CAR RESALE VALUE PREDICTOR

TEAM NUMBER: PNT2022TMID35766

<u>TEAM MEMBERS</u>

SHRI NIKHITHAA R.M

PRIYADHARSHINI R

GAYATHRISRI R

JOTHIKA S

INTRODUCTION

PROJECT OVERVIEW

Predicting the value of used cars is a hotly debated subject because of the extraordinary volume of vehicles being bought and sold. People tend to buy used automobiles more frequently in developing nations since they are more affordable.It is expected that sales of old cars and second-hand imported (reconditioned) autos will rise in tough economic times. Leasing a car rather than purchasing one entirely is typical in many developed nations. 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, being able to accurately anticipate the salvage value (residual value) of cars is in the best interest of sellers and financiers from a business standpoint.

We suggest an intelligent, adaptable, and efficient method that is based on applying regression algorithms to forecast the resale value of the vehicle. A regression model needs to be constructed that would provide the vehicle's closest resale value, taking into account the key variables that would impact this value. The method with the highest accuracy will be chosen from among the several regression algorithms we employ, and it will then be implemented into the web-based application that notifies the user of the status of his product.

PURPOSE

The main objective of the project is to build a model for predicting the resale value of a car. A regression model needs to be constructed that would provide the vehicle's closest resale value, taking into account the key variables that would impact this value.The user will register their cars with the decsription of the features. Then the resale value of the car will be predicted and given as the final result.

LITERATURE SURVEY

EXISTING PROBLEM

Predicting the value of used cars is a hotly debated subject because of the extraordinary volume of vehicles being bought and sold. People tend to buy used automobiles more frequently in developing nations since they are more affordable. Leasing a car rather than purchasing one entirely is typical in many developed nations. 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, being able to accurately anticipate the salvage value (residual value) of cars is in the best interest of sellers and financiers from a business standpoint.

REFERENCES

S.NO	PAPER	PROPOSED SYSTEM AND RELATED WORK	LIMITATIONS
1.	Machine Learning Techniques To Predict The Price Of Used Cars Predictive Analytics in Retail Business IEEE journal 2021 Chejarla Venkat Narayana hinta Lakshmi Likhitha Syed Bademiya Karre Kusumanjali	In this paper, we gather the data and perform exploratory analysis on it to obtain a summary view of the data, Next, we will perform preprocessing tasks, such as handling of missing data, categorical variables, and scaling of features. Later, we'll handle outliers, split the dataset into train and test divisions, and evaluate the significance of features in the model's construction, among other feature engineering chores. Finally, using machine learning methods, we will create a range of prediction models and assess them.	One of the major drawbacks of this paper is the data they had taken 1105 samples of limited features only. Since data gathered using web scraper, so there are several samples with few attribute values.
2.	Prediction Of Used Car Prices Using Artificial Neural Networks And Machine Learning IEEE journal 2022 Janke Varshitha,K Jahnavi,Dr. C. Lakshmi	Here ,In any AI/ML models the first step is to pre-process the raw data based on the requirements of the project and then train and test the data based on the algorithm used. Firstly, data processing, training and followed by testing.	This model designed here is restricted to predict the price of used cars it can be extended to any electric gadget or household appliance as well.

3.	Used Car Price Prediction using Machine Learning IEEE journal 2021 Feng Wang Xusong Zhang Qiang Wang	In this paper, a simple linear regression problem is used. We investigate the relationship between one independent variable and another one dependent variable. Multiple linear regression models are used to estimate a linear (or nonlinear) relationship between multiple input variables and an output variable.	The R2 score is not sufficient to determine whether the model is a good fit. Also the GBR model recorded a low RMSE error minimization value.
4.	A New Model for Residual Value Prediction of the Used Car Based on BP Neural Network and Nonlinear Curve Fit IEEE journal Shen Gongqi, Wang Yansong, Zhu Qiang 2011	In this paper, a comprehensive method combined by the BP neural network and nonlinear curve fit was introduced for optimising the model due to its flexible nonlinearity paper. A comprehensive method combined by the BP neural network and nonlinear curve fit was introduced for optimizing the model due to its flexible nonlinearity.	The car manufacturer, model, mileage, age, maintenance record, physical condition, market occupancy, after-sale service,and the driving habit of owner factors were the original factors which were interconnected with each other, affecting the residual value coherently. If the relationship among these factors was neglected and the function of factors to residual value was calculated respectively, it will lower the accuracy of the prediction model.
5.	Predicting the Selling Price of Cars Using Business Intelligence with the Feed-forward Backpropagation Algorithms	In this paper, using the concepts of descriptive, predictive, and prescriptive they implemented Business Intelligence and use the feed-forward backpropagation algorithm	However, the engine learning techniques (ANN, SVM SVM, and Random Forest) can only be implemented as an ensemble so they have to use the data

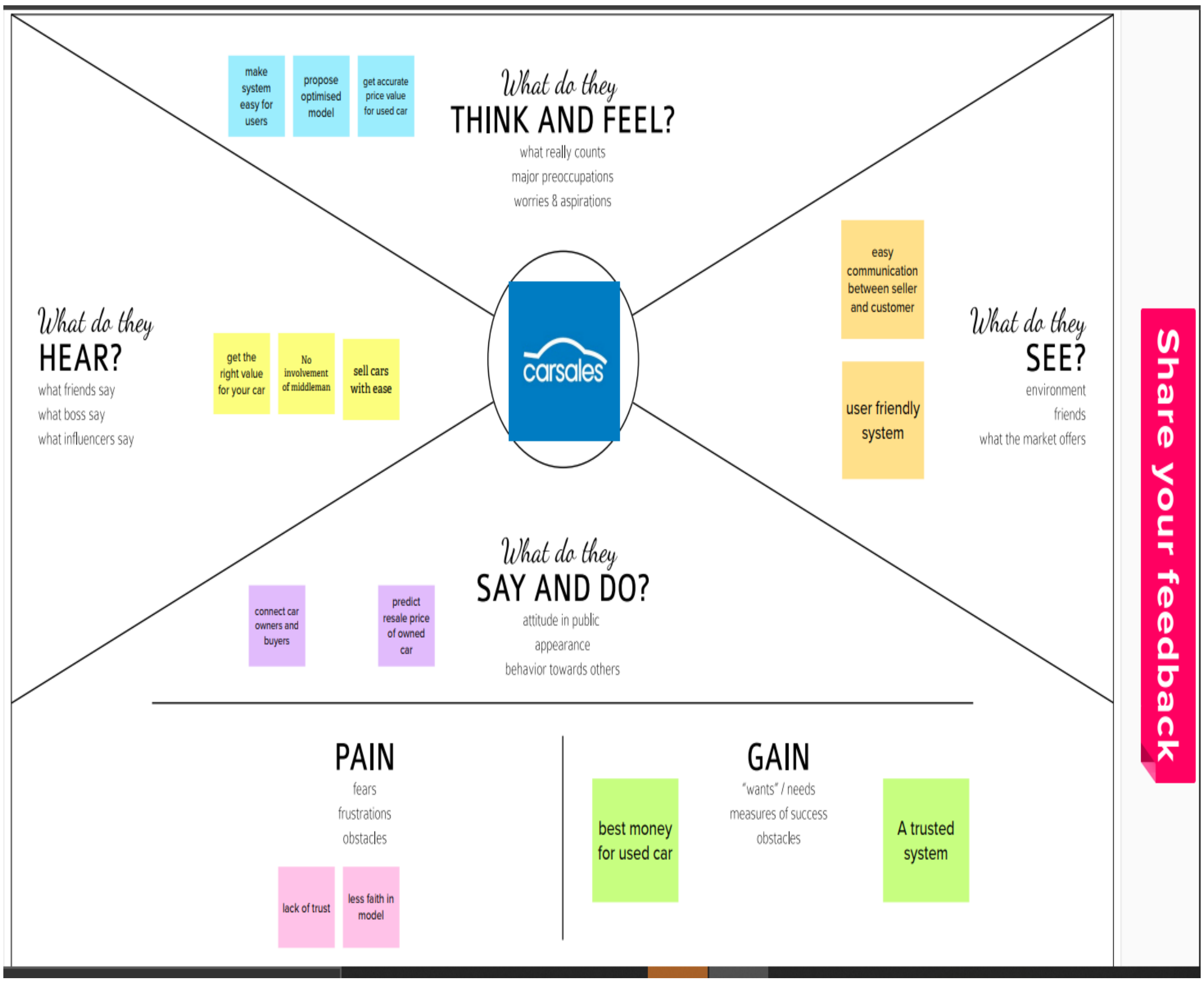
	IEEE journal Nur Oktavin Idris Aspian Achban Siti Andini Utiarahman Jorry Karim Fuad Pontoiyo	to predicts the selling price of a car based on its specification and predict a car price based on the latest specification which has never been on sale.	collected through web scraper using PHP programming to make predictions.
6.	Second-Hand Car Trading Framework Based on Blockchain in Cloud Service Environment IEEE journal 2021 Yimin Yu Chuanjia Yao Yi Zhang Rong Jiang	In this paper, attempts to use blockchain technology as an auxiliary means to solve the long-standing problems in the used car market. This paper proposes a framework of used car trading based on blockchain in cloud service environment, and explains the working principle of the framework.	There are also some limitations. Firstly, if a car manufacturer didn't obey the law and forged documents in the very beginning in order to make more profits, it would be very difficult for consumers to testify. Secondly, even through the framework can trail a car's life track, it can not reach 100% accuracy for mileage during usage. Thirdly, how to improve the work efficiency between nodes and make sure the normal operation of the framework is also a problem.

PROBLEM STATMENT

Predicting the value of used cars is a hotly debated subject because of the extraordinary volume of vehicles being bought and sold. People tend to buy used automobiles more frequently in developing nations since they are more affordable. Leasing a car rather than purchasing one entirely is typical in many developed nations. 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, being able to accurately anticipate the salvage value (residual value) of cars is in the best interest of sellers and financiers from a business standpoint. The main objective of the project is to predict used cars prices using machine learning techniques, by analysing the different aspects and factors that lead to the actual used car price valuation. To enable the consumers to know the actual worth of their car or desired car, by simply providing the program with a set of attributes from the desired car to predict the car price. The purpose of this project is to understand and evaluate used car prices and to develop a strategy that utilizes machine learning techniques to predict used car prices. An adaptable, and efficient method which is based on applying regression algorithms is developed to predict the resale value of the cars

IDEATION AND PROPOSED SOLUTION

EMPATHY MAP CANVAS



IDEATION AND BRAINSTORMING

The ideation phase involves creating the empathy map and proposed solution. The ideas where brainstormed and the best solution is suggested. For the proposed model for predicting car resale value, an empathy map is drawn.The propsed solution is stated and the solution fit architecture diagram is also drawn to give a clear picture of the proposed solution.

The purpose of this project is to understand and evaluate used car prices and to develop a strategy that utilizes machine learning techniques to predict used car prices. An adaptable, and efficient method which is based on applying regression algorithms is developed to predict the resale value of the cars

PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	This project's primary goal is to employ regression techniques to predict a used car's resale value. This might make it easier for clients to determine the best price for the used vehicle being offered.
2.	Idea / Solution description	A car's resale value is influenced by a variety of variables, including its price, fuel type, model, gearbox, and vehicle type. In order to manage missing values and outliers, standardize the data, and divide it into dependent and independent variables, the data is per-processed. The model is then created using regression methods to forecast the car's resale value.
3.	Novelty / Uniqueness	This is a current issue that can help both the buyer and the vendor. This proposal's originality lies in trying to estimate the resale value as closely as feasible to the real value.
4.	Social Impact / Customer Satisfaction	It is more likely that use of used cars will rise given the current economic climate. Customers and sellers have a shared commercial interest in this. This creates a sense of trust between the seller and the buyer by predicting the resale prices of the car based on all of its qualities and preventing over- or under-pricing.
5.	Business Model (Revenue Model)	The suggested model may be offered for sale to resellers who would then use it to determine the ideal bid price. If more users started using it to determine the best price for a used automobile, it could be turned into an application and

		generate income from it.
6.	Scalability of the Solution	The primary model is targeted only for a lower number of audiences. However, as the customer base increases for the model it can be extended to the cloud for effective services.

PROBLEM SOLUTION FIT

Define CS, fit into CC

1. CUSTOMER SEGMENT(S) CS

Customers
Car mechanic

6. CUSTOMER CONSTRAINTS CC

All the necessary information information about the car is to be known by the customer to find the correct resale value.

5. AVAILABLE SOLUTIONS AS

We use an intelligent and effective system to predict the resale value of the car.

Explore AS, differentiate

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

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

Customer should know clearly about the details of their car

9. PROBLEM ROOT CAUSE RC

There is no proper platform to find the resale value of the cars.

7. BEHAVIOUR BE

Customers have to to enter the details of the cars in the web application to find the resale price of the car

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

3. TRIGGERS TR • When customers sell their car. • When a customer buys a used car.	**10. YOUR SOLUTION** SL The proposed solution is to develop a care resale value predictor using machine learning techniques like regression and random forest.	**8.CHANNELS of BEHAVIOR** CH **8.1 ONLINE** Car details that have to be entered in a web application. **8.2 OFFLINE** Customers have to collect the necessary information about the specifications of the car.
4. EMOTIONS: BEFORE / AFTER EM Customers get an awareness of the resale price of their car.		

REQUIREMENT ANALYSIS

FUNCTIONAL REQUIREMENTS

The follwing are the functional requirements for the project:

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

NON-FUNCTIONAL REQUIREMENTS

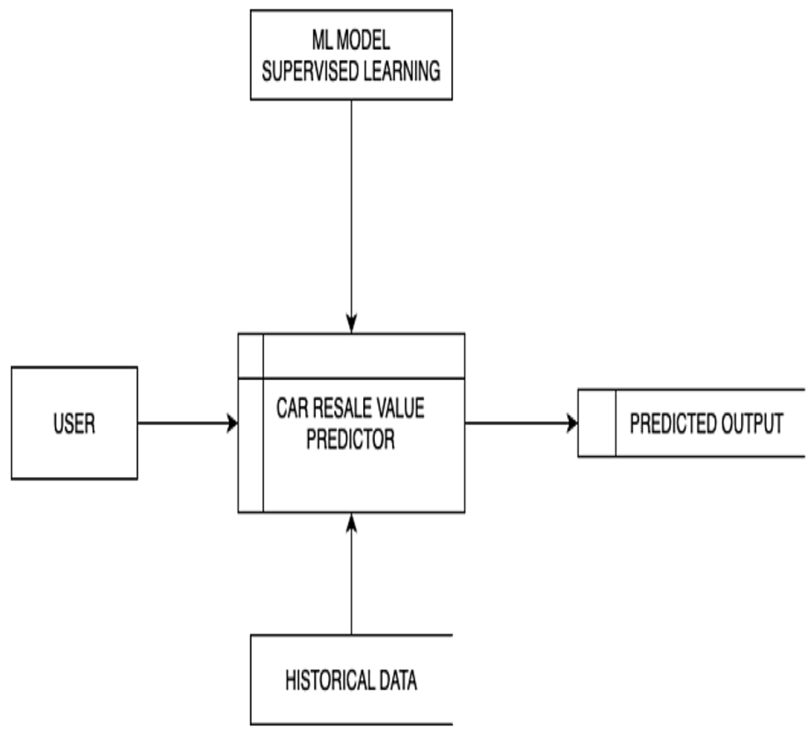
The follwing are the non- functional requirements for the project:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Predicting the resale value of used cars
NFR-2	**Security**	Providing security to the website which is developed
NFR-3	**Reliability**	Providing reliability by predicting the resale values values for different types of cars.
NFR-4	**Performance**	Providing high performance by using some machine learning techniques like regression ,random forest.
NFR-5	**Availability**	Resale values of all types of cars can be predicted.
NFR-6	**Scalability**	Resale value is predicted for different types of cars.

PROJECT DESIGN

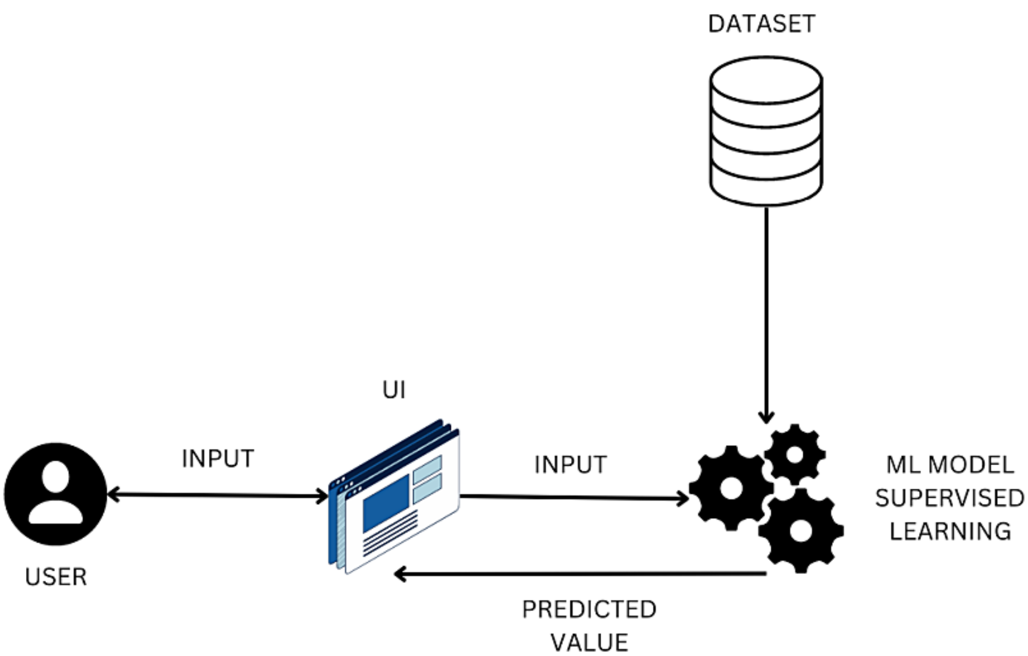
DATA FLOW DIAGRAM

Data Flow Diagrams: DFD LEVEL- 0

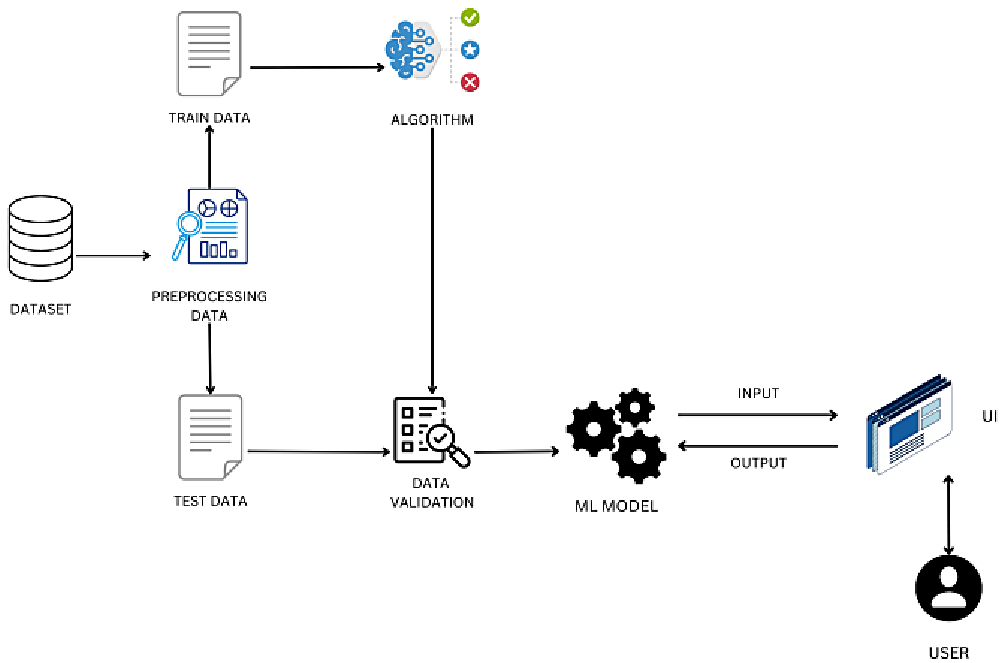


SOLUTION AND TECHNICAL ARCHITECTURE

SOLUTION ARCHITECTURE



TECHNICAL ARCHITECTURE



USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (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
Customer (Web user)	Login	USN-2	As a user, I can log into the application by entering email & password	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Web user)	Dashboard	USN-3	As a user, I can view my Dashboard and all the elements in the Dashboard are in working condition	I can view and access Dashboard	Low	Sprint-1
Customer (Web user)	Data Entry	USN-4	As a user, I can enter the details of the car whose value is to be predicted	I can enter the details of the car	High	Sprint-2
Customer (Web user)	Predicted Output	USN-5	As a user, I can view the predicted price of the car using the details provided	I can view the predicted price of the car	High	Sprint-2
Administrator	ML Model	USN-6	As an Administrator, I can modify or upgrade the ml model used for predicting the price of a car	I can modify or upgrade the model	low	Sprint-3
Administrator	Website	USN-7	As an Administrator, I can maintain the website	I can maintain the website	Low	Sprint-3
Administrator	Database	USN-8	As an Administrator, I can maintain the databases	I can maintain the databases	Low	Sprint-3

PROJECT PLANNING AND EXECUTION

SPRINT PLANNING AND ESTIMATION

Sprint 1 -Collecting dataset and preprocessing of data

Sprint 2- Creating the User Interface

Sprint 3-Running the flask application.

Sprint 4-Integrating the project

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	15	5 Days	20 Oct 2022	25 Oct 2022	15	25 Oct 2022
Sprint-2	15	5 Days	27 Oct 2022	01 Nov 2022	15	01 Nov 2022
Sprint-3	15	5 Days	03 Nov 2022	08 Nov 2022	15	03 Nov 2022
Sprint-4	15	5 Days	10 Nov 2022	15 Nov 2022	15	10 Nov 2022

CODING AND SOLUTIONING

FEATURE - PREDICTING THE RESALE VALUE OF THE CAR USING RANDOM FOREST ALGORITHM

The proposed solution for predicitng the resale value of the cars is done by using the random forest algorithm.The user has to enter the details of the car and the resale value will be predicted using random forest algorithm.The coding is as given below:

1 IMPORTING LIBRARIES

[2]:
```
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import cross_val_score, train_test_split,␣
↪StratifiedShuffleSplit , RandomizedSearchCV
import pickle
import datetime
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor
```

2 LOAD DATA

[3]:
```
file = '/Users/rpriyadharshini/Desktop/autos.csv'
```

[4]:
```
import chardet
with open(file, 'rb') as rawdata:
    result = chardet.detect(rawdata.read(100000))
result
```

[4]:
```
{'encoding': 'Windows-1252', 'confidence': 0.73, 'language': ''}
```

[5]:
```
df = pd.read_csv(file, encoding = 'Windows-1252')
df.head()
```

[5]:
```
           dateCrawled                            name  seller offerType  \
0  2016-03-24 11:52:17                      Golf_3_1.6  privat   Angebot
1  2016-03-24 10:58:45            A5_Sportback_2.7_Tdi  privat   Angebot
2  2016-03-14 12:52:21  Jeep_Grand_Cherokee_"Overland"  privat   Angebot
3  2016-03-17 16:54:04              GOLF_4_1_4__3TÜRER  privat   Angebot
4  2016-03-31 17:25:20  Skoda_Fabia_1.4_TDI_PD_Classic  privat   Angebot
```

```
   price abtest vehicleType  yearOfRegistration    gearbox  powerPS  model  \
0    480   test         NaN                1993    manuell        0   golf
1  18300   test       coupe                2011    manuell      190    NaN
2   9800   test         suv                2004  automatik      163  grand
3   1500   test  kleinwagen                2001    manuell       75   golf
4   3600   test  kleinwagen                2008    manuell       69  fabia

   kilometer  monthOfRegistration fuelType       brand notRepairedDamage  \
0     150000                    0   benzin  volkswagen               NaN
1     125000                    5   diesel        audi                ja
2     125000                    8   diesel        jeep               NaN
3     150000                    6   benzin  volkswagen              nein
4      90000                    7   diesel       skoda              nein

           dateCreated  nrOfPictures  postalCode             lastSeen
0  2016-03-24 00:00:00             0       70435  2016-04-07 03:16:57
1  2016-03-24 00:00:00             0       66954  2016-04-07 01:46:50
2  2016-03-14 00:00:00             0       90480  2016-04-05 12:47:46
3  2016-03-17 00:00:00             0       91074  2016-03-17 17:40:17
4  2016-03-31 00:00:00             0       60437  2016-04-06 10:17:21
```

3 EXPLORATORY DATA ANALYSIS

3.1 SHAPE

[6]:
```
df.shape
```

[6]:
```
(371528, 20)
```

[7]:
```
print(df['fuelType'].unique())
print(df['seller'].unique())
print(df['gearbox'].unique())
```

```
['benzin' 'diesel' nan 'lpg' 'andere' 'hybrid' 'cng' 'elektro']
['privat' 'gewerblich']
['manuell' 'automatik' nan]
```

3.2 CHECK NULL VALUES

[8]:
```
df.isnull().sum()
```

[8]:
```
dateCrawled              0
name                     0
seller                   0
offerType                0
price                    0
abtest                   0
```

```
vehicleType            37869
yearOfRegistration         0
gearbox                20209
powerPS                    0
model                  20484
kilometer                  0
monthOfRegistration        0
fuelType               33386
brand                      0
notRepairedDamage      72060
dateCreated                0
nrOfPictures               0
postalCode                 0
lastSeen                   0
dtype: int64
```

3.3 DESCRIBE THE DATA

[9]:
```
df.describe()
```

[9]:
```
              price  yearOfRegistration         powerPS      kilometer  \
count  3.715280e+05       371528.000000  371528.000000  371528.000000
mean   1.729514e+04         2004.577997     115.549477  125618.688228
std    3.587954e+06           92.866598     192.139578   40112.337051
min    0.000000e+00         1000.000000       0.000000    5000.000000
25%    1.150000e+03         1999.000000      70.000000  125000.000000
50%    2.950000e+03         2003.000000     105.000000  150000.000000
75%    7.200000e+03         2008.000000     150.000000  150000.000000
max    2.147484e+09         9999.000000   20000.000000  150000.000000

       monthOfRegistration  nrOfPictures    postalCode
count        371528.000000      371528.0  371528.00000
mean              5.734445           0.0   50820.66764
std               3.712412           0.0   25799.08247
min               0.000000           0.0    1067.00000
25%               3.000000           0.0   30459.00000
50%               6.000000           0.0   49610.00000
75%               9.000000           0.0   71546.00000
max              12.000000           0.0   99998.00000
```

4 PREPROCESSING DATA

[10]:
```
df1 = df.drop(columns='name')
df1.shape
```

[10]:
```
(371528, 19)
```

[11]:
```
df.seller.value_counts()
```

[11]:
```
privat        371525
gewerblich         3
Name: seller, dtype: int64
```

[12]:
```
df=df[df.seller != 'gewerblich']
```

[13]:
```
df=df.drop('seller',1)
```

```
/var/folders/13/mdvvnj0d299_920wwhpnhy5m0000gn/T/ipykernel_1648/1037493778.py:1:
FutureWarning: In a future version of pandas all arguments of DataFrame.drop
except for the argument 'labels' will be keyword-only
  df=df.drop('seller',1)
```

[14]:
```
df.offerType.value_counts()
```

[14]:
```
Angebot    371513
Gesuch         12
Name: offerType, dtype: int64
```

[15]:
```
df=df[df.offerType != 'Gesuch']
```

[16]:
```
df
```

[16]:
```
                dateCrawled                                    name  \
0       2016-03-24 11:52:17                              Golf_3_1.6
1       2016-03-24 10:58:45                    A5_Sportback_2.7_Tdi
2       2016-03-14 12:52:21              Jeep_Grand_Cherokee_"Overland"
3       2016-03-17 16:54:04                        GOLF_4_1_4__3TÜRER
4       2016-03-31 17:25:20              Skoda_Fabia_1.4_TDI_PD_Classic
…                       …                                         …
371523  2016-03-14 17:48:27                   Suche_t4___vito_ab_6_sitze
371524  2016-03-05 19:56:21         Smart_smart_leistungssteigerung_100ps
371525  2016-03-19 18:57:12            Volkswagen_Multivan_T4_TDI_7DC_UY2
371526  2016-03-20 19:41:08                        VW_Golf_Kombi_1_9l_TDI
371527  2016-03-07 19:39:19  BMW_M135i_vollausgestattet_NP_52.720____Euro

       offerType  price   abtest vehicleType  yearOfRegistration    gearbox  \
0        Angebot    480     test         NaN                1993    manuell
1        Angebot  18300     test       coupe                2011    manuell
2        Angebot   9800     test         suv                2004  automatik
3        Angebot   1500     test  kleinwagen                2001    manuell
4        Angebot   3600     test  kleinwagen                2008    manuell
…            …     …      …        …          …                 …        …
371523   Angebot   2200     test         NaN                2005        NaN
371524   Angebot   1199     test      cabrio                2000  automatik
```

```
371525  Angebot   9200     test         bus                 1996    manuell
371526  Angebot   3400     test       kombi                 2002    manuell
371527  Angebot  28990  control   limousine                 2013    manuell

        powerPS        model  kilometer  monthOfRegistration fuelType  \
0             0         golf     150000                    0   benzin
1           190          NaN     125000                    5   diesel
2           163        grand     125000                    8   diesel
3            75         golf     150000                    6   benzin
4            69        fabia      90000                    7   diesel
…           …          …      …                      …       …
371523        0          NaN      20000                    1      NaN
371524      101       fortwo     125000                    3   benzin
371525      102  transporter     150000                    3   diesel
371526      100         golf     150000                    6   diesel
371527      320      m_reihe      50000                    8   benzin

                 brand notRepairedDamage          dateCreated  nrOfPictures  \
0           volkswagen               NaN  2016-03-24 00:00:00             0
1                 audi                ja  2016-03-24 00:00:00             0
2                 jeep               NaN  2016-03-14 00:00:00             0
3           volkswagen              nein  2016-03-17 00:00:00             0
4                skoda              nein  2016-03-31 00:00:00             0
…                  …               …                   …             …
371523  sonstige_autos               NaN  2016-03-14 00:00:00             0
371524           smart              nein  2016-03-05 00:00:00             0
371525      volkswagen              nein  2016-03-19 00:00:00             0
371526      volkswagen               NaN  2016-03-20 00:00:00             0
371527             bmw              nein  2016-03-07 00:00:00             0

        postalCode            lastSeen
0            70435  2016-04-07 03:16:57
1            66954  2016-04-07 01:46:50
2            90480  2016-04-05 12:47:46
3            91074  2016-03-17 17:40:17
4            60437  2016-04-06 10:17:21
…              …                   …
371523       39576  2016-04-06 00:46:52
371524       26135  2016-03-11 18:17:12
371525       87439  2016-04-07 07:15:26
371526       40764  2016-03-24 12:45:21
371527       73326  2016-03-22 03:17:10

[371513 rows x 19 columns]
```

[17]:

```
df=df.drop('offerType',1)
```

```
/var/folders/13/mdvvnj0d299_920wwhpnhy5m0000gn/T/ipykernel_1648/2498620258.py:1:
FutureWarning: In a future version of pandas all arguments of DataFrame.drop
except for the argument 'labels' will be keyword-only
  df=df.drop('offerType',1)
```

[18]:
```
df.shape
```

[18]: (371513, 18)

[19]:
```
df = df[(df.powerPS > 50) & (df.powerPS <900)]
```

[20]:
```
df.shape
```

[20]: (319704, 18)

[21]:
```
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)]
```

[22]:
```
df.shape
```

[22]: (309166, 18)

[23]:
```
df.head()
```

[23]:
```
           dateCrawled                                              name  \
1  2016-03-24 10:58:45                               A5_Sportback_2.7_Tdi
2  2016-03-14 12:52:21                     Jeep_Grand_Cherokee_"Overland"
3  2016-03-17 16:54:04                                 GOLF_4_1_4__3TÜRER
4  2016-03-31 17:25:20                     Skoda_Fabia_1.4_TDI_PD_Classic
5  2016-04-04 17:36:23  BMW_316i___e36_Limousine___Bastlerfahrzeug__Ex…

   price abtest vehicleType  yearOfRegistration    gearbox  powerPS  model  \
1  18300   test       coupe                2011    manuell      190    NaN
2   9800   test         suv                2004  automatik      163  grand
3   1500   test  kleinwagen                2001    manuell       75   golf
4   3600   test  kleinwagen                2008    manuell       69  fabia
5    650   test   limousine                1995    manuell      102    3er

   kilometer  monthOfRegistration fuelType       brand notRepairedDamage  \
1     125000                    5   diesel        audi                ja
2     125000                    8   diesel        jeep               NaN
3     150000                    6   benzin  volkswagen              nein
4      90000                    7   diesel       skoda              nein
5     150000                   10   benzin         bmw                ja

           dateCreated  nrOfPictures  postalCode             lastSeen
1  2016-03-24 00:00:00             0       66954  2016-04-07 01:46:50
2  2016-03-14 00:00:00             0       90480  2016-04-05 12:47:46
```

```
3   2016-03-17 00:00:00            0        91074  2016-03-17 17:40:17
4   2016-03-31 00:00:00            0        60437  2016-04-06 10:17:21
5   2016-04-04 00:00:00            0        33775  2016-04-06 19:17:07
```

[24]:
```
df.columns
```

[24]:
```
Index(['dateCrawled', 'name', 'price', 'abtest', 'vehicleType',
       'yearOfRegistration', 'gearbox', 'powerPS', 'model', 'kilometer',
       'monthOfRegistration', 'fuelType', 'brand', 'notRepairedDamage',
       'dateCreated', 'nrOfPictures', 'postalCode', 'lastSeen'],
      dtype='object')
```

[25]:
```
df.drop(['name', 'abtest', 'dateCrawled','nrOfPictures', 'lastSeen',␣
 ↪'postalCode', 'dateCreated'], axis='columns', inplace=True)
```

[26]:
```
df
```

[26]:
```
        price vehicleType  yearOfRegistration    gearbox  powerPS  \
1       18300       coupe                2011    manuell      190
2        9800         suv                2004  automatik      163
3        1500  kleinwagen                2001    manuell       75
4        3600  kleinwagen                2008    manuell       69
5         650   limousine                1995    manuell      102
…         …         …                 …        …       …
371520   3200   limousine                2004    manuell      225
371524   1199      cabrio                2000  automatik      101
371525   9200         bus                1996    manuell      102
371526   3400       kombi                2002    manuell      100
371527  28990   limousine                2013    manuell      320

              model  kilometer  monthOfRegistration fuelType       brand  \
1               NaN     125000                    5   diesel        audi
2             grand     125000                    8   diesel        jeep
3              golf     150000                    6   benzin  volkswagen
4             fabia      90000                    7   diesel       skoda
5               3er     150000                   10   benzin         bmw
…               …         …                   …      …        …
371520         leon     150000                    5   benzin        seat
371524       fortwo     125000                    3   benzin       smart
371525  transporter     150000                    3   diesel  volkswagen
371526         golf     150000                    6   diesel  volkswagen
371527      m_reihe      50000                    8   benzin         bmw

       notRepairedDamage
1                     ja
2                    NaN
3                   nein
```

```
4                  nein
5                    ja
…                   …
371520               ja
371524             nein
371525             nein
371526              NaN
371527             nein

[309166 rows x 11 columns]
```

[27]:
```python
new_df = df.copy()
```

[28]:
```python
new_df.columns
```

[28]:
```
Index(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
       'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand',
       'notRepairedDamage'],
      dtype='object')
```

[29]:
```python
new_df.head()
```

[29]:
```
   price vehicleType  yearOfRegistration    gearbox  powerPS  model  \
1  18300       coupe                2011    manuell      190    NaN
2   9800         suv                2004  automatik      163  grand
3   1500  kleinwagen                2001    manuell       75   golf
4   3600  kleinwagen                2008    manuell       69  fabia
5    650   limousine                1995    manuell      102    3er

   kilometer  monthOfRegistration fuelType       brand notRepairedDamage
1     125000                    5   diesel        audi                ja
2     125000                    8   diesel        jeep               NaN
3     150000                    6   benzin  volkswagen              nein
4      90000                    7   diesel       skoda              nein
5     150000                   10   benzin         bmw                ja
```

[30]:
```python
new_df = new_df.drop_duplicates(['price', 'vehicleType',␣
↪'yearOfRegistration','gearbox', 'powerPS', 'model', 'kilometer',␣
↪'monthOfRegistration', 'fuelType' , 'notRepairedDamage'])
```

[31]:
```python
new_df.shape
```

[31]:
```
(285140, 11)
```

[32]:
```python
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)
```

[33]:
```
new_df.head()
```

[33]:
```
   price vehicleType  yearOfRegistration    gearbox  powerPS  model  \
1  18300       coupe                2011     manual      190    NaN
2   9800         suv                2004  automatic      163  grand
3   1500   small car                2001     manual       75   golf
4   3600   small car                2008     manual       69  fabia
5    650   limousine                1995     manual      102    3er

   kilometer  monthOfRegistration fuelType       brand notRepairedDamage
1     125000                    5   diesel        audi               Yes
2     125000                    8   diesel        jeep               NaN
3     150000                    6   petrol  volkswagen                No
4      90000                    7   diesel       skoda                No
5     150000                   10   petrol         bmw               Yes
```

[34]:
```
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)
```

[35]:
```
new_df.head()
```

[35]:
```
   price vehicleType  yearOfRegistration    gearbox  powerPS         model  \
1  18300       coupe                2011     manual      190  not-declared
2   9800         suv                2004  automatic      163         grand
3   1500   small car                2001     manual       75          golf
4   3600   small car                2008     manual       69         fabia
5    650   limousine                1995     manual      102           3er

   kilometer  monthOfRegistration fuelType       brand notRepairedDamage
1     125000                    5   diesel        audi               Yes
2     125000                    8   diesel        jeep      not-declared
3     150000                    6   petrol  volkswagen                No
4      90000                    7   diesel       skoda                No
5     150000                   10   petrol         bmw               Yes
```

[36]:
```
new_df.to_csv("/Users/rpriyadharshini/Desktop/autos_p.csv")
```

[37]:
```
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_)
    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)
```

```
Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
       'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
       'model_labels', 'brand_labels', 'fuelType_labels',
       'vehicleType_labels'],
      dtype='object')
```

[39]:
```
Y = labeled.iloc[:,0].values
X = labeled.iloc[:,1:].values
```

[40]:
```
X
```

[40]:
```
array([[  2011,    190, 125000, ...,      1,      1,      3],
       [  2004,    163, 125000, ...,     14,      1,      8],
       [  2001,     75, 150000, ...,     38,      7,      7],
       ...,
       [  1996,    102, 150000, ...,     38,      1,      0],
       [  2002,    100, 150000, ...,     38,      1,      1],
       [  2013,    320,  50000, ...,      2,      7,      4]])
```

[41]:
```
Y
```

[41]:
```
array([18300,  9800,  1500, ...,  9200,  3400, 28990])
```

[42]:
```
Y = Y.reshape(-1,1)
```

[43]:
```
print(X.shape,Y.shape)
```

```
(278575, 10) (278575, 1)
```

[44]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X ,Y ,test_size = 0.
↪3,random_state = 3)
```

[45]:
```
X_train
```

[45]:
```
array([[  2009,     90, 150000, ...,     10,      1,      0],
       [  2006,    150, 150000, ...,      2,      1,      4],
       [  1999,    102, 150000, ...,     20,      7,      4],
       ...,
       [  1994,    102, 150000, ...,      2,      7,      4],
       [  1997,    170, 125000, ...,     20,      7,      4],
       [  2012,    313,  50000, ...,      1,      1,      1]])
```

[46]:
```
multiple_lin_reg = LinearRegression()
multiple_lin_reg.fit(X_train,Y_train)
```

[46]:
```
LinearRegression()
```

[47]:
```
y_pred_mlr = multiple_lin_reg.predict(X_test)
```

[48]:
```
mae = mean_absolute_error(Y_test, y_pred_mlr)
mse = mean_squared_error(Y_test, y_pred_mlr)
rmse = np.sqrt(mse)
rmsle = np.log(rmse)
n,k = X_train.shape
r2=r2_score(Y_test,y_pred_mlr)
adj_r2= 1 - ((1-r2)*(n-1)/(n-k-1))
print(mae,mse,rmse,rmsle,r2,adj_r2)
```

```
3137.89414643729 29434090.807211053 5425.3194198324445 8.598832055907133
0.5759870208048179 0.5759652755458473
```

[49]:
```
regressor = RandomForestRegressor(n_estimators␣
↪=1000,max_depth=10,random_state=34)
```

[51]:
```
regressor.fit(X_train,np.ravel(Y_train,order='C'))
```

[51]:
```
RandomForestRegressor(max_depth=10, n_estimators=1000, random_state=34)
```

[52]:
```
y_pred = regressor.predict(X_test)
```

[53]:
```
mae = mean_absolute_error(Y_test, y_pred)
mse = mean_squared_error(Y_test, y_pred)
rmse = np.sqrt(mse)
rmsle = np.log(rmse)
n,k = X_train.shape
r2=r2_score(Y_test,y_pred)
```

```
adj_r2= 1 - ((1-r2)*(n-1)/(n-k-1))
print(mae,mse,rmse,rmsle,r2,adj_r2)
```

```
1624.2919043250051 10729873.982609846 3275.648635401826 8.094271185069976
0.8454307332421266 0.8454228062472008
```

[54]:
```
filename = '/Users/rpriyadharshini/Desktop/mod.sav'
pickle.dump(regressor,open(filename,'wb'))
```

[ ]:
```

```

TESTING

TEST CASES

TEST CASES	EXPECTED OUTCOME	PASS/FAIL
Check if data can be entered in the 'registration year' text box	Data can be entered properly	PASS
Check if data can be entered in the 'registration month' text box	Data can be entered properly	PASS
Check if data can be entered in the 'Power of car' text box	Data can be entered properly	PASS
Check if data can be entered in the 'kilometers' text box	Data can be entered properly	PASS
Check if radio buttons are working for 'Gear type'	Radio buttons are working	PASS
Check if radio buttons are working for 'is car damaged'	Radio buttons are working	PASS
Check if the drop down list is showing for 'model type'	Drop down list is showing	PASS
Check if the drop down list is showing for	Drop down list is showing	PASS

'brand'		
Check if the drop down list is showing for 'fuel type'	Drop down list is showing	PASS
Check if the drop down list is showing for 'vehicle type'	Drop down list is showing	PASS
Check if the 'Submit' Button is working	'Submit' button is working	PASS

USER ACCEPTANCE TESTING

USER TEST CASES	EXPECTED OUTCOME	PASS/FAIL
Login	User can login and access the account	PASS
Dashboard	User can view and access the dashboard	PASS
Data Entry	User can enter the details of the car	PASS
Predicted Output	User can view the predicted price of the car	PASS
ML Model	User can modify or upgrade the model	PASS
Website	User can access the website	PASS

RESULTS

PERFORMANCE METRICS

```
mae = mean_absolute_error(Y_test, y_pred)
mse = mean_squared_error(Y_test, y_pred)
rmse = np.sqrt(mse)
rmsle = np.log(rmse)
n,k = X_train.shape
r2=r2_score(Y_test,y_pred)
```

```
adj_r2= 1 - ((1-r2)*(n-1)/(n-k-1))
print(mae,mse,rmse,rmsle,r2,adj_r2)
```

```
1624.2919043250051 10729873.982609846 3275.648635401826 8.094271185069976
0.8454307332421266 0.8454228062472008
```

ADVANTAGES AND DISADVANTAGES

ADVANTAGES

The main objective of this project is to predict the resale values of cars. Since it is becoming essential for the users to know the correct price of the car for resaling,this proposed solution will help them to get accurate predictions for the prices of their cars.This will help them to easily predict the price and sell their cars.

DISADVANTAGES

The prices of the cars will be predicted based on the model created using the datasets given.So, as time passes, the accuracy of predicted prices may be little low as the value of the cars will be constantly changing.This is one disadvantage of this proposed solution.

CONCLUSION

In this project,we have proposed a solution to predict the prices of the cars for resaling.The users have to enter the details of their cars .A model will be trained based on the datasets.Based on the trained model ,the predicted prices for the cars of the users will be displayed as the final result.This is done using random forest algorithm.Thus this proposed solution will help the users to accurately predict the prices of their cars.

FUTURE SCOPE

The prices of the cars will be predicted based on the model created using the datasets given.So, as time passes, the accuracy of predicted prices may be little low as the value of the cars will be constantly changing.This will a disadvantage .So,the solution sholud be given in such a way that , as time goes on the model can be upraded using new datasets. This will definetly increase the accuacy of the model.

APPENDIX

GITHUB LINK

https://github.com/IBM-EPBL/IBM-Project-4978-1658744652

PROJECT DEMO LINK

https://drive.google.com/file/d/1kBEudJ7v6Gx6PVl90CXah4dyEdDerAXv/view?usp=sharing