PROJECT REPORT

CAR RESALE VALUE PREDICTION

By team – PNT2022TMID53349

Batch no - B2-2M4E

SRI VENKATESWARA COLLEGE OF ENGINEERING

Under the guidance of,

DR. N.RAJGANESH (Mentor)

PROF SWETHA(Industry Mentor)

PROJECT REPORT

1. INTRODUCTION

1.1 PROJECT OVERVIEW

Today, one of the cornerstones of the economy is considered to be the transportation sector. In affluent countries, the automobile industry is alluded to as the "Industry of Industries." Professionals in the sector claim that the UAE's automotive sector has expanded significantly. It symbolizes its global prominence in relation to being the country with the automobile industry's strongest development. In most societies, both the native community and the former people who work there are already becoming progressively enthusiastic about cars. All makes and models of second hand cars, including those manufactured by well known brands, are readily available to purchase. Nowadays, almost everyone wants their own car, but many people choose to buy used cars due to issues with price or the state of the economy. Because used car prices depend on so many different features and conditions, it takes expertise to anticipate them accurately. Prices for used cars fluctuate on the market, therefore both buyers and sellers require an intelligence system to accurately anticipate the price. The collection of the dataset, which includes all crucial details like the car's manufacturing year, fuel type, kilometers driven, body type, the total number of previous owners, painted, transmission, the number of services, damage condition and Bharat stage is the most challenging task facing this intelligent system. It is obvious that a variety of elements influence the product's pricing, but sadly, details regarding these qualities are not always easily accessible. Since the market is the primary focus, the benchmark dataset with all essential features is scraped. 2 Before feeding the data straight into the data mining model, it is important to pre-process and transform the

obtained data into the appropriate format. The dataset was first statistically examined and plotted. The presence of duplicate, null, and missing values was found and corrected. To build an efficient model, the most correlated features were retained, and others were discarded. This prediction problem can be considered a regression problem since it belongs to the supervised learning domain. Two regressors known as random forest, XGBoost regression were trained and compared. A random forest Regressor outperformed all others in this project, so it was chosen as the main algorithm model.

1.2 PURPOSE

The objective of this study is to predict used cars prices using data mining techniques, by scraping data from websites that sell used cars, and analyzing the different aspects and factors that lead to the actual used car price valuation. To enable consumers to know the actual worth of their car or desired car, by simply providing the program with a set of attributes from the desired car to predict the car price. The purpose of this study is to understand and evaluate used car prices, and to develop a strategy that utilizes data mining techniques to predict used car prices. In aim of assisting consumers who want to purchase or sell cars in addition to providing them with a deeper understanding of the automotive industry, the project seeks to disseminate price prediction models to the general populace. Because some dealers are infamous for using dishonest sales techniques to complete a deal, purchasing a used automobile from a dealer can be a tedious and unsatisfactory experience. Therefore, this study aims to arm customers with the necessary tools to aid them in their purchasing experience and help them avoid falling prey to such strategies.

2. LITERATURE SURVEY

2.1 EXISTING PROBLEM

With the pandemic-related shortages of semiconductors throughout the past year, the secondhand car market has undergone a significant transformation. As a result, there was a rapid change in automobile prices during this study, which will have an impact on future predictions of actual car pricing. The autos on the market will be undervalued by the present dataset. As a result, the ideal approach would be to create a model that is based on real-time data and can be easily integrated into a public service.

2.2 REFERENCES

Several studies and related works have been done previously to predict used car prices around the world using different methodologies and approaches, with varying results of accuracy from 50% to 90%.

2.2.1 PREDICTING THE PRICE OF USED CARS USING MACHINE LEARNING TECHNIQUES

Sameerchand Pudaruth, (2014) the researcher proposed to predict used car prices in Mauritius, where he applied different machine learning techniques to achieve his results like decision tree, K-nearest neighbors, Multiple Regression and Naïve Bayes algorithms to predict the used cars prices, based on historical data gathered from the newspaper. The predictions are then evaluated and compared in order to find those which provide the best performance.

Main advantage in this paper was, four different machine learning techniques have been used to forecast the price of used cars in Mauritius. The mean error with linear regression was about Rs 51,000 while for KNN it was about Rs 27,000 for Nissan cars and about Rs 45,000 for Toyota cars. J48 and Naive

Bayes accuracy dangled between 60-70% different combinations of parameters. Main weakness of the decision tree and naïve Bayes that it is required to discretize the price and classify it which accrue to more inaccuracies. Moreover, he suggested a larger set of data to train the models hence the data gathered was not sufficient. Hence, the price attribute had to be classified into classes which contained a range of prices but this evidently introduced further grounds for inaccuracies. The main limitation of this study is the low number of records that have been used.

2.2.2 A STUDY ON USED CARS PRICE PREDICTION USING REGRESSION MODEL WITH REFERENCE TO CARTADE.COM

Monburinon, et al., (2018) Gathered data from a German e-commerce site that totaled to 304,133 rows and 11 attributes to predict the prices of used car using different techniques and measured their results using Mean Absolute Error (MAE) to compare their results. Same training dataset and testing dataset was given to each model. Growing Demand for Luxury Used Cars to Play Key Role in the Market. The Indian pre-owned car market is growing due to a steady increase in the demand for luxury cars. The sales of used luxury cars observed a 20% growth. Market, by Vehicle Types are Small, Mid-Size, Luxury.

Highest results achieved was by using gradient boosted regression tree with a MAE of 0.28, and MSE of 0.35 and 0.55 for mean absolute error and multiple linear regression respectively. Authors suggested adjusting the parameters in future works to yield better results, as well as using one hot encoding instead of label encoding for more realistic data interpretations on categorical data.

2.2.3 CAR PRICE PREDICTION USING MACHINE LEARNING TECHNIQUES

Gegic, Isakovic, Keco, Masetic, & Kevric (2019) To build a model for predicting the price of used cars in Bosnia and Herzegovina, Using data scrapped from a local Bosnian website for used cars totaled at 797 car samples after preprocessing, and proposed using these methods: Support Vector Machine, Random Forest and Artificial Neural network. However, the mentioned techniques were applied to work as an ensemble. The data used for the prediction was

collected from the web portal autopijaca.ba using web scraper that was written in PHP programming language. Respective performances of different algorithms were than compared to find one that best suits the available dataset. The final prediction model was integrated into Java application.

Main Advantage in this paper was, Due to the high number of attributes were considered for the accurate prediction. PHP scripts were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms. 12 Results have shown using only one machine learning algorithm achieved results less than 50%, whereas after combing the algorithms with pre calcification of prices using Random Forest, results with accuracies up to 87.38% was recorded.

2.2.4 VEHICLE PRICE PREDICTION SYSTEM USING MACHINE LEARNING TECHNIQUES

Noor & Jan (2017) were able to achieve high level of accuracy using Multiple linear regression models to predict the price of cars collected from used cars website in Pakistan called PakWheels that totaled to 1699 records after pre-processing, and where able to achieve accuracy of 98%, this was done after reducing the total amount of attributes using variable selection technique to include significant attributes only and to reduce the complexity of the model.

The data set used in this paper can be very valuable in conducting similar research using different prediction techniques. The prices of vehicles can be predicted using this data set on same or different prediction software as well. The data obtained under this research facilitated in prediction of prices of used cars through linear regression method. Many assumptions were made on the basis of the data set. The proposed system evaluated variables and selected the most relevant variables out of the dataset and reduced the complexity of model by eliminating unrelated variables during processing and analysis phase.

2.2.5 USED CAR PRICE PREDICTION USING K-NEAREST NEIGHBOR MODEL

K.Samruddhi & Kumar (2020) Proposed using Supervised machine leaning model using K-Nearest Neighbour to predict used car prices from a data set obtained from Kaggle containing 14 different attributes, using this method accuracy reached up to 85% after different values of K as well as Changing the percent of training data to testing data, expectedly when increasing the percent of data that is tested better accuracy results are achieved. The model was also cross validated with 5 and 10 folds by using the K fold method.

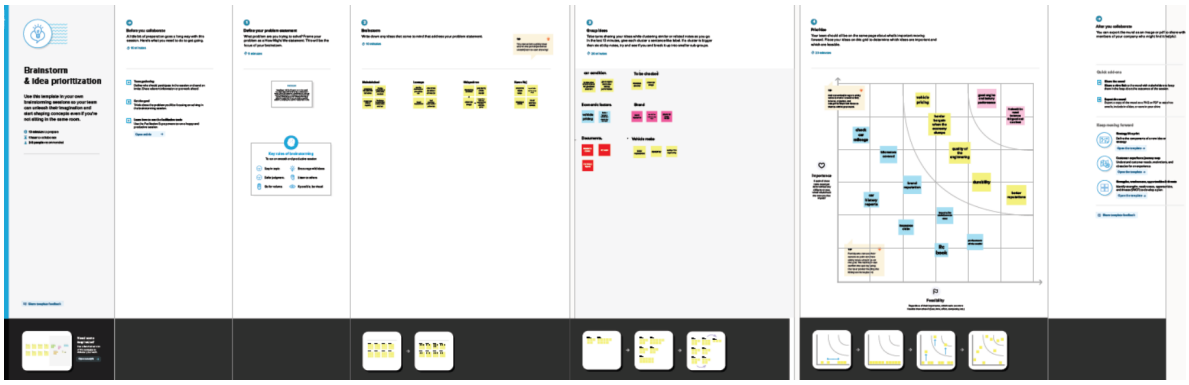
2.3 PROBLEM STATEMENT DEFINITION

Nowadays, the majority of consumers purchase used vehicles without understanding the true cost of each vehicle. With difficult economic conditions, it is likely the sales of second-hand imported cars and used cars will increase. Considering the main factors which would affect the resale value of a vehicle a regression model is to be built that would give the nearest resale value of the vehicle.

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS

3.2 IDEATION & BRAINSTORMING



Define your problem statement

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

Nowadays, a majority of customers purchase used vehicles without understanding the true cost of each vehicle. With difficult economic conditions, it is likely that sales of second-hand imported cars and used cars will increase. Considering the main factors which would affect the resale price of a car ,we predict a possible solution by analyzing the customer needs and demands.

Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP

You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

Mahalakshmi

- Analyzing the car after test drive
- Getting checked by a professiona; mechanic
- Check car history reports
- To change batteries that comes with a used car

Lavanya

- see the depreciation price of the car
- **check the condition of the car**
- **brand reputatation**
- **fuel economy and number pf owners**

Nithyashree

- kilometers covered
- popularity of the model
- check car millage
- keep all the necessary documents (RC book,Car insurance) perfect

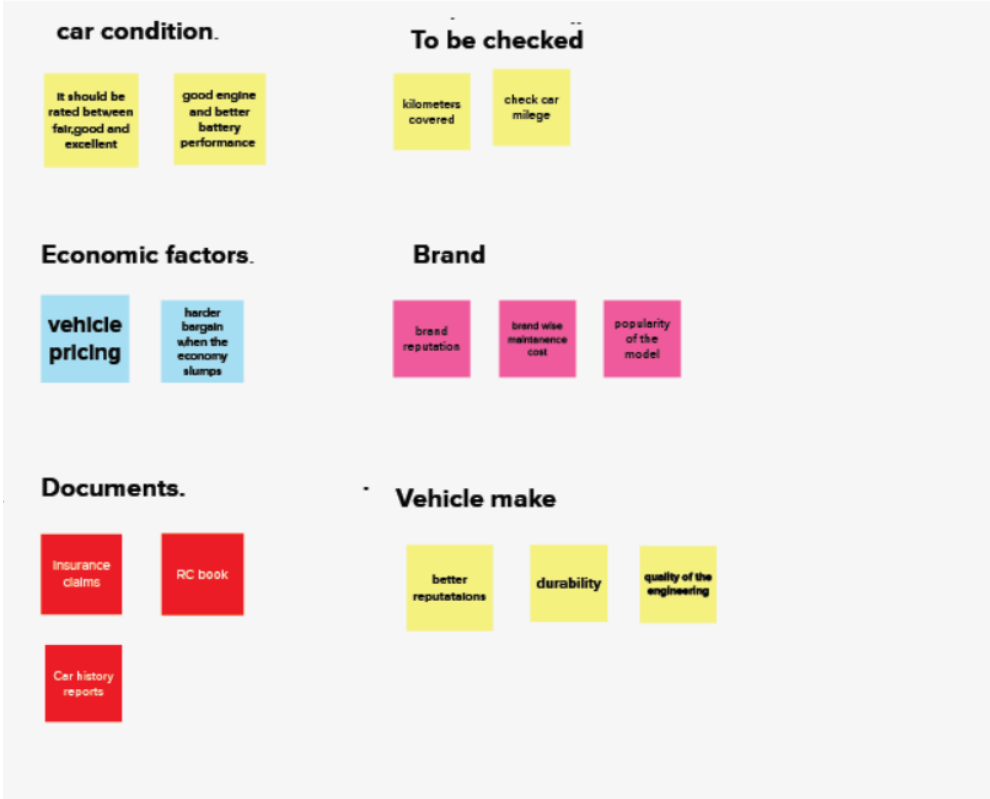
Karan Raj

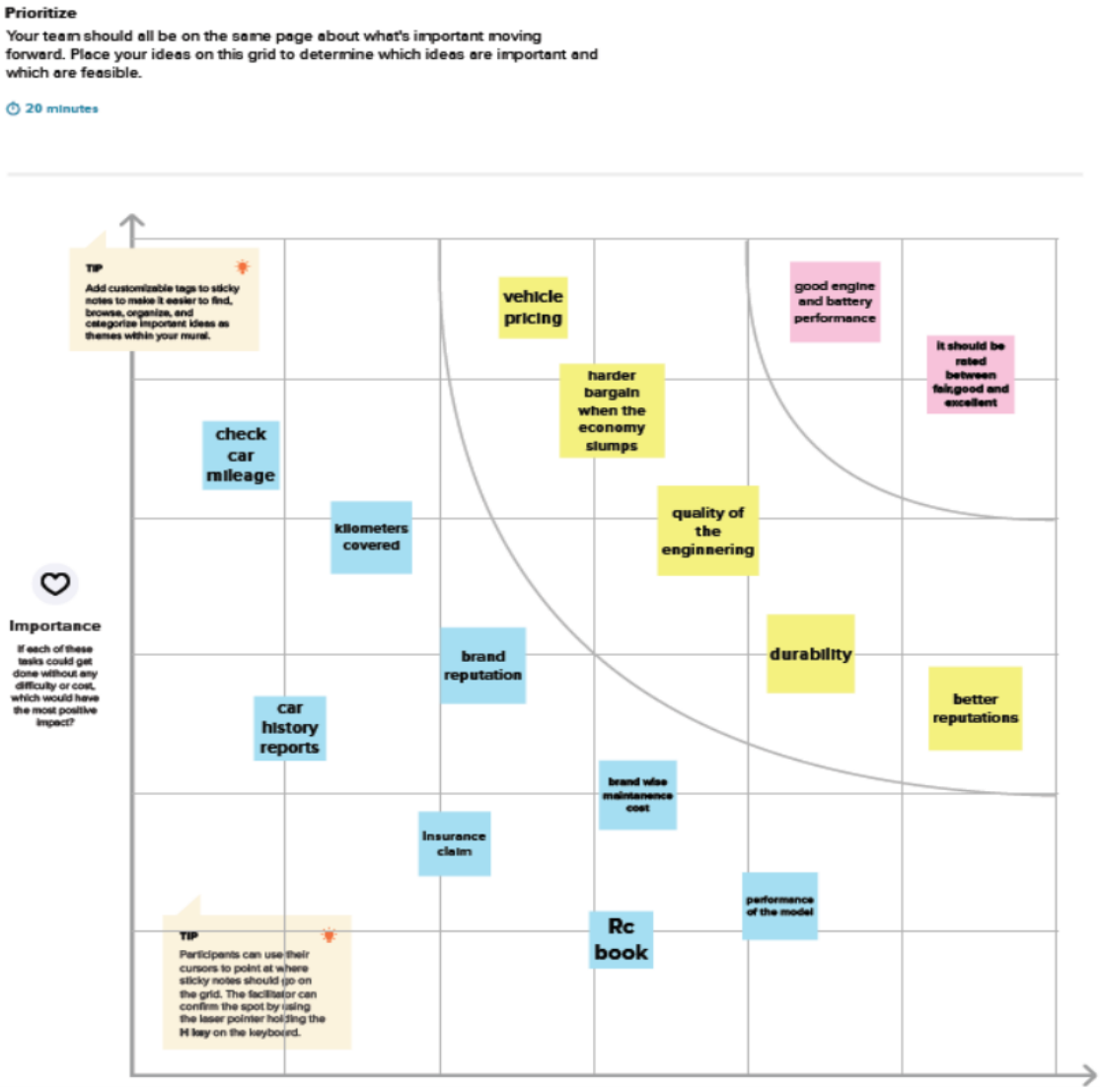
- check the resale value
- brand-wise maintenance cost
- insurance claims
- No of previous owners

Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. In the last 10 minutes, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

⏱ 20 minutes





3.3 PROPOSED SOLUTION

Despite buying a new car, most people prefer to purchase a used automobile. Even if buying a used automobile is less expensive, it may not be worth as much as the seller has said. To overcome this, one may decide to seek the advice of a professional with years of expertise in this sector. However, he could charge more for this reason. Or, if a person decides to purchase on their own, they may be duped into believing that the cost of purchasing a car is less than the actual offered price. The used vehicle price forecast based on the sale price of previously sold cars is a solution to this problem. by utilizing models from machine learning. These

machine learning models forecast the price of an automobile based on prior sales of cars in the same condition.

S.No.	Parameter	Description
1	Problem Statement (Problem to be solved)	The prices of new cars in the industry is fixed by the manufacturer with some additional costs incurred by the Government in the form of taxes. So, customers buying a new car can be assured of the money they invest to be worthy. But due to the increased price of new cars and the incapability of customers to buy new cars due to the lack of funds, used cars sales are on a global increase .There is a need for a used car price prediction system to effectively determine the worthiness of the car using a variety of features.
2	Idea / Solution description	Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car.We implement and evaluate various learning methods on a dataset consisting of the sale prices of different makes and models across cities in the United States. Our results show that Random Forest model and K-Means clustering with linear regression yield the best results.
3	Novelty / Uniqueness	Deciding whether a used car is worth the posted price when you see listings online can be difficult. Hence car resale value prediction gives the accurate price of the used cars according to their features.
4	Social Impact / Customer Satisfaction	Many websites, such as cars24.com, cardekho.com, and OLX.com, provide these buyers with a place to sell their old cars, but what should the car\’s price be? Machine learning algorithms may be able to overcome

		this problem. Regardless of how large or little the dataset is, the results show that both methods are highly accurate in prediction.
5	Business Model (Revenue Model)	Business model revolves around the parameters of buying any car from its owner at the best price compared with other car resale services in the area. Basically,it aims to offer consumers is an alternative to other tedious means of selling a used car, thus making this process simple and convenient.It offers a guaranteed price on any car, regardless of the model, age, or condition
6	Scalability of the Solution	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.

3.4 PROBLEM SOLUTION FIT

4. EMOTIONS: BEFORE / AFTER EM Before buying a car they experience a state of elation, joy , delight etc., After buying a car the sense of achievement they feel when they drive it home is linked to the fact that you are now the owner of the car.		8.2 OFFLINE When customers wanted to buy a car they would visit one auto dealership after another, talking with salespeople and seeing where they could get the best price.

4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

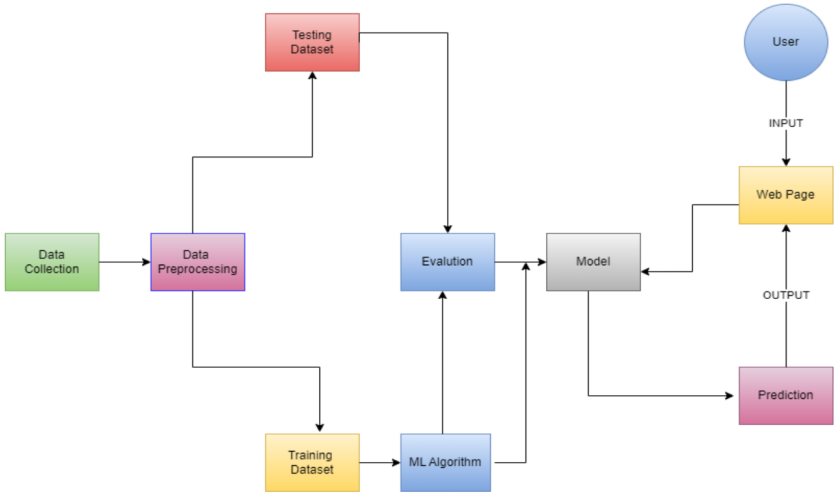
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Form Registration through Gmail Registration through LinkedIN
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Profile	User Details
FR-4	Database	Car Database 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

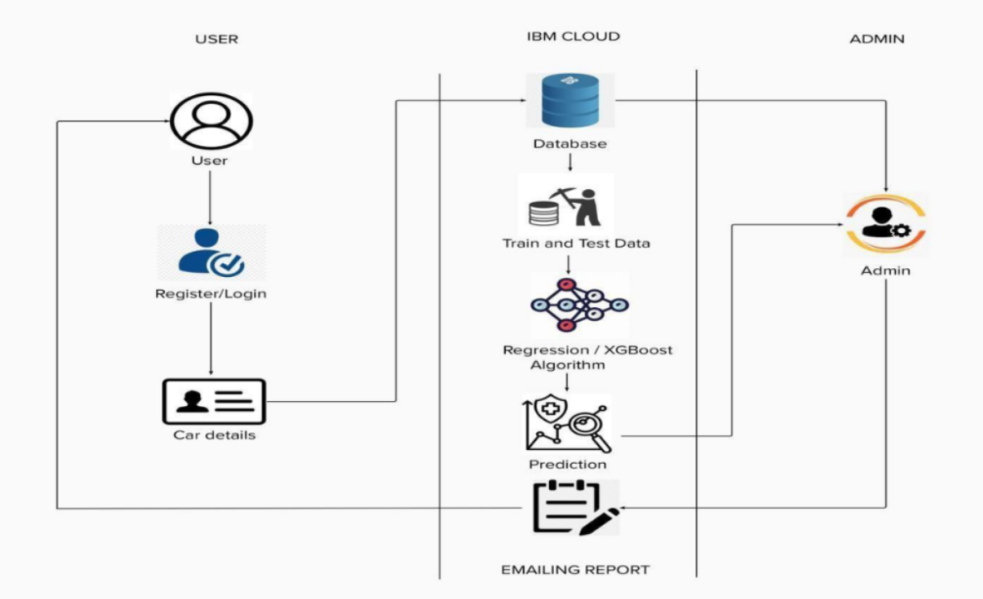
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	Great UI(User Interface), Accuracy in value prediction.
NFR-2	**Security**	Protect user password. Personal details.
NFR-3	**Reliability**	Rate of occurrence of failure is less. Failure free.
NFR-4	**Performance**	Perform correct prediction value, The landing page support several users and must provide 5 seconds or less response time.
NFR-5	**Availability**	Uninterrupted services must be available in all time except the time of server updation.
NFR-6	**Scalability**	Can handle any amount of data and perform many computations in a cost-effective and timesaving way to instantly serve millions of users residing at global locations.

5. PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS



5.2 SOLUTION & TECHNICAL ARCHITECTURE



A sophisticated, versatile, and effective approach based on regression algorithms is used to anticipate the resale value of the car. Taking into account the major elements that influence a vehicle's resale value, a regression model needs to be developed that will provide the vehicle's closest resale value. Various regression methods will be used, and the approach with the highest accuracy will be chosen as

a solution, which will then be integrated into the web-based application, where the user will be notified of the status of his product.

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 car details application by entering my email, password, and confirming my password.	I can access my account/ dashboard and view car details.	High	Sprint-1
		USN-2	As a user, I will receive car resale value in the application	I can receive car resale value in the application.	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	I can register & access the dashboard with Gmail Login	Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password	I can access the dashboard and view car details.	High	Sprint-1
	Dashboard		Display the details of different varieties of previously used cars.	I can know the resale value of a used car.	High	Sprint-1

6. PROJECT PLANNING & SCHEDULING

6.1 SPRINT PLANNING & ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Point	Priority	Team Members
Sprint-1	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	2	High	4
Sprint-1	Registration	USN-1	As a user, I can enter into the website through browser in Android	1	High	4
Sprint-1	Registration	USN-2	As a user, I can enter into the website through browser in ios	2	Medium	4
Sprint-1	Login	USN-3	As a user, I can find the car resale value prediction page in the website	1	High	4
Sprint-2	Home Page	USN-4	As a user, I need to select the parameters like Year, Showroom price, Kilometers driven, fuel type etc and click on the submit button	2	High	4
Sprint-3	Home Page	USN-5	As a user, I can see the accurate price for car resale after entering the details.	2	High	4
Sprint-4	Home Page	USN-6	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 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

Velocity:
Imagine we have a 10-day sprint duration, and the velocity of the team is 20 (points per sprint). Let's calculate the team's average velocity (AV) per iteration unit (story points per day)

$$AV = \frac{sprint\ duration}{velocity} = \frac{20}{10} = 2$$

7. CODING & SOLUTIONING

7.1 FEATURE 1

DATA PREPROCESSING

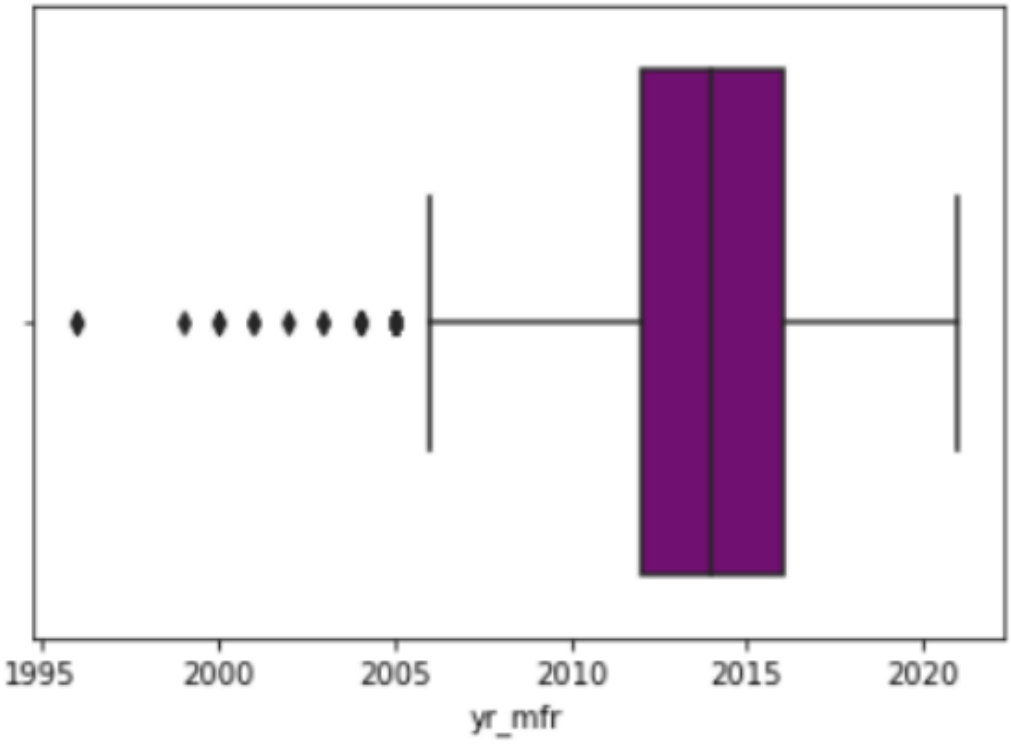
Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put it in a formatted way. So, for this, we use data preprocessing tasks. Real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data preprocessing is required for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

It involves below steps:
• Getting the dataset
• Importing libraries

• Importing datasets
• Finding Missing Data
• Encoding Categorical Data
• Splitting dataset into training and test set

DATA CLEANING

The practice of correcting or deleting inaccurate, damaged, improperly formatted, duplicate, or incomplete data from a dataset is known as data cleaning. There are numerous ways for data to be duplicated or incorrectly categorized when merging multiple data sources. Even if results and algorithms appear to be correct, they are unreliable if the data is inaccurate. Because the procedures will differ from dataset to dataset, there is no one definitive way to specify the precise phases in the data cleaning process. But it is essential to create a template for your 35 data cleaning procedure so you can be sure you are carrying it out correctly each time.



Before Outlier Detection - Box Plot

Step 1: Remove duplicate or irrelevant observations

Remove duplicate or pointless observations as well as undesirable observations from your dataset. The majority of duplicate observations will occur during data gathering.

Step 2: Filter unwanted outliers

There will frequently be isolated findings that, at first look, do not seem to fit the data you are evaluating. Removing an outlier if you have a good reason to, such as incorrect data entry, will improve the performance of the data you are working with. But occasionally, the emergence of an outlier will support a theory you are investigating.

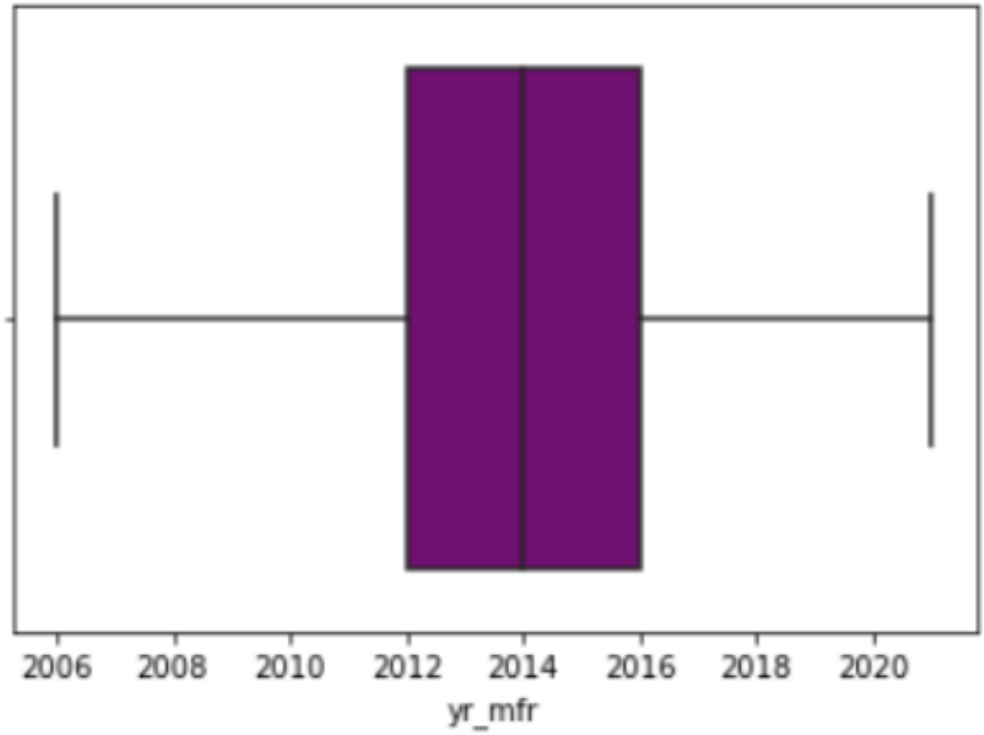
Step 3: Handle missing data

This can't ignore missing data because many algorithms will not accept missing values. There are a couple of ways to deal with missing data. Neither is optimal, but both can be considered.

1. As a first option, you can drop observations that have missing values, but doing this will drop or lose information, so be mindful of this before you remove it.

2. As a second option, you can input missing values based on other observations; again, there is an opportunity to lose integrity of the data because you may be operating from assumptions and not actual observations.

3. As a third option, you might alter the way the data is used to effectively navigate null values.



After Outlier Detection - Box plot

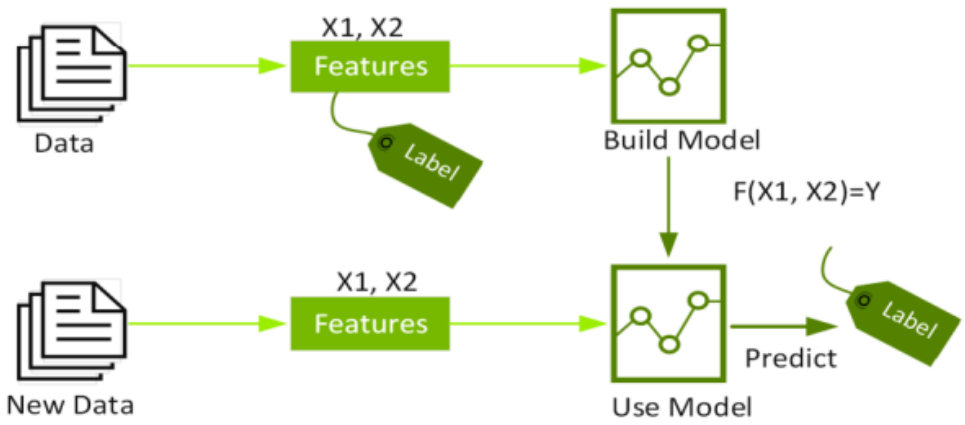
7.2 FEATURE 2

MODEL BUILDING

After performing feature selection, the dataset is split into a training and testing set. Then various Regression algorithms like Random Forest Regression and XGBoost Regression algorithm are applied on the training set. The model is first trained using the training set by fitting the X_train and y_train in each model. Then tested using the test set for each algorithm.

XGBoost Regression algorithm

XgBoost stands for Extreme Gradient Boosting, which was proposed by the researchers at the University of Washington. It is a library written in C++ which optimizes the training for Gradient Boosting.



XGBoost Regressor Model Building

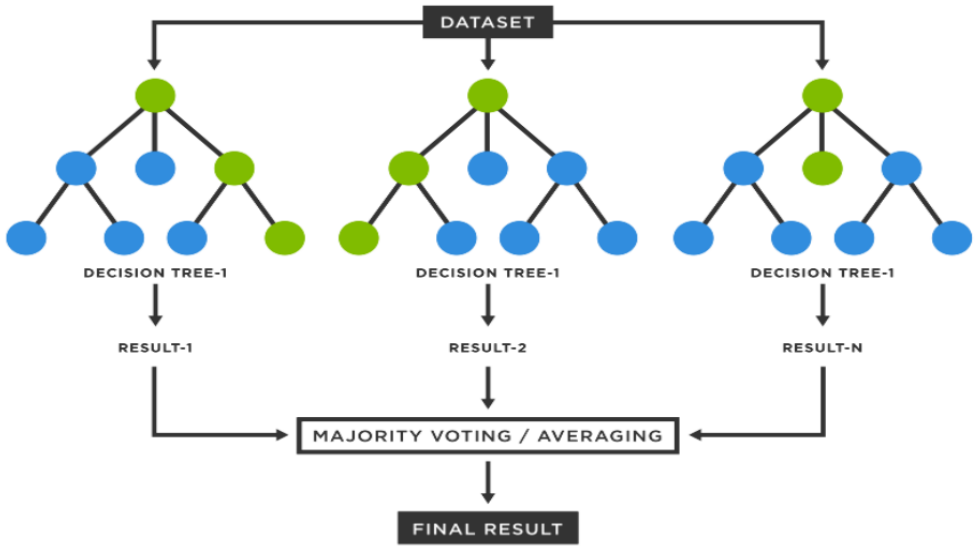
Random forest uses a technique called bagging to build full decision trees in parallel from random bootstrap samples of the data set. The final prediction is an average of all of the decision tree predictions. The term "gradient boosting" comes from the idea of "boosting" or improving a single weak model by combining it with a number of other weak models in order to generate a collectively strong model. Gradient boosting is an extension of boosting where the process of additively generating weak models is formalized as a gradient descent algorithm over an objective function. Gradient boosting sets targeted outcomes for the next model in an effort to minimize errors. Targeted outcomes for each case are based on the gradient of the error (hence the name gradient boosting) with respect to the prediction.

XGBoost Benefits and Attributes: A large and growing list of data scientists globally that are actively contributing to XGBoost open source development. Usage on a wide range of applications, including solving problems in regression, classification, ranking, and user-defined prediction challenges. A library that's highly portable and currently runs on OS X, Windows, and Linux platforms.

Random Forest Regression algorithm:

Random Forest is already revealing that it creates forest and then somehow randomizes it. It builds the forest through the ensemble of Decision Trees and most

of the time trains it using a method called the Bagging Method. Since it uses the ensemble method, the result is improved. Decision tree and bagging classifier parameters are the same. Each feature in the tree can be made random simply by adding thresholds.



Flowchart for Random Forest Algorithm

The following steps enable us to understand how Random Forest operates:

1. First, choose random samples from the given dataset.

2. Following that, each sample will be given a choice tree. Based on those choices, it will get a predicted end result.

3. For each anticipated outcome, voting may occur in this step.

4. In the end, choose the prediction outcome with the most votes because it is the very last prediction outcome.

7.3 DATA

This dataset contains over 6000+ true value cars data across all major tier 1 and tier 2 cities in India which is ready to accept a different owner. The information includes car name, make, model, fuel type, year of manufacture to mention a few.

Content:
car_name: Name of a car
yr_mfr: Car manufactured year
fuel_type: Type of fuel car runs on
kms_run: Number of kilometers run
body_type: Car body type. Ex: Sedan, hatchback etc.
transmission: Type of transmission. Ex: Manual, Automatic
make: Car manufacturing company
model: Car model name
total_owners: How many owners have already owned it?
original_price: Original price of a car (in INR)
warranty_avail: Warranty availability status
sale_price: Selling price of a car (in INR)

The 15 Independent attributes are,

1. Car name
2.Year Manufacture
3. Kilometers run
4. Transmission
5. Total owners
6. Model
7. Warranty available
8. Damage
9. Bharat stage
10. Fuel Type
11. Body Type

12. Make

13. Original Price

14. Paint

15. No. of Service.

The 1 Dependent attribute is,

1. Sale Price The 2 Unnecessary attributes are,

1. Id

2. Reserved

```
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              6399 non-null   int64
 1   car_name        6399 non-null   object
 2   yr_mfr          6399 non-null   int64
 3   fuel_type       6399 non-null   object
 4   kms_run         6399 non-null   int64
 5   sale_price      6399 non-null   int64
 6   body_type       6309 non-null   object
 7   transmission    5925 non-null   object
 8   make            6399 non-null   object
 9   model           6399 non-null   object
 10  total_owners    6399 non-null   int64
 11  orginal_price   3575 non-null   float64
 12  reserved        6399 non-null   bool
 13  warranty_avail  6399 non-null   bool
 14  paint           6399 non-null   object
 15  damage          6399 non-null   object
 16  no_of_service   6399 non-null   int64
 17  bs              6399 non-null   int64
dtypes: bool(2), float64(1), int64(7), object(8)
memory usage: 812.5+ KB
None
```

8. TESTING

Test Cases

- **Missing values**

The trained ML model requires 4 feature inputs for predicting the output. Failing which, the model throws invalid Input error. All the fields in the html form have been marked required using CSS and thus the user must input all fields.

Output: User must input all the fields, failing which, form shows warning message "this field needs to be filled". Thus, there can be no errors in model prediction.

- **Invalid Input**

The trained ML model requires only numerical input for all 4 features. Thus, if the user uses symbols such as a comma while input, the model may throw an error. To overcome the same, preprocessing script is deployed in the backend which removes all unwanted characters like comma, whitespaces etc. so that model gets required input.
Output: Due to python preprocessing script, model will get the desired input and thus will give accurate prediction.

- **Unseen year of purchase**

The model is trained with data from cars purchased from 2011 to 2020. If the user inputs details of the car purchased after that i.e., 2021, model may get confused since that data is quite new and unseen to model.

Output: Model has been trained with a boosting algorithm and thus it gives quite accurate results with around RMSE 65,000 INR.

User Acceptance Testing

The accuracy score of the various classification models is identified. The accuracy score percentage for XGBoost Regressor and Random Forest Regression is 94.58% and 99.11% respectively after performing feature selection. It is identified that Random Forest Regression has the highest accuracy score and low mean absolute

error, low mean square error and low root. The regression model can be evaluated on following parameters:

1. Mean Square Error (MSE): MSE is the single value that provides information about the goodness of regression line. Smaller the MSE value, better the fit because smaller value implies smaller magnitude of errors. $\boldsymbol{MSE = 1\ N\ \sum|yi - y|\ 2\ N\ i=1}$

2. Root Mean Square Error (RMSE): RMSE is the quadratic scoring rule that also measures the average magnitude of the error. It is the square root of average squared difference between prediction and actual observation.

3. Mean Absolute Error (MAE): This measure represents the average absolute difference between the actual and predicted values in the dataset. It represents the average residual from the dataset. $\boldsymbol{MAE = 1\ /\ N\ \sum|yi - y|}$

```
**************************************************
ERROR Table:
                    Random Forest Reg    XGBoost Regressor
Mean Absolute Error  1410.1145878378377   5658.9170507490635
Mean Square Error    803301237.6289136    4773133290.331509
RootMeanSquareError  28342.569354751762   69087.86644796254
**************************************************
```

Error Table

A software's user interface is very crucial. It serves as a connection point between the user and the machine where the user provides data and the computer processes it to provide results. The user can provide data to be processed in the context of machine learning, and the machine learning model fitted on the back end predicts the results and shows them in the user interface. In this prediction system it has been implemented with react which is a front-end framework of JavaScript. The user is required to fill out all of the facts in this prediction system, such as car name, year of manufacture, fuel type, kms run, body type, transmission, model, total owners, original price, and warranty availability. The ML model predicts and provides the price of the car in its current condition based on the data provided.

9. RESULTS

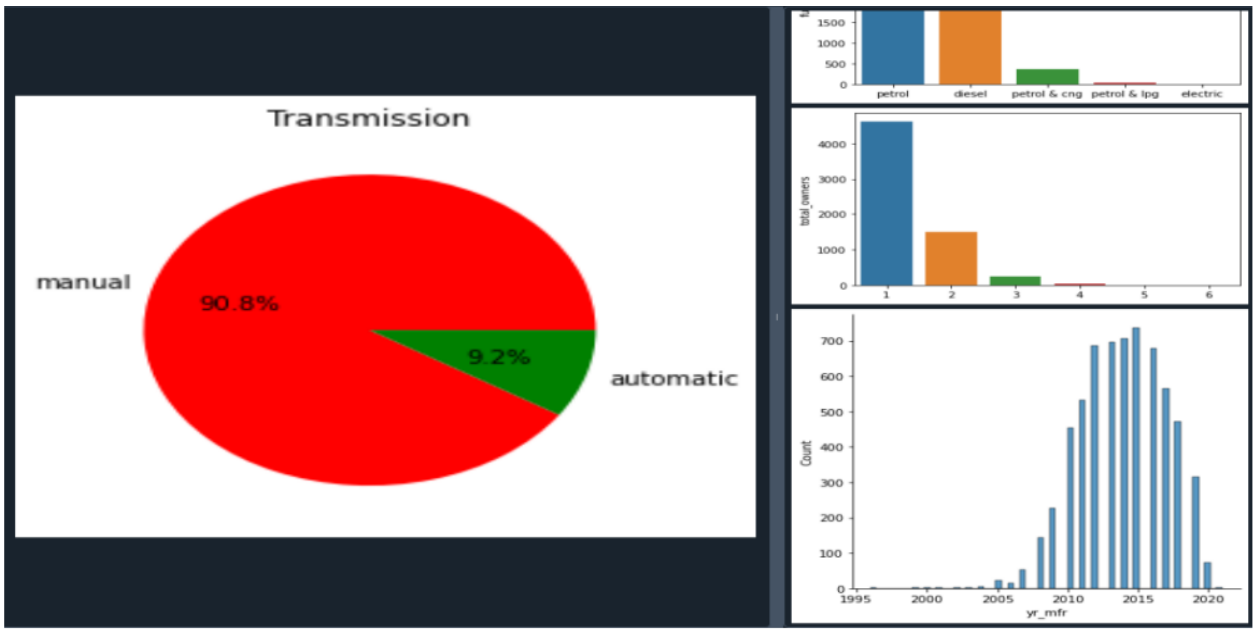
9.1 PERFORMANCE METRICS

9.1.1 DATA LOADING

```
In [6]: runfile('C:/Users/3108p/.spyder-py3/temp.py', wdir='C:/Users/3108p/.spyder-py3')
Reloaded modules: xgboost
   id           car_name  yr_mfr fuel_type  ...  paint  damage no_of_service bs
0   1       maruti swift    2015    petrol  ...    Yes     Yes             1  1
1   2    maruti alto 800    2016    petrol  ...    Yes      No             2  3
2   3  hyundai grand i10    2017    petrol  ...     No      No             0  3
3   4       maruti swift    2013    diesel  ...     No     Yes             5  3
4   5  hyundai grand i10    2015    petrol  ...     No      No             4  3

[5 rows x 18 columns]
(6399, 18)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6399 entries, 0 to 6398
Data columns (total 18 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              6399 non-null   int64
 1   car_name        6399 non-null   object
 2   yr_mfr          6399 non-null   int64
 3   fuel_type       6399 non-null   object
 4   kms_run         6399 non-null   int64
 5   sale_price      6399 non-null   int64
 6   body_type       6309 non-null   object
 7   transmission    5925 non-null   object
 8   make            6399 non-null   object
 9   model           6399 non-null   object
 10  total_owners    6399 non-null   int64
 11  orginal_price   3575 non-null   float64
 12  reserved        6399 non-null   bool
 13  warranty_avail  6399 non-null   bool
 14  paint           6399 non-null   object
 15  damage          6399 non-null   object
```

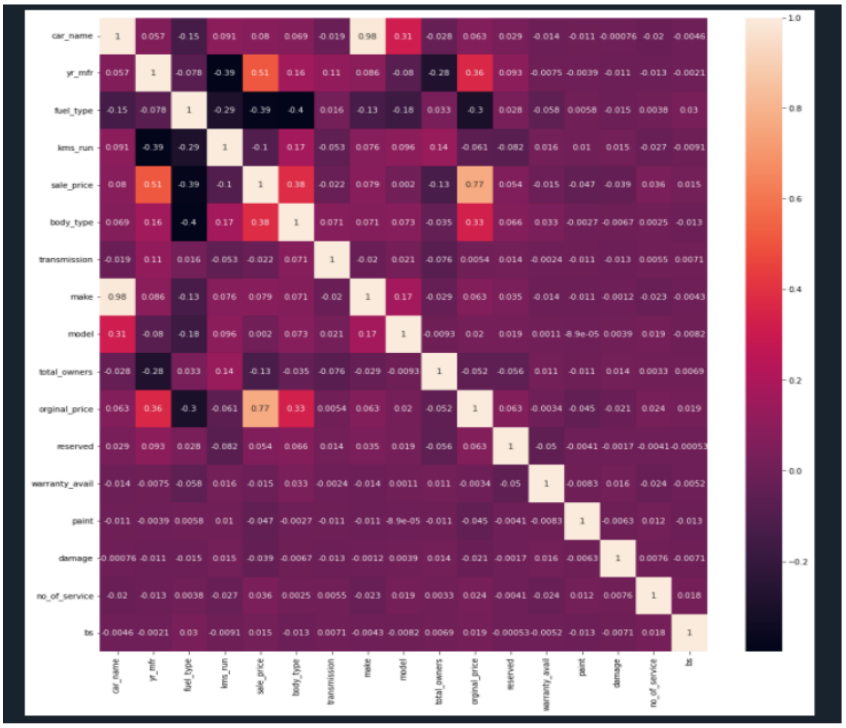
Implementation of Data Loading

9.1.2 EXPLORATORY DATA ANALYSIS



Implementation of EDA

9.1.3 DATA CLEANING



Correlation matrix

9.1.4 MODEL BUILDING

```
['car_name', 'yr_mfr', 'fuel_type', 'kms_run', 'body_type', 'transmission', 'make', 'model', 'total_owners',
'orginal_price', 'warranty_avail', 'paint', 'damage', 'no_of_service', 'bs']
(5119, 15)
(1280, 15)
R2 score:
                 XGBoost Regressor    Random Forest Reg
Before FS        0.9710601499422246   0.9745915261600064
----------------------------------------------------
```

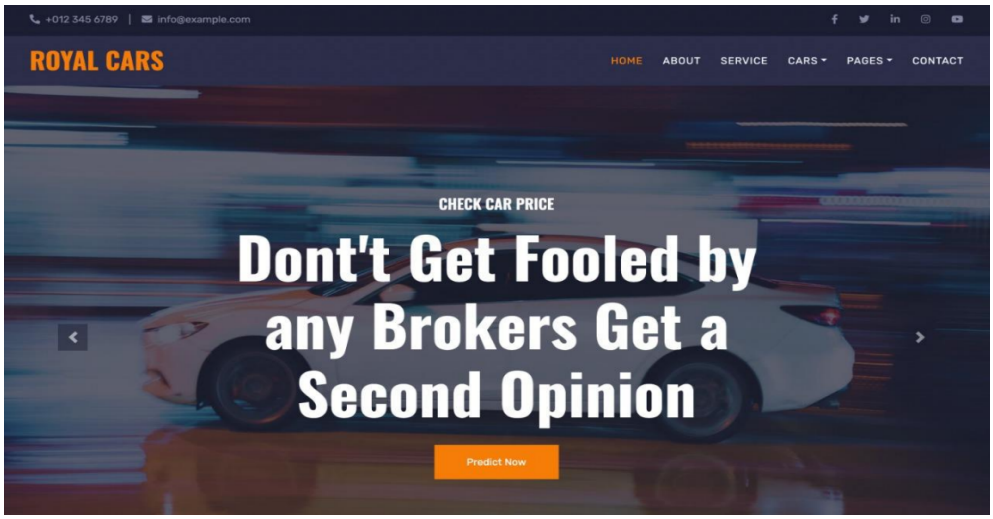
Model Building

9.1.5 EVALUATION METRICS AND ERROR DETECTION

```
**************************************************
ERROR Table:
                      Random Forest Reg    XGBoost Regressor
Mean Absolute Error   29549.262062499998   29202.80895014151
Mean Square Error     1726656926.4566293   1901837144.2898064
RootMeanSquareError   41553.06157741725    43610.057834057116
**************************************************
```

Performance Metrics

9.1.6 USER INTERFACE



9.1.7 RESULTS

10. ADVANTAGES & DISADVANTAGES

10.1 Advantages

- No need to seek out a specialist to assess the condition of the vehicle.
- There is no need to contact a third party to check the price.
- Always be knowledgeable of the vehicle's current condition price.
- The prediction method can provide a second viewpoint on purchasing a used car.
- The accuracy of the prediction system can be enhanced by increasing the volume of information provided to the machine learning model.

10.2 Disadvantages

- Gathering more data can yield more robust predictions.
- The data cleaning process can be done more rigorously with the help of more technical information.

11. CONCLUSION

The proposed system enables a clear vision for the customers, investors, and the people who don't have car resale knowledge to identify the price of a car which was already used by a person. And with the help of our application and by entering details about the car, customers can make sure that they are buying the car at a reasonable price. This work makes it useful for all people without car knowledge to gain knowledge about the car prices in real-time.

12. FUTURE SCOPE

The system can be made more efficient by the real-time updating of new car details into the training dataset such that the model becomes more efficient in terms of its identification as well as its feature characteristics. This model along with identification, corresponding links are displayed such that to provide quality means of knowledge about the subject to the consumer. The real-time updating of the model recurrently increases the training dataset providing higher accuracy score.

13. APPENDIX

13.1 REQUIREMENT.TXT

Flask == 2.2.2

numpy == 1.23.4

pandas == 1.5.1

scikit-learn == 1.1.3

xgboost == 1.7.1

matplotlib == 3.6.2

reactJS

13.2 SERVER.PY

server.py > getdata

```python
import pickle
from flask import Flask, request
from flask_cors import CORS, cross_origin
import json
from json import JSONEncoder
import numpy
import dataParsing
import time

app = Flask(__name__)
cors = CORS(app)
app.config['CORS_HEADERS'] = 'Content-Type'

# Load the model
model = pickle.load(open("./model.pkl",'rb'))

class NumpyArrayEncoder(JSONEncoder):
    def default(self, obj):
        if isinstance(obj, numpy.ndarray):
            return obj.tolist()
        return JSONEncoder.default(self, obj)

#• Members API Route
@app. route("/members")
def members ():
    # car_name(48),yr_mfr(2017.0),fuel_type(2),kms_run(17406),body_type
    # ['car_name', 'yr_mfr', 'fuel_type', 'kms_run', 'body_type', 'trar
    result=model.predict([[48,2017.0,2,17406,0,2,7,71,1,0,1,0,2,1]])
    data={"carvalue": result }
    encodedNumpyData = json.dumps(data, cls=NumpyArrayEncoder)
    return encodedNumpyData

@app.route("/home")
@cross_origin()
def getdata():
    data = dataParsing.Data.getData()
    return data
```

13.3 XGBOOSTREG.PY

xgboostreg.py > ...

```python
# -*- coding: utf-8 -*-
"""
Created on Sun Oct  9 14:25:54 2022

@author: 3108p
"""
import math
import pandas as pd
import numpy as np
from collections import defaultdict

class XGBoostModel():
    def __init__(self, params, random_seed=None):
        self.params = defaultdict(lambda: None, params)
        self.subsample = self.params['subsample'] \
            if self.params['subsample'] else 1.0
        self.learning_rate = self.params['learning_rate'] \
            if self.params['learning_rate'] else 0.3
        self.base_prediction = self.params['base_score'] \
            if self.params['base_score'] else 0.5
        self.max_depth = self.params['max_depth'] \
            if self.params['max_depth'] else 5
        self.rng = np.random.default_rng(seed=random_seed)

    def fit(self, X, y, objective, num_boost_round, verbose=False):
        current_predictions = self.base_prediction * np.ones(shape=y.shape)
        self.boosters = []
        for i in range(num_boost_round):
            gradients = objective.gradient(y, current_predictions)
            hessians = objective.hessian(y, current_predictions)
            sample_idxs = None if self.subsample == 1.0 \
                else self.rng.choice(len(y),
                                     size=math.floor(self.subsample*len(y)),
                                     replace=False)
            booster = TreeBooster(X, gradients, hessians,
                                  self.params, self.max_depth, sample_idxs)
            current_predictions += self.learning_rate * booster.predict(X)
```

13.4 MODEL.PY

model.py > ...

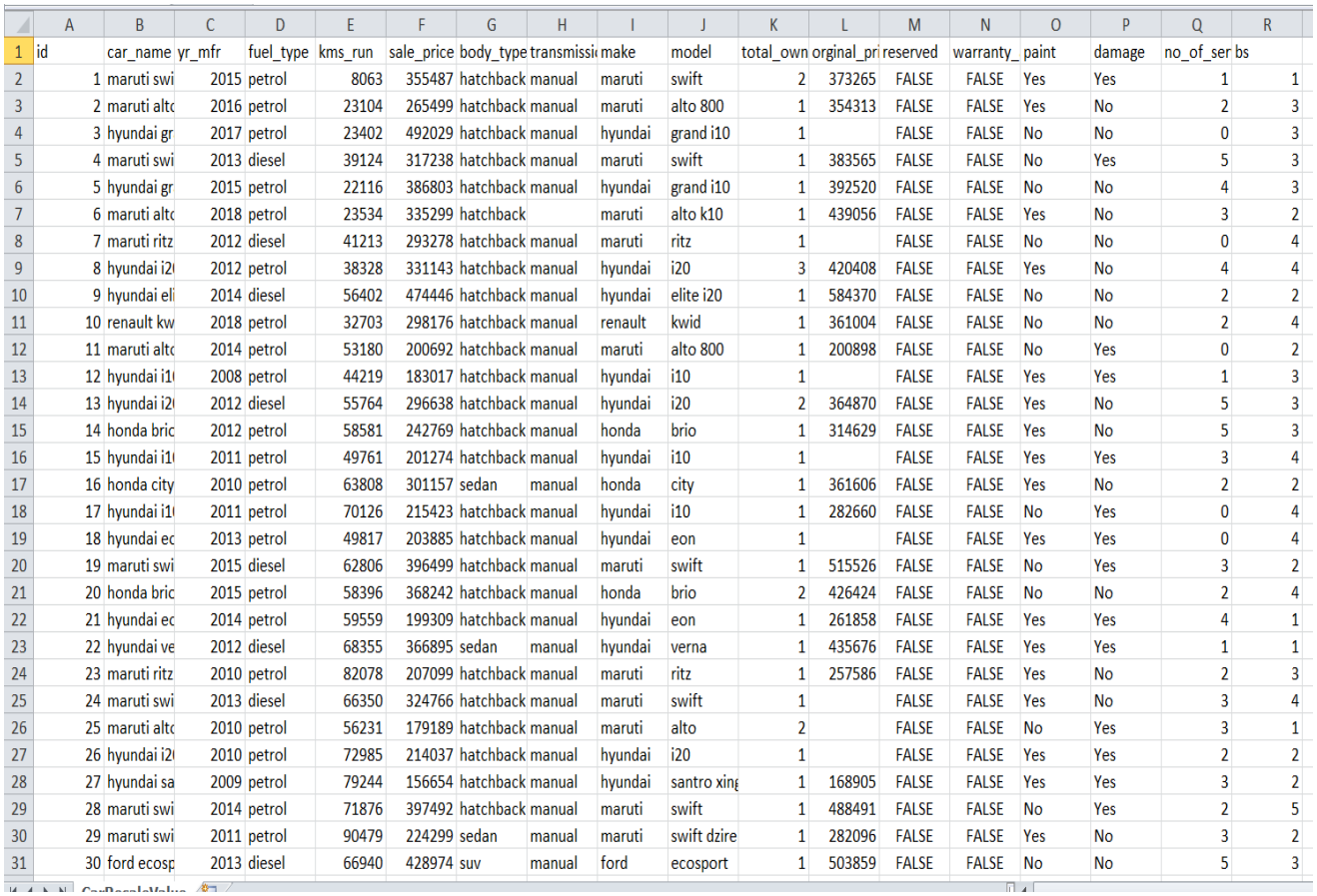
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import r2_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
import xgboostreg
import pickle


#Data Loading
df = pd.read_csv("./CarResaleValue.csv")
#printing first 5 rows of dataset
#print(df.head())
#Printing no. of Rows and Columns in the dataset
print(df.shape)

# print(df.describe())# only shows numerical column
df = df.drop(columns=['id'],axis=1)

# print(df.info())

#sns.boxplot(df.yr_mfr)
q1 = df.yr_mfr.quantile(0.25)
q3 = df.yr_mfr.quantile(0.75)
IQR = q3-q1
upper_limit = q3 + 1.5*IQR
lower_limit = q1 - 1.5*IQR
df['yr_mfr']=np.where(df['yr_mfr']<lower_limit,df['yr_mfr'].median(),df['yr_mfr'])
#sns.boxplot(df.yr_mfr)

le = LabelEncoder()
for column in df.columns:
    if df[column].dtype == object or df[column].dtype == bool:
```

13.5 DATAPARSING.PY

dataParsing.py > Data > getData

```python
import numpy as np
import pandas as pd
import json

class Data:
    def getData():
        car_csv = pd.read_csv('./CarResaleValue.csv')
        car_make_name = car_csv['make'].unique()
        body_type = car_csv['body_type'].unique()
        car_data = car_csv['car_name'].tolist()
        fuel_type=car_csv['fuel_type'].unique().tolist()
        body_type=car_csv['body_type'].unique()
        body_type=[x for x in body_type if x == x]
        transmission=car_csv['transmission'].unique()
        transmission=[x for x in transmission if x == x]
        cars = []

        for i in car_csv['make']+":"+car_csv['model'].tolist():
            cars.append(str(i).split(":",1))

        car_name = {}
        for i in car_make_name:
            car_name[i] = []

        for car in cars:
            if car[1] not in car_name[car[0]]:
                car_name[car[0]].append(car[1])
        data={}
        data["car_name"] = car_name
        data["fuel_type"]=fuel_type
        data["body_type"]=body_type
        data["transmission"]=transmission
        return data
    def dataLabels():
        df = pd.read_csv('./CarResaleValue.csv')
        datalabels={}
        carname =df.car_name.unique()
```

13.6 CARRESALEVALUE.CSV



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
1	id	car_name	yr_mfr	fuel_type	kms_run	sale_price	body_type	transmissi	make	model	total_own	orginal_pr	reserved	warranty_	paint	damage	no_of_ser	bs
2	1	maruti swi	2015	petrol	8063	355487	hatchback	manual	maruti	swift	2	373265	FALSE	FALSE	Yes	Yes	1	1
3	2	maruti alto	2016	petrol	23104	265499	hatchback	manual	maruti	alto 800	1	354313	FALSE	FALSE	Yes	No	2	3
4	3	hyundai gr	2017	petrol	23402	492029	hatchback	manual	hyundai	grand i10	1		FALSE	FALSE	No	No	0	3
5	4	maruti swi	2013	diesel	39124	317238	hatchback	manual	maruti	swift	1	383565	FALSE	FALSE	No	Yes	5	3
6	5	hyundai gr	2015	petrol	22116	386803	hatchback	manual	hyundai	grand i10	1	392520	FALSE	FALSE	No	No	4	3
7	6	maruti alto	2018	petrol	23534	335299	hatchback		maruti	alto k10	1	439056	FALSE	FALSE	Yes	No	3	2
8	7	maruti ritz	2012	diesel	41213	293278	hatchback	manual	maruti	ritz	1		FALSE	FALSE	No	No	0	4
9	8	hyundai i2	2012	petrol	38328	331143	hatchback	manual	hyundai	i20	3	420408	FALSE	FALSE	Yes	No	4	4
10	9	hyundai eli	2014	diesel	56402	474446	hatchback	manual	hyundai	elite i20	1	584370	FALSE	FALSE	No	No	2	2
11	10	renault kw	2018	petrol	32703	298176	hatchback	manual	renault	kwid	1	361004	FALSE	FALSE	No	No	2	4
12	11	maruti alto	2014	petrol	53180	200692	hatchback	manual	maruti	alto 800	1	200898	FALSE	FALSE	No	Yes	0	2
13	12	hyundai i1	2008	petrol	44219	183017	hatchback	manual	hyundai	i10	1		FALSE	FALSE	Yes	Yes	1	3
14	13	hyundai i2	2012	diesel	55764	296638	hatchback	manual	hyundai	i20	2	364870	FALSE	FALSE	Yes	No	5	3
15	14	honda bric	2012	petrol	58581	242769	hatchback	manual	honda	brio	1	314629	FALSE	FALSE	Yes	No	5	3
16	15	hyundai i1	2011	petrol	49761	201274	hatchback	manual	hyundai	i10	1		FALSE	FALSE	Yes	Yes	3	4
17	16	honda city	2010	petrol	63808	301157	sedan	manual	honda	city	1	361606	FALSE	FALSE	Yes	No	2	2
18	17	hyundai i1	2011	petrol	70126	215423	hatchback	manual	hyundai	i10	1	282660	FALSE	FALSE	No	Yes	0	4
19	18	hyundai ec	2013	petrol	49817	203885	hatchback	manual	hyundai	eon	1		FALSE	FALSE	Yes	Yes	0	4
20	19	maruti swi	2015	diesel	62806	396499	hatchback	manual	maruti	swift	1	515526	FALSE	FALSE	No	Yes	3	2
21	20	honda bric	2015	petrol	58396	368242	hatchback	manual	honda	brio	2	426424	FALSE	FALSE	No	No	2	4
22	21	hyundai ec	2014	petrol	59559	199309	hatchback	manual	hyundai	eon	1	261858	FALSE	FALSE	Yes	Yes	4	1
23	22	hyundai ve	2012	diesel	68355	366895	sedan	manual	hyundai	verna	1	435676	FALSE	FALSE	Yes	Yes	1	1
24	23	maruti ritz	2010	petrol	82078	207099	hatchback	manual	maruti	ritz	1	257586	FALSE	FALSE	Yes	No	2	3
25	24	maruti swi	2013	diesel	66350	324766	hatchback	manual	maruti	swift	1		FALSE	FALSE	Yes	No	3	4
26	25	maruti alto	2010	petrol	56231	179189	hatchback	manual	maruti	alto	2		FALSE	FALSE	No	Yes	3	1
27	26	hyundai i2	2010	petrol	72985	214037	hatchback	manual	hyundai	i20	1		FALSE	FALSE	Yes	Yes	2	2
28	27	hyundai sa	2009	petrol	79244	156654	hatchback	manual	hyundai	santro xing	1	168905	FALSE	FALSE	Yes	Yes	3	2
29	28	maruti swi	2014	petrol	71876	397492	hatchback	manual	maruti	swift	1	488491	FALSE	FALSE	No	Yes	2	5
30	29	maruti swi	2011	petrol	90479	224299	sedan	manual	maruti	swift dzire	1	282096	FALSE	FALSE	Yes	No	3	2
31	30	ford ecosp	2013	diesel	66940	428974	suv	manual	ford	ecosport	1	503859	FALSE	FALSE	No	No	5	3

LINKS

GITHUB :

https://github.com/IBM-EPBL/IBM-Project-42472-1660663996

DRIVE LINK :

https://drive.google.com/drive/u/0/folders/1TVaqQq9r6ZqJMmlsiZm-SqtGrw8pvIDk