Car Resale Value Prediction

A PROJECT REPORT

Submitted by

P UDAYAKUMAR(210919104047)

S MATHI SHANKAR(210919104024)

D MUTHUKUMARAN(210919104026)

K DIVAKAR(210919104010)

for the course

HX8001 – Professional Readiness for Innovation,Employability and Enterpreneurship

in

Computer Science

LOYOLA INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

November-2022

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report "**CAR RESALE VALUE PREDICTION"** is the bonafide work of "**UDAYAKUMAR P,MATHI SHANKAR S, MUTHUKUMARAN D,DIVAKAR K"** who carried out the project work under my supervision.

SIGNATURE
Dr. G. BHUVANESWARI,
HEAD OF THE DEPARTMENT,
PROFESSOR,
Computer Science and Engineering,
Loyola Institute of Technology,
Palanchur,
Chennai- 600 123.

SIGNATURE
Mr. S. NOOR MOHAMMED,
MENTOR,
ASSISTANT PROFESSOR,
Computer Science and Engineering,
Loyola Institute of Technology,
Palanchur,
Chennai- 600 123.

ABSTRACT

This paper aims to build a model to predict used cars' reasonable prices based on multiple aspects, including vehicle mileage, year of manufacturing, fuel consumption, transmission, road tax, fuel type, and engine size. This model can benefit sellers, buyers, and car manufacturers in the used cars market. Upon completion, it can output a relatively accurate price prediction based on the information that user input. The model building process involves machine learning and data science. The dataset used was scraped from listings of used cars. Various regression methods, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression, were applied in the research to achieve the highest accuracy. Before the actual start of model-building, this project visualized the data to understand the dataset better. The dataset was divided and modified to fit the regression, thus ensure the performance of the regression. To evaluate the performance of each regression, Rsquare was calculated. Among all regressions in this project, random forest achieved the highest R-square of 0.90416. Compared to previous research, the resulting model includes more aspects of used cars while also having a higher prediction accuracy.

TABLE OF CONTENTS

1.INTRODUCTION

1.1 PROJECT OVERVIEW:

- The main idea of making a car resale value prediction system is to get hands-on practice for python using Data Science.
- Car resale value prediction is the system to predict 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.
- The system is defined in the python language that predicts the amount of resale value based on the given information.
- The system works on the trained dataset of the machine learning program that evaluates the precise value of the car.
- User can enter details only of fields like purchase price of car, kilometers driven, fuel of car, year of purchase.

1.2 PURPOSE

- This resale value prediction system is made for general purpose to just predict the amount that can be roughly acquired by the user.
- We try to predict the amount of resale by best 70% accuracy so the user can get estimated value before he resales the car and doesn't make a deal in loss.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

The problem is defined as the optimised way to estimate insurance cost based on the manufacturer with some additional costs incurred by the Government in the form of taxes. As the existing methods for estimating the cost takes a lot of time and energy and due to the increased price of new cars and the inability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase. The prices of new cars in the industry is fixed by the So, customers buying a new car can be assured of the money they invest to be worthy. There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features. Even though there are websites that offer this service, their prediction method may not be the best. Besides, different models and systems may contribute to predicting power for a used car's actual market value. It is important to know their actual market value while both buying and selling.

2.2 REFERENCES

1. Sameerchand Pudaruth, "Predicting the Price of Used Cars using Machine Learning Techniques";(IJICT 2014)

2. Enis gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric, "Car Price Prediction Using Machine Learning"; (TEM Journal 2019)

3. Shonda Kuiper (2008) Introduction to Multiple Regression: How Much Is Your Car Worth?, Journal of Statistics Education, 16:3.

4. Geurts P. (2009) Bias vs Variance Decomposition for Regression and Classification. In: Maimon O., Rokach L. (eds) Data Mining and Knowledge Discovery Handbook. Springer, Boston, MA

5. Ning sun, Hongxi Bai, Yuxia Geng, Huizhu Shi, "Price Evaluation Model In Second Hand Car System Based On BP Neural Network Theory"; (Hohai University Changzhou, China)

6. Nitis Monburinon, Prajak Chertchom, Thongchai Kaewkiriya, Suwat Rungpheung, Sabir Buya, Pitchayakit Boonpou, "Prediction of Prices for Used Car by using Regression

Models”

7. Doan Van Thai, Luong Ngoc Son, Pham Vu Tien, Nguyen Nhat Anh, Nguyen Thi Ngoc Anh, “Prediction car prices using qualify qualitative data and knowledge-based system”

8. Robert T. (1996) Regression Shrinkage and Selection Via the Lasso. In: Journal of the Royal Statistical Society: Series B (Methodological) Volume 58.

2.3 PROBLEM STATEMENT DEFNITION

Problem Statement (PS)	**I am (Customer)**	**I’m trying to**	**But**	**Because**	**Which makes me feel**
PS-1	I’m owner of Car Travels	I’m trying to buy a Latest Resale Models of Cars for my Travel agency.	I can’t find the one that I expect in local.	There is only few Resale Cars are Available But I want a Bunch of Cars.	Like I should just by a new car instead of resale cars.
PS-2	I’m a Car Seller	I’m trying to sell a Car with Best Price for my Purpose.	I’m not able to find the value/price of my old car.	My old car doesn’t have a AC & Auto Pilot Mode So I want to sell it and buy a one.	Like I don’t want to sell my old car.
PS-2	I’m a Son	I’m trying to buy a Resale Car for My Father who is working in a Factory that is 10km away from my home.	I don’t have any idea that I can buy a car with good condition within my budget.	It's my first time buying Resale car and I don’t want to embarrass myself Infront of my family.	Like I’m overdoing and overthinking it, just because of a Car for My Father and also for My Family.

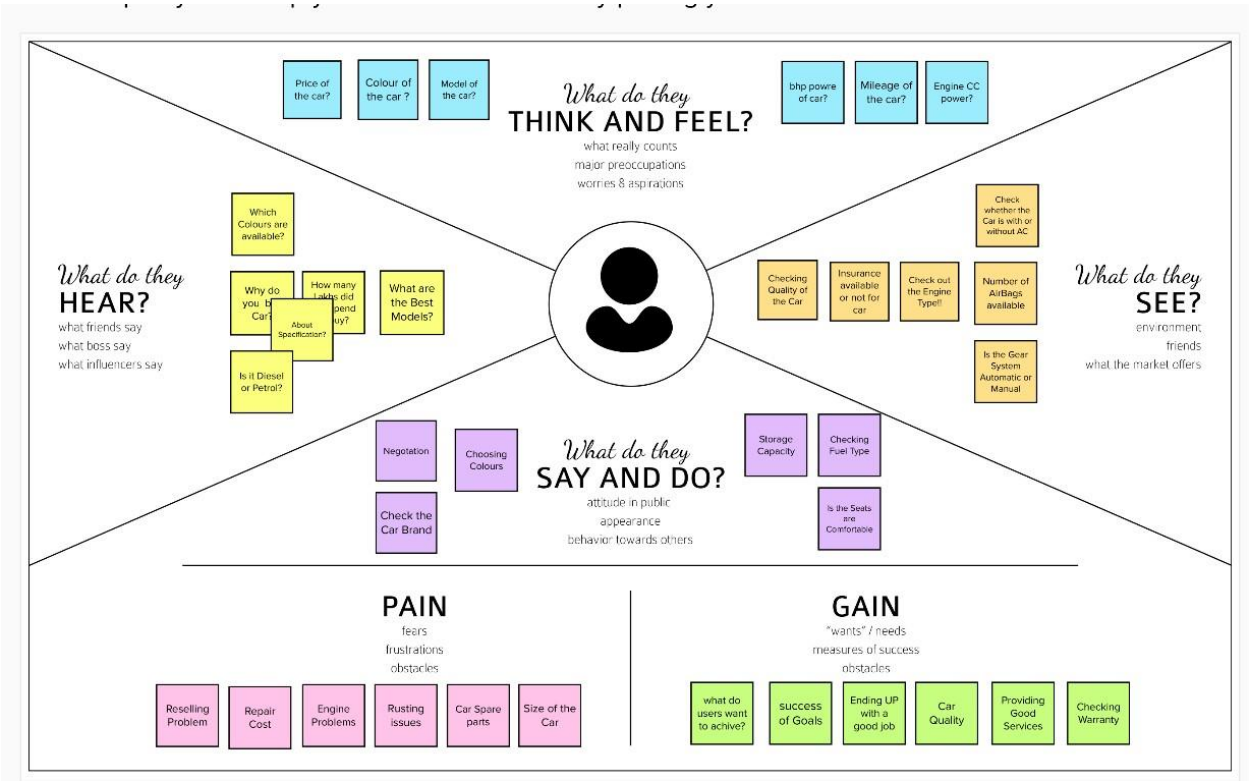
3.IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

Empathy Map :

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviours and attitudes. It is a useful tool to helps teams better understand their users.

Creating an effective solution requires understanding the true problem and the person who is experiencing it. The exercise of creating the map helps participants consider things from the user's perspective along with his or her goals and challenges.

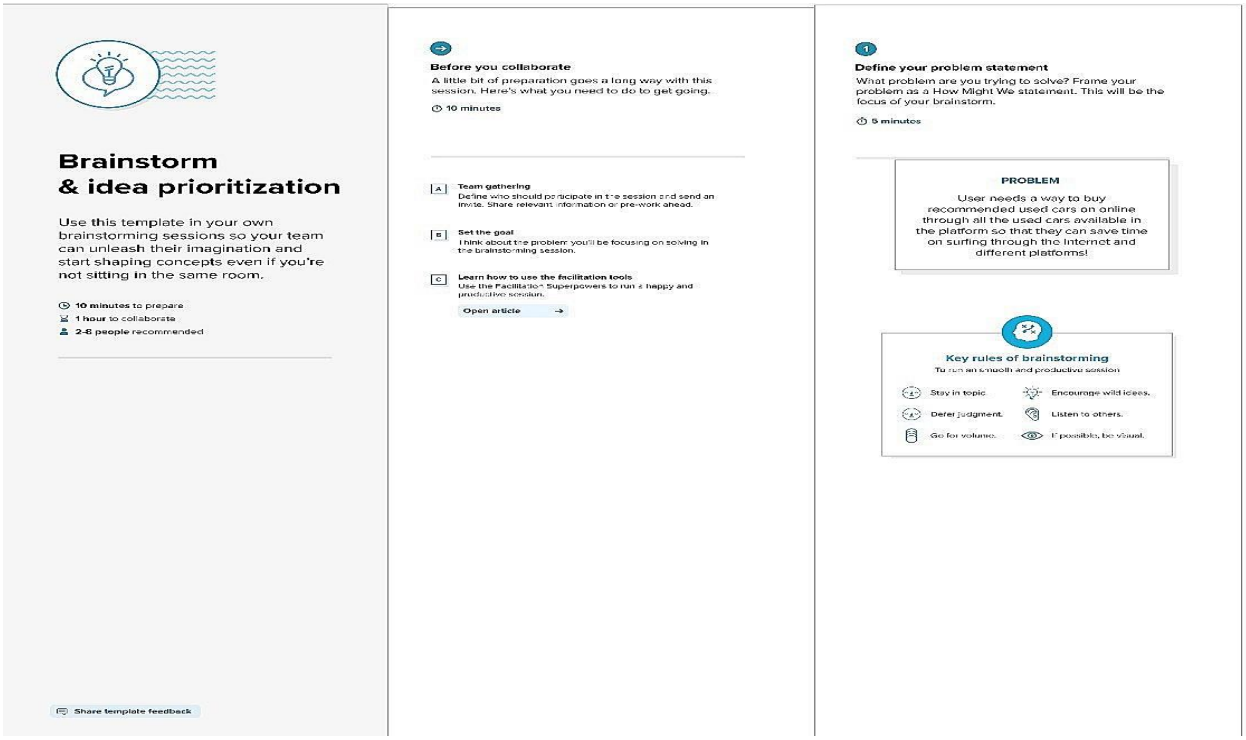


3.2 IDEATION AND BRAINSTORMING

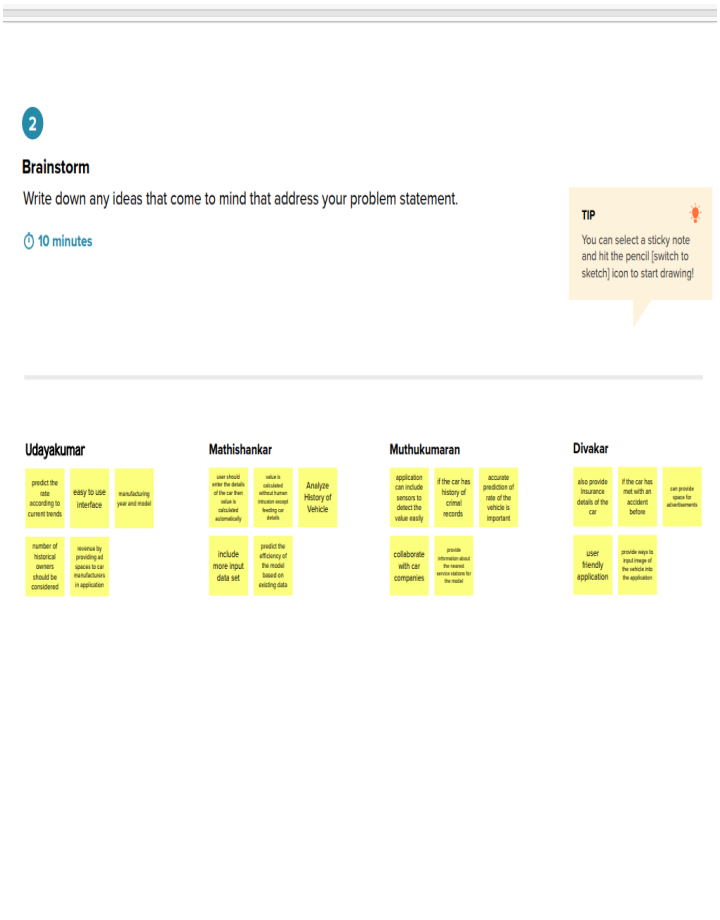
Brainstorm:

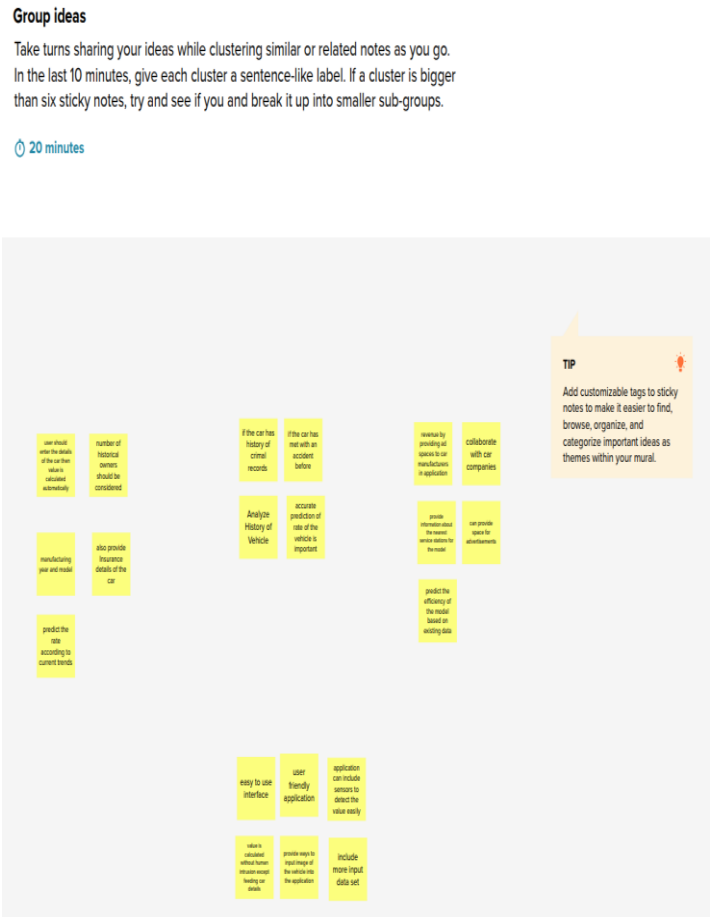
Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, 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 Statemen

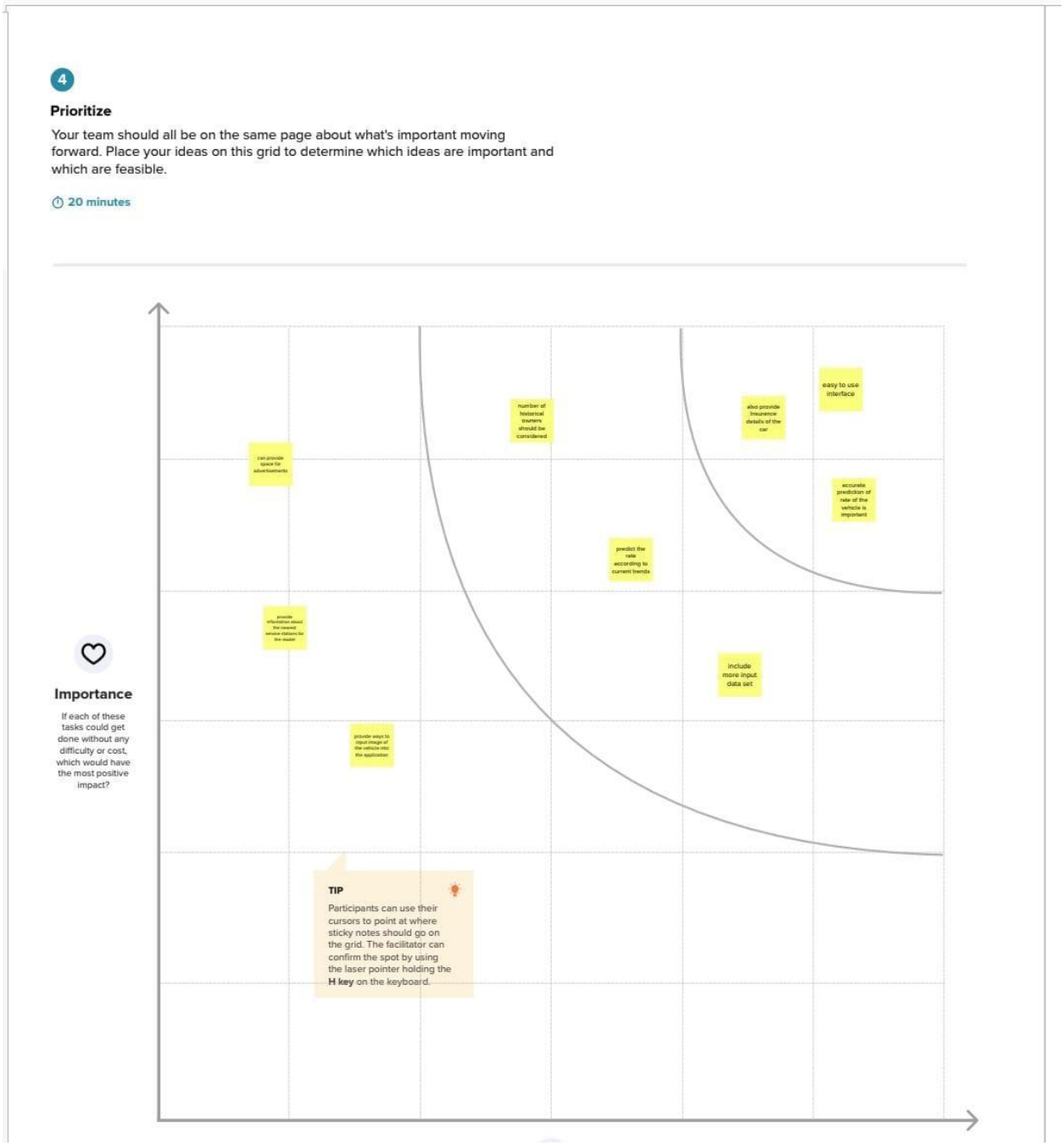


Step-2: Brainstorm, Idea Listing and Grouping





Step-3: Idea Prioritization

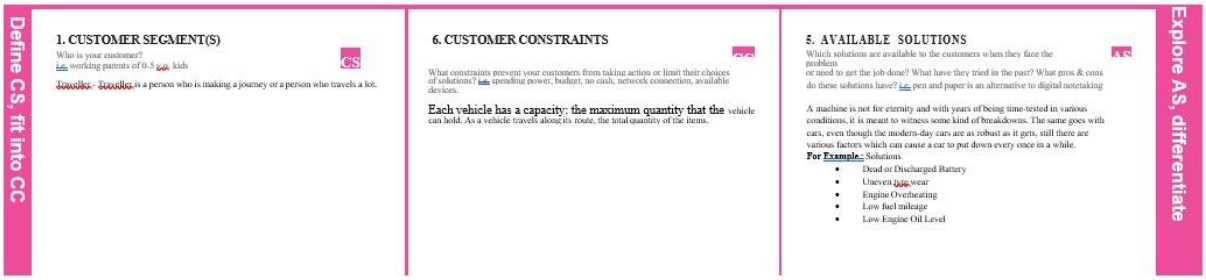


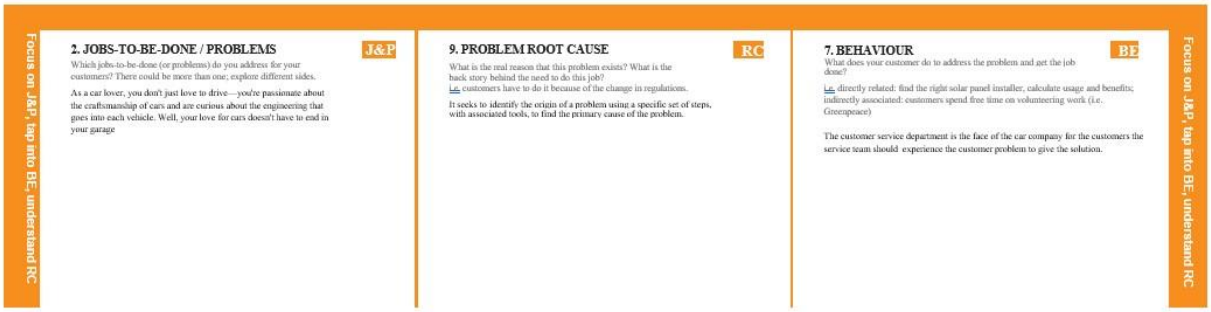
3.3 PROPOSED SOLUTION

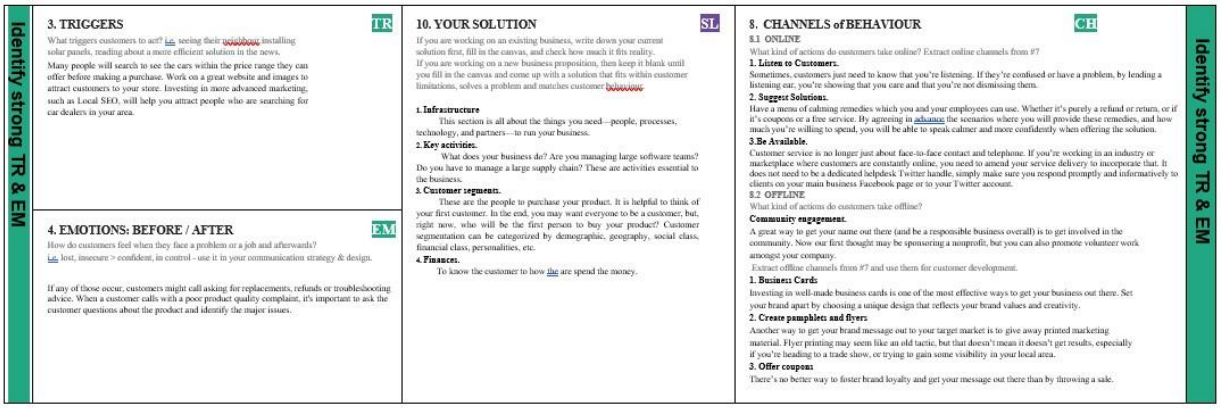
S. No	Parameter	Description
1.	Problem Statement (Problem to be solved)	User needs a way to buy recommended used cars on online through all the used cars available in the platform so that they can save time on surfing through the Internet and different platforms!
2.	Idea / Solution description	To develop a efficient and effective model which predicts the price of a used car according to user's inputs. To develop a User Interface(UI) which is user-friendly and takes input from the user and predicts the price.
3.	Novelty / Uniqueness	Accuracy in Price Prediction.
4.	Social Impact / Customer Satisfaction	A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. The f prediction model was integrated into J application. Furthermore, the model w evaluated using test data and the accu of 87.38% was obtained.

5.	Business Model (Revenue Model)	With the development of the used car market, the demand for a more accurate and scientific price prediction model of used cars becomes urgent. With the development of the used car market, the demand for a more accurate and scientific price prediction model of used cars becomes urgent. It uses multiple linear regression decision tree and random forest to build up the automobile price forecasting model. We use means to cluster cars and find out that some factors like power, kilometres , gearbox have an influence on the price.
6.	Scalability of the Solution	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.

3.4 PROBLEM SOLUTION FIT







4.REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Process of Value Prediction	Accuracy in Price Prediction

4.2 NON-FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Indicates how effectively and easy users can learn and use a system.
NFR-2	**Security**	Assures all data inside the system or its part will be protected against malware attacks or unauthorized access.
NFR-3	**Reliability**	Specifies the probability of the software performing without failure for a specific number of uses or amount of time.
NFR-4	**Performance**	Deals with the measure of the system's response time under different load conditions.
NFR-5	**Availability**	Describes how likely the system is accessible for a user at a given point in time.

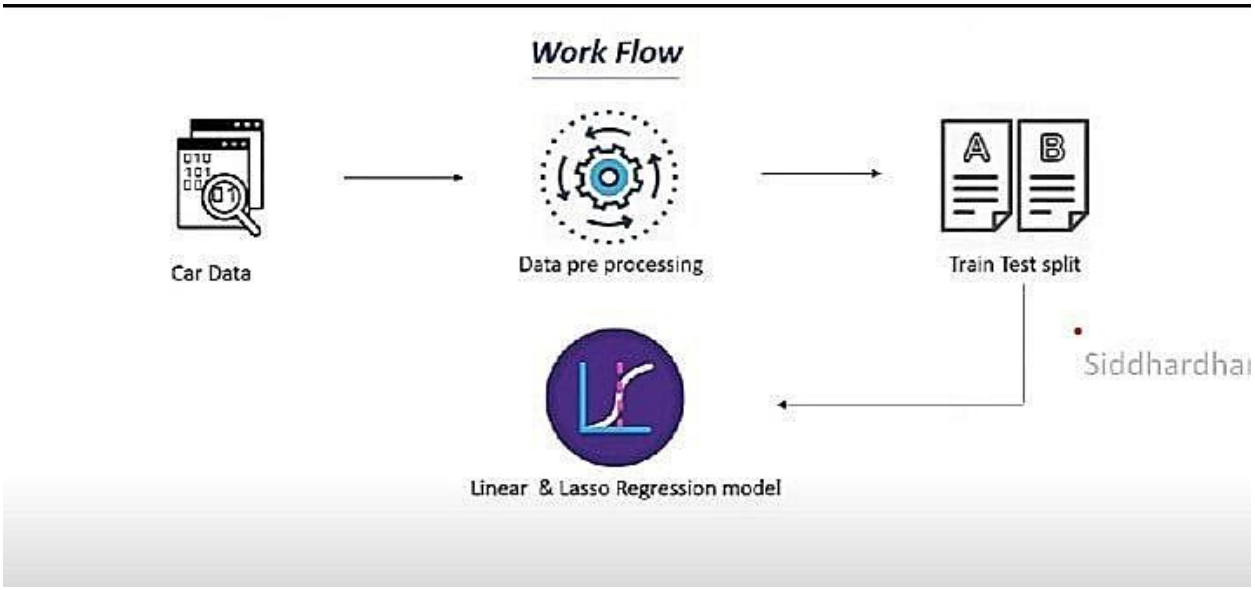
NFR-6	**Scalability**	Accesses the highest workload under which the system will still meet the performance requirements.

5.PROJECT DESIGN

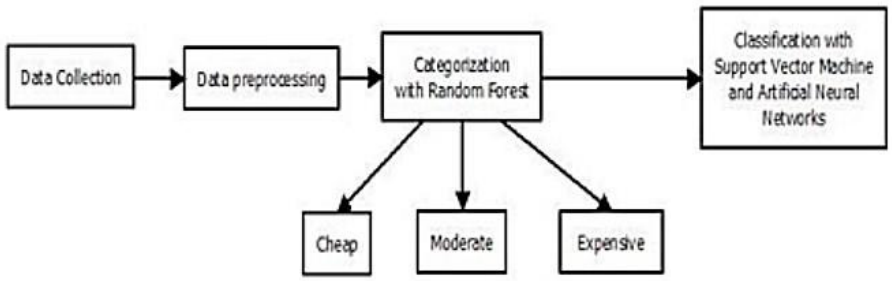
5.1 DATA FLOW DIAGRAMS

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

Simplified:



Car resale value model:



Acceptance criteria **Priority** **Release**

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

i.SOLUTION ARCHITECTURE

Functional Requirements:

Following are the functional requirements of the proposed solution.

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	Process of Value Prediction	Accuracy in Price Prediction

Non-functional Requirements:

Following are the non-functional requirements of the proposed solution.

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Indicates how effectively and easy users can learn and use a system.
NFR-2	**Security**	Assures all data inside the system or its part will be protected against malware attacks or unauthorized access.
NFR-3	**Reliability**	Specifies the probability of the software performing without failure for a specific number of uses or amount of time.
NFR-4	**Performance**	Deals with the measure of the system's response time under different load conditions.
NFR-5	**Availability**	Describes how likely the system is accessible for a user at a given point in time.
NFR-6	**Scalability**	Accesses the highest workload under which the system will still meet the performance requirements.

ii.TECHNICAL ARCHITECTURE

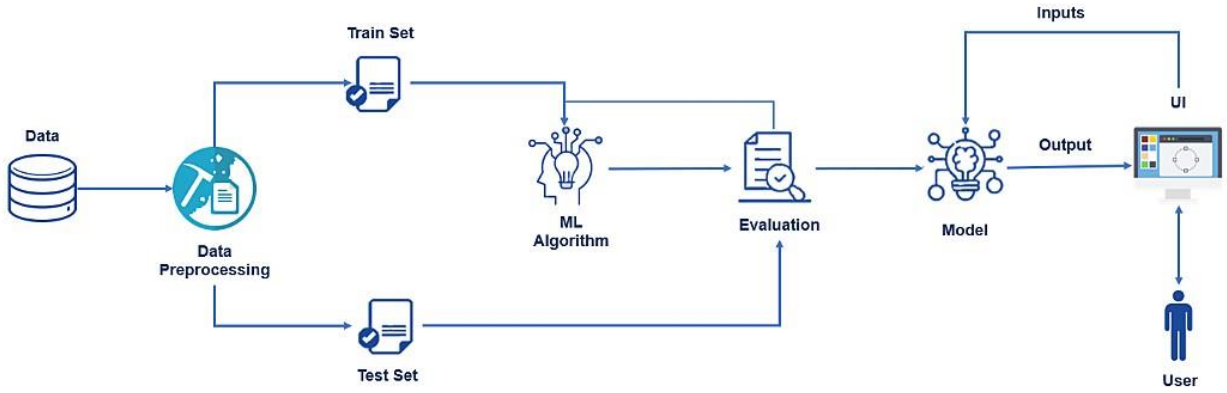


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web Ui only	HTML, CSS,Python, Flask
2.	Application Logic-1	Load the data set and find the test data and train data	Python
3.	Application Logic-2	Logic for a process in the application	Pandas,numpy,sklearn
4.	Application Logic-3	Logic for a process in the application	flask
5.	Database	Data Type, Configurations etc.	Dataset
6.	Cloud Database	Database Service on Cloud	IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage or Other Storage Service or Local Filesystem
8.	External API-1	Purpose of External API used in the application	IBM cloud API, etc.
9.	Machine Learning Model	Purpose of Machine Learning Model	Regression Model.

10.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Cloud Foundry, Kubernetes, etc.

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Technology of Opensource framework
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	e.g. SHA-256, Encryptions, IAM Controls, OWASP etc.
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, services)	Machine Learning
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Machine Learning
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Pyhton Flask,html,css Micro

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
(Mobile user)			entering my email, password, and confirming my password.	dashboard		
		USN-2	As a user, I will receiveconfirmation email onceI 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 Facebook	I can register &access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard	USN-6	Entering the car details in the application		High	Sprint-1

Customer (Web user)	Process	USN-1	As a user, I can enterthe car whichI want to predict theprice		Medium	Sprint-2
Customer Care Executive	Maintenance	USN-2	As a executive, I can rectify Customer's Problems as well as Comments	I can interact throughcomments	High	Sprint-2
Administrator	Developing		As a administrator, I can checkthe car prediction values are up to date	I cangather the details of each car	High	Sprint-2

6. PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

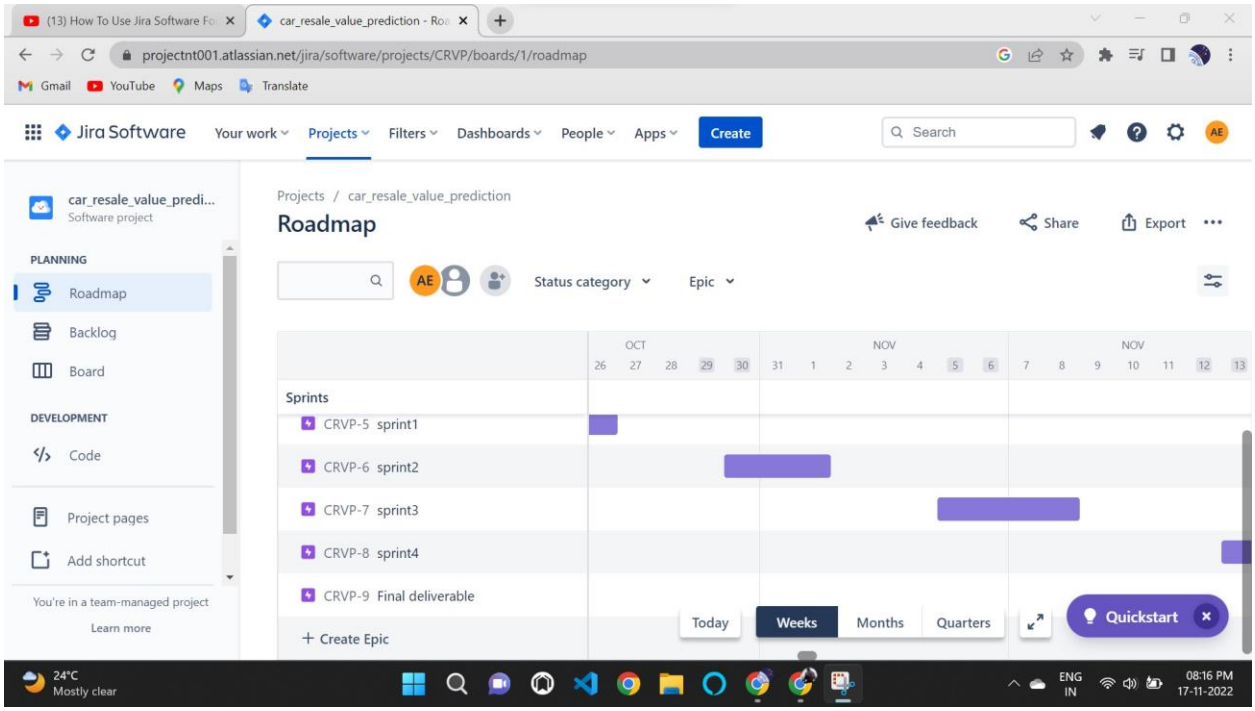
Sprint	Functional Requirement (Epic)	UserStory Number	UserStory / Task	Story Points	Priority	Team Members
Sprint-1	Dataset reading and Preprocessing	USN-1	Cleaning the dataset and splitting to dependent andindependent variables	2	High	Ajith.E A.Navinvarma
Sprint-2	Building the model	USN-2	Choosing the appropriate model for building and saving the model as pickle file	1	High	Ajith.E Rahul.M Jothi murugan.A

Sprint-3	Applicati on building	USN-3	Using flaskdeploying the ML model	2	Medi um	Ajith.E Navinvarma .A
Sprint-4	Train themodel in IBM	USN-4	Finally trainthe model on IBM cloudand deploy the application	2	Medi um	Ajith.E Jothi murugan.A Rahul.M Navinvarma .A

6.2 SPRINT DELIVERY SCHEDULE

Sprint	**Total Story Poin ts**	**Durati on**	**Sprint Start Date**	**Sprint End Date (Planne d)**	**Story Points Complet ed (as on Planned End Date)**	**Sprint Release Date (Actual)**
Sprint-1	15	5 Days	24 Oct2022	29 Oct2022	15	29Oct 2022
Sprint-2	15	5 Days	31 Oct2022	05 Nov2022	15	05Nov 2022
Sprint-3	15	5 Days	07 Nov 2022	12 Nov2022	15	12Nov 2022
Sprint-4	15	5 Days	14 Nov 2022	19 Nov2022	15	19Nov 2022

6.3 REPORTS FROM JIRA



7.CODING AND SOLUTIONING

7.1 FEATURE 1

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

#Load the dataset df = pd.read_csv(r"E:\car_resale\Data\autos.csv", header=0, sep=',',
encoding='Latin1', )

#print all the different sellers
print(df.seller.value_counts())

#remove the seller type haveing only 3 car
df[df.seller != 'gewerblich']

#now all the sellers are same so we can get rid of this column
df=df.drop(columns=['seller'])  #1 refer the columns & 0 refer the index

#print all different seller
print(df.offerType.value_counts())

#remove the offers type having only 12 listings df[df.offerType
!= 'Gesuch']
```

```
#now all offer are sameso we can get rid this collumn
df=df.drop(columns=['offerType']) # 1 refer the columns & 0 refer the index

'''car having power les then 50ps and above 900ps seems a little suspicious, let's remove them and see
what we have got now''' print(df.shape)
df = df[(df.powerPS > 50) & (df.powerPS < 900)] print(df.shape)
#around 50000 cars ahave been removed which could have introunduced error toour data


#Simlarly, filtering our the cars having registeration years not in the mentioned range
#print(df.shape)
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)] print(df.shape)
#not much of a difference but still, 10000 rows have been reduced. it's better to
#get rid of faulty data instead of keeping them just to increase the size

'''removing irrelevent columns which are either the same for all the cars in the dataset, or can
introduce bias, so removing them too..'''

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

'''dropping the duplicates from the dataframe and stroing it in a new here all row having
same value in all the mentioned columns will be deleted and by defult, only first
occurance of any such row is kept''' new_df = df.copy()
new_df = new_df.drop_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox',
'powerPS',
                        'model', 'kilometer', 'monthOfRegistration', 'fuelType',
'notRepairedDamage'])
```

```
#As the dataset contained same german words for many features, changing them to engilsh
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','canvertible','combination','others'), inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'), ('Yes','No'), inplace=True)

#### Removing the outliers new_df = new_df[(new_df.price >= 100) &
(new_df.price <= 150000)]

''' Filling NaN values for columns whose data might not be there with the information provider,
which might lead to some variance but our model but we will still be able to give some estimate
to the user'''

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)

#can save the csv for future purpose.
new_df.to_csv("autos_preprocessed.csv")

#Columns which contain categorical values, which we'll need to convert via label encoding
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

'''looping over the labels to the label encoding for all at once and
saveing the LABEL ENCODING FILES''' 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)

#Final data to be put in a new dataframe called "LABELED", labeled
= new_df[
    [
        'price',
        'yearOfRegistration',
        'powerPS',
        'kilometer',
        'monthOfRegistration'
    ] + [x+"_labels" for x in labels]
]

print(labeled.columns)

#Storing price in Y and reset of the data in X
Y = labeled.iloc[:,0].values
X  = labeled.iloc[:,1:].values

#need to reshape the Y values
Y  = Y.reshape(-1,1)

#traing data and test data 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 and fitting
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 of test
y_pred = regressor.predict(X_test)

#printing the Accuraccy for test set
print(r2_score(Y_test, y_pred))

#saving the model for future use.
filename = 'resale_model.pkl'
pickle.dump(regressor, open(filename, 'wb'))
```

The System is defined in the python language that predicts the amount of resale value based on the given information.The system works on the trained dataset of the machine learning program that evaluates the precise value of the car.User can enter details only of fields like purchase price of car,kilometers driven,fuel of car,year of purchase.

7.1 FEATURE 2

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

app = Flask(__name__)  filename
= 'resale_model.pkl'
model_rand = pickle.load(open(filename, 'rb'))

@app.route('/')
def index():
return
render_template('
resaleintro.html')


@app.route('/predict')
def predict():
   return render_template('resalepredict.html')

@app.route('/y_predict', methods=['GET', 'POST']) def
y_predict():
   regyear = int(request.form['regyear'])
   powerps = float(request.form['powerps'])
   kms = float(request.form['kms'])
   regmonth = int(request.form.get('regmonth'))
```

```
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}
print(new_row)
new_df = pd.DataFrame (columns =['vehicleType', 'yearOfRegistration', 'gearbox',
                        'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType',
                        'brand', 'notRepairedDamage'])
new_df = new_df.append(new_row, ignore_index = True)
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
mapper = {}
for i in labels:
    mapper[i]= LabelEncoder()
    mapper[i].classes_ = np.load(str('classes'+i+'.npy'), allow_pickle=True)
    tr = mapper[i].fit_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
```

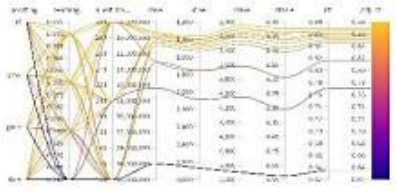
```
    print(X)
    y_prediction = model_rand.predict(X)
    print(y_prediction)
    return render_template('resalepredict.html',ypred = 'The resale value predicted is
{:.2f}$'.format(y_prediction[0]))

if __name__== '__main__':
    app.run(host='localhost', debug=True, threaded=False)
```

Upon from submission,the data is sent to the ML model via Flask API and the model responds with a predicted resale value of the car based on user input.The prediction is displayed on the web page using a render template .Thus,with minimal information and without human intervention or manual examination,a user can predict the resale value of his car.

8.TESTING

8.1 TEST CASES

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Regression Model:** LGBM Regressor MAE: 1327.56 MSE: 9492244.25 RMSE: 3080.93 RMSLE: 8.05 R2 Score: 0.8664 Adjusted R2 Score: 0.8666	
2.	Tune the Model	**Hyperparameter Tuning** 1) Learning Rate: [0.01, 0.03, 0.05, 0.07] 2) Boosting Type: ['gbdt','dart','goss','rf'] 3) Number of Estimators: [100,200,300] **Validation Method**: Grid Search Cross Validation **Best Parameters:** Learning Rate – 0.07 Boosting Type – 'gbdt' Number of Estimators - 300	 

8.2 USER ACCEPTANCE TESTING :

				Date	17-Nov-22								
				Team ID	PNT2022TMID40127								
				Project Name	Project - Car Resale Value Prediction								
				Maximum Marks	4 marks								
Test case ID	**Feature Type**	**Component**	**Test Scenario**	**Pre-Requisite**	**Steps To Execute**	**Test Data**	**Expected Result**	**Actual Result**	**Status**	**Comments**	**TC for Automation(Y/N)**	**BUG ID**	**Executed By**
HomePage_TC_001	UI	Home Page	Verify all the UI elements in Home page rendered properly		1. Enter URL and click go 2. Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Ajith
HomePage_TC_002	Functional	Home Page	Verifiy the Data Entry page can be reachable.		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button.	-	User should navigate to Data Entry Page	Working as expected	Pass		N		Navinvarma
DataEntryPage_TC_001	UI	Data Entry Page	Verify all the UI elements in Data Entry page rendered properly		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button in the home page 4. Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Rahul
DataEntryPage_TC_002	Functional	Data Entry Page	Verify user is able to enter all values		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered	2012 12 12 Manual Yes Golf Volkswagen Petrol Coupe	User should be able to enter all values in data entry page	Working as expected	Pass		N		Jothi murugan
DataEntryPage_TC_003	Functional	Data Entry Page	Verifiy the Output Display page can be reachable.		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered 6. Press the submit Button	-	User should navigate to Output Display Page	Working as expected	Pass		N		Rahul
OutputDisplayPage_TC_001	UI	Output Display Page	Verify all the UI elements in Output Display page rendered properly		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered 6. Press the submit Button 7. Verify all the UI elements displayed or not	-	All the UI elements rendered properly	Working as expected	Pass		N		Navinvarma
OutputDisplayPage_TC_002	Functional	Output Display Page	Verify user is able to get predicted result		1. Enter URL and click go 2. Verify all the UI elements displayed or not. 3. Press the Check Price button in the home page 4. Verify all the UI elements displayed or not 5. Verify if all values can be entered 6. Press the submit Button 7. Verify all the UI elements displayed or not 8. Verify if the predicted value is displayed or not	-	Predited Car Resale Value is displayed on the page	Working as expected	Pass		N		Ajith

Test Scenarios :

Verify user is able to see home page?
Verify user is able to navigate to data entry page?
Verify user is able to see data entry page?
Verify user is able to enter values in the fields?
Verify user is able to navigate to output display page?
Verify user is able to view the output display page?

9.RESULTS

9.1 PERFORMANCE METRICS

Use the below template to create product backlog and sprint schedule

Sprint	Functional Requirement (Epic)	User Story Number	UserStory / Task	StoryPoints	Priority	Team Members
Sprint-1	Dataset reading and Preprocessing	USN-1	Cleaning thedataset and splitting to dependent andindependent variables	2	High	Ajith.E A.Navinvarma
Sprint-2	Building the model	USN-2	Choosing the appropriate model for building andsaving the modelas pickle file	1	High	Ajith.E Rahul.M Jothi murugan.A
Sprint-3	Application building	USN-3	Usingflask deploying the ML model	2	Medium	Ajith.E Navinvarma.A
Sprint-4	Train the model in IBM	USN-4	Finally train the model on IBM cloud and deploy the application	2	Medium	Ajith.EJothi murugan.A Rahul.M Navinvarma.A

Project Tracker, Velocity& Burndown

Sprint	Total Story Points	Duration	SprintStart Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	15	5 Days	24 Oct2022	29 Oct 2022	15	29 Oct2022
Sprint-2	15	5 Days	31 Oct2022	05 Nov 2022	15	05 Nov 2022

Sprint-3	15	5 Days	07 Nov 2022	12 Nov 2022	15	12 Nov 2022
Sprint-4	15	5 Days	14 Nov 2022	19 Nov 2022	15	19 Nov 2022

10.ADVANTAGES AND DISDVANTAGES

i.ADVANTAGES

- Value for money. Pre-owned cars come with a lower price tag and offer a much better value for the amount paid.
- Slow rate of depreciation.
- Lower insurance and registration charges.
- Higher inflation.
- Lower loan amount to be borrowed.

ii. DISDVANTAGES

- Little to No Warranty
- New models not avaliable
- Little to No Financing
- No accurate prediction

11.CONCLUSION

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. Data cleaning is one of the processes that increases prediction performance, yet insufficient for the cases of complex data sets as the one in this research. Applying single machine algorithm on the data set accuracy was less than 50%. Therefore, the ensemble of multiple machine learning algorithms has been proposed and this combination of ML methods gains approximate price prediction.

This is significant improvement compared to single machine learning method approach. However, the drawback of the proposed system is that it consumes much more computational resources than single machine learning algorithm.

12.FUTURE SCOPE

Currently, system can only deal with Swift Dzire cars due to lack of data. Also, data has been collected of only 5 cities of India. This can be extended to multiple car models and cities so as to improve accuracy and usability.

Efficient use of deep learning such as LSTM (Long shortterm memory) or RNN (Recurrent Neural networks) can be implemented once enough data is collected. This can improve accuracy and decrease RMSE drastically.One can also implement CNN to determine physical condition of the car from images like identifying dents, scratches etc. and thus predicting more relevant resale value of a car.

13.APPENDIX CODING

Source code:

Resale value prediction final.py

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

#Load the dataset df = pd.read_csv(r"E:\car_resale\Data\autos.csv", header=0, sep=',',
encoding='Latin1', )

#print all the different sellers
print(df.seller.value_counts())
#remove the seller type haveing only 3 car
df[df.seller != 'gewerblich']
#now all the sellers are same so we can get rid of this column
df=df.drop(columns=['seller'])  #1 refer the columns & 0 refer the index  #print all different
seller print(df.offerType.value_counts())
#remove the offers type having only 12 listings df[df.offerType
!= 'Gesuch']
#now all offer are sameso we can get rid this collumn
df=df.drop(columns=['offerType']) # 1 refer the columns & 0 refer the index
```

```
'''car having power les then 50ps and above 900ps seems a little suspicious, let's
remove them and see what we have got now''' print(df.shape)
df = df[(df.powerPS > 50) & (df.powerPS < 900)] print(df.shape)
#around 50000 cars ahave been removed which could have introunduced error toour data
#Simlarly, filtering our the cars having registeration years not in the mentioned range
#print(df.shape)
df = df[(df.yearOfRegistration >= 1950) & (df.yearOfRegistration < 2017)] print(df.shape)
#not much of a difference but still, 10000 rows have been reduced. it's better to
#get rid of faulty data instead of keeping them just to increase the size

'''removing irrelevent columns which are either the same for all the cars in the dataset, or
can introduce bias, so removing them too..''' df.drop(['name', 'abtest', 'dateCrawled',
'nrOfPictures', 'lastSeen',
      'postalCode', 'dateCreated'], axis='columns', inplace=True)

'''dropping the duplicates from the dataframe and stroing it in a new here all row having
same value in all the mentioned columns will be deleted and by defult, only first
occurance of any such row is kept''' new_df = df.copy() new_df =
new_df.drop_duplicates(['price', 'vehicleType', 'yearOfRegistration', 'gearbox', 'powerPS',
                        'model', 'kilometer', 'monthOfRegistration', 'fuelType',
'notRepairedDamage'])
#As the dataset contained same german words for many features, changing them to engilsh
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','canvertible','combination','others'), inplace=True)
new_df.notRepairedDamage.replace(('ja','nein'), ('Yes','No'), inplace=True)
#### Removing the outliers new_df = new_df[(new_df.price >= 100) &
(new_df.price <= 150000)]
```

```
''' Filling NaN values for columns whose data might not be there with the information provider, which might lead to some variance but our model but we will still be able to give some estimate to the user'''
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)


#can save the csv for future purpose.
new_df.to_csv("autos_preprocessed.csv")
#Columns which contain categorical values, which we'll need to convert via label encoding
labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']

'''looping over the labels to the label encoding for all at once and
saveing the LABEL ENCODING FILES''' 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)

#Final data to be put in a new dataframe called "LABELED", labeled
= new_df[
    [
        'price',
```

```
        'yearOfRegistration',
        'powerPS',
        'kilometer',
        'monthOfRegistration'     ] +
[x+"_labels" for x in labels]]
print(labeled.columns)
#Storing price in Y and reset of the data in X
Y = labeled.iloc[:,0].values X
= labeled.iloc[:,1:].values
#need to reshape the Y values
Y = Y.reshape(-1,1) #traing data and test data 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 and fitting
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 of test
y_pred = regressor.predict(X_test)
#printing the Accuraccy for test set
print(r2_score(Y_test, y_pred))
#saving the model for future use.
filename = 'resale_model.pkl'
pickle.dump(regressor,
open(filename, 'wb'))
```

Resale_flask.py

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

app = Flask(__name__)  filename =
'resale_model.pkl'  model_rand =
pickle.load(open(filename, 'rb'))

@app.route('/')
def index():
   return render_template('resaleintro.html')
@app.route('/predict')
def predict():
   return render_template('resalepredict.html') @app.route('/y_predict', methods=['GET',
'POST']) def y_predict():
   regyear = int(request.form['regyear'])
   powerps = float(request.form['powerps'])
   kms = float(request.form['kms'])
   regmonth = int(request.form.get('regmonth'))
   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}
    print(new_row)
    new_df = pd.DataFrame (columns =['vehicleType', 'yearOfRegistration', 'gearbox',
                            'powerPS', 'model', 'kilometer', 'monthOfRegistration', 'fuelType',
                            'brand', 'notRepairedDamage'])
    new_df = new_df.append(new_row, ignore_index = True)
    labels = ['gearbox', 'notRepairedDamage', 'model', 'brand', 'fuelType', 'vehicleType']
    mapper = {}
    for i in labels:
        mapper[i]= LabelEncoder()
        mapper[i].classes_ = np.load(str('classes'+i+'.npy'), allow_pickle=True)
        tr = mapper[i].fit_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
    print(X)
    y_prediction = model_rand.predict(X)
    print(y_prediction)
    return render_template('resalepredict.html',ypred = 'The resale value predicted is
{:.2f}$'.format(y_prediction[0]))
```

```
if __name__== '__main__':
    app.run(host='localhost', debug=True, threaded=False)
```

resalepredict.html

```
 <html lang="en">
<head>
<link rel="stylesheet" href="../static/style.css" type="text/css">
</head>
 <body>
 <div class="background-image">
  <marquee behavior="" direction="" scrollamount="15"><center><h1>GET THE
ACCURATE RESALE VALUE OF YOUR CAR...!</h1></marquee>
  <center><h2>Please fill the following details of your car:</h2>
<div class="form">
<form action="/y_predict" method="post">
<label>Registration year</label>
 <input  type="number" id="pj" name="regyear" placeholder="Enter the year" >
<br><br>
<label>Registration month</label>
 <input type="number" name="regmonth" id="pj" placeholder="Enter the month">
<br><br>
<label>Power of Car in PS</label>
 <input type="number" name="powerps" id="pj" placeholder="Enter power of car">
<br><br>
<label>Kilometers the car as driven</label>
 <input type="number" name="kms" id="pj" placeholder="Enter no of kms">
<br><br>
```

```
<label>GearBox type</label>
 <input type="radio" id="manual" name="gearbox" value="manual" for="manual">Manual
 <input type="radio" id="automatic" name="gearbox" value="automatic"
for="automatic">Automatic
 <input type="radio" name="gearbox" value="not-declared" for="not-declared">Not declared
<br><br>
<label>Your car is damaged(or)repaired</label>
 <input type="radio" id="Yes" name="dam" value="Yes" for="Yes">Yes
 <input type="radio" id="No" name="dam" value="No" for="No">No
 <input type="radio" id="not-declared" name="dam" value="not-declared"
for="notdeclared">Not declared<br>
 <br><br>
<label>Modeltype</label>
 <input type="text" name="modeltype" id="pj" placeholder="Enter the Model">
<br><br>
<label>Brand of the car</label>
 <input type="text" name="brand" id="pj" placeholder="Enter the Brand">
<br><br>
<label>Fuel type of the car</label>
 <input type="text" name="fuel" id="pj" placeholder="Enter fule type">
<br><br>
<label>Vehicaltype</label>
 <input type="text" name="vehicletype" id="pj" placeholder="Enter Vchicle type">
 <br><br>
<p><input  type="submit" value="PREDICT" style="background-color:#F28C0F;color:white; width:
125px; height:50px;"/></p>

</form>
</div>
```

```
<b><h3>{{ypred}}</h3></b>
<br> <br>
</div>
</body>
</html>
```

GitHub & Project Demo Link

GitHub link- https://github.com/IBM-EPBL/IBM-Project-53496-1661412125

Demo link- https://drive.google.com/file/d/1FfQEj0qN41bD3AohuFRGEGWwacwPoyJA/view?usp=drivesdk