ANNA UNIVERSITY

COLLEGE OF ENGINEERING GUINDY

PROJECT TITLE: CAR RESALE VALUE PREDICTION

DOMAIN: APPLIED DATA SCIENCE

TEAM ID: PNT2022TMID35419

PROJECT REPORT

SUBMITTED BY

NAPA SAI SANDEEP	–	**2019105069**
SHAIK ANWAR BASHA	–	**2019105602**
PENGANI VIKKI	–	**2019105037**
KADAGOTI PRUDHVI VARA PRASAD RAO	–	**2019105543**

OF

BACHELOR OF ENGINEERING

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

1. **INTRODUCTION**
    1.1 Project Overview
    1.2 Purpose
2. **LITERATURE SURVEY**
    2.1 Existing problem
    2.2 References
    2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
    3.1 Empathy Map Canvas
    3.2 Ideation & Brainstorming
    3.3 Proposed Solution
    3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
    4.1 Functional requirement
    4.2 Non-Functional requirements
5. **PROJECT DESIGN**
    5.1 Data Flow Diagrams
    5.2 Solution & Technical Architecture
    5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
    6.1 Sprint Planning & Estimation
    6.2 Sprint Delivery Schedule
    6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
    7.1 Feature 1
    7.2 Feature 2
    7.3 Database Schema (if Applicable)
8. **TESTING**
    8.1 Test Cases
    8.2 User Acceptance Testing
9. **RESULTS**
    9.1 Performance Metrics

10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**

Source Code

GitHub & Project Demo Link

1. INTRODUCTION:

1.1 Project Overview:

Determining whether the listed price of a used car is a challenging task, due to the many factors that drive a used vehicle's price on the market. The focus of this project is developing machine learning models that can accurately predict the price of a used car based on its features, in order to make informed purchases. We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models . We will compare the performance of various machine learning

algorithms like Linear Regression, Ridge Regression, Lasso Regression, Elastic Net, Decision Tree Regressor and choose the best out of it. Depending on various parameters we will determine the price of the car. Regression Algorithms are used because they provide us with continuous value as an output and not a categorized value because of which it will be possible to predict the actual price a car rather than the price range of a car. User Interface has also been developed which acquires input from any user and displays the Price of a car according to user's inputs.

1.2 Purpose:

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 predictipon system which effectively determines the worthiness of the car using a variety of features.Car Resale Value Prediction helps the user to predict the resale value of the car depending upon the various features like kilometers driven,fuel type etc.,This resale value prediction system is made for general purpose to just predict the amount that can be roughly acquired by the user.

2. LITERATURE SURVEY:

2.1 Existing Problem

'Car Price Prediction Using Machine Learning Techniques' according to authors Enis Gegic, Becir Isakovic, Dino Keco, Zerina Masetic, Jasmin Kevric, in this paper they mainly concentrate on collecting various data from web portal by using web scrap techniques. And those have been compared with the help of different machine learning algorithms to predict the vehicle price in easy manner. They classified the price according to different ranges of price that is already given. Artificial neural network, support vector machine, random forest algorithms were used on different datasets to build classifiers model. In the existing system, to predict the price of vehicles both two wheelers and four wheelers, a lot of data mining algorithms and machine learning algorithms were widely used. The major drawback of this existing system is they need more attributes in order to predict the

vehicle price. More comparison techniques must be used to get the result more efficiently. It is highly complicated to get sufficient data sets that were spread widely all over the world. The datasets can be collected only through online. But not on the offline mode. It is not possible for everyone to collect the data sets through online mode particularly in rural areas. The data sets will not have about the vehicles which were not used for long time and also the traditional model vehicles may or may not be included in the data sets.

'Predicting the Price of Used Cars Using Machine Learning Techniques' according to author Sameerchand Pudaruth they have done the predictions of car price from the historical data that has been collected from daily newspapers. For predicting the price of vehicles, they 4 have used supervised machine learning techniques. Other algorithms were also used to predict such as multiple linear regression, some decision tree algorithms. All these algorithms were compared and found the best algorithm for prediction. They have faced some difficulties in comparing the algorithms, somehow they have managed. The major drawbacks of existing system are the system is very slow due to most of the works about the keyword query just analyze individual points, and they are inappropriate to many applications that call for analysis of groups of different vehicle points. In the existing system shown above, authors proposed prediction model based on the single machine learning algorithm. However, it is noticeable that single machine learning algorithm approach did not give remarkable prediction results and could be enhanced by assembling various machine learning methods in an ensemble. In future, gaining advancement in this system model we could rely on this to predict the value. Generally, e-commerce platforms attract customers in different ways to use their system for buying or selling and the algorithm which is used in their system is such that the value is not accurate It is overpriced when a customer wants to buy a car and vice versa while selling. Also referring to the tremendous loss reported of worth more than a billion dollars in Germany due to mis-calculation of the car value which could be overcome using this prediction application. Further, 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 for better user

interaction and for better performance, we plan to judiciously design deep learning network structures, useadaptive learning rates and train on clusters of data rather than the whole dataset.

2.2 References:

Car Price Prediction Using Machine Learning Techniques

Ref- TEM Journal. Volume 8, Issue 1, Pages 113-118, ISSN 2217- 8309, DOI:10.18421/TEM81-16, February 2019.

Predicting the Price of Used Cars Using Machine Learning Techniques

Ref: International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 4, Number 7 (2014).

2.3 Problem statement definition:

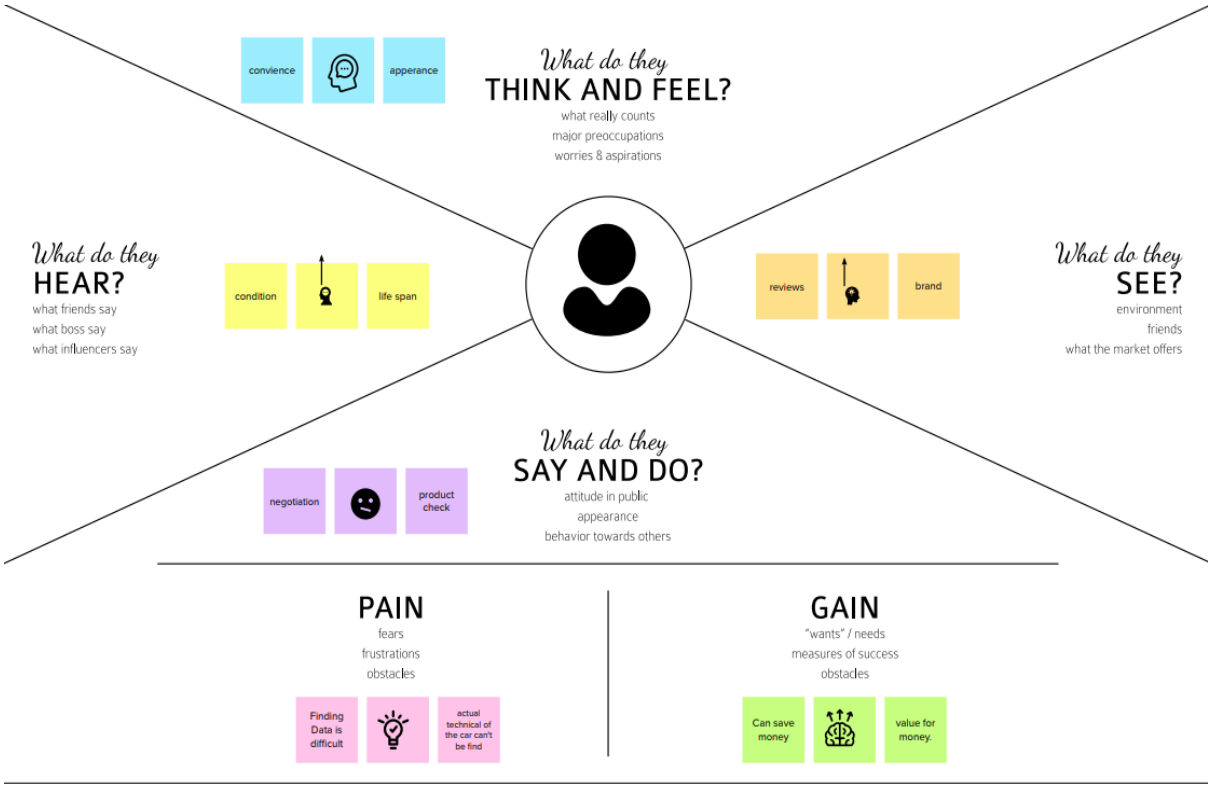
Machine learning has become a tool used in almost every task that requires estimation.We need to build a model to estimate the price of used cars. The model should take car related parameters and gives output as a selling price. The selling price of a used car depends on certain features as Fuel type, Manufacturing year,Miles driven,Number of owners .This is a supervised learning problem and can be solved using regression techniques.We need to predict the selling price of a car based on the given car's features.Regression problems require labelled data where our target or dependent variable is the selling price of a car.All other features are independent variables. Some regression algorithms are linear regression,Decision tree Regressor, Random Forest Regressor etc.,

3. IDEATION AND PROPOSED SOLUTION :

3.1Empathy Map Canvas:

An empathy map is a simple,easy to digest visual that captures knowledge about a users behaviours and attitude.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 users perspective along with his or her goals and challenges.

Empathy map for Car Resale value prediction:



3.2 Ideation and 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.

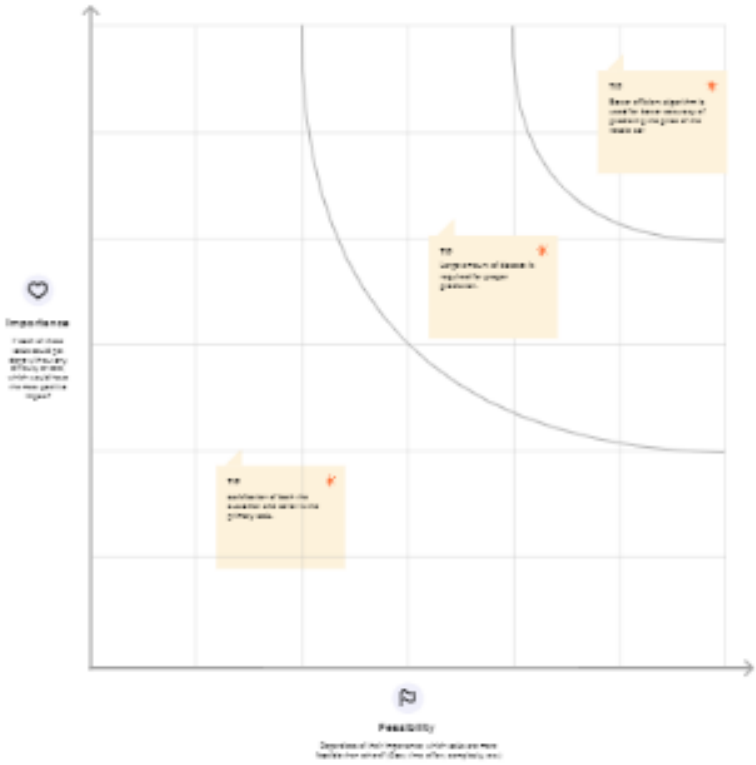
Brainstorming and idea prioritization Template for Car Resale Value Prediction

Brainstorm

Group Ideas

Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on the grid to determine which ideas are important and which are feasible.



After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

Keep moving forward

3.3 Proposed Solution:

S.No.	Parameter	Description
1.	Problem Statement (Problem to besolved)	Currently, if anyone wants to sell their car, they have to take their car to a respective company workshop or have a to make an appointment for the company to get an estimate of the price. This process involves of lot of time and resources.
2.	Idea / Solution description	Especially for the first timers, a used purchase ismore practical an d affordable at the same time. Unless you really want the latest car in the market or that new car smell is all you are looking for, a used car can very well cater to almost alltypes of buyers quite conveniently
3.	Novelty / Uniqueness	Looks Matterfor A Better Car ResaleValue. A Service Ensures Good Car Resale Value. Keep All Papers in Place. Novelty is car resale Get PhoneNumbers, Address, Photos, Maps of Novelty Tata.
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 and speed.

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 hugecapital to invest. Dealer license.

		Location of thebusiness. Keep a watch on the market.Make yourcatalog. Use a perfect marketing strategy.
6 .	Scalability of the Solution	The size of the used car market in India wasover 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

The individual who has to sell a car	when the number of prediction parameters grows. The prediction's accuracy decreases	use various machine learning algorithms to predict the car price more accuratly based on the details of the car.

Identify the cost of car based on the specifications and suggest the accurate price of the car based on the details shared	The Individual user does not have a well knowledge about the car to estimate the car price on their own.So they need a mechanic or a person who is having well knowledge about car to analyse the details and correct estimation about the car	Evaluate the advantages and utilisation. It must be necessary to have complete knowledge of the various machine learning techniques and implementations. Model needs to have good training.
The seller is impressed and feels the platform as user-friendly when they need money by selling the car. Individual car sellers were unable to determine the cost of the vehicle. It results in financial loss. and displeased the user. Before: Stress to determine the price After: Feels peaceful and relax	obtain information from the user and utilise as a dataset. applying different machine learning algorithms to the dataset and providing the accurate automobile price forecast based on the dataset's details.	The seller will be in their home itself find the price predictions by sharing their car details

4.REQUIREMENT ANALYSIS:

4.1 Functional Requirement:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story/ Sub-Task)
FR-1	User Registration	Registration through Form Registration throughGmail Registrationthrough LinkedIN
FR-2	User Confirmation	Confirmation viaEmail 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

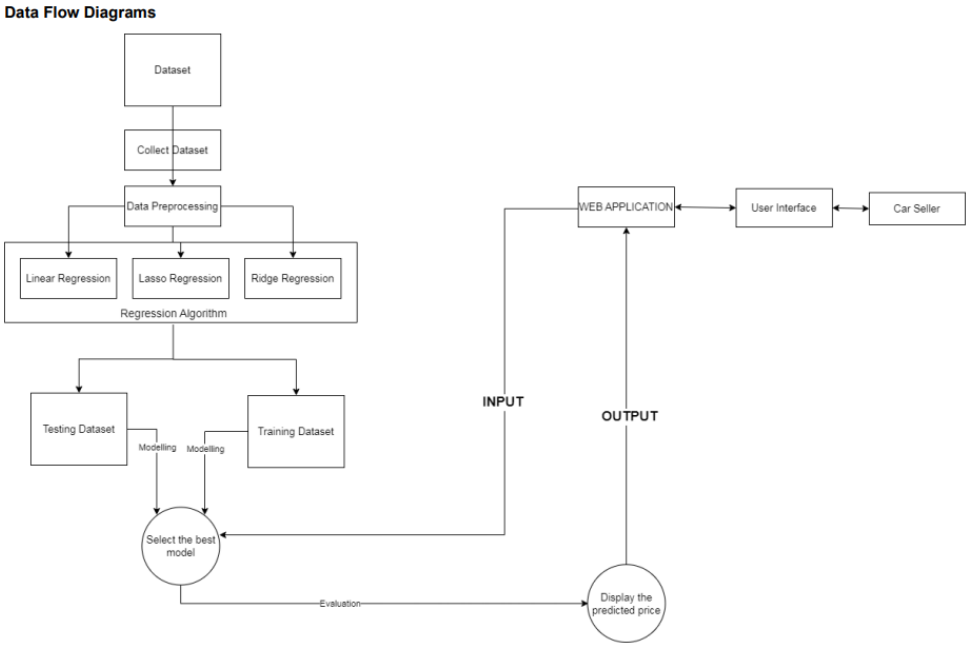
4.2 Non Functional Requirement:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Great UI(User Interface), Accuracy in value prediction
NFR-2	**Security**	Protect user password and Personal details
NFR-3	**Reliability**	Rate of occurrence of failure is less, Failure free.
NFR-4	**Performance**	Perform correct prediction value, The landing page support several users and mustprovide 5 secondsor less response time
NFR-5	**Availability**	Uninterrupted services mustbe available in all time except the time of server updation
NFR-6	**Scalability**	that can handle any amount of data and perform many computations in a cost-effective and timesaving wayto instantly servemillions of usersresiding at globallocations

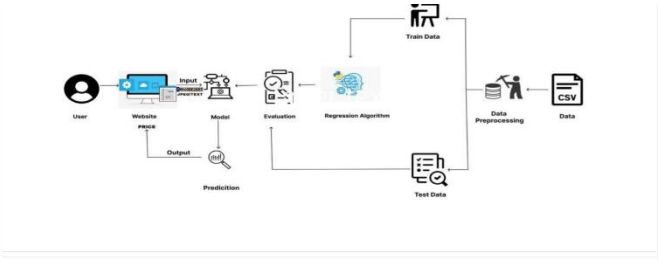
5. PROJECT DESIGN:

5.1Data flow diagram

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.



5.2 Solution and Technical Architecture:



5.3 User Stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
		USN-2	As a user, I will receive confirmation email once I 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
Customer (Web user)	Dashboard	USN-6	As a user, I can give my car details by giving car brand,registration year,model,varient,state of registration,number of owners,kilometre driven.		High	Sprint-1
	Dashboard	USN-7	As a user, I can able to view all the car model to pick my car.		High	Sprint-1
	Dashboard	USN-8	As a user, I can get my car specification using the car registration number		High	Sprint-1
	Dashboard	USN-9	As a user, I manually modify the car health and damages, kilometres driven,odometer,mileage and relevant added accessories		High	Sprint-1
Administrator			As a administrator, I can add the car dataset.		High	Sprint-1

6. PROJECT PLANNING AND SCHEDULING:

6.1Sprint Planning and Estimation

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can enter into the website with the help of the Google chrome browser in Windows	2	High	4
Sprint-1	Registration	USN-2	As a user, I can enter into the website through browser in Android	1	High	4
Sprint-1	Registration	USN-3	As a user, I can enter into the website through browser in ios	2	Medium	4
Sprint-1	Login	USN-4	As a user, I can find the car resale value prediction page in the website	1	High	4
Sprint-2	Home Page	USN-5	As a user, I need to select the parameters like Year, Showroom price, Kilometres driven, fuel type etc and click on the submit button	2	High	4
Sprint-3	Home Page	USN-6	As a user, I can see the accurate price for car resale after entering the details.	2	High	4
Sprint-4	Home Page	USN-7	As a user, If I done a mistake while providing the details , I can reset the details and click the submit button.	1	Low	4

6.2 Sprint Delivery and Schedule

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

6.3

Reports from JIRA

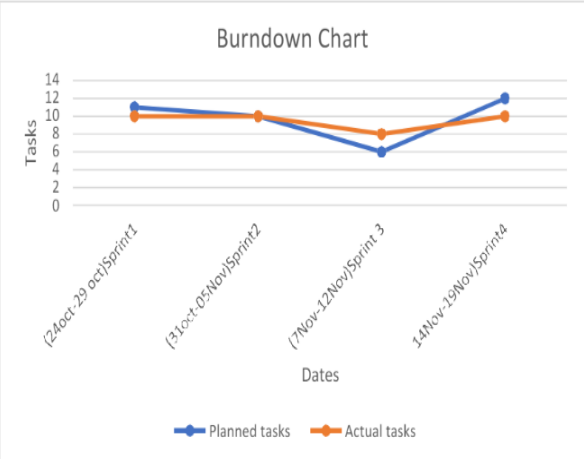
Velocity:

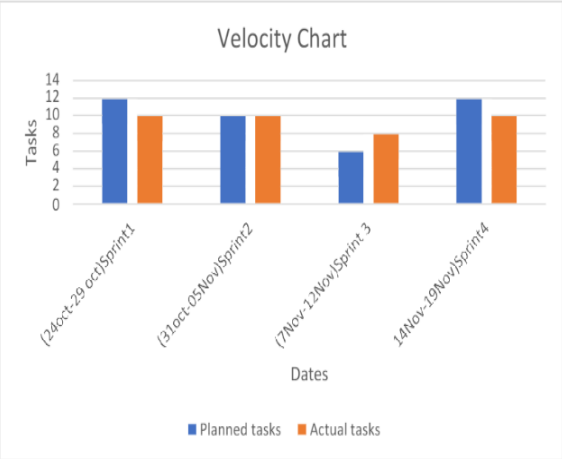
Imagine we have a 10-day sprint duration , and the velocity of the team is 20 (point per sprint). Lets calculate the teams average velocity(AV) per iteration unit(story points per day).

AV = Sprint duration/velocity = 20/10 =2

Burndown Chart:

A burndown chart is a graphical representation of work left to do versus time. It is often used in agile software development methodologies such as Scrum. However, burndown chart can be applied to any project containing measurable progress over time.





7.1CODING AND SOLUTIONING:

7.1Analysis of Algorithm:

There are several algorithms a person can use to predict a value as every algorithm itself has different significance and its own equation to predict. Algorithms like linear regression, Lasso regression, Ridge regression, Random forest regression, XGBoost, etc. have different equation through which we get a predicted value using past data.

Random Forest Regression:

We will see steps that has helped us to build this model for car value prediction; Firstly, we have import dataset of different cars and their year, kms

driven, fuel type, transmission, etc. These are the attributes which we are going to need to predict. Then we will be using feature engineering where we will correlate between attributes in our dataset.

Dataset Description:

Dataset was collected from the website called Kaggle.Kaggle allows users to find and publish datasets,explore and build in a web based data science environment,work with other data science environment,work with other data scientists and machine learning engineers.The data set includes car name,year,selling price,present price,kms driven,fuel type,seller type,Transmission type and owner.

Dataset:

Data Pre-Processing:

Data pre-processing 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.While creating a machine learning project,it is not always a case that we come across the clean and formated data and while doing a any operation with data,it is mandatory to clean it and put in a formatted way.So for this,we use data preprocessing task.

Explanation of code:

Importing Dataset:

The data consists of some rows and columns.Since our target is to find the selling price,the target attribute y is also selling price,remaining features are taken for analysis and predictions.

```
In [1]: import pandas as pd

In [2]: df=pd.read_csv('Car Dataset.csv')

In [3]: df.shape

Out[3]: (301, 9)
```

```
In [6]: df.describe()
```

Out[6]:

	Year	Selling_Price	Present_Price	Kms_Driven	Owner
count	301.000000	301.000000	301.000000	301.000000	301.000000
mean	2013.627907	4.661296	7.628472	36947.205980	0.043189
std	2.891554	5.082812	8.644115	38886.883882	0.247915
min	2003.000000	0.100000	0.320000	500.000000	0.000000
25%	2012.000000	0.900000	1.200000	15000.000000	0.000000
50%	2014.000000	3.600000	6.400000	32000.000000	0.000000
75%	2016.000000	6.000000	9.900000	48767.000000	0.000000
max	2018.000000	35.000000	92.600000	500000.000000	3.000000

Correlation:

```
In [20]: final_dataset.corr()
```

Out[20]:

	Selling_Price	Present_Price	Kms_Driven	Owner	no_year	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
Selling_Price	1.000000	0.878983	0.029187	-0.088344	-0.236141	0.552339	-0.540571	-0.550724	-0.367128
Present_Price	0.878983	1.000000	0.203647	0.008057	0.047584	0.473306	-0.465244	-0.512030	-0.348715
Kms_Driven	0.029187	0.203647	1.000000	0.089216	0.524342	0.172515	-0.172874	-0.101419	-0.162510
Owner	-0.088344	0.008057	0.089216	1.000000	0.182104	-0.053469	0.055687	0.124269	-0.050316
no_year	-0.236141	0.047584	0.524342	0.182104	1.000000	-0.064315	0.059959	0.039896	-0.000394
uel_Type_Diesel	0.552339	0.473306	0.172515	-0.053469	-0.064315	1.000000	-0.979648	-0.350467	-0.098643
uel_Type_Petrol	-0.540571	-0.465244	-0.172874	0.055687	0.059959	-0.979648	1.000000	0.358321	0.091013
_Type_Individual	-0.550724	-0.512030	-0.101419	0.124269	0.039896	-0.350467	0.358321	1.000000	0.063240
mission_Manual	-0.367128	-0.348715	-0.162510	-0.050316	-0.000394	-0.098643	0.091013	0.063240	1.000000

Now we are slicing the data into training and test set.

```
In [26]: ##independent and dependent features
         X=final_dataset.iloc[:,1:]
         y=final_dataset.iloc[:,0]
```

Elimination of unwanted features:

The Extra Trees Regressor library allows you to view feature importance and thereby remove the less important features from the data.It is always advised to remove the unnecessary feature because they can definitely yield better accuracy scores.

```
In [29]: ### Feature Importance

         from sklearn.ensemble import ExtraTreesRegressor
         model = ExtraTreesRegressor()
         model.fit(X,y)
```

```
Out[29]: ExtraTreesRegressor()
```

```
In [30]: print(model.feature_importances_)
```

```
[0.37357135 0.04555725 0.00107922 0.07372164 0.23153707 0.01240199
 0.12378753 0.13834395]
```

Training and Test Split:

The train-test split used to estimate the performance of machin learning algorithms that are applicable for prediction based algorithms.This method is a fast and easy procedure to perform such that we can compare our own machine learning model results to machine results.

```
In [32]: 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, random_state=0)
```

Training the model:

We have used the random forest Regressor to predict the selling prices since this is a regression problem and that random forest uses multiple decision trees and has shown good results for my model.

In [33]:
```
from sklearn.ensemble import RandomForestRegressor
rf_random=RandomForestRegressor()
```

In [34]:
```
##Hyperparameters

import numpy as np
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
print(n_estimators)
```

```
[100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200]
```

Hyper Parameter Optimization:

In [35]:
```
#Randomized Search CV

# Number of trees in random forest
n_estimators = [int(x) for x in np.linspace(start = 100, stop = 1200, num = 12)]
# Number of features to consider at every split
max_features = ['auto', 'sqrt']
# Maximum number of levels in tree
max_depth = [int(x) for x in np.linspace(5, 30, num = 6)]
# max_depth.append(None)
# Minimum number of samples required to split a node
min_samples_split = [2, 5, 10, 15, 100]
# Minimum number of samples required at each leaf node
min_samples_leaf = [1, 2, 5, 10]
```

In [36]:
```
from sklearn.model_selection import RandomizedSearchCV
```

In [37]:
```
random_grid = {'n_estimators': n_estimators,
               'max_features': max_features,
               'max_depth': max_depth,
               'min_samples_split': min_samples_split,
               'min_samples_leaf': min_samples_leaf}

print(random_grid)
```

```
{'n_estimators': [100, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1100, 1200], 'max_features': ['auto', 'sqrt'], 'max_dept
h': [5, 10, 15, 20, 25, 30], 'min_samples_split': [2, 5, 10, 15, 100], 'min_samples_leaf': [1, 2, 5, 10]}
```

```
In [40]: rf_random.fit(X_train,y_train)

Fitting 5 folds for each of 10 candidates, totalling 50 fits
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   0.9s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   0.8s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   0.8s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   0.8s
[CV] END max_depth=10, max_features=sqrt, min_samples_leaf=5, min_samples_split=5, n_estimators=900; total time=   0.8s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   1.1s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   0.9s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   1.1s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   0.9s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=1100; total time=   1.0s
[CV] END max_depth=15, max_features=auto, min_samples_leaf=5, min_samples_split=100, n_estimators=300; total time=   0.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=15, max_features=sqrt, min_samples_leaf=1, min_samples_split=15, n_estimators=300; total time=   0.2s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   0.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   0.7s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   0.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   0.6s
[CV] END max_depth=5, max_features=sqrt, min_samples_leaf=2, min_samples_split=10, n_estimators=700; total time=   0.6s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   0.7s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   0.7s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   0.7s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   0.7s
[CV] END max_depth=20, max_features=auto, min_samples_leaf=1, min_samples_split=15, n_estimators=700; total time=   0.6s

Out[40]: RandomizedSearchCV(cv=5, estimator=RandomForestRegressor(), n_jobs=1,
                            param_distributions={'max_depth': [5, 10, 15, 20, 25, 30],
                                                 'max_features': ['auto', 'sqrt'],
                                                 'min_samples_leaf': [1, 2, 5, 10],
                                                 'min_samples_split': [2, 5, 10, 15,
                                                                       100],
                                                 'n_estimators': [100, 200, 300, 400,
                                                                  500, 600, 700, 800,
                                                                  900, 1000, 1100,
                                                                  1200]},
                            random_state=42, scoring='neg_mean_squared_error',
                            verbose=2)
```

Now we are finally use the model to predict the test dataset.

```
In [45]: predictions=rf_random.predict(X_test)

In [46]: predictions

Out[46]: array([ 7.02401874,  0.52168574,  4.93711392,  8.37186064, 12.32479143,
                 5.23432801,  3.36838423,  0.42969473,  3.93824504,  5.05868389,
                 2.85141883,  0.66126344,  5.06808233,  7.2587716 ,  7.45821224,
                12.63274799,  6.98157088,  4.1717805 ,  0.48428483,  1.30795166,
                 3.25008974,  5.1754389 ,  5.43804929, 10.44034459,  0.23723337,
                 0.6824487 ,  0.32796592,  0.68567623,  0.51177523,  4.85071808,
                 2.84296786,  5.88991838,  0.51368733,  7.11751535,  3.27218103,
                 1.14759745,  5.73088704,  5.50097307,  0.25750374,  7.76900176,
                 7.5946165 , 22.0146006 ,  5.06754404,  4.51565913,  5.60947326,
                10.30841657,  0.25539516,  0.75706951,  5.44377104,  6.76426193,
                 6.83967661,  2.98452932,  5.34145427, 22.01674346,  1.14759745,
                 1.14669807,  0.42096162,  2.75978594,  3.67714684,  2.53817549,
                 4.63683139])
```

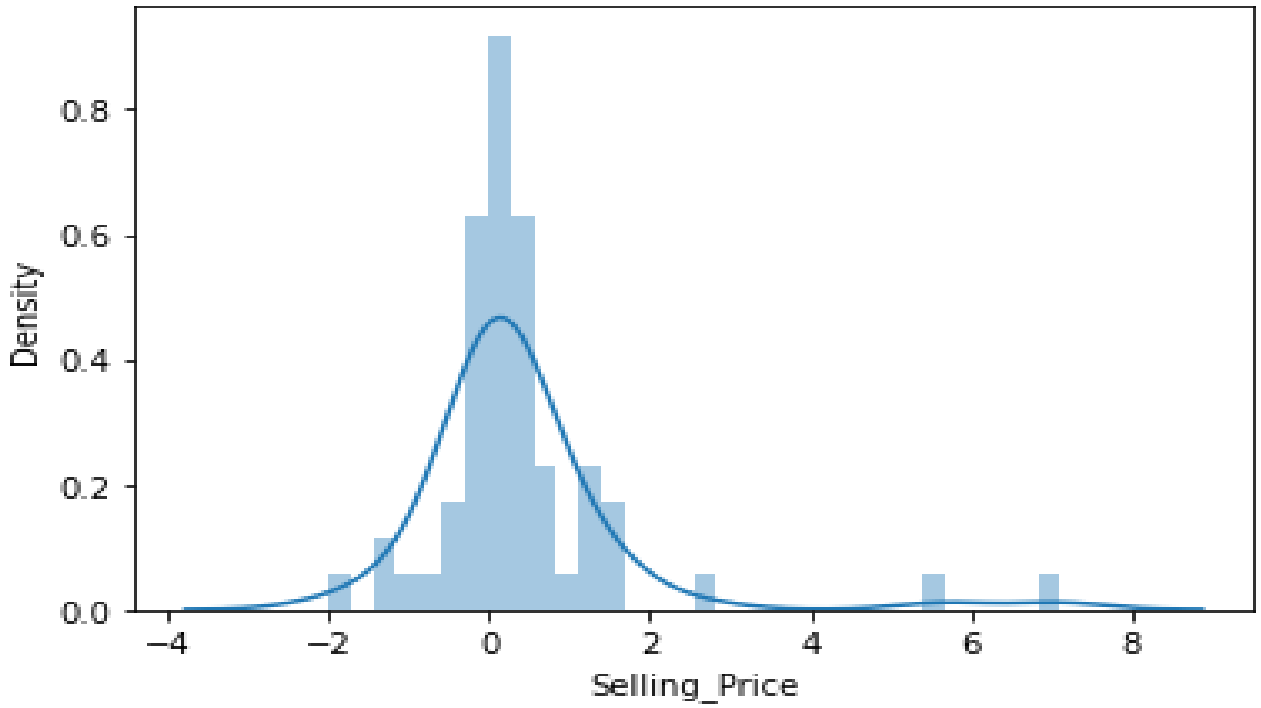
```
In [50]: import pickle
         # open a file, where you ant to store the data
         file = open('random_forest_regression_model.pkl', 'wb')

         # dump information to that file
         pickle.dump(rf_random, file)
```

To use the flask framework for deployment it is necessary to pack this whole and import it into the python file for creating web application.Hence, we dump our model into the pickle file.

8.TESTING:

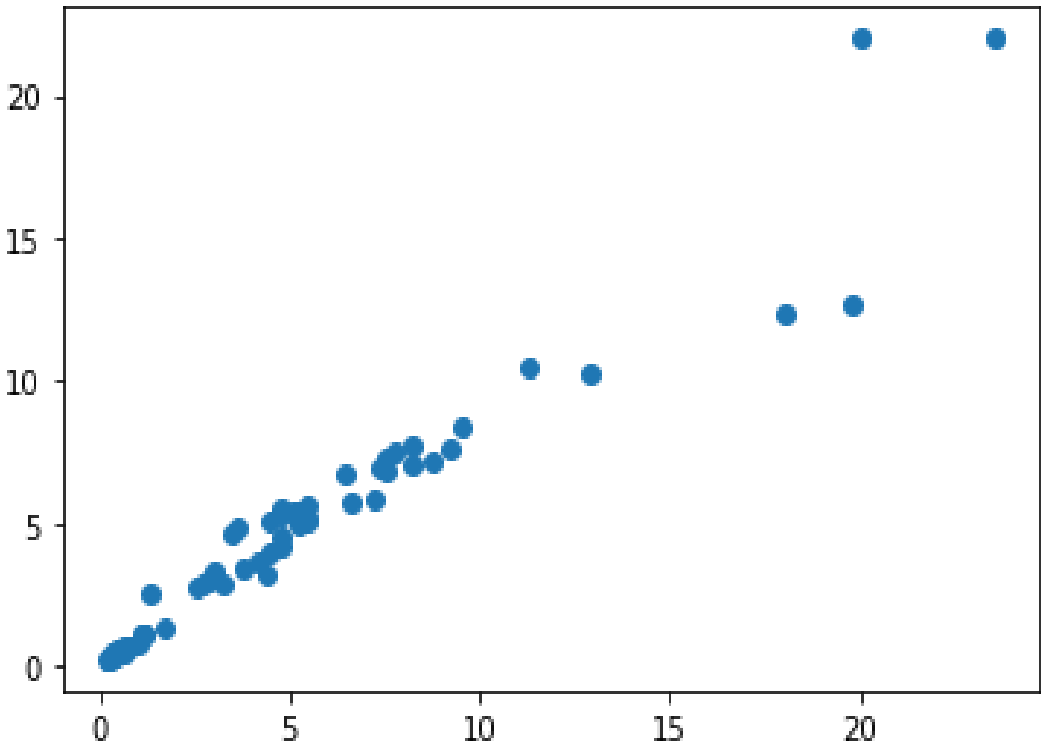
In the testing phase, we test the model using cross validation,we check the model is well or not and going is right or not,there are some technique of cross validation and we use confusion matrix for checking the model performance.We will test in all algorithms and we will get best suitable Seaborn is a python based data visualization library based on matplotlib.

The graph,basically, shows that the model which we have created gives us a very good result as it is a normal distribution representation.The close range on the graph gives us the idea that is predicting more accurately.

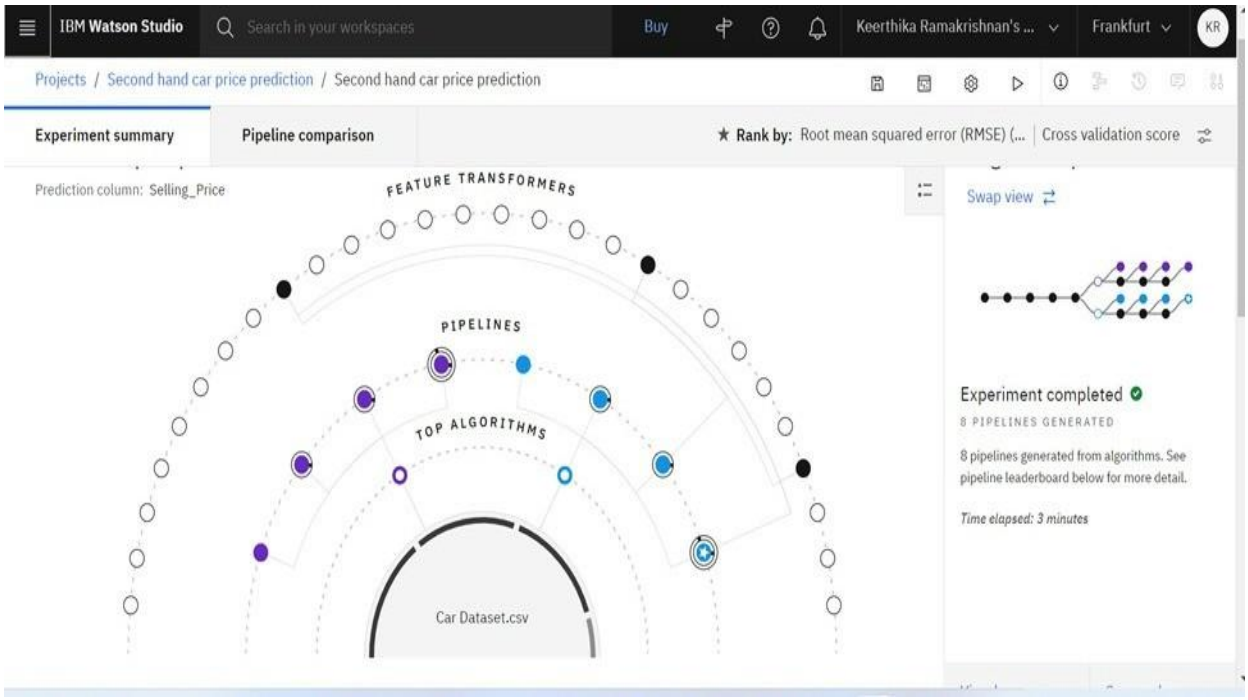


Distplot showing the distribution(Selling Price):

Scatterplot showing the Distribution:

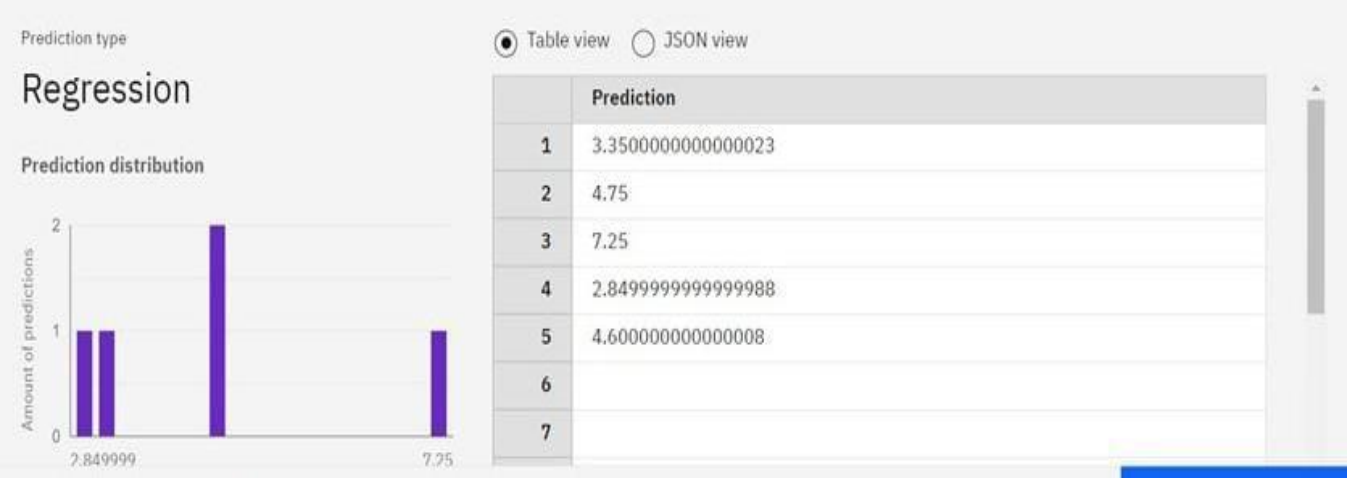


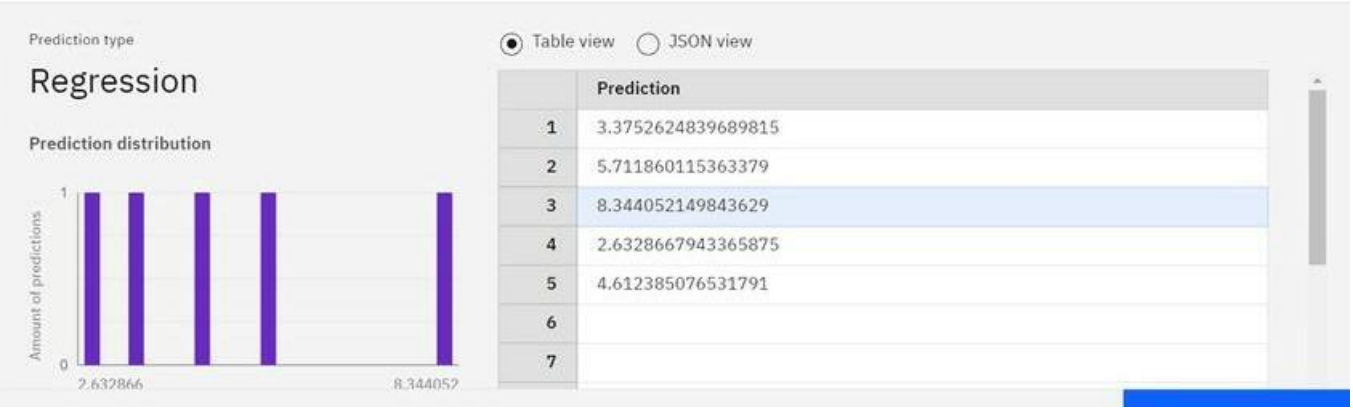
8.1 Test Cases:



ALGORITHMS	RESULTS
Random Forest Regression	0.970449
Lasso Regression	0.836209
Ridge Regression	0.872582
Linear Regression	0.872812

Prediction Results:





8.2 User Acceptance Testing:

Purpose of Document: The purpose of this document is to briefly explain the test coverage and open issuesof the [ProductName] project at the time of the release to User Acceptance Testing (UAT).

Defect Analysis:

This reportshows the numberof resolved or closed bugs at each severity level, and how they were resolved

Resolution	Severity 1	Severity 2	Severity 3	Severity 4	Subtotal
By Design	10	4	2	3	20
Duplicate	1	0	2	0	3
External	3	3	0	2	8
Fixed	12	2	4	16	34
Not Reproduced	0	0	1	0	1
Skipped	0	0	1	1	2
Won't Fix	0	0	0	0	0
Totals	26	9	10	22	70

Test Case Analysis:

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

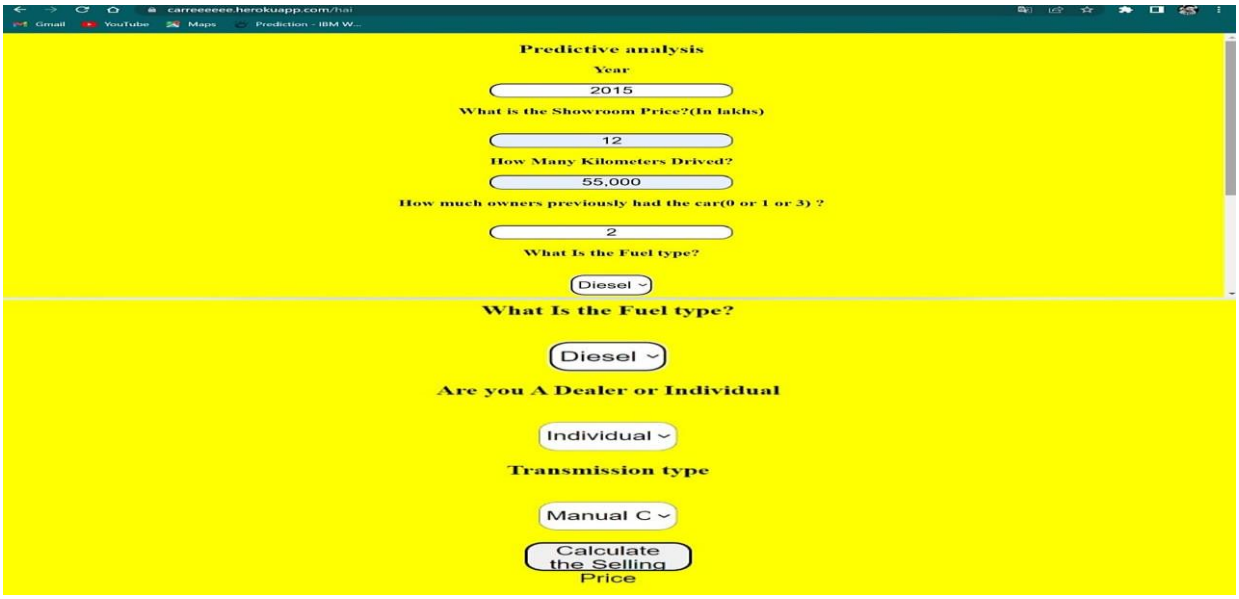
Section	Total Cases	Not Tested	Fail	Pass
Print Engine	8	0	0	8
Client Application	55	0	0	55
Security	5	0	0	5
Outsource Shipping	3	0	0	3
Exception Reporting	9	0	0	9
Final ReportOutput	5	0	0	5
Version Control	2	0	0	2

9 RESULTS:

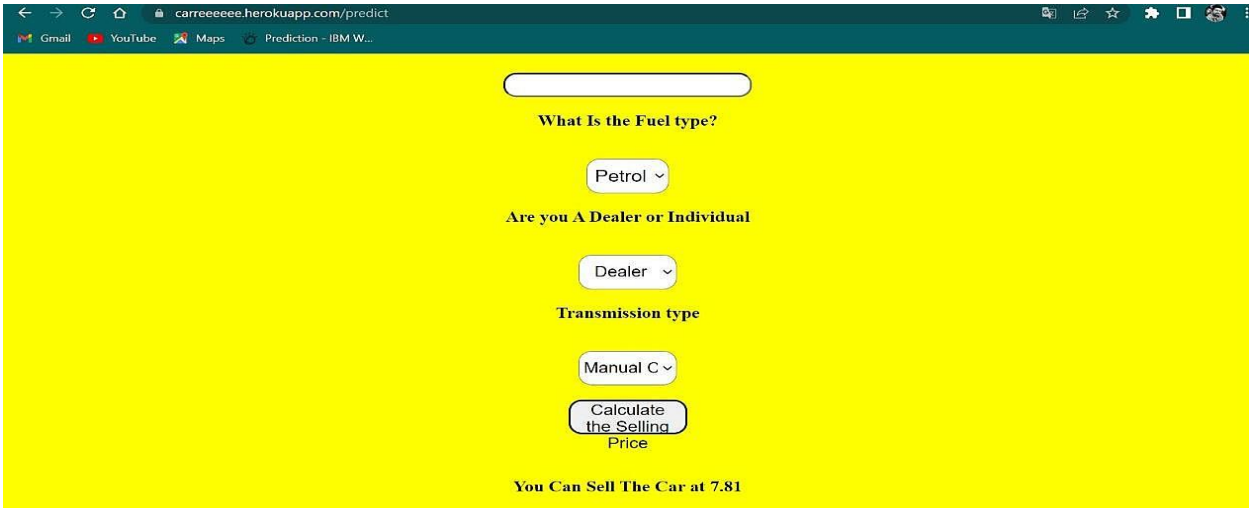
9.1 Performance Metrics:

In this we will be deploying the model on web.After testing and getting an output it would be stored in pickle file format using Python.We have used version controlled system as GitHub and further deployed on Heroku platform making the whole process more dynamic.

Input:



Output:



9 ADVANTAGES AND DISADVANTAGES:

Advantages:

- Variants usually don't matter in the used car market.If you search well, you can get a top-spec less driven car in the used car market at a price which you would have otherwise paid for a lower variant in case of buying a new car.
- If you buy a car from a brand authorized dealership,you get a warranty on the repair.
- If we are buying a used car that was launched a year ago,you can save upto 20% on its original cost.

Disadvantages:

- Some cars may be lemons.They look fine on outside but can land in huge repair costs while you use them.
- Be a very informed customer and check each and every possible details before buying.

10 CONCLUSION:

The model which we were making is to predict a value of second hand car using machine learning techniques. We have collected the data of cars from Kaggle having attributes like different cars and their year, kms driven, fuel type, transmission, etc. The data is then processed using different algorithms where results of each algorithms is compared, getting Random Forest algorithm as the most accurate amongst them, so we have used Random Forest because irrespective of size it runs efficiently and gives more accuracy than any other algorithm. We have also used ExtraTressRegressor for averaging to improve the predictive accuracy and control overfitting. It works by creating a large number of unpruned decision trees from the training dataset. It adds randomization but still has optimization. Further it would be available in GUI as a Web- application developed using Python flask making it user friendly so that users could give input and get the price of a car according to it.

12. FUTURE SCOPE:

- As a part of future work, we aim at the variable choices over the algorithms that were used in the project. We could only explore two algorithms whereas many other algorithms exist and might be more accurate. More specifications will be added to a system or provide more accuracy in terms of price in the system i.e.
    1) Horsepower
    2) Battery power
    3) Suspension
    4) Cylinder
    5) Torque
- As we know technologies are improving day by day and there is also advancement in-car technology, so our next upgrade will include hybrid cars, electric cars, and Driverless cars.

13. APPENDIX:

Detailed information, lengthy derivations, raw experimental observations etc. are to be presented in the separate appendices.

Web Application Code:

Python Flask Application:

```python
from flask import Flask, render_template, request
import jsonify
import requests
import pickle
import numpy as np
import sklearn
from sklearn.preprocessing import StandardScaler

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "v_ncy_zvGopd1_8JyfXQH6-6-4r3tL3v-MgJmGwBzhwV"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
```

```python
from flask import Flask, render_template, request
import jsonify
import requests
import pickle
import numpy as np
import sklearn
from sklearn.preprocessing import StandardScaler

import requests

# NOTE: you must manually set API_KEY below using information retrieved from your IBM Cloud account.
API_KEY = "v_ncy_zvGopd1_8JyfXQH6-6-4r3tL3v-MgJmGwBzhwV"
token_response = requests.post('https://iam.cloud.ibm.com/identity/token', data={"apikey":
 API_KEY, "grant_type": 'urn:ibm:params:oauth:grant-type:apikey'})
mltoken = token_response.json()["access_token"]

header = {'Content-Type': 'application/json', 'Authorization': 'Bearer ' + mltoken}
app = Flask(__name__)


model = pickle.load(open('random_forest_regression_model.pkl', 'rb'))
@app.route('/')
def index():
    return render_template('index1.html')

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

@app.route('/hello',methods=['GET'])
def Home():
    return render_template('index.html')
```

```python
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'])
        Kms_Driven2=np.log(Kms_Driven)
        Owner=int(request.form['Owner'])
        Fuel_Type_Petrol=request.form['Fuel_Type_Petrol']
        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([[Present_Price,Kms_Driven2,Owner,Year,Fuel_Type_Diesel,Fuel_Type_Petrol,Seller_Type_Individual,Transmission_Mannual]])


        output=round(prediction[0],2)
        payload_scoring = {"input_data": [{"field": [['Present_Price','Kms_Driven2','Owner','Year','Fuel_Type_Diesel','Fuel_Type_Petrol','Seller_Type_Individual','Transmission_Mannual'
```

```python
        response_scoring = requests.post('https://us-south.ml.cloud.ibm.com/ml/v4/deployments/5b9b8783-5cf9-4cde-b1c2-38957732f5d9/predictions?version=2022-11-19', json=payload_scoring
         headers={'Authorization': 'Bearer ' + mltoken})
        print("response_scoring")
        predictions=response_scoring.json()
        predict=predictions['predictions'][0]['values'][0][0]
        if output<0:
            return render_template('index.html',prediction_texts="Sorry you cannot sell this car")
        else:
            return render_template('index.html',prediction_text="You Can Sell The Car at {}".format(output))
    else:
        return render_template('index.html')

if __name__=="__main__":
    app.run(debug=True)
```

User Interface :

Index1.html:

```
<!DOCTYPE
html>
            <html lang="en">
              <head>
                <meta charset="UTF-8" />
                <meta name="viewport" content="width=device-width, initial-scale=1.0" />
                <title>Document</title>
                <link rel="stylesheet" href="../static/css/style.css" />
              </head>
              <body>

                <section class="hero">
                  <div class="hero-container">
                    <div class="column-left">
                      <h1>Welcome to Car Resale Value Prediction</h1>
                      <p>
                        With difficult economic condition, it is likely that sales of
            second-hand car will increase. In many developed countries, it is common to
            lease
            a car rather than buying. In order to predict the resale value of the car we
            proposed a effective system based on using regression algorithms.
                      </p>
                   <a href ="{{ 'hai' }}">
                      <button>Get Started</button></a>
                    </div>
                    <div class="column-right">
                      <img
                        src="../static/img/car%20image.jpg"
                        alt="illustration
                    "
                        class="hero-image"
                      />
                    </div>
                  </div>
                </section>
              </body>
            </html>
```

Index.html:

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

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

            <body>


                <div style="color:">
                    <form action="{{ url_for('predict')}}" method="post">
                        <h2>Predictive analysis</h2>
                        <h3>Year</h3>
                        <input id="first" name="Year" type="number ">
                        <h3>What is the Showroom Price?(In lakhs)</h3><br><input
            id="second" name="Present_Price" required="required">
                        <h3>How Many Kilometers Drived?</h3><input id="third"
            name="Kms_Driven" required="required">
                        <h3>How much owners previously had the car(0 or 1 or 3)
            ?</h3><br><input id="fourth" name="Owner" required="required">
                        <h3>What Is the Fuel type?</h3><br><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>
                        <h3>Are you A Dealer or Individual</h3><br><select
            name="Seller_Type_Individual" id="resea" required="required">
                            <option value="Dealer">Dealer</option>
                            <option value="Individual">Individual</option>
                        </select>
                        <h3>Transmission type</h3><br><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 ">Calculate the Selling
Price</button>
            <br>
        </form>
    </div>
    <style>
        body {
            background-color: yellow;
            text-align: center;
            padding: 0px;
        }

        #research {
            font-size: 18px;
            width: 100px;
            height: 23px;
            top: 23px;
        }

        #box {
            border-radius: 60px;
            border-color: 45px;
            border-style: solid;
            font-family: cursive;
            text-align: center;
            background-color: rgb(168, 131, 61);
            font-size: medium;
            position: absolute;
            width: 700px;
            bottom: 9%;
            height: 850px;
            right: 30%;
            padding: 0px;
            margin: 0px;
            font-size: 14px;
        }

        #fuel {
            width: 83px;
            height: 43px;
            text-align: center;
            border-radius: 14px;
            font-size: 20px;
        }
```

```
#fuel:hover {
    background-color: white;
}

#research {
    width: 99px;
    height: 43px;
    text-align: center;
    border-radius: 14px;
    font-size: 18px;
}

#research:hover {
    background-color: white;
}

#resea {
    width: 99px;
    height: 43px;
    text-align: center;
    border-radius: 14px;
    font-size: 18px;
}

#resea:hover {
    background-color: white;
}

#sub {
    width: 120px;
    height: 43px;
    text-align: center;
    border-radius: 14px;
    font-size: 18px;
}

#sub:hover {
    background-color: #fdcc04;
}

#first {
    border-radius: 14px;
    height: 25px;
    font-size: 20px;
    text-align: center;
}
```

```
        #second {
            border-radius: 14px;
            height: 25px;
            font-size: 20px;
            text-align: center;
        }

        #third {
            border-radius: 14px;
            height: 25px;
            font-size: 20px;
            text-align: center;
        }

        #fourth {
            border-radius: 14px;
            height: 25px;
            font-size: 20px;
            text-align: center;
        }
    </style>
</body>

</html>
```

GitHuB Link:

https://github.com/IBM-EPBL/IBM-Project-6053-1658822598