Project Report

on

Car Resale Price Predictor

(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

Under the Supervision of
Dr D Rajesh Kumar (Assistant Professor)

Submitted By:
Ankur
Singh(18021180027)
Bhanu Kalia
(18021011543)

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING GALGOTIAS UNIVERSITY, GREATER NOIDA
INDIA

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING
GALGOTIAS UNIVERSITY, GREATER NOIDA

CANDIDATE’S DECLARATION

We hereby certify that the work which is being presented in the project, entitled **“Car Resale Price Prediction”** in partial fulfillment of the requirements for the award of the **BACHELORS OF TECHNOLOGY** submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of month, July 2021 to December 2021, under the supervision of Mr. Hradesh Kumar (Assistant Professor), Department of Computer Science and Engineering, Galgotias University, Greater Noida

The matter presented in the project has not been submitted by us for the award of any other degree of this or any other places.

Ankur Singh, 1802118002

Bhanu Kalia, 18021011543

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr D Rajesh Kumar

Assistant Professor

CERTIFICATE

The Final Project Viva-Voce examination of Ankur Singh (18021180027) and Bhanu Kalia (18021011543) has been held on ________________________ and his/her work is recommended for the award of Bachelors of Technology.

Signature of Examiner(s) **Signature of Supervisor(s)**

Signature of Project Coordinator **Signature of Dean**

Date: December, 2021

Place: Greater Noida

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our guide Mr. Hradesh Kumar, as well as our esteemed university who gave us the golden opportunity to do this wonderful project on the topic Classifying tweets based on climate change stance which also helped us in doing a lot of Research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame.

Ankur Singh (18021180027)

Bhanu Kalia (18021011543)

ABSTRACT

Car Resale is a problem that is faced by almost everyone who thinks about sale of his used car , there can be different ways to rate a car but mostly people trust brokers and there is no mathematical way to predict this based on the previously sold cars. There are certain apps where you can sale your used products but there is no efficient method to suggest what should be actual price to resale the car.

The designed system deals with using previous data from various sources to predict price of the car and will use different machine learning algorithms that will create a regression model and will help in predicting price of car.

This Project will also be deployed with help of flask so it can be used in real time with a user interface as well, this will be a good project for daily use of people who face a dilemma of choosing right price for their vehicle before selling it again, to built the model we will be using different ML models like SVM , LR , Trees etc

Keywords—Resale , Prediction , Linear Regression,SVM

TABLE OF CONTENTS

CHAPTER 1
Introduction

1.1 PURPOSE

Car resale is a problem faced by people almost on daily basis , in this the major problem faced is to decide the best price of car and based on what variables should be the price of car be decided and how to decide those factors and how to make a standard market evaluation of car.

Car resale is multi million dollar worth business world-wide and the most important of aspect of car sale is solely based on third party evaluation which is usually done by dealers and brokers without any standard measure so we need to evaluate previously sold cars and its data to create a better system which will be helpful in producing a better system for car selling.

With the advent of ML technologies and data science , this is the field where it can be used so much efficiently to produce a system for better and efficient application of resale prediction product.

The purpose of undertaking this project is to produce a system and web app that will help people to get a better valuation of their product and to use it in their daily life

.

1.2 PRODUCT SCOPE

There are many products out there that provide the platform to provide the customers to buy second hand products but there is not a single system which recommends a price predictor for the product so we will be trying to produce a tool that will help us to get the predicted value for the product and which will be a easy to use web app based on very easy and very little information.

1.3 PROBLEM DEFINITION

Finally it comes down to a regression problem which we have to solve to get a better understanding of resale market and the tool that we will create to provide price and to pin point the major factors that will help us to get the better resale price based on certain conditions of vehicle.

Chapter 2
Literatue Survey

2.1 Hardware Requirements

The section of hardware configuration is an important task related to software development, insufficient random-access memory may affect adversely on the speed and efficiency of the entire system. The process should be powerful to handle the entire operations. The hard disk should have sufficient capacity to store the file and application.

Processor : Intel Core i3 or more RAM : 4GB or more
Hard disk : 10GB or more (depending on input file size)
Peripherals : Keyboard, Compatible mouse
Cache Memory : L2-1 MB
GPU : Intel HD Graphics or Nvidia chip for better performance Monitor Resolution : 1024*768 or 1336*768 or 1280*1024 3.2

2.2 Software Requirments

A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the system, function to be carried out the performance level to be obtained and corresponding interfaces to be established

2.2.1 Python

Python Language Introduction Python is a general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is a high-level, interpreted, interactive and object-oriented scripting language.

Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

• Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to

PERL and PHP.
• Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
• Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
• Python is a Beginner's Language − Python is a great language for the beginnerlevel programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games. History of Python Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands. Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages. Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL). Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress. Python Features Python's features include –
• Easy-to-learn − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
• Easy-to-read − Python code is more clearly defined and visible to the eyes.
• Easy-to-maintain − Python's source code is fairly easy-to-maintain.

2.2.2 Pandas

Pandas is an open source python library providing high performance data manipulation and analysis tool using its data structure. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Pandas can be used to accomplish five important steps in the processing and analysis of data, irrespective of the origin of data — load, prepare, manipulate, model, and analyse. Python with Pandas is used in many fields like academic and commercial domains including finance, economics, Statistics, analytics, etc. Library features of Pandas:
• Fast and efficient DataFrame object with default and customized indexing.
• Tools for loading data into in-memory data objects from different file formats.
• Very useful in data manipulation.
• Data alignment and integrated handling of missing data.
• Reshaping and pivoting of data sets.
• Label-based slicing and sub setting of large data sets.
• Columns from Data Structure can be deleted or inserted.

2.2.3 Numpy

NumPy is a library for the Python programming language, adding support for large, multidimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The ancestor of NumPy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created NumPy by incorporating features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors. NumPy targets the CPython reference implementation of Python, which is a nonoptimizing bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

2.2.4 Scikit

Learn Scikit-learn (formerly scikits learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR. Some popular groups of models provided by scikit-learn include:

- Clustering: for grouping unlabeled data such as KMeans.
- Cross Validation: for estimating the performance of supervised models on unseen data.
- Datasets: for test datasets and for generating datasets with specific properties for investigating model behaviours.
- Dimensionality Reduction: for reducing the number of attributes in data for summarization, visualization and feature selection such as Principal component analysis.
- Ensemble methods: for combining the predictions of multiple supervised models.
- Feature extraction: for defining attributes in image and text data.
- Feature selection: for identifying meaningful attributes from which to create supervised models.
- Parameter Tuning: for getting the most out of supervised models.
- Manifold Learning: For summarizing and depicting complex multi-dimensional data.
- Supervised Models: a vast array not limited to generalized linear models, discriminate analysis, naive Bayes, lazy methods, neural networks, support vector machines and decision trees

2.2.5 Flask

Flask is a web application framework written in Python. It was developed by Armin Ronacher, who led a team of international Python enthusiasts called Poocco. Flask is based on the Werkzeg WSGI toolkit and the Jinja2 template engine.Both are Pocco projects
Flask is often referred to as a microframework. It is designed to keep the core of the application simple and scalable.
Instead of an abstraction layer for database support, Flask supports extensions to add such capabilities to the application.
Flask is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries
It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.
Jinja, also by Ronacher, is a template engine for the Python programming language and is licensed under a BSD License. Similar to the Django web framework, it handles templates in a sandbox.

WHY FLASK ??
Development server and debugger
Integrated support for unit testing
RESTful request dispatching
Uses Jinja templating
Support for secure cookies (client side sessions)
100% WSGI 1.0 compliant
Unicode-based
Extensive documentation
Google App Engine compatibility
Extensions available to enhance features desired

2.2.6 Pickle

Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling. We can converts the byte stream (generated through pickling) back into python objects by a process called as unpickling.

Why Pickle?: In real world sceanario, the use pickling and unpickling are widespread as they allow us to easily transfer data from one server/system to another and then store it in a file or database.

Precaution: It is advisable not to unpickle data received from an untrusted source as they may pose security threat. However, the pickle module has no way of knowing or raise alarm while pickling malicious data.

Only after importing pickle module we can do pickling and unpickling.

2.2.6 Seaborn

Seaborn is an amazing visualization library for statistical graphics plotting in Python. It provides beautiful default styles and color palettes to make statistical plots more attractive. It is built on the top of matplotlib library and 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.

2.2.7 GradientBoostRegressor

Gradient Boosting for regression.

GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage a regression tree is fit on the negative gradient of the given loss function.

2.2.8 Train Test Split

Before discussing train_test_split, you should know about Sklearn (or Scikit-learn). It is a Python library that offers various features for data processing that can be used for classification, clustering, and model selection.

Model_selection is a method for setting a blueprint to analyze data and then using it to measure new data. Selecting a proper model allows you to generate accurate results when making a prediction.

To do that, you need to train your model by using a specific dataset. Then, you test the model against another dataset.

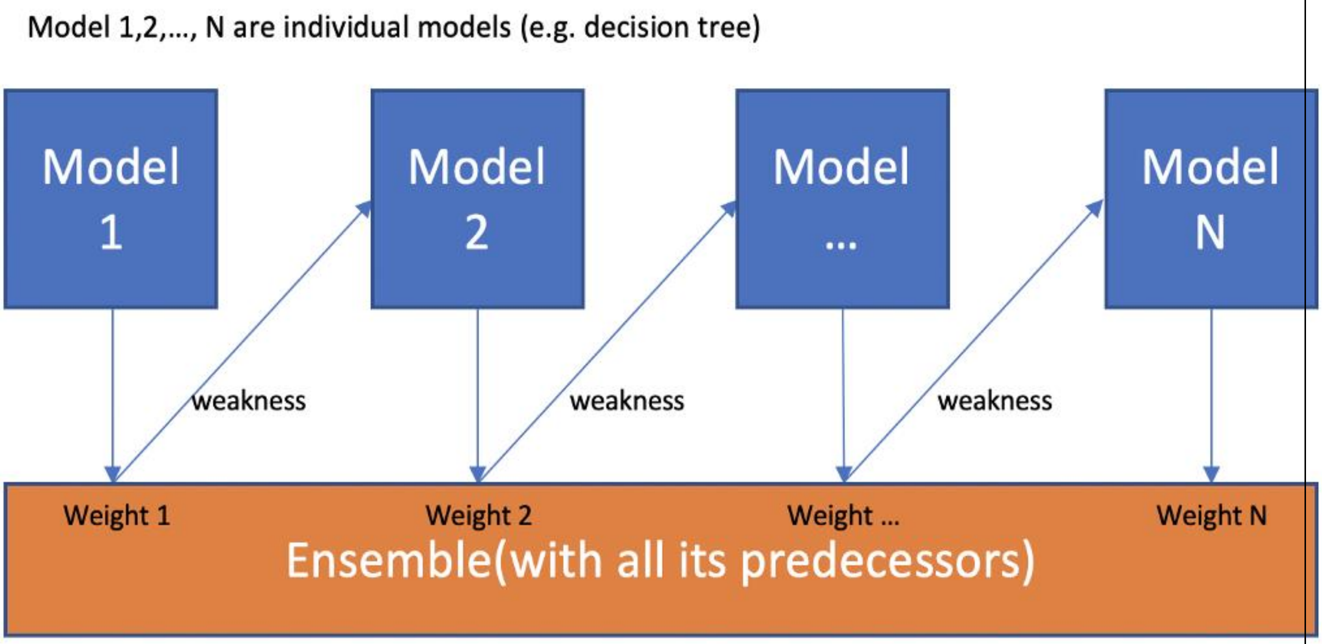
2.2.9 Understanding Regression Algorithms

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables).

Regression helps investment and financial managers to value assets and understand the relationships between variables, such as commodity prices and the stocks of businesses dealing in those commodities.

2.2.10 Understanding Gradient Boosting Regression Algorithm

Unlike many ML models which focus on high quality prediction done by a single model, boosting algorithms seek to improve the prediction power by training a sequence of weak models, each compensating the weaknesses of its predecessors.



Gradient Boosting

Gradient boosting approaches the problem a bit differently. Instead of adjusting weights of data points, Gradient boosting focuses on the difference between the prediction and the ground truth.

$$To\ improve\ a\ model\ F: We\ want\ to\ \text{minimize}\ Loss(Y, F(X))$$
$$The\ loss\ function\ L = func(F(X_1), F(X_2), \dots, F(X_n), Y)$$
$$Minimization\ is\ performed\ by\ fitting\ an\ estimator\ H\ on\ \left(X_i, \frac{\partial L}{\partial X_i}\right) \forall i$$
$$F(X) + H(X)\ is\ an\ approximation\ of\ gradient\ descent\ \hat{F}(X_i) = F(X_i) - \frac{\partial L}{\partial F(X_i)}$$

Gradient boosting requires a differential loss function and works for both regression and classifications. I’ll use a simple Least Square as the loss function (for regression). The algorithm for classifications shares the same idea, but the math is slightly more complicated.(J. Friedman, Greedy Function Approximation: A Gradient Boosting Machine)

$$fit\ estimator\ F^1$$
$$for\ i\ in\ [1, M] // \text{M weak estimators}$$
$$Loss^i = \sum_{j=1}^{n} \left(Y_j - F^i(X_j)\right)^2 \ // \text{ loss in } i^{th} \text{ iteration}$$
$$calculate\ neg\ gradient: -\frac{\partial L^i}{\partial X_j} = -\frac{2}{n} * \left(Y_j - F^i(X_j)\right) \forall i$$
$$Fit\ a\ weak\ estimator\ H^i\ on\ \left(X, \frac{\partial L}{\partial X}\right)$$
$$// \rho\ changes\ the\ step\ size$$
$$Prediction: F^m(X) = F^i(X) + \rho * H^i(X) = F^1 + \rho * \sum_{i=1}^{m} H^i(X)$$

Finding Best Parametera For Model

```
GradientBoostingRegressor(alpha=0.9, criterion='friedman_mse', init=None,
             learning_rate=0.1, loss='ls', max_depth=3, max_features=None,
             max_leaf_nodes=None, min_impurity_decrease=0.0,
             min_impurity_split=None, min_samples_leaf=1,
             min_samples_split=2, min_weight_fraction_leaf=0.0,
             n_estimators=100, n_iter_no_change=None, presort='auto',
             random_state=None, subsample=1.0, tol=0.0001,
             validation_fraction=0.1, verbose=0, warm_start=False)
```

Regression:
loss : {‘ls’, ‘lad’, ‘huber’, ‘quantile’}, optional (default=’ls’)
Classification:
loss : {‘deviance’, ‘exponential’}, optional (default=’deviance’)
The rest are the same
learning_rate : float, optional (default=0.1)
n_estimators : int (default=100)
Gradient boosting is fairly robust to over-fitting so a large number usually results in better performance.
subsample : float, optional (default=1.0)
The fraction of samples to be used for fitting the individual base learners. If smaller than 1.0 this results in Stochastic Gradient Boosting. subsample interacts with the parameter n_estimators. Choosing subsample < 1.0 leads to a reduction of variance and an increase in bias.
criterion : string, optional (default=”friedman_mse”)
The function to measure the quality of a split.

2.2.11 Data Used

We have used data of cardekho website which was available on kaggle and then we used that on our system and then further more modifications on data to create data in presentable form and in the form that can be fed into ML model

Original Data :

	Car_Name	Year	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner
0	ritz	2014	3.35	5.59	27000	Petrol	Dealer	Manual	0
1	sx4	2013	4.75	9.54	43000	Diesel	Dealer	Manual	0
2	ciaz	2017	7.25	9.85	6900	Petrol	Dealer	Manual	0
3	wagon r	2011	2.85	4.15	5200	Petrol	Dealer	Manual	0
4	swift	2014	4.60	6.87	42450	Diesel	Dealer	Manual	0

Since Data can not be passed in this format , we need to convert data into some other format so that it can be passed into ML algorithm, we will use dummy variable technique to improvise the data

Pandas Dummy Variable : **pandas.get_dummies()** is used for data manipulation. It converts categorical data into dummy or indicator variables.

These columns will be used for making data based model for ML

	Selling_Price	Present_Price	Kms_Driven	Owner	year_diff	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmissio
0	3.35	1.720979	10.203592	0	6	0	1	0	
1	4.75	2.255493	10.668955	0	7	1	0	0	
2	7.25	2.287471	8.839277	0	3	0	1	0	
3	2.85	1.423108	8.556414	0	9	0	1	0	
4	4.60	1.927164	10.656082	0	6	1	0	0	

2.2.12 Frontend and User Interface

HTML : HTML (Hypertext Markup Language) is the code that is used to structure a web page and its content. For example, content could be structured within a set of paragraphs, a list of bulleted points, or using images and data tables

HTML is a HyperText Markup Language file format used as the basis of a web page. HTML is a file extension used interchangeably with HTM. ... The HTML tags can be used to define headings, paragraphs, lists, links, quotes, and interactive forms. It can also be used to embed Javascript, and CSS (cascading

style sheets).
Any file containing HTML code is saved using the extension ". HTML". All modern browsers -- such as Google Chrome, Safari and Mozilla Firefox -- recognize this format and can open these files, so all you need to do to run an HTML file is open it in your Web browser of choice.
appearance of the document.

HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. Tags such as <img /> and <input /> directly introduce content into the page. Other tags such as <p> surround and provide information about document text and may include other tags as sub-elements. Browsers do not display the HTML tags, but use them to interpret the content of the page.

CSS : Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML
CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility; provide more flexibility and control in the specification of presentation characteristics; enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, which reduces complexity and repetition in the structural content; and enable the .css file to be cached to improve the page load speed between the pages that share the file and its formatting.
Separation of formatting and content also makes it feasible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

The name cascading comes from the specified priority scheme to determine which style rule applies if more than one rule matches a particular element. This cascading priority scheme is predictable.
CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.

A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

BOOTSTRAP
Bootstrap is the most popular HTML, CSS and JavaScript framework for developing a responsive and mobile friendly website.
It is absolutely free to download and use.
It is a front-end framework used for easier and faster web development.
It includes HTML and CSS based design templates for typography, forms,

buttons, tables, navigation, modals, image carousels and many others.
It can also use JavaScript plug-ins.
It facilitates you to create responsive designs.

JQUERY

jQuery is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax
It is free, open-source software using the permissive MIT License. As of May 2019, jQuery is used by 73% of the 10 million most popular websites. Web analysis indicates that it is the most widely deployed JavaScript library by a large margin, having at least 3 to 4 times more usage than any other JavaScript library.

jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications. jQuery also provides capabilities for developers to create plug-ins on top of the JavaScript library. This enables developers to create abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets. The modular approach to the jQuery library allows the creation of powerful dynamic web pages and Web applications.

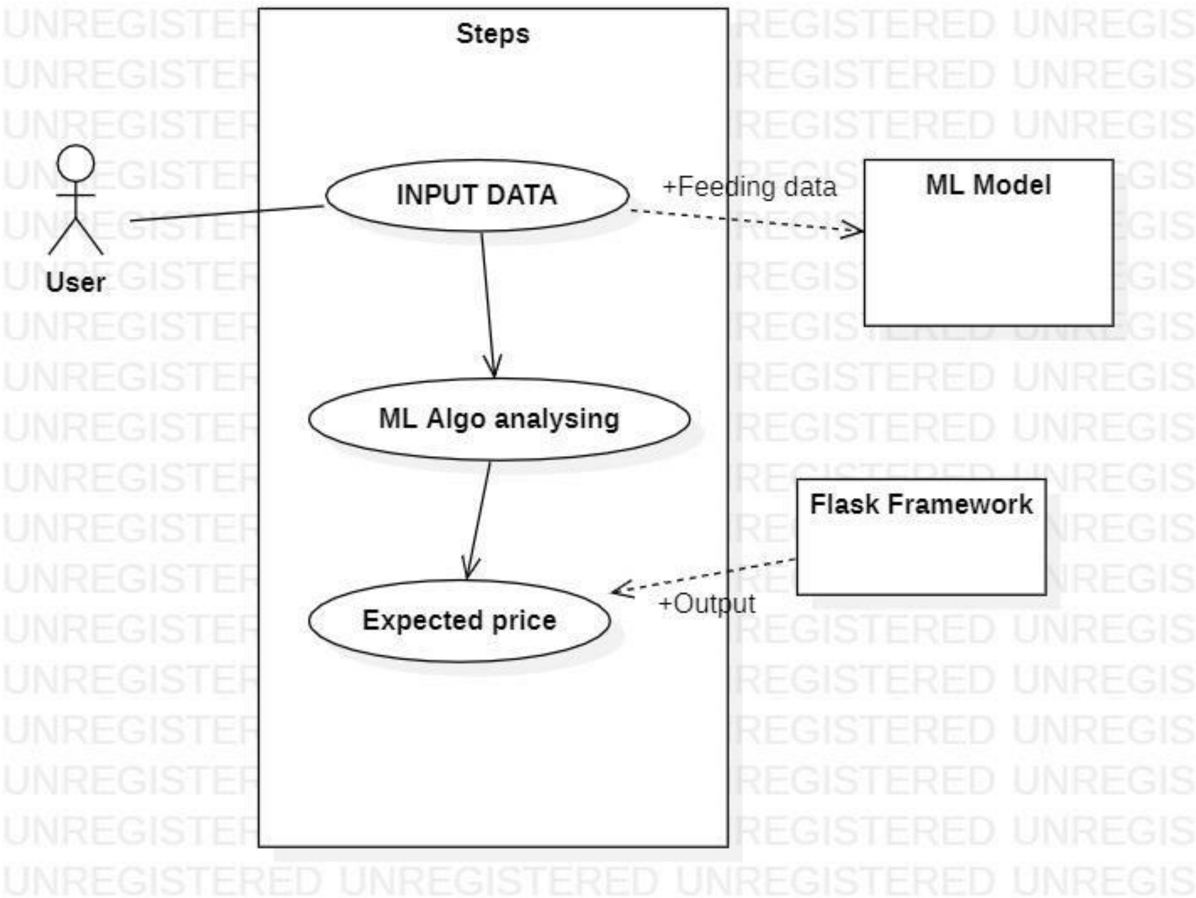
Chapter 3

System Design

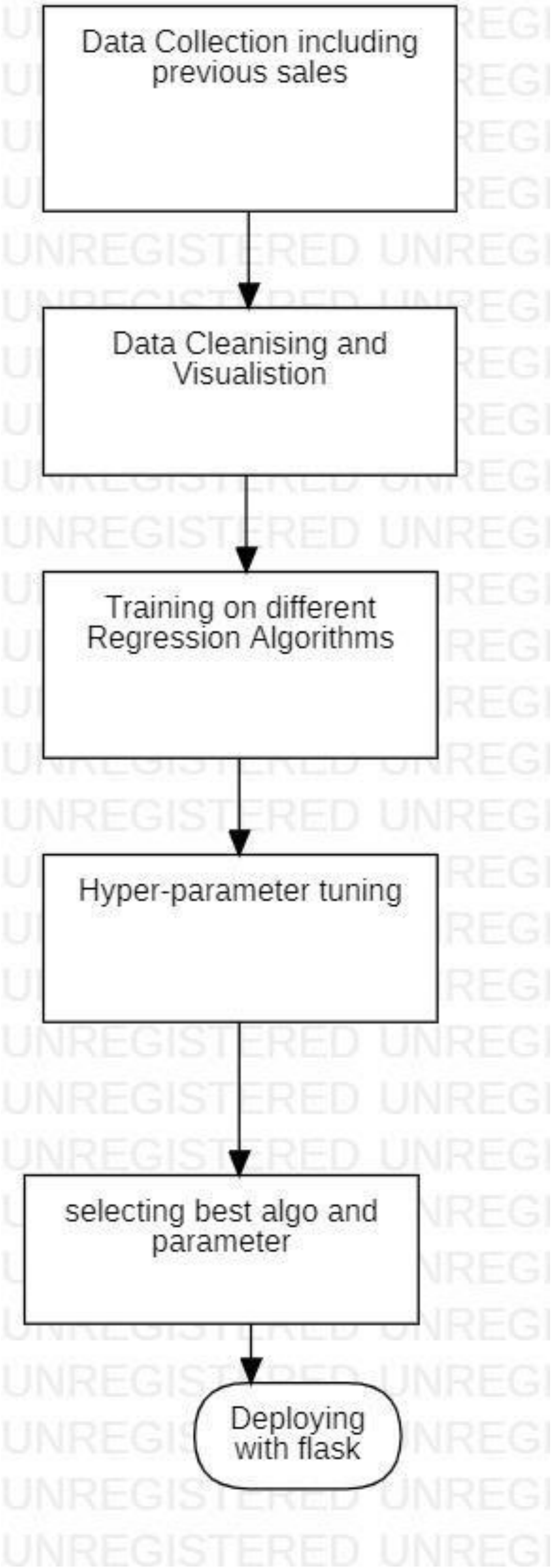
3.1 System Framework

Our solution to above mentioned problem is ‘car sale price predictor’ , a web based application that would enable service sector organizations to automate the review management system and reward the best performers of the month/week and to motivate to those who are under performing .

3.2 USECASE DIAGRAM

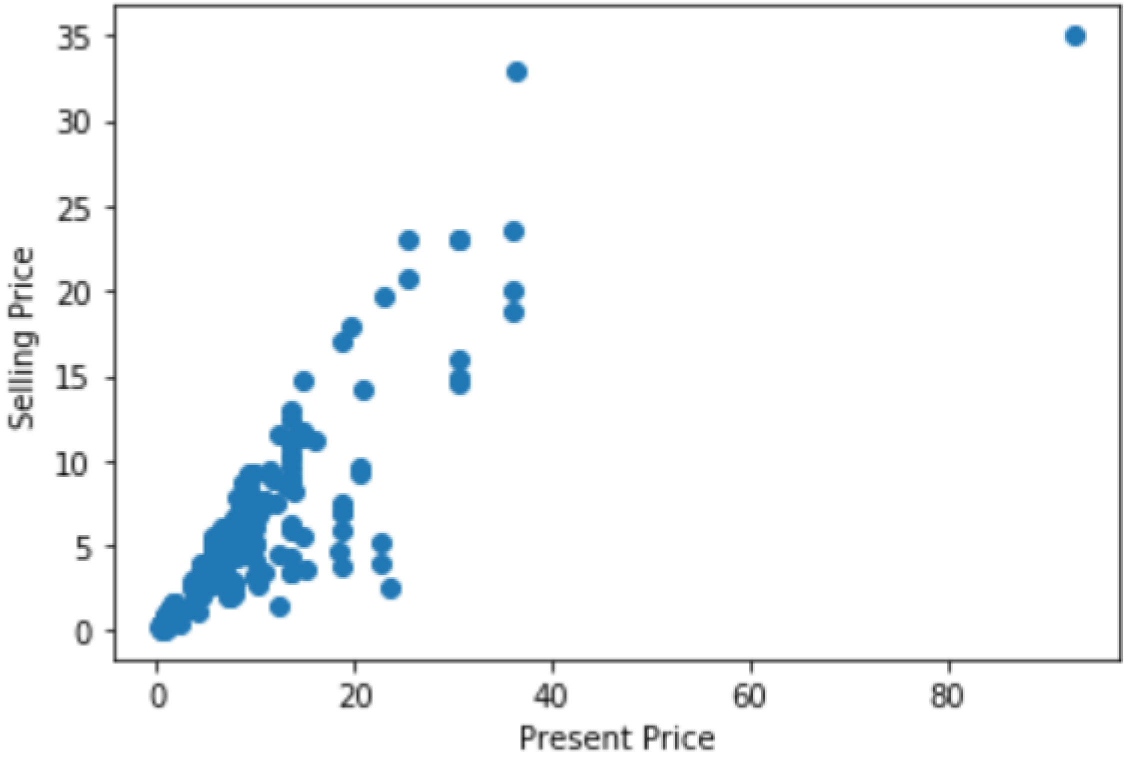


3.3 FLOWCHART FOR ML MODEL

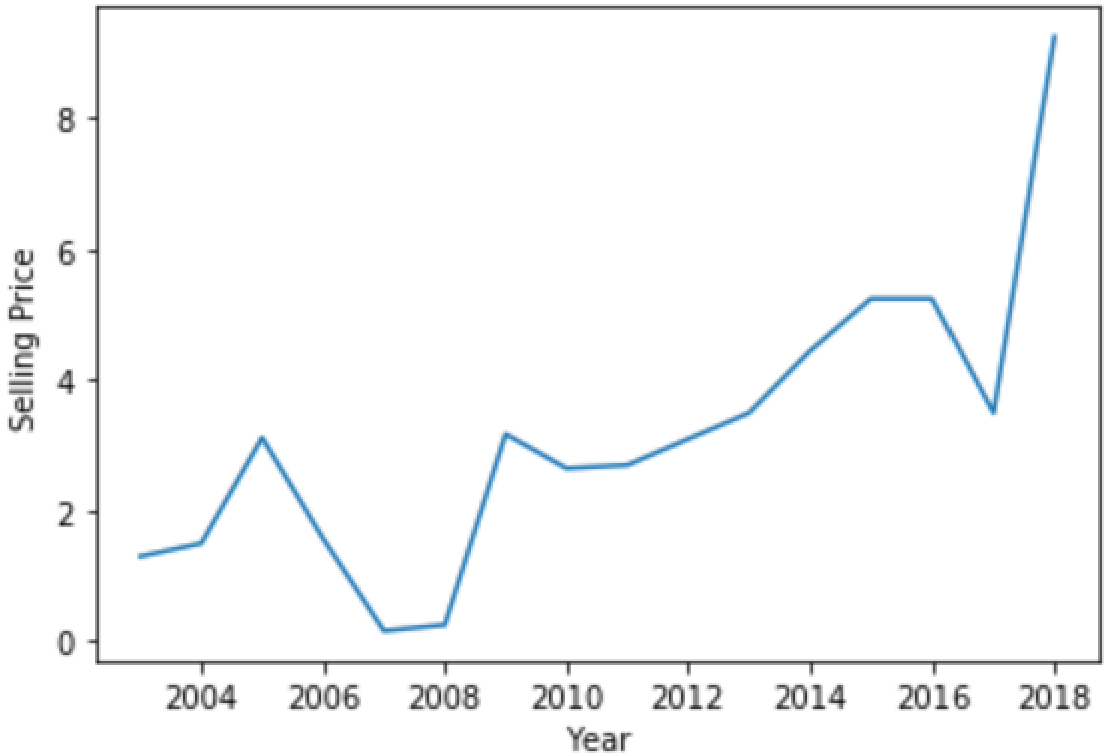


3.4 STATISTICAL ANALYSIS

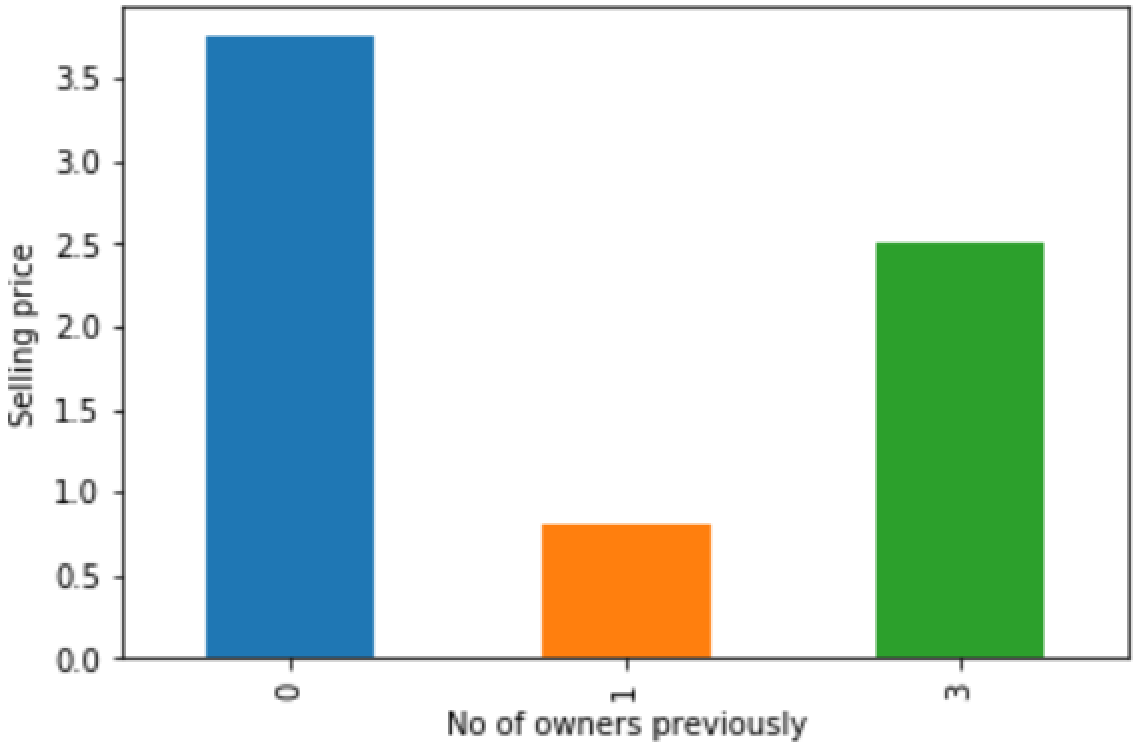
Analysis of Present Price and Selling Price



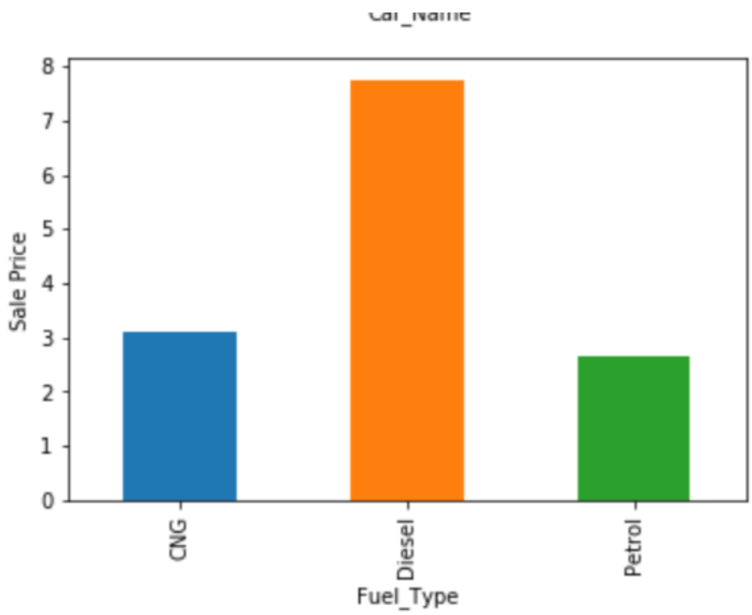
Relation between year and selling price

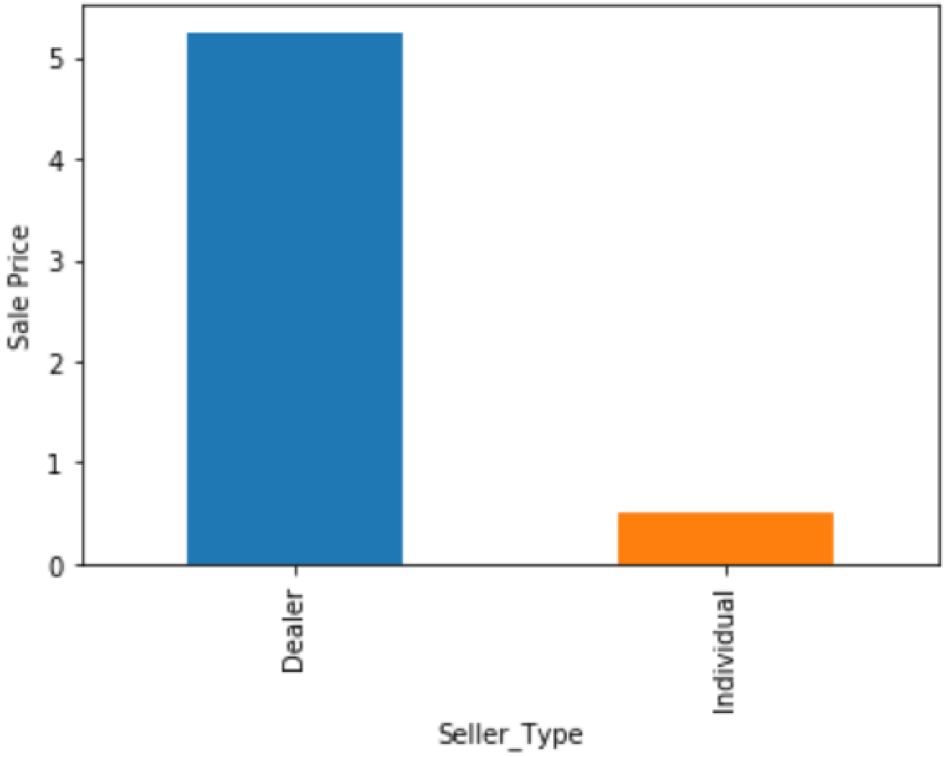


Selling Price v/s previous owners



Fuel Type v/s Pricing





Feature Engineering and Using Algorithm

Code Snippets

```
df = df.drop('Car_Name',axis = 1)
```

```
df.head()
```

	Selling_Price	Present_Price	Kms_Driven	Fuel_Type	Seller_Type	Transmission	Owner	year_diff
0	3.35	1.720979	10.203592	Petrol	Dealer	Manual	0	6
1	4.75	2.255493	10.668955	Diesel	Dealer	Manual	0	7
2	7.25	2.287471	8.839277	Petrol	Dealer	Manual	0	3
3	2.85	1.423108	8.556414	Petrol	Dealer	Manual	0	9
4	4.60	1.927164	10.656082	Diesel	Dealer	Manual	0	6

In [45]:

```
final_dataset.head()
```

Out[45]:

	Selling_Price	Present_Price	Kms_Driven	Owner	year_diff	Fuel_Type_Diesel	Fuel_Type_Petrol	Seller_Type_Individual	Transmission_Manual
0	3.35	1.720979	10.203592	0	6	0	1	0	1
1	4.75	2.255493	10.668955	0	7	1	0	0	1
2	7.25	2.287471	8.839277	0	3	0	1	0	1
3	2.85	1.423108	8.556414	0	9	0	1	0	1
4	4.60	1.927164	10.656082	0	6	1	0	0	1

In [46]:

```
final_dataset.columns
```

Out[46]:

```
Index(['Selling_Price', 'Present_Price', 'Kms_Driven', 'Owner', 'year_diff',
       'Fuel_Type_Diesel', 'Fuel_Type_Petrol', 'Seller_Type_Individual',
       'Transmission_Manual'],
      dtype='object')
```

In [47]:

```
x = final_dataset[['Present_Price', 'Kms_Driven', 'Owner', 'year_diff',
        'Fuel_Type_Diesel', 'Fuel_Type_Petrol', 'Seller_Type_Individual',
        'Transmission_Manual']]
y = final_dataset['Selling_Price']
```

Running Algorithm and getting results

```
from sklearn.ensemble import GradientBoostingRegressor
boost = GradientBoostingRegressor()
boost.fit(x_train,y_train)
```

```
GradientBoostingRegressor(alpha=0.9, criterion='friedman_mse', init=None,
             learning_rate=0.1, loss='ls', max_depth=3, max_features=None,
             max_leaf_nodes=None, min_impurity_decrease=0.0,
             min_impurity_split=None, min_samples_leaf=1,
             min_samples_split=2, min_weight_fraction_leaf=0.0,
             n_estimators=100, n_iter_no_change=None, presort='auto',
             random_state=None, subsample=1.0, tol=0.0001,
             validation_fraction=0.1, verbose=0, warm_start=False)
```

```
boost.score(x_test,y_test)
```

```
0.921426886307747
```

USING FLASK TO DEPLOY ON WEB APP

```python
from flask import Flask , render_template , request , url_for
import numpy as np
import pickle

app = Flask(__name__)
model = pickle.load(open('model.pkl', 'rb'))
@app.route('/')
def home():
        return render_template('home.html')

@app.route('/survey',methods=['POST'])
def main():
    name = request.form['name']
    email = request.form['email']

    return render_template('main.html',name=name,email=email)
```

```python
def predict():
    Fuel_Type_Diesel = 0
    Present_Price = float(request.form['Present_Price'])
    Present_Price = np.log(Present_Price)
    Kms_Driven = int(request.form['Kms_Driven'])
    Kms_Driven = np.log(Kms_Driven)
    owner= int(request.form['Owner'])
    Year = int(request.form['Year'])
    Year = 2020 - Year
    Fuel_Type_Petrol=request.form['Fuel_Type_Petrol']
    if(Fuel_Type_Petrol=='Petrol'):
            Fuel_Type_Petrol=1
            Fuel_Type_Diesel=0
    else:
            Fuel_Type_Petrol=0
            Fuel_Type_Diesel=1

    Seller_Type_Individual=request.form['Seller_Type_Individual']
    if(Seller_Type_Individual=='Individual'):
            Seller_Type_Individual=1
    else:
            Seller_Type_Individual=0

    Transmission_Mannual=request.form['Transmission_Mannual']
    if(Transmission_Mannual=='Mannual'):
            Transmission_Mannual=1
    else:
            Transmission_Mannual=0
```

CHAPTER 4

FUTURE SCOPE

For future work, we can use this regression process to obtain a loss risk class as a first step in improving the performance of our model, and we could try to solve price prediction using deep learning methods to continue comparing our model to even more sophisticated methods, but this would require a larger dataset. We also intend to use regression techniques in the dataset to forecast the size of profit and loss rather than the class (Profit or Loss). There are two well-known prediction difficulties in this field.