“CAR PRICE PREDICTOR”

Project report Submitted in partial fulfillment of the requirements forthe award of degree of

Bachelor of Technology

in

Information Technology

by

SYED ARHAAM DANIYAL	**(19P71A1206)**
NEHA YADAV	**(19P71A1236)**
SHAIK RAHEEMUDDIN	**(20P75A1209)**

Under the Esteemed Guidance of

Mrs. A.Jyotsna, Assistant Professor

Department of Information Technology

SWAMI VIVEKANANDA INSTITUTE OF TECHNOLOGY

Mahbub College Campus, Secunderabad-500003

Affiliated to JNTU-H)

2019-2023

CERTIFICATE

This is to certify that the project report entitled **"CAR PRICE PREDICTOR"** is being submitted by SYED ARHAAM DANIYAL(19P71A1206), NEHA YADAV(19P71A1236),SHAIK RAHEEMUDDIN(20P75A1209) in partial fulfillment for the award of Degree of BACHELOR OF TECHNOLOGY in INFORMATION TECHNOLOGY to the Jawaharlal Nehru Technological University is a record of bonafide work carried out by him/her under my guidance and supervision.

Date:

Internal Guide

(Mrs.A. Jyotsna)

HOD-IT
(Mrs.M. Supriya)

External Examiner

Principal

(Dr.V.Usha Shree)

ACKNOWLEDGEMENT

First of all, we thank our project Guide Mrs.A.Jyotsna Assistant Professor, and HOD Mrs.M. Supriya of the Department of Information Technology for giving us this opportunity in developing this project. This project has really helped us in enhancing our skills of programming, the perspective with which we should view projects and above all, our presentation and interpersonal skills. Last but not the least, we also thank the professors of our department for lending their helping hand whenever it was necessary.

SYED ARHAAM DANIYAL
(19P71A1206)

NEHA YADAV
(19P71A1236)

SHAIK RAHEEMUDDIN
(20P75A1209)

DECLARATION

We hereby declare that the work which is being presented in this Mini Project entitled, **"CAR PRICE PREDICTOR"** submitted to **JNTU-H**, in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **INFORMATION TECHNOLOGY**, is an authentic record of my own work carried out from September 2022 to January 2023 under the supervision of **Mrs.A.Jyotsna, Assistant Professor, IT Dept., SVIT, Mahbub Campus.**

The matter embodied in this project report has not been submitted by me for the award of anyother degree.

Place: SECUNDERABAD

Date:

SYED ARHAAM DANIYAL

(19P71A1206)

NEHA YADAV

(19P71A1236)

SHAIK RAHEEMUDDIN

(20P75A1209)

INDEX

ABSTRACT

A car is the four-wheeler vehicle which plays a major role in life now-a- days. So here comes the query, what if a person buys the new car and he wants to sell the old one. For this our web application THE CAR PRICE PREDICTOR gives the best price to the car which the owner wanted to sell. This application predicts the value of the car which is to be sold. So that the predicted price helps the owner to take the right decision. The main focus of this application is developing a machine learning model which can accurately predict the price of a used car depending on its features, in order to make informed purchases. Here we implement and evaluate various learning techniques and methods on a data set. This data set consists of different sale prices of various manufacturers and models across the cities of India. Several factors like mileage, manufacturer, year of purchase, kilometers driven, etc.can influence the actual worth of your car. Based on the existing data, the main aim is to use machine learning algorithms for developing models to predict the prices for used cars. For this application we are mainly using the data set by doing a survey. The survey has the features like manufacturer company, model, year of purchase, kilometers driven, fuel type, number of owners.

This project aims to build a model to predict used cars' reasonable prices based on multiple aspects, including vehicle mileage, year of manufacturing, fuel consumption, transmission, road tax, fuel type, and engine size. This model can benefit sellers, buyers, and car manufacturers in the used cars market. Upon completion, it can output a relatively accurate price prediction based on the information that users input. The dataset used was scraped from listings of used cars. Various regression methods, including linear regression, polynomial regression, support vector regression, decision tree regression, and random forest regression, were applied in the research to achieve the highest accuracy. Before the actual start of model-building, this project visualized the data to understand the dataset better. The dataset was divided and modified to fit the regression, thus ensure the performance of the regression..

LIST OF FIGURES

1. INTRODUCTION

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 across cities. Linear regression also yielded satisfactory results, with the advantage of a significantly lower training time in comparison to the aforementioned methods.

Deciding whether a used car is worth the posted price when you see listings online can be difficult. Several factors, including mileage, make, model, year, etc. can influence the actual worth of a car. From the perspective of a seller, it is also a dilemma to price a used car appropriately. Based on existing data, the aim is to use machine learning algorithms to develop models for predicting used car prices.

The main focus of this application is developing a machine learning model which can accurately predict the price of a used car depending on its features, in order to make informed purchases. Here we implement and evaluate various learning techniques and methods on a data set. This data set consists of different sale prices of various manufacturers and models across the cities of India. A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars.

Our Application Gives User an Approximate Estimated Price of a Car Based on the Specifications of the User's Car. The Attributes which are considered are Year of Purchase, Present Showroom Price, Kilo meters Driven, Fuel Type, Seller Type, Transmission, Number of Owners.

2. SYSTEM REQUIREMENTS

2.1 Existing System

In the existing system, to predict the price of a four wheeler, 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. The major drawbacks of existing system is 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. There are no fast query retrieval methods.

2.1.1Disadvantages of Existing Systems:

1. 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.
2. It is highly complicated to get sufficient data sets that were spread widely all over the world.
3. The drawback of existing system is 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. There are no fast query retrieval methods.

2.2 Proposed System

The main focus of this project is developing a machine learning model which can accurately predict the price of a used car depending on its features, in order to make informed purchases. Here we implement and evaluate various learning techniques and methods on a data set. This data set consists of different sale prices of various manufacturers and models across the cities of India. A car price prediction has been a high-interest research area, as it requires noticeable effort and knowledge of the field expert. Considerable number of distinct attributes are examined for the reliable and accurate prediction. To build a model for predicting the price of used cars.

Our results show that Random Forest model with linear regression yield the best results, but are compute heavy. Linear regression also yielded satisfactory results, with the advantage of a significantly lower training time in comparison to the aforementioned methods.

2.2.1 Advantages of proposed System:

1) No Complexity.
2) Here the User need not to Load Multiple Interfaces.
3) It even Reduces the User Time.
4) This Application can be accessed on any Screen Size such as Laptop, Desktop,Tablet, Phones etc.
5) User can easily predict the value.

2.3System Requirements

Hardware Requirements

RAM : 4GB and Higher

Processor : i3processor and Higher

Hard Disk : 500GB

Software Requirements

Operating System: Windows 7,10 /macOS

Technologies: Python, ML Algorithims,Jupyter Notebook

Python Libraries: Numpy,Pandas,Seaborn

3. SYSTEM DESIGN

3.1 INTRODUCTION TO UML

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic, semantic and pragmatic rules. A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows:

3.1User Model View

In this module admin login into the system page.(using user name & password) Admin maintains the database and he provides the details of the donor to the recipient when the recipient is in need of blood.

3.1.1 Structural Model View

In this model, the data and functionality are arrived from inside the system. This model view models the static structures.

3.1.2 Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting he interactions of collection between various structural elements described in the user model and structural model view.

3.1.3 Implementation Model View

In this view, the structural and behavioral as parts of the system are represented as they are to be built.

3.1.4 Environmental Model View

In this view, the structural and behavioral aspects of the environment in which the system is to be implemented are represented.

3.2UML Diagrams

3.2.1 Use-Case Diagram

To model a system, the most important aspect is to capture the dynamic behavior.. To clarify a bit in details, dynamic behavior means the behavior of the system when it is running/operating.

So only static behavior is not sufficient to model a system rather dynamic behavior is more important than static behavior. In UML there are five diagrams available to model dynamic nature and use case diagram is one of them. Now as we have to discuss that the use case diagram is dynamic in nature there should be some internal or external factors for making the interaction.

These internal and external agents are known as actors. So use case diagrams are consisting of actors, use cases and their relationships. The diagram is used to model the system/subsystem of an application. A single use case diagram captures a particular functionality of a system. So to model the entire system numbers of use case diagrams are used.

Use case diagrams are used to gather the requirements of a system including internal and external influences. These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified. In brief, the purposes of use case diagrams can be as follows:

a. Used to gather requirements of a system.

b. Used to get an outside view of a system.

c. Identify external and internal factors influencing the system.

d. Show the interacting among the requirements are actors.

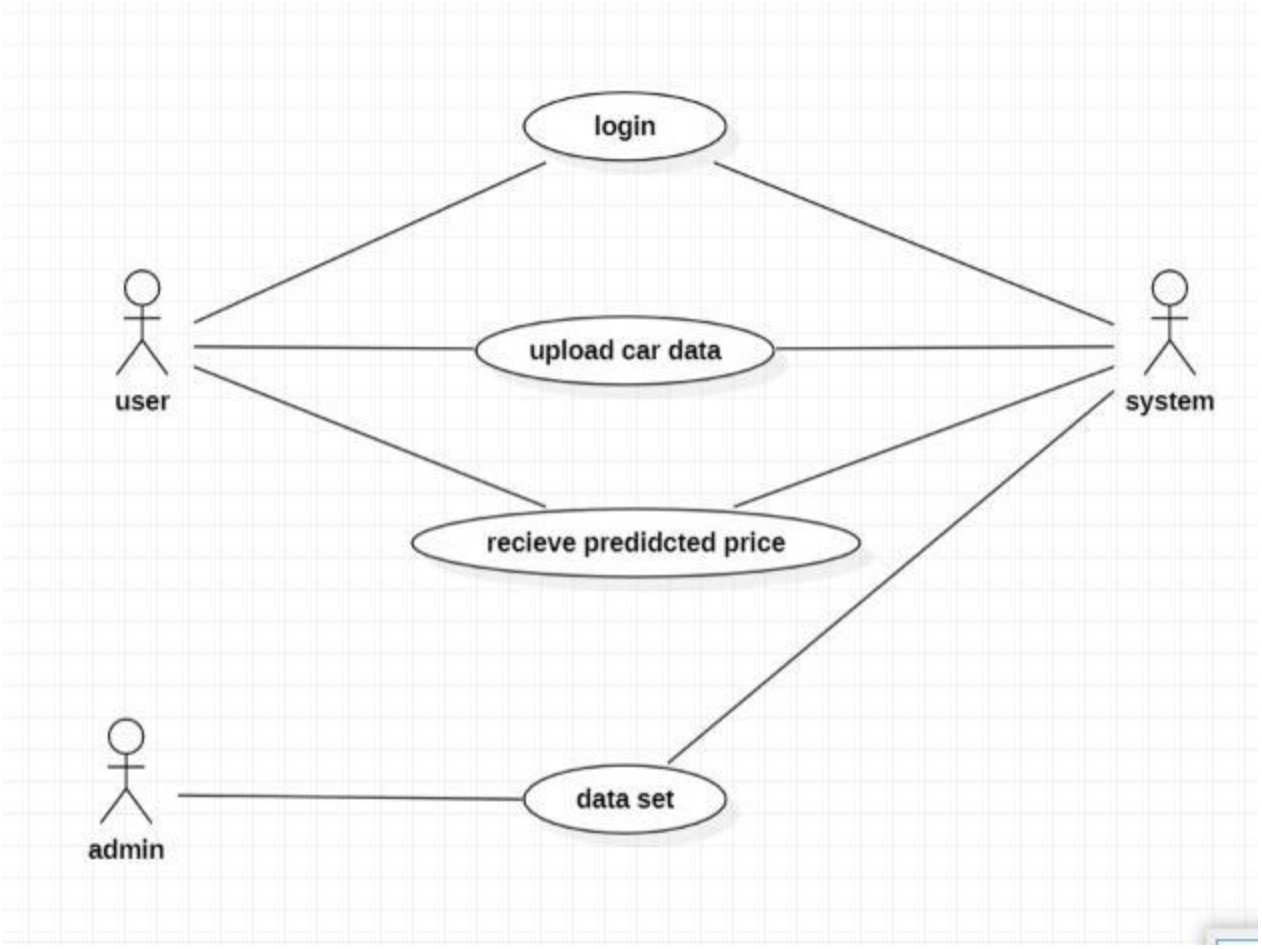

Figure 3.2.1 Use-Case Diagram

3.2.2 Sequence diagram

Sequence diagrams describe interactions among classes in terms of an exchange of messages over time. They're also called event diagrams. A sequence diagram is a good way to visualize and validate various runtime scenarios. These can help to predict how a system will behave and to discover responsibilities a class may need to have in the process of modeling a new system.

The aim of a sequence diagram is to define event sequences, which would have a desired outcome. The focus is more on the order in which messages occur than on the message per se.the majority of sequence diagrams will communicate what messages are sent and the order in which they tend to occur.

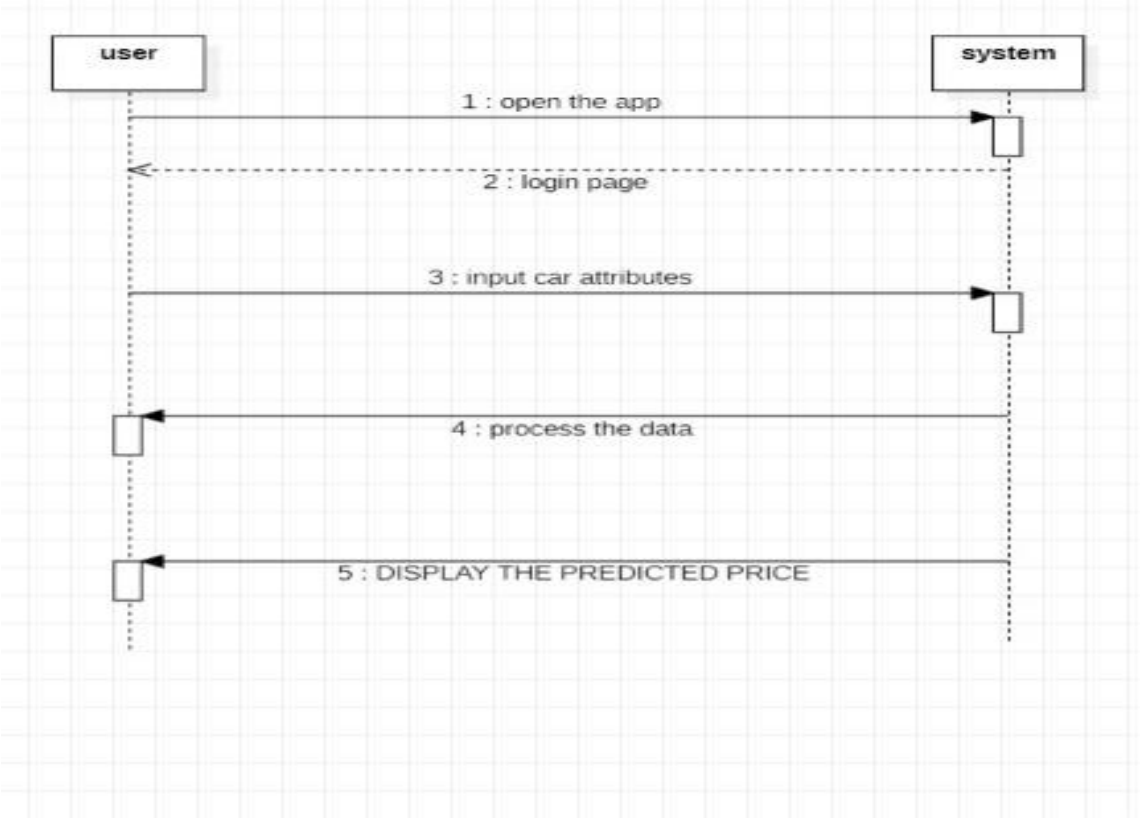

Figure 3.2.2 Sequence Diagram

3.2.3 Activity diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination. It is also suitable for modeling how a collection of use cases coordinate to represent business workflows

3.2.3.1 Identify candidate use cases, through the examination of business workflows

3.2.3.2 Identify pre- and post-conditions (the context) for use cases

3.2.3.3 Model workflows between/within use cases

3.2.3.4 Model complex workflows in operations on objects

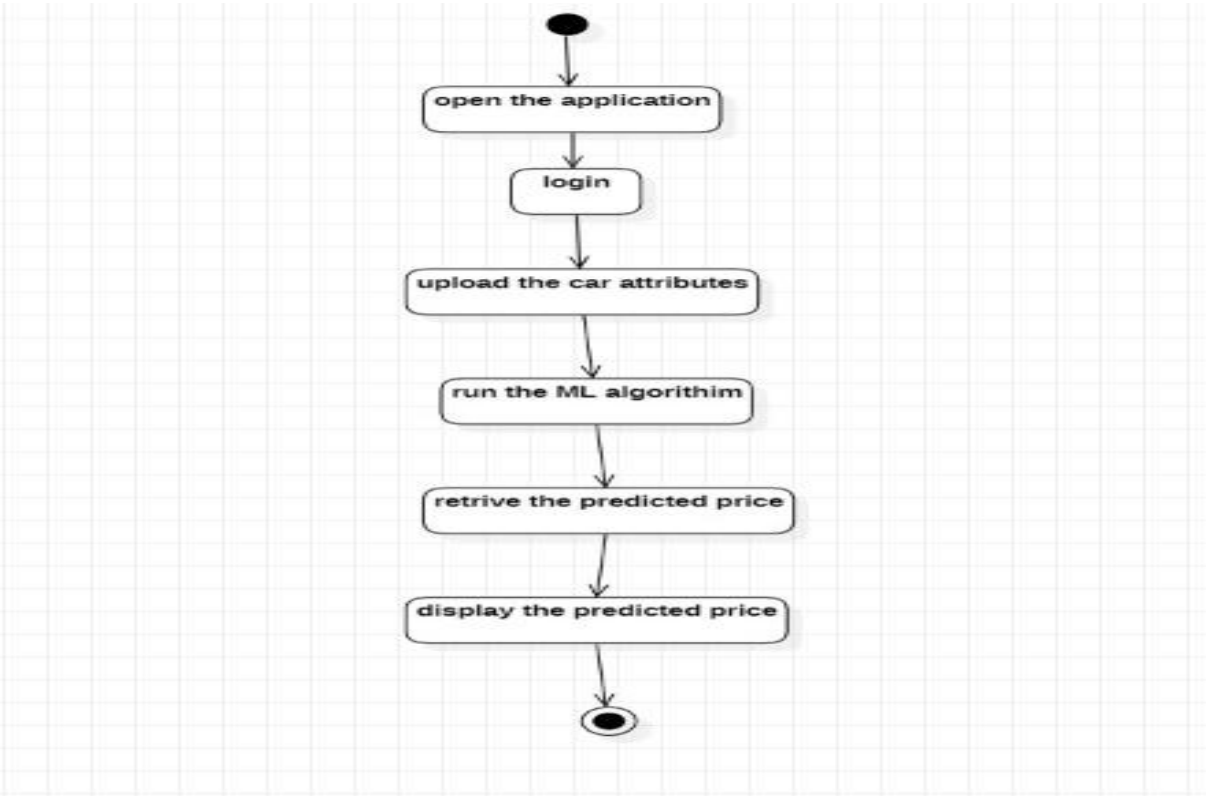

Figure 3.2.3.Activity Diagram

3.2.4 Class diagram

Class diagrams are the main building blocks of every object oriented methods. The classdiagram can be used to show the classes, relationships, interface, association, and collaboration. UML is standardized in class diagrams. Since classes are the building block of an application that is based on OOPs, so as the class diagram has appropriate structure to represent the classes, inheritance, relationships, and everything that OOPs have in its context. It describes various kinds of objects and the static relationship in between them.

The main purpose to use class diagrams are:

1. This is the only UML which can appropriately depict various aspects of OOP's concept.
2. Proper design and analysis of application can be faster and efficient.
3. It is base for deployment and component diagram.

4. Each class is represented by a rectangle having a subdivision of three: Compartmentsname, attributes and operation.

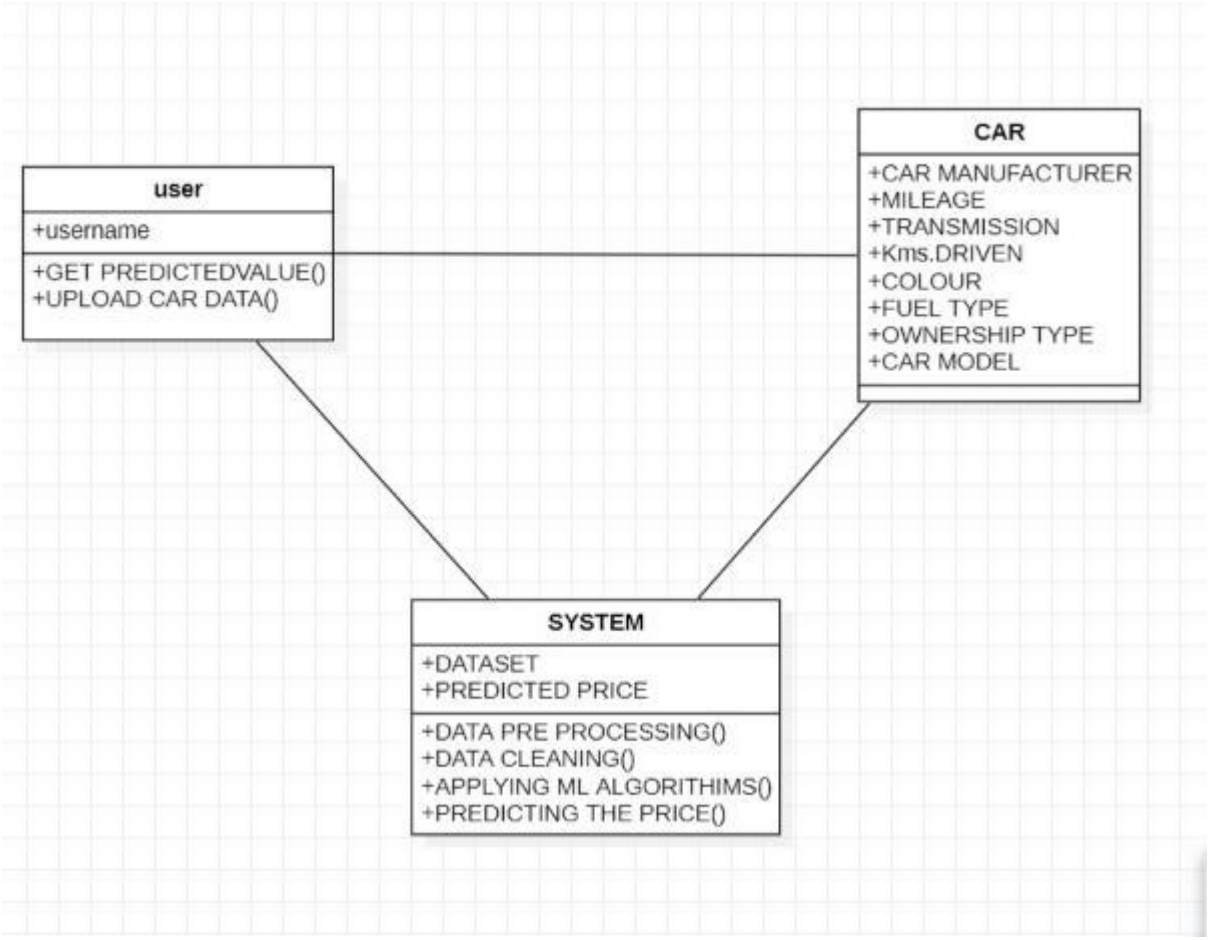

Figure 3.2.4.Class Diagram

4. IMPLEMENTATION DETAILS

➢ PYTHON

For our Application we used Python Programming Language, is a high-level, general-purpose and a very popular programming language. Python programming language (latest Python 3) is being used in web development, Machine Learning applications, along with all cutting-edge technology in Software Industry.

Python Programming Language is very well suited for Beginners, also for experienced programmers with other programming languages like C++ and Java.We also used Jupyter Notebook and Visual Studio Platforms. In simple words, JupyterNotebook is used to Clean and Train the Dataset and Create Models. Visual Studio is used forFront end Purpose.

Visual Studio Code is a code editor redefined and optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favourite platform - Linux, macOS, and Windows. It is a Community A fully-featured, extensible, free IDE for creating modern applications for Android, iOS, Windows, as well as web applications and cloud services. Jupyter Notebook is an open-source, web-based interactive environment, which allows you to create and share documents that contain live code, mathematical equations, graphics, maps, plots,

visualizations, and narrative text**.** It integrates with many programming languages like Python, PHP, R, C#, etc.

- **NumPy**

NumPy, is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, fourier transform, and matrices. NumPy was created in 2005 by Travis Oliphant. It is an open-source project and you can use it freely. NumPy stands for Numerical Python. In Python we have lists that serve the purpose of arrays, but they are slow to process. NumPy aims to provide an array object that is up to 50x faster than traditional Python lists. The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy. Arrays are very frequently used in data science, where speed and resources are very important.

- **Pandas**

The Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data. The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008. Pandas allows us to analyze big data and make conclusions based on statistical theories. Pandas can clean messy data sets, and make them readable and relevant. Relevant data is veryimportant in data science.

- **Matplotlib**

Matplotlib is a low-level graph plotting library in python that serves as a visualization utility. Matplotlib was created by John D. Hunter. Matplotlib is open source and we can

use itfreely. Matplotlib is mostly written in python, a few segments are written in C, Objective-C and JavaScript for Platform compatibility. Matplotlib is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.

In this article, we will learn about Python plotting with Matplotlib from basics to advance with the help of a huge dataset containing information about different types of plots and their customizations.

- **Seaborn**

Seaborn is a library mostly used for statistical plotting in Python. It is built on top of Matplotlib and provides beautiful default styles and colour palettes to make statistical plots more attractive. Seaborn comes with some customized themes and a high-level interface for customizing the looks of the graphs. Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and colour palettes to make statistical plots more attractive. It is built on the top of matplotlib libraryand also closely integrated to the data structures from pandas. Seaborn aims to make visualization the central part of exploring and understanding data. It provides dataset- oriented APIs, so that we can switch between different visual representations for same variables for better understanding of dataset.

- **StreamLit**

Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build wed apps. Streamlit allows you to create a stunning-looking

application with only a few lines of code.

From a Survey Dataset has been designed which is Uncleaned Dataset. We used Data Wrangling tasks, sometimes referred to as data munging, is the process of transforming and mapping data from one raw data form into another format with the intent of making it more appropriate and valuable for a variety of downstream purposes such as analytics. The goal of data wrangling is to assure quality and useful data. Data analysts typically spend the majority of their time in the process of data wrangling compared to the actual analysis of the data.

The process of data wrangling may include further munging, data visualization, data aggregation, training a statistical model, as well as many other potential uses. Data wrangling typically follows a set of general steps which begin with extracting the data in a raw form from the data source, "munging" the raw data (e.g., sorting) or parsing the data into predefined data structures, and finally depositing the resulting content into a data sink for storage and future use. There are 6 Steps of Data Wrangling:

1. Data Discovering.
2. Data Structuring.
3. Data Cleaning.
4. Data Enriching.
5. Data Validating.
6. Data Publishing.

Step 1: Data Discovering

The first step in the Data Wrangling process is Discovery**.** This is an all-encompassingterm for understanding or getting familiar with your data. You must take a look at the datayou have and think about how you would like it organized to make it easier to consume and analyze. So, you begin with an Unruly Crowd of Data collected from multiple sources in a wide range of formats. At this stage, the goal is to compile the Disparate, Siloed data sources and configure each of them so they can be understood and examined to find patterns and trends in the data.

Step 2: Data Structuring

When raw data is collected, it's in a wide range of formats and sizes. It has no definite structure, which means that it lacks an existing model and is completely disorganized. It needs to be restructured to fit in with the Analytical Model deployed by your business, and giving it a structure allows for better analysis. Unstructured data is often text-heavy and contains things such as Dates, Numbers, ID codes, etc. At this stage of the Data Wrangling process, the dataset needs to be parsed. This is a process whereby relevant information is extracted from fresh data. For example, if you are dealing with code scrapped from a website, you might parse HTML code, pull out what you need, and discard the rest. This will result ina more user-friendly spreadsheet that contains useful data with columns, classes, headings,and so on.

Step 3: Data Cleaning

Most people use the words Data Wrangling and Data Cleaning interchangeably. However, these are two very different processes. Although a complex process in itself, Cleaning is just a single aspect of the overall Data Wrangling process. For the most part, raw data comes with a lot of errors that have to be cleaned before the data can move on to the nextstage. Data Cleaning involves Tackling Outliers, Making Corrections, Deleting Bad Data completely, etc. This is done by applying algorithms to tidy up and sanitize the dataset.

Cleaning the data does the following:

- It removes outliers from your dataset that can potentially skew your results whenanalyzing the data.
- It changes any null values and standardizes the data format to improve quality andconsistency.
- It identifies duplicate values and standardizes systems of measurements, fixes structural errors and typos, and validates the data to make it easier to handle.

Step 4: Data Enriching

At this stage of the Data Wrangling process, you've become familiar with, and have a deep understanding of the data at hand. Combining your raw data with additional data from other sources such as internal systems, third-party providers, etc. will help you accumulate even more data points to improve the accuracy of your analysis. Alternatively, your goal might be to simply fill in gaps in the data. For instance, combining two databases of customer information where one contains customer addresses, and the other one doesn't. Enriching the data is an optional step that you only need to take if your current data doesn't meet your requirements.

Step 5: Data Validating

Validating the data is an activity that services any issues in the quality of your data so they can be addressed with the appropriate transformations.

The rules of data validation require repetitive programming processes that help to verify the following:

- Quality
- Consistency
- Accuracy
- Security
- Authenticity

This is done by checking things such as whether the fields in the datasets are accurate, and if attributes are normally distributed. Preprogramed scripts are used to compare the data'sattributes with defined rules. This is a great example of the overlap that sometimes happens between Data Cleaning and Data Wrangling.

Step 6: Data Publishing

By this time, all the steps are completed and the data is ready for analytics. All that's

left is to publish the newly Wrangled Data in a place where it can be easily accessed and usedby you and other stakeholders. You can deposit the data into a new architecture or database.

4.1 Process Structuring

There are five steps in Process Structuring, means gaining insight into the way your organisation operates. It introduces a layering in the way of working that gives you and your employees insight into which processes there are and how these relate to each other. Which method to choose for the structuring of your processes depends on your organisation. They are:

1. Data Collection
2. Pre Processing
3. Data Analysis
4. Application of Algorithm
5. Evaluating The Models

- DATA COLLECTION

Data collection is the procedure of collecting, measuring and analyzing accurate insights for research using standard validated techniques. A researcher can evaluate their hypothesis on the basis of collected data. In most cases, data collection is the primary and most important step for research, irrespective of the field of research. The approach of data collection is different for different fields of study, depending on the required information.

- PRE PROCESSING

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

➢ DATA ANALYSIS

Data Analysis is a process of inspecting cleansing, transforming, and modelling data with the goal of discovering useful information, informing conclusions, and supporting decision- making. Data analysis has multiple facets and approaches, encompassing diverse techniques under a variety of names, and is used in different business, science, and social science domains. In today's business world, data analysis plays a role in making decisions more scientific and helping businesses operate more effectively.

➢ APPLICATION OF ALGORITHMS

An Algorithm in data mining (or machine learning) is a set of heuristics and calculations that creates a model from data. To create a model, the algorithm first analyzes the data you provide, looking for specific types of patterns or trends. The algorithm uses the results of this analysis over many iterations to find the optimal parameters for creating the mining model. These parameters are then applied across the entire data set to extract actionable patterns and detailed statistics.

➢ EVALUATING THE MODELS

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring. It is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluating model performance with the data used for training is not acceptable in data science because it can easily generate overoptimistic and overfitted models.

PROCESS STRUCTURING MODEL

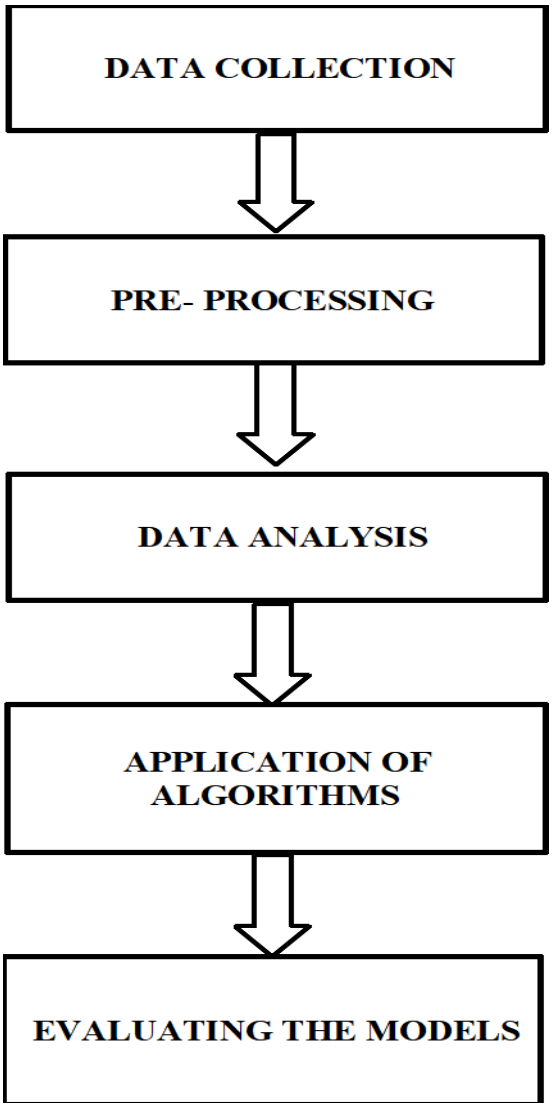

Figure.4.0 process structuring model

	Selling_Price	Present_Price	Kms_Driven	Owner	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	5.59	27000	0	0	1	0	1
1	4.75	9.54	43000	0	1	0	0	1
2	7.25	9.85	6900	0	0	1	0	1
3	2.85	4.15	5200	0	0	1	0	1
4	4.60	6.87	42450	0	1	0	0	1
...	...	...	...	...	...	...	...	...
296	9.50	11.60	33988	0	1	0	0	1
297	4.00	5.90	60000	0	0	1	0	1
298	3.35	11.00	87934	0	0	1	0	1
299	11.50	12.50	9000	0	1	0	0	1
300	5.30	5.90	5464	0	0	1	0	1

TABLE : STRUCTURED DATASET FROM CLEANED DATASET

This is a Structured dataset, which is consisting of 300rows and 8 columns selling_price, present_price, kms_driven, owner, fuel_type_diesel, fuel_type_petrol, seller_type_individual, transmission_manual.

Here by using a syntax as get_dummies which means if a car is of petrol fuel type. Then in fuel_type_diesel it will be given as 0 and fuel_type_petrol will be given as 1and vice versa. If not Petrol and Diesel that means if it is a CNG fuel type then both fuel_type_diesel and fuel_type_petrol will be given as 0 and if the Seller Type is not an individual then the column is given as 0 when the transmission type is manual, it is given as 1.Structured Data Set means the Input Data Set that has been annotated, tagged or otherwise processed. Structured dataset is dataset that has been organized into a formatted repository, typically a database, so that its elements can be made addressable for more effective processing and analysis.

5.CODE SNIPPETS

```
#importing the libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
#reading the dataset
data=pd.read_csv('car data.csv')
print(data.shape)
data.head()
print('Unique elements in Seller_Type are',data['Seller_Type'].unique())
print('Unique elements in Fuel_Type are',data['Fuel_Type'].unique())
print('Unique elements in Transmission are',data['Transmission'].unique())
print('Unique elements in Owner are',data['Owner'].unique())
print('Unique elements in Year are',data['Year'].unique())
print('Unique elements in Car_Name are',data['Car_Name'].nunique())
#98 unique elements
#so, rather than encoding it, we can just drop this columbn as it doesn' make sense
data.describe()
dataset=data[['Year','Selling_Price','Present_Price','Kms_Driven','Fuel_Type','Seller_Ty
pe','Transmission','Owner']]
dataset.head()
ataset['Present_Year']=2020
dataset['Number_of_Years_Old']=dataset['Present_Year']- dataset['Year']
dataset.head()
dataset.drop(labels=['Year', 'Present_Year'],axis=1,inplace=True)
dataset.head()
#select categorical variables from then dataset, and then implement categorical encoding
for nominal variables
Fuel_Type=dataset[['Fuel_Type']]
Fuel_Type=pd.get_dummies(Fuel_Type, drop_first=True)
```

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

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

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

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

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

print(model.feature_importances_)
```

```
#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()
X=dataset.iloc[:,:-1].values
y=dataset.iloc[:,-1].values
rom sklearn.model_selection import cross_val_score
from sklearn import metrics
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
#from sklearn.model_selection import RandomizedSearchCV
#from sklearn.model_selection import GridSearchCV
#from sklearn.model_selection import StratifiedKFold
#kfold = StratifiedKFold(n_splits=3)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)
#Decision Tree Regressor
from sklearn.tree import DecisionTreeRegressor
dt_reg = DecisionTreeRegressor(random_state = 0)
dt_reg.fit(X_train, y_train)
y_pred=dt_reg.predict(X_test)

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

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

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

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

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

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

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

lt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()


#Random Forest Regression
from sklearn.ensemble import RandomForestRegressor
rf_reg =
RandomForestRegressor(n_estimators=400,min_samples_split=15,min_samples_leaf=2
,
max_features='auto', max_depth=30)
rf_reg.fit(X_train, y_train)
y_pred=rf_reg.predict(X_test)

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

Accuracy

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

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

mse=mean_squared_error(y_test, y_pred)
print("Mean Squared Error:" , mse)
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))

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

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

plt.scatter(y_test, y_pred, alpha = 0.5)
plt.xlabel("y_test")
plt.ylabel("y_pred")
plt.show()

import pickle
pickle.dump(vot_reg, open("vot_reg.pkl", "wb"))

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

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

6. METHODOLOGY

Here In this Application, we used three Models LinearRegression, ShuffleSplit, and DecisionTreeRegressor. Calculating the Accuracy from the above three Models we got the highest acuracy in RandomForestRegressor, is an ensemble learning technique. In ensemble learning, you take multiple algorithms or same algorithm multiple times and put together a model that's more powerful than the original. Prediction based on the trees is more accurate because it takes into account many predictions. This is because of the average value used. These algorithms are more stable because any changes in dataset can impact one tree but not the forest of trees. Which has got a Highest Accuracy of 92.

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

ShuffleSplit will randomly sample your entire dataset during each iteration to generate a training set and a test set. The test_size and train_size parameters control how large the test and training test set should be for each iteration. Since you are sampling from the entire dataset during each iteration, values selected during one iteration, could be selected again during another iteration.

Decision Tree is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs, and utility. Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables. These Models are imported from sklearn, Scikit-learn (Sklearn) is the most useful and robust library for

machine learning in Python. It provides a selection of efficient tools for machine learning and statisticalmodeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy**,** SciPy and Matplotlib. Packages such as pandas, numpy, matplotlib, pickle, seaborn are Imported.

7.TESTING

In order to uncover present in different phases we have the concept of levels of testing.

- **Code Testing**

This examines the logic of the program. For example, the logic for updating various sample data and with the sample files and directories were tested and verified.

- **Unit testing**

In the unit testing we test each module individually and integrate with the overall system. Unit testing focuses verification efforts on the smallest unit of software design in the module. This is also known as module testing. The module of the system is tested separately. This testing is carried out during the programming stage itself. In the testing step, each module is found to work satisfactorily as regard to expected output from the module. There are some validation checks forfields also. For example, the validation check is done for varying the user input given by the user which validity of the data entered. It is very easy to find errors debug the system. Each Module can be tested using the following two Strategies.

1. Black Box Testing

2. White Box Testing

7.1 Black Box Testing

What is Black Box Testing?

Black box testing is a software testing technique in which functionality of the software under test (SUT) is tested without looking at the internal code structure, implementation details and knowledge of internal paths of the software This type of testing is based entirely on the software requirements and specifications.

In Black Box Testing we just focus on inputs and output of the software system without bothering about internal knowledge of the software program.

Fig 7.1 Black box testing

The above Black Box can be any software system you want to test. For example: an operating system like Windows, a website like Google, a database like Oracle or even your own custom application. Under Black Box Testing, you can test these applications by just focusing on the inputs and outputs without knowing their internal code implementation.

Black box testing - Steps

- Here are the genetic steps followed to carry out any type of Black Box Testing.
- Initially requirements and specifications of the system are examined.

- Tester chooses valid inputs (positive test scenario) to check whether SUTprocesses
- them correctly. Also, some invalid inputs (negative test scenario) are chosen toverify
- that the SUT is able to detect them.
- Tester determines expected outputs for all those inputs.
- Software tester constructs test cases with the selected inputs.
- The test cases are executed.
- Software tester compares the actual outputs with the expected outputs.
- Defects if any are fixed and re-tested

Types of Black Box Testing

- There are many types of Black Box Testing but following are the prominentones –
- Functional testing - This black box testing type is related to functional requirements of a system, it is done by software testers.
- Non-functional testing - This type of black box testing is not related to testing of a specific functionality, but non-functional requirements such as performance, scalability, usability
- Regression testing -Regression testing is done after code fixes, upgrades or any other system maintenance to check the new code has not affected the existing code.

7.2 White Box Testing

White Box testing of software solution's internal ending and infrastructure. It focuses primarily on strengthening security, the flow of inputs and outputs through the application, and improving design and usability. White box testing is also known as clear, open, structural, and glass box testing.

It is one of parts of “box testing” approach of software testing. Its counterpart black box testing from an external or end-user type perspective. On the other hand, White box testing is based on the inner workings of an application and revolves around internal testing. The term “white box” was used because of the see-through box concept. The clear box or white box names symbolizes the ability to see through the software’s outer shell (or “box”) into its inner workings. Likewise, the “black box” in “black box testing” symbolizes not being able to see the inner workings of the software so that the only end-user experience can be tested.

What do you verify in White Box Testing?

The testing can be done at system, integration and unit levels of software development. One of the basic goals of white box testing is to verify a working flow for an application. It involves testing a series of predefined inputs against expected or desired outputs so that when a specific input does not result in the expected output, you have encountered a bug.

8.OUTPUT

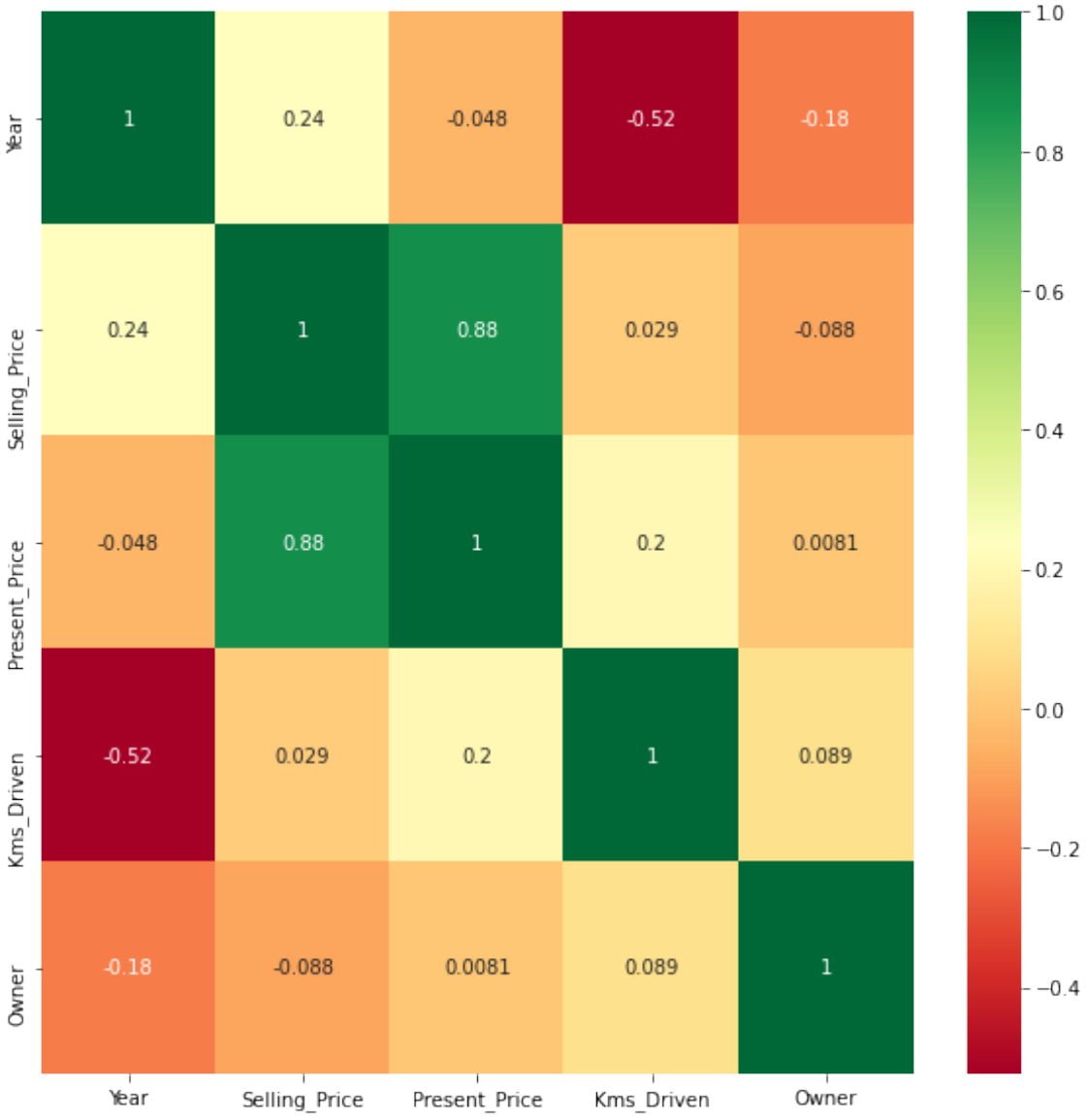

Figure.8.1 output after correlation

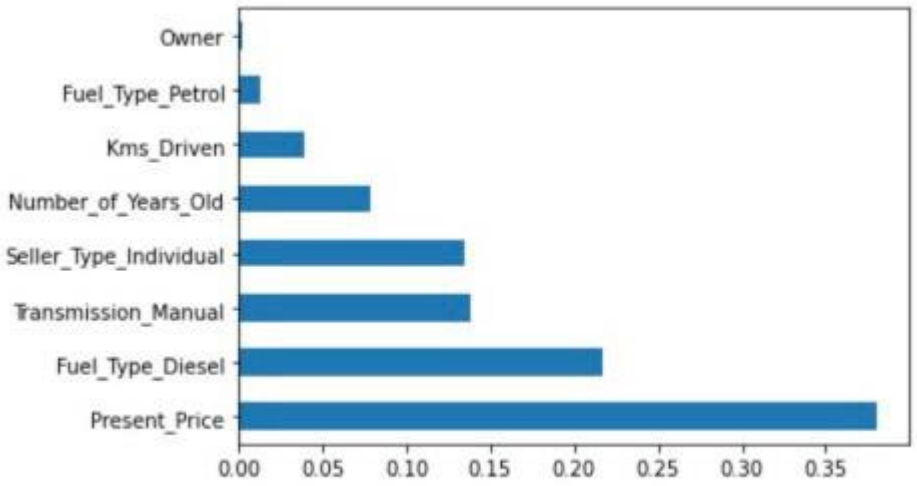

Figure.8.2 output after data visualisation

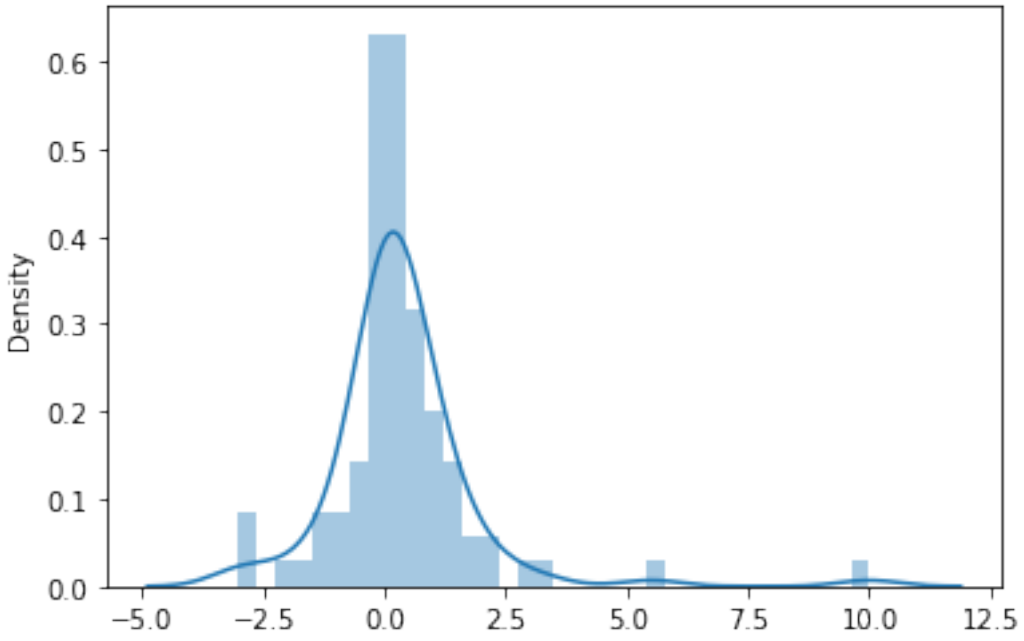

Figure.8.3 output of decision tree

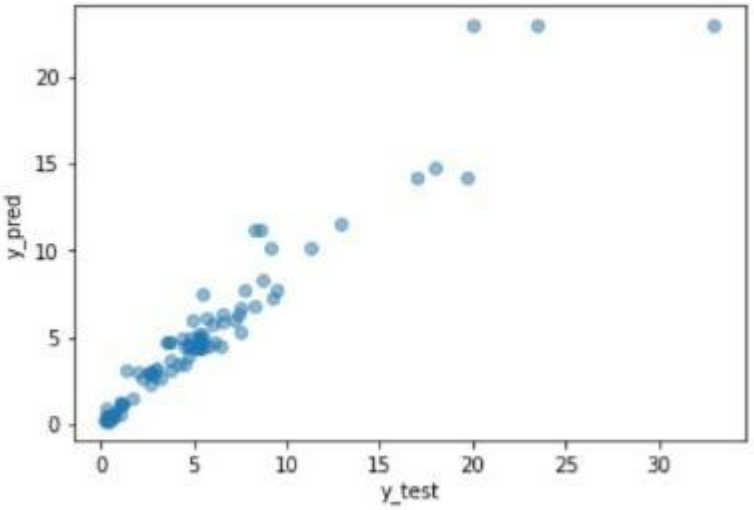

Figure.8.4 output of decision tree

dj

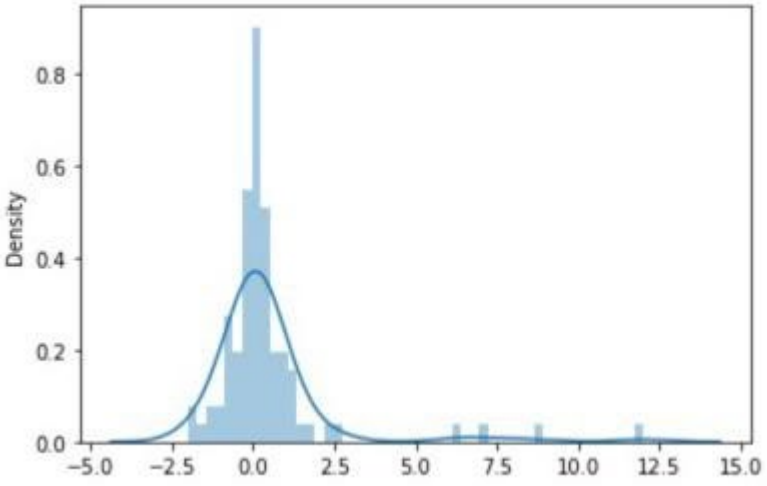

Figure.8.5 output after random forest regressor

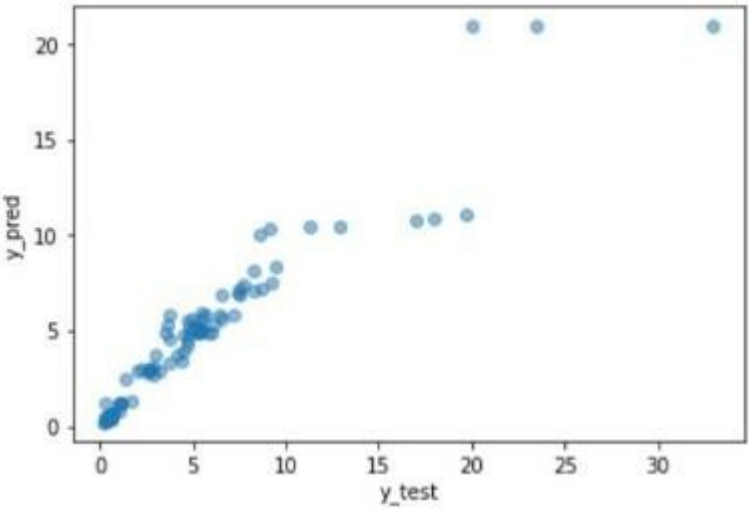

Figure 8.6.Output of random forest regressor

CONCLUSION

The increased prices of new cars and the financial incapability of the customers to buy them, Used Car sales are on a global increase. Therefore, there is an urgent need for a Used Car Price Prediction system which effectively determines the worthiness of the car using a variety of features. The proposed system will help to determine the accurate price of used car price prediction. This paper compares 3 different algorithms for machine learning : Decision tree, Random forest regressor.

Car price prediction can be a challenging task due to the high number of attributes that should be considered for the accurate prediction. The major step in the prediction process is collection and preprocessing of the data. In this research, PHP scripts were built to normalize, standardize and clean data to avoid unnecessary noise for machine learning algorithms.

Using data mining and machine learning approaches, this project proposed a scalable framework for Car Price Prediction. An efficient machine learning model is built by training, testing, and evaluating three machine learning regressors named Random Forest Regressor, Linear Regression. As a result of pre-processing and transformation, Random Forest Regressor came out on top with 95% accuracy.

REFERENCES

[1] Listiani, M. (2009). Support vector regression analysis for price prediction in a car leasing application (Doctoral dissertation, Master thesis, TU Hamburg-Harburg)

[2] 3.2.4.3.1. sklearn.ensemble.RandomForestClassifier — scikit-learn 0.19.2 documentation. (n.d.). Retrieved from: http://scikitlearn.org/stable/modules/generated/sklearn.ensemble. .RandomForestClassifier.html [accessed: August 30, 2018].

[3] Information regarding machine learning techniques and algorithmshttps://en.wikipedia.org/wiki/Machine_learning

[4] S. Pudaruth, "Predicting the price of used cars using machine learning techniques," Int. J. Inf. Comput. Technol, vol. 4, no. 7, pp. 753–764, 2014. 183 Authorized licensed use limited to: Carleton University. Downloaded on May 29,2021 at 09:56:13 UTC from IEEE Xplore.

[5] Cars price dataset- https://www.kaggle.com/datasets/avikasliwal/used-cars-price-prediction?select=traindata.csv.

[6] no. 22, pp. 12 693–12 700, 2017. [12] E. Gegic, B. Isakovic, D. Keco, Z. Masetic, and J. Kevric, "Car price prediction using machine learning techniques," 2019.

[7] https://scikit-learn.org/stable/modules/classes.html: Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[8] Python for Data Analysis (2nd Edition) – by Wes McKinney.

[9] Machine Learning Web Development | medium.com