CAR RESALE VALUE PREDICTION

TEAM ID: PNT2022TMID04349

Team Details

AKASH M	737819CSR006
CHARANRAJ T	737819CSR021
CIBIKUMAR M V	737819CSR022
DANUSHRAJ K S	737819CSR023

INDEX

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.

In order 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.

1.2 PURPOSE

Predicting the price of used cars in both an important and interesting problem. According to data obtained from the National Transport Authority [1], the number of cars registered between 2003 and 2013 has witnessed a spectacular increase of 234%. From 68, 524 cars registered in 2003, this number has now reached 160, 701. With difficult economic conditions, it is likely that sales of second-hand imported (reconditioned) cars and used cars will increase. It is reported in [2] that the sales of new cars has registered a decrease of 8% in 2013. In many developed countries, it is common to lease a car rather than buying it outright. A lease is a binding contract between a buyer and a seller (or a third party – usually a bank, insurance firm or other financial institutions) in which the buyer must pay fixed instalments for a pre-defined number of months/years to the seller/financer. 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 make the predictions. Keywords-car; price; machine learning; artificial intelligence 754 Sameerchand Pudaruth seller/financers to be able to predict the salvage value (residual value) of cars with accuracy. If the residual value is under-estimated by the seller/financer at the beginning, the instalments will be higher for the clients who will certainly then opt for another seller/financer. If the residual value is over-estimated, the instalments will be lower for the clients but then the seller/financer may have much difficulty at selling these high-priced used cars at this over-estimated residual value. Thus, we can see that estimating the price of used cars is of very high commercial importance as well. Manufacturers' from Germany made a loss of 1 billion Euros in their USA market because of mis-calculating the residual value of leased cars [3]. Most individuals in Mauritius who buy new cars are also very apprehensive about the resale value of their cars after certain number of years when they will possibly sell it in the used cars market. Predicting the resale value of a car is not a simple task. It is trite knowledge that the value of used cars depends on a number of factors. The most important ones are usually the age of the car, its make (and model), the origin of the car (the original country of the manufacturer), its mileage (the number of kilometers it has run) and its horsepower. Due to rising fuel prices, fuel economy is also of prime importance. Unfortunately, in practice, most people do not know exactly how much fuel their car consumes for each km driven. Other factors such as the type of fuel it uses, the interior style, the braking system, acceleration, the volume of its cylinders (measured in cc), safety index, its size, number of doors, paint colour, weight of the car, consumer reviews, prestigious awards won by the car manufacturer, its physical state, whether it is a sports car, whether it has cruise control, whether it is automatic or manual transmission, whether it belonged to an individual or a company and other options such as air conditioner, sound system, power steering, cosmic wheels, GPS navigator all may influence the price as well. Some special factors which buyers attach importance in Mauritius is the local of previous owners, whether the car had been involved in serious accidents and whether it is a lady-driven car. The look and feel of the car certainly contributes a lot to the price. As we can see, the price depends on a large number of factors. Unfortunately, information about all these factors are not always available and the buyer must make the decision to purchase at a certain price based on few factors only.

The main objective of this project is to predict the amount of resale and thereby eliminating the human intervention and biased valuation. So this project is made with the purpose of predicting the correct valuation of used cars that helps users to sell the car remotely with perfect valuation. It predicts the amount of resale value based on the parameters provided by the user. User enters the details of the car into the form given and accordingly the car resale value is predicted.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

- Customers who want to purchase a used automobile are vulnerable to fraud since they do not have a thorough understanding of the vehicles.
- The cost of new automobiles might be expensive since the government must pay some additional expenses in the form of taxes. So, for those just starting out, purchasing a used automobile will be the best option.
- Customers run the risk of paying excessive prices for old-model or old-brand vehicles that aren't really worth the money.
- If buyers are uninformed about the models and brands of the used automobiles being sold, they risk being duped.
- Although there are websites that provide information on automobiles, they might not provide an estimate of the cost of the car.
- Customers need a reliable website or model to estimate the cost of a secondhand car.

2.2 REFERENCES

1. **CAR RESALE PRICE FORECASTING**[Stefan Lessmann, Stefan Vob, 2017]

Resale price forecasting is first done with Random Forest Regression. Then the same price forecastign is done with externally generated residual value estimates and finally the two results are compared to determine the best approach.

2. **PREDICTION OF RESALE VALUE OF THE CAR USING LINEAR REGRESSION ALGORITHM** [Kiran S, 2020]

A correlation with each attribute to that of target attribute is found and linear regression curve with the target attribute is drawn. As a final step the total error and accuracy is measured

3. **CAR PRICE PREDICTION IN THE USA BY USING LINER REGRESSION** [Huseyn Mammadov, 2021]

They proposed a model using linear regression since the dependent variable price is linearly related to many independent variables and they have eliminated the irrelevant features by using the recursive feature elimination to reduce the dimensionality. Then R-square and root mean squared error is used to reduce the errors produced.

4. **PREDICTING THE PRICE OF USED CARS USING MACHINE LEARNING TECHNIQUES** [Sameerchand Pudaruth, 2013]

Different techniques like multiple linear regression analysis, k-nearest neighbors, naïve bayes and decision trees have been used to make the predictions. The predictions are then evaluated and compared in order to find those which provide the best performances.

5. **USED CARS PRICE PREDICTION USING SUPERVISED LEARNING TECHNIQUES**[Pattabiraman Venkatasubbu, Mukkesh Ganesh, 2019]

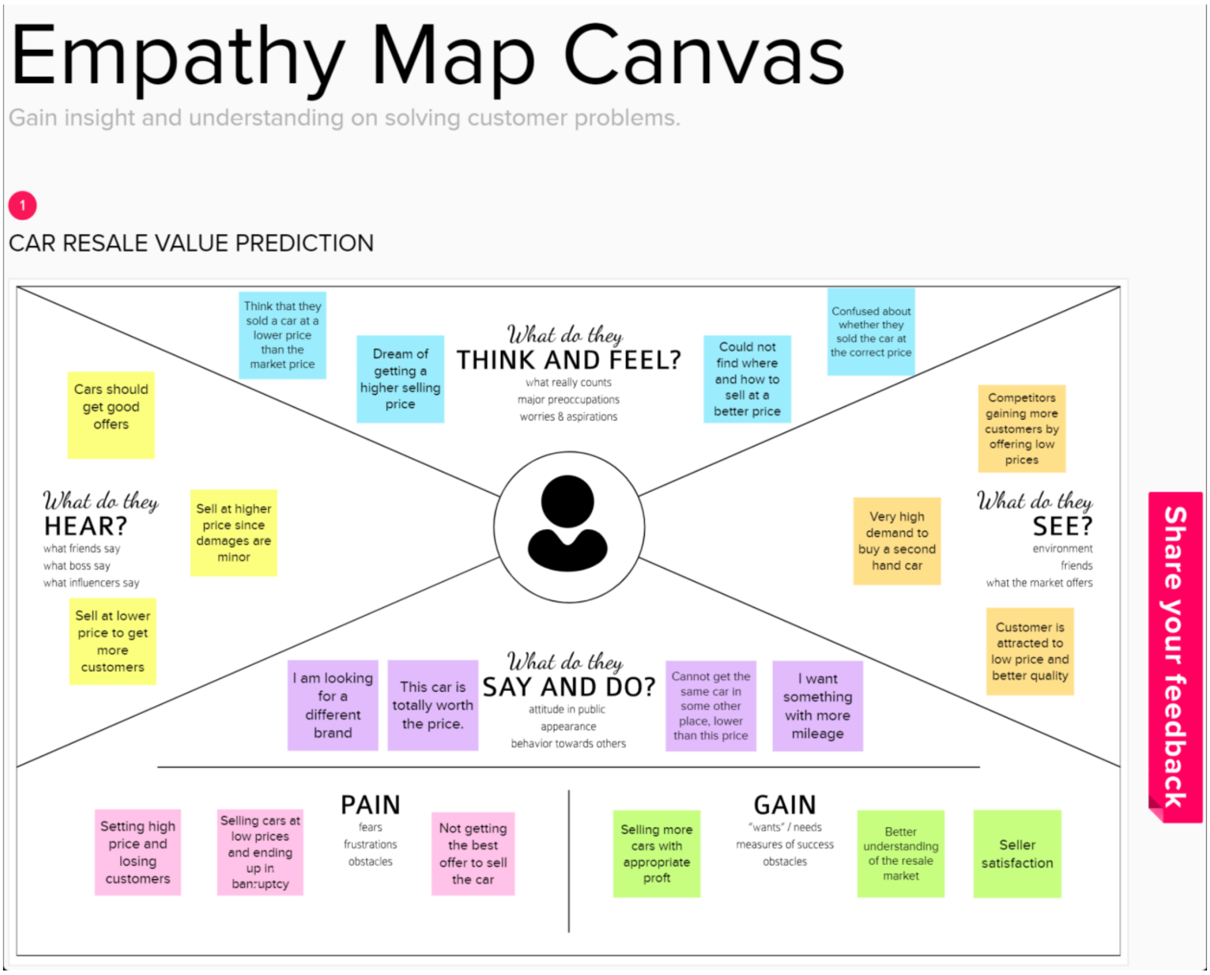
They proposed a model using multiple and lasso regression. Using Lasso regression on the training data set, we first select the subset of attributes that lead to less error while predicting the price. It makes use of 10-fold cross-validation and L1 regularization. A general linear model, which models price to the set of selected attributes from lasso regression is used for multiple regression training.

2.3 PROBLEM STATEMENT DEFINITION

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Car dealer	sell used cars	Increasing price	rising interest rates, tariffs, and energy concerns, car dealerships are expected to have fewer sales, especially with newer vehicles	Sad and Worried
PS-2	Common People	Buy 2nd hand cars	Can't decide on cars	Too many car models & prices. Common man can't decide correct car.	Fear of making wrong choice
PS-3	Budget Oriented People	Buy cheap cars	Price is not justified	Price is increased and cannot justify price	Paying more for cars
PS-4	Seller	To sell my car at reasonable price	Deciding on the price is hard	Too many complications in calculating the correct price for selling cars	Unhappy for not selling car at correct price.

3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas



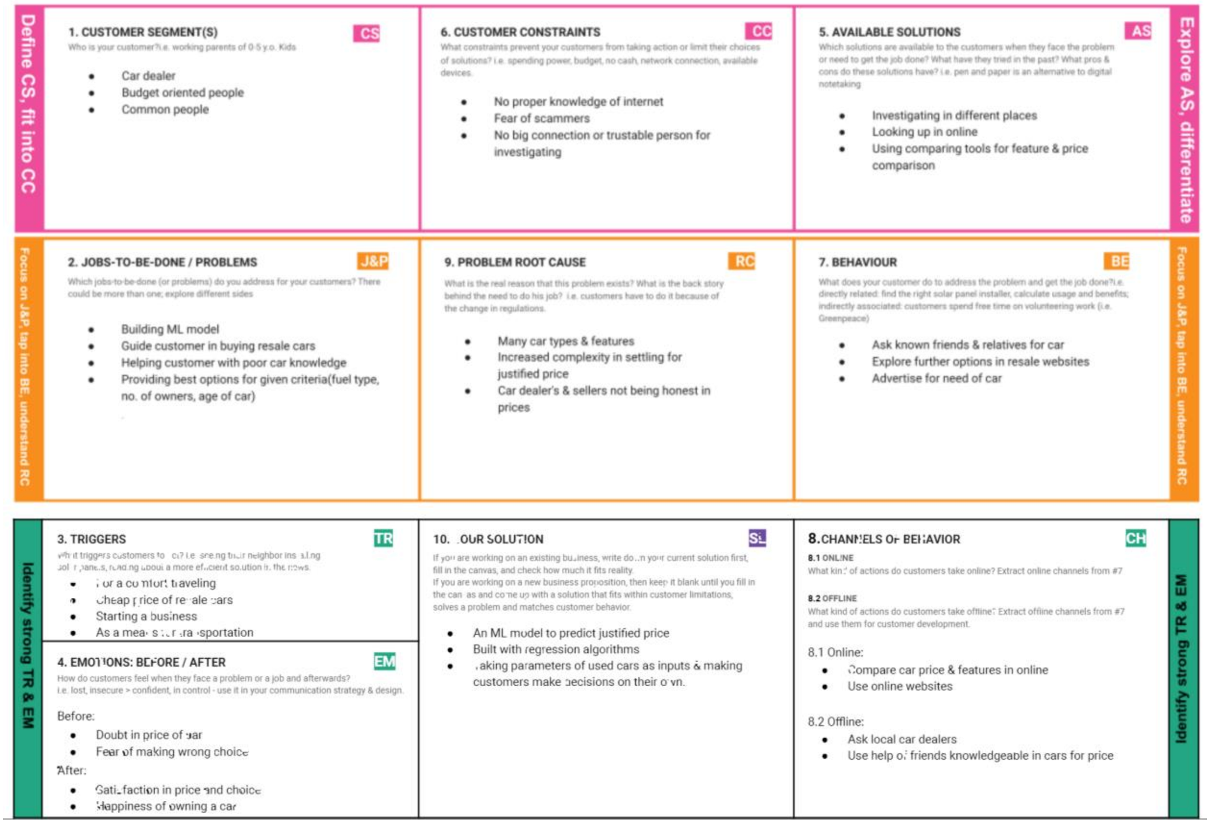
3.2 IDEATION & BRAINSTORMING


Collect the cost price of car
Analyzing current condition of vehicle
Using ML Model
Getting user feedback
Analyze the engine condition
Integrating AI Algorithms To Improve accuracy
Collect car damage details
Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Getting user feedback
Get accident history
Analyze economic conditions of car
Filter Options Based On User Preference
Cost of fixing damages in old car
TIP
Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the H key on the keyboard.
User support/ Query center
Seek friends who have knowledge about cars
Discuss with car dealers to get idea
Survey to get personalized results
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	• The main aim of this project is to predict the resale value of a used car using regression algorithms. • This could help the customers to find the best price of the used car that is going to be sold.
2.	Idea / Solution description	• The resale value of a car depends on factors such as price, vehicle type, gearbox, model, kilometres run, fuel type, etc. • The data is then pre-processed to handle missing values and outliers, to normalize the data and split it into dependent and independent variables. • After that the model is developed using regression algorithms to predict the resale price of the car.
3.	Novelty / Uniqueness	• This is a real-time problem which can benefit both customer and seller. • The novelty of this proposal is to predict the resale value as near as possible to the actual value.
4.	Social Impact / Customer Satisfaction	• Provided the current economic times, it is more likely that the usage of second-hand cars will increase. • This is a mutual commercial interest to both the customers and the sellers. • It predicts the resale values of the car based on all its features and prevents over-pricing or under-pricing. • This sets an understanding or trust between the seller and the customer.
5.	Business Model (Revenue Model)	• The proposed model could be sold to resellers so that they could use it to find the perfect price for bidding. • It could be developed into an application and get revenue from it if more no of users started to using it to find the best value of a second-hand car.
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.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1) FUNCTIONAL REQUIREMENT

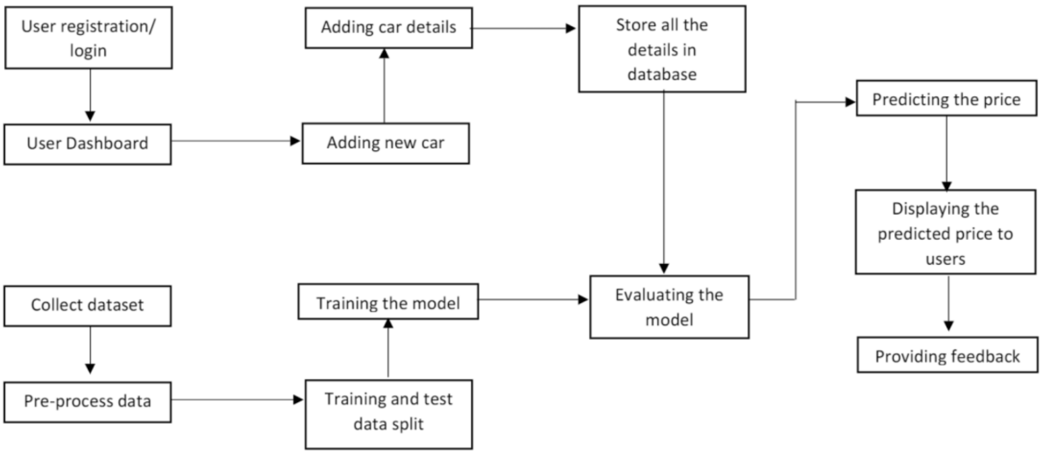
FR No	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form
FR-2	User Confirmation	Confirmation via Email
FR-3	User Profile	View account details
FR-4	Car predictions	View the previous predicted prices along with model, brand and vehicle type
FR-5	Maintain database	Maintain database to store user and their car details
FR-6	Value prediction	Predict the value of the resale car using the regression model and details entered
FR-7	Result display	Display the predicted value of the used car

4.2) NON-FUNCTIONAL REQUIREMENTS

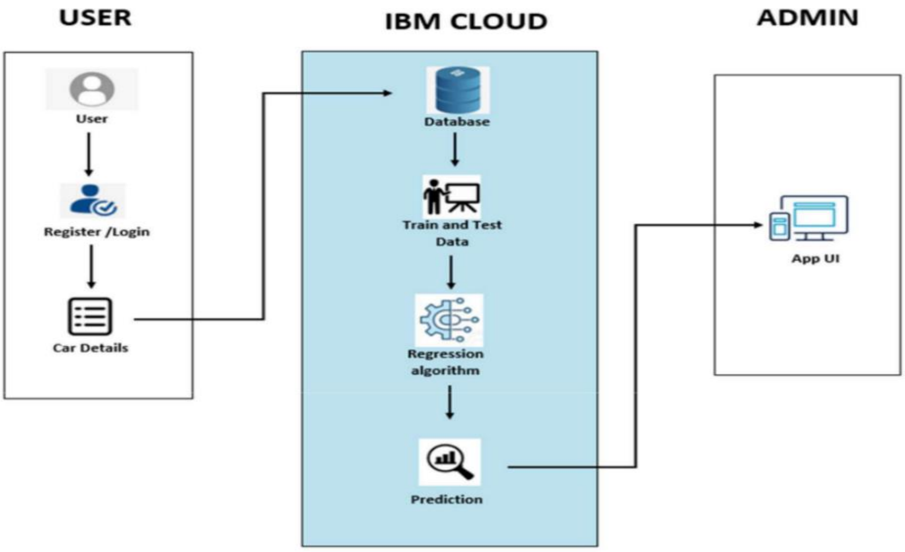
FR No	Non-Functional Requirement	Description
NFR-1	Usability	• User friendly UI • Easy process flow to predict value
NFR-2	Security	• User authentication while entering website • No information is shared with third party • User can see only his details
NFR-3	Reliability	• Data will be stored and replicated so that data loss can be avoided • Rate of occurrence of failure is very less
NFR-4	Performance	• Quick prediction results • Fast website loading • Efficient ML algorithm to provide accurate result with less time complexity
NFR-5	Availability	• Application can be accessed from both mobile and desktop • Single page failure does not affect the whole website • Uninterrupted user services
NFR-6	Scalability	• Able to handle large amount of data and traffic globally without failure • Database can be scaled according to the usage in a cost effective manner

5) PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Admin	Dataset	USN-1	Collect the required data for the Car resale prediction.	Enough data collected for training model	High	Sprint-1
	Data preprocessing	USN-2	Perform data cleaning to optimize the dataset	Clean Dataset enough to make correct predictions	High	Sprint-1
	Training & Building Model	USN-3	Build the model using regression algorithms to classify the data	Model should be predicting prices with acceptable accuracy	High	Sprint-1
	Deploy the model	USN-4	Deployment of ML model using IBM Cloud	Model should be working fine from the cloud	High	Sprint-2
	Integrate the web app with the IBM model	USN-5	Use flask for the integration purpose.	Model should be easy to use & working fine from the web app	High	Sprint-2
Customer	Homepage	USN-6	Details about the application and the car resale process	I can get an idea about the app	Medium	Sprint-2
	Registration	USN-7	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-3
		USN-8	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email	High	Sprint-3
	Login	USN-9	As a user, I can log into the application by entering email & password	I can login to my account	High	Sprint-3
	Dashboard	USN-10	As a user, I can add new cars and get access to insert and update their details	I can add new cars	Medium	Sprint-4
	Car Details	USN-11	As a user, I should give the car details like car model, engine and fuel type, etc...	Car details should be accepted & taken for further processing	High	Sprint-4
	Car Price	USN-12	As a user, I can view the current rate of the used car price	Car Prices must be shown based on the predicted result by the model	High	Sprint-4

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

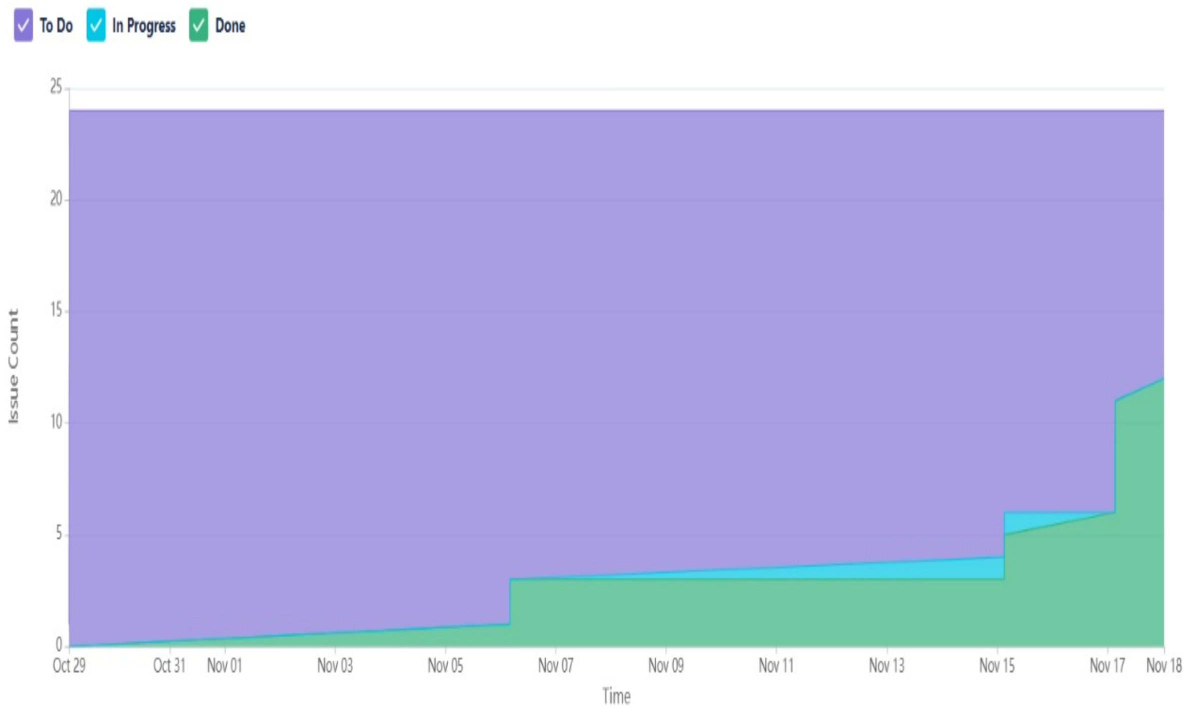
TITLE	DESCRIPTION	DUE DATE
Literature survey & Information gathering	Literature survey on the selected topic and collect information by referring to the related papers and research projects, journals etc.	3 September 2022
Prepare Empathy Map	Prepare empathy map canvas to understand about the user problems, pains and gains. From the empathised details, prepare the problem statements to be solved.	10 September 2022
Ideation	Conduct a brainstorming session with the teammates and discuss ideas to solve the problem. Prioritize the top 3 ideas based on feasibility.	17 September 2022
Proposed Solution	Prepare the proposed solution which includes the novelty, feasibility, revenue, social impact, scalability etc.	24 September 2022
Problem Solution Fit	Prepare the problem solution fit which includes the causes, problems and solutions of the problem.	1 October 2022
Solution Architecture	Prepare solution architecture that indicates the data flow from the user, model and the website.	1 October 2022

Customer Journey	Prepare the customer journey map to understand the user needs and experience with the application.	8 October 2022
Functional Requirement	Prepare the functional requirement which includes all the features that will be available in the application.	15 October 2022
Technology Architecture	Prepare the technology architecture that defines about the technologies and the IBM cloud features used in the application.	15 October 2022
Data Flow Diagrams	Draw the data flow diagram to indicate the data flow from the user, during the model building and while predicting the result,	15 October 2022
Prepare Milestone & Activity List	Split the entire project into simpler tasks and prepare milestones and activity list of the project.	22 October 2022
Sprint Delivery Plan	Prepare a delivery plan of the project with specific due dates to complete each sprint consisting of a set of functional requirements.	22 October 2022
Project Development - Delivery of Sprint-1, 2, 3 & 4	Develop, test and submit the code.	19 November 2022

6.2 SPRINT DELIVERY SCHEDULE

User	Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Admin	Sprint 1	Dataset collection	USN-1	Collect the required data for the Car resale prediction	2	High	Cibikumar, Charanraj, Akash, Danushraj
	Sprint 1	Data pre-processing	USN-2	Perform data cleaning to optimize the dataset	4	Medium	Charanraj, Cibikumar
	Sprint 1	Training & Building Model	USN-3	Build the model using regression algorithms to classify the data	6	High	Cibikumar, Charanraj, Akash, Danushraj
	Sprint 2	Deploy the model	USN-4	Deployment of ML model using IBM Cloud	5	High	Akash, Danushraj
	Sprint 4	Integration	USN-5	Integrate the web app developed using flask with IBM model	5	High	Charanraj, Cibikumar
Customer	Sprint 2	Homepage	USN-6	Details about the application and the car resale process	2	Low	Akash, Danushraj
	Sprint 2	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	5	High	Cibikumar, Danushraj
	Sprint 3	Confirmation	USN-8	As a user, I will receive confirmation email once I have registered for the application	3	Medium	Charanraj, Akash
	Sprint 3	Login	USN-9	As a user, I can log into the application by entering email & password	4	High	Cibikumar, Danushraj
	Sprint 3	Dashboard	USN-10	As a user, I can add new cars and get access to insert and update their details	5	High	Cibikumar, Charanraj, Akash
	Sprint 4	Car Details	USN-11	As a user, I should give the car details like car model, engine and fuel type, etc…	2	Medium	Akash, Danushraj
	Sprint 4	Car Price	USN-12	As a user, I can view the current rate of the used car price	5	High	Charanraj, Akash, Danushraj

6.3) REPORTS FROM JIRA



7. CODING & SOLUTIONING

7.1 FEATURE-1 (MODEL CREATION)

Importing the libraries

In [1]:
```
# importing libraries

import pandas as pd
import numpy as np
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
import pickle
```

In [2]:
```
#importing dataset

df=pd.read_csv("Data/autos.csv",header=0,sep=",",encoding="Latin1")
```

Cleaning the dataset

In [3]:
```
df.dtypes
```

Out[3]:
```
dateCrawled            object
name                   object
seller                 object
offerType              object
price                   int64
abtest                 object
vehicleType            object
yearOfRegistration      int64
gearbox                object
powerPS                 int64
model                  object
kilometer               int64
monthOfRegistration     int64
fuelType               object
brand                  object
notRepairedDamage      object
dateCreated            object
nrOfPictures            int64
postalCode              int64
lastSeen               object
dtype: object
```

In [4]:
```
print(df.seller.value_counts())
df[df.seller!='gewerblich']
df=df.drop('seller',1)

print(df.offerType.value_counts())
df[df.offerType!='Gesuch']
df=df.drop('offerType',1)
```

```
privat        371525
gewerblich         3
Name: seller, dtype: int64
```

```
C:\Users\charan\AppData\Local\Temp\ipykernel_9104\587171588.py:3: FutureWarning: In a
future version of pandas all arguments of DataFrame.drop except for the argument 'lab
els' will be keyword-only.
  df=df.drop('seller',1)
Angebot    371516
Gesuch         12
Name: offerType, dtype: int64
C:\Users\charan\AppData\Local\Temp\ipykernel_9104\587171588.py:7: FutureWarning: In a
future version of pandas all arguments of DataFrame.drop except for the argument 'lab
els' will be keyword-only.
  df=df.drop('offerType',1)
```

In [5]:
```
print(df.shape)
df=df[(df.powerPS>50) & (df.powerPS)<900]
print(df.shape)
df=df[(df.yearOfRegistration>=1950)&(df.yearOfRegistration<2017)]
print(df.shape)
```

```
(371528, 18)
(371528, 18)
(356559, 18)
```

In [6]:
```
df.drop(['name','abtest','dateCrawled','nrOfPictures','lastSeen',
'postalCode','dateCreated'],axis='columns',inplace=True)
```

In [7]:
```
new_df=df.copy()
new_df=new_df.drop_duplicates(['price','vehicleType','yearOfRegistration',
'gearbox','powerPS','model','kilometer','monthOfRegistration','fuelType',
'notRepairedDamage'])
```

In [8]:
```
new_df.gearbox.replace(('manuell','automatik'),('manual', 'automatic'), inplace=True)
new_df.fuelType.replace(('benzin','andere','elektro'),('petrol','others','electric'),
new_df.vehicleType.replace(('kleinwagen','cabrio','kombi','andere'),('small car','conv
new_df.notRepairedDamage.replace(('ja','nein'),('Yes','No'),inplace=True)
```

In [9]:
```
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)
```

In [10]:
```
new_df.to_csv('autos_preprocessed.csv')
```

In [11]:
```
new_df
```

Out[11]:

	price	vehicleType	yearOfRegistration	gearbox	powerPS	model	kilometer	monthOf
0	480	not-declared	1993	manual	0	golf	150000	
1	18300	coupe	2011	manual	190	not-declared	125000	
2	9800	suv	2004	automatic	163	grand	125000	
3	1500	small car	2001	manual	75	golf	150000	
4	3600	small car	2008	manual	69	fabia	90000	
...	...	...	...	...	...	...	...	
371523	2200	not-declared	2005	not-declared	0	not-declared	20000	
371524	1199	convertible	2000	automatic	101	fortwo	125000	
371525	9200	bus	1996	manual	102	transporter	150000	
371526	3400	combination	2002	manual	100	golf	150000	
371527	28990	limousine	2013	manual	320	m_reihe	50000	

317379 rows × 11 columns

In [12]:

```
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','monthOfRegistra

print(labeled.columns)
```

```
gearbox : LabelEncoder()
notRepairedDamage : LabelEncoder()
model : LabelEncoder()
brand : LabelEncoder()
fuelType : LabelEncoder()
vehicleType : LabelEncoder()
Index(['price', 'yearOfRegistration', 'powerPS', 'kilometer',
       'monthOfRegistration', 'gearbox_labels', 'notRepairedDamage_labels',
       'model_labels', 'brand_labels', 'fuelType_labels',
       'vehicleType_labels'],
      dtype='object')
```

In [ ]:

Splitting Data Into Independent And Dependent Variables

In [13]:
```
Y = labeled.iloc[:,0].values
X = labeled.iloc[:,1:].values

Y = Y.reshape(-1,1)
```

In [14]:
```
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=
```

In [15]:
```
X_train,X_test,Y_train,Y_test
```

Out[15]:
```
(array([[  2009,    101,  40000, ...,     36,      7,      4],
        [  1998,    115, 150000, ...,     10,      5,      1],
        [  2003,    109, 150000, ...,      1,      7,      7],
        ...,
        [  2005,    209, 150000, ...,     39,      7,      8],
        [  2007,    143, 150000, ...,      2,      7,      4],
        [  1999,    136, 150000, ...,      2,      7,      4]], dtype=int64),
 array([[  2006,    140, 100000, ...,     24,      7,      4],
        [  2001,    179, 150000, ...,      1,      1,      1],
        [  1999,    211, 150000, ...,     24,      7,      1],
        ...,
        [  2003,    113, 150000, ...,     27,      4,      7],
        [  1998,    140, 150000, ...,     39,      7,      4],
        [  1994,     75, 150000, ...,     38,      7,      2]], dtype=int64),
 array([[ 7499],
        [  450],
        [ 2990],
        ...,
        [10500],
        [ 6995],
        [ 1899]], dtype=int64),
 array([[5990],
        [2999],
        [ 899],
        ...,
        [2700],
        [ 850],
        [1000]], dtype=int64))
```

Building the model

In [17]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score

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

rf_regressor.fit(X_train, np.ravel(Y_train, order='C'))
```

Out[17]:

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

In [18]:

```
y_pred = rf_regressor.predict(X_test)

print(r2_score(Y_test, y_pred))
```

```
0.8191322832483275
```

In [20]:

```
filename='resale_model.sav'
pickle.dump(rf_regressor,open(filename,'wb'))
```

Accuracy metrics

In [21]:

```
from sklearn.metrics import mean_absolute_error

print(mean_absolute_error(Y_test, y_pred))
```

```
1655.534681561534
```

In [22]:

```
from sklearn.metrics import mean_squared_error

print(mean_squared_error(Y_test, y_pred))
```

```
11832644.335139675
```

In [23]:

```
from sklearn.metrics import mean_squared_error

root_mean_squared_error = mean_squared_error(Y_test, y_pred, squared=False)
print(root_mean_squared_error)
```

```
3439.861092419238
```

In [24]:

```
from sklearn.metrics import r2_score

print(r2_score(Y_test, y_pred))
```

```
0.8191322832483275
```

7.2 FEATURE-2 (FLASK)

```
import pandas as pd
import numpy as np
from datetime import timedelta
import matplotlib as plt
from flask import Flask, render_template, session, request, redirect
from sklearn.preprocessing import LabelEncoder
import pickle
import requests
import pyrebase

app = Flask(__name__)

firebaseConfig = {
  "apiKey": "FIREBASE_API_KEY",
  "authDomain": "carresale-fd630.firebaseapp.com",
  "databaseURL": "https://carresale-fd630-default-rtdb.asia-southeast1.firebasedatabase.app",
  "projectId": "carresale-fd630",
  "storageBucket": "carresale-fd630.appspot.com",
  "messagingSenderId": "217742759498",
  "appId": "1:217742759498:web:8c55a5fa220ca091fa38b3",
  "measurementId": "G-VHBR6L69Q1"
}

firebase = pyrebase.initialize_app(firebaseConfig)
db = firebase.database()
app.secret_key = 'secret'
app.permanent_session_lifetime = timedelta(minutes=60)

@app.route("/")
def home():
    return render_template('home.html')

@app.route("/auth")
def auth():
    return render_template('login.html')

@app.route("/login", methods=['POST'])
def login():
    auth = firebase.auth()
    email = request.form['email']
    password = request.form.get('password')
    try:
        user = auth.sign_in_with_email_and_password(email, password)
        userDetails = auth.get_account_info(user['idToken'])
        if {userDetails['users'][0]['emailVerified']} == {True}:
```

```
            session['user'] = user
        else:
            return "Verify email"
    except Exception as e:
        print(f'error:{e}')
    return redirect('/dashboard')

@app.route("/register", methods=['POST'])
def register():
    auth = firebase.auth()
    name = request.form['name']
    email = request.form['email']
    password = request.form.get('password')
    userDetails = {
        'name': name,
        'email': email,
        'history': False
    }
    try:
        user = auth.create_user_with_email_and_password(email, password)
        auth.update_profile(user['idToken'], display_name=name)
        db.child('users').child(user['localId']).set(userDetails, user['idToken'])
        auth.send_email_verification(user['idToken'])
    except Exception as e:
        print(f'error:{e}')
    return redirect('/dashboard')

@app.route('/logout')
def logout():
    if 'user' in session:
        session.pop('user',None)
        return redirect('/')
    else:
        return redirect('/')

@app.route("/dashboard")
def dashboard():
    if 'user' in session:
        user = dict(db.child('users').get().val())[session['user']['localId']]
        return render_template('history.html', name=session['user']['displayName'],history=user['history'])
    else:
        return redirect('/auth')

@app.route("/predict")
def predict():
    if 'user' in session:
        return render_template('predict.html', name=session['user']['displayName'],modelData=modelData,
fuelData=fuelData, vehicleData=vehicleData, brandData=brandData)
```

```
    else:
        return redirect('/auth')

def predictFromDeploymentModel(userInput):
    API_KEY = "IBM_CLOUD_API_KEY"
    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}
    payload_scoring = {"input_data": [{"fields":
['yearOfRegistration'  ,'powerPS'  ,'kilometer'    ,'monthOfRegistration'  ,'gearbox_labels',  'notRepairedDa
mage_labels', 'model_labels', 'brand_labels', 'fuelType_labels',  'vehicleType_labels'], "values":
[userInput]}]}
    response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/b0df73c1-
d3dd-4e66-8f0d-90534cf3fe4a/predictions?version=2022-11-14', json=payload_scoring,
    headers={'Authorization': 'Bearer ' + mltoken})
    predictions = response_scoring.json()
    return predictions['predictions'][0]['values'][0][0]

@app.route("/y_predict", methods=['GET', 'POST'])
def y_predict():
    if 'user' not in session:
        return redirect('/auth')

    regyear = int(request.form['regyear'])
    powerps = float(request.form['powerps'])
    kms = float(request.form['kms'])
    regmonth = int(request.form.get('month'))
    gearbox = request.form['gearbox']
    damage = request.form['dam']
    model = request.form.get('modeltype')
    brand = request.form.get('brand')
    fuelType = request.form.get('fuel')
    vehicletype = request.form.get('vehicletype')

    new_row = {'yearOfRegistration': regyear, 'powerPS': powerps, 'kilometer': kms, 'monthOfRegistration':
regmonth,
               'gearbox': gearbox, 'notRepairedDamage': damage, 'model': model, 'brand': brand, 'fuelType':
fuelType, 'vehicleType': vehicletype}
    new_df = pd.DataFrame(columns=['vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
                        'model', 'kilometer', 'monthOfRegistration', 'fuelType', 'brand', 'notRepairedDamage'])
    new_df = new_df.append(new_row, ignore_index=True)

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

    mapper = {}
```

```python
    for i in labels:
        mapper[i] = LabelEncoder()
        mapper[i].fit(np.load(str('classes'+i+'.npy'), allow_pickle=True))
        tr = mapper[i].transform(new_df[i])
        new_df.loc[:, i+'_labels'] = pd.Series(tr, index=new_df.index)

    labeled = new_df[['yearOfRegistration', 'powerPS', 'kilometer',
                      'monthOfRegistration']+[x+'_labels' for x in labels]]
    X = labeled.values
    res = predictFromDeploymentModel(list(X[0]))
    # y_prediction = model_rand.predict(X)

    data = [{
        'model': model,
        'brand': brand,
        'vehicle': vehicletype,
        'price': res
    }]

    user = dict(db.child('users').get().val())[session['user']['localId']]
    if(user['history']):
        user['history'].extend(data)
        db.child('users').child(session['user']['localId']).child('history').set(user['history'],
session['user']['idToken'])
    else:
        db.child('users').child(session['user']['localId']).child('history').set(data, session['user']['idToken'])

    return render_template('result.html', price="{:.2f}$".format(res))


if __name__ == '__main__':
    app.run(host='localhost', port=3001)
```

8. TESTING

8.1 TEST CASES

Sprint-1

MODEL USED = RANDOM FOREST REGRESSION MODEL

MAE (MEAN ABSOLUTE ERROR) – 1655.53

```
from sklearn.metrics import mean_absolute_error

print(mean_absolute_error(Y_test, y_pred))

1655.534681561534
```

MSE (MEAN SQUARED ERROR) –11832644.33

```
from sklearn.metrics import mean_squared_error

print(mean_squared_error(Y_test, y_pred))

11832644.335139675
```

RMSE (ROOT MEAN SQUARED ERROR) –3439.86

```
from sklearn.metrics import mean_squared_error

root_mean_squared_error = mean_squared_error(Y_test, y_pred, squared=False)
print(root_mean_squared_error)

3439.861092419238
```

R2 SCORE – 0.82

```
from sklearn.metrics import r2_score

print(r2_score(Y_test, y_pred))

0.8191322832483275
```

Sprint-2

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
Register_Page_01	UI	Landing Page	The Landing page must be responsive and the user must be redirected to register page once get started is clicked	1) Enter the URL and go 2) Click get started	https://crvp-anxious-fox-py.mybluemix.net/	The page should load and once the user clicks get started the user must be navigated to register page	Worked as expected	Pass	Akash
Register_Page_02	UI	Register Page	The Register page must allow the user to register to the website	1) Enter the URL and go 2) Click get started 3) Fill your credentials 4) Click register	https://crvp-anxious-fox-py.mybluemix.net/	The page should render three text boxes to fill name, email and password and the user should also be able to click register	Worked as expected	Pass	Danushraj
Register_Page_03	Functional	Register Page	The Register page should register the user to backend service	1) Enter the URL and go 2) Navigate to register page 3) Fill credentials and click register	https://crvp-anxious-fox-py.mybluemix.net/	The page should add a user to backend authentication service	Worked as expected	Pass	Charanraj

	Test scenarios
1	Verify if the user is able to see the landing page
2	Verify if the user is able to navigate to register page
3	Verify if the user is able to click all the buttons in the application
4	Verify if the user is able to register to backend authentication service

Sprint-3

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
Login_Page_01	UI	Login Page	The login page must load properly and be responsive	1) Visit the URL and go 2) Click get started button in the landing page 3) Navigate to the login tab	https://crvp-anxious-fox-py.mybluemix.net/	The user should be able to visit the login table in the registration page	Worked as expected	Pass	Cibikumar
Login_Page_02	Functional	Login Page	Verify the user to login and continue to the application dashboard page when the user logs in first time	1) Visit the URL and go 2) Register by providing credentials 3) Navigate to login page 4) Fill credentials and login	https://crvp-anxious-fox-py.mybluemix.net/	The user should receive a mail asking to confirm their registration and then should be able to login to the application dashboard	Worked as expected	Pass	Akash
Login_Page_03	UI	Dasboard Page	The dashobard page must load and provide options for making a new prediction, logging out and display cards of previous predictions	1) Visit the URL and go 2) Login using credentials 3) View your previous predictions	https://crvp-anxious-fox-py.mybluemix.net/	The user should be navigated to the dashboard page and must be able to see buttons for viewing previous predictions, logging out and making new predictions	Worked as expected	Pass	Danushraj

	Test Scenarios
1	Verify if the user is able to view the login page
2	Verify if th user recieves email for verfication during first time login
3	Verify if ther user is navigated to the dashboard page
4	Verify if the user sees buttons for making new predictions, logging out and cards for previous predictions

Sprint-4

Test case ID	Feature Type	Component	Test Scenario	Steps To Execute	Test Data	Expected Result	Actual Result	Status	Executed By
Prediction_Form_01	UI	Prediction Form	The form should accept different inputs from the user	1) Visit the URL and go 2) Login using credentials 3) Make new predictions by clicking the button	https://crvp-anxious-fox-py.mybluemix.net/	The form must be accepting inputs from the user	Worked as expected	Pass	Charanraj
Prediction_Form_02	Functionality	Prediction Form	From the form inputs the app should retreive, predict the values and display it to the user	1) Visit the URL and go 2) Login using credentials 3) Make new predictions by clicking the button 4) Input data regarding the car and click predict	https://crvp-anxious-fox-py.mybluemix.net/	After submitting the app should display the predicted result	Worked as expected	Pass	Cibikumar
Prediction_Form_03	Functionality	Dashboard Page	The previous predictions must be displayed to the user	1) Visit the URL and go 2) Login using credentials 3) Make new predictions by clicking the button 4) Input data regarding the car and click predict 5) Click view all predictions	https://crvp-anxious-fox-py.mybluemix.net/	The user must be able to see their previous predictions	Worked as expected	Pass	Akash

	Test Scenarios
1	Verify if the user is able to input data to the prediction form
2	Verify if the user is able to view the output of the prediction
3	Verify if the user is able to view their previous predictions

8.2 USER ACCEPTANCE TESTING

2. Defect Analysis

This report shows the number of resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	6	3	2	2	13
Duplicate	1	0	3	0	4
External	2	2	0	1	5
Fixed	7	3	4	5	19
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	5	2	1	8
Totals	16	13	13	10	52

3. Test Case Analysis

This report shows the number of test cases that have passed, failed, and untested

Section	Total Cases	Not Tested	Fail	Pass
Print Engine	5	0	0	5
Client Application	9	0	0	9
Security	3	0	0	3
Outsource Shipping	1	0	0	1
Exception Reporting	2	0	0	2
Final Report Output	4	0	0	4
Version Control	2	0	0	2

9. RESULTS

9.1 PERFORMANCE METRICS

MODEL USED = RANDOM FOREST REGRESSION MODEL

MAE (MEAN ABSOLUTE ERROR) – 1655.53

```
from sklearn.metrics import mean_absolute_error

print(mean_absolute_error(Y_test, y_pred))

1655.534681561534
```

MSE (MEAN SQUARED ERROR) –11832644.33

```
from sklearn.metrics import mean_squared_error

print(mean_squared_error(Y_test, y_pred))

11832644.335139675
```

RMSE (ROOT MEAN SQUARED ERROR) –3439.86

```
from sklearn.metrics import mean_squared_error

root_mean_squared_error = mean_squared_error(Y_test, y_pred, squared=False)
print(root_mean_squared_error)

3439.861092419238
```

R2 SCORE – 0.82

```
from sklearn.metrics import r2_score

print(r2_score(Y_test, y_pred))

0.8191322832483275
```

10. ADVANTAGES & DISADVANTAGES

ADVANTAGES:

- Common People will get the most use from this. Both people including those who wants to sell their cars and also buy used cars can know the best price.

- Car Dealers are the next set of people who can benefit from it. They can use this service and will get to know proper pricing which will help them to set a competitive price for a car.

- All the online services used for selling and buying cars can use this service to give a fair price for a car.

DISADVANTAGES:

- The accuracy & speed of the performance limits the model. Although this can be improved overtime.

- Predicted price accuracy depends on the accuracy of dataset used.

- End user still have to verify with other sources and may use the predicted prices as only an estimation.

11. CONCLUSION

Car Resale Value Prediction is a useful project that has many advantages, especially to those who are looking to buy used cars or sell old cars for justified price. This can also help car dealers and also online services to better run their business with competitive prices.

Also, as an improvement, we can feed better and bigger dataset to improve its accuracy, test the model with different combinations of ML algorithms and find the one that best suits it and increase the speed of prediction with efficient solutions.

Since the model can be hosted online, it can be used by anyone, anywhere and at any time. This level of ease of access makes it even more useful. In conclusion, it is a useful project which will continue to evolve with future ML algorithms.

12. FUTURE SCOPE

The good thing about any ML models is, it can always be enhanced in a better way. As with CRVP model, we can use the next great ML algorithms for greater improvement in performance, use bigger and better datasets and make the model's prediction time lesser.

Instead of stopping with cars, we can also expand it to all vehicle types. The process remains the same with the exception of dataset. This will make it even more useful and won't be limited to just cars.

Another part that can be improved is its ease of access. Currently we can host it in separate website. Instead of it being separate, we can integrate it with existing online services where buying and selling old cars are provided. This saves the end users time immensely.

13. APPENDIX

GITHUB LINK:

https://github.com/IBM-EPBL/IBM-Project-12077-1659369468

PROJECT DEMO LINK:

https://drive.google.com/file/d/14MJhLObGHL5Zrp6Gu-JZ421qjdAGLy36/view?usp=sharing

APPLICATION HOSTED URL:

https://crvp-anxious-fox-py.mybluemix.net/