CAR RESALE VALUE PREDICTION

NALAIYA THIRAN PROJECT BASED LEARNING

ON

PROFESSIONAL READINESS FOR INNOVATION, EMPLOYABILITY AND ENTREPRENEURSHIP

Submitted By

TEAM ID: PNT2022TMID31751

HARIPRIYA R	**711719205020**
JEEVA H	**711719205022**
KUBERA PRIJESH D	**711719205028**
SNEHA N S	**711719205052**

TABLE OF CONTENTS

1. INTRODUCTION

Car resale value prediction system is made with the purpose of predicting the correct valuation of used cars that helps users to sell the car remotely with perfect valuation and without human intervention in the process to eliminate biased valuation. For this project, we developed the car resale value using a variety of algorithms and methodologies.prediction algorithms that take into account the car's many qualities. In a nutshell, automobile resale value prediction enables users to forecast the resale worth of a car based on a variety of factors, such as the number of miles traveled and the type of fuel used. The project focuses on getting input like kilometers a vehicle has run, year of manufacture, the owner, number of services the car has been given since its purchase, previous damages and so on.

1.1 PROJECT OVERVIEW

In this project we have used different algorithms with different techniques for developing Car resale value prediction systems considering different features of the car. In a nutshell, car resale value prediction helps the user to predict the resale value of the car depending upon various features like kilometers driven, fuel type, etc.

1.2 PURPOSE

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.

This resale value prediction system is made for general purpose to just predict the amount that can be accurately acquired by the user. We try to predict the amount of resale by best 90% accuracy so the user can get an estimated value before he resales the car and doesn't make a deal in loss.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

1.Car Price Prediction using Machine Learning

This proposed system helps to study car price detection and resale prediction. The results showed that there is a positive correlation between price and kilometers traveled, year of registration and kilometers traveled and a negative correlation between price and year of registration.

2.Used Car Price Prediction using K-Nearest Neighbor Based Model

This paper proposed a supervised machine learning model using KNN algorithm to analyze the price of used cars. We trained our model with data of used cars which is collected from the Kaggle website.

3.Prediction of Prices for Used Car by using Regression Models

This proposed approach aids in the investigation of a pricing evaluation model based on big data analysis.Given time itself could have a significant impact on the pricing of cars, the data was gathered over a shorter period of time than a month

4.Prediction of Used Car Price Based on Supervised Learning Algorithm (2021)

Extra Trees Regressor, Random Forest Regressor was used.The algorithm was optimized by using the hyper parameter function.

5.Car resale value prediction using quantify qualitative data and knowledge-based system

The difficult procedure has been the main method for acquiring knowledge.Explore different data types of car data.

6.Vehicle Price Prediction using SVM Techniques

The study sector has seen a rise in the popularity of car price forecasting, which calls for significant effort and knowledge of the subject matter specialists. This the merit as follows, distinct machine learning algorithms of varying performances.

7.Car resale value prediction using quantify qualitative data and knowledge-based system

In this paper which is titled Doan Van thai(2019) quantify qualitative data and knowledge-based systems. The primary technique for knowledge acquisition has Been the time-consuming process.

8.Used Car price prediction (2021)

The data in the data set are used to train the system, which then creates a model. The inputs are sent to the system, which is then tested to see how it functions.

2.2 REFERENCES

[1].M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in Proc. of USENIX Security Symposium, 2017.

[2].Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani,―Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization‖, 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal,January 2018

[3].Hossein Hadian Jazi, Hugo Gonzalez, Natalia Stakhanova, and Ali A. Ghorbani. "Detecting HTTP-based Application Layer DoS attacks on Web Servers in the presence of sampling." Computer Networks, 2017

[4]. A. Shiravi, H. Shiravi, M. Tavallaee, A.A. Ghorbani, Toward developing a systematic approach to generate benchmark datasets for intrusion detection, Comput. Security 31 (3) (2012) 357–374.

[5].Z. He, T. Zhang, and R. B. Lee, ―Machine Learning Based DDoS Attack Detection from Source Side in Cloud,‖ in Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), pp. 114–120, New York, NY, USA, June 2017

[6].R. Doshi, N. Apthorpe and N. Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," 2018 IEEE Security and Privacy Workshops (SPW), San Francisco, CA, 2018, pp. 29-35.

[7].Jerome H. Friedman, (2002), Stochastic gradient boosting, Computational Statistics & Data Analysis, 38, (4), 367-378

[8].Friedman, Jerome H. Greedy function approximation: A gradient boosting machine. Ann. Statist. 29 (2001), no. 5, 1189--1232

2.3 PROBLEM STATEMENT DEFINITION

Customer Problem Statement

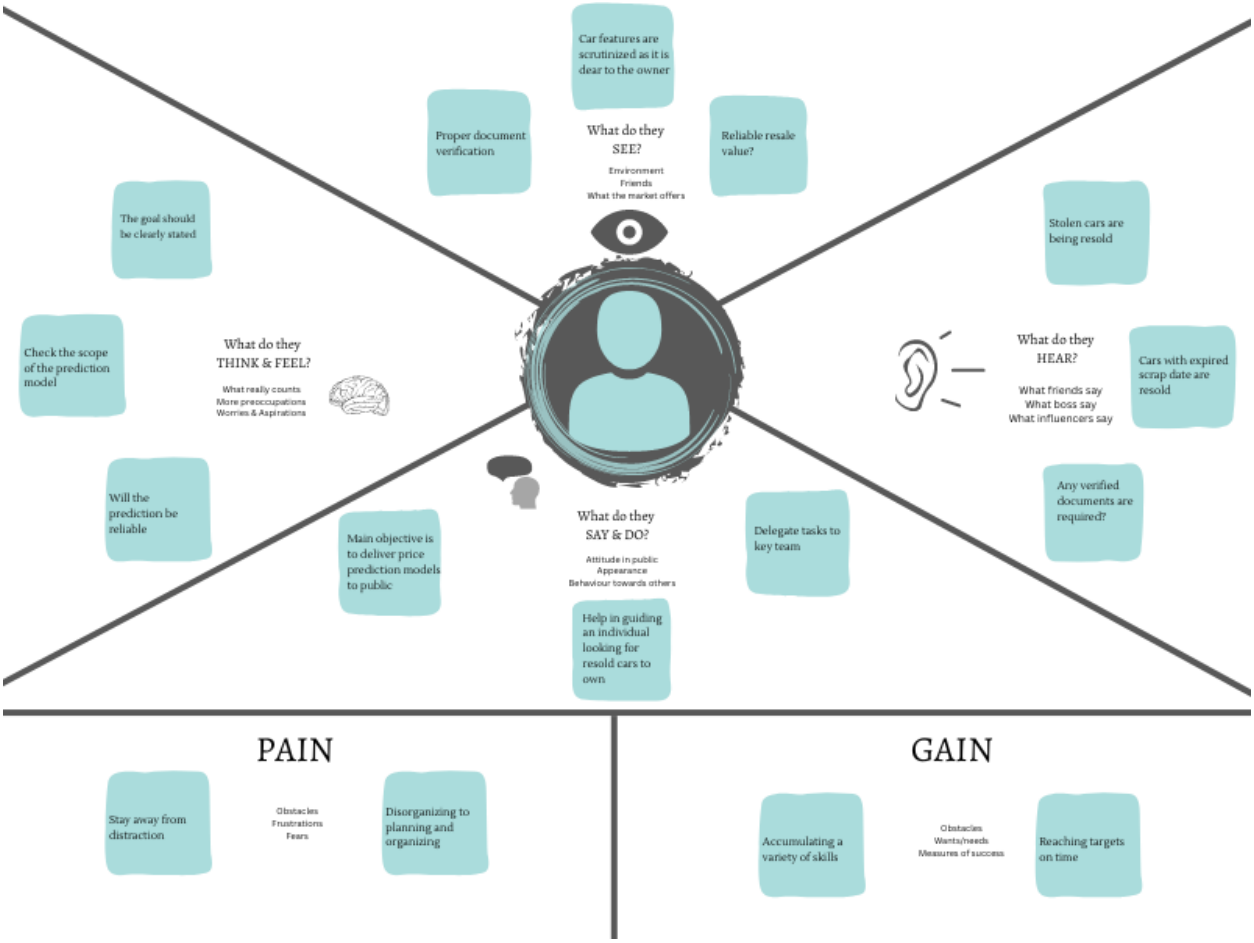
There is a critical need to close the gap between sellers and buyers due to the enormous demand for used cars and the shortage of professionals who can evaluate the proper valuation. This project focuses on developing a system that can impartially and accurately forecast the resale value of cars based on simple features like miles driven, the year of purchase, fuel type, etc. In this research, we have developed automotive resale value prediction systems using machine learning approaches while taking into account various car attributes. Only a few features are currently used to forecast a car's resale value. This can be expanded to include more features and input sets.

Problem Statement (PS)	I am (Customer)	I'm trying to	But	Because	Which makes me feel
PS-1	Entrepreneur	Buy a used car	I am unaware of the varieties available	I don't have correct guidance	Exhausted
PS-2	Normal man	Find a good second hand car	I am unaware about the price and other factors	I am unable to know the current updates	Stressed
PS-3	Owner of a car	Sell	I don't know about its current value	Of its ear and tear	Confused

3. IDEATION & PROPOSED SOLUTION

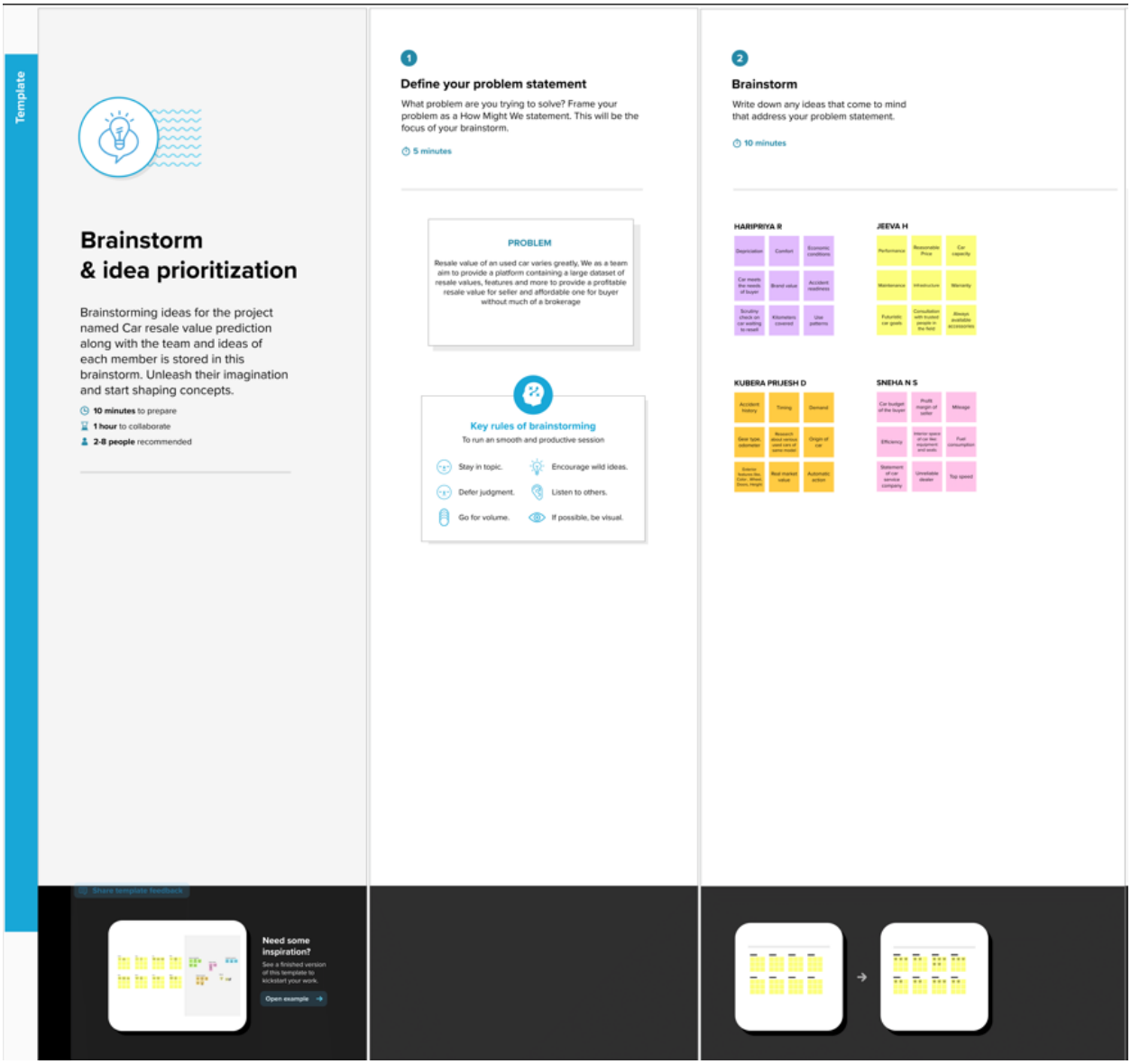
3.1 EMPATHY MAP CANVAS

An empathy map is a simple, easy-to-digest visual that captures knowledge about a user's behaviors and attitudes. It is a useful tool to help 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.

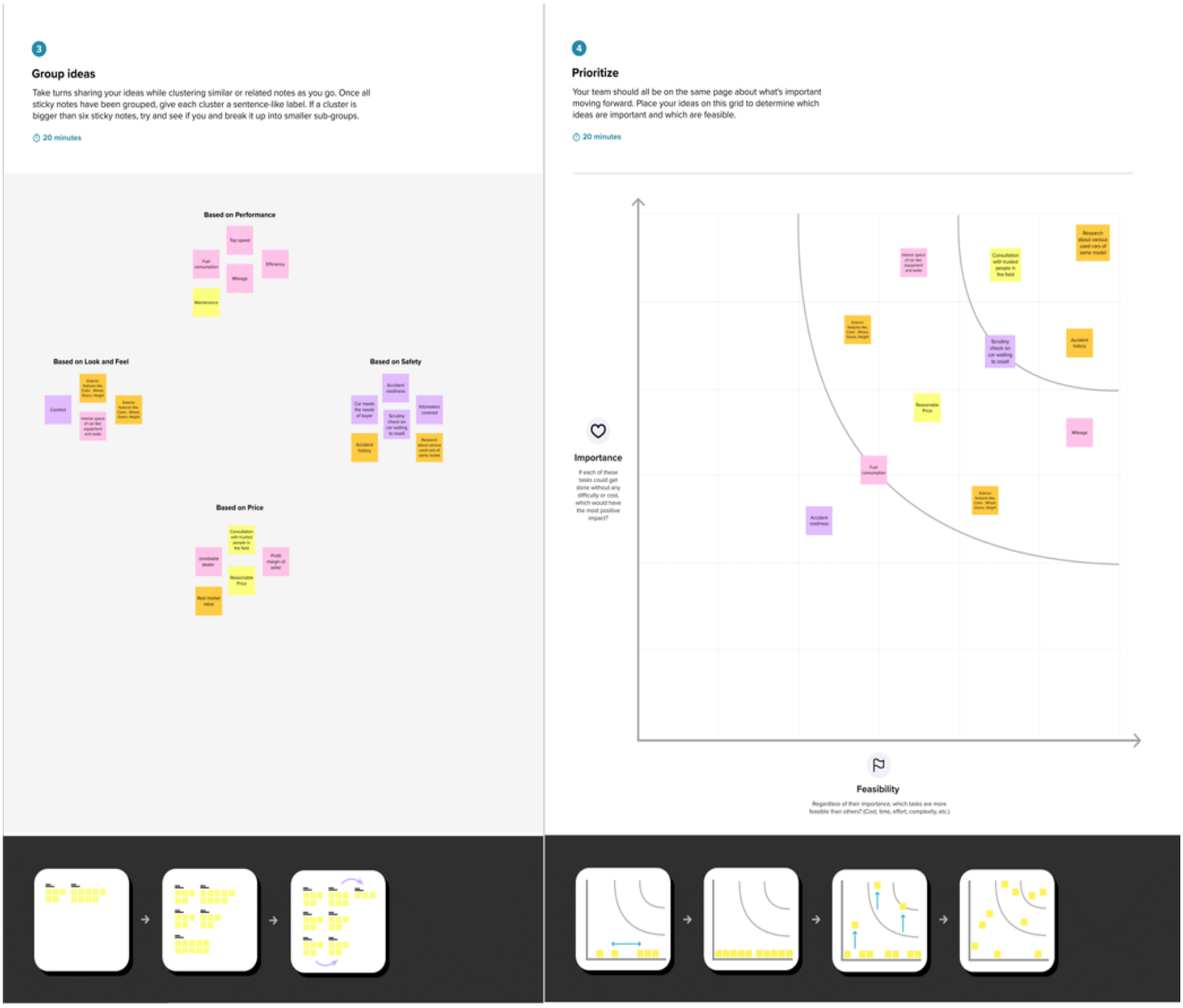


3.2 IDEATION & BRAINSTORMING

Step 1: Team Gathering, collaborating & selecting the problem statement



Step 2: Brainstorming, Idea Listing & Grouping

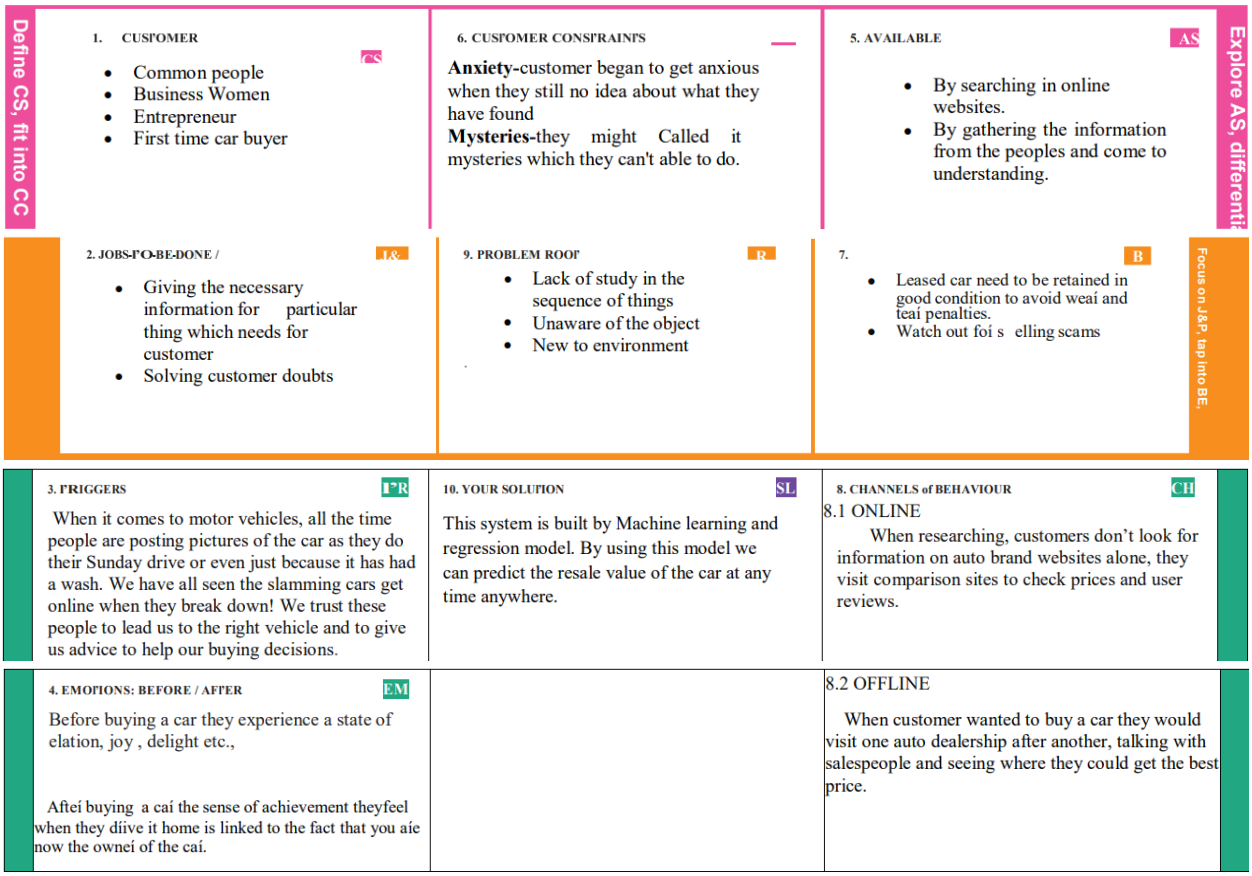


3.3 PROPOSED SOLUTION

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	Therefore, anyone who wants to sell their car must bring it to the appropriate corporate workshop or schedule an appointment with the business to acquire a price estimate. It takes a lot of time and energy to complete this process.
2.	Idea / Solution description	A used buy is more economical as well as more practical, especially for first-timers. A used automobile may easily accommodate practically all types of consumers, as long as you don't absolutely need the newest model on the market or just want the fragrance of a new car.
3.	Novelty / Uniqueness	For a higher car resale value, appearance matters. Good car resale value is guaranteed by a service. Keep all documents organised. Reselling cars is novel. Find Novelty Tata's Phone Numbers, Address, Photos, and Maps.
4.	Social Impact / Customer Satisfaction	Became obsessed with customer feedback, Create a sense of convenience, Deliver fast responses, satisfaction is a company –wide focus. Customer Satisfaction Look and Style Fuel consumption Pulling Power Seating Capacity Riding Comfort Safety Features Speed Shock Absorbs & transmission Tyre mileage Braking Efficiency
5.	Business Model (Revenue Model)	How to start a car merchant business. Generally, it is considered that if you want to start a car merchant business, you need a huge capital to invest. Dealer license.

		Location of the business. Keep a watch on the market. Make your catalog. Use a perfect marketing strategy.
6.	Scalability of the Solution	The size of the used car market in India was over 4.4 million units in 2020, according to Statista. The startup has managed to strive ahead by leveraging a robust managed marketplace business model, while proving that it is economically viable and independent of scale due to the use of technology, economy of scale, economy of scope, asset light, and network effects.

3.4 PROBLEM SOLUTION FIT



4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

The functional requirements for the suggested solution are listed below.

FR No.	Functional Requirement	Sub Requirement
FR-1	User Registration	Using a Form to Register using Gmail to sign up using LinkedIN to sign up
FR-2	User Confirmation	Email confirmation required Confirmation via OTP
FR-3	User Profile	User Details
FR-4	Database	Car Database Customer Database
FR-5	Features and technology	Performance of the car , fuel capacity , mileage etc.,
FR-6	Feedback	Feedback through Form Feedback through Gmail Feedback through LinkedIN

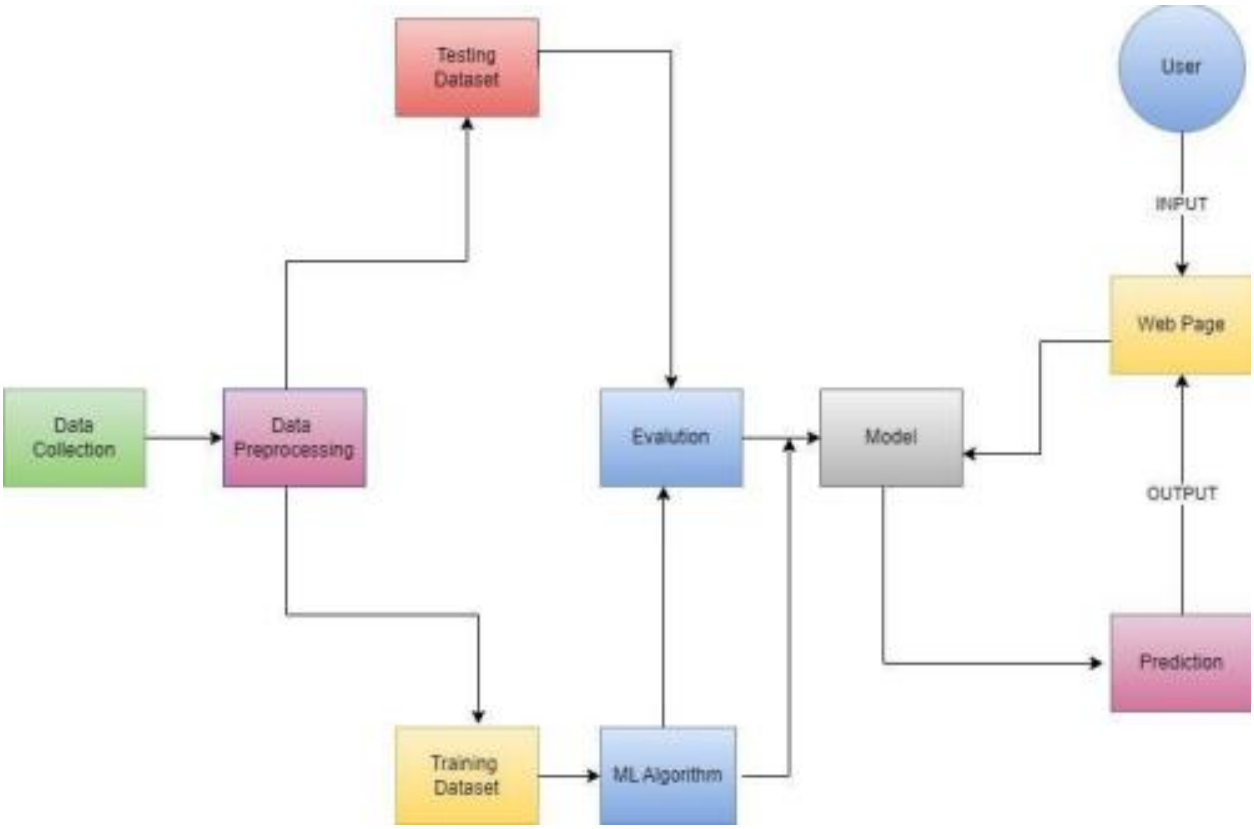
4.2 NON-FUNCTIONAL REQUIREMENTS

The non-functional requirements for the suggested solution are listed below.

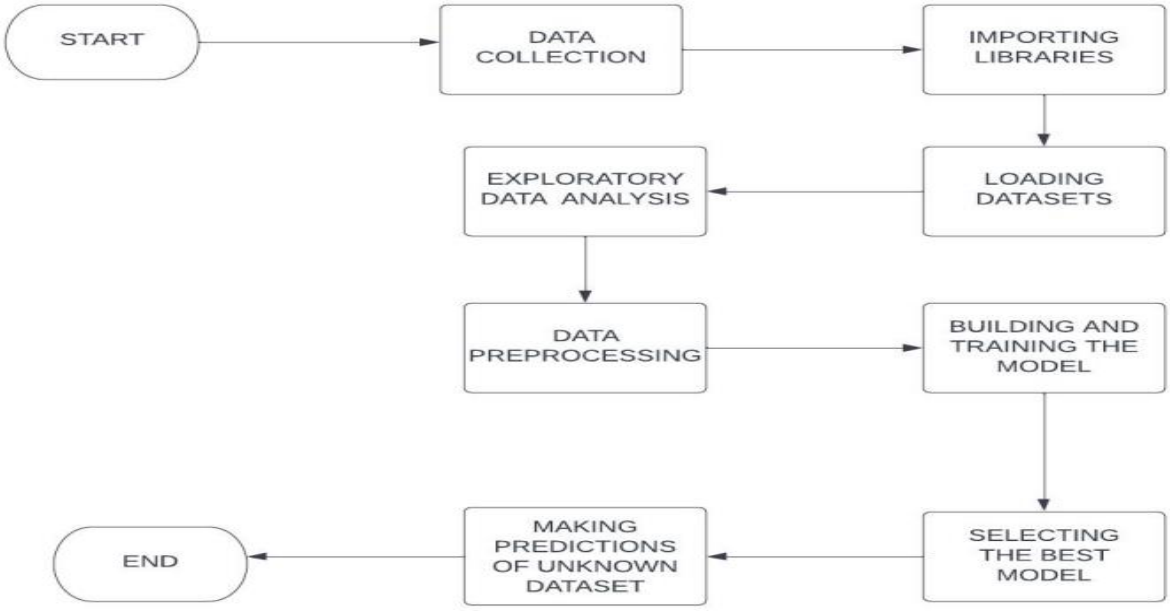
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Excellent user interface, Precise value prediction
NFR-2	**Security**	Securing user passwords and personal information
NFR-3	**Reliability**	Failure occurs less frequently and is failure free.
NFR-4	**Performance**	execute accurate prediction value, The landing page must accommodate multiple users and have a response time of no more than five seconds.
NFR-5	**Availability**	Services must always be available without interruption, with the exception of when servers are being updated.
NFR-6	**Scalability**	It can instantaneously serve millions of users located all over the world and manage any amount of data and computations in a time- and money-saving manner.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



5.2 SOLUTION & TECHNICAL ARCHITECTURE



5.3 USER STORIES

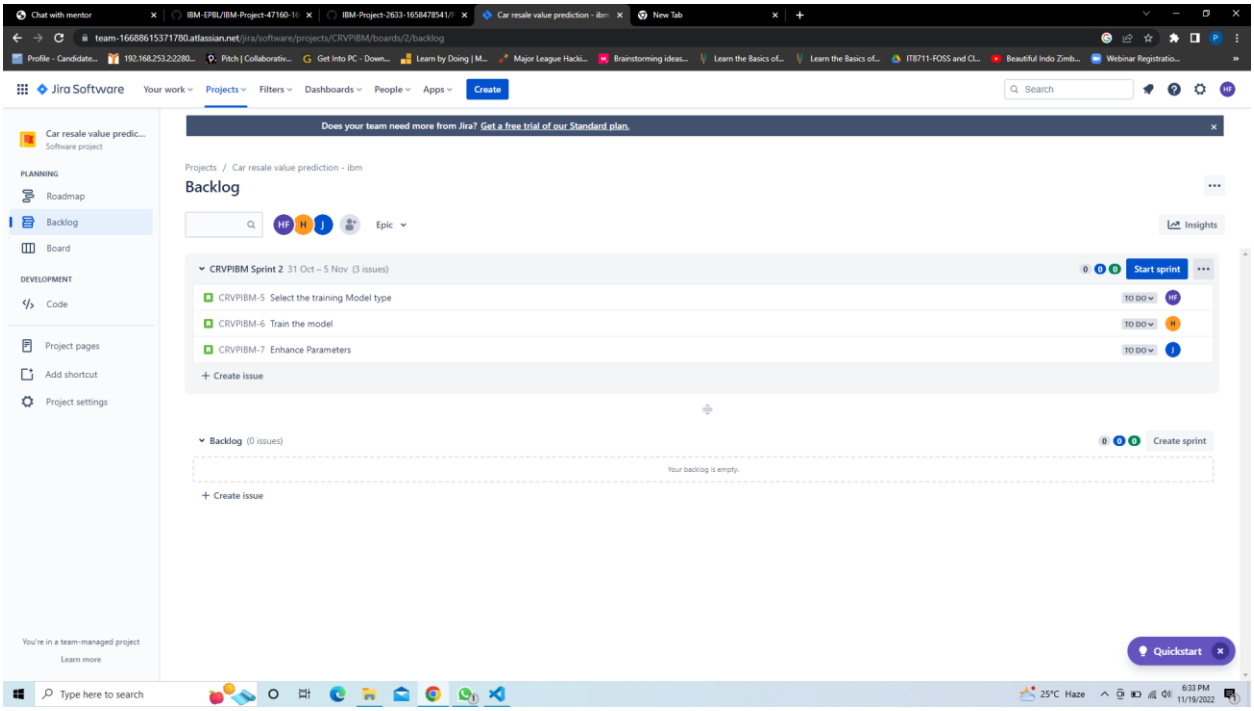
User type	Functional requirement (epic)	User Story Number	User Story/Task	Acceptance criteria	Priority	release
Customer (mobile user)	Data entry	USN-1	I can add information about my car to the application as a user.	I can input the automobile information.	Medium	Sprint-1
Customer (mobile user)	Obtain output	USN-2	I will be given access to the application's car resale value as a user.	I'm able to get my car's resale value.	High	Sprint-1
Customer (mobile user)	Data entry	USN-1	I can enter the car's information into the application as a user.	I can enter the car details	Medium	Sprint-1
Customer (mobile user)	Obtain output	USN-2	I will be given access to the application's car resale value as a user.	I'm able to get my car's resale value.	High	Sprint-1

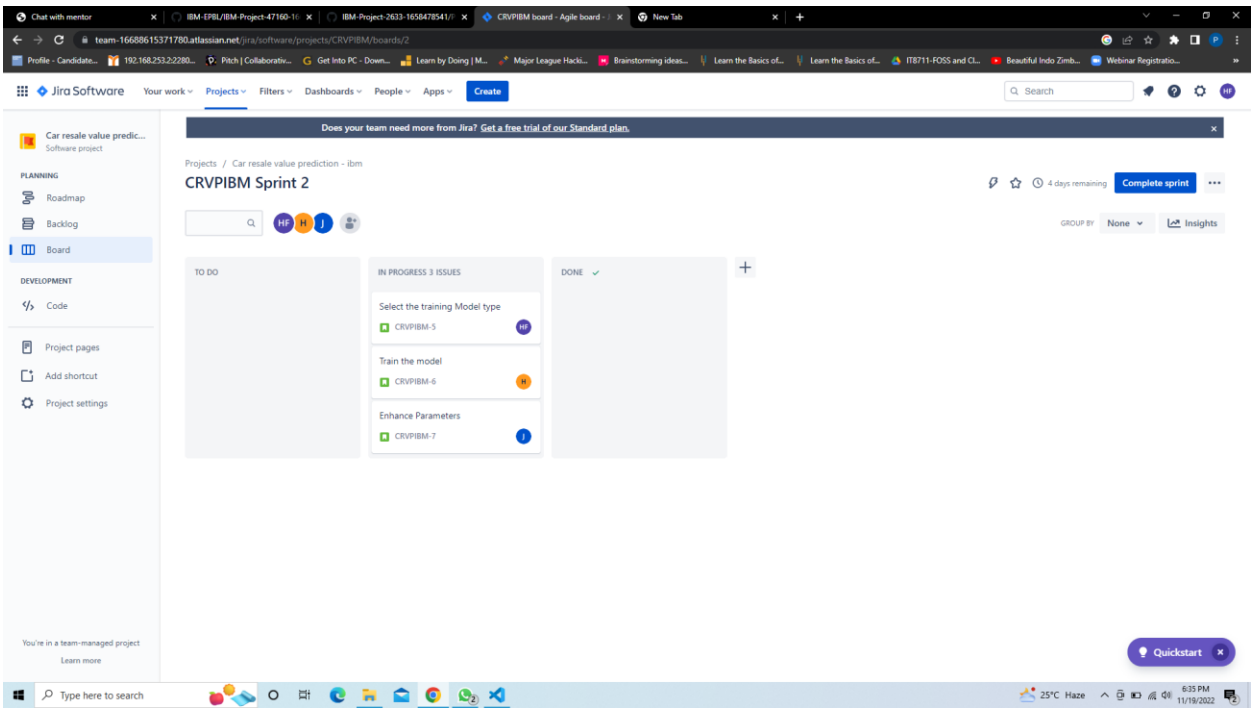
6. PROJECT PLANNING & SCHEDULING

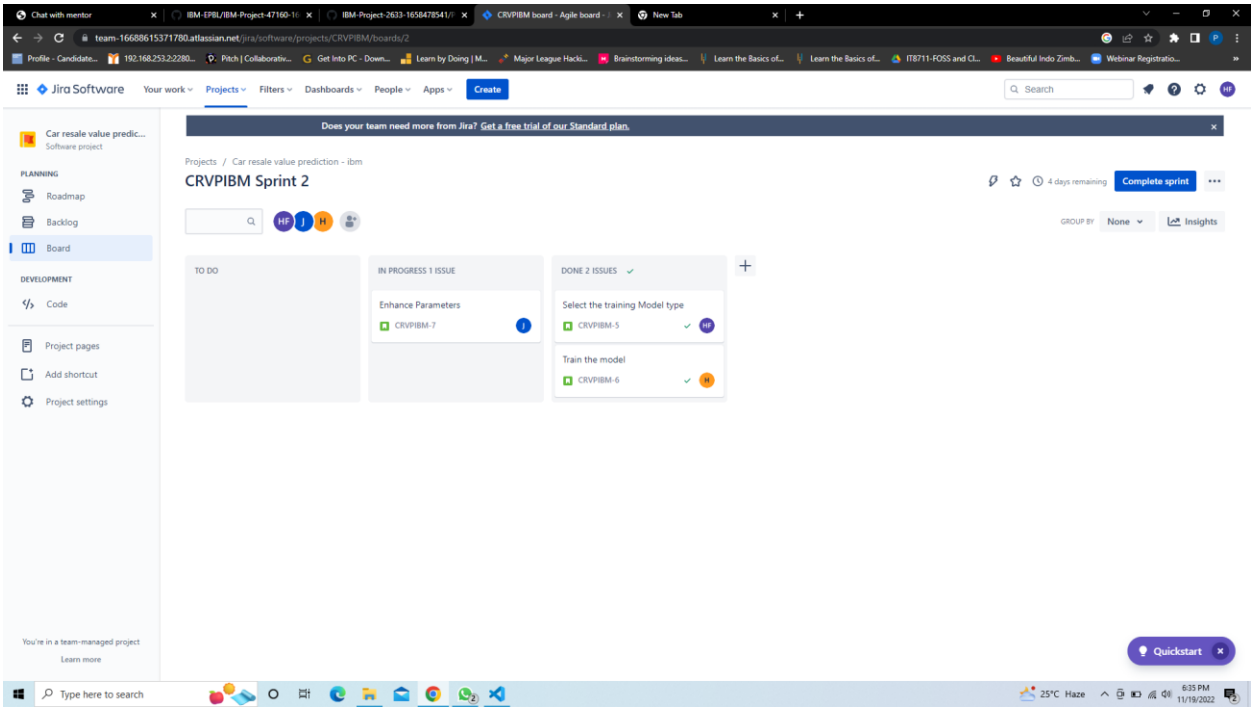
6.1 SPRINT PLANNING & ESTIMATION

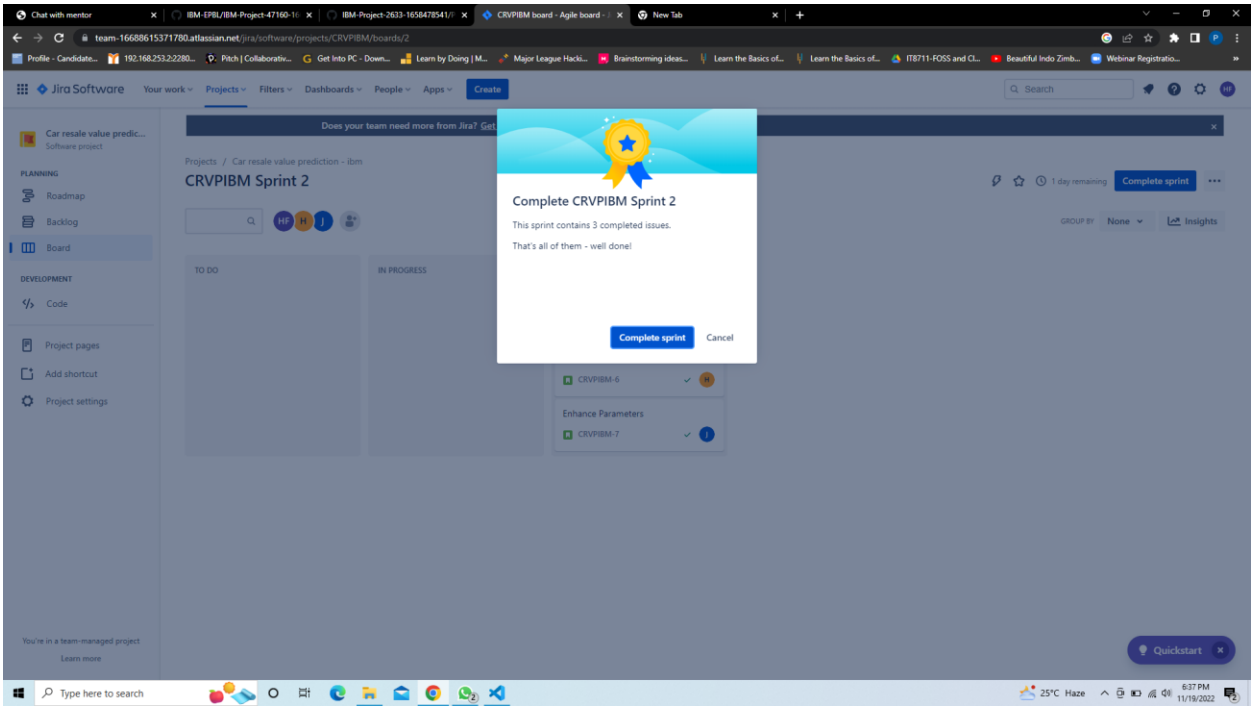
6.2 SPRINT DELIVERY SCHEDULE

6.3 REPORTS FROM JIRA


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Car resale value predic...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Car resale value prediction - ibm
Backlog
Epic
Insights
CRVPIBM Sprint 2 31 Oct – 5 Nov (3 issues)
Start sprint
CRVPIBM-5 Select the training Model type
CRVPIBM-6 Train the model
CRVPIBM-7 Enhance Parameters
TO DO
+ Create issue
Backlog (0 issues)
Create sprint
Your backlog is empty.
+ Create issue
Quickstart
Type here to search
25°C Haze
6:33 PM
11/19/2022


Jira Software
Your work
Projects
Filters
Dashboards
People
Apps
Create
Search
Car resale value predic...
Software project
PLANNING
Roadmap
Backlog
Board
DEVELOPMENT
Code
Project pages
Add shortcut
Project settings
You're in a team-managed project
Learn more
Does your team need more from Jira? Get a free trial of our Standard plan.
Projects / Car resale value prediction - ibm
CRVPIBM Sprint 2
4 days remaining
Complete sprint
GROUP BY
None
Insights
TO DO
IN PROGRESS 3 ISSUES
DONE
Select the training Model type
CRVPIBM-5
Train the model
CRVPIBM-6
Enhance Parameters
CRVPIBM-7
Quickstart
Type here to search
25°C Haze
6:35 PM
11/19/2022





7. CODING

7.1 FEATURE 1

7.2 FEATURE 2

8. TESTING

app.py:

```
import pickle
import numpy as np
import pandas as pd
from flask import Flask, render_template, request

app = Flask(__name__)

model = pickle.load(open(r'LinearRegressionModel.pkl', 'rb'))
car = pd.read_csv('Cleaned_dataset.csv')


@app.route('/')
def index():
    companies = sorted(car['Brands'].unique())
    car_models = sorted(car['Car_names'].unique())
    year = sorted(car['year'].unique(), reverse=True)
    fuel_type = car['fuel_type'].unique()

    companies.insert(0, 'Select Company')
    return render_template('index.html', companies=companies,
car_models=car_models, years=year, fuel_types=fuel_type)


@app.route('/predict', methods=['POST'])
def predict():
    company = request.form.get('company')

    car_model = request.form.get('car_models')
    year = request.form.get('year')
    fuel_type = request.form.get('fuel_type')
    driven = request.form.get('kilo_driven')

    prediction = model.predict(pd.DataFrame(columns=['name',
'company', 'year', 'kms_driven', 'fuel_type'],

data=np.array([car_model, company, year, driven,
fuel_type]).reshape(1, 5)))
    print(prediction)
```

```
    return str(np.round(prediction[0], 2))


if __name__ == '__main__':
    app.run()
```

car_resale_value_prediction.ipynb:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as mpl
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import *
from sklearn.model_selection import *
from sklearn.ensemble import *
from sklearn.feature_selection import *
from sklearn.metrics import *
from sklearn.tree import *
from sklearn.svm import *
from sklearn.preprocessing import LabelEncoder
import pickle
import warnings

warnings.filterwarnings('ignore')
%matplotlib inline
mpl.style.use('ggplot')
data = pd.read_csv('Datasets_of_cars.csv')

data.head()
data.shape
data.info()
```

```
backup=data.copy()
data['year'].unique()
data['Price'].unique()
data['kms_driven'].unique()
data['fuel_type'].unique()
data['Car_names'].unique()
#year has many non-year values
data=data[data['year'].str.isnumeric()]
#year is in object. Change to integer
data['year']=data['year'].astype(int)
#Price has Ask for Price
data=data[data['Price']!='Ask For Price']
#Price has commas in its prices and is in object
data['Price']=data['Price'].str.replace(',','').astype(int)
#kms_driven has object values with kms at last
data['kms_driven']=data['kms_driven'].str.split().str.get(0).str.replace(',','')
#It has nan values and two rows have 'Petrol' in them
data=data[data['kms_driven'].str.isnumeric()]
data['kms_driven']=data['kms_driven'].astype(int)
#fuel_type has nan values
data=data[~data['fuel_type'].isna()]
data.shape
data['Car_names']=data['Car_names'].str.split().str.slice(start=0,stop=3).str.join(' ')
#Resetting the index of the final cleaned data
data=data.reset_index(drop=True)
data
data.to_csv('Cleaned_datasets.csv')
data.info()
data.describe()
data['Brands'].unique()
data.hist(figsize=(8,8))
sns.pairplot(data)
sns.boxplot(data.Price)
Extracting Training Data
X=data[['Car_names','Brands','year','kms_driven','fuel_type']]
y=data['Price']
X
```

```
y.shape
#Encoding Techniques
le=LabelEncoder()
data.head()
plt.figure(figsize=(10,8))
sns.heatmap(data.corr(),annot=True)
# dependent variable
y=data['Price']
y
#independent variable
X=data.drop(columns=['Price'],axis=1)
X.head()
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
X_train.shape
y_train.shape
X_test.shape
y_test.shape
X = data.drop(columns='Price')
y = data['Price']
from sklearn.model_selection import train_test_split
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2)
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import OneHotEncoder
from sklearn.compose import make_column_transformer
from sklearn.pipeline import make_pipeline
from sklearn.metrics import r2_score
enc=OneHotEncoder()
enc.fit(X[['Car_names','Brands','fuel_type']])
column_trans=make_column_transformer((OneHotEncoder(categories=enc.catego
ries_),['Car_names','Brands','fuel_type']),
                                     remainder='passthrough')
# Linear Regression model
mod=LinearRegression()
# Making a pipeline
pipe=make_pipeline(column_trans,mod)
# Fitting the model
```

```
pipe.fit(X_train,y_train)
#Predicting the values to test
y_pred=pipe.predict(X_test)
y_pred
#printing the accuracy
r2_score(y_test,y_pred)
scores=[]
for i in range(1000):

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=i)
    mod=LinearRegression()
    pipe=make_pipeline(column_trans,mod)
    pipe.fit(X_train,y_train)
    y_pred=pipe.predict(X_test)
    scores.append(r2_score(y_test,y_pred))

np.argmax(scores)

scores[np.argmax(scores)]

pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti
Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))

X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.1,random_state=np.
argmax(scores))

lr=LinearRegression()

pipe=make_pipeline(column_trans,lr)

pipe.fit(X_train,y_train)

y_pred=pipe.predict(X_test)

#R2score
r2_score(y_test,y_pred)
```

 # Saving the Model to a Pickle File

```
pickle.dump(pipe,open('LinearRegressionModel.pkl','wb'))
pipe.predict(pd.DataFrame(columns=X_test.columns,data=np.array(['Maruti
Suzuki Swift','Maruti',2019,100,'Petrol']).reshape(1,5)))
pipe.steps[0][1].transformers[0][1].categories[0]
```

13.2 GITHUB & PROJECT DEMO LINK