TEAM ID	**PNT2022TMID12421**
TITLE	**CAR RESALE VALUE PREDICTION**
DATE	**19.11.2022**

1.INTRODUCTION

1.1 Project Overview

With the rapid growth of the number of private cars and the development of the second-hand car market, second hand cars have become the main choice when people buy cars. The online second-hand car platform provides both buyers and sellers the chance of online P2P trade. In such systems, the accuracy of secondhand car price evaluation largely determines whether the seller and the buyer can get more efficient trading experience. In this paper, the price evaluation model based on big data analysis is proposed, which takes advantage of widely circulated vehicle data and a large number of vehicle transaction data to analyze the price data for each type of vehicles by using the optimized BP neural network algorithm. It aims to establish a second-hand car price evaluation model to get the price that best matches the car. In this paper, the random forest algorithm is build a model, which improves the accuracy of the prediction model. This system is compared with other regression models. As a result, the random forest algorithm is better than others as well as the accuracy is higher.

1.2 Purpose

The used car market is an ever-rising industry, which has almost doubled its market value in the last few years. The emergence of online portals such as CarDheko,Quikr, Carwale,Cars24, and many others has facilitated the need for both the customer and the seller to be better informed about the trends and patterns that determine the value of the used car in the market. Considering the demand for private car all around the world, the demand of secondhand car market has been rising and creating a chance in business for both buyer and seller. In several countries, buying a used car is the best choice for customer because its price is reasonable and affordable by buyer. After few years of using them, it may get a profit from resell again. However, various factors influence the price of a used car such as how old of those vehicles and the condition in current scenario of them. Normally, the price of used cars in the market is not constant. Thus, car price evaluation model is required for helping in trading. In this paper,

we conducted a comparative study using multiple linear regression and random forest regression to build a price model of used car. In this project we have used different algorithms with different techniques for developing car resale value prediction

systems considering different features of the car. The data set primarily comprises of categorical attributes along with quantitative attributes.

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. User enters the details of the car into the form given and accordingly the car resale value is predicted.

2.LITREATURE SURVEY

2.1 Existing problem

As a commodity, a used car not only has its attributes that will affect the price, but also some external factors will also affect the value of the used car. Factors affecting the value of used cars should be fully considered, as well as the availability of indicator data. This paper will analyze the factors of the vehicle itself and the market and analyze the factors that affect the price of used cars in combination with the parameters of used cars, vehicle condition factors, and transaction factors.

2.2 References

At present, under the guidance of the new generation of information technology, the rapid accumulation of data, the continuous improvement of computing power, the continuous optimization of algorithm models, and the rapid rise of multi-scene applications have made profound changes in the development environment of Machine Learning.

1. Aparna Varma; Harsh Valecha; Ishita Khare; Aakash Sachdeva; Mukta Goyal (2018), "Prediction of Consumer Behaviour using Random Forest Algorithm DOI:10.1109/ UPCON.2018.8597070 , Conference Paper , Publisher: IEEE.

2. Doan Van Thai; Luong Ngoc Son; Pham Vu Tien; Nguyen Nhat Anh; Nguyen Thi Ngoc Anh (2019), "Prediction car prices using quantify qualitative data and knowledge-based system ", Conference Paper , Publisher: IEEE.

3. Geetha. V; A. Punitha ;M. Abarna; M. Akshaya; S. Illakiya; A.P. Janani (2020), "An Effective Crop Prediction Using Random Forest Algorithm", Conference Paper , Publisher: IEEE.

4. Nazmia Kurniawati; Dianing Novita Nurmala Putri; Yuli Kurnia Ningsih (2020), "Random Forest Regression for Predicting Metamaterial Antenna Parameters" , Conference Paper , Publisher: IEEE.

5. Ning Sun; Hongxi Bai; Yuxia Geng; Huizhu Shi (2017), " Price evaluation model in second-hand car system based on BP neural network theory ", DOI:10.1109 /SNPD.2017.8022758 , Conference Paper , Publisher: IEEE.

6. Rita Samikannu; Jitendra Kumar Jaiswal; (2017), " Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression" , World Congress on Computing and Communication Technologies (WCCCT)

7. Sabir Buya; Nitis Monburinon; Prajak Chertchom; Thongchai Kaewkiriya; Suwat Rungpheung; Pitchayakit Boonpou (2018), "Prediction of prices for used car by using regression models",DOI:10.1109/ICBIR.2018.8391177 , Conference Paper , Publisher: IEEE.

8. Suganya G; Pranav Motarwar; Ankita Duraphe; M Premalatha (2020), "Cognitive Approach for Heart Disease Prediction using Machine Learning" , Conference Paper, Publisher: IEEE.

9. Varma; Amuluru Devi Chaitrasree; Penmetsa Chidananda Varma; C. Lakshmi (2017), "Random Forest Algorithm for the Prediction of Diabetes", DOI:10.1109/ICSCAN.2019.8878802 , Conference: 2019 IEEE InternationalConference on System, Computation, Automation and Networking (ICSCAN) , Conference Paper , Publisher: IEEE.

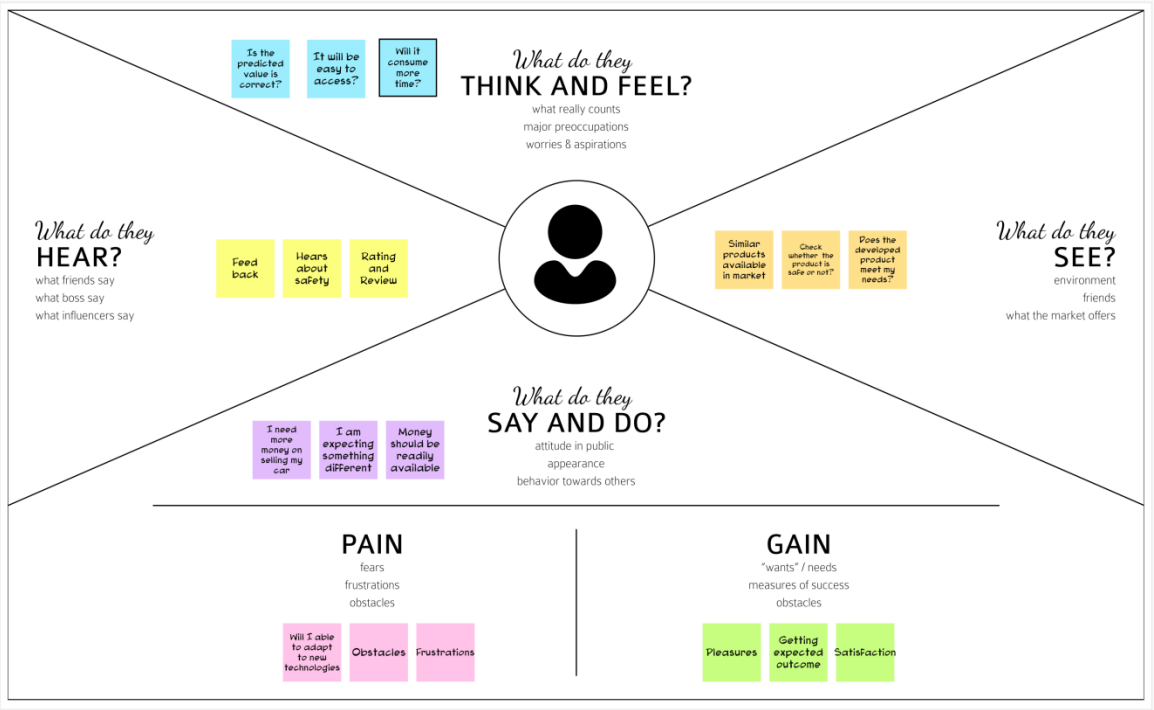
10. Vijay Kumar; S. V. Patel; Veena N. Jokhakar in "A random forest based machine learning approach for mild steel defect diagnosis" , 2016 IEEE International Conference on Computational Intelligence and Computing Research(ICCIC) , Conference Paper , Publisher: IEEE.

2.3 Problem statement definition

Considering the demand for private car all around the world, the demand of second-hand car market has been rising and creating a chance in business for both buyer and seller. In several countries, buying a used car is the best choice for customer because its price is reasonable and affordable by buyer. After few years of using them, it may get a profit from resell again. However, various factors influence the price of a used car such as how old of those vehicles and the condition in current scenario of them. Normally, the price of used cars in the market is not constant. Thus, car price evaluation model is required.

3.IDEATION & PROPOSED SOLUTION

3.1 Empathy map canvas



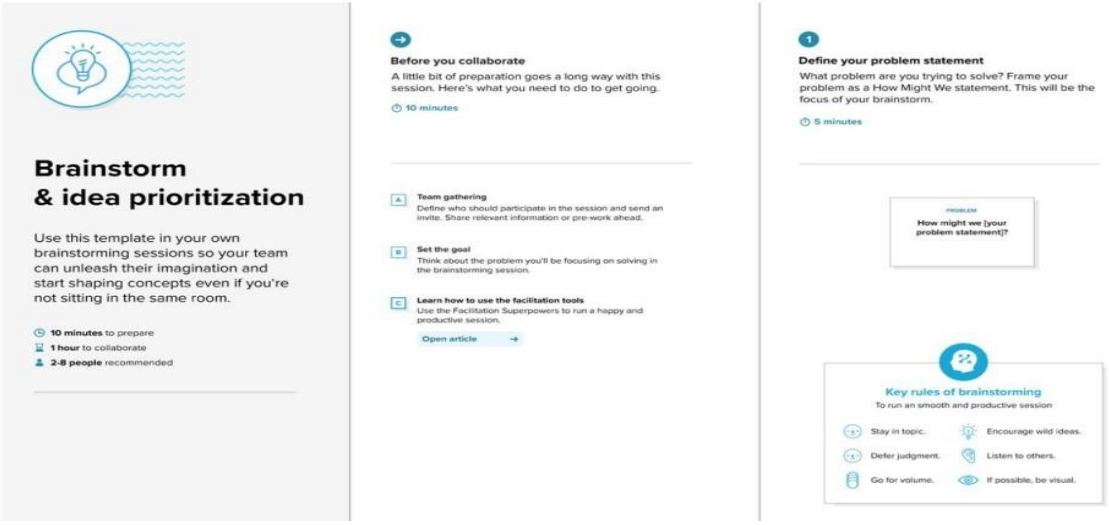
3.2 Ideation &Brainstorming

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

STEP 1

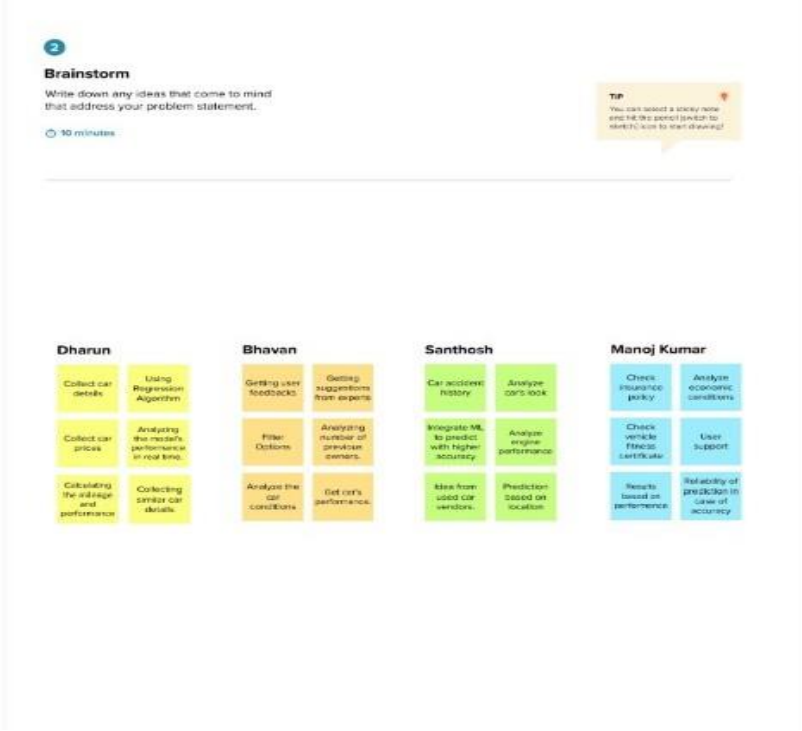
- Prediction using Car image.
- By using the exterior and interior image of the car.
- The value will be predicted based on the appearance of the car.
- By using neural network value of the car can be predicted.

- Neural network algorithm is developed by considering the human brain that takes a set of units as input and transfers results to a predefined output.



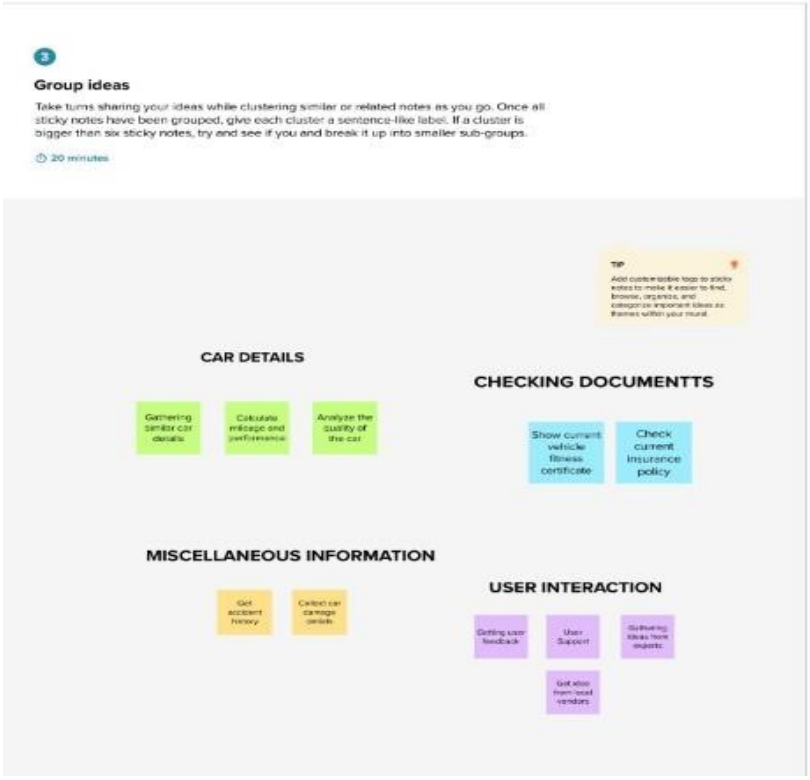
STEP 2

- The main objective of this project is to predict the Prices of used cars, compare the prices and also estimate the lifespan of a particular car.
- Insurance, Company claims, etc.
- Regression Algorithm is used to predict the value.
- Regression model based on k-nearest neighbour machine learning algorithm was used to predict the price of a car.



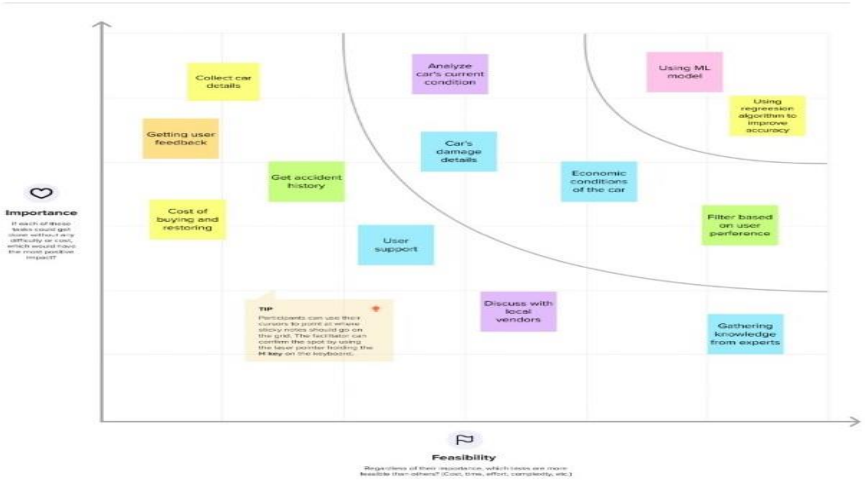
STEP 3

- Prediction using engine car condition. User should upload engine sound in the format of audio file.
- CNNs for Machine Learning on sound data by spectrogram approach that was just converts each song (or song segment) into a spectrogram: a two dimensional matrix 8.



STEP 4

- Economic Conditions.
- Kilo-metres Covered. Its mileage (the number of kilometres it has run) and its horsepower



3.3 Proposed Solution

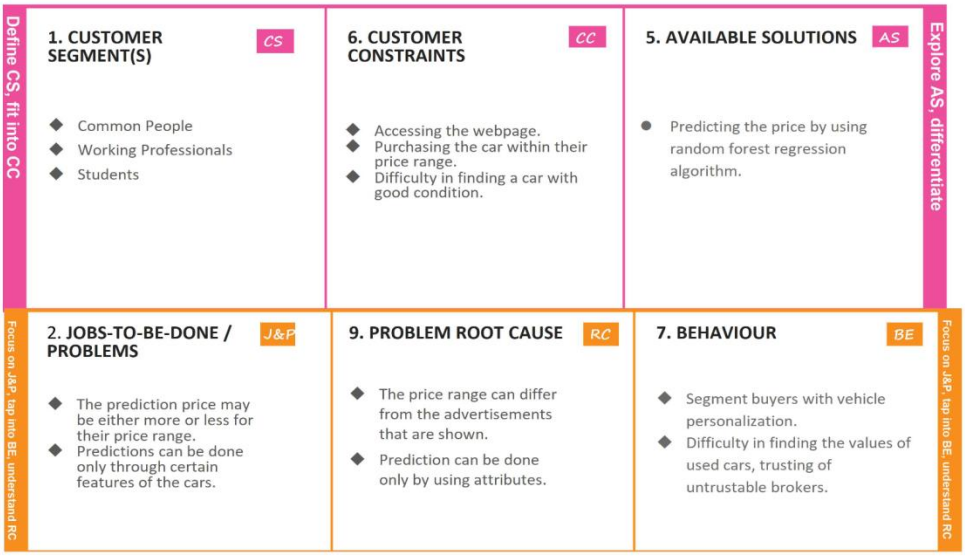
S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict the price of the used cars with user's favorable features.
2.	Idea / Solution description	● To develop a ML model that predicts the price of the used cars with higher accuracy. ● The developed model should posses minimal error. ● Random Forest Algorithm is used to predict the price of the used cars.
3.	Novelty / Uniqueness	By using the regression algorithm(Random Forest), the prediction will be more accurate.
4.	Social Impact / Customer Satisfaction	● Both the seller and buyer can get the idea about the price of the used cars and the prediction rate will be more accurate as compared to any other algorithms. ● User friendly environment that can easily understand by anyone. ● More availability and reliability.
5.	Business Model (Revenue Model)	● Our model will be deployed on cloud and it can be accessed by anyone. ● New companies can use this model for predicting the car price and make this as a startup product.
6.	Scalability of the Solution	● Our model designed in such a manner that it can accept multiple users at a time. ● By using random forest, the prediction speed is faster than other algorithms. ● Our model is cost efficient.

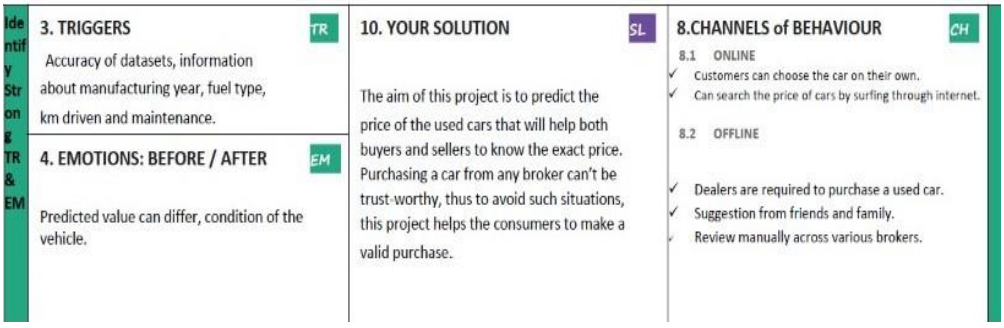
3.3 Proposed Solution

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	To predict the price of the used cars with user's favorable features.
2.	Idea / Solution description	● To develop a ML model that predicts the price of the used cars with higher accuracy. ● The developed model should posses minimal error. ● Random Forest Algorithm is used to predict the price of the used cars.
3.	Novelty / Uniqueness	By using the regression algorithm(Random Forest), the prediction will be more accurate.
4.	Social Impact / Customer Satisfaction	● Both the seller and buyer can get the idea about the price of the used cars and the prediction rate will be more accurate as compared to any other algorithms. ● User friendly environment that can easily understand by anyone. ● More availability and reliability.
5.	Business Model (Revenue Model)	● Our model will be deployed on cloud and it can be accessed by anyone. ● New companies can use this model for predicting the car price and make this as a startup product.
6.	Scalability of the Solution	● Our model designed in such a manner that it can accept multiple users at a time. ● By using random forest, the prediction speed is faster than other algorithms. ● Our model is cost efficient.

3.4 Problem solution fit

Project Title: Car Resale Value Prediction **Project Design Phase-I - Solution Fit Template** **Team ID:** PNT2022TMID12421





4.REQUIREMENT ANALYSIS

4.1 Functional requirement

The functional Requirements of this projects involves the better understanding of Pre-processing, Application designing using HTML & CSS and IBM Watson Cloud. IBM Watson provides the services such as Database, deployment etc. **Hardware requirements**

- Processor - Dual Core 2.4 GHz [i5 or i7 or equivalent AMD]
- RAM - Min 4 GB
- Memory - Min 4 GB

Software requirement

- OS - Windows 8, 10 or 11
- Pycharm
- VsCode
- Google Colab

4.2 Non Functional requirement

The Non - Functional Requirements of this project are

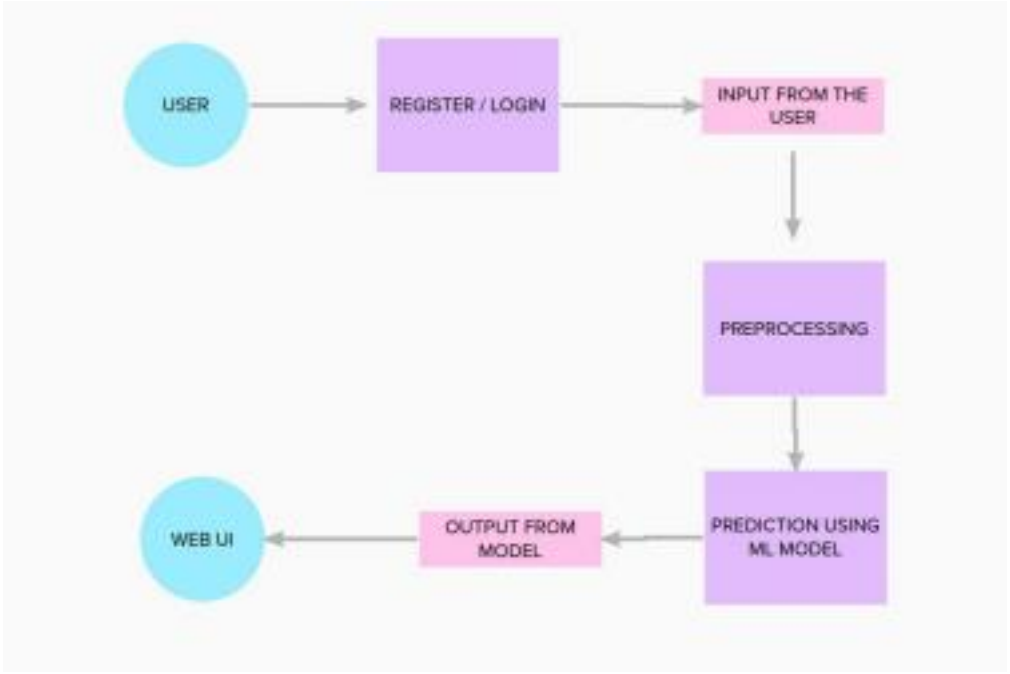
- Highly accurate Image Predictive model
- Better user responsive web application
- Cloud database for storing the informations

5.PROJECT DESIGN

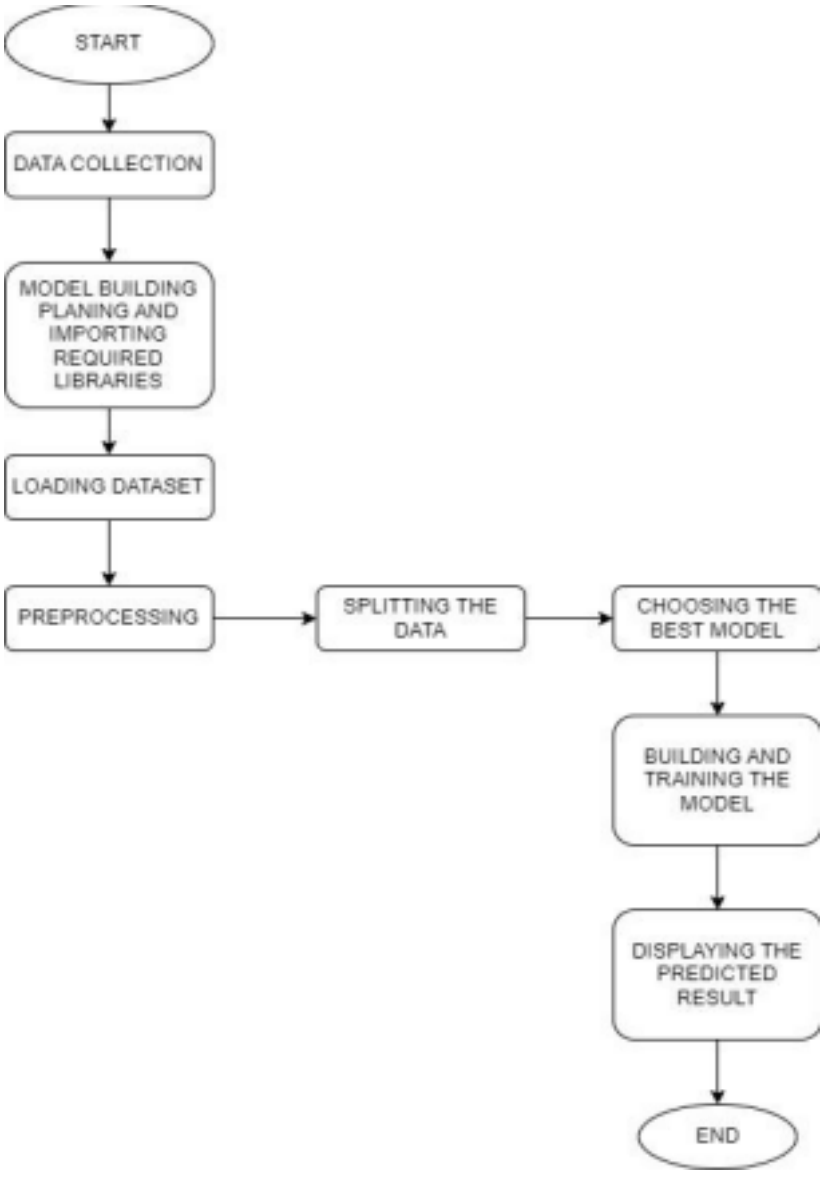
5.1Data flow diagrams

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

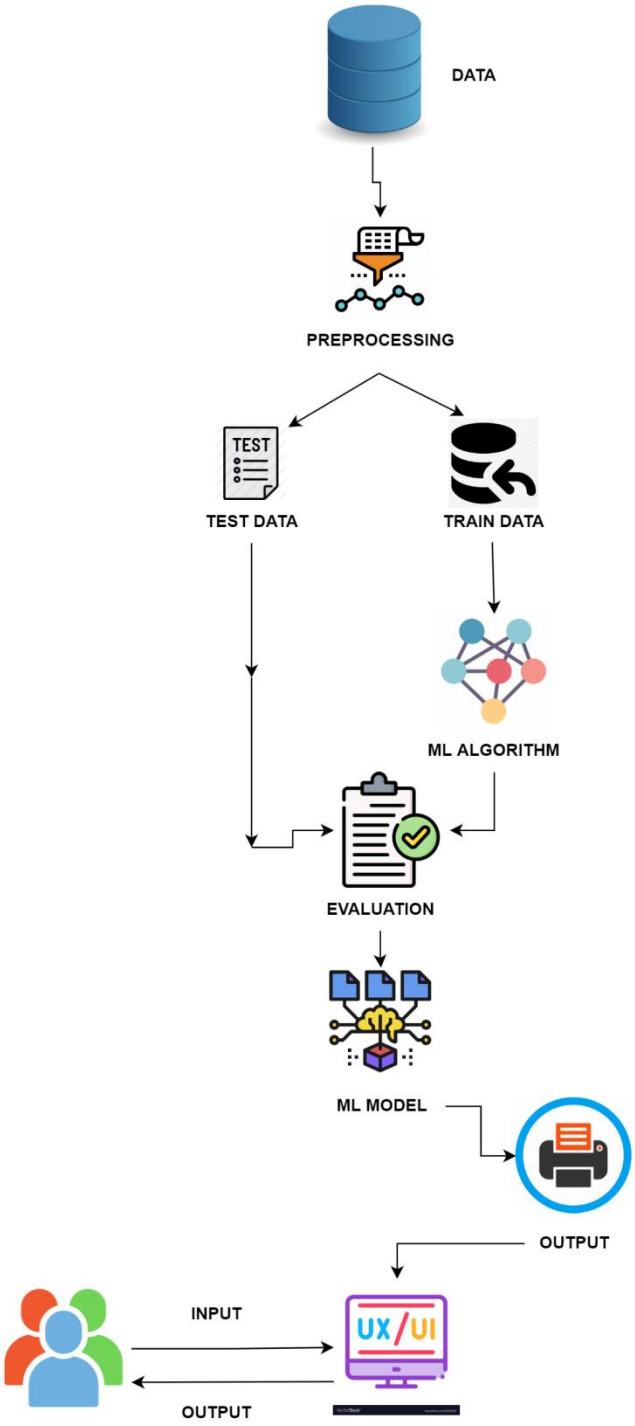
DATA FLOW (Simplified)



DFD:

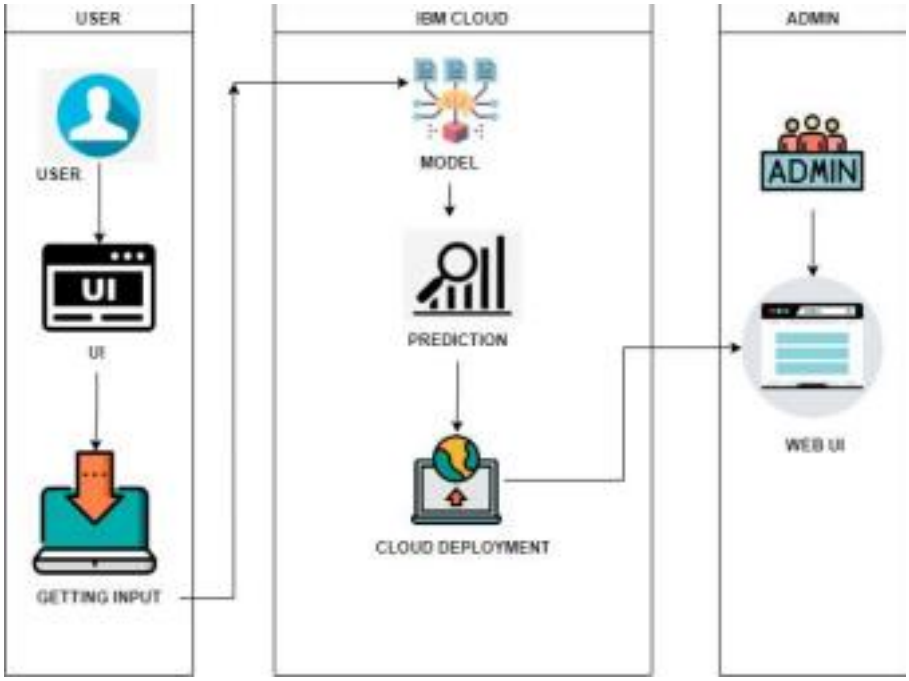

START
DATA COLLECTION
MODEL BUILDING PLANING AND IMPORTING REQUIRED LIBRARIES
LOADING DATASET
PREPROCESSING
SPLITTING THE DATA
CHOOSING THE BEST MODEL
BUILDING AND TRAINING THE MODEL
DISPLAYING THE PREDICTED RESULT
END

5.2 Solution architecture



5.3 Technology architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2



Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Web UI, Mobile App, Chatbot etc.	HTML, CSS, JavaScript / Angular Js / React Js etc.
2.	Value Prediction	ML model to predict the price of the used cars.	Python
3.	Database	Big data that are stored In nosql databases.	NoSQL, etc.
4.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant etc.
5.	File Storage	File storage requirements	IBM Block Storage.
6.	Machine Learning Model	Purpose of Machine Learning Model	Used Car value Prediction model.
7.	Infrastructure (Server / Cloud)	Application Deployment on Cloud Cloud Server Configuration :	Cloud Foundry.

Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List of open-source frameworks used	Google Colab, Anaconda Navigator, Python flask.
2.	Security Implementations	User Encryption	SHA-256, Encryptions,
3.	Scalable Architecture	Must supports higher workload.	IBM cloud, Model performance.
4.	Availability	Availability of application.	Deployment on IBM cloud
5.	Performance	Performance of the application is high as we used Random forest algorithm and the model deployment on cloud.	Random forest regressor, IBM cloud.

5.3 User stories

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data collection	USN-1	Collect the dataset from various sources.	1	Low
	Preprocess the data	USN-2	Import required libraries	1	Low
		USN-3	Read the data	2	Medium
		USN-4	Clean the data	2	Medium
Sprint-2	Model Building	USN-1	Choose appropriate model	3	High
		USN-2	Check the metrics of the model	2	Medium
		USN-3	Applying regression model	3	High
Sprint-3	Application Building	USN-1	Build a HTML page	2	Low
		USN-2	Build python flask application	5	High
		USN-3	Execute and test	5	High
Sprint-4	Train the model	USN-1	Train ML model	5	High

		USN-2	Integrate flask	5	High

6.PROJECT PLANNING

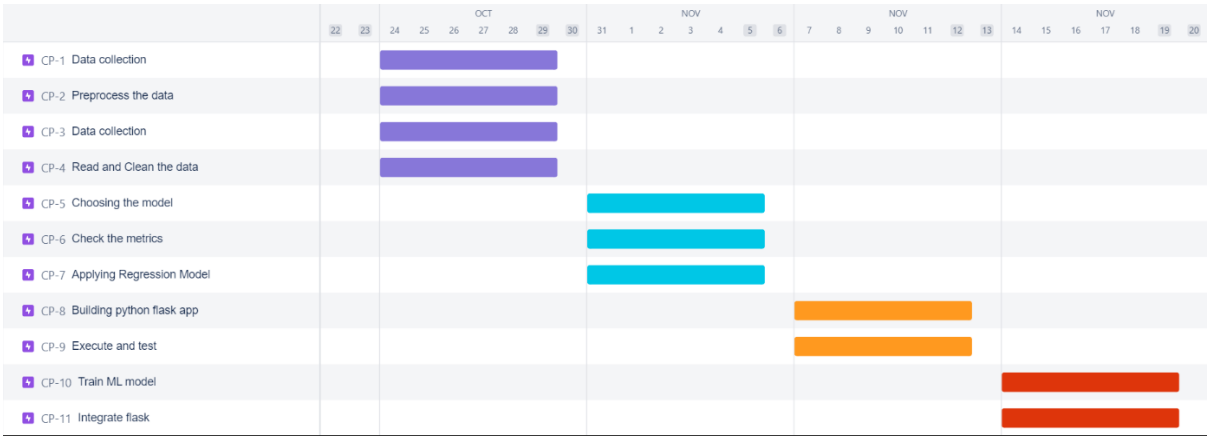
6.1 Sprint planning & estimation

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	6 Days	24 Oct 2022	29 Oct 2022	20	29 Oct 2022
Sprint-2	20	6 Days	31 Oct 2022	05 Nov 2022	20	On progress
Sprint-3	20	6 Days	07 Nov 2022	12 Nov 2022	20	On progress
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	20	On progress

6.2 Sprint Delivery Schedule

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority
Sprint-1	Data collection	USN-1	Collect the dataset from various sources.	1	Low
	Preprocess the data	USN-2	Import required libraries	1	Low
		USN-3	Read the data	2	Medium
		USN-4	Clean the data	2	Medium
Sprint-2	Model Building	USN-1	Choose appropriate model	3	High
		USN-2	Check the metrics of the model	2	Medium
		USN-3	Applying regression model	3	High
Sprint-3	Application Building	USN-1	Build a HTML page	2	Low
		USN-2	Build python flask application	5	High
		USN-3	Execute and test	5	High
Sprint-4	Train the model	USN-1	Train ML model	5	High
		USN-2	Integrate flask	5	High

6.2 Repotrs from JIRA



7.CODING AND SOLUTIONS

Source code:

Car_Resal_pred.ipynb :

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#reading the dataset
data=pd.read_csv('/content/car_data.csv')
print(data.shape)
data.head()
data.isnull().sum()
print('Unique elements in Seller_Type are',data['Seller_Type'].unique())
print('Unique elements in Fuel_Type are',data['Fuel_Type'].unique())
print('Unique elements in Transmission are',data['Transmission'].unique())
print('Unique elements in Owner are',data['Owner'].unique())
print('Unique elements in Year are',data['Year'].unique())
print('Unique elements in Car_Name are',data['Car_Name'].nunique())
data.describe()
dataset=data[['Year','Selling_Price','Present_Price','Kms_Driven','Fuel_Type','Seller_Type','Transmissio
n','Owner']]
dataset.head()
dataset['Present_Year']=2020
dataset['Number_of_Years_Old']=dataset['Present_Year']- dataset['Year']
dataset.head()
dataset.drop(labels=['Year', 'Present_Year'],axis=1,inplace=True)
dataset.head()
#select categorical variables from then dataset, and then implement categorical encoding for nominal
variables
Fuel_Type=dataset[['Fuel_Type']]
Fuel_Type=pd.get_dummies(Fuel_Type, drop_first=True)

Seller_Type=dataset[['Seller_Type']]
Seller_Type=pd.get_dummies(Seller_Type, drop_first=True)

Transmission=dataset[['Transmission']]
Transmission=pd.get_dummies(Transmission, drop_first=True)

dataset=pd.concat([dataset,Fuel_Type, Seller_Type, Transmission], axis=1)

dataset.drop(labels=['Fuel_Type', 'Seller_Type', 'Transmission'], axis=1, inplace=True)

dataset.head()
dataset.columns
dataset.corr()
```

```
corrmat = data.corr()
top_corr_features = corrmat.index
plt.figure(figsize=(10,10))
#Plot heat map
sns.heatmap(data[top_corr_features].corr(),annot=True,cmap="RdYlGn")
sell=dataset['Selling_Price']
dataset.drop(['Selling_Price'], axis=1, inplace=True)
dataset=dataset.join(sell)
dataset.head()
X=dataset.iloc[:,:-1]
y=dataset.iloc[:,-1]
from sklearn.ensemble import ExtraTreesRegressor
model = ExtraTreesRegressor()
model.fit(X,y)

print(model.feature_importances_)

#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
from sklearn.model_selection import cross_val_score
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
from sklearn.tree import DecisionTreeRegressor
dt_reg = DecisionTreeRegressor(random_state = 0)
dt_reg.fit(X_train, y_train)
y_pred=dt_reg.predict(X_test)

print("Decision Tree Score on Training set is",dt_reg.score(X_train, y_train))#Training Accuracy
print("Decision Tree Score on Test Set is",dt_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(dt_reg, X_train, y_train, cv = 5)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)
```

```
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))

sns.distplot(y_test-y_pred)
plt.show()

plt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()
from sklearn.ensemble import RandomForestRegressor
rf_reg = RandomForestRegressor(n_estimators=400,min_samples_split=15,min_samples_leaf=2,
max_features='auto', max_depth=30)
rf_reg.fit(X_train, y_train)
y_pred=rf_reg.predict(X_test)

print("Random Forest Score on Training set is",rf_reg.score(X_train, y_train))#Training Accuracy
print("Random Forest Score on Test Set is",rf_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(rf_reg, X_train, y_train, cv = 5)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))

sns.distplot(y_test-y_pred)
plt.show()

plt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()
from sklearn.ensemble import VotingRegressor
vot_reg = VotingRegressor([('DecisionTree', dt_reg), ('RandomForestRegressor', rf_reg)])
vot_reg.fit(X_train, y_train)
y_pred=vot_reg.predict(X_test)
```

```
print("Voting Regresssor Score on Training set is",vot_reg.score(X_train, y_train))#Training Accuracy
print("Voting Regresssor Score on Test Set is",vot_reg.score(X_test, y_test))#Testing Accuracy

accuracies = cross_val_score(vot_reg, X_train, y_train, cv = 5)
print(accuracies)
print("Accuracy: {:.2f} %".format(accuracies.mean()*100))
print("Standard Deviation: {:.2f} %".format(accuracies.std()*100))

mae=mean_absolute_error(y_pred, y_test)
print("Mean Absolute Error:" , mae)

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)

print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

print('The r2_score is', metrics.r2_score(y_test, y_pred))

sns.distplot(y_test-y_pred)
plt.show()

plt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()
import pickle
pickle.dump(vot_reg, open("vot_reg.pkl", "wb"))

# load model from file
model = pickle.load(open("vot_reg.pkl", "rb"))

model.predict([[9.85, 6900, 0, 3, 0, 1, 0, 1]])
```

resaleintro.html :

```
<!DOCTYPE html>
<!-- Coding by CodingLab | www.codinglabweb.com  -->
<html lang="en">

<head>
  <meta charset="UTF-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
```

```
  <title>Website Image Slider</title>

  <!-- Link Swiper's CSS -->
  <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min.css"
integrity="sha384-
xOolHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
crossorigin="anonymous">

  <!-- CSS -->
  <!-- <link rel="stylesheet" href="style.css"> -->
  <style>
  /* Google Fonts - Poppins */
  @import
url("https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600&display=swap");

  * {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
    font-family: "Poppins", sans-serif;
  }
  .main {
    height: 100vh;
    width: 100%;
  }
  .wrapper,
  .slide {
    position: relative;
    width: 100%;
    height: 100%;
  }
  .slide {
    overflow: hidden;
  }
  .slide::before {
    content: "";
    position: absolute;
    height: 100%;
    width: 100%;
    background-color: rgba(0, 0, 0, 0.4);
    z-index: 10;
  }
  .slide .image {
    height: 100%;
    width: 100%;
    object-fit: cover;
```

```
}
.slide .image-data {
  position: absolute;
  top: 50%;
  left: 50%;
  transform: translate(-50%, -50%);
  text-align: center;
  width: 100%;
  z-index: 100;
}
.image-data span.text {
  font-size: 14px;
  font-weight: 400;
  color: #fff;
}
.image-data h2 {
  font-size: 45px;
  font-weight: 600;
  color: #fff;
}
a.button {
  display: inline-block;
  padding: 10px 20px;
  border-radius: 25px;
  color: #333;
  background: #fff;
  text-decoration: none;
  margin-top: 25px;
  transition: all 0.3s ease;
}
a.button:hover {
  color: #fff;
  background-color: #E83A14;
}

/* swiper button css */
.nav-btn {
  height: 50px;
  width: 50px;
  border-radius: 50%;
  background: rgba(255, 255, 255, 0.3);
}
.nav-btn:hover {
  background: rgba(255, 255, 255, 0.4);
}
.swiper-button-next {
```

```
    right: 50px;
  }
  .swiper-button-prev {
    left: 50px;
  }
  .nav-btn::before,
  .nav-btn::after {
    font-size: 25px;
    color: #fff;
  }
  .swiper-pagination-bullet {
    opacity: 1;
    height: 12px;
    width: 12px;
    background-color: #fff;
    visibility: hidden;
  }
  .swiper-pagination-bullet-active {
    border: 2px solid #fff;
    background-color: #c87e4f;
  }

  @media screen and (max-width: 768px) {
    .nav-btn {
      visibility: hidden;
    }
    .swiper-pagination-bullet {
      visibility: visible;
    }
  }

  .profile{
    margin-left:950px;
  }

  ion-icon{
    color: #E83A14;
  }

  </style>
</head>

<body>
  <nav class="navbar navbar-expand-lg navbar-dark bg-dark fixed-top">
    <a class="navbar-brand" href="#">Navbar</a>
```

```
    <button class="navbar-toggler" type="button" data-toggle="collapse" data-
target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-expanded="false" aria-
label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
      <div class="navbar-nav">
        <a class="nav-link active" href="#">Home <span class="sr-only">(current)</span></a>
        <a class="nav-link" href="#">About Us</a>
        <a class="nav-link" href="#">Contat Us</a>
        <a class="profile nav-link" href="#">Profile</a>
        <ion-icon name="people-outline" size="large"></ion-icon>
      </div>
    </div>
  </nav>
  <section class="main swiper mySwiper">
    <div class="wrapper swiper-wrapper">
      <div class="slide swiper-slide">
        <img src="https://wallpaperaccess.com/full/1838837.jpg" alt="" class="image" />
        <div class="image-data">
          <span class="text" style="color:#E83A14;">We predict the best price for your car.</span>
          <h2>
            What do you wanna do?<br />
          </h2>
          <a href="resalepredict.html" onclick="predict()" class="button">Let's start Predicting..</a>
          <!-- <form action="/predict/" method="post"> -->
            <!-- <button class="button"type="submit" onclick="predict()">Let's start predicting..</button> -
->
          <!-- </form> -->
        </div>
      </div>
    </div>
  </section>

  <!-- Swiper JS -->
  <!-- <script src="swiper-bundle.min.js"></script> -->

  <!-- Initialize Swiper -->
  <!-- <script>
      var swiper = new Swiper(".mySwiper", {
        slidesPerView: 1,
        loop: true,
        pagination: {
          el: ".swiper-pagination",
          clickable: true,
        },
```

```
        navigation: {
          nextEl: ".swiper-button-next",
          prevEl: ".swiper-button-prev",
        },
      });
    </script> -->
  <script type="module"
src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.esm.js"></script>
  <script nomodule src="https://unpkg.com/ionicons@5.5.2/dist/ionicons/ionicons.js"></script>
</body>

</html>
```

resalePredict.html :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Car Price Prediction</title>


    <!-- BootStrap -->
    <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/css/bootstrap.min.css"
        integrity="sha384-
9aIt2nRpC12Uk9gS9baDl411NQApFmC26EwAOH8WgZl5MYYxFfc+NcPb1dKGj7Sk"
crossorigin="anonymous">


    <!-- css -->
    <!-- <link rel="stylesheet" href="/static/css/styles.css"> -->
<style>
  body {
    background-image:
url(https://wallpaperaccess.com/full/13647.jphttps://wallpaperaccess.com/full/3540065.jpgg);
    background-size: cover;
    text-align: center;
}

.navbar {
    background-color: #8bbde8;
}
```

```
a {
    color: #c73f0a;
}

a:hover {
    color: #0e536e;
  }
</style>

</head>

<body>

    <!-- As a heading -->
    <nav class="navbar navbar-inverse navbar-fixed-top">
        <div class="container-fluid">
            <div class="navbar-header">

        <br>
            </div>
        </div>
    </nav>

    <br>


    <div class="container">
<a2>Please fill the parameters below and click on Selling Price button to check car price</a2>


        <form action="/predict" method="post">

            <h3>Year</h3><input id="first" name="Year" placeholder="eg. like the year '2010'
"type="number " >
        <br>
        <br>
            <h3>Showroom Price(lakhs)</h3><input id="second" name="Present_Price" placeholder="eg.
'10.45' lakhs"required="required">
        <br>
        <br>
            <h3>Kilometers Driven</h3><input id="third" name="Kms_Driven" placeholder="eg. 10000
Km driven before"required="required">
        <br>
        <br>
```

```
        <h3>Previous Owners</h3><input id="fourth" name="Owner" placeholder="0 or 1 or 2
Previous Owners" required="required">
      <br>
      <br>

        <h3>Fuel Type</h3><select name="Fuel_Type_Petrol" id="fuel" required="required">
          <option value="Petrol">Petrol</option>
          <option value="Diesel">Diesel</option>
          <option value="Diesel">CNG</option>
        </select>
      <br>
      <br>
        <h3>Dealer or Individual</h3><select name="Seller_Type_Individual" id="resea"
required="required">
          <option value="Dealer">Dealer</option>
          <option value="Individual">Individual</option>
        </select>
      <br>
      <br>
        <h3>Transmission Type</h3><select name="Transmission_Mannual" id="research"
required="required">
          <option value="Mannual">Manual Car</option>
          <option value="Automatic">Automatic Car</option>
        </select>

        <br><br><button id="sub" type="submit ">Selling Price</button>
      </form>
      <h3>{{ prediction_texts }}</h3>
      <br>
    </div>




    <!-- JavaScript -->
    <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
      integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXaRkfj"
      crossorigin="anonymous"></script>
    <script src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
      integrity="sha384-
Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
      crossorigin="anonymous"></script>
    <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.5.0/js/bootstrap.min.js"
```

```
        integrity="sha384-
OgVRvuATP1z7JjHLkuOU7Xw704+h835Lr+6QL9UvYjZE3Ipu6Tp75j7Bh/kR0JKI"
        crossorigin="anonymous"></script>
    <script src="https://kit.fontawesome.com/5f3f547070.js" crossorigin="anonymous"></script>

   <!-- Footer -->

</body>

</html>
```

App.py :

```
import pickle

import joblib
import numpy as np
import sklearn
from flask import Flask, render_template, request
from sklearn.preprocessing import StandardScaler

app = Flask(__name__)
model = pickle.load(open("vot_reg.pkl", "rb"))
@app.route('/',methods=['GET'])
def Home():
    return render_template('resalepredict.html')

# @app.route('/',methods=['GET'])
# def predictpage():
#     return render_template('resalepredict.html')


standard_to = StandardScaler()
@app.route("/predict", methods=['POST'])
def predict():

    Fuel_Type_Diesel=0

    if request.method == 'POST':
        Year = int(request.form['year'])
        Present_Price=float(request.form['Present_Price'])
        Kms_Driven=int(request.form['Kms_Driven'])
        Owner=int(request.form['Owner'])
        Fuel_Type_Petrol=request.form['fueltype']
```

```
        if(Fuel_Type_Petrol=='petrol'):
                Fuel_Type_Petrol=1
                Fuel_Type_Diesel=0
        else:
            Fuel_Type_Petrol=0
            Fuel_Type_Diesel=1
        Year=2020-Year
        Seller_Type_Individual=request.form['Seller_Type_Individual']
        if(Seller_Type_Individual=='Individual'):
            Seller_Type_Individual=1
        else:
            Seller_Type_Individual=0
        Transmission_Mannual=request.form['Transmission_Mannual']
        if(Transmission_Mannual=='Mannual'):
            Transmission_Mannual=1
        else:
            Transmission_Mannual=0
        prediction=model.predict(np.array([[Year,
                                    Present_Price,
                                    Kms_Driven,
                                    Owner,
                                    Fuel_Type_Diesel,
                                    Fuel_Type_Petrol,
                                    Seller_Type_Individual,
                                    Transmission_Mannual]]))
        output=round(prediction[0],2)
        if output<0:
            return render_template('resaleresult.html',prediction_texts="Sorry you cannot sell this car")
        else:
            return render_template('resaleresult.html',prediction_texts="You can sell the Car at {}
lakhs".format(output))
    else:
        return render_template('resaleresult.html')
if __name__=="__main__":
    app.run(debug=True)
```

8. TESTING :

8.1 HOME PAGE TESTING :

Test Case ID	Test Scenario	Test Case Description	Test Inputs	Expected Output	Positive Result	Negative Result
CRVP_TC_01	Site link	To check whether the site link is open or not	Tap the site link	Visiblity of the website	When the website was opened after tap the site means the result as home page	When the website was not opened or error acquiring like 402,505,etc..,
CRVP_TC_02	Prediction button in home page	To check whether the prediction is working or not	Tap the button to open the prediction page	Move to the prediction form	Successfully open the prediction form	When the button was tapped by user error acquiring the frontend code connectivity code or button was not enabling , **Your file couldn't be accessed**

8.2 PREDICTION PAGE TESTING :

Test Case ID	Test Scenario	Test Case Description	Test Inputs	Expected Output	Results
CRVP_TC_01	Show room released year	To check whether the year was in the given limit	Input type Limited range of year	Prediction credational is valid / Not valid	Test case will be pass / fail
CRVP_TC_02	What is the Showroom Price?(In lakhs)	To check whether the Showroom Price? Is (In lakhs)	Car rate in lakh	Prediction credational is valid / Not valid	Test case will be pass / fail
CRVP_TC_03	How Many Kilometers Drived?	To check whether the kilometers were in given input input format	Limited Kilometer as a input	Prediction credational is valid / Not valid	Test case will be pass / fail
CRVP_TC_04	How much owners previously	To check whether the owner	Input type as number in given range	Prediction credational is valid / Not	Test case will be pass / fail

9. RESULTS :

```
[ ]  1 import pickle
     2 pickle.dump(vot_reg, open("vot_reg.pkl", "wb"))
     3
     4 # load model from file
     5 model = pickle.load(open("vot_reg.pkl", "rb"))
     6
     7 #model.predict([[Year,Present_Price,Kms_Driven,Owner,Fuel_Type_Petrol,Fuel_Type_Diesel,Seller_Type,Transmission]])
     8
     9 new_val = model.predict([[10, 380000.00, 252300, 2, 1, 0, 0, 1]])
    10 print("The resale value will be around {} lakhs.".format(np.round(new_val,2)))
```

```
The resale value will be around [7.5] lakhs.
```

```
Voting Regresssor Score on Training set is 0.977609234660353
Voting Regresssor Score on Test Set is 0.9018209429652577
[0.9696831  0.83755812 0.68204825 0.90301588 0.94145951]
Accuracy: 86.68 %
Standard Deviation: 10.24 %
Mean Absolute Error: 0.8177898930707259
Mean Squared Error: 2.93614844723144
RMSE: 1.713519316270301
The r2_score is 0.9018209429652577
```

11. ADVANTAGES :

The data needed for the price estimation of the used cars is more Compared to the existing system. Other than the variant , model and brand of the car , the mileage and Service records of the car is also needed for estimating price for greater accuracy . In the older or existing systems, the owners of the cars code the selling price of the cars which is favorable to the owner’s hand. The satisfaction of the both seller and buyer are more and there is no brokerage and brokers involved as an external mediator. User friendly, more reliable ,easily understandable , high accuracy in estimation of the price of used cars. There is less need for a mechanic to give a price estimate of the car after inspection. The more reasonable and best resale price is estimated through the newly proposed system which fulfills both the seller’s and buyer’s satisfaction.

DISADVANTAGES :

The data needed for the price estimation of the used cars is less in the existing system. Only variant , model, brand and the model year of the car and the estimated price of the car given by the seller are the only information given in most of the existing systems. In the existing systems, the owners of the cars code the selling price of the cars which is favorable to the owner's hand. The satisfaction of the buyer and the seller is less and the estimated price isn't reasonable. And the brokerage and brokers are involved in the existing systems. In the existing system, mechanics are needed to give the final estimate of the cars after the inspection. Mileage and horsepower are neglected in the price estimation data.

MODULE DESCRIPTION

Data Pre-Processing:

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So for this, the user can use data pre-processing task

Training:

A training model is a dataset that is used to train an ML algorithm. It consists of the sample output data and the corresponding sets of input data that have an influence on the output. The training model is used to run the input data through the algorithm to correlate the processed output against the sample output. The result from this correlation is used to modify the model.

Testing:

In machine learning, model testing is referred to as the process where the performance of a fully trained model is evaluated on a testing set. This kind of ML testing is more similar to traditional testing. Users can write and run tests checking the performance of the program. Applying the tests, users catch bugs in different components of the ML program. For example, users can test that the hidden layers

in a neural network are configured correctly.

Random forest Regression :

Random Forest Regression is a supervised learning algorithm that uses ensemble learning methods for regression. Ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model.

Prediction:

"Prediction" refers to the output of an algorithm .It has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome. Just like a hypothesis, a prediction is a type of guess. However, a prediction is an estimation made from observations.

CONCLUSION AND SCOPE FOR FUTURE WORK

The increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction . Car Price Prediction was aimed to get different perspectives and eventually compared their performance with different models. Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data. In this research, linear regression and lasso regression , Random forest regression were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms. Data cleaning is one of the processes that increases prediction performance.