CAR RESALE VALUE PREDICTION

1. INTRODUCTION

1.1 PROJECT OVERVIEW:

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. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy.

1.2 PURPOSE:

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.

2. LITERATURE SURVEY

Title: Used Cars Price Prediction using Supervised Learning Techniques **Author:** Mukkesh Ganesh , Pattabiraman Venkatasubbu
Year: December 2019
Abstract:

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.

Title: Used Cars Price Prediction and Valuation using Data Mining Techniques
Author: Abdulla AlShared
Year: December 2021
Abstract:

Due to the unprecedented number of cars being purchased and sold, used car price prediction is a topic of high interest. Because of the affordability of used cars in developing countries, people tend more purchase used cars. A primary objective of this project is to estimate used car prices by using attributes that are highly correlated with a label (Price). To accomplish this, data mining technology has been employed. Null, redundant, and missing values were removed from the dataset during pre-processing. In this supervised learning study, three regressors (Random Forest Regressor, Linear Regression, and Bagging Regressor) have been trained, tested, and compared against a benchmark dataset. Among all the experiments, the Random Forest Regressor had the highest score at 95%, followed by 0.025 MSE, 0.0008 MAE, and 0.0378 RMSE respectively. In addition to Random Forest Regression, Bagging Regression performed well with an 88% score, followed by Linear Regression having an 85% mark. A train-test split of 80/20 with 40 random states was used in all experiments. The researchers of this project anticipate that in the near future, the most sophisticated algorithm is used for making predictions, and then the model will be integrated into a mobile app or web page for the general public to use.

Title: Used Car Price Prediction
Author: Praful Rane, Deep Pandya, Dhawal Kotak
Year: April 2021
Abstract:

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.

Title: Car Price Prediction Using Machine Learning
Author: Ketan Agrahari, Ayush Chaubey, Mamoor Khan, Manas Srivastava **Year:** June 2021
Abstract:

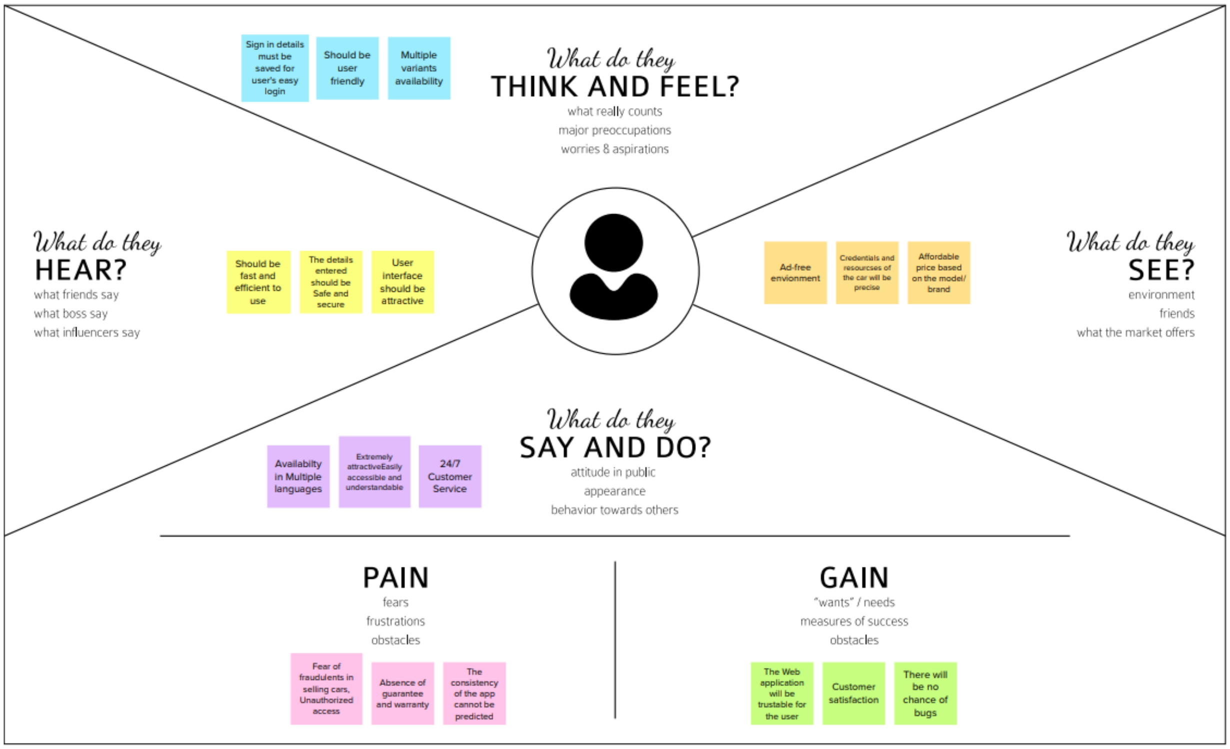
The demand for used cars has increased significantly in the past decade and it is prognosticated that with Covid-19 outbreak this requirement will augment considerably. Hence to enhance the reliability, with the expansion of the used car market, a model that can forecast the current market price of a used automobile on the basis of a variety of criteria. This analysis can be used to study the trends in the industry, offer better insight into the market, and aid the community in its smooth workflow. The aim of this research paper is to predict the car price as per the data set (previous consumer data like engine capacity, distance traveled, year of manufacture, etc.). The result of these algorithms will be analyzed and based on the efficiency and accuracy of these algorithms, the best one of them can be used for the said purpose.

Title: Vehicle Price Prediction using SVM Techniques
Author: S.E.Viswapriya, Durbaka Sai Sandeep Sharma, Gandavarapu Sathya kiran
Year: June 2020
Abstract:

The prediction of price for a vehicle has been more popular in research area, and it needs predominant effort and information about the experts of this particular field. The number of different attributes is measured and also it has been considerable to predict the result in more reliable and accurate. To find the price of used vehicles a well defined model has been developed with the help of three machine learning techniques such as Artificial Neural Network, Support Vector Machine and Random Forest. These techniques were used not on the individual items but for the whole group of data items. This data group has been taken from some web portal and that same has been used for the prediction. The data must be collected using web scraper that was written in PHP programming language. Distinct machine learning algorithms of varying performances had been compared to get the best result of the given data set. The final prediction model was integrated into Java application.

3. IDEATION & PROPOSED SOLUTION

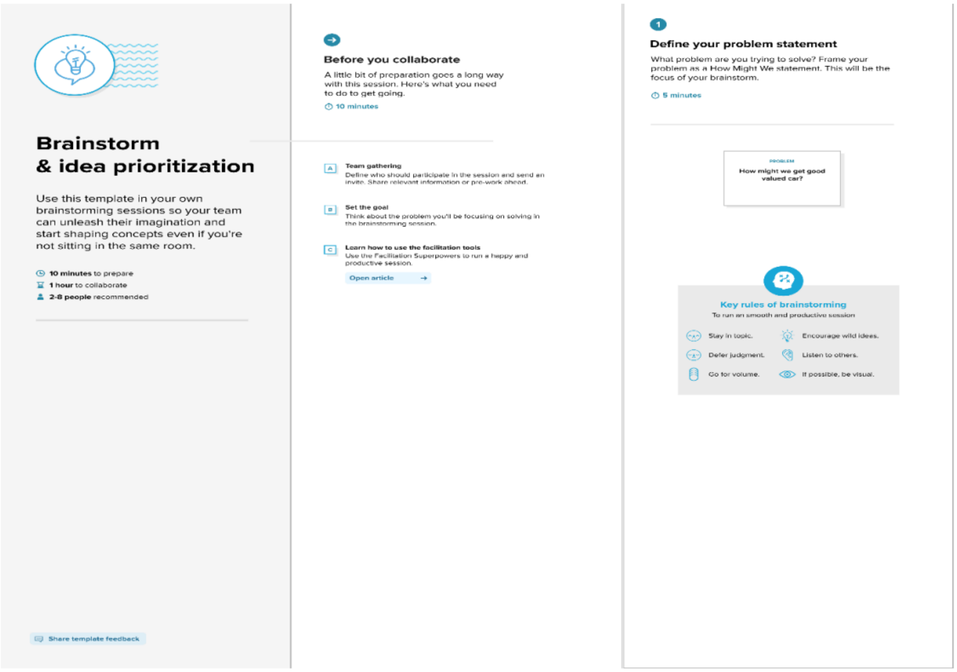
3.1 EMPATHY MAP CANVAS:


Sign in details must be saved for user's easy login
Should be user friendly
Multiple variants availability
What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations
What do they
HEAR?
what friends say
what boss say
what influencers say
Should be fast and efficient to use
The details entered should be Safe and secure
User interface should be attractive
Ad-free envionment
Credentials and resourcses of the car will be precise
Affordable price based on the model/ brand
What do they
SEE?
environment
friends
what the market offers
Availabilty in Multiple languages
Extremely attractiveEasily accessible and understandable
24/7 Customer Service
What do they
SAY AND DO?
attitude in public
appearance
behavior towards others
PAIN
fears
frustrations
obstacles
Fear of fraudulents in selling cars, Unauthorized access
Absence of guarantee and warranty
The consistency of the app cannot be predicted
GAIN
"wants" / needs
measures of success
obstacles
The Web application will be trustable for the user
Customer satisfaction
There will be no chance of bugs

3.2 IDEATION & BRAINSTORMING:

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions. Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

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



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

Share template feedback

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A **Team gathering**
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B **Set the goal**
Think about the problem you'll be focusing on solving in the brainstorming session.

C **Learn how to use the facilitation tools**
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

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.

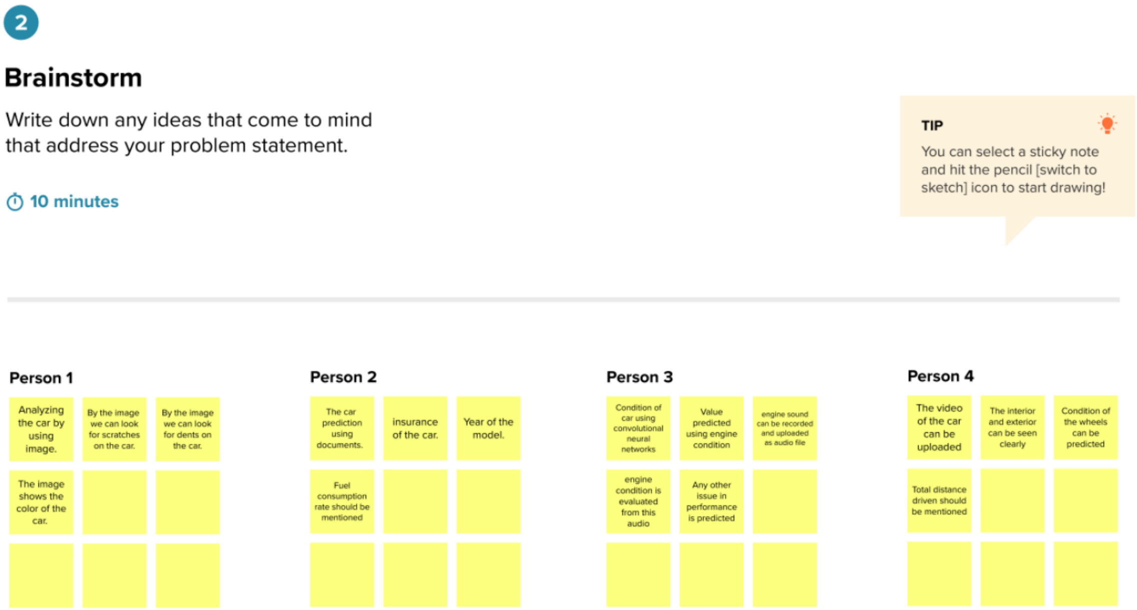
5 minutes

PROBLEM
How might we get good valued car?

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.

Step-2: Brainstorm



2

Brainstorm

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

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Person 1

Analyzing the car by using image.
By the image we can look for scratches on the car.
By the image we can look for dents on the car.
The image shows the color of the car.

Person 2

The car prediction using documents.
insurance of the car.
Year of the model.
Fuel consumption rate should be mentioned

Person 3

Condition of car using convolutional neural networks
Value predicted using engine condition
engine sound can be recorded and uploaded as audio file
engine condition is evaluated from this audio
Any other issue in performance is predicted

Person 4

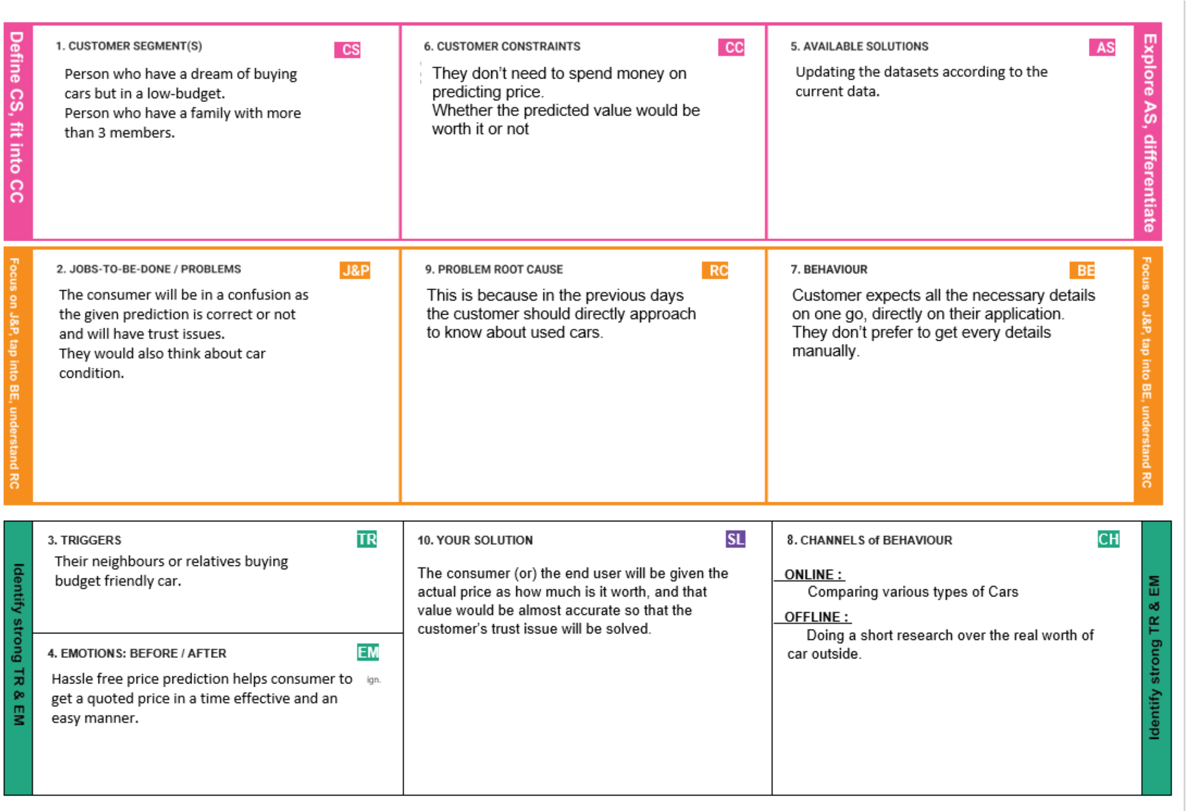
The video of the car can be uploaded
The interior and exterior can be seen clearly
Condition of the wheels can be predicted
Total distance driven should be mentioned

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	**Car Resale Value Prediction** 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. Thus, it is of commercial interest to sellers/financers to be able to predict the salvage value (residual value) of cars with accuracy.
2.	Idea / Solution description	Using regression algorithms, we proposed an intelligent, flexible, and effective system to predict the value of the car. By regression algorithms and other algorithms is used to predict the accuracy value of the cars. Depend on major parts and damages on the car will affect the price of the car.
3.	Novelty / Uniqueness	To predict the value, the most essential elements for forecast are brand and model, period use of vehicle, mileage of vehicle, gear type and fuel type utilized in the vehicle just as fuel utilization per mile profoundly influence cost of a vehicle because of continuous changes in the cost of a fuel. By forecasting the above details, AI can predict the value accurately.

4.	Social Impact / Customer Satisfaction	Customer Satisfaction plays a vital role in this, i.e for customer, he/she need to get profit from his car so customer expect that the predict value need to be good which gives him/her profit, but it is depend on the car condition. Depend on the customer satisfaction our application will create a social impact and may customer will increase.
5.	Business Model (Revenue Model)	A Revenue model is a framework for generating financial income. It identifies which revenue source to pursue ,what value to offer ,how to price the value ,and who pays for the value.
6.	Scalability of the Solution	The value of the car is predicting by using different regression algorithms like linear regression , random forest regression , decision tree regression and so on. Thus the car will got accurate price. Those algorithms gives the results with the user given details about the car, but the best and approximate result is got by random forest algorithm. As random forest regression algorithm gives more as 15% then other algorithms.

3.4 PROBLE SOLUTION FIT:



Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Person who have a dream of buying cars but in a low-budget.
Person who have a family with more than 3 members.

6. CUSTOMER CONSTRAINTS CC

They don't need to spend money on predicting price.
Whether the predicted value would be worth it or not

5. AVAILABLE SOLUTIONS AS

Updating the datasets according to the current data.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

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

The consumer will be in a confusion as the given prediction is correct or not and will have trust issues.
They would also think about car condition.

9. PROBLEM ROOT CAUSE RC

This is because in the previous days the customer should directly approach to know about used cars.

7. BEHAVIOUR BE

Customer expects all the necessary details on one go, directly on their application.
They don't prefer to get every details manually.

Focus on J&P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS TR

Their neighbours or relatives buying budget friendly car.

4. EMOTIONS: BEFORE / AFTER EM

Hassle free price prediction helps consumer to get a quoted price in a time effective and an easy manner. ign.

10. YOUR SOLUTION SL

The consumer (or) the end user will be given the actual price as how much is it worth, and that value would be almost accurate so that the customer's trust issue will be solved.

8. CHANNELS of BEHAVIOUR CH

ONLINE :
Comparing various types of Cars

OFFLINE :
Doing a short research over the real worth of car outside.

Identify strong TR & EM

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS:

Operating system- Windows 7,8,10
Processor- dual core 2.4 GHz (i5 or i7 series Intel processor or equivalent AMD)
RAM-4GB

4.2 NON-FUNCTIONAL REQUIREMENTS:

Python Pycharm
PIP 2.7
Jupyter Notebook
Chrome

5. PROJECT DESIGN

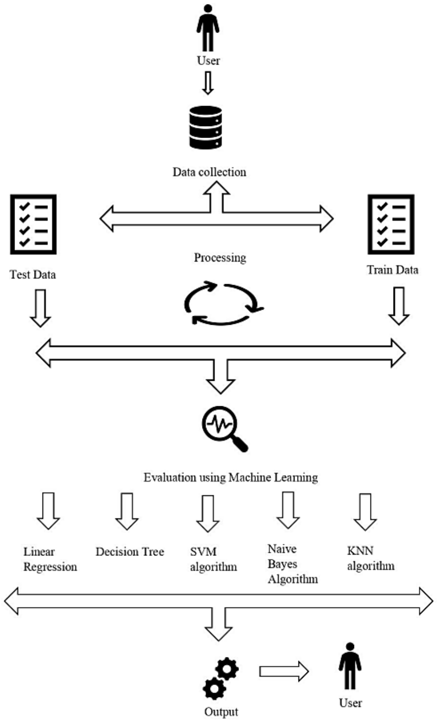
5.1 DATA FLOW DIAGRAMS:

A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored. Example: (Simplified)

5.2 SOLUTION & TECHNICAL ARCHITECTURE:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

- Find the best tech solution to solve existing business problems.
- Describe the structure, characteristics, behavior, and other aspects of the software to project stakeholders.



5.3 USER STORIES:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-3	As a user, I can register for the application through Google, Facebook.	I can register & access the dashboard with Google, Facebook Login	Low	Sprint-2
	Login	USN-4	As a user, I can log into the application by entering email & password	I can login using email and password	High	Sprint-1
	Dashboard	USN-5	As a user, I can access the dashboard after login	I can access the dashboard	High	Sprint-2

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	High	Sprint-1
		USN-7	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
		USN-8	As a user, I can register for the application through Google, Facebook.	I can register & access the dashboard with Google, Facebook Login	Low	Sprint-2
	Login	USN-9	As a user, I can log into the application by entering email & password	I can login using email and password	High	Sprint-1
	Dashboard	USN-10	As a user, I can access the dashboard after login	I can access the dashboard	High	Sprint-2
Customer Care Executive	Customer Support	USN-11	As a user, I can contact the customer care and chat with us	I can contact the customer care and chat with the person in-charge	High	Sprint-2
Administrator		USN-12	As a user, my data is maintained by admin	Admin maintain customer data	High	Sprint-2

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION:

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password.	10	High	Sivaguru, Rajeshkanth
Sprint-1	Confirmation	USN-2	As a user, I will receive confirmation email once I have registered for the application	10	High	Suresh, Rajeshkanth
Sprint-1	Login	USN-3	As a user, I can log into the application by entering email & password	8	Medium	Sivaguru, Rajeshkanth
Sprint-2	Dataset	USN-4	Collect dataset, Import required libraries, Test and Train data.	10	High	Shamsheer ahamed, Sivaguru
Sprint-2	Algorithm	USN-5	Apply Regression algorithm and got the data (.pkl	10	High	Suresh, Shamsheer ahamed

			file).			
Sprint-3	Dashboard	USN-6	HTML page contains Login, Details to be entered to predict the car price and a customer support.	10	High	Shamsheer ahamed, Rajeshkanth
Sprint-4	Building application	USN-7	Build python flask application	10	High	Suresh, Shamsheer ahamed

6.2 SPRINT DELIVERY SCHEDULE:

Sprint	**Total Story Points**	**Duration**	**Sprint Start Date**	**Sprint End Date (Planned)**	**Story Points Completed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	20	3 Days	08 Nov 2022	10 Nov 2022	20	11 Nov 2022
Sprint-2	20	3 Days	11 Nov 2022	13 Nov 2022	20	14 Nov 2022
Sprint-3	20	2 Days	14 Nov 2022	15 Nov 2022	20	16 Nov 2022
Sprint-4	20	2 Days	16 Nov 2022	17 Nov 2022	20	17 Nov 2022

7. CODING & SOLUTIONING

7.1 FEATURE 1:

Using the following code we have created the categories and included text boxes for getting input from the user, also in some categories the user will be able to select an option from the drop down list.

```
    <tr>
    <td><label for="month">Registration Month : </label></td>
    <td><input id="month" maxlength="50" name="regmonth" type="text" />
    <br>
    <br>
    </td>
    </tr>
  <tr>
    <td><label for="year" padding:10px>Registration year : </label></td>
    <td><input id="year" maxlength="50" name="regyear" type="text" />
    <br>
    <br>
    </td>
    </tr>
```

The following code is for getting kilometres driven information and power and other information about the car.

```
    <tr>
    <td><label for="kilometer">Kilometers driven : </label></td>
    <td><input id="kilometer" maxlength="50" name="kms" type="text" />
    <br>
    <br>
    </td>
    </tr>
  <tr>
```

```
    <td><label for="power">Car power in PS: </label></td>
    <td><input id="power" maxlength="50" name="powerps" type="text" />
    <br>
    <br>
    </td>
    </tr>

    <tr>
    <td><label for="geartype">Gear type : </label></td>
    <td><input type="radio" name="geartype" value="manual"/> Manual
    <input type="radio" name="geartype" value="automatic"/> Automatic
    <input type="radio" name="geartype" value="not-declared"/> Not declared
    <br>
    <br>
    </td>
    </tr>
```

In the next code we have included a bunch of car models list and their brands list.

```
    <tr>
    <td><label for="model">Model Type : </label></td>
    <td>
    <select name="model" id="model">
    <option value="" disabled selected hidden>Choose Model Name...</option>
 <option value="golf">Golf </option>
    <option value="grand">Grand </option>
    …...........................................................
    …...........................................................
    <option value="serie_1">Serie 1 </option>
    <option value="discovery_sport">Discovery Sport </option>
    </select>
    <br>
    <br>
    </td>
    </tr>
```

And For Brands:

```
    <tr>
    <td><label for="brand">Brand :</label></td>
    <td>
```

```
<select name="brand" id="brand">
<option value="" disabled selected hidden>Choose Brand Name...</option>
<option value="volkswagen">Volkswagen </option>
<option value="land_rover">Land Rover </option>
<option value="lada">Lada </option>
</select>
<br>
<br>
</td>
</tr>
```

7.2 FEATURE 2:

In our project we have created an interactive design of car resale value prediction using advanced data science. We have Label encoded the categorical data.

```
from sklearn.preprocessing import LabelEncoder
 labels = ['gearbox','notRepairedDamage','model','brand','fuelType','vehicleType']
 mapper = {}
for i in labels:
    mapper[i] = LabelEncoder()
    mapper[i].fit(new_df[i])
    tr = mapper[i].transform(new_df[i])
    np.save(str('classes'+i+'.npy'), mapper[i].classes_)
    print(i,":",mapper[i])
    new_df.loc[:,i+'_labels'] = pd.Series(tr, index=new_df.index)

labeled = new_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']
                + [x+"_labels" for x in labels]]
 print(labeled.columns)
```

8. TESTING

8.1 TEST CASES:

In order to understand what affects change in price of a used car, the relation between features available in the data sat will be examined by using inferential statistic methods. The primary assumption based on figures and tables is price must be affected by odometer and condition. There must be other features that affects price significantly. It will be investigated in the later phase of the study.

Checking normality: For checking normality, q-q plot helps us. Figure 9 tells that there is a violation of normality. This means that the data points that are used are not distributed normally. In addition, Shapiro-Wilk test was performed for checking normality.

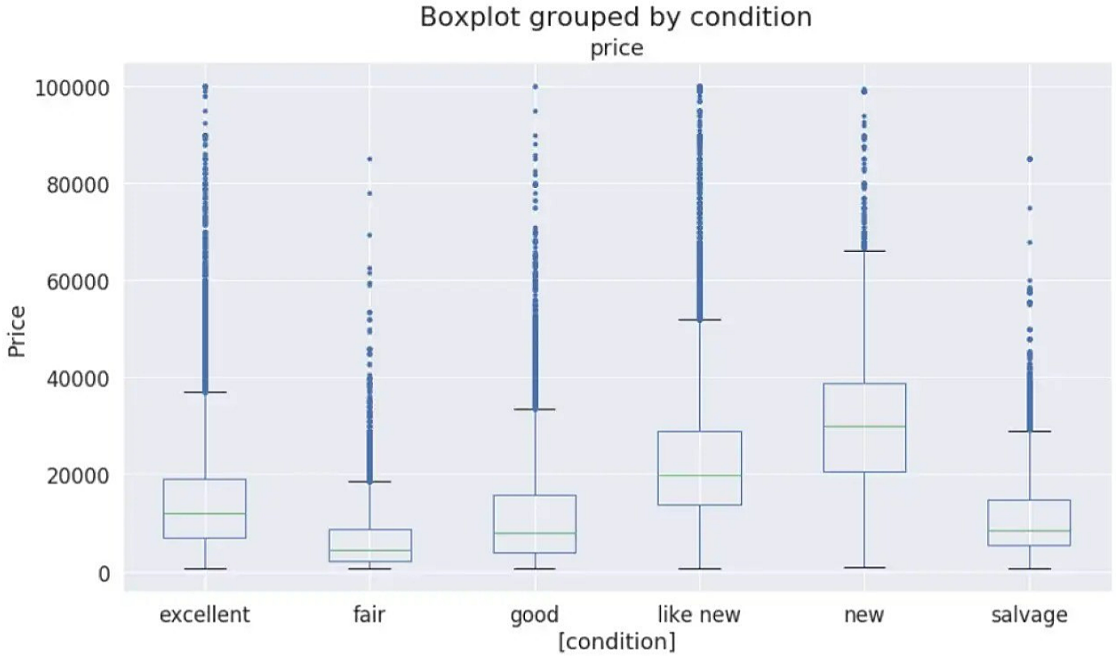
Result:(0.9586305022239685, 0.0)

Here, the first value is W-test statistic and the second value is the p-value. For N > 5000, the W test statistic is accurate but the p-value may not be. By considering p-value of Shapiro-Wilk test, it can be concluded that the data is not normally distributed.

In this situation, we have problem with initial data points. May be, filtering data can solve this issue. For this purpose, the values of odometer and price that are two standard deviation away from mean were dropped and independent t-test applied.

Condition vs Price

The second hypothesis of this study focuses on effect of a car's condition on its price. In order to understand this relation, Table 6 and Figure 6 can be useful. By looking at Figure 10, it can be said that 'condition' effects median price of cars seriously. On the other hand, there are a lot of outliers in the condition values which is an expected result for such a lar dataset. We do not see outliers at the bottom of the Figure 10. This is mostly because during data cleaning, cars that lower than $750 price were dropped.



9. RESULTS

9.1 PERFORMANCE METRICS:

Performance metrics are a collection of data that employers evaluate against an established objective. It is important to note the difference between a performance metric and a key performance indicator.

We have used the Random Forest Regression method to evaluate the performance metrics.

We use the following code for it:

```
regressor = RandomForestRegressor(n_estimators=1000, max_depth=10,random_state=34)
#fitting the model
regressor.fit(X_train, np.ravel(Y_train,order='C'))
```

And we get:

```
RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

By considering all four metrics, it can be concluded that random forest the best model for the prediction for used car prices. Random Forest as a regression model gave the best.

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- It is very easy to use.
- It contains all the available models and their predictions.
- It gives almost 90% accurate prediction.
- No user is asked for their personal details.
- Can be very useful for people who are going to buy or sell used cars.

DISADVANTAGES:

- They ask for so many data about the cars.
- We have to know everything about the car precisely.

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 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 rather than the whole dataset.

13. APPENDIX

Source Code:

Multi Linear Regression.ipynb

```
import sklearn
sklearn.__version__
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data = pd.read_csv("autos.csv", header=0, sep=',', encoding='Latin1',)
data.head()
data.info()
data.describe()
data.isnull().any()
data.isnull().sum()
# filling the missing values with mean

data["vehicleType"].fillna(data["vehicleType"].mode()[0],inplace = True)
data["gearbox"].fillna(data["gearbox"].mode()[0],inplace = True)
data["model"].fillna(data["model"].mode()[0],inplace = True)
data["fuelType"].fillna(data["fuelType"].mode()[0],inplace = True)
data["notRepairedDamage"].fillna(data["notRepairedDamage"].mode()[0],inplace = True)
data.info()
print(data.offerType.value_counts())
data[data.offerType != 'Gesuch']
data=data.drop('offerType',1)
data[data.seller != 'gewerblich']
data = data.drop('seller',1)
print(data.shape)
data = data[(data.powerPS>50)&(data.powerPS<900)]
print(data.shape)
data = data[(data.yearOfRegistration >= 1950) & (data.yearOfRegistration <2017)]
print(data.shape)
```

```
new_df = data.copy()
new_df = new_df.drop_duplicates(['price','vehicleType','yearOfRegistration','gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType','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 = new_df[(new_df.price >= 100) & (new_df.price <= 150000)]

new_df['notRepairedDamage'].fillna(value='not-declared',inplace=True)
new_df['fuelType'].fillna(value='not-declared',inplace=True)
new_df['gearbox'].fillna(value='not-declared',inplace=True)
new_df['vehicleType'].fillna(value='not-declared',inplace=True)
new_df['model'].fillna(value='not-declared',inplace=True)


from sklearn.preprocessing import LabelEncoder

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

mapper = {}
for i in labels:
    mapper[i] = LabelEncoder()
    mapper[i].fit(new_df[i])
    tr = mapper[i].transform(new_df[i])
    np.save(str('classes'+i+'.npy'), mapper[i].classes_)
    print(i,":",mapper[i])
    new_df.loc[:,i+'_labels'] = pd.Series(tr, index=new_df.index)

labeled = new_df[['price','yearOfRegistration','powerPS','kilometer','monthOfRegistration']
                + [x+"_labels" for x in labels]]

print(labeled.columns)

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

```
X = labeled.iloc[:,1:].values
X.shape
Y.shape
Y = Y.reshape(-1,1)

from sklearn.model_selection import cross_val_score, train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.3, random_state = 3)
#model building
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
regressor = RandomForestRegressor(n_estimators=1000, max_depth=10,random_state=34)
#fitting the model
regressor.fit(X_train, np.ravel(Y_train,order='C'))

#predicting the values to test set
y_pred =regressor.predict(X_test)

#printing the Accuracy
print(r2_score(Y_test,y_pred))
#plot for predicted and actual price
plt.figure(figsize=(10,5))
plt.plot(y_pred[0:20])
plt.plot(np.array(Y_test[0:20]))
plt.legend(["predicted","actual"])
plt.show()
#saving the model
import pickle
filename = 'resale_model.sav'
pickle.dump(regressor, open(filename, 'wb'))
```

Car.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>Car resale value </title>
    <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">
      <center>
  <div class="container-text">

</div>
</center>

        <div class="text-box" style= "text-align:left;">
          <h1>Car resale value Predictor</h1> <br><br><br>
          <pre class="tab1">
          <a href="./predict_page" class="visit-btn ">Check price</a>
         </pre>
        </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 Predicted Value</title>

</head>
<body>

            <section class="header">

        <div class="text-box">
<br><br><br><br><br><br><br><br><br>
          <h1>The Predicted Value of the car: </h1>
                          <h1>{{predict}}</h1>
        </div>
    </section>


</body>
</html>
```

Value.html

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head>
<link rel="stylesheet" href="../static/css/value.css">
<title>Car resale value</title>
</head>
<body>
<center>
      <section class="form">
      <form action="http://localhost:5000/predict" method="GET">
  <table border="0" align="center">
  <tbody>
            <h3>Calculate Resale Value of Your Car</h3>


      <tr>
      <td><label for="month">Registration Month : </label></td>
      <td><input id="month" maxlength="50" name="regmonth" type="text" />
      <br>
      <br>
      </td>
      </tr>
    <tr>
      <td><label for="year" padding:10px>Registration year : </label></td>
      <td><input id="year" maxlength="50" name="regyear" type="text" />
      <br>
      <br>
      </td>
      </tr>



      <tr>
      <td><label for="kilometer">Kilometers driven : </label></td>
      <td><input id="kilometer" maxlength="50" name="kms" type="text" />
```

```
        <br>
        <br>
        </td>
        </tr>
     <tr>
        <td><label for="power">Car power in PS: </label></td>
        <td><input id="power" maxlength="50" name="powerps" type="text" />
        <br>
        <br>
        </td>
        </tr>

        <tr>
        <td><label for="geartype">Gear type : </label></td>
        <td><input type="radio" name="geartype" value="manual"/> Manual
        <input type="radio" name="geartype" value="automatic"/> Automatic
        <input type="radio" name="geartype" value="not-declared"/> Not declared
        <br>
        <br>
        </td>
        </tr>

        <tr>
        <td><label for="model">Model Type : </label></td>
        <td>
        <select name="model" id="model">
        <option value="" disabled selected hidden>Choose Model Name...</option>
<option value="golf">Golf </option>
        <option value="grand">Grand </option>
        <option value="fabia">Fabia </option>
        <option value="3er">3er </option>
        <option value="2_reihe">2 Reihe </option>
        <option value="andere">Andere </option>
        <option value="c_max">C Max </option>
        <option value="3_reihe">3 Reihe </option>
        <option value="passat">Passat </option>
        <option value="navara">Navara </option>
```

```
<option value="ka">Ka </option>
<option value="polo">Polo </option>
<option value="twingo">Twingo </option>
<option value="a_klasse">A klasse </option>
<option value="scirocco">Scirocco </option>
<option value="5er">5er </option>
<option value="meriva">Meriva </option>
<option value="arosa">Arosa </option>
<option value="c4">C4 </option>
<option value="civic">Civic </option>
<option value="transporter">Transporter </option>
<option value="punto">Punto </option>
<option value="e_klasse">E Klasse </option>
<option value="clio">Clio </option>
<option value="kadett">Kadett </option>
<option value="kangoo">Kangoo </option>
<option value="corsa">Corsa </option>
<option value="one">One </option>
<option value="fortwo">Fortwo </option>
<option value="1er">1er </option>
<option value="b_klasse">B Klasse </option>
<option value="signum">Signum </option>
<option value="astra">Astra </option>
<option value="a8">A8 </option>
<option value="jetta">Jetta </option>
<option value="fiesta">Fiesta </option>
<option value="c_klasse">C Klasse </option>
<option value="micra">Micra </option>
<option value="vito">Vito </option>
<option value="sprinter">Sprinter </option>
<option value="156">156 </option>
<option value="escort">Escort </option>
<option value="forester">Forester </option>
<option value="xc_reihe">Xc Reihe </option>
<option value="scenic">Scenic </option>
<option value="a4">A4 </option>
<option value="a1">A1 </option>
```

```html
<option value="insignia">Insignia </option>
<option value="combo">Combo </option>
<option value="focus">Focus </option>
<option value="tt">Tt </option>
<option value="a6">A6 </option>
<option value="jazz">Jazz </option>
<option value="omega">Omega </option>
<option value="slk">Slk </option>
<option value="7er">7er </option>
<option value="80">80 </option>
<option value="147">147 </option>
<option value="glk">Glk </option>
<option value="100">100 </option>
<option value="z_reihe">Z Reihe </option>
<option value="sportage">Sportage </option>
<option value="sorento">Sorento </option>
<option value="v40">V40 </option>
<option value="5er">5er </option>
<option value="ibiza">Ibiza </option>
<option value="3er">3er </option>
<option value="mustang">Mustang </option>
<option value="eos">Eos </option>
<option value="touran">Touran </option>
<option value="getz">Getz </option>
<option value="a3">A3 </option>
<option value="almera">Almera </option>
<option value="megane">Megane </option>
<option value="7er">7er </option>
<option value="1er">1er </option>
<option value="lupo">Lupo </option>
<option value="r19">R19 </option>
<option value="zafira">Zafira </option>
<option value="caddy">Caddy </option>
<option value="2_reihe">2 Reihe </option>
<option value="mondeo">Mondeo </option>
<option value="cordoba">Cordoba </option>
<option value="colt">Colt </option>
```

```
<option value="impreza">Impreza </option>
<option value="vectra">Vectra </option>
<option value="berlingo">Berlingo </option>
<option value="80">80 </option>
<option value="m_klasse">M Klasse </option>
<option value="tiguan">Tiguan </option>
<option value="i_reihe">I Reihe </option>
<option value="espace">Espace </option>
<option value="sharan">Sharan </option>
<option value="6_reihe">6 Reihe </option>
<option value="panda">Panda </option>
<option value="up">Up </option>
<option value="seicento">Seicento </option>
<option value="ceed">Ceed </option>
<option value="5_reihe">5 Reihe </option>
<option value="yeti">Yeti </option>
<option value="octavia">Octavia </option>
<option value="mii">Mii </option>
<option value="rx_reihe">Rx Reihe </option>
<option value="6er">6er </option>
<option value="modus">Modus </option>
<option value="fox">Fox </option>
<option value="matiz">Matiz </option>
<option value="beetle">Beetle </option>
<option value="c1">C1 </option>
<option value="rio">Rio </option>
<option value="touareg">Touareg </option>
<option value="logan">Logan </option>
<option value="spider">Spider </option>
<option value="cuore">Cuore </option>
<option value="s_max">S Max </option>
<option value="a2">A2 </option>
<option value="x_reihe">X Reihe </option>
<option value="a5">A5 </option>
<option value="galaxy">Galaxy </option>
<option value="c3">C3 </option>
<option value="viano">Viano </option>
```

```
<option value="s_klasse">S Klasse </option>
<option value="1_reihe">1 Reihe </option>
<option value="avensis">Avensis </option>
<option value="sl">Sl </option>
<option value="roomster">Roomster </option>
<option value="q5">Q5 </option>
<option value="kaefer">Kaefer </option>
<option value="santa">Santa </option>
<option value="cooper">Cooper </option>
<option value="leon">Leon </option>
<option value="4_reihe">4 Reihe </option>
<option value="500">500 </option>
<option value="laguna">Laguna </option>
<option value="ptcruiser">Ptcruiser </option>
<option value="clk">Clk </option>
<option value="primera">Primera </option>
<option value="exeo">Exeo </option>
<option value="159">159 </option>
<option value="transit">Transit </option>
<option value="juke">Juke </option>
<option value="qashqai">Qashqai </option>
<option value="carisma">Carisma </option>
<option value="accord">Accord </option>
<option value="corolla">Corolla </option>
<option value="lanos">Lanos </option>
<option value="phaeton">Phaeton </option>
<option value="boxster">Boxster </option>
<option value="verso">Verso </option>
<option value="swift">Swift </option>
<option value="rav">Rav </option>
<option value="kuga">Kuga </option>
<option value="picanto">Picanto </option>
<option value="kalos">Kalos </option>
<option value="superb">Superb </option>
<option value="stilo">Stilo </option>
<option value="alhambra">Alhambra </option>
<option value="911">911 </option>
```

```
<option value="mx_reihe">Mx Reihe </option>
<option value="m_reihe">M Reihe </option>
<option value="roadster">Roadster </option>
<option value="ypsilon">Ypsilon </option>
<option value="cayenne">Cayenne </option>
<option value="galant">Galant </option>
<option value="justy">Justy </option>
<option value="90">90 </option>
<option value="sirion">Sirion </option>
<option value="crossfire">Crossfire </option>
<option value="6_reihe">6 Reihe </option>
<option value="agila">Agila </option>
<option value="duster">Duster </option>
<option value="cr_reihe">Cr Reihe </option>
<option value="v50">V50 </option>
<option value="discovery">Discovery </option>
<option value="c_reihe">C Reihe </option>
<option value="v_klasse">V Klasse </option>
<option value="yaris">Yaris </option>
<option value="c5">C5 </option>
<option value="aygo">Aygo </option>
<option value="cc">Cc </option>
<option value="carnival">Carnival </option>
<option value="fusion">Fusion </option>
<option value="bora">Bora </option>
<option value="forfour">Forfour </option>
<option value="100">100 </option>
<option value="cl">Cl </option>
<option value="tigra">Tigra </option>
<option value="156">156 </option>
<option value="300c">300c </option>
<option value="100">100 </option>
<option value="147">147 </option>
<option value="q3">Q3 </option>
<option value="spark">Spark </option>
<option value="v70">V70 </option>
<option value="x_type">X Type </option>
```

```
<option value="5_reihe">5 Reihe </option>
<option value="ducato">Ducato </option>
<option value="s_type">S Type </option>
<option value="x_trail">X Trail </option>
<option value="toledo">Toledo </option>
<option value="altea">Altea </option>
<option value="7er">7er </option>
<option value="voyager">Voyager </option>
<option value="calibra">Calibra </option>
<option value="bravo">Bravo </option>
<option value="range_rover">Range Rover </option>
<option value="antara">Antara </option>
<option value="tucson">Tucson </option>
<option value="q7">Q7 </option>
<option value="citigo">Citigo </option>
<option value="jimny">Jimny </option>
<option value="cx_reihe">Cx Reihe </option>
<option value="wrangler">Wrangler </option>
<option value="lybra">Lybra </option>
<option value="range_rover_sport">Range Rover Sport </option>
<option value="lancer">Lancer </option>
<option value="159">159 </option>
<option value="freelander">Freelander </option>
<option value="captiva">Captiva </option>
<option value="c2">C2 </option>
<option value="500">500 </option>
<option value="range_rover_evoque">Range Rover Evoque </option>
<option value="sandero">Sandero </option>
<option value="note">Note </option>
<option value="900">900 </option>
<option value="147">147 </option>
<option value="defender">Defender </option>
<option value="cherokee">Cherokee </option>
<option value="clubman">Clubman </option>
<option value="samara">Samara </option>
<option value="2_reihe">2 Reihe </option>
<option value="1er">1er </option>
```

```
<option value="3er">3er </option>
<option value="601">601 </option>
<option value="3_reihe">3 Reihe </option>
<option value="4_reihe">4 Reihe </option>
<option value="5er">5er </option>
<option value="6_reihe">6 Reihe </option>
<option value="legacy">Legacy </option>
<option value="pajero">Pajero </option>
<option value="auris">Auris </option>
<option value="niva">Niva </option>
<option value="5_reihe">5 Reihe </option>
<option value="s60">S60 </option>
<option value="nubira">Nubira </option>
<option value="vivaro">Vivaro </option>
<option value="g_klasse">G Klasse </option>
<option value="lodgy">Lodgy </option>
<option value="850">850 </option>
<option value="serie_2">Serie 2 </option>
<option value="6er">6er </option>
<option value="charade">Charade </option>
<option value="croma">Croma </option>
<option value="outlander">Outlander </option>
<option value="gl">Gl </option>
<option value="doblo">Doblo </option>
<option value="musa">Musa </option>
<option value="amarok">Amarok </option>
<option value="156">156 </option>
<option value="move">Move </option>
<option value="9000">9000 </option>
<option value="v60">V60 </option>
<option value="145">145 </option>
<option value="aveo">Aveo </option>
<option value="200">200 </option>
<option value="300c">300c </option>
<option value="b_max">B Max </option>
<option value="delta">Delta </option>
<option value="terios">Terios </option>
```

```
<option value="rangerover">RangeRover </option>
<option value="90">90 </option>
<option value="materia">Materia </option>
<option value="kalina">Kalina </option>
<option value="elefantino">Elefantino </option>
<option value="i3">I3 </option>
<option value="kappa">Kappa </option>
<option value="serie_3">Serie 3 </option>
<option value="48429">48429 </option>
<option value="serie_1">Serie 1 </option>
<option value="discovery_sport">Discovery Sport </option>

</select>
<br>
<br>
</td>
</tr>

<tr>
<td><label for="brand">Brand :</label></td>
<td>
<select name="brand" id="brand">
<option value="" disabled selected hidden>Choose Brand Name...</option>
<option value="volkswagen">Volkswagen </option>
<option value="audi">Audi </option>
<option value="jeep">Jeep </option>
<option value="skoda">Skoda </option>
<option value="bmw">Bmw </option>
<option value="peugeot">Peugeot </option>
<option value="ford">Ford </option>
<option value="mazda">Mazda </option>
<option value="nissan">Nissan </option>
<option value="renault">Renault </option>
<option value="mercedes_benz">Mercedes Benz </option>
<option value="opel">Opel </option>
<option value="seat">Seat </option>
<option value="citroen">Citroen </option>
```

```
    <option value="honda">Honda </option>
    <option value="fiat">Fiat </option>
    <option value="mini">Mini </option>
    <option value="smart">Smart </option>
    <option value="hyundai">Hyundai </option>
    <option value="sonstige_autos">Sonstige Autos </option>
    <option value="alfa_romeo">Alfa Romeo </option>
    <option value="subaru">Subaru </option>
    <option value="volvo">Volvo </option>
    <option value="mitsubishi">Mitsubishi </option>
    <option value="kia">Kia </option>
    <option value="suzuki">Suzuki </option>
    <option value="lancia">Lancia </option>
    <option value="porsche">Porsche </option>
    <option value="toyota">Toyota </option>
    <option value="chevrolet">Chevrolet </option>
    <option value="dacia">Dacia </option>
    <option value="daihatsu">Daihatsu </option>
    <option value="trabant">Trabant </option>
    <option value="saab">Saab </option>
    <option value="chrysler">Chrysler </option>
    <option value="jaguar">Jaguar </option>
    <option value="daewoo">Daewoo </option>
    <option value="rover">Rover </option>
    <option value="land_rover">Land Rover </option>
    <option value="lada">Lada </option>
    </select>
    <br>
    <br>
    </td>
    </tr>

<tr>
    <td><label for="damage">Repaired or damaged : </label></td>
    <td><input type="radio" name="damage" value="yes"/> Yes
    <input type="radio" name="damage" value="no"/> No
    <input type="radio" name="damage" value="not-declared"/> Not declared
    <br>
    <br>
    </td>
    </tr>

    <tr>
    <td><label for="fuelType">Fuel Type :</label></td>
    <td>
    <select name="fuelType" id="brand">
    <option value="" disabled selected hidden>Choose Fuel Type...</option>
    <option value="petrol"> Petrol </option>
    <option value="diesel"> Diesel </option>
    <option value="not-declared"> Not Declared </option>
    <option value="lpg">LPG </option>
    <option value="cng">CNG </option>
    <option value="hybrid">Hybrid </option>
    <option value="others">Others </option>
    <option value="electric">Electric </option>
    </select>
    <br>
    <br>
    </td>
    </tr>

    <tr>
    <td><label for="vehicletype">Vehicle type:</label></td>
    <td>
    <select name="vehicletype" id="vehicle" >
    <option value="" disabled selected hidden>Choose Vehicle Type...</option>
    <option value="coupe">Coupe </option>
<option value="suv">SUV </option>
<option value="kleinwagen">Kleinwagen </option>
<option value="limousine">Limousine </option>
    <option value="cabrio">Cabrio </option>
    <option value="bus">Bus </option>
    <option value="kombi">Kombi </option>
    <option value="andere">Andere </option>
```

```
            <option value="volkswagen">Volkswagen </option>
            </select>
            <br>
            <br>
            </td>
            </tr>
            </tbody>
            </table>
            <input name="Submit" type="Submit" value="Submit" id="button"/>
            </form>
  </section>
</center>
</body>
</html>
```

Predict.css

```
.header{
  min-height: 100vh;
  width: 100%;
  background-image: url("https://media.istockphoto.com/id/1134703571/photo/front-view-of-
fictitious-car.jpg?s=612x612&w=0&k=20&c=UGUtuDpFuCLgzWtTCP4-
LQvJjJ91E16OomI2mAfTkKM=");
  background-position: center;
  background-size: cover;
  position: relative;
background-repeat: no-repeat;
overflow:hidden;

}

.text-box{
  text-align: center;
  position: relative;
  color:  #FFE4C4;
  top:50%;
}
.text-box h1{
  margin-top: 50px;
  font-size: 55px;
}

.text-box p{
  margin: 10px 0 40px;
  font-size: 15px;
}

nav{
  display:flex;
  padding: 2% 6%;
  justify-content: space-between;
  align-items: center;
}
```

Style.css

```
*{
  margin: 0;
  padding: 0;
}

.header{
  min-height: 100vh;
  width: 100%;
  background-image: url("https://media.istockphoto.com/id/1272936966/photo/sports-car-speeding-in-urban-highway.jpg?b=1&s=170667a&w=0&k=20&c=njAx2rXb-aURFG3_QG0jUfzhuPmjtiwH_wrTwGX9Rwo=");
  background-position: center;
  background-size: cover;
  position: relative;
}
nav{
  display:flex;
  padding: 2% 6%;
  justify-content: space-between;
  align-items: center;
```

```
}
.nav-links{
  flex: 1;
  text-align: right;
}
.nav-links ul li{
  list-style: none;
  display: inline-block;
  padding: 8px 12px;
  position: relative;
}
.nav-links ul li a{
  color:white;
  text-decoration: none;
  font-size: 13px;
}
.text-box{
  text-align: center;
  position: relative;
  color:  #FFE4C4;
  top:50%;
}
.text-box h1{
  margin-top: 50px;
  font-size: 55px;
}
.text-box p{
  margin: 10px 0 40px;
  font-size: 15px;
}
.visit-btn{
  display: inline;
  border: 3px solid #fff;
  padding:10px 14px;
  font-size: 15px;
  background: transparent;
  color: white;
  text-decoration:none;
}
@import url('https://fonts.googleapis.com/css2?family=Poppins:wght@500&display=swap');
.container-text {
  display: flex;
  justify-content: center;
  align-items: center;
  width: 50%;
  height: 30vh;
}

h1 {
  text-transform: uppercase;
            background-image: linear-gradient(to right top,#DAE2F8, #DD5E89);
   background-size: 50% auto;
            -webkit-background-clip: text;
            -webkit-text-fill-color: transparent;
  font-family: 'Poppins', sans-serif;
  font-size: 3.5vw;
  animation: gradientAnim 7.5s ease  alternate infinite;
}
.button{
          background:  #70e1f5;
          border: 0;
          width: 10%;
          height: 40px;
          border-radius: 3px;
          color: black;
          cursor: pointer;
          transition: background 0.3s ease-in-out;
        }

.tab1{
   tab-size:16;
   margin-left: 250px;
}
```

```
@keyframes gradientAnim {
  to {
    background-position: 50% top;
  }
}
```

Value.css

```
body {
  background-image: linear-gradient(#292E49, #536976);
  width: 700px;
  border: 15px solid #DAE2F8;
  padding: 10px;
  margin: 400px;
  /*overflow: hidden;*/


}
input[type=submit]:hover {
  background-color: #45a049;
}
.header{
  width: 100%;
  text-align: center;
  padding-top: 10px;
  font-size:20px;
  font-family: "Lucida Console";
  background-color:#43FFB6;
  border:0%;
  top:0px;
  bottom:0px;
  right:0px;
  left:0px;
  overflow-y:auto;
}

.form{
```

```
background-image: linear-gradient(#536976,#292E49);
background-position: center;
  background-size: cover;
  position: relative;
}
.form{
text-align: center;
padding:10px;
text-top:10px;
display: flex;
flex-direction: column;
align-items: center;
}
.form{
font-size:22px;
}
textarea {
  width: 100%;
  height: 50px;
  padding: 12px 20px;
  box-sizing: border-box;
  border: 2px solid #ccc;
  border-radius: 4px;
  background-color: #f8f8f8;
  resize: none;
}
input[type=text] {
  transition: width 0.4s ease-in-out;
}


#model{
width: 70%;
}
#brand{
width:70%;
}
```

```
#vehicle{
width:70%;
}
*{
color:black;
}

#button{
  padding: 10px 10px;
  margin: 0;

  text-align:center;
  width:100px;
}
```

Integrate_flask.py

```
# Import Libraries
import pickle

import numpy as np
import pandas as pd
import requests
from sklearn.preprocessing import LabelEncoder

from flask import Flask, Response, render_template, request

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "MIfDRZYQhDHWH7dNHo2oQrSY2ajDfwJGV8PLQI9NIX36"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey": API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}


app = Flask(__name__)#initiate flask app
```

```
def load_model(file='resale_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('Result\\'+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)

    # NOTE: manually define and pass the array(s) of values to be scored in the next line
    payload_scoring = {"input_data": [{"field": [['yearOfReg', 'powerPS', 'kilometer',
'monthOfRegistration','gearbox_labels', 'notRepairedDamage_labels',
'model_labels','brand_labels', 'fuelType_labels', 'vehicletype_labels']], "values": X}]}
    #payload_scoring = {"input_data": [{"fields": [array_of_input_fields], "values":
[array_of_values_to_be_scored, another_array_of_values_to_be_scored]}]}

    response_scoring = requests.post('https://eu-
de.ml.cloud.ibm.com/ml/v4/deployments/99a4f93d-9a11-4878-95ed-
d5395db2f283/predictions?version=2022-11-16', json=payload_scoring,
headers={'Authorization': 'Bearer ' + mltoken})
    predictions = response_scoring.json()
    print(response_scoring.json())
    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(host='localhost', debug=True, threaded=False)
```

GitHub : https://github.com/IBM-EPBL/IBM-Project-35385-1660284370

Project Demo Link: https://www.youtube.com/embed/IhsMjd1OCQM