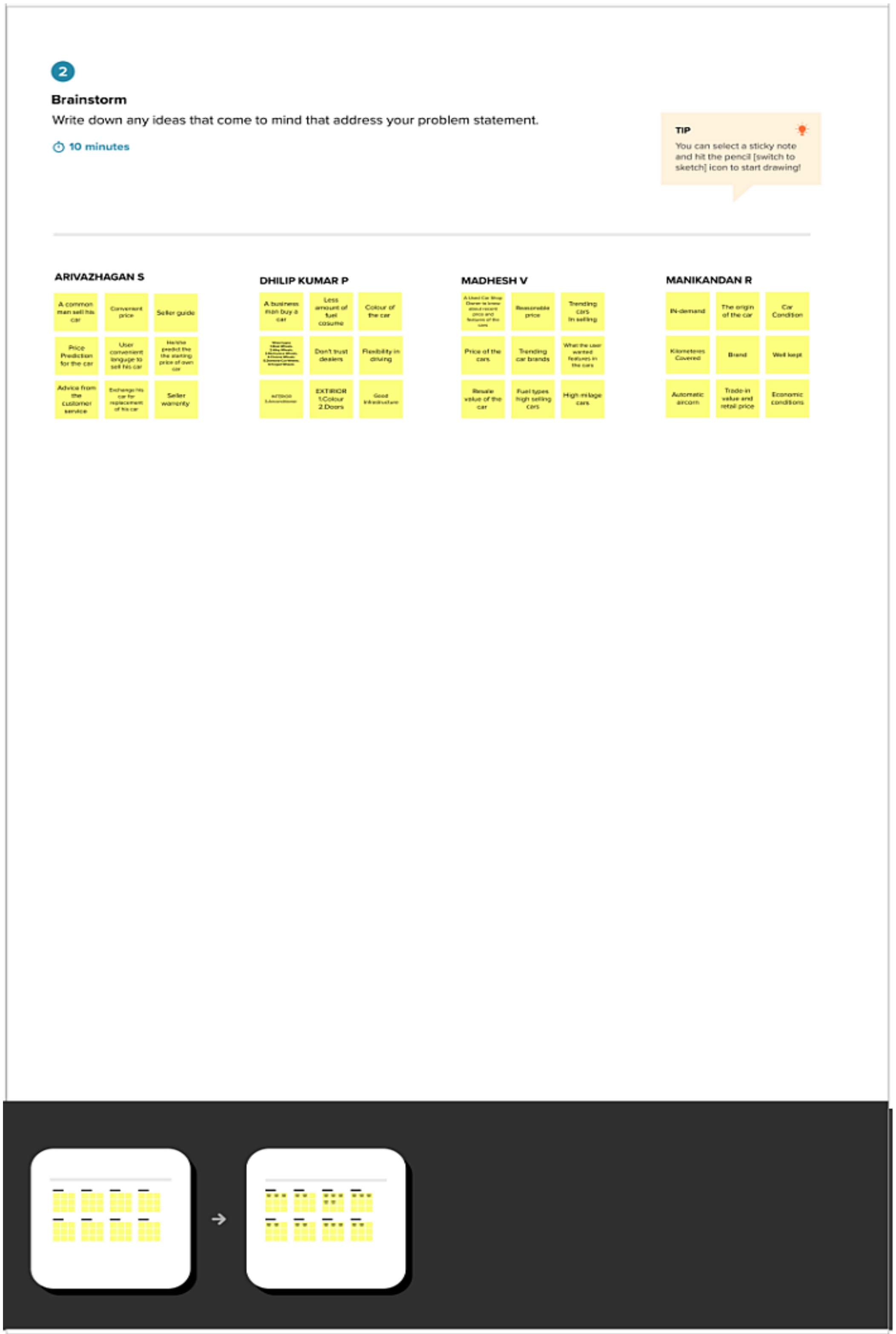
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate,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

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

Reference:https://www.mural.co/templates/empathy-map-canvas

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

Step-2: Brainstorm, Idea Listing and Grouping



3

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

BASED ON PERFORMANCE

- Wants to be futuristic
- Fuel consumption
- Automatic aircon
- Good mileage

BASED ON PRICE

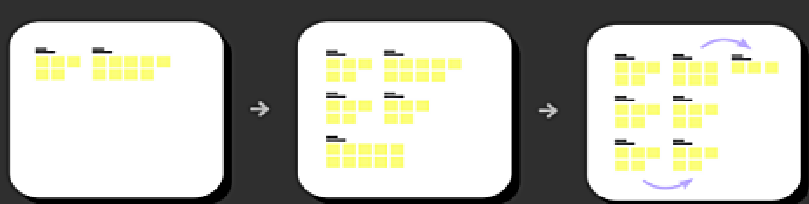
- Explores sites and looks at all cars in budget
- Research more specifically about price

BASED ON SAFETY

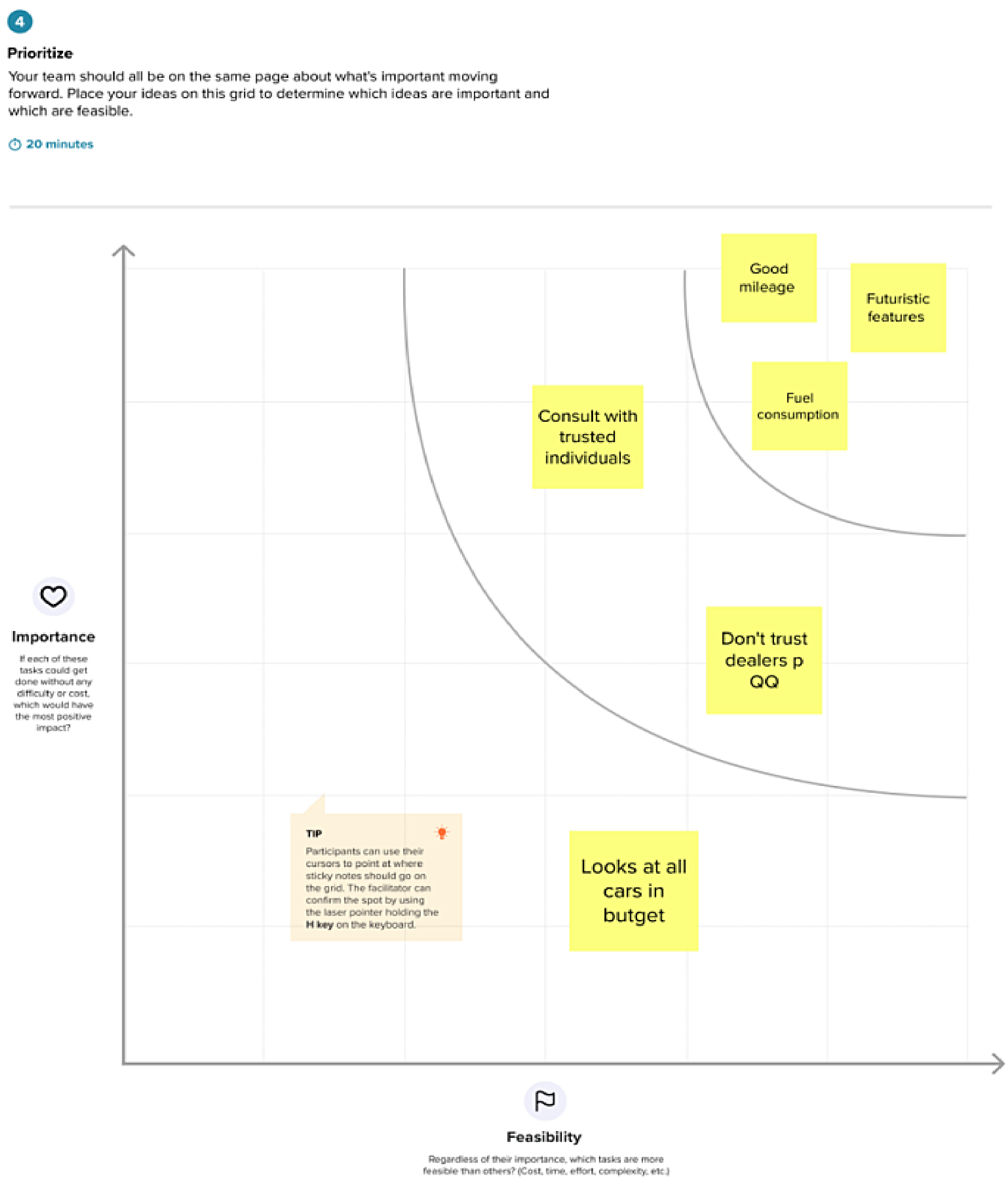
- Buys a car that meets our needs
- Don't trust Dealers
- Flexibility in driving
- Consult with trusted individuals

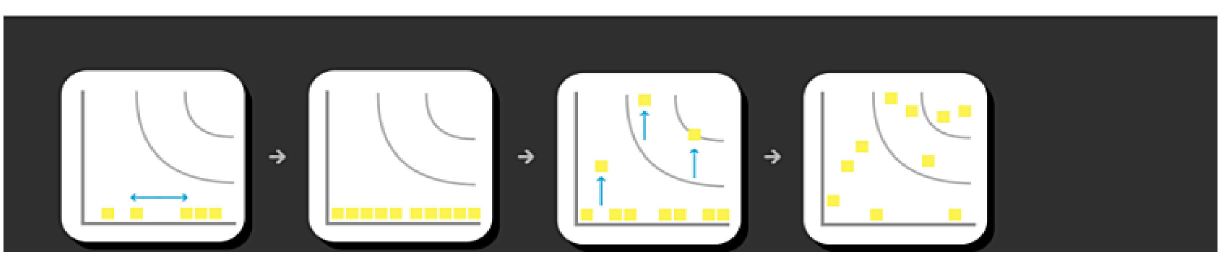
TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.



Step-3:idea prioritization





3.3.Proposed Solution

Proposed Solution Template:

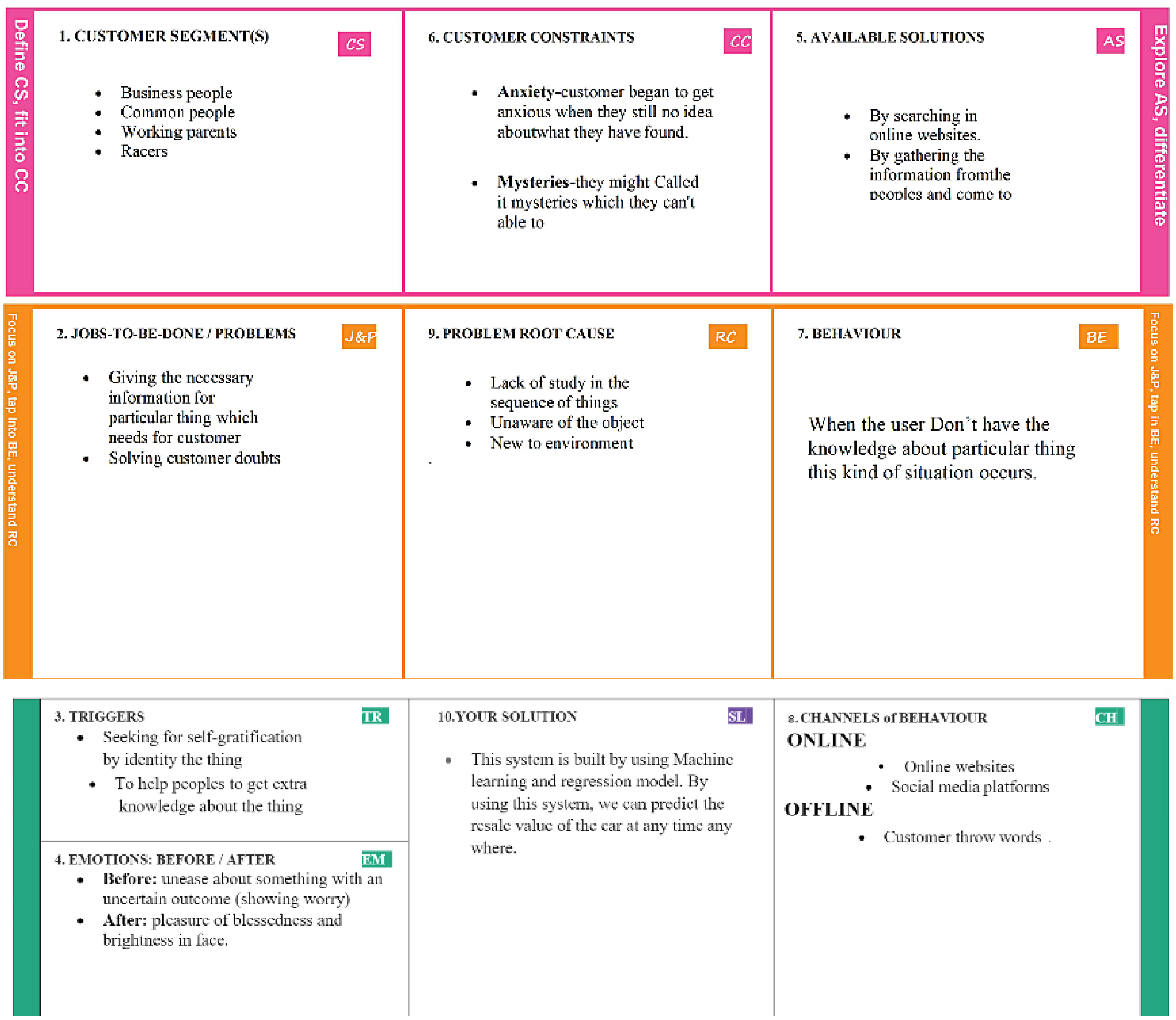
Project team shall fill the following information in proposed solution template

S.No.	**Parameter**	**Description**
1.	Problem Statement (Problem to besolved)	Currently, if anyone wants to sell their car, they have to take their car to a respective company workshop or have a to make an appointment for the company to get an estimate of the price. This process involves of lot of time and resources.
2.	Idea / Solution description	Especially for the first timers, a used purchase ismore practical an d affordable at the same time. Unless you really want the latest car in the market or that new car smell is all you are looking for, a used car can very well cater to almost all types of buyers quite conveniently
3.	Novelty / Uniqueness	Looks Matter for A Better Car Resale Value. A Service Ensures Good Car Resale Value. Keep All Papers in Place. Novelty is car resale Get Phone Numbers, Address, Photos, Maps of Novelty Tata.

4.	Social Impact / Customer Satisfaction	Became obsessed with customer feedback, Create a sense of convenience, Deliver fast responses, satisfaction is a company –wide focus. Customer Satisfaction Look and Style Fuel consumption Pulling Power Seating Capacity Riding Comfort Safety Features Speed Shock Absorbs & transmission Tyre mileage Braking Efficiency Tyre mileage Braking Efficiency
5.	Business Model (Revenue Model)	How to start a car merchant business. Generally, it is considered that if you want to start a car merchant business, you need a huge capital to invest. Dealer license. Location of the business. Keep a watch on the market. Make your catalog. Use a perfect marketing strategy.
6.	Scalability of the Solution	The size of the used car market in India was over 4.4 million units in 2020, according to Statista. The startup has managed to strive ahead by leveraging a robust managed marketplace business model, while proving that it is economically viable and independent of scale due to the use of technology, economy

			of scale, economy of scope, asset light, and network effects.

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 Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User Login	Login via Email Login via password
FR-4	Car registration	Registering the car details
FR-5	Value Prediction	Predicting the car resale value

4.2 NON FUNCTIONAL REQUIREMENTS

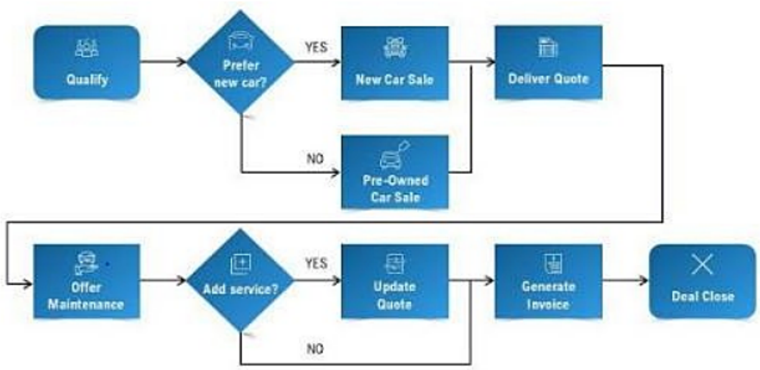
Following are the non-functional requirements of the proposed solution.

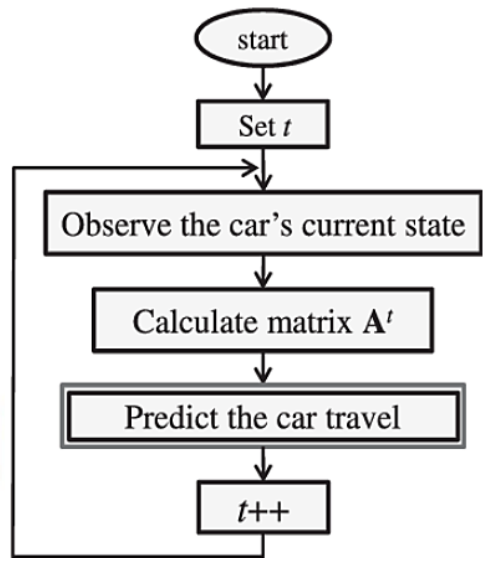
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Predicting the resale value
NFR-2	**Security**	Providing security to the website
NFR-3	**Reliability**	Providing high reliability by predicting values fordifferent types of cars
NFR-4	**Performance**	Providing high performance by using some machinelearning 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 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.





FLOW OF THE PROCESS

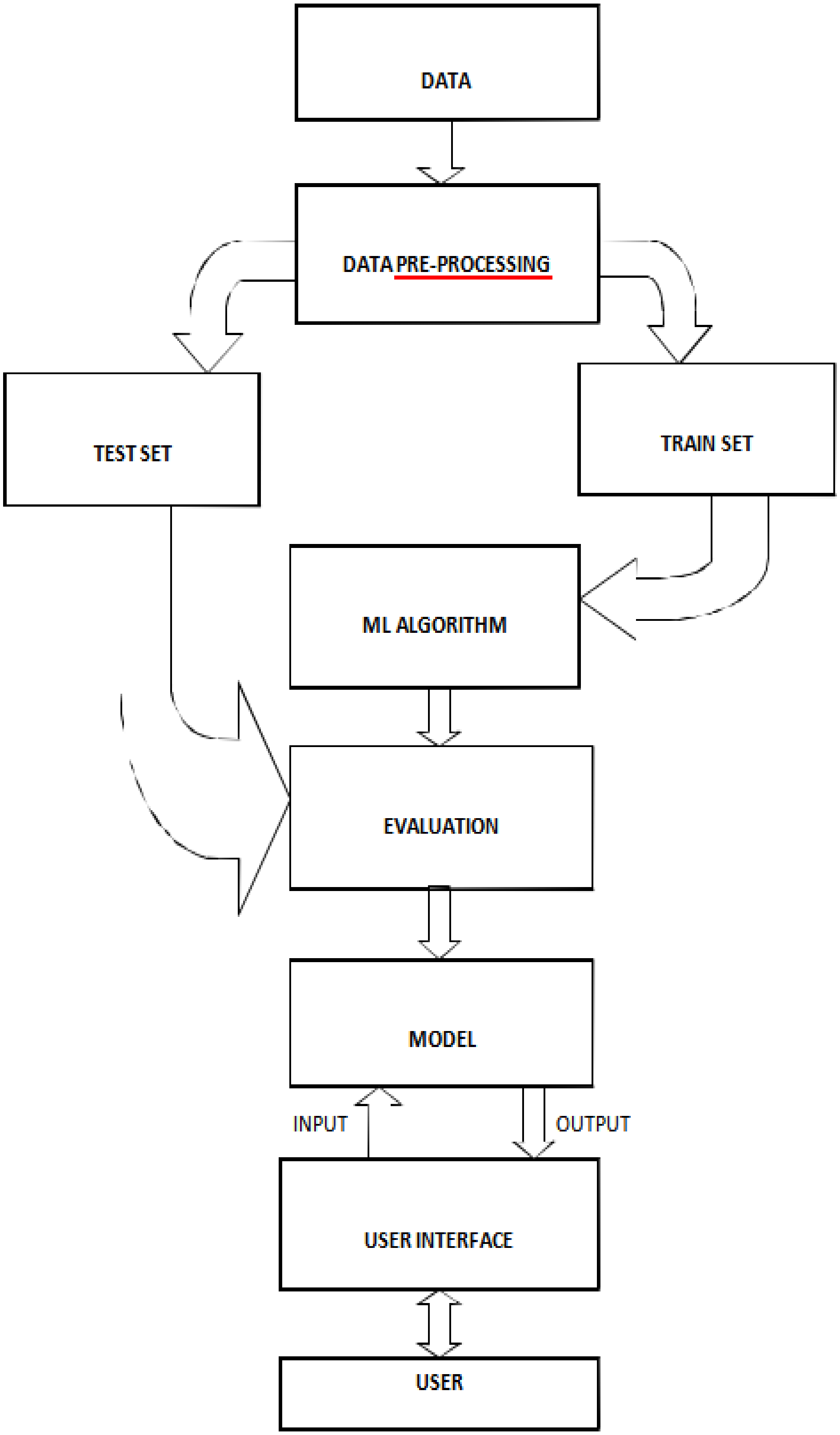
User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering	I can access my account / dashboard	High	Sprint-1

			my email, password, and confirming my password.			
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the application through G-mail	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can log into the application by entering email & password	High	Sprint-1

	Dashboard	USN-6	As a user, I can register & access the dashboa rdwith Facebook Login	I can access the dashboard thorugh facebook login and get access to various tools	Medi um	Sprin t-1
Customer (Web user)	Registration	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboa rd	High	Sprin t-1
Custom er Care Executi ve	Access	USN-7	As a user, I can connect to the custmer care executive through contact number or email.	I can connect to the custmer care executive and clarify mydoubts through contact number or email.	High	Sprin t-1

Administrator	Documents verification	USN-8	As a user, I can get my details and documents verified virtually from the comfort of my home.	, I can get my details and documents verified virtually from the comfort of my home.	High	Sprint-1
	Login verification	USN-9	As a user, I can get my login details verified virtually from the comfort of my home through OTP.	I can get my login details verified virtually from the comfort of my home through OTP.	High	Sprint-1

5.2 Solution & Technical Architecture



5.3 User Stories

User Type	Functional Requirement(Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail	I can register & access the application through G-mail	Medium	Sprint-1

	Login	USN-5	As a user, I can log into the application by entering email & password	I can log into the applicationby entering email & password	High	Sprint-1
	Dashboard	USN-6	As a user, I can register & access the dashboard with Facebook Login	I can access the dashboard thorugh facebook login and get access to various tools	Medium	Sprint-1
Customer (Webuser)	Registration	USN-6	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer Care Executive	Access	USN-7	As a user, I can connect to the custmer care executive through contact number or email.	I can connect to the custmer care executive and clarify my doubts through contact number or email.	High	Sprint-1
Administrator	Documents verification	USN-8	As a user, I can get my details and documents verified virtually from the comfort of my home.	, I can get my details and documents verified virtually from the comfort of my home.	High	Sprint-1

	Login verification	USN-9	As a user, I can get my login details verified virtually from the comfort of my home through OTP.	I can get my login details verified virtually from the comfort of my home through OTP.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 Sprint Planning & Estimation

Product Backlog, Sprint Schedule, and 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	Registration	USN-1	As a user, I can enter into the website with the help of the Google chrome browser in Windows	2	High	4
Sprint-1	Registration	USN-2	As a user, I can enter into the website through browser in Android	1	High	4
Sprint-1	Registration	USN-3	As a user, I can enter into the website through	2	Medium	4

			browser in ios			
Sprint-1	Login	USN-4	As a user, I can find the car resale value prediction page in the website	1	High	4
Sprint-2	Home Page	USN-5	As a user, I need to select the parameters like Year, Showroom price, Kilometres driven, fuel type etc and click on the submit button	2	High	4
Sprint-3	Home Page	USN-6	As a user, I can see the accurate price for car resale after entering the details.	2	High	4
Sprint-4	Home Page	USN-7	As a user, If I done a mistake while providing the details , I can reset the details and click the submit button.	1	Low	4

6.2 Sprint Delivery Schedule

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

6.3 Reports from JIRA

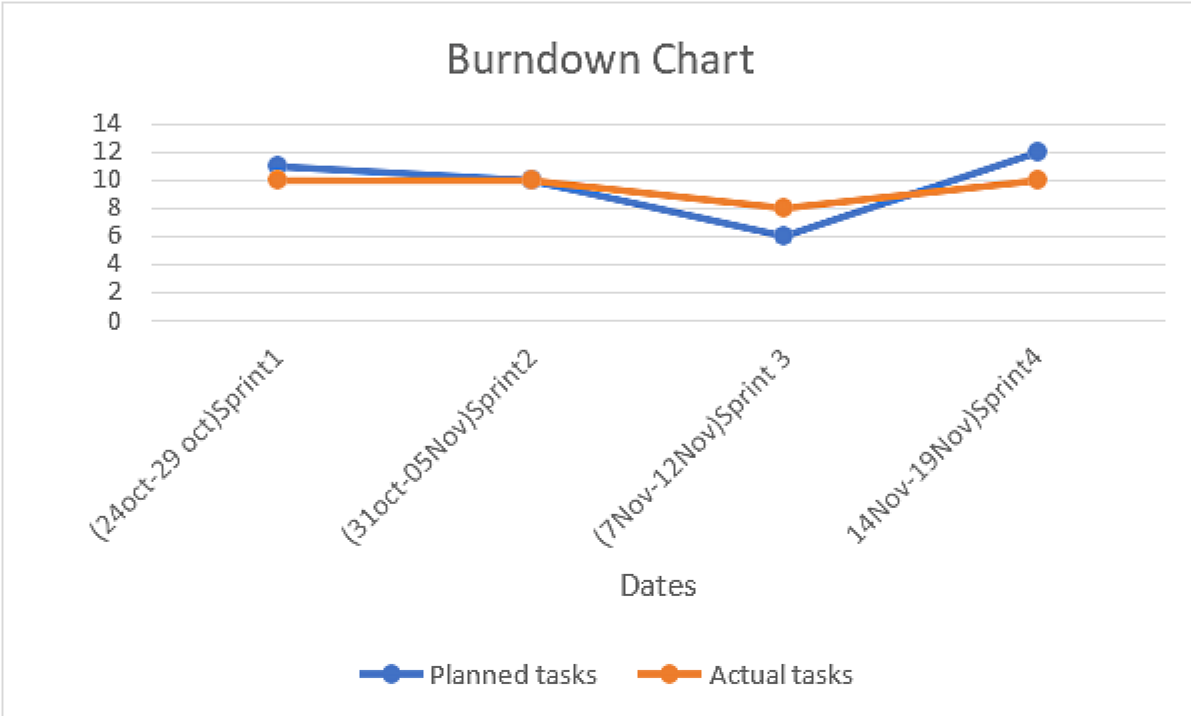
Velocity:

Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

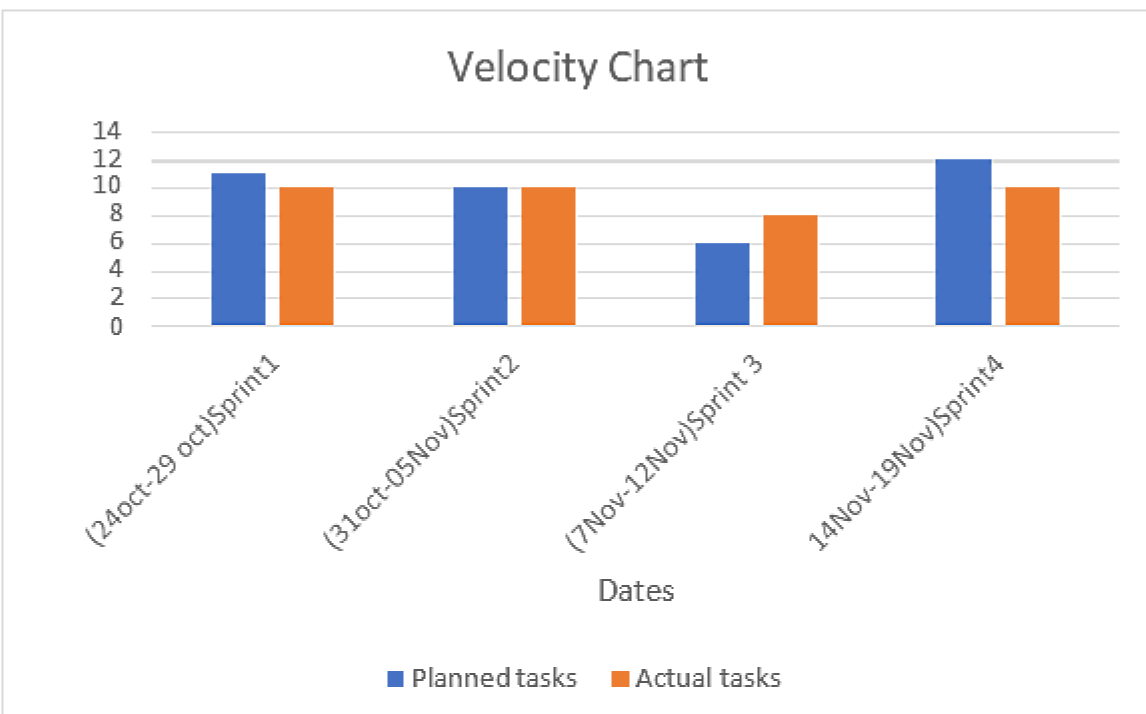
$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

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 suchas Scrum. However, burn down charts can be applied to any project containing measurable progress over time



velocity chart:



7. CODING & SOLUTIONING

7.1 Feature 1

In [1]:
```
import pandas as pd
```

In [2]:
```
df=pd.read_csv('car data.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

In [4]:
```
print(df['Seller_Type'].unique())
print(df['Fuel_Type'].unique())
print(df['Transmission'].unique())
print(df['Owner'].unique())
```

```
['Dealer' 'Individual']
['Petrol' 'Diesel' 'CNG']
['Manual' 'Automatic']
[0 1 3]
```

In [5]:
```
##check missing values
df.isnull().sum()
```

Out[5]:
```
Car_Name         0
Year             0
Selling_Price    0
Present_Price    0
Kms_Driven       0
Fuel_Type        0
Seller_Type      0
Transmission     0
Owner            0
dtype: int64
```

In [6]:
```
df.describe()
```

Out[6]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

In [7]:
```python
final_dataset=df[['Year','Selling_Price','Present_Price','Kms_Driven','Fuel_Type','Seller_Type','Transmission','Owner']]
```

In [8]:
```python
final_dataset.head()
```

Out[8]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

In [9]:
```python
final_dataset['Current Year']=2022
```

In [10]:
```python
final_dataset.head()
```

Out[10]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Current Year
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0	2022
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0	2022
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0	2022
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0	2022
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0	2022

In [11]:
```python
final_dataset['no_year']=final_dataset['Current Year']- final_dataset['Year']
```

In [12]:
```python
final_dataset.head()
```

Out[12]:

	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Current Year	no_year
0	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0	2022	8
1	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0	2022	9
2	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0	2022	5
3	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0	2022	11
4	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0	2022	8

In [13]:
```python
final_dataset.drop(['Year'],axis=1,inplace=True)
```

In [14]:
```
final_dataset.head()
```

Out[14]:

	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	Current Year	no_year
0	3.35	5.59	27000	Petrol	Dealer	Manual	0	2022	8
1	4.75	9.54	43000	Diesel	Dealer	Manual	0	2022	9
2	7.25	9.85	6900	Petrol	Dealer	Manual	0	2022	5
3	2.85	4.15	5200	Petrol	Dealer	Manual	0	2022	11
4	4.60	6.87	42450	Diesel	Dealer	Manual	0	2022	8

In [15]:
```
final_dataset=pd.get_dummies(final_dataset,drop_first=True)
```

In [16]:
```
final_dataset.head()
```

Out[16]:

	Selling_Price	Present_Price	Kms_Driven	Owner	Current Year	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	5.59	27000	0	2022	8	0	1	0	1
1	4.75	9.54	43000	0	2022	9	1	0	0	1
2	7.25	9.85	6900	0	2022	5	0	1	0	1
3	2.85	4.15	5200	0	2022	11	0	1	0	1
4	4.60	6.87	42450	0	2022	8	1	0	0	1

In [17]:
```
final_dataset.head()
```

Out[17]:

	Selling_Price	Present_Price	Kms_Driven	Owner	Current Year	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	5.59	27000	0	2022	8	0	1	0	1
1	4.75	9.54	43000	0	2022	9	1	0	0	1
2	7.25	9.85	6900	0	2022	5	0	1	0	1
3	2.85	4.15	5200	0	2022	11	0	1	0	1
4	4.60	6.87	42450	0	2022	8	1	0	0	1

In [18]:
```
final_dataset=final_dataset.drop(['Current Year'],axis=1)
```

In [19]:
```
final_dataset.head()
```

Out[19]:

	Selling_Price	Present_Price	Kms_Driven	Owner	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	5.59	27000	0	8	0	1	0	1
1	4.75	9.54	43000	0	9	1	0	0	1
2	7.25	9.85	6900	0	5	0	1	0	1
3	2.85	4.15	5200	0	11	0	1	0	1
4	4.60	6.87	42450	0	8	1	0	0	1

In [20]:
```
final_dataset.corr()
```

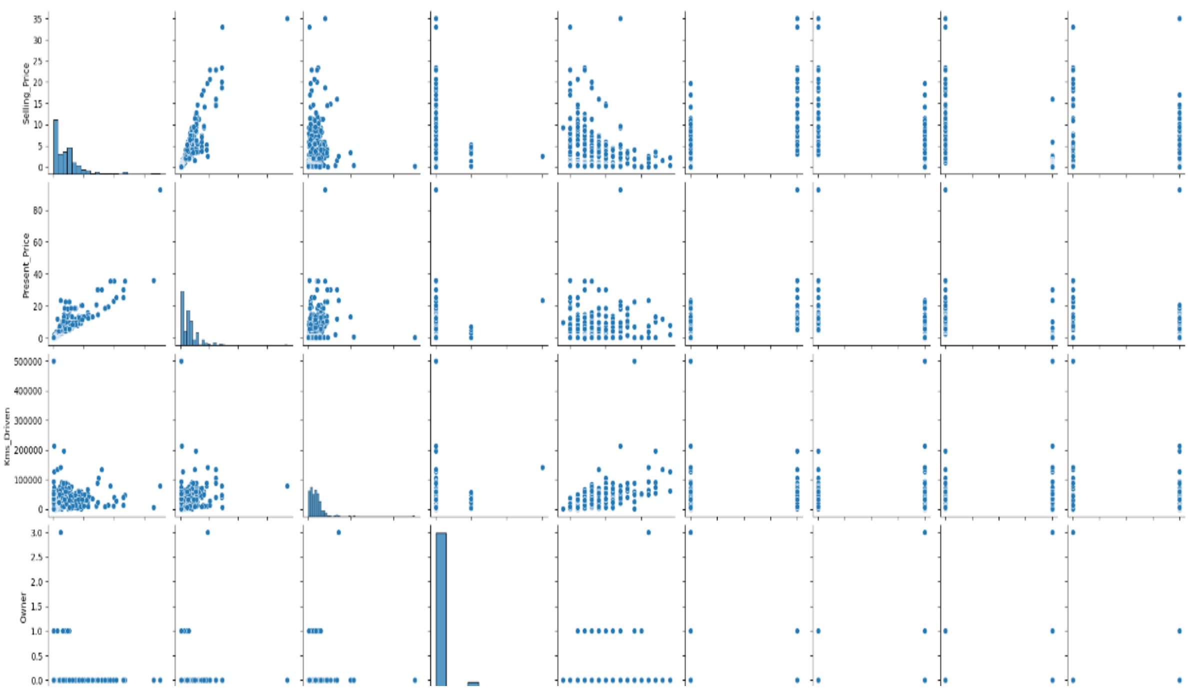
Out[20]:

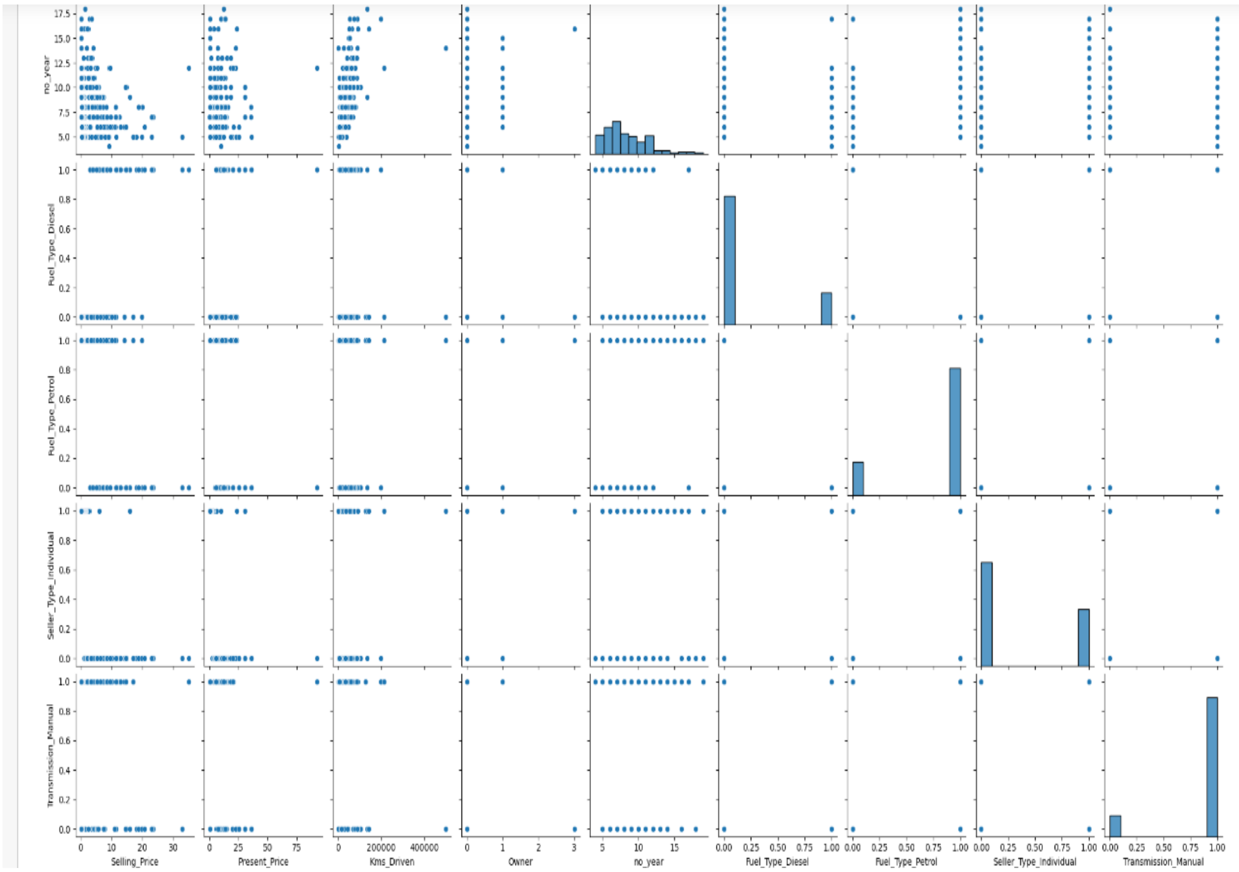
	Selling_Price	Present_Price	Kms_Driven	Owner	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Ma
Selling_Price	1.000000	0.878983	0.029187	-0.088344	-0.236141	0.552339	-0.540571	-0.550724	-0.36
Present_Price	0.878983	1.000000	0.203647	0.008057	0.047584	0.473306	-0.465244	-0.512030	-0.34
Kms_Driven	0.029187	0.203647	1.000000	0.089216	0.524342	0.172515	-0.172874	-0.101419	-0.16
Owner	-0.088344	0.008057	0.089216	1.000000	0.182104	-0.053469	0.055687	0.124269	-0.05
no_year	-0.236141	0.047584	0.524342	0.182104	1.000000	-0.064315	0.059959	0.039896	-0.00
Fuel_Type_Diesel	0.552339	0.473306	0.172515	-0.053469	-0.064315	1.000000	-0.979648	-0.350467	-0.09
Fuel_Type_Petrol	-0.540571	-0.465244	-0.172874	0.055687	0.059959	-0.979648	1.000000	0.358321	0.09
Seller_Type_Individual	-0.550724	-0.512030	-0.101419	0.124269	0.039896	-0.350467	0.358321	1.000000	0.06
Transmission_Manual	-0.367128	-0.348715	-0.162510	-0.050316	-0.000394	-0.098643	0.091013	0.063240	1.00

In [21]:
```
import seaborn as sns
```

In [22]:
```
sns.pairplot(final_dataset)
```

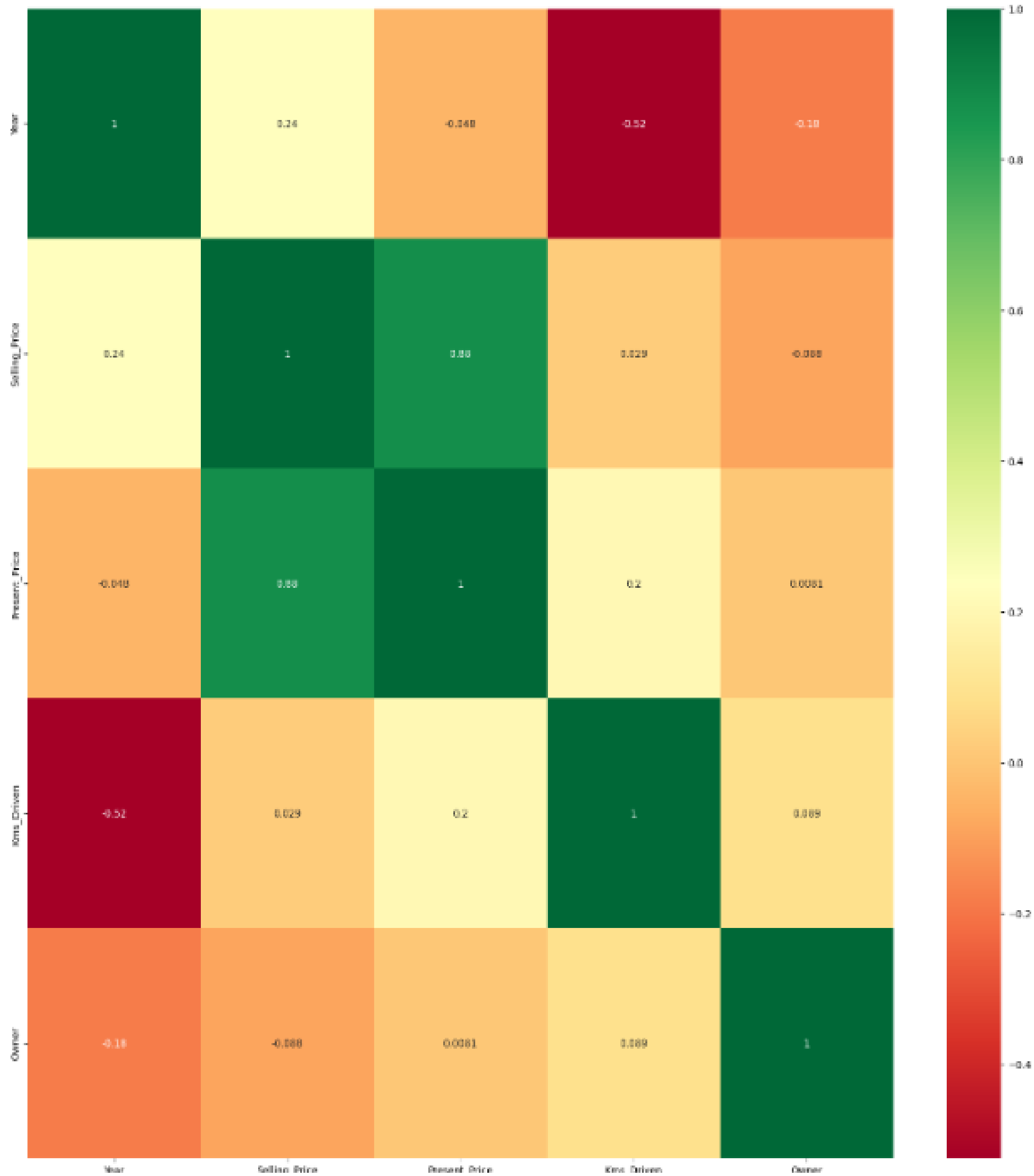
Out[22]: <seaborn.axisgrid.PairGrid at 0x1d5806dfb80>





In [26]:
```
import matplotlib.pyplot as plt
%matplotlib inline
```

In [27]:
```
#get correlations of each features in dataset
corrmat = df.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(20,20))
#plot heat map
g=sns.heatmap(df[top_corr_features].corr(),annot=True,cmap="RdYlGn")
```


Year
Selling_Price
Present_Price
Kms_Driven
Owner
1
0.24
-0.048
-0.52
-0.18
0.24
1
0.88
0.029
-0.088
-0.048
0.88
1
0.2
0.0081
-0.52
0.029
0.2
1
0.089
-0.18
-0.088
0.0081
0.089
1
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

In [28]:
```
X=final_dataset.iloc[:,1:]
y=final_dataset.iloc[:,0]
```

In [29]:
```
X['Owner'].unique()
```

Out[29]:
```
array([0, 1, 3], dtype=int64)
```

In [30]:
```
X.head()
```

Out[30]:

	Present_Price	Kms_Driven	Owner	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	5.59	27000	0	8	0	1	0	1
1	9.54	43000	0	9	1	0	0	1
2	9.85	6900	0	5	0	1	0	1
3	4.15	5200	0	11	0	1	0	1
4	6.87	42450	0	8	1	0	0	1

In [31]:
```
y.head()
```

Out[31]:
```
0    3.35
1    4.75
2    7.25
3    2.85
4    4.60
Name: Selling_Price, dtype: float64
```

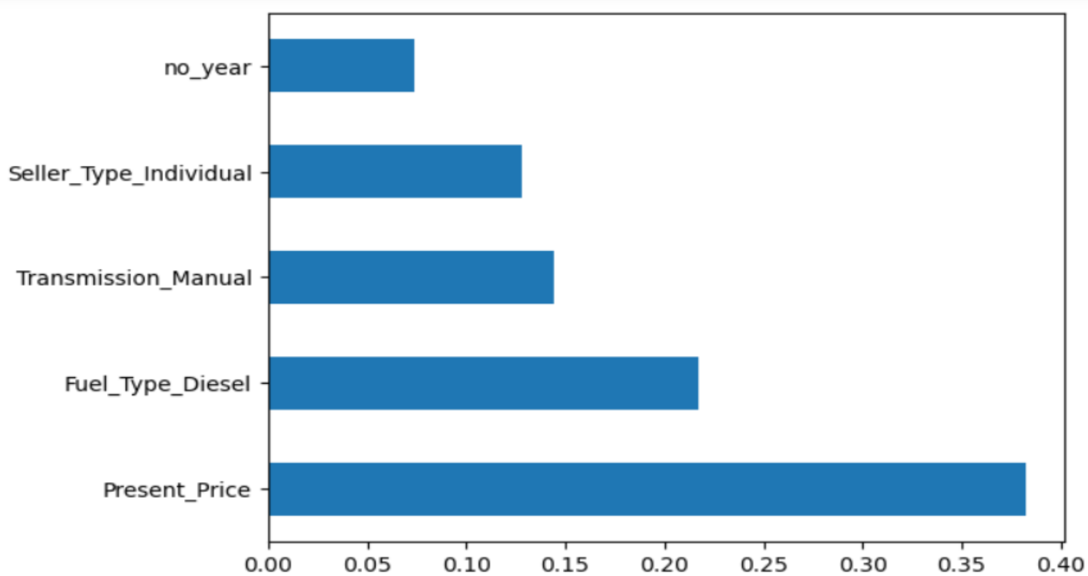
In [32]:
```
### Feature Importance
from sklearn.ensemble import ExtraTreesRegressor
import matplotlib.pyplot as plt
model = ExtraTreesRegressor()
model.fit(X,y)
```

Out[32]:
```
ExtraTreesRegressor()
```

In [33]:
```
print(model.feature_importances_)
```

```
[0.3825746  0.04212456 0.0009353  0.07392163 0.21694732 0.01140884
 0.1278428  0.14424496]
```

In [34]:
```
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(5).plot(kind='barh')
plt.show()
```



In [35]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.2)
```

In [39]:
```
X_train.shape
```

Out[39]:
```
(210, 8)
```

In [43]:
```
regressor=RandomForestRegressor()
```

In [45]:
```
###hyperparameters
import numpy as np
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
print(n_estimators)
```

```
[100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200]
```

In [46]:
```
from sklearn.model_selection import RandomizedSearchCV
```

In [47]:
```
 #Randomized Search CV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
# max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

In [48]:
```
# Create the random grid
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}

print(random_grid)
```

```
{'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200], 'max_features': ['auto', 'sqrt'], 'max_dept
h': [5, 10, 15, 20, 25, 30], 'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_leaf': [1, 2, 5, 10]}
```

In [49]:
```
# Use the random grid to search for best hyperparameters
# First create the base model to tune
rf = RandomForestRegressor()
```

In [50]:
```
# Random search of parameters, using 3 fold cross validation,
# search across 100 different combinations
rf_random = RandomizedSearchCV(estimator = rf, param_distributions = random_grid,scoring='neg_mean_squared_error', n_iter = 10, c
```

In [51]: rf_random.fit(X_train,y_train)

```
Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.7s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.7s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.7s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.9s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   1.8s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   2.3s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   2.0s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   2.1s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   2.0s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   2.1s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.6s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   0.8s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   0.8s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   0.8s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   0.8s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=5, n_estimators=400; total time=   0.9s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   1.4s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   1.6s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   1.3s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   1.4s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=10, min_samples_split=5, n_estimators=700; total time=   1.4s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=   2.1s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=   2.1s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=   2.1s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=   1.9s
[CV] END max_depth=25, max_features=sqrt, min_samples_leaf=1, min_samples_split=2, n_estimators=1000; total time=   2.0s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.0s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.0s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.1s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=10, min_samples_split=15, n_estimators=1100; total time=   2.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.6s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.5s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.2s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.3s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.2s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.3s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   1.3s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   1.4s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   1.4s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   1.5s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   1.4s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   1.3s
```

Out[51]:

```
RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=1,
                   param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                        'max_features': ['auto', 'sqrt'],
                                        'min_samples_leaf': [1, 2, 5, 10],
                                        'min_samples_split': [2, 5, 10, 15,
                                                              100],
                                        'n_estimators': [100, 200, 300, 400,
                                                         500, 600, 700, 800,
                                                         900, 1000, 1100,
                                                         1200]},
                   random_state=42, scoring='neg_mean_squared_error',
                   verbose=2)
```

In [52]:
```
rf_random.best_params_
```

Out[52]:
```
{'n_estimators': 1000,
 'min_samples_split': 2,
 'min_samples_leaf': 1,
 'max_features': 'sqrt',
 'max_depth': 25}
```

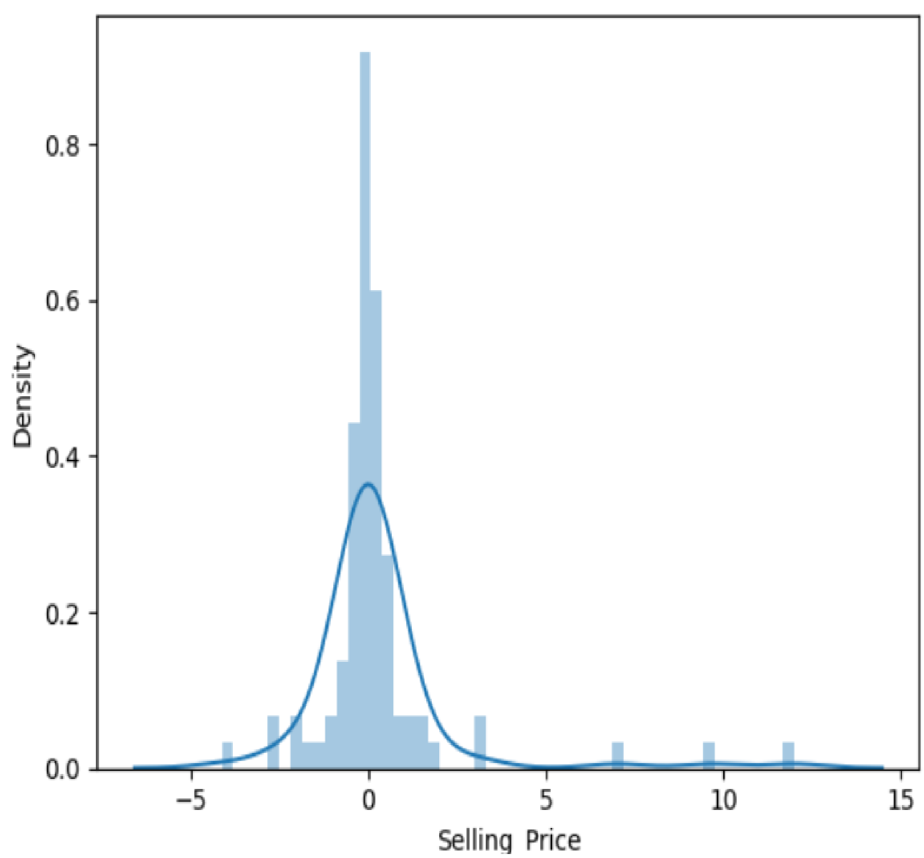
In [53]:
```
rf_random.best_score_
```

Out[53]:
```
-4.037144151006656
```

In [54]:
```
predictions=rf_random.predict(X_test)
```

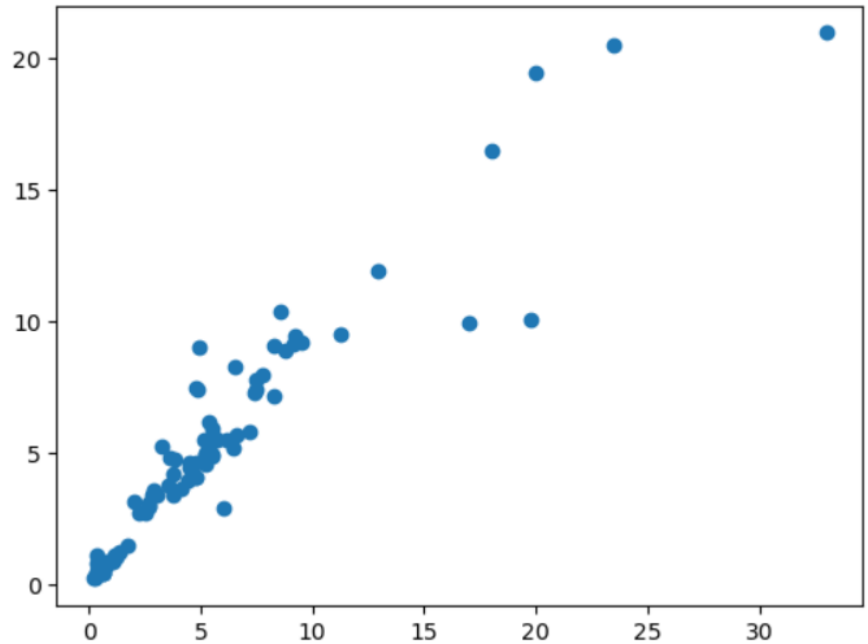
In [56]:
```
sns.distplot(y_test-predictions)
```

Out[56]:
```
<AxesSubplot:xlabel='Selling_Price', ylabel='Density'>
```



In [57]:
```
plt.scatter(y_test,predictions)
```

Out[57]: <matplotlib.collections.PathCollection at 0x1d589d05550>



In [58]:
```
from sklearn import metrics
```

In [59]:
```
print('MAE:', metrics.mean_absolute_error(y_test, predictions))
print('MSE:', metrics.mean_squared_error(y_test, predictions))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MAE: 0.8928115384615382
MSE: 4.090603201029677
RMSE: 2.0225239679740947
```

In [60]:
```
import pickle
# open a file, where you ant to store the data
file = open('random_forest_regression_model.pkl', 'wb')

# dump information to that file
pickle.dump(rf_random, file)
```

7.2 Feature 2

HTML HOME PAGE:

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
</head>
<body>
  <div style="color:rgb(0, 0, 0)">
      <form action="{{ url_for('predict')}}" method="post">
         <h2>Predictive analysis</h2>
         <h3> model Year</h3>
         <input id="first" name="Year" type="number"><br>

         <br><h3>Showroom Price?(In lakhs)</h3>
         <input id="second" name="Present_Price" required="required"><br>

         <br><h3>Kilometers Drived?</h3>
         <input id="third" name="Kms_Driven" required="required"><br>

         <br><h3>owners(0 or 1 or 3) ?</h3>
         <input id="fourth" name="Owner" required="required"><br>
         <br><h3>Select Fuel type?</h3>
         <select name="Fuel_Type_Petrol" id="fuel" required="required">
            <option value="Petrol">Petrol</option>
            <option value="Diesel">Diesel</option>
            <option value="Diesel">CNG</option>
         </select><br><br>
```

```
          <br><h3>Owner type</h3>
           <select name="Seller_Type_Individual" id="resea" required="required">
               <option value="Dealer">Dealer</option>
               <option value="Individual">Individual</option>
           </select><br>
           <br><h3>Transmission type</h3>
           <select name="Transmission_Mannual" id="research"
required="required">
               <option value="Mannual">Manual Car</option>
               <option value="Automatic">Automatic Car</option>
           </select><br><br><br>
           <br><input type="submit" value="submit">
       </form>
       <br><br><h3>{{ prediction_text }}<h3>
   </div>
   <style>
       body {
            background: linear-gradient(135deg, #cfd4d6, #9b59b6);
           text-align: center;
           padding: 0px;
       }
       input[type="submit"]{
       width: 20%;
       height: 50px;
       border: 1px solid;
       border-radius: 25px;
       font-size: 18px;
       color: #e9f4fb;
       padding-right: px;
       font-weight: 700;
       cursor: pointer;
       background: linear-gradient(135deg, #cfd4d6, #9b59b6);
```

```
margin-left: 5px;
}
input[type="submit"]:hover{
background: linear-gradient(-135deg, #cfd4d6, #9b59b6);
transition: .5s;
}
#research {
font-size: 16px;
width: 100px;
height: 45px;
top: 23px;
}
#box {
border-radius: 60px;
border-color: 45px;
border-style: solid;
font-family: cursive;
text-align: center;
background-color: rgb(168, 131, 61);
font-size: medium;
position: absolute;
width: 700px;
bottom: 9%;
height: 850px;
right: 30%;
padding: 0px;
margin: 0px;
font-size: 14px;
}

#fuel {
width: 199px;
```

```
height: 45px;
text-align: center;
border-radius: 5px;
font-size: 16px;
}

#fuel:hover {
background-color: rgb(255, 255, 255);
}

#research {
width: 199px;
height: 45px;
text-align: center;
border-radius: 5px
}

#research:hover {
background-color: rgb(255, 255, 255);
}

#resea {
width: 199px;
height: 45px;
text-align: center;
border-radius: 5px;
font-size: 16px;
}

#resea:hover {
background-color: rgb(255, 255, 255);
}
```

```
#sub {
width: 236px;
height: 45px;
text-align: center;
border-radius: 5px;
font-size: 16px;
}

#sub:hover {
background-color: rgb(255, 255, 255);
}

#first {
height: 45px;
font-size: 16px;
text-align: center;
border-radius: 5px;
border: 1px solid #ccc;
}

#second {
height: 45px;
font-size: 16px;
text-align: center;
border-radius: 5px;
border: 1px solid #ccc;
}

#third {
height: 45px;
font-size: 16px;
```

```
        text-align: center;
        border-radius: 5px;
        border: 1px solid #ccc;
        }

        #fourth {
        height: 45px;
        font-size: 16px;
        text-align: center;
        border-radius: 5px;
        border: 1px solid #ccc;
        }
    </style>
</body>
</html>
```

APP.PY FLASK:

```
from flask import Flask, render_template, request
import jsonify
import requests
import pickle
import numpy as np
import sklearn
from sklearn.preprocessing import StandardScaler
app = Flask(__name__)
model = pickle.load(open('random_forest_regression_model.pkl', 'rb'))
@app.route('/',methods=['GET'])
def Home():
    return render_template("first.html")
standard_to = StandardScaler()
```

```
@app.route("/red")
def red():
    return render_template('index.html')


@app.route("/predict", methods=['POST'])
def predict():
    Fuel_Type_Diesel=0
    if request.method == 'POST':
        Year = int(request.form['Year'])
        Present_Price=float(request.form['Present_Price'])
        Kms_Driven=int(request.form['Kms_Driven'])
        Kms_Driven2=np.log(Kms_Driven)
        Owner=int(request.form['Owner'])
        Fuel_Type_Petrol=request.form['Fuel_Type_Petrol']
        if(Fuel_Type_Petrol=='Petrol'):
                Fuel_Type_Petrol=1
                Fuel_Type_Diesel=0
        else:
            Fuel_Type_Petrol=0
            Fuel_Type_Diesel=1
        Year=2020-Year
        Seller_Type_Individual=request.form['Seller_Type_Individual']
        if(Seller_Type_Individual=='Individual'):
            Seller_Type_Individual=1
        else:
            Seller_Type_Individual=0
        Transmission_Mannual=request.form['Transmission_Mannual']
        if(Transmission_Mannual=='Mannual'):
            Transmission_Mannual=1
        else:
            Transmission_Mannual=0
```

```
prediction=model.predict([[Present_Price,Kms_Driven2,Owner,Year,Fuel_Type_D
iesel,Fuel_Type_Petrol,Seller_Type_Individual,Transmission_Mannual]])
        output=round(prediction[0],2)
        if output<0:
            return render_template('index.html;',prediction_texts="Sorry you cannot
sell this car")
        else:
            return render_template('index.html',prediction_text="You Can Sell The Car
at {}".format(output))
    else:
        return render_template('index.html')

if __name__=="__main__":
    app.run(debug=True)
```

FIRST.HTML:

```
<!DOCTYPE html>
<!-- Created By CodingLab - www.codinglabweb.com -->
<html lang="en" dir="ltr">
  <head>
    <meta charset="UTF-8">
    <!---<title> Responsive Registration Form | CodingLab </title>--->
    <link rel="stylesheet" href="first.css"/>
     <meta name="viewport" content="width=device-width, initial-scale=1.0">
   </head>
    <body>
        <div class="predict">
             <div><h1>Car price prediction</h1></div>
             <br>           <h3>Introduction</h3>
            <p> car resale valu prediction is the system to predict the amount of resale
```

```
<br>        value based on the parameters provided by the user</p>
        <h3>Car price prediction using random forest regressor</h3>
        <p> leading organaizations are collecting tons of data every day to drive
buisness
            <br> decisions and solutions from it.
        </p>
        <h3>Requirements for predicting price</h3>
        .Manufacturing Year <br>
        .Showroom price<br>
        .Miles driven<br>
        .Number of historical owners<br>
        .Fuel type<br>
        .Owner type<br>
        .Transmission type<br>
        <h3>Some Regression Algorithms</h5>
            .Linear Regression<br>
            .Decision Tree Regressor<br>
            .Support Vector Regressor<br>
            .KNN Regressor<br>
            .Random Forest Regressor<br>
        <br><div class="submit">
            <a href="{{url_for('red')}}">Predict</a>
            </div>
        </div>
    </body>
</html>
```

FIRST.CSS:

```
body{
    background: linear-gradient(135deg, #cfd4d6, #9b59b6);
```

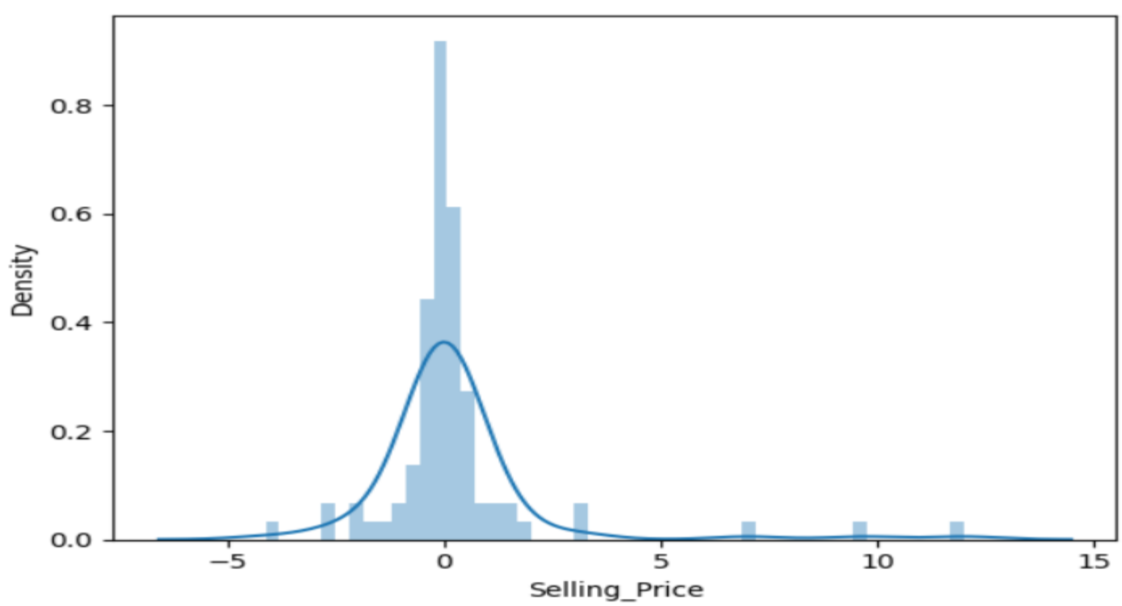
```
}
.predict{
    background: linear-gradient(135deg, #cfd4d6, #9b59b6);
    padding-right:0%;
}
input[type="submit"]
{
    width: 20%;
    height: 50px;
    border: 1px solid;
    border-radius: 50px;
    font-size: 18px;
    color: #e9f4fb;
    padding-right: 0px;
    font-weight: 700;
    cursor: pointer;
    background: linear-gradient(135deg, #cfd4d6, #9b59b6);
    margin-left: 5px;
    align-content: center;
    }

  .predict h1{
        text-align: center;
    }
 .submit a{
    border: 2px solid blueviolet;
    border-radius: 20px;
    font-size:26px;
    text-decoration: none;
    padding: 3px;
margin-left: 250px;
 }
```

```
.submit a:hover{
  background-color: blueviolet;
  color:white;
  transition: 0.5s;
}
```

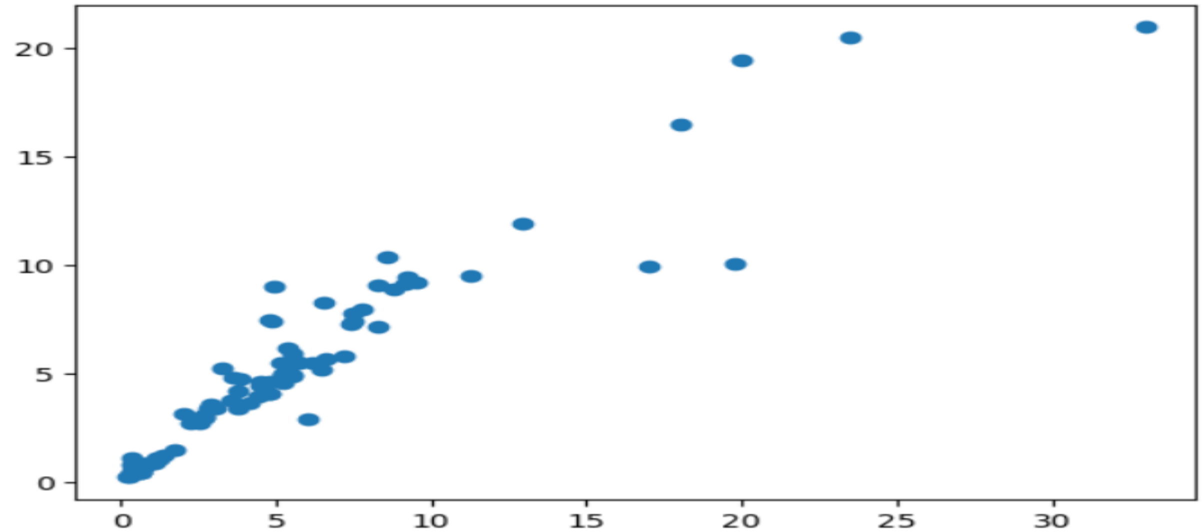
8. TESTING:

8.1 Test Cases:

Random Forest is an adaptable, simple to utilize AI calculation that produces, even without hyperboundary tuning, an incredible outcome more often than not. It is likewise perhaps the most utilized calculations, in view of its effortlessness and variety (it very well may be utilized for both order and relapse errands). In this post we'll figure out how the arbitrary woodland calculation functions, how it contrasts from different calculations and how to utilize it and grapically represented in fig.



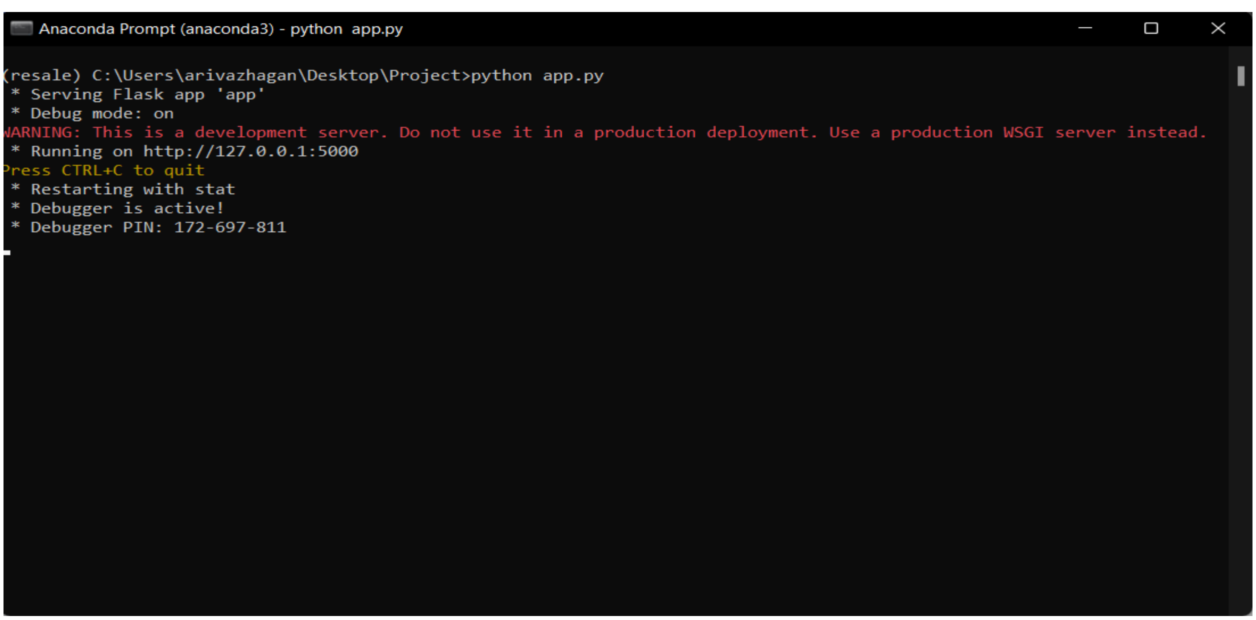
8.2 User Acceptance Testing:

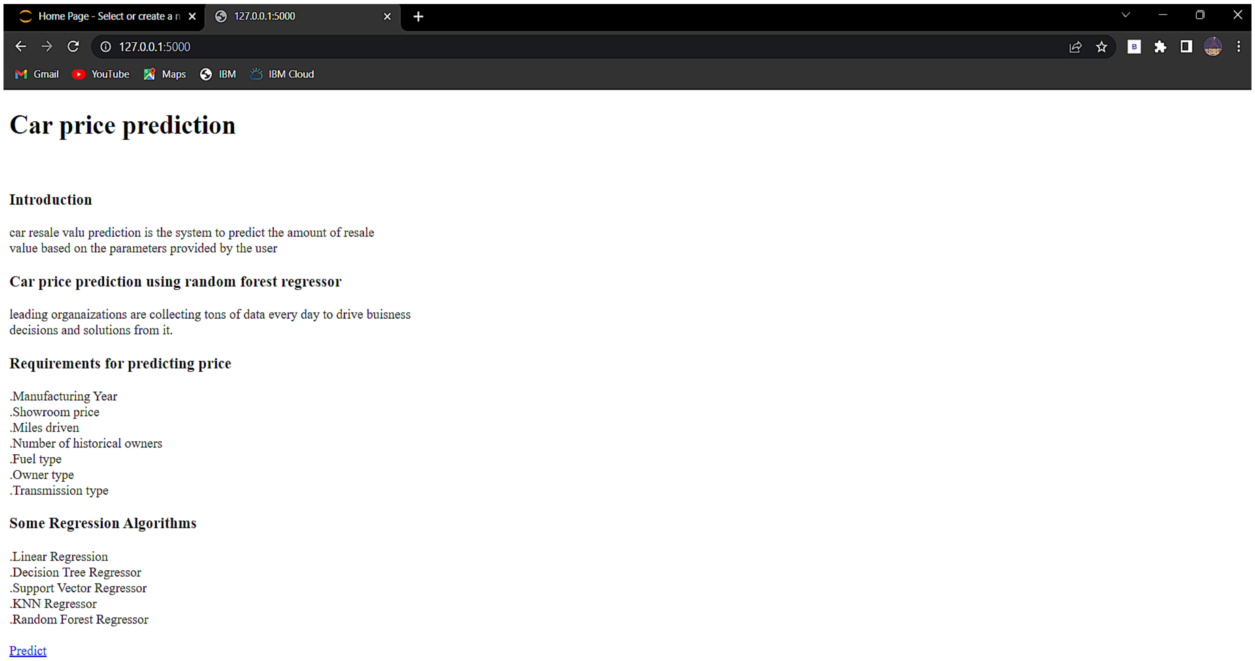


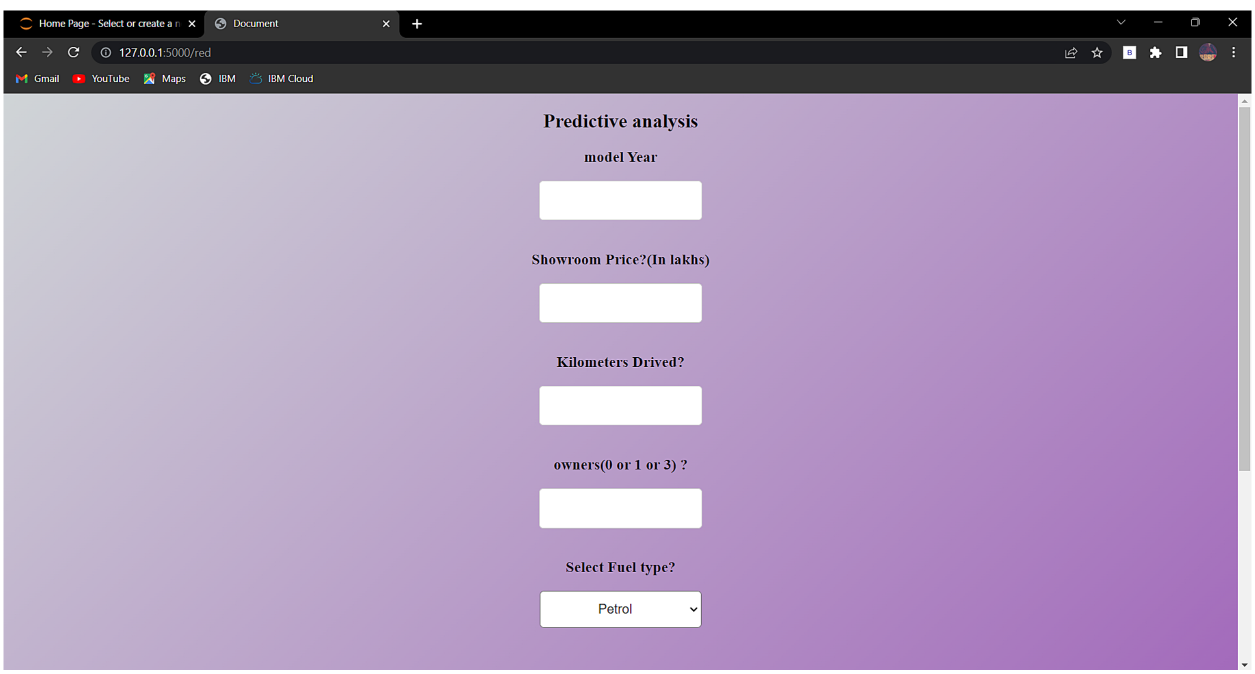
9. RESULTS

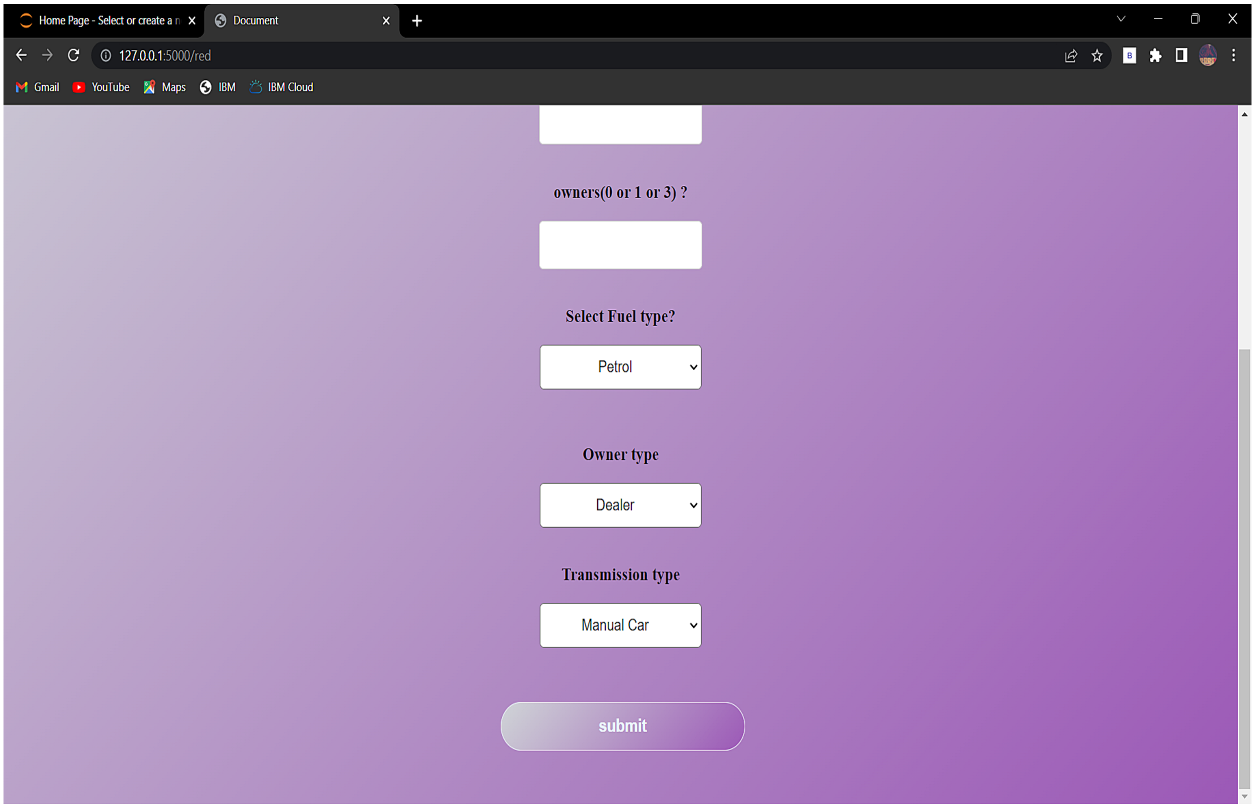
9.1 Performance Metrics

Metrics considered to test the strength of the algorithm are Mean Absolute Error, Mean Squared Error, Root Mean Squared Error. These three metrics are used to evaluate both the regression algorithms. The three metrics have lower values for Xgboost, hence it has higher accuracy over the random forest algorithm. Table 1 shows the result analysis of both the algorithm.




Home Page - Select or create a n
127.0.0.1:5000
127.0.0.1:5000
Gmail
YouTube
Maps
IBM
IBM Cloud
Car price prediction
Introduction
car resale valu prediction is the system to predict the amount of resale value based on the parameters provided by the user
Car price prediction using random forest regressor
leading organaizations are collecting tons of data every day to drive buisness decisions and solutions from it.
Requirements for predicting price
.Manufacturing Year
.Showroom price
.Miles driven
.Number of historical owners
.Fuel type
.Owner type
.Transmission type
Some Regression Algorithms
.Linear Regression
.Decision Tree Regressor
.Support Vector Regressor
.KNN Regressor
.Random Forest Regressor
Predict


Home Page - Select or create a n
Document
127.0.0.1:5000/red
Gmail
YouTube
Maps
IBM
IBM Cloud
Predictive analysis
model Year
Showroom Price?(In lakhs)
Kilometers Drived?
owners(0 or 1 or 3) ?
Select Fuel type?
Petrol



10. ADVANTAGES & DISADVANTAGES

Advantages and Disadvantages of the Random Forest Algorithm

1. Random forest is a very versatile algorithm capable of solving both classification and regression tasks.
2. Also, the hyperparameters involved are easy to understand and usually, their default values result in good prediction.
3. Random forest solves the issue of overfitting which occurs in decision trees.
4. One limitation of Random forest is, too many trees can make the processing of the algorithm slow thereby making it ineffective for prediction on real-time data.

11. CONCLUSION

we started with uderstanding the use casde of machine learning in the automative industry and how machine learning has transformed the driving experience.moving on we looked at the various factors that affect the resale value of a used car performed exploratory data analysis (EDA).Further,we build a random forest rtegression model to predict the resale value of a used car.finally,we evaluated the perfomance of the model using the R squared score and residual plot.

we could also used simpler regression algorithm like linear regression and lesso Regression.stil,we need to make sure there are no outliers in the dataset befor implementing them.pair plots and scatter plots help visalize the outliers.

12. FUTURE SCOPE

To get a clear picture of current industry trends and their implications for the future of car sales and the automotive aftermarket, including impact on oem revenues and profit, this study seeks to answer the following key questions:

- how will trends affect traditional business segment's vehicle sales and aftersales.
- what are opportunities and risks involved with new business segments such as mobility and digital services
- when and how will these developments and trend effects differ across core markets

13. APPENDIX

source: https://github.com/IBM-EPBL/IBM-Project-42423-1660662404