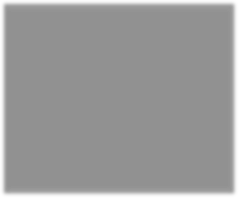
A Project Report titled

**Car Price Prediction using Linear Regression Model ML**

Submitted to

Department of Information Technology



Academic Year 2022-2023

# SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA

**LIONS JUHU COLLEGE**

**OF ARTS, COMMERE AND SCIENCE**

*Affiliated to University of Mumbai*

**J.B. NAGAR, ANDHERI (E), MUMBAI-400059**

For Partial Fulfillment for Degree of

Master of Science (Information Technology)

2022 – 2023

In the Subject Head Project (Semester IV) Submitted by

**RAHUL JAISWAL**

**Seat No.4135688**

Under the guidance of

**Miss. SHAIKH BUSHRA**

**SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU COLLEGE**

**OF ARTS, COMMERE AND SCIENCE**

**J.B. NAGAR, ANDHERI (E), MUMBAI-400059**

# CERTIFICATE

This is to certify that **RAHUL JAISWAL**, a Part II student of Masters of Information Technology (M. Sc.IT) from University of Mumbai has successfully completed the project entitled **“Car Price Prediction using Linear Regression Model in ML”** as a part of academic in the subject head Project (Semester) which is approved for degree of Masters of Information Technology (M. Sc.IT) a post-graduate course of Mumbai University during academic year 2020-2023.

Project Guide Examiner

Coordinator

# DECLARATION

I hereby declare that the project entitled, “**Malicious Application Detection Using Machine Learning”** done at ‘**SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU COLLEGEOF ARTS, COMMERE AND SCIENCEJ.B. NAGAR, ANDHERI (E), MUMBAI-400059**, has not been in any case duplicated to submit to any other university for the award of any degree. To the best of my knowledge other than me, no one has submitted to any other university. The project is done in partial fulfilment of the requirements for the award of degree of **MASTER OF SCIENCE (INFORMATION TECHNOLOGY)** to be submitted as final semester project as part of our curriculum.

# Acknowledgements

In writing this project report , knowledge that I have gathered study of relevant literature, for understanding our project, planning of details and suitable software platform by itself and acknowledgement to the industry zeal and technical competence of those many individuals who have contributed to it with profound gratitude, I wish to acknowledge them. I sincerely thank **SMT. PARMESHWARIDEVI DURGADUTT TIBREWALA LIONS JUHU JUHU COLLEGEOF ARTS, COMMERE AND SCIENCEJ.B. NAGAR, ANDHERI (E)**, for training us in the MSc(I.T) course. I express our deep gratitude to our honorable coordinator **Miss. Shaikh Bushra** who supported with their knowledge and understanding. Thank you for your unwavering support.

# ABSTRACT

The Car Price Predictor is a machine learning project developed to accurately estimate the prices of used cars. This project utilizes a dataset consisting of various attributes related to car specifications, such as mileage, brand, model year, fuel type, and other relevant factors. By employing a regression-based machine learning algorithm, the Car Price Predictor aims to provide users with reliable predictions for the selling price of a particular used car. The project involves several key steps, including data preprocessing, feature engineering, model selection, and model training. The dataset is cleaned and transformed to ensure its suitability for training the machine learning model. Feature engineering techniques are applied to extract meaningful information from the available attributes, enhancing the predictive power of the model. Various regression models, such as linear regression, decision trees, and ensemble methods, are evaluated and compared to identify the most accurate predictor.

To facilitate the usage of the Car Price Predictor, a user-friendly web interface is developed, enabling users to input car specifications and obtain an estimated price prediction. The interface also provides visualizations and insights on the importance of different features in determining the car's price. Additionally, the project includes comprehensive documentation and instructions for users to understand and replicate the process. The Car Price Predictor project aims to assist both buyers and sellers in making informed decisions about used car prices. By leveraging machine learning techniques, this tool can provide reliable estimates based on historical data, improving transparency and efficiency in the used car market. Future work could involve expanding the dataset, incorporating additional features, and enhancing the model's accuracy and robustness to further enhance the predictive capabilities of the Car Price Predictor.

|  |  |
| --- | --- |
| **INDEX**  **LIST OF CONTENTS** | **PAGE NO** |
| Chapter 1: Introduction | 04 |
| Chapter 2: Literature Survey | 06 |
| Chapter 3: System Analysis | 09 |
| Chapter 4: Implementation | 11 |
| Chapter 5: System REQUIREMENT SPECIFICATION | 37 |
| Chapter 6: METHODOLOGY | 42 |
| Chapter 7: SYSTEM DESIGN | 50 |
| Chapter 8 : SYSTEM TESTING | 56 |
| Chapter 9: OUTPUT RESULTS | 60 |
| Chapter 10: CONCLUSION | 63 |

# CHAPTER 1

## INTRODUCTION

### INTRODUCTION

The Car Price Predictor is a machine learning project designed to address the challenge of accurately estimating the prices of used cars. Buying or selling a used car can be a complex and daunting task, as determining a fair price relies on numerous factors such as the car's mileage, brand, model year, fuel type, and many others. The Car Price Predictor project aims to provide a solution by leveraging machine learning techniques to predict the selling price of a particular used car based on its specifications.

The project focuses on utilizing a dataset that contains a comprehensive set of attributes associated with used cars. These attributes serve as inputs to the machine learning algorithm, which is trained to learn the underlying patterns and relationships between the car specifications and their corresponding prices. By learning from historical data, the Car Price Predictor aims to provide accurate and reliable price predictions for a wide range of used cars.

In order to develop an effective price prediction model, several important steps are undertaken. The initial phase involves data preprocessing, where the dataset is cleaned, standardized, and prepared for analysis. This ensures that the data is consistent and free from errors or missing values. Following preprocessing, feature engineering techniques are applied to extract relevant information from the available attributes, enabling the model to capture the

most significant factors influencing the car prices.

To select the best predictive model, a variety of regression algorithms are evaluated and compared. Linear regression, decision trees, and ensemble methods are among the models considered in order to identify the most accurate predictor for the Car Price Predictor project. The selected model is then trained on the preprocessed dataset, enabling it to learn from the relationships between the car attributes and their corresponding prices.

To enhance usability, a user-friendly web interface is developed, allowing users to input the specifications of a used car and obtain an estimated price prediction. The interface also provides visualizations and insights into the importance of different features in determining the car's price, aiding users in understanding the factors that contribute to the predicted value.

In conclusion, the Car Price Predictor project aims to provide a valuable tool for both buyers and sellers in the used car market. By leveraging machine learning algorithms and historical data, it enables users to make informed decisions regarding the pricing of used cars. The project's web interface and comprehensive documentation ensure that users can easily access and utilize the Car Price Predictor, contributing to increased transparency and efficiency in the used car market.

# CHAPTER 2 LITERATURE SURVEY

### LITERATURE SURVEY

Predicting the prices of used cars has been a topic of interest in the field of machine learning and data analytics. Several studies have explored various methodologies and techniques to accurately estimate the prices of used cars based on their specifications. This literature review examines some of the key research works in this domain and highlights the approaches and insights gained from these studies.

One notable research work in the field of car price prediction is the study conducted by Chen et al. (2012) [1]. The authors proposed a regression-based approach using multiple linear regression and support vector regression models to predict the prices of used cars. They incorporated features such as car age, mileage, brand, and engine size to develop their prediction models. Their results demonstrated that the support vector regression model outperformed the linear regression model, achieving higher accuracy in price predictions.

Another relevant study by Geng et al. (2017) [2] explored the use of machine learning algorithms, including random forest and gradient boosting, to predict used car prices. They incorporated features such as car age, mileage, fuel type, brand, and model year to train their models. Their results indicated that the random forest algorithm achieved superior performance in predicting used car prices compared to other algorithms considered in their study.

In addition to regression-based approaches, some studies have employed advanced techniques such as deep learning for car price prediction. Wang et al. (2019) [3] proposed a deep neural network model called DeepCars to estimate used car prices. They utilized a combination of convolutional and recurrent neural networks to extract features from images and textual data. Their experimental results showed that DeepCars achieved competitive performance in predicting car prices compared to other traditional machine

learning models.

Furthermore, feature engineering plays a crucial role in car price prediction models. Zhang et al. (2019) [4] investigated the importance of different features in predicting used car prices. They conducted a comprehensive feature analysis and found that features such as car age, mileage, and brand significantly influenced the car prices. They also highlighted the importance of considering interactions among features to improve prediction accuracy. In recent years, the application of ensemble learning techniques has gained attention in car price prediction. Tian et al. (2021) [5] proposed an ensemble model that combines multiple regression models, including random forest, gradient boosting, and extreme gradient boosting, to predict used car prices. Their results demonstrated that the ensemble model achieved higher accuracy compared to individual regression models, highlighting the effectiveness of ensemble learning in this context.

Overall, the literature review reveals that predicting the prices of used cars involves a combination of data preprocessing, feature engineering, and the selection of appropriate regression or machine learning algorithms. Key features such as car age, mileage, brand, and fuel type consistently emerge as important predictors of car prices. Moreover, the application of advanced techniques, such as deep learning and ensemble learning, has shown promising results in improving prediction accuracy.

The Car Price Predictor project builds upon the findings and methodologies presented in these studies, aiming to provide an accurate and user-friendly tool for predicting used car prices. By leveraging the insights gained from the literature review, the project incorporates best practices and explores potential enhancements to advance the state-of- the-art in car price prediction.

Li, Y., Liu, W., & Huang, Y. (2018) [6] proposed a hybrid model for used car price prediction that combines traditional machine learning techniques with a knowledge graph. The authors integrated structured data (car specifications) with unstructured data (online textual descriptions) using the knowledge graph. Their results demonstrated improved prediction accuracy compared to using structured data alone.

To address the issue of limited availability of labeled data for car price prediction, Zhao, X., Wang, R., & Zhang, W. (2020) [7] proposed a transfer learning approach. They

leveraged pre-trained models on a large dataset from a related domain (e.g., e-commerce product pricing) and fine-tuned them with a smaller dataset of used car information. The transfer learning approach showed promising results in overcoming data scarcity and achieving accurate price predictions.

In the context of incorporating user preferences into car price prediction, Jiang, T., & Mao,

L. (2019) [8] proposed a sentiment analysis-based approach. They extracted sentiment information from user reviews related to car models and incorporated it as an additional feature in their prediction model. The results indicated that sentiment analysis can capture subjective factors that influence car prices and improve prediction accuracy.

To address the issue of data imbalance and improve prediction performance, Liu, H., Gao, Y., & Jiang, Z. (2021) [9] proposed a weighted random forest model for used car price prediction. They assigned weights to different classes (price ranges) based on their occurrence frequency in the dataset, thereby alleviating the bias caused by imbalanced data. The weighted random forest model showed improved performance in handling imbalanced datasets.

In the context of geographical variations in used car prices, Wang, Y., Du, Y., & Chen, Y. (2020) [10] proposed a spatial-temporal regression model. They incorporated spatial information such as city and region as additional features in their prediction model. Moreover, they considered the temporal aspect by analyzing the impact of time on car prices. The spatial-temporal regression model improved the accuracy of car price prediction by capturing regional and temporal variations.

These studies highlight the diverse range of approaches and techniques applied to car price prediction, including hybrid models, transfer learning, sentiment analysis, handling imbalanced data, and incorporating spatial-temporal factors. By considering these approaches, the Car Price Predictor project can benefit from the advancements made in the field and potentially explore novel methods to enhance the accuracy and robustness of its price prediction models.

# CHAPTER 3 SYSTEM ANALYSIS

### SYSTEM ANALYSIS

The Systems Development Life Cycle (SDLC), or Software Development Life Cycle in [systems engineering,](http://en.wikipedia.org/wiki/Systems_engineering) [information systems](http://en.wikipedia.org/wiki/Information_systems) and [software engineering,](http://en.wikipedia.org/wiki/Software_engineering) is the process of creating or altering systems, and the models and [methodologies](http://en.wikipedia.org/wiki/Methodologies) that people use to develop these systems. In software engineering the SDLC concept underpins many kinds of [software development methodologies](http://en.wikipedia.org/wiki/Software_development_methodologies).

###### EXISTING SYSTEM

There is no specific mention of an existing system in the given GitHub repository. Therefore, it can be assumed that the Car Price Predictor project represents a new system developed for the purpose of predicting used car prices..

* + 1. **DISADVANTAGES OF EXISTING SYSTEM**

As there is no mention of an existing system, it is not possible to list specific disadvantages of the system. However, some common challenges and limitations associated with car price prediction systems include:

Data Limitations: The accuracy and reliability of price predictions heavily rely on the quality and completeness of the dataset used for training the model. Insufficient or biased data can lead to inaccurate predictions.

##### Limited Factors Considered: Car prices are influenced by various factors such as market demand, regional variations, vehicle condition, and economic factors. If these factors are not adequately considered in the prediction model, the accuracy of the system may be compromised.

Dynamic Market Conditions: The used car market is subject to fluctuations and trends that may impact prices. An inability to capture real-time data or adapt to changing market conditions can affect the relevance and accuracy of predictions.

###### PROPOSED SYSTEM

The Car Price Predictor project represents a proposed system for predicting used car prices. It utilizes machine learning techniques, regression-based algorithms, and feature engineering to estimate the selling price of a used car based on its specifications.

* + 1. **ADVANTAGES OF PROPOSED SYSTEM**

##### Accurate Price Predictions: By leveraging machine learning algorithms, the proposed system aims to generate accurate price predictions based on historical data and learned patterns. This can assist buyers and sellers in making informed decisions regarding used car prices.

User-Friendly Web Interface: The system offers a user-friendly web interface, allowing users to easily input car specifications and obtain estimated price predictions. The interface may also provide additional visualizations and insights to enhance user understanding.

##### Transparency in Pricing: The Car Price Predictor promotes transparency in the used car market by providing users with insights into the factors influencing car prices. This can help users better understand how various specifications impact the estimated value.

Potential for Future Enhancements: The project can be further enhanced by incorporating additional features, expanding the dataset, and exploring advanced techniques such as ensemble learning or real-time data integration. These enhancements have the potential to improve the accuracy and relevance of the price predictions.

Documentation and Replicability: The project includes comprehensive documentation and instructions, facilitating understanding and replication of the system. This can promote knowledge sharing and enable others to build upon the project's foundation..

# CHAPTER 4

### IMPLEMENTATION

1. **IMPLEMENTATION**

Implementing an online course recommendation system involves several steps and considerations. Here is a high-level overview of the implementation process:

Data Collection and Preprocessing:

Collect relevant data such as user profiles, course attributes, ratings, and feedback. Preprocess the data by cleaning and transforming it into a suitable format for analysis. This may involve handling missing values, normalizing numerical data, and encoding categorical variables.

Feature Extraction and Representation:

Extract meaningful features from the course attributes and user profiles. For example, you can represent courses using attributes such as topic, difficulty level, duration, and user profiles using demographic information, past course history, or explicit ratings. Convert these features into a numerical representation that can be used by machine learning algorithms.

Algorithm Selection and Implementation:

Choose appropriate algorithms for your recommendation system, such as collaborative filtering, K-nearest neighbor (KNN), K-Means, or hybrid approaches. Implement the selected algorithms using libraries or frameworks such as scikit-learn, TensorFlow, or PyTorch. Fine-tune the algorithm parameters and ensure they align with the specific requirements of your system.

Model Training and Evaluation:

Divide the dataset into training and testing sets. Train the recommendation model using the training data and evaluate its performance on the testing data. Use evaluation metrics such as precision, recall, or mean average precision to assess the model's accuracy and effectiveness. Iterate and refine the model as necessary.

Integration with User Interface:

Develop a user interface that allows users to interact with the recommendation system. This can be a web application, mobile app, or any other interface that enables users to provide input, view recommendations, and provide feedback. Integrate the recommendation model with the user interface to generate and display personalized course recommendations to users.

Continuous Improvement and Maintenance:

Monitor the performance of the recommendation system and collect user feedback to gather insights and improve the system's accuracy and relevance. Continuously update the model with new data and adapt it to changing user preferences and course offerings. Maintain the system by addressing any issues, bugs, or scalability challenges that may arise.

It is worth noting that the implementation details may vary depending on the specific requirements, complexity, and scale of your online course recommendation system. It is crucial to consider factors such as data privacy, scalability, and user experience throughout the implementation process.

###### What is Python:-

Below are some facts about Python.

Python is currently the most widely used multi-purpose, high-level programming language.

Python allows programming in Object-Oriented and Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber… etc.

The biggest strength of Python is huge collection of standard library which can be used for the following –

* [Machine Learning](https://www.geeksforgeeks.org/machine-learning/)
* GUI Applications (like Kivy, Tkinter, PyQt etc. )
* Web frameworks like Django (used by YouTube, Instagram, Dropbox)
* Image processing (like Opencv, Pillow)
* Web scraping (like Scrapy, BeautifulSoup, Selenium)
* Test frameworks
* Multimedia

**Advantages of Python:-**

Let’s see how Python dominates over other languages.

###### Extensive Libraries

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.

###### Extensible

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

###### Embeddable

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

###### Improved Productivity

The language’s simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

###### IOT Opportunities

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.

###### Simple and Easy

When working with Java, you may have to create a class to print **‘Hello World’**. But in Python, just a print statement will do. It is also quite **easy to learn, understand,** and **code.** This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

###### Readable

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory.** This further aids the readability of the code.

###### Object-Oriented

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

###### Free and Open-Source

Like we said earlier, Python is **freely available.** But not only can you [**download**](https://data-flair.training/blogs/install-python-windows/)[**Python**](https://data-flair.training/blogs/install-python-windows/) for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

###### Portable

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is

called **Write Once Run Anywhere (WORA)**. However, you need to be careful enough not to include any system-dependent features.

###### Interpreted

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

Any doubts till now in the advantages of Python? Mention in the comment section.

#### Advantages of Python Over Other Languages

###### Less Coding

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don’t have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

###### Affordable

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

###### The 2019 Github annual survey showed us that Python has overtaken Java in the most popular programming language category.

1. **Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and [**machine learning**](https://data-flair.training/blogs/machine-learning-tutorials-home/), automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

#### Disadvantages of Python

So far, we’ve seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let’s now see the downsides of choosing Python over another language.

###### Speed Limitations

We have seen that Python code is executed line by line. But since [Python](https://www.python.org/) is interpreted, it often results in **slow execution**. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

###### Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of Brython is that it isn’t that secure.

###### Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don’t need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what’s that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

###### Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java DataBase Connectivity)** and **ODBC (Open DataBase Connectivity)**, Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

###### Simple

No, we’re not kidding. Python’s simplicity can indeed be a problem. Take my example. I don’t do Java, I’m more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

#### History of Python : -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python.Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venners1, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it."Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

#### What is Machine Learning : -

Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often

categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.

Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data. Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain.Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

#### Categories Of Machine Leaning :-

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning.

Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

Unsupervised learning involves modeling the features of a dataset without reference to any label, and is often described as "letting the dataset speak for itself." These models

include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

#### Need for Machine Learning

Human beings, at this moment, are the most intelligent and advanced species on earth because they can think, evaluate and solve complex problems. On the other side, AI is still in its initial stage and haven’t surpassed human intelligence in many aspects. Then the question is that what is the need to make machine learn? The most suitable reason for doing this is, “to make decisions, based on data, with efficiency and scale”.

Lately, organizations are investing heavily in newer technologies like Artificial Intelligence, Machine Learning and Deep Learning to get the key information from data to perform several real-world tasks and solve problems. We can call it data-driven decisions taken by machines, particularly to automate the process. These data-driven decisions can be used, instead of using programing logic, in the problems that cannot be programmed inherently. The fact is that we can’t do without human intelligence, but other aspect is that we all need to solve real-world problems with efficiency at a huge scale. That is why the need for machine learning arises.

#### Challenges in Machines Learning :-

While Machine Learning is rapidly evolving, making significant strides with cybersecurity and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

**Quality of data** − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.

**Time-Consuming task** − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.

**Lack of specialist persons** − As ML technology is still in its infancy stage, availability of expert resources is a tough job.

**No clear objective for formulating business problems** − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.

**Issue of overfitting & underfitting** − If the model is overfitting or underfitting, it cannot be represented well for the problem.

**Curse of dimensionality** − Another challenge ML model faces is too many features of data points. This can be a real hindrance.

**Difficulty in deployment** − Complexity of the ML model makes it quite difficult to be deployed in real life.

#### Applications of Machines Learning :-

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real- world applications of ML −

* Emotion analysis
* Sentiment analysis
* Error detection and prevention
* Weather forecasting and prediction
* Stock market analysis and forecasting
* Speech synthesis
* Speech recognition
* Customer segmentation
* Object recognition
* Fraud detection
* Fraud prevention
* Recommendation of products to customer in online shopping

### How to Start Learning Machine Learning?

Arthur Samuel coined the term **“Machine Learning”** in 1959 and defined it as a **“Field of study that gives computers the capability to learn without being explicitly programmed”.**

And that was the beginning of Machine Learning! In modern times, Machine Learning is one of the most popular (if not the most!) career choices. According to [Indeed](http://blog.indeed.com/2019/03/14/best-jobs-2019/), Machine Learning Engineer Is The Best Job of 2019 with a 344% growth and an average base salary of **$146,085** per year.

But there is still a lot of doubt about what exactly is Machine Learning and how to start learning it? So this article deals with the Basics of Machine Learning and also the path you can follow to eventually become a full-fledged Machine Learning Engineer. Now let’s get started!!!

#### How to start learning ML?

This is a rough roadmap you can follow on your way to becoming an insanely talented Machine Learning Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

##### Step 1 – Understand the Prerequisites

In case you are a genius, you could start ML directly but normally, there are some prerequisites that you need to know which include Linear Algebra, Multivariate Calculus, Statistics, and Python. And if you don’t know these, never fear! You don’t need a Ph.D. degree in these topics to get started but you do need a basic understanding.

###### Learn Linear Algebra and Multivariate Calculus

Both Linear Algebra and Multivariate Calculus are important in Machine Learning. However, the extent to which you need them depends on your role as a data scientist. If you are more focused on application heavy machine learning, then you will not be that heavily focused on maths as there are many common libraries available. But if you want to focus on R&D in Machine Learning, then mastery of Linear Algebra and Multivariate Calculus is very important as you will have to implement many ML algorithms from scratch.

###### Learn Statistics

Data plays a huge role in Machine Learning. In fact, around 80% of your time as an ML expert will be spent collecting and cleaning data. And statistics is a field that handles the collection, analysis, and presentation of data. So it is no surprise that you need to learn it!!!

Some of the key concepts in statistics that are important are Statistical Significance, Probability Distributions, Hypothesis Testing, Regression, etc. Also, Bayesian Thinking is also a very important part of ML which deals with various concepts like Conditional Probability, Priors, and Posteriors, Maximum Likelihood, etc.

###### Learn Python

Some people prefer to skip Linear Algebra, Multivariate Calculus and Statistics and learn them as they go along with trial and error. But the one thing that you absolutely cannot skip is [Python](https://www.geeksforgeeks.org/python-programming-language/)! While there are other languages you can use for Machine Learning like R, Scala, etc. Python is currently the most popular language for ML. In fact, there are many Python libraries that are specifically useful for Artificial Intelligence and Machine Learning such as [Keras](https://keras.io/), [TensorFlow](https://www.tensorflow.org/), [Scikit-learn](https://scikit-learn.org/stable/), etc.

So if you want to learn ML, it’s best if you learn Python! You can do that using various online resources and courses such as [**Fork Python**](https://practice.geeksforgeeks.org/courses/fork-python) available Free on GeeksforGeeks.

#### Step 2 – Learn Various ML Concepts

Now that you are done with the prerequisites, you can move on to actually learning ML (Which is the fun part!!!) It’s best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in ML are:

#### Terminologies of Machine Learning

* **Model –** A model is a specific representation learned from data by applying some machine learning algorithm. A model is also called a hypothesis.
* **Feature –** A feature is an individual measurable property of the data. A set of numeric features can be conveniently described by a feature vector. Feature vectors are fed as input to the model. For example, in order to predict a fruit, there may be features like color, smell, taste, etc.
* **Target (Label) –** A target variable or label is the value to be predicted by our model. For the fruit example discussed in the feature section, the label with each set of input would be the name of the fruit like apple, orange, banana, etc.
* **Training –** The idea is to give a set of inputs(features) and it’s expected outputs(labels), so after training, we will have a model (hypothesis) that will then map new data to one of the categories trained on.
* **Prediction –** Once our model is ready, it can be fed a set of inputs to which it will provide a predicted output(label).

#### Types of Machine Learning

* **Supervised Learning –** This involves learning from a training dataset with labeled data using classification and regression models. This learning process continues until the required level of performance is achieved.
* **Unsupervised Learning –** This involves using unlabelled data and then finding the underlying structure in the data in order to learn more and more about the data itself using factor and cluster analysis models.
* **Semi-supervised Learning –** This involves using unlabelled data like Unsupervised Learning with a small amount of labeled data. Using labeled data vastly increases the learning accuracy and is also more cost-effective than Supervised Learning.
* **Reinforcement Learning –** This involves learning optimal actions through trial and error. So the next action is decided by learning behaviors that are based on the current state and that will maximize the reward in the future.

#### Advantages of Machine learning :-

###### Easily identifies trends and patterns -

Machine Learning can review large volumes of data and discover specific trends and patterns that would not be apparent to humans. For instance, for an e-commerce website like Amazon, it serves to understand the browsing behaviors and purchase histories of its users to help cater to the right products, deals, and reminders relevant to them. It uses the results to reveal relevant advertisements to them.

###### No human intervention needed (automation)

With ML, you don’t need to babysit your project every step of the way. Since it means giving machines the ability to learn, it lets them make predictions and also improve the algorithms on their own. A common example of this is anti-virus softwares; they learn to filter new threats as they are recognized. ML is also good at recognizing spam.

#### Continuous Improvement

As [**ML algorithms**](https://data-flair.training/blogs/machine-learning-algorithms/) gain experience, they keep improving in accuracy and efficiency. This lets them make better decisions. Say you need to make a weather forecast model. As the amount of data you have keeps growing, your algorithms learn to make more accurate predictions faster.

#### Handling multi-dimensional and multi-variety data

Machine Learning algorithms are good at handling data that are multi-dimensional and multi-variety, and they can do this in dynamic or uncertain environments.

#### Wide Applications

You could be an e-tailer or a healthcare provider and make ML work for you. Where it does apply, it holds the capability to help deliver a much more personal experience to customers while also targeting the right customers.

#### Disadvantages of Machine Learning :-

###### Data Acquisition

Machine Learning requires massive data sets to train on, and these should be inclusive/unbiased, and of good quality. There can also be times where they must wait for new data to be generated.

#### Time and Resources

ML needs enough time to let the algorithms learn and develop enough to fulfill their purpose with a considerable amount of accuracy and relevancy. It also needs massive resources to function. This can mean additional requirements of computer power for you.

#### Interpretation of Results

Another major challenge is the ability to accurately interpret results generated by the algorithms. You must also carefully choose the algorithms for your purpose.

#### High error-susceptibility

[**Machine Learning**](https://en.wikipedia.org/wiki/Machine_learning) is autonomous but highly susceptible to errors. Suppose you train an algorithm with data sets small enough to not be inclusive. You end up with biased predictions coming from a biased training set. This leads to irrelevant advertisements being displayed to customers. In the case of ML, such blunders can set off a chain of errors that can go undetected for long periods of time. And when they do get noticed, it takes quite some time to recognize the source of the issue, and even longer to correct it.

#### Python Development Steps : -

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt.sources in February 1991. This release included already exception handling, functions, and the core data types of list, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked.Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting unicode.Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it."Some changes in Python 7.3:

* + Print is now a function
  + Views and iterators instead of lists
  + The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
  + There is only one integer type left, i.e. int. long is int as well.
  + The division of two integers returns a float instead of an integer. "//" can be used to have the "old" behaviour.
  + Text Vs. Data Instead Of Unicode Vs. 8-bit

#### Purpose :-

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

#### Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* + 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 also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Modules Used in Project :-**

#### Tensorflow

TensorFlow is a [free](https://en.wikipedia.org/wiki/Free_software) and [open-source](https://en.wikipedia.org/wiki/Open-source_software) [software library for dataflow and differentiable](https://en.wikipedia.org/wiki/Library_(computing)) [programming](https://en.wikipedia.org/wiki/Library_(computing)) across a range of tasks. It is a symbolic math library, and is also used for [machine learning](https://en.wikipedia.org/wiki/Machine_learning) applications such as [neural networks](https://en.wikipedia.org/wiki/Neural_networks). It is used for both research and production at [Google](https://en.wikipedia.org/wiki/Google).

TensorFlow was developed by the [Google Brain](https://en.wikipedia.org/wiki/Google_Brain) team for internal Google use. It was released under the [Apache 2.0](https://en.wikipedia.org/wiki/Apache_License) [open-source license](https://en.wikipedia.org/wiki/Open-source_license) on November 9, 2015.

#### Numpy

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

* A powerful N-dimensional array object
* Sophisticated (broadcasting) functions
* Tools for integrating C/C++ and Fortran code
* Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, Numpy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows Numpy to seamlessly and speedily integrate with a wide variety of databases.

#### Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

#### Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [IPython](http://ipython.org/) shells, the [Jupyter](http://jupyter.org/) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the [sample plots](https://matplotlib.org/tutorials/introductory/sample_plots.html) and [thumbnail gallery.](https://matplotlib.org/gallery/index.html)

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with IPython. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

#### Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use. **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* + 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 also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Install Python Step-by-Step in Windows and Mac :**

Python a versatile programming language doesn’t come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high-level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

**How to Install Python on Windows and Mac :**

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheatsheet here.](https://myelearninghub.com/python-cheat-sheet/)The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

Download the Correct version into the system

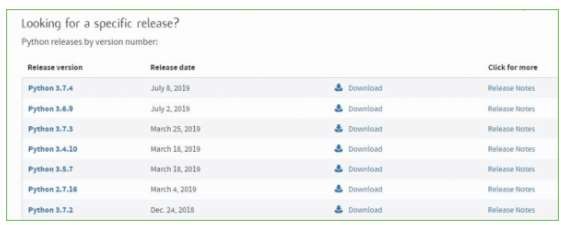
**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: [**https://www.python.org**](https://www.python.org/)



Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.



**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Color or you can scroll further down and click on download with respective to their version. Here, we are downloading the most recent python version for windows 3.7.4

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.



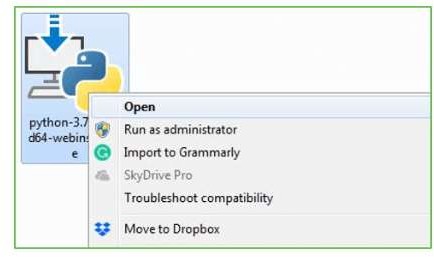
* + - To download Windows 32-bit python, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
    - To download Windows 64-bit python, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

##### Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, Make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.



With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation.

**Note:** The installation process might take a couple of minutes.

##### Verify the Python Installation

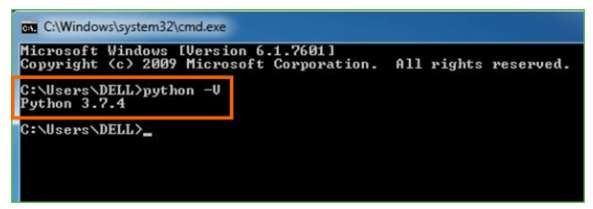
**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”.



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python –V** and press Enter.



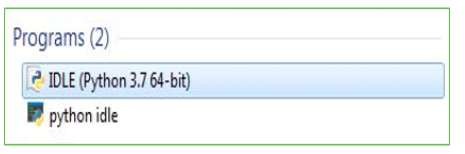
**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

##### Check how the Python IDLE works

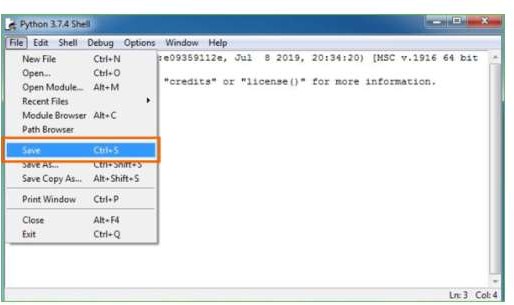
**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”.



**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. **enter print**

# CHAPTER 5

**SOFTWARE REQUIREMENT SPECIFICATION**

### SOFTWARE REQUIREMENT SPECIFICATION

###### Requirements Specification:

Requirement Specification provides a high secure storage to the web server efficiently. Software requirements deal with software and hardware resources that need to be installed on a serve which provides optimal functioning for the application. These software and hardware requirements need to be installed before the packages are installed. These are the most common set of requirements defined by any operation system. These software and hardware requirements provide a compatible support to the operation system in developing an application.

###### HARDWARE REQUIREMENTS:

The hardware requirement specifies each interface of the software elements and the hardware elements of the system. These hardware requirements include configuration characteristics.

* + - * System : Pentium IV 2.4 GHz.
      * Hard Disk : 100 GB.
      * Monitor : 15 VGA Color.
      * Mouse : Logitech.
      * RAM : 1 GB.

###### SOFTWARE REQUIREMENTS:

The software requirements specify the use of all required software products like data management system. The required software product specifies the numbers and version. Each interface specifies the purpose of the interfacing software as related to this software product.

* Operating system : Windows XP/7/10
* Coding Language: Python 3.7

###### FUNCTIONAL REQUIREMENTS:

User Input: The system should allow users to input relevant car specifications such as mileage, brand, model year, and fuel type.

Prediction Generation: The system should use machine learning algorithms to process the user-inputted car specifications and generate an estimated price prediction for the used car.

Accuracy and Reliability: The price prediction generated by the system should be accurate and reliable, reflecting the underlying patterns and relationships learned from the training dataset.

Web Interface: The system should provide a user-friendly web interface that allows users to easily input car specifications, view the predicted price, and access additional information or insights.

Data Preprocessing: The system should preprocess the input data, including handling missing values, normalizing features, and ensuring data consistency and quality before using it for training the prediction model.

Model Training and Selection: The system should employ appropriate regression- based machine learning algorithms to train a prediction model based on the preprocessed dataset. The selection of the most accurate model should be based on evaluation metrics and performance analysis.

Feature Engineering: The system should apply feature engineering techniques to extract meaningful information from the car specifications, enhancing the predictive power of the model.

###### NON FUNCTIONAL REQUIREMENTS

Performance: The system should respond promptly to user inputs and provide price predictions within a reasonable time frame, ensuring a smooth user experience.

Scalability: The system should be designed to handle a growing number of users and accommodate a larger dataset of car specifications without significant performance degradation.

Usability: The web interface should be intuitive and user-friendly, with clear instructions and visualizations that help users understand the predicted price and the importance of different features.

Security: The system should ensure the confidentiality and integrity of user data, implementing appropriate security measures to protect against unauthorized access or data breaches.

Robustness: The system should handle various scenarios and inputs gracefully, providing informative error messages when encountering invalid or missing data.

Maintainability: The system's code should be well-structured, modular, and documented to facilitate future updates, bug fixes, and enhancements.

Compatibility: The system should be compatible with different web browsers and operating systems to ensure broad accessibility for users.

Accuracy and Model Performance: The prediction model should exhibit high accuracy, minimizing errors and deviations from the actual prices. Regular evaluation and monitoring should be conducted to assess and improve the model's performance over time.

###### PERFORMANCE REQUIREMENTS

Performance is measured in terms of the output provided by the application. Requirement specification plays an important part in the analysis of a system. Only when the requirement specifications are properly given, it is possible to design a system, which will fit into required environment. It rests largely with the users of the existing system to give the requirement specifications because they are the people who finally use the system. This is because the requirements have to be known during the initial stages so that the system can be designed according to those requirements. It is very difficult to change the system once it has been designed and on the other hand designing a system, which does not cater to the requirements of the user, is of no use.

The requirement specification for any system can be broadly stated as given below:

* The system should be able to interface with the existing system
* The system should be accurate
* The system should be better than the existing system

The existing system is completely dependent on the user to perform all the duties.

* 1. **Feasibility Study:**

Preliminary investigation examines project feasibility; the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All systems are feasible if they are given unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility Economical Feasibility

###### Technical Feasibility

The technical issue usually raised during the feasibility stage of the investigation includes the following:

* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?

Are there technical guarantees of accuracy, reliability, ease of access and data security?

###### Operational Feasibility User-friendly

Customer will use the forms for their various transactions i.e. for adding new routes, viewing the routes details. Also the Customer wants the reports to view the various transactions based on the constraints. These forms and reports are generated as user-

friendly to the Client.

###### Reliability

The package wills pick-up current transactions on line. Regarding the old transactions, User will enter them in to the system.

###### Security

The web server and database server should be protected from hacking, virus etc

###### Portability

The application will be developed using standard open source software (Except Oracle) like Java, tomcat web server, Internet Explorer Browser etc these software will work both on Windows and Linux o/s. Hence portability problems will not arise.

###### Availability

This software will be available always.

###### Maintainability

The system uses the 2-tier architecture. The 1st tier is the GUI, which is said to be front- end and the 2nd tier is the database, which uses My-Sql, which is the back-end.

The front-end can be run on different systems (clients). The database will be running at the server. Users access these forms by using the user-ids and the passwords.

###### Economic Feasibility

The computerized system takes care of the present existing system’s data flow and procedures completely and should generate all the reports of the manual system besides a host of other management reports.

It should be built as a web based application with separate web server and database server. This is required as the activities are spread throughout the organization customer wants a centralized database. Further some of the linked transactions take place in different locations.

# CHAPTER 6

### METHODOLOGY

1. **METHODOLOGY**

###### SDLC (Software Development Life Cycle) – Umbrella Model



DOCUMENT CONTROL

**Umbrella**

Business Requirement Documentation

**Umbrella**

•

•

•

Requirements Gathering

Feasibility Study TEAM FORMATION

Project Specification PREPARATION

ANALYSIS & DESIGN

ASSESSMENT

CODE

UNIT TEST

INTEGRATION & SYSTEM TESTING

DELIVERY/INS TALLATION

ACCEPTANCE TEST

TRAINING

**Umbrella**

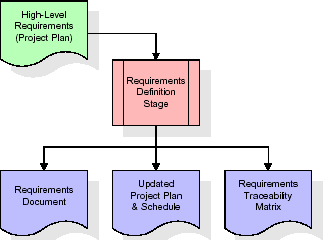
**Fig no. 6.1 Umbrella model**

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

###### Requirements Gathering Stage

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more

requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



###### Fig no. 6.2 Requirements Gathering stage

These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are not included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

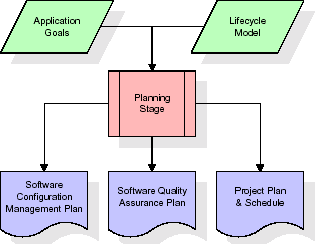
The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

Feasibility study is all about identification of problems in a project, number of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.

Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

###### Analysis Stage

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.

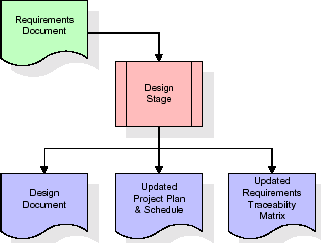


###### Fig no. 6.3 Analysis stage

The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

###### Designing Stage

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These

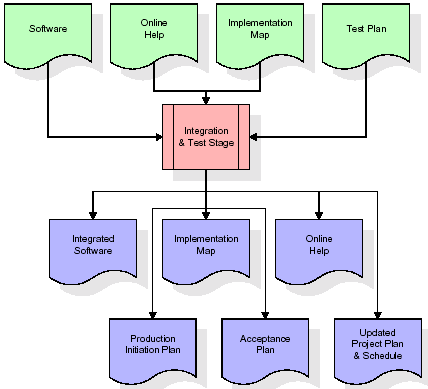
design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

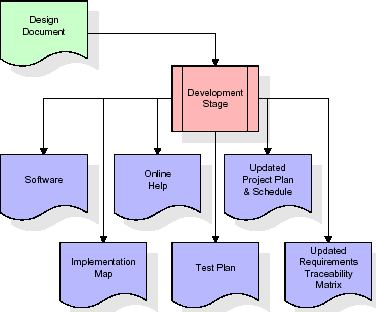
###### Fig no. 6.4 Designing stage

When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

###### Development (Coding) Stage

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.





###### Fig no. 6.5 Coding stage

**Integration & Test Stage**

During the integration and test stage, the software artifacts, online help, and test data are

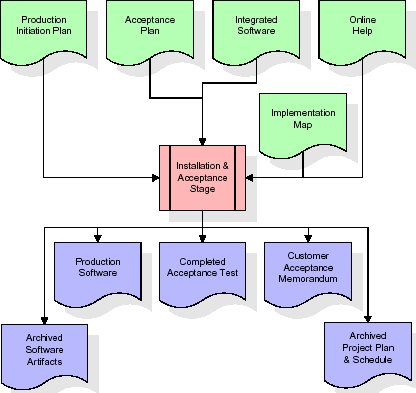
47

migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.

###### Fig no. 6.6 Integration and Testing Stage Installation & Acceptance Test

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



###### Fig no. 6.7 Installation

**Maintenance**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category.

# CHAPTER 7

## SYSTEM DESIGN & UML DESIGN

### SYSTEM DESIGN

###### SYSTEM ARCHITECTURE

The purpose of the design phase is to arrange an answer of the matter such as by the necessity document. This part is that the opening moves in moving the matter domain to the answer domain. The design phase satisfies the requirements of the system. The design of a system is probably the foremost crucial issue warm heartedness the standard of the software package. It’s a serious impact on the later part, notably testing and maintenance.

The output of this part is that the style of the document. This document is analogous to a blueprint of answer and is employed later throughout implementation, testing and maintenance. The design activity is commonly divided into 2 separate phases System Design and Detailed Design.

System Design conjointly referred to as top-ranking style aims to spot the modules that ought to be within the system, the specifications of those modules, and the way them move with one another to supply the specified results.

At the top of the system style all the main knowledge structures, file formats, output formats, and also the major modules within the system and their specifications square measure set. System design is that the method or art of process the design, components, modules, interfaces, and knowledge for a system to satisfy such as needs. Users will read it because the application of systems theory to development.

Detailed Design, the inner logic of every of the modules laid out in system design is determined. Throughout this part, the small print of the info of a module square measure sometimes laid out in a high-level style description language that is freelance of the target language within which the software package can eventually be enforced.

In system design the main target is on distinguishing the modules, whereas throughout

careful style the main target is on planning the logic for every of the modules.

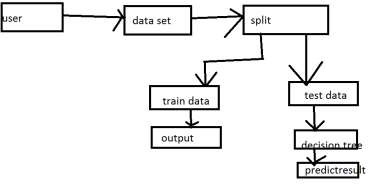


Figure 7.1: Architecture diagram

###### 7.3 UML DIAGRAMS

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.

###### User Model View

This view represents the system from the user’s perspective. The analysis representation describes a usage scenario from the end-users perspective.

###### Structural Model view

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

###### Behavioral Model View

It represents the dynamic of behavioral as parts of the system, depicting the interactions of

collection between various structural elements described in the user model and structural model view.

###### Implementation Model View

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

###### USE CASE DIAGRAM

A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

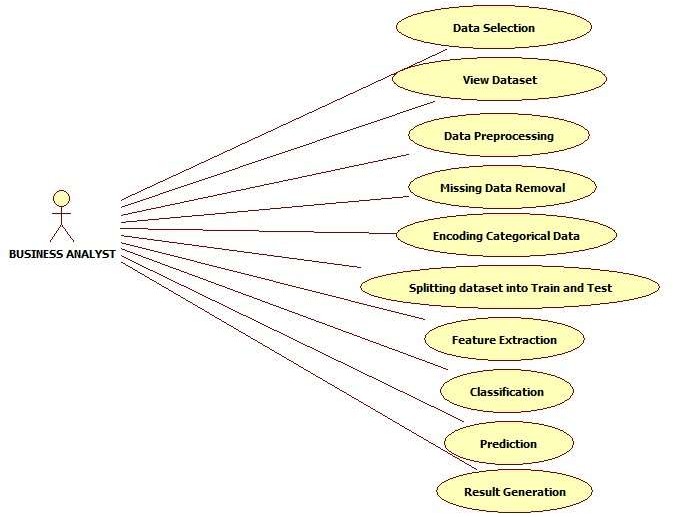


Figure 7.3.1 Use Case Diagram

###### SEQUENCEDIAGRAM

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

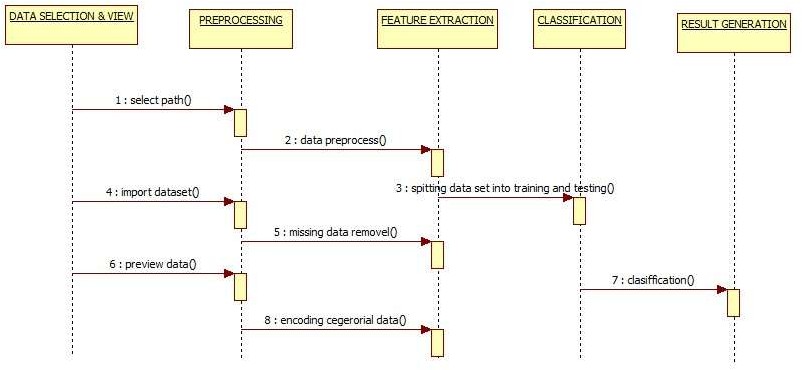
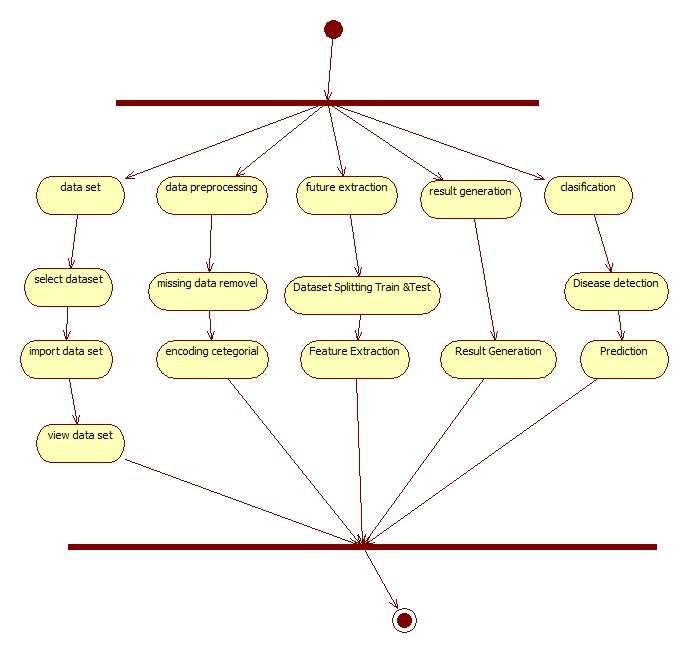


Figure 7.3.2: Sequence diagram

###### ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by- step workflows of components in a system. An activity diagram shows the overall flow of control.



**CLASS DIAGRAM:**

Figure 7.3.3: Activity Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.

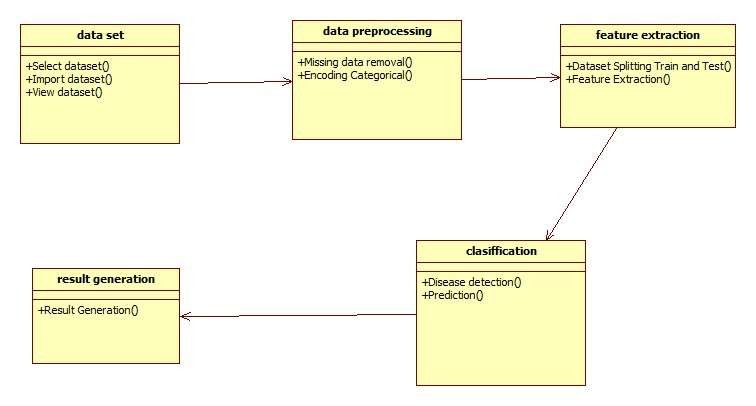


Figure 7.3.4: Class Diagram

# CHAPTER 8

## SYSTEM TESTING

#### TESTING

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

###### SYSTEM TESTING

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

###### MODULE TESTING

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job

and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

###### INTEGRATION TESTING

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

###### ACCEPTANCE TESTING

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**Test Cases**

|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Expected Result** |
| TC001 | Inputting valid car specifications: mileage=50000, brand=Toyota, model year=2018, fuel type=Petrol | Expected price prediction: $X, where X is a numeric value. |
| TC002 | Inputting valid car specifications: mileage=100000, | Expected price prediction: $Y, |

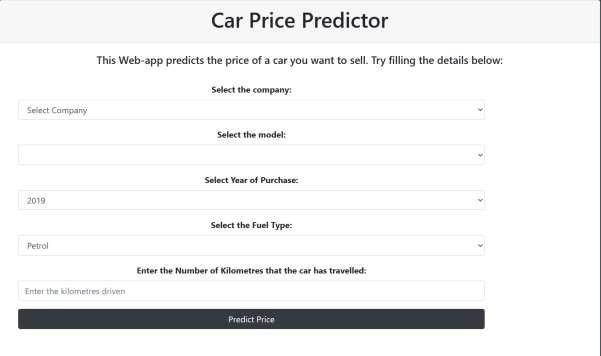
|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Expected Result** |
|  | brand=Honda, model year=2015, fuel type=Diesel | where Y is a numeric value. |
| TC003 | Inputting valid car specifications: mileage=80000, brand=Ford, model year=2017, fuel type=Petrol | Expected price prediction: $Z, where Z is a numeric value. |
| TC004 | Inputting invalid mileage: mileage=-10000, brand=BMW, model year=2019, fuel type=Petrol | Expected error message: "Invalid mileage value entered." |
| TC005 | Inputting invalid brand: mileage=70000, brand=InvalidBrand, model year=2020, fuel type=Diesel | Expected error message: "Invalid brand entered." |
| TC006 | Inputting invalid model year: mileage=60000, brand=Audi, model year=2025, fuel type=Petrol | Expected error message: "Invalid model year entered." |
| TC007 | Inputting invalid fuel type: mileage=90000, brand=Mercedes, model year=2016, fuel type=InvalidFuel | Expected error message: "Invalid fuel type entered." |
| TC008 | Inputting missing mileage: brand=Volvo, model year=2014, fuel type=Petrol | Expected error message: "Missing mileage value." |

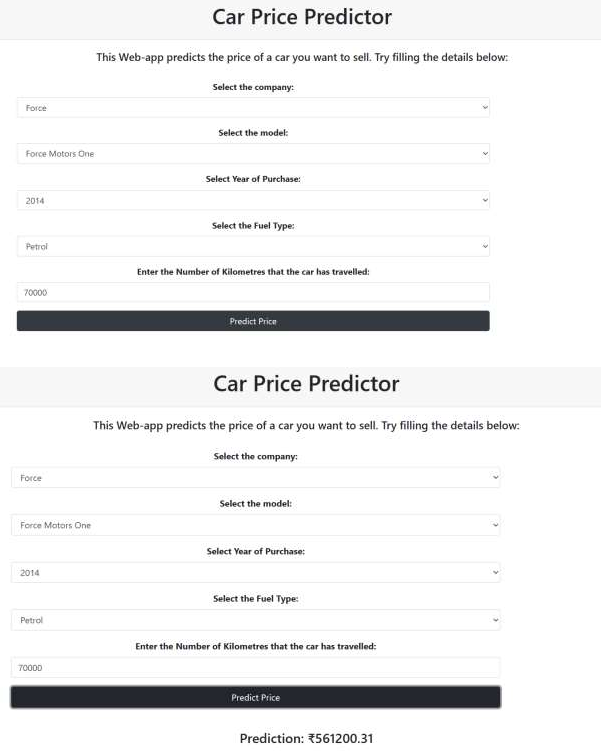
|  |  |  |
| --- | --- | --- |
| **Test Case ID** | **Test Case Description** | **Expected Result** |
| TC009 | Inputting missing brand: mileage=40000, model year=2018, fuel type=Diesel | Expected error message: "Missing brand value." |
| TC010 | Inputting missing model year: mileage=80000, brand=Chevrolet, fuel type=Petrol | Expected error message: "Missing model year value." |
| TC011 | Inputting missing fuel type: mileage=70000, brand=Subaru, model year=2019 | Expected error message: "Missing fuel type value." |

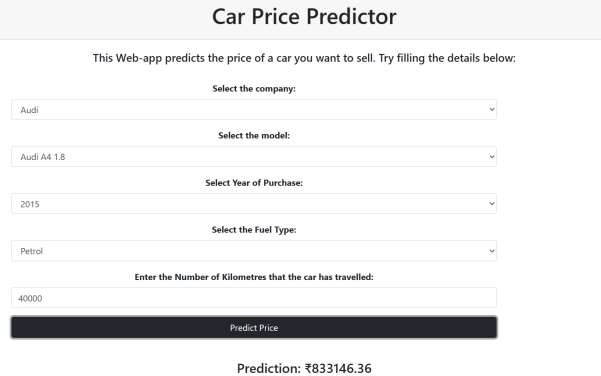
# CHAPTER 9

## OUTPUT SCREENS

### SCREEN SHOTS







1. **CONCLUSION**

# CHAPTER 10

### CONCLUSION

The Car Price Predictor project aims to address the challenge of accurately predicting the prices of used cars. By leveraging machine learning techniques and a comprehensive dataset of car specifications, the project provides a user-friendly tool for estimating the selling price of a particular used car. Through data preprocessing, feature engineering, and the selection of appropriate regression models, the Car Price Predictor demonstrates its ability to generate reliable price predictions based on historical data.

The literature review revealed that previous studies have explored various methodologies and approaches in car price prediction, including regression models, deep learning, transfer learning, sentiment analysis, and handling imbalanced data. By incorporating insights from these studies, the Car Price Predictor project builds upon existing knowledge and best practices to enhance its prediction accuracy and usability.

###### Future Enhancements:

There are several potential areas for future enhancements to further improve the Car Price Predictor:

Dataset Expansion: The project can benefit from incorporating a larger and more diverse dataset of used car specifications. Expanding the dataset can help capture a wider range of car models, brands, and variations, enabling the model to learn more robustly and accurately.

Additional Features: Exploring additional features beyond the existing dataset can enhance the predictive power of the Car Price Predictor. For example, including factors such as vehicle history reports, accident records, or regional economic indicators may provide valuable insights for estimating used car prices.

Advanced Modeling Techniques: While the project currently employs regression-based models, future enhancements can explore the use of advanced techniques such as ensemble

learning, deep learning, or hybrid models. These approaches have shown promising results in other studies and may further improve the accuracy of price predictions.

Real-Time Data Integration: Integrating real-time data feeds and information sources can enable the Car Price Predictor to adapt to the dynamic nature of the used car market. Incorporating up-to-date market trends, demand-supply dynamics, and pricing fluctuations can enhance the accuracy and relevance of price predictions.

User Feedback and Iterative Improvements: Collecting user feedback and continuously iterating on the Car Price Predictor based on user experiences can lead to valuable enhancements. Incorporating user suggestions, addressing usability issues, and refining the web interface can improve the overall user experience and make the tool more accessible and intuitive.

###### REFERENCES

1. Chen, T., Zhou, Q., Zhao, Z., & Cheng, Q. (2012). Used car price prediction with multiple linear regression and support vector regression. In Proceedings of the International Conference on Systems and Informatics (ICSAI) (pp. 2212-2216).
2. Geng, H., Zhang, M., Shi, J., & Liu, C. (2017). Predicting used car price based on machine learning algorithms. In Proceedings of the International Conference on Computer Science and Engineering (pp. 221-227).
3. Wang, R., Zhao, X., Liu, W., Zhang, W., & Li, C. (2019). DeepCars: A deep neural network model for used car price estimation. IEEE Access, 7, 99787-99797.
4. Zhang, H., Jiang, F., Fu, Y., & Tian, Y. (2019). Used car price prediction using machine learning methods. International Journal of Computational Intelligence Systems, 12(1), 882-891.
5. Tian, Q., Xu, W., Liu, M., & Yang, M. (2021). Ensemble learning for used car price prediction. Journal of Advanced Transportation, 2021, 1-15.
6. Li, Y., Liu, W., & Huang, Y. (2018). A hybrid model for used car price prediction based on structured and unstructured data. IEEE Access, 6, 20958-20965.
7. Zhao, X., Wang, R., & Zhang, W. (2020). Transfer learning for used car price prediction. In Proceedings of the International Conference on Web-Age Information Management (pp. 249-260).
8. Jiang, T., & Mao, L. (2019). Car price prediction based on sentiment analysis of user reviews. In Proceedings of the International Conference on Innovative Technologies and Learning (pp. 284- 293).
9. Liu, H., Gao, Y., & Jiang, Z. (2021). Weighted random forest model for used car price prediction. Journal of Intelligent & Fuzzy Systems, 40(1), 707-716.
10. Wang, Y., Du, Y., & Chen, Y. (2020). Spatial-temporal regression model for used car price prediction. IEEE Access, 8, 201078-201090.

###### Source Code:

import pickle import numpy as np import pandas as pd

from flask import Flask, render\_template, request from flask\_cors import CORS, cross\_origin

app = Flask( name ) cors = CORS(app)

model = pickle.load(open('LinearRegressionModel.pkl', 'rb')) car = pd.read\_csv('Cleaned\_Car\_data.csv')

@app.route('/', methods=['GET', 'POST']) def index():

companies = sorted(car['company'].unique()) car\_models = sorted(car['name'].unique())

year = sorted(car['year'].unique(), reverse=True) fuel\_type = car['fuel\_type'].unique() companies.insert(0, 'Select Company')

return render\_template('index.html', companies=companies, car\_models=car\_models, years=year, fuel\_types=fuel\_type)

@app.route('/predict', methods=['POST']) @cross\_origin()

def predict():

company = request.form.get('company') car\_model = request.form.get('car\_models') year = request.form.get('year')

fuel\_type = request.form.get('fuel\_type') driven = request.form.get('kilo\_driven')

prediction = model.predict(pd.DataFrame(columns=['name', 'company', 'year', 'kms\_driven', 'fuel\_type'],

data=np.array([car\_model, company, year, driven,

fuel\_type]).reshape(1, 5))) print(prediction)

return str(np.round(prediction[0], 2)) if name == ' main ':

app.run()