# Project On

Car Resale Value prediction

Powered by IBM India

Submitted by

Veera Ashwin S

Thankakumar K

Sakthivel N

Sanjay Ram R

**PROJECT ID: PNT2022TMID28152**

Mentor: **Jaspin K**

**Industrial Mentor:Swetha**

**1.1 INTRODUCTION**

This project “Car Resale Value Prediction” aims to build a model to predict used cars' reasonable prices based on multiple aspects, including vehicle mileage, year of manufacturing, fuel consumption, transmission, road tax, fuel type, and engine size. This model can benefit sellers, buyers, and car manufacturers in the used cars market. Upon completion, it can output a relatively accurate price prediction based on the information that users input. The model building process involves machine learning and data science. The dataset used was scraped from listings of used cars. Various regression methods, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression, were applied in the research to achieve the highest accuracy. Before the actual start of model-building, this project visualised the data to understand the dataset better. The dataset was divided and modified to fit the regression, thus ensuring the performance of the regression.

## 1.1 Project Overview

A car price prediction has been a high interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars, the applied three machine learning techniques are random forest ,KN-N and linear regression algorithm. Respective performances of different algorithms were then compared to find one that best suits the

available data set. This ability to capture data, analyse it and use it to personalise a shopping experience or implement is the future of retail.

Parameters involved :

Car_name;Year;Selling_Price;Present_Price;Kms_Driven;Fuel_type;

Seller_type;Transmission;Owner and so on.

## 1.2 Purpose

Car resale value prediction helps the user to predict the re sale value of the car depending upon various features like kilo-metres driven, fuel type, etc. This resale value prediction system is made for general purpose to just predict the amount that can be roughly acquired by the user. The most essential elements for forecast are brand and model, period use of vehicle, mileage of vehicle, gear type and fuel type utilised in the vehicle just as fuel utilisation per mile profoundly influences cost of a vehicle because of continuous changes in the cost of a fuel. In view of the differing highlights and factors, and furthermore with the assistance of master information the vehicle value forecast has been done precisely.

## 2. LITERATURE SURVEY

### 2.1 Existing problem

The problem is defined as the optimised way to estimate in surance cost based on the manufacturer with some additional costs incurred by the Government in the form of taxes. As the existing methods for estimating the cost takes a lot of time and energy and due to the increased price of new cars and the inability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase.

The prices of new cars in the industry is fixed by the So, customers buying a new car can be assured of the money they invest to be worthy. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offer this service, their prediction method may not be the best. Besides, different models and systems may contribute to predicting power for a used car's actual market value. It is important to know their actual market value while both buying and selling.

## 2.2 References

At present, under the guidance of the new generation of information technology,the rapid accumulation of data, the continuous improvement of computing power, the continuous optimization of algorithm models, and the rapid rise of multi-scene applications have made profound changes in the development environment of Machine Learning.


**[1] Kanwal Noor, 2017, Vehicle Price Prediction System using Machine Learning Techniques International Journal of Computer Applications. Volume 167 - Number 9**

**[2] Mariana Lusitania et al, (2009). Support vector regression analysis for price prediction in a vehicle leasing application [3] Richardson, M. S. (2009). Determinants of used vehicle resale value.**

**[3] Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, Master thesis, TU Hamburg-Harburg).**

**[4]T. D. Phan, "Housing Price Prediction Using Machine Learning Algorithms: The Case of Melbourne City Australia", *2018 International Conference on Machine Learning and Data Engineering (iCMLDE)*, pp. 35-42, 2018.**

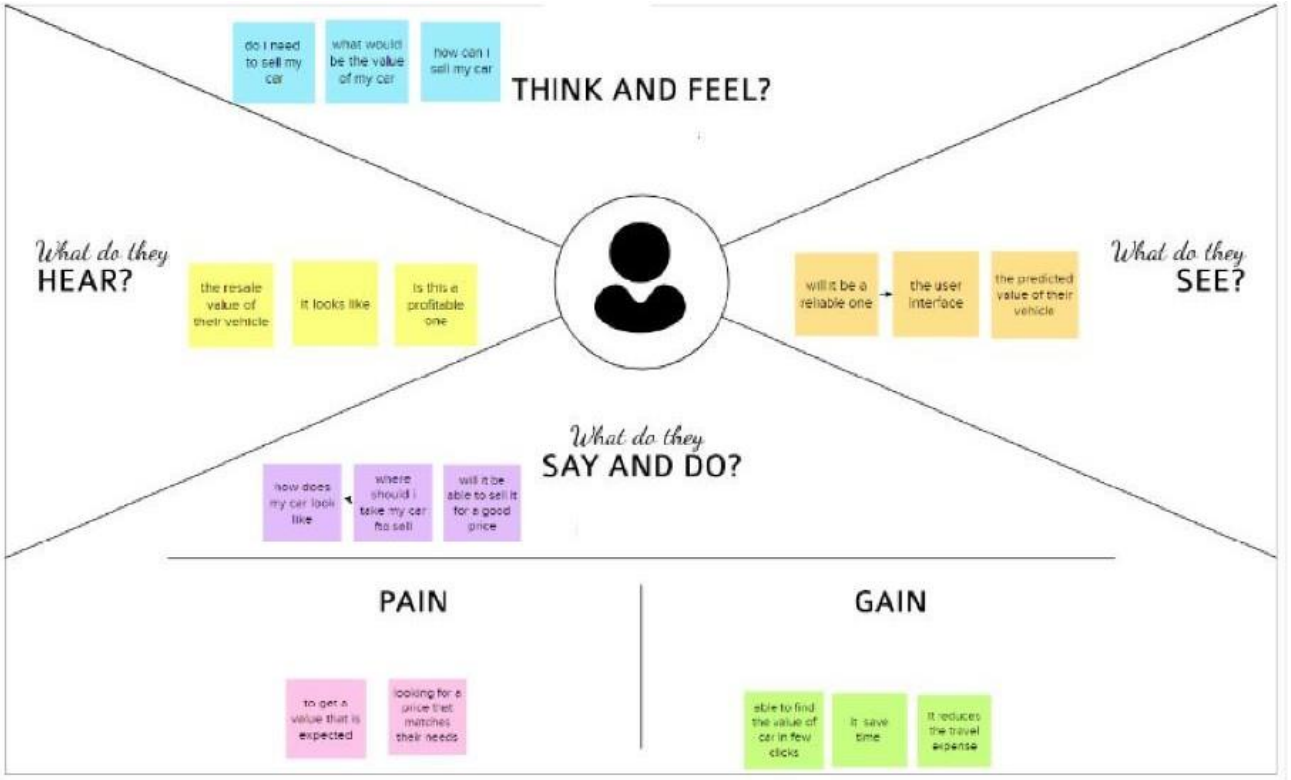
**[5]K. Samruddhi and R. Ashok Kumar, "Used Car Price Prediction using K-Nearest Neighbor Based Model", *International Journal of Innovative Research in Applied Sciences and Engineering*, vol. 4, no. 3, pp. 686-689, 2020.**

**[6]O. Celik and U. O. Osmanoglu, "Prediction of The Prices of Second-Hand Cars", *Avrupa Bilim ve Teknoloji Dergisi*, no. 16, pp. 77-83, Aug. 2019.**

### *2.3 Problem statement definition*

## *3 . IDEATION& PROPOSED SOLUTION*

### *3.1 Empathy Map Canvas*



### *3.2 Ideation & Brainstorming*

**STEP 1:**

- Prediction using Car image.
- By using the exterior and interior image of the car.
- The value will be predicted based on the appearance of the car.If there any damage or n numbers scratches the car resale value will be quite affected.
- By using neural network value of the car can be predicted.
- Neural network algorithm is developed by considering the human brain that takes a set of units as input and transfers results to a predefined output.

**STEP 2:**

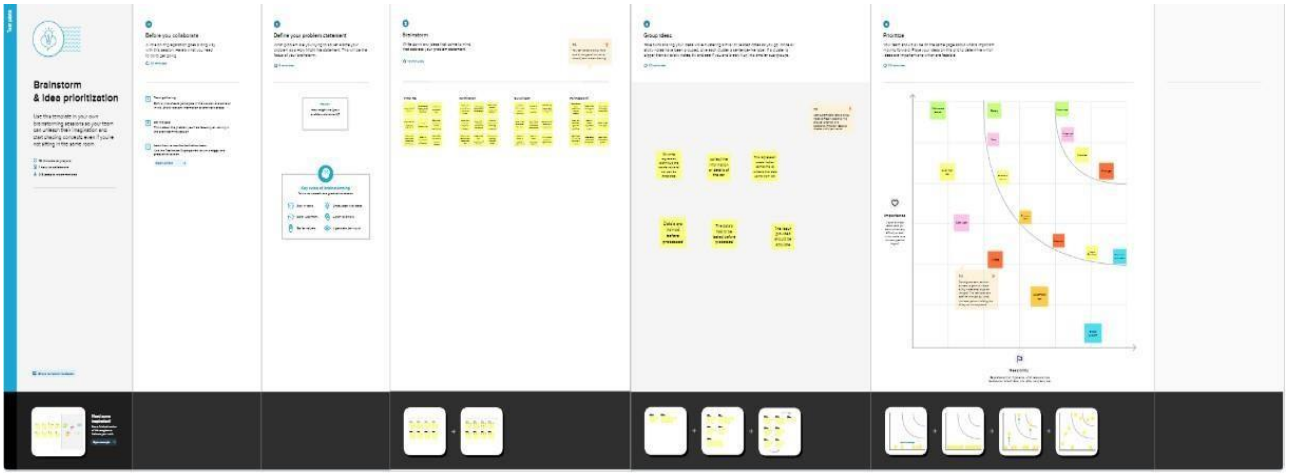
- The main objective of this project is to predict the Prices of used cars, compare the prices and also estimate the lifespan of a particular car.
- Insurance, Company claims,etc.
- regression Algorithm is used to predict the value.
- Regression model based on k-nearest neighbour machine learning algorithm was used to predict the price of a car.

**STEP 3:**

- Prediction using engine car condition.
- User should upload engine sound in the format of audio file.
- By using Convolutional Neural Networks methodology price can be predicted.
- CNNs for Machine Learning on sound data by spectrogram approach that was just converts each song (or song segment) into a spectrogram: a two dimensional matrix

**STEP 4:**

- Economic Conditions.
- Kilo-metres Covered.
- Its mileage (the number of kilometres it has run) and its horsepower
- Car prediction using XGBoost algorithm accurate results will be monitored.
- XGBoost as a regression model gave the best MSLE and RMSE values.

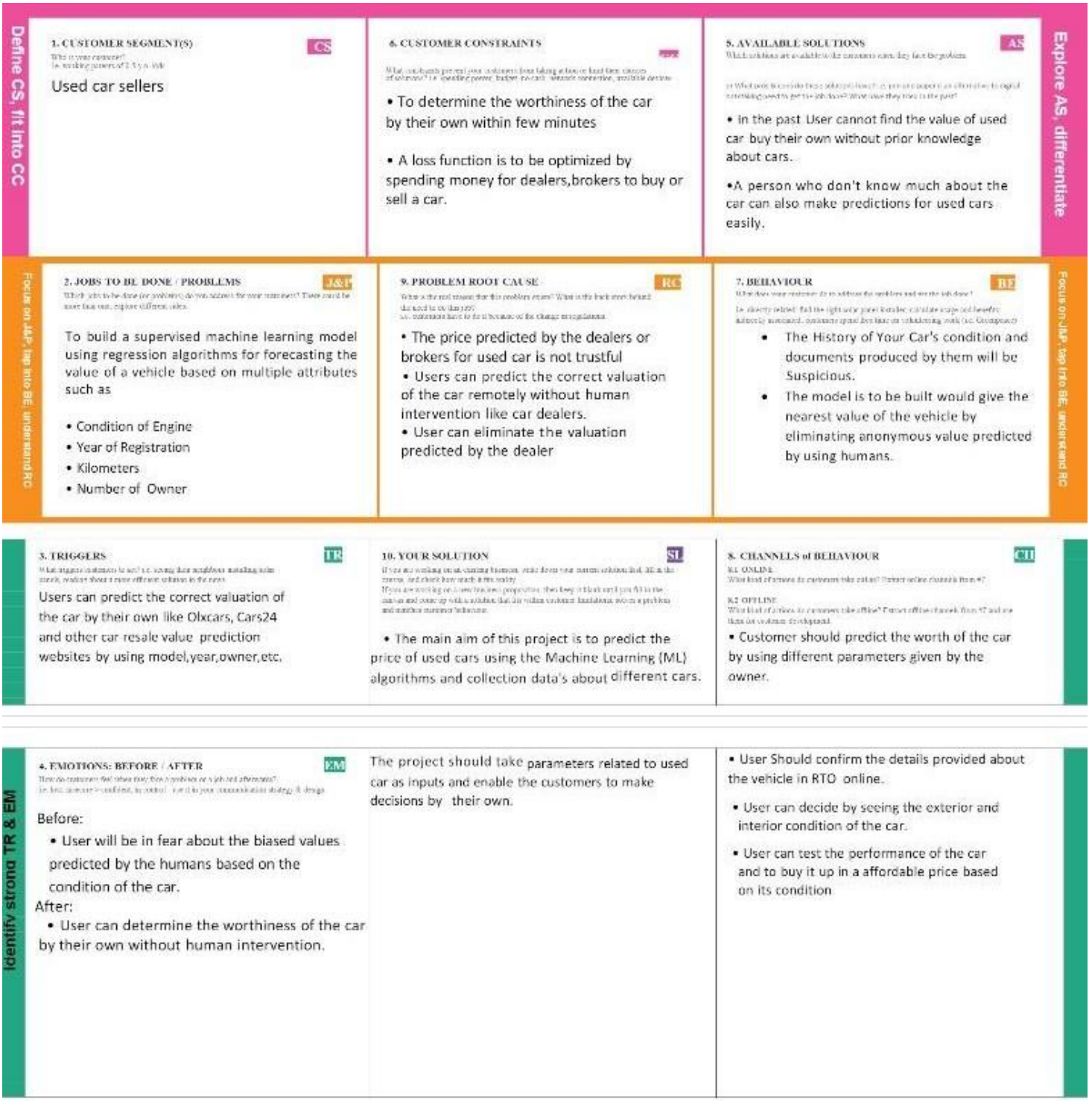



### *3.3 Proposed Solution*

| S.No | Parameter | Description |
|---|---|---|
| 1. | Problem Statement (Problem to be solved) | To develop a webpage to predict the resale value of a car |
| 2. | Idea / Solution description | To train the system with the dataset using a regression model |
| 3. | Novelty / Uniqueness | By using the optimal regression model to predict the value in a less amount to time and predict its value |

| | | |
|---|---|---|
| 4. | Social Impact / Customer Satisfaction | The customer can get an idea about the resale value of their car .to have an idea whether to sell their vehicle or not based on their financial condition |
| 5. | Business Model (Revenue Model) | The web based application has a friendly UI for the customer to enter their vehicles detail and the system predicts the value within few seconds |
| 6. | Scalability of the Solution | T he solution given by the trained system is efficient and is nearly accurate value of the vehicle. |

### *3.4* Proposed Solution fit



## *4* REQUIREMENT ANALYSIS

### 4.1. Functional Requirement

The functional Requirements of this projects involves the better understanding of

Pre-processing, Application designing using HTML & CSS and

IBM Watson Cloud. IBM Watson provides the services such as Database, deployment etc.

Hardware requirements Operating system- Windows 7,8,10

Software Requirements

- Python
- Pycharm
- PIP 2.7
- Jupyter Notebook
- Chrome

### 4.2. Non Functional Requirement

The Non - Functional Requirements of this project are,

- Highly accurate Image Predictive model
- better user responsive web application
- Cloud database for storing the informations

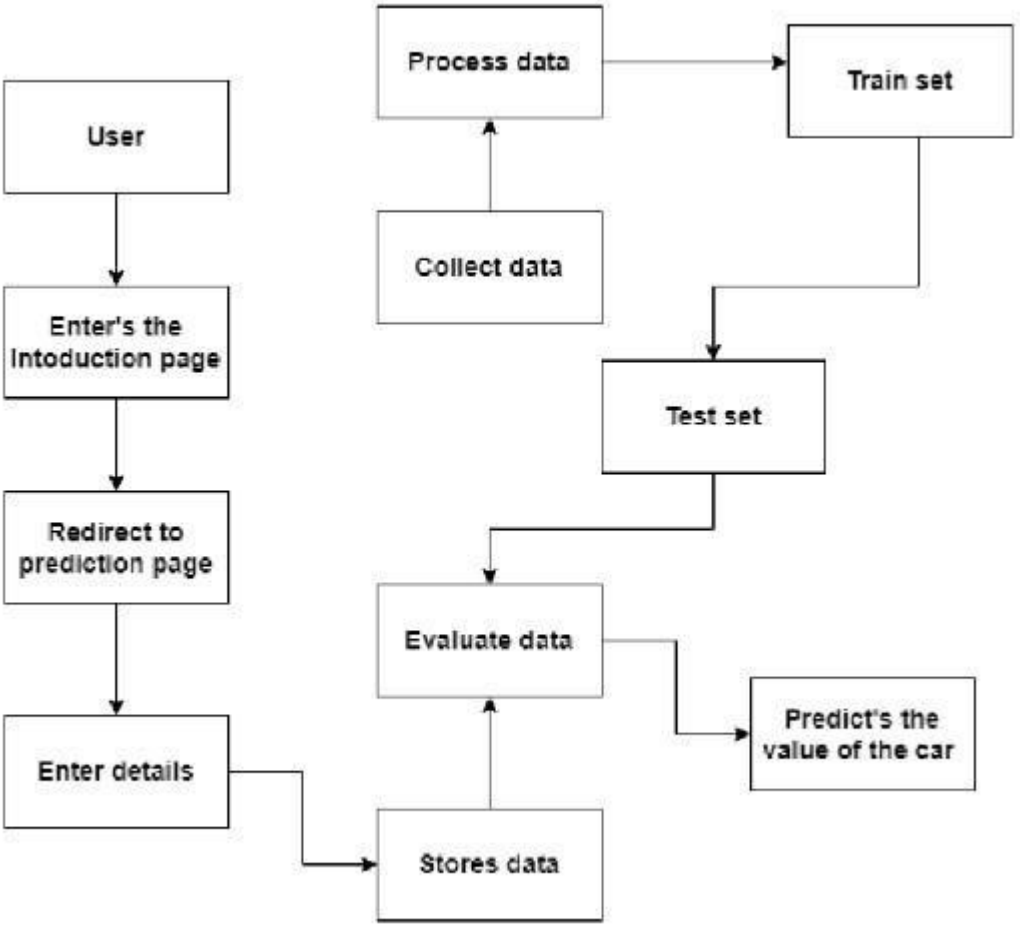
## 5. PROJECT DESIGN

### 5.1 Data Flow Diagrams

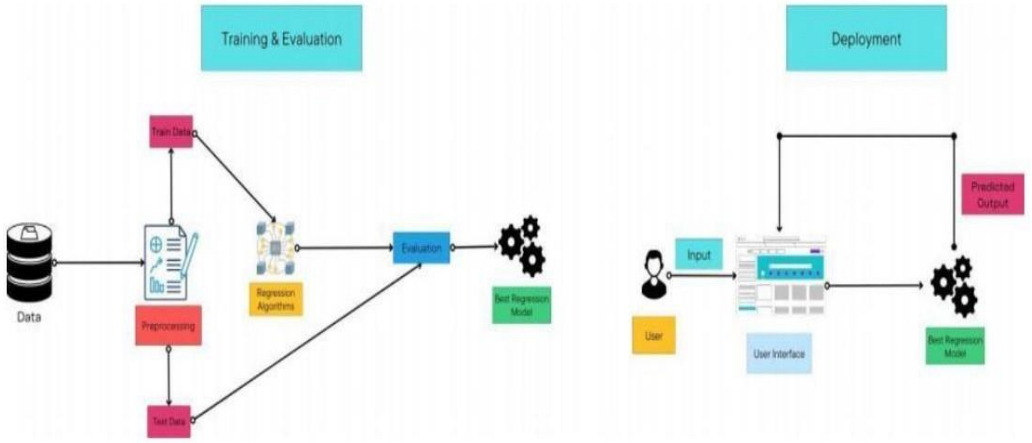
### 5.2 Solution & Technical Architecture

### 5.2 Solution & Technical Architecture

### 5.2 Solution & Technical Architecture




### 5.2 Solution & Technical Architecture



### 5.3 User Stories

| 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 | Vimal raj.k |
| Sprint-2 | Car resale value display | USN-2 | As a user, I can be redirected to the data entry page | 20 | Medium | Gurunivash.v.j |
| Sprint-3 | Data Entry | USN-3 | As a user, I can enter my car details in the re4quired fields. | 20 | Medium | Karthikeyan.s |
| Sprint-4 | Resale Value Prediction | USN-4 | As a user, I expect the application to predict the resale value of my car. | 20 | Medium | Parthasarathy.v |

## 6. PROJECT PLANNING

### 6.1 Sprint Planning and Estimation

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

## 6.2 Sprint Delivery Schedule

- Pre -requisites
- Import Required libraries
- Collect Data Set
- Pre the process the data
- Choose the Appropriate Model
- Train the model on IBM
- Integrate with Flask endpoint
- Index..html
- Registration form.html
- Flask application
- App.py

# 7. CODING & SOLUTION

## 7.1 Feature 1

### Import Libraries

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import pickle
from lightgbm import LGBMRegressor
```

### Import Dataset

In [2]:
```
df=pd.read_csv("Dataset.csv", header=0, sep=',', encoding='Latin1',)
```

### Read Dataset

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

Out[3]:

| | dateCrawled | name | seller | offerType | price | abtest | vehicleType | yearOfRegistration | gearbox | powerPS | model | kilometer | month |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 2016-03-24 11:52:17 | Golf_3_1.6 | privat | Angebot | 480 | test | NaN | 1993 | manuell | 0 | golf | 150000 | |
| 1 | 2016-03-24 10:58:45 | A5_Sportback_2.7_Tdi | privat | Angebot | 18300 | test | coupe | 2011 | manuell | 190 | NaN | 125000 | |
| 2 | 2016-03-14 12:52:21 | Jeep_Grand_Cherokee_"Overland" | privat | Angebot | 9800 | test | suv | 2004 | automatik | 163 | grand | 125000 | |

## 7.2 Feature 2

```
# Import Libraries
 import pandas as pd
 import numpy as np
 from flask import Flask, render_template, Response, request
 import pickle
 from sklearn.preprocessing import LabelEncoder

 app = Flask(__name__)#initiate flask app

 def load_model(file='model.sav'):#load the saved model
    return pickle.load(open(file, 'rb'))

 @app.route('/')
 def index():#main page
    return render_template('car.html')

 @app.route('/predict_page')
 def predict_page():#predicting page
     return render_template('value.html')

 @app.route('/predict', methods=['GET','POST'])
 def predict():
     reg_year = int(request.args.get('regyear'))
     powerps = float(request.args.get('powerps'))
     kms=  float(request.args.get('kms'))
     reg_month = int(request.args.get('regmonth'))

     gearbox = request.args.get('geartype')
     damage = request.args.get('damage')
     model = request.args.get('model')
     brand = request.args.get('brand')
     fuel_type = request.args.get('fuelType')
     veh_type = request.args.get('vehicletype')

     new_row = {'yearOfReg':reg_year, 'powerPS':powerps,
'kilometer':kms,
                'monthOfRegistration':reg_month, 'gearbox':gearbox,
                'notRepairedDamage':damage,
                'model':model, 'brand':brand, 'fuelType':fuel_type,
                'vehicletype':veh_type}

     print(new_row)

     new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',
        'powerPS','model','kilometer','monthOfRegistration','fuelType',
        'brand','notRepairedDamage'])
     new_df = new_df.append(new_row, ignore_index=True)
```

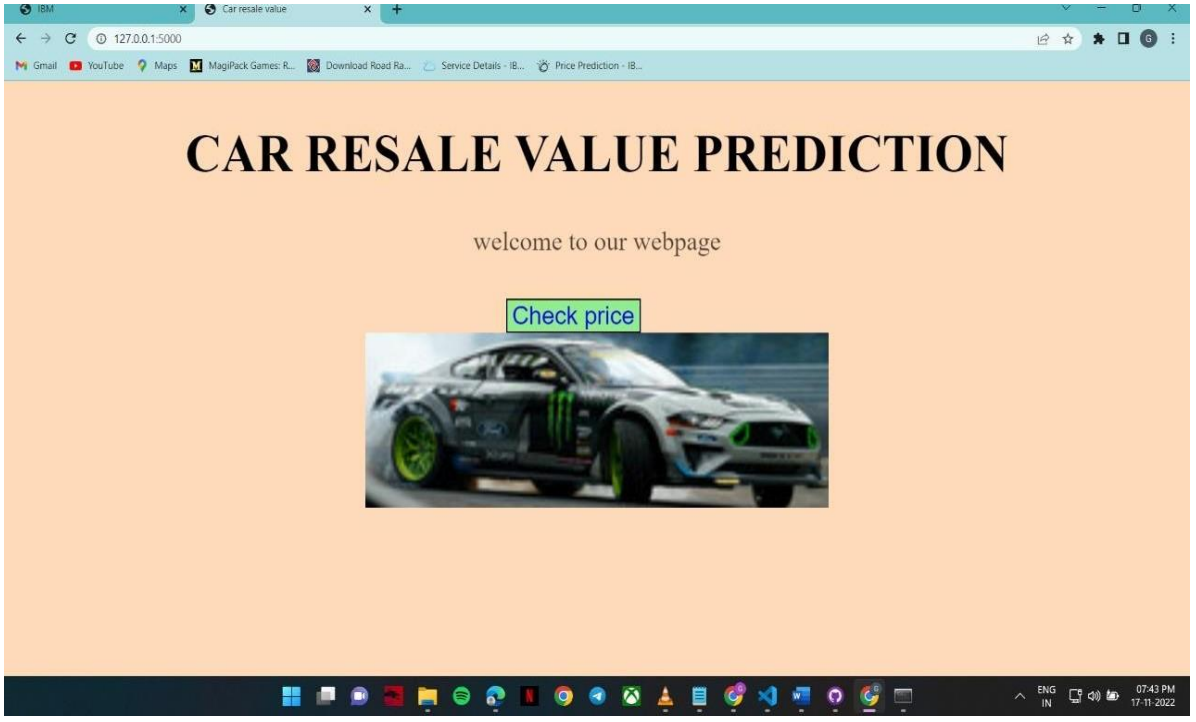
```
    labels =
['gearbox','notRepairedDamage','model','brand','fuelType','vehicletype']




    mapper = {}

    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].classes = np.load(str('classes'+i+'.npy'),
allow_pickle=True)
        transform = mapper[i].fit_transform(new_df[i])
        new_df.loc[:,i+'_labels'] = pd.Series(transform,
index=new_df.index)
    labeled =
new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] +
[x+'_labels' for x in labels]]

    X = labeled.values.tolist()
    print('\n\n', X)
    predict = reg_model.predict(X)


    #predict = predictions['predictions'][0]['values'][0][0]
    print("Final prediction :",predict)

    return render_template('predict.html',predict=predict)

if__name__=='__main__':
    reg_model = load_model()#load the saved model
    app.run(debug=True)
```

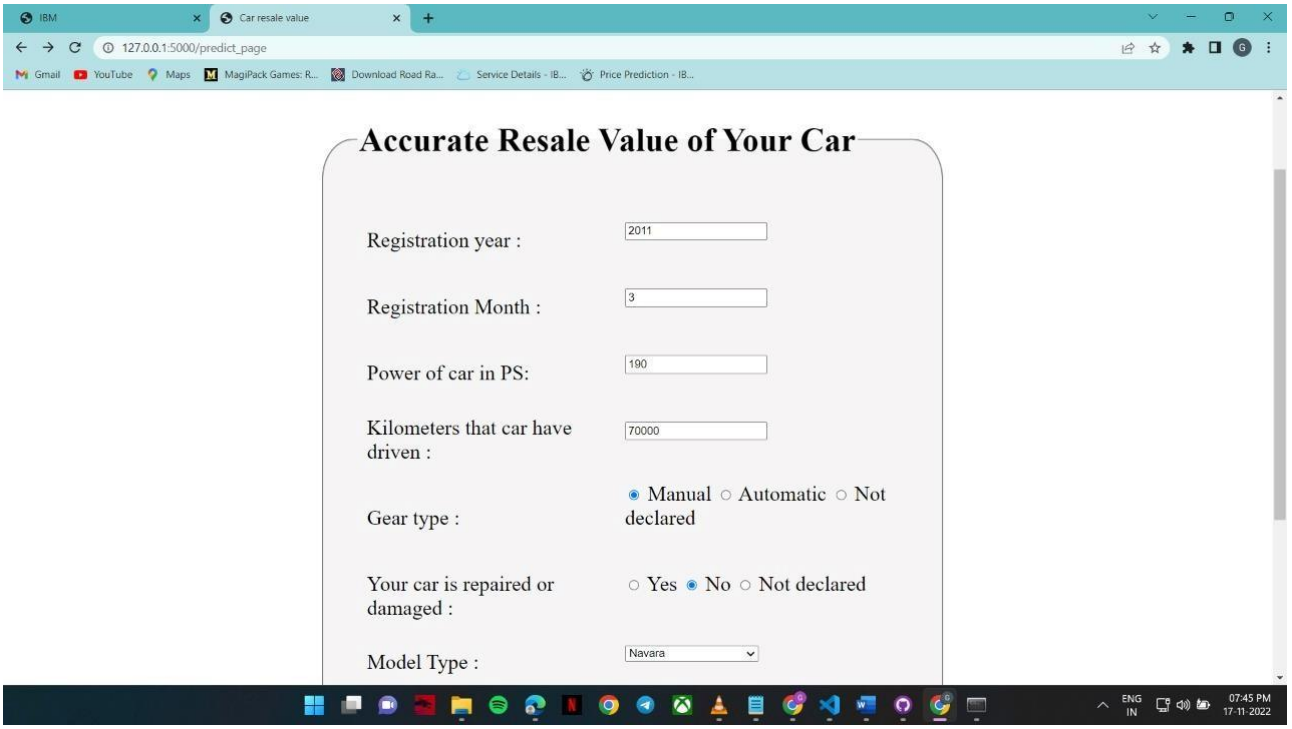
## 8. TESTING

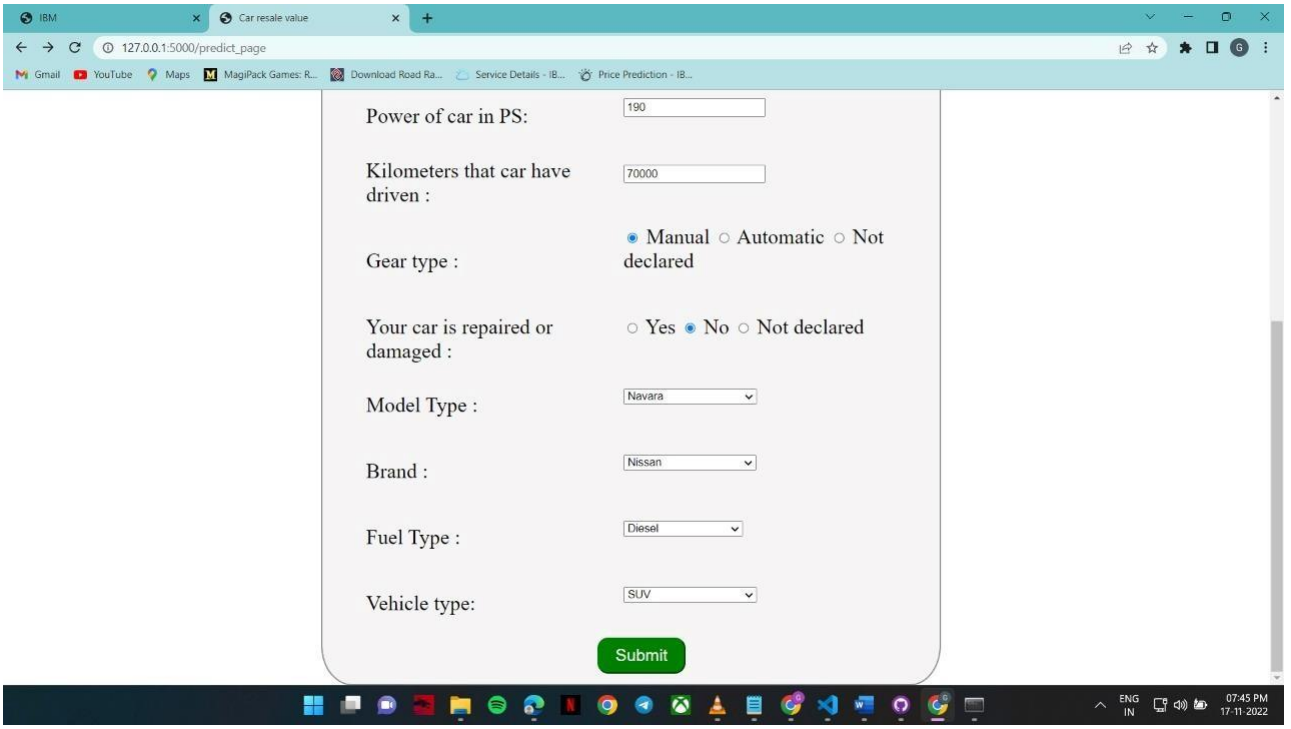
### 8.1. Test Cases

1. User Login and Registration test
2. Database Update test
3. Prediction test

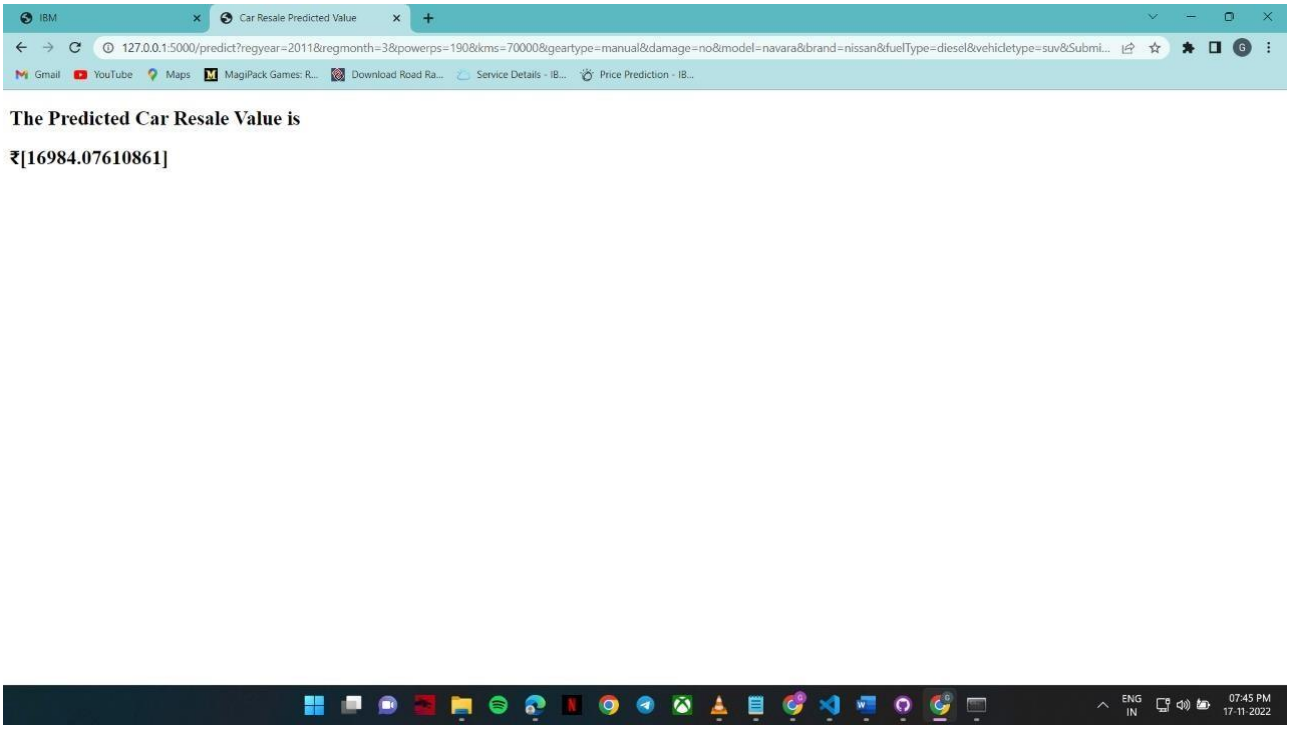
### 8.2. User Acceptance Testing






The login web page is tested with the invalid user information to check the invalid login testing into the webpage.




# Estimated Car resale value is 16984.07610861

## 9. PERFORMANCE

### 9.1 Performance metrics

**{'mae': 1325.112086905962,**

**'mse': 9577053.62710202,**

**'rmse': 3094.6815065692977,**

**'rmsle': 8.03744027403009,**

**'r2': 0.8661221626879432,**

**'adj_r2_score': 0.8661152969113608}**

The model is tested with the various damaged car

images which is not used during the training and validation of the model which also

shows that the model works with the accuracy of about 98% in the overall performance

## 10. ADVANTAGES AND DISADVANTAGES

- To develop an efficient and effective model which predicts the price of a used car according to the user's inputs and achieve good accuracy.

**CONS:**

- Less effective

## 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.

## 12. FUTURE SCOPE

In future this machine learning model may bind with various websites 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 a user interface for interacting with users. For better performance, we plan to judiciously design deep learning network structures, use adaptive learning rates and train on clusters of data rather than the whole dataset.

**Github Repo: https://github.com/IBM-EPBL/IBM-Project-25568-1659967505**

**VideoLink: https://drive.google.com/drive/folders/1mZi3E2JDHpmpEwhQLnEbNj046wxJNTtr?usp=sharing**

## 13. APPENDIX

**App.py**

```
# Import Libraries
import pandas as pd
import numpy as np
from flask import Flask, render_template, Response, request
import pickle
from sklearn.preprocessing import LabelEncoder

app = Flask(__name__)#initiate flask app

def load_model(file='model.sav'):#load the saved model
        return pickle.load(open(file, 'rb'))

@app.route('/')
def index():#main page
        return render_template('car.html')
```

```
@app.route('/predict_page')
def predict_page():#predicting page
        return render_template('value.html')


@app.route('/predict', methods=['GET','POST'])
def predict():
        reg_year = int(request.args.get('regyear'))
        powerps = float(request.args.get('powerps'))
        kms=  float(request.args.get('kms'))
        reg_month = int(request.args.get('regmonth'))

        gearbox = request.args.get('geartype')
        damage = request.args.get('damage')
        model = request.args.get('model')
        brand = request.args.get('brand')
        fuel_type = request.args.get('fuelType')
        veh_type = request.args.get('vehicletype')

        new_row = {'yearOfReg':reg_year, 'powerPS':powerps, 'kilometer':kms,
                                'monthOfRegistration':reg_month, 'gearbox':gearbox,
                                'notRepairedDamage':damage,
                                'model':model, 'brand':brand, 'fuelType':fuel_type,
                                'vehicletype':veh_type}

        print(new_row)

        new_df = pd.DataFrame(columns=['vehicletype','yearOfReg','gearbox',
                'powerPS','model','kilometer','monthOfRegistration','fuelType',
                'brand','notRepairedDamage'])
        new_df = new_df.append(new_row, ignore_index=True)
        labels = ['gearbox','notRepairedDamage','model','brand','fuelType','vehicletype']
        mapper = {}
```

```
        for i in labels:
                mapper[i] = LabelEncoder()
                mapper[i].classes = np.load(str('classes'+i+'.npy'), allow_pickle=True)
                transform = mapper[i].fit_transform(new_df[i])
                new_df.loc[:,i+'_labels'] = pd.Series(transform, index=new_df.index)
        labeled = new_df[['yearOfReg','powerPS','kilometer','monthOfRegistration'] +
[x+'_labels' for x in labels]]


        X = labeled.values.tolist()
        print('\n\n', X)
        predict = reg_model.predict(X)



        #predict = predictions['predictions'][0]['values'][0][0]
        print("Final prediction :",predict)


        return render_template('predict.html',predict=predict)


if__name_=='__main__':
        reg_model = load_model()#load the saved model
        app.run(debug=True)
```