Car Resale Value Prediction

Project report

Team Id : PNT2022TMID54519

Team Members:

- **Charles.M**
- **Angel Sneha.A**
- **Anisha.S**
- **Vignesh.V**
- **Bala Suthakar.N**

INTRODUCTION:

Project Overview:

In this project we have used different algorithms with different techniques for developing Car resale value prediction systems considering different features of the car. In a nutshell, car resale value prediction helps the user to predict the resale value of the car depending upon various features like kilometres driven, fuel type, etc.

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.

LITERATURE SURVEY

Existing problem:

"What will you give me for my car?" That's an important question to determine the used car trade-in value, which often serves as a down payment on a new car. Getting more for your trade-in can mean you'll have less to finance, which can save you money over the length of your loan.

Here are a few factors that go into determining used car trade-in value.

The age of your trade-in:
With some exceptions, cars tend to lose their value rapidly. They can drop by 20-30% in the first year and 60% or more of the original price after five years. The curve flattens out from there, but until and unless your car becomes a collectible or a classic, the value continues to go down. In many cases, the demand for your vehicle on the used car market also decreases. Newer trade-ins typically bring higher prices and bigger profits for the dealer.

The mileage on your trade-in:
Low mileage can tip the balance in favor of an older trade-in. According to the U.S. Department of Transportation, the average miles driven in the United States is roughly 13,500 a year. Anything under the average is a selling point for the dealer looking to make a profit reselling your trade. A 10 year-old car with 67,500 miles on it (half the national average) could be more attractive to potential buyers than one with 135,000 miles or more.

The overall condition of your trad”-in:
Regardless of age or mileage, condition plays a huge part in used car trade-in value. Dealers may not look at scratches, dents, stained interiors, cracked windshields, and mechanical issues and think "stuff happens." They look at what it will cost them to repair or replace the things necessary for the car to sell at a strong price. Additionally, they'll deduct that from what otherwise could have been a decent offer for your trade-in.

The history of your car's maintenance:
Even if your car is in excellent condition, one way you can improve your chances for the highest trade-in value is to have it maintained according to the manufacturer's schedule in your owners' manual and keep the paperwork from every one of those visits. Bring those with you at trade-in time. Being able to document that you've provided the tender loving care your car deserves does count.

In addition, some dealers love to see that all that maintenance was done in their service department. It's one more thing they can use to sell a prospective buyer on your trade, and it establishes you as a potentially loyal customer in the future.

The equipment in your car:
The standard and optional equipment that came with your car can increase the money you’re likely to get for it in trade. The most valuable is an automatic transmission. The number of people who can drive a stick shift has been declining for years. So, if your car has one, it could be worth hundreds of dollars less to a dealer than the same car with an automatic. Sure, there might be

someone who’d love to have that car with a manual transmission, but there are simply more potential customers for automatics, improving the dealer’s odds of selling your trade for a solid profit.

Other pluses at trade-in time are upgraded factory audio systems, automatic climate control, power seats, upgraded upholstery, and sunroofs. In colder climates, all-wheel drive, heated seats, and steering wheels can also .

Current demand dictates the deal:

Beyond equipment, there are other factors that can vary. An attractive, somewhat neutral paint color can appeal to a larger number of used car customers. A model that sold poorly or a make that's gone out of business could be at a disadvantage. Likewise, demand for convertibles is likely to be much stronger in the Sunbelt than the Snowbelt. Also, cars with good gas mileage can be worth more in regions with higher fuel prices.

Make a good impression at trade-in time:

Before you head down to the dealership, clean the car. At minimum, vacuum it out, remove your personal items and run it through a car wash. A full detailing of the exterior, interior and engine bay might be an even better idea. It conveys a well-cared for car and could easily pay for itself in trade-in value.

Ultimately, getting the best used car trade-in value is a process that should begin when you buy that car. Look for one with historically strong resale value in a popular color with popular equipment. Keep the mileage below average if you can. Stick to the manufacturer's recommended maintenance schedules, keep the receipts, and fix anything that breaks right away. Finally, monitor your car's value. Wait for the right moment when you have enough equity in your old car to make a difference in the negotiation for your new one.

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.

Problem Statement Definition:

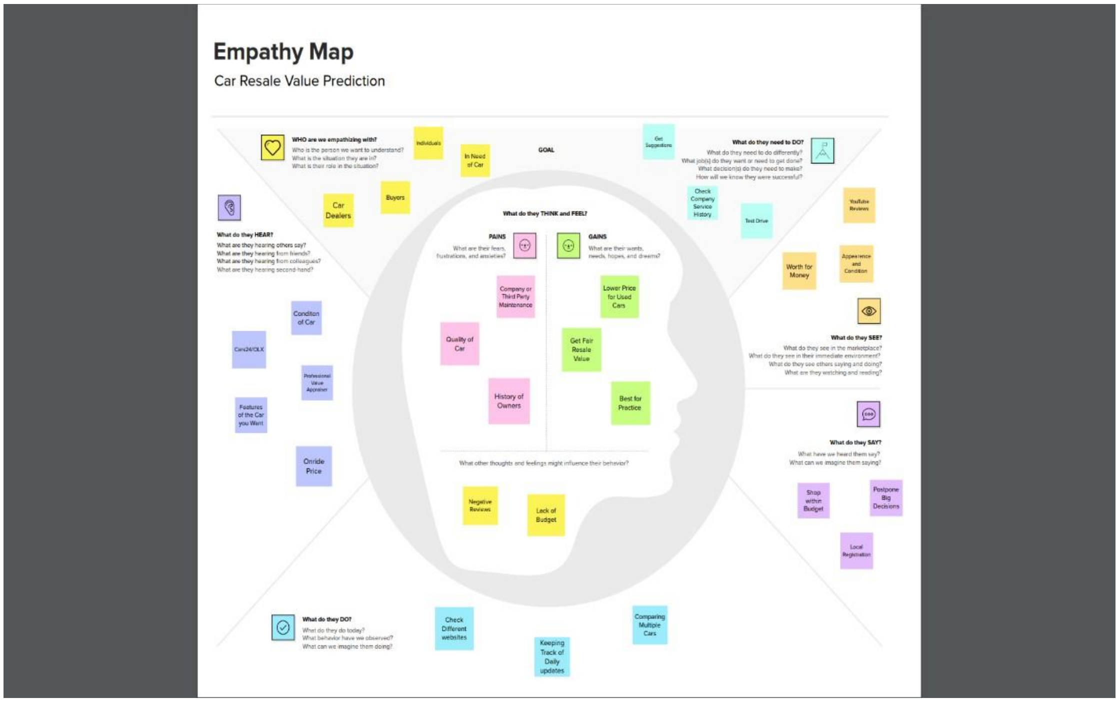
The main aim of this project is to predict the price of used cars using the various Machine Learning (ML) models. This can enable the customers to make decisions based on different inputs or factors namely

- Brand or Type of the car one prefers like Ford, Hyundai
- Model of the car namely Ford Figo, Hyundai Creta
- Year of manufacturing like 2020, 2021
- Type of fuel namely Petrol, Diesel

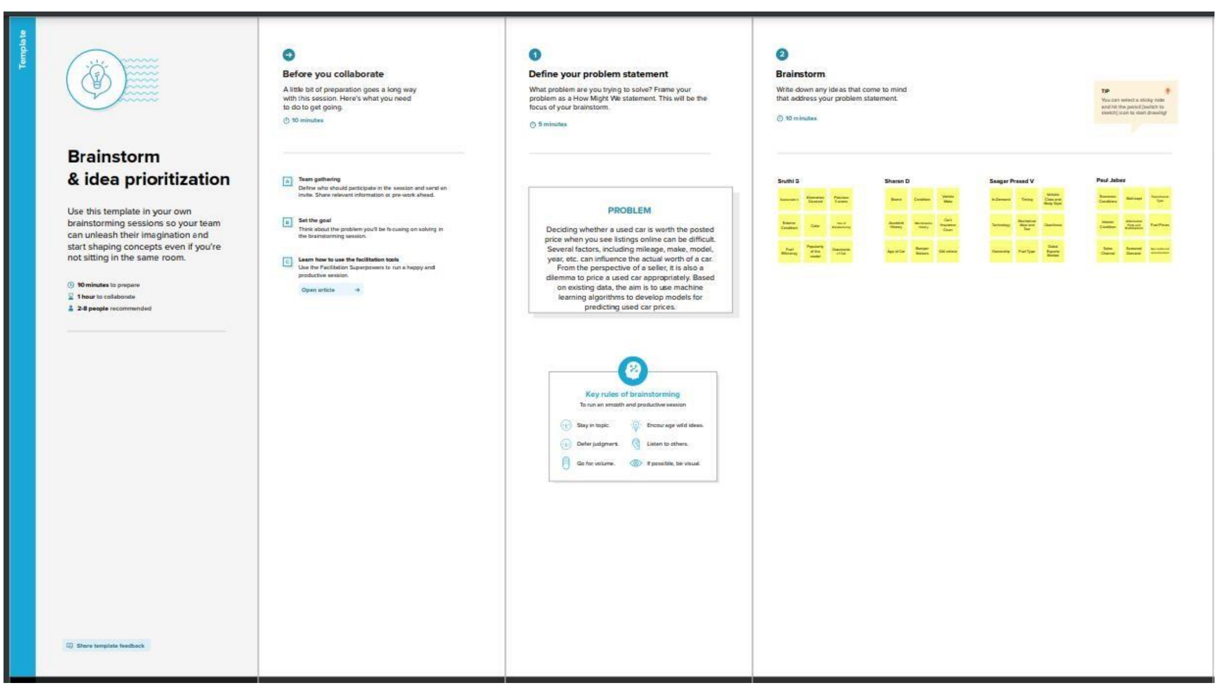
to name a few characteristic features required by the customer. The project Car Price Prediction deals with providing the solution to these problems. Through this project, we will get to know which of the factors are significant and tell us how they affect the car’s worth in the market

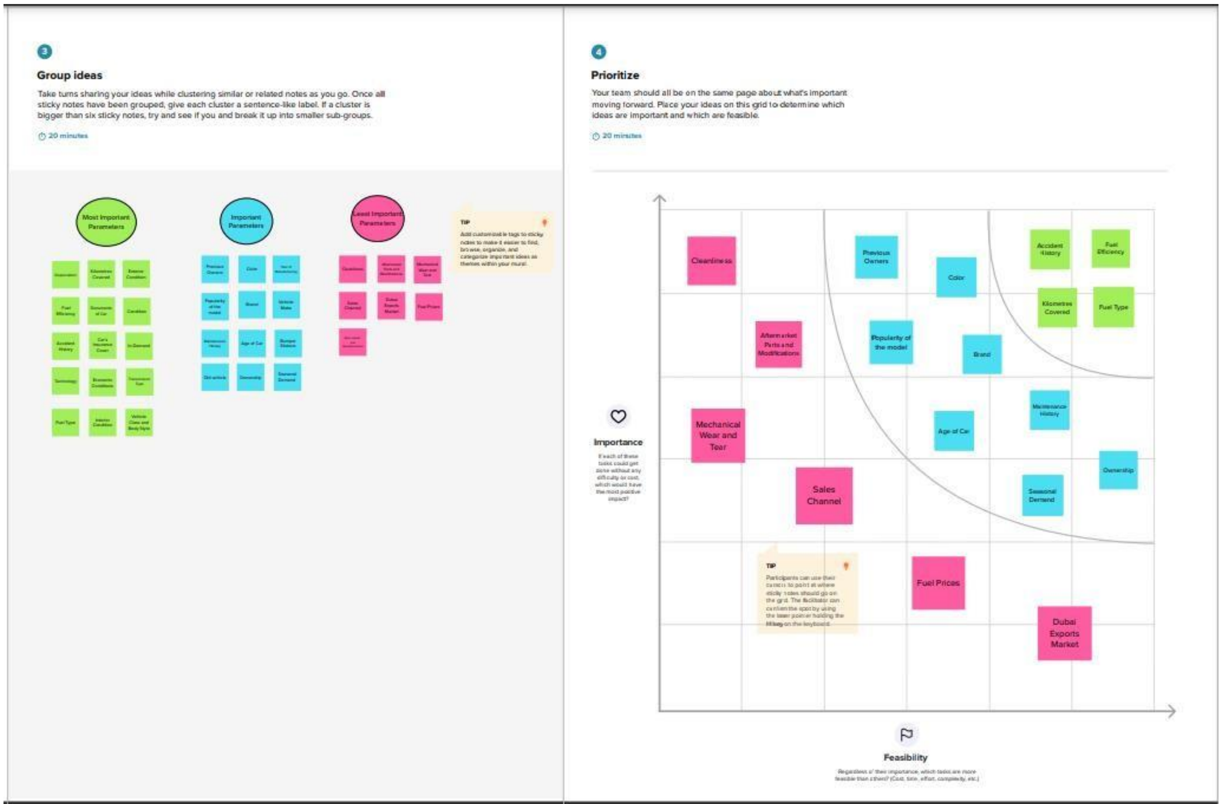
IDEATION & PROPOSED SOLUTION

Empathy Map Canvas:

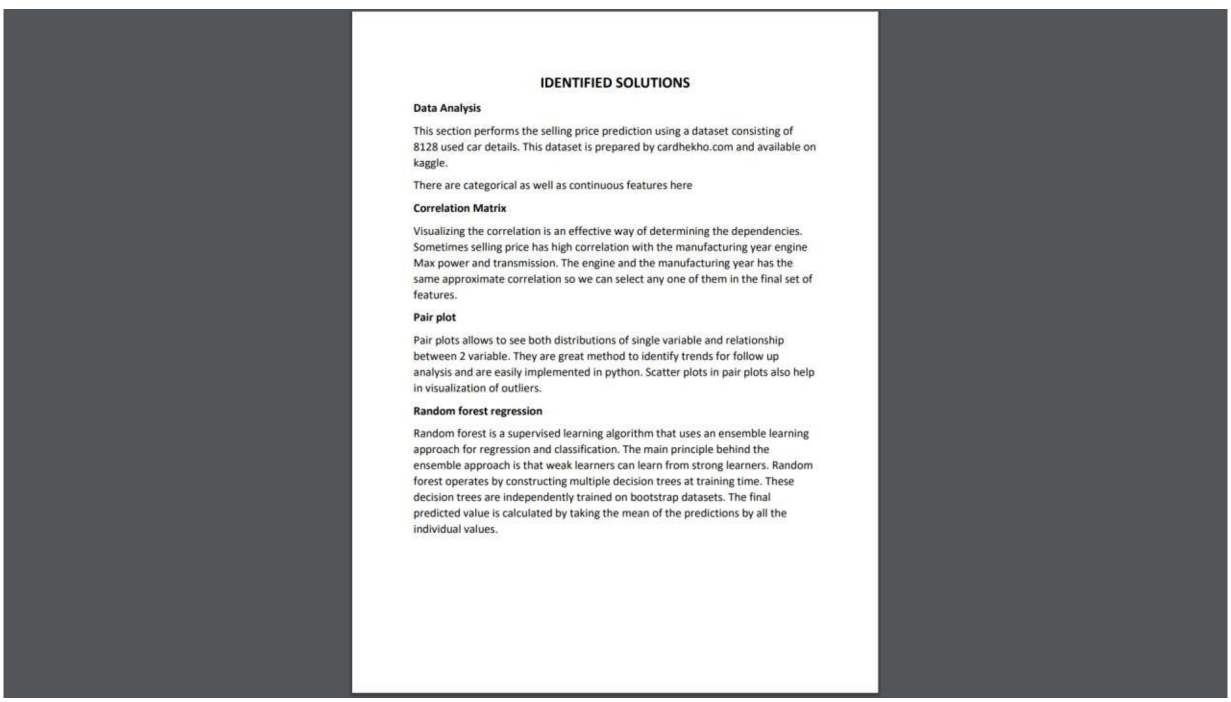


Ideation & Brainstorming:





Proposed Solution:



IDENTIFIED SOLUTIONS

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

Visualizing 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.

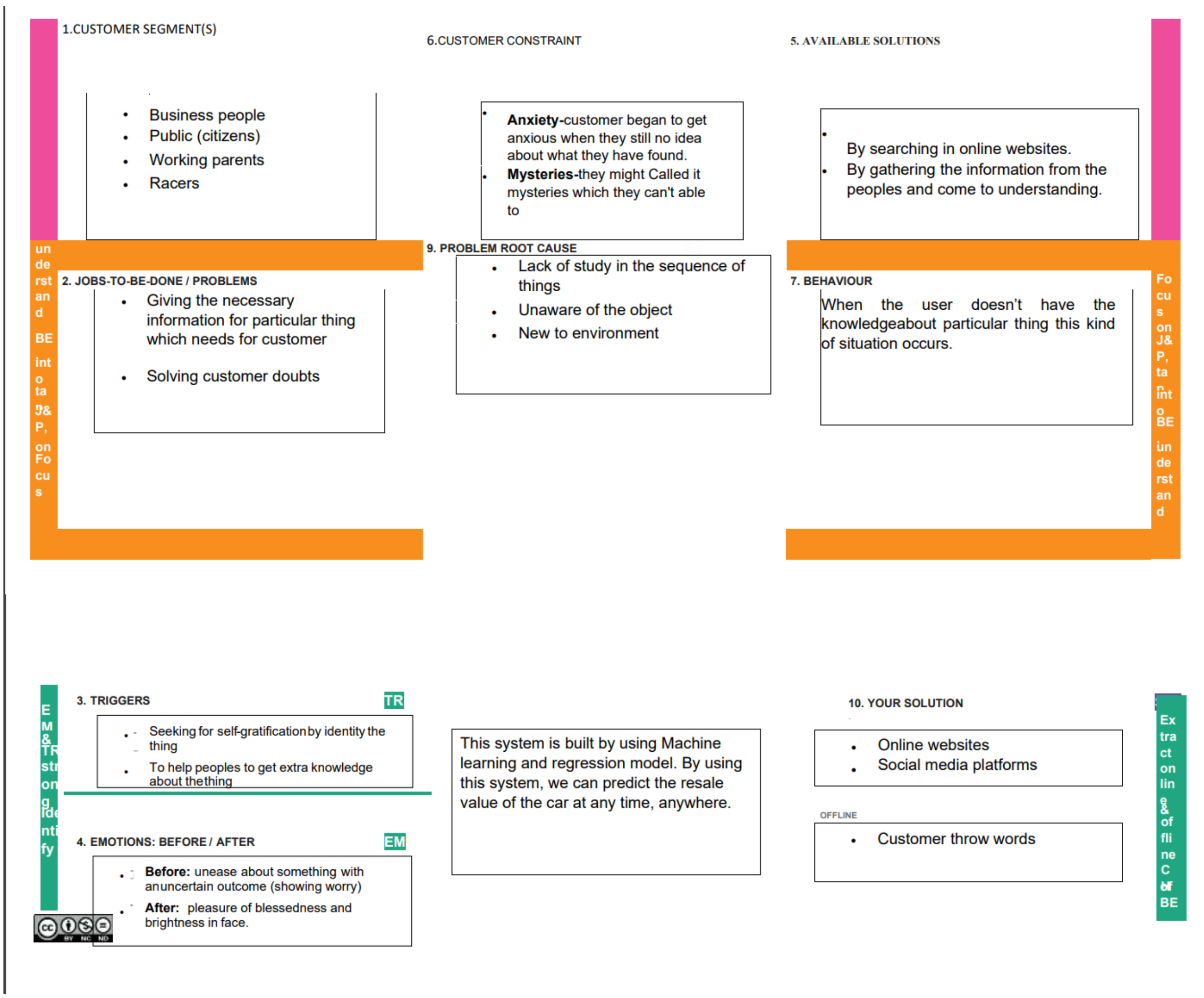
Pair plot

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 visualization 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.

Problem Solution fit:



REQUIREMENT ANALYSIS:

Functional requirement:

Following are the functional requirements of the proposed solution.

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

Non-Functional requirements:

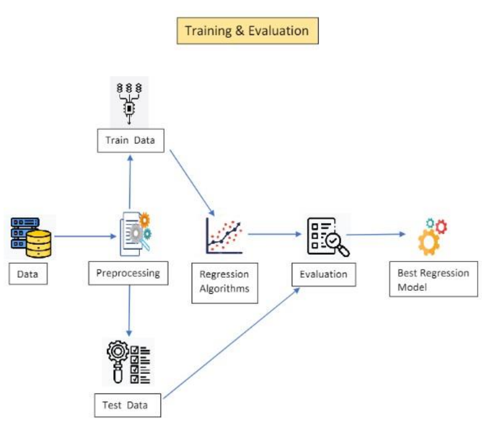
Following are the non-functional requirements of the proposed solution.

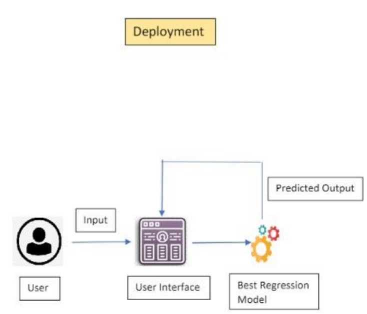
NFR No.	Non-Functional Requirement	Description
NFR-1	Usability	Predicting the resale value
NFR-2	Security	Providing security to the website
NFR-3	Reliability	Providing high reliability by predicting values for different types of cars
NFR-4	Performance	Providing high performance by using some machine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

PROJECT DESIGN:

Data Flow Diagrams:

Solution & Technical Architecture:





User Stories:

PROJECT PLANNING & SCHEDULING:

Sprint Planning & Estimation:

Project Name			Car Resale Value Prediction			
Sprint	**Functional Requirement (Epic)**	**User Story Number**	**User Story / Task**	**Story Points**	**Priority**	**Team Members**
Sprint-1	Pre-process data	USN-1	Collect Dataset	1	Low	Sruthi G
Sprint-1		USN-2	Import required libraries	1	Low	Sruthi G
Sprint-1		USN-3	Read and clean data sets	2	Low	Sharan D, Saagar Prasad V
Sprint-2	Model building	USN-1	Split data into independent and dependent variables	3	Medium	Paul Jabez
Sprint-2		USN-2	Apply using regression model	3	Medium	Sruthi G, Sharan D, Saagar Prasad V, Paul Jabez
Sprint-3	Application building	USN-1	Build python flask application and HTML page	5	High	Sruthi G, Sharan D,Saagar Prasad V
Sprint-3		USN-2	Execute and test	5	High	Paul Jabez
Sprint-4	Training the model	USN-1	Train machine learning model	5	High	Sruthi G, Sharan D
Sprint-4		USN-2	Integrate flask	5	High	Saagar Prasad V, Paul Jabez

Sprint Delivery Schedule:

Product Backlog, Sprint Schedule, and Estimation (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

CODING & SOLUTIONING

Feature 1:

Autonomous Cars: Companies like Tesla has already implemented the self-driving feature in their cars. Tesla relies on its complex Computer Vision Algorithms and Sensors to gain full control over roads. Machine Learning allows self-driving cars to adapt to changing road conditions instantaneously.

Feature 2:

Real-Time Car Parking: There is a tremendous increase in the number of vehicles, and with the increasing number of cars, parking problems arise. Smart Parking Systems has been addressing such problems by leveraging the power of Machine Learning and loT devices. The system minimizes human intervention and saves time, money, and energy.

TESTING

Test Cases:

• Missing values

The trained ML model requires 4 feature inputs for predicting the output. Failing which, the model throws invalid Input error. All the fields in the html form have been marked required using CSS and thus user must input all fields.

Output: User must input all the fields, failing which, form shows warning message "this field needs to be filled". Thus, there can be no errors in model prediction.

• Invalid Input

The trained ML model requires only numerical input for all 4 features. Thus, if user uses symbols such as comma while input, model may throw error. To overcome the same, preprocessing script is deployed in backend which removes all unwanted characters like comma, whitespaces etc. so that model gets required input.

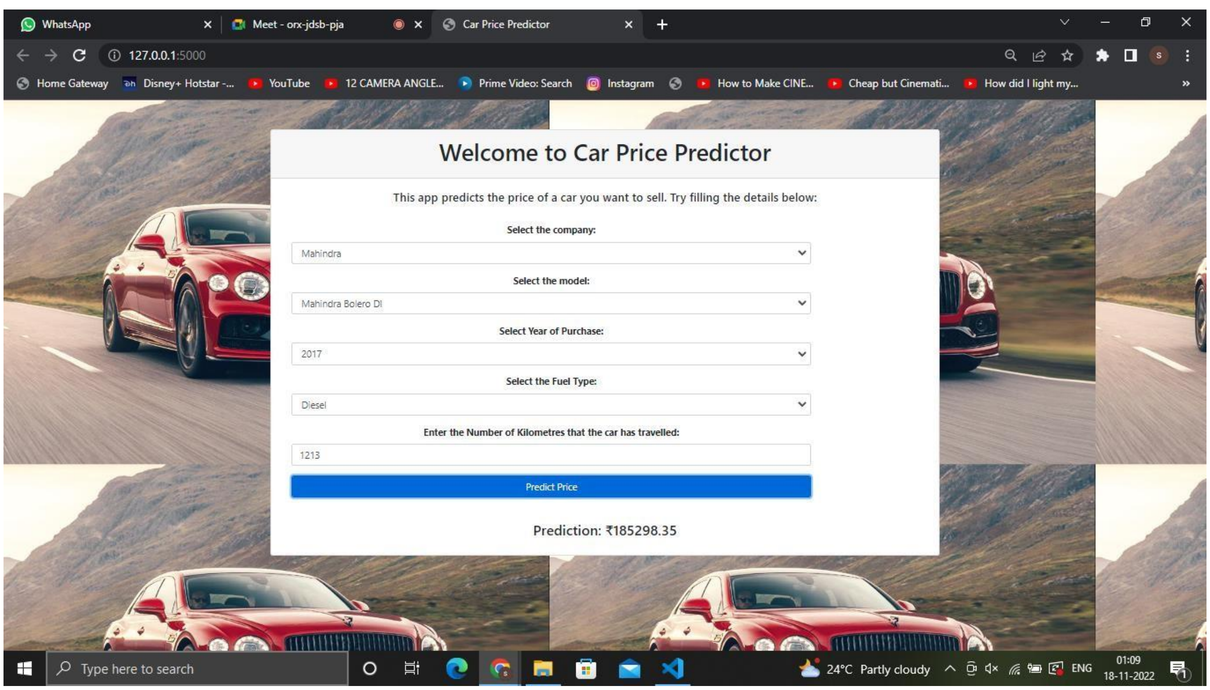
Output: Due to python preprocessing script, model will get the desired input and thus will give accurate prediction.

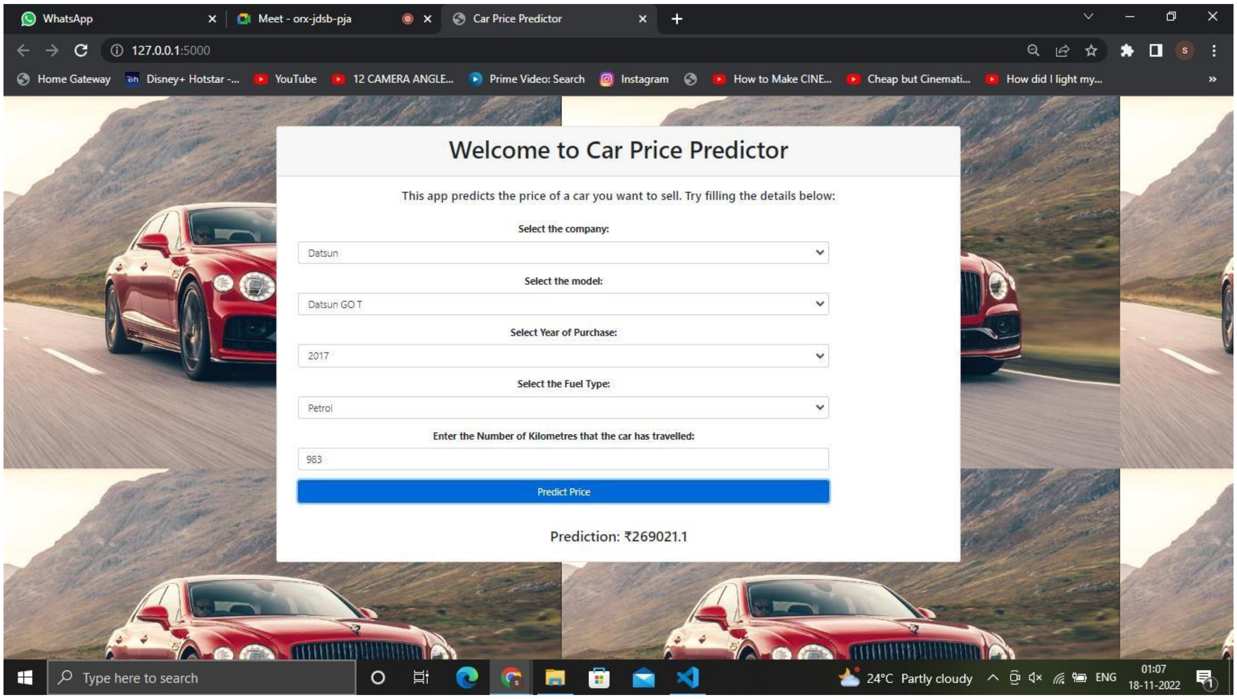
• Unseen year of purchase

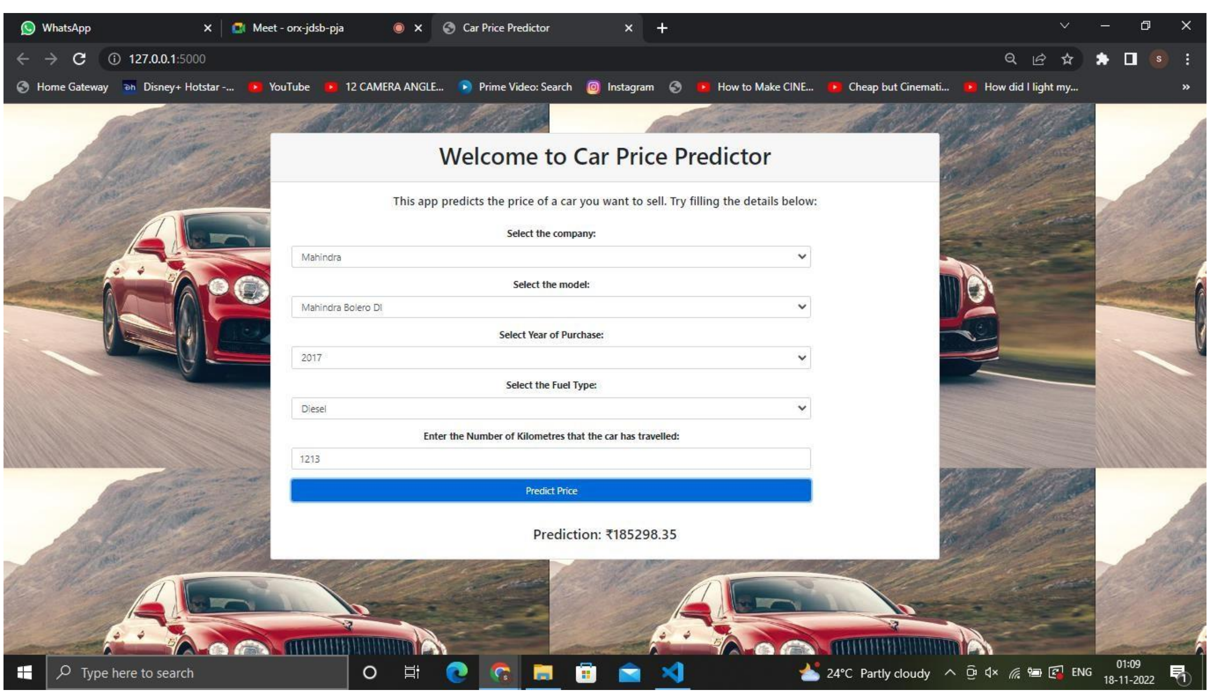
The model is trained with data from cars purchased since 2011 to 2020. If the user inputs details of car purchased after that i.e., 2021, model may get confused since that data is quite new and unseen to model.

Output: Model has been trained with boosting algorithm and thus it gives quite accurate results with around RMSE 65,000 INR.

User Acceptance Testing:




WhatsApp
Meet - orx-jdsb-pja
Car Price Predictor
127.0.0.1:5000
Home Gateway
Disney+ Hotstar -...
YouTube
12 CAMERA ANGLE...
Prime Video: Search
Instagram
How to Make CINE...
Cheap but Cinemati...
How did I light my...
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:
Datsun
Select the model:
Datsun GO T
Select Year of Purchase:
2017
Select the Fuel Type:
Petrol
Enter the Number of Kilometres that the car has travelled:
983
Predict Price
Prediction: ₹269021.1
Type here to search
24°C Partly cloudy
ENG
01:07
18-11-2022


WhatsApp
Meet - orx-jdsb-pja
Car Price Predictor
127.0.0.1:5000
Home Gateway
Disney+ Hotstar -...
YouTube
12 CAMERA ANGLE...
Prime Video: Search
Instagram
How to Make CINE...
Cheap but Cinemati...
How did I light my...
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:
Mahindra
Select the model:
Mahindra Bolero DI
Select Year of Purchase:
2017
Select the Fuel Type:
Diesel
Enter the Number of Kilometres that the car has travelled:
1213
Predict Price
Prediction: ₹185298.35
Type here to search
24°C Partly cloudy
ENG
01:09
18-11-2022

RESULTS

Performance Metrics:

The r2 score varies between 0 and 100%. It is closely related to the MSE ,but not the same. Wikipedia defines r2 as the proportion of the variance in the dependent variable that is predictable from the independent variable(s).”

```
[43] 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))

[44] np.argmax(scores)

     655

[45] scores[np.argmax(scores)]

     0.920087093218515

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

     array([400642.51767152])
```

ADVANTAGES & DISADVANTAGES

Advantages:

- Linear regression performs exceptionally well for linearly separable data

- Easier to implement, interpret and efficient to train
- It handles overfitting pretty well using dimensionally reduction techniques, regularization, and cross-validation
- One more advantage is the extrapolation beyond a specific data set

Disadvantages:

- The assumption of linearity between dependent and independent variables.
- It is often quite prone to noise and overfitting
- Linear regression is quite sensitive to outliers
- It is prone to multicollinearity

CONCLUSION

In recent years, online used car trading platforms have developed rapidly, but they still face many problems. In practice, institutions and individuals differ in how they screen the characteristic variables of used car prices and predict used car prices. Under such conditions, it is easy to lead to the unsound development of the market, and it is difficult to establish a unified evaluation system, which causes great difficulties in the transaction of used cars. In terms of theory, traditional used car price evaluation methods rely too much on the subjective judgment of evaluators, which can no longer meet the needs of online transactions in the used car market. Therefore, it is necessary to establish an efficient, reasonable, fair, and accurate used car price evaluation system.

This paper analyzes the factors affecting the price of used cars from three aspects—used car parameters, vehicle condition factors, and transaction factors—and establishes a used car price evaluation system including 12 characteristic variables. Using web crawler technology to obtain used car transaction data, three prediction models of BPNN, GRABPNN, and PSO-GRA-BPNN were constructed to conduct comparative verification and result analysis. In a rough comparison, the BPNN model has lower accuracy, with an error range of about 19.979%, and it is unstable. In the case of feature variable screening, the prediction accuracy of the GRA-BPNN model is higher than that of the BPNN, and the error range is about 13.986%. However, the constructed PSO-GRA-BPNN used car price prediction model not only has high accuracy but also has an error range of about 7.978%, which has good scalability. Although the PSO-GRA-BPNN used car price prediction model has high prediction accuracy, it has lost time. It is mainly analyzed from two aspects. First, when selecting the hidden layers of the BP neural network, accuracy is given priority and time is ignored. The second is to select the iteration number and population size of the particle swarm optimization algorithm. When the iteration number and population size are larger, the accuracy of the model is higher, but with the continuous increase of the value, the improved accuracy is very small, almost stable. In addition, in view of the fact that the prediction accuracy of high-end used cars is lower than that of low-end used cars, it is suggested that when pricing high-end used cars, you need to check other configuration information in order to make a more reasonable judgment.

FUTURE SCOPE

Currently, system can only deal with Swift Dzeren cars due to lack of data. Also, data has been collected of only 5 cities of India. This can be extended to multiple car models and cities to improve accuracy and usability. Efficient use of deep learning such as LSTM (Long short-term memory) or RNN (Recurrent Neural networks) can be implemented once enough data is collected. This can improve accuracy and decrease RMSE drastically. Currently, only few features are used to predict resale value of the car. This can be extended to more features. One can also implement CNN to determine physical condition of the car from images like identifying dents, scratches etc. and thus predicting more relevant resale value of a car**.**

APPENDIX

Source Code:

index.html:-

```
<!DOCTYPE html>
<html lang="en">
<head xmlns="http://www.w3.org/1999/xhtml">
   <meta charset="UTF-8">
   <title>Car Price Predictor</title>
   <link rel="stylesheet" href="static/css/style.css">
   <link rel="stylesheet" type="text/css"
      href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.11.2/css/all.css">
      <script
    src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```html
    <script
  src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.16.0/umd/popper.min.js"></script>

    <script
  src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"

      integrity="sha384-
  Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"

      crossorigin="anonymous"></script>


  <!-- Bootstrap CSS -->

  <link rel="stylesheet"
  href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"

    integrity="sha384-
  9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7
  Sk" crossorigin="anonymous">

  <script
  src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@2.0.0/dist/tf.min.js"></script>


</head>

<body class="bg-dark">

<center>

<div class="container">

  <div class="row">

    <div class="card mt-50" style="width: 100%; height: 100%">

      <div class="card-header" style="text-align: center">

        <h1>Welcome to Car Price Predictor</h1>

      </div>

      <div class="card-body">

        <div class="col-12" style="text-align: center">

          <h5>This app predicts the price of a car you want to sell. Try filling the
  details below: </h5>

        </div>

        <br>

        <form method="post" accept-charset="utf-8" name="Modelform">

          <div class="col-md-10 form-group" style="text-align: center">

            <label><b>Select the company:</b> </label><br>
```

```
<select class="selectpicker form-control" id="company" name="company"
required="1"
        onchange="load_car_models(this.id,'car_models')">
    {% for company in companies %}
    <option value="{{ company }}">{{ company }}</option>
    {% endfor %}
</select>
</div>
<div class="col-md-10 form-group" style="text-align: center">
<label><b>Select the model:</b> </label><br>
<select class="selectpicker form-control" id="car_models"
name="car_models" required="1">
</select>
</div>
<div class="col-md-10 form-group" style="text-align: center">
<label><b>Select Year of Purchase:</b> </label><br>
<select class="selectpicker form-control" id="year" name="year"
required="1">
    {% for year in years %}
    <option value="{{ year }}">{{ year }}</option>
    {% endfor %}
</select>
</div>
<div class="col-md-10 form-group" style="text-align: center">
<label><b>Select the Fuel Type:</b> </label><br>
<select class="selectpicker form-control" id="fuel_type" name="fuel_type"
required="1">
    {% for fuel in fuel_types %}
    <option value="{{ fuel }}">{{ fuel }}</option>
    {% endfor %}
</select>
</div>
<div class="col-md-10 form-group" style="text-align: center">
<label><b>Enter the Number of Kilometres that the car has travelled:</b>
</label><br>
<input type="text" class="form-control" id="kilo_driven"
name="kilo_driven"
```

```
                        placeholder="Enter the kilometres driven ">
                </div><br>
                <div class="col-md-10 form-group" style="text-align: center">
                    <button style="width: 25%; height: 5%" class="btn btn-primary form-
control" onclick="send_data()">Predict Price</button>
                </div>
            </form>
            <br>
            <div class="row">
                <div class="col-12" style="text-align: center">
                    <h4><span id="prediction"></span></h4>
                </div>
            </div>
        </div>
    </div>
  </div>
</div></center>

<script>

  function load_car_models(company_id,car_model_id)
  {
      var  company=document.getElementById(company_id);
      var car_model= document.getElementById(car_model_id);
      console.log(company.value);
      car_model.value="";
      car_model.innerHTML="";
      {% for company in companies %}
          if( company.value == "{{ company }}")
          {
              {% for model in car_models %}
                  {% if company in model %}

                      var newOption= document.createElement("option");
                      newOption.value="{{ model }}";
```

```
                newOption.innerHTML="{{ model }}";
                car_model.options.add(newOption);
            {% endif %}
        {% endfor %}
    }
  {% endfor %}
}

function form_handler(event) {
    event.preventDefault(); // Don't submit the form normally
}
function send_data()
{
    document.querySelector('form').addEventListener("submit",form_handler);

    var fd=new FormData(document.querySelector('form'));

    var xhr= new XMLHttpRequest({mozSystem: true});

    xhr.open('POST','/predict',true);
    document.getElementById('prediction').innerHTML="Wait! Predicting Price ....";
    xhr.onreadystatechange = function(){
        if(xhr.readyState == XMLHttpRequest.DONE){
            document.getElementById('prediction').innerHTML="Prediction:
 ₹"+xhr.responseText;

        }
    };

    xhr.onload= function(){};

    xhr.send(fd);
}
</script>
```

```
<!-- jQuery first, then Popper.js, then Bootstrap JS -->
<script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
    integrity="sha384-
   DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj
   "
    crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
    integrity="sha384-
   Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
    crossorigin="anonymous"></script>
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
    integrity="sha384-
   OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI
   "
    crossorigin="anonymous"></script>
</body>
</html>
```

style.css:-

```
.{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
.bg-dark{
   background-color: none;
}
body {
  background-image: url("car.jpg");
 }
.mt-50{
   margin-top: 50px;
}
#canvas{
  border: 2px solid black;
}
```

application.py:-

```
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 new.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.debug = True
  app.run()
```

GitHub & Project Demo Link:-

https://github.com/IBM-EPBL/IBM-Project-54857-1662615484