**CAR RESALE VALUE PREDICTION**

**TEAM ID:** **PNT2022TMID12591**

Submitted by,

Indra Shekar G - 19Z219

Sanjai M - 19Z241

Manoj S - 19Z263

Mathana Sekaran T - 19Z264

1. **INTRODUCTION** 
   1. Project Overview
   2. Purpose
2. **LITERATURE SURVEY**
   1. Existing problem
   2. References
   3. Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas
   2. Ideation & Brainstorming
   3. Proposed Solution
   4. Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   1. Functional requirement
   2. Non-Functional requirements
5. **PROJECT DESIGN**
   1. Data Flow Diagrams
   2. Solution & Technical Architecture
   3. User Stories
6. **PROJECT PLANNING & SCHEDULING**
   1. Sprint Planning & Estimation
   2. Sprint Delivery Schedule
   3. Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   1. Feature 1
   2. Feature 2
   3. Database Schema (if Applicable)
8. **TESTING** 
   1. User Acceptance Testing and Test Cases
9. **RESULTS**
   1. Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **APPENDIX**

Source Code

GitHub & Project Demo Link

**CHAPTER -1**

**INTRODUCTION**

* 1. **PROJECT OVERVIEW**

Prediction of car resale value depends on various factors of the car so expert knowledge is needed in this task. There are many features that affect the price of car like fuel type, age, model, mileage, year of registration, damage of car, engine power, capacity, etc. Various other features like color, seater, precious owner, braking system, Presence of extra features like air conditioner and other factors also affect the pricing of car. Machine Learning can be used in this case to predict the price. More specifically supervised regression model can be used in this task.

* 1. **PURPOSE**

Car resale value prediction help people who are interested in buying used cars by showing the predicted price for the given price so he/she will be knowing whether the amount he/she pays is worth the car. Also, it helps seller to find the price of his used car so he will be saved from under selling his/her car than actual price. Car Resale value prediction also removes the third-party people involved in resale care trade resulting in higher profit for both seller and buyer. 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 algorithms with the best accuracy will be taken as a solution, then it will be integrated to the web-based application

**CHAPTER - 2**

**LITERATURE SURVEY**

**2.1 EXISTING PROBLEM**

The price of new cars is set by the respective manufactures which will be always high which people may afford. Their only option is to buy second hand used cars. Even though they are exploited by some sellers who set their car for high prices and the buyers who are not aware of it will buy it as it will be lower compared to new cars. So, there is need for predicting the worthiness of a used car based on its features.

**2.2 REFERENCES**

[1] Abdulla ALShared, Used Car Price Prediction and valuation using Data Mining Techniques

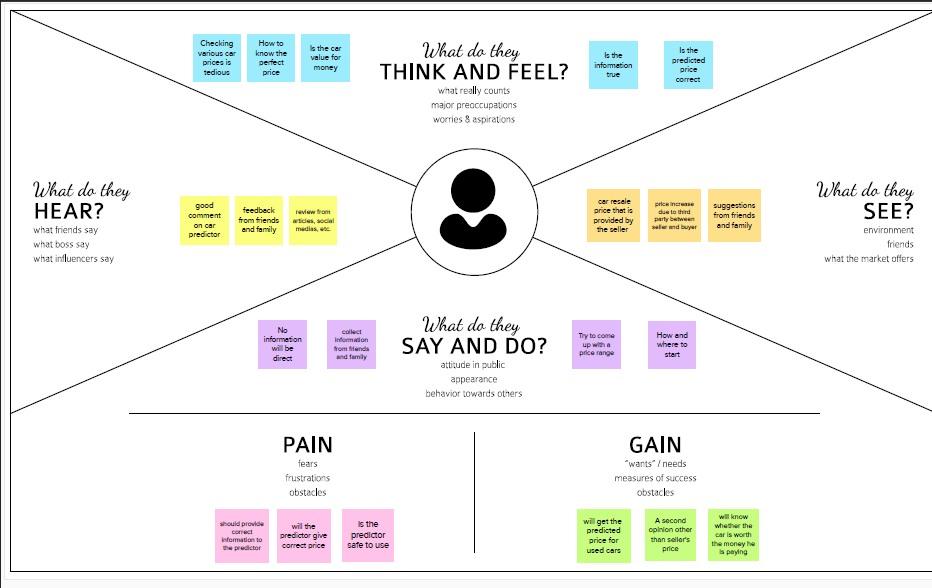
[2] Ning sun, Hongxi Bai, Yuxia Geng, Huizhu Shi, Price Evaluation model in secondhand car system

[3] Pudaruth Sameerchand, Pudaruth Sameerchand, Predicting the price of Used Car Using Machine Learning Techniques

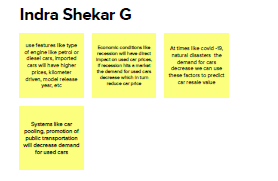
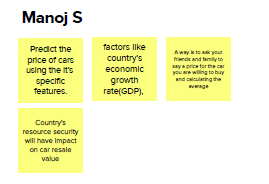
[4] Kiran S, Prediction of resale value of the car Using Linear Regression Algorithm

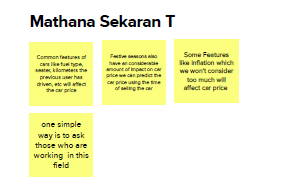
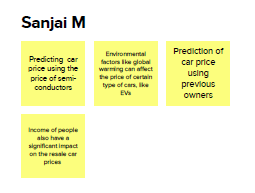
**CHAPTER - 3**

**IDEATION AND PROPOSED SOLUTION**

**3.1 EMPATHY MAP CANVAS**

**3.2 IDEATION AND BRAINSTORMING**

****

****

**3.3 PROPOSED SOLUTION**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Parameter** | **Description** |
|  | Problem Statement (Problem to be solved) | Car Resale Value Prediction |
|  | Idea / Solution description | Use Machine Learning Models (supervised) to predict the car resale value using the features like kilometre driven, fuel type, registration year, model, etc |
|  | Novelty / Uniqueness | The Application will have an interface specific for car resale value that will display the predicted price in a easy to read manner, Also This application is also secure as it doesn’t need any personal information about the users to predict the price |
|  | Social Impact / Customer Satisfaction | Will make sellers provide cars with reasonable price, also will make the customers to buy cars that are value for the money they pay |
|  | Business Model (Revenue Model) | Making it large scale will enable the service providers to charge a small amount for price prediction which the customer will be ready to pay as it will be less than the seller’s price |
|  | Scalability of the Solution | The proposed solution can be scaled to any vehicles by adding their features to the dataset and creating a new ML model and using the new ml model can be used to predict prices |

**3.4 PROBLEM SOLUTION FIT**

**CHAPTER - 4**

**REQUIREMENT ANALYSIS**

**4.1 FUNCTIONAL REQUIREMENTS**

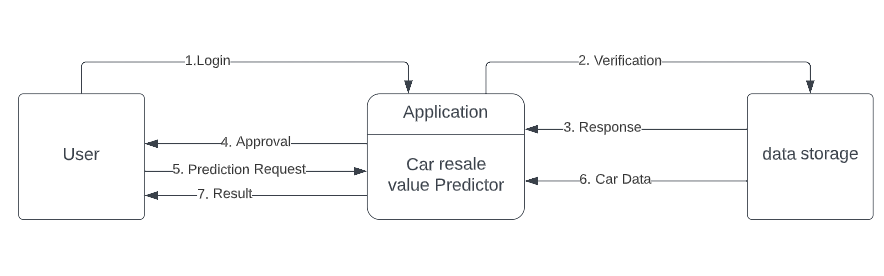
|  |  |  |
| --- | --- | --- |
| **FR No.** | **Functional Requirement (Epic)** | **Sub Requirement (Story / Sub-Task)** |
| FR-1 | User Opens Website | Server opens the site |
| FR-2 | User Enters the values | User enters Car details |
| FR-3 | Validation | Checks values and perform validation |
| FR-4 | Prediction | Predicting the Car Resale Value by using the features present in dataset using a ML model |

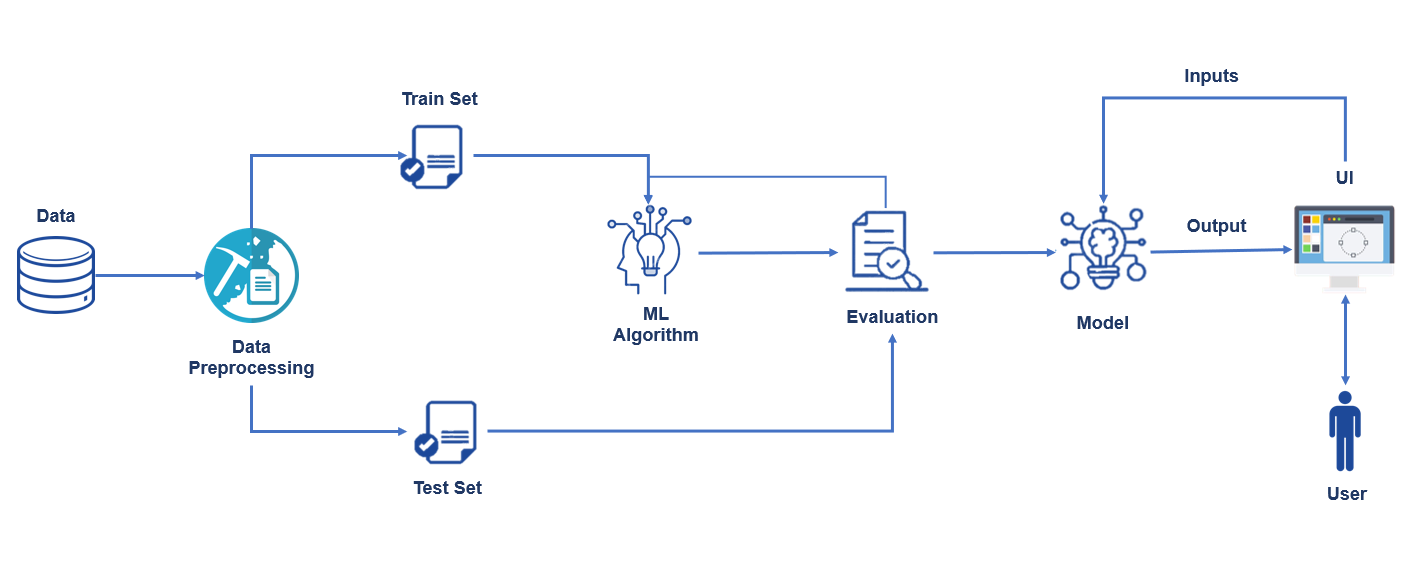
**4.2 NON-FUNCTIONAL REQUIREMENTS**

|  |  |  |
| --- | --- | --- |
| **FR No.** | **Non-Functional Requirement** | **Description** |
| NFR-1 | **Usability** | The user interface should be easy to use and there should be any prerequisite that the user should know |
| NFR-2 | **Security** | The information provider by the user should be secure |
| NFR-3 | **Reliability** | The Application should be reliable compared to other solutions available for this problem |
| NFR-4 | **Performance** | The performance of the system should be highly effective. I.e., the accuracy should be higher |
| NFR-5 | **Availability** | The application should be able to be accessed by all types of users, also it should be able to perform both with and without internet |
| NFR-6 | **Scalability** | The application should perform well in case of multiple users using it on the same time, also it should be able to be updated over the time |

**CHAPTER - 5**

**PROJECT DESIGN**

**5.1 DATA FLOW DIAGRAM**

**5.2 SOLUTION ARCHITECTURE**

**5.3 USER STORIES**

| **User Type** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Acceptance criteria** | **Priority** | **Release** |
| --- | --- | --- | --- | --- | --- | --- |
| Customer | Registration | USN-1 | A user can register for the application by entering my email, password, and confirming my password. | Account specific tasks | High | Sprint-1 |
|  |  | USN-2 | A user will receive confirmation email once registered for the application | Verify the registered account | High | Sprint-1 |
|  |  | USN-3 | Validation of user using Gmail | Account is validated and got access to dashboard | Medium | Sprint-1 |
|  | Login | USN-4 | Using username and password login to application | Right information should be given | High | Sprint-1 |
|  | Dashboard | USN-5 | The user can give the car features that he is planning to buy | Only cars will be accepted, and only certain features will be accepted | High | Sprint-2 |
| Support Team | Support | USN-6 | Responds to user queries via telephone, email etc. | Queries can be raised in case of any doubts | Medium | Sprint-3 |
|  |  | USN-7 | The team should be able to provide effective solution to the queries | The user will get all their doubt clarified | Low | Sprint-3 |
|  |  | USN-8 | The team must respond to the queries based on the importance of the query | Queries get resolved | Low | Sprint-3 |
| Development Team | Ain function | USN-9 | Design the application with good user interface and build a ml model that provides high accuracy | Interfaces understandable to all | High | Sprint-4 |

**CHAPTER - 6**

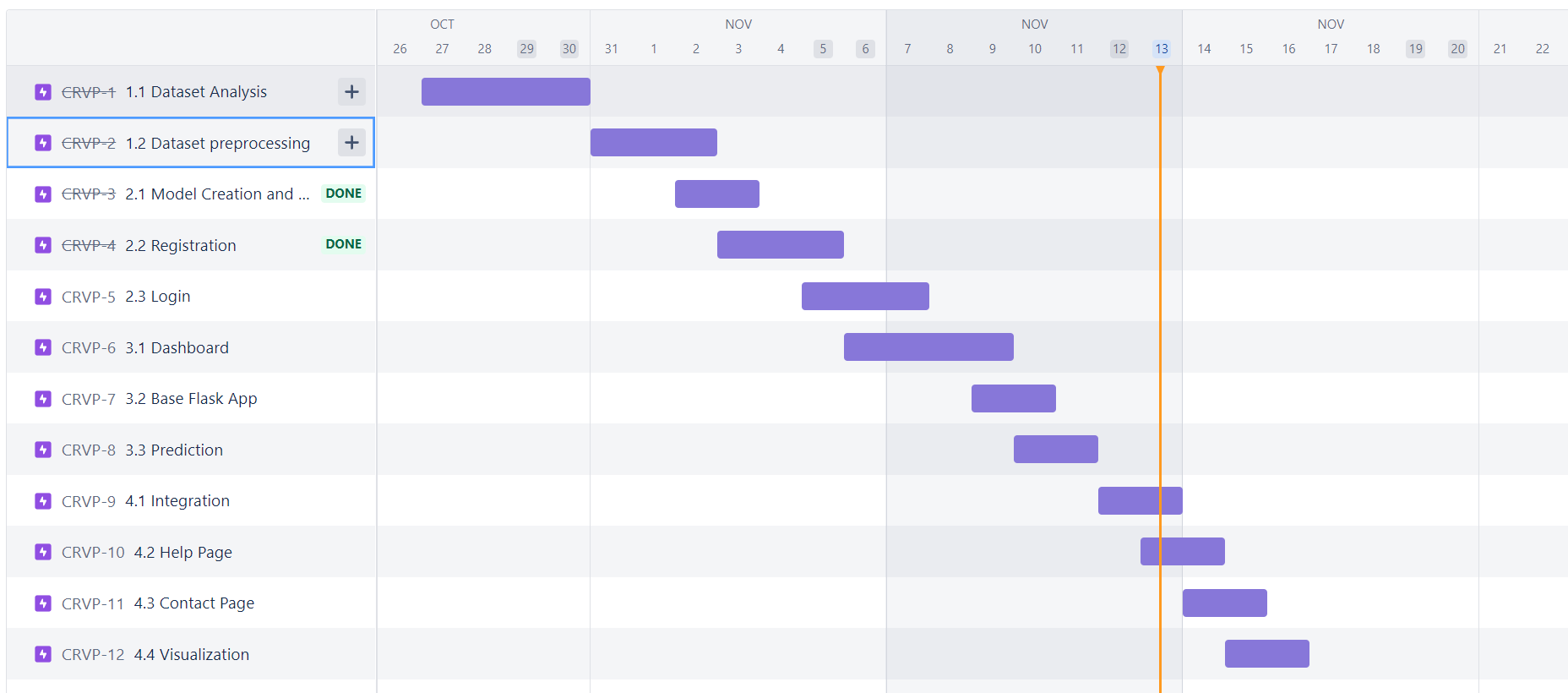
**PROJECT PLANNING AND SCHEDULING**

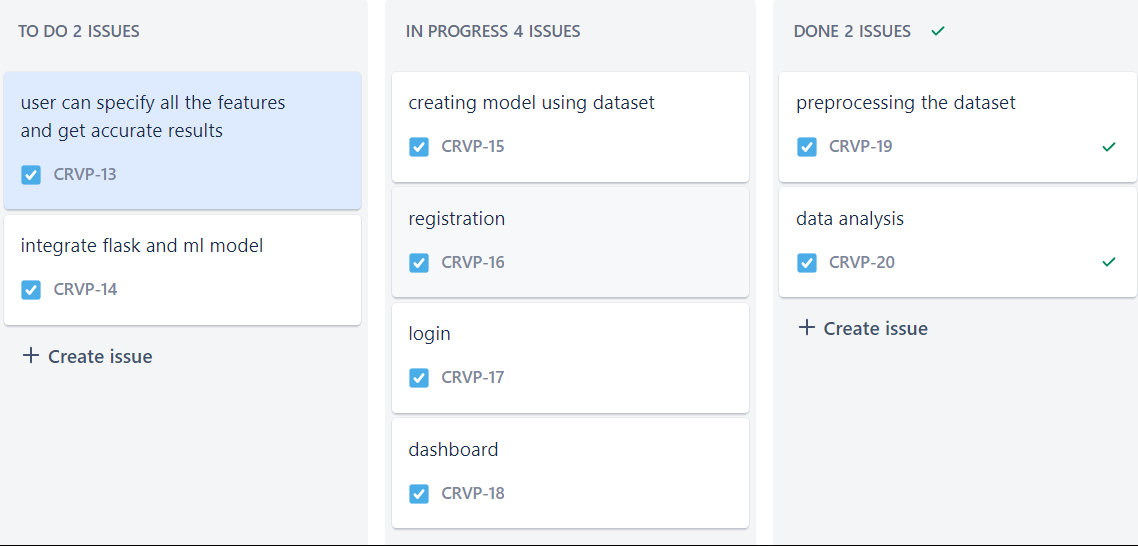
* 1. **SPRINT PLANNING AND ESTIMATION**

| **Sprint** | **Functional Requirement (Epic)** | **User Story Number** | **User Story / Task** | **Story Points** | **Priority** | **Team Members** |
| --- | --- | --- | --- | --- | --- | --- |
| Sprint-1 | Build model and Flask App | USN-1 | Create a regression model and a flask web page | 10 | High | Sanjai M  Manoj S |
| Sprint -1 | Integrate with model | USN-2 | Get inputs, integrate with model, and check the outputs | 10 | High | Sanjai M  Indra Shekar G |
| Sprint-2 | Building the model | USN-3 | Build Ml model using Python | 5 | Medium | Sanjai M  Mathana Sekaran T |
| Sprint-2 | Deploy with Dataset | USN-4 | Building a python code for predicting price | 10 | Medium | Manoj S  Indra Shekar G |
| Sprint-3 | Build Model in cloud | USN-5 | Predict the data using the model | 20 | High | Mathana Sekaran T  Indra Shekar G |
| Sprint-4 | Display the predicted values | USN-6 | Integrate Flask with Scoring End Pont | 20 | High | Manoj S  Mathana Sekaran T |

* 1. **SPRINT DELIEVERY 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 | 4 Days | 25 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 |

* 1. **REPORTS FROM JIRA**

****

**CHAPTER - 7**

**CODING AND SOLUTIONING**

**7.1 FEATURE 1**

**index.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" type ="text/css" href="{{ url\_for('static', filename='css/styles.css') }}" />

<!-- Bootstrap -->

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"

rel="stylesheet"

integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"

crossorigin="anonymous"

/>

<title>Resale Value Prediction</title>

</head>

<body>

<div class="heading\_div">

<h1 class="heading">Get The Accurate Resale Value Of Your Car!!</h1>

</div>

<div class="wrapper">

<img class="image" src="{{ url\_for('static', filename='images/CarMain.jpg') }}" alt="carImg" />

<div class="rightContainer">

<h1 class="para">

It is expected that sales of reconditioned used automobiles would rise

in light of the current challenging economic climate. Leasing cars

rather than owning them entirely is a prevalent practise in the

majority of the world's nations. A lease is a legally binding

agreement between a buyer and a seller wherein the buyer is required

to pay the seller or financier in set payments over a certain period

of time, usually months or years. After the lease term is up, the

buyer will have the option of purchasing the vehicle for its residual

value, or anticipated resale value. Therefore, the ability for sellers

and financiers to accurately anticipate the residual value of

automobiles is of economic significance.

</h1>

<a href="{{ url\_for('prediction\_page') }}">

<button type="button" class="btn button btn-dark">

WANT TO KNOW THE RESALE VALUE OF YOUR CAR??

</button></a

>

</div>

</div>

<!-- Bootstrap -->

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"

integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"

crossorigin="anonymous"

></script>

</body>

</html>

**form.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" />

<title>Resale Value Prediction</title>

<link rel="stylesheet" type ="text/css" href="{{ url\_for('static', filename='css/styles\_form.css') }}" />

<!-- Bootstrap -->

<link

href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/css/bootstrap.min.css"

rel="stylesheet"

integrity="sha384-Zenh87qX5JnK2Jl0vWa8Ck2rdkQ2Bzep5IDxbcnCeuOxjzrPF/et3URy9Bv1WTRi"

crossorigin="anonymous"

/>

</head>

<body>

<div class="heading\_div">

<h1 class="heading">Get The Accurate Resale Value Of Your Car!!</h1>

</div>

<h3 class="heading-sub">Please fill the following details of your car</h3>

<div class="wrapper">

<div><form action="{{ url\_for('prediction') }}" method="post" id="form">

<!-- Registration Year -->

<div class="mb-3">

<label for="registrationYear" class="form-label"

>Registration Year</label

>

<input placeholder="YYYY" type="number" class="form-control" name="registrationYear" id="registrationYear" required />

</div>

<!-- Registration Month -->

<div class="mb-3">

<label for="registrationMonth" class="form-label"

>Registration Month</label

>

<input placeholder="(0-12)" type="number" class="form-control" name="registrationMonth" id="registrationMonth" required />

</div>

<!-- Power Of Car In PS -->

<div class="mb-3">

<label for="powerOfCarInPS" class="form-label"

>Power Of Car In PS</label

>

<input placeholder="234" type="number" class="form-control" name="powerOfCarInPS" id="powerOfCarInPS" required />

</div>

<!-- Kilometers The Car As Driven -->

<div class="mb-3">

<label for="KmTheCarAsDriven" class="form-label"

>Kilometers The Car As Driven</label

>

<input placeholder="38593" type="number" class="form-control" name="KmTheCarAsDriven" id="KmTheCarAsDriven" required />

</div>

<!-- Gear Box Type -->

<h6 class="form-check-inline">Gear Box Type</h6>

<div class="form-check form-check-inline mb-3">

<input

class="form-check-input"

type="radio"

name="gearbox"

id="inlineRadio1"

value="manuell"

required

/>

<label class="form-check-label" for="inlineRadio1">Manual</label>

</div>

<div class="form-check form-check-inline">

<input

class="form-check-input"

type="radio"

name="gearbox"

id="inlineRadio2"

value="automatik"

required

/>

<label class="form-check-label" for="inlineRadio2">Automatic</label>

</div>

<div class="form-check form-check-inline">

<input

class="form-check-input"

type="radio"

name="gearbox"

id="inlineRadio2"

value=""

required

/>

<label class="form-check-label" for="gearbox"

>Not declared</label

>

</div>

</div>

<div class="second-div">

<!-- Model Type -->

<div class="mb-3">

<label for="modelType" class="form-label">Model</label>

<input placeholder="2\_reihe" type="text" class="form-control" name="modelType" id="modelType" required />

</div>

<!-- Brand Of The Car -->

<div class="mb-3">

<label for="brandOfTheCar" class="form-label">Brand Of The Car</label>

<input placeholder="jaguar" type="text" class="form-control" name="brandOfTheCar" id="brandOfTheCar" required />

</div>

<!-- Fuel Type Of The Car -->

<div class="mb-3">

<label for="fuelTypeOfTheCar" class="form-label"

>Fuel Type Of The Car</label

>

<input placeholder="diesel / benzin / lpg" type="text" class="form-control" name="fuelTypeOfTheCar" id="fuelTypeOfTheCar" required />

</div>

<!-- Vehicle Type -->

<div class="mb-3">

<label for="vehicleType" class="form-label">Vehicle Type</label>

<input placeholder="bus" type="text" class="form-control" name="vehicleType" id="vehicleType" required />

</div>

<!-- Car is damaged -->

<h6 class="form-check-inline">Car Is Damaged</h6>

<div class="form-check form-check-inline mb-3">

<input

class="form-check-input"

type="radio"

name="damage"

id="inlineRadio1"

value="ja"

required

/>

<label class="form-check-label" for="inlineRadio1">Yes</label>

</div>

<div class="form-check form-check-inline">

<input

class="form-check-input"

type="radio"

name="damage"

id="inlineRadio2"

value="nein"

required

/>

<label class="form-check-label" for="inlineRadio2">No</label>

</div>

<div class="form-check form-check-inline">

<input

class="form-check-input"

type="radio"

name="damage"

id="inlineRadio2"

value=""

required

/>

<label class="form-check-label" for="damage"

>Not declared</label

>

</div>

</form>

</div>

</div>

<div class="btn-wrapper">

<button class="btn button btn-dark" type="submit" form="form" value="Submit">Predict</button>

</div>

<div>

<h3 class="output">Predicted Price: {{ pred\_result }}</h3>

<!-- <h3 class="output">{{ pred\_result }}</h3> -->

</div>

<script

src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.2/dist/js/bootstrap.bundle.min.js"

integrity="sha384-OERcA2EqjJCMA+/3y+gxIOqMEjwtxJY7qPCqsdltbNJuaOe923+mo//f6V8Qbsw3"

crossorigin="anonymous"

></script>

</body>

</html>

**7.2 FEATURE 2**

from flask import Flask, request, render\_template

import pickle

import math

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

app = Flask(\_\_name\_\_)

model = pickle.load(open('model.sav', 'rb'))

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/prediction\_page')

def prediction\_page():

return render\_template('form.html')

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

def prediction():

# regyear = request.form.get('registrationYear')

# powerps = request.form.get('powerOfCarInPS')

# kms = request.form.get('KmTheCarAsDriven')

# regmonth = request.form.get('registrationMonth')

# gearbox = request.form.get('gearbox')

# damage = request.form.get('damage')

# model\_type = request.form.get('modelType')

# brand = request.form.get('brandOfTheCar')

# fuel\_type = request.form.get('fuefuelTypeOfTheCar')

# vehicle\_type = request.form.get('vehicleType')

# print(regyear, powerps, kms, regmonth, gearbox, damage, model\_type, brand, fuel\_type, vehicle\_type)

regyear = int(request.form.get('registrationYear'))

powerps = float(request.form.get('powerOfCarInPS'))

kms = float(request.form.get('KmTheCarAsDriven'))

regmonth = int(request.form.get('registrationMonth'))

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

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

model\_type = request.form.get('modelType')

brand = request.form.get('brandOfTheCar')

fuel\_type = request.form.get('fuelTypeOfTheCar')

vehicle\_type = request.form.get('vehicleType')

print(regyear, powerps, kms, regmonth, gearbox, damage, model\_type, brand, fuel\_type, vehicle\_type)

new\_row = {

'vehicleType' : vehicle\_type,

'yearOfRegistration' : regyear,

'gearbox' : gearbox,

'powerPS' : powerps,

'model' : model\_type,

'kilometer' : kms,

'monthOfRegistration' : regmonth,

'fuelType' : fuel\_type,

'brand' : brand,

'notRepairedDamage' : damage

}

print(new\_row)

new\_df = pd.DataFrame(columns=['vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand', 'notRepairedDamage'])

new\_df = new\_df.append(new\_row, ignore\_index=True)

labels = ['vehicleType', 'gearbox', 'model', 'fuelType', 'brand', 'notRepairedDamage']

mapper = {}

for i in labels:

mapper[i] = LabelEncoder()

mapper[i].classes\_ = np.load(str('numpy\_classes/classes' + i + '.npy'), allow\_pickle=True)

tr = mapper[i].fit\_transform(new\_df[i])

new\_df.loc[:, i + '\_labels'] = pd.Series(tr, index=new\_df.index)

labeled = new\_df[[

'yearOfRegistration',

'kilometer',

'monthOfRegistration',

'powerPS'

]

+ [x + '\_labels' for x in labels]]

x = labeled.values

print(x)

result = model.predict(x)[0]

result = math.ceil(result)

result = '$' + str(result)

print('The predicted result: ', result)

return render\_template('form.html', pred\_result=result)

if \_\_name\_\_ == '\_\_main\_\_':

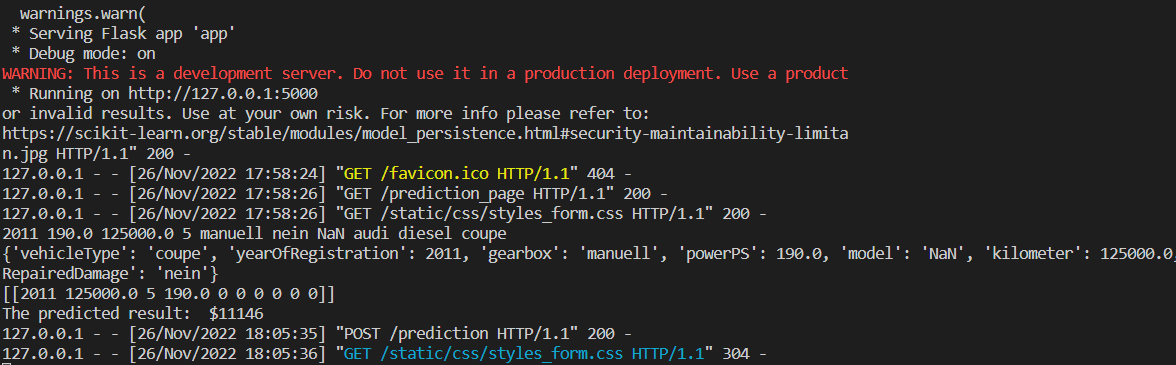
app.run(debug=True)

**CHAPTER - 8**

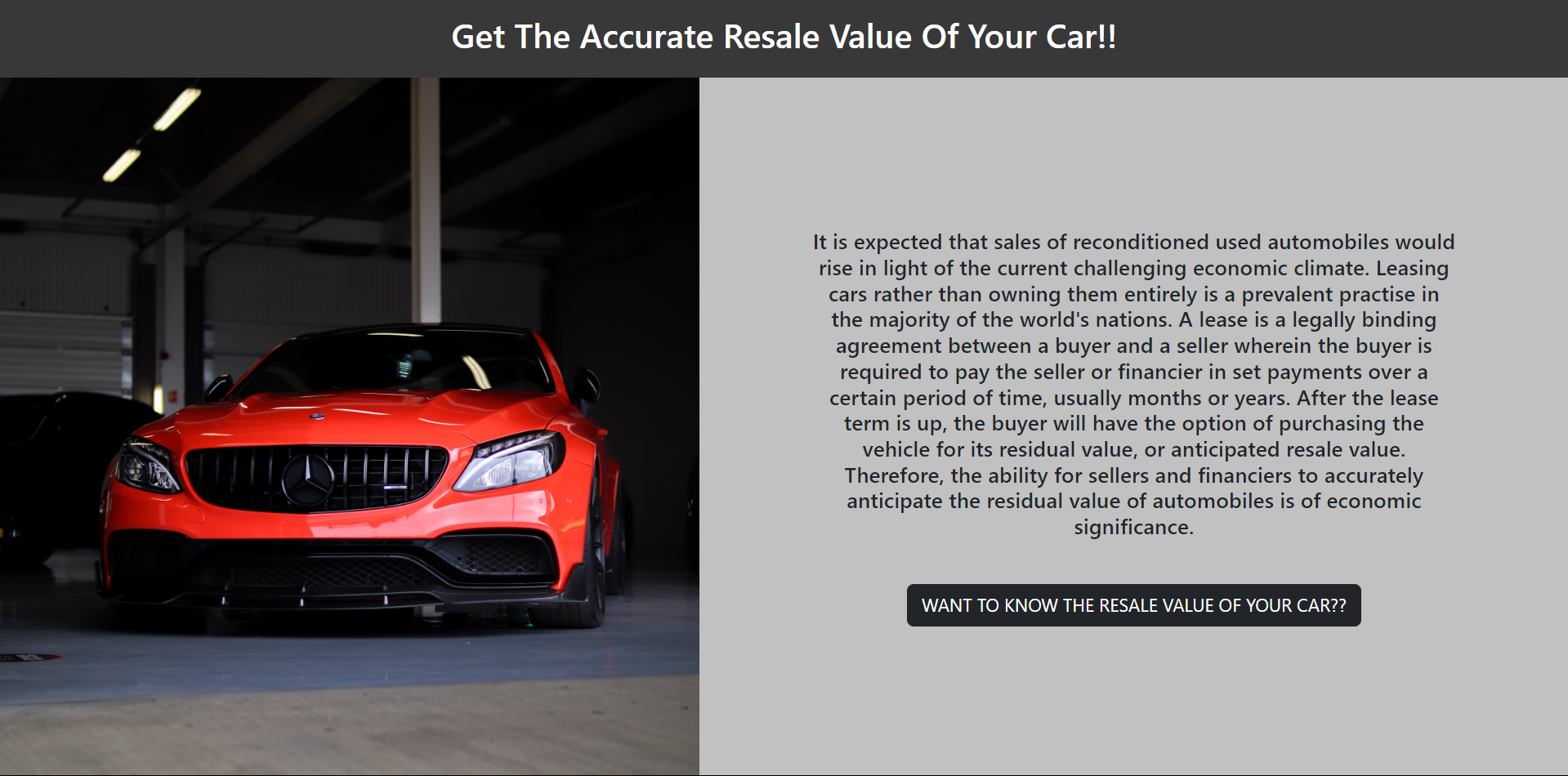
**TESTING**

**8.1 USER ACCEPETENCE TESTING AND TEST CASES**

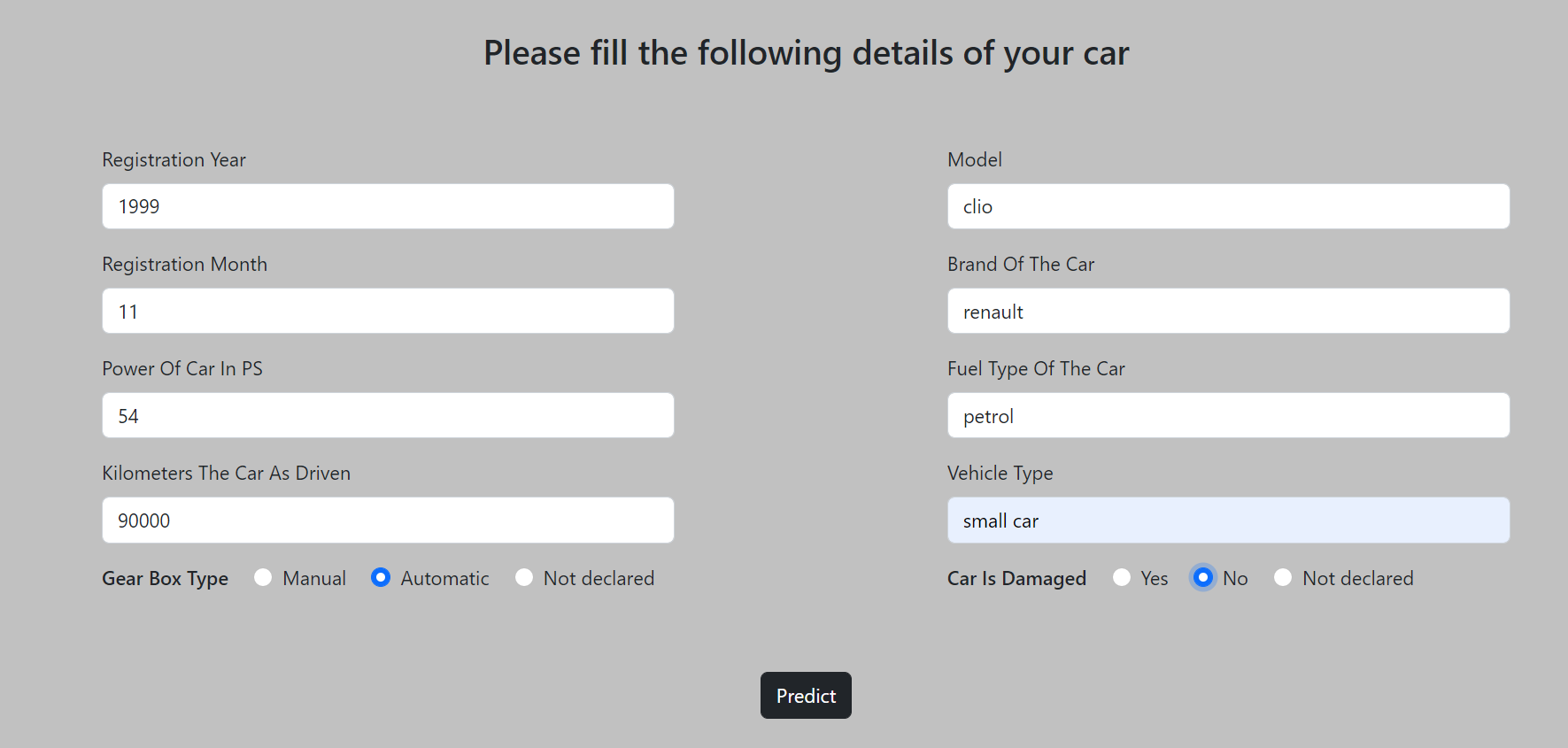
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TEST CASE ID** | **FEATURE TYPE** | **COMPONENT** | **TEST SCENARIO** | **PREREQUISITE** |
| **HOME PAGE** | Functional | Home Page | User can see the Images and Hyperlinks and visit predict button | Enter the URL  Button to redirect the page |
| **PREDICTION AND RESULT PAGE** | Functional | Predict and Result | User can enter the values and see the prediction price | Enter Predict Button  User Should enter the details  See Predicted Price |

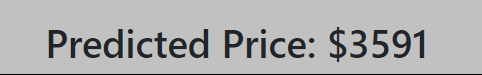


**index.html**

****

**forms.html**

****



**CHAPTER – 9**

**RESULTS**

**9.1 PERDFORMANCE METRICS**

Mean Absolute Error: 1421.45607

Mean Squared Error: 8095671.045234

Root Mean Squared Error: 2845.28922

R2 Score: 0.747891214137351

**CHAPTER – 10**

**ADVANTAGES AND DISADVANTAGES**

**ADVANTAGES:**

* Helps in increasing the profit of both seller and buyer by eliminating the brokerage
* Able to find best car for the budget
* Good pricing for both buyer and seller

**DISADVANTAGES:**

* User Should provide accurate and valid information
* Price of the cars also depends other factors like presence of extra accessories

**CHAPTER – 11**

**CONCLUSION**

Price prediction analyzes a good or service based on its attributes, demand and current market trends using an algorithm. The pricing is then adjusted by the program at a level that it believes would both draw people and optimize sales.

**CHAPTER – 12**

**APPENDIX**

**MODEL BUILDING:**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import pickle

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split, cross\_val\_score

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import r2\_score

df=pd.read\_csv('autos.csv',header=0,sep=',',encoding='Latin1')

df.drop(columns=['seller','offerType','name','abtest','dateCrawled','nrOfPictures','lastSeen','postalCode','dateCreated'],inplace=True,axis=1)

df=df[(df['powerPS']>50) & (df['powerPS']<90)]

df=df[(df['yearOfRegistration']>1949) & (df['yearOfRegistration']<2017)]

df=df[(df['price']>=100) & (df['price']<=150000)]

new\_df=df.copy()

new\_df=new\_df.drop\_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS','model', 'kilometer', 'monthOfRegistration', 'fuelType','brand','notRepairedDamage'])

new\_df['gearbox'].replace(('manuell','automatik'),('manual','automatic'),inplace=True)

new\_df['fuelType'].replace(('benzin','andere','elektro'),('petrol','others','electric'),inplace=True)

new\_df['vehicleType'].replace(('kleinwagen','cabrio','kombi','andere'),('small car','convertible','combination','others'),inplace=True)

new\_df['notRepairedDamage'].replace(('ja','nein'),('yes','no'),inplace=True)

new\_df.to\_csv('autos\_preprocessed.csv')

labels=['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']

dicti={}

for feature in labels:

dicti[feature]=LabelEncoder()

dicti[feature].fit(new\_df[feature])

temp=dicti[feature].transform(new\_df[feature])

np.save(str('classes'+feature+'.npy'),dicti[feature].classes\_)

print(feature,':',dicti[feature])

new\_df.loc[:,feature+'\_Labels']=pd.Series(temp,index=new\_df.index)

labeled=new\_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']+[x+'\_Labels' for x in labels]]

X=labeled.iloc[:,1:].values

Y=labeled.iloc[:,0].values

Y=Y.reshape(-1,1)

X\_train,X\_test,y\_train,y\_test=train\_test\_split(X,Y,test\_size=.3)

regressor=RandomForestRegressor(n\_estimators=1000,max\_depth=10)

regressor.fit(X\_train,np.ravel(y\_train,order='C'))

file='model.sav'

pickle.dump(regressor,open(file,'wb'))

**GitHub Link:** [**https://github.com/IBM-EPBL/IBM-Project-13434-1659518403**](https://github.com/IBM-EPBL/IBM-Project-13434-1659518403)