PROJECT BASED EXPERIENTIAL LEARNING PROGRAM (NALAIYA THIRAN)

CAR RESALE VALUE PREDICTION

A PROJECT REPORT

Submitted by

PAVAN KALYANN K	**(111719106109)**
MUGUNDAN P	**(111719106094)**
MAGANI MUNEENDRA	**(111719106088)**
SAI KARAN T	**(111719106303)**

TEAM ID : PNT2022TMID16117

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

R.M.K.ENGINEERING COLLEGE
(An Autonomous Institution)
R.S.M. Nagar, Kavaraipettai-601 206

NOVEMBER 2022

PROJECT REPORT

Car Resale Value Prediction

TEAM ID : PNT2022TMID16117

1. INTRODUCTION

1.1 Project Overview :

The production of cars has been steadily increasing in the past decade, with over 70 million passenger cars being produced in the year 2016. This has given rise to the used car market, which on its own has become a booming industry. The recent advent of online portals has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of a used car in the market. Using Machine Learning Algorithms such as Lasso Regression, Multiple Regression and Regression trees, we will try to develop a statistical model which will be able to predict the price of a used car, based on previous consumer data and a given set of features. We will also be comparing the prediction accuracy of these models to determine the optimal one.

The price of a new car 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 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. 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.

To overcome this problem we have developed a model which will be highly effective. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value. Because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user's inputs.

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models . We will compare the performance of various machine learning algorithms like Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Decision Tree Regressor and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user's inputs.

1.2 Purpose / Objectives :

1) You’ll be able to understand the problem to classify if it is a regression or a classification kind of problem.

2) You will be able to know how to pre-process/clean the data using different data pre-processing techniques.

3) Applying different algorithms according to the dataset

4) You will be able to know how to evaluate the model.

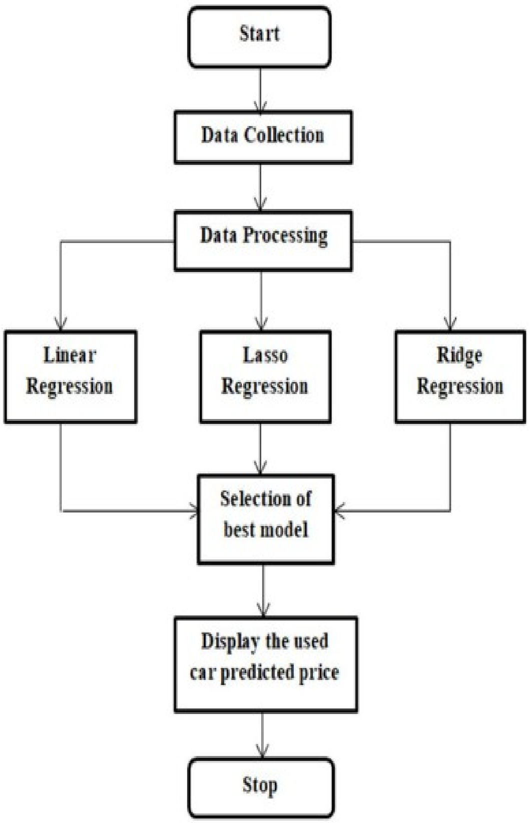
5)You will be able to build web applications using the Flask framework.

6) To develop a efficient and effective model which predicts the price of a used car according to user’s inputs.

7) To achieve good accuracy.

8) To develop a User Interface(UI) which is user-friendly and takes input from the user and predicts the price.

Project Overview



2. LITERATURE SURVEY

2.1 Existing Problem :

Predicting the exact price for a car in present days is becomming a difficult task. With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. In many developed countries, it is common to lease a car rather than buying it outright. After the lease period is over, the buyer has the possibility to buy the car at its residual value, i.e. its expected resale value. Therefore, this project helps in predicting the correct resale value of the car based on its condition.

The market for used cars has seen an upsurge in demand, which has impacted both purchasers and sellers' businesses increased. Expertise is needed for dependable and accurate prediction. knowledge of the subject because the cost of cars is a factor on a number of crucial criteria. In this study, a supervised Regression using the KNN (K Nearest Neighbor) machine learning algorithm method to evaluate used-car prices. Through this investigation, A variety of trained to test ratios were used to analyse the data. As as a consequence, the suggested model is fitted with an accuracy of about 85% like the improved model. The predictions are then evaluated and compared in order to find those which provide the best performances. A seemingly easy problem turned out to be indeed very difficult to resolve with high accuracy. All the four methods provided comparable performance. In the future, we intend to use more sophisticated algorithms to make the predictions . Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. On a dataset made up of the selling prices of various makes and models across American cities, we put several learning techniques into practise and evaluate their effectiveness. Our findings demonstrate that, while computationally intensive, the Random Forest model and K-Means clustering with linear regression produce the best outcomes.

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) Ananth is an Auto Sales Representative who needs a way to accurately predict the value of used car because he needs to satisfies his customer.

2) Sujith needs a way to predict the value of used car by taking its model name and seller type because he doesn’t want old model cars provided that seller type is inguinal.

3) Sruthi needs a way to predict the value of used car because it’s difficult to anticipate the selling price of a used car.

4) User is an explorer who needs a way to predict the value of used car based on mileage driven and transmission types because he wants to be low level petrol in run more kilometers and automatic types.

5) User is an owner who needs a way to predict the accurate value of used car because he wants to know the actual worth of their car and to sell it.

2.2 References :

[1] Pudaruth, Sameerchand. "Predicting the price of used cars using machine learning techniques." Int. J. Inf. Comput. Technol 4, no. 7 (2014): 753-764.

[2] Monburinon, Nitis, Prajak Chertchom, Thongchai Kaewkiriya, Suwat Rungpheung, Sabir Buya, and Pitchayakit Boonpou. "Prediction of prices for used car by using regression models." In 2018 5th International Conference on Business and Industrial Research (ICBIR),
pp. 115-119. IEEE, 2018.

[3] Gegic, Enis, BecirIsakovic, Dino Keco, Zerina Masetic, and Jasmin Kevric. "Car price prediction using machine learning techniques." TEM Journal 8, no. 1 (2019): 113.

[4] Noor, Kanwal, and Sadaqat Jan. "Vehicle price prediction system using machine learning techniques." International Journal of Computer Applications 167, no. 9 (2017): 27-31.

[5] https://ieeexplore.ieee.org/Xplore/home.jsp

[6]https://www.analyticsvidhya.com/blog/2018/08/knearestneighbor-introductionregression-python/

[7] https://machinelearningmastery.com/k-fold-cross-validation

2.3 Problem Statement Definition :

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle’s price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models . We will compare the performance of various machine learning algorithms like Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Decision Tree Regressor and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user’s inputs.

The Project Flow is as follows:

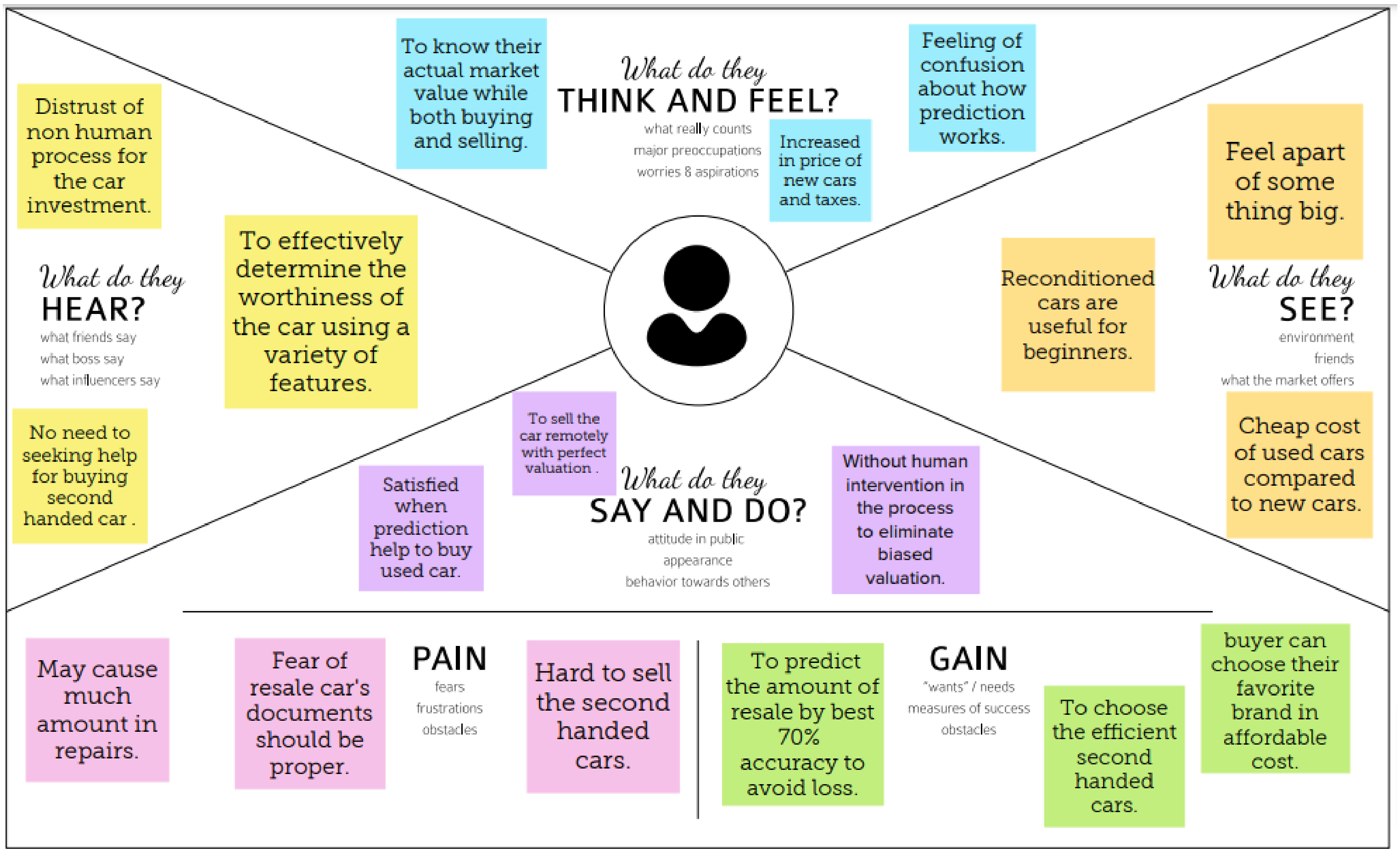
1) The user interacts with the UI (User Interface) to enter the input features.

2) Entered input features are analyzed by the model which is integrated.

3) Once the model analyses the input, the prediction is showcased on the UI.

3. IDEATION & PROPOSED SOLUTION :

3.1 Empathy Map Canvas



What do they THINK AND FEEL?

what really counts
major preoccupations
worries & aspirations

To know their actual market value while both buying and selling.

Increased in price of new cars and taxes.

Feeling of confusion about how prediction works.

What do they HEAR?

what friends say
what boss say
what influencers say

Distrust of non human process for the car investment.

To effectively determine the worthiness of the car using a variety of features.

No need to seeking help for buying second handed car .

What do they SEE?

environment
friends
what the market offers

Feel apart of some thing big.

Reconditioned cars are useful for beginners.

Cheap cost of used cars compared to new cars.

What do they SAY AND DO?

attitude in public
appearance
behavior towards others

To sell the car remotely with perfect valuation .

Satisfied when prediction help to buy used car.

Without human intervention in the process to eliminate biased valuation.

PAIN

fears
frustrations
obstacles

May cause much amount in repairs.

Fear of resale car's documents should be proper.

Hard to sell the second handed cars.

GAIN

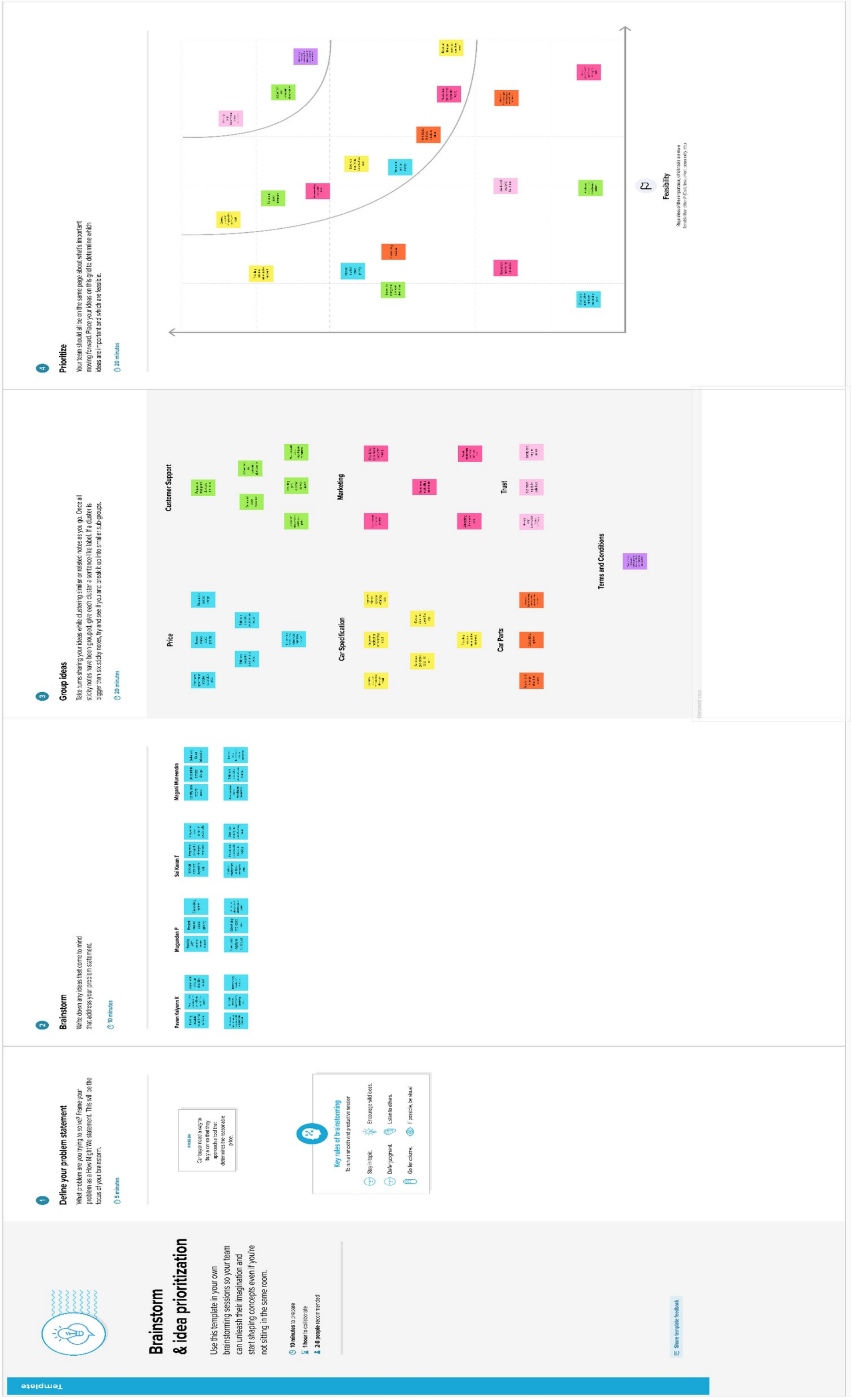
"wants" / needs
measures of success
obstacles

To predict the amount of resale by best 70% accuracy to avoid loss.

To choose the efficient second handed cars.

buyer can choose their favorite brand in affordable cost.

3.2 IDEATION & BRAINSTORMING



Template

Brainstorm & idea prioritization

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.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

1 Define your problem statement

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

5 minutes

PROBLEM

Car buyer need a way to buy a car so that they approach a tool that determines the resonable price.

Key rules of brainstorming

To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

2 Brainstorm

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

10 minutes

3 Group ideas

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

20 minutes

4 Prioritize

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

20 minutes

Feasibility

Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 Proposed Solution

Data Analysis

This section performs the selling price prediction using a dataset consisting of 8128 used car details. This dataset is prepared by cardhekho.com and available on kaggle. There are categorical as well as continuous features here.

Correlation Matrix

Visualising the correlation is an effective way of determining the dependencies.sometimes selling price has high correlation with the manufacturing year engine Max power and transmission. The engine and the manufacturing year has the same approximate correlation so we can select any one of them in the final set of features.

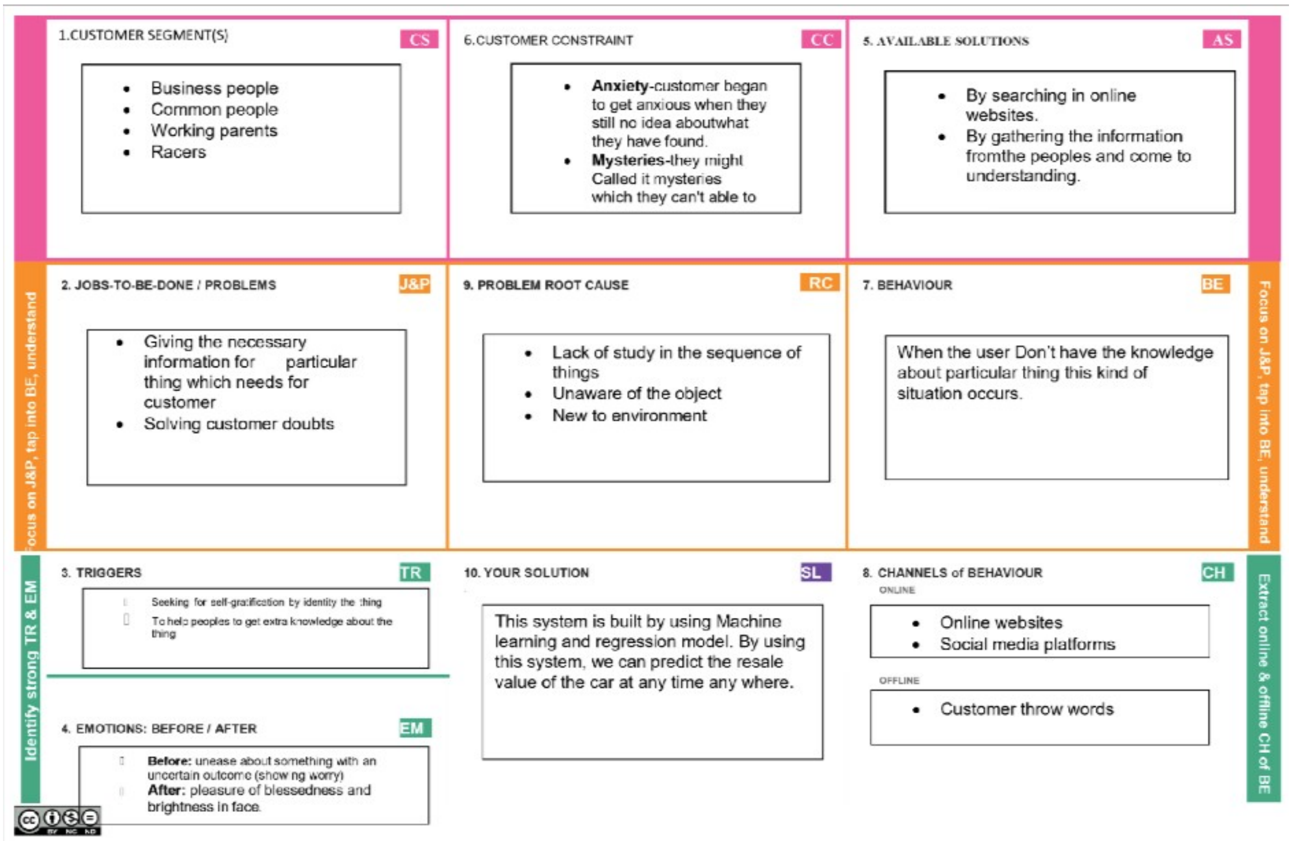
Pair plot

Pair plots allows to see both distributions of single variable and relationship between 2 variable. They are great method to identify trends for follow up analysis and are easily implemented in python. Scatter plots in pair plots also help in visualisation of outliers.

Random forest regression

Random forest is a supervised learning algorithm that uses an ensemble learning approach for regression and classification.The main principle behind the ensemble approach is that weak learners can learn from strong learners.Random forest operates by constructing multiple decision trees at training time.These decision trees are independently trained on bootstrap datasets.The final predicted value is calculated by taking the mean of the predictions by all the individual values.

3.4 Problem Solution Fit


Focus on J&P, tap into BE, understand
Identify strong TR & EM
Focus on J&P, tap into BE, understand
Extract online & offline CH of BE
CC BY NC ND

1.CUSTOMER SEGMENT(S) CS

- Business people
- Common people
- Working parents
- Racers

6.CUSTOMER CONSTRAINT CC

- **Anxiety**-customer began to get anxious when they still no idea aboutwhat they have found.
- **Mysteries**-they might Called it mysteries which they can't able to

5. AVAILABLE SOLUTIONS AS

- By searching in online websites.
- By gathering the information fromthe peoples and come to understanding.

2. JOBS-TO-BE-DONE / PROBLEMS J&P

- Giving the necessary information for particular thing which needs for customer
- Solving customer doubts

9. PROBLEM ROOT CAUSE RC

- Lack of study in the sequence of things
- Unaware of the object
- New to environment

7. BEHAVIOUR BE

When the user Don't have the knowledge about particular thing this kind of situation occurs.

3. TRIGGERS TR

- Seeking for self-gratification by identity the thing
- To help peoples to get extra knowledge about the thing

4. EMOTIONS: BEFORE / AFTER EM

- **Before:** unease about something with an uncertain outcome (showing worry)
- **After:** pleasure of blessedness and brightness in face.

10. YOUR SOLUTION SL

This system is built by using Machine learning and regression model. By using this system, we can predict the resale value of the car at any time any where.

8. CHANNELS of BEHAVIOUR CH

ONLINE

- Online websites
- Social media platforms

OFFLINE

- Customer throw words

4. REQUIREMNT ANALYSIS :

Anaconda Navigator :

Anaconda Navigator is a free and open-source distribution of the Python and R programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS. Conda is an open-source, cross-platform, package management system. Anaconda comes with great tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code.

For this project, we will be using Jupyter notebook and Spyder.

To install Anaconda navigator and to know how to use Jupyter Notebook & Spyder using Anaconda watch the video.

To build Machine learning models you must require the following packages.

Anaconda Navigator

Sklearn:

Scikit-learn is a library in Python that provides many unsupervised and supervised learning algorithms.

NumPy:

NumPy is a Python package that stands for 'Numerical Python'. It is the core library for scientific computing, which contains a powerful n-dimensional array object.

Pandas:

pandas is a fast, powerful, flexible, and easy to use open-source data analysis and manipulation tool, built on top of the Python programming language.

Matplotlib:

It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits.

Flask:

Web framework used for building Web applications.
1. Open anaconda prompt.
2. Type “pip install numpy” and click enter.
3. Type “pip install pandas” and click enter.
4. Type “pip install matplotlib” and click enter.
5. Type “pip install scikit-learn” and click enter.
6. Type “pip install Flask” and click enter.

4.1 Functional requirements

User Registration:

Registration through Website.

User Confirmation:

Confirmation via the Website.

Car Registration:

Registering the car details.

Value Prediction:

Predicting the car resale value.

4.2 Non- functional Requirements

Usability:

The platform will be a user friendly one as the only form of input received form the user is the vehicle data in the form of documents and ID Proofs and the verification is done.

Security:

The platform shall be made secure such a way that no data shall be leaked or accesed by unauthorised users.

Reliability:

The platform shall be made a more reliable one through proper prediction of car resale price and satisfying the customer needs.

Performance:

The price of the cars shall be predicted with appreciable amount of efficieny by using these machine learning techniques.

- Multiple Linear Regression Analysis
- K-Nearest Neighbours Algorithm
- Decision Trees
- Naïve Bayes classification

Availability:

The platform shall be made available for all the users who wish to sell their cars in the form of a website or a dedicated application.

Scalability:

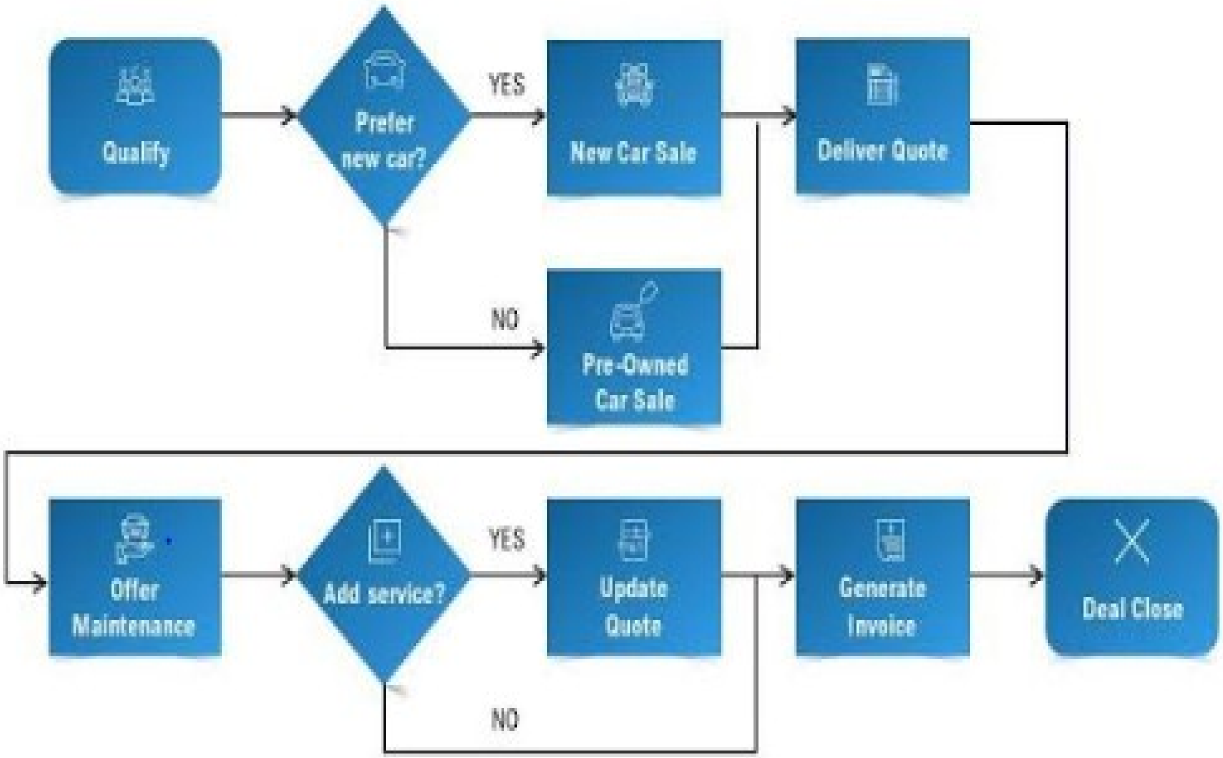
By using the 4 different machine learning techniques, the resale price shall be perdicted with nearly 60-70% of accuracy.

Other documents to be verified:

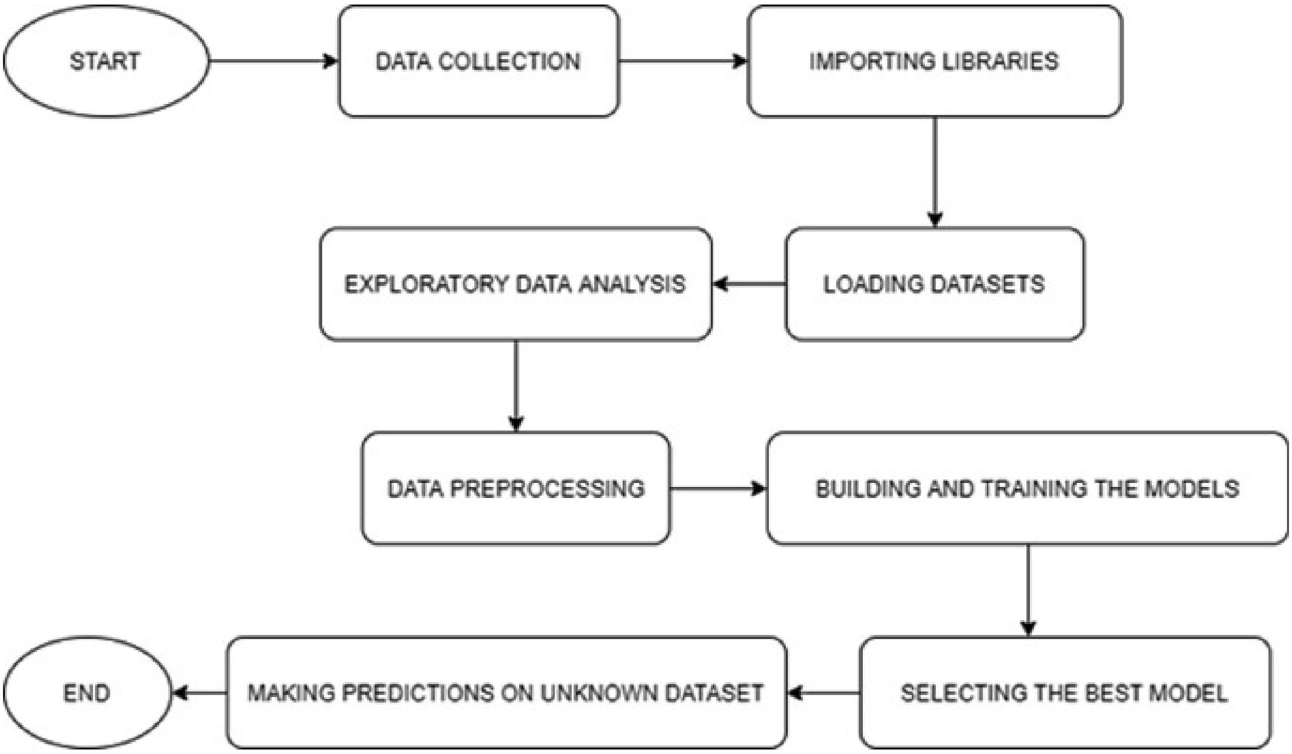
- Service records book.
- No Objection Certificate(NOC).
- Road Tax Receipt.
- Emission Certificate shall be verified

5. PROJECT DESIGN :

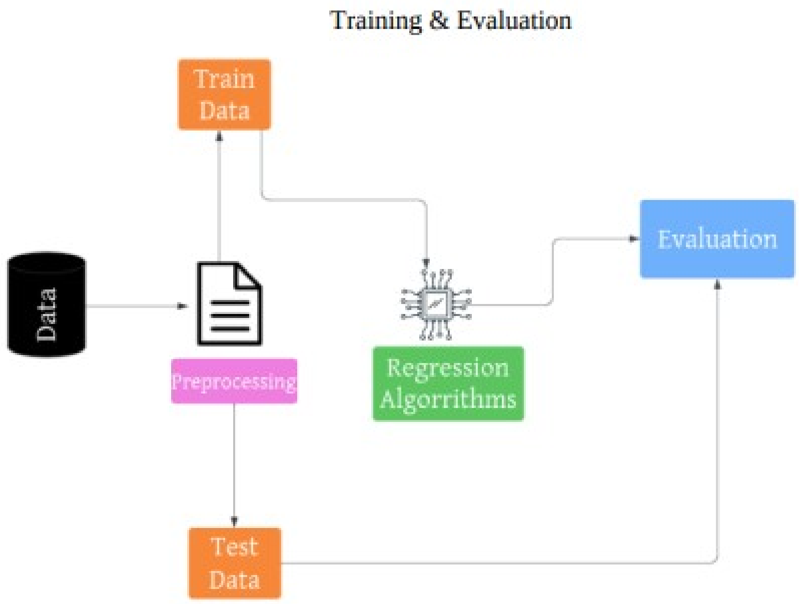
5.1 Data Flow Diagram

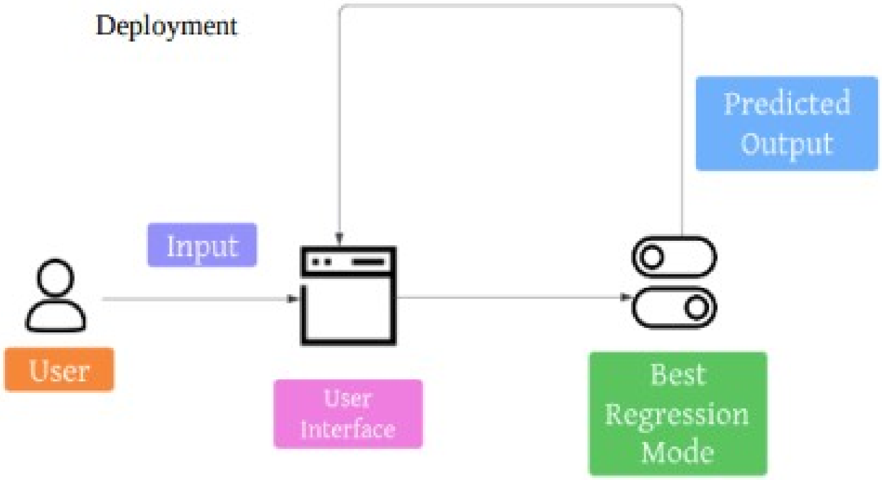


Flow of the process



5.2 Solution & Technical Architecture



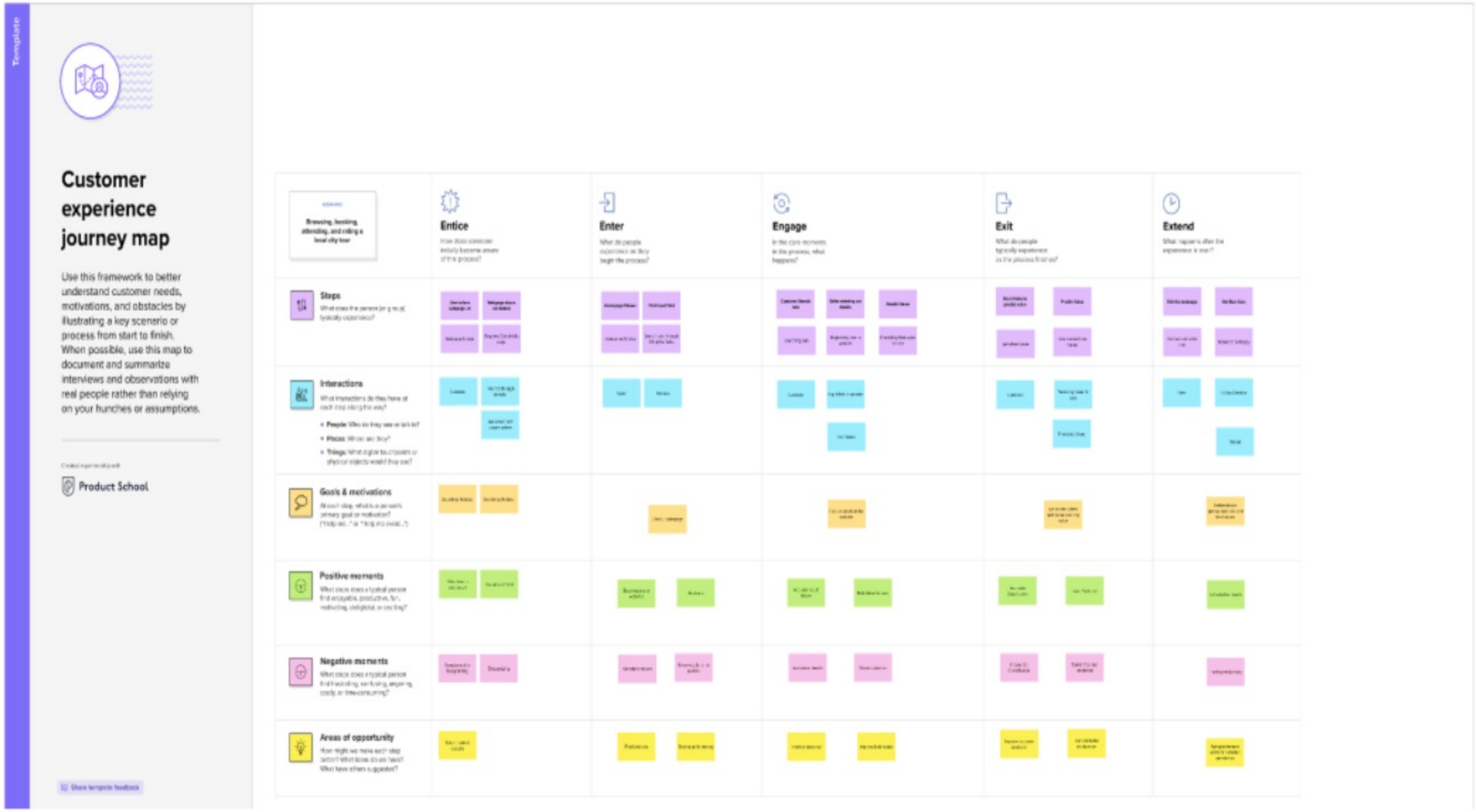


5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria
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
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login
		USN-4	As a user, I can register for the application through Gmail	I can register & access the application through G-mail
	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
	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
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 / dashboard

User Type	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Acceptance criteria**
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.
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.
	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 my home comfortably through OTP.

Customer Journey Map



Template

Customer experience journey map

Use this framework to better understand customer needs, motivations, and obstacles by illustrating a key scenario or process from start to finish. When possible, use this map to document and summarize interviews and observations with real people rather than relying on your hunches or assumptions.

Product School

Share template feedback

Browsing, booking, attending, and riding a local city tour

Entice
How does someone initially become aware of this process?

Enter
What do people experience as they begin the process?

Engage
In the core moments in the process, what happens?

Exit
What do people typically experience as the process finishes?

Extend
What happens after the experience is over?

Steps
What does the person (or group) typically experience?

Interactions
What interactions do they have at each step along the way?

- **People:** Who do they see or talk to?
- **Places:** Where are they?
- **Things:** What digital touchpoints or physical objects would they use?

Goals & motivations
At each step, what is a person's primary goal or motivation? ("Help me..." or "Help me avoid...")

Positive moments
What steps does a typical person find enjoyable, productive, fun, motivating, delightful, or exciting?

Negative moments
What steps does a typical person find frustrating, confusing, angering, costly, or time-consuming?

Areas of opportunity
How might we make each step better? What ideas do we have? What have others suggested?

6. PROJECT PLANNING & SCHEDULING :

6.1 Sprint Planning & Estimation

Sprint 1 :

Car.html

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

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

  </body>
</html>
```

Styles.css

```
.{
   margin: 0;
   padding: 0;
   box-sizing: border-box;
}
.bg-dark{
      background-color: #75767B;
}


.mt-50{
      margin-top: 50px;
}
#canvas{
   border: 2px solid black;
}
```

Sprint 2 :

Car.html

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

     </nav>
```

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

 </body>
</html>
```

Styles.css

```
<!DOCTYPE html>

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



</body>
</html>
```

Sprint 3 :

Car.html

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

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

  </body>
</html>
```

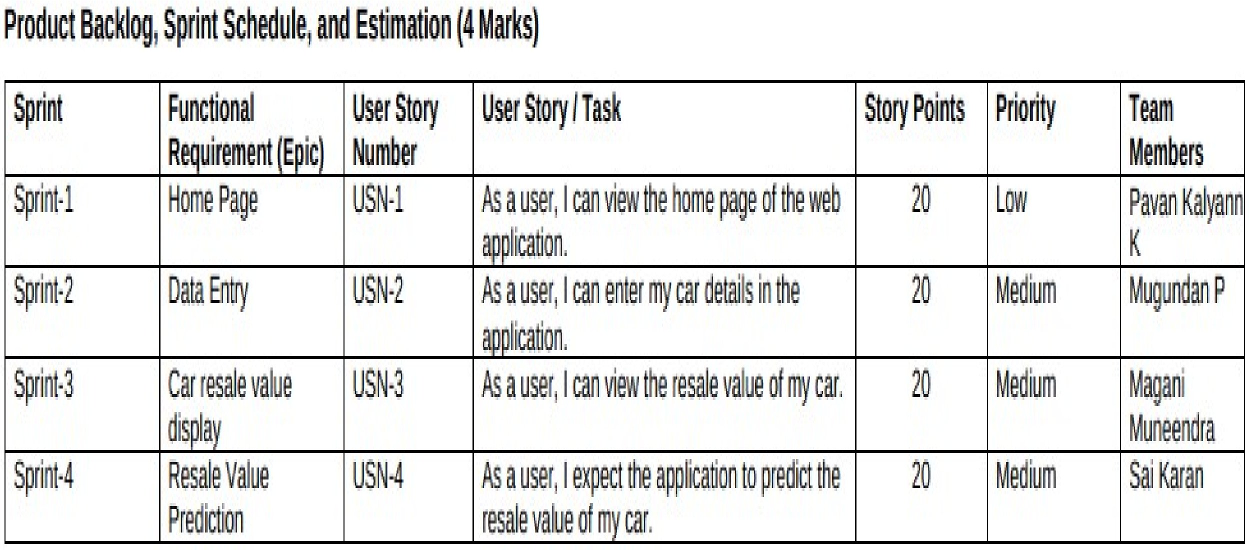
Predict.html

```
<!DOCTYPE html>

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


</body>
</html>
```

Sprint 4 : Result

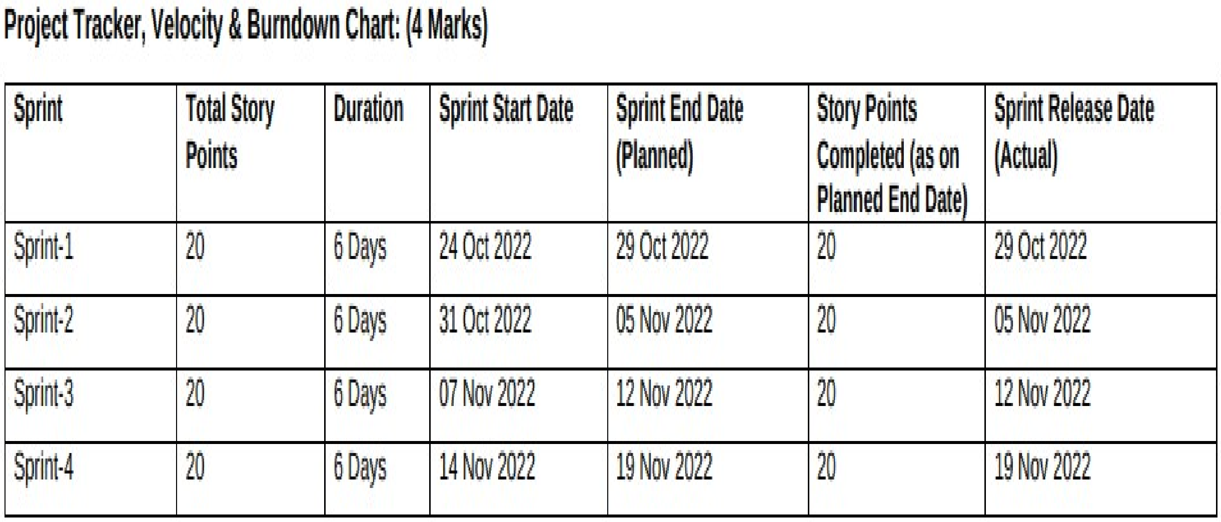
```
<!DOCTYPE html>

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



</body>
</html>
```

6.2 Sprint Delivery Plan



Product Backlog, Sprint Schedule, and Estimation (4 Marks)

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Home Page	USN-1	As a user, I can view the home page of the web application.	20	Low	Pavan Kalyann K
Sprint-2	Data Entry	USN-2	As a user, I can enter my car details in the application.	20	Medium	Mugundan P
Sprint-3	Car resale value display	USN-3	As a user, I can view the resale value of my car.	20	Medium	Magani Muneendra
Sprint-4	Resale Value Prediction	USN-4	As a user, I expect the application to predict the resale value of my car.	20	Medium	Sai Karan



Project Tracker, Velocity & Burndown Chart: (4 Marks)

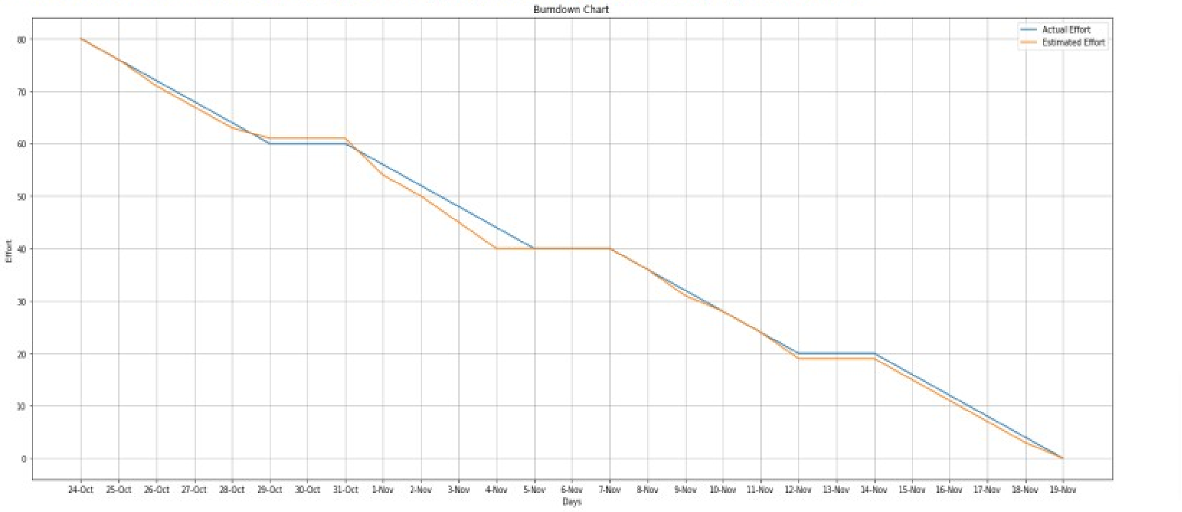
Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	05 Nov 2022
Sprint-3	20	6 Days	07 Nov 2022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

Velocity:
Imagine we have a 6-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)

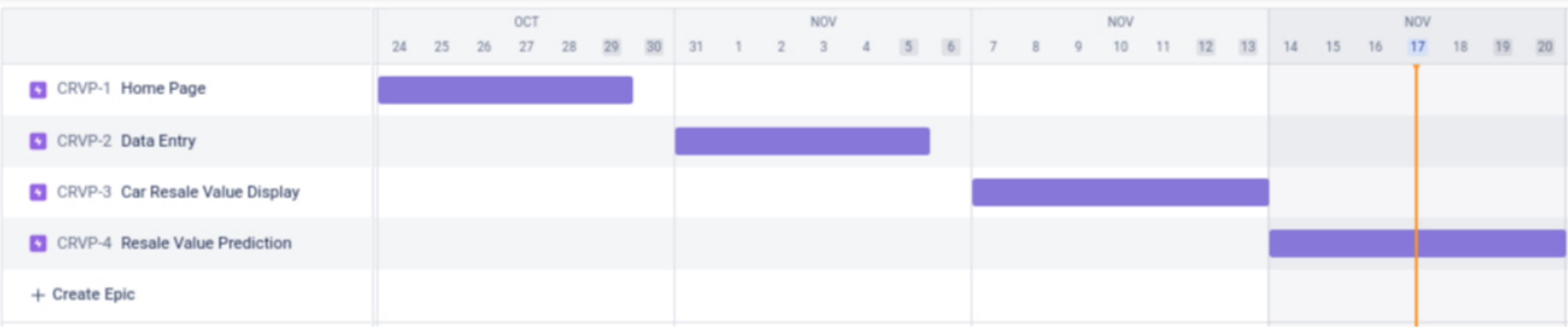
$$\textbf{Average Velocity} = \frac{20}{6} = 3.33$$

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.



6.3 Reports from JIRA



CRVP-1 Home Page

CRVP-2 Data Entry

CRVP-3 Car Resale Value Display

CRVP-4 Resale Value Prediction

+ Create Epic

7. CODING & SOLUTIONING :

Application

This is the final code used in the designing of the app or website for the resale prediction and it is done with the regression models.

```
from flask import Flask,render_template,request,redirect

from flask_cors import CORS,cross_origin
import pickle
import pandas as pd
import numpy as np

app=Flask(__name__)
cors=CORS(app)
model=pickle.load(open('LinearRegressionModel.pkl','rb'))
car=pd.read_csv('Cleaned_Car_data.csv')

@app.route('/',methods=['GET','POST'])
def index():
    companies=sorted(car['company'].unique())
    car_models=sorted(car['name'].unique())
    year=sorted(car['year'].unique(),reverse=True)
    fuel_type=car['fuel_type'].unique()

    companies.insert(0,'Select Company')
    return render_template('index.html',companies=companies,
car_models=car_models, years=year,fuel_types=fuel_type)


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

    company=request.form.get('company')
    car_model=request.form.get('car_models')
    year=request.form.get('year')
    fuel_type=request.form.get('fuel_type')
    driven=request.form.get('kilo_driven')

    prediction=model.predict(pd.DataFrame(columns=['name', 'company', 'year',
'kms_driven', 'fuel_type'],

data=np.array([car_model,company,year,driven,fuel_type]).reshape(1, 5)))
    print(prediction)

    return str(np.round(prediction[0],2))



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

Feature 1:

Car Details

This helps in collecting the details from the user like what is the model of the car, fuel type, etc.

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

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

 </body>
</html>
```

Styles

```
<!DOCTYPE html>

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

```
</body>
</html>
```

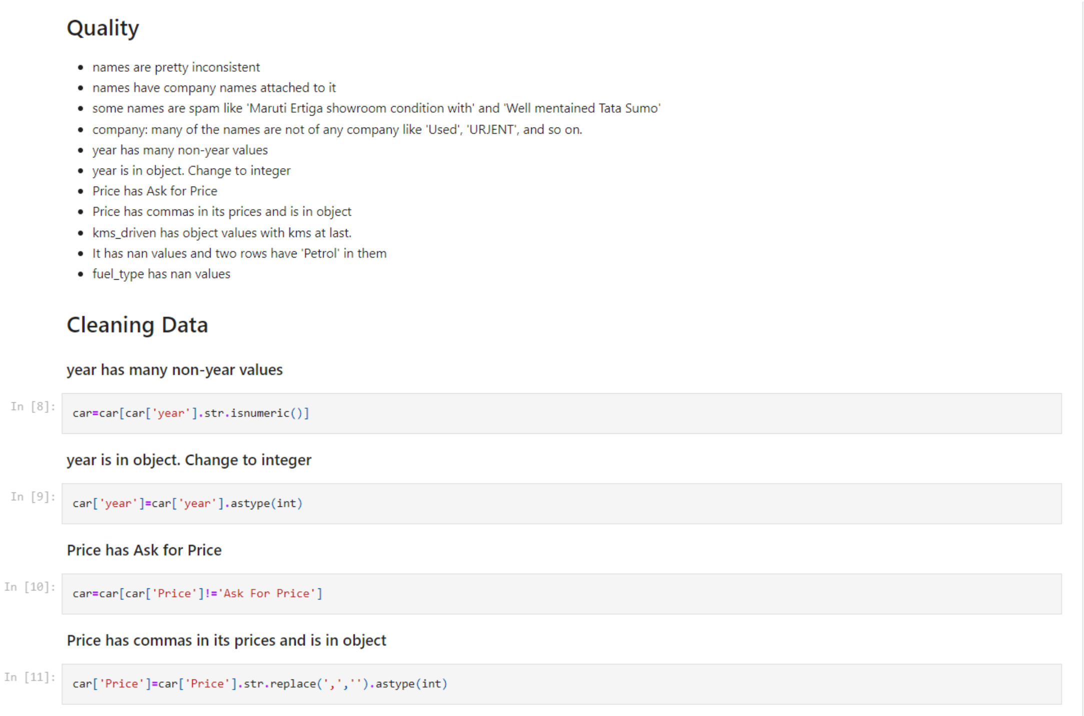
Feature 2 :

Prediction Purpose

This helps in predicting the final output based on the inputs given by the user.

This also helps in predicting accurate price for the car.

```
<!DOCTYPE html>

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


</body>
</html>
```



Quality

- names are pretty inconsistent
- names have company names attached to it
- some names are spam like 'Maruti Ertiga showroom condition with' and 'Well mentained Tata Sumo'
- company: many of the names are not of any company like 'Used', 'URJENT', and so on.
- year has many non-year values
- year is in object. Change to integer
- Price has Ask for Price
- Price has commas in its prices and is in object
- kms_driven has object values with kms at last.
- It has nan values and two rows have 'Petrol' in them
- fuel_type has nan values

Cleaning Data

year has many non-year values

```
In [8]: car=car[car['year'].str.isnumeric()]
```

year is in object. Change to integer

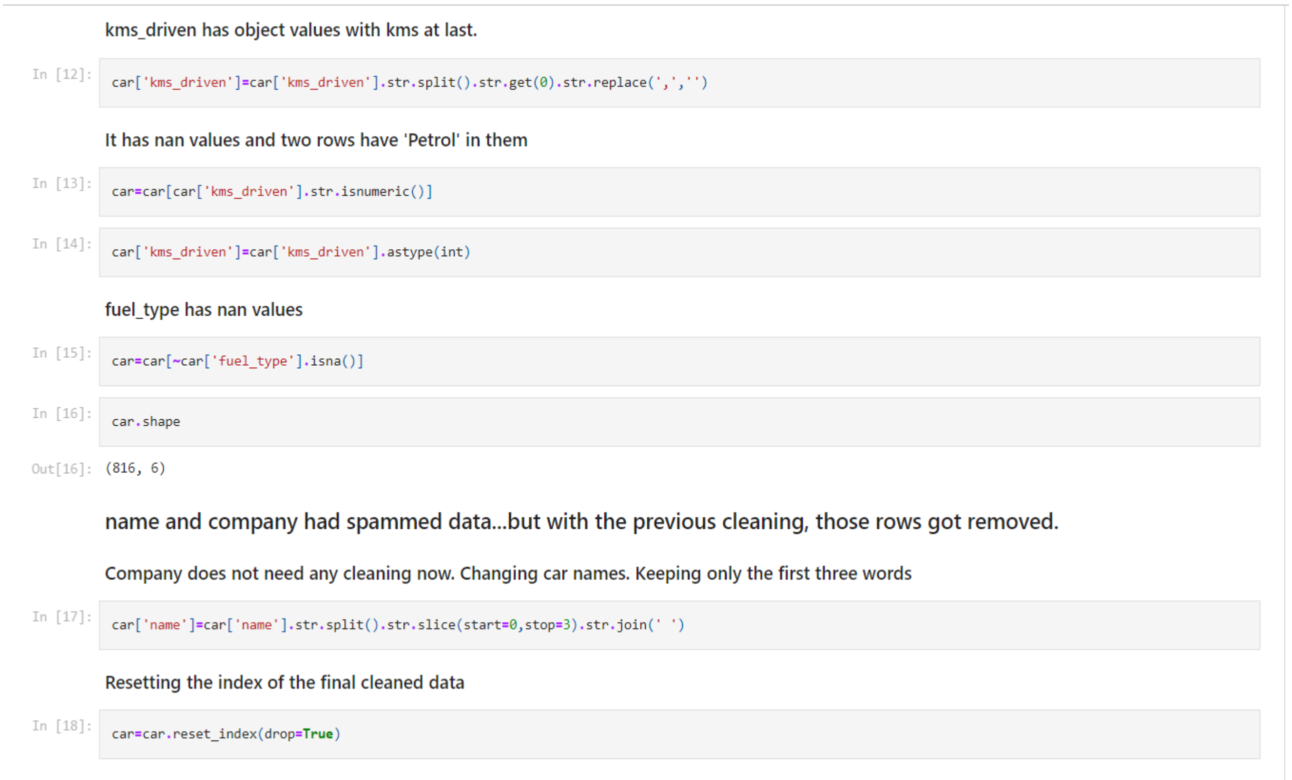
```
In [9]: car['year']=car['year'].astype(int)
```

Price has Ask for Price

```
In [10]: car=car[car['Price']!='Ask For Price']
```

Price has commas in its prices and is in object

```
In [11]: car['Price']=car['Price'].str.replace(',','').astype(int)
```



kms_driven has object values with kms at last.

```
In [12]: car['kms_driven']=car['kms_driven'].str.split().str.get(0).str.replace(',','')
```

It has nan values and two rows have 'Petrol' in them

```
In [13]: car=car[car['kms_driven'].str.isnumeric()]
```

```
In [14]: car['kms_driven']=car['kms_driven'].astype(int)
```

fuel_type has nan values

```
In [15]: car=car[~car['fuel_type'].isna()]
```

```
In [16]: car.shape
```

```
Out[16]: (816, 6)
```

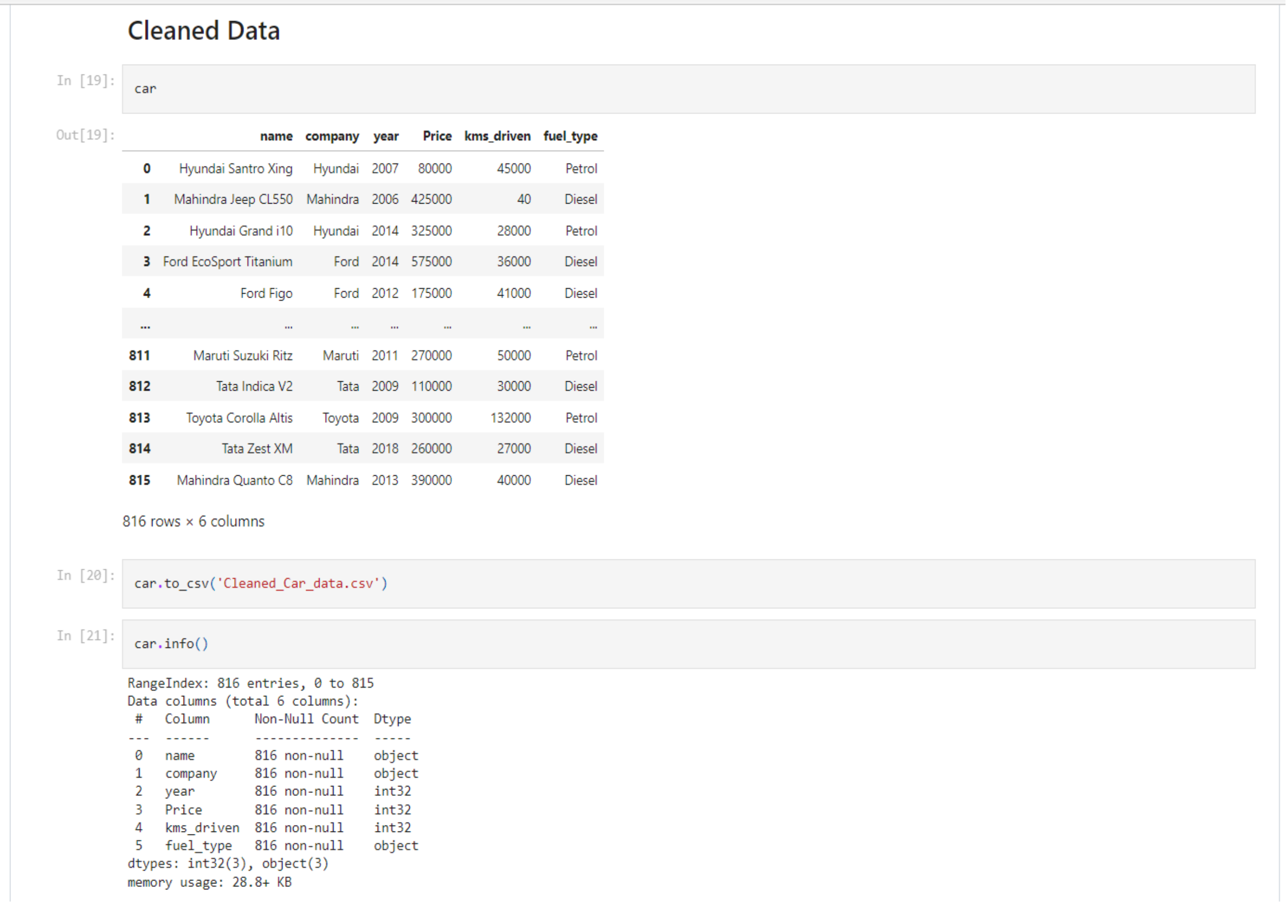
name and company had spammed data...but with the previous cleaning, those rows got removed.

Company does not need any cleaning now. Changing car names. Keeping only the first three words

```
In [17]: car['name']=car['name'].str.split().str.slice(start=0,stop=3).str.join(' ')
```

Resetting the index of the final cleaned data

```
In [18]: car=car.reset_index(drop=True)
```



Cleaned Data

In [19]:
```
car
```

Out[19]:

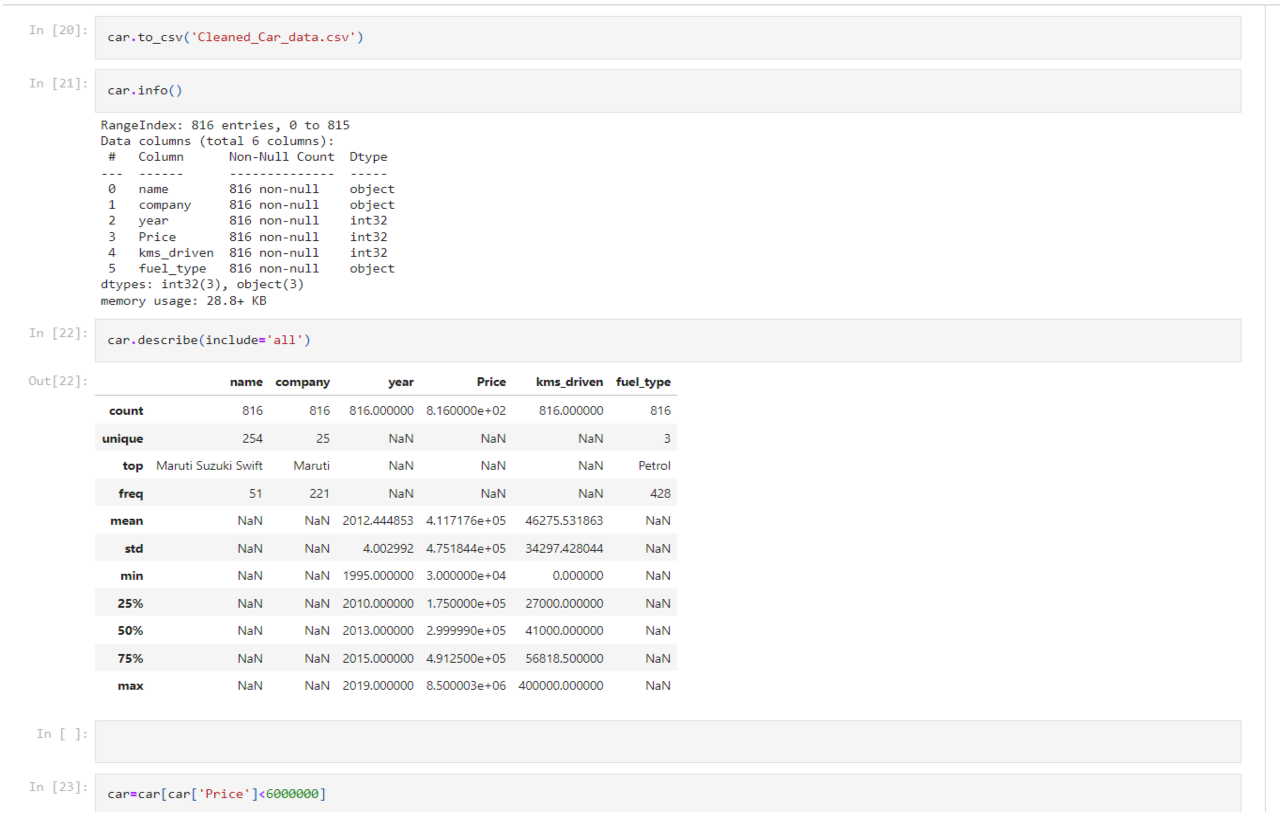
	name	company	year	Price	kms_driven	fuel_type
0	Hyundai Santro Xing	Hyundai	2007	80000	45000	Petrol
1	Mahindra Jeep CL550	Mahindra	2006	425000	40	Diesel
2	Hyundai Grand i10	Hyundai	2014	325000	28000	Petrol
3	Ford EcoSport Titanium	Ford	2014	575000	36000	Diesel
4	Ford Figo	Ford	2012	175000	41000	Diesel
...	...	...	...	...	...	...
811	Maruti Suzuki Ritz	Maruti	2011	270000	50000	Petrol
812	Tata Indica V2	Tata	2009	110000	30000	Diesel
813	Toyota Corolla Altis	Toyota	2009	300000	132000	Petrol
814	Tata Zest XM	Tata	2018	260000	27000	Diesel
815	Mahindra Quanto C8	Mahindra	2013	390000	40000	Diesel

816 rows × 6 columns

In [20]:
```
car.to_csv('Cleaned_Car_data.csv')
```

In [21]:
```
car.info()
```

```
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        816 non-null    object
 1   company     816 non-null    object
 2   year        816 non-null    int32 
 3   Price       816 non-null    int32 
 4   kms_driven  816 non-null    int32 
 5   fuel_type   816 non-null    object
dtypes: int32(3), object(3)
memory usage: 28.8+ KB
```



In [20]:
```
car.to_csv('Cleaned_Car_data.csv')
```

In [21]:
```
car.info()
```

```
RangeIndex: 816 entries, 0 to 815
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   name        816 non-null    object
 1   company     816 non-null    object
 2   year        816 non-null    int32 
 3   Price       816 non-null    int32 
 4   kms_driven  816 non-null    int32 
 5   fuel_type   816 non-null    object
dtypes: int32(3), object(3)
memory usage: 28.8+ KB
```

In [22]:
```
car.describe(include='all')
```

Out[22]:

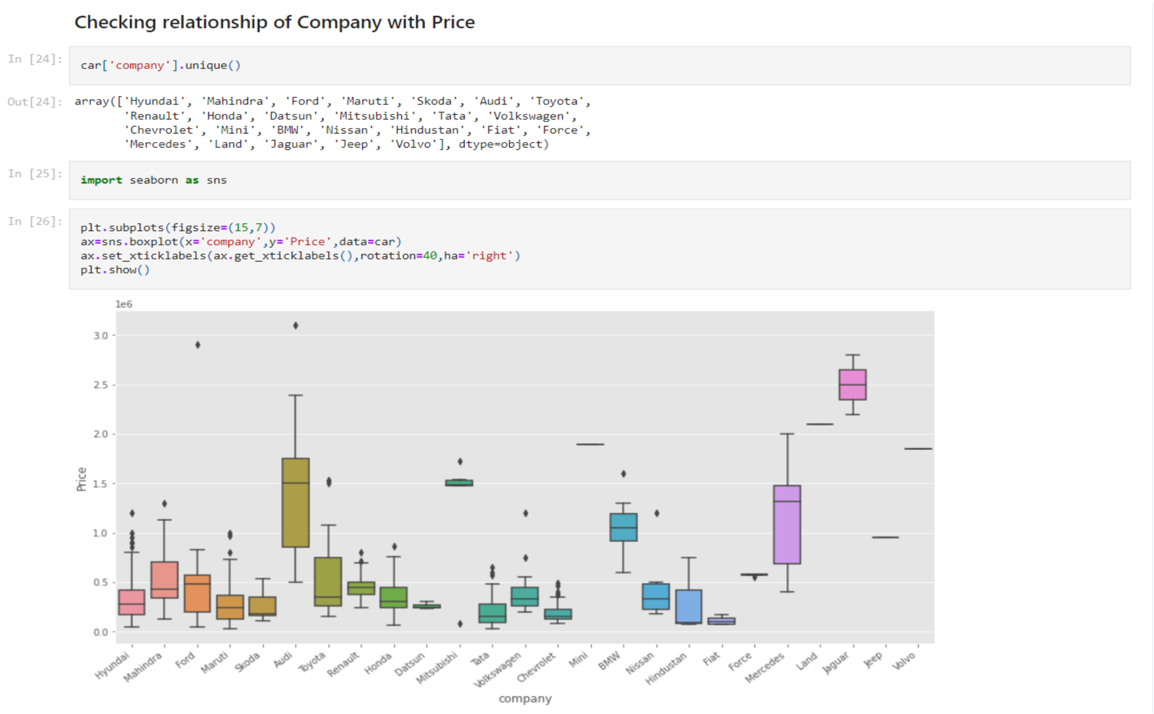
	name	company	year	Price	kms_driven	fuel_type
count	816	816	816.000000	8.160000e+02	816.000000	816
unique	254	25	NaN	NaN	NaN	3
top	Maruti Suzuki Swift	Maruti	NaN	NaN	NaN	Petrol
freq	51	221	NaN	NaN	NaN	428
mean	NaN	NaN	2012.444853	4.117176e+05	46275.531863	NaN
std	NaN	NaN	4.002992	4.751844e+05	34297.428044	NaN
min	NaN	NaN	1995.000000	3.000000e+04	0.000000	NaN
25%	NaN	NaN	2010.000000	1.750000e+05	27000.000000	NaN
50%	NaN	NaN	2013.000000	2.999990e+05	41000.000000	NaN
75%	NaN	NaN	2015.000000	4.912500e+05	56818.500000	NaN
max	NaN	NaN	2019.000000	8.500003e+06	400000.000000	NaN

In [ ]:

In [23]:
```
car=car[car['Price']<6000000]
```

DATA VISUALIZATIONS :

1) Checking relationship of Company with Price



Checking relationship of Company with Price

In [24]:

```
car['company'].unique()
```

Out[24]:

```
array(['Hyundai', 'Mahindra', 'Ford', 'Maruti', 'Skoda', 'Audi', 'Toyota',
       'Renault', 'Honda', 'Datsun', 'Mitsubishi', 'Tata', 'Volkswagen',
       'Chevrolet', 'Mini', 'BMW', 'Nissan', 'Hindustan', 'Fiat', 'Force',
       'Mercedes', 'Land', 'Jaguar', 'Jeep', 'Volvo'], dtype=object)
```

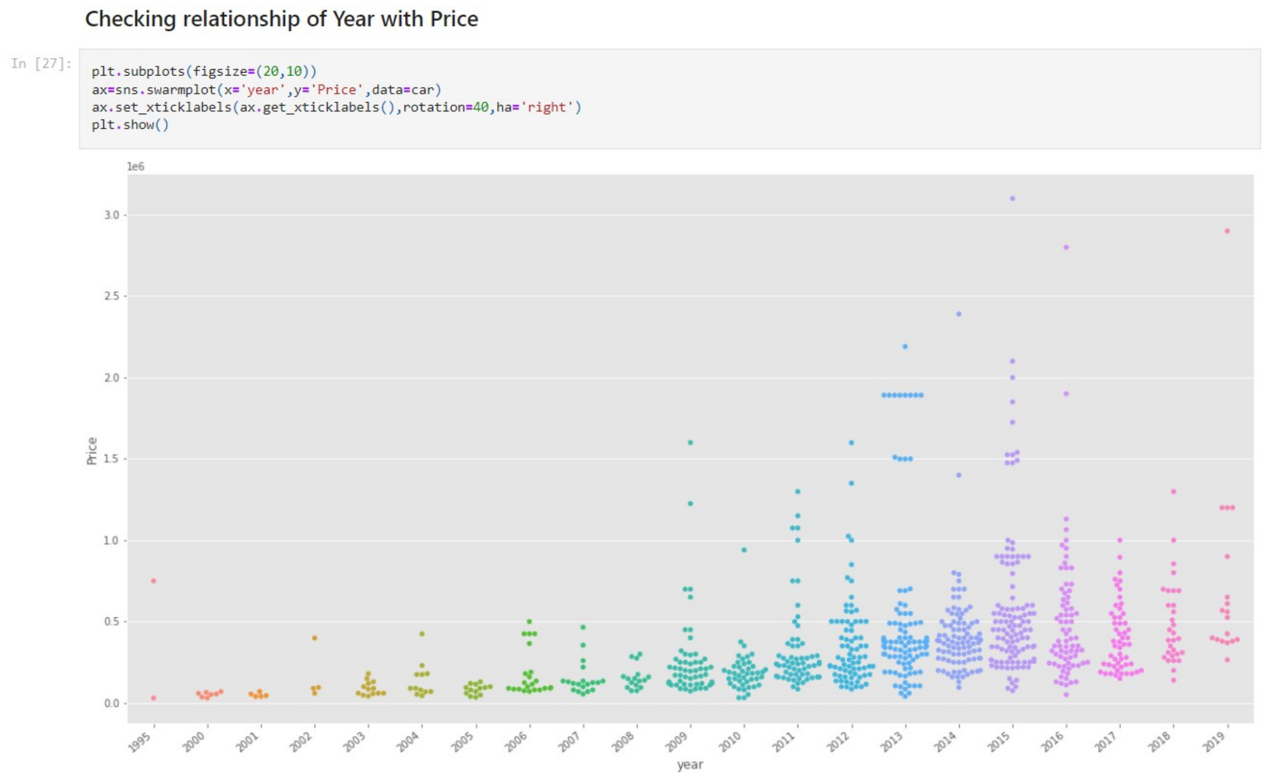
In [25]:

```
import seaborn as sns
```

In [26]:

```
plt.subplots(figsize=(15,7))
ax=sns.boxplot(x='company',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

2) Checking relationship of Year with Price

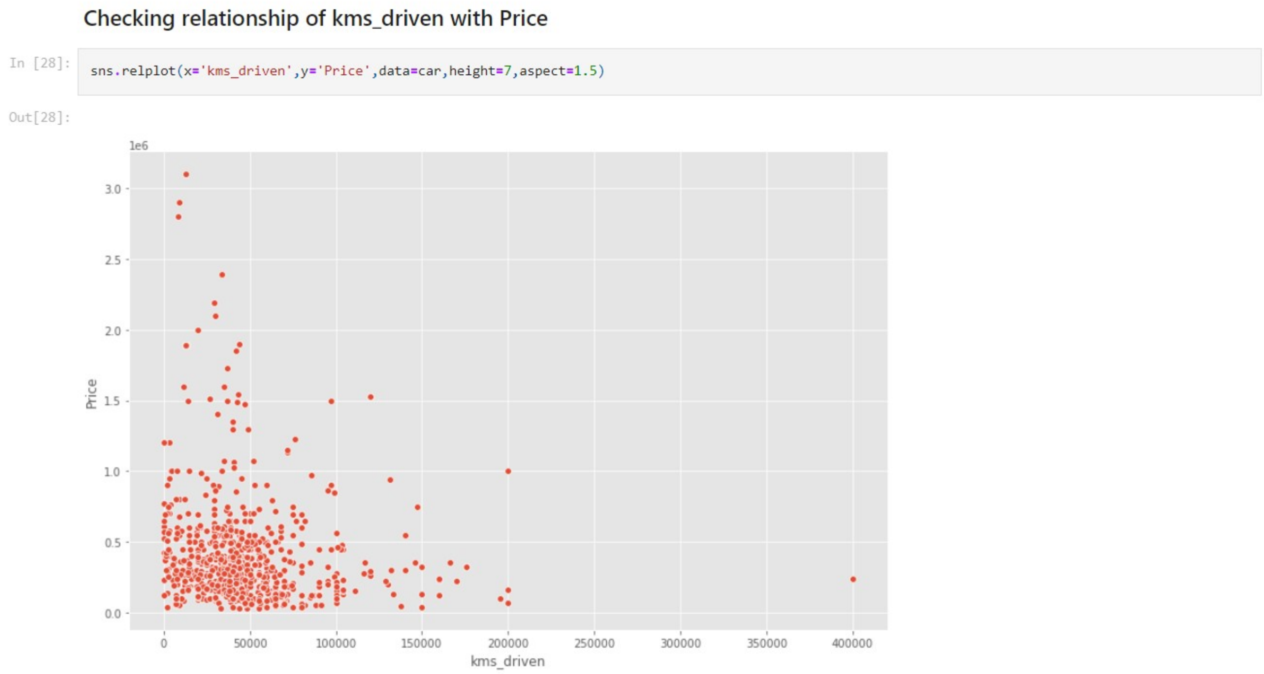


Checking relationship of Year with Price

In [27]:

```
plt.subplots(figsize=(20,10))
ax=sns.swarmplot(x='year',y='Price',data=car)
ax.set_xticklabels(ax.get_xticklabels(),rotation=40,ha='right')
plt.show()
```

3) Checking relationship of kms_driven with Price



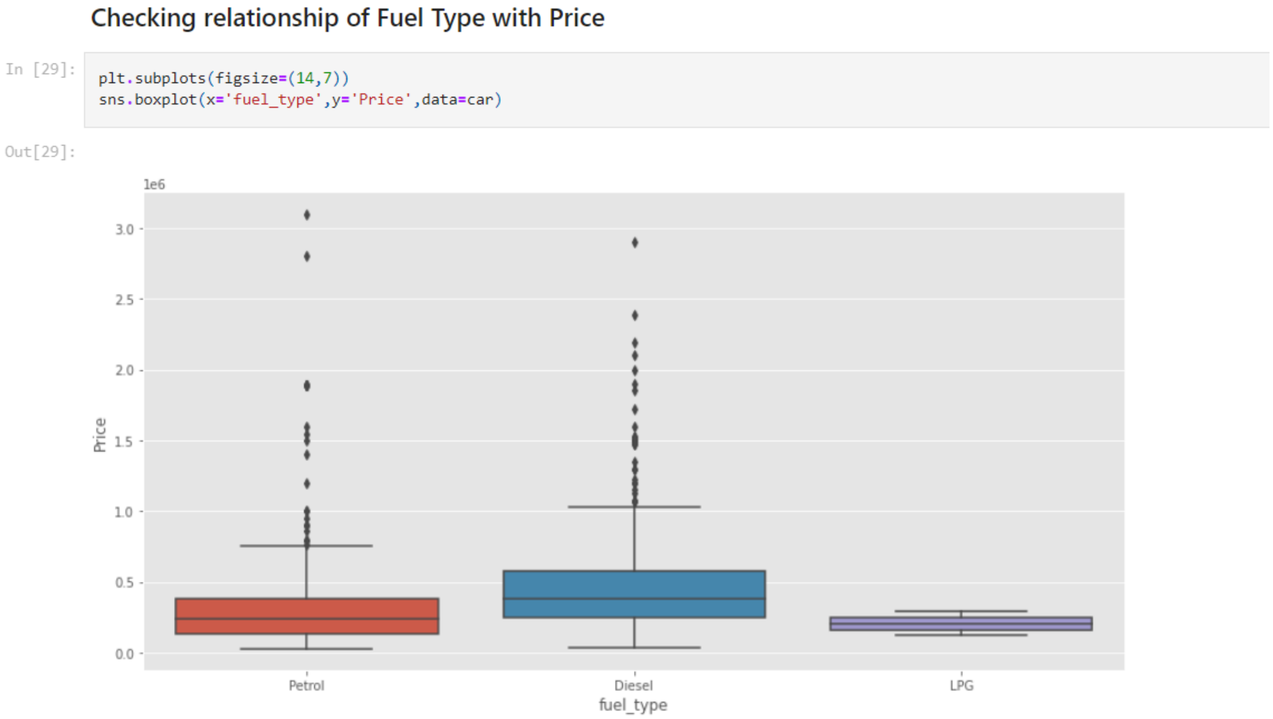
Checking relationship of kms_driven with Price

In [28]:

```
sns.relplot(x='kms_driven',y='Price',data=car,height=7,aspect=1.5)
```

Out[28]:

4) Checking relationship of Fuel type with Price



Checking relationship of Fuel Type with Price

In [29]:

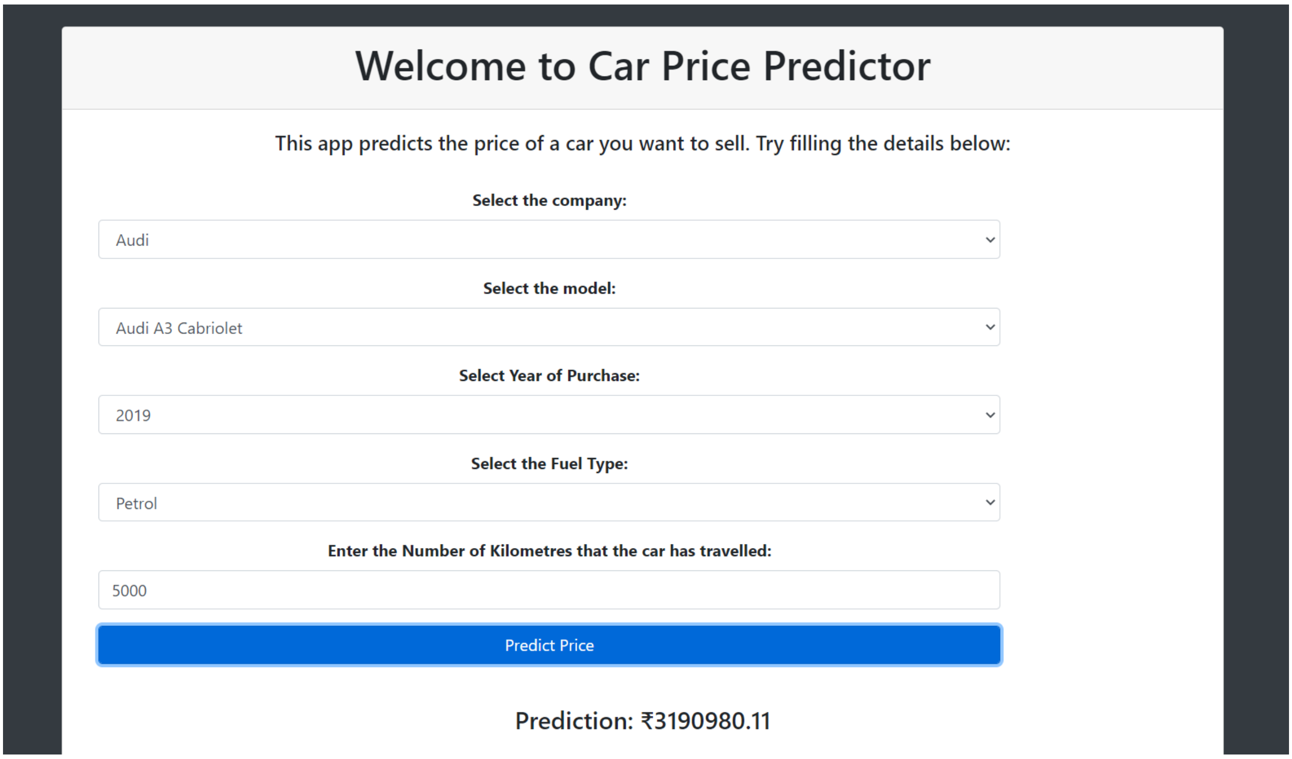
```
plt.subplots(figsize=(14,7))
sns.boxplot(x='fuel_type',y='Price',data=car)
```

Out[29]:

8. TESTING :

8.1 Test Cases : Various test cases are tested and their results are given below.

Case 1:



Welcome to Car Price Predictor

This app predicts the price of a car you want to sell. Try filling the details below:

Select the company: Audi

Select the model: Audi A3 Cabriolet

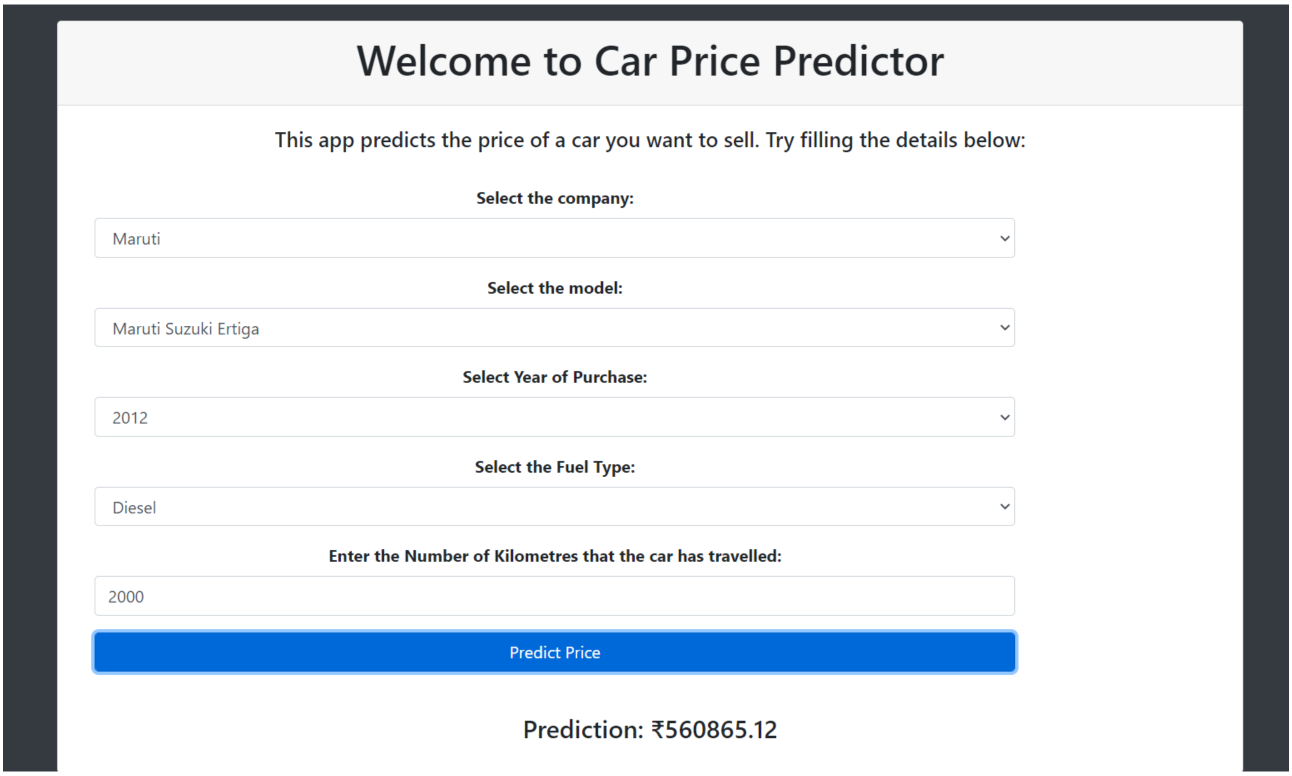
Select Year of Purchase: 2019

Select the Fuel Type: Petrol

Enter the Number of Kilometres that the car has travelled: 5000

Prediction: ₹3190980.11

Case 2:



Welcome to Car Price Predictor

This app predicts the price of a car you want to sell. Try filling the details below:

Select the company: Maruti

Select the model: Maruti Suzuki Ertiga

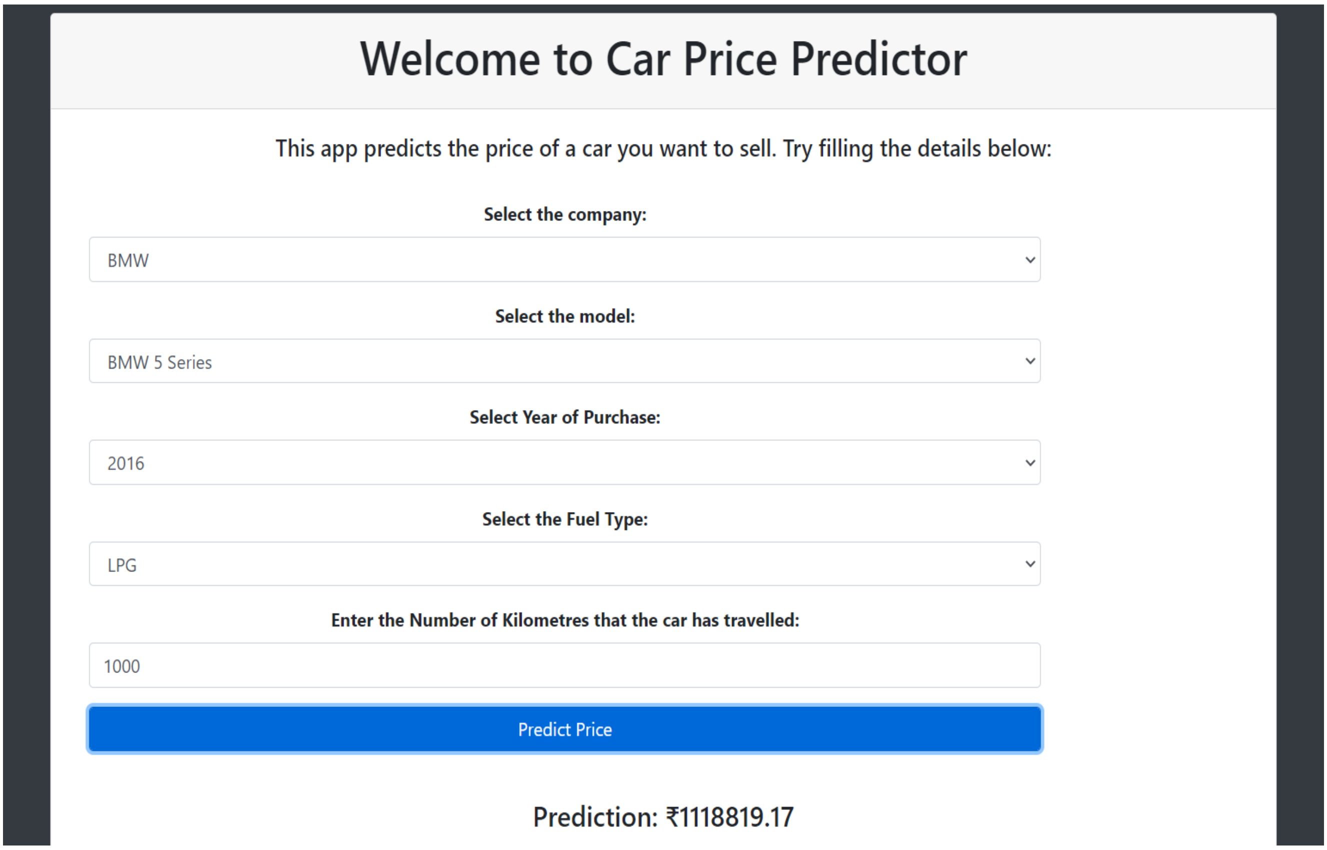
Select Year of Purchase: 2012

Select the Fuel Type: Diesel

Enter the Number of Kilometres that the car has travelled: 2000

Prediction: ₹560865.12

Case 3:



Welcome to Car Price Predictor

This app predicts the price of a car you want to sell. Try filling the details below:

Select the company: BMW

Select the model: BMW 5 Series

Select Year of Purchase: 2016

Select the Fuel Type: LPG

Enter the Number of Kilometres that the car has travelled: 1000

Prediction: ₹1118819.17

8.2 User Acceptance Testing

1) Verify all the UI elements in Home page rendered properly.

2) Verifiy the Data Entry page can be reachable.

3) Verify all the UI elements in Data Entry page rendered properly.

4) Verify user is able to enter all values.

5) Verifiy the Output Display page can be reachable.

6) Verify all the UI elements in Output Display page rendered properly.

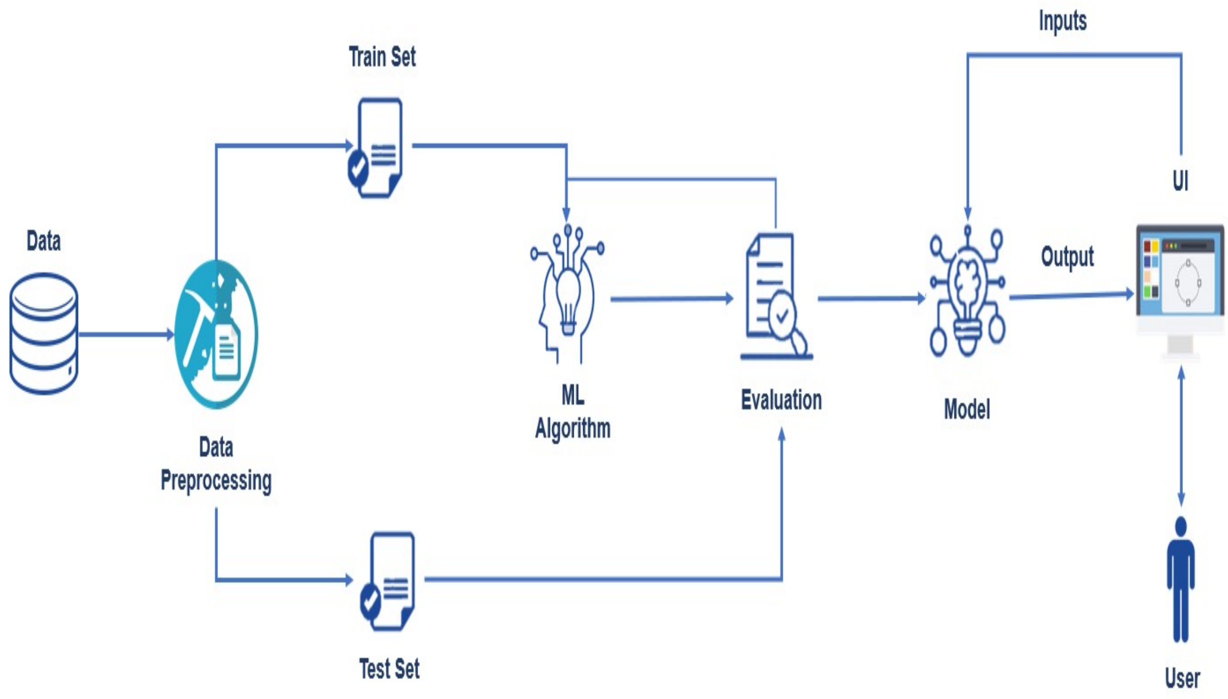
7) Verify user is able to get predicted result.

Here is the excel document:

https://github.com/IBM-EPBL/IBM-Project-30691-1660154083/blob/main/Final%20Deliverables/Model%20Building/Testcases%20Report.xlsx

9. RESULTS :

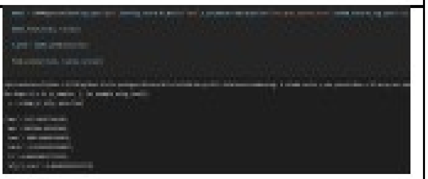
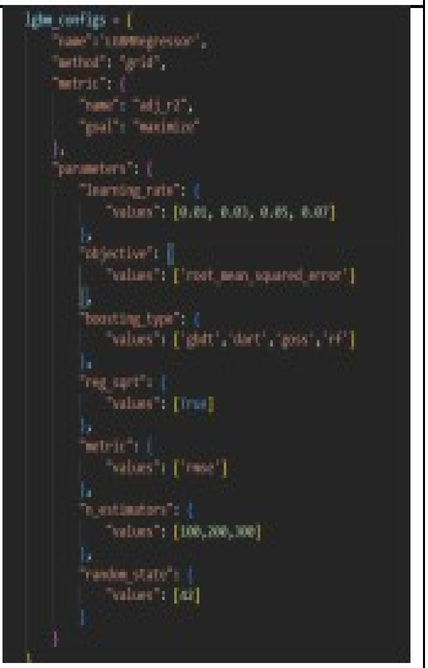
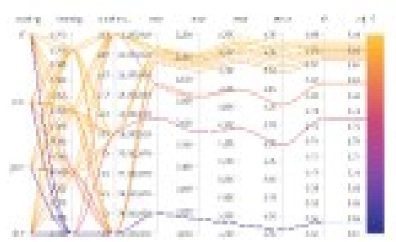
Performance Metrics



Model Performance Testing

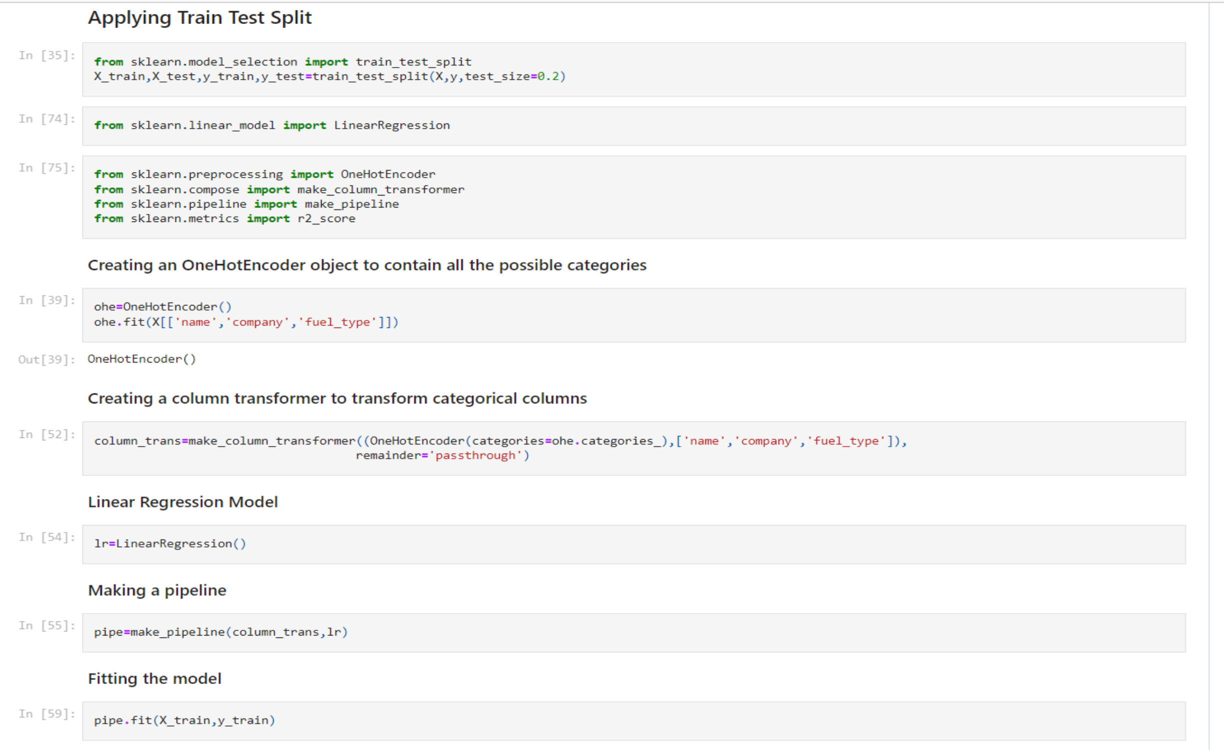
Model Performance Testing:

Project team shall fill the following information in model performance testing template.

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

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

By applying train test split



Applying Train Test Split

```
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 [74]: from sklearn.linear_model import LinearRegression
```

```
In [75]: from sklearn.preprocessing import OneHotEncoder
         from sklearn.compose import make_column_transformer
         from sklearn.pipeline import make_pipeline
         from sklearn.metrics import r2_score
```

Creating an OneHotEncoder object to contain all the possible categories

```
In [39]: ohe=OneHotEncoder()
         ohe.fit(X[['name','company','fuel_type']])

Out[39]: OneHotEncoder()
```

Creating a column transformer to transform categorical columns

```
In [52]: column_trans=make_column_transformer((OneHotEncoder(categories=ohe.categories_),['name','company','fuel_type']),
                                             remainder='passthrough')
```

Linear Regression Model

```
In [54]: lr=LinearRegression()
```

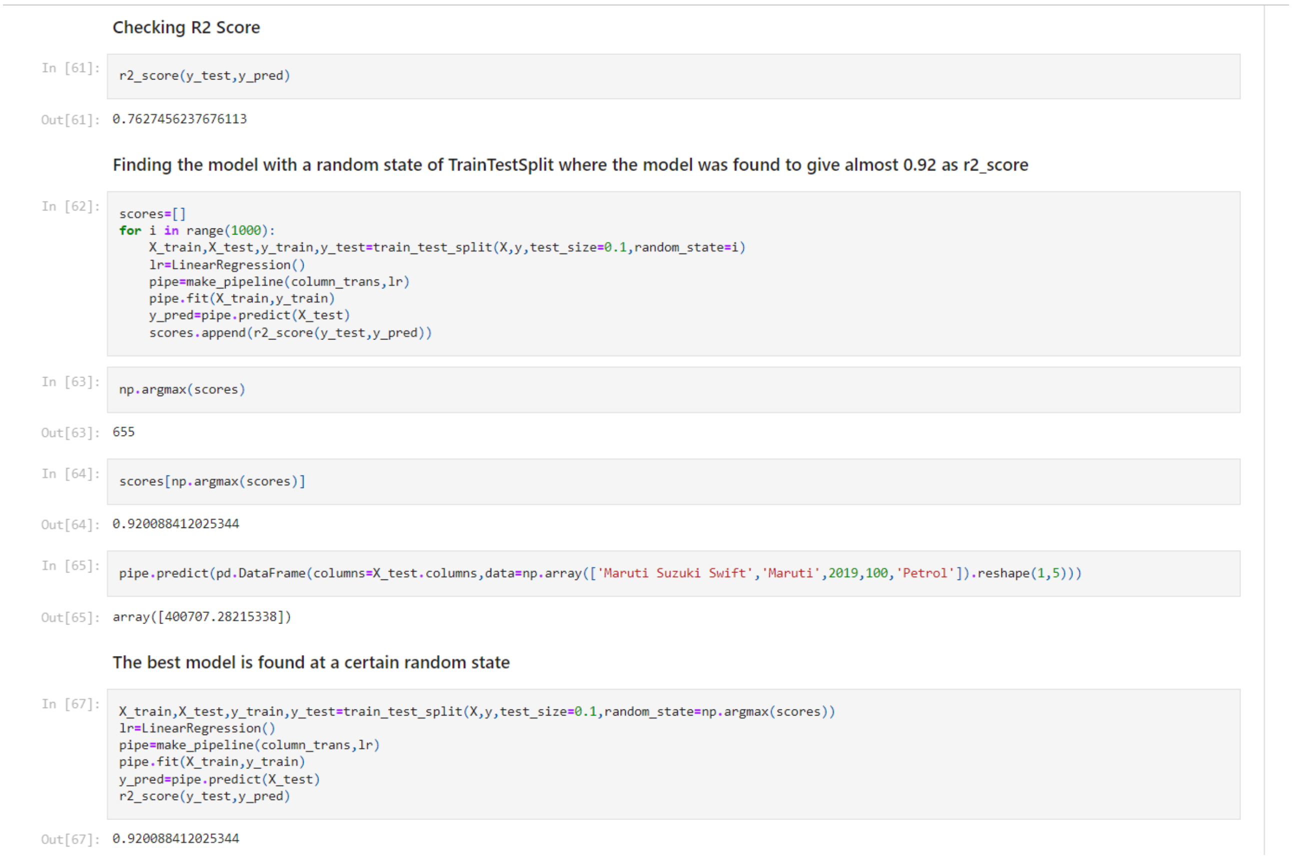
Making a pipeline

```
In [55]: pipe=make_pipeline(column_trans,lr)
```

Fitting the model

```
In [59]: pipe.fit(X_train,y_train)
```

Checking R2 Score



Checking R2 Score

```
In [61]: r2_score(y_test,y_pred)

Out[61]: 0.7627456237676113
```

Finding the model with a random state of TrainTestSplit where the model was found to give almost 0.92 as r2_score

```
In [62]: scores=[]
         for i in range(1000):
             X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
             lr=LinearRegression()
             pipe=make_pipeline(column_trans,lr)
             pipe.fit(X_train,y_train)
             y_pred=pipe.predict(X_test)
             scores.append(r2_score(y_test,y_pred))
```

```
In [63]: np.argmax(scores)

Out[63]: 655
```

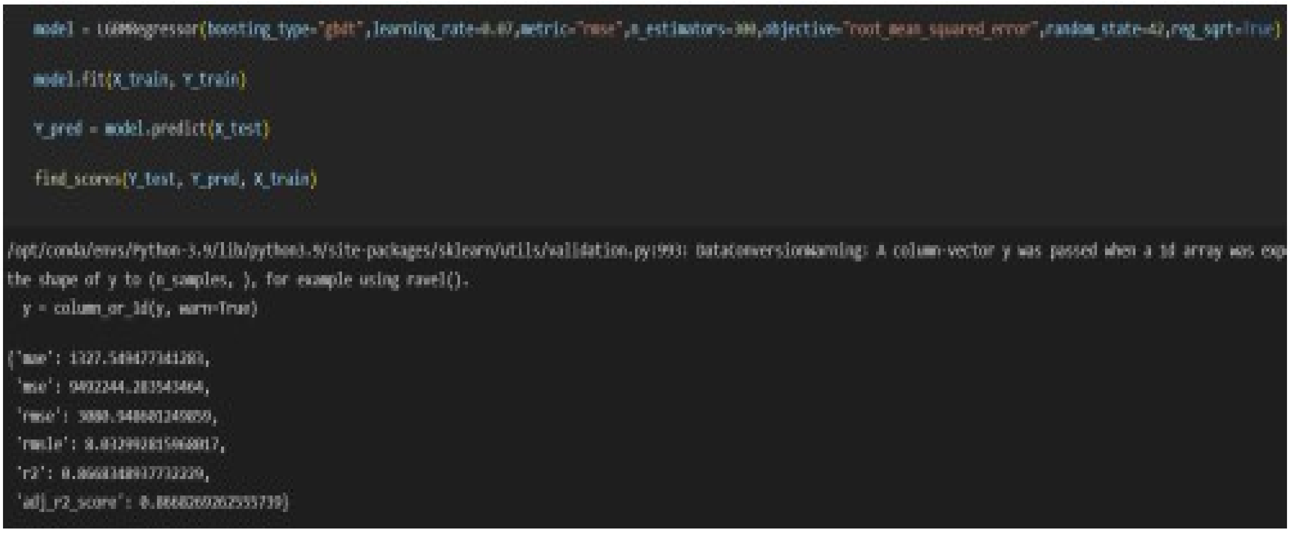
```
In [64]: scores[np.argmax(scores)]

Out[64]: 0.920088412025344
```

```
In [65]: pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))

Out[65]: array([400707.28215338])
```

The best model is found at a certain random state

```
In [67]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.argmax(scores))
         lr=LinearRegression()
         pipe=make_pipeline(column_trans,lr)
         pipe.fit(X_train,y_train)
         y_pred=pipe.predict(X_test)
         r2_score(y_test,y_pred)

Out[67]: 0.920088412025344
```

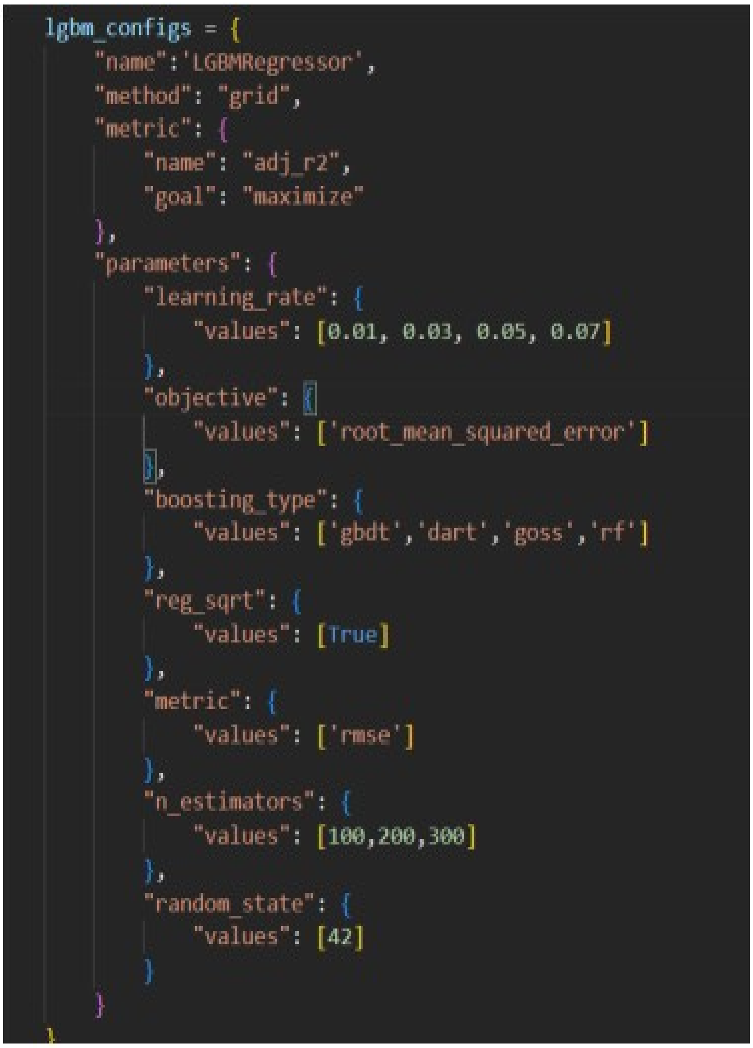
Metrics



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

model.fit(X_train, Y_train)

Y_pred = model.predict(X_test)

find_scores(Y_test, Y_pred, X_train)
```

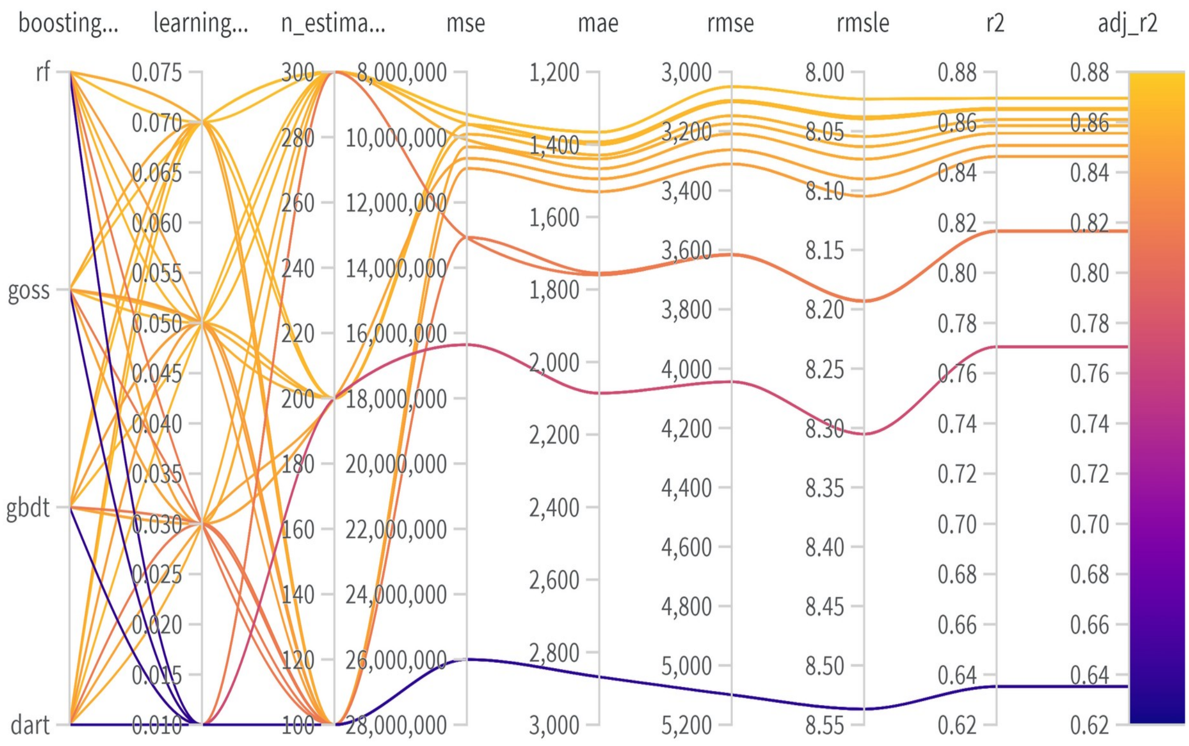
```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/sklearn/utils/validation.py:993: DataConversionWarning: A column-vector y was passed when a 1d array was exp
the shape of y to (n_samples, ), for example using ravel().
  y = column_or_1d(y, warn=True)

{'mae': 1327.5494773441283,
 'mse': 9492244.283543464,
 'rmse': 3080.940601249859,
 'rmsle': 0.032992815968017,
 'r2': 0.8668348937732229,
 'adj_r2_score': 0.8668269262555739}
```

Tuning the model

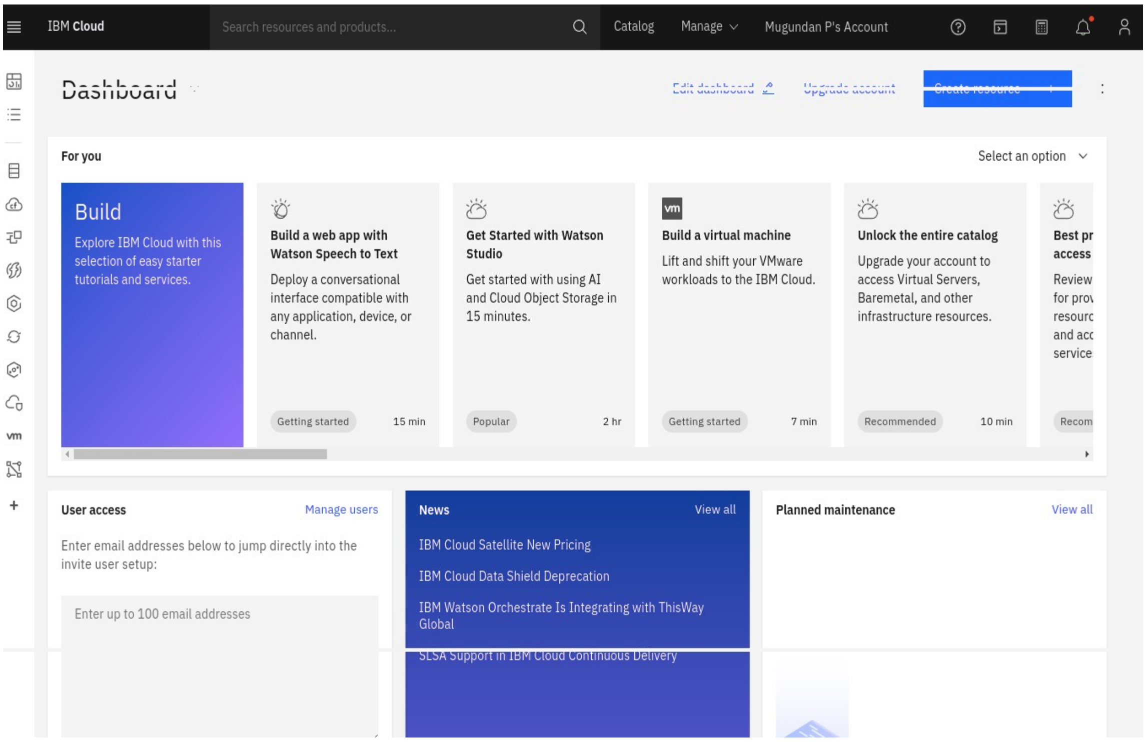


```
lgbm_configs = {
    "name":'LGBMRegressor',
    "method": "grid",
    "metric": {
        "name": "adj_r2",
        "goal": "maximize"
    },
    "parameters": {
        "learning_rate": {
            "values": [0.01, 0.03, 0.05, 0.07]
        },
        "objective": {
            "values": ['root_mean_squared_error']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

LGBM Regressor



Importing model on IBM :



For you

Select an option

Build

Explore IBM Cloud with this selection of easy starter tutorials and services.

Build a web app with Watson Speech to Text

Deploy a conversational interface compatible with any application, device, or channel.

Getting started 15 min

Get Started with Watson Studio

Get started with using AI and Cloud Object Storage in 15 minutes.

Popular 2 hr

Build a virtual machine

Lift and shift your VMware workloads to the IBM Cloud.

Getting started 7 min

Unlock the entire catalog

Upgrade your account to access Virtual Servers, Baremetal, and other infrastructure resources.

Recommended 10 min

Best pr access

Review for prov resourc and acc service

Recom

User access

Manage users

Enter email addresses below to jump directly into the invite user setup:

Enter up to 100 email addresses

News

View all

IBM Cloud Satellite New Pricing

IBM Cloud Data Shield Deprecation

IBM Watson Orchestrate Is Integrating with ThisWay Global

SLSA Support in IBM Cloud Continuous Delivery

Planned maintenance

View all

10. ADVANTAGES :

1) Good at learning complex and non- linear relationships.

2) Highly explainable and easy to interpret.

3) Robust to outliers.

4) No feature scaling is required.

5) Easy evaluation of the model.

6) Ability to build applications using Frameworks.

DISADVANTAGES:

1) Consumes more time.

2) Requires high computational power.

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. To get even more accurate models, we can also choose more advanced machine learning algorithms such as random forests, an ensemble learning algorithm which creates multiple decision/regression trees, which brings down overfitting massively or Boosting, which tries to bias the overall model by weighing in the favor of good performers. More data from newer websites and different countries can also be scraped and this data can be used to retrain these models to check for reproducibility.

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. We can build an android app as user interface for interacting with user. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data ratherthan the whole dataset.

13. APPENDIX :

Source Code:

Car.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Car Resale Value Predicting Application</title>
    <link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
    <link rel="stylesheet" href="../static/css/style.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
  </head>
  <body>
    <section class="header">
      <nav>
        <a href="/"><img src="../static/Images/sang.png" width="100" height="100"></a>
      </nav>
        <div class="text-box">
          <h1>Car resale value Predictor</h1>
          <p>Best system to predict the amount of resale value based on the parameters provided by the user .</p>
          <a href="./predict_page" class="visit-btn ">Check price</a>
        </div>
    </section>

  </body>
</html>
```

Styles.css

```
<!DOCTYPE html>

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



</body>
</html>
```

Prediction Purpose

```
<!DOCTYPE html>

<html lang="en">
<head>
   <meta charset="UTF-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <link rel="stylesheet" href="../static/css/predict.css">
   <title>Car Resale Value Predicting Application</title>
   <link rel="icon" type="image/x-icon" href="../static/Images/favicon.ico">
</head>
<body>
```

```
        <section class="header">
      <nav>
        <a href="/"><img src="../static/Images/sang.png" width="100"
height="100"></a>
      </nav>
        <div class="text-box">
          <h1>The Predicted Car Resale Value is </h1>
                  <h1>{{predict}}</h1>
        </div>
    </section>


</body>
</html>
```

Application

```
from flask import Flask,render_template,request,redirect

from flask_cors import CORS,cross_origin
import pickle
import pandas as pd
import numpy as np

app=Flask(__name__)
cors=CORS(app)
model=pickle.load(open('LinearRegressionModel.pkl','rb'))
car=pd.read_csv('Cleaned_Car_data.csv')

@app.route('/',methods=['GET','POST'])
def index():
    companies=sorted(car['company'].unique())
    car_models=sorted(car['name'].unique())
    year=sorted(car['year'].unique(),reverse=True)
    fuel_type=car['fuel_type'].unique()

    companies.insert(0,'Select Company')
    return render_template('index.html',companies=companies,
car_models=car_models, years=year,fuel_types=fuel_type)


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

```
@cross_origin()
def predict():

    company=request.form.get('company')

    car_model=request.form.get('car_models')
    year=request.form.get('year')
    fuel_type=request.form.get('fuel_type')
    driven=request.form.get('kilo_driven')

    prediction=model.predict(pd.DataFrame(columns=['name', 'company', 'year', 'kms_driven', 'fuel_type'],

data=np.array([car_model,company,year,driven,fuel_type]).reshape(1, 5)))
    print(prediction)

    return str(np.round(prediction[0],2))



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

Github Link :

https://github.com/IBM-EPBL/IBM-Project-30691-1660154083

Project Demo Link :

https://drive.google.com/file/d/1PW-_yoxcoxOxPyQTIhq-Sr-RAzNxtEqO/view?usp=share_link